

**ЎЗБЕКИСТОН РЕСПУБЛИКАСИ ОЛИЙ ВА ЎРТА МАХСУС ТАЪЛИМ
ВАЗИРЛИГИ**

**МИРЗО УЛУҒБЕК НОМИДАГИ ЎЗБЕКИСТОН МИЛЛИЙ
УНИВЕРСИТЕТИ**

МАТЕМАТИКА ФАКУЛЬТЕТИ

Р.Д.АЛОЕВ, Д.Э.НЕЪМАТОВА

**БЕРИЛГАНЛАР БАЗАЛАРИ ТИЗИМЛАРИДА ЗАМОНАВИЙ
ТЕХНОЛОГИЯЛАРИ
фанидан
МАЪРУЗАЛАР МАТНИ**

Тошкент-2018

Ушбу маърузалар матни Oracle берилганлар базасини бошқариш тизимини илк бор ўрганувчиларига мўлжалланган бўлиб, унда SQL\Oracle ва PL/SQL ORACLE элементлари ва ORACLE концепцияси қисқача баён этилади. Ушбу маърузалар матнидан, Ахборот тизимларининг математик ва дастурий таъминоти йўналиши ва мутахассислиги бўйича таҳсил олаётган талабаларга мўлжалланган “Берилганлар базалари тизимларида замонавий технологиялар” фани учун таянч ўқув қўлланма сифатида фойдаланиш мумкин.

SQL ORACLE

§1. ORACLE/SQL тилига кириш.

ORACLE/SQL тили (Structured Query Language - Тузилмаланган сўровлар тили) - бу шундай буйруқлар тўпламики, улар ёрдамида Oracle тизимидаги берилганлар базасига кириб берилганлар устида ҳар хил амаллар бажариш мумкиндир. Кўп ҳолларда Oracle тизимидаги амалий дастурлар ва ускунавий тизимлар ёрдамида юқорида айтилган SQL буйруқларидан фойдаланмасдан туриб берилганлар базасига кириш имконияти туғилади. Аммо, лекин ушбу амалий дастурлар ва ускунавий тизимларнинг ўзи, ўз навбатида SQL буйруқларидан фойдаланиб фойдаланувчиларнинг сўровларини бажаради. Ушбу бандда биз реляцион берилганлар базасига муносиб бўлган асосий маълумотлар билан танишамиз.

Бу бандда қуйидаги саволлар муҳокама қилинади:

- SQL тарихи;
- SQL стандартлари;
- SQL ижобий томонлари;
- Ички қурилган SQL;
- ORACLE/SQL - даги ёзиш қоидалари;
- ORACLE ускунавий воситаларида SQL дан фойдаланиш.

1.1. SQL тарихи.

Реляцион модел биринчи мартаба Э.Ф.Кодд томонидан 1970 йилларининг бошида ишлаб чиқилган. SQL тилининг биринчи версияси реляцион берилганлар базасини бошқариш System R номли тизим прототипи учун IBM фирмаси томонидан 1970 йилларининг ўрталарида яратилган. SQL тилининг ушбу версияси (SEQUEL2) IBM фирмасининг IBM Journal of R&D журналининг 1976 й. ноябридаги сонидида chop этилган. 1979 йилда эса ORACLE корпорацияси SQL тилининг тижорат вариантини тавсия этди. Бундан ташқари SQL тили IBM фирмаси ишлаб чиққан DB2 ва SQL/DS берилганлар базасини бошқариш учун ҳам ишлатилган эди. Ҳозирги кунларда SQL тили кенг ишлатилади ва берилганлар базаси билан ишлашга рухсат берадиган стандарт тил сифатида тан олинган.

1.2. SQL стандартлари

Американинг стандартлар институти (ANSI) томонидан SQL тилининг реляцион берилганлар базасини бошқариш учун янги стандартни (ANSI X3.135 - 1989 << Database Language SQL with Integrity Enhancement >> хужжати ёки ANSI SQL 89) қабул қилди. Бу стандарт унинг олдинги ANSI X3. - 135 - 1986 стандартини алмаштиради. Ушбу стандарт яна бир қатор қуйидаги ташкилотларда қабул қилинди:

- International Standard Organization (ISO), ISO 9075 - 1989 << Database Language SQL with Integrity Enhancement >> хужжати асосида;

- United States Federal Government, Federal Information Processing Standard Publication (FIPS PUB) 127 хужжати асосида.

Американинг стандартлар институти ички қурилган SQL тили стандартини ҳам ANSI X3. 168 – 1989 <<Database Language Embedded SQL >> хужжати асосида қабул қилди.

Реляцион берилганлар базасини бошқарувчи барча асосий тизимлар SQL тилининг у ёки бу вариантларини қўллаб қувватлайди ва иложи борида ANSI SQL 89 стандартига мослашишга ҳаракат қилади.

1.3. SQL тилининг ижобий томонлари.

Ушбу бандостида SQL тилининг шуҳрат қозонганлигининг асосий сабаблари ёритилади. SQL тили ҳар хил тоифадаги фойдаланувчилар учун, шу жумладан амалий дастурчилар, берилганлар базасининг администраторлари, менеджерлар ва охириги фойдаланувчилар учун қатор кучли воситаларни тавсия этади.

SQL процедурасиз тил.

SQL тили процедурасиз тил бўлиб унда

- бир вақтнинг ўзида фақат бир қатор устида ишлов бермасдан, балки қаторлар тўплами устида амаллар бажариш мумкин;
- автоматик навигацияни таъминлайди.

Ҳар хил тоифадаги фойдаланувчилар учун тил.

SQL тилини берилганлар базаси устида ҳар хил тоифадаги фойдаланувчилар ҳар хил турдаги амалларни бажариш учун фойдаланишлари мумкин:

- тизимий администраторлар;
- берилганлар базасининг администраторлари;
- берилганларни ҳимоя қилувчи тизимнинг администраторлари;
- амалий дастурчилар;
- охириги фойдаланувчиларнинг барча турлари.

SQL командалари жуда тез ўрганилади.

Универсал тил.

SQL командалари ёрдамида турли хил масалаларни ҳал қилиш мумкин:

- берилганларни чиқариб олиш;
- жадвалларга янги қаторларни қўшиш, янгилаш, ўчириш;
- объектларни яратиш, ўринларини алмаштириш, йўқотиш, модификация қилиш;
- берилганлар базасидаги объектларга киришга ҳуқуқ беришни бошқариш;

Барча реляцион берилганлар базаси учун умумий тил.

Барча реляцион берилганлар базасини бошқариш тизимларида SQL тили ишлайди.

1.4. Ички қурилган SQL

Ички қурилган SQL бирор процедурали тилда ёзилган программаларда SQL командаларидан фойдаланишга имкон беради. Ички қурилган SQL командаларини икки гуруҳга бўлиш мумкин:

- SELECT ва INSERT-га ўхшаган интерактив фойдаланиладиган SQL командалари;
- процедурали тилда тузилган программа командалари билан ахборот алмашишни бошқариш командалари: PREPARE, OPEN ва ҳоказо.

Ички қурилган SQL - ни ORACLE тизимининг

- Pro * Ada;
- Pro * C;
- Pro * COBOL;
- Pro * FORTRAN;
- Pro * PASCAL;
- Pro * PL/I.

предкомпиляторларида ишлатиш мумкин.

Ички қурилган SQL терминлари.

- * *:host_variable* - база ўзгарувчиси , процедурали тил қоидаларига асосан эълон қилинади. Улар SQL командаларида фойдаланилади.
- * *CURSOR* - курсор идентификатори;
- * *DB_NAME* - берилганлар базасининг номи;
- * *DB_String* - SQL*Net даги берилганлар базасининг идентификатори;
- * *Statement_name* - SQL операторининг идентификатори;
- * *block_name* - PL/SQL блокининг идентификатори ва ҳоказо.

1.5. Лексик келишувлар.

Oracle тизимида SQL операторини киритишда бир ёки ундан кўп табуляция белгилари, қаторни ўтказиш, бўш жой ёки изоҳларни қўшиш мумкин. Масалан ORACLE қуйидаги иккита командаларни бир хил қабул қилади:

- ◆ `SELECT ENAME, SAL * 12, MONTH_BETWEEN (HIREDATE, SYSDATE) FROM EMP`
Ва
- ◆ `SELECT ENAME,
SAL * 12,
MONTH_BETWEEN (HIREDATE, SYSDATE)
FROM EMP`

Бунда командалар номлари, идентификатор ва параметрлар номларида катта ёки кичик ҳарфларнинг ишлатилиши фарқланмайди. Аммо берилганлар ичида катта ва кичик ҳарфлар фарқланади.

2-§. SQL элементлари

Бу бандда биз қуйидаги SQL элементлари ҳақида қисқача ахборот берамиз:

- схема объектлари;
- объектларни номлаш қоидалари,
- объектларга мурожаат қилиш қоидалари ва тавсиялар;
- литераллар;
- берилганлар турлари;
- бўш қийматлар (null - қийматлар)
- псевдоустунлар;
- изоҳлар.

Энди биз ушбу элементлар ҳақида алоҳида-алоҳида сўз юритамиз:

2.1. Схема объектлари

Схема бу берилганларнинг мантиқий тузилмавий мажмуаси бўлиб у фойдаланувчига тегишли бўлади ва унинг номи фойдаланувчи номи билан номланади. Схема объектларини қуйидаги турларга бўлиш мумкин:

- кластерлар;
 - боғланиш каналлари;
 - автоном сақланадиган функциялар *;
 - индекслар;
 - пакетлар *;
 - автоном сақланадиган процедуралар ;
 - кетма кетликлар;
 - оний нусхалар *;
 - нусха баёнотлири *;
 - синонимлар;
 - жадваллар;
 - триггерлар *;
 - кўринишлар.
- * - белгиси орқали белгиланган объектлар фақат процедурали режимга эга бўлган ORACLE-да ишлатилади;
 - | - белгиси орқали белгиланган объектлар эса тақсимланган режимга эга бўлган ORACLE-да ишлатилади.

Берилганлар базасида схемага тегишли бўлмаган қуйидаги объектлар ҳам мавжуд:

- профиллар;
- роллар;
- қайтариш сегментлари;
- жадвалли фазолар.

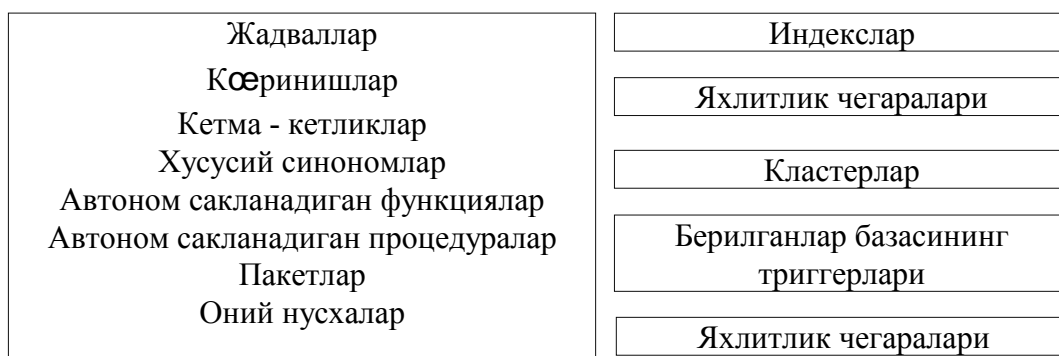
2.2. Объектларни номлаш ва унинг қисмлари.

Объект номлари 1 байтдан 30 байтгача бўлган белгилар кетма кетлиги ёрдамида белгилинади. Берилганлар базасининг номи бундан мустасно бўлиб у 8 байтдан ошмаслиги лозим. Боғланиш каналларининг номи эса 128 байтгача бўлиши мумкин.

Номларда катта ва кичик ҳарфлар фарқланмайди ва улар берилганлар базасидаги белгилар тўпламига тегишли бўлган белгидан бошланиши керак. Номларда сонли белгилар, алфавитдаги барча белгилар ва `_`, `$` ва `#` белгиларидан фойдаланиш мумкин. Бунда Oracle корпорацияси `$` ва `#` белгиларини ишлатишга тавсия бერмайди. Номларда қўштирноқ белгисини ишлатиш мумкин эмас. Боғланиш каналларини номлашда `.` ва `@` белгиларидан фойдаланиш мумкин. Номланишда берилганлар базасини бошқариш тизимидаги (БББТ) захираланган хизматчи сўзлардан фойдаланиш мумкин эмас. DUAL сўзи SQL*Plus ва SQL*Forms - ларда тез-тез ишлатиладиган махсус бўш жадвалнинг номи ва шу сабабли уни объект ёки унинг қисмини номлашда фойдаланмаслик керак.

Хар бир схема объектларининг номлари ўзининг шахсий *номлар фазоларидан* ташкил топган бўлиб, хар бир номлар фазоси ичида номлар такрорланмайди. Қуйида биз схема объектлари учун хар бир номлар фазосини алоҳида рамка ичида келтирамыз.

2.1 расм. Схема объектлари учун номлар фазолари



Қуйида биз схемага таалуқли бўлмаган, балки бутун берилганлар базасига мансуб бўлган объектларнинг номлар фазоларини келтирамыз

2.2 расм. Бошқа объектлар учун номлар фазоси:



Хар бир жадвалда устунларнинг номлари такрорланмаслиги лозим. Номларга доир қатор мисоллар келтирамыз:

- emp;
- “emp”;
- Emp;
- “Emp”

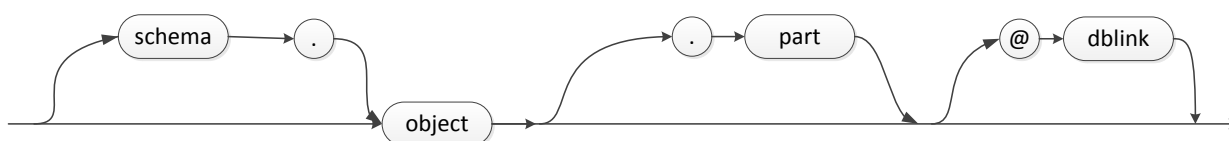
Oracle бу номларни бир хил ном сифатида қабул қилади.

- `ename;`
- `horse ;`
- `scott.hiredate;`
- “EVEN THIS & THAT !”
- `a_very_very_very_long_name.`

2.3. Объектларга мурожаат қилиш қоидалари ва тавсиялар.

ORACLE тизимида объект ва унинг қисмларига мурожаат қилиш қуйидаги диаграммалар асосида амалга оширилади.

а) Объектга мурожаат қилиш умумий синтаксис диаграммаси қуйидагича:



Бу ерда

object - объектнинг номи ;

schema - объектни ўз ичига олувчи схема;

part - объект қисмлари (масалан жадвалнинг устуни);

dblink - объектни ўз ичига олувчи берилганлар базасининг номи ;

Мисоллар :

```

1. INSERT INTO Dept
VALUES ( 50, 'SUPPORT' , 'PARIS' )
  
```

бу мисолда Dept номли жадвалга янги қатор қўшиляпти. Жадвал номига тўғридан-тўғри мурожаат қилинаяпти.

```

2. DROP TABLE Scott.Emp
  
```

Ушбу иккинчи мисолда SCOTT схемасига тегишли бўлган EMP жадвали берилганлар базасидан йўқотиляпти. Бу ерда жадвал номига унинг қайси схемага мансуб эканлигини ҳисобга олиб мурожаат қилинаяпти.

б) Боғланиш каналларини ташкил қилиш диаграммасини келтирамиз:

dblink::=



Бу ерда:

database - ушбу канал орқали боғланадиган узоқлашган берилганлар базасининг номи;

domain - ушбу канал орқали боғланадиган узоқлашган берилганлар базасининг домени.

Connection_descriptor - битта берилганлар базаси учун бир нечта боғланиш каналларини ташкил қилишга имкон беради.

2.4. Литераллар.

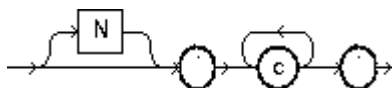
Аввал литераллар ҳақида тушунча ҳосил қилиш учун уларга доир бир қатор мисоллар билан бошлаймиз:

- * 'JACK', 'BLUEISLAND' ва '101' - белгили литераллар;
- * 5001 - сонли литерал.

Ушбу мисолларга таяниб литералларга таъриф берамиз. Литерал деб берилганларнинг фиксирланган (ўзгармас) қийматларига айтилади.

а) *Белгили литераллар* қуйидаги 'text' номли диаграммаси асосида берилади:

text::=



Бу ерда *C* - белгилар тўпламидаги ихтиёрий белги (бунда ' мустасно)

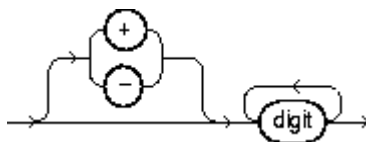
Мисоллар :

- * 'Салом; '
- * 'ORACLE.dbs '
- * 'oq_MAP '

б) *Сонли литераллар* -га доир 'integer' диаграммаси :

Ушбу литерал бутун мусбат сонларни ифодалаш учун ишлатилади.

integer::=



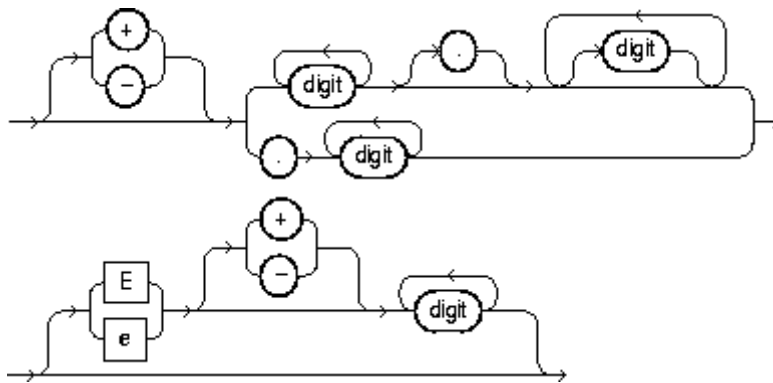
Бу ерда *digit* - 0 , 1, 2, 3, 4, 5, 6, 7, 8, 9 қийматларни қабул қилиши мумкин.

Мисоллар.

- 9
- 395

в) *Сонли литераллар* га доир 'Number' диаграммаси. Бу литерал бутун ёки ҳақиқий сонларни ифодалаш учун ишлатилади. Унинг диаграммаси:

number::=



Бу ерда digit - 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, - сонларни қабул қилиши мумкин ;

e - сонлар нормалланган кўринишда берилганлигини билдиради.

E - белгисидан кейин келадиган рақамлар даража кўрсаткичини аниқлайди ва у -130 дан 125 гача ўзгариши мумкин.

Мисоллар 445, + 8.67, 0.5, 76e - 05

2.5. Берилганлар турлари

Хар бир литерал ёки жадвал устуни берилганлар турига эга. Берилганларнинг қуйидаги ички турлари мавжуд:

- VARCHAR2 (size) - ўзгарувчи узунликдаги белгили қатор. Максимал узунлиги 2000 байтгача бўлиши мумкин;
- NUMBER (p,s) - p ($1 < p < 38$) ишончли рақам ва s ($-84 < s < 127$) масштабга эга бўлган сон;
- LONG - узунлиги 2 гигабайтгача ўзгарувчи узунликдаги белгили берилганлар;
- DATE - эрамиздан олдин 1 янв. 4712 йилдан эрамиздаги 31 дек. 4712 йилгача бўлган барча тўғри кунлар;
- RAW (SIZE) - иккилик СС(санок системаси) берилганлар. Максимал узунлиги 2000 байтгача бўлиши мумкин;
- LONG RAW - иккилик СС берилганлар. Максимал узунлиги 2 гигабайтгача бўлиши мумкин;
- ROWID - жадвал қаторининг 16 - лик СС уникал адреси;
- CHAR (size) - фиксирланган узунликдаги (1 байтдан 255 байтгача) белгили берилганлар;
- MLSLABEL - операцион системанинг иккилик СС меткасининг формати. Бу тур фақат Trusted ORACLE да ишлатилади

Белгили берилганлар турлари.

Ушбу тур сўз ва матнлар устида амаллар бажариш учун ишлатилади. Ушбу турни ифодалаш учун CHAR ва VARCHAR2 турлари ишлатилади.

CHAR турдаги берилганлар.

CHAR турдаги берилганлар ўзгармас узунликдаги белгили қаторлар учун ишлатилади. Агарда CHAR турдаги устуннинг ўлчови кўрсатилмаса, ORACLE тизими уни 1 байт деб қабул қилади. CHAR турдаги устуннинг ўлчови кўпи билан 255 байтдан ошмаслиги лозим.

VARCHAR2 турдаги берилганлар.

VARCHAR2 турдаги берилганлар ўзгарувчи узунликдаги белгили қаторлар учун ишлатилади. VARCHAR2 турдаги устунлар кўпи билан 2000 байтгача бўлиши мумкин.

NUMBER турдаги берилганлар.

NUMBER турдаги берилганлар 1.0×10^{-130} дан $9.9 \dots 9 \times 10^{125}$ гача бўлган нул, манфий ва мусбат сонлар учун ишлатилади. ʻзгармас нуқтали сон учун NUMBER(p,s) тури ишлатилади. Бу ерда

p - ишончли рақамлар сони (1 дан 38 гача);

s - ўнли нуқтадан кейинги рақамлар сони(-84 дан 127 гача).

Булардан ташқари қуйидаги NUMBER турлари мавжуд :

а) NUMBER (p) - яъни s = 0

б) NUMBER - 38 аниқликга эга бўлган сузувчи нуқтали сон

Мисоллар :	сон	тур	сақланиш ҳолати
	7456123.89	NUMBER	7456123.89
	7456123.89	NUMBER (7, -2)	7456100

LONG турдаги берилганлар.

Бу тур ўзгарувчи узунликдаги узунлиги 2 гигабайтдан ошмаган белгили қаторлар учун ишлатилади.

а) Хар қандай жадвал биттадан кўп бўлмаган LONG турдаги устунга эга бўлиши мумкин.

б) LONG турдаги устунлар яхлитлик чеклагичларида пайдо бўлиши мумкин эмас (бундан NULL ва NOT NULL мустасно);

в) LONG турдаги устунларни индекслаш мумкин эмас;

г) LONG турдаги устунларни процедура ва функцияларнинг аргументлари сифатида ишлатиш мумкин эмас.

д) Функциялар LONG турдаги қийматларни қайтариш мумкин эмас;

е) SQL операторларида LONG турдаги барча устунлар, кетма - кетликлар, янгиланувчи жадваллар ва маҳкамланган (блокированные) жадваллар битта берилганлар базасида бўлиши лозим.

ж) WHERE, GROUP BY, ORDER BY, CONNECT BY, DISTINCT операторларида LONG турини ишлатиш та'қиқланади.

* SQL функцияларида (SYBSTR ёки INSTR га ўхшаган);

* Тўпламлар устида амаллар бажаришидаги танлаш рўйхатида;

* CREATE TABLE AS SELECT операторининг танлаш рўйхатида

LONG туридаги устун ишлатиш мумкин эмас.

DATE туридаги берилганлар.

Date тури кун ва вақт ҳақидаги ахборотни сақлаш учун ишлатилади. ҳар бир date туридаги қиймат учун қуйидаги ахборот сақланади:

- аср;
- йил;
- ой;
- кун;
- соат;
- минут;
- * секунд.

SYSDATE функцияси оний кун ва вақтни қайтаради. Date турдаги қийматларни сонли қийматлар билан қўшиб, айириш мумкин.

Мисол :

SYSDATE + 1 - эрта;

SYSDATE - 7 - бир ҳафта олдин.

(SYSDATE - (10/1440)) - 10 минут олдин.

Юлиан санаси : Юлиан санаси - эрамиздан олдин 1 январь 4712 йилдан бошлаб кунлар сонини ифодалайди.

Мисол :

```
SELECT TO_CHAR (TO_Date('01-01-1992', 'MM-DD-YYYY '), 'J' )
FROM DUAL
```

Бу мисол 1 январь 1992 йилга эквивалент бўлган Юлиан санасини қайтаради.

RAW ва LONG RAW турдаги берилганлар.

Бу тур байтли берилганларни, белгили қаторларни, сузувчи нуқтали берилганларни ва иккилик СС берилганларни сақлаш учун ишлатилади. RAW турдаги берилганлар устида амаллар бажариш мумкин эмас.

ROWID турдаги берилганлар.

Ҳар бир қатор берилганлар базасида адресга эга бўлиб, у ROWID турга тегишли бўлади.

ROWID псевдоустуннинг белгили қийматлари block.row.file кўринишда бўлади. Бу ерда

а) block - қаторни ўз ичига оловчи берилганлар файлдаги блок бўлиб, у 16-лик СС да берилади;

б) row - 4 байтли узунликга эга бўлган 16-ли СС да блокдаги қаторни белгилайди.

в) file - берилганлар базасидаги файлнинг номерини билдиради.

Мисол: 0000000f.0000.0002

15-нчи блок 1-нчи бет 2-нчи файл.

MLSLABEL турдаги берилганлар.

Меткаларни иккилик форматда сақлаш учун ишлатилади.

2.7. Берилганларни солиштириш қоидалари.

а) *Сонли қийматлар.* Сонли қийматлар арифметика қоидаларига асосан солиштирилади.

Мисол. $-1 < 100$; $-100 < -5$

б) *Саналар.* Саналар ҳам а) бўлимидагидек солиштирилади:

Мисол : 29-Март-1991 < 05-январ-1992 ;

05-Январ-1992 1:35 pm > 05-Январ-1992 10:09 am.

в) *Белгили қаторларни солиштириш.* Белгиларни кодига қараб солиштирилади:

Мисол 'a b' > 'a a' ; 'a b' > 'a' ; 'a b' > 'a', 'a b' = 'a b' ; 'a ' = 'a '

2.8. Берилганларни турини ўзгартириш.

Берилганларни турини бевосита ёки билвосита ўзгартириш мумкин.

Берилганларни ошкормас равишда турларини ўзгартириш.

Мисол 1.

```
SELECT ename
      FROM emp
      WHERE empno = '7936'
```

Бу мисолда оператор '7936' ни ошкормас равишда 7936 - га айлантиради.

Мисол 2.

```
SELECT ename
      FROM emp
      WHERE hiredate = '12-MAR - 1993'
```

Мисол 3.

```
SELECT ename
      FROM emp
      WHERE ROWID = '00002514.0001.0001'
```

Берилганларни турини ошкор равишда ўзгартириш.

Берилганларни турини ошкор равишда қуйидаги жадвалда келтирилган функциялар ёрдамида ўзгартириш мумкин:

	CHAR	NUMBER	DATE	RAW	ROWID
CHAR		TO_NUMBER	TO_DATE	HEXTORAW	CHARTOROWID
NUMBER	TO_CHAR		TO_DATE (number, 'J')		
DATE	TO_CHAR	TO_CHAR (date, 'J')			

RAW	RAWTOHEX				
ROWID	ROWIDTOCHAR				

Бу функциялар тўғрисида кейинроқ батафсил изоҳ берамиз.

2.9. Бўш қиймат.

Агарда қатордаги бирор устунда қиймат бўлмаса, бу устун бўш қиймат ёки null қийматга эга дейилади. Солиштириш шартларида COMM = NULL кўринишдаги солиштириш натижаси номаълум. муыйидаги жадвалда бўш қийматлар қатнашган амалларнинг натижалари келтирилган:

В қиймати	Шарт	Натижа
10	В IS NULL	FALSE
10	В IS NOT NULL	TRUE
NULL	В IS NULL	TRUE
NULL	В IS NOT NULL	FALSE
10	В = NULL	Номаълум
10	В ! = NULL	Номаълум
NULL	В = NULL	Номаълум
NULL	В ! = NULL	Номаълум

2.10. Псевдоустунлар.

Псевдоустунлар ўзини худди устунлардек олиб боради, лекин улар ҳақиқатда жадвалда сақланмайди. Псевдоустунлардан берилганларни фақат ўқиш мумкин холос. Бу ерда биз асосан

CURRVAL;
NEXTVAL;
LEVEL;
ROWID;
ROWNUM

псевдоустунларни ўрганиш билан чегараланамиз.

а) CURRVAL ва NEXTVAL. Кетма - кетлик (sequence) - бу схеманинг объекти бўлиб, у уникал бўлган кетма - кет қийматларни ишлаб чиқаради.

* CURRVAL - кетма кетликнинг оний қийматини қайтаради

* NEXTVAL - кетма кетликнинг оний қийматини ошириб навбатдаги қийматини қайтаради.

Мисоллар. 1) (Кетма кетликга мурожаат қилиш)

Sequence.CURRVAL

Sequence. NEXTVAL
 Schema.sequence.CURRVAL
 Schema.sequence.NEXTVAL
 Schema.sequence.CURRVAL@dblink
 Schema.sequence.NEXTVAL@dblink

2) Ишлатиш.

1. SELECT empseq.currval
 FROM DUAL;
2. INSERT INTO emp
 VALUES (empseq.nextval, ' LEWIS ' , 'CLERK ' , 7902,
 SYSDATE, 1200, NULL, 20);
3. INSERT INTO master_order (orderno , customer, orderdate)
 VALUES (orderseq.nextval, ' all "s Auto Shop ' , SYSDATE)
 INSERT INTO detail_order (orderno, part, quantity)
 VALUES (orderseq.currval, ' SPARKPLUG' ,4)
 INSERT INTO default_order (orderno, part, quantity)
 VALUES (orderseq.currval, ' FUEL PUMP , 1)

б) LEVEL - Иерархия дарахтидаги даража тартибини қайтади.

в) ROWID - қатор адресини қайтади.

Мисол. SELECT ROWID, ename

 FROM emp
 WHERE deptno = 20

г) ROWNUM. Берилганлар базасидан суғуриб олинган қатор тартибларини қайтади.

Мисол.

 SELECT *
 FROM emp
 WHERE ROWNUM < 10
 10 та қаторни қайтади.

Изоҳлар. SQL операторларида изоҳларни қўшиш мумкин.

| *.....* | - кўп қаторли изоҳлар учун ишлатилади. -- (иккита минус белгиси) бир қаторли изоҳ учун ишлатилади.

Мисол SELECT ename, -- номни танлаш
 job, -- лавозим
 loc, -- шаҳар
 FROM emp, -- барча хизматчилар учун
 dept
 WHERE emp.deptno = dept. deptno

§ 3. Амаллар

Амалларни 2 синфга бўлиш мумкин:

- 1) Унар амаллар битта ҳад (операнд) устида бажарилади :

амал операнд

- 2) Бинар амаллар эса иккита ҳад устида бажарилади:
операнд1 амал операнд2

3.1. Амаллар бажариш тартиби

- 1) Унар арифметик амаллар (+ -) ва PRIOR амали;
- 2) Арифметик амаллар (* /);
- 3) Бинар арифметик амаллар (+ -), белгили амал |
|;
- 4) Солиштириш амаллари (барчаси);
- 5) NOT мантиқий амал ;
- 6) AND мантиқий амал;
- 7) OR мантиқий амал.

Мисол. $1 + 2 * 3$. Бу ерда ORACLE аввал 2 ни 3 га кўпайтириб натижани 1га қўшади.

3.2. Арифметик амалларга мисоллар

- а) + , - унар амаллар :
- ```
* SELECT * FROM orders
 WHERE qtysold = -1 ;
```
- \* SELECT \* FROM emp
 WHERE -sal <0
- б) \*, / - бинар амаллар:
- ```
UPDATE emp
      SET sal = sal * 1.1
```
- в) + - бинар амаллар :
- ```
SELECT sal + comm FROM emp
 WHERE sysdate - hiredate > 365
```

### 3.3. Белгили амалларга мисол.

- а) || - Конкатенация амали
- ```
SELECT 'Name is ' || ename FROM emp
```
- б) CREATE TABLE tab1 (col1 VARCHAR2(6), col2 CHAR(6) ,
col3 VARCHAR2(6), col4 CHAR (6))
INSERT INTO tab (col1, col2, col3, col4,)
VALUES (' abc', 'def', ' dnc', ' jkl ')
SELECT col1 || col2 || col3 || col4 "Concatenation" FROM tab1

absdef dhi jkl

3.4. Солиштириш амаллари.

- а) = (тенглик) солиштириш амалига мисол:
 SELECT * FROM emp
 WHERE sal = 1500
- б) !=, ^=, ^=, <> (тенгсизлик) солиштириш амалларига мисол:
 SELECT * FROM emp
 WHERE sal != 1500
- в) <, > (кичик, катта) солиштириш амалларига мисоллар:
 * SELECT * FROM emp
 WHERE sal < 1500
 * SELECT * FROM emp
 where SAL > 1500
- г) >=, <= (катта ёки тенг, кичик ёки тенг) солиштириш амалларига мисоллар:
 * SELECT * FROM emp
 WHERE sal >= 2500
 * SELECT * FROM emp
 WHERE sal <= 2500
- д) IN (тўпламнинг ихтиёрий элементига тенг) солиштириш амалига мисоллар:
 * SELECT * FROM emp
 WHERE JOB IN ('CLERK' , 'ANALYST')
 * SELECT * FROM emp
 WHERE sal IN (SELECT sal FROM emp WHERE deptno = 30)
- е) NOT IN (тўпламга тегишли эмас) солиштириш амалига мисоллар:
 * SELECT * FROM emp WHERE sal NOT IN (SELECT sal
 FROM emp WHERE deptno = 30)
 * SELECT * FROM emp
 WHERE job NOT IN ('CLERK' , 'ANALYST')
- ё) ANY, SOME (тўпламнинг ихтиёрий элементи) солиштириш амалига мисол:
 SELECT * FROM emp WHERE sal = ANY
 (SELECT sal FROM emp WHERE deptno = 30)
- ж) ALL (тўпламнинг барча элементи) солиштириш амалига мисол:
 SELECT * FROM emp
 WHERE sal >= ALL (1400, 3000)

з) [NOT] BETWEEN x AND y (x ва y интервалида етиш [етмаслик]) солиштириш амалига мисол:

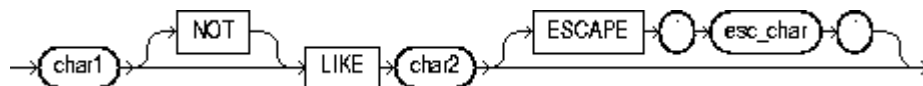
```
SELECT * FROM emp WHERE sal
      BETWEEN 2000 AND 3000
```

к) EXISTS (мавжуд) солиштириш амалига мисол:

```
SELECT dname, deptno FROM dept
      WHERE EXISTS ( SELECT * FROM emp
                     WHERE dept. Deptno = emp. Deptno )
```

л) x [NOT] LIKE y [ESCAPE z] (x ва y-қолип устма-уст тушиш [тушмаслик]) солиштириш амалига мисоллар. LIKE амали белгили қаторларни шаблон орқали солиштириш учун ишлатилади. Ушбу амал синтаксисининг диаграммаси қуйидагича:

LIKE condition : : =



Мисоллар:

- SELECT sal FROM emp WHERE ename LIKE 'SM%'
- SELECT sal FROM emp WHERE ename LIKE 'MA% '

Ушбу шартни MARTIN, MA, MARK, MARY қаноатлантиради.
- SELECT sal FROM emp WHERE ename LIKE 'SMITH '

Ушбу шартни SMITHE , SMITHY, SMITHS қаноатлантиради.

3.5. Мантиқий амаллар.

1) NOT (инкор қилиш) мантиқий амалига доир мисоллар:

- * SELECT * FROM emp WHERE NOT (job IS NULL)
- * SELECT * FROM emp WHERE NOT

(sal BETWEEN 1000 AND 2000)

2) AND(ва) мантиқий амалига доир мисол:

- * SELECT * FROM emp WHERE lob = 'CLERK' AND deptno = 10

3) OR(ёки) мантиқий амалига доир мисол:

- * SELECT * FROM emp WHERE job = 'CLERK ' OR deptno = 10

3.6. Тўпламлар ўстида амаллар.

1) UNION(қўшиш) - тузилиши бир хил бўлган тўпламларни қўшиш амали. Ушбу амал бажарилиши жараёнида устма уст тушадиган қаторлар бир қатор сифатида қабул қилинади.

Мисол : SELECT part FROM orders_list1
UNION
SELECT part FROM orders_list2

2) UNION ALL (қўшиш) - тузилиши бир хил бўлган тўпламларни қўшиш амали. Ушбу амал бажарилиши жараёнида устма уст тушадиган қаторлар, такрорланиш сонига тенг бўлган қаторлар сифатида қабул қилинади.

Мисол : SELECT part FROM orders_list1
UNION ALL
SELECT part FROM orders_list2

3) INTERSECT (кесишма) тузилиши бир хил бўлган тўпламларни кесишмасини аниқлаш амали.

Мисол : SELECT part FROM orders_list1
INTERSECT
SELECT part FROM orders_list2

4) MINUS (минус) тузилиши бир хил бўлган тўпламларни айирмасини аниқлаш амали.

Мисол : SELECT part FROM orders_list1
MINUS
SELECT part FROM orders_list2

Бошқа амаллар :

1) (+) - бу амалдан олдин келадиган жадвал устуни ташқи уланиш устуни бўлишини билдиради.

SELECT ename, dname
FROM emp, dept
WHERE dept. Deptno = emp. Deptno (+)

- dept жадвалида бўлими бўлмаган қаторларни ҳам қайтаради

2) PRIOR - бўлиниш даражаси (унар амал)

§ 4. Функциялар

4.1. Сонли функциялар

Сонли функциялар - сонли ўзгарувчилар устида амал бажариб сонли қиймат қайтаради.

1) ABS (n) - n соннинг абсолют қийматини қайтаради.

SELECT ABS (-15) “ Абсолют қиймат ” FROM DUAL

Бу функция 15 қайтаради;

2) CEIL(n) - n га тенг ёки ундан катта бўлган энг кичик бутун сонни қайтаради.

Мисол. CEIL (15.7) = 16;

3) COS(n) - радианда ифодаланган n бурчакнинг косинусини қайтаради;

4) COSH(n) -n-нинг гиперболик косинусини қайтаради;

5) EXP (n) - e^n қайтаради;

6) FLOOR (n) - n га тенг ёки ундан кичик бўлган энг катта бутун сонни қайтаради;

Мисол. FLOOR (15.7) = 15

7) LN(n) - n-нинг натурал логарифмини қайтаради;

8) LOG(m,n) - m асосли n дан логарифмини қайтаради;

9) MOD(m,n) - m-ни n-га бўлишидан қолган қолдиқни қайтаради;

10) POWER(m,n) - m^n қайтаради;

11) ROUND(n[,m]) - ўнли нуқтадан кейин m хонагача яхлитланган n сонини қайтаради;

Мисол. ROUND (15.193, 1) = 15.2

$$12) \text{SIGN}(n) = \begin{cases} -1, & n < 0 \\ 0, & n = 0 \\ 1, & n = 1 \end{cases}$$

13) SIN (n) радианда ифодаланган n бурчакнинг синусини қайтаради;

14) SINH(n) n-нинг гиперболик синусини қайтаради;

15) SQRT(n) - $\sqrt{n}$ қайтаради;

16) TAN(n) - радианда ифодаланган n бурчакнинг тангенсини қайтаради;

17) TANH(n) - n-нинг гиперболик тангенсини қайтаради;

18) TRUNC(n[,m]) - n соннинг ўнли нуқтасидан бошлаб m хонадан кейинги қисмини кесиб ташлайди

Мисол. TRUNC (15.79,1) = 15.7

4.2. Белгили функциялар

Белгили функциялар - белгили ўзгарувчилар устида амал бажариб белгили ёки сонли қийматни қайтаради.

1) CHR(n) - n кодга эга бўлган белгини қайтаради.

2) LPAD(char1,n[,char2]) - Char1 га чап томондан Char 2 - даги белгиларни кетма-кет қўшиб узунлиги n-га тенг бўлган матнни қайтаради.

Мисол. LPAD('Page1', 15 , '*') = *.*.*.*.*Page1

3) LTRIM(char[,set]) - Char - даги барча белгиларни чап томондан бошлаб set тўпламда мавжуд бўлмаган биринчи белгигача чиқариб ташлайди.

Мисол. LTRIM('хухXхуLAST WORD' , 'ху') = XхуLAST WORD

4) NLS_INITCAP(char[,nlsparams']) - char сўзларидаги биринчи ҳарфларни бош ҳарфлар билан ва қолган ҳарфларни кичик ҳарфлар билан алмаштириб қайтаради. Бу ерда nlsparams-нинг қиймати қуйидаги форматга эга:

'NLS_SORT = sort', sort эса лингвистик саралаш кетма-кетлиги ёки BINARY бўлиши мумкин.

Мисол. NLS_INITCAP('Ijsland', 'NLS_SORT = XDutch') = Ijsland.

5) NLS_LOWER (char [, 'nlsparams']) - char-даги ҳарфларни кичик ҳарфларга алмаштириб қайтаради

Мисол. NLS_LOWER ('CITTA', 'NLS_SORT = XItalian') = citta

6) NLS_LOWER (char [, 'nlsparams']) - char-даги ҳарфларни катта ҳарфларга алмаштириб қайтаради

Мисол. NLS_UPPER ('groß', 'NLS_SORT = XGerman') = GROSS.

7) REPLACE (char, search_string [, replacement_string]) - char-даги search_string қаторқисмларини replacement_string қаторқисмларига алмаштириб қайтаради.

Мисол. REPLACE ('JACK and JUE', 'J', 'BL') = BLACK and BLUE.

8) RPAD(char1, n [,char2,]) - Char1 га ўнг томондан Char 2 - даги белгиларни кетма-кет қўшиб узунлиги n-га тенг бўлган матнни қайтаради.

Мисол. RPAD('TURNER', 11, 'ab') = TURNERababa

9) RTRIM(char[,set]) - Char - даги барча белгиларни ўнг томондан бошлаб set тўпламда мавжуд бўлмаган биринчи белгигача кесиб ташлайди.

Мисол. RTRIM('TURNERxyXxy', 'xy') = TURNERxyX

10) SUBSTR (char , m [, n]) - char-нинг m-нчи белгисидан бошлаб n дона белгисини қайтаради.

Мисол. SUBSTR ('ABCDEFGFG', 3, 2) = CD

11) TRANSLATE (char , from , to) - char-даги ҳар бир белги агар у from-да мавжуд бўлса уни to-даги мос белги билан алмаштиради.

Мисол. TRANSLATE ('2KRW229',
'0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ',
'9999999999XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX') =
= 9XXX999.

12) UPPER(char) - char-даги барча ҳарфларни катта ҳарфларга алмаштириб қайтаради

Мисол. UPPER (' Large ') = LARGE.

13) ASCII (char) - char-даги биринчи байтни ўнли кодини қайтаради.

Мисол. ASCII (' Q ') = 81.

14) INSTR (char1, char2[,n[,m]]) - char1-нинг n-нчи белгисидан бошлаб char2-нинг char1-га m-нчи маротаба қатнашишидаги биринчи белгисининг жойланиш тартибини қайтаради

Мисол. INSTR ('CORPORATE FLOOR', 'OR', 3, 2) = 14

16) INSTRB(char1, char2[,n[,m]]) - худди INSTR га ўхшайди фақат n , m - лар байт билан ўлчанади. Масалан икки байтли белгилар учун :

INSTRB ('CORPORATE FLOOR', 'OR', 5, 2) = 27 бўлади.

17) LENGTH(char) - char-даги белгилар сонини қайтаради.

Мисол. LENGTH('CANDIDE') = 7.

18) LENGTHB (char) - char-нинг узунлигини байтда қайтаради.

19) NLSSORT (char [, ' nlsparams ']) - char ни саралаш учун ишлатадиган байтлар занжири қайтаради. Бу ерда 'nlsparams' қиймати юқорида келтирилган.

Мисол : SELECT * FROM emp
WHERE NLSSORT (ename, 'NLS_SORT = GERMAN') >
NLSSORT ('B' , 'NLS_SORT = GERMAN')

4.3. Сана функциялари

20) ADD_MONTHS(d, n) - d кунига n ойни қўшиб қайтаради. Фараз қилайлик hiredate ўзгарувчиси DATE турига мансуб бўлиб 17- dec - 1980 қийматга эга бўлсин. У ҳолда

TO_CHAR(ADD_MONTHS(hiredate,1),'DD-MM-YYYY')=17-JAN-1981

21) LAST_DAY (d) - d да кўрсатилган ойнинг охири кунини қайтаради.

Мисол : LAST_DAY (SYSDATE) = 30 - NOV - 92

22) MONTHS_BETWEEN (d1, d2) - d1 ва d2 кунлари ўртасидаги ойлар сонини қайтаради

Мисол : MONTHS_BETWEEN(TO_DATE('02-02-1992' , 'MM-DD-YYYY'), TO_DATE('01-01-1992' , 'MM-DD-YYYY'))=1.03225

23)NEW_TIME (d, z1, z2) - d вақтни z1 минтақавий вақтдан z2 минтақавий вақтга айлантириб қайтаради. z1 ва z2 : ўзгарувчилар қуйидаги қийматларни қабул қилишлари мумкин:

- AST/ADT Атлантик Стандарт
- BST/BDTБеринг Стандарти
- CST/CDTМарказий Стандарт
- EST/EDTШарқий Стандарт
- GMT Greenwich 'рта Вақти
- HST/HDT Стандарт Аляска-Гавайи Стандарт Вақти.
- MST/MDT ТоғлиСтандарт
- NST Newfoundland Стандарти
- PST/PDT Тинч Океани Стандарти
- YST/YDT Yukon Стандарти

Мисол :

TO_CHAR(NEW_TIME(TO_DATE('17:47' , 'hh24:mi') , 'PST' , 'GMT'), 'hh24:mi') = 01 : 47

24) NEXT_DAY(d, char) - char да номи кўрсатилган ҳафта кунининг d кунидан кейинги биринчи кунини қайтаради.

Мисол. NEXT_DAY ('15-MAR-92', 'TUESDAY') = 17-MAR-92

25) ROUND(d[,fmt]); - fmt форматда кўрсатилган формат бўйича d-ни яхлитлаб қайтаради.

Мисол. ROUND(TO_DATE('27-OCT-92'), 'YEAR') = 01-JAN-93.

26) SYSDATE - оний вақт ва кунни қайтаради .

27) TRUNC(d[,fmt]) - fmt форматда кўрсатилган формат бўйича d-ни кесиб қайтади.

Мисол. TRUNC(TO_DATE('27-окт-92' , 'DD-MM-YY') , 'YEAR') = 01-JAN-92.

4.4. Турларни ўзгартириш функциялари.

28) CHARTOROWID (char) - CHAR турдаги char-ни ROWID турига айлантириб беради.

Мисол : SELECT ename from emp WHERE
ROWID = CHARTOROWID ('000000F.0003.0002')
SWITH-ни қайтади.

29) CONVERT (char , dest_char_set [, source_char_set]) - char белгили қаторни бир source_char_set белгилар тўпламидан бошқа dest_char_set белгилар тўпламига ўтказиб беради.

Мисол. CONVERT ('GroB' , ' WE8HP' , 'WE8DEC') = GroB

Белгилар тўплами :

US7ASCII US 7-битли ASCII белгилар тўплами □

-
- WE8DEC DEC □арбий Европа 8-битли белгилар тўплами
- WE8HP HP □арбий Европа Laserjet 8-битли белгилар тўплами
- F7DECDCE Франция 7-битли белгилар тўплами
- WE8EBCDIC500 IBM Гарбий Европа EBCDIC 500 Кодлар бети
- WE8PC850 IBM PC 850 Кодлар бети
- WE8ISO8859P1 ISO 8859-1 □арбий Европа 8-битли белгилар тўплами

30) HEXTORAW (char) - char-даги 16-лик рақамларни RAW туридаги қийматларга айлантириб беради.

Мисол. INSERT INTO graphics (raw-column)
SELECT HEXTORAW ('7D')
FROM DUAL

31) RAWTOHEX (char) - char-даги RAW туридаги қийматларни 16-лик эквивалентига айлантириб беради.

Мисол. SELECT RAWTOHEX (raw-column) "Graphics"
FROM graphics

Graphics

7D

32) ROWIDTOCHAR (rowid) - rowid (ROWID туридаги) қийматини VARCHAR2 туридаги қийматга айлантириб беради

Мисол. ROWIDTOCHAR (ROWID) = 00000F38.0001.0001

33) TO_CHAR(d[,fmt[, 'nlsparams']]) - DATE туридаги d-нинг қийматини fmt-форматли маскада кўрсатилган формат бўйича VARCHAR2

туридаги қийматга айлантиради. Бунда ‘ nlsparams ‘-параметрнинг қиймати ҳафта куни ва ойнинг номи ҳамда аббревиатурасини қайси тилда ёзилишини кўрсатади. У қуйидагича берилади:

‘NLS_DATE_LANGUAGE = language ‘

Мисол: TO_CHAR(HIREDATE, ‘Month DD, YYYY’) = Декабрь 11 , 1980

34) TO_CHAR(n[,fmt[,‘nlsparams’]]) - NUMBER туридаги n сонни fmt формат бўйича VARCHAR2 туридаги қийматга айлантириб беради. Бунда ‘nlsparams’-параметри сонли форматли маскадаги элементлар қайтарадиган қуйидаги белгиларни кўрсатади

‘NLS_NUMERIC_CHARACTERS = “dg”, бу ерда d ва g мос равишда ўнли нуқта ва гуруҳларга бўлиш белгиларини билдиради;

NLS_CURRENCY = “text”

NLS_ISO_CURRENCY = territory’

Мисол: TO_CHAR(17145,‘L099G99’, NLS_NUMERIC_CHARACTERS = “.,” NLS_CURRENCY = “AUD”) = AVD017,145

35) TO_DATE (char [, fmt [, ‘ nlsparams ‘]]) - CHAR туридаги char-нинг қийматини fmt-форматли маскада кўрсатилган формат бўйича DATE туридаги қийматга айлантиради.

Мисол. TO_DATE(‘january 15, 1989, 11:00 A.M.’ ,
‘ Montn dd , YYYY, HH:MI A.M. ‘ ,
, NLS_DATE_LANGUAGE = American’)

36) TO_MULTI_BYTE (char) - char-даги бир байтли белгиларни кўп байтли белгиларга айлантириб беради.

37) TO_NUMBER (char [, fmt [, ‘ nlsparams ‘]]) - CHAR ёки VARCHAR2 туридаги char-нинг қийматини fmt-форматли маскада кўрсатилган формат бўйича NUMBER туридаги қийматга айлантиради.

Мисол. TO_NUMBER (‘AUD100.00’ , ‘L999D99’ ,
NLS_NUMERIC_CHARACTERS = “.,”
NLS_CURRENCY = “AUD” ‘) = 100.00

38) TO_SINGLE_BYTE- char-даги кўпбайтли белгиларни бир байтли белгиларга айлантиради

4.5. Бошқа функциялар

39) DUMP(expr[,return_format[,start_position[,length]]]) expr-қийматини return_format санок системасидаги кодини start-position тартибидан бошлаб length узунликда қайтаради. Фараз қилайлик ename ўзгарувчи ‘scott’ қийматига эга бўлсин. У ҳолда

DUMP(ename, 8, 3, 2) = 117,124;

DUMP(ename, 10, 3, 2) = 79,84;

DUMP(ename, 16, 3, 2) = 4f,54;

DUMP(ename, 17, 3, 2) = 0,T

40) GREATEST(expr[,expr]. . .) - берилган ифодалар ўртасидан энг каттасини аниқлаб унинг қийматини қайтаради.

Мисол. GREATEST (' HARRY ' , ' HARRIOT ' , ' HAROLD ') = HARRY

41) LEAST(expr[,expr] . . .) - берилган ифодалар ўртасидан энг кичигини аниқлаб унинг қийматини қайтаради.

Мисол. LEAST(' HARRY ' , ' HARRIOT ' , ' HAROLD ').

42) NVL(expr1, expr2) - агар expr1 - бўш қийматга эга бўлса expr2 қийматини қайтаради , акс ҳолда expr1 - ни қайтаради.

43) UID - оний фойдаланувчи уникал кодини қайтаради.

44) USER - оний фойдаланувчи номини қайтаради.

Мисол. SELECT USER, UID FROM DUAL

45) UserEnv (option) - оний сеанс ҳақида маълумот қайтаради. Бу ерда option қуйидаги қийматлардан биттасини қабул қилиши мумкин:

- * 'ENTRYID' - ёзув протоколи идентификаторини қайтаради;
- * 'LABEL' - оний сеанс меткасини қайтаради;
- * 'LANGUAGE' - оний сеансдаги тил, жой ва белгилар тўпламини қайтаради;
- * 'SESSIONID' - оний сеанс идентификатори қайтаради ;
- * 'TERMINAL' - оний сеансдаги операцион системанинг идентификатори;

Мисол. SELECT USERENV (' LANGUAGE ') "Язык" From DUAL

46) VSIZE (expr) - expr-нинг узунлигини байтда қайтаради.

Мисол. VSIZE(ename) = 5.

4.6. Гуруҳли функциялар.

Гуруҳли функциялар қаторлар тўплами устида амал бажариб (скаляр функциялар алоҳида қатор устида амал бажаради) натижа қайтаради.

47) AVG ([DISTINCT | ALL n) - n ўзгарувчининг ўрта қийматини қайтаради.

Мисол. SELECT AVG (Sal) "ўрта қиймат" From emp .

Бу мисолда emp жадвалидаги Sal устунининг ўрта қиймати қайтарилади.

48) COUNT({* | [DISTINCT|ALL] expr}) - сўровларда суғуриб олинadиган қаторлар сонини қайтаради. Агар expr параметри кўрсатилса, у ҳолда бу функция expr бўш бўлмаган қийматга эга бўлган қаторлар сонини қайтаради. мўшимча равишда DISTINCT параметри кўрсатилса, у ҳолда фақат expr қиймати устма-уст тушмайдиган қаторлар сони қайтарилади. ALL параметри кўрсатилса, у ҳолда бу функция expr бўш бўлмаган қийматга эга бўлган қаторлар сонини қайтаради.

Мисол. SELECT COUNT (*) "Барчаси" From emp

Ушбу мисол emp жадвалидаги барча қаторлар сонини қайтаради.

49) GLB ([DISTINCT | ALL] label) - label меткаси қийматининг энг катта қуйи чегарасини қайтаради.

50) LUB ([DISTINCT | ALL] label) - label меткаси қийматининг энг кичик қуйи чегарасини қайтаради

51) MAX ([DISTINCT | ALL] expr) - expr ифодасининг энг катта қийматини қайтаради.

Мисол. SELECT MAX (sal) “Максимум” From emp

52) MIN ([DISTINCT | ALL] expr) - expr ифодасининг энг кичик қийматини қайтаради

Мисол. SELECT MIN (hiredate) “Энг биринчи” FROM emp

53) STDDEV ([DISTINCT | ALL] x) - x катталик қийматини стандарт оғишмасини қайтаради.

54) SUM ([DISTINCT | ALL] n) - n қийматларининг йиғиндисини қайтаради

Мисол. SELECT SUM (sal) “ҳаммаси” From emp.

55) VARIANCE ([DISTINCT | ALL] x катталикнинг дисперсиясини қайтаради

§ 5. Формат маскалари

Формат маскаси DATE ва NUMBER турдаги берилганларни белгили қаторда қандай форматда сақлашни тасвирлайдиган белгили литералдир.

5.1. NUMBER форматининг элементлари.

муайян биз NUMBER форматининг элементларини жадвал кўринишда келтирамиз:

Элемент	Мисол	Изоҳ
9	9999	9-ларнинг сони натижадаги «ишончли» рақамлар сонини кўрсатади. Ноллар ўрнига бўш жой қайтаради.
0	0999,9990	Етакчи нолни қайтаради ва нол қийматини кўрсатилган жойга (бўш қиймат эмас) 0 сифатида қайтаради.
\$	\$9999	Соннинг олдида доллар аломати қўшиб қайтаради.
B	B9999	Нол ўрнига бўш қиймат қайтаради. Бунда форматли маскадаги нол ҳисобга олинмайди.
MI	9999MI	Манфий сонлардан кейин ‘-’ аломатини қайтаради.
S	S99999999 S	Кўрсатилган жойда мусбат қийматлар учун ‘+’ аломатини ва манфий сонлар учун ‘-’ аломатини қайтаради.
PR	9999PR	Манфий қийматларни бурчакли қавсларда қайтаради. Мусбат қийматлар учун қиймат бошиди ва охирида бўш жой қайтаради.
D	99D99	Кўрсатилган жойда сонни бутун ва бўлак қисмини ажратадиган ўнли нуқтани тасвирлайдиган белгини

		қайтаради.
G	9G999	Кўрсатилган жойда сонни гуруҳларга бўладиган белгини қайтаради.
C	C999	Кўрсатилган жойда ISO стандарти бўйича валюта аломатини қайтаради.
L	L999	Кўрсатилган жойда маҳаллий валюта аломатини қайтаради.
, (comma)	9,999	Кўрсатилган жойда вергул аломатини қайтаради.
. (period)	99.99	Кўрсатилган жойда нукта аломатини қайтаради.
V	999V99	ийматни 10 ⁿ -га кўпайтиради, бу ерда n V-дан кейинги 9-ларнинг сони.
EEEE	9.9EEEE	ийматни нормалланган формада қайтаради.
RN _m	RN	Рим рақамларни кичик ёки катта ҳарфларда қайтаради.
FM	FM90.9	ийматни олдида ва охирида бўш жойсиз қайтаради.

Мисол. SELECT ename employee, TO_CHAR (Comm, '\$ 9, 990. 99')
FROM emp;

5.2. DATE форматининг элементлари.

қуйида биз DATE форматининг элементларини жадвал кўринишда келтирамиз:

Элемент	Тасвирланиши
-,.,: "text"	Пунктуация аломатлари ва матн қайтариладиган қийматда ўзгаришсиз қайтарилади
AD/A.D.	Эрамиздаги давр.
AM/A.M.	Яримкунгача бўлган даврни кўрсатади.
BC/B.C.	Эрамиздан олдинги давр.
CC/SCC	Аср; агар "S" кўрсатилган бўлса, унда эрамиздан олдинги саналар олдида "-" аломати қўйилади.
D	ҳафтанинг куни (1-7).
DAY	9 белгигача бўш жой билан тўлдирилган кун номини қайтаради.
DD	Ойнинг куни (1-31).
DDD	Йилнинг куни (1-366).
DY	Куннинг қисқартирилган номи.
IW	ISO стандарти бўйича йилнинг ҳафтаси (1-52 ёки 1-53).
IYY/IY/I	ISO стандарти бўйича йилнинг охириги3,2 ёки 1-та рақами.
IYYY	ISO стандарти бўйича йилнинг 4-та рақами.
HH/HH12	Куннинг соати (1-12).
HH24	Куннинг соати(0-23).

J	Юлиан куни; Эрамиздан олдин 1-январь 4712 йилдан бошлаб кунлар сони.
MI	Минут(0-59).
MM	Ой(01-12; JAN = 01)
MONTH	9 белгигача бўш жой билан тўлдирилган ой номи
MON	Ойнинг қисқартирилган номи.
RM	Ойнинг римли рақамларда ифодаланиши (I-XII; JAN = I).
Q	Йилнинг чораки(квартали) (1, 2, 3, 4; JAN-MAR = 1)
RR	Бошқа асрлар учун йилнинг охири иккита рақами.
WW	Йилнинг ҳафтаси (1-53). Биринчи ҳафта йилнинг биринчи кунда бошлаб 7 кун давом эттирилади.
W	Ойнинг ҳафтаси (1-5). Биринчи ҳафта ойнинг биринчи кунда бошлаб 7 кун давом эттирилади.
PM/P.M.	Яримкундан кейинги даврни кўрсатади.
SS	Сония (0-59).
SSSSS	Яримтундан кейинги сониялар миқдори (0-86399).
Y,YYY	Кўрсатилган жойда вергул қўйиб йил қайтарилади.
YEAR/SYEA	Сўз билан ёзилган йил. Агар "S" префикси қўйилган бўлса унда
R	эрамиздан олдинги саналар олдида "-" аломати қўйилади.
YYYY/SYYY	Йилнинг 4 рақами. Агар "S" префикси қўйилган бўлса унда
Y	эрамиздан олдинги саналар олдида "-" аломати қўйилади.
YYY/YY/Y	Йилнинг охири 3, 2 ёки 1 рақами;

Мисол 1. `Select ename, TO_CHAR(hiredate, 'fmMonth DD, YYYY')`
`hiredate FROM emp;`

hiredate

December 17, 1980.

Мисол 2. `UPDATE emp`
`SET hiredate = TO_DATE('1992 05 20', 'YYYY MM DD')`
`WHERE ename = 'JONES'`

Мисол 3. `SELECT TO_CHAR(TO_DATE('27-OCT-95', 'DD-MON-RR'), 'YYYY') FROM DUAL`

Мисол 4. а) `SELECT TO_CHAR (SYSDATE, 'fmDDTH "of" Month, YYYY') "Кун" FROM DUAL`

Кун

3RD of April, 1992

б) `SELECT TO_CHAR(SYSDATE, 'DDTH "of" Month, YYYY') FROM DUAL;`

Кун

3RD of April , 1992

Мисол 7. `SELECT TO_CHAR (SYSDATE,'fmDay ' 's Special'
') "Кун" FROM DUAL`

Кун

Tuesday's Special

Мисол 8. `UPDATE table
SET date_column = TO_DATE(char,'fmt');`

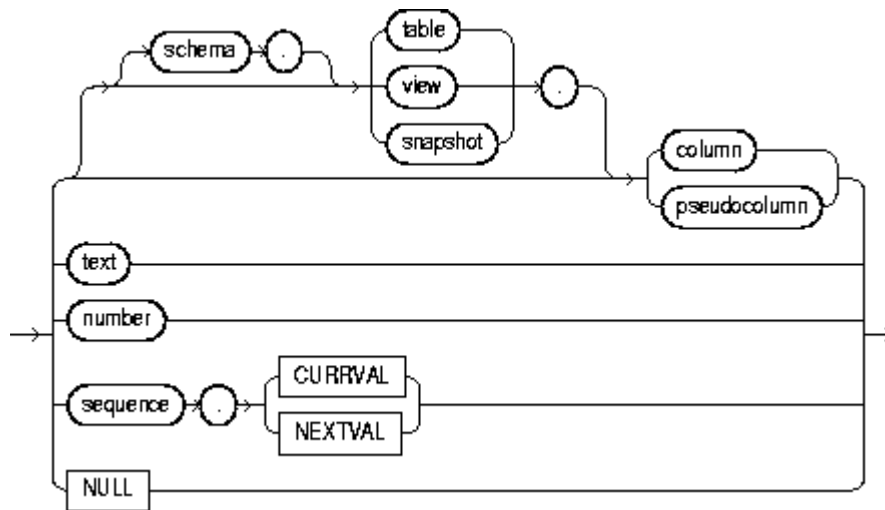
§ 6. Ифода (expr параметри)

expr - ихтиёрий.турдаги берилганларнинг ифодасини белгилайди. Ифода бир қатор формаларга эга бўлиши мумкин.

6.1. Формалар

Форма I. Бу форма устун, псевдоустун, ўзгармас ва кетма-кетлик тартибини ифодалайди.

1).expr (Form I) :: =



Мисоллар:

- * emp. Ename;
- * 'бу қатор';
- * 10.

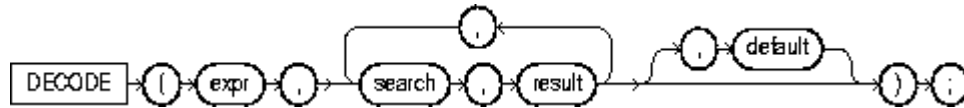
Форма II. Бу форма базавий ўзгарувчи ва шарт бўлмаган индикатор-ўзгарувчини ифодалайди.

2) expr (Form II) :: =

6.2. Ифодани декодлаштириш.

Махсус DECODE синтаксисидан фойдаланадиган ифода:

decode_expr ::=



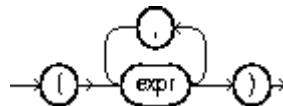
Навбат билан expr-нинг қиймати search-нинг навбатдаги қиймати билан солиштирилади, агар улар устма-уст тушса, унда мос равишдаги result-нинг қиймати қайтарилади, акс ҳолда default қийматини қайтаради ва агар default қиймати берилмаган бўлса бўш қийматни қайтаради.

Мисол: DECODE (deptno, 10, 'ACCOUNTING',
20, 'RESEARCH',
30, 'SALES',
40, 'OPERATION',
'NONE')

6.3. Ифодалар рўйхати.

қавслар ичида олинган ифодалар рўйхати:

expr-list ::=



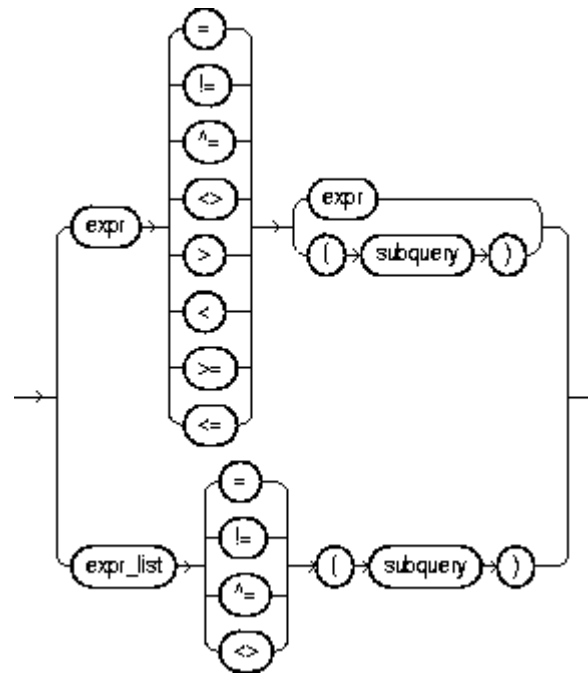
Мисоллар:

- * (10, 20, 40)
- * ('SSOTT', 'BLAKE', 'TAVLOR')
- * (LENGTH('MOOSE')*57, - SQRT(144)+72,69).

§ 7. Шарт

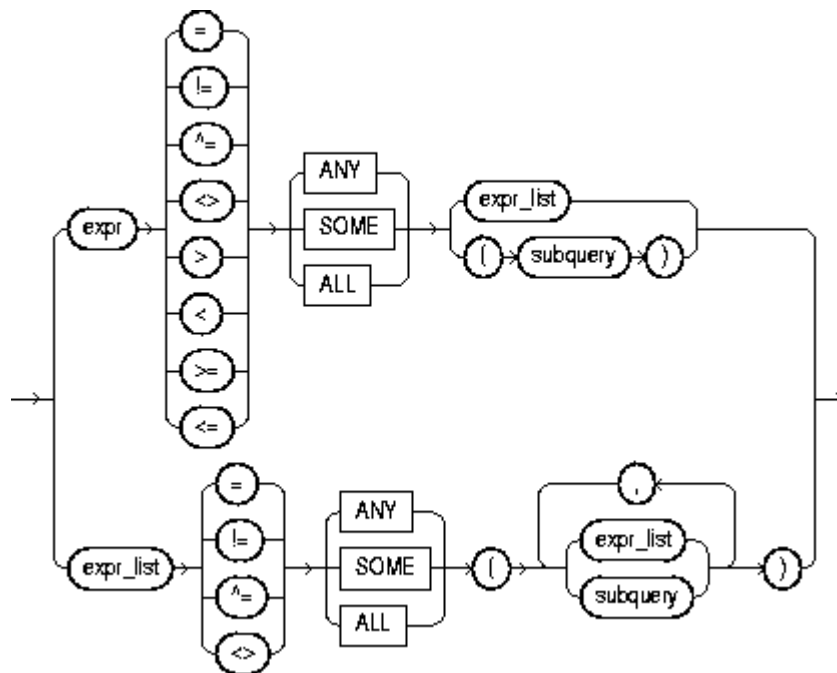
Шарт бу ифодалар ёки мантикий амаллар комбинациясидан иборат бўлиб TRUE, FALSE ёки номаълум натижа қайтаради. Шарт қатор формаларга эга бўлиши мумкин:

Форма I - Ифода ёки сўровости натижаси билан солиштириш.
condition(Form I) ::=



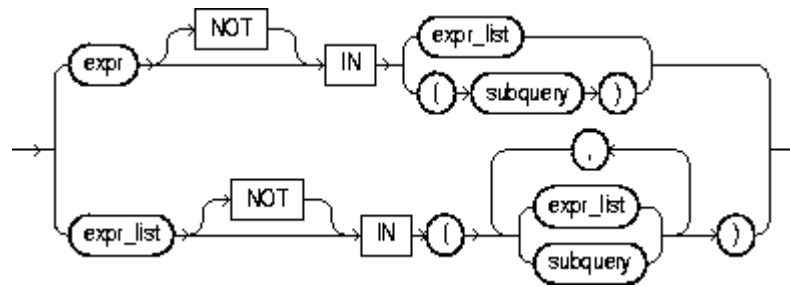
Форма II - Рўйхат ёки сўровостининг бирор элементи ёки барча элементлари билан солиштириш.

condition (Form II) ::=

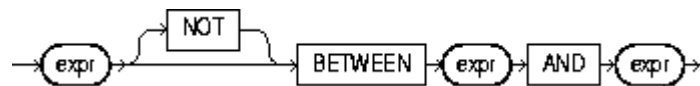


Форма III Рўйхат ёки сўровости ичида мавжудлигини текшириш.

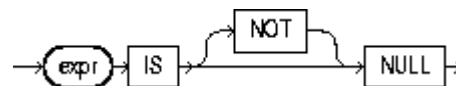
condition (Form III) ::=



Форма IV Диапазонда тушишини текшириш
condition (Form IV) ::=



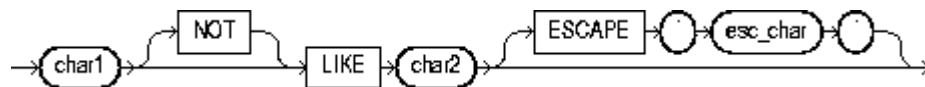
Форма V Бўш қийматга текшириш
condition (Form V) ::=



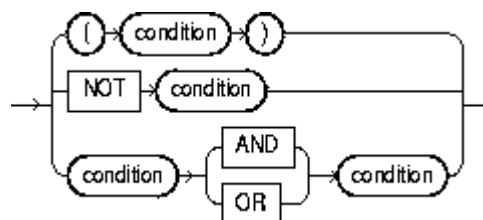
Форма VI Сўровости ҳеч бўлмаганда битта қатор қайтаришини текшириш:
condition (Form VI) ::=



Форма VII қолипга мос келишини текшириш
LIKE Condition (Form VII) ::=



Форма VIII Бошқа шартларнинг комбинацияси:
condition (Form VIII) ::=



Мисоллар:

* 1=1

```

*   NVL(sal,0)+NVL (comm,0)>2500
*   AND (5<7)
*   ename = 'SMITH'
*   emp.deptno = dert.deptno
*   hiredate > '01-JAN-88'
*   job IN ('PRESIDENT','CLERK','ANALIST')
*   sal BETWEEN 500 AND 1000
*   comm IS NULL AND sal =2000

```

Шартлар одатда DELETE
SELECT
UPDATE

→ командаларининг WHERE жумласида

ишлатилади.

Шартларни SELECT командасининг қуйидаги жумлаларида ҳам ишлатиш мумкин:

```

*   WHERE
*   START WITH
*   CONNECT BY
*   HAVING

```

§ 8. SQL командалари

Командалар қуйидаги қисмларга бўлинади:

- 1) Берилганларнинг аниқлаш тили (DDL);
- 2) Берилганларни манипуляция қилиш (бошқариш) тили (DML);
- 3) Транзакцияларни бошқариш;
- 4) Сеансни бошқариш;
- 5) Системани бошқариш;
- 6) Ички қурилган SQL.

8.1. Берилганларнинг аниқлаш тили командалари:

қуйида берилганларнинг аниқлаш тили командалари жадвалини келтирамиз:

Команда	Мақсад
ALTER CLUSTER	Кластернинг storage характеристикаларини ўзгартириш .
ALTER DATABASE	Берилганлар базаси параметрларини ўзгартириш.
ALTER FUNCTION	Сақланадиган функцияни қайтадан компиляция қилиш.
ALTER INDEX	Индексни қайтадан ташкил қилиш.
ALTER PACKAGE	Сақланадиган пакетни қайтадан компиляция қилиш.
ALTER PROCEDURE	Сақланадиган процедурани қайтадан компиляция қилиш.
ALTER PROFILE	Профилдаги ресурсларни ўзгартириш

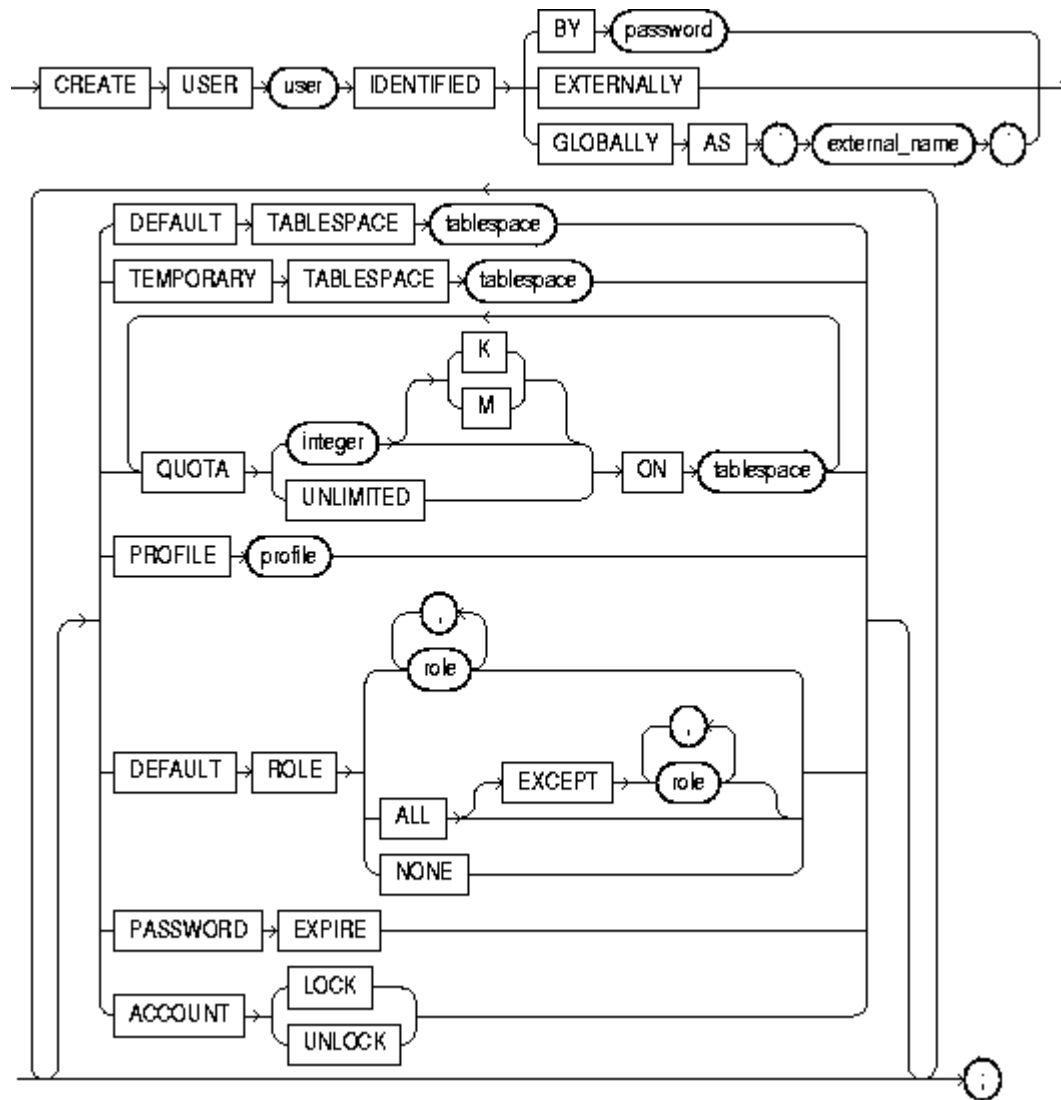
ALTER RESOURCE COST	Сессиядаги умумий ресурсларни баҳолаш формуласини аниқлаш.
ALTER ROLE	Ролни ўзгартириш.
ALTER ROLLBACK SEGMENT	дайтариш сегменти параметрларини ўзгартириш.
ALTER SEQUENCE	Кетма-кетлик параметрларини ўзгартириш.
ALTER SNAPSHOT	Оний нусха параметрларини ўзгартириш.
ALTER SHAPSHOT LOG	Оний нусха журнали параметрларини ўзгартириш.
ALTER TABLE	Жадвал параметрларини ўзгартириш.
ALTER TABLESPACE	Жадвалли фазо параметрларини ўзгартириш.
ALTER TRIGGER	Базадаги триггерни ишчи ёки ишсиз ҳолатга келтириш.
ALTER USER	Фойдаланувчи параметрларини ўзгартириш
ALTER VIEW	Кўринишни қайтадан компиляция қилиш.
ANALYZE	Статистик маълумотларни йиғиш
AUDIT	Аудит қилиш.
COMMENT	Изоҳ бериш.
CREATE CLUSTER	Кластер очиш.
CREATE CONTROLFILE	Контрол файлини ташкил қилиш.
CREATE DATABASE	Берилганлар базасини(ББ) ташкил қилиш.
CREATE DATABASE LINK	Узоқланган ББ алоқа каналини ташкил қилиш.
CREATE FUNCTION	Сақланадиган функцияни ташкил қилиш.
CREATE INDEX	Индексни ташкил қилиш.
CREATE PACKAGE	Пакет аниқланишини ташкил қилиш.
CREATE PACKAGE BODY	Пакет танасини ташкил қилиш.
CREATE PROCEDURE	Процедурани ташкил қилиш.
CREATE PROFILE	Профилни ташкил қилиш.
CREATE ROLE	Ролни ташкил қилиш.
CREATE ROLLBACK SEGMENT	дайтариш сегментини ташкил қилиш.
CREATE SCHEMA	Схемани ташкил қилиш.
CREATE SEQUENCE	Кетма-кетликни ташкил қилиш.
CREATE SHAPSHOT	Оний нусхани ташкил қилиш.
CREATE SNAPSHOT LOG	Оний нусха журналини ташкил қилиш.
CREATE SYNONYM	Синонимни ташкил қилиш.
CREATE TABLE	Жадвални ташкил қилиш.
CREATE TABLESPACE	Жадвалли фазони ташкил қилиш.
CREATE TRIGGER	Берилганлар базадаги триггерни ташкил қилиш.
CREATE USER	Фойдаланувчини ташкил қилиш.

CREATE VIEW	Кўринишни ташкил қилиш.
DROP CLUSTER	Кластерни ББ ўчириб ташлаш.
DROP DATABASE LINK	Алоқа каналини ўчириб ташлаш.
DROP FUNCTION	ББ сақланадиган функцияни ўчириб ташлаш.
DROP INDEX	ББ индексни ўчириб ташлаш.
DROP PACKAGE	ББ сақланадиган пакетни ўчириб ташлаш.
DROP PROCEDURE	ББ сақланадиган процедурани ўчириб ташлаш.
DROP PROFILE	ББ профилни ўчириб ташлаш.
DROP ROLE	ББ ролни ўчириб ташлаш.
DROP ROLLBACK SEGMENT	ББ қайтариш сегментини ўчириб ташлаш.
DROP SEQUENCE	ББ кетма-кетликни ўчириб ташлаш.
DROP SNAPSHOT	ББ оний нусхани ўчириб ташлаш.
DROP SNAPSHOT LOG	ББ оний нусха журналини ўчириб ташлаш.
DROP SYNONYM	ББ синонимни ўчириб ташлаш.
DROP TABLE	ББ жадвални ўчириб ташлаш.
DROP TABLESPACE	ББ жадвалли фазони ўчириб ташлаш.
DROP TRIGGER	ББ триггерни ўчириб ташлаш.
DROP USER	ББ фойдаланувчини ўчириб ташлаш.
DROP VIEW	ББ кўринишни ўчириб ташлаш.
GRANT	Грант бериш.
NOAUDIT	Аудитни олиб ташлаш..
RENAME	Схема номини ўзгартириш.
REVOKE	Берилган грантларни қайтариб олиш.
TRUNCATE	Жадвалнинг барча қаторларни ўчириш.

Юқоридаги жадвалда келтирилган командалар айримини изоҳлаб ўтамыз:

Create user.

Берилганлар базасининг фойдаланувчисини ташкил қилади
CREATE USER command ::=



Бу ерда

user - фойдаланувчининг номи;

IDENTIFIED - ORACLE-га фойдаланувчи кириш услубини кўрсатади:

* BY password - ORACLE тизимида password-пароли ёрдамида кириш;

* EXTERNALLY - Операцион системаси орқали кириш;

* DEFAULT TABLESPACE - агар ошкор равишда жадвалли фазо кўрсатилмаса, фойдаланувчи объектлари ушбу фазода очилади.

* TEMPORARY TABLESPACE - фойдаланувчиларнинг вақтинчалик сегментлари учун вақтинчалик жадвалли фазони аниқлайди. Агар сиз ушбу жумлани қолдириб юборсангиз, у ҳолда бу мақсад учун ORACLE SYSTEM жадвалли фазосини ишлатади.

* QUOTA - фойдаланувчига жадвалли фазодан фойдаланишга квотани(жойни) ажратиб беради

Квотани сиз байт, К ёки М-ёрдамида (мас равишда килобайт ёки мегабайт) кўрсатишингиз мумкин.

* **UNLIMITED** - кўрсатилган жадвалли фазодан хоҳлаганча фойдаланиш мумкин.

* **PROFILE** - фойдаланувчига profile профилини тайинлайди. Профил берилганлар базасининг ресурсларини чегаралайди. Агар профил ошкор равишда кўрсатилмаса

* **DEFAULT** профили тайинланади.

Мисол:

1). **CREATE USER chief**

IDENTIFIED BY teacher

DEFAULT TABLESPACE students

TEMPORARY TABLESPACE teachers

QUOTA 10 M ON students

QUOTA 5 M ON teachers

QUOTA 5M ON system

PROFILE professor

2). Фақат операцион системанинг фойдаланувчисига рухсат бўлган фойдаланувчини ташкил қилиш учун мисол

CREATE USER ops\$george

IDENTIFIED EXTERNALLY

DEFAULT TABLESPACE accs_ts

TEMPORARY TABLESPACE temp_ts

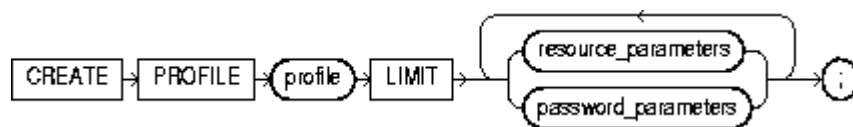
QUOTA UNLIMITED ON accs_ts

QUOTA UNLIMITED on temp_ts

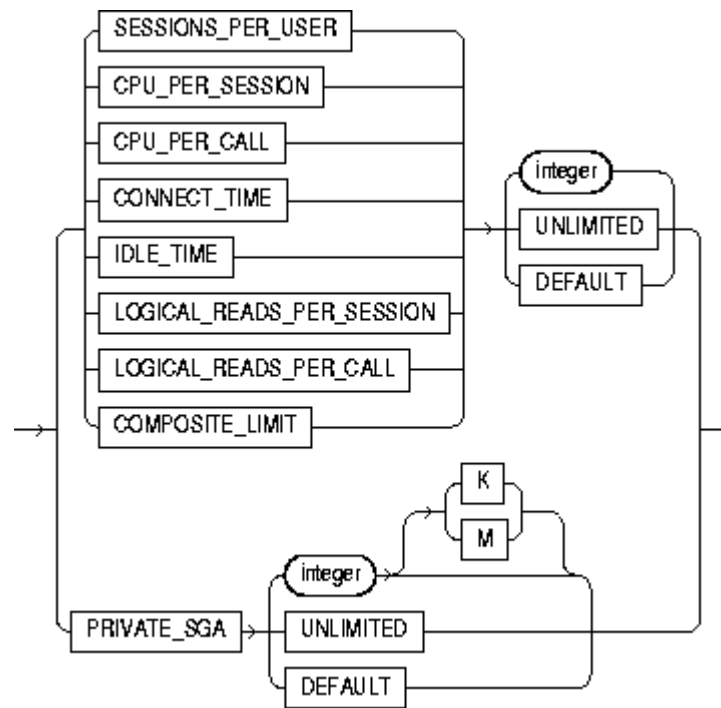
Бу ерда ops\$ - OS_AUTHENT_PREFIX инициализация параметрининг қиймати.

Create profile.

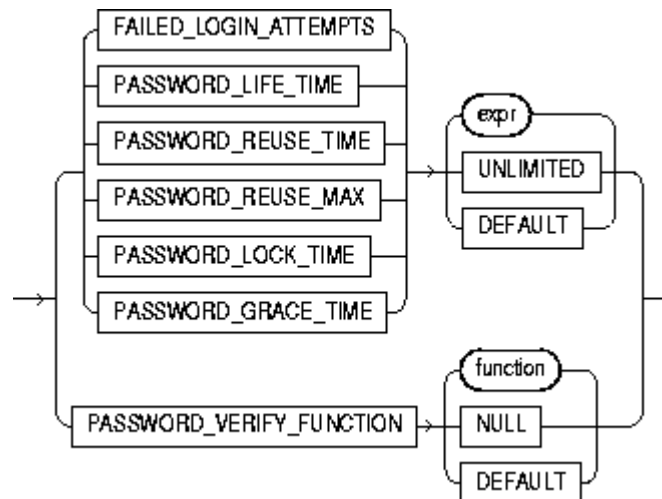
Берилганлар базасида профилни ташкил қилади.



[resource_parameters::=](#)



password_parameters::=



Бу ерда

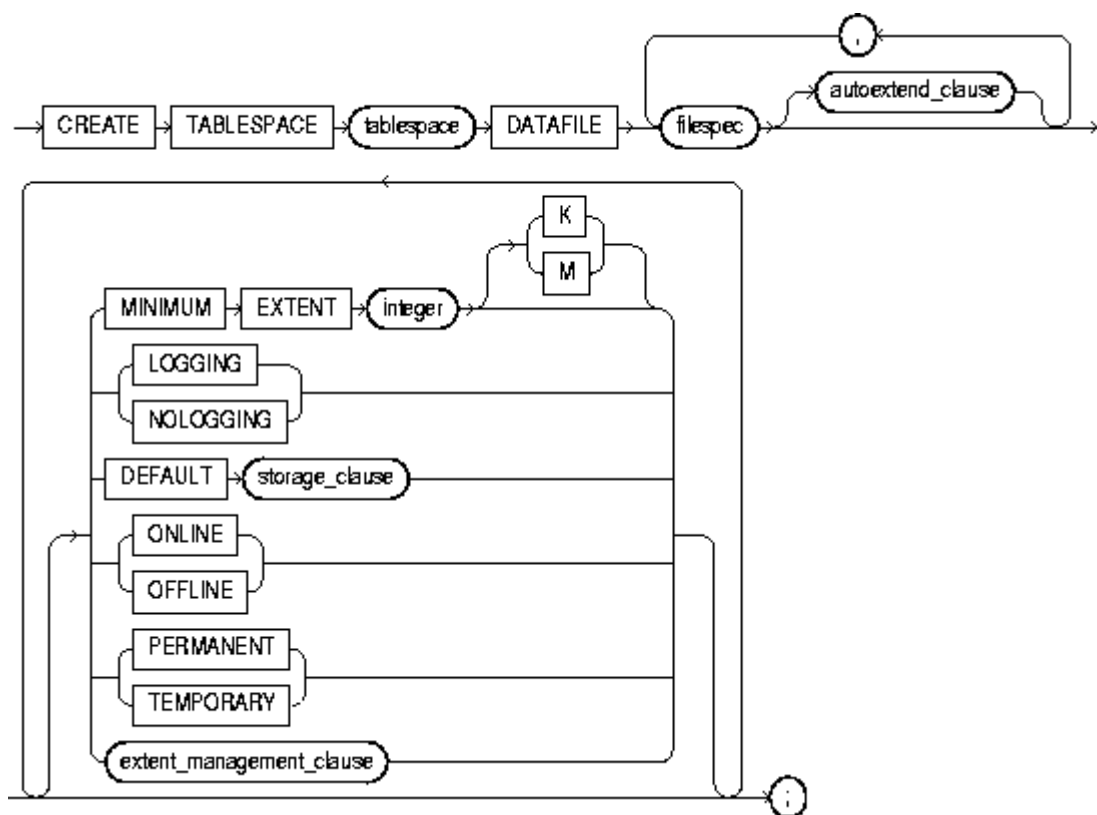
- * profile - профил номи
- * SESSION_PER_USER - параллел сеанслар сони
- * CPU_PER_SESSION - CPU вақти битта сеанс учун (секундда юздан бир қисми)
- * CPU_PER_CALL - CPU вақти битта чакирув учун (секундда юздан бир қисми)
- * CONNECT_TIME - Битта сеансинг умумий вақти(минутда)
- * IDLE_TIME - Бир сеанс жараенида актив бўлмаган вақт(минутда)
- * LOGICAL_READS_PER_SESSION - берилганлар блокларининг ўқишлар сони(бир сеансда)

- * LOGICAL_READS_PER_CALL - берилганлар блокларига мурожаат қилишлар сони битта чақиришда
- * PRIVATE_SGA - хусусий хотира соҳаси (SGA да)
- * COMPOSITE_LIMIT - ресурслар йиғиндисининг нархи (сервис бирлигида)

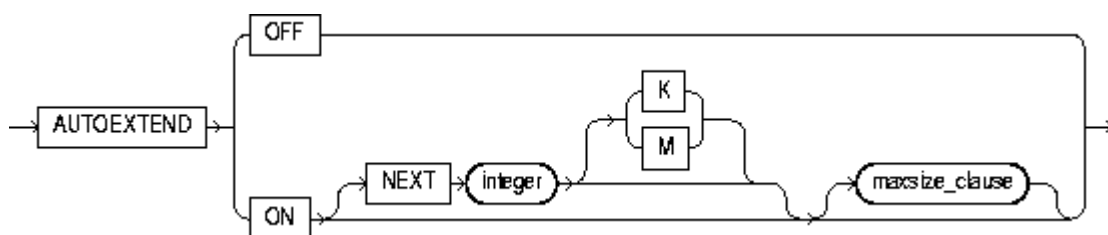
Create tablespace.

Жадвалли фазо - берилганлар базасидаги фазо бўлиб ва у қуйидаги сегментларни ўз ичига олиши мумкин:

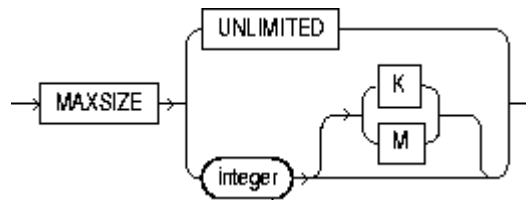
- * Берилганлар сегментлари
- * Индекслар сегментлари
- * җайтариш сегментлари
- * Вақтинчалик сегментлари



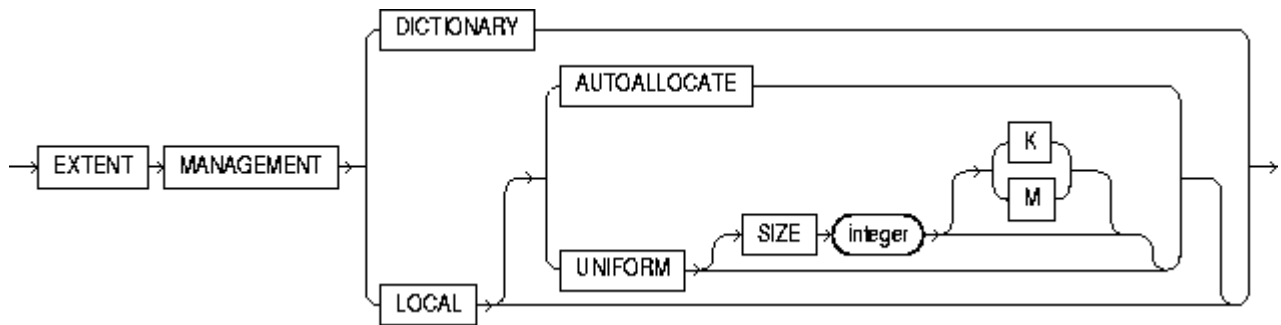
autoextend_clause::=



maxsize_clause::=



extent_management_clause::=



Бу ерда:

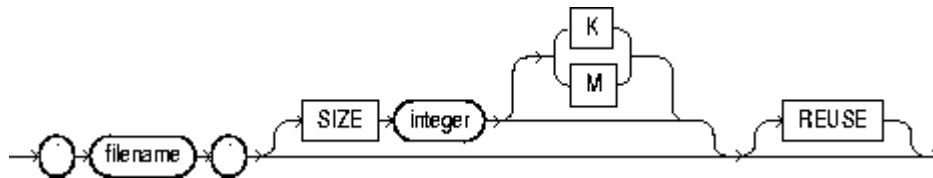
- `tablespace` - жадвалли фазонинг номи;
- `DATAFILE` - жадвалли фазони ўз ичига олувчи берилганлар файли ёки файллари;
- `DEFAULT STORAGE` - объектларни сақлаш параметрлари; (агар улар ошкор равишда кўрсатилмаган бўлса);
- `ONLINE` - жадвалли фазода ишлашга рухсат этилади;
- `OFFLINE` - жадвалли фазода ишлашга рухсат этилмайди.

Мисол: `CREATE TABLESPACE tablespace_2
DATAFILE 'diska: tablespace_2. dat' SIZE 20M
DEFAULT STORAGE (INITIAL 10K NEXT 50K
MENEXTENTS 1 MAXEXTENTS 999
PCTINCREASE 10)
ONLINE.`

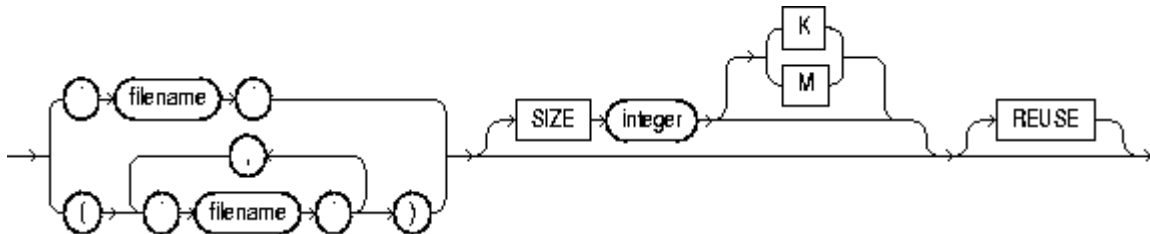
Filespace.

Файлни берилганлар файли сифатида тайинлайди ёки файл ёки файллар гуруҳини журналлар файли гуруҳи сифатида тайинлайди:

a) *filespace* (Data Files)::=



б) `filesize (redo Lod FileGroups) ::=`



Бу ерда

- 'filename' - берилганлар файлининг номи
- SIZE -файл ўлчови
- REUSE - мавжуд бўлган файлни қайтадан ишлатишга рухсат беради

Мисоллар:1) `CREATE DATABASE payable`

`LOGFILE GROUP1 ('diska:log1.log','diskb:log1.log')SIZE 50K`

`GROUP2 ('diska:log2.log','diskb:log2.log')SIZE 50K`

`DATAFILE 'diskc: dbone.date'SIZE 30M`

2) `ALTER DATABASE stocks`

`ADD LOGFILE GROUP 3 (diska:log3.log','diskb:log3.log')`

`SIZE 50K REUSE`

3) `GREATE TABLESPACE stocks`

`DATAFILE 'diskc:stock1.dat',`

`'diskc:stock2.dat',`

`'diskc:stock3.dat'`

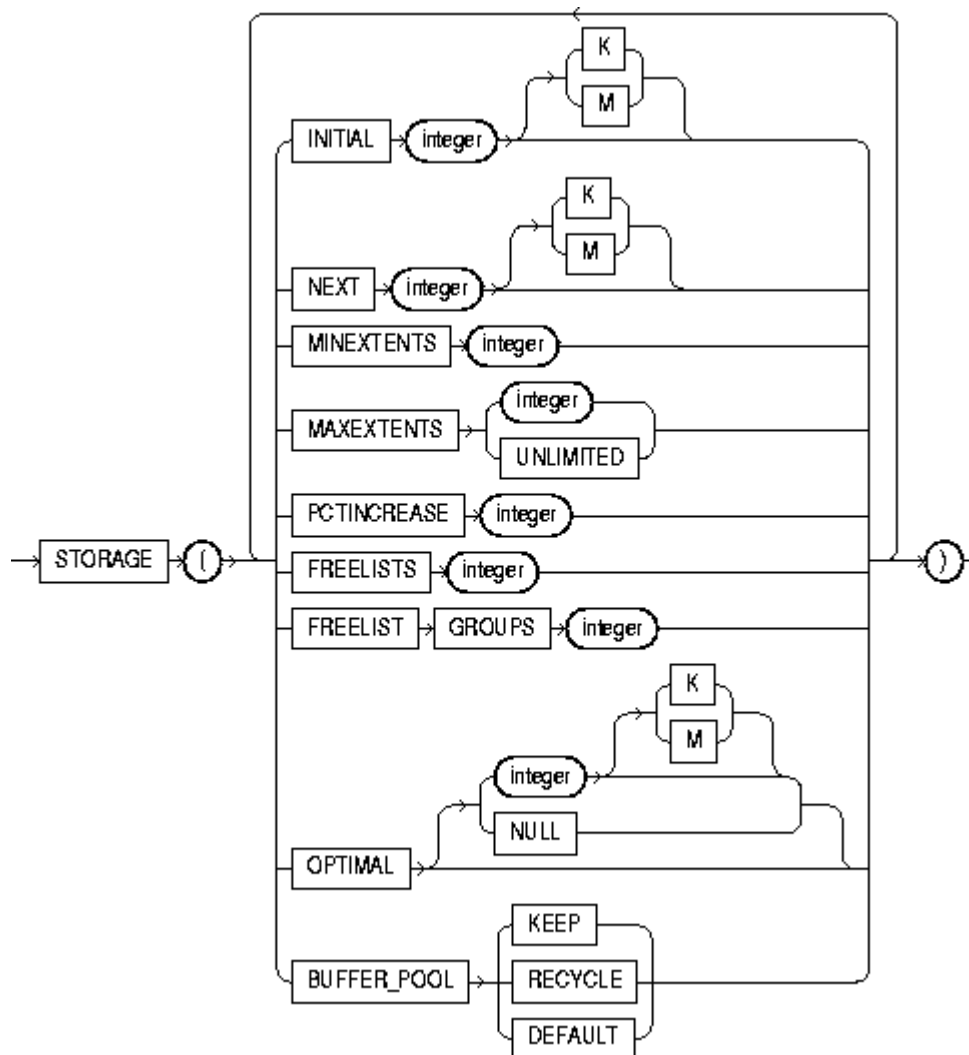
4) `ALTER TABLESPACE stocks`

`ADD DATAFILE 'diskc:stocks4.dat' REUSE`

Storage.

Жадвал, индекс, кластер, қайтариш сегментларини жадвалли фазода сақлаш параметрларини аниқлайди.

`STORAGE storage_clause ::=`



Бу ерда

- 1) **INITIAL** - биринчи экстенст ўлчовини байтда кўрсатади
- 2) **NEXT** - навбатдаги экстенст ўлчовини байтда кўрсатади
- 3) **PCTINCREASE** - навбатдаги экстенст олдинги экстенстда нисбатан неча процентга ошишини кўрсатади
- 4) **MINEXTENTS** - сегмент очилганда ажратиладиган экстенстларнинг умумий сони
- 5) **MAXEXTENS** - кўпи билан берилган объект учун ажратиладиган экстенстлар сони
- 6) **OPTIMAL** - қайтариш сегментда оптимал ўлчовини байтда кўрсатади. Бу параметр фақат қайтариш сегменти учун ишлатилади
- 7) **FREELIST GROUPS** - бўш блоklar рўйхати гуруҳларининг сонини кўрсатади. Параллел режимда ишлайди.
- 8) **FREELISTS** - бўш блоklar рўйхати гуруҳидаги рўйхатлар сонини кўрсатади

Мисоллар 1) `CREATE TABLE dept`
 `( deptno NUMBER(2),`
 `dname VARCHAR2(14),`

```

loc    VARCHAR2(13) )
STORAGE ( INITIAL 100 K NEXT 50 k MINEXTENTS 1
          MAXEXTENTS 50 PCTINCREASE 5 )
2) CREATE ROLLBACK SEGMENT  rsone
STORAGE ( INITIAL 10 K NEXT 10 K MINEXTENTS 2
          MAXEXTENTS 25 OPTIMAL 50

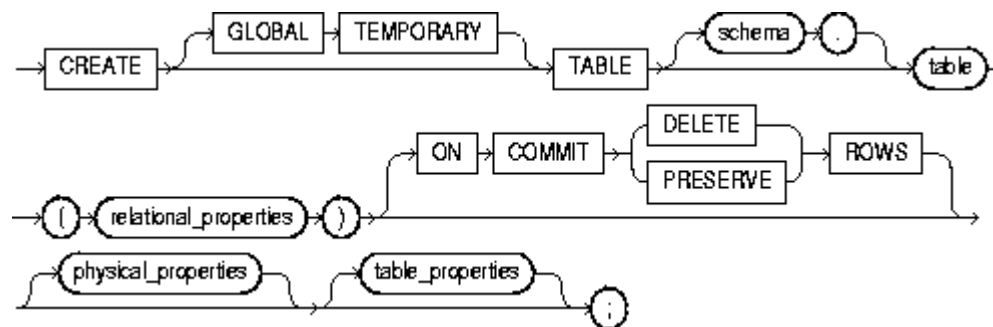
```

k)

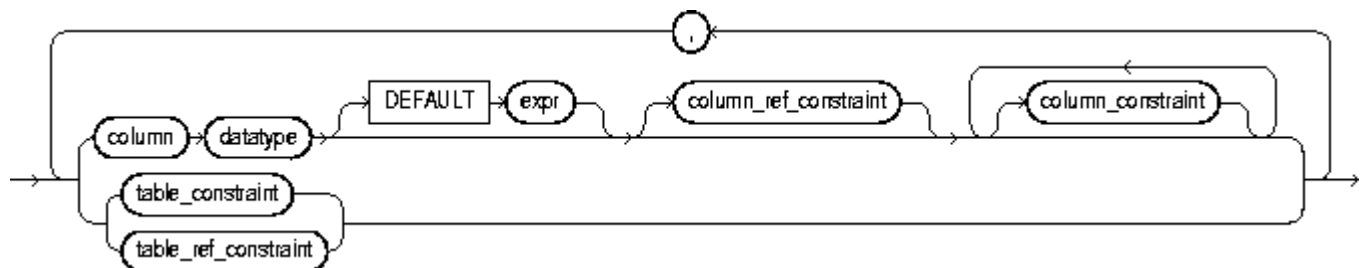
Create table.

Бу - жадвални ташкил қилиш командаси.

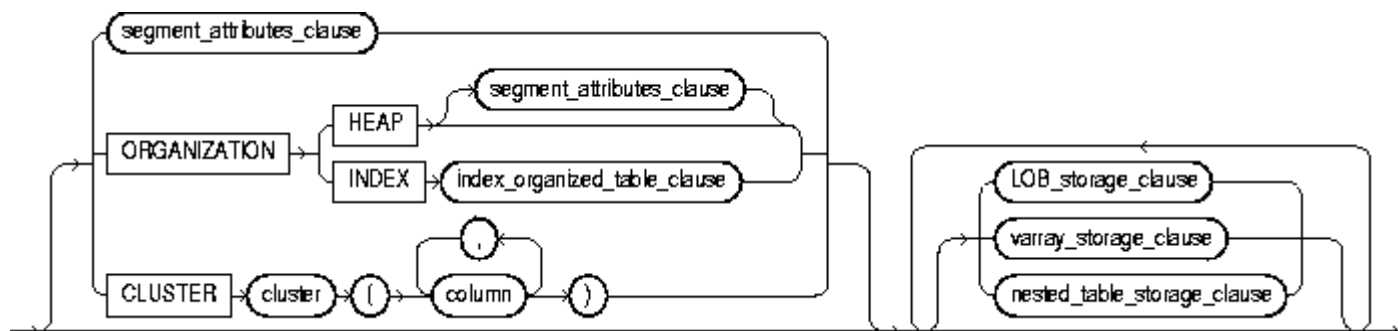
CREATE TABLE command ::=



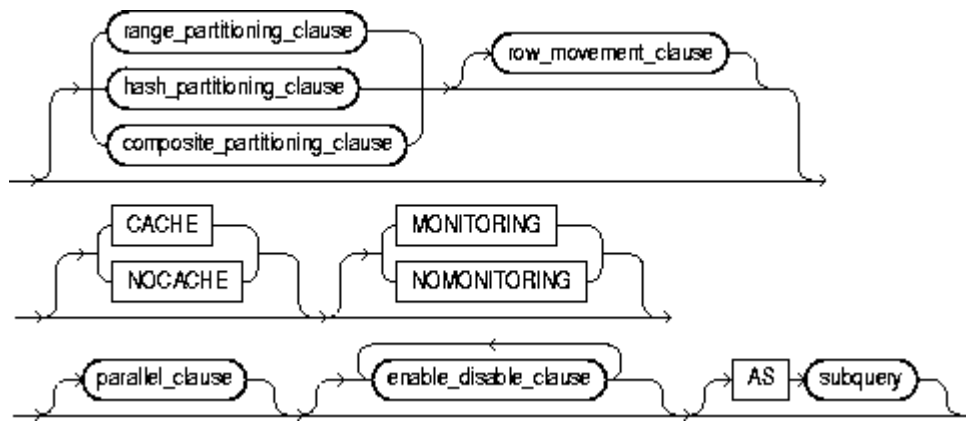
relational_properties::=



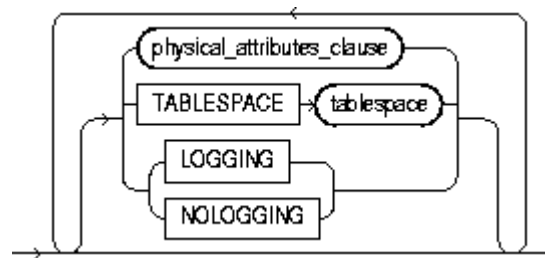
physical_properties::=



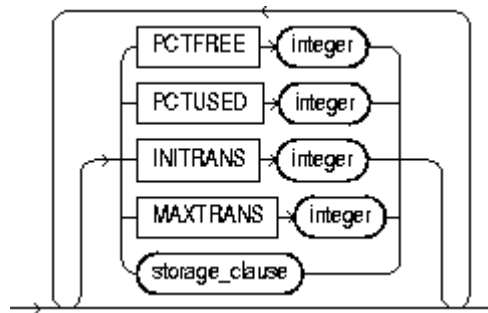
table_properties::=



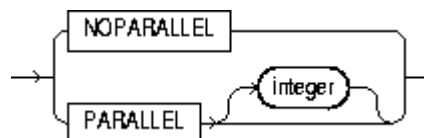
segment_attributes_clause::=



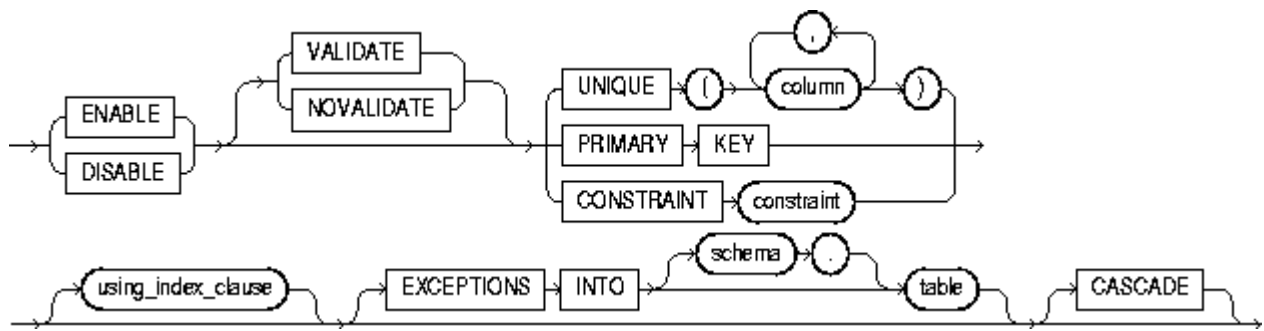
physical_attributes_clause::=



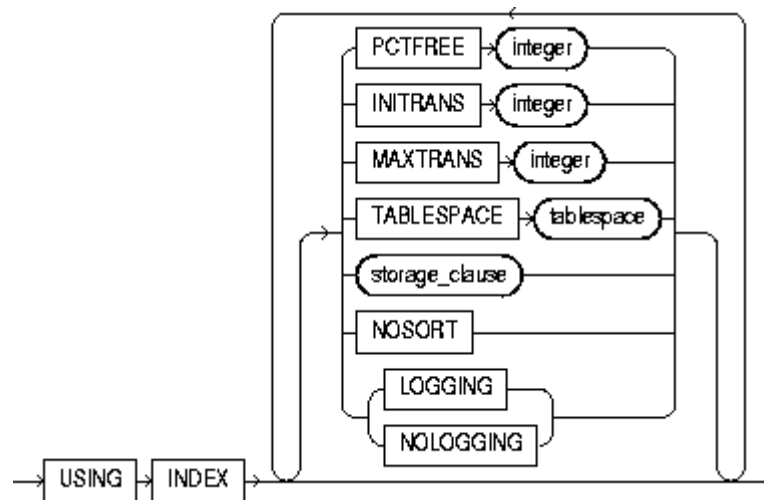
parallel_clause::=



enable_disable_clause::=



using_index_clause::=



Бу ерда

schema	схема номи ;
table	жадвал номи ;
column	жадвал устунининг номи;
datatype	устундаги берилганларнинг тури;
DEFAULT	устунда қўйиладиган қиймат (INSERT вақтида,агар у кўрсатилмаса);
column_constraint	устун даражасидаги чеклагич ;
table_constraint	жадвал даражасидаги чеклагич ;
PCTFREE	келажакда ҳар бир блокда қаторларни янгилаш учун бўш фазонинг улуши (процентда);
PCTUSED	ҳар бир блокнинг минимал ишлатилиши мумкин бўлган жой (% да)
INITRANS	блокга транзакциялар кириши сони
MAXTRANS	блокга транзакциялар кириши

	максимал сони
TABLESPACE	жадвал ташкил қилинадиган жадвалли фазо
STORAGE	жадвалнинг сақлаш параметрлари
CLUSTER	жадвалнинг қайси кластерда ётиши
ENABLE	яхлитлик чеклагични ишчи ҳолатга келтиради
DISABLE	яхлитлик чеклагични ишсиз ҳолатга келтиради
AS subquery	жадвал ташкил қилиниши билан биргаликда сўровости қаторлари орқали тўлдирилади.

Мисоллар :

```

1). CREATE TABLE scott.emp
  ( empno NUMBER CONSTRAINT pk_emp PRIMARY KEY,
    ename VARCHAR2(10) CONSTRAINT nn_ename NOT NULL
                                CONSTRAINT upper_ename
                                CHECK ( ename = UPPER(ename)),
    job   VARCHAR2(9),
    mgr   NUMBER          CONSTRAINT fk_mgr
                                REFERENCES scott.emp(empno))
  PCTFREE 5 PCTUSED 75
2). CREATE TABLE dept
      ( deptno NUMBER (2 ) ,
        dname  VARCHAR 2 (14 ) ,
        loc    VARCHAR 2 (13 ) )

      STORAGE ( INITIAL 100 K NEXT 50 k MINEXTENTS 1
                MAXEXTENTS 50 PCTINCREASE 5 )

3). CREATE ROLLBACK SEGMENT rsone
      STORAGE ( INITIAL 10 K NEXT 10 K MINEXTENTS 2
                MAXEXTENTS 25 OPTIMAL 50 )

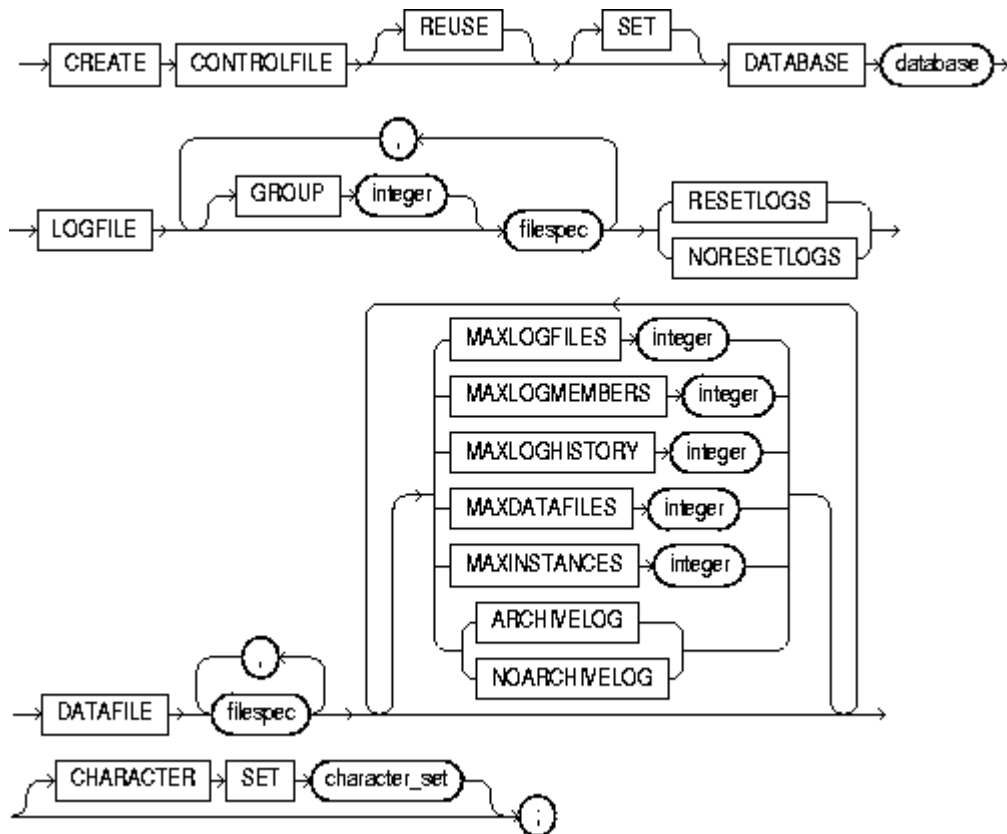
```

k)

Create controlfile

Бошқарувчи файлни қайтадан ташкил қилиш учун ишлатилади

CREATE CONTROLFILE command : : =



Бу ерда :

а) REUSE - CONTROL_FILES параметри орқали аниқланган мавжуд файлларни тозалаб ишлатади;
б) SET DATABASE - берилганлар базасини номини ўзгартиради
в) LOGFILE - журналли файллар гуруҳини кўрсатади
г) RESETLOGS - LOGFILE-да кўрсатилган файлларнинг ичини рад этади берилганлар базаси очилганда қандай фойдаланган бўлса шундай фойдаланишини кўрсатади
е) MAXLOGFILE - журналли файл гуруҳлари максимал сонини кўрсатади
ж) MAXLOGMEMBERS - битта файлли журнал гуруҳидаги максимал файллар сонини кўрсатади
з) MAXLOGHISTORY - архивлаштирилган файл журналлари гуруҳларини максимал сонини кўрсатади.
и) MAXDATAFILES - Берилган файлларининг максимал сони

к) MAXINSTANCES - Очилиши мумкин бўлган берилганлар базасининг экземплярининг максимал сони
л) ARCHIVELOG - Журналли файлларни (қайтадан ишлатиш олдида) архивлаштириш режимини ўрнатади
м) NOARCHIVELOG - л) пунктининг акси

Мисоллар :

```

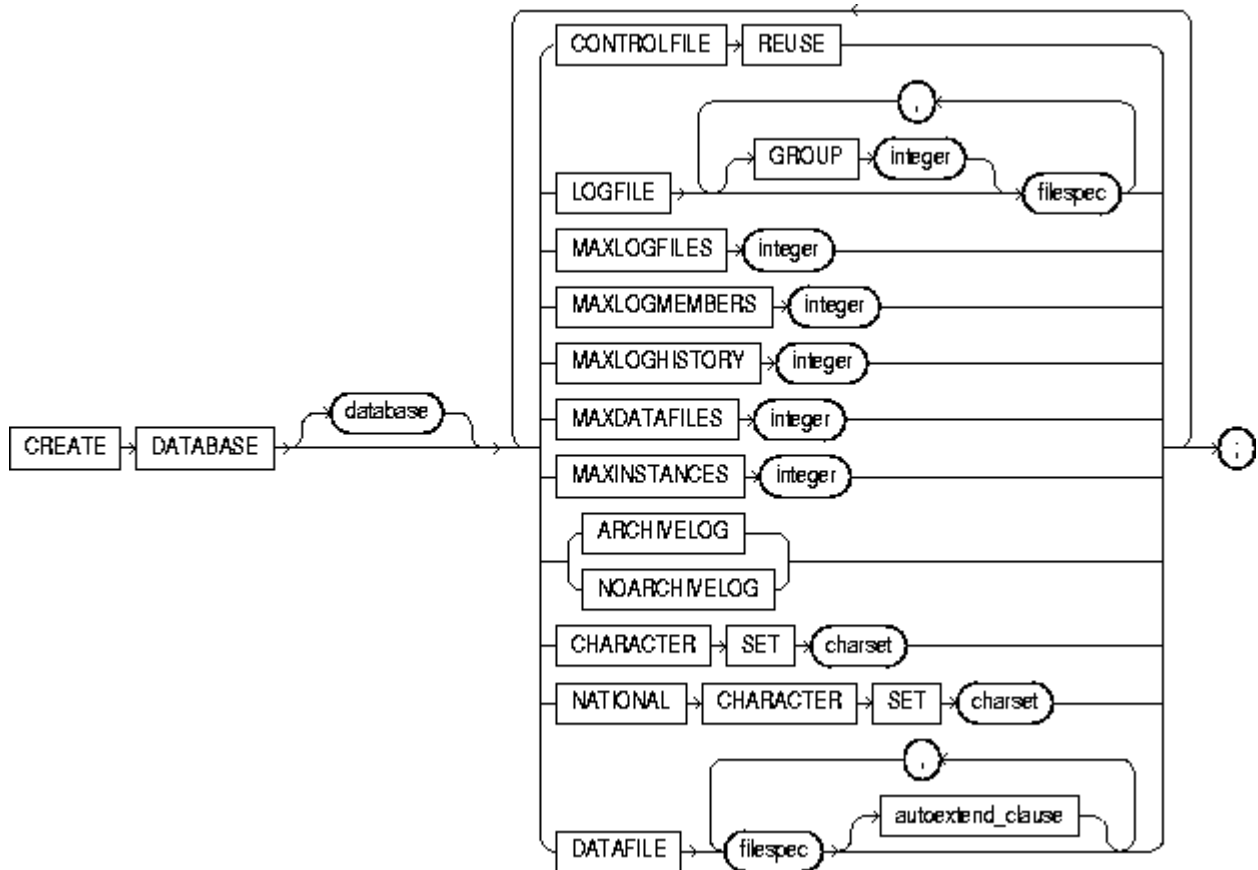
1). CREATE CONTROLFILE REUSE
    SET DATABASE orders_2
    LOGFILE GROUP1 ('diskb:log1.Log', 'diskc:log1.Log') SIZE 50 k
        GROUP2 ('diskb:log2.Log', 'diskc:log2.Log' ) SIZE50 k
    NORESETLOGS
    DATAFILE 'diska:dbone.dat' SIZE 2M
    MAXLOGFILES 5
    MAXLOGHISTORY 100
    MAXDATAFILES 10
    MAXINSTANCES 2
    ARCHIVELOG

```

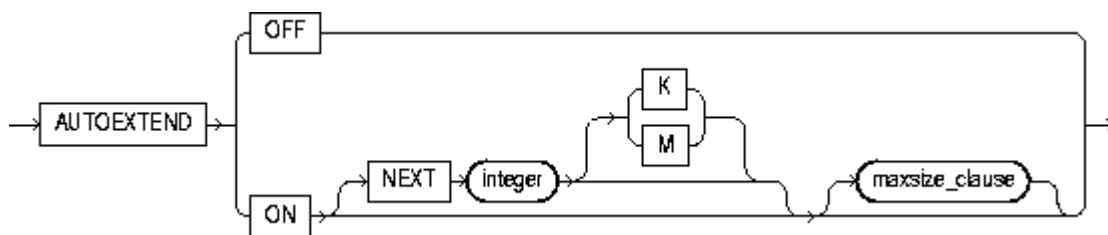
Create database

Берилганлар базасини ташкил қилиш .

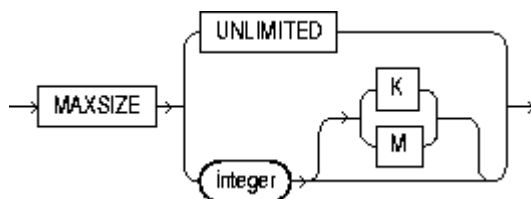
```
2). CREATE DATABASE command :: =
```



autoextend_clause::=



maxsize_clause::=



- а) database - берилганлар базасининг номи (узунлиги 8 байтгача)
- б) CHARACTER SET - белгиларни кодлаштириш схемасини кўрсатади

Мисол : CREATE DATABASE newtest
 CONTROLFILE REUSE
 LOGFILE GROUP1 ('diskb:log1.Log', 'diskc:log1.Log') SIZE 50 k

```

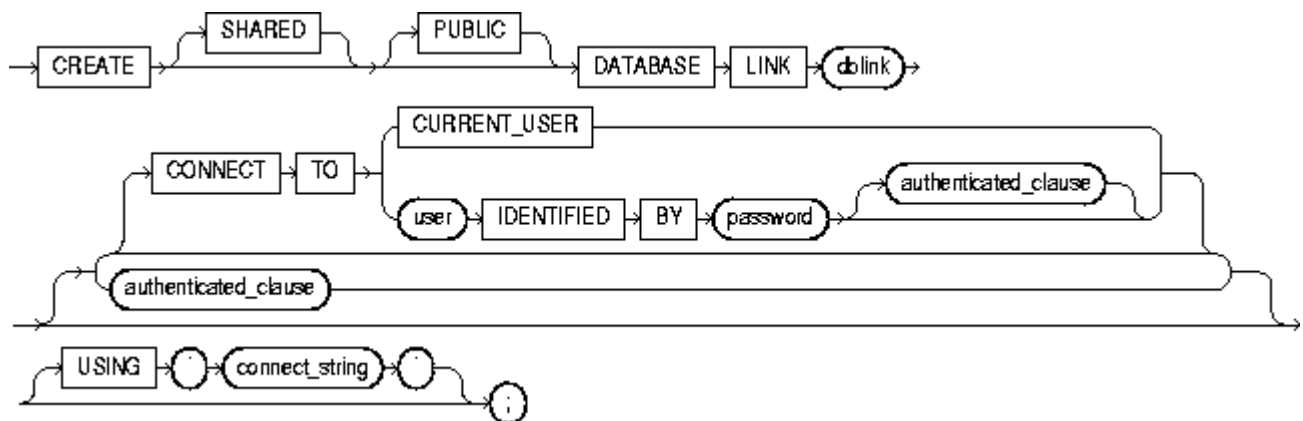
GROUP2 ('diskb:log2.Log', 'diskc:log2.Log' ) SIZE 50 k
MAXLOGFILES 5
MAXLOGHISTORY 100
DATAFILE 'diska:dbone.dat' SIZE 2M
MAXDATAFILES 10
MAXINSTANCES 2
ARCHIVELOG
EXCLUSIVE

```

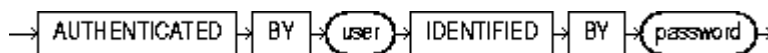
Create database link.

Узоқлашган берилганлар базаси билан боғланиш каналини ташкил қилиш.

CREATE DATABASE LINK ::=



authenticated_clause::=



- а) PUBLIC - умумий алоқа канали
- б) dblink - алоқа каналнинг тўлиқ ёки қисқа номи
- в) USING - узоқланган берилганлар базасининг аниқланиши

Мисоллар:

```

I CREATE DATABASE LINK sales.hd.acme.com
  CONNECT TO Scott IDENTIFIED BY tiger
  USING 'D:BOSTON-MEG'

```

```

II. SELECT * FROM emp@sales.hd.acme.com

```

8.2. Берилганларнинг манипуляция қилиш тили командалари:

қуйида берилганларнинг манипуляция қилиш тили командаларини келтирамиз.

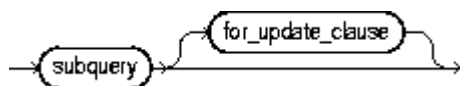
Команда	Изоҳ
DELETE	Жадвалдаги қаторларни ўчириб ташлайди.
EXPLAIN PLAN	SQL жумласининг бажарилиш планини қайтаради .
INSERT	Жадвалга. янги қатор қўшади
LOCK TABLE	Жадвални бошқа фойдаланувчилар учун маҳкамлайди.
SELECT	Жадваллардан қаторларни танлаб олиш .
UPDATE	Жадвалдаги қаторлар қийматини ўзгартириш.

Ушбу жадвалда келтирилган айрим командаларни изоҳлаб ўтамиз.

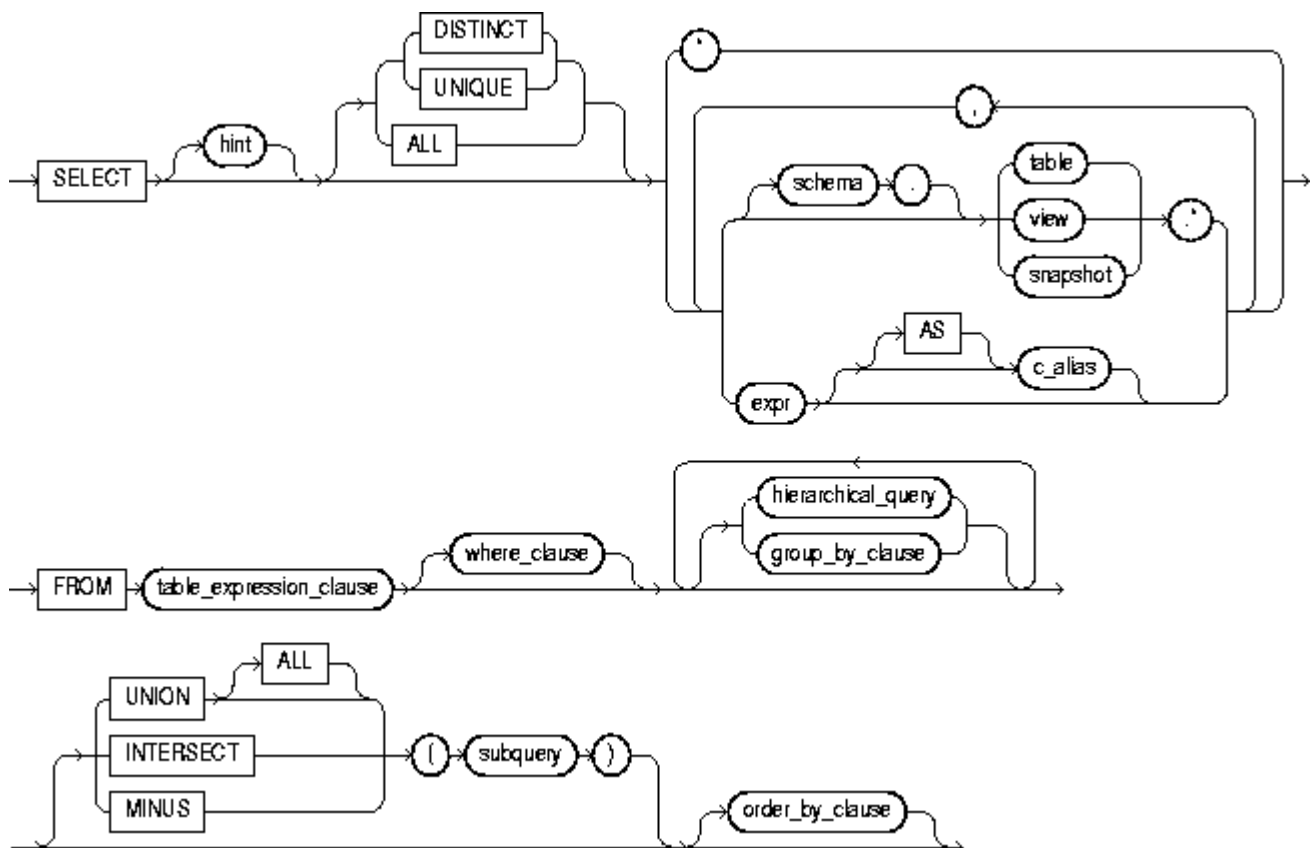
Select.

Берилганлар базасидан берилганларни танлаб чиқариш учун ишлатилади

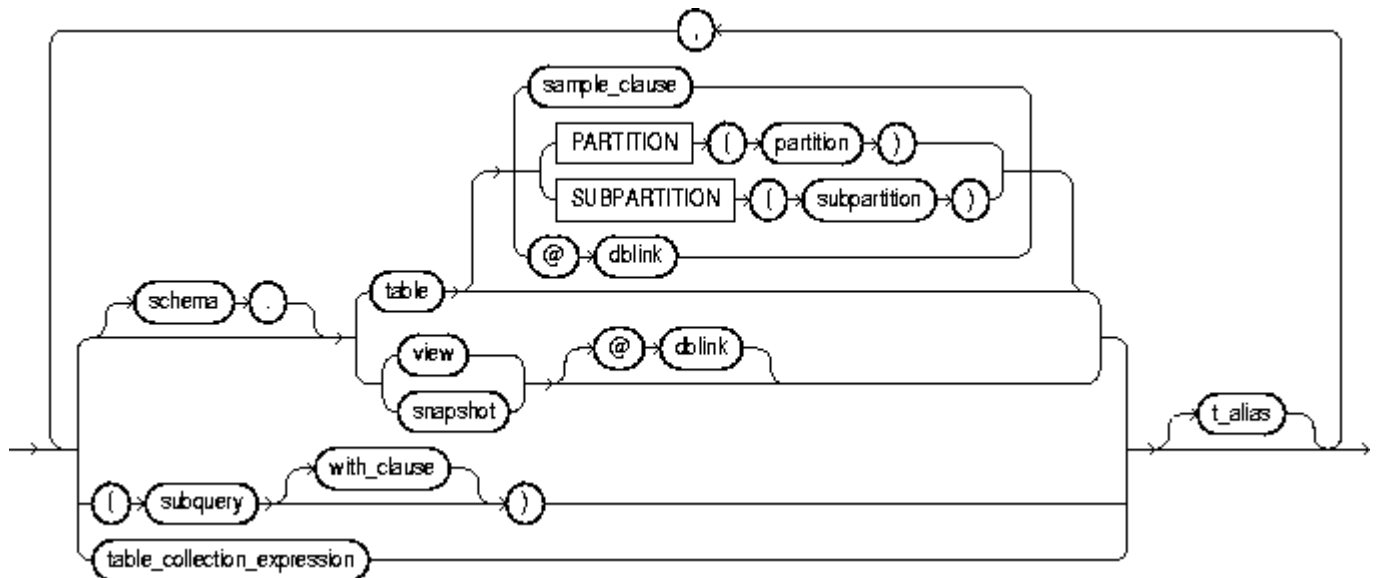
SELECT command ::=



subquery::=



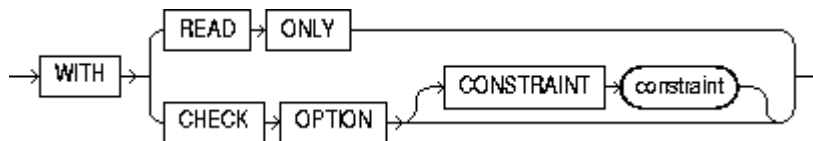
table_expression_clause::=



sample_clause::=



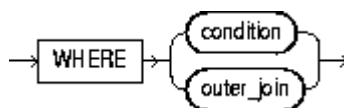
with_clause::=



table_collection_expression::=



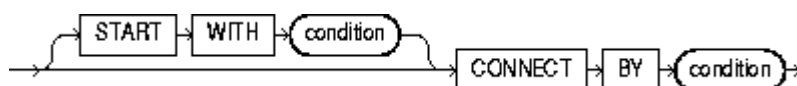
where_clause::=



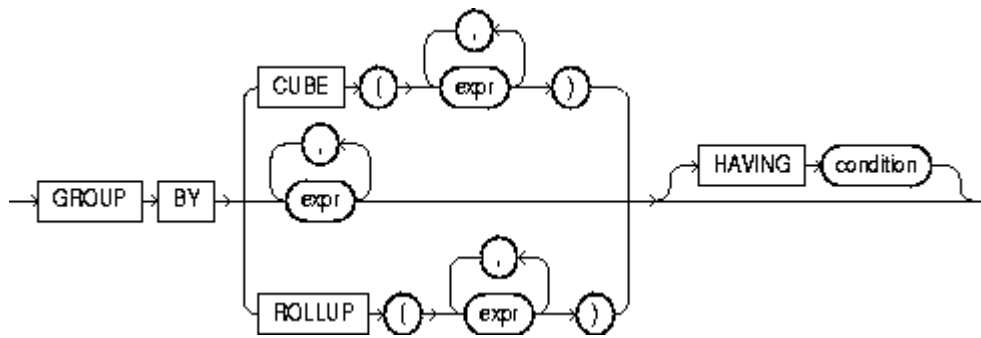
outer_join::=



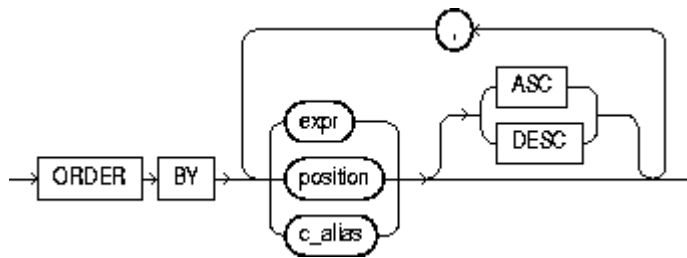
hierarchical_query_clause::=



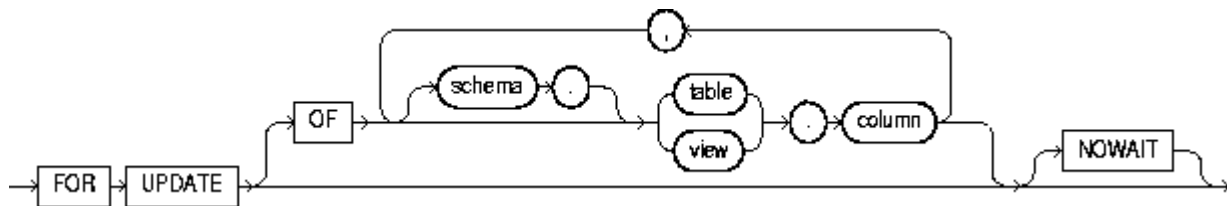
group by clause::=



order by clause::=



for update clause::=



Insert.

Берилганлар базасидаги жадвалга янги қатор қўшиш учун ишлатилади
`INSERT command :: =`

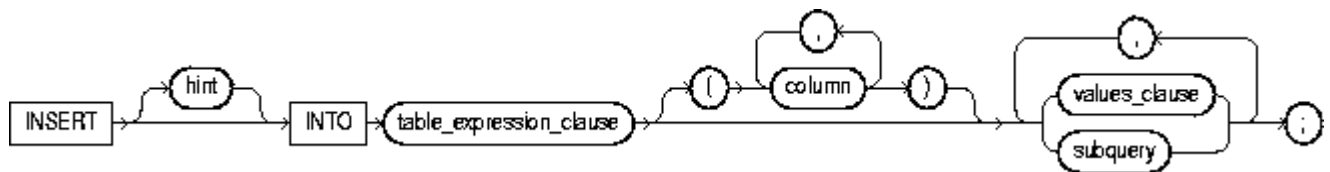
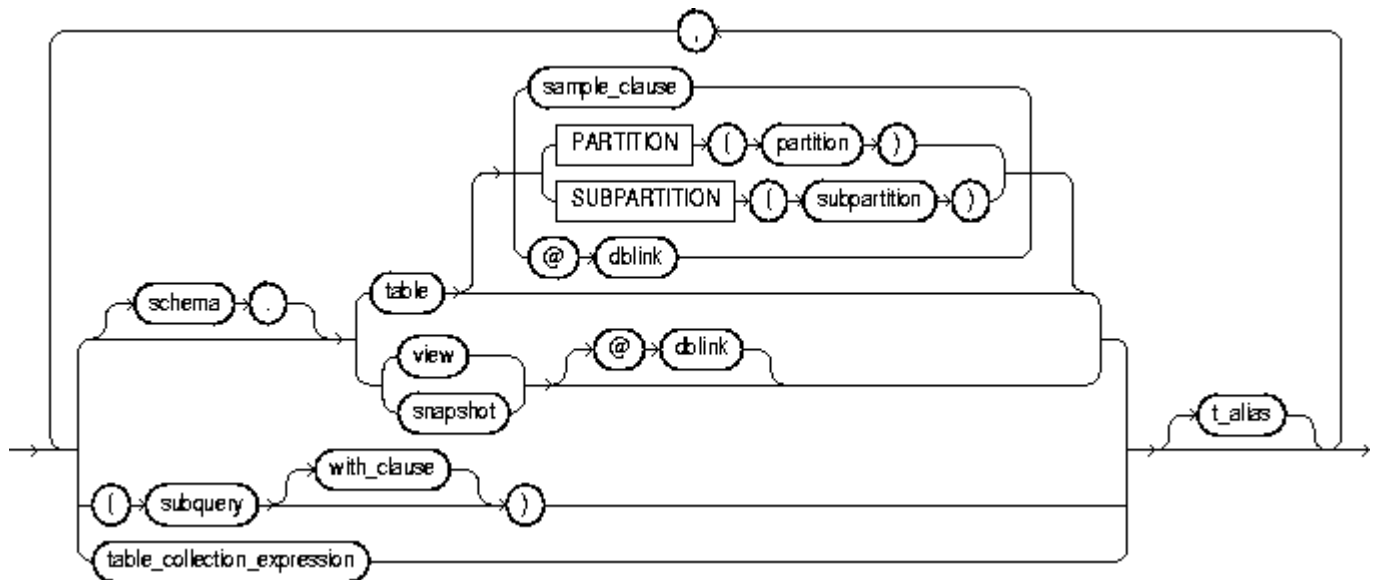


table expression clause::=



with clause::=

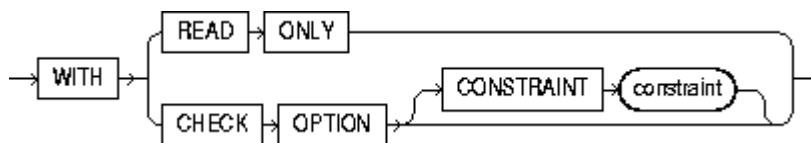
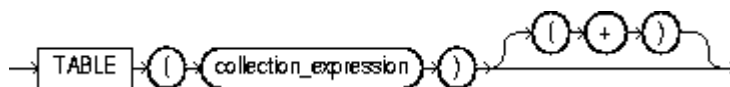
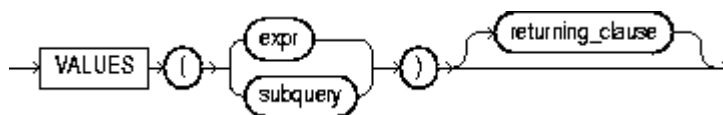


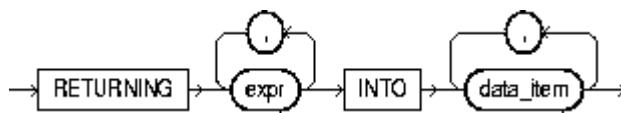
table collection expression::=



values clause::=



returning clause::=

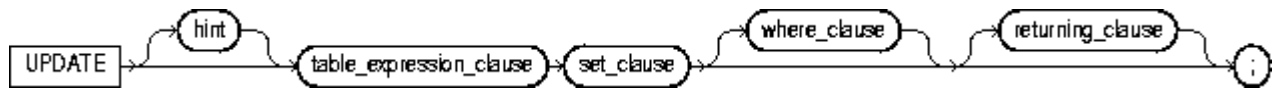


Мисол 1. INSERT INTO dept
VALUES (50 , 'PRODUCTION' , 'SAN FRANCISCO ')

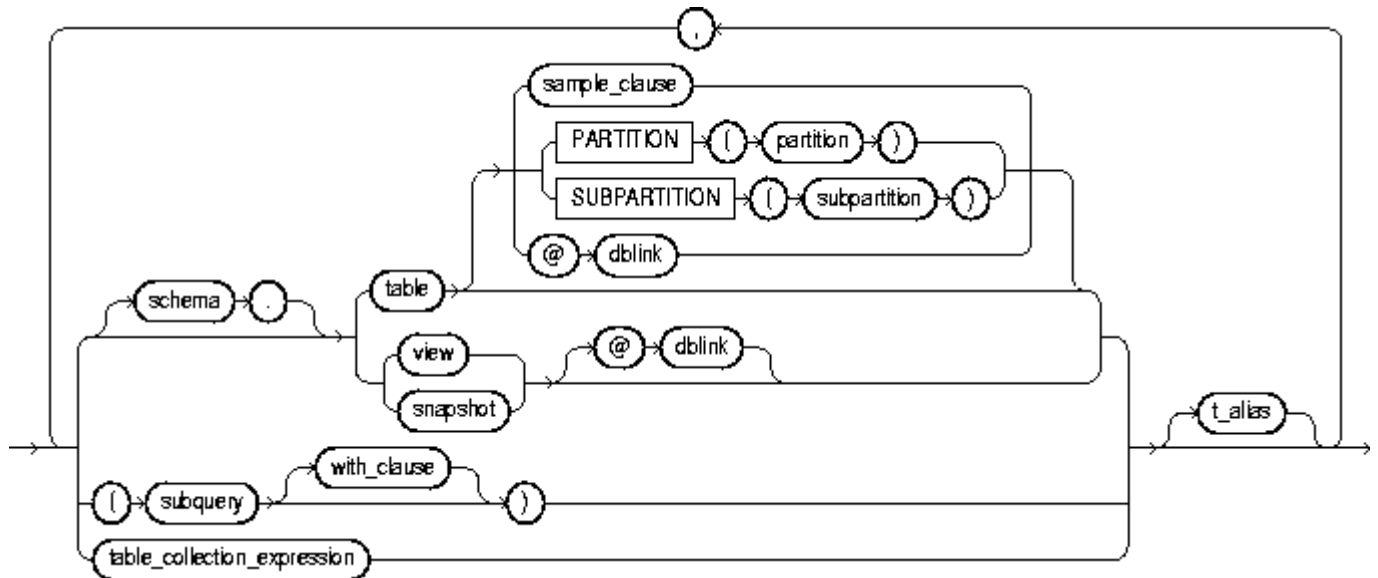
Update.

Берилганлар базасидаги жадвал қаторини янгилаш учун ишлатилади.

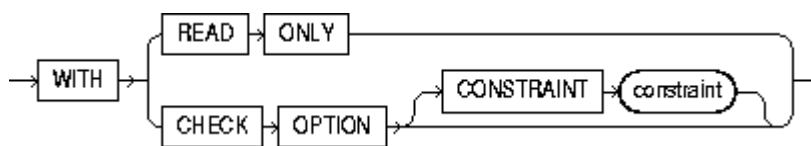
UPDATE command ::=



table_expression_clause::=



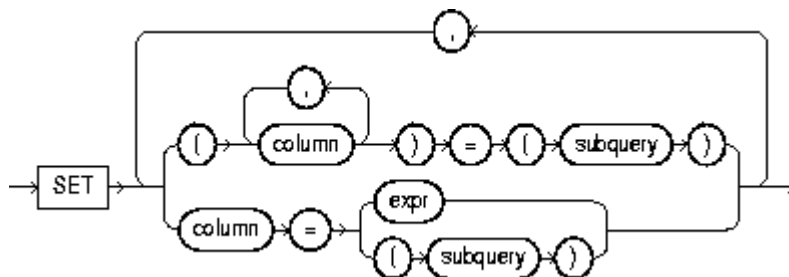
with_clause::=



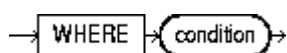
table_collection_expression::=



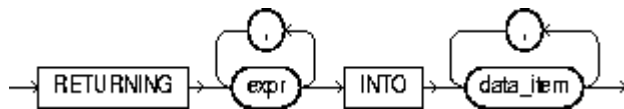
set_clause::=



where_clause::=



returning_clause::=



Delete.

Берилганлар базасидаги жадвал каторини ўчириш учун ишлатилади.

DELETE command ::=

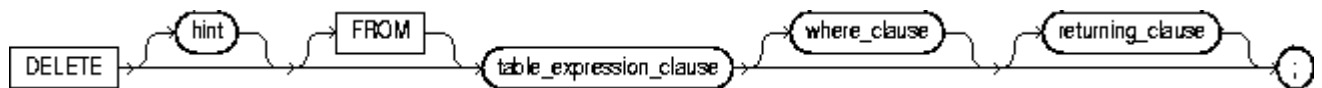
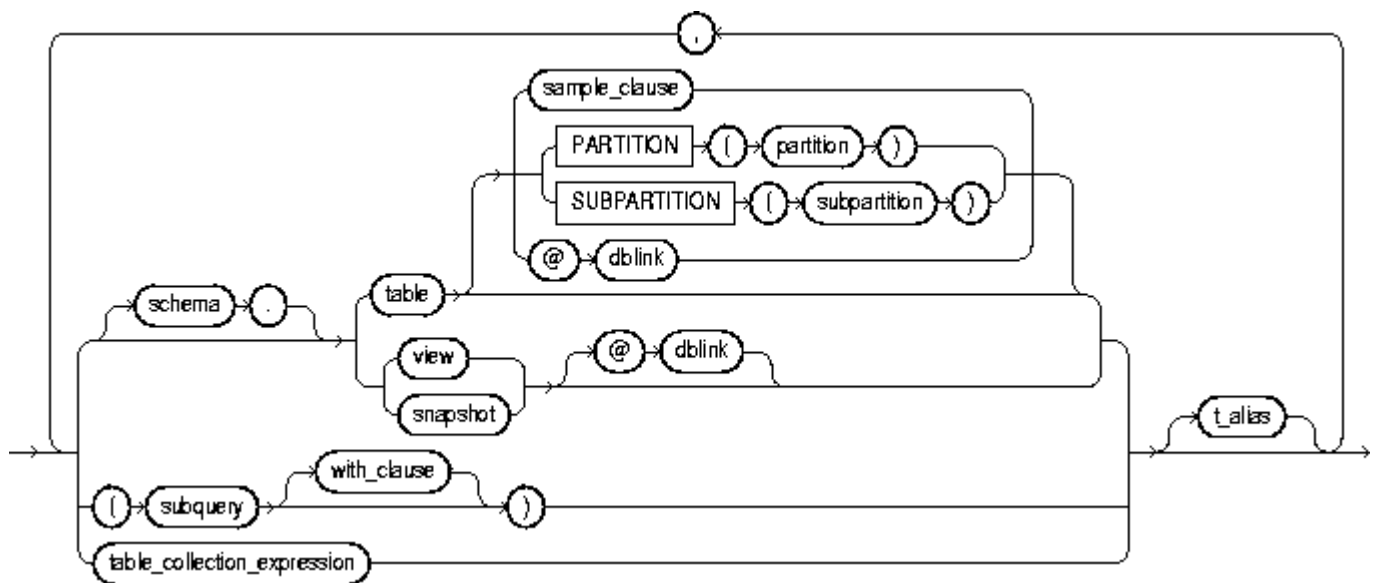


table expression clause ::=



with clause ::=

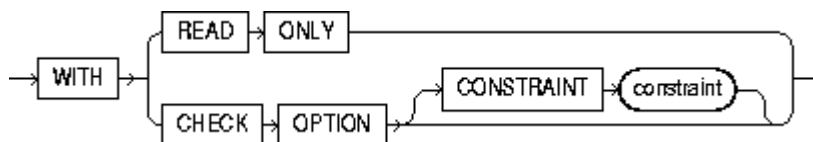
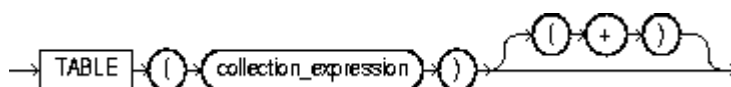
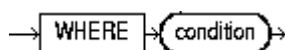


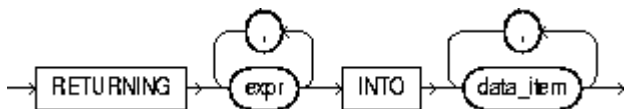
table collection expression ::=



where clause ::=



returning_clause::=



8.3. Транзакцияларни бошқариш командалари:

муайида транзакцияларни бошқариш командаларини келтирамиз.

Команда	Изоҳ
COMMIT	Охирги транзакция амалга оширган барча ўзгаришларни базага фиксирлайди.
ROLLBACK	Охирги транзакция амалга оширган барча ўзгаришларни бекор қилади.
SAVEPOINT	майтиш нуқтасини белгилайди.
SET TRANSACTION	Оний транзакция хоссаларини белгилайди.

8.4. Сеансни бошқариш командалари:

муайида сеансни бошқариш командаларини келтирамиз.

Команда	Изоҳ
ALTER SESSION	Сессия параметрларини ўзгартириш.
SET ROLE	Оний сессияда ролни ишчи ёки ишсиз ҳолатга келтириш.

Set role.

SET ROLE командаси оний сессияда ролни ишчи ёки ишсиз ҳолатга келтириш учун ишлатилади

Мисоллар:

- SET ROLE gardener IDENTIFIED BY marigolds

marigolds пароли орқали аниқланган gardener ролни оний сеансда ишчи ҳолатга келтиради.

- SET ROLE ALL

Барча ролларни оний сеансда ишчи ҳолатга келтиради.

- SET ROLE ALL EXCEPT banker

banker ролдан ташқари барча ролларни оний сеансда ишчи ҳолатга келтиради

- SET ROLE NONE

Барча ролларни оний сеансда ишсиз ҳолатга келтиради

8.5. Системани бошқариш командалари.

қуйида системани бошқариш командаларини келтирамиз.

Команда	Изоҳ
ALTER SYSTEM	Oracle7 экземплярини ўзгартиради.

8.6. Ички қурилган SQL командалари.

Команда	Изоҳ
ALLOCATE	Курсор ўзгарувчисини ўрнатиш .
CLOSE	Курсорни ёпиш.
CONNECT	Oracle7 экземплярига уланиш.
DECLARE CURSOR	Курсорни эълон қилиш.
DECLARE DATABASE	Узоқланган база номини эълон қилиш.
DECLARE STATEMENT	SQL жумласига SQL ўзгарувчиси номини тайинлаш.
DECLARE TABLE	Жадвални эълон қилиш.
DESCRIBE	Дескрипторни тасвирлаш.
EXECUTE	SQL жумласи ёки PL/SQL блокини бажариш.
EXECUTE IMMEDIATE	Host ўзгарувчисига эга бўлмаган SQL сжумласини бажариш.
FETCH	Сўров орқали танлаб олинган қаторларни очиш.
OPEN	Курсорни очиш.
PREPARE	SQL жумласини анализ қилиш.
TYPE	Фойдаланувчи аниқлаган «equivalencing»-ни бажариш.
WHENEVER	Хатоликларга ишлов беришни аниқлаш.

PL/SQL ORACLE

1. PL/SQL АСОСЛАРИ

PL/SQL процедурали тил бўлиб, у қуйидаги асосий элементлардан иборат:

- 1) белгилар тўплами;
- 2) захиранган сўзлар;
- 3) чеклагичлар;
- 4) берилганлар турлари;
- 5) синтаксис қоидалари;
- 6) операторларни ёзиш ва ишлатиш қоидалари.

Ушбу асосий элементлар ҳақида батафсил алоҳида-алоҳида тўхталиб ўтамиз.

Белгилар тўплами

А) Лотин кичик ва катта ҳарфлари:

A...Z, a ...z;

Б) 0 ... 9 рақамлари;

В) табулятор, бўш жой, курсорни қайтариш;

Г) () + - * / < > = ! ~ ; : . ' @ % , " # \$ ^ & _ | { } ? [] белгилари.

Лексемалар

PL/SQL матни қаторлари лексемалардан (белгилар гуруҳи) таркиб топган:

- Чеклагичлар;
- Идентификаторлар;
- Литераллар;
- Изоҳлар.

Масалан

`bonus: = salary * 0.10;` - - мукофот пулини ҳисоблаш қатори қуйидаги лексемалардан таркиб топган:

- `bonus` ва `salary` -идентификаторлар
- `:` `=` -мураккаб чеклагич
- `*` ва `;` -содда белгилар
- `0.10` -сонли литерал
- `--`Премияни ҳисоблаш -изоҳ.

қуйидаги қатор эса нотўғри ёзилган:

```
count := count + 1;  -- нотўғри
```

чунки унда (:=) мураккаб белгилар орасида бўш жой қолдирилган.

Тузилмани кўрсатиш учун қаторни бир неча қаторга бўлишингиз мумкин:

```
IF x>y THEN max:=x;ELSE max:=y;END IF;|    IF x > y THEN
|                                     max:= x;
|    ELSE
|                                     max := y;
|    END IF;
```

Чеклагичлар

Чеклагичлар PL/SQL махсус маънога эга бўлган

- Содда белгилар
- Мураккаб белгилар

иборат.

Содда белгилар

Содда белгилар битта белгидан иборат бўлиб улар қуйида келтирилган рўйхатдан таркиб топган:

- + қўшиш амали;
- айириш еки унар минус амали;
- * кўпайтириш амали;
- / бўлиш амали;
- = муносабат амали;
- < муносабат амали;
- > муносабат амали;
- (ифода еки рўйхат чеклагичи;
-) ифода еки рўйхат чеклагичи;
- ; оператор тугаши аломати;
- % атрибут аломати;

- , рўйхатда элементларни ажратгичи;
- . ҳаднинг селектори;
- @ узоқланган кириш ҳуқуқи / аломати;
- ‘ белгилар қаторининг чеклагичи;
- “ куштирноқдаги идентификаторнинг чеклагичи;
- : базали тил узгарувчисининг аломати;

Мураккаб белгилар

Мураккаб белгилар иккита белгилардан иборат:

- ** даражага кўтариш амали;
- <> муносабат амали;
- != муносабат амали;
- ~= муносабат амали;
- ^= муносабат амали;
- <= муносабат амали;
- >= муносабат амали;
- := қиймат бериш оператори;
- => боғлаш оператори;
- .. диапазон амали;
- || конкатенация операцияси;
- << метка чеклагичи;
- >> метка чеклагичи;
- бир қаторли изоҳнинг аломати;
- /* кўп қаторли изоҳнинг боши;
- */ кўп қаторли изоҳнинг охири.

Идентификаторлар

Ўзгармас, ўзгарувчи, мустасно (исключение), курсор, қисмдастур ва пакетларни номлаш учун ишлатилади. Идентификатор ҳарфдан бошланади ва

унда @, - , / бўш жойга ўхшаган белгилар қатнашиши мумкин эмас.

Идентификатор узунлиги 30 ошмаслиги лозим. Намуналар келтирамиз:

x

t2

phone#

credit_limit

LastName

oracle\$number

қуйидагилар эса:

mine&yours -- нотўғри (амперсанд қатнашяпти)

debit-amount -- нотўғри (минус қатнашяпти)

on/off -- нотўғри (слеш қатнашяпти)

user id -- нотўғри (бўш жой қатнашяпти)

идентификатор бўлолмайдилар.

Захираланган сўзлар

PL/SQL да захираланган сўзлар махсус вазифаларни бажаради ва шу сабабли уларни қайтадан аниқлаш мумкин эмас.

Мисоллар: DECLARE , BEGIN,

Олдиндан аниқланган идентификаторлар

STANDARD пакетида эълон қилинган глобал (INVALID_ NUMBER мустасно каби) идентификаторларни қайтадан эълон қилиш қилиш мумкин. Аммо бу хатоликга олиб келиши мумкин, чунки локал эълонлар (юқори приоритетга эга) глобал эълонлар устида ёзилади.

Қўштирноқ ичидаги идентификаторлар

PL/SQL идентификаторларни қўштирноқ ичига олишга имкон беради ва бу қатор қулайликларга эга. Масалан белги сифатида энди бўш жойни ҳам ишлатиш мумкин, лекин иккиланган қўштирноқни ишлатиш мумкин эмас. Намуналар келтирамиз:

"X+Y"

"last name"

```
"on/off switch"
"employee(s) "
"*** header info ***"
```

Литераллар

Литераллар сонли, белгили, қаторли еки бўл бўлиши мумкин.

Сонли литераллар

Арифметик ифодаларда икки хил турдаги бутун ва ҳақиқий сонли литераллардан фойдаланиш мумкин. Масалан:

```
030 6 -14 0 +32767
6.6667 0.0 -12.0 3.14159 +8300.00 .5 2E5 1.0E-7 3.14159e0 -1E38 -
9.5e-3
5E3 = 5 10**3 = 5 1000 = 5000
5E-3 = 5 10**-3 = 5 0.001 = 0.005
```

Белгили литераллар

Белгили литерал қўштирноқ ичига олинган ягона белгидан иборат:

```
'Z' '%' '7' ' ' 'z' '('
```

Қаторли литераллар:

Қаторли литераллар қўштирноқ ичига олинган белиглар тўпламидан иборат:

```
'Hello, world!'
'XYZ Corporation'
'10-NOV-91'
'He said "Life is like licking honey from a thorn."'
'$1,000,000'
```

Бул литераллар

Бул литераллари олдиндан аниқланган TRUE ва FALSE, ҳамда NULL қийматларидан иборат

Изоҳлар

Изоҳлар дастурни изоҳлаш учун ишлатилади. Изоҳни PL/SQL компилятори инкор қилади. Икки хил турдаги изоҳ мавжуд:

- бир қаторли
- кўп қаторли.

Бир қаторли изох

Бир қаторли изох қатордаги ихтиёрий жойидан иккита минус белгисидан башланиб қатор охиригача давом этади. Мисоллар келтирамиз:

```
-- begin processing
SELECT sal INTO salary FROM emp -- get current salary
WHERE empno = emp_id;
bonus := salary * 0.15; -- compute bonus amount
```

Кўп қаторли изох

Кўп қаторли изох (/*) слеш-юлдузча билан бошланади ва (*/) юлдузча слеш билан тугайди. Мисоллар келтирамиз:

```
BEGIN
    /* Compute a 15% bonus for top-rated employees. */
    IF rating > 90 THEN
        bonus := salary * 0.15 /* bonus is based on salary
*/
    ELSE
        bonus := 0;
    END If;
    ...
    /*
```

2. Берилганлар турлари

Хар қандай ўзгармас ва ўзгарувчи уларнинг сақлаш формати, чеклагичлари ва ўзгариш соҳасини аниқлаб берувчи берилганлар турига эга бўлади.

PL/SQL берилганлар турлари

Скаляр турлар

Мураккаб турлар

BINARY_INTEGER DEC DECIMAL DOUBLEPRECISION FLOAT INT INTEGER NATURAL NATURALN NUMBER NUMERIC PLS_INTEGER POSITIVE POSITIVEN REAL SIGNTYPE SMALLINT	CHAR CHARACTER LONG LONG RAW NCHAR NVARCHAR2 RAW ROWID STRING VARCHAR VARCHAR2 BOOLEAN DATE	RECORD TABLE VARRAY Хавола турлари REF CURSOR REF object_type LOB турлари BFILE BLOB CLOB NCLOB
--	---	---

Number турлари

Number турлари сонли берилганларни сақлашга имкон беради.

BINARY_INTEGER

BINARY_INTEGER ишорали бутун сонлар (-2147483647 .. 2147483647) интервалда иккилик санок системада сақланади. Ишорани ўзгартирмасдан ҳисоблаш жараенида ишлатиш мумкин. NUMBER туридан камроқ жой талаб қилади.

BINARY_INTEGER тур остилари:

PL/SQL тилида қуйидаги тур остилар аниқланган:

NATURAL

NATURALN

POSITIVE

POSITIVEN

SIGNTYPE

NATURAL ва POSITIVE тур остилари мос равишда манфий бўлмаган ва мусбат бутун ўзгарувчиларни тавсифлайди. NATURALN ва POSITIVEN эса бутун ўзгарувчилар қиймати олдига нул қўйишни тақиқлайди. SIGNTYPE тури $-1, 0$ ва 1 қийматга эга бўлган бутун ўзгарувчиларни тавсифлайди.

NUMBER

NUMBER тури сузувчи еки фиксирланган нуқтали сонларни ифодалаш учун ишлатилади. У қуйидаги синтаксисга эга

NUMBER (аниқлик, масштаб)

Бу ерда аниқлик умумий рақамлар белгилар сони (≤ 38) ва $1.OE\ 129 \dots 9.99E\ 125$ қийматларни қабул қилиши мумкин. Масштаб эса ўнлик нуқтадан кейинги рақамлар сони бўлиб – 84 дан 127гача ўзгаради.

NUMBER тур остилари

Сиз қуйидаги NUMBER тур остиларини ишлатишингиз мумкин:

DEC

DECIMAL

DOUBLE

PRECISION

FLOAT

INTEGER

INT

NUMERIC

REAL

SMALLINT

DEC, DECIMAL ва NUMERIC тур остиларини максимал аниқлиги 38 ўнлик рақамдан иборат фиксирланган нуқтали сонларни ифодалаш учун ишлатишингиз мумкин.

INTEGER, INT ва SMALLINT тур остиларини эса максимал аниқлиги 38 ўнлик рақамдан иборат бутун сонларни ифодалаш учун ишлатишингиз мумкин.

DOUBLE PRECISION ва FLOAT тур остиларини максимал аниқлиги 38 ўнлик рақамдан иборат сузувчи – нуқтали сонларни ифодалаш учун ишлатишингиз мумкин.

REAL тур остини максимал аниқлиги 18 ўнлик рақамдан иборат сузувчи – нуқтали сонларни ифодалаш учун ишлатишингиз мумкин.

PLS_INTEGER

PLS_INTEGER тури -2147483647 .. 2147483647 интервалдаги ишорали бутун сонларни ифодалаш учун ишлатилади.

Белгили турлар CHAR

CHAR – фиксирланган узунликдаги белгили берилганларни ифодалайди.
Унинг синтаксиси:

CHAR [(максимал узунлиги)],

Максимал узунлиги CHAR турдаги базавий устунлар учун ≤ 2000 байт.
Мумкин бўлган максимал узунлиги ≤ 32767 .

Тур остилари: CHARACTER STRING

VARCHAR2

VARCHAR2 - ўзгарувчи узунликдаги белгили берилганларни ифодалайди.
Унинг синтаксиси:

VARCHAR 2 (максимал узунлиги)

Бу ерда максимал узунлиги VARCHAR2 турдаги базавий устунлар учун ≤ 2000 байт. Мумкин бўлган максимал узунлиги ≤ 32767 .

Тур остилари: VARCHAR

STRING

LONG

LONG - ўзгарувчи узунликдаги белгили қаторларни ифодалайди. Унинг синтаксиси:

LONG (максимал узунлиги)

Бу ерда максимал узунлиги LONG турдаги базавий устунлар учун ≤ 2147483647 байт. Максимал узунлиги LONG турдаги ўзгарувчилар учун ≤ 32760 байт.

LONG RAW

LONG RAW - ўзгарувчи узунликдаги бинар берилганларини ва байтли қаторларни ифодалайди. Унинг синтаксиси:

LONG RAW (максимал узунлиги)

Бу ерда максимал узунлиги LONG RAW турдаги базавий устунлар учун ≤ 2147483647 байт. Максимал узунлиги LONG RAW турдаги ўзгарувчилар учун ≤ 32760 байт. LONG RAW турдаги берилганларни PL/SQL томонидан интерпретация қилинмайди.

RAW

RAW - ўзгарувчи узунликдаги бинар берилганларини ва байтли қаторларни ифодалайди. Унинг синтаксиси:

RAW (максимал узунлиги)

Бу ерда максимал узунлиги RAW турдаги базавий устунлар учун ≤ 2000 байт. Максимал узунлиги RAW турдаги ўзгарувчилар учун ≤ 32760 байт. RAW турдаги берилганларни PL/SQL томонидан интерпретация қилинмайди. Расмлар ва графикларни сақлаш учун қулайдир.

BOOLEAN

BOOLEAN тури TRUE, FALSE ва NULL қийматларни ифодалаш учун ишлатилади.

DATE

DATE тури 1 январь 4717 эрамиздан олдин 31 декабрь 4712 йилгача фиксирланган узунликдаги санани ифодалаш учун ишлатилади.

ROWID

ROWID – псевдоустун бўлиб бир қийматли равишда қаторни ифодалайди.

Унинг формати BBBB BBBB.RRRR.FFFF.

Бу ерда

BBBB BBBB – берилганлар базасидаги блокнинг номери

RRRR - блокда қаторнинг тартиб номери

FFFF - берилганлар базасидаги файлнинг номери.

(Хаммаси 16 лик системада)

Мисол. 0000000E . 000A. 0007. псевдоустун жисмонан

7-нчи файл

15-нчи блок

11-нчи қатор

сақланади.

NLS белгили турлари

NLS белгили турлари миллий тиллардаги белгилар тўпламини ифодалаш учун ишлатилади. NCHAR ва NVARCHAR2 буларга мисол бўла олади.

LOB турлари

LOB (large object) турлари BFILE, BLOB, CLOB ва NCLOB иборат бўлиб (матн, расм, видеоклип ва товуш каби) тузилмаланмаган берилганларни ифодалаш учун ишлатилади. Унинг ўлчови 4 ГБ ошмаслиги лозим.

BFILE

BFILE берилганлар тури катта ҳажмдаги бинар объектларни базадан ташқарида жойлашган операцион тизим файлларида сақлаш учун ишлатилади. Хар бир BFILE ўзгарувчиси файл локаторини сақлайди. Файл локатори серверда жойлашган катта бинар файл ҳақида маълумотдан таркиб

топган. Ушбу маълумот файл тўлиқ йўли номини аниқловчи директория ҳамроҳидан иборат. BFILE фақат ўқиб бўлади холос.

BLOB

BLOB берилганлар тури катта ҳажмдаги бинар объектларни базада сақлаш учун ишлатилади. Хар бир BLOB ўзгарувчиси локаторни сақлайди. Локатор катта бинар объект ҳақидаги маълумотдан таркиб топган. BLOB транзакцияларда қатнашиши мумкин ва уларни DBMS_LOB ёки OCI пакетлари ёрдамида ўзгартириш мумкин.

CLOB

CLOB берилганлар тури катта ҳажмдаги бир байтли белгили берилганлар блокини базада сақлаш учун ишлатилади. Хар бир CLOB ўзгарувчиси локаторни сақлайди. Локатор катта ҳажмдаги бир байтли белгили берилганлар блоки ҳақидаги маълумотдан таркиб топган. CLOB транзакцияларда қатнашиши мумкин ва уларни DBMS_LOB ёки OCI пакетлари ёрдамида ўзгартириш мумкин.

NCLOB

NCLOB берилганлар тури катта ҳажмдаги кўп байтли белгили NCHAR турдаги берилганлар блокини базада сақлаш учун ишлатилади. Хар бир NCLOB ўзгарувчиси локаторни сақлайди. Локатор катта ҳажмдаги кўп байтли белгили NCHAR турдаги берилганлар блоки ҳақидаги маълумотдан таркиб топган. NCLOB транзакцияларда қатнашиши мумкин ва уларни DBMS_LOB ёки OCI пакетлари ёрдамида ўзгартириш мумкин.

Фойдаланувчи томонидан аниқланган турлар

PL/SQL блоки, қисмдастур ва пакетларнинг эълон қилиш қисмида сизх ўзингизни туростиларингизни аниқлашингиз мумкин. Намуналар келтирамиз:

```
DECLARE
```

```
    SUBTYPE BirthDate IS DATE NOT NULL; -- based on DATE
type
```

```

    SUBTYPE Counter IS NATURAL;    -- based on NATURAL
subtype
    TYPE NameList IS TABLE OF VARCHAR2(10);
    SUBTYPE DutyRoster IS NameList;    -- based on TABLE
type
    TYPE TimeRec IS RECORD (minutes INTEGER, hours
INTEGER);
    SUBTYPE FinishTime IS TimeRec;    -- based on
RECORD type
    SUBTYPE ID_Num IS emp.empno%TYPE;    -- based on
column type
    CURSOR c1 IS SELECT * FROM dept;
    SUBTYPE DeptFile IS c1%ROWTYPE;    -- based on
cursor rowtype

```

Аммо, тур ўлчовини кўрсатиш мумкин эмас. Масалан қуйидаги эълонлар нотўғри бўлади:

```

DECLARE
    SUBTYPE Accumulator IS NUMBER(7,2); -- illegal; must
be NUMBER
    SUBTYPE Delimiter IS CHAR(1);    -- illegal; must
be CHAR

```

Фойдаланувчи томонидан аниқланган турлар одатдагидек ишлатилади:

```

DECLARE
    SUBTYPE Counter IS NATURAL;
    rows Counter;

DECLARE
    SUBTYPE Accumulator IS NUMBER;
    total Accumulator(7,2);

DECLARE
    temp NUMBER(1,0);

```

```

SUBTYPE Scale IS temp%TYPE;
x_axis Scale;  -- magnitude range is -9 .. 9
y_axis Scale;
BEGIN
    x_axis := 10;  -- raises VALUE_ERROR

```

Берилганлар турларини ўзгартириш

Баъзан қийматларни бир турдан бошқа турга айлантиришга тўғри келади. Бу ишни ошкор ёки ошқормас тарзда амалга ошириш мумкин. Ошкор усулда ички қурилган функциялардан фойдаланиш мумкин.

Ошқормас тарзда ўзгаритириш жадвали.

	BIN_INT	CHAR	DATE	LONG	NUMBER	PLS_INT	RAW	UROWID	VARCHAR2
BIN_INT		X		X	X	X			X
CHAR	X		X	X	X	X	X	X	X
DATE		X		X					X
LONG		X					X		X
NUMBER	X	X		X		X			X
PLS_INT	X	X		X	X				X
RAW		X		X					X
UROWID		X							X
VARCHAR2	X	X	X	X	X	X	X	X	

Ўзгарувчи ва ўзгармасларни эълон қилиш

Сизнинг дастурингиз қийматларни ўзгарувчилар ва ўзгармасларда сақлайди. Сиз ўзгарувчи ва ўзгармасларни PL/SQL блоки, қисмдастур ва пакетларнинг эълон қилиш қисимида эълон қилишингиз мумкин.

```

birthday DATE;
emp_count SMALLINT := 0;
pi REAL := 3.14159;
radius REAL := 1;
area REAL := pi * radius**2;

```

```

birthday DATE;
birthday DATE := NULL;
credit_limit CONSTANT REAL := 5000.00;
blood_type CHAR := 'O';
blood_type CHAR DEFAULT 'O';
hours_worked INTEGER DEFAULT 40;
employee_count INTEGER := 0;
acct_id INTEGER(4) NOT NULL := 9999;
credit REAL(7,2);
debit credit%TYPE;
balance NUMBER(7,2);
minimum_balance balance%TYPE := 10.00;
my_dname scott.dept.dname%TYPE;

```

%ROWTYPE ишлатиш

%ROWTYPE атрибути жадвал қатори кўринишидаги ёзув турини эълон қилишга имкон беради:

```

DECLARE
    emp_rec emp%ROWTYPE;
    CURSOR c1 IS SELECT deptno, dname, loc FROM dept;
    dept_rec c1%ROWTYPE;
    .....
emp_rec.ename := 'JOHNSON';
emp_rec.sal := emp_rec.sal * 1.15;
    .....
CREATE PACKAGE emp_actions AS
    CURSOR c1 RETURN emp%ROWTYPE;  -- declare cursor
specification
    ...
END emp_actions;

CREATE PACKAGE BODY emp_actions AS

```

```

    CURSOR c1 RETURN emp%ROWTYPE IS  -- define cursor
body
    SELECT * FROM emp WHERE sal > 3000;
    ...
END emp_actions;

```

Ёзувни ҳамма майдончаларини бирданига 2 хил вариантда тўлдириш мумкин.

I вариант:

```

DECLARE
    dept-rec1          dept % ROWTYPE:
    dept-rec2          dept % ROWTYPE;
    CURSOR              C1  IS  SELECT  deptno, dname,
loc  FROM dept;
    dept-rec3          C1 % ROWTYPE;
    dept-rec4          C1 % ROWTYPE;
BEGIN
    ...
    dept-rec 1:      = dept-rec 2;
    dept-rec 4:      = dept-rec 3;
END
    dept-rec 2:      = dept-rec 3;          нотўғри!

```

II вариант:

A) DECLARE

```

    dept-rec          dept % ROWTYPE
    ...
BEGIN
    SELECT deptno, dname , loc  INTO dept-rec FROM
dept  WHERE deptno =30;
    ...

```

Б) INSERT INTO dept VALUES (dept-rec1); -- нотўғри

Хамроҳ номларни ишлатиш

Ифодалар қатнашган курсор қаторларига мурожаат қилиш қилиш учун , уларни олдин албатта содда ҳамроҳ номлаш зарур:

```

DECLARE

```

```

        CURSOR my_cursor IS SELECT sal + NVL
(comm,0) wages, ename
        FROM emp;
        my-rec my_cursor % ROWTYPE;
BEGIN
        OPEN my_cursor;
        LOOP
                FETCH my_cursor INTO my-rec
                EXIT WHEN my_cursor % NOT
FOUND;

                IF my_rec.wages > 2000 THEN.
                        INSERT INTO temp VALVES
(NULL, my_rec.wages, my_rec.ename);
                END IF;
        END LOOP;
        CLOSE my_cursor;
END

```

3. Номлаш қоидалари

Ўзгармаслар, ўзгарувчилар, курсорлар, курсор ўзгарувчилари, процедуралар, функциялар ва пакетлар каби барча PL/SQL дастури бирликлари ягона номлаш қоидаларига бўйсундилар. Номлаш содда, мураккаб, узоқланган ёки мураккаб ва узоқланган бўлиши мумкин:

```

raise_salary(...);           -- содда
emp_actions.raise_salary(...); -- мураккаб
raise_salary@newyork(...);   -- узоқланган
emp_actions.raise_salary@newyork(...); -- мураккаб ва
узоқланга

```

SQL командаларида локал ва формал ўзгарувчилар жадвал номидан устунлик қилади (агарда улар бир хил номланган бўлса);

Мисол 1.

```
FOR emp IN 1... 5 LOOP
```

```

...
UPDATE emp SET bonus = 500 WHERE      --нотўғри!
END LOOP;      цикл ўзгарувчиси деб тушунилади.

```

Мисол 2.

Нотўғри!

```

PROCEDURE      calc_bonus      (emp      NUMBER,
      bonus OUT
      REAL ) IS
avg_sal      REAL;
BEGIN
SELECT      AVG (sal)      INTO avg_sal      FROM
emp
WHERE
...
END;      формал      параметр      деб      қабул      қилади.

```

Бу ҳолатда жадвал номини фойдаланувчи номини кўрсатиш орқали аниқлаш мумкин:

Мисол 3.

Тўғри!

```

PROCEDURE      calc_bonus      (emp.NUMBER,      bonus
OUT
      REAL ) IS
Arg_sal      REAL;
...
BEGIN
SELECT      AVG (sal)      INTO avg_sal      FROM      scott.emp
WHERE
...
END;

```

Жадвал устунларининг номи локал ва формал ўзгарувчилар номидан устунлик қилади (агарда улар бир хил номланган бўлса);

Мисол 1.

```

{
  DECLARE
  Ename CHAR (10) : = ' KING';
    BEGIN
  DELETE FROM emp WHERE  ename = ename;
  ...
  END;

```

Ушбу процедура бажарилганда EMP жадвалининг ҳамма қаторлари ўчирилади. Бу ноқулай вазиятдан чиқиш учун еки бошқа ўзгарувчи киритиш лозим:

```

DECLARE
My-ename CHAR (10);
...

```

ёинки блок меткасини ишлатиш лозим:

```

{
  <<main>>
  DECLARE
  Ename CHAR ( 10): = 'KING';
    BEGIN
  DELETE FROM emp WHERE  ename = main.ename;
  ...
  END;

```

ёки қисм дастур номини фойдаланиш мумкин:

```

{

```

```

PROCEDURE calc_bonus (empno NUMBER, bonus out
REAL) IS
    avg_sal    REAL;
name CHAR (10);
job CHAR (15) := 'SALESMAN'
    BEGIN
        SELECT  AVG (sal)  INTO avg_sal  FROM emp
            WHERE  job = cal_bonus.job;      /* локал
ўзгарувчига мурожаат */
        SELECT  ename  INTO name  FROM emp
            WHERE  empno = cal_bonus.empno: --
параметрга мурожаат
    END;

```

Таъсир ва кўриниш соҳалари

Идентификаторнинг таъсир соҳаси деб, уни ишлатиш мумкин бўлган дастурнинг қисмига айтилади. Идентификаторнинг кўриниш соҳаси деб, унга номи орқали мурожаат қилиш мумкин бўлган дастурнинг қисмига айтилади.

Мисол 1.

Таъсир соҳаси

(Ташқи х учун):

DECLARE

```

X REAL;
  BEGIN
    ...
    DECLARE
      X REAL,
    BEGIN
      ...
    END;
    ...
  END;

```

(Ички х учун):

Кўриниш соҳаси

DECLARE

```

X REAL;
  BEGIN
    ...
    DECLARE
      X REAL,
    BEGIN
      ...
    END;
    ...
  END;

```

DECLARE

...
DECLARE

```
DECLARE
```

```
    X REAL;
```

```
X REAL
```

```
BEGIN
```

```
    BEGIN
```

```
        ...
```

```
        DECLARE
```

```
            X REAL,
```

```
        BEGIN
```

```
            ...
```

```
        END;
```

```
    ...
```

```
END;
```

Ички блокдан туриб глобал ўзгарувчига мурожаат қилиш учун меткалаш усулидан фойдаланиш мумкин:

Мисол 2. (Квалификатор)

```
<< Outer >>
```

```
DECLARE
```

```
    Birthday DATE
```

```
BEGIN
```

```
    ...
```

```
    DECLARE.
```

```
        Birthday DATE;
```

```
    BEGIN
```

```
        ...
```

```
        IF birthdate = outer.birthday THEN
```

```
            ...
```

```
        END IF;
```

```
    END;
```

```
END outer;
```

Қиймат бериш оператори

Ўзгарувчи ва ўзгармаслар ҳар гал блок ёёки қисмдастур киритилганда қиймат берилади. Ўзгарувчиларга қиймат бергунга қадар уларнинг қиймати NULL бўлади. Масалан:

```
DECLARE
    Count    INTEGER;
BEGIN
    Count: = Count + 1;    -- Count ўзгарувчига NULL
    қиймат беради
    ...
END;
```

Демак ўзгарувчи ва ўзгармасларнинг қиймати аниқланган бўлиши учун уларга ошкор тарзда қиймат бериш лозим.

Бул қийматлар

Бул турдаги ўзгарувчиларига 3 хил қиймат бериш мумкин:

- FALSE
- TRUE
- NULL

Масалан

```
DECLARE
    done    BOOLEAN
    ...
BEGIN
    done: =    FALLS
    WHILE    NOT    done    LOOP
        ...
    END    LOOP;
END;
```

ёки

```
done: = (count > 500);
```

Берилганлар базасидаги қийматлар

Берилганлар базасида жойлашган қийматларни ўзгарувчиларга бериш учун SELECT жумласидан фойдаланиш мумкин:

```
DECLARE
    my_empno emp.empno%TYPE;
    my_ename emp.ename%TYPE;
    wages    NUMBER(7,2);
BEGIN
    ...
    SELECT ename, sal + comm
        INTO last_name, wages FROM emp
        WHERE empno = emp_id;
```

Аmmo сиз устун қийматларини бул ўзгарувчиларига беролмайсиз.

Ифода ва солиштириш

Ифодалар операнд ва амаллар ёрдамида тузилади. Операнд ўзгарувчи, ўзгармас, литерал ёки чақираладиган функция бўлиши мумкин. Содда арифметик ифода мисолини келтирамиз:

$-X / 2 + 3$

Унар амаллар (- минус амали каби) битта операнд устида бажарилади, бинар амаллар (/ бўлиш амали каби) эса иккита операнд устида бажарилади.

Амалларни бажариш тартиби

Ифодаларда амаллар қуйидаги жадвалда кўрсатилган тартибда (юқоридан пастга қараб) бажарилади.

Оператор	Амал
**, NOT	Даражага кўтариш, инкор қилиш
+, -	Унар минус ва плюс
*, /	Кўпайтириш, бўлиш
+, -,	қўшиш, айириш, конкатенация амали

=, !=, <, >, <=, >=, IS NULL, LIKE, BETWEEN, IN	Солиштириш
AND	Мантикий кўпайтириш
OR	Мантикий қўшиш

Мантикий амаллар

Мантикий амаллар қуйидаги жадвалда кўрсатилган тарзда амалга оширилади.

x	y	x AND y	x OR y	NOT x
TRUE	TRUE	TRUE	TRUE	FALSE
TRUE	FALSE	FALSE	TRUE	
TRUE	NULL	NULL	TRUE	
FALSE	TRUE	FALSE	TRUE	TRUE
FALSE	FALSE	FALSE	FALSE	
FALSE	NULL	FALSE	NULL	
NULL	TRUE	NULL	TRUE	NULL
NULL	FALSE	FALSE	NULL	
NULL	NULL	NULL	NULL	

Солиштириш амаллари

Қуйидаги жадвалда солиштириш амалларининг маъноси келтирилган:

Оператор	Маъноси
=	тенг
<>, !=, ~=	тенг эмас

<	кичик
>	катта
<=	кичик ёки тенг
>=	катта ёки тенг

IS NULL оператори

IS NULL оператори TRUE бул қиймат қайтаради, агар унинг операнди NULL қийматга эга бўлса ва FALSE қийматни қайтаради, агар унинг операнди NULL фарқли қийматга эга бўлса. Масалан

```
IF variable IS NULL THEN ...
```

LIKE

LIKE оператори белгили қийматларни солиштириш учун ишлатилади ва у бул қиймат қайтаради. Масалан еname ўзгарувчиси 'JONSON' қийматига эга бўлса

```
ename LIKE 'J%SON'
```

ифода TRUE қиймат қайтаради. Бу ерда % белгилар тўпламини билдиради, у бўш ҳам бўлиши мумкин.

BETWEEN оператори

BETWEEN оператори қийматни берилган интервалда ётиш ёки ётмаслигини аниқлаб беради. Масалан

```
45 BETWEEN 38 AND 44
```

ифода FALSE қийматни қайтаради.

IN оператори

IN оператори қиймат берилган тўпламнинг бирор элементи билан устма-уст тушишини текширади. Тўпланда NULL қатнашиши мумкин, аммо у инобатга олинмайди. Масалан

```
DELETE FROM emp WHERE ename IN (NULL, 'KING', 'FORD');
```

ифода `ename` устунининг қиймати `NULL` бўлган қаторларни ўчирмайди. `NOT IN` оператори эса тўпلامда `NULL` иштирок этганда `FALSE` қиймат қайтаради. Масалан

```
DELETE FROM emp WHERE ename NOT IN (NULL, 'KING');
```

ифода ҳеч қандай қаторларни ўчирмайди.

Конкатенация оператори

Конкатенация оператори битта белгили қаторни иккинчисига қўшиш учун ишлатилади. Масалан

```
'suit' || 'case' = 'suitcase'
```

4. Бул ифодалар

PL/SQL ўзгарувчи ва ўзгармасларни солиштиришга имкон беради. Ушбу солиштиришлар бул ифодалар дейилади.

Арифметик ифодалар

Сонларни тенгликга ёки тенг эмасликга текшириш мумкин:

```
number1 = 75;
number2 = 70;
```

бўлганда қуйидаги ифода `TRUE` қиймат қайтаради:

```
number1 > number2
```

Белгили ифодалар

Белгили қийматларни ҳам тенгликга ёки тенг эмасликга текшириш мумкин:

```
string1 := 'Kathy';
string2 := 'Kathleen';
```

бўлганда, қуйидаги ифода `TRUE` қиймат қайтаради:

```
string1 > string2
```

Санавий ифодалар

Саналарни ҳам солиштириш мумкин:

```
date1 = '01-JAN-91';
date2 = '31-DEC-90';
```

бўлганда, қуйидаги ифода TRUE қиймат қайтаради:

```
date1 > date2
```

NULL қиймати

NULL қиймати билан ишлаганда қуйидаги қоидаларни унутманг:

- NULL қатнашган солиштиришлар ҳамма вақт NULL қайтаради.
- NOT мантиқий операторини NULL қўллаганда NULL ҳосил бўлади.
- Шартли ифодаларда шарт натижаси NULL бўлса, унга боғлиқ бўлган жумлалар кетма-кетлиги бажарилмайди. Масалан

Мисол 1.

```
x := 5;
y := NULL;
...
IF x != y THEN  -- TRUE ни эмас NULL қайтаради.
    Операторлар кетма-кетлиги -- бажарилмайди.
END IF;
```

Мисол 2.

```
a := NULL;
b := NULL;
...
IF a = b THEN  -- TRUE ни эмас NULL қайтаради.
    Операторлар кетма-кетлиги -- бажарилмайди.
END IF;
```

NOT оператори

NOT мантиқий операторини NULL қўллаганда NULL ҳосил бўлишини эслатиб ўтамиз. Шундай қилиб қуйидаги иккита жумла ҳамма вақт ҳам эквивалент эмас:

IF x > y THEN		IF NOT x > y THEN
high := x;		high := y;
ELSE		ELSE
high := y;		high := x;

```
END IF; | END IF;
```

ELSE жумласидаги операторлар кетма-кетлиги бажарилади, агарда IF шарти FALSE ёки NULL бўлса. Демак *x* ва *y* бири ёки иккаласи NULL бўлса, биринчи IF жумла *y* қийматини *high* ўзгарувчига беради, лекин иккинчи IF жумла *x* қийматини *high* ўзгарувчига беради.

Нул узунликдаги қатор

PL/SQL нул узунликдаги қаторларни NULL деб қабул қилади:

```
null_string := TO_VARCHAR2('');
zip_code := SUBSTR(address, 25, 0);
valid := (name != '');
```

ўзгарувчилар NULL қийматни қабул қилади.

Конкатенация амали

Конкатенация амали NULL операндларни инкор қилади:

```
'apple' || NULL || NULL || 'sauce' = 'applesauce'
```

5. Функциялар

Агар функцияга бўш қиймат узатилса *y* ҳолда *y* ҳам бўш қиймат қайтаради, қуйидаги ҳоллар бундан мустасно.

Қуйидаги мисолда *rating* устунининг қиймати NULL бўлганда DECODE функцияси 1000 қийматни қайтаради:

```
SELECT DECODE(rating, NULL, 1000, 'C', 2000, 'B', 4000,
'A', 5000)
INTO credit_limit FROM accts WHERE acctno =
my_acctno;
```

Қуйидаги мисолда *hire_date* - нинг қиймати NULL бўлганда

NVL функцияси SYSDATE қийматни қайтаради:

```
start_date := NVL(hire_date, SYSDATE);
new_string := REPLACE(old_string, NULL, my_string);
```

мисолда *old_string* ва *new_string* қийматлар бир хил.

```
syllabified_name := 'Gold-i-locks';
```

```
name := REPLACE(syllabified_name, '-', NULL);
```

мисолда name қиймати 'goldilocks' бўлади.

Ички қурилган функциялар

Ички қурилган функцияларни қуйидаги категорияларга бўлиш мумкин:

- Хатоларни рўйхатга олиш функциялари
- Сонли функциялар
- Белгили функциялар
- Берилганлар турларини алмаштириш функциялари
- Саналар билан ишлайдиган функциялар
- Объект билан ишлайдиган функциялар
- Аралаш функциялар

Хар бир категориядаги функциялар рўйхатини қуйидаги жадвалда келтирамиз:

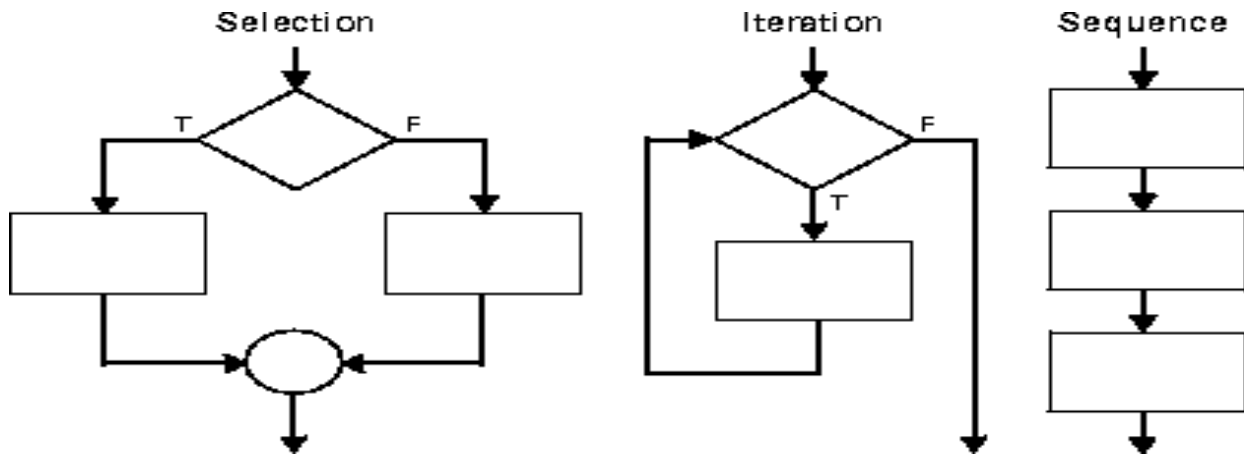
Хатолик	Сонли	Белгили	Алмаштириш	Санали	Объект	Аралаш
SQLC ODE SQLE RRM	ABS	ASCII	CHARTOROW	ADD_MONT	DEREF	BFILENAM
	ACOS	CHR	ID	HS	REF	E
	ASIN	CONCAT	CONVERT	LAST_DAY	VALUE	DECODE
	ATAN	INITCAP	HEXTORAW	MONTHS_B		DUMP
	ATAN2	INSTR	RAWTOHEX	ETWEEN		EMPTY_BLOB
	CEIL	INSTRB	ROWIDTOCHAR	NEW_TIME		EMPTY_CLOB
	COS	LENGTH	TO_CHAR	ROUND		EMPTY_CLOB
	COSH	LENGTHB	TO_DATE	SYSDATE		GREATEST
	EXP	LOWER	TO_MULTI_BYTE	TRUNC		LEAST
	FLOOR	LPAD				NLS_CHAR
	LN	LTRIM	TO_NUMBER			SET_DECL

LOG	NLS_INI	TO_SINGLE			_LEN
MOD	TCAP	_BYTE			NLS_CHAR
POWER	NLS_LOWER				SET_ID
ROUND	NLSSORT				NLS_CHAR
SIGN	NLS_UPPER				SET_NAME
SIN	REPLACE				NVL
SINH	RPAD				SYS_CONTEXT
SQRT	RTRIM				SYS_GUID
TAN	SOUNDEX				UID
TANH	SUBSTR				USER
TRUNC	SUBSTRB				USERENV
	TRANSLATE				VSIZE
	TRIM				
	UPPER				

6. Назорат тузилмалари

Ушбу бўлимда PL/SQL дастурларида назорат оқимлари қандай тузилганлиги ҳақида сўз юритилади. Назорат тузилмалари қуйидаги жадвалда келитирлган:

Назорат тузилмалари



Шартли назорат: IF жумлалари

IF жумлаларнинг учта формаси мавжуд: IF-THEN, IF-THEN-ELSE, ва IF-THEN-ELSIF.

IF-THEN

IF-THEN жумласи қуйидаги синтаксисга эга

IF шарт THEN

Жумлалар_кетма_кетлиги

END IF;

Бу ерда жумлалар_кетма_кетлиги бажарилади, агарда шарт рост бўлса.

Масалан

```
IF sales > quota THEN
    compute_bonus(empid);
    UPDATE payroll SET pay = pay + bonus WHERE empno =
emp_id;
END IF;
```

Сиз IF жумласини битта қаторда ҳам ёзишингиз мумкин:

```
IF x > y THEN high := x; END IF;
```

IF-THEN-ELSE

IF-THEN-ELSE жумласи қуйидаги синтаксисга эга

IF шарт THEN

Жумлалар_кетма_кетлиги1

```
ELSE
    Жумлалар_кетма_кетлиги2
END IF;
```

Бу ерда жумлалар_кетма_кетлиги1 бажарилади, агарда шарт рост бўлса, акс ҳолда жумлалар_кетма_кетлиги2 бажарилади. Масалан

```
IF trans_type = 'CR' THEN
    UPDATE accounts SET balance = balance + credit WHERE
...
ELSE
    UPDATE accounts SET balance = balance - debit WHERE
...
END IF;
```

IF жумлалари ўзаро ичма-ич жойлашиши мумкин:

```
IF trans_type = 'CR' THEN
    UPDATE accounts SET balance = balance + credit WHERE
...
ELSE
    IF new_balance >= minimum_balance THEN
        UPDATE accounts SET balance = balance - debit
WHERE ...
    ELSE
        RAISE insufficient_funds;
    END IF;
END IF;
```

IF-THEN-ELSIF

IF-THEN-ELSIF жумласи қуйидаги синтаксисга эга

```
IF шарт1 THEN
    Жумлалар_кетма_кетлиги1
ELSIF шарт2 THEN
    Жумлалар_кетма_кетлиги2
ELSE
    Жумлалар_кетма_кетлиги3
```

```
END IF;
```

Бу ерда жумлалар_кетма_кетлиги1 бажарилади, агарда шарт1 рост бўлса, акс ҳолда жумлалар_кетма_кетлиги2 бажарилади, агарда шарт2 рост бўлса, акс ҳолда жумлалар_кетма_кетлиги1 бажарилади. Масалан

```
BEGIN
```

```
...
```

```
IF sales > 50000 THEN
```

```
    bonus := 1500;
```

```
ELSIF sales > 35000 THEN
```

```
    bonus := 500;
```

```
ELSE
```

```
    bonus := 100;
```

```
END IF;
```

```
INSERT INTO payroll VALUES (emp_id, bonus, ...);
```

```
END;
```

IF жумлаларида ELSIF жумлалари сони ихтиёрий бўлиши мумкин. Ичма-ич жойлашган IF жумлалари ўрнига ELSIF жумлаларидан фойдаланиш тавсия этилади.

```
IF condition1 THEN
```

```
    statement1;
```

```
ELSE
```

```
    IF condition2 THEN
```

```
        statement2;
```

```
    ELSE
```

```
        IF condition3 THEN
```

```
            statement3;
```

```
        END IF;
```

```
    END IF;
```

```
END IF;
```

```
| IF condition1 THEN
```

```
|     statement1;
```

```
| ELSIF condition2 THEN
```

```
|     statement2;
```

```
| ELSIF condition3 THEN
```

```
|     statement3;
```

```
|         END IF;
```

```
|
```

```
|
```

```
|
```

```
|
```

Итератив назорат: LOOP ва EXIT жумлалари

LOOP жумлалари жумлалар кетма-кетлигини кўп марта бажариш учун ишлатилади. LOOP жумласининг уч хил шакли мавжуд:

LOOP, WHILE-LOOP, ва FOR-LOOP.

LOOP

LOOP жумласи қуйидаги синтаксисга эга:

```
LOOP
```

```
    Жумлалар_кетма_кетлиги
```

```
END LOOP;
```

Ушбу циклик жараёндан EXIT орқали чиқиш мумкин

EXIT

EXIT жумласи шартсиз тарзда LOOP жумласини ишини тугатишга мажбурлайди. Масалан

```
LOOP
```

```
    ...
```

```
    IF credit_rating < 3 THEN
```

```
        ...
```

```
        EXIT; -- циклдан дарҳол чиқади
```

```
    END IF;
```

```
END LOOP;
```

```
-- бошқарув шу ерга узатилади
```

Қуйидаги мисол PL/SQL блоки ишини тугатиш учун EXIT жумласидан фойдаланиб бўлмаслигини намойиш этади

```
BEGIN
```

```
    ...
```

```
    IF credit_rating < 3 THEN
```

```
        ...
```

```
        EXIT; -- қоидага ҳилоф
```

```
    END IF;
```

```
END;
```

EXIT-WHEN

EXIT-WHEN жумласи LOOP жумласини шартли тарзда тугатишга имкон беради.

```

LOOP
    FETCH c1 INTO ...
    EXIT WHEN c1%NOTFOUND; -- шарт рост бўлса, loop
    тугатилади
    ...
END LOOP;
CLOSE c1;

```

EXIT-WHEN жумласи содда IF жумласини алмаштиради:

```

IF count > 100 THEN      |      EXIT WHEN count > 100;
    EXIT;                |
END IF;                  |

```

Loop меткалари

Loop меткалани мумкин:

Мисол 1.

```

<<label_name>>
LOOP
    sequence_of_statements
END LOOP;

```

Мисол 2.

```

<<my_loop>>
LOOP
    ...
END LOOP my_loop;

```

```

<<outer>>
LOOP
    ...
    LOOP
        ...
        EXIT outer WHEN ... - иккала loop чиқади
    
```

```

    END LOOP;

    ...
END LOOP outer;

```

WHILE-LOOP

WHILE-LOOP жумласининг синтаксиси:

```

WHILE condition LOOP
    Жумлалар_кетма_кетлиги
END LOOP;

```

Агар шарт рост бўлса, жумлалар кетма-кетлиги бажарилади.

Мисол 1.

```

WHILE total <= 25000 LOOP
    ...
    SELECT sal INTO salary FROM emp WHERE ...
    total := total + salary;
END LOOP;

```

Мисол 2.

```

LOOP
    sequence_of_statements
    EXIT WHEN boolean_expression;
END LOOP;

```

Мисол 3.

```

done := FALSE;
WHILE NOT done LOOP
    sequence_of_statements
    done := boolean_expression;
END LOOP;

```

Мисол 4. Эквивалент мисоллар

WHILE TRUE LOOP		LOOP
...		...

```
END LOOP; | END LOOP;
```

FOR-LOOP

Олдиндан аниқ бўлган чекли сондаги итерацияларни амалга ошириш учун FOR-LOOP конструкцияси ишлатилади. Унинг синтаксиси:

```
FOR счётчик IN [REVERSE] қуйи_чегара..юқори_чегара LOOP
    Жумлалар_кетма_кетлиги
END LOOP;
```

Мисол 1.

```
FOR i IN 1..3 LOOP -- 1,2,3 қийматларни i беради
    Жумлалар_кетма_кетлиги -- 3 марта бажарилади
END LOOP;
```

Мисол 2.

```
FOR i IN 3..3 LOOP -- 3 қийматни i беради
    Жумлалар_кетма_кетлиги -- 1 марта бажарилади
END LOOP;
```

Мисол 3.

```
FOR i IN REVERSE 1..3 LOOP --3,2,1 қийматларни i беради
    Жумлалар_кетма_кетлиги -- 3 марта бажарилади END LOOP;
```

Мисол 3.

```
FOR ctr IN 1..10 LOOP
    IF NOT finished THEN
        INSERT INTO ... VALUES (ctr, ...); -- мумкин
        factor := ctr * 2; -- мумкин
    ELSE
        ctr := 10; -- қоидага хилоф
    END IF;
END LOOP;
```

Итерация схемалари

Loop счётчикининг ўзгариши қуйдаги схемалар остида амалга оширилади:

```
j IN -5..5
k IN REVERSE first..last
step IN 0..TRUNC(high/low) * 2
```

Мисол 1. (BASIC тилида)

FOR J = 5 TO 15 STEP 5 :REM 5,10,15 қийматлар J берилади

Жумлалар_кетма_кетлиги -- J 5,10,15 қийматларга эга

NEXT J

Мисол 2. (PL/SQL да)

```
FOR j IN 5..15 LOOP  -- 5,6,7,... қийматлар J берилади
    IF MOD(j, 5) = 0 THEN  -- 5 тадан ўтади
        Жумлалар_кетма_кетлиги -- J 5,10,15 қийматларга эга
    END IF;
END LOOP;
```

Мисол 3.

```
FOR j IN 1..3 LOOP  -- 1,2,3 қийматлар J берилади
    Жумлалар_кетма_кетлиги
END LOOP;
```

Динамик интервал

PL/SQL цикл ўзгариши соҳасини динамик тарзда аниқлашга имкон беради:

```
SELECT COUNT(empno) INTO emp_count FROM emp;
FOR i IN 1..emp_count LOOP
    ...
END LOOP;
```

Агар цикл ўзгарувчисининг юқори чегараси пастги чегарасидан кичик бўлса, цикл бажарилмайди ва бошқарув навбатдаги жумлага узатилади:

```
-- limit қиймати 1 бўлсин
FOR i IN 2..limit LOOP
    Жумлалар_кетма_кетлиги -- бажарилмайди
END LOOP;
-- бошқарув шу ерга узатилади
```

Таъсир қилиш соҳаси

Цикл ўзгарувчиси фақат цикл ичида аниқланган:

```
FOR ctr IN 1..10 LOOP
    ...
END LOOP;
sum := ctr - 1; -- қоидага хилоф
```

Цикл ўзгарувчисини эълон қилиш керак эмас:

```
DECLARE
    ctr INTEGER;
BEGIN
    ...
    FOR ctr IN 1..25 LOOP
        ...
        IF ctr > 10 THEN ... -- refers to loop counter
    END LOOP;
END;
```

Глобал ўзгарувчига мурожаат қилиш учун, меткадан фойдаланишингиз зарур:

```
<<main>>
DECLARE
    ctr INTEGER;
    ...
BEGIN
```

```

...
FOR ctr IN 1..25 LOOP
    ...
    IF main.ctr > 10 THEN --глобал ўзгарувчига
мурожаат
        ...
    END IF;
END LOOP;
END main;

```

Айнан шу қоидалар ичма-ич жойлашган цикллар учун ҳам ўринлидир:

```

<<outer>>
FOR step IN 1..25 LOOP
    FOR step IN 1..10 LOOP
        ...
        IF outer.step > 15 THEN ...
    END LOOP;
END LOOP outer;

```

EXIT жумласидан фойдаланиш

Цикл жараёнини дарҳол бузиш учун EXIT жумласидан фойдаланиш мумкин:

```

FOR j IN 1..10 LOOP
    FETCH c1 INTO emp_rec;
    EXIT WHEN c1%NOTFOUND;
    ...
END LOOP;

```

EXIT жумласи орқали ичма-ич жойлашган цикллардан ҳам чиқиш имконияти мавжуд:

```

<<outer>>
FOR i IN 1..5 LOOP

```

```

...
FOR j IN 1..10 LOOP
    FETCH c1 INTO emp_rec;
    EXIT outer WHEN c1%NOTFOUND; --иккала циклдан
чиқади
    ...
END LOOP;
END LOOP outer;
-- бошқарув шу ерга узатилади.

```

Кетма-кет бошқарув: GOTO ва NULL жумлалари

GOTO жумласи

GOTO жумласи меткаланган бўлимга шартсиз равишда ўтиш учун ишлатилади:

Мисол 1.

```

BEGIN
    ...
    GOTO insert_row;
    ...
    <<insert_row>>
    INSERT INTO emp VALUES ...
END;

```

Мисол 2.

```

BEGIN
    ...
    <<update_row>>
    BEGIN
        UPDATE emp SET ...
    ...

```

```

    END;

    ...

    GOTO update_row;

    ...
END;
```

Мисол 3.

```

DECLARE
    done    BOOLEAN;
BEGIN
    ...

    FOR i IN 1..50 LOOP
        IF done THEN
            GOTO end_loop;
        END IF;

        ...

        <<end_loop>>  -- қоидага хилоф
    END LOOP;      -- бажарилмайдиган жумла
END;
```

Мисол 4.

```

FOR i IN 1..50 LOOP
    IF done THEN
        GOTO end_loop;
    END IF;

    ...

    <<end_loop>>
NULL;  -- бажариладиган жумла
END LOOP;
```

Мисол 5.

```

DECLARE
```

```

    my_ename CHAR(10);
BEGIN
    <<get_name>>
    SELECT ename INTO my_ename FROM emp WHERE ...
    BEGIN
        ...
        GOTO get_name;
    END;
END;

```

Чегаралар

GOTO жумласини ишлатишда айрим чегаралар мавжуд. Хусусан IF жумласи ичига, LOOP жумласига, ёки қисмблок ичига бошқарувни узатиш мумкин эмас:

Мисол 1.

```

BEGIN
    ...
    GOTO update_row;  -- қоидага хилоф жумла
    ...
    IF valid THEN
        ...
        <<update_row>>
        UPDATE emp SET ...
    END IF;
END;

```

Мисол 2.

```

BEGIN
    ...
    IF valid THEN
        ...

```

```

        GOTO update_row;  -- қоидага хилоф жумла
ELSE
    ...
    <<update_row>>
    UPDATE emp SET ...
END IF;
END;
```

Мисол 3.

```

BEGIN
    ...
    IF status = 'OBSOLETE' THEN
        GOTO delete_part;  -- қоидага хилоф жумла
    END IF;
    ...
    BEGIN
        ...
        <<delete_part>>
        DELETE FROM parts WHERE ...
    END;
END;
```

Мисол 4.

```

DECLARE
    ...
    PROCEDURE compute_bonus (emp_id NUMBER) IS
    BEGIN
        ...
        GOTO update_row;  -- қоидага хилоф жумла
    END;
BEGIN
    ...
    <<update_row>>
```

```
    UPDATE emp SET ...
END;
```

Мисол 5.

```
DECLARE
    ...
    pe_ratio REAL;
BEGIN
    ...
    SELECT price / NVL(earnings, 0) INTO pe_ratio FROM
    ...
    <<insert_row>>
    INSERT INTO stats VALUES (pe_ratio, ...);
EXCEPTION
    WHEN ZERO_DIVIDE THEN
        pe_ratio := 0;
        GOTO insert_row;  -- қоидага хилоф жумла
END;
```

NULL Statement

NULL жумласи ошкор тарзда ҳаракат қилмасликни аниқлайди. У ҳеч қандай амал бажармайди ва бошқарувни навбатдаги жумлага узатади.

Мисол 1.

```
EXCEPTION
    WHEN ZERO_DIVIDE THEN
        ROLLBACK;
    WHEN VALUE_ERROR THEN
        INSERT INTO errors VALUES ...
        COMMIT;
    WHEN OTHERS THEN
        NULL;
END;
```

Мисол 2.

```
IF rating > 90 THEN
    compute_bonus(emp_id);
ELSE
    NULL;
END IF;
```

Мисол 3.

```
PROCEDURE debit_account (acct_id INTEGER, amount REAL)
IS
BEGIN
    NULL;
END debit_account;
```

7. Коллекциялар ва ёзувлар

Берилганлар базасининг анъанавий илова дастурларида дастурчилар массив, рўйхат, ичма-ич жойлашган жадваллар, тўпламлар ва дарахтлар каби коллекциялардан фойдаланадилар. Ушбу талабларни қондириш мақсадда PL/SQL сизга индекслашган жадваллар, ичма–ич жойлашган жадваллар ва ўзгарувчан-ўлчовли массивларни эълон қилиш учун TABLE ва VARRAY берилганлар турларини ҳавола қилади. Бу ерда сиз RECORD берилганлар турини ҳам ўрганасиз.

Асосий бўлимлар

Коллекция нима?

Коллекцияларни инициализация қилиш ва уларга ҳавола этиш

Коллекциларга қиймат бериш ва солиштириш

Коллекцияларни бошқариш

Коллекция методларидан фойдаланиш

Коллекция мустасноларидан холи бўлиш

Ёзув нима?

Ёзувларни аниқлаш ва эълон қилиш

Ёзувларни инициализация қилиш ва уларга ҳавола этиш

Ёзувларга қиймат бериш ва солиштириш

Ёзувларни бошқариш

Коллекция нима?

Коллекция – бу бир хил турдаги тартиблаштирилган элементлар тўпламидир. Хар бир элемент ягона индексга эга бўлиб, у элементнинг коллекциядаги жойлашув жойини аниқлайди. PL/SQL икки турдаги коллекцияни тавсия қилади. TABLE туридаги элементлар *индекслаштирилган жадваллар* (2 Версия *PL/SQL жадваллари*) ёки *ичма-ич жойлашган жадваллар* бўлиши мумкин VARRAY туридаги элементлар массивлар (ўзгарувчан-ўлчовли массивлар) демакдир.

Коллекция нусхалари сифатида объект тури ҳам бўлиши мумкин ва аксинча улар объект турининг атрибути ҳам бўлиши мумкин. Коллекцияларни параметр сифатида ҳам узатиш мумкин.

Ичма-ич жойлашган жадваллар

Берилганлар базасида ичма-ич жойлашган жадвалларни берилганлар базасидаги бир устунли жадвал сифатида қараш мумкин.

PL/SQL-да эса ичма-ич жойлашган жадваллар бир ўлчовли массивга ўхшаган бўлади. Аммо, лекин ичма-ич жойлашган жадваллар массивлардан икки жиҳатдан кескин фарқ қилади. Биринчиси, массивлар фиксирланган юқори чегарага эга бўлади, ичма-ич жойлашган жадваллар эса чегараланмаган. Демак, ичма-ич жойлашган жадвалнинг ўлчови динамик тарзда ўзгариши мумкин.

Ичма-ич жойлашган жадвал ва массив

Array of Integers

321	17	99	407	83	622	105	19	67	278
x(1)	x(2)	x(3)	x(4)	x(5)	x(6)	x(7)	x(8)	x(9)	x(10)

Fixed
Upper
Bound

Nested Table after Deletions

321		99	407		622	105	19		278
x(1)		x(3)	x(4)		x(6)	x(7)	x(8)		x(10)

Unbounded

Иккинчидан массив индивидуал элементларини ўчириб бўлмайди, шу билан бирга сиз ички қурилган DELETE процедураси орқали ичма-ич жойлашган жадвал элементини ўчиришингиз мумкин.

Индекслаштирилган жадваллар ва ичма-ич жойлашган жадваллар

Индекслаштирилган жадваллар ва ичма-ич жойлашган жадваллар жуда ўхшашдирлар. Масалан, улар бир хил тузилмага эга ва уларнинг элементларига бар хил тарзда кирилади. Улар орасидаги асосий фарқ эса шундаки, ичма-ич жойлашган жадвалларни берилганлар базасидаги устунларда сақлаш мумкин, индекслаштирилган жадвалларни эса мумкин эмас. Яна таъкидлаб ўтиш жоизки, индекслаштирилган жадвал берилганлар базасида сақланмайди.

Массивлар

VARRAY туридаги элементлар *массивлар* дейилади. Массив элементига мурожаат қилиш учун стандарт индекслаш синтаксисидан фойдаланасиз:

10 ўлчовли массив

Varray Grades

B	C	A	A	C	D	B			
(1)	(2)	(3)	(4)	(5)	(6)	(7)			

Maximum
Size = 10

Массив максимал ўлчовига эга бўлиб, уни унинг турини тавсифлашда аниқлашингиз лозим. Унинг индекси фиксирланган 1-қуйи чегарага эга ва кенгайиши мумкин бўлган юқори чегарага эгадир. Шундай қилиб массив ўзгарувчан сондаги элементлардан (нолдан унинг турини тавсифида аниқланган максимумигача) иборат бўлиши мумкин.

Массивлар ва ичма-ич жойлашган жадваллар

Ичма-ич жойлашган жадваллар массивлардан қуйидаги жиҳатлардан фарқ қилади:

- Массивлар максимал ўлчовга эга, ичма-ич жойлашган жадваллар бундай эмас.
- Массив индивидуал элементларини ўчириб бўлмайди, шу билан бирга ичма-ич жойлашган жадвал элементини ўчиришингиз мумкин.
- Oracle массивни қаторда сақлайди, ичма-ич жойлашган жадвални эса сақланадиган жадвалда сақлайди.
- Берилганлар базасида сақлашда массивлар ўзининг тартибини ва индексларини сақлаб қолади, ичма-ич жойлашган жадваллар эса бундай эмас.

Коллекцияларни аниқлаш ва эълон қилиш

Коллекцияларни ташкил қилиш учун, аввал коллекция турини аниқлаб, сўнг ушбу турдаги колекцияни эълон қилиш лозим. TABLE ва VARRAY турларини ихтиёрий PL/SQL блоки, қисмдастур ёки пакетнинг эълон қилиш соҳасида эълон қилиш мумкин. Ичма-ич жойлашган жадвалларда қуйидаги синтаксисдан фойдаланинг

`TYPE тур_номи IS TABLE OF элемент_тури [NOT NULL];` ва массивлар учун қуйидаги синтаксисдан фойдаланинг:

```
TYPE тур_номи IS {VARRAY | VARYING ARRAY}
(ўлчов_чегараси)
OF элемент_тури [NOT NULL];
```

Бу ерда тур_номи тур аниқловчиси бўлиб, келгусида коллекцияларни эълон қилиш учун ишлатилади, ўлчов_чегараси мусбат бутун литерал ва элемент_тури ихтиёрий PL/SQL берилганлар тури (қуйидагилардан ташқари):

BINARY_INTEGER, PLS_INTEGER

BOOLEAN

BLOB, CLOB (фақат массивлар учун қўлланилади)

LONG, LONG RAW

NATURAL, NATURALN

NCHAR, NCLOB, NVARCHAR2

BLOB ёки CLOB атрибутли объект тури (фақат массивлар учун қўлланилади)

TABLE ва VARRAY атрибутли объект тури

POSITIVE, POSITIVEN

REF CURSOR

SIGNTYPE

STRING

TABLE

VARRAY

Агар элемент_тури ёзув тури бўлса, ёзувдаги ҳар бир қатор скаляр турга ёки объект турига тегишли бўлиши лозим.

Индексلاштирилган жадваллар учун қуйидаги синтаксисдан фойдаланинг:

```
TYPE тур_номи IS TABLE OF элемент_тури [NOT NULL]
    INDEX BY BINARY_INTEGER;
```

Ичма-ич жойлашган жадваллар ва массивлардан фаркли равишда индекслаштирилган жадвалларда қуйидаги элемент турлари ҳам бўлиши мумкин: BINARY_INTEGER, BOOLEAN, LONG, LONG RAW, NATURAL, NATURALN, PLS_INTEGER, POSITIVE, POSITIVEN, SIGNTYPE, ва STRING. қуйида келтирилган мисолда индекслаштирилган ёзувлар жадвали эълон қилинади. Жадвалнинг ҳар бир элементи берилганлар базасидаги emp жадвали қаторини сақлайди.

DECLARE

```

TYPE EmpTabTyp IS TABLE OF emp%ROWTYPE
    INDEX BY BINARY_INTEGER;
emp_tab EmpTabTyp;
BEGIN
    SELECT * INTO emp_tab(7468) FROM emp WHERE empno =
7788;

```

VARARRAY турини аниқлашда, сиз унинг максимал ўлчовини кўрсатишингиз лозим. Қуйидаги мисолда 366-тагача саналарни сақловчи массив тури аниқланган:

```

DECLARE
    TYPE Calendar IS VARRAY(366) OF DATE;

```

Ўзгарувчи ёки берилганлар базаси устунидаги берилганлар турини аниқлашга имкон берувчи элемент турини аниқлаш учун %TYPE фойдаланиш мумкин. Берилганлар базасидаги жадвал ёки курсор қатори кўринишидаги турни аниқлашга имкон берувчи %ROWTYPE ҳам фойдаланишингиз мумкин:

```

DECLARE
    TYPE EmpList IS TABLE OF emp.ename%TYPE;  -- устунга
асосланган
    CURSOR c1 IS SELECT * FROM dept;
    TYPE DeptFile IS VARRAY(20) OF c1%ROWTYPE;  --
курсорга асосланган

```

Навбатдаги мисолда, элемент турини аниқлаш учун RECORD туридан фойдаланишингиз мумкин:

```

DECLARE
    TYPE AnEntry IS RECORD (
        term      VARCHAR2(20),
        meaning   VARCHAR2(200));
    TYPE Glossary IS VARRAY(250) OF AnEntry;

```

Охирги мисолда, элемент турига NOT NULL чеклагич қўйилади:

```
DECLARE
    TYPE EmpList IS TABLE OF emp.empno%TYPE NOT NULL;
```

Коллекцияларни эълон қилиш

Коллекцияни бир марта эълон қилиб, сўнг ушбу турдаги коллекцияларни эълон қилишингиз мумкин:

```
CREATE TYPE CourseList AS TABLE OF VARCHAR2(10)  --
турни аниқлайди
/
CREATE TYPE Student AS OBJECT (--объект ташкил этиш
    id_num    INTEGER(4),
    name      VARCHAR2(25),
    address   VARCHAR2(35),
    status    CHAR(2),
    courses   CourseList)  -- ичма-ич жойлашган жадвални
атрибут сифатида эълон қилиш
/
```

Courses идентификатори ичма-ич жойлашган жадвални ифодалайди.

Қуйида келтирилган скрипт, массивларни сақловчи берилганлар базасидаги устунни очади. Массивларнинг ҳар бир элементи Project объектини сақлаш учун мўлжалланган .

```
CREATE TYPE Project AS OBJECT(  --объект очади
    project_no NUMBER(2),
    title      VARCHAR2(35),
    cost       NUMBER(7,2))
/
CREATE TYPE ProjectList AS VARRAY(50) OF Project  --
VARRAY тури аниқланади
/
CREATE TABLE department (  -- берилганлар базасидаги
жадвални очиш
```

```

dept_id  NUMBER(2),
name     VARCHAR2(15),
budget   NUMBER(11,2),
projects ProjectList)  -- массивни устун сифатида
ЭЪЛОН ҚИЛИШ
/

```

Қуйидаги мисол олдин эълон қилинган коллекция кўринишидаги берилганлар турини аниқлаш учун %TYPE фойдаланишни намоиш этади:

```

DECLARE
    TYPE Platoon IS VARRAY(20) OF Soldier;
    p1 Platoon;
    p2 p1%TYPE;

```

Коллекцияларни функция ва процедураларнинг параметрлари сифатида эълон қилишингиз мумкин. Қуйидаги мисолда пакетдаги процедура учун формал параметр сифатида коллекция эълон қилинган:

```

CREATE PACKAGE personnel AS
    TYPE Staff IS TABLE OF Employee;
    ...
    PROCEDURE award_bonuses (members IN Staff);
END personnel;

```

RETURN жумласида ҳам коллекция турини аниқлашингиз мумкин:

```

DECLARE
    TYPE SalesForce IS VARRAY(25) OF Salesperson;
    FUNCTION top_performers (n INTEGER) RETURN SalesForce
IS ...

```

Коллекцияларни инициализация қилиш ва унга ҳавола этиш

Ичма-ич жойлашган жадвал ёки массивни инициализация қилинмагунча, улар атомик бўш бўлади (яъни, коллекцияни ўзи (унинг

элементлари эмас) бўш бўлади). Ичма-ич жойлашган жадвал ёки массивни инициализация қилиш учун тизим томонидан аниқланган ва номи коллекция номи билан устма-уст тушадиган *конструктор* функциясидан фойдаланасиз. Ушбу функция унга узатилган элементлардан коллекцияларни «кўради». Қуйидаги мисолда `CourseList()`, конструкторга олти элемент узатилади ва у ушбу элементлардан таркиб топган ичма-ич жойлашган жадвални қайтаради:

```
DECLARE
    my_courses CourseList;
BEGIN
    my_courses := CourseList('Econ 2010', 'Acct 3401',
    'Mgmt 3100',
        'PoSc 3141', 'Mktg 3312', 'Engl 2005');
    ...
END;
```

Қуйидаги мисолда `ProjectList()`, конструкторга учта объект узатилади:

```
DECLARE
    accounting_projects ProjectList;
BEGIN
    accounting_projects := ProjectList(Project(1, 'Design
New Expense Report', 3250),
        Project(2, 'Outsource Payroll', 12350),
        Project(3, 'Audit Accounts Payable', 1425));
    ...
END;
```

Бутун массивни инициализация қилиш зарурияти йуқ. Масалан, агар массивнинг максимал ўлчови 50 тенг бўлса, сиз бир нечта 50-дан ошмаган элементларни конструкторга узатишингиз мумкин. Агар сиз `NOT NULL` чеклагичини қўймаган бўлсангиз ёки элемент учун ёзув турини аниқлаган бўлсангиз, бўш элементни конструкторга узатишингиз мумкин:

```
BEGIN
    my_courses := CourseList('Math 3010', NULL, 'Stat
3202', ...);
```

Навбатдаги мисол, коллекцияларни эълон қилишда инициализация қилиш мумкинлигини намоёиш этади:

```
DECLARE
    my_courses CourseList :=
        CourseList('Art 1111', 'Hist 3100', 'Engl 2005',
...);
```

Агар сиз конструкторни аргументсиз чақирсангиз, сиз null бўлмаган бўш коллекцияни ҳосил қиласиз:

```
DECLARE
    TYPE Clientele IS VARRAY(100) OF Customer;
    vips Clientele := Clientele(); -- бўш массивни
инициализация қилиш
BEGIN
    IF vips IS NOT NULL THEN --шарт TRUE қайтаради
        ...
    END IF;
END;
```

PL/SQL (индекслаштирилган жадваллардан ташқари) конструкторни ошқормас тарзда конструкторни ҳеч қачон чақирмайди, шу сабабли уни ошқор тарзда чақиришингиз шарт. Функцияни чақириш қаерда рухсат этилган бўлса, ўша жойда конструкторни ҳам чақириш рухсат этилган. Булар SELECT, VALUES ва SET жумлаларни ўз ичига олади.

Қуйидаги мисолда, сиз Student объектини sophomores объектли жадвалга киритасиз. CourseList() жадвал конструктори courses атрибутига қиймат киритади.

```
BEGIN
```

```

INSERT INTO sophomores
    VALUES (Student(5035, 'Janet Alvarez', '122 Broad
St', 'FT',
        CourseList('Econ 2010', 'Acct 3401', 'Mgmt
3100', ...)));
...

```

Охирги мисолда берилганлар базасидаги department жадвалига қатор киритилади. ProjectList() массив конструктори projects устунига қиймат киритади

```

BEGIN
    INSERT INTO department
        VALUES(60, 'Security', 750400,
            ProjectList(Project(1, 'Issue New Employee
Badges', 9500),
                Project(2, 'Find Missing IC Chips',
2750),
                Project(3, 'Inspect Emergency
Exits', 1900)));
...

```

Коллекция элементларига мурожаат қилиш

Хар бир элементга мурожаат қилиш учун коллекция номи ва қавс ичига олинган индексни кўрсатиш лозим. Индекс қайси элеменга мурожаат қилинаётганлигини аниқлаб беради. Мурожаат қилишнинг синтаксиси қуйидагича:

Коллекция_номи(индекс)

бу ерда индекс бутун сонни қайтарувчи ифода. Индекслашган жадваллар учун рухсат этилган индекслар интервали -2147483647 .. 2147483647 ташкил қилади. Ичма-ич жойлашган жадваллар учун рухсат

этилган индекслар интервали 1 .. 2147483647 иборат. Ва массивлар учун 1 .. ўлчов_чегараси.

Барча ифодаларда коллекцияларга мурожаат қилиш мумкин. Қуйидаги мисолда ичма-ич жойлашган names жадвали элементларига мурожаат қилинапти:

```
DECLARE
    TYPE Roster IS TABLE OF VARCHAR2(15);
    names Roster := Roster('J Hamil', 'D Caruso', 'R
Singh', ...);
    i BINARY_INTEGER;
BEGIN
    ...
    IF names(i) = 'J Hamil' THEN
        ...
    END IF;
END;
```

Қуйидаги мисолда қисмдастурларни чақиришда коллекцияларга мурожаат қилиш намоиш этилган:

```
DECLARE
    TYPE Roster IS TABLE OF VARCHAR2(15);
    names Roster := Roster('J Hamil', 'D Piro', 'R
Singh', ...);
    i BINARY_INTEGER;
BEGIN
    ...
    verify_name(names(i)); -- процедурани чақириш
END;
```

Коллекцияни қайтарувчи функцияни чақиришда колекция элементига мурожаат қилиш учун қуйидаги синтаксисдан фойдаланинг:

Функция_номи(параметрлар_рўйхати)(индекс)

Қуйидаги мисолда new_hires функция қайтарадиган массивнинг учинчи элементиға мурожаат қилинапти

```
DECLARE
    TYPE Staff IS VARRAY(20) OF Employee;
    staffer Employee;
    FUNCTION new_hires (hiredate DATE) RETURN Staff IS
...
BEGIN
    staffer := new_hires('16-ОКТ-96')(3); /* функцияни
чақириш */
    ...
END;
```

Коллекцияларға қиймат бериш ва солиштириш

Битта коллекцияни бошқасига қиймат сифатида INSERT, UPDATE, FETCH, ёки SELECT жумлалари ёрдамида бериш мумкин. Бунда коллекциялар бир хил турға эга бўлиши лозим. Бир хил турдаги Элементларининг бир хил турлилиги етарли эмас.

```
DECLARE
    TYPE Clientele IS VARRAY(100) OF Customer;
    TYPE Vips IS VARRAY(100) OF Customer;
    group1 Clientele := Clientele(...);
    group2 Clientele := Clientele(...);
    group3 Vips := Vips(...);
BEGIN
    group2 := group1;
    group3 := group2; -- нотўғри, берилганлар тури ҳар
хил
```

Атомик null коллекцияни бошқа коллекцияга берсангиз, бошқа коллекция атомик null бўлиб қолади (ва қайтадан инициализация қилиниши лозим). Қуйидаги мисолни қараймиз:

```
DECLARE
    TYPE Clientele IS TABLE OF Customer;
    group1 Clientele := Clientele(...); -- инициализация
    қилинди
    group2 Clientele; -- атомик null
BEGIN
    IF group1 IS NULL THEN ... -- шарт FALSE қайтаради
    group1 := group2;
    IF group1 IS NULL THEN ... -- шарт TRUE қайтаради
    ...
END;
```

Шунга ўхшаб, агар коллекцияга NULL қиймат берилса, коллекция атомик null бўлади.

Коллекция элементларига қиймат бериш

Коллекциянинг муайян элементига ифодали қийматни бериш учун қуйидаги синтаксисдан фойдаланинг:

Коллекция_номи(индекс) := ифода;

Бу ерда ифода коллекция турини тавсифида элемент учун аниқланган турдаги қийматни қайтаради. Агар индекс null ёки бутун турга айлантирилмайдиган бўлса, PL/SQL олдиндан аниқланган VALUE_ERROR номли мустаснони вужудга келтиради. Агар коллекция атомик null бўлса, PL/SQL COLLECTION_IS_NULL номли мустаснони вужудга келтиради. Бир нечта мисоллар келтирамиз:

```
DECLARE
    TYPE NumList IS TABLE OF INTEGER;
    nums NumList := NumList(10,20,30);
```

```

    ints NumList;
    ...
BEGIN
    ...
    nums(1) := TRUNC(high/low);
    nums(3) := nums(1);
    nums(2) := ASCII('B');
    nums('A') := 40; --VALUE_ERROR содир бўлади
    ints(1) := 15;    --COLLECTION_IS_NULL содир бўлади
END;
```

Коллекцияларни яхлит солиштириш

Ичма-ич жойлашган жадваллар ва массивлар атомик null бўлиши мумкин, шу сабабли уларни null бўлиш шартига текшириш мумкин:

```

DECLARE
    TYPE Staff IS TABLE OF Employee;
    members Staff;
BEGIN
    ...
    IF members IS NULL THEN ... - шарт TRUE қайтаради;
END;
```

Аммо, лекин коллекцияларни тенглик ёки тенгсизликка солиштириб бўлмайди. Масалан, қуйидаги IF шарти нотўғри:

```

DECLARE
    TYPE Clientele IS TABLE OF Customer;
    group1 Clientele := Clientele(...);
    group2 Clientele := Clientele(...);
BEGIN
    ...
    IF group1 = group2 THEN -- компиляция хатолиги
вужудга келади
    ...
```

```
END IF;
END;
```

Ушбу чегара ошкормас солишитиришлар учун ҳам қўлланилади. Масалан, коллекциялар DISTINCT, GROUP BY, ёки ORDER BY рўйхатида қатнашиши мумкин эмас.

Коллекцияларни бошқариш

PL/SQL-да коллекциялар қатор қулайликлар туғдиради. Жумладан, дастурингиз коллекцияларни хотирада сақлаш учун SQL фойдаланиб билади.

Ичма-ич жойлашган жадвалга доир мисоллар

Фараз қиламиз SQL*Plus, Course, объект турини қуйидагича аниқладингиз:

```
SQL> CREATE TYPE Course AS OBJECT (
2   course_no NUMBER(4),
3   title      VARCHAR2(35),
4   credits    NUMBER(1));
```

Сўнг, Course объектларини ўз ичида сақловчи TABLE туридаги CourseList номли турини аниқладингиз:

```
SQL> CREATE TYPE CourseList AS TABLE OF Course;
```

Ниҳоят, берилганлар базасидаги CourseList турдаги устун қатнашган department жадвали қуйидагича ташкил этилади:

```
SQL> CREATE TABLE department (
2   name      VARCHAR2(20),
3   director  VARCHAR2(20),
4   office    VARCHAR2(20),
5   courses   CourseList)
6   NESTED TABLE courses STORE AS courses_tab;
```

courses устундаги ҳар бир элемент ичма-ич жойлашган жадвал бўлиб, берилган бўлим тақдим этадиган курсларни сақлайди. NESTED TABLE жумласи орқали department жадвали ичма-ич жойлашган устунга эга эканлиги белгиланади. Ушбу жумла ичма-ич жойлашган жадвал ва берилганларни сақловчи тизим томонидан ишлаб чикувчи жадвал номини аниқлайди.

Энди department жадвалини тўлдиришингиз мумкин:

```
BEGIN
```

```
    INSERT INTO department
```

```
        VALUES('Psychology', 'Irene Friedman', 'Fulton
Hall 133',
```

```
    CourseList(Course(1000, 'General Psychology', 5),
    Course(2100, 'Experimental Psychology', 4),
    Course(2200, 'Psychological Tests', 3),
    Course(2250, 'Behavior Modification', 4),
    Course(3540, 'Groups and Organizations', 3),
    Course(3552, 'Human Factors in Busines', 4),
    Course(4210, 'Theories of Learning', 4),
    Course(4320, 'Cognitive Processes', 4),
    Course(4410, 'Abnormal Psychology', 4)));
```

```
    INSERT INTO department
```

```
VALUES('History', 'John Whalen', 'Applegate Hall 142',
    CourseList(Course(1011, 'History of Europe I', 4),
    Course(1012, 'History of Europe II', 4),
    Course(1202, 'American History', 5),
    Course(2130, 'The Renaissance', 3),
    Course(2132, 'The Reformation', 3),
    Course(3105, 'History of Ancient Greece', 4),
    Course(3321, 'Early Japan', 4),
    Course(3601, 'Latin America Since 1825', 4),
    Course(3702, 'Medieval Islamic History', 4)));
```

```

INSERT INTO department
VALUES('English', 'Lynn Saunders', 'Breakstone Hall
205',
      CourseList(Course(1002, 'Expository Writing', 3),
                  Course(2020, 'Film and Literature', 4),
                  Course(2418, 'Modern Science Fiction', 3),
                  Course(2810, 'Discursive Writing', 4),
                  Course(3010, 'Modern English Grammar', 3),
                  Course(3720, 'Introduction to Shakespeare', 4),
                  Course(3760, 'Modern Drama', 4),
                  Course(3822, 'The Short Story', 4),
                  Course(3870, 'The American Novel', 5)));
END;

```

Қуйидаги мисолда, Инглиз бўлими томонидан тақдим этилган курслар янгиланади:

```

DECLARE
new_courses CourseList :=
  CourseList(Course(1002, 'Expository Writing', 3),
             Course(2020, 'Film and Literature', 4),
             Course(2810, 'Discursive Writing', 4),
             Course(3010, 'Modern English Grammar', 3),
             Course(3550, 'Realism and Naturalism', 4),
             Course(3720, 'Introduction to Shakespeare', 4),
             Course(3760, 'Modern Drama', 4),
             Course(3822, 'The Short Story', 4),
             Course(3870, 'The American Novel', 4),
             Course(4210, '20th-Century Poetry', 4),
             Course(4725, 'Advanced Workshop in Poetry', 5));
BEGIN
  UPDATE department
  SET courses = new_courses WHERE name = 'English';

```

```
END;
```

Қуйидаги мисолда Физиология Бўлими такдам этган курсларни локал ичма-ич жойлашган жадвалга олиш намойиш этилган.

```
DECLARE
    psyc_courses CourseList;
BEGIN
    SELECT courses INTO psyc_courses FROM department
        WHERE name = 'Psychology';
    ...
END;
```

Массивларга мисоллар

Фараз қиламиз SQL*Plus Project объект турини қуйидагича аниқладингиз:

```
SQL> CREATE TYPE Project AS OBJECT (
2   project_no NUMBER(2),
3   title       VARCHAR2(35),
4   cost        NUMBER(7,2));
```

Сўнг Project объектларини ўз ичида сақловчи VARRAY туридаги ProjectList аниқладингиз:

```
SQL> CREATE TYPE ProjectList AS VARRAY(50) OF Project;
```

Ниҳоят, берилганлар базасидаги ProjectList турдаги устун қатнашган department жадвали қуйидагича ташкил этилади:

```
SQL> CREATE TABLE department (
2   dept_id  NUMBER(2),
3   name     VARCHAR2(15),
4   budget   NUMBER(11,2),
5   projects ProjectList);
```

Энди department жадвалини тўлдиришингиз мумкин:

```

BEGIN
    INSERT INTO department
        VALUES(30, 'Accounting', 1205700,
ProjectList(Project(1, 'Design New Expense Report',
3250),
        Project(2, 'Outsource Payroll', 12350),
        Project(3, 'Evaluate Merger Proposal', 2750),
        Project(4, 'Audit Accounts Payable', 1425)));
    INSERT INTO department
        VALUES(50, 'Maintenance', 925300,
        ProjectList(Project(1, 'Repair Leak in Roof',
2850),
        Project(2, 'Install New Door Locks', 1700),
        Project(3, 'Wash Front Windows', 975),
        Project(4, 'Repair Faulty Wiring', 1350),
        Project(5, 'Winterize Cooling System', 1125)));
    INSERT INTO department
        VALUES(60, 'Security', 750400,
        ProjectList(Project(1, 'Issue New Employee
Badges', 13500),
        Project(2, 'Find Missing IC Chips', 2750),
        Project(3, 'Upgrade Alarm System', 3350),
        Project(4, 'Inspect Emergency Exits', 1900)));
END;

```

Қийматни янгилашга доир мисол:

```

DECLARE
    new_projects ProjectList :=
        ProjectList(Project(1, 'Issue New Employee
Badges', 13500),
        Project(2, 'Develop New Patrol Plan', 1250),
        Project(3, 'Inspect Emergency Exits', 1900),
        Project(4, 'Upgrade Alarm System', 3350),

```

```

        Project(5, 'Analyze Local Crime Stats', 825));
BEGIN
    UPDATE department
        SET projects = new_projects WHERE dept_id = 60;
END;
```

Қийматларни локал массивга олишга доир мисол:

```

DECLARE
    my_projects ProjectList;
BEGIN
    SELECT projects INTO my_projects FROM department
        WHERE dept_id = 30;
    ...
END;
```

Department: жадвалидан қийматларни ўчиришга доир мисол:

```

BEGIN
    DELETE FROM department WHERE dept_id = 30;
END;
```

Индивидуал элементларни бошқариш

Агар сиз коллекциянинг алоҳида элементини бошқармоқчи бўлсангиз TABLE операторидан фойдаланинг. TABLE операнди қисмсуров бўлиб, ягона устун қийматини бошқариш учун қайтарилади. Ушбу қиймат массив ёки ичма-ич жойлашган жадвалдир.

Қуйидаги мисолда courses устунида сақловчи History Бўлими ичма-ич жойлашган жадвалга қатор қўшади:

```

BEGIN
    INSERT INTO
        TABLE(SELECT courses FROM department WHERE name =
'History')
        VALUES (3340, 'Modern China', 4);
```

END;

Қуйидаги мисолда алоҳида олинган коллекция элементини янгилаш намойиш этилган:

```
DECLARE
    adjustment INTEGER DEFAULT 1;
BEGIN
    UPDATE TABLE (SELECT courses FROM department
        WHERE name = 'Psychology')
        SET credits = credits + adjustment
        WHERE course_no IN (2200, 3540);
END;
```

Қуйидаги мисол алоҳида коллекция элементини сўров орқали олинад:

```
DECLARE
    my_course_no NUMBER(4);
    my_title VARCHAR2(35);
BEGIN
    SELECT course_no, title INTO my_course_no, my_title
        FROM TABLE (SELECT courses FROM department
            WHERE name = 'History')
        WHERE course_no = 3105;
    ...
END;
```

Коллекция алоҳида элементларини ўчириб ташлаш қуйидаги мисолда намойиш этилган:

```
BEGIN
    DELETE TABLE (SELECT courses FROM department
        WHERE name = 'English')
        WHERE credits = 5;
END;
```

Массив устунидан алоҳида элементларни сўровлаш:

```

DECLARE
    my_cost    NUMBER(7,2);
    my_title   VARCHAR2(35);
BEGIN
    SELECT cost, title INTO my_cost, my_title
        FROM TABLE(SELECT projects FROM department
                     WHERE dept_id = 50)
        WHERE project_no = 4;
    ...
END;
```

Айни пайтда массивнинг алоҳида элементларига INSERT, UPDATE, ёки DELETE жумлаларида мурожаат қилиб билмайсиз. statement. Шу мақсадда PL/SQL процедуравий жумлаларидан фойдаланишингиз лозим:

```

CREATE PROCEDURE add_project (
    dept_no      IN NUMBER,
    new_project  IN Project,
    position     IN NUMBER) AS
    my_projects  ProjectList;
BEGIN
    SELECT projects INTO my_projects FROM department
        WHERE dept_no = dept_id FOR UPDATE OF projects;
    my_projects.EXTEND; --янги проект у-н жой очиш    /*
    массив элементларини олдинга силжитиш*/
    FOR i IN REVERSE position..my_projects.LAST - 1 LOOP
        my_projects(i + 1) := my_projects(i);
    END LOOP;
    my_projects(position) := new_project;          UPDATE
    department SET projects = my_projects
        WHERE dept_no = dept_id;
END add_project;
```

Қуйидаги сақланадиган процедура жадвали қийматини ўзгартиради:

```

CREATE PROCEDURE update_project (
    dept_no    IN NUMBER,
    proj_no    IN NUMBER,
    new_title  IN VARCHAR2 DEFAULT NULL,
    new_cost   IN NUMBER DEFAULT NULL) AS
    my_projects ProjectList;
BEGIN
    SELECT projects INTO my_projects FROM department
        WHERE dept_no = dept_id FOR UPDATE OF projects;
    /* Проектни топиш, уни янгилаш, сўнг циклдан дарҳол
чиқиБ кетиш. */
    FOR i IN my_projects.FIRST..my_projects.LAST LOOP
        IF my_projects(i).project_no = proj_no THEN
            IF new_title IS NOT NULL THEN
                my_projects(i).title := new_title;
            END IF;
            IF new_cost IS NOT NULL THEN
                my_projects(i).cost := new_cost;
            END IF;
            EXIT;
        END IF;
    END LOOP;
    UPDATE department SET projects = my_projects
        WHERE dept_no = dept_id;
END update_project;

```

8. PL/SQL орқали Oracle билан ўзаро мулоқот қилиш

Ушбу бўлимда сиз PL/SQL тили ердамида SQL командалари, функциялари ва операторлари орқали Oracle берилганларини бошқаришни ўрганасиз. Шу билан бирга курсорларни бошқариш, курсор ўзгарувчиларидан фойдаланиш ва транзакция жараёнлари ҳам кўриб чиқилади.

SQL қўллаб қувватлаш

PL/SQL тили SQL тилидаги берилганларни бошқариш жумлалари, транзакцияни назорат қилувчи жумлаларини, функцияларини, псевдоустунларини ва операторларини қўллаб қувватлайди. Бундан ташқари динамик SQL ҳам қўллаб қувватланади.

Берилганларни бошқариш

Oracle берилганларини INSERT, UPDATE, DELETE, SELECT, ва LOCK TABLE командалари орқали бошқарилади. INSERT берилганлар базасидаги жадвалларга янги қатор қўшади; UPDATE қаторларни ўзгартиради; DELETE кераксиз қаторларни ўчириб ташлайди; SELECT сўровни бажаради ва LOCK TABLE вақтинча жадвалга киришни тақиқлаб қўяди.

Транзакцияни бошқариш

Яхлит мантикий ишни бажарувчи SQL берилганларни бошқариш жумлалари тўплами транзакция деб номланади.

COMMIT, ROLLBACK, SAVEPOINT ва SET TRANSACTION командалар орқали транзакцияларни бошқариш мумкин. COMMIT командаси жорий транзакция давомида базада юз берган ўзгаришларни фиксирлаб қўяди. ROLLBACK эса жорий транзакция ишини тўхтатиб, транзакция бошланган қадар ҳолатга қайтаради. SAVEPOINT транзакция жараёнининг жорий нуқтасини белгилайди. SET TRANSACTION транзакция хоссаларини ўрнатади.

SQL функциялари

PL/SQL тили барча SQL функцияларини жумладан AVG, COUNT, GROUPING, MAX, MIN, STDDEV, SUM, ва VARIANCE функцияларни ишлатишга имкон беради. Функция синтаксиси қуйидагича:

Функция_номи([ALL | DISTINCT] ифода)

Бу ерда ифода битта ёки ундан кўп устунларга мурожаат қилади. Агар ALL танласангиз, функция устуннинг барча қийматларини (жумладан такрорланганларини) инобатга олади. Агар DISTINCT танласангиз, функция устуннинг фақат фаркли қийматларини инобатга олади. Масалан қуйидаги жумла берилганлар базасидаги emp жадвалида жойлашган бир-биридан фарк қиладиган job қийматлар сонини қайтаради:

```
SELECT COUNT(DISTINCT job) INTO job_count FROM emp;
```

The function COUNT lets you specify the asterisk (*) option, which returns the number of rows in a table. For example, the following Қуйидаги мисол эса emp жадвалининг қаторлар сонини қайтаради:

```
SELECT COUNT(*) INTO emp_count FROM emp;
```

SQL Pseudocolumns

PL/SQL қуйидаги SQL псевдоустунларини ишлатиш мумкин: CURRVAL, LEVEL, NEXTVAL, ROWID, ва ROWNUM.

CURRVAL and NEXTVAL

Кетма-кетлик схема объекти бўлиб кетма кет сонларни ишлаб чиқади. CURRVAL аниқланган кетма-кетликдаги жорий қийматни қайтаради. NEXTVAL навбатдаги қийматни қайтаради. Мурожаат қилиш мисоллари:

```
Кетма_кетлик_номи.CURRVAL
```

```
Кетма_кетлик_номи.NEXTVAL
```

```
INSERT INTO emp VALUES (empno_seq.NEXTVAL, my_ename,
...);
```

```
INSERT INTO sals VALUES (empno_seq.CURRVAL, my_sal,
...);
```

ROWID

ROWID қаторнинг бинар адресини қайтаради. Бинар адресларни сақловчи ўзгарувчини UROWID турда эълон қилиш лозим:

```
DECLARE
```

```
    row_id UROWID;
```

ROWNUM

ROWNUM жадвалдан сўров орқали олинган қатор тартибини кўрсатади.

Масалан

```
DECLARE
```

```
    CURSOR c1 IS SELECT empno, sal FROM emp
```

```
        WHERE sal > 2000 AND ROWNUM < 10;
```

10 қатор қайтаради.

SQL операторлари

PL/SQL тили SQL солиштириш, тўплamlар билан ишлаш ва қатор операторларини барчасини ишлатишга имкон беради.

Солиштириш операторлари

Солиштириш операторлари қуйидаги жадвалда келтирилган:

Оператор	Тавсифи
ALL	Қийматни рўйхатда келтирилган қийматлар билан солиштиради ва TRUE қийматни қайтаради, агар барча индивидуал солиштиришлар TRUE қиймат қайтарса.
ANY, SOME	Қийматни рўйхатда келтирилган қийматлар билан солиштиради ва TRUE қийматни қайтаради, агар ихтиёрий индивидуал солиштириш TRUE қиймат қайтарса.
BETWEEN	Қиймат олдиндан аниқланган интервалда ётишини текширади.
EXISTS	TRUE қийматни қайтаради, агар сўровости камида битта қатор қайтарса.

IN	Тўпламга тегишлилигини текширади.
IS NULL	Null-га текширади.
LIKE	Белгили қаторда кўрсатилган қисм мавжудлигини текширади.

Тўплам операторлари

Тўплам операторлари иккита сўров натижаларини битта натижага бирлаштиради. INTERSECT иккала сўров томонидан суғуриб олинган натижаларда қатнашган фарқли қаторларни қайтаради. MINUS биринчи сўров суғуриб олган лекин иккинчи сўров натижасида қатнашмаган фарқли қаторларни қайтаради. UNION иккала сўров томонидан суғуриб олинган фарқли қаторларни қайтаради. UNION ALL иккала сўров томонидан суғуриб олинган барча қаторларни қайтаради.

Қатор операторлари

ALL сўров натижасида такрорланувчи қаторларни қолдиради. DISTINCT сўров натижасидан такрорланувчи қаторларни чиқариб ташлайди.

9. PL/SQLда курсорлар билан ишлаш

Курсорларни бошқариш

PL/SQL икки хил турдаги курсорлардан фойдаланади: ошкор ва ошкормас.

Ошкор курсорлар

Агарда сўров натижаси биттадан кўп қаторларни қайтарса, сиз ошкор тарзда курсор эълон қилишингиз мумкин. Курсорларни бошқариш учун урта команда мавжуд: OPEN, FETCH, and CLOSE. OPEN жумласи орқали курсор

сўров натижаси билан тўлдирилади. Сўнг FETCH жумласидан фойдаланиб биринчи қаторни олишингиз мумкин. Охирги қаторни олгандан сўнг CLOSE жумласи орқали курсорни ёпишингиз мумкин.

Курсорни эълон қилиш

Курсордан фойдаланишдан олдин уни қуйидаги синтаксис асосида эълон қилиш лозим:

CURSOR курсор_номи [(параметр[,параметр]...)]
[RETURN қайтариш_тури] IS select_жумласи;

Бу ерда қайтариш_тури берилганлар базасидаги жадвал қатори ёки ёзуви ва параметр қуйидаги синтаксисга эга:

курсор_параметри_номи [IN] берилганлар_тури [{:= | DEFAULT} ифода]

Масалан c1 ва c2 деб номланган курсорларни қуйидагича эълон қилишингиз мумкин:

```
DECLARE
    CURSOR c1 IS SELECT empno, ename, job, sal FROM emp
        WHERE sal > 2000;
    CURSOR c2 RETURN dept%ROWTYPE IS
        SELECT * FROM dept WHERE deptno = 10;
```

Параметрли курсорларга мисол келтирамиз:

```
DECLARE
    CURSOR c1 (low INTEGER DEFAULT 0,
        high INTEGER DEFAULT 99) IS SELECT ...
```

Курсорни очиш

Курсор очиш учун OPEN командасидан фойдаланиш лозим:

```
DECLARE
    CURSOR c1 IS SELECT ename, job FROM emp WHERE sal <
3000;
    ...
BEGIN
    OPEN c1;
    ...
END;
```

Курсор параметрларини узатиш

OPEN жумласи орқали курсорга параметрларни узатиш мумкин:

```
DECLARE
    emp_name emp.ename%TYPE;
    salary    emp.sal%TYPE;
    CURSOR c1 (name VARCHAR2, salary NUMBER) IS SELECT
...

OPEN c1(emp_name, 3000);
OPEN c1('ATTLEY', 1500);
OPEN c1(emp_name, salary);
```

Курсордан қаторларни олиш

FETCH жумласи орқали курсор қаторларини битта-битта суғуриб олиш мумкин:

```
FETCH c1 INTO my_empno, my_ename, my_deptno;
```

ёки

```
LOOP
    FETCH c1 INTO my_record;
    EXIT WHEN c1%NOTFOUND;
    -- process data record
END LOOP;
```

Сўровдаги ҳар бир ўзгарувчи курсор очилганда ҳисобланади. Қуйидаги мисолда ҳар бир SALARY қиймати 2 га кўпайтириб олинади:

```
DECLARE
    my_sal emp.sal%TYPE;
    my_job emp.job%TYPE;
    factor INTEGER := 2;
    CURSOR c1 IS SELECT factor*sal FROM emp WHERE job =
my_job;
BEGIN
    ...
    OPEN c1;  -- бу ерда factor 2-га тенг
    LOOP
        FETCH c1 INTO my_sal;
        EXIT WHEN c1%NOTFOUND;
        factor := factor + 1;  -- FETCH таъсири йўқ
    END LOOP;
END;
```

Курсор қийматини ўзгартириш учун қатор INTO жумлаларидан фойдаланиш мумкин:

```
DECLARE
  CURSOR c1 IS SELECT ename FROM emp;
  name1 emp.ename%TYPE;
  name2 emp.ename%TYPE;
  name3 emp.ename%TYPE;
BEGIN
  OPEN c1;
  FETCH c1 INTO name1; -- биринчи қатор олинади
  FETCH c1 INTO name2; -- иккинчи қатор олинади      FETCH
c1 INTO name3;      -- учинчи қатор олинади      ...
  CLOSE c1;
END;
```

Курсорни ёпиш

Курсорни ёпиш учун CLOSE жумласидан фойдаланиш лозим:

```
CLOSE c1;
```

Сўровостилардан фойдаланиш

Курсорларда сўровостилардан фойдаланиш мумкин:

Мисол 1.

```
DECLARE
  CURSOR c1 IS SELECT empno, ename FROM emp
    WHERE deptno IN (SELECT deptno FROM dept
    WHERE loc <> 'CHICAGO');
```

Мисол 2.

```
DECLARE
  CURSOR c1 IS SELECT t1.deptno, dname, "STAFF"
    FROM dept t1, (SELECT deptno, COUNT(*) "STAFF"
    FROM emp GROUP BY deptno) t2
    WHERE t1.deptno = t2.deptno AND "STAFF" >= 5;
```

Мисол 3.

```
DECLARE
  CURSOR c1 IS SELECT deptno, ename, sal FROM emp t
    WHERE sal > (SELECT AVG(sal) FROM emp WHERE
t.deptno = deptno)
    ORDER BY deptno;
```

Ошкормас курсорлар

Oracle ҳар бир SQL жумласини бажариш учун ошкормас тарзда курсор очади. Аммо OPEN, FETCH, ва CLOSE жумлалари ёрдамида SQL курсорларини бошқариб бўлмайди.

Курсорларни пакетларда бирлаштириш

Пакетда курсор аниқланиши ва танасини алоҳида ажратишингиз мумкин. Пакетдаги курсорнинг аниқланиши синтаксиси қуйидагича:

CURSOR курсор_номи [(параметр[,параметр]...)] RETURN қайтариш_тури;

Курсор аниқланишида %ROWTYPE фойдалани мумкин:

```
CREATE PACKAGE emp_actions AS
    /* Курсор аниқланиш соҳасини эълон қилиш */
    CURSOR c1 RETURN emp%ROWTYPE;
    ...
END emp_actions;

CREATE PACKAGE BODY emp_actions AS
    /* Курсор танасини эълон қилиш */
    CURSOR c1 RETURN emp%ROWTYPE IS
        SELECT * FROM emp
            WHERE sal > 3000;
    ...
END emp_actions;
```

Келажакда курсор аниқланишини ўзгартирмасдан туриб уни танасини ўзгартиришингиз мумкин:

```
CREATE PACKAGE BODY emp_actions AS
    /* Курсор танасини эълон қилиш */
    CURSOR c1 RETURN emp%ROWTYPE IS
        SELECT * FROM emp
            WHERE deptno = 20;  -- янги WHERE жумласи ...
END emp_actions;
```

FOR Loops курсоридан фойдаланиш

Кўп ҳолларда курсорни ошкор тарзда эълон қилишда OPEN, FETCH ва CLOSE жумлалари ўрнига FOR Loop курсоридан фойдаланишингиз мумкин.

FOR loop цикл индексини %ROWTYPE ёзув кўринишида ошкормас тарзда эълон қилиб, курсорни очади, қаторларни кетма-кет суғуриб ёзувга олади ва барча қаторлар олингандан сўнг курсорни ёпади:

```
DECLARE
    result temp.col1%TYPE;
    CURSOR c1 IS
        SELECT n1, n2, n3 FROM data_table WHERE exper_num = 1;
BEGIN
    FOR c1_rec IN c1 LOOP
        /* ҳисоблайди ва натижаларни сақлайди */
        result := c1_rec.n2 / (c1_rec.n1 + c1_rec.n3);
        INSERT INTO temp VALUES (result, NULL, NULL);
    END LOOP;
    COMMIT;
END;
```

ёзув фақат циклнинг ичида аниқланган холос ва унинг майдонларига циклдан ташқарида мурожаат қилиб бўлмайди:

```
FOR c1_rec IN c1 LOOP
    ...
END LOOP;
result := c1_rec.n2 + 3;  -- қоидага ҳилоф
```

Сўровостилардан фойдаланиш

FOR loop жумласида сўровостилардан фойдаланиш мумкин:

```
DECLARE
    bonus REAL;
BEGIN
    FOR emp_rec IN (SELECT empno, sal, comm FROM emp) LOOP
        bonus := (emp_rec.sal * 0.05) + (emp_rec.comm * 0.25);
        INSERT INTO bonuses VALUES (emp_rec.empno, bonus);
    END LOOP;
    COMMIT;
END;
```

Ҳамроҳ номлардан фойдаланиш

SELECT жумласида ифодаларни ҳамроҳ ном орқали белиглаш мумкин:

```
CURSOR c1 IS
    SELECT empno, sal+NVL(comm,0), job FROM ...
```

Қуйидаги мисолда sal+NVL(comm,0) ифода wages ҳамроҳ ном орқали белигиланган:

```
CURSOR c1 IS
  SELECT empno, sal+NVL(comm,0) wages, job FROM ...
```

Мурожаатда ҳамроҳ номидан фойдаланиш мумкин:

```
IF emp_rec.wages < 1000 THEN ...
```

Параметрларни узатиш

FOR loop курсорига параметрларни узатиш мумкин:.

```
DECLARE
  CURSOR emp_cursor(dnum NUMBER) IS
    SELECT sal, comm FROM emp WHERE deptno = dnum;
  total_wages NUMBER(11,2) := 0;
  high_paid    NUMBER(4)    := 0;
  higher_comm  NUMBER(4)    := 0;
BEGIN
  /* итерациялар сони emp_cursor қайтарадиган қаторлар
  сонига тенг бўлиши лозим */
  FOR emp_record IN emp_cursor(20) LOOP
    emp_record.comm := NVL(emp_record.comm, 0);
    total_wages := total_wages + emp_record.sal +
      emp_record.comm;
    IF emp_record.sal > 2000.00 THEN
      high_paid := high_paid + 1;
    END IF;
    IF emp_record.comm > emp_record.sal THEN
      higher_comm := higher_comm + 1;
    END IF;
  END LOOP;
  INSERT INTO temp VALUES (high_paid, higher_comm,
    'Total Wages: ' || TO_CHAR(total_wages));
  COMMIT;
END;
```

Курсор ўзгарувчиларидан фойдаланиш

PL/SQL тилида курсор ўзгарувчиси мавжуд.

Курсор ўзгарувчиси нима?

Курсор ўзгарувчиси худди C ёки Паскал кўрсаткичларига ўхшаб бирор элементнинг ўрнига уни хотирадаги адресини сақлайди.

Нима учун курсор ўзгарувчисидан фойдаланиш керак?

Кўпинча сиз курсор ўзгарувчисини сўров натижасида олинган тўпламни PL/SQL қисмдастуридан ҳар хил мижозларга узатиш учун фойдаланасиз.

REF CURSOR турини аниқлаш

Курсор ўзгарувчиси икки қадамда аниқланади. Аввал сиз REF CURSOR турини аниқлаб, сўнг ушбу турдаги курсор ўзгарувчисини эълон қиласиз. REF CURSOR турини аниқлаш синтаксиси:

TYPE ref_тури_номи IS REF CURSOR RETURN қайтариш_тури;

Мисол.

```
DECLARE
    TYPE DeptCurTyp IS REF CURSOR RETURN dept%ROWTYPE;
```

REF CURSOR тури кучли ёки кучсиз бўлиши мумкин:

```
DECLARE
    TYPE EmpCurTyp IS REF CURSOR RETURN emp%ROWTYPE;    --
кучли
    TYPE GenericCurTyp IS REF CURSOR;    -- кучсиз
```

Курсор ўзгарувчисини эълон қилиш

Бир марта REF CURSOR турини аниқлаб, ушбу турдаги курсор ўзгарувчиларини ихтиёрий:

```
DECLARE
    TYPE DeptCurTyp IS REF CURSOR RETURN dept%ROWTYPE;
    dept_cv DeptCurTyp;    -- курсор ўзгарувчисини эълон
қилиш
```

Изоҳ: Курсор ўзгарувчисини пакетда эълон қилиш қилиш мумкин эмас.

Мураккаб мисоллар:

I)

```
DECLARE
    TYPE TmpCurTyp IS REF CURSOR RETURN emp%ROWTYPE;
    tmp_cv TmpCurTyp;    -- курсор ўзгарувчисини эълон
қилиш
    TYPE EmpCurTyp IS REF CURSOR RETURN tmp_cv%ROWTYPE;
    emp_cv EmpCurTyp;    -- курсор ўзгарувчисини эълон
қилиш
```

II)

```

DECLARE
    dept_rec dept%ROWTYPE; -- курсор ўзгарувчисини эълон қилиш
    TYPE DeptCurTyp IS REF CURSOR RETURN dept_rec%TYPE;
    dept_cv DeptCurTyp;    -- курсор ўзгарувчисини эълон қилиш

```

III)

```

DECLARE
    TYPE EmpRecTyp IS RECORD (
        empno NUMBER(4),
        ename VARCHAR2(10),
        sal    NUMBER(7,2));
    TYPE EmpCurTyp IS REF CURSOR RETURN EmpRecTyp;
    emp_cv EmpCurTyp;    -- курсор ўзгарувчисини эълон қилиш.

```

Курсор ўзгарувчиси параметр сифатида

Курсор ўзгарувчисини функция ва процедуранинг формал парметри сифатида эълон қилишингиз мумкин:

```

DECLARE
    TYPE EmpCurTyp IS REF CURSOR RETURN emp%ROWTYPE;
    PROCEDURE open_emp_cv (emp_cv IN OUT EmpCurTyp) IS
    ...

```

Курсор ўзгарувчиларини бошқариш

OPEN-FOR, FETCH, ва CLOSE жумлалари орқали курсор ўзгарувчисини бошқариш мумкин.

Курсор ўзгарувчисини очиш

OPEN-FOR жумласи ўзгарувчисини кўп-қаторли сўров билан боғлайди, сўровни бажаради ва натижавий тўпламни аниқлайди. Унинг синтаксиси қуйидагича:

```

OPEN {курсор_ўзгарувчиси_номи |
:host_курсор_ўзгарувчиси_номи }
FOR select_жумласи;

```

Масалан:

```

IF NOT emp_cv%ISOPEN THEN
    /* Курсор ўзгарувчисини очиш */

```

```

    OPEN emp_cv FOR SELECT * FROM emp;
END IF;

```

Курсор ўзгарувчиси формал параметр сифатида:

```

CREATE PACKAGE emp_data AS
    ...
    TYPE EmpCurTyp IS REF CURSOR RETURN emp%ROWTYPE;
    PROCEDURE open_emp_cv (emp_cv IN OUT EmpCurTyp);
END emp_data;

CREATE PACKAGE BODY emp_data AS
    ...
    PROCEDURE open_emp_cv (emp_cv IN OUT EmpCurTyp) IS
    BEGIN
        OPEN emp_cv FOR SELECT * FROM emp;
    END open_emp_cv;
END emp_data;

```

Курсор ўзгарувчиларини очиш учун алоҳида-сақланадиган
процедуралардан ҳам фойдаланиш мумкин:

```

CREATE PACKAGE cv_types AS
    TYPE GenericCurTyp IS REF CURSOR;
    TYPE EmpCurTyp IS REF CURSOR RETURN emp%ROWTYPE;
    TYPE DeptCurTyp IS REF CURSOR RETURN dept%ROWTYPE;
    ...
END cv_types;

```

Пакетда аниқланган курсордан фойдаланиш:

```

CREATE PROCEDURE open_emp_cv (emp_cv IN OUT
cv_types.EmpCurTyp) AS
BEGIN
    OPEN emp_cv FOR SELECT * FROM emp;
END open_emp_cv;

```

Шартга қараб курсорни очиш:

```

CREATE PACKAGE emp_data AS
    TYPE GenericCurTyp IS REF CURSOR;
    TYPE EmpCurTyp IS REF CURSOR RETURN emp%ROWTYPE;
    PROCEDURE open_emp_cv (emp_cv IN OUT EmpCurTyp,
choice NUMBER);
END emp_data;

CREATE PACKAGE BODY emp_data AS
    PROCEDURE open_emp_cv (
        emp_cv IN OUT EmpCurTyp,
        choice NUMBER) IS
    BEGIN

```

```

        IF choice = 1 THEN
            OPEN emp_cv FOR SELECT * FROM emp WHERE comm IS
NOT NULL;
        ELSIF choice = 2 THEN
            OPEN emp_cv FOR SELECT * FROM emp WHERE sal >
2500;
        ELSIF choice = 3 THEN
            OPEN emp_cv FOR SELECT * FROM emp WHERE deptno
= 20;
        END IF;
    END open_emp_cv;
END emp_data;

```

Қулайроқ бўлиши учун курсор ўзгарувчисини ҳар хил турни қайтарувчи сўровларни бажарувчи процедурага узатишингиз мумкин:

```

CREATE PACKAGE BODY emp_data AS
    PROCEDURE open_cv (
        generic_cv IN OUT GenericCurTyp,
        choice     NUMBER) IS
    BEGIN
        IF choice = 1 THEN
            OPEN generic_cv FOR SELECT * FROM emp;
        ELSIF choice = 2 THEN
            OPEN generic_cv FOR SELECT * FROM dept;
        ELSIF choice = 3 THEN
            OPEN generic_cv FOR SELECT * FROM salgrade;
        END IF;
    END open_cv;
END emp_data;

```

Курсор ўзгарувчисидан қаторларни олиш

FETCH жумласи кўп-қаторли сўровнинг натижавий тўпламидан қаторларни битта-битта олиш учун ишлатилади. Унинг синтаксиси:

```

FETCH {курсор_ўзгарувчиси_номи | :host_
курсор_ўзгарувчиси_номи }
    INTO { ўзгарувчи_номи [,ўзгарувчи_номи]... |
ёзув_номи};

```

Масалан:

```

LOOP
    /* курсор ўзгарувчисидан қатор олиш. */
    FETCH emp_cv INTO emp_rec;
    EXIT WHEN emp_cv%NOTFOUND; /* охирги қатор олиниши
билан цикл тугатилади */
    --берилганларни ёзиш жараёни

```

```
END LOOP;
```

Курсор ўзгарувчисини ёпиш

CLOSE жумласи курсор ўзгарувчисини ёпади. Унинг синтаксиси:

```
CLOSE      {      курсор_ўзгарувчиси_номи      |      :host_
курсор_ўзгарувчиси_номи) ;
```

Масалан:

```
LOOP
    FETCH emp_cv INTO emp_rec;
    EXIT WHEN emp_cv%NOTFOUND;
    -- берилганларни ёзиш жараёни
END LOOP;
/* курсор_ўзгарувчисини_ёпиш. */
CLOSE emp_cv;
```

1 мисол

Берилганлар базасидаги асосий библиотека даврий китоблари излаб топиб берувчи процедурани қараб чиқамиз.

```
CREATE PACKAGE cv_types AS
    TYPE LibCurTyp IS REF CURSOR;
    ...
END cv_types;

CREATE PROCEDURE find_item (title  VARCHAR2(100),
                           lib_cv  IN OUT
cv_types.LibCurTyp) AS
    code BINARY_INTEGER;
BEGIN
    SELECT item_code FROM titles INTO code
        WHERE item_title = title;
    IF code = 1 THEN
        OPEN lib_cv FOR SELECT * FROM books
            WHERE book_title = title;
    ELSIF code = 2 THEN
        OPEN lib_cv FOR SELECT * FROM periodicals
            WHERE periodical_title = title;
    ELSIF code = 3 THEN
        OPEN lib_cv FOR SELECT * FROM tapes
            WHERE tape_title = title;
    END IF;
END find_item;
```

2 мисол

Библиотека филиалидаги мижоз томонидаги дастур олинган ахборотни кўриш учун қуйидаги PL/SQL блокидан фойдаланиш мумкин.

```
DECLARE
    lib_cv          cv_types.LibCurTyp;
    book_rec        books%ROWTYPE;
    periodical_rec  periodicals%ROWTYPE;
    tape_rec        tapes%ROWTYPE;
BEGIN
    get_title(:title);  -- title is a host variable
    find_item(:title, lib_cv);
    FETCH lib_cv INTO book_rec;
    display_book(book_rec);
EXCEPTION
    WHEN ROWTYPE_MISMATCH THEN
        BEGIN
            FETCH lib_cv INTO periodical_rec;
            display_periodical(periodical_rec);
        EXCEPTION
            WHEN ROWTYPE_MISMATCH THEN
                FETCH lib_cv INTO tape_rec;
                display_tape(tape_rec);
        END;
END;
```

3 мисол

Pro*C дастурига мисол:

```
#include <stdio.h>
#include <sqlca.h>
void sql_error();
main()
{
    char temp[32];
    EXEC SQL BEGIN DECLARE SECTION;
    char * uid = "scott/tiger";
    SQL_CURSOR generic_cv;  /* cursor variable */
    int table_num;          /* selector */
    struct                  /* EMP record */
    {
        int emp_num;
        char emp_name[11];
        char job_title[10];
```

```

        int    manager;
        char   hire_date[10];
        float  salary;
        float  commission;
        int    dept_num;
    } emp_rec;
    struct          /* DEPT record    */
    {
        int    dept_num;
        char   dept_name[15];
        char   location[14];
    } dept_rec;
    struct          /* BONUS record    */
    {
        char   emp_name[11];
        char   job_title[10];
        float  salary;
    } bonus_rec;
EXEC SQL END DECLARE SECTION;
/* Handle Oracle errors. */
EXEC SQL WHENEVER SQLERROR DO sql_error();

/* Connect to Oracle. */
EXEC SQL CONNECT : uid;

/* Initialize cursor variable. */
EXEC SQL ALLOCATE : generic_cv;

/* Exit loop when done fetching. */
EXEC SQL WHENEVER NOT FOUND DO break;

for (;;)
{
    printf("\n1 = EMP, 2 = DEPT, 3 = BONUS");
    printf("\nEnter table number (0 to quit): ");
    gets(temp);
    table_num = atoi(temp);
    if (table_num <= 0) break;

    /* Open cursor variable. */
    EXEC SQL EXECUTE
        BEGIN
        IF : table_num = 1 THEN
            OPEN : generic_cv FOR SELECT * FROM emp;

```

```

ELSIF:table_num = 2 THEN
OPEN : generic_cv FOR SELECT * FROM dept;
ELSIF:table_num = 3 THEN
OPEN : generic_cv FOR SELECT * FROM bonus;
    END IF;
    END;
    END - EXEC;
    for (;;)
    {
        switch (table_num)
        {
        case 1: /* Fetch row into EMP record. */
EXEC SQL FETCH : generic_cv INTO : emp_rec;
break;
        case 2: /* Fetch row into DEPT record. */
EXEC SQL FETCH : generic_cv INTO : dept_rec;
break;
        case 3: /* Fetch row into BONUS record. */
EXEC SQL FETCH : generic_cv INTO : bonus_rec;
break;
        }
        /* Process data record here. */
    }
    /* Close cursor variable. */
EXEC SQL CLOSE : generic_cv;
}
exit(0);
}
void sql_error()
{
    /* Handle SQL error here. */
}

```

4 мисол

Host ўзгарувчисини масалан SQL*Plus –да VARIABLE командаси орқали эълон қилиш мумкин:

VARIABLE return_code NUMBER

PL/SQL Host ўзгарувчисига мурожаат қилиш учун уни олдида икки нукта (:) қўйиш лозим:

BEGIN

```

:return_code := 0;
IF credit_check_ok(acct_no) THEN
    :return_code := 1;
END IF;
...
END;

```

SQL*Plus Host ўзгарувчиси қийматини кўриш учун PRINT командасидан фойдаланинг:

```
SQL> PRINT return_code
```

```
RETURN_CODE
```

```
-----
```

```
1
```

SQL*Plus -даги REFCURSOR берилганлар тури курсор ўзгарувчисини эълон қилишга имкон беради.

```

CREATE PACKAGE emp_data AS
    TYPE EmpRecTyp IS RECORD (
        emp_id      NUMBER(4),
        emp_name    CHAR(10),
        job_title   CHAR(9),
        dept_name   CHAR(14),
        dept_loc    CHAR(13));
    TYPE EmpCurTyp IS REF CURSOR RETURN EmpRecTyp;
    PROCEDURE get_staff (
        dept_no IN NUMBER,
        emp_cv IN OUT EmpCurTyp);
END;
/
CREATE PACKAGE BODY emp_data AS
    PROCEDURE get_staff (
        dept_no IN NUMBER,
        emp_cv IN OUT EmpCurTyp) IS
    BEGIN
        OPEN emp_cv FOR
            SELECT empno, ename, job, dname, loc FROM emp,
dept
                WHERE emp.deptno = dept_no AND emp.deptno =
dept.deptno
                ORDER BY empno;
    END;
END;
/
COLUMN EMPNO HEADING Number

```

```

COLUMN ENAME HEADING Name
COLUMN JOB HEADING JobTitle
COLUMN DNAME HEADING Department
COLUMN LOC HEADING Location
SET AUTOPRINT ON
VARIABLE cv REFCURSOR
EXECUTE emp_data.get_staff(20, :cv)

```

Курсор атрибутларидан фойдаланиш

Хар бир курсор ёки курсор ўзгарувчиси тўртта атрибутга эга: %FOUND, %ISOPEN %NOTFOUND ва %ROWCOUNT. Ушбу атрибутлар курсор ҳақида фойдали ахборотлар беради.

Ошкор курсор атрибутлари

%FOUND

Курсор ёки курсор ўзгарувчиси очилганда, лекин биринчи қатор олингунга қадар %FOUND атрибут NULL қийматни қайтаради. Сўнг қатор олинса TRUE, қатор тугаса FALSE қийматни қайтаради.

```

LOOP
    FETCH c1 INTO my_ename, my_sal, my_hiredate;
    IF c1%FOUND THEN  -- қатор олинди
        ...
    ELSE  -- қатор олиш муваффақиятсиз бўлди
        EXIT;
    END IF;
END LOOP;

```

%ISOPEN

%ISOPEN атрибут TRUE қиймат қайтаради, агар курсор ёки курсор ўзгарувчиси очилган бўлса, акс ҳолда %ISOPEN атрибут FALSE қиймат қайтаради.

```

IF c1%ISOPEN THEN  -- курсор очиқ
    ...
ELSE  -- курсор ёпиқ
    OPEN c1;
END IF;

```

%NOTFOUND

%NOTFOUND мантиқан %FOUND қарама қаршидир. %NOTFOUND атрибут FALSE қиймат қайтаради, агар охирги уриншида қатор олинган бўлса, акс ҳолда TRUE қиймат қайтаради.

```
LOOP
  FETCH c1 INTO my_ename, my_sal, my_hiredate;
  EXIT WHEN c1%NOTFOUND;
  ...
END LOOP;
```

Биринчи қатор олгунга қадар %NOTFOUND атрибути NULL қийматга эга. Ушбу хавфдан ҳимоялаш учун қуйидаги жумладан фойдаланишингизни тавсия берамиз:

```
EXIT WHEN c1%NOTFOUND OR ci%NOTFOUND IS NULL;
```

%ROWCOUNT

%ROWCOUNT олинган қаторлар сонини қайтаради.

```
LOOP
  FETCH c1 INTO my_ename, my_deptno;
  IF c1%ROWCOUNT > 10 THEN
    ...
  END IF;
  ...
END LOOP;
```

Курсор атрибутлари қийматлари

		%FOUND	%ISOPEN	%NOTFOUND	%ROWCOUNT
OPEN	before	exception	FALSE	exception	exception
	after	NULL	TRUE	NULL	0
First FETCH	before	NULL	TRUE	NULL	0
	after	TRUE	TRUE	FALSE	1

Next FETCH(es)	before	TRUE	TRUE	FALSE	1
	after	TRUE	TRUE	FALSE	data dependent
Last FETCH	before	TRUE	TRUE	FALSE	data dependent
	after	FALSE	TRUE	TRUE	data dependent
CLOSE	before	FALSE	TRUE	TRUE	data dependent
	after	exception	FALSE	exception	exception

Мисоллар

1 мисол.

```

DECLARE
    num1    data_table.n1%TYPE;
    num2    data_table.n2%TYPE;
    result  temp.col1%TYPE;
    CURSOR c1 IS
        SELECT n1, n2, n3 FROM data_table WHERE exper_num
= 1;
BEGIN
    OPEN c1;
    LOOP
        FETCH c1 INTO num1, num2, num3;
        EXIT WHEN c1%NOTFOUND;
        result := num2/(num1 + num3);
        INSERT INTO temp VALUES (result, NULL, NULL);
    END LOOP;
    CLOSE c1;
    COMMIT;
END;
```

2 мисол.

```

DECLARE
    CURSOR bin_cur(part_number NUMBER) IS
```

```

SELECT amt_in_bin FROM bins
  WHERE part_num = part_number AND amt_in_bin > 0
  ORDER BY bin_num
  FOR UPDATE OF amt_in_bin;
bin_amt      bins.amt_in_bin%TYPE;
total_so_far  NUMBER(5) := 0;
amount_needed CONSTANT NUMBER(5) := 1000;
bins_looked_at NUMBER(3) := 0;
BEGIN
  OPEN bin_cur(5469);
  WHILE total_so_far < amount_needed LOOP
    FETCH bin_cur INTO bin_amt;
    EXIT WHEN bin_cur%NOTFOUND;
    -- if we exit, there's not enough to fill the
order
    bins_looked_at := bins_looked_at + 1;
    IF total_so_far + bin_amt < amount_needed THEN
      UPDATE bins SET amt_in_bin = 0
        WHERE CURRENT OF bin_cur;
      -- take everything in the bin
      total_so_far := total_so_far + bin_amt;
    ELSE -- we finally have enough
      UPDATE bins SET amt_in_bin = amt_in_bin
        - (amount_needed - total_so_far)
        WHERE CURRENT OF bin_cur;
      total_so_far := amount_needed;
    END IF;
  END LOOP;

  CLOSE bin_cur;
  INSERT INTO temp
    VALUES (NULL, bins_looked_at, '<- bins looked
at');
  COMMIT;
END;
```

Ошкормас курсор атрибутлари

Ошкормас курсор атрибутлари INSERT, UPDATE, DELETE ёки SELECT INTO жумлаларнинг бажарилиши ҳақидаги ахборотларни қайтаради.

%FOUND

SQL жумласи бажарилгунга қадар %FOUND атрибути NULL қиймат қайтаради. Сўнг %FOUND атрибути TRUE қиймат қайтаради, агар INSERT, UPDATE ёки DELETE жумласи битта ёки ундан кўп қаторларга таъсир қилса ёки SELECT INTO жумласи битта ёки ундан кўп қаторларни

қайтарса. Акс ҳолда %FOUND атрибути FALSE қиймат қайтаради. Қуйидаги мисолда қаторларни йўқотиш амали муваффақиятли бажарилса янги қаторни ёзиш амали бажарилади:

```
DELETE FROM emp WHERE empno = my_empno;
IF SQL%FOUND THEN  -- йўқотиш муваффақиятли
    INSERT INTO new_emp VALUES (my_empno, my_ename, ...);
```

%ISOPEN

Oracle SQL курсорини уни бажаришдан сўнг автомат тарзда ёпади. Шу сабабли %ISOPEN атрибут ҳамма вақт FALSE қиймат қайтаради..

%NOTFOUND

%NOTFOUND мантиқан %FOUND атрибутининг қарама-қаршисидир.

%ROWCOUNT

%ROWCOUNT атрибути INSERT, UPDATE ёки DELETE жумлалари томнидан таъсир қилган қаторлар сонини SELECT INTO жумласи қайтарган қаторлар сонини қайтаради.

```
DELETE FROM emp WHERE ...
IF SQL%ROWCOUNT > 10 THEN --10 кўп қаторлар ўчирилган
    ...
END IF;
```

10.Хатоликларга ишлов бериш

Ушбу бўлимда биз қуйидаги мавзуларга тўхталиб ўтамиз:

Белгилар тўплами;

[Хатоликлар](#)

Мустасноларнинг ижобий томонлари

Олдиндан аниқланган мустаснолар

Фойдаланувчи томонидан аниқланган мустаснолар

Мустаснолар қандай пайдо бўлади

Мустаснолар қандай тарқалади

Мустанолар қандай такроран вужудга келади
 Вужудга келган мустасноларга ишлов бериш
 Фойдали усуллар

Хатоликлар

PL/SQL-да хато содир бўлиш шарти мустасно дейилади.

Мустаснолар тизим томонидан ёки фойдаланувчи томонидан аниқланиши мумкин. Тизим томонидан аниқланган мустаснога мисол сифатида *нолга бўлиш* ва *хотира тўлиб кетиши* кабиларни келитирш мумкин. Баъзи умумий тизим томонидан аниқланган мустаснолар `ZERO_DIVIDE` ва `STORAGE_ERROR` каби олдиндан аниқланган номларга эга.

Сиз ҳам фойдаланувчи сифатида ихтиёрий PL/SQL блокида, қисмдастурда ёки пакетда мустаснони аниқлашингиз мумкин. Фойдаланувчи томонидан аниқланган мустасно албатта номга эга бўлиши лозим.

Хатолик юз бериши билан мустасно *содир* бўлади. Тизим томонидан аниқланган мустаснолар автоматик тарзда тизим ишлаши жараёнида содир бўлади. Фойдаланувчи томонидан аниқланган мустасноларни эса сунъий тарзда `RAISE` жумласи ёрдамида содир этмоқ зарур.

Содир бўлган мустасноларга ишлов бериш учун алоҳида *мустасно ишлов берувчиси* деб номланувчи дастурни ёзишингиз лозим. Ишлов берувчи ишга тушишиши билан, жорий блокнинг бажарилиши тўхтатилади. Ишлов берувчи ишини тугатгандан сўнг бошқарув навбатдаги блокка узатилади.

Мисол:

```
DECLARE
    pe_ratio NUMBER(3,1);
BEGIN
    SELECT price / earnings INTO pe_ratio FROM stocks
        WHERE symbol = 'XYZ'; /* нолга бўлиш
                               хатолиги юз бериши мумкин */
    INSERT INTO stats (symbol, ratio) VALUES ('XYZ',
        pe_ratio);
```

```

COMMIT;
EXCEPTION  -- мустасно ишлов берувчиси бошланиши
    WHEN ZERO_DIVIDE THEN  /* 'нолга бўлиш'
                           хатолигига ишлов беради */
        INSERT INTO stats (symbol, ratio) VALUES ('XYZ',
NULL);
    COMMIT;
    ...
    WHEN OTHERS THEN  /* бошқа барча хатоликларга
                       ишлов беради */

        ROLLBACK;
END; -- мустасно ишлов берувчиси ва блокнинг охири

```

Нолга бўлиш хатолигини қуйидагича тузатиш мумкин:

```

INSERT INTO stats (symbol, ratio)
    SELECT symbol, DECODE(earnings, 0, NULL, price /
earnings)
    FROM stocks WHERE symbol = 'XYZ';

```

Мустасноларнинг ижобий томонлари

Хатоликларга ишлов бериш учун мустаснолардан фойдаланиш қатор ижобий томонларга эга. Агарда мустасно ишлов берувчисидан фойдаланмасангиз, ҳар галда хатолик юз беришини текширишингиз лозим бўлади:

```

BEGIN
    SELECT ...
    -- 'берилганлар топилмади' хатолик юз беришини текшириш
    SELECT ...
    -- 'берилганлар топилмади' хатолик юз беришини текшириш
    SELECT ...
    -- 'берилганлар топилмади' хатолик юз беришини текшириш

```

Агар мустасно ишлов берувчисидан фойдалансангиз, ушбу код қисқаради:

```

BEGIN
    SELECT ...
    SELECT ...
    SELECT ...
    ...
EXCEPTION
    WHEN NO_DATA_FOUND THEN  -- барча 'берилганлар
топилмади'
    хатоликларини ушлайди

```

Олдиндан аниқланган хатоликлар

Тизим томонидан олдиндан аниқланган мустаснолар, ошкормас тарзда, PL/SQL дастури Oracle тизими қоидаларини бузганда содир этилади. Хар бир Oracle хатолиги ўзининг тартиб номерига эга бўлиб, лекин номи орқали ищлов берилиши лозим. PL/SQL бир қатор Oracle умумий хатоликларини мустаснолар тарзда олдиндан аниқлаб қўйган. Масалан агарда SELECT INTO жумласи қатор қайтармаса, унда PL/SQL олдиндан аниқланган NO_DATA_FOUND мустаснони содир этади.

Бошқа Oracle хатоликларига ищлов бериш учун, сиз OTHERS ищлов берувчидан фойдаланишингиз мумкин. Хусусан SQLCODE ва SQLERRM функциялари OTHERS ищлов берувчи учун жуда фойдали бўлади, чунки Oracle хатолиги коди ва хабари матнини қайтаради. Алтернатив равишда сиз EXCEPTION_INIT прагмасидан фойдаланиб мустасно номини Oracle хатолиги коди билан боғлашингиз мумкин.

PL/SQL олдиндан аниқланган мустасноларни глобал равишда STANDARD пакетида эълон қилади. Шу сабабли, уларни сиз эълон қилишингизга зарурат йўқ. Сиз номлари қуйидаги рўйхатда келтирилган олдиндан аниқланган мустаснолар учун ищлов берувчиларни ёзишингиз мумкин. Мос Oracle хатолиги коди ва SQLCODE қайтарадиган қийматлар ҳам келтирилган.

Мустасно номи	Oracle хатолиги	SQLCODE қиймати
---------------	-----------------	-----------------

ACCESS_INTO_NULL	ORA-06530	-6530
COLLECTION_IS_NULL	ORA-06531	-6531
CURSOR_ALREADY_OPEN	ORA-06511	-6511
DUP_VAL_ON_INDEX	ORA-00001	-1
INVALID_CURSOR	ORA-01001	-1001
INVALID_NUMBER	ORA-01722	-1722
LOGIN_DENIED	ORA-01017	-1017
NO_DATA_FOUND	ORA-01403	+100
NOT_LOGGED_ON	ORA-01012	-1012
PROGRAM_ERROR	ORA-06501	-6501
ROWTYPE_MISMATCH	ORA-06504	-6504
SELF_IS_NULL	ORA-30625	-30625
STORAGE_ERROR	ORA-06500	-6500
SUBSCRIPT_BEYOND_COUNT	ORA-06533	-6533
SUBSCRIPT_OUTSIDE_LIMIT	ORA-06532	-6532
SYS_INVALID_ROWID	ORA-01410	-1410
TIMEOUT_ON_RESOURCE	ORA-00051	-51
TOO_MANY_ROWS	ORA-01422	-1422
VALUE_ERROR	ORA-06502	-6502
ZERO_DIVIDE	ORA-01476	-1476

Олдиндан аниқланган мустасноларнинг тавсифи:

Мустасно номи	Содир бўлади,агар ...
ACCESS_INTO_NULL	Сизнинг дастурингиз инициализация қилинмаган (атомик нол) объектга қиймат беришга уринса.

COLLECTION_IS_NULL	Сизнинг дастурингиз EXISTS методидан ташқари бошқа коллекция методларини инициализация қилинмаган (атомик нол) ичма-ич жойлашган жадвал ёки массивга қўлланишга уринса, ёки дастур инициализация қилинмаган (атомик нол) ичма-ич жойлашган жадвал ёки массив элементлариг қиймат беришга уринса.
CURSOR_ALREADY_OPEN	Сизнинг дастурингиз очилган курсорни қайта очишга уринса. Курсорни қайта очишдан олдин уни ёпиш лозим. FOR loop курсори у мурожаат қиладиган курсорни автоматик тарзда очади. Шу сабабли, сизнинг дастурингиз циклнинг ичида ушбу курсорни очиб билмайди.
DUP_VAL_ON_INDEX	Сизнинг дастурингиз берилганлар базасидаги уникал чеклагичга эга бўлган устунга икки марта бир хил қиймат беришга уринса.
INVALID_CURSOR	Сизнинг дастурингиз курсорлар устида қоидага ҳилоф бўлган (масалан очилмаган курсорни ёпиш) амалларни бажаришга уринса.

INVALID_NUMBER	SQL жумлаларида белгили қаторни сонга алмаштириш уринсиз, чунки қатор ўринли сонни ифодаламайди. (Процедуравий жумлаларда, VALUE_ERROR мустасно содир бўлади.)
LOGIN_DENIED	Сизнинг дастурингиз мавжуд бўлмаган ном ёки нотўғри парол орқали Oracle-га уланишга уринса.
NO_DATA_FOUND	SELECT INTO жумласи қатор қайтармаса, ёки сизнинг дастурингиз ичма-ич жойлашган жадвалнинг ўчирилган элементиға ҳавола қилса.
NOT_LOGGED_ON	Сизнинг дастурингиз Oracle-га уланмасдан базаға мурожаат қилишға уринса.
PROGRAM_ERROR	PL/SQL ички муаммолари бўлса.
ROWTYPE_MISMATCH	Хост курсор ўзгарувчиси ва PL/SQL курсор ўзгарувчиси биргаликда бўлмаган ҳар хил турларға эға бўлса

SELF_IS_NULL	Сизнинг дастурингиз бўш нусха учун MEMBER методини чақиришга уринса. Яъни ички-қурилган SELF параметри (ҳамма вақт MEMBER методига узатиладиган биринчи параметр) null қийматга эга бўлади.
STORAGE_ERROR	PL/SQL ишлашида хотира етмаса.
SUBSCRIPT_BEYOND_COUNT	Сизнинг дастурингиз ичма-ич жойлашган жадвалга ёки коллекциядаги элементлар сонидан катта индекслар номерли массив элементига ҳавола қилса.
SUBSCRIPT_OUTSIDE_LIMIT	Сизнинг дастурингиз ичма-ич жойлашган жадвалга ёки ўрнатилган интервалдан ташқарида ётувчи индекс номерли (масалан -1) массив элементига ҳавола қилса.
SYS_INVALID_ROWID	Белгили қаторни универсал rowid алмаштиришга уринса.
TIMEOUT_ON_RESOURCE	Oracle ресурсга муҳтож бўлса.
TOO_MANY_ROWS	SELECT INTO жумласи биттадан кўп қаторни қайтарса.

VALUE_ERROR	Арифметик, алмаштириш, кесиш ёки ўлчов чеклагичи билан боғлиқ хатолик содир бўлса.
ZERO_DIVIDE	Сизнинг дастурингиз сонни нолга бўлишга уринса.

Фойдаланувчи томонидан аниқланган мустаснолар

PL/SQL фойдаланувчига ўзининг мустасносини эълон қилишга имкон беради. Олдиндан аниқланган мустаснодан фарқли равишда фойдаланувчи томонидан аниқланган мустаснолар эълон қилиниши лозим ва RAISE жумласи ёрдамида ошкор тарзда содир этилиши шарт.

Мустасноларни эълон қилиш

Мустасноларни PL/SQL блоки, қисмпрограммаси ёки пакетнинг эълон қилиш қисмида эълон қилиш мумкин холос. Мустаснони унинг номини киритиш ва EXCEPTION таянч хизматчи сўз билан давом этириш лозим. Масалан:

```
DECLARE
    past_due EXCEPTION;
```

Эълон қилиш қоидалари

Битта блокнинг ичида мустаснони икки марта эълон қилиш мумкин эмас. Аммо битта мустаснони иккита ҳар хил блокларда эълон қилиш мумкин ва уларга масалан қуйидагича мурожаат қилиш мумкин:

Блок_меткаси.мустасно_номи

Қуйидаги мисол битта мустаснони иккита ҳар хил блокларда эълон қилиш мумкинлигини намоён этилади:

```
DECLARE
```

```

past_due EXCEPTION;
acct_num NUMBER;
BEGIN
    DECLARE  -- қисм-блок бошланиши
        past_due EXCEPTION;  -- эълон қилиш prevails
        acct_num NUMBER;
    BEGIN
        ...
        IF ... THEN
            RAISE past_due; -- бунга ишлов берилмайди
        END IF;
    END;  ----- қисм-блок охири
EXCEPTION
    WHEN past_due THEN  -- содир этилган мустаснога ишлов
        берилмайди
        ...
END;

```

Охирги блокдаги мустаснога ишлов бериш қисми ишланмайди, чунки `past_due` мустаснонинг содир этиш ички блокда юз бериб, унда ишлов бериш қисми (ички блок ичида)мавжуд эмас. Бу ерда `RAISE` жумласи ва `WHEN` жумласи ҳар хил мустасноларга мурожаат қиляпти. Охирги блокдаги мустаснога ишлов бериш қисми ишлаши учун `past_due` мустаснонинг эълон қилиш қисмини ички блокдан олиб ташлаш лозим ёки `OTHERS` ишлов бериш қисмини киритиш керак.

EXCEPTION_INIT фойдаланиш

Номланмаган ички мустасноларга ишлов бериш учун `OTHERS` ишлов берувчидан ёки `EXCEPTION_INIT` прагмадан фойдаланишингиз мумкин. *Прагма* – бу компиляторга директива бўлиб хизмат қилади.

PL/SQL-да `EXCEPTION_INIT` прагмасы мустасно номини Oracle хатолиги номери билан боғлашга компиляторга кўрсатма беради. Бу эса ўз навбатида ихтиёрий ички мустаснога унинг номи орқали мурожаат қилишга ва унга ишлов берувчи ёзишга имкон беради. `EXCEPTION_INIT` прагмасы қуйидаги синтаксисга эга:

```

PRAGMA                                EXCEPTION_INIT (мустасно_номи ,
Oracle_хатолиги_номери) ;

```

Бу ерда мустасно_номи олдин эълон қилинган мустаснонинг номи. Прагма мустасно эълон қилингандан сўнг ихтиёрий жойда келиши мумкин. Масалан:

```
DECLARE
    deadlock_detected EXCEPTION;
    PRAGMA EXCEPTION_INIT(deadlock_detected, -60);
BEGIN
    ...
EXCEPTION
    WHEN deadlock_detected THEN
        -- хатолик ишлов берилади
END;
```

Raise_application_error фойдаланиш

DBMS_STANDARD пакети приложения ва Oracle орасидаги муносабатни осонлаштириш учун хизмат қилади. Масалан ундаги raise_application_error процедураси фойдаланувчи томонидан аниқланган хатолик хабарини сақланадиган қисмдастурдан чиқариб беришга имкон беради. Шундай билан сиз хатоликлар ҳақида мижоз дастурига хабар бериш имконига эгасиз. raise_application_error процедурасини чақириш учун қуйидаги синтаксисдан фойдаланинг:

raise_application_error(хатолик_номери, хабар[, {TRUE | FALSE}]);

бу ерда хатолик_номери -20000 .. -20999 оралиқда ётувчи манфий бутун сон ва хабар белгили қатор бўлиб узунлиги 2048 байтдан ошмайди. Агар учинчи параметрнинг қиймати TRUE бўлса, хатолик олдинги хатоликларнинг стёкида жойлашади. Агар учинчи параметрнинг қиймати FALSE бўлса, хатолик олдинги барча хатоликларини алмаштиради. DBMS_STANDARD пакети STANDARD пакетининг кенгайтмасидир.

Қуйидаги мисолда, агар ишчи маоши қолдириб юборилган бўлса, raise_application_error чақирилади:

```
CREATE PROCEDURE raise_salary (emp_id NUMBER, amount
NUMBER) AS
    curr_sal NUMBER;
BEGIN
    SELECT sal INTO curr_sal FROM emp WHERE empno =
emp_id;
```

```

IF curr_sal IS NULL THEN
    /*Фойдаланувчи аниқланган хатолик
      чиқарилади. */
    raise_application_error(-20101, 'Маош қолдириб
юборилган');
ELSE
    UPDATE emp SET sal = curr_sal + amount WHERE empno
= emp_id;
END IF;
END raise_salary;

```

Олдиндан аниқланган мустасноларни қайта эълон қилиш

Шуни таъкидлаб ўтиш жоизки PL/SQL олдиндан аниқланган мустасноларни STANDARD, пакетида глобал тарзда эълон қилади. Олдиндан эълон қилинган мустасноларни қайта эълон қилсангиз, хатолик юз бериши мумкин, чунки локал эълон қилинган мустасно глобал эълон қилинган мустасно устига ёзилади. Масалан, агар сиз *invalid_number* номли мустасно эълон қилсангиз ва PL/SQL олдиндан аниқланган INVALID_NUMBER мустаснони ички содир этса, INVALID_NUMBER учун ёзилган ишлов берувчи ички мустаснони ушлаб билмайди. Бу нохуш ҳолатдан нукта белгисидан фойдаланиш орқали чиқиш мумкин:

```

EXCEPTION
    WHEN invalid_number OR STANDARD.INVALID_NUMBER THEN
        -- хатолик ишлов берилади.
END;

```

Мустасно қандай содир этилади

Ички мустаснолар ва EXCEPTION_INIT орқали Oracle хатолиги номери билан боғланган, фойдаланувчи томонидан аниқланган мустаснолар ошқормас тарзда тизим ишлаши жараёнида содир этилади. Аммо фойдаланувчи томонидан аниқланган бошқа мустаснолар RAISE жумласи орқали ошқор тарзда содир этилиши лозим.

RAISE жумласидан фойдаланиш

Берилган мустасно учун RAISE жумласини мустасно аниқланган блокнинг ихтиёрий жойида ишлатишингиз мумкин:

```

DECLARE
    out_of_stock    EXCEPTION;
    number_on_hand  NUMBER(4);
BEGIN
    ...
    IF number_on_hand < 1 THEN
        RAISE out_of_stock;
    END IF;
EXCEPTION
    WHEN out_of_stock THEN
        -- хатолик ишлов берилади
END;
```

Олдиндан аниқланган мустаснони ошкор тарзда ҳам содир этишингиз мумкин:

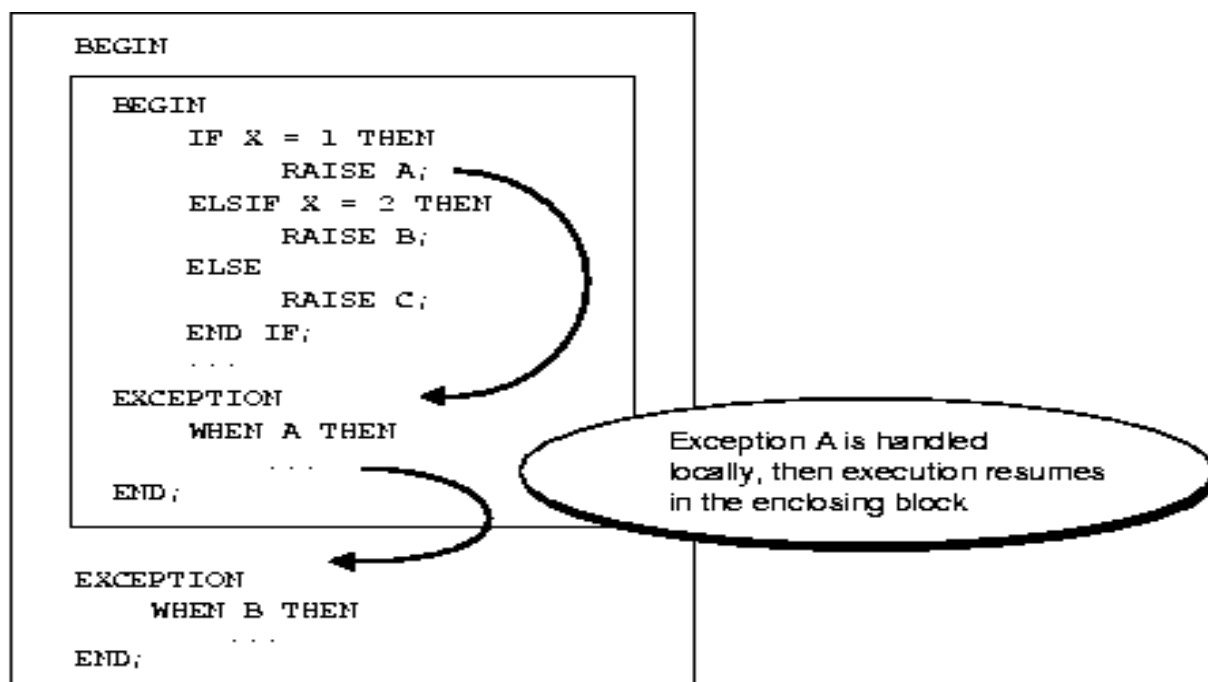
```

DECLARE
    acct_type INTEGER;
BEGIN
    ...
    IF acct_type NOT IN (1, 2, 3) THEN
        RAISE INVALID_NUMBER; /* Олдиндан аниқланган
мустасно ошкор тарзда содир этиляпти*/
    END IF;
EXCEPTION
    WHEN INVALID_NUMBER THEN
        ROLLBACK;
    ...
END;
```

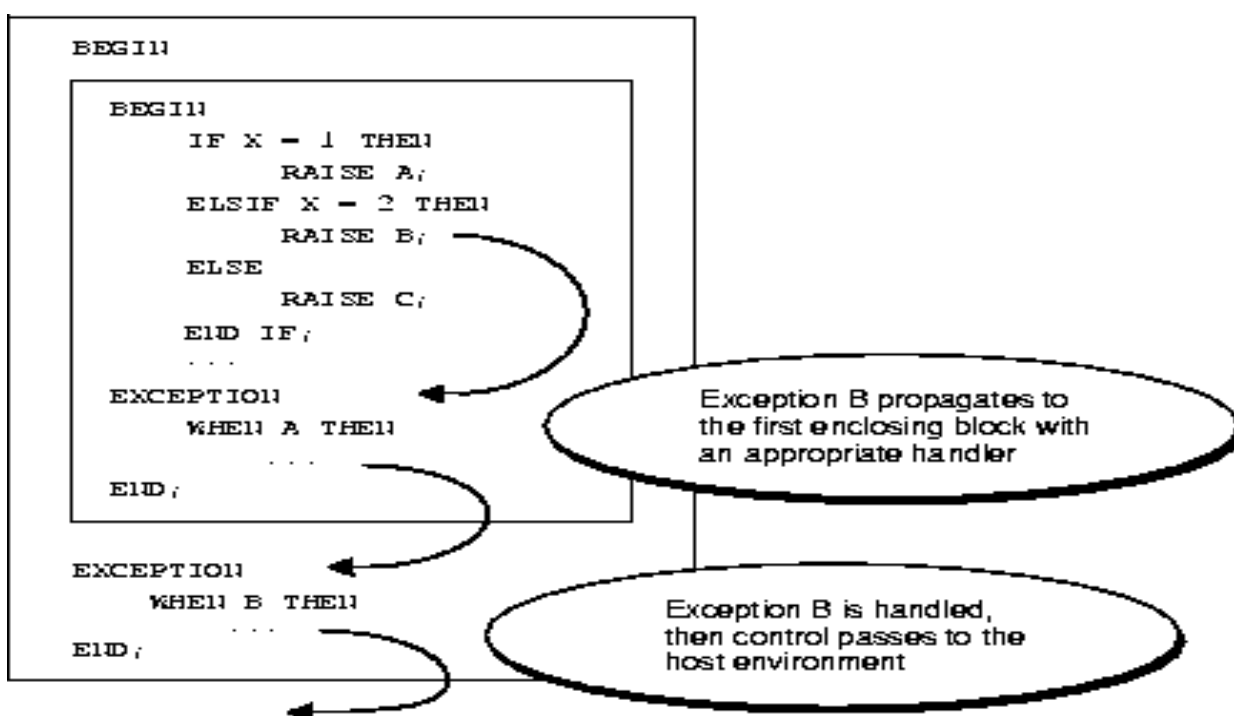
Мустаснолар қандай тарқалади

Мустасно содир этилиши билан, агар PL/SQL унга ишлов берувчини жорий блок ёки қисмдастурда тополмаса, мустасно *тарқалади*. Бу эса шуни англатадики, мустасно ташқи блокдан то охирги блокга қадар ишлов берувчини излайди. Агар изланиш муваффақиятсиз тугаса, унда PL/SQL *ишлов берилмаган мустасно* хатолигини хост муҳитга қайтаради.

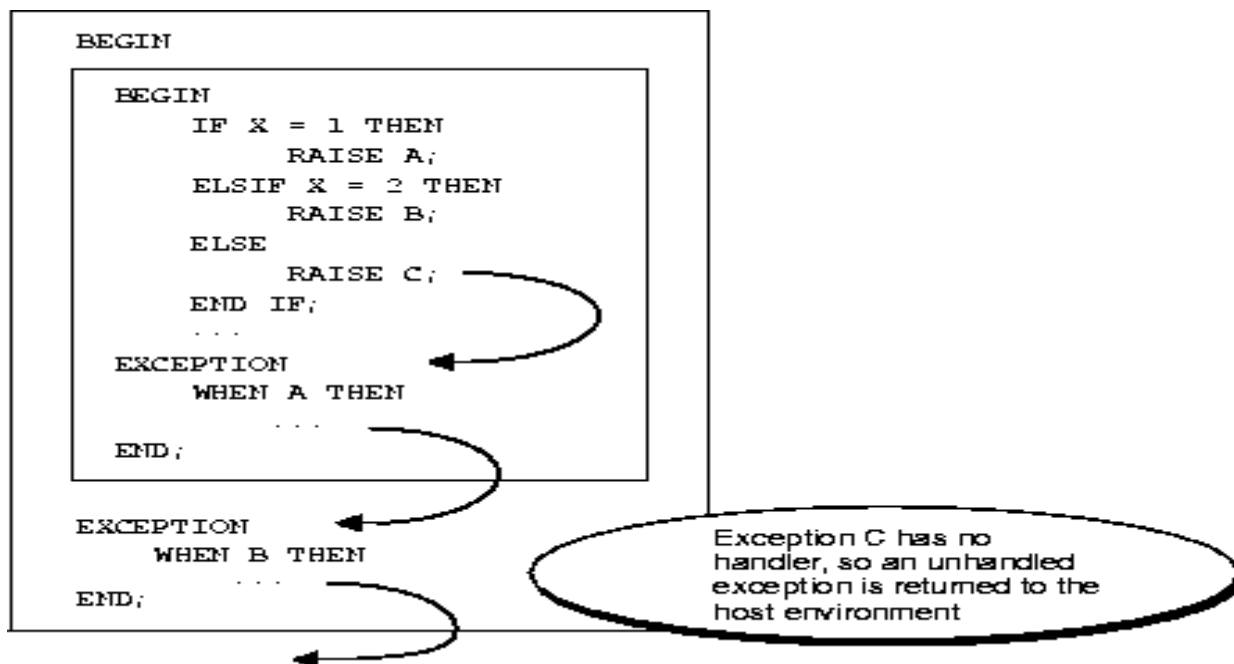
Расм. Тарқалиш қоидалари: 1 мисол



Расм. Тарқалиш қоидалари: 2 мисол



Расм. Тарқалиш қоидалари: 3 мисол



Мустасно фақат ўзи эълон қилинган блок миқёсида тарқалиши мумкин холос.

Қуйидаги мисолни кўрамиз:

```

BEGIN
  ...
  DECLARE ----- қисм-блок боши
    past_due EXCEPTION;
  BEGIN
    ...
    IF ... THEN
      RAISE past_due;
    END IF;
  END; ----- қисм-блок охири
EXCEPTION
  ...
  WHEN OTHERS THEN
    ROLLBACK;
END;

```

Past_due мустасно эълон қилинган блокда уни ишлов берувчиси мавжуд эмас ва шу сабабли у навбатдаги блокга тарқалади. Лекин, мустаснони таъсир қилиш қоидаларига асосан, охирги блок қисм-блокда эълон қилинган мустаснога мурожаат қилиб билмайди. Шу сабабли фақат OTHERS ишлов берувчиси ушбу мустаснони ушлаши мумкин.

Мустаснони қайта содир этиш

Баъзан мустаснони қайтадан содир этиш зарурияти туғилади. Sometimes, you want to *reraise* an exception, that is, handle it locally, then pass it to an enclosing block. For example, you might want to roll back a transaction in the current block, then log the error in an enclosing block.

Мустаснони қайта содир этиш учун ишлов берувчи қисми ичида RAISE жумласини ёзиш кифоя бўлади:

```
DECLARE
    out_of_balance EXCEPTION;
BEGIN
    ...
    BEGIN ----- қисм-блок боши
        ...
        IF ... THEN
            RAISE out_of_balance; -- мустасно содир
            этилади
        END IF;
    EXCEPTION
        WHEN out_of_balance THEN
            -- хатоликга ишлов берилади
            RAISE; -- жорий мустасно қайта содир этилади
        END; ----- қисм-блок охири
    EXCEPTION
        WHEN out_of_balance THEN
            -- хатоликга фарқли ишлов берилади
        ...
END;
```

Содир этилган мустасноларга ишлов бериш

Мустасно содир этилса, одатда қуйидагича уни ишлов бериш мақсадга мувофиқ бўлади:

```
EXCEPTION
    WHEN мустасно_номи1 THEN --ишлов берувчи
        жумлалар_кетма_кетлиги
    WHEN мустасно_номи2 THEN --бошқа ишлов берувчи
        жумлалар_кетма_кетлиги2
    ...
    WHEN OTHERS THEN --ишлов берувчи
```

```

        жумлалар_кетма_кетлиги3
END;

```

Мустаснони ушлаб олиш учун, мустаснога ишлов берувчиси ёзилади. Хар бир ишлов берувчи мустаснони аниқловчи WHEN жумласи ва бажарилиши лозим бўлган жумлалар кетма-кетлигидан таркиб топган. Ушбу жумлалар кетма-кетлиги блок ёки қисмдастурни бажарилишини тугатади ва бошқарув мустасно содир этилган жойга қайтарилмайди.

The optional OTHERS ишлов берувчиси барча махсус номланмаган мустасноларга ишлов бериш учун ишатилади ва у сўнги ишлов берувчиси бўлиши лозим:

```

EXCEPTION
    WHEN ... THEN
        -- хатоликга ишлов берилади
    WHEN ... THEN
        -- хатоликга ишлов берилади
    WHEN OTHERS THEN
        -- барча бошқа хатоликларга ишлов берилади
END;

```

Иккита ва ундан кўп мустасноларга бирданига бир хил ишлов бериш мумкин:

```

EXCEPTION
    WHEN over_limit OR under_limit OR VALUE_ERROR THEN
        -- хатоликга ишлов берилади

```

Эълон қилиш қисмида мустасноларнинг содир этилиши

Эълон қилиш қисмида мустаснолар фақат қиймат бериш жараёнида содир этилиши мумкин. Масалан қуйидаги эълон қилиш қисмида мустасно содир этилади, чунки credit_limit ўзгармас 999 катта бўлган сонни сақлаб билмайди:

```

DECLARE
    credit_limit CONSTANT NUMBER(3) := 5000; -- мустасно
    содир этилади
BEGIN
    ...
EXCEPTION
    WHEN OTHERS THEN - мустаснони ушлаб билмайди
    ...
END;

```

Жорий блокдаги ишлов берувчилар содир этилган мустаснони ушлаб билмайди, чунки эълон қилиш қисмида содир этилган мустаснолар *ўша заҳотиёқ* ташқи блокга тарқалади.

Ишлов берувчида содир этилган мустаснолар

Бир вақтда фақат битта мустасно актив бўлиши мумкин холос, шу сабабли ишлов берувчининг ичида содир этилган мустасно *ўша заҳотиёқ* ташқи блокга тарқалади ва янги содир этилган мустасно учун ишлов берувчини излайди:

```
EXCEPTION
  WHEN INVALID_NUMBER THEN
    INSERT INTO ... -- DUP_VAL_ON_INDEX мустасно
    содир бўлиши мумкин
  WHEN DUP_VAL_ON_INDEX THEN ... - мустаснони ушлаб
  билмайди
END;
```

Мустаснога ишлов берувчидан чиқиш ва унга кириш

GOTO жумласи ёрдамида мустаснога ишлов берувчи қисмига кириш мумкин эмас. *statement cannot branch into an exception handler*. GOTO жумласи ёрдамида ишлов берувчи қисмидан жорий блокга ўтиш ҳам мумкин эмас:

```
DECLARE
  pe_ratio NUMBER(3,1);
BEGIN
  DELETE FROM stats WHERE symbol = 'XYZ';
  SELECT price / NVL(earnings, 0) INTO pe_ratio FROM
stocks
  WHERE symbol = 'XYZ';
  <<my_label>>
  INSERT INTO stats (symbol, ratio) VALUES ('XYZ',
pe_ratio);
EXCEPTION
  WHEN ZERO_DIVIDE THEN
    pe_ratio := 0;
    GOTO my_label; -- жорий блокга нотўғри ўтиш END;
```

SQLCODE ва SQLERRM фойдаланиш

Мустаснога ишлов бериш қисмида SQLCODE ва SQLERRM функцияларидан қандай хатолик юз берганини ва у билан боғланган хатолик

матнини топиш учун фойдаланиш мумкин. Ички хатоликлар учун `SQLCODE` функцияси Oracle хатолиги номерини қайтаради. Агар Oracle хатолиги *no data found* бўлса `SQLCODE` функцияси +100 ва бошқа барча ҳолларда манфий сонни қайтаради. `SQLERRM` функцияси эса мос хатолик хабарини қайтаради. **The message begins with the Oracle error code.**

Фойдаланувчи томонидан аниқланган мустаснолар учун `SQLCODE` функцияси +1 ва `SQLERRM` функцияси қуйидаги хабарни қайтаради:

User-Defined Exception

Агар мустасно номи `EXCEPTION_INIT` прагма орқали Oracle хатолиги номери билан боғланган бўлса, `SQLCODE` функцияси хатолик номерини қайтаради ва `SQLERRM` функцияси эса мос хатолик хабарини қайтаради. The maximum length of an Oracle хатолиги хабарининг максимал узунлиги error message is 512 белгидан иборат бўлиши мумкин. Агар мустасно содир бўлмаса, `SQLCODE` функцияси нол қиймат қайтаради ва `SQLERRM` функцияси

ORA-0000: normal, successful completion

хабарни қайтаради. Oracle барча хатоликлар хабарларини қуйидаги дастур ёрдамида кўриш мумкин

```
DECLARE
    ...
    err_msg VARCHAR2(100);
BEGIN
    /* Get all Oracle error messages. */
    FOR err_num IN 1..9999 LOOP
        err_msg := SQLERRM(-err_num);
        INSERT INTO errors VALUES (err_msg);
    END LOOP;
END;
```

`SQLCODE` ва `SQLERRM` функцияларини бевосита SQL жумлаларида ишлатиш мумкин эмас, балки уларни аввал локал ўзгарувчида олиб, сўнг ишлатиш мумкин:

```
DECLARE
    err_num NUMBER;
```

```

    err_msg VARCHAR2(100);
BEGIN
    ...
EXCEPTION
    ...
    WHEN OTHERS THEN
        err_num := SQLCODE;
        err_msg := SUBSTR(SQLERRM, 1, 100);
        INSERT INTO errors VALUES (err_num, err_msg);
END;
```

Фойдали усуллар

Ушбу бўлимда қулайлик туғдирадиган учта усул билан танишасиз:

Мустасно содир этилгандан сўнг ишни давом этириш

Сизга маълумки мустасно содир этилгандан сўнг бошқарув мустаснога ишлов бериш қисмига узатилади ва ундан кейин блок иши тугатилади, яъни жорий блокга қайтишнинг иложи йўқ. Қуйидаги мисолда агарда SELECT INTO жумласида ZERO_DIVIDE содир бўлса, сиз INSERT жумласини ишлатолмайсиз:

```

DECLARE
    pe_ratio NUMBER(3,1);
BEGIN
    DELETE FROM stats WHERE symbol = 'XYZ';
    SELECT price / NVL(earnings, 0) INTO pe_ratio FROM
stocks
    WHERE symbol = 'XYZ';
    INSERT INTO stats (symbol, ratio) VALUES ('XYZ',
pe_ratio);
EXCEPTION
    WHEN ZERO_DIVIDE THEN
        ...
END;
```

Ушбу мисолдаги муаммони қуйидагича ҳал қилиш мумкин::

```

DECLARE
    pe_ratio NUMBER(3,1);
BEGIN
    DELETE FROM stats WHERE symbol = 'XYZ';
    BEGIN ----- қисм-блок боши
```

```

SELECT price / NVL(earnings, 0) INTO pe_ratio FROM
stocks
    WHERE symbol = 'XYZ';
EXCEPTION
    WHEN ZERO_DIVIDE THEN
        pe_ratio := 0;
END; ----- қисм-блок охири
INSERT INTO stats (symbol, ratio) VALUES ('XYZ',
pe_ratio);
EXCEPTION
    WHEN OTHERS THEN
        ...
END;
```

Ушбу мисолда агар SELECT INTO жумласи ZERO_DIVIDE мустаснони содир этса, локал ишлов берувчи уни ушлаб олади ва pe_ratio нол қиймат беради. Ишлов берувчи ўз ишини тугатади ва иш INSERT жумласи билан давом этирилади.

Транзакцияни қайта бажариш

Агар мабода бирор сабабга кўра транзакциянгиз бажарилмай қолса, сиз уни қуйидаги мисолда кўрсатилган усулда қайта бажаришга уриниб кўришингиз мумкин:

```

DECLARE
    name    VARCHAR2(20);
    ans1    VARCHAR2(3);
    ans2    VARCHAR2(3);
    ans3    VARCHAR2(3);
    suffix  NUMBER := 1;
BEGIN
    ...
    LOOP /* буни ўрнига FOR i IN 1..10 LOOP бўлиши
мумкин, унда ўнта уринишга имконият берилади */
        BEGIN --қисм-блок боши
            SAVEPOINT start_transaction; /* қайтиш
нуктасини белгилаш */
            DELETE FROM results WHERE answer1 = 'NO';
            INSERT INTO results VALUES (name, ans1,
                                        ans2, ans3);

            /* DUP_VAL_ON_INDEX содир этиши
мумкин, агар номлар бир хил бўлса */
            COMMIT;
            EXIT;
        EXCEPTION
```

```

        WHEN DUP_VAL_ON_INDEX THEN
            ROLLBACK TO start_transaction;
            /* асл ҳолатга қайтиш */
            suffix := suffix + 1;
            /* муаммони ҳал этишга уриниш */
            name := name || TO_CHAR(suffix);
        END; -- қисм-блок охири
    END LOOP;
END;

```

Локатор ўзгарувчиларидан фойдаланиш

Қуйидаги мисолни қараймиз:

```

BEGIN
    SELECT ...
    SELECT ...
    SELECT ...
    ...
EXCEPTION
    WHEN NO_DATA_FOUND THEN ...
        -- Қайси SELECT жумласида хатолик юз берди?
END;

```

Ушбу мисода содир бўлган хатолик қайси SELECT жумласида юз берганини аниқлаш учун қуйидаги мисолда кўрсатилганидек локатор ўзгарувчиларидан фойдаланишимиз мумкин:

```

DECLARE
    stmt INTEGER := 1; -- 1SELECT жумласи у-н
BEGIN
    SELECT ...
    stmt := 2; -- 2SELECT жумласи у-н
    SELECT ...
    stmt := 3; -- 3SELECT жумласи у-н
    SELECT ...
    ...
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        INSERT INTO errors VALUES ('Error in statement '
        || stmt);
END;

```

11.PL/SQLда объект турлари ва улар билан ишлаш

Объект турлари

Объект тури нима?

Нима учун объект тури ишлатилади?

Объект турининг тузилиши

Объект турининг компоненталари

Объект турининг аниқланиши

Объектларни эълон қилиш ва қиймат бериш

Атрибутларга мурожаат қилиш

Конструкторларни чақириш

Методларни чақириш

Объектлардан умумий фойдаланиш

Объектларни бошқариш

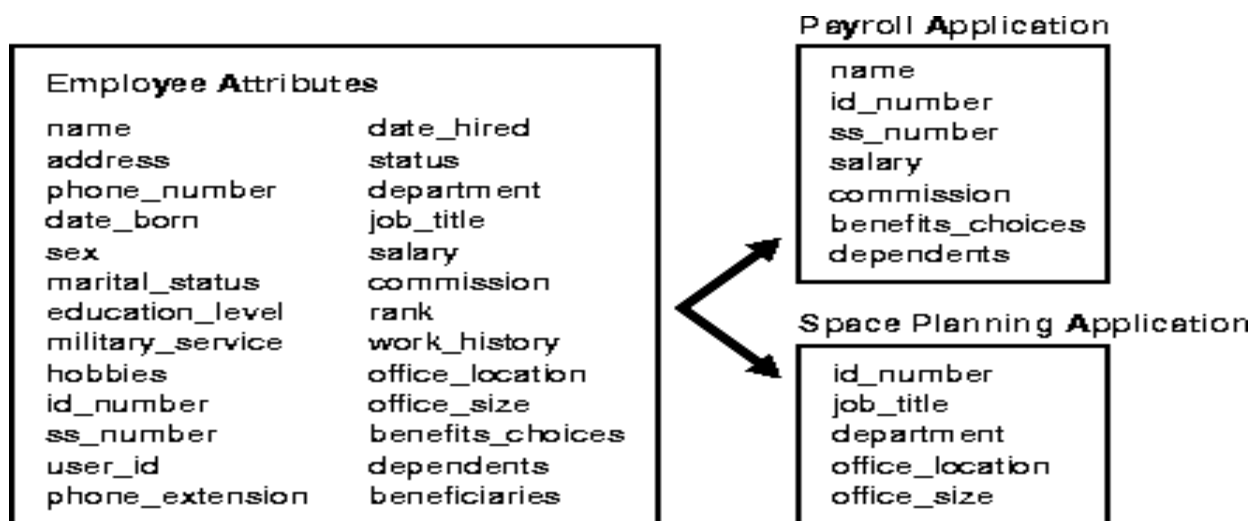
Объект тури нима?

Объект тури фойдаланувчи томонидан аниқланган мураккаб берилганлар тури бўлиб, берилганлар тури ва берилганларни бошқариш учун зарур бўладиган функция ва процедуралардан таркиб топган. Берилганлар тузилишини шаклантирувчи ўзгарувчилар атрибутлар деб номланади. Объект тури ҳаракат тартибини характерловчи функция ва процедуралар эса методлар деб номланади.

Биз одатда объект (шахс, машина ёки банк ҳисоб рақами каби) деганда унинг атрибутлари ва ҳаракат тартибини тушунамиз. Масалан бола объекти – ёши ва вазни атрибутларига эга, ҳаракат тартиби эса ейиш, ичиш ва ухлашдан иборат деб тасаввур қилиш мумкин.

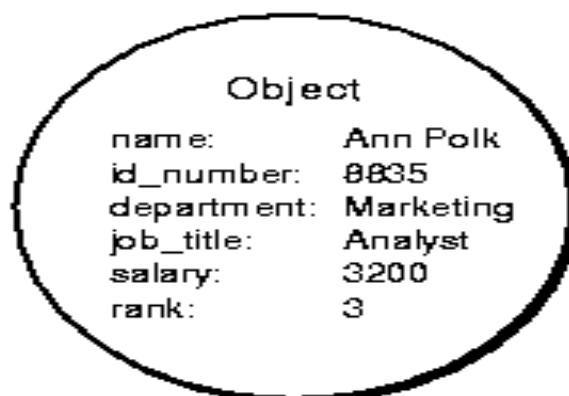
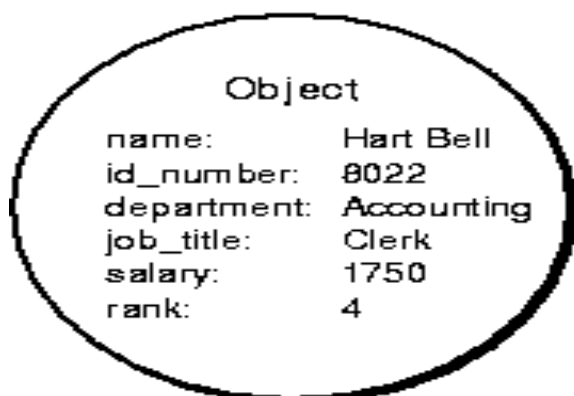
When you define an object type using the CREATE TYPE жумласи ёрдамида объект турини аниқлашда сиз дунёдаги бирор ҳақиқий объектнинг абстракт қолипini яратасиз. Қолип эса амалий дастурда зарур бўладиган объект атрибутлари ва ҳаракат тартибини

аниқлайди. Масалан хизматчи жуда кўп атрибутларга эга, аммо маълум варақаларни тўлдириш учун айрим атрибутларни тўлдириш талаб қилинади ХОЛОС:



Берилганлар тузилмасини қийматлар билан тўлдиришда, сиз абстракт хизматчи *нусхасини* яратасиз. Сиз хоҳлаганча нусхаларни (одатда *объектлар* деб номланади) яратиш имкониятига эгасиз. Хар бир объект ўзининг номига, тартиб рақамига, иш номи ва шу каби бошқа атрибутларга эга. Ушбу берилганлар уларга боғлиқ бўлган методлар ёрдамида киритилади ёки ўзгартирилади. Шундай қилиб:

Object Type <i>Employee</i>	
Attributes name id_number department job_title salary rank	Methods calculate_bonus change_dept change_job_title change_salary change_rank



Нима учун объект тури ишлатилади?

Объект тури катта тизимни мантиқий борлиқ бирликларига бўлиш ёрдамида мураккабликни соддалаштиришга имкон беради. Бу эса ўз навбатида дастурлаш жараёнида модуллик принципини қўллашда имкон яратади. Бундан ташқари дастур қисмларини ҳар хил командалар ишлаб чиқишига имкон беради.

Объект тури берилганларга кириш жараёнини фақат методлар ёрдамида ташкил қилишга ва берилганларни ортиқча таъсирлардан ҳимоя қилади. Объект тури мижоз дастурини ўзгартирмасдан туриб метод кодини ўзгартиришга имконият яратади.

Объект тури берилганларни ҳақиқий моделлаштиришга йўл очиб беради. Бундан ташқари объект турлари бевосита Java ва C++ каби объектга – мўлжалланган тилларнинг синфларига акслантирилади.

Объект турининг тузилиши

Пакетларга ўхшаб объект тури икки қисмдан иборат: аниқланиши ва танаси:

Объект турининг тузилмаси.

аниқланиши
Атрибутларни эълон қилиш
Методларнинг аниқланиши

танаси
Методларнинг таналари

Объект турининг *аниқланиши* сизнинг дастурингиз учун интерфейс бўлиб хизмат қилади; у берилганлар тузилмаси (атрибутлар тўплами) билан бирга уларни бошқарувчи амаллар (методларни) эълон қилади. *Танаси* эса методларни тўла аниқлайди.

Мижоз дастурида зарур бўладиган барча методлар объект турининг аниқланишида эълон қилинади. Аниқланишни операцион интерфейс ва танани қора ящик деб тасаввур қилишингиз мумкин. Сиз мижоз дастурига таъсир қилмасдан туриб танани ўзгартиришингиз мумкин.

Объект тури аниқланишида барча атрибутлар методлардан олдин эълон қилиниши лозим. Шундай қилиб, агар объект тури аниқланишида фақат атрибутлар эълон қилинса, унда объект турининг танаси лозим бўлмайди. Танада атрибутларни эълон қилиш мумкин эмас. Объект тури аниқланишидаги барча эълонлар объект турининг ташқарисида кўринади.

Тузилмани яхшироқ тушуниб олиш учун қуйидаги мисолга мурожаат қиламиз. Унда комплекс сонлар объект тури аниқланган. Биз учун ҳозирча комплекс сонлар икки қисмдан иборат ва улар учун бир нечта арифметик амалларнинг аниқланиши етарли бўлади.

```
CREATE TYPE Complex AS OBJECT (
  rpart REAL,  -- атрибут
```

```

    ipart REAL,
    MEMBER FUNCTION plus (x Complex) RETURN Complex, --
метод
    MEMBER FUNCTION less (x Complex) RETURN Complex,
    MEMBER FUNCTION times (x Complex) RETURN Complex,
    MEMBER FUNCTION divby (x Complex) RETURN Complex
);

CREATE TYPE BODY Complex AS
    MEMBER FUNCTION plus (x Complex) RETURN Complex IS
    BEGIN
        RETURN Complex(rpart + x.rpart, ipart + x.ipart);
    END plus;

    MEMBER FUNCTION less (x Complex) RETURN Complex IS
    BEGIN
        RETURN Complex(rpart - x.rpart, ipart - x.ipart);
    END less;

    MEMBER FUNCTION times (x Complex) RETURN Complex IS
    BEGIN
        RETURN Complex(rpart * x.rpart - ipart * x.ipart,
                        rpart * x.ipart + ipart * x.rpart);
    END times;

    MEMBER FUNCTION divby (x Complex) RETURN Complex IS
        z REAL := x.rpart**2 + x.ipart**2;
    BEGIN
        RETURN Complex((rpart * x.rpart + ipart * x.ipart)
/ z,
                        (ipart * x.rpart - rpart * x.ipart)
/ z);
    END divby;
END;

```

Объект турининг компоненталари

Объект тури берилганлар турлари ва амалларни аниқлайди. Объект тури аниқланишида атрибутлар ва методларни эълон қилишингиз мумкин, аммо ўзгармасларни, мустасноларни, курсорларни ёки турларни эълон қилиш мумкин эмас. Камида битта атрибут (максимум 1000) эълон қилиниши лозим. Мавжуд бўлган объект турига атрибутлар қўшиш мумкин эмас.

Атрибутлар

Атрибутлар худди ўзгарувчилар каби номи ва берилганлар тури орқали эълон қилинади. Номи объект тури ичида уникал бўлиши лозим. Берилганлар тури эса Oracle тизимидаги ихтиёрий тур бўлиши мумкин, қуйидаги турлар бундан мустасно:

- LONG ва LONG RAW
- NCHAR, NCLOB, ва NVARCHAR2
- ROWID ва UROWID
- PL/SQL-даги махсус турлар: BINARY_INTEGER (ва уларнинг қисмтурлари), BOOLEAN, PLS_INTEGER, RECORD, REF CURSOR, %TYPE, ва %ROWTYPE
- PL/SQL пакети ичида аниқланган турлар

Атрибутни эълон қилишда унга қиймат бериш мумкин эмас. Атрибутга NOT NULL чеклагични қўйиш мумкин эмас. Аммо объектларни берилганлар базасининг жадвалида сақлаш мумкин ва унда NOT NULL чеклагичини қўйиш мумкин.

Объект турининг атрибути ўз навбатида бошқа объект тури бўлиши мумкин.

Методлар

Умуман айтганда, метод бу объект тури аниқланишида эълон қилинган MEMBER ва STATIC таянч сўзлари ишлатилган қисмдастурдир. Методнинг номи объект тури номи ёки унинг ихтиёрий атрибути номидан фарқ қилиши лозим. MEMBER методлари нусхалардан чақирилади:

instance_expression.method()

Аммо STATIC методлар объект турларидан чақирилади:

object_type_name.method()

Пакетда бирлаштирилган қисмдастурларидек кўп методлар икки қисмдан таркиб топган: аниқланиши ва танаси. *Аниқланиши* – метод номи, параметрлар рўйхати ва функциялар учун қайтариш туридан иборат. Танаси эса махсус вазифани бажарувчи коддан иборат.

Объект тури аниқланишидаги ҳар бир методга объект тури танасида мос метод танаси мавжуд бўлиши лозим.

Объект турида методлар атрибутларга ва бошқа методларга бевосита (объект турини кўрсатмасдан) мурожаат қилишлари мумкин:

```
CREATE TYPE Stack AS OBJECT (
    top INTEGER,
    MEMBER FUNCTION full RETURN BOOLEAN,
    MEMBER PROCEDURE push (n IN INTEGER),
    ...
);

CREATE TYPE BODY Stack AS
    ...
    MEMBER PROCEDURE push (n IN INTEGER) IS
    BEGIN
        IF NOT full THEN
            top := top + 1;
            ...
        END push;
    END;
```

SELF параметри

MEMBER методлари ички қурилган номланган SELF параметрига эга бўлиб, у объект тури нусхасини билдиради. STATIC методлар эса ушбу хоссага эга эмаслар.

Метод танасида SELF параметри методи чакирилган объектни белгилайди. Масалан қуйидаги мисолда transform методи SELF параметрини IN OUT параметри тарзида эълон қилади:

```
CREATE TYPE Complex AS OBJECT (
    MEMBER FUNCTION transform (SELF IN OUT Complex) ...
```

Сиз SELF параметри учун бошқа берилганлар турини аниқлаб билмайсиз. Агарда MEMBER функцияларида SELF параметри эълон қилинмаган бўлса, ушбу параметр IN тарзда қабул қилинади. Аммо агар MEMBER процедураларида SELF параметри эълон қилинмаган бўлса, ушбу параметр IN OUT тарзда қабул қилинади. Сиз OUT туридаги SELF параметрини аниқлаб билмайсиз.

Методлар SELF атрибутларига бевосита мурожаат қилишлари мумкин:

```
CREATE FUNCTION gcd (x INTEGER, y INTEGER) RETURN
INTEGER AS
-- х ва у энг катта бўлинувчисини топади
  ans INTEGER;
BEGIN
  IF (y <= x) AND (x MOD y = 0) THEN ans := y;
  ELSIF x < y THEN ans := gcd(y, x);
  ELSE ans := gcd(y, x MOD y);
  END IF;
  RETURN ans;
END;

CREATE TYPE Rational AS OBJECT (
  num INTEGER,
  den INTEGER,
  MEMBER PROCEDURE normalize,
  ...
);

CREATE TYPE BODY Rational AS
  MEMBER PROCEDURE normalize IS
    g INTEGER;
  BEGIN
    g := gcd(SELF.num, SELF.den);
    g := gcd(num, den);  -- олдинги жумлага эквивалент
    num := num / g;
    den := den / g;
  END normalize;
  ...
END;
```

Агар SQL жумласидан бўш нусха MEMBER методини чақирсангиз (яъни SELF бўш) , метод чақирилмайди ва бўш қиймат қайтарилади. Агар процедуравий жумладан бўш нусха MEMBER методини чақирсангиз (яъни

SELF бўш) , PL/SQL олдиндан аниқланган SELF_IS_NULL номли мустаснони чақиради.

Номлаш қондаси

Пакетлаштирилган процедуралар каби ҳар хил методларни (функция ёки процедурани) бир ном билан номлаш мумкин, агар ушбу методларнинг формал параметрлари сони, жойлашиш тартиби ва берилганлар тури оиласи фарқ қилса. Аммо иккита ‘member’ функциялар фақат қайтариш қийматининг тури билан фарқ қилса, бу функцияларни бир хил номлаш мумкин эмас.

Акслантириш ва тартиб методлари

CHAR ёки REAL каби скаляр берилганлар турларининг қийматлари олдиндан аниқланган тартибга эга ва шу сабабали уларни солиштириш имкони мавжуд. Аммо лекин объект тури нусхалари олдиндан аниқланган тартибга эга эмас. Уларни тартиблаштириш учун PL/SQL сиз аниқланган *акслантириш методни* чақиради. Қуйидаги мисолда MAP таянч сўзи кўрсатадики convert методи Rational объектларни уларни REAL қийматларга акслантириш орқали тартиблаштиради:

```
CREATE TYPE Rational AS OBJECT (
    num INTEGER,
    den INTEGER,
    MAP MEMBER FUNCTION convert RETURN REAL,
    ...
);
CREATE TYPE BODY Rational AS
    MAP MEMBER FUNCTION convert RETURN REAL IS
    BEGIN
        RETURN num / den;
    END convert;
    ...
END;
```

Объект тури фақат битта акслантириш методга эга бўлиши мумкин. Бу метод параметрсиз функция бўлиб қуйидаги турлардан бирини қайтаради: DATE, NUMBER, VARCHAR2, ёки CHARACTER ё REAL каби ANSI SQL турлари.

Атернатив тарзда *тартиб методлари* ишлатилади. Объект тури фақат битта тартиб методига эга бўлиши мумкин ва у сонли натижа қайтарувчи функция бўлиши лозим:

```
CREATE TYPE Customer AS OBJECT (
    id    NUMBER,
    name  VARCHAR2(20),
    addr  VARCHAR2(30),
    ORDER MEMBER FUNCTION match (c Customer) RETURN
INTEGER
);

CREATE TYPE BODY Customer AS
    ORDER MEMBER FUNCTION match (c Customer) RETURN
INTEGER IS
    BEGIN
        IF id < c.id THEN
            RETURN -1;  -- any negative number will do
        ELSIF id > c.id THEN
            RETURN 1;   -- any positive number will do
        ELSE
            RETURN 0;
        END IF;
    END;
END;
```

Хар бир тартиб методи фақат иккита параметрга эга: SELF ва ушбу турдаги бошқа объект. Агар c1 ва c2 объектлар Customer объеклари бўлса c1 > c2 каби солиштириш автомат тарзда match методини чақиради. Метод мос равишда манфий сонни, нолни ёки мусбат сонни қайтаради. Бу эса SELF параметри мос равишда бошқа параметрдан кичик , тенг ёки катта эканлигини билдиради. NULL қиймат тартиб методига узатилса, у ҳам NULL қиймат қайтаради.

Эслатма

Акслантериш методи объект қийматини скаляр қийматга акслантиради, сўнг скаляр қийматларни солиштиради. Тартиб методи битта объектни бошқа объект билан солиштиради.

Сиз акслантириш методни ёки тартиб методни эълон қилишингиз мумкин, аммо иккаласини бирданига эълон қилиш мумкин эмас. Агар сиз иккала методни ҳам эълон қилмасангиз, унда объектларни фақат тенгликга ёки тенг бўлмасликга солиштиришингиз мумкин холос. (Бир хил турдаги иккита объект тенг бўлади фақат ва фақат уларнинг мос атрибутларининг қийматлари тенг бўлса)

Конструктор методлари

Хар бир объект тури тизим томонидан аниқланган функция - *конструктор методига* (қисқача *конструктор*) эга, унинг номи объект турининг номи билан устма – уст тушади. Сиз конструкторни объект тури нусхасини тўлдириш ва қайтариш учун ишлатасиз.

Объект турларини аниқлаш

Объект тури қуйидагича аниқланади:

```
CREATE [OR REPLACE] TYPE тур_номи
  [AUTHID {CURRENT_USER | DEFINER}] {IS | AS} OBJECT (
    атрибут_номи берилганлар_тури[, атрибут_номи
    берилганлар_тури]...
    [{MAP | ORDER} MEMBER функция_аниқланиши,]
    [{MEMBER | STATIC} {қисмдастур_аниқланиши |
    чақириш_аниқланиши}
    [, {MEMBER | STATIC} { қисмдастур_аниқланиши |
    чақириш_аниқланиши}]...]
  );

[CREATE [OR REPLACE] TYPE BODY тур_номи {IS | AS}
  { {MAP | ORDER} MEMBER функция_танаси;
    | {MEMBER | STATIC} {қисмдастур_танаси |
    чақириш_аниқланиши };}
  [{MEMBER | STATIC} { қисмдастур_танаси |
    чақириш_аниқланиши };}]...
END;]
```

AUTHID жумласи барча member методларининг бажарилиши уларни ташкил қилган фойдаланувчи ёки уларни чақирувчи фойдаланувчи имтиёзи орқали амалга оширилишини аниқлаб беради.

Чақириш аниқланиши эса Java методи ёки ташқи C функциясини Oracle берилганлар луғатида фойдаланишга имкон беради.

Стек объект тури

Стек тартиблаштирилган берилганлар майдончалари тўпламини сақлайди. Номидан кўриниб турибтики, стекнинг боши ва охири мавжуд. Аммо майдонларни қўшиш ёки олиб ташлаш бошидан бошланади. Охирги майдон стекга қўшилди деганда, биринчи майдон чиқарилди деб тушуниш лозим (охиргиси кирди, биринчиси чиқди).

Стеки жуда кўп жойларда ишлатиш мумкин. Масалан тизимий дастурлашда узилишлар приоритетини ўрнатиш ва рекурсияларни бошқаришда ишлатилади. Уларни бутун қийматли массивларда ҳам ишлатса бўлади.

PL/SQL тилида VARRAY, берилганлар тури мавжуд бўлиб, ўзгарувчан ўлчовли массивларни эълон қилишга имкон беради. Varray атрибутини эълон қилиш учун аввал унинг турни аниқлашингиз лозим:

```
CREATE TYPE IntArray AS VARRAY(25) OF INTEGER;
```

Сўнг объект тури аниқланишини ёзишингиз мумкин:

```
CREATE TYPE Stack AS OBJECT (
    max_size INTEGER,
    top      INTEGER,
    position IntArray,
    MEMBER PROCEDURE initialize,
    MEMBER FUNCTION full RETURN BOOLEAN,
    MEMBER FUNCTION empty RETURN BOOLEAN,
    MEMBER PROCEDURE push (n IN INTEGER),
    MEMBER PROCEDURE pop (n OUT INTEGER)
);
```

Ниҳоят объект тури танасини ёзишингиз мумкин:

```
CREATE TYPE BODY Stack AS
    MEMBER PROCEDURE initialize IS
    BEGIN
        top := 0;
```

```

    /* Varray учун конструкторни чақиради ва 1-нчи
элементга NULL қиймат беради. */
    position := IntArray(NULL);
    max_size := position.LIMIT; /*Varray ўлчовини
аниқлайди */
    position.EXTEND(max_size - 1, 1); /* 1-нчи
элементини 2..25 элементларига нусхалайди */
END initialize;

MEMBER FUNCTION full RETURN BOOLEAN IS
BEGIN
    RETURN (top = max_size); /* TRUE қиймат
қайтаради, агарда стек тўлган бўлса */
END full;

MEMBER FUNCTION empty RETURN BOOLEAN IS
BEGIN
    RETURN (top = 0); /* TRUE қиймат қайтаради,
агарда стек бўш бўлса */
END empty;

MEMBER PROCEDURE push (n IN INTEGER) IS
BEGIN
    IF NOT full THEN
        top := top + 1; -- push integer onto stack
        position(top) := n;
    ELSE -- стек тўлган
        RAISE_APPLICATION_ERROR(-20101, 'стек тўлган');
    END IF;
END push;

MEMBER PROCEDURE pop (n OUT INTEGER) IS
BEGIN
    IF NOT empty THEN
        n := position(top);
        top := top - 1; -- pop integer off stack
    ELSE -- stack is empty
        RAISE_APPLICATION_ERROR(-20102, 'stack
underflow');
    END IF;
END pop;
END;

```

Push ва pop member процедураларида сиз ички қурилган raise_application_error процедурани фойдаланувчи томонидан аниқланган хатолик хабарини мижоз дастурига чиқариш учун ишлатасиз. Мижоз дастури ушбу PL/SQL мустасносини SQLCODE ва SQLERRM хатолик-

ҳисоботи функцияларини OTHERS мустаснога ишлов берувчида ишлатишингиз мумкин:

```
DECLARE
    err_num NUMBER;
    err_msg VARCHAR2(100);
BEGIN
    ...
EXCEPTION
    WHEN OTHERS THEN
        err_num := SQLCODE;
        err_msg := SUBSTR(SQLERRM, 1, 100);
        DBMS_OUTPUT.PUT_LINE(TO_CHAR(err_num) || ': ' ||
err_msg);
```

Алтернатив тарзда дастур EXCEPTION_INIT прагмани raise_application_error томонидан қайтарган хатолик рақамларини номланган мустасноларга акслантиришга имкон беради:

```
DECLARE
    stack_overflow EXCEPTION;
    stack_underflow EXCEPTION;
    PRAGMA EXCEPTION_INIT(stack_overflow, -20101);
    PRAGMA EXCEPTION_INIT(stack_underflow, -20102);
BEGIN
    ...
EXCEPTION
    WHEN stack_overflow THEN ...
```

Банк_ҳисоб_рақами объекти тури

Банк ҳисоб рақамини ифодаловчи объект турини аниқлаш учун, лозим бўладиган берилганлар ва амалларни қараб чиқишингиз зарур. Содда банк ҳисоб рақами учун объект тури учун ҳисоб рақами тартиби, баланси ва статус атрибутлари лозим бўлади. Бундан ташқари қуйидаги амалларни бажарувчи методлар ҳам зарур бўлади: ҳисоб рақамини очиш, ҳисоб рақами тартибини текшириш, ҳисоб рақамини ёпиш, депозит пуллари ва балансни қайтарувчи.

Аввал сиз объект тури аниқланишини ёзишингиз лозим:

```
CREATE TYPE Bank_Account AS OBJECT (
    acct_number INTEGER(5),
    balance     REAL,
```

```

        status          VARCHAR2(10),
        MEMBER PROCEDURE open (amount IN REAL),
        MEMBER PROCEDURE verify_acct (num IN INTEGER),
        MEMBER PROCEDURE close (num IN INTEGER, amount OUT
REAL),
        MEMBER PROCEDURE deposit (num IN INTEGER, amount IN
REAL),
        MEMBER PROCEDURE withdraw (num IN INTEGER, amount IN
REAL),
        MEMBER FUNCTION curr_bal (SELF IN OUT Bank_Account,
        num IN INTEGER) RETURN REAL
);

```

Сўнг объект тури танасини ёзасиз:

```

CREATE TYPE BODY Bank_Account AS
    MEMBER PROCEDURE open (amount IN REAL) IS
        -- open account with initial deposit
    BEGIN
        IF NOT amount > 0 THEN
            RAISE_APPLICATION_ERROR(-20104, 'bad amount');
        END IF;
        SELECT acct_sequence.NEXTVAL INTO acct_number FROM
dual;
        status := 'open';
        balance := amount;
    END open;

    MEMBER PROCEDURE verify_acct (num IN INTEGER) IS
        -- check for wrong account number or closed account
    BEGIN
        IF (num <> acct_number) THEN
            RAISE_APPLICATION_ERROR(-20105, 'wrong
number');
        ELSIF (status = 'closed') THEN
            RAISE_APPLICATION_ERROR(-20106, 'account
closed');
        END IF;
    END verify_acct;

    MEMBER PROCEDURE close (num IN INTEGER, amount OUT
REAL) IS
        -- close account and return balance
    BEGIN
        verify_acct(num);
        status := 'closed';
        amount := balance;
    END close;

```

```

MEMBER PROCEDURE deposit (num IN INTEGER, amount IN
REAL) IS
BEGIN
    verify_acct(num);
    IF NOT amount > 0 THEN
        RAISE_APPLICATION_ERROR(-20104, 'bad amount');
    END IF;
    balance := balance + amount;
END deposit;

MEMBER PROCEDURE withdraw (num IN INTEGER, amount IN
REAL) IS
    -- if account has enough funds, withdraw
    -- given amount; else, raise an exception
BEGIN
    verify_acct(num);
    IF amount <= balance THEN
        balance := balance - amount;
    ELSE
        RAISE_APPLICATION_ERROR(-20107, 'insufficient
funds');
    END IF;
END withdraw;

MEMBER FUNCTION curr_bal (SELF IN OUT Bank_Account,
num IN
INTEGER)
RETURN REAL IS
BEGIN
    verify_acct(num);
    RETURN balance;
END curr_bal;
END;
```

Рационал объект тури

Рационал сон иккита бутун сон орқали ифодаловчи сондир. PL/SQL тилида рационал сон тури аниқланмаган. Биз шуни Rational объект тури орқали аниқлаймиз:

```

CREATE TYPE Rational AS OBJECT (
    num INTEGER,
    den INTEGER,
    MAP MEMBER FUNCTION convert RETURN REAL,
    MEMBER PROCEDURE normalize,
    MEMBER FUNCTION reciprocal RETURN Rational,
    MEMBER FUNCTION plus (x Rational) RETURN Rational,
```

```

MEMBER FUNCTION less (x Rational) RETURN Rational,
MEMBER FUNCTION times (x Rational) RETURN Rational,
MEMBER FUNCTION divby (x Rational) RETURN Rational,
PRAGMA RESTRICT REFERENCES (DEFAULT,
RNDS, WNDS, RNPS, WNPS)
);

```

PL/SQL –да ҳар хил операторларни бир хил номлаш таъқиқланган. Шу сабабали сиз +, -, *, ва / операторлар учун номланган plus, less (minus сўзи заҳираланган), times, ва divby методларини аниқлашингиз лозим.

Сўнг қуйидаги алоҳида – сақланувчи ва normalize методи деб номланувчи функцияни аниқлашингиз лозим:

```

CREATE FUNCTION gcd (x INTEGER, y INTEGER) RETURN
INTEGER AS
-- find greatest common divisor of x and y
ans INTEGER;
BEGIN
IF (y <= x) AND (x MOD y = 0) THEN
ans := y;
ELSIF x < y THEN
ans := gcd(y, x); -- recursive call
ELSE
ans := gcd(y, x MOD y); -- recursive call
END IF;
RETURN ans;
END;

```

Кейин эса объект тури танасини ёзишингиз мумкин:

```

CREATE TYPE BODY Rational AS
MAP MEMBER FUNCTION convert RETURN REAL IS
-- convert rational number to real number
BEGIN
RETURN num / den;
END convert;

MEMBER PROCEDURE normalize IS
-- reduce fraction num / den to lowest terms
g INTEGER;
BEGIN
g := gcd(num, den);
num := num / g;
den := den / g;
END normalize;

MEMBER FUNCTION reciprocal RETURN Rational IS

```

```

-- return reciprocal of num / den
BEGIN
    RETURN Rational(den, num); -- call constructor
END reciprocal;

MEMBER FUNCTION plus (x Rational) RETURN Rational IS
-- return sum of SELF + x
    r Rational;
BEGIN
    r := Rational(num * x.den + x.num * den, den *
x.den);
    r.normalize;
    RETURN r;
END plus;

MEMBER FUNCTION less (x Rational) RETURN Rational IS
-- return difference of SELF - x
    r Rational;
BEGIN
    r := Rational(num * x.den - x.num * den, den *
x.den);
    r.normalize;
    RETURN r;
END less;

MEMBER FUNCTION times (x Rational) RETURN Rational IS
-- return product of SELF * x
    r Rational;
BEGIN
    r := Rational(num * x.num, den * x.den);
    r.normalize;
    RETURN r;
END times;

MEMBER FUNCTION divby (x Rational) RETURN Rational IS
-- return quotient of SELF / x
    r Rational;
BEGIN
    r := Rational(num * x.den, den * x.num);
    r.normalize;
    RETURN r;
END divby;
END;

```

Объектларни эълон қилиш ва қиймат бериш

Объект турини бир марта эълон қилиб , сўнг уни ушбу турдаги объектларни ихтиёрий PL/SQL блокада, қисмдастурда ва пакетда эълон қилишда ишлатишингиз мумкин.

Объектларни эълон қилиш

Сиз объект турини CHAR ёки NUMBER каби ички қурилган турлари ишлатиладиган ихтиёрий жойда ишлатишингиз мумкин:

```
DECLARE
    r Rational;
BEGIN
    r := Rational(6, 8);
    DBMS_OUTPUT.PUT_LINE(r.num); -- 6 чоп этади
```

Объектларни функция ёки процедура формал параметрлари сифатида эълон қилишингиз мумкин:

```
DECLARE
    ...
    PROCEDURE open_acct (new_acct IN OUT Account) IS ...
```

Қуйидаги мисолда Account объект тури функция қайтарилиши лозим бўлган қиймат турини аниқлайди:

```
DECLARE
    ...
    FUNCTION get_acct (acct_id IN INTEGER) RETURN Account
IS ...
```

Объектларга қиймат бериш

Объектга унинг тури конструкторини чақириш йўли билан қиймат бермагунга қадар у *ягона бирлик сифатида бўш* бўлади. Яъни нафақат объект атрибутлари, балки унинг ўзи ҳам бўш бўлади:

```
DECLARE
    r Rational; -- r ягона бирлик сифатида бўш
BEGIN
    r := Rational(2,3); -- r 2/3 тенг
```

Бўш объект ҳеч қачон бошқа объектга тенг бўлмайди. Амалда бўш объектни ихтиёрий бошқа объект билан солиштириш натижаси NULL қиймат

қайтаради. Худди шунингдек ягона бирлик сифатида бўш объектни бошқа объектга қиймат сифатида берганда, бошқа объект ягона бирлик сифатида бўш объект бўлади:

```
DECLARE
    r Rational;
BEGIN
    r Rational := Rational(1,2); --r ? бўлади
    r := NULL; -- r ягона бирлик сифатида бўш бўлади
    IF r IS NULL THEN ... -- шарт TRUE қайтаради
```

Дастурлашнинг яхши усули объектни эълон қилишда унга қиймат беришдан иборат:

```
DECLARE
    r Rational := Rational(2,3); -- r 2/3 бўлади
```

PL/SQL бўш объектни қандай текширади

Ифодаalarda бўш объектнинг атрибутлари NULL деб олинади. Бўш объектнинг атрибутларига қиймат беришга уринсангиз, олдиндан аниқланган ACCESS_INTO_NULL мустасно содир бўлади. Агар бўш объект ёки унинг атрибутларига IS NULL солиштириш амалани қўлласангиз натижа TRUE бўлади.

Қуйидаги мисол бўш объект ва бўш атрибутли объект фарқини намоиш этади:

```
DECLARE
    r Rational; -- r ягона бирлик сифатида бўш
BEGIN
    IF r IS NULL THEN ... -- TRUE бўлади
    IF r.num IS NULL THEN ... -- TRUE бўлади
    r := Rational(NULL, NULL); -- r бўш атрибутли объект
    r.num := 4; -- r ягона бирлик сифатида бўш эмас
    r := NULL; -- r яна бўш бўлди
    r.num := 4; -- ACCESS_INTO_NULL содир бўлади
EXCEPTION
    WHEN ACCESS_INTO_NULL THEN
        ...
END;
```

Бўш объект методларини чақириш мумкин, унда SELF параметри NULL билан чегараланган бўлади. Агар IN турдаги параметрга бўш объект атрибутини аргумент сифатида узатсангиз NULL ҳосил бўлади.

Атрибутларга мурожаат қилиш

Сиз атрибутга фақат унинг номи орқали (объект туридаги жойлашган жойи орқали эмас) мурожаат қилишингиз мумкин. Қуйидаги мисолда den атрибути қийматини denominator ўзгарувчига берилади. Сўнг numerator ўзгарувчидаги қиймат num Атрибутига берилади:

```
DECLARE
    r Rational := Rational(NULL, NULL);
    numerator   INTEGER;
    denominator INTEGER;
BEGIN
    ...
    denominator := r.den;
    r.num := numerator;
```

Ичма-ич жойлашган объект туридаги атрибутларга кириш имконияти мавжуд:

```
CREATE TYPE Address AS OBJECT (
    street   VARCHAR2(30),
    city     VARCHAR2(20),
    state    CHAR(2),
    zip_code VARCHAR2(5)
);

CREATE TYPE Student AS OBJECT (
    name           VARCHAR2(20),
    home_address   Address,
    phone_number   VARCHAR2(10),
    status         VARCHAR2(10),
    advisor_name   VARCHAR2(20),
    ...
);
```

Агар s Student кўринишдаги объект бўлса, унинг zip_code атрибутига қуйидагича мурожаат қилишингиз мумкин:

s.home_address.zip_code

Конструкторни чақириш

Функциялар каби конструктор ифоданинг ихтиёрий жойида чақирилиши мумкин:

```
DECLARE
    r1 Rational := Rational(2, 3);
    FUNCTION average (x Rational, y Rational) RETURN
Rational IS
    BEGIN
        ...
    END;
BEGIN
    r1 := average(Rational(3, 4), Rational(7, 11));
    IF (Rational(5, 8) > r1) THEN
        ...
    END IF;
END;
```

Агар конструкторга параметр узатсангиз, мос атрибутларга (n-чи параметр n-чи атрибутга қиймат узатади) бошланғич қиймат берилади:

```
DECLARE
    r Rational;
BEGIN
    r := Rational(5, 6);  -- num 5, den 6 бўлади
    -- now r is 5/6
```

Атрибутга номи бўйича ҳам қиймат бериш мумкин:

```
BEGIN
    r := Rational(den => 6, num => 5);  --
```

Методларни чақириш

Пакетлаштирилган қисмдастурлари каби методлар нукта белигисидан фойдаланиб чақирилади. Қуйидаги мисолда, `normalize` методи чақирилади:

```
DECLARE
    r Rational;
BEGIN
    r := Rational(6, 8);
    r.normalize;
    DBMS_OUTPUT.PUT_LINE(r.num);  -- prints 3
```

Қуйидаги мисол кўрсатадики методларни занжирсимон тарзда кетма-кет чақириш мумкин. Улар чапдан ўнгга қараб бажарилади. Аввал

`reciprocal member` функцияси чақирилади, сўнг эса `normalize member` процедураси чақирилади.

```
DECLARE
```

```
  r Rational := Rational(6, 8);
```

```
BEGIN
```

```
  r.reciprocal().normalize;
```

```
  DBMS_OUTPUT.PUT_LINE(r.num);  -- prints 4
```

SQL жумлаларида параметрсиз методларни чақирсангиз, бўш параметрлар рўйхати талаб қилинади.

Методларни занжирсимон шаклдаги чақиришда процедурадан кейин методни чақириш тақиқ қилинади, чунки процедура ифоданинг қисми сифатида эмас, балки жумла тарзида чақирилади. Масалан қуйидаги мурожаат нотўғри бўлади:

```
r.normalize().reciprocal; -- нотўғри
```

Яна шуни таъкидлаш жоизки, агар сиз иккита функцияни кетма-кет занжирсимон тарзда чақирсангиз, биринчи функция иккинчи функция қабул қилиши мумкин бўлган қийматни қайтариши лозим.

Объектлардан умумий тарзда фойдаланиш

Кўпинча ҳақиқий дунё объектлари `Rational` объектига қараганда каттароқ ва мураккаброкдир. Қуйидаги объект турини қараймиз:

```
CREATE TYPE Address AS OBJECT (
  street_address VARCHAR2(35),
  city            VARCHAR2(15),
  state           CHAR(2),
  zip_code        INTEGER
);
```

```
CREATE TYPE Person AS OBJECT (
  first_name      VARCHAR2(15),
  last_name       VARCHAR2(15),
  birthday        DATE,
  home_address    Address,  -- ичма-ич жойлашган объект
  phone_number    VARCHAR2(15),
  ss_number       INTEGER,
```

тури

```
); ...
```

Address объекти туртта атрибутдан иборат ва Person объекти эса ундан ҳам кўп атрибутларга эга бўлиб, улардан бири Address туридаги объектдир. Объект катта бўлса, уни бир қисмдастурдан иккинчи қисмдастурга кўчириш самарасиз бўлади. Шу сабабли объектлардан умумий тарзда фойдаланиш фойдалироқ бўлади. Бу ишни амалга ошириш учун объект ўзининг идентификаторига эга бўлиши лозим. Объектдан умумий тарзда фойдаланиш учун сиз унга ҳаволани (қисқача refs, references сўзидан) ишлатасиз. Ref бу объектга кўрсаткичдир. Объектдан умумий тарзда фойдаланишнинг иккита муҳим ижобий томони мавжуд. Биринчиси – берилганлар заруриятсиз репликация қилинмайди. Иккинчиси – умумий тарзда фойдаланиши лозим бўлган объект қиймати ўзгартирилганда, ўзгартириш фақат бир жойда амалга оширилади ва ихтиёрий ҳавола ўзгартирилган қийматларни очиб ўқиши мумкин.

Қуйидаги мисолда биз умумий тарзда фойдаланиладиган объект ижобий томонларидан бирини Home объект турини аниқлаб ва сўнг ушбу турдаги объект нусхаларидан таркиб топган жадвални очиш орқали намойиш этамиз:

```
CREATE TYPE Home AS OBJECT (
  address    VARCHAR2(35),
  owner      VARCHAR2(25),
  age        INTEGER,
  style       VARCHAR(15),
  floor plan BLOB,
  price      REAL(9,2),
  ...
);
...
CREATE TABLE homes OF Home;
```

Refs-дан фойдаланиш

Person турдаги объектда бир хил уй адресига эга бўлган одамларни моделлаштириш мумкин. Объект кўрсатгичини ўз ичига олган атрибутни эълон қилиш учун REF –дан фойдаланасиз:

```

CREATE TYPE Person AS OBJECT (
    first_name    VARCHAR2(10),
    last_name     VARCHAR2(15),
    birthday      DATE,
    home_address  REF Home,    /* оила аъзолари умумий адрес
учун фойдаланишлари мумкин */
    phone_number  VARCHAR2(15),
    ss_number     INTEGER,
    mother        REF Person,  /* оила аъзолари бир-бирига
ҳавола қилиши мумкин */
    father        REF Person,
    ...
);

```

Notice how references from persons to homes and between persons model real-world relationships.

Сиз refs-ни ўзгарувчи, параметр, майдон ёки атрибут тарзида ишлатишингиз мумкин. Refs –ни SQL жумлаларида киритиш ёки чиқариш ўзгарувчилари сифатида ҳам фойдаланишингиз мумкин. Аммо refs йўналиши бўйлаб мурожаат қилиб бўлмайди. Агар `x.attribute`, ифодада `x ref` бўлса, PL/SQL, ҳавола этилаётган объектни сақлаётган жадвалга ўтолмайди. Масалан қуйидаги қиймат бериш қоидага хилоф:

```

DECLARE
    p_ref    REF Person;
    phone_no VARCHAR2(15);
BEGIN
    phone_no := p_ref.phone_number;  -- нотўғри

```

Буни ўрнига сиз объектга киришингиз учун Deref операторидан фойдаланишингиз лозим .

Турни олдиндан аниқлаш

Сиз фақат мавжуд бўлган схема объектига ҳавола этишингиз мумкин. Қуйидаги мисолда биринчи CREATE TYPE жумласи ўринсиз бўлади , чунки у ҳали мавжуд бўлмаган Department, турдаги объектга ҳавола этаяпти:

```

CREATE TYPE Employee AS OBJECT (
    name VARCHAR2(20),
    dept REF Department,  -- нотўғри
    ...
);

```

```
CREATE TYPE Department AS OBJECT (
    number INTEGER,
    manager Employee,
    ...
);
```

CREATE TYPE жумлалари ўрнини алмаштириб муаммони ҳал этиб бўлмайди, чунки бу турлар бир-бирига мурожаат қилаяпти. Ушбу муаммони ҳал этиш учун, махсус *турни олдиндан аниқлаш* деб номланувчи CREATE TYPE жумласини ишлатиш лозим:

```
CREATE TYPE Department; --турни олдиндан аниқлаш /*бу ерда
Department—тугатилмаган объект тури*/
```

Тури олдиндан аниқланган объект тури *тугатилмаган объект тури* (то у тўлиқ аниқлангунча)деб номланади, чунки у хали атрибутлар ва методларга эга эмас.

Объектларни бошқариш

Объект турини CREATE TABLE жумласида устуннинг берилганлари турини аниқлашда ишлатишингиз мумкин. Жадвални бир марта аниқлаб, SQL жумлаларини объектни киритиш, унинг атрибутларини сўров қилиш, унинг методларини чақириш ва унинг ҳолатини ўзгартириш учун ишлатиш мумкин.

Қуйидаги SQL*Plus скриптида INSERT жумласи Rational объект тури учун конструкторни чақиради ва натижавий объектни жадвалга киритади. SELECT жумласи num атрибути қийматини сўров қилиб олади. UPDATE жумласи reciprocal member методини чақиради.

```
CREATE TABLE numbers (rn Rational, ...)
/
INSERT INTO numbers (rn) VALUES (Rational(3, 62)) --
3/62
/
SELECT n.rn.num INTO my_num FROM numbers n ... - 3
қайтаради
/
```

```
UPDATE numbers n SET n.rn = n.rn.reciprocal ... -- 62/3
қайтаради
/
```

Агар сиз объектларни шу тарзда ташкил қилсангиз, улар жадвалдан ташқарида ўзининг идентификаторига эга бўлмайди. Аммо объект тури ихтиёрий жадвалга боғлиқ бўлмаган тарзда мавжуд бўлади ва уларни объектларни бошқа усулда ташкил қилишда ишлатиш мумкин.

Қуйидаги мисолда, қаторлари Rational турдаги объектлардан иборат жадвал ташкил қилинади. Объектли қаторлардан таркиб топган бундай жадваллар объектли жадваллар деб номланади. Қатордаги ҳар бир устун объект турининг атрибутига мос келади.

```
CREATE TABLE rational_nums OF Rational;
```

Объектли жадвалнинг ҳар бир қатори объектни бир қийматли равишда аниқлаб берувчи ва айнан шу қаторда сақланувчи *объект идентификаторига* эга бўлиб объектга ҳавола қилиш учун хизмат қилади.

Объектларга сўров бериш

Фараз қиламиз Person объект тури ва persons объектли жадвални ташкил қилувчи қуйидаги SQL*Plus скриптини бажаришга юбордингиз ва ушбу жадвални тўлидириш ниятидасиз:

```
CREATE TYPE Person AS OBJECT (
    first_name    VARCHAR2(15),
    last_name     VARCHAR2(15),
    birthday      DATE,
    home_address  Address,
    phone_number  VARCHAR2(15))
/
CREATE TABLE persons OF Person
/
```

Қуйидаги сўровоти фақат Person объектлари атрибутларидан таркиб топган натижавий қаторлар тўпламини ишлаб чиқади:

```
BEGIN
```

```

INSERT INTO employees -- Person туридаги бошқа
объектлик жадвал
SELECT * FROM persons p WHERE p.last_name LIKE
'%Smith';

```

Нативий объектлар тўпламини қайтариш учун VALUE операторидан фойдаланишингиз лозим.

VALUE операторидан фойдаланиш

VALUE оператори объект қийматини қайтаради. VALUE оператори аргумент сифатида корреляцион ўзгаришчи қабул қилади. Биз бу ерда корреляцион ўзгаришчи деганда, қаторли ўзгаришчи ёки объектлик жадвалнинг қатори билан боғлиқ бўлган жадвал ҳамроҳини тушунамиз. Масалан Person туридаги нативий объектлар тўпламини қайтариш учун VALUE операторидан қуйидагича фойдаланинг:

```

BEGIN
  INSERT INTO employees
    SELECT VALUE(p) FROM persons p
    WHERE p.last_name LIKE '%Smith';

```

Қуйидаги мисолда Person туридаги махсус объектни қайтариш учун VALUE операторидан фойдаланган:

```

DECLARE
  p1 Person;
  p2 Person;
  ...
BEGIN
  SELECT VALUE(p) INTO p1 FROM persons p
    WHERE p.last_name = 'Kroll';
  p2 := p1;
  ...
END;

```

Бу ерда p1 локал Person объектидан таркиб топган бўлиб, охириги номи 'Kroll' бўлган сақланадиган объект нусхасидир ва p2 бошқа локал Person объекти бўлиб p1 объект нусхасидир:

```

BEGIN
  p1.last_name := p1.last_name || 'Jr';

```

Энди p1 локал Person объектидаги охирги ном 'Kroll Jr' иборат.

REF операторидан фойдаланиш

Refs-ни REF операторидан фойдаланиб очишингиз мумкин, бунда у аргумент сифатида корреляцион ўзгарувчини қабул қилади. Қуйидаги мисол ёрдамида сиз Person объектлари учун битта ёки ундан кўп refs очиб, сўнг уларни person_refs жадвалига ёзиб қўйишингиз мумкин:

```
BEGIN
  INSERT INTO person_refs
    SELECT REF(p) FROM persons p
    WHERE p.last_name LIKE '%Smith';
```

Қуйидаги мисолда сиз ref ва атрибутни бир вақтнинг ўзида очишингиз мумкин:

```
DECLARE
  p_ref      REF Person;
  taxpayer_id VARCHAR2(9);
BEGIN
  SELECT REF(p), p.ss_number INTO p_ref, taxpayer_id
    FROM persons p
    WHERE p.last_name = 'Parker'; -- битта қитор
    қайтариши лозим
  ...
END;
```

Охирги мисолда сиз Person объекти қийматини ўзгартирасиз:

```
DECLARE
  p_ref      REF Person;
  my_last_name VARCHAR2(15);
  ...
BEGIN
  ...
  SELECT REF(p) INTO p_ref FROM persons p
    WHERE p.last_name = my_last_name;
  UPDATE persons p
    SET p = Person('Jill', 'Anders', '11-NOV-67', ...)
    WHERE REF(p) = p_ref;
END;
```

Хавола осилишини текшириш

Агарда ref кўрсатиб турган объект йўқотилса ref *осилган* ҳолатда (мавжуд бўлмаган объектни кўрсатиб) қолади. Ушбу шартни текшириш учун IS DANGLING SQL предикатидан фойдаланишингиз мумкин:

```
BEGIN
```

```
UPDATE department SET manager = NULL WHERE manager IS
DANGLING;
```

DEREF операторидан фойдаланиш

Refs кўрсатаётган объект қийматларини ўқиб олиш учун Deref операторидан фойдаланишингиз мумкин. in a SQL statement. (Deref is short for dereference. When you *dereference* a pointer, you get the value to which it points.) Deref аргумент сифатида объектга ҳаволани қабул қилиб, сўнг ҳавола этилаётган объект қийматини қайтаради.

```
DECLARE
```

```
    p1      Person;
    p_ref REF Person;
    name    VARCHAR2(15);
```

```
BEGIN
```

```
    ...
    /* Фараз қиламиз p_ref объектли жадвалда сақланаётган
    объектга ҳаволасини сақлайди*/
    SELECT Deref(p_ref) INTO p1 FROM dual;
    name := p1.last_name;
```

```
CREATE TYPE PersonRef AS OBJECT (p_ref REF Person)
/
```

```
DECLARE
```

```
    name    VARCHAR2(15);
    pr_ref REF PersonRef;
    pr      PersonRef;
    p       Person;
```

```
BEGIN
```

```
    ...
    /* Фараз қиламиз pr_ref тўғри ҳаволадан тиркиб
    топган. */
    SELECT Deref(pr_ref) INTO pr FROM dual;
    SELECT Deref(pr.p_ref) INTO p FROM dual;
```

```

    name := p.last_name;
    ...
END
/

```

Қуйидаги мисол Deref операторини процедуравий жумлаларда ишлатиш мумкин эмаслигини кўрсатади:

```

BEGIN
    ...
    p1 := Deref(p_ref);  -- нотўғри

```

Қуйидаги синтаксис ёзувлар тўғри:

```

table_alias.object_column.ref_attribute
table_alias.object_column.ref_attribute.attribute
table_alias.ref_column.attribute

```

Фараз қиламиз Address ва Person, ҳамда persons объектли жадвални ташкил этувчи қуйидаги SQL*Plus скрипти бажарилди.

```

CREATE TYPE Address AS OBJECT (
    street    VARCHAR2(35),
    city      VARCHAR2(15),
    state     CHAR(2),
    zip_code  INTEGER)
/
CREATE TYPE Person AS OBJECT (
    first_name  VARCHAR2(15),
    last_name   VARCHAR2(15),
    birthday    DATE,
    home_address REF Address,  -- бошқа Person объектлари
    phone_number VARCHAR2(15))
/
CREATE TABLE persons OF Person
/

```

Ref турдаги home_address атрибути persons объектли жадвалнинг устунига мос келиб, у бошқа жадвалда сақланаётган Address объектларига ҳаволадан таркиб топган. Жадвални тўлдирганингиздан кейин, сиз қуйида кўрсатилганидек хусусий адресларни сўров қилиб олишингиз мумкин:

```

DECLARE
    addr1 Address,
    addr2 Address,
    ...
BEGIN

```

```
SELECT Deref(home_address) INTO addr1 FROM persons p
WHERE p.last_name = 'Derringer';
```

Қуйдаги мисолда home_address ref устун орқали street атрибутига мурожаат қилинади. Бу ҳолатда жадвал ҳамроҳи талаб қилинади.

```
DECLARE
    my_street VARCHAR2(25),
    ...
BEGIN
    SELECT p.home_address.street INTO my_street FROM
persons p
    WHERE p.last_name = 'Lucas';
```

Объектларни киритиш

Объектли жадвалга объектларни киритиш учун INSERT жумласидан фойдаланиш лозим. Қуйдаги мисолда Person объекти persons объектли жадвалга киритилади.

```
BEGIN
    INSERT INTO persons
    VALUES ('Jenifer', 'Lapidus', ...);
```

Alternatively, you can use the constructor for object type Person to insert an object into object table persons: объектли жадвалга объектни киритиш учун Person объект тури конструкторидан фойдаланиш мумкин:

```
BEGIN
    INSERT INTO persons
    VALUES (Person('Albert', 'Brooker', ...));
```

Қуйдаги мисолда Person ҳаволасини локал ўзгарувчида сақлаш учун RETURNING жумласидан фойдаланган:

```
DECLARE
    p1_ref REF Person;
    p2_ref REF Person;
    ...
BEGIN
    INSERT INTO persons p
    VALUES (Person('Paul', 'Chang', ...))
    RETURNING REF(p) INTO p1_ref;
    INSERT INTO persons p
    VALUES (Person('Ana', 'Thorne', ...))
    RETURNING REF(p) INTO p2_ref;
```

Объектли жадвалга объектларни киритиш учун айнан шу турдаги объектларни қайтарувчи сўровостилардан ҳам фойдаланиш мумкин:

```
BEGIN
  INSERT INTO persons2
    SELECT VALUE(p) FROM persons p
    WHERE p.last_name LIKE '%Jones';
```

Қуйидаги мисолда Person турдаги устунга эга бўлган department деб номланувчи реляцион жадвал ташкил қилинади.

```
CREATE TABLE department (
  dept_name VARCHAR2(20),
  manager    Person,
  location   VARCHAR2(20))
/
INSERT INTO department
  VALUES ('Payroll', Person('Alan', 'Tsai', ...), 'Los
  Angeles')
/
```

Объектларни ўзгартириш

Объектли жадвалдаги объект атрибутларини ўзгартириш учун UPDATE жумласидан қуйидаги мисолда кўрсатилганидек фойдаланиш лозим:

```
BEGIN
  UPDATE persons p SET p.home_address = '341 Oakdene Ave'
    WHERE p.last_name = 'Brody';
  ...
  UPDATE persons p SET p = Person('Beth', 'Steinberg',
  ...)
    WHERE p.last_name = 'Steinway';
  ...
END;
```

Объектларни йўқотиш

Объектли жадвалдан объектларни чиқариб ташлаш учун DELETE жумласидан фойдаланиш лозим:

```
BEGIN
  DELETE FROM persons p
    WHERE p.home_address = '108 Palm Dr';
  ...
END;
```

1-§ Oracle сервер концепцияси

Oracle сервер концепциясини ва унда ишлатиладиган терминлар ҳақида фикрларимизни баён этамиз.

Oracle серверига кириш

- Берилганлар базаси ва ахборотларни бошқариш
- Берилганлар базаси тузилмаси ва бошқарув фазоси
- Хотира тузилмаси ва жараёнлар
- Берилганлар конкуренцияси ва яхлитлиги
- Тақсимланган жараёнлар ва тақсимланган берилганлар базаси
- Юклатиш ва ёпиш амаллари
- Берилганлар базаси махфийлиги
- Берилганлар базасини тиклаш
- Берилганлар базаси бошқаруви учун объект-реляцион модел
- Берилганларга кириш имкониятлари

Берилганлар базаси ва ахборотларни бошқариш

Берилганлар базасининг сервери ахборотни бошқариш муаммосини ҳал қилишга калит бўлиб хизмат қилади. Умуман айтганда сервер жуда катта ҳажмдаги ахборотни ўз ичига сақлаб, кўплаб фойдаланувчиларни бир вақтда бир хил ахборот билан ишлашига имкон беради. Шу билан бирга, берилганлар базасининг сервери берилганларни бегоналардан самарали ҳимоя қилиб қайта тиклаш имконига эга.

Oracle сервери мижоз/сервер (тақсимланган жараён) муҳитларига эга бўлиб, берилганлар базасининг сервери ва мижозлари орасида жараённи ташкил қилишга имконият яратади. Сервер қисми берилганлар устида амалларни бажаради, мижоз дастури эса асосий эътиборни берилганларни таҳлил қилиб экранда кўрсатишга масъулдир.

Oracle жуда катта ҳажмдаги (терабайтларгача) берилганлар базасини сақлашга имкон беради. Айни пайтда ушбу берилганлар базаси билан бир вақтнинг ўзида минглаб фойдаланувчилар айнан бир хил берилганлар билан

ишлашлари мумкин. Бунда улар жараён секин кечишини деярли сезмайдилар ва берилганлар яхлитлиги таъминланади.

Шуни таъкидлаш жоизки, Oracle берилганларга кириш стандарт тилларини, операцион тизимларини, фойдаланувчи интерфейсларини ва тармоқ коммуникацион протоколларини қўллаб қувватлайди. У очик турдаги тизимдир.

Oracle сервери ANSI/ISO SQL92 (Structured Query Language) стандартини қўллаб қувватлайди. Oracle тизимини бошқариш учун Simple Network Management Protocol (SNMP) стандартини қўллаб қувватлайди.

Oracle берилганлар яхлитлигини таъминлайди. Oracle тизимида ҳар хил жойларда жойлашган (тармоқга уланган) берилганлар базаларини мантиқан битта берилганлар базаси сифатида қараш имконияти мавжуд. Бундай тарзда тузилган берилганлар базаси **таксимланган берилганлар базаси** дейилади.

Oracle тизими ҳар хил операцион тизимлар асосида ишлайди, бунда бир вақтнинг ўзида битта тармоқда ҳар хил турдаги компьютерлар ва операцион тизимлар мавжуд бўлиши мумкин.

Oracle Сервери

Oracle Сервери - бу объект-реляцион берилганлар базасини бошқарув тизими бўлиб, Oracle берилганлар базаси ва Oracle сервери нусхасидан таркиб топган.

SQL (Structured Query Language) Тузилмаланган сўровлар тили

SQL (SEQUEL деб ўқилади) дастурлаш тили бўлиб, берилганлар базасини аниқлаш ва бошқариш учун хизмат қилади. SQL берилганлар базаси реляцион берилганлар базасидир. Содда қилиб айтганда, берилганлар муносабатлар тўплами сифатида сақланади. Берилганлар базаси битта ёки ундан кўп жадвалларни ўз ичига олиши мумкин. Ҳар бир жадвал устунлар ва қаторлардан иборат. Сиз берилганларни SQL командалари орқали бошқаришингиз мумкин. SQL командалари DDL (data definition language) командалари ва DML (data manipulation commands) командаларидан иборат.

Бундан ташқари Oracle сервери PL/SQL - процедуравий тил билан таъминланган. PL/SQL тили SQL жумлаларини дастурлашга имкон беради.

Берилганлар базасининг тузилиши

Oracle берилганлар базаси жисмоний ва мантикий тузилмалардан иборат

Oracle берилганлар базаси жисмоний тузилмаси.

Oracle берилганлар базаси жисмоний тузилмаси берилганлар базасини ташкил қилувчи операцион тизим файлларидан иборат. Хар бир Oracle берилганлар базаси учта турдаги файллардан: битта ва ундан кўп берилганлар файллари, иккита ва ундан кўп "redo log" файллари, битта ва ундан кўп назорат файлларидан иборат.

Oracle берилганлар базаси мантикий тузилмаси

Oracle берилганлар базаси мантикий тузилмаси қуйидагилардан таркиб топган:

- битта ва ундан кўп жадваллар фазоси;

Жадваллар фазоси - мантикий сақлаш жойи бўлиб, уни кейинроқ батафсил тушунтирамиз.

- берилганлар базаси схемасининг объектлари ;

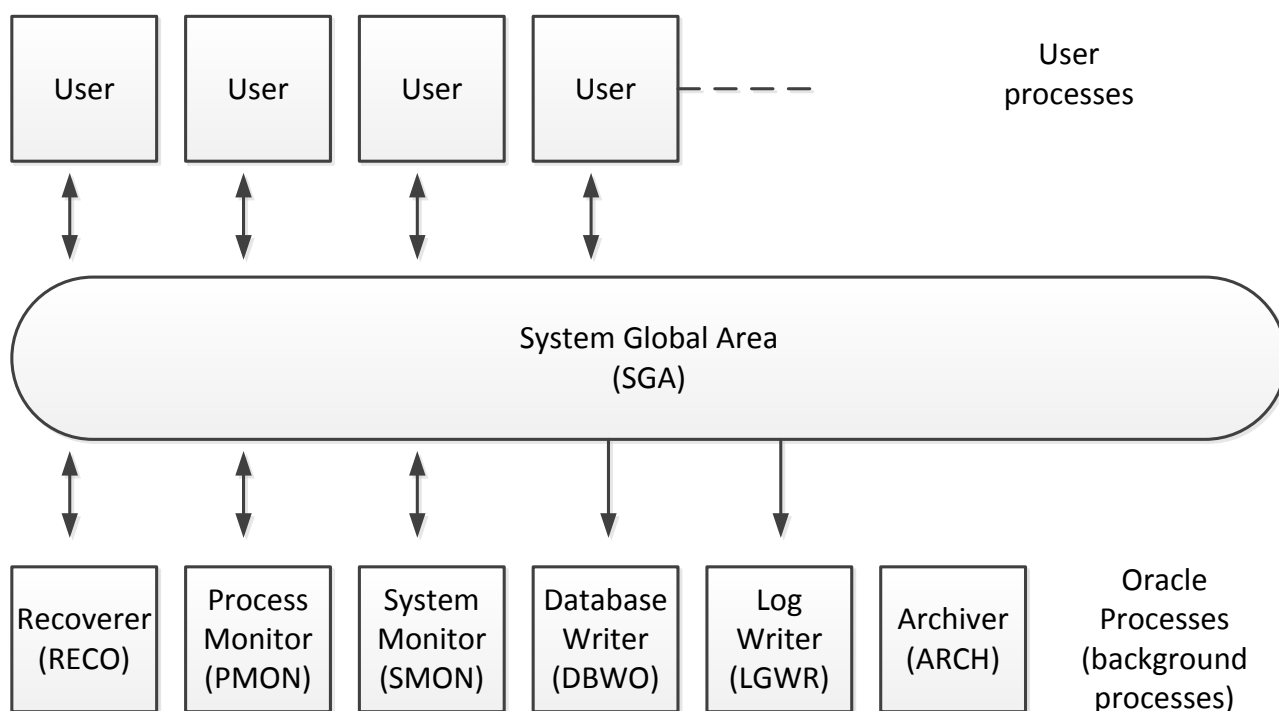
Схема - бу объектлар мажмуасидир. Схема объектлари мантикий тузилма бўлиб, берилганлар базасидаги берилганларга бевосита хавола этади. Схема объектлари жадваллар, кўринишлар, кетма-кетликлар, сақланадиган процедуралар, синонимлар, индекслар, кластерлар ва берилганлар базасининг алоқалари каби тузилмалардан иборат.

Oracle нусхаси

Берилганлар базаси ҳар гал юклатилганда, тизимий глобал соҳаси тақсимланади, ва Oracle фоний жараёнлари бошланади. Тизимий глобал соҳа - бу берилганлар базасининг ахбороти учун фойдаланадиган хотира соҳаси бўлиб, берилганлар базасининг барча фойдаланувчилари учун фойдаланиш имкони мавжуд. Фон жараёнлари ва хотира буферлари комбинацияси Oracle нусхаси дейилади.

Oracle нусхаси икки турдаги жараёнларга эга: фойдаланувчи жараёнлари ва Oracle жараёнлари.

- Фойдаланувчи жараёнлари (Oracle Forms -га ўхшаган дастур) дастур кодини бажаради.
- Oracle жараёнлари сервер жараёнлари ҳисобланиб фойдаланувчи жараёнлари учун ва Oracle серверига хизмат кўрсатувчи фон жараёнлари ишини бажаради.



1.1- расм. Oracle нусхаси.

1.1- расм Кўп-жараёнли Oracle нусхасини намоиш этади.

Алоқавий дастурий таъминоти ва Net8

Агар фойдаланувчи ва сервер жараёнлари тармоқнинг ҳар хил компьютерларида бўлса ёки фойдаланувчи жараёнлари сервернинг умумий жараёнларига диспетчер жараёни орқали мурожаат қилса, фойдаланувчи жараёнлари ва сервер жараёнлари ўзаро Net8 орқали алоқа қилишади. Net8 - стандарт алоқа протоколларига “Oracle”-нинг интерфейси бўлиб, компьютерлар орасидаги берилганлар алмашинувини таъминлайди.

Oracle параллел сервери: Кўп нусхали тизимлар

Айрим аппарат қурилмаларнинг ахитектураси (масалан умумий фойдаланадиган диск тизимлари) бир вақтнинг ўзида кўп компьютерларнинг берилганлар, дастурий таъминот, ёки периферия қурилмалари билан биргаликда ишлашига имкон беради.

Oracle параллел сервери бундай архитектура афзаллигини битта берилганлар базаси билан ишлашда, кўп нусхасини юклатиш орқали амалга оширади.

Oracle берилганлар базаси

Oracle берилганлар базаси - яхлит бирлик сифатида ишлов берадиган берилганлар мажмуасидир. Берилганлар базасининг умумий мақсади чамбарчас боғланган ахборотни сақлаш ва излашдан иборат. Берилганлар базаси мантиқий ва жисмоний тузилмаларга эга.

Очиқ ва ёпиқ берилганлар базаси

Oracle берилганлар базаси очиқ (кириш мумкин) ва ёпиқ (кириш мумкин эмас) ҳолатда бўлиши мумкин. Нормал ҳолатда, берилганлар базаси очиқ ҳолатда бўлиб, фойдаланишга имконият мавжуд бўлади. Аммо, айрим ҳолларда махсус маъмурий ишларни бажариш мақсадида, берилганлар базаси ёпиқ ҳолда бўлиб, фойдаланувчиларнинг берилганлар базасидаги берилганларга киришни таъқиқлаб қўйиш талаб этилади.

2-§ Берилганлар базасининг тузилмаси ва фазони бошқариш

Бу бўлимда Oracle берилганлар базасининг архитектураси, жумладан берилганлар базасини ташкил этувчи жисмоний ва мантиқий тузилмалари тавсифланади.

Oracle берилганлар базаси - яхлит бирлик сифатида ишлов берадиган берилганлар мажмуасидир. Берилганлар базасининг умумий мақсади чамбарчас боғланган ахборотни сақлаш ва излашдан иборат. Берилганлар базаси мантиқий ва жисмоний тузилмаларга эга.

Oracle берилганлар базаси мантиқий тузилмаси

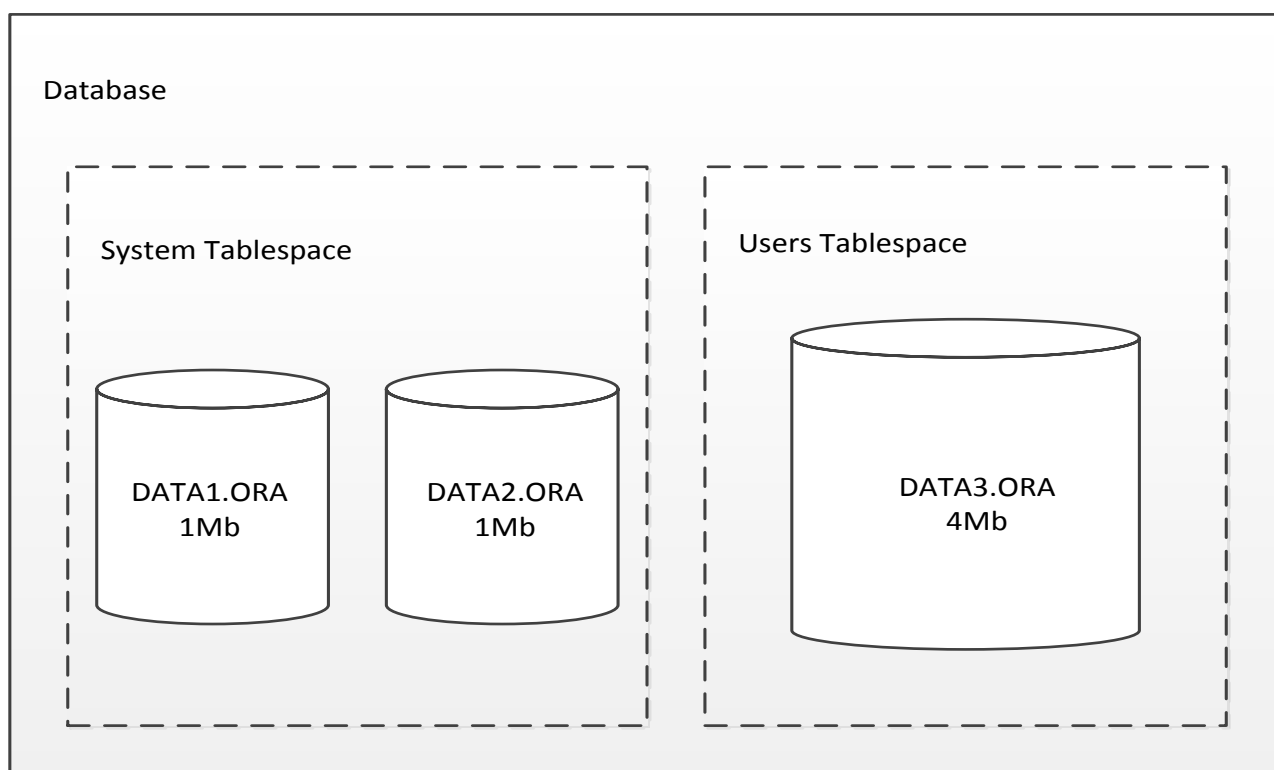
Қуйида биз берилганлар базаси мантиқий тузилмасини, жумладан жадвалий фазоларни, схема объектларини, берилганлар блокини, экстендларни ва сегментларни тушунтириб ўтамиз.

Жадвалий фазолар

Берилганлар базаси жадвалий фазолар деб номланувчи мантиқий сақловчи birlikларга бўлинади. Жадвалий фазолар мантиқий тузилмаларни бир гуруҳга йиғиш учун ишлатилади.

Берилганлар базалари, жадвалий фазолар, ва берилганлар файллари

Берилганлар базалари, жадвалий фазолар, ва берилганлар файллари орасидаги боғланиш 1-2 расмида намоиш этилган.



1-2 расм. Берилганлар базалари, жадвалий фазолар, ва берилганлар файллари.

Бу расм қуйидагини намоиш этади:

Хар бир берилганлар базаси мантиқан битти ёки ундан кўп жадвалий фазолардан таркиб топган. Хар бир жадвалий фазо учун, ундаги барча мантиқий тузилмаларни жисмонан сақлаш мақсадида, битта ёки ундан кўп берилганлар файллари ошкор тарзда очилади. Жадвалий фазодаги

берилганлар файллариинг умумий ўлчами, жадвалий фазонинг умумий сақлаш сиғимлиги дейилади (SYSTEM жадвалий фазоси 2 MB сақлаш сиғимлигига эга, шу билан бирга USERS жадвалий фазоси эса 4 MB сақлаш сиғимлигига эга).

Актив ва автоном жадвалий фазолар

Жадвалий фазолар актив (кириш мумкин) ёки автоном (кириш мумкин эмас) ҳолатларда бўлиши мумкин. Одатда нормал ҳолатда жадвалий фазо актив ҳолатда бўлиб, фойдаланувчилар ундаги маълумотларга кириш имкониятлари мавжуд бўлади. Аммо, баъзан жадвалий фазони автоном ҳолатга, яъни ундаги бир қисм берилганларига киришни таъқиқлаб, қолган қисм берилганларга нормал кириш ҳуқуқини беришга тўғри келади. Бу эса баъзи маъмурий ишларни осон ҳал қилишга имконият яратади.

Схемалар ва схема объектлари

Схема - бу берилганлар базаси объектларининг мажмуасидир. Схема объектлари эса берилганлар базасидаги берилганларга бевосита хавола этувчи мантиқий тузилмалардир. Схема объектлари жадваллар, кўринишлар, кетма-кетликлар, сақланадиган процедуралар, синонимлар, индекслар ва берилганлар базаси боғланишлари каби тузилмаларни ўз ичига олади.

Берилганлар блоклари, экстендлари ва сегментлари

Oracle тизими мантиқий сақлаш тузилмалари, жумладан берилганлар блоклари, экстендлари ва сегментлари ёрдамида диск фазосини ишлатилишини назорат қилиш имкониятига эга.

Oracle берилганлар блоклари

Oracle берилганлар базасининг берилганларини базада сақлаш учун энг кичик сиғимликга эга бўлган блоklar ишлатилади. Берилганларнинг бир блоки берилганлар базаси жисмоний фазосининг дискдаги маълум миқдордаги байтларга мос келади. Блок ўлчови ҳар бир Oracle берилганлар

базасини ташкил қилиш пайтида аниқланади. Берилганлар базаси ундаги бўш фазони Oracle берилганлар блокларига бўлиб ишлатади.

Экстентлар

Мантиқий берилганлар базаси фазосининг навбатдаги даражаси экстент дейилади.

Экстент - бу маълум миқдордаги узлуксиз берилганлар блоклари бўлиб , махсус турдаги ахборотни сақлаш учун ишлатилади.

Сегментлар

Экстентдан юқори турадиган мантиқий берилганлар базасининг сақловчи сатхи - сегмент дейилади. Сегмент маълум мантиқий тузилмада жойлашган экстентлар тўпламидан иборат.

Берилганлар сегменти

Хар бир кластерланмаган жадвал берилганлар сегментига эга. Жадвалнинг барча берилганлари унинг берилганлар сегментида сақланади. Хар бир кластер берилганлар сегментига эга. Кластерда жойлашган хар бир жадвалнинг берилганлари кластернинг берилганлар сегментида сақланади.

Индекс сегменти

Хар бир индекс унинг берилганларини сақлаш учун индекс сегментига эга.

Тиклаш сегменти (Rollback Segment)

Олдинги ҳолатга қайтиш учун зарур бўладиган берилганларни вақтинча сақлаш учун, битти ёки ундан кўп миқдорда тиклаш сегменти очиб қўйилади.

Вақтинчалик сегмент

Вақтинчалик сегмент, SQL жумласи ўз ишини тугатиш учун лозим бўладиган вақтинчалик соҳа тарзида Oracle томонидан ажратилади. SQL жумласи бажарилгандан сўнг вақтинчалик сегмент экстентлари келгусида ишлатиш учун системага қайтарилади.

Oracle динамик тарзда, сегментнинг экстенглари тўлганда, фазо ажратади. Шу сабабли сегмент экстенглари дискда узлуксиз равишда жойлашмаган бўлиши мумкин.

Жисмоний берилганлар базасининг тузилмаси.

Қуйидаги бўлимларда жисмоний берилганлар базасининг тузилмаси, жумладан берилганлар файллари, тиклаш журнали файллари, ва назорат файллари ҳақида сўз юритилади.

Берилганлар файллари

Хар бир Oracle берилганлар базаси битта ёки ундан кўп жисмоний берилганлар файлларига эга бўлиши мумкин. Берилганлар базасининг берилганлар файллари берилганлар базасидаги барча берилганларни сақлайди.

Берилганлар файллари қуйидаги характеристикаларга эга:

- Берилганлар файли фақат битта берилганлар базаси билан боғлиқ бўлади.
- Берилганлар базасининг файллари автоматик тарзда, берилганлар базаси тўлиб кетганда, кенгайиш хусусиятига эга бўлиши мумкин.
- Битта ёки ундан кўп берилганлар файллари жадвал фазоси деб номланувчи мантиқий бирлик сақловчисини ташкил этади.

Тиклаш журнали файллари

Хар бир Oracle берилганлар базаси икки ва ундан кўп тиклаш журнали файлларига эга бўлиши мумкин. Тиклаш журнали файллари тўпламини берилганлар базасининг қайд журнали дейилади. Қайд журналининг бош вазифаси берилганлар барча ўзгаришини ёзиб боришдир.

Назорат файллари

Хар бир Oracle берилганлар базаси назорат файлига эга. Назорат файли берилганлар базаси жисмоний тузилмасини аниқлаб беради. Масалан, бу қуйидаги турдаги ахборотларни ўз ичига олади:

- берилган базасининг номи;
- берилганлар базасининг берилганлар файллари ва тиклаш журнали файлларининг номлари ва жойланиши;
- берилганлар базаси ташкил қилинган вақти.

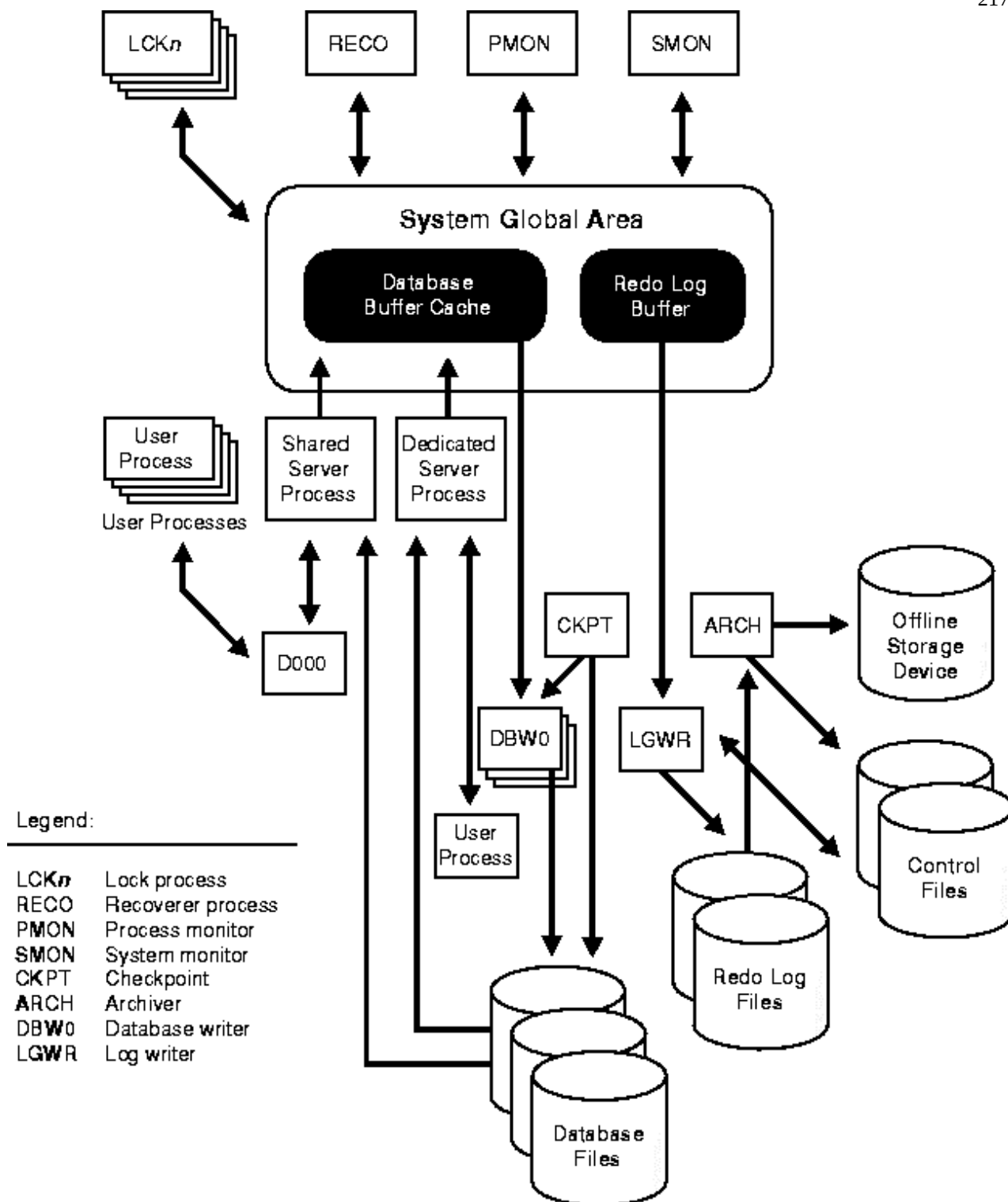
3-§ Хотира ва жараёнлар тузилмаси

Ушбу бўлимда Oracle сервери томонидан берилганлар базасини бошқариш учун ишлатиладиган хотира ва жараёнлар тузилмасини муҳофаза қиламиз. Шу билан бирга, Oracle сервери қуйидаги қобилиятларини тушунтириб берувчи архитектуравий имкониятларини ёритилади:

- бир вақтнинг ўзида битта берилганлар базасига кўп фойдаланувчиларнинг муносабат қилиши;
- параллел равишда кўп-фойдаланувчи ва кўп-амалий дастурли берилганлар базавий тизимлари юқори самарадорликни талаб қилади.

Oracle сервери хотира ва жараёнлар тузилмасини берилганлар базасига кириш ва уни бошқариш учун ишлатади. Барча хотира тузилмалари компьютернинг асосий хотирасида мавжуд бўлади. Жараёнлар - ушбу компьютерлар хотирасида ишлайдиган топшириқлардан таркиб топади.

1.3-расмда Oracle сервери хотираси ва жараёнлари тузилмалари намоёниш этилган.



1.3-расм. Oracle жараёнлари ва хотира тузилмаси

Хотира тузилмаси

Тизимнинг глобал соҳаси (System Global Area) (SGA)

Тизимнинг глобал соҳаси (SGA) умумий ишлатиладиган хотира соҳаси бўлиб, Oracle битта нусхаси учун берилганлар ва назорат маълумотларини сақлайди. SGA ва Oracle фон жараёнлари Oracle нусхасини ташкил қилади.

Oracle SGA-ни нусха ўз ишини бошлаганда ажратади ва нусхани ёпганда бекор қилади. Хар бир нусха ўз SGA эга.

Фойдаланувчилар Oracle серверига уланиши билан SGA берилганларига кириш ҳукуқига эга. Оптимал ишлаш учун SGA, кўпроқ берилганларни хотирада сақлаб камроқ дискга мурожаат қилиши учун, иложи борида катта бўлиши лозим.

SGA – да сақланаётган маълумотлар бир нечта турдаги хотира тузилмаларига бўлинади, жумладан берилганлар базасининг буферлари, қайд журнали буфери, умумий пул (shared pool). Бу соҳалар фиксирланган ўлчовга эга бўлиб, нусха юклатиш жараёнида очилади.

Берилганлар базасининг хеш буфери

SGA-даги берилганлар базасининг буферлари сўнги ишлатилган берилганлар базаси блокларини сақлайди; нусхадаги берилганлар базасининг буферлари тўплами – берилганлар базаси хеш буфери дейилади. Хеш буфери ўзгартирилган, ҳамда ўзгартирилмаган блокларни ўз таркибида сақлайди. Сўнги (тез-тез) ишлатилган берилганлар хотирада сақланиши сабабли дискга ёзиш-ўқиш амалларини бажариш камроқ талаб этилади ва иш тезлиги ошади.

Қайд журнали буфери

SGA қайд журнали буфери берилганлар базасида вужудга келган ўзгаришларни сақлайди. Қайд журнали буферидаги маълумотлар «online» қайд журнали файлига ёзилиб, берилганлар базасини тиклаш лозим бўлганда ишлатилади. Уларнинг ўлчови статикдир (ўзгармас).

Умумий пул (Shared Pool)

Умумий пул SGA қисми бўлиб, умумий SQL соҳаларига ўхшаган умумий хотирадан таркиб топади.

Курсорлар

Курсор – бу махсус жумлалар билан боғланган хотира номидир.

Программа глобал соҳаси (Program Global Area) (PGA)

Программа глобал соҳаси хотира буфери бўлиб сервер жараёни учун берилганлар ва назорат ахборотларни сақлайди. PGA Oracle томонидан сервер жараёни ўз ишини бошлаганда очилади. PGA – даги ахборот Oracle конфигурациясига боғлиқ бўлади.

Жараён архитектураси

Жараён – операцион тизимдаги шундай механизмки, унда қадамлар серияси бажарилиши мумкин. Айрим операцион тизимлар иш ёки масала тушунчасини ишлатади. Одатда жараён ўз шахсий хотира соҳасига эга бўлади.

Oracle сервери икки турдаги жараёнларга эга: фойдаланувчи жараёнлари ва Oracle жараёнлари.

Фойдаланувчи (мижоз) жараёнлари

Фойдаланувчи жараёнлари амалий дастурлар (масалан Pro*C/C++ дастури) ёки Oracle инструментларини (масалан Oracle Enterprise Manager) дастурий таъминот кодини бажариш учун очилади ва қўллаб-қувватланади. Фойдаланувчи жараёнлари сервер жараёнлари билан алоқаларни бошқариш учун хизмат қилади.

Фойдаланувчи жараёнлари сервер жараёнлари билан интерфейс дастури ёрдамида алоқа қилади.

Oracle жараёнинг архитектураси.

Oracle жараёнлари бошқа жараёнлар томонидан чақирилади ва чақирувчи жараён номидан унинг топшириқларини бажаради. Келгуси бўлимларда Oracle жараёнларининг ҳар хил турлари ва уларнинг махсус функциялари муҳокама қилинади. Улар сервер жараёнлари ва фон жараёнларидан таркиб топган.

Сервер жараёнлари

Oracle, сервер жараёнларни уланган фойдаланувчи жараёнлари талабларини ишлаб чиқиш учун очади. Сервер жараёни фойдаланувчи жараёнининг хабар ва суровлари билан боғлиқ бўлган Oracle билан ўзаро алоқа қилишга масъулдир. Масалан, агар фойдаланувчи SGA – нинг берилганлар базаси буферларида мавжуд бўлмаган берилганларни чакириб олишни талаб қилса, унда мос сервер жараёни керакли берилганлар блокларини берилганлар файлларидан ўқиб SGA – га кўчиради.

Фон жараёнлари

Oracle ҳар бир нусха учун фон жараёнлари мажмуасини очади.

SGA ва фон жараёнлари мажмуаси Oracle нусхасини ташкил қилади. Хар бир Oracle нусхаси бир нечта фон жараёнларини ишлатиши мумкин. Бу жараёнларнинг номлари: DBWn, LGWR, CKPT, SMON, PMON, ARCH, RECO, Dnnn, LCKn, SNPn, QMNn.

Берилганлар базасининг ёзувчиси (Database Writer) (DBWn)

Берилганлар базасининг ёзувчиси берилганлар базаси хеш буферидаги ўзгартирилган блокларни берилганлар файлларига ёзади. Кўп тизимлар учун битта берилганлар базасининг ёзувчиси жараёни (DBW0) етарли бўлишига қарамасдан, сиз ёзиш жараёнини яхшилаш учун қўшимча (DBW1 -дан DBW9 -гача) жараёнларни очиш имкониятига эгасиз. DB_WRITER_PROCESSES – инициализация параметри DBWn жараёнлар сонини аниқлайди.

Журнал ёзувчиси (Log Writer) (LGWR)

Журнал ёзувчиси қайд журналининг дискга ёзади. Қайд журналининг берилганлари SGA – нинг қайд журнали буфери ишлаб чиқилади. Транзакция тугатилиши билан ва қайд журнали буфери тўлиб кетиши билан, LGWR қайд журнали берилганларини қайд журнали актив файлига ёзади.

Текширув нуқтаси (Checkpoint) (CKPT)

Маълум вақт интервалларида SGA – даги берилганлар базасининг барча ўзгартирилган буферлари берилганлар файлларига DBWN томонидан

ёзилади; бу ходиса текширув нуктаси дейилади. Текширув нуктаси жараёни , текширув нукталарида DBWn – га сигнал бериш учун ва барча берилганлар файллари, ҳамда бошқарув файллари янгилаш учун масъулдир.

Тизимий монитор (System Monitor) (SMON)

Тизимий монитор нусха юклатилганда, уни тиклашини амалга оширади. Кўп нусхали тизимларда, битта нусханинг SMON бошқа бузилган нусхалар учун тиклаш вазифасини бажариши мумкин. SMON ишлатилмаётган вақтинчалик сегментларини ҳам тозалайди. Айрим сабабларга кўра ўтмаган танзакцияларни тиклайди.

Жараён монитори (Process Monitor) (PMON)

Жараён монитори фойдаланувчи жараёни бузилганда уни тиклашини амалга оширади. PMON жараён ишлатган хешни тозалашга ва ресурсларни бўшатишга масъулдир. PMON диспетчер ва сервер жараёнларни ҳам текширади ва агар улар бузилса қайта юклайди.

Архиватор (Archiver) (ARCH)

Архиватор “online” қайд журнали файлларида (агар улар тўлган бўлса) нусха олиб архивда сақлайди. ARCH актив ҳолда бўлади, агар берилганлар базасининг қайд журнали ARCHIVELOG ҳолатда бўлса.

Тикловчи (Recoverer) (RECO)

Тикловчи, таксимланган берилганлар базасида, тизим ёки турнинг ишдан чиқиши муносабати билан юзага келган таксимланган транзакция муаммосини ҳал қилиш учун ишлатилади.

Диспетчер (Dispatcher) (Dnnn)

Диспетчер фон жараёнлари бўлиб ҳисобланиб, кўп-жараёнли сервер (multi-threaded server) конфигурацияси учун ишлатилади.

Қулфлаш (Lock) (LCKn)

Қулфлаш жараёнлари (LCK0, . . . , LCK9) Oracle параллел серверида нусхаларо муносабат учун ишлатилади.

Иш навбати (Job Queue) (SNPn)

Тақсимланган берилганлар базасида ўттиз олтигача иш навбати жараёнлари (SNP0, ..., SNP9, SNPA, ..., SNPZ) автомат тарзда жадвалларнинг лахзавий расмини (snapshot) тўлдириши мумкин.

Навбат монитори (Queue Monitor) (QMNn)

Навбат монитори Oracle AQ опцияси хабар навбатларини назорат қилувчи фон жараёнидир. Сиз ўнтагача навбат монитори жараёнини очиш имкониятига эгасиз.

Дастур интерфейси

Дастур интерфейси – бу шундай механизмки у орқали фойдаланувчи жараёни сервер жараёни билан алоқа қилади.

Oracle ишлашига мисол

Қуйидаги мисол тур орқали уланган фойдаланувчи ва унга мос сервер жараёни ҳар хил компьютерда бўлган ҳолда Oracle созлашини намоиш этади.

1. Айни пайтда компьютерда Oracle нусхаси ишляпти.
2. Дастур ишлаётган компьютер (локал машина ёки мижоз ишчи станцияси) дастурни фойдаланувчи жараёнида ижро этади. Мижоз дастури мос Net8 драйвери ёрдамида сервер билан уланишга уриниб кўради.
3. Сервер мос Net8 драйверни ишлатади. Сервер мижоз дастури томонидан келган сўровни аниқлайди ва фойдаланувчи жараёни номидан сервер жараёнини очади.
4. Фойдаланувчи SQL жумлани бажаради ва транзакцияни «commit» қилади. Масалан, фойдаланувчи жадвал каторидаги номни ўзгартиради.
5. Сервер жараёни жумлани қабул қилиб олади ва умумий «пул»да худди шундай SQL жумласидан таркиб топган умумий SQL соҳаси мавжудлигини текширади. Агар бундай умумий SQL соҳаси топилса, сервер жараёни талаб этилган берилганларга фойдаланувчи кириш ҳуқуқини текширади ва олдинги мавжуд бўлган умумий SQL соҳаси жумлани ишлаб чиқиш учун ишлатилади. Акс ҳолда жумлани ишлаб чиқиш учун янги умумий SQL соҳаси ажратилади.

6. Сервер жараёни ихтиёрий керакли берилганлар қийматларини берилганлар файлидан ёки SGA-дан тиклаши мумкин.
7. Сервер жараёни SGA-даги қийматларни ўзгартиради. DBWn жараёни ўзгартирилган берилганларни доимо дискга ёзиб боради. Транзакция «commit» қилиниши биланоқ, LGWR жараёни тезда транзакцияни «online» қайд журнали файлига ёзади.
8. Агар транзакция муваффақиятли ўтса, сервер жараёни тур орқали мижоз дастурига хабар етказади. Акс холда, мос хатолик хабари етказилади.
9. Ушбу узун процедура билан параллел равишда фон жараёнлари ишлаб, аралаштириш лозим бўлган ишларни бажарадилар. Буларга қўшимча равишда берилганлар базасининг сервери бошқа фойдаланувчилар транзакцияларини бошқарадилар ва айнан бир хил берилганларга мурожаат қилувчи транзакцияларни тартибга солади.

4-§ Берилганлар биргаликдалиги ва конкуренцияси

Ушбу бўлимда Oracle томонидан ишлатиладиган ахборий бошқарув тизимини дастурий таъминоти механизми ёритиб берилади:

- Берилганларни ўқиш ва ўзгартириш кетма-кет тарзда амалга оширилиши лозим.
- Кўп-фойдаланувчи тизимнинг берилганларга кириш конкуренцияси максималлаштирилиши лозим.
- Максимал самарадорликка эга бўлиш учун берилганлар базаси тизимининг аксарият фойдаланувчиларидан юқори савияда ишлашни талаб этади.

Конкуренция

Кўп – фойдаланувчили берилганлар базасининг бошқарув тизимининг дастлабки муаммоси параллелик ёки бир вақтнинг ўзида айнан бир хил берилганларга кўп фойдаланувчи мурожаат қилишини назорат қилишдир.

Агар параллелик адекват тарзда назорат қилинмаса, берилганлар янгиланиши ва ўзгариши керакли тарзда амалга оширилмаслиги мумкин ва шу сабабли берилганлар яхлитлигини хавф остида қолдириши мумкин.

Ўқиш биргаликдалиги

Oracle томонидан қўллаб-қувватланадиган ўқиш кетма-кетлиги куйидагиларни амалга оширади:

- жумла томонидан кўринадиган берилганлар биргаликдалиги битта вақт нуқтасига нисбатан кафолатланади ва жумла бажарилиши давомида ўзгармайди (жумла сатхидаги ўқиш кетма-кетлиги).
- берилганлар базасидаги берилганларни ўқувчилари, айнан ўша берилганларни бошқа ўқувчилари ёки ёзувчиларини кутмаслиги кафолатланади.
- берилганлар базасидаги берилганларни ёзувчилари, айнан ўша берилганларни ўқувчиларини кутмаслиги кафолатланади.
- ёзувчилар фақат бошқа ёзувчиларни кутадилар, агар улар параллел транзакция ёрдамида айний қаторни янгилашга уринаётган бўлсалар.

Ўқиш биргаликдалиги, қайтиш(rollback) сегментлари ва транзакциялар

Фойдаланувчи берилганларни янгिलाётганда жадвалнинг асл қийматлари қайтиш сегментида ёзилади. Ушбу янгиланиш «commit» қилинмаган транзакциянинг қисми бўлиб қолгунга қадар, янгиланган берилганларга мурожаат қилувчи фойдаланувчи жадвалнинг асл қийматларини кўради.

Транзакция «commit» қилиниши билан транзакциянинг ўзгаришлари фиксирланади (ўзгармас бўлиб қолади). Фойдаланувчи транзакцияси «commit» қилингандан сўнг, жумлалар фақат «commit» қилинган транзакция томонидан амалга оширилган ўзгаришларни кўради холос.

Қулфлаш механизми

Берилганларга параллел киришни назорат қилиш учун Oracle қулфлаш механизмидан ҳам фойдаланади. Қулфлаш механизми – бу берилганларга

параллел равишда мурожаат қилувчи фойдаланувчилар орасидаги ўзаро бузувчи таъсир кўрсатиш олдини олишдир.

Қулфлаш берилганлар базасининг иккита муҳим мақсадига эришиш учун ишлатилади:

1) биргаликдалиги

Фойдаланувчи кўраётган ёки ўзгартираётган берилганлар, у ўз ишини берилганлар билан тугатгунга қадар ўзгартирмаслиги (бошқа фойдаланувчилар томонидан) кафолатланади.

2) яхлитлик

Берилганлар базасидаги берилганлар ва тузилмалар, улардаги ўзгаришларни тўғри кетма-кетликда акс этиради.

Автомат тарзда қулфлаш

Oracle қулфлаш жараёнини автоматик тарзда амалга оширади, бунда фойдаланувчининг ҳеч қандай ҳаракати талаб қилинмайди. SQL жумласи талабига мувофиқ ошкормас тарзда қулфлаш содир бўлади.

Oracle автоматик тарзда жадвал берилганларини қатор даражасида қулфлаб қўяди.

Фойдаланувчи томонидан қулфлаш

Айрим ҳолларда фойдаланувчи қулфлаш жараёнини ўзи амалга оширмоқчи бўлади. Oracle бунга (қатор даражасида ҳам, жадвал даражасида ҳам) имконият беради.

5-§ Тақсимланган жараёнлар ва тақсимланган берилганлар базалари

Мижоз/Сервер архитектураси: Тақсимланган жараёнлар

Тақсимланган жараёнлар биттадан кўп бўлган процессорларни, жараённи ўзаро боғланган топшириқлар тўпламига бўлиш учун ишлатилади.

Oracle берилганлар базаси тизими тақсимланган жараёнлар ютуқларидан, унинг Мижоз/Сервер архитектурасини ишлатиш орқали,

фойдаланади. Ушбу архитектурада берилганлар базаси тизими икки қисмга бўлинади:

Мижоз қисми ва сервер қисми.

Мижоз

Мижоз қисми берилганлар базаси дастурининг (приложение) охири бўлиб, фойдаланувчи билан клавиатура, монитор ва сичқонча орқали ўзаро мулоқот қилади.

Сервер

Сервер қисми Oracle дастурий таъминотини ишлатади ва берилганларга параллел, умумий кириш қўйилган талаб функцияларини ишлов беради. Сервер қисми мијоз дастури томонидан юборилган SQL ва PL/SQL жумлаларини қабул қилади ва бажаради.

Тақсимланган берилганлар базаси

Тақсимланган берилганлар базаси берилганлар базалари тури бўлиб кўплаб берилганлар базаси серверлари орқали бошқарилади, бунда улар фойдаланувчига ягона мантиқий берилганлар базаси тарзида кўринади. Тақсимланган берилганлар базасидаги барча берилганлар базаларининг берилганларига бир вақтнинг ўзида ҳам мурожаат қилиш ва ҳам ўзгартириш мумкин.

Тақсимланган берилганлар базасидаги берилганлар базасини бошқарувчи ҳар бир компьютерга тугун деб номланади. Фойдаланувчи бевосита уланган берилганлар базаси локал берилганлар базаси дейилади. Ушбу фойдаланувчи томонидан кирадиган ихтиёрий бошқа берилганлар базаси узоқланган берилганлар базаси дейилади.

Жойлашиш шаффофлиги (Location Transparency)

Агарда берилганларнинг жисмоний жойлашуви дастурларга ва берилганлар базаси фойдаланувчиларига бевосита куринса (шаффоф), жойлашув шаффофлиги содир бўлди дейилади. Бундай ҳолатни кўриниш, процедуралар ва синонимлар орқали амалга ошириш мумкин. Масалан, бир нечта базаларда жойлашган берилганларни кўриниш орқали шаффоф қилиш

мумкин, бунда кўринишга мурожаат қилувчи фойдаланувчи, берилганлар жисмонан қаердан олинаётгани ҳақида хабари бўлмайди.

Тугун автономлиги

Тугун автономлиги деганда тақсимланган берилганлар базасининг ҳар бир берилганлар базаси алоҳида, бошқа берилганлар базаларига боғлиқ бўлмаган равишда бошқариш мумкинлиги тушунилади.

Тақсимланган берилганларни бошқаруви

Oracle тақсимланган берилганлар базасининг архитектураси барча DML командаларини, жумладан сўров (select), янги қиймат қўйиш (insert), қийматни ўзгартириш (update), ва қийматни ўчириш (delete) амалларини қўллаб қувватлайди. Масалан узоқланган берилганлар базасидаги SALES-нинг EMP жадвалига сўров бериш учун

```
SELECT * FROM emp@sales;
```

команда орқали мурожаат қилинади.

Икки-фазавий «Commit»

Oracle тизими тақсимланган берилганлар базасида худди тақсимланмаган берилганлар базасидек икки-фазавий «Commit» механизмини ишлатади. Яъни транзакция тақсимланган берилганлар базасининг барча тугунларида яхлит бирлиқ сифатида ё «Commit» қилинади ёки «Rollback» қилинади.

Тизим ёки тур носозлигини автоматик тарзда тиклаш.

RECO фон жараёни, тақсимланган транзакцияларда тизим ёки тур носозлиги туфайли «Commit» узилиб қолса, уни автомат тарзда носозлик тузатилганда тиклайди ва мос равишда ё «Commit» ёки «Rollback» қилади .

Шубхали транзакцияни фойдаланувчи томонидан бошқариш

Oracle тизими ҳар бир локал администраторга шубхали тақсимланган транзакцияни «Commit» ёки «Rollback» қилиш имконини беради.

Тиклашга хизмат кўрсатувчи программалар

Агар берилганлар базасини ўтган вақтдаги бирор нуқтасига қайтариш лозим бўлса, Oracle-нинг тиклашга хизмат кўрсатувчи программалари берилганлар базалари администраторларига уларнинг базаларини вақтга нисбатан олдинги нуқтага қайтаришга имкон берадилар. Бу эса глобал берилганлар базаси яхлитлигини таъминлайди.

Жадвал репликацияси

Таксимланган берилганлар базаси тизимида локал фойдаланувчилар томонидан кўп сўров берадиган узокланган жадвал учун нусха олиб қўйилади.

Oracle ва Net8

Net8 — Oracle механизми бўлиб, тақсимланган жараёнлар ва тақсимланган берилганлар базаларига хизмат кўрсатувчи тур коммуникацион протоколлари орасидаги интерфейс хизматини бажаради. Коммуникацион протоколлар берилганларни тур ёрдамида жўнатиш ва қабул қилиш усулини аниқлайди. Тур муҳитида Oracle берилганлар базаси сервери мижоз ишчи станцияси ва бошқа Oracle берилганлар базалари серверлари билан Net8 деб номланувчи Oracle дастурий таъминоти орқали алоқа қилади.

6-§ Юклатиш ва ёпиш амаллари

Oracle берилганлар базасига фойдаланувчилар то Oracle сервери юклатилмагунча ва очилмагунча, кириб билмайдилар. Ушбу амаллар берилганлар базаси администратори томонидан бажарилади. Берилганлар базасини юклатиш ва уни бошқалар ишлатишга яроқли холга келтириш уч қадамда амалга оширилади:

- Oracle сервери нусхасини ишга тушириш.
- Берилганлар базасини ўрнатиш.
- Берилганлар базасини очиш.

Oracle сервери ишга туширилганда, у параметрлар қийматини ўз ичига олувчи параметрлар файлидан фойдаланади. Ушбу параметрлар берилганлар базасининг номи, жойлашув хотира миқдори, назорат файллари номлари, ва ҳар хил чеклагичлар, ҳамда тизим параметрларидан иборатдир.

Уланган нусха ва берилганлар базасини ёпиш ҳам уч кадамда бажарилади:

1. Берилганлар базасини ёпиш.
2. Берилганлар базасини узиш (Dismount the database).
3. Oracle сервер нусхасини ёпиш.

Oracle ҳамма кадамларни автоматик тарзда нусхани ёпишда бажаради.

7-§ Берилганлар базаси махфийлиги

Oracle-га ўхшаган кўп-фойдаланувчили берилганлар базаси тизими хавфсизликни таъминловчи хоссаларга эга бўлиб, берилганлар базасига кириш ва уни ишлатишни назорат қилади. Масалан, хавфсизликни таъминловчи омилларга доир мисол:

- берилганлар базасига рухсатсиз киришни таъқиқлайди
- схема объектларига рухсатсиз киришни таъқиқлайди
- диск ишлатилишини назорат қилади
- тизим ресурслари (CPU вақти каби) ишлатилишини назорат қилади
- фойдаланувчи ҳатти-ҳаракатини текширади.

Хар бир берилганлар базасининг фойдаланувчиси айнан ўша ном билан номланган схема боғлангандир. Схема бу берилганлар базаси объектларининг (жадваллар, кўринишлар, кетма-кетликлар, синонимлар, индекслар, кластерлар, процедуралар, функциялар, пакетлар ва берилганлар базаси алоқалари) мантиқий мажмуасидир.

Берилганлар базаси махфийлигини икки категорияларга булинади: тизим махфийлиги ва берилганлар махфийлиги. Тизим махфийлиги берилганлар базасига кириш ва ундан фойдаланишни тизим даражасида

назорат қилишни ўз ичига олади. Масалан, тизим махфийлиги қуйидагиларни ўз ичига олади:

- фойдаланувчи номи/пароли комбинациясини тўғрилиги;
- фойдаланувчи схемаси объектлари учун ажратилган дискдаги жой микдори;
- фойдаланувчи учун ресурс чегараси.

Тизим махфийлиги механизми қуйидагиларни текширади:

- фойдаланувчи берилганлар базага уланишга ҳуқуқи борми?
- берилганлар базасини текшириш актив ҳолатдами?
- фойдаланувчи қандай тизимий амалларни бажариши мумкин?

Берилганлар махфийлиги схема объектлари даражасида берилганлар базасига кириш ва ундан фойдаланишни назорат қилиш механизмини ўз ичига олади. Масалан берилганлар махфийлиги қуйидагиларни ўз ичига олади:

- қайси фойдаланувчилар схеманинг маълум объектларига кириш ҳуқуқига эга ва ҳар бир фойдаланувчи учун схема объектлари устида маълум турдаги амалларни бажаришга рухсат этилган (масалан фойдаланувчи SCOTT EMP жадвали устида SELECT ва INSERT жумлаларни бажариб билади, аммо DELETE жумласини бажаролмайди)
- схема ҳар бир объекти учун бажариладиган амалларни текшириш

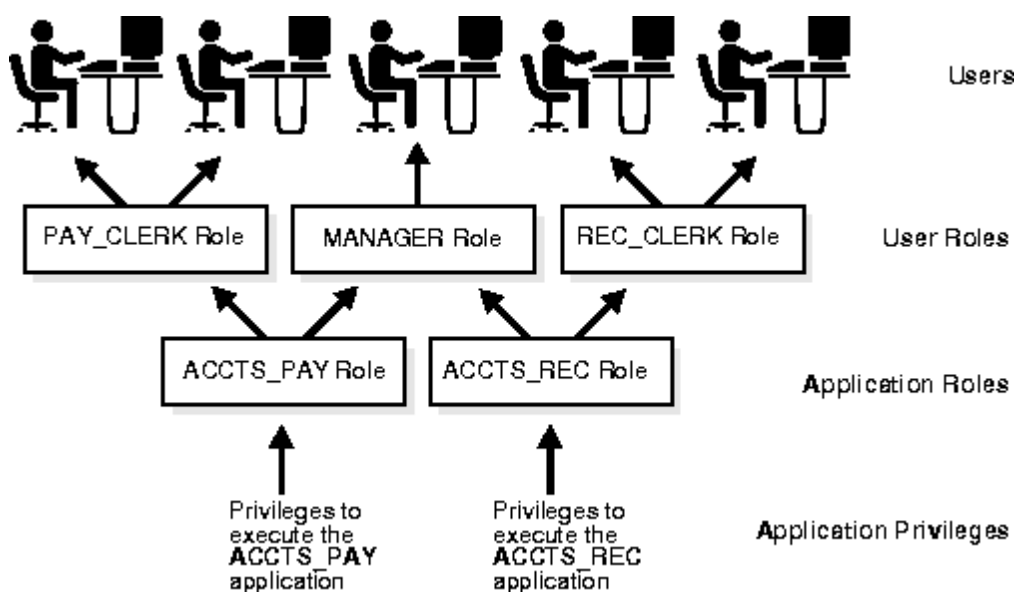
Махфийлик механизми

Oracle сервери берилганлар базаси маълумотларига киришни имтиёзлар орқали назорат қилади. Ушбу имтиёзларни мос фойдаланувчиларга олдиндан тайинлаш лозимки, улар тегишли схема объектларига киришлари мумкин бўлсин. Фойдаланувчилар ўз имтиёзларини бошқа фойдаланувчиларга ҳам узатишлари мумкин.

Oracle берилганлар базаси махфийлигини бошқариш учун ҳар хил хизмат кўрсатиш воситаларидан фойдаланади:

- берилганлар базаси фойдаланувчилари ва схемалари
- имтиёзлар
- роллар
- жойлашув параметрларини қўйиш ва квоталар
- ресурс чегаралари
- аудит қилиш

1.4-расмда Oracle махфийлигини таъминловчи воситалар орасидаги боғланишлар намоён этилган.



1.4 -расм. Oracle хавфсизлиги хусусиятлари.

Берилганлар базаси фойдаланувчилари ва схемалари

Хар бир Oracle берилганлар базаси фойдаланувчилар рўйхатига эга. Берилганлар базасига кириш учун , фойдаланувчи берилганлар базаси дастури (приложенияси) ёрдамида берилганлар базаси рўйхатида мавжуд бўлган фойдаланувчи номидан базага уланишга уринади. Хар бир фойдаланувчи номи ўз паролига эга бўлиб, уни номидан бошқа фойдаланувчилар базага киришини олдини олади.

Махфийлик домени

Хар бир фойдаланувчи махфийлик соҳасига эга бўлиб, у хоссалар мажмуасидан таркиб топган ва қуйидагиларни аниқлайди:

- фойдаланувчи бажарилиши мумкин бўлган ҳатти-ҳаракатлар (имтиёзлар ва роллар)
- фойдаланувчига бериладиган жадвалий фазодаги квоталар (фойдаланишлиши мумкин бўлган диск фазосининг қисми)
- фойдаланувчи учун тизим ресурслари чегаралари (масалан CPU иш вақти)

Хар бир хоссани батафсил ўрганиб чиқамиз.

Имтиёзлар

Имтиёз – бу SQL жумлаларини бажариш ҳуқуқидир. Мисоллар келтирамиз:

- берилганлар базасига уланиш ҳуқуқи (сессия очиш)
- ўзингизни схемангизда жадвал очиш ҳуқуқи
- бошқа бировнинг жадвалидан қаторларни сўраб олиш ҳуқуқи
- бошқа бировнинг сақланадиган процедурасини бажариш ҳуқуқи

Oracle берилганлар базасининг имтиёзларини иккита ҳар хил категорияларга бўлиш мумкин: тизимий имтиёзлар ва схема объектлари имтиёзлари

Тизимий имтиёзлар

Тизимий имтиёзлар фойдаланувчиларга тизимга тегишли баъзи бир ишларни бажаришга ёки айрим схема объектлари устида баъзи бир амалларни бажариш имконини беради. Масалан, жадвалий фазоларни очиш ёки берилганлар базасидаги ихтиёрий жадвал қаторларини ўчириш ҳуқуқлари тизимий имтиёзга мансуб бўлади. Тизимий имтиёзларнинг аксарияти администраторларга ва дастур (приложения) ишлаб чиқувчиларига берилади.

Схема объектлари учун имтиёзлар

Схема объектлари учун имтиёзлар айрим схема объектлари устида баъзи бир амалларни бажариш имконини беради. Масалан, берилганлар базасидаги айрим жадвал қаторлари устида айрим амалларни бажариш ҳуқуқи схема объектлари учун имтиёзларга мансуб бўлади. Объект имтиёзлари сўнги фойдаланувчиларга тайинланади.

Имтиёзларни тайинлаш

Имтиёзлар фойдаланувчиларга, берилганлар базасига кириб берилганлар базасидаги берилганларни ўзгартиришлари учун, тайинланади. Фойдаланувчи имтиёзни икки усулда олиши мумкин:

- Имтиёзлар фойдаланувчиларга ошкор тарзда берилиши мумкин. Масалан, EMP жадвалига қатор киритиш имтиёзини SCOTT деган фойдаланувчига ошкор тарзда бериш мумкин.
- Имтиёзларни ролларга (номланган имтиёзлар гуруҳи) тайинлаш мумкин ва сўнг ролни битта ёки ундан кўп фойдаланувчиларга тайинлаш мумкин. Масалан, EMP жадвалига қатор киритиш имтиёзини CLERK деб номланган ролга тайинлаш мумкин, уни ўз навбатида SCOTT ва BRIAN фойдаланувчиларига тайинлаш мумкин.

Роллар

Oracle тизими имтиёзларни роллар орқали содда тарзда бошқаради. Роллар номланган ўзаро боғланган имтиёзлар гуруҳи бўлиб, уларни фойдаланувчиларга ёки бошқа ролларга тайинлаш мумкин. Ролларнинг қуйидаги ҳоссалари уларни бошқаришни соддалаштиради:

- имтиёзлар тайинлаш жараёнини қисқартиради – Айнан бир хил имтиёзлар тўпламини кўп фойдаланувчиларга тайинлаш ўрнига, берилганлар базасининг администратори ўзаро боғланган фойдаланувчилар гуруҳига мўлжалланган имтиёзларни ролга тайинлаши мумкин. Сўнг берилганлар базасининг администратори ушбу ролни гуруҳнинг ҳар бир аъзосига тайинлаши мумкин.

- Имтиёзларни динамик бошқариш – Фойдаланувчилар гуруҳи имтиёзларини ўзгартирилиши лозим бўлса, фактагина рол имтиёзларини ўзгартириш кифоя бўлади. Ушбу ўзгартиришлар автоматик тарзда ўз аксини мос фойдаланувчиларда топади.
- Имтиёзларни танлаб тайинлаш – Фойдаланувчиларга берилган ролларни танланган ҳолда ишлатиш учун яроқли ёки яроқсиз ҳолга келтириш мумкин. Бу эса фойдаланувчиларга берилган имтиёзларни ихтиёрий ҳолатда назорат қилишга имкон беради
- Дастурлар (приложениялар) учун имтиёзлар – Берилганлар базаси дастурини шундай тузиш мумкинки, фойдаланувчи дастурни ишлатишга ўринганда танланган роллар автоматик тарзда яроқли ёки яроқсиз ҳолга келтирилади.

Берилганлар базасининг администратори (DBA) кўпинча берилганлар базаси дастурлари учун роллар очади. DBA дастур ролига дастурни ишга тушириш учун зарур бўладиган барча имтиёзларни тайинлайди. Сўнг DBA дастур ролини бошқа ролларга ёки фойдаланувчиларга тайинлаши мумкин. Дастур ҳар бири алоҳида имтиёзлар гуруҳидан таркиб топган бир нечта ҳар хал ролларга эга бўлиши мумкин.

DBA ролдаги имтиёзларни рухсатсиз ишлатмаслик мақсадида паролга эга бўлган ролни ҳам очиши мумкин. Одатда дастурлар шундай тузиладики, уларни юклатиш пайтида барча лозим бўладиган роллар яроқли ҳолга келтирилади. Натижада дастур фойдаланувчиси дастур ролининг паролини билиши лозим бўлмайди.

Жой ва квоталарни тайинлаш

Oracle берилганлар базаси учун ажратилган дискдаги жойни, жумладан фойдаланувчи ишчи жадвалий фазосини, вақтинчалик жадвалий фазолар ва жадвалий фазолар квоталарини бошқарув воситаларига эга.

Ишчи жадвалий фазо

Ҳар бир фойдаланувчи ишчи жадвалий фазо билан боғланган. Фойдаланувчи жадвал, индекс ёки кластер очмоқчи бўлиб, унда уларни

жисмонан сақлаш учун жадвалий фазо кўрсатилмаган бўлса, шу мақсадда ишчи жадвалий фазо ишлатилади, агарда фойдаланувчи схема объектларини очиш имтиёзига эга ва ишчи жадвалий фазода квотаси мавжуд бўлса.

Вақтинчалик жадвалий фазо

Хар бир фойдаланувчи вақтинчалик жадвалий фазога эга. Агар фойдаланувчи, вақтинчалик сегментларни очишни талаб қиладиган SQL жумласини бажарса (масалан, индекс очиш), фойдаланувчининг вақтинчалик жадвалий фазоси ишлатилади.

Жадвалий фазо квоталари

Oracle схема объектлари ишлатадиган диск фазосини чеклаб қўйиши мумкин. Квоталар (фазо чеглагичи) фойдаланувчига ҳар бир жадвалий фазо учун алохида-алохида берилиши мумкин.

Профиллар ва ресурс чеклагичлари

Хар бир фойдаланувчига профил тайинланади. Профил фойдаланувчига мансуб бўлган тизим ресурсларини чеклаб қўяди, жумладан

- фойдаланувчи ўрнатилиши мумкин бўлган параллел сессиялар сони
- CPU ишлаш вақти
- фойдаланувчи сессияларига кириши имконияти
- мантиқий киритиш-чиқариш миқдори
- фойдаланувчи сессиясига рухсат этилган уланиш вақти
- парол чеклагичи
- муваффақиятсиз уланишлар сонини санаш

Хар хили профиллар очиб, берилганлар базасининг фойдаланувчиларига индивидуал тарзда тайинлаш мумкин. Агар профил фойдаланувчига тайинланмаса, унда Oracle тизими унга умумий профилни тайинлайди.

Аудит қилиш

Oracle тизими танланган текширувни амалга оширишга имкон беради. Текширувни уч хил даражада амалга ошириш мумкин: жумла даражасида, имтиёз даражасида, схема объекти даражасида.

Жумла даражасидаги текширув

Жумла даражасидаги текширув, схема объектларини аниқ кўрсатилмаган ҳолда SQL жумлаларни назорат қилади. Масалан, SCOTT ва LORI фойдаланувчилар берилганлар базасига уланиш ва узишини назорат остига олиш мумкин.

Имтиёз даражасида текширув

Имтиёз даражасида текширув – бу схема объектларини аниқ кўрсатилмаган ҳолда тизимий имтиёзларни назорат қилишдир. Имтиёз даражасида текширувда барча фойдаланувчиларни ёки айрим фойдаланувчиларни назорат остига олиш мумкин.

Схема объектлари даражасидаги текширув

Схема объектлари даражасидаги текширув –бу фойдаланувчига боғлиқ бўлмаган равишда схема айрим объектларига киришни назорат қилишдир. Масалан берилган жадвал учун SELECT ва DELETE жумлалар бажарилишини назорат қилиш. Назорат натижалари назорат жадвалига ёзиб борилади ва уни ўқишнинг имконияти бор.

8-§ Берилганлар базасини тиклаш

Ушбу бўлимда Oracle томонидан фойдаланадиган:

- ҳар хил сабабларга кўра бузилган берилганлар базасини тиклаш;
- ихтиёрий ҳолатда тиклашни таъминлаш;
- тиклаш жараёнида фойдаланувчиларнинг берилганлар билан ишлаш имконияти мавжудлигини таъминлаш

Тиклашнинг зарурийлиги

Хар қандай берилганлар базаси тизимида аппарат воситалари ёки тизимнинг бузилиш эҳтимоллиги мавжуд. Тизим ёки аппарат воситалари

бузилса уни тиклаш зарурияти туғилади. Бузилишдан сўнг энг асосий мақсад, барча «commit» қилинган транзакциялар ўз аксини тикланган берилганлар базасида топиши лозимлиги ва имкон қадар тезроқ нормал ҳолатга қайтишдир.

Бузилиш турлари

Кўп ҳолатлар Oracle берилганлар базасининг бузилишига олиб келиши мумкин. Айрим ҳолатларини келтирамиз:

Фойдаланувчи хатолиги

Фойдаланувчи хатолиги берилганлар базасини хатолик содир бўлишидан олдинги вақтга қайтаришни талаб қилиши мумкин. Масалан, фойдаланувчи тасодифан жадвални йўқотиши мумкин.

Жумла ва жараённинг бузилиши

Oracle дастурида жумлани таҳлил қилиб ишлов беришда мантиқий бузилиш вужудга келса, жумла бузилиши юз беради (масалан, жумла нотўғри SQL конструкциясидан ташкил топган). Жумла бузилиши вужудга келганда Oracle тизими автомат тарзда жумла таъсирини қайтариб назоратни фойдаланувчига қайтаради.

Фойдаланувчи жараёни Oracle тизимига киришда (масалан, нормал бўлмаган узилиш) бузилиш вужудга келса – жараён бузилиши юз берди дейилади. Бузилган фойдаланувчи жараёни ишни давом этиролмайди, лекин Oracle ва бошқа фойдаланувчи жараёнлари ўз ишини давом этиради. Oracle PMON фон жараёни бузилган фойдаланувчи жараёнида «commit» қилинмаган транзакцияларни «rollback» қилиб, фойдаланувчи жараёни ишлатилаётган барча ресурсларни бўшатиб юборади.

Нусха бузилиши

Нусха ўз ишини давом этиришга монетьлик билдирадиган муаммолар пайдо бўлса, нусха бузилиши вужудга келди дейилади. Нусха бузилиши юз берганда SGA буферидаги берилганлар, берилганлар файллариға ёзилмайди.

Нусха бузилиши нусхани тиклашни талаб қилади. Нусха юклатилаётганда нусха тикланиши Oracle томонидан автомат тарзда бажарилади.

Диск бузилиши

Файлни ёзиш ёки уни ўқишда вужудга келадиган хатолик диск бузилиши дейилади.

Тиклаш учун ишлатиладиган тузилмалар

Нусха ёки диск бузилганда тўлиқ тиклашни амалга ошириш учун Oracle тизими бир нечта тузилмалардан фойдаланади: қайд журнали, «rollback» сегменти, назорат файли, ва берилганлар базасини зарурий лахзагача орқага қайтариш.

Қайд журнали

Олдин тавсифлаганимиздек қайд журнали файллар тўплами бўлиб, хали берилганлар файлига ёзиб улгурилмаган, хотирадаги ўзгартирилган берилганлар базасидаги берилганларини сақлаш учун ишлатилади. Қайд журнали икки қисмдан иборат бўлиши мумкин: ишчи қайд журнали ва архивланган қайд журнали.

Ишчи қайд журнали

Ишчи қайд журнали икки ёки ундан кўп ишчи қайд журнали файллари тўпамидан иборат бўлиб, берилганлар базасидаги барча «commit» қилинган ўзгартиришларни ёзади. Бу амал LGWR фон жараёни орқали бажарилади.

Берилганлар базасини йўқотмаслик мақсадида Oracle тизими кўп каррали ишчи қайд журнали файллари тўпламига эга бўлиши мумкин. Бунда кўп каррали ишчи қайд журнали, жисмонан ҳар хил дискда жойлашган ишчи қайд журнали файллари нусхаларидан таркиб топган.

Архивланган қайд журнали

Тўлган ишчи қайд журналини қайтадан ишлатиш учун, уни архивлаш мумкин. Архивланган қайд журнали файллари архив қайд журналини сақлайди.

«ARCHIVELOG» вазият

Тўлиб кетган ишчи қайд журнали файлларини даврий тарзда қайтадан фойдаланиш учун, олдин улар архивланади.

«NOARCHIVELOG» вазият

Тўлиб кетган ишчи қайд журнали файллари архивланмайди.

«ARCHIVELOG» вазиятида берилганлар базасини тўлиқ тиклаш мумкин. Агарда «NOARCHIVELOG» вазиятида нусха бузилса берилганлар базасини тўлиқ тиклаш мумкин, аммо диск бузилиши вужудга келса базани тиклаб бўлмайди.

Назорат файллари

Назорат файллари, бошқа маълумотлар билан бирга берилганлар базаси тузилмаси файли ва LGWR томонидан ёзиладиган жорий қайд кетма-кетлик тартибини ҳам сақлайди. Нормал тиклаш жараёнида назорат файлидаги маълумотлар тиклаш жараёнини автоматлаштириш учун ишлатилади.

9-§ Берилганлар базасини бошқариш учун объектли-реляцион модел

Oracle ўз имкониятларини реляцион моделдан объектли-реляцион моделгача кенгайтди. Бунда реляцион берилганлар базасида комплекс бизнес моделни сақлаш имконияти пайдо бўлди.

Реляцион модел

Реляцион модел уч аспектдан иборат:

Тузилмалар

Тузилмалар – бу жадваллар, кўринишлар, индекслар ва шу каби объектлар бўлиб берилганлар базасидаги берилганларни сақлаш ёки уларга кириш учун ишлатилади. Тузилмалар ва улардаги берилганлар амаллар орқали бошқарилиши мумкин.

Амаллар

Амаллар – бу содда тарзда аниқланган ҳаракат бўлиб, улар ердамида фойдаланувчилар берилганлар базасидаги тузилмалар ва берилганларни

бошқарадилар. Амаллар олдиндан аниқланган қатъий яхлитлик қоидалар тўпламига бўйсунилари лозим.

Яхлитлик қоидалари

Яхлитлик қоидалари – бу қонун бўлиб, берилганлар базасидаги берилганлар ва тузилмалар устида бажарилиши мумкин бўлган амалларни аниқлайди. Яхлитлик қоидалари берилганлар базасидаги тузилмалар ва берилганларни ҳимоялайди.

Реляцион берилганлар базасини бошқарув тизимининг қатор ижобий хоссалари мавжуд:

- берилганларни жисмонан сақланиши ва берилганлар базасининг мантикий тузилмаси орасидаги боғлиқсизлик
- барча берилганларга ўзгарувчан ва озод тарзда кириш имконияти
- берилганлар базаси лойиҳасини кўришда тўлиқ эркинлиги
- берилганларни қисқартириб сақланиши

Объект-реляцион модель

Объект-реляцион модель - фойдаланувчиларга объект турини аниқлаш ва берилганлар тузилмаси ҳамда берилганларга ишлов бериш усулини аниқлашга имкон беради.

Объект тури борлиқнинг ҳақиқий объектининг абстракциясидир. Объект тури учта ясовчисидан иборат:

- номи
- атрибутлари
- методлари (берилганлар базасида сақланадиган PL/SQL тилида ёзилган функция ёки процедуралар, ёки С тилига ўхшаган тилда ёзилиб базадан ташқарида сақланадиган)

Схемалар ва схема объектлари

Схема – фойдаланувчи кириши имконияти бўлган берилганлар базаси объектларининг мажмуасидир. Схема объектлари берилганлар базасидаги берилганларга бевосита алоқаси бўлган мантикий тузилмалардир. Схема

объектларига жадваллар, кўринишлар, кетма-кетликлар, сақланадиган процедуралар, синонимлар, индекслар, кластерлар, ва берилганлар базаси боғланишлари мансуб бўлади. (Жадвалий фазо ва схема орасида боғланиш муносабати мавжуд эмас; битта схема объектлари ҳар хил жадвалий фазоларда ётиши мумкин).

Жадваллар

Жадвал Oracle берилганлар базасидаги берилганларни сақловчи асосий бирликдир. Жадвал берилганлари қаторлар ва устунларда сақланади. Ҳар бир жадвал – жадвалнинг номи ва устунлар тўплами орқали аниқланади. Ҳар бир устун – устун номи, берилганлар тури (масалан CHAR, DATE, ёки NUMBER), ва узунлиги ёки масштаб ва аниқликдан иборат. Жадвал яратилиши билан уни тўғри қаторлар билан тулдириш мумкин. Сўнг жадвал қаторлари қийматларини ўзгартириш, йўқотиш ва сўраб олиш мумкин. Жадвал берилганларини маълум бизнес қоидаларига бўйсундириш учун жадвал учун яхлитлик чеклагичлари ва триггерларни аниқлаш мумкин.

Кўринишлар

Кўриниш – жадвалларга асосланган сақланадиган сўров деб тасаввур қилиш мумкин. Кўринишлар берилганларни сақламайди, балки у асосланган жадваллардан олади. Асосий жадваллар ўз навбатида жадваллар ёки кўринишлар бўлиши мумкин. Жадваллардагидек кўринишларга сўров бериш, қийматларини ўзгартириш, янги қиймат киритиш ва қийматларни йўқотиш (баъзи чекланишлар билан). Барча амаллар ҳақиқатда кўриниш асосланган жадваллар устида бажарилади.

Кетма-кетликлар

Кетма-кетликлар жадвал устунларини тўлдириш учун уникал бўлган сонлар кетма-кетлигини ишлаб чиқаради.

Дастур бирликлари

Дастур бирликлари термини ушбу қўлланмада сақланадиган процедуралар, функциялар, пакетлар, триггерлар ва номсиз блоklar учун ишлатилади. Процедура ёки функция SQL ва PL/SQL жумлалар тўпламидан

иборат бўлиб муайян вазифани бажарувчи бажариладиган бирликлардир. Процедура ва функцияларни берилганлар базасида сақлаб, исталган вақтда бажариш мумкин. Функция ҳамма вақт чақирувчига битта қиймат қайтаради. Бошқа томонларидан функция ва процедура айнан бир хил бирликлардир.

Пакет эса ўзаро боғланган процедуралар, функциялар ва бошқа пакетларни берилганлар базасида ягона бирлик сифатида сақлаш учун ишлатилади.

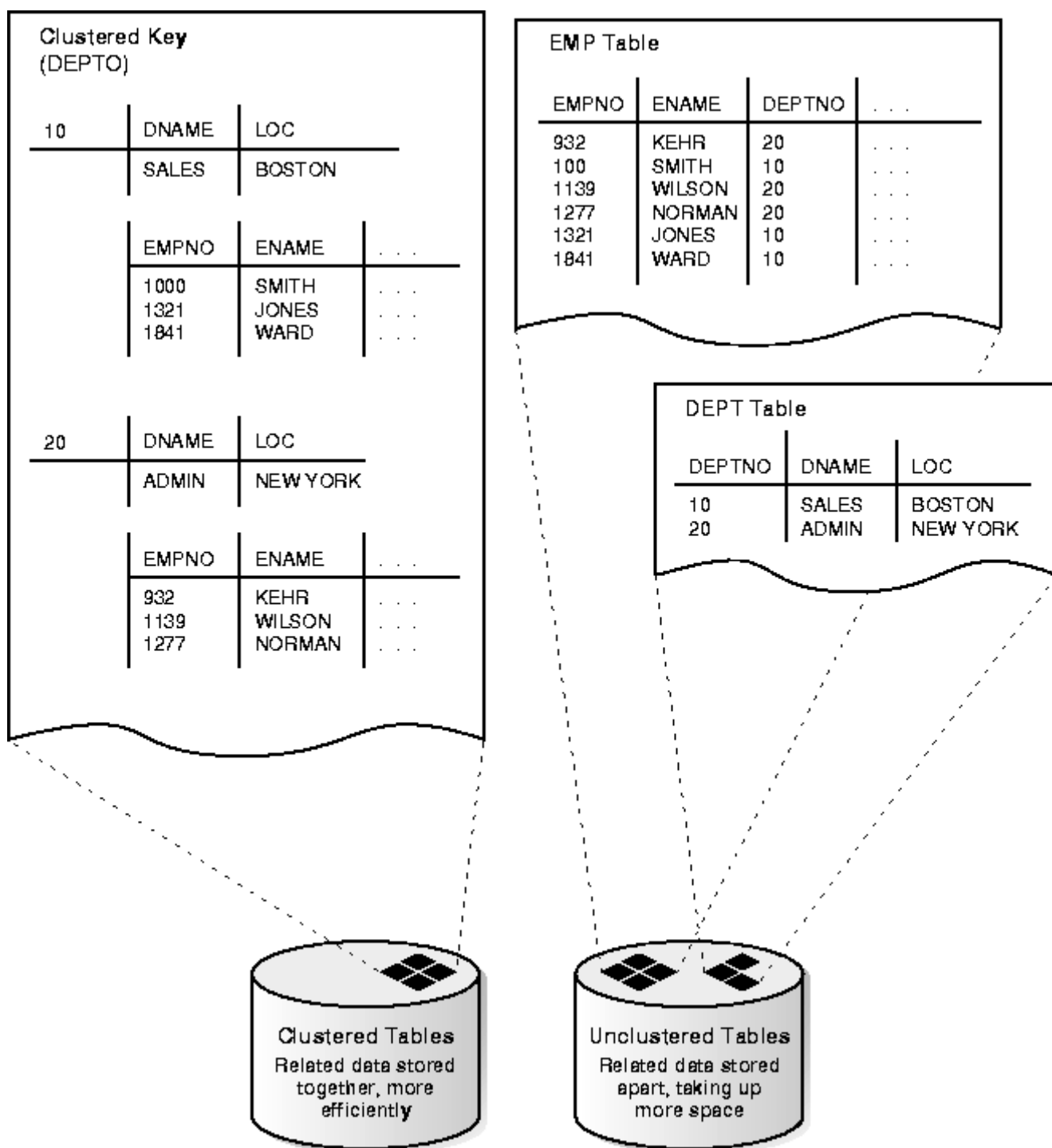
Синонимлар

Синонимлар – жадвал, кўриниш, кетма-кетлик ёки бошқа дастур бирлиги учун ҳамроҳ бўлиб ҳисобланади. Синоним ҳақиқатда схема объекти эмас балки унга ҳаволадир.

Индекслар, Кластерлар ва Хеш кластерлари

Индекслар, Кластерлар ва Хеш кластерлари жадвалларга боғланган тузилмалар бўлиб берилганларни саралаб олишини тезлаштириш учун ишлатилади. Индекслар жадвалнинг битта ёки ундан кўп устунлари учун очилади. Кластерлар жадвал берилганларини сақлаш усули бўлиб жисмонан бирга сақланадиган битта ёки ундан кўп жадваллар гуруҳидир. Жадвалларнинг боғланган устунлари кластернинг калити деб номланади ва индексланади.

1.5-расмда кластерланган ва кластерланмаган берилганларни жисмонан сақланиши намоён қилинган.



1.5- расм. Кластерланган ва кластерланмаган жадваллар.

Хеш кластер нормал кластерга ўхшаб жадвал берилганларини гурухлайди. Аммо хеш кластерда сақланадиган қатор кластер калити қиймат

қаторларига хеш функцияни қўллаш натижасига асосланган. Бир хил калит қийматига эга бўлган қаторлар дискда бирга сақланади.

Берилганлар базаси боғланишлари

Берилганлар базаси боғланиши номланган схема объекти бўлиб битта берилганлар базасидан иккинчи берилганлар базасигача бўлган йўлни тавсифлайди.

Берилганлар луғати

Хар бир Oracle берилганлар базаси ўзининг берилганлар луғатига эга. Берилганлар луғати фақат ўқиш учун мулжалланган жадваллар ва кўринишлар тўпламидан иборат бўлиб берилганлар базаси ҳақидаги маълумотларни сақлайди. Масалан, берилганлар луғати берилганлар базасининг мантиқий ва жисмоний тузилмаси ҳақидаги маълумотларни сақлайди. Бундан ташқари берилганлар луғати:

- Oracle берилганлар базаси фойдаланувчилари
- берилганлар базасидаги жадваллар учун аниқланган яхлитлик чеклагичлари ҳақидаги маълумотлар
- схема объектлари учун ажратилган жой ва унинг ишлатилган қисми

10-§ Берилганларга кириш

Берилганларга кириш имконияти:

- берилганларга кириш тили саноат стандартларига қатъий риоя қилинади
- берилганлар базасидаги берилганларини бошқаришда маълумотлар кетма-кетлигини назорат қилиб, унга риоя қилинади.
- берилганлар базасидаги маълумотлар яхлитлигини таъминлаш учун қоидаларни аниқлаш ва бошқариш тизимини хавола этади.
- юқори самарадорликни таъминлайди.

SQL – Тузилмаланган сўровлар тили (The Structured Query Language)

SQL - содда, қувватли, берилганлар базаси билан ишловчи тил бўлиб, реляцион берилганлар базасини бошқарувчи тизим учун стандарт тил бўлиб хизмат қилади.

SQL жумлалари

Oracle берилганлар базаси маълумотлари устида барча амаллар SQL жумлалари орқали бажарилади. SQL жумласи Oracle бажарилиши лозим бўлган SQL матни қаторидан иборат. Масалан:

```
SELECT ename, deptno FROM emp;
```

Фақатгина тўлиқ SQL жумлалари бажарилади холос. Чала жумлалар бажарилмайди:

SELECT ename

SQL жумласини жуда содда, лекин кучли дастур тарзида тасаввур қилиш мумкин. SQL жумлаларини қуйидаги категорияларга бўлиш мумкин:

- берилганларни аниқлаш тили (DDL) жумлалари
- берилганларни бошқариш тили (DML) жумлалари
- транзакцияни назорат қилувчи жумлалари
- сессияни назорат қилувчи жумлалари
- тизимни назорат қилувчи жумлалари
- ички қўрилган SQL жумлалари

Берилганларни аниқлаш тили (DDL) (Data Definition Statements)

DDL жумлалари схема объектларини аниқлаш, сақлаш ва керак бўлмаганда йўқотиш учун ишлатилади. DDL жумлалари фойдаланувчи бошқа фойдаланувчиларга берилганларга кириш имтиёзларини бериш учун ҳам ишлатилади.

Берилганларни бошқариш тили (DML)

DML жумлалари берилганлар базасидаги берилганларни бошқаради. Масалан, сўров бериш, янги қиймат қўйиш, қийматни ўзгартириш, ва жадвал қаторларини ўчириш DML амалларига мисол бўлаолади.

Транзакцияни назорат қилувчи жумлалар

Транзакцияни назорат қилувчи жумлалар DML жумлалари томонидан бажарилган ўзгаришларни бошқаради. Улар фойдаланувчи ёки дастурчиларга ўзгаришларни мантиқий транзакцияларга гурухлашга имкон беради. COMMIT, ROLLBACK, ва SAVEPOINT шуларга мисол бўлаолади..

Сессияни назорат қилувчи жумлалари

Сессияни назорат қилувчи жумлалари фойдаланувчига унинг оний сессия хоссаларини, жумладан ролни актив ва пассив ҳолатга келтиришни, бошқаради. ALTER SESSION ва SET ROLE буларга мисол бўлаолади.

Тизимни назорат қилувчи жумлалари

Тизимни назорат қилувчи жумлалари Oracle сервери нусхаси хоссаларини ўзгартиради. Тизимни назорат қилувчи ягона командаси

ALTER SYSTEM мавжуд; у сизга умумий ишлатилиши мумкин бўлган серверларнинг минимал сонини ўрнатиш, сессияни тугатиш ва шу каби бошқа вазифаларни бажаришга имкон беради.

Ички қурилган SQL жумлалари

Ички қурилган SQL жумлалари DDL, DML ва транзакцияни назорат қилувчи жумлаларини процедуравий тилдаги дастурларда тасвирлаш учун ишлатилади. OPEN, CLOSE, FETCH, ва EXECUTE буларга мисол бўлаолади.

Транзакциялар

Транзакция – ягона фойдаланувчи томонидан бажариладиган битта ёки ундан кўп SQL жумлаларидан иборат ишнинг мантиқий бирлигидир. ANSI/ISO SQL стандартига асосан транзакция фойдаланувчининг биринчи бажариладиган SQL жумласидан бошланади. Транзакция тугатилган деб ҳисобланади, агар ўша фойдаланувчи томонидан у ошкор тарзда “commit” ёки ‘rollback’ қилинса.

Транзакцияларни «Commit» ва «RollBack» қилиш

Транзакцияни ташкил қилувчи SQL жумлалари томонидан амалга оширилган ўзгартиришларни бажариш ёки қайтариш мумкин. Транзакция бажарилиши ёки қайтарилиши билан навбатдаги транзакция навбатдаги SQL жумласидан бошланади.

Транзакцияни «Commit» қилиш транзакциядаги барча SQL жумлалари томонидан амалга оширилган ўзгаришларни фиксирлайди ва бу ҳолат транзакция 'commit' қилингандан сўнг очилган бошқа фойдаланувчилар сессияларига кўринади.

Транзакцияни 'rollback' қилингандан сўнг транзакциядаги барча SQL жумлалари томонидан амалга оширилган ўзгаришлар инкор қилинади ва берилганлар асл ҳолатига қайтади (худди транзакциядаги SQL жумлалари ҳеч қачон бажарилмаганидек).

Оралик меткалари (Savepoints)

Жуда кўп SQL жумлаларидан иборат узун транзакциялар учун оралик меткаларни эълон қилиш мумкин. Оралик меткалар узун транзакцияларни кичик қисмларга бўлиш учун фойдаланиш мумкин.

Оралик метка ёрдамида сиз узун транзакциядаги ишингизни ихтиёрий жойини белгилаб қўйишингиз мумкин. Бу эса келгусида оний нуқтадан белгиланган нуқтагача 'rollback' қилиш имконига эга бўлишингизни таъминлайди.

PL/SQL

PL/SQL Oracle-нинг процедуравий тили бўлиб SQL тилининг кенгайтмаси ҳисобланади. PL/SQL тили SQL тилининг содда ва қулайлиги томонлари билан бирга IF ... THEN, WHILE, ва LOOP каби тузилмаланган дастурлаш тили процедуравий имкониятларини ўзида мужассамлайди.

Берилганлар базаси дастурини яратишда, ишлаб чикувчи сақланадиган PL/SQL қатор ижобий ютуқларини инобатга олиши лозим:

- PL/SQL коди марказлашган ҳолда берилганлар базасида сақланиши мумкинлиги туфайли берилганлар базаси ва дастур орасидаги тур трафикини қисқартириш мумкин
- Берилганларга киришни сақланадиган PL/SQL коди орқали назорат қилиш мумкин. Бу ҳолда фойдаланувчилар берилганларга фақат дастур рухсат этилган берилганларга кириш имконига эга бўладилар ва бошқа йўл билан берилганларга кириб бўлмаслиги кафолатланади
- Тур трафикини оширмасдан амаллар мажмуасини бажарувчи PL/SQL блокини дастур ёрдамида берилганлар базасига юбориш мумкин.

Хатто PL/SQL коди берилганлар базасида сақланмаса ҳам, дастурлар индивидуал SQL жумлалар ўрнига PL/SQL блокини берилганлар базасига юбориш мумкин. Бу эса ўз навбатида тур трафикини қисқартиради.

Процедуралар ва функциялар

Процедуралар ва функциялар – муайян масалани ҳал қилувчи ягона бирлик тарзда гуруҳланган SQL ва PL/SQL жумлаларидан иборат. Процедура компиляция қилинган ҳолда берилганлар базасида сақланади ва фойдаланувчи ёки дастур томонидан бажарилиши мумкин. Функциялар доимо чақирувчига ягона қиймат қайтарилиши лозим, процедуралар эса чақирувчига қиймат қайтармайди. Бошқа ҳолларда улар айнан бир хилдирлар.

Пакетлар

Пакетлар ўзаро боғланган процедуралар, функциялар, ўзгарувчилар ва бошқа пакет тузилмаларини берилганлар базасида ягона бирлик сифатида сақлаш учун ишлатилади.

Берилганлар базаси триггерлари

Oracle тизими сизга олдин процедурани ёзиб сўнг уни автоматик тарзда жадвалда янги қиймат қўйиш, қийматни ўзгартириш ва ўчириш натижаси сифатида бажаришга имкон беради. Бундай процедуралар берилганлар базаси триггерлари деб номланади.

Методлар

Метод процедура ёки функция бўлиб фойдаланувчи аниқланган берилганлар турининг (объект тури, ичма-ич жойлашган жадвал, ёки ўзгарувчи массив) аниқланиш қисмидир.

Методлар сақланадиган процедуралардан икки жиҳатдан фарқ қилишади:

- Сиз методни у билан боғлиқ бўлган турдаги объектга мурожаат қилиш орқали чақиришингиз мумкин.
- Метод, у билан боғланган объект атрибутларига ва уларнинг тури ҳақидаги маълумотга тўлиқ кириш имкониятига эга.

Берилганлар яхлитлиги

Берилганлар - дастур ишлаб чиқувчи ёки берилганлар базаси администратори томонидан аниқланган иш юритиш қоидаларга бўйсунилишни кафолатлаш жуда муҳимдир. Берилганлар базасидаги берилганлар яхлитлиги қоидаларини бошқариш ечими сифатида Oracle яхлитлик чеклагичлари ва берилганлар базаси триггерларини тавсия этади.

Яхлитлик чеклагичлари

Яхлитлик чеклагичлари — жадвал устунлари учун бизнес қоида аниқловчи декларатив усулдир. Яхлитлик чеклагичи жадвал берилганлари учун ҳамма вақт рост бўлган жумладир:

- Агарда яхлитлик чеклагичи бирор жадвал учун очилган бўлиб ва жадвалдаги мавжуд муайян берилганлар чеклагични қаноатлантирмаса, чеклагични олдиндан ишлатиб бўлмайди.
- Чеклагич аниқланиши билан, агарда бирорта DML жумлаларни натижаси яхлитлик чеклагични бузса, жумла 'rollback' қилинади ва хатолик қайтарилади.

Яхлитлик чеклагичлари жадвал билан бирга аниқланади ва жадвал аниқланиши қисми сифатида марказлашган ҳолда берилганлар луғатида сақланади.

Қуйидаги яхлитлик чеклагичлари Oracle томонидан қўллаб қувватланади:

NOT NULL

Жадвал устунини null (бўш қиймат) бўлишига йўл қўймайди.

UNIQUE

Устун ёки устунлар тўпламида қиймат такрорланишига йўл қўймайди.

PRIMARY KEY

Устун ёки устунлар тўпламида қиймат такрорланишига ва null бўлишига йўл қўймайди.

FOREIGN KEY

Устун ёки устунлар тўплами қиймати боғланган жадвалнинг UNIQUE ёки PRIMARY KEY қийматларига мос келишини талаб қилади.

CHECK

Чеклагичдаги мантикий ифодани қаноатлантирмайдиган қийматларга йўл қўймайди.

11-§ Oracle берилганлар базасининг администратори

Ушбу бўлимда Oracle серверини бошқарувчи шахс-берилганлар базаси администраторининг вазифалари ёритилади.

Oracle фойдаланувчиларининг турлари

Oracle фойдаланувчиларини қуйидаги турларга бўлиш мумкин.

- Берилганлар базасининг администраторлари
- Хавфсизлик администраторлари
- Дастурларни ишлаб чиқувчилар
- Дастурлар администраторлари
- Берилганлар базасининг фойдаланувчилари
- Тармоқ администраторлари

Берилганлар базасининг администраторлари

Oracle берилганлар базаси тизими жуда катта бўлиб, кўп фойдаланувчиларга эга бўлиши мумкин, шу сабабли бир ёки бир нечта одамлар гуруҳи ушбу тизимни бошқариши лозим. Берилганлар базасининг администратори (DBA) ушбу бошқарувчидир. Ҳар бир берилганлар базаси, маъмурий вазифаларни бажариш учун камида бир кишини талаб этади.

Берилганлар базаси администраторининг вазифалари қўйидагилардан иборат бўлиши мумкин:

- Oracle сервери ва дастурий ускуналарни ўрнатиш ва янгилаш
- Берилганлар базаси тизими талаб этадиган фазони жойлаштириш ва келгуси ривожига талабларини режалаштириш
- Дастур ишлаб чиқувчилар ўзларининг дастурларини яратганларидан сўнг, берилганлар базасининг дастлабки сақлагич тузилмаларини (жадвалий фазолар) ташкил қилиш
- Дастур ишлаб чиқувчилар ўзларининг дастурларини яратганларидан сўнг, берилганлар базасининг дастлабки объектларини (жадваллар, кўринишлар, индекслар) яратиш
- Дастур ишлаб чиқувчиларнинг маълумотларига таянган ҳолда зарурият пайдо бўлганда берилганлар базаси тузилмаларини янгилаш
- Фойдаланувчиларга ролларни тайинлаш ва тизим махфийлигини таъминлаш
- Oracle лицензияси шартларини кафолатлаш
- Фойдаланувчиларнинг берилганлар базасига кириш имкониятларини назорат қилиш
- Берилганлар базасида бажариладиган амалларни кузатиш ва оптималлаштириш
- Берилганлар базасидаги маълумотларни эски ҳолатига қайтариш ва тиклашни режалаштириш
- Архивланган маълумотларни алоҳида сақлагичларда сақлаш
- Берилганлар базаси ҳолатини орқага қайтариш ва қайта тиклаш
- Техникавий қўллаб қувватлаш учун Oracle корпорацияси билан алоқада бўлиш

Хавфсизликни таъминловчи администраторлар

Баъзи ҳолларда, берилганлар базаси хавфсизлигини таъминловчи администраторларга эга бўлиши мумкин. *Берилганлар базаси хавфсизлигини таъминловчи администраторлар* фойдаланувчиларга ролларни тайинлаш, фойдаланувчиларнинг берилганлар базасига кириш имкониятларини доимий назорат қилиб бориш ва бошқариш, ҳамда тизим сирлигини таъминлашга масъулдир.

Дастур ишлаб чиқувчилар

Дастур ишлаб чиқувчилар берилганлар базаси билан ишлашга мўлжалланган дастурларни ишлаб чиқадилар ва уларни жорий этадилар. Дастур ишлаб чиқувчиларнинг вазифалари қўйидагилардан иборат:

- Берилганлар базаси билан ишлашга мўлжалланган дастурларни ишлаб чиқиш ва дизайн бериш
- Дастур учун зарур бўладиган берилганлар базаси тузилмасини чизиш
- Дастурни жорий этиш учун сақлаш жойига қўйиладиган талабларни баҳолаш
- Дастур учун зарур бўладиган берилганлар базаси тузилмалари ўзгаришини аниқлаб бериш
- Юқоридаги маълумотларни берилганлар базаси администраторига етказиш
- Ишлаб чиқиш жараёнида дастурни ишга туширмаслик
- Ишлаб чиқиш жараёнида дастур сирлиги ўлчовларини ўрнатиш

Дастур администраторлари

Oracle тизими битта ёки ундан кўп дастур администраторларига эга бўлиши мумкин. *Дастур администратори* махсус дастурларни бошқариш учун масъул шахсдир.

Берилганлар базасининг фойдаланувчилари

Берилганлар базасининг фойдаланувчилари берилганлар базаси билан дастурлар ёки утилиталар ёрдамида ўзаро мулоқот қиладилар. Фойдаланувчиларнинг намунавий вакилининг вазифалари қўйидагилардан иборат бўлиши мумкин:

- Берилганларни берилганлар базасига киритиш, ўзгартириш ва ўчириш
- Берилганлар базасидаги маълумотлардан ҳар хил ҳисоботлар олиш

Тармоқ администраторлари

Битта ёки ундан кўп тармоқ администраторлари мавжуд бўлиши мумкин. Тармоқ администраторлари Oracle-нинг Net8 каби тармоқ билан ишлайдиган маҳсулотларини бошқариш учун масъул бўлишлари мумкин.

12-§ Берилганлар базаси администраторининг имтиёзлари ва махфийлиги

Oracle тизимида маъмурий ишларни бажаришингиз учун (берилганлар базаси билан ишлашда ва эҳтимол берилганлар базаси ишга туширилиши лозим бўлган сервернинг операцион тизими билан ҳам ишлашда) қўшимча имтиёзларга эга бўлишингиз лозим.

Берилганлар базаси администраторининг операцион тизими билан ишлаш ҳуқуқлари

Берилганлар базаси учун аксарият маъмурий вазифаларни бажаришда (масалан, Oracle дастурий таъминотларини ўрнатиш учун) операцион тизим командаларини бажариш имконига эга бўлишингиз лозим.

Берилганлар базаси администраторининг номлари

Берилганлар базасини яратишда автоматик тарзда иккита фойдаланувчи ташкил этилади ва уларга DBA роли тайинланади. Булар қўйидагилар:

- SYS (дастлабки пароли: CHANGE_ON_INSTALL)
- SYSTEM (дастлабки пароли: MANAGER)

***Изоҳ.** Берилганлар лугавий жадвалларига рухсат этилмаган кириш имкониятларини чегаралаш мақсадида, SYS ва SYSTEM фойдаланувчилари паролларини дарҳол ўзгартириши лозим.*

SYS

Ихтиёрий берилганлар базаси яратилганда автоматик тарзда CHANGE_ON_INSTALL паролига эга бўлган SYS фойдаланувчиси ташкил этилади ва унга DBA роли тайинланади.

Берилганлар базасидаги берилганлар луғати таркибида бўлган барча асосий жадвал ва кўринишлар SYS схемасида сақланади. Берилганлар луғати яхлитлигини таъминлаш мақсадида SYS схемасининг жадваллари Oracle томонидан бошқарилади; уларни ҳеч қачон ҳеч қайси турдаги фойдаланувчи ўзгартириши мумкин эмас ва SYS схемасида жадвал яратиш қатъиян манъ этилади.

Берилганлар базасининг аксарият фойдаланувчилари SYS номи билан берилганлар базасига улаш имкониятига эга бўлмасликлари лозим.

SYSTEM

Ихтиёрий берилганлар базаси яратилганда автоматик тарзда MANAGER паролига эга бўлган SYSTEM фойдаланувчиси ташкил этилади ва унга берилганлар базаси билан ишлашда зарур бўладиган барчи тизимий имтиёзлар тайинланади.

SYSTEM фойдаланувчиси, маъмурий маълумотларни кўрсатиш учун ва Oracle ускуналари томонидан фойдаланадиган қўшимча жадвал ва кўринишларни яратади. SYSTEM схемасида шахсий фойдаланувчилар учун ҳеч қачон жадвал яратманг.

DBA роли

"DBA" деб номланган ва олдиндан аниқланган рол автоматик тарзда Oracle берилганлар базаси билан бирга яратилади. Ушбу рол барча тизимий имтиёзлардан таркиб топган. Шу сабабли бу рол тўлиқ ҳуқуқга эга бўлган берилганлар базаси администраторларига тайинланиши лозим.

13-§ Берилганлар базасининг администратори уланишини назорат қилиш методлари

- Берилганлар базасининг администраторлари кўпинча махсус амалларни (берилганлар базасини очиш ёки ёпиш каби) бажаришларига тўғри келади. Бундай амалларни оддий фойдаланувчилар бажариб билмасликлари сабабли берилганлар базасининг администраторларининг схемалари янада сирлироқ бўлган текширувдан ўтишлари лозим.

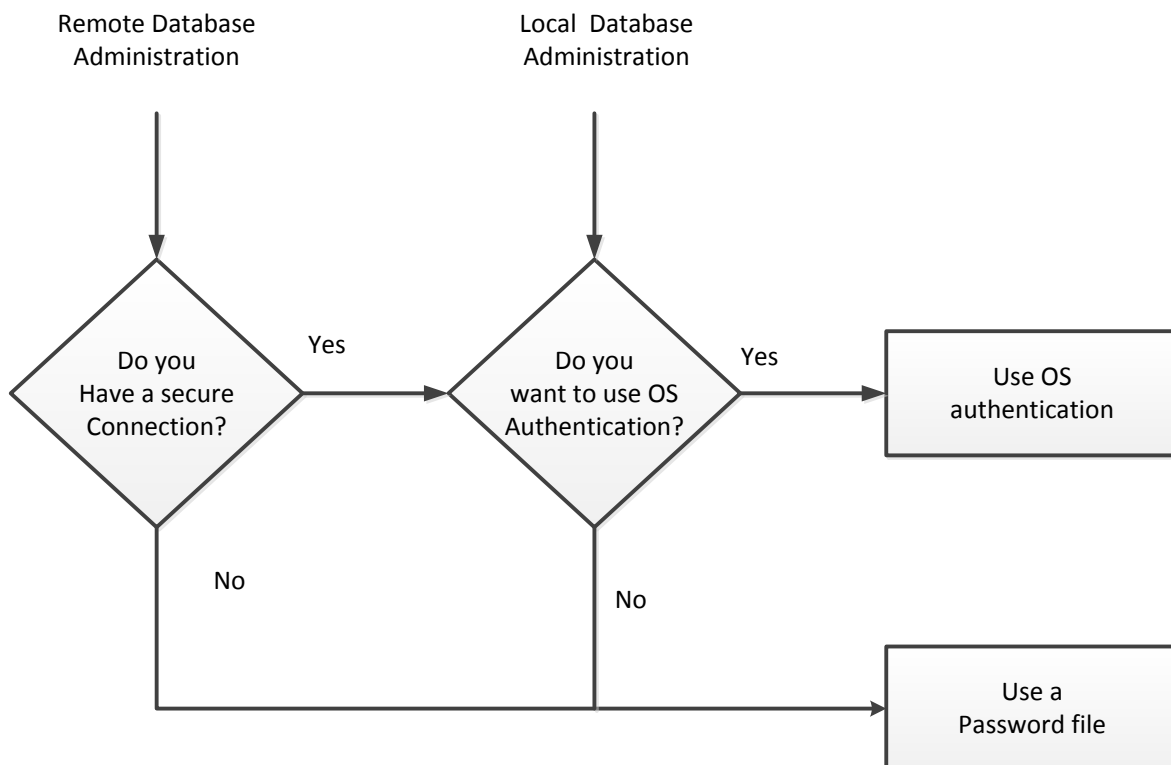
Текширув методини танлаш

Қўйидаги методлар берилганлар базаси администраторларининг тизимга кириш ҳуқуқларини текширишни амалга ошириш учун олдинги версияларидаги CONNECT INTERNAL синтаксисини алмаштиради:

- Операцион тизим текшируви
- Парол файллари

Қаердан туриб (узоқланган бошқарув ёки локал тўрдан бошқарув) берилганлар базасини бошқариш бўйича маъмурий ишларни бажаришингизга қараб у ёки бу методни танлашингиз мумкин. 1.1 расмда ушбу схема намоиш этилган.

13.1- расм. Берилганлар базаси администраторини текширув усуллари



Операцион тизим текширувидан фойдаланиш

Операцион тизим текширувини танласангиз, қўйидаги кадамларни амалга оширинг:

1. Операцион тизим текширувини ўрнатинг.
2. REMOTE_LOGIN_PASSWORDFILE инициализация параметрини NONE ўрнатилганлигига ишонч ҳосил қилишингиз лозим.
3. Текшириладиган фойдаланувчилар қўйидаги уланиш командалари орқали локал ёки узоқланган базаларга уланишлари лозим:

CONNECT / AS SYSOPER

CONNECT / AS SYSDBA

OSOPER and OSDBA

Операцион тизимининг иккита роллари берилганлар базаси администраторини текширувдан ўтказди, агар OSOPER ва OSDBA операцион тизим текшируви ишлатилаётган бўлса.

OSOPER Фойдаланувчига STARTUP, SHUTDOWN, ALTER DATABASE OPEN/MOUNT, ALTER DATABASE BACKUP, ARCHIVE LOG ва RECOVER командаларини

ишлатишга имкон беради ва RESTRICTED SESSION имтиёзини ҳам ўз ичига олади.

OSDBA ADMIN OPTION билан барча тизимий имтиёзлар ва OSOPER ролини ўз ичига олади; CREATE DATABASE ва вақтга асосланган қайта тиклашни бажаришга имкон беради.

OSOPER ва OSDBA операцион тизимга боғлиқ равишда ҳар хил ном ва функционалга эга бўлиши мумкин.

OSOPER ва OSDBA роллари фақат операцион тизими ёрдамида берилиши мумкин. Уларни GRANT жумласи ёрдамида тайинлаб, ўчириб, ҳаттоки чақириб олиб бўлмайди. Администратор имтиёзи билан уланганда ва REMOTE_LOGIN_PASSWORDFILE параметрининг қиймати NONE ўрнатилганда, Oracle операцион тизим билан алоқа қилади ва аввал OSDBA актив ҳолатга келтиришга уринади ва агар муваффақиятсиз бўлса OSOPER ролини актив ҳолга келитиришга ўтади. Агар иккала уриниш ҳам муваффақиятсиз бўлса, уланиш содир бўлмайди.

Пароллар файли текширувидан фойдаланиш

Агар сиз пароллар файли текширувидан фойдаланмоқчи бўлсангиз, қўйидаги қадамларни бажаринг:

1. ORAPWD утилити ёрдамида пароллар файлини ташкил қилинг.

```
ORAPWD FILE=filename PASSWORD=password
ENTRIES=max_users
```

2. REMOTE_LOGIN_PASSWORDFILE инициализация параметрини EXCLUSIVE ўрнатинг.

3. Пароллар файлига берилганлар базасининг администратори функциясини бажарувчи фойдаланувчиларни SQL ёрдамида қўйидаги тарзда қўшинг:

```
GRANT SYSDBA TO scott;
GRANT SYSOPER TO scott;
```

SYSDBA имтиёзи фойдаланувчига OSDBA орқали бажарилиши мумкин бўлган амалларни бажаришга имкон беради. Худди шундай SYSOPER имтиёзи фойдаланувчига OSOPER имтиёзи

орқали бажариш мумкин бўлган амалларни бажаришга имкон беради..

4. Имтиёзи бор фойдаланувчи берилганлар базасига қўйидаги тарзда уланиб билиши лозим.

```
CONNECT scott/tiger@acct.hq.com AS SYSDBA
```

14-§ Пароллар файлини бошқариш

Пароллар файлини ORAPWD - пароллар файлини яратувчи утилита ёрдамида ёки стандарт инсталляциянинг қисми сифатида ҳам очишингиз мумкин.

ORAPWD фойдаланиш

Агар сиз пароллар файлини яратувчи утилитани чақирсангиз, қўйидагича хабар оласиз:

```
orapwd
```

```
Usage: orapwd file=<fname> password=<password>
entries=<users>
```

Бу ерда

`file` - пароллар файлининг номи,

`password` - SYS ва INTERNAL учун парол,

`entries` - ҳар хил DBA ва and OPER максимал сони.

Масалан, қўйидаги команда ACCT.PWD номли пароллар файлини ташкил қилади ва ҳар хил паролларга эга бўлган 30та имтиёзи бор фойдаланувчиларни очишга имкон беради. Файл даставвал, SYSOPER ёки SYSDBA сифатида уланувчилар учун, SECRET пароли остида ташкил қилинади:

```
ORAPWD FILE=acct.pwd PASSWORD=secret ENTRIES=30
```

Қўйида ORAPWD утилитадаги параметрлар тавсифини келтирамыз.

FILE

Ушбу параметр ташкил қилиниши лозим бўлган пароллар файлининг номини ўрнатади. Бунда сиз файлининг тўлиқ йўлини аниқлашингиз лозим. Ушбу файл таркиби шифрланади ва фойдаланувчи ўқиб билмайди.

Пароллар файли учун мўлжалланган файл номининг турлари операцион тизимга боғлиқдир. Баъзи платформалар пароллар файлини махсус форматда бўлиши ва махсус директорияда жойланишини талаб этади. Бошқа платформалар пароллар файлининг номи ва жойлашиш жойи учун ўзгарувчилар муҳитидан фойдаланади. Бу параметрни ўрнатиш мажбурийдир.

ИЗОҲ: Тизимнинг махфийлигини таъминлаш мақсадида пароллар файлини ва пароллар файли жойини аниқлаб берувчи ўзгарувчилар муҳитини ҳимоялаш жуда муҳим ҳисобланади. Ушбуларга кириш имконияти мавжуд бўлган ихтиёрий фойдаланувчи уланиш ҳимоясини хавф остида қолдиради.

PASSWORD

Ушбу параметр SYSOPER ва SYSDBA учун парол ўрнатади. Агар сиз берилганлар базасига улангандан кейин ALTER USER командаси орқали паролни ўзгартирсангиз, унда берилганлар луғатида сақланувчи парол ва пароллар файлида сақланувчи парол бир вақтнинг ўзида янгиланади. INTERNAL фойдаланувчиси фақат олдинги версияларни қўллаб қувватлаш учун зарур. Бу параметрни ўрнатиш мажбурийдир.

ENTRIES

Ушбу параметр пароллар файлига рухсат этилган максимал киришлар сонини ўрнатади. Бу эса берилганлар базасига SYSDBA ёки SYSOPER сифатида уланиши мумкин бўлган фойдаланувчиларнинг максимал сонига мос келади.

ИЗОҲ: Агар сиз ушбу чегарадан ошиб кетмоқчи бўлсангиз, янги пароллар файлини очишингиз зарур. Шу сабабли ушбу сонни захира билан очишингиз мақсадга мувофиқ бўлади.

REMOTE_LOGIN_PASSWORDFILE параметрини ўрнатиш

Пароллар файлини ташкил қилиш билан бирга REMOTE_LOGIN_PASSWORDFILE параметрининг мос қийматини ўрнатиш лозим.

Изоҳ: Нусха ёки берилганлар базасини юклатиш учун Enterprise Manager фойдаланишингиз лозим. Нусха параметрларини ўрнатиш учун берилганлар базасининг номи ва параметрлар файлини аниқлашингиз лозим. Узоқланган берилганлар базасининг тўлиқ номини аниқлашингиз учун Net8 фойдаланишингиз мумкин. Аммо, лекин инициализация параметрлар файли ва унга боғлиқ бўлган конфигурация файли каби файллар мижоз компютерида мавжуд бўлиши лозим. Яъни бу Enterprise Manager қаерда юклатсангиз, ўша ерда параметрлар файли мавжуд бўлиши кераклигини англатади.

NONE

Агар ушбу параметрни NONE ўрнатсангиз, Oracle худди пароллар файли мавжуд бўлмагандай ҳаракат қилади. Ушбу параметрнинг қиймати ошкор тарзда ўрнатилмаса, у NONE бўлади.

EXCLUSIVE

EXCLUSIVE пароллар файли фақат битта берилганлар базаси томонидан ишлатилади. Фақат EXCLUSIVE файли SYSOPER ва SYSDBA фарқли фойдаланувчиларни ўз ичига олиши мумкин. EXCLUSIVE пароллар файлидан фойдаланиш SYSDBA ва SYSOPER тизимий имтиёзларни индивидуал фойдаланувчиларга бериш имкониятини беради ва улар ўзларининг номлари орқали уланадилар.

SHARED

SHARED пароллар файли кўп берилганлар базалари томонидан ишлатилиши мумкин. Аммо, лекин SHARED пароллар файли орқали фақат SYSDBA ва SYSOPER текшириб танилади. SHARED пароллар файлига фойдаланувчиларни қўшиб билмайсиз. SYSDBA ёки SYSOPER тизимий имтиёзга мўхтож барча фойдаланувчилар, айнан шу номлар, SYS ва паролдан фойдаланиб уланишлари лозим. Бу опция кўп берилганлар базасини ягона администратор ёрдамида бошқаришда жуда фойдалидир.

Маслаҳат: *Юқори сатҳдаги ҳимояга эга бўлиш учун, пароллар файлини ташкил қилиниши биланоқ REMOTE_LOGIN_PASSWORDFILE параметр қийматини EXCLUSIVE ўрнатиш лозим.*

Пароллар файлига фойдаланувчиларни қўшиш

SYSDBA ёки SYSOPER имтиёзларни фойдаланувчиларга тайинлашингиз захоти, уларнинг номлари ва имтиёзлари пароллар файлига қўшилади. Агар серверда EXCLUSIVE пароллар файли мавжуд бўлмаса, яъни REMOTE_LOGIN_PASSWORDFILE параметри NONE ёки SHARED ўрнатилган бўлса, ушбу имтиёзларни тайинлашга уриниб кўрсангиз хатолик ҳақида хабар оласиз.

Агар фойдаланувчи ушбу имтиёзлардан бирига эга бўлса, унинг номи пароллар файлида сақланади. Фойдаланувчидан охирги имтиёзни чақириб олганингиздан кейин, у фойдаланувчи пароллар файлидан чиқарилади.

Пароллар файлини ташкил қилиш ва унга янги фойдаланувчиларни қўшиш

1. Пароллар файлини ташкил қилиш қондасига риоя қилинг.
2. REMOTE_LOGIN_PASSWORDFILE инициализация параметрини EXCLUSIVE ўрнатиш.
3. SYSDBA имтиёзи билан қўйидаги мисолдек уланинг:
CONNECT SYS/change_on_install AS SYSDBA
4. Нусхани юклатинг ва керак бўлса берилганлар базасини ташкил қилинг ёки мавжуд бўлган берилганлар базасини ўрнатиш ва очинг.
5. Зарур бўлса фойдаланувчиларни ташкил қилинг. SYSOPER ёки SYSDBA имтиёзларини ўзингизга ва бошқа фойдаланувчиларига мос равишда тайинланг.
6. Энди ушбу фойдаланувчилар пароллар файлига қўшилади ва берилганлар базасига SYSOPER ёки SYSDBA сифатида ўзининг номи ва пароли орқали уланиши мумкин (SYS фойдаланиш ўрнига). Пароллар файлидан фойдаланиш фойдаланувчиларни уланиш пайтида операцион тизим текширувидан қутқармайди.

SYSOPER ва SYSDBA имтиёзларини тайинлаш ва чақириб олиш

Агар серверингиз EXCLUSIVE пароллар файлидан фойдаланса, фойдаланувчиларга SYSDBA ёки SYSOPER тизимий имтиёзларни тайинлаш учун GRANT командасидан қўйидагича фойдаланинг:

```
GRANT SYSDBA TO scott;
```

SYSDBA ёки SYSOPER тизимий имтиёзларни фойдаланувчилардан чақириб олиш учун REVOKE командасидан қўйидагича фойдаланинг:

```
REVOKE SYSDBA FROM scott;
```

Ушбу имтиёзларни ролларга тайинлаш мумкин эмас, чунки роллар берилганлар базасини юклатгандан сўнг кучга киради. Уларда ADMIN OPTION ишлатилмайди..

Пароллар файлининг аъзоларини кўриш

Берилганлар базасида қайси фойдаланувчилар SYSDBA ва SYSOPER тизимий имтиёзларга эга эканлигини кўрмоқчи бўлсангиз V\$PFILE_USERS кўринишга мурожаат қилинг. Ушбу кўриниш қўйидаги устунларга эга:

USERNAME

Пароллар файли орқали танилиши лозим бўлган фойдаланувчининг номи.

SYSDBA

Агар ушбу устуннинг қиймати TRUE бўлса, фойдаланувчи SYSDBA тизимий имтиёз билан уланиши мумкин.

SYSOPER

Агар ушбу устуннинг қиймати TRUE бўлса, фойдаланувчи SYSOPER тизимий имтиёз билан уланиши мумкин.

Администратор имтиёзлари билан уланиш

Агар сиз When you connect with SYSOPER ёки SYSDBA имтиёзлари билан фойдаланувчи номи ва паролидан фойдаланиб улансангиз, сиз сизнинг номингиз билан боғланган схемага эмас, балки SYS схемасига уланасиз.

Администратор имтиёзлари билан уланишга мисол

Масалан, фараз қиламиз фойдаланувчи SCOTT қўйидаги командани берди:

```
CONNECT scott/tiger
```

```
CREATE TABLE scott_test (name VARCHAR2 (20)) ;
```

Кейинроқ, SCOTT қўйидаги командаларини берди:

```
CONNECT scott/tiger AS SYSDBA
```

```
SELECT * FROM scott_test;
```

У SCOTT_TEST мавжуд эмас деган хатолик хабарини олди. Бунга сабаб жадвал SCOTT схемасида очилган бўлиб, аини пайтда SCOTT SYS схемасига мурожаат қилади.

Махфий бўлмаган узоқланган уланиш

Oracle тизимига имтиёзланган фойдаланувчи сифатида махфий бўлмаган усулда уланиш учун қўйидаги шартларни учратасиз:

- Уланишингиз лозим бўлган сервер пароллар файлига эга бўлиши лозим.
- Сиз SYSOPER ёки SYSDBA тизимий имтиёзга эга бўлишингиз лозим.
- Сиз номингиз ва паролингиз орқали уланишингиз лозим.

Локал ва махфий узоқланган уланишлар

Oracle тизимига имтиёзланган фойдаланувчи сифатида локал ва махфий узоқланган усулда уланиш учун қўйидаги шартлардан бирини учратасиз:

- Пароллар файлидан фойдаланиб уланишингиз мумкин (худди олдинги рўйхатдагидек).
- Агар сервер пароллар файлини ишлатмаса ёки сизга SYSOPER ёки SYSDBA имтиёзлари тайинланмаган ва шу сабабли пароллар файлида мавжуд бўлмасангиз, сизнинг операцион тизимидаги номингиз операцион тизими орқали уланиш имтиёзига эга бўлган имтиёзланган фойдаланувчи сифатида текширувдан ўтиши лозим.

Пароллар файлини сақлаш

Ушбу бандда пароллар файлини кенгайтириш, қайта жойлаштириш ва ҳолатини ўзгартиришдан асраш кўриб чиқилади.

Пароллар файлидаги фойдаланувчилар сонини кенгайтириш

Агар сиз фойдаланувчиларга SYSDBA ёки SYSOPER тизимий имтиёзларни тайинлашда, файл тўлиб кетди деган хатолик хабарини олсангиз (ORA-1996), унда каттароқ бўлган пароллар файлини ташкил қилиб имтиёзларни фойдаланувчиларга қайтадан тайинлаб чиқишингиз лозим.

Пароллар файлини алмаштириш

1. V\$PWFILERS_USERS кўринишга сўров бериш орқали қайси фойдаланувчилар SYSDBA ёки SYSOPER имтиёзларга эга эканлигини аниқланг.
2. Берилганлар базасини ёпинг.
3. Мавжуд бўлган пароллар файлини ўчириг.
4. ORAPWD утилитаси ёрдамида янги пароллар файлини (кўрсатмага риоя қилиб) ташкил қилинг. ENTRIES параметри қийматини етарлича катта қилиб ўрнатганингизга ишонч ҳосил қилинг.
5. «Пароллар файлига фойдаланувчиларни қўшиш» кўрсатмасига риоя қилинг.

Пароллар файлини қайта жойлаштириш

Пароллар файлини яратишингиздан сўнг, уни хоҳлаган жойингизда қайта жойлаштиришингиз мумкин. Пароллар файлини қайта жойлаштирганингиздан сўнг, мос муҳит ўзгарувчилари қийматларини янги йўлларга ўзгартиришингиз лозим. Агар операцион тизимингиз олдиндан аниқланган йўл номини ишлатса, унда пароллар файли жойини ўзгартириб билмайсиз.

Пароллар файлини чиқариб ташлаш

Агар сиз фойдаланувчиларни текшириш учун пароллар файлидан бошқа ҳеч қачон фойдаланмаслик қарорига келсангиз, пароллар файлини ўчиришингиз мумкин ва REMOTE_LOGIN_PASSWORDFILE инициализация параметри қийматини NONE ўрнатинг. Ушбу файлини ўчиришингиздан сўнг берилганлар базасини фақат операцион тизими текширувидан ўтадиган фойдаланувчиларгина администраторлик амалларни бажариб биладилар.

ДИҚҚАТ: Агар берилганлар базаси ёки нусхаси REMOTE_LOGIN_PASSWORDFILE=EXCLUSIVE (ёки SHARED) параметри билан ўрнатилган бўлса, пароллар файлини ўчирманг ёки мукаммаллаштирманг. Агар барибир шу ишни қилсангиз, унда пароллар файлидан фойдаланиб узоқдан қайта уланиб билмайсиз. ҳаттоки пароллар файлини алмаштирсангиз ҳам, янги пароллар файлини ишлатиб билмайсиз, чунки ташкил қилингин вақти ва текширув

суммаси хато бўлади.

Пароллар файли ҳолатини ўзгартириш

Пароллар файлининг ҳолати пароллар файлида сақланади. Биринчи маротаба файлни яратишда, унинг ҳолати одатда SHARED бўлади. Пароллар файли ҳолатини REMOTE_LOGIN_PASSWORDFILE параметрини ўрнатиш орқали ўзгартиришингиз мумкин. Нусхани юклатишингиз билан, Oracle ушбу параметр қийматини мижоз машинасида сақланаётган параметрларнинг инициализация файлидан ўқиб олади. Берилганлар базасини ўрнатишингиз билан, Oracle ушбу параметр қийматини пароллар файлида сақланаётган қиймат билан солиштиради. Агар ушбу қийматлар устма-уст тушмаса, файлда сақланаётган қиймат устидан янги қиймат ёзилади.

15-§ Берилганлар базаси администратори учун мўлжалланган утилиталар

Oracle серверини ишлатиш ва назорат қилиш учун сизга бир нечта утилиталар тавсия этилади.

SQL*Loader

SQL*Loader ҳам берилганлар базаси администратори ва ҳам Oracle фойдаланувчилари томонидан фойдаланилади. У берилганларни стандарт операцион тизими файлларидан (матнли ёки С форматидаги файлларни) Oracle берилганлар базаси жадвалларига кўчиради.

Экспорт ва импорт

Экспорт ва импорт утилиталари Oracle форматидаги мавжуд бўлган берилганларни Oracle берилганлар базасига кўчиришга ва ундан чиқаришга имкон беради. Масалан, экспорт файллари берилганлар базасидаги берилганларни архив қилиши мумкин; бир хил ёки ҳар хил операцион тизимларида ишлаётган ҳар хил Oracle берилганлар базаси орасида берилганларни кўчириши мумкин.

16-§ Берилганлар базаси администраторининг приоритетлари

Умуман айтганда берилганлар базасини яратиш ва ишлатиш учун бир нечта кадамларни бажаришингиз лозим.

Ихтиёрий турдаги компьютер тизимида Oracle серверини конфигурациялаш учун қўйидаги кадамларни амалга ошириш лозим.

Oracle серверини конфигурациялаш

- 1 кадам: Oracle дастурий таъминотини инсталляциялаш
- 2 кадам: Берилганлар базаси серверининг винчестерини ҳисоблаш
- 3 кадам: Берилганлар базасини режалаштириш
- 4 кадам: Берилганлар базасини яратиш ва очиш
- 5 кадам: Берилганлар базасини кўриш
- 6 кадам: Берилганлар базасини орқага қайтариш
- 7 кадам: Тизим фойдаланувчиларига ролларни тайинлаш
- 8 кадам: Берилганлар базасини ишлатиш

1 кадам: Oracle дастурий таъминотини инсталляциялаш

Берилганлар базасининг администратори сифатида, сиз Oracle серверининг барча дастурий таъминоти, дастурий ускуналари ва берилганлар базаси билан ишлайдиган берилганлар базаси дастурларини инсталляция қилишингиз лозим. Баъзи тақсимланган жараёнларни инсталляциялаш учун, берилганлар базаси марказий компьютердан бошқарилади ва берилганлар базасининг дастурий ускуналари ва берилганлар базаси билан ишлайдиган берилганлар базаси дастурлари узоқланган машинада бажарилади; ушбу ҳолатда сиз узоқланган машинадан Oracle ишлайдиган компьютерга уланишингиз учун Oracle Net8 драйверларини инсталляция қилишингиз лозим.

2 кадам: берилганлар базаси серверининг винчестерини ҳисоблаш

Инсталляция жараёни тугагандан сўнг, Oracle ва унинг дастурлари ишлатиши мумкин бўлган компьютер ресурсларидан унумли равишда фойдаланиш усуллари ҳисобланг. Ушбу ҳисоб-китоб қўйидагиларни аниқлаб бериши лозим:

- Oracle ва унинг берилганлар базаси учун рухсат этилган мантиқий дисклар нечта
- Мўлжалланган Oracle нусхаларини юклатиш учун хотирадан қанча жой ажратилган.

3 кадам: Берилганлар базасини режалаштириш

Берилганлар базасининг администратори сифатида, сиз қўйидагиларни режалаштиришингиз лозим:

- Берилганлар базасининг мантикий тузилмаси
- Берилганлар базасининг тўлиқ тузилмаси
- Берилганлар базасини тиклаш стратегияси

Берилганлар базасининг мантикий тузилмаси тизимнинг ишлашига ва берилганлар базаси бошқаруви учун зарур бўлган амалларни бажаришга таъсирини билиш жуда муҳим ҳисобланади. Масалан, берилганлар базаси учун ихтиёрий жадвалий фазони яратишдан олдин, жадвалий фазо учун нечта берилганлар файллари зарур эканлигини билишингиз, ушбу берилганлар файллари жисмонан қаерда жойланиши (қайси мантикий дискларда) ва қандай турдаги маълумотлар ҳар бир жадвалий фазода сақланишини аниқлашингиз лозим. Берилганлар базасининг мантикий тузилмасини режалаштиришда, ушбу тузилманинг берилганлар базаси яратилганда ва ишлаганда қандай таъсир кўрсатишини ҳисобга олиш зарур. Бунда берилганлар базасининг мантикий сақловчи тузилмаларининг қўйидаги ҳолатларга таъсири тушунилади:

- Oracle тизимини ишлатувчи компьютернинг ишлаш ҳолати;
- берилганларга кириш амалларини бажаришда берилганлар базасининг ишлаш ҳолати
- берилганлар базаси учун орқага қайтариш ва тиклаш процедураларининг ишлаш унумдорлиги

Берилганлар базасидаги объектлар орасидаги боғланишларни ва ушбу объектларнинг ҳар бирининг сақлаш характеристикаларини режалаштиринг. Берилганлар базасидаги объектлар орасидаги боғланишларни ва ушбу объектларнинг ҳар бирининг сақлаш характеристикаларини режалаштиришда, уларни яратишдан олдин берилганлар базасини яхлит бир бирлик сифатида ишлашини инобатга олишингиз лозим. Режалаштиришда берилганлар базасини ўсишини ҳисобга олинг.

Тақсимланган берилганлар базаси муҳитида режалаштиришнинг ушбу қадами жуда муҳим бўлиб ҳисобланади. Берилганларнинг жисмонан жойланиши дастурларнинг ишлашига катта таъсир кўрсатади.

Режалаштиришнинг юқорида келтирилган фазаларида, берилганлар базасини орқага қайтариш стратегиясини режалаштиринг.

4 қадам: Берилганлар базасини яратиш ва очиш

Берилганлар базаси тузилишини ишлаб чиқишдан сўнг, уни яратишингиз мумкин ва ундан нормал ҳолатда фойдаланиш учун уни очинг. Операцион тизимга боғлиқ равишда, Oracle тизимини инсталляция қилиш пайтида берилганлар базаси яратилиши мумкин. Унда сизга нусхани юклатиш ва дастлабки берилганлар базасини ўрнатиш ва очишингиз қолади холос.

5 қадам: Берилганлар базаси мантикий тузилмасини қуриш

Берилганлар базасини яратиб ва уни юклатганингиздан сўнг, берилганлар базасининг режалаштирилган мантикий тузилмасини зарур бўлган ҳамма rollback сегментлари ва жадвалий фазоларни яратиш орқали қуришингиз мумкин. Ушбунни қуришдан кейин, берилганлар базаси объектларини яратишингиз мумкин.

6 қадам: Берилганлар базасини орқага қайтариш

Берилганлар базаси тузилмасини яратишдан сўнг, қўшимча redo log файлларини яратиш орқали, берилганлар базасини режалаштирилган орқага қайтариш стратегиясини амалга оширинг, бунда биринчидан тўлиқ орқага қайтариш (online or offline) ва келгусида муайян интервалларда орқага қайтаришни ҳисобга олинг.

7 қадам: Тизим фойдаланувчиларига тегишли ролларни тайинлаш

Берилганлар базаси тузилмасини орқага қайтариш стратегиясини амалга оширгандан сўнг, Oracle лицензияси шартномасига таянган ҳолда, тизим фойдаланувчиларини яратишингиз ва уларга тегишли ролларни очиб тайинлашингиз мумкин.

8 қадам: Берилганлар базаси ишини бошқариш

Берилганлар базасининг тизимий ишлашини оптималлаштириш сизнинг доимий вазифангиз бўлиб ҳисобланади.

Oracle дастурий таъминоти версияларини аниқлаш

Oracle маҳсулотлари доимий равишда ривожланиб ўзгариб туради. Шу сабабли бир вақтнинг ўзида ҳар хил версиядаги маҳсулотлар бирга ишлаши мумкин.

Оний версияни аниқлаш

Айни пайтда Oracle тизими ва унинг қайси компоненталаридан фойдаланаётганингизни кўрмоқчи бўлсангиз берилганлар лугатидаги PRODUCT_COMPONENT_VERSION кўринишга сўров беринг:

```
SELECT * FROM product_component_version;
```

	PRODUCT	VERSION	STATUS
1	NLSRTL	11.2.0.2.0	Production
2	Oracle Database 11g Express Edition	11.2.0.2.0	Production
3	PL/SQL	11.2.0.2.0	Production
4	TNS for 32-bit Windows:	11.2.0.2.0	Production

Oracle берилганлар базасини яратиш

Ушбу бобда Oracle берилганлар базасини яратиш учун зарур бўладиган қадамлар келтирилади:

- Берилганлар базасини яратишдан олдин қилинадиган ишлар
- Oracle берилганлар базасини яратиш
- Параметрлар
- Берилганлар базасини яратгандан сўнг қилинадиган ишлар
- Дастлабки бошқарув кўрсатмалари

17-§ Берилганлар базасини яратишдан олдин қилинадиган ишлар

Ушбу бўлимда қўйидаги мавзулар ёритилади:

- Яратиш учун қўйиладиган талаблар
- Бошлангич берилганлар базасидан фойдаланиш
- Берилганлар базасининг олдинги версиясини кўчириш

Берилганлар базасини яратиш деганда бир нечта операцион тизими файлларини Oracle берилганлар базаси сифатида биргаликда ишлашига тайёрлашни тушуниш лозим.

Берилганлар базасини яратиш жараёни қўйидаги амалларни ўз ичига олади:

- Янги берилганлар файлларини яратиш ёки мавжуд бўлган олдинги берилганлар файлларидан фойдаланиш
- Oracle талаб этадиган берилганлар базасидан фойдаланиш учун зарур бўладиган маълумот тузилмаларини (берилганлар лугати) яратиш
- Берилганлар базаси учун назорат ва redo log файлларини яратиш ва инициализация қилиш

Берилганлар базасини яратишдан олдин қўйидаги ишларни амалга оширинг:

- Берилганлар базасининг жадвалларини ва индексларини режалаштиринг ва улар қанча фазони эгаллашини баҳоланг.

- Янги берилганлар базасини химоялашни режалаштиринг, жумладан online режимидаги ишлаши ва redo log архивини (ва у қанча жойни талаб этади) конфигурациялаш, ҳамда қайтариш стратегияси буларга мисолдир.
- Берилганлар базасидаги белгилар тўпламини (character set) танланг. Берилганлар базасидаги белгилар тўпламини берилганлар базасини яратишда аниқлашингиз лозим. Берилганлар базаси яратилгандан сўнг, белгилар тўпламини ўзгартириб бўлмайди. Берилганлар базасида барча белгилар берилганлар (жумладан берилганлар лугатидаги берилганлар) берилганлар базасидаги белгилар тўпламида сақланади. Шу сабабли мижоз томонидаги белгилар тўплами билан берилганлар базасидаги белгилар тўплами мос келиши лозим.

Яратиш учун қўйиладиган талаблар

Янги берилганлар базасини яратиш учун қўйидагиларга эга бўлишингиз лозим:

- Берилганлар базаси администраторининг тўлиқ вазифасини бажариш учун зарур бўладиган операцион тизим имтиёзлари
- Oracle нусхасини ишлатиш учун зарур бўладиган етарли хотира
- Oracle тизими ишлаётган компютер дискларида режалаштирилган берилганлар базасини сақлаш учун етарли сақлаш жойи.

Дастлабки берилганлар базасидан фойдаланиш

Операцион тизимига боглиқ равишда, берилганлар базаси Oracle тизимини инсталляция қилиш жараёнининг қисми сифатида автомат тарзида яратилиши мумкин. Сиз ушбу берилганлар базасидан ўз мақсадларингизга мослаштириб фойдаланишингиз мумкин ёки уни алмаштириш учун битта ёки ундан кўп янги берилганлар базаларини яратишингиз мумкин. Берилганлар базасининг олдинги версиясини кўчириш.

Берилганлар базасининг олдинги версиясини кўчириш

Сиз Oracle олдинги версияси томонидан бошқариладиган мавжуд бўлган берилганлар базасини Oracle дастурий таъминотининг янги версияларига кўчириб ишлатишингиз мумкин.

18-§ Oracle берилганлар базасини яратиш

Ушбу бўлимда қўйидаги мавзулар муҳокама қилинади:

- Oracle берилганлар базасини яратиш қадамлари
- Берилганлар базасини яратишга мисол
- Берилганлар базасини яратишда вужудга келадиган муаммолар
- Берилганлар базасини ўчириб ташлаш

Oracle берилганлар базасини яратиш қадамлари

Oracle берилганлар базасини яратиш учун қўйидаги қадамларни кўрсатилган тартибда амалга ошириш лозим.

Янги берилганлар базасини яратиш ва ундан тизим фойдаланишига имкон бериш

1. Барча мавжуд бўлган берилганлар базаларини орқага қайтариш.
2. Параметрлар файлларини яратиш.
3. Янги параметрлар файлларини таҳрирлаш.
4. Тизимингиз учун нусха идентификаторини текширинг.
5. SQL*Plus юклатинг ва Oracle тизимига SYSDBA сифатида уланинг.
6. Нусхани юклатинг.
7. Берилганлар базасини яратинг.
8. Берилганлар базасидан захиравий нусха олиб қолиш.

1 қадам: Ихтиёрий мавжуд бўлган берилганлар базасидан захиравий нусха олиш

Янги берилганлар базасини яратишдан олдин мавжуд бўлган барча берилганлар базаларидан захиравий нусха олишни Oracle корпорацияси қатъий тавсия этади. Захиравий нусха параметрлар файллари, берилганлар файллари, қайд журнали файллари ва назорат файлларини ўз ичига олиши лозим.

2 қадам: Параметрлар файлини яратиш.

Ихтиёрий Oracle берилганлар базаси учун нусха (Тизимнинг глобал соҳаси ва фон жараёнлари) ўз ишини параметрлар файлидан фойдаланиб бошлайди.

Тизимингиздаги ҳар бир берилганлар базаси энг камида битта айнан шу берилганлар базасига мос параметрлар файлига эга бўлиши лозим. Бир нечта берилганлар базалари учун айнан битта параметрлар файлини ишлатманг.

Берилганлар базаси учун параметрлар файлини яратиш учун Oracle тизимининг дистрибютив муҳитидаги параметрлар файлидан операцион тизими ёрдамида нусха олинг. Ушбу нусхага бошқа ном беринг. Сўнг ушбу файлни таҳрирлаб янги берилганлар базаси учун янги файл сифатида ишлатинг.

3 қадам: Янги параметрлар файлини таҳрирлаш.

Янги берилганлар базасини яратиш учун, янги параметрлар файлидаги қўйидаги параметрларни текшириб мос равишда таҳрирланг:

DB_NAME
DB_DOMAIN
CONTROL_FILES
DB_BLOCK_SIZE
DB_BLOCK_BUFFERS
PROCESSES
ROLLBACK_SEGMENTS

Мос лицензия параметрларини ҳам таҳрирлашингиз лозим:

LICENSE_MAX_SESSIONS
LICENSE_SESSION_WARNING
LICENSE_MAX_USERS

4 қадам: Тизимингиз учун нусха идентификаторини текширинг

Агар сиз бошқа берилганлар базаларига эга бўлсангиз, Oracle нусхалари идентификаторини текширинг.

5 қадам: SQL*Plus ишга туширинг ва Oracle тизимига SYSDBA сифатида уланинг.

Берилганлар базасига SYSDBA сифатида уланинг.

```
$ SQLPLUS /nolog
connect username/password as sysdba
```

6 қадам: Нусхани юклатинг.

Берилганлар базасини ўрнатмасдан туриб нусхани ишга туширингиз мумкин, одатда янги берилганлар базасини яратишда шундай иш қиласиз. Бунда STARTUP командасидан NOMOUNT вазиятидан фойдаланинг:

```
STARTUP NOMOUNT;
```

Ушбу вазиятда берилганлар базаси мавжуд эмас. Янги берилганлар базасини яратиш учун фақат SGA ва фон жараёнлари ишга тушдилар.

7 қадам: Берилганлар базасини яратиш.

Янги берилганлар базасини яратиш учун SQL тилидаги CREATE DATABASE жумласидан фойдаланинг. Бунда берилганлар базасининг номини, файлларнинг максимал сонини, файллар номлари ва ўлчовларини ва ҳ.к. кўрсатиш лозим.

CREATE DATABASE жумласини ишлатишда Oracle қўйидаги ишларни бажаради:

- берилганлар базаси учун берилганлар файлларини яратади
- берилганлар базаси учун назорат файлларини яратади
- берилганлар базаси учун қайд журнали (redo log) файлларини яратади
- SYSTEM жадвалий фазо ва SYSTEM қайтарув сегментини яратади
- Берилганлар лугатини яратади
- SYS ва SYSTEM фойдаланувчиларини яратади
- Берилганларни берилганлар базасида сақлаш учун ишлатиладиган белгилар тўпламини аниқлайди
- Берилганлар базасини ўрнатади ва уни ишлатиш учун очади

ДИҚҚАТ: *Сиз аниқлаган берилганлар файли ва қайд журнали файлларининг номлари бошқа берилганлар базасидаги номлар билан устма-уст тушмасликка қатъий ишонч ҳосил қилинг.*

8-қадам: Берилганлар базасидан захиравий нусха олиш.

Берилганлар базасидан тўлиқ захиравий нусха олишингиз лозим. Чунки мабодо муҳит ишдан чикса, унда сизда берилганлар базасини тиклаш учун зарур бўладиган барча файллар мавжудлигига қатъий ишонч ҳосил қилишингиз лозим.

Берилганлар базасини яратиш: Мисол

Қўйидаги жумла CREATE DATABASE жумласига мисол бўла олади:

```
CREATE DATABASE test
DATAFILE 'test_system' SIZE 10M
LOGFILE GROUP 1 ('test_log1a', 'test_log1b') SIZE
500K,
GROUP 2 ('test_log2a', 'test_log2b') SIZE 500K;
```

Ушбу мисолда MAXLOGFILES, MAXLOGMEMBERS, MAXDATAFILES, MAXLOGHISTORY ва MAXINSTANCES параметрларнинг қийматлари операцион тизимга боғлиқ бўлиб, олдиндан Oracle тизими томонидан аниқланган деб фараз қилинади.

Берилганлар базаси NOARCHIVELOG ва EXCLUSIVE вазиятда ўрнатилган деб фараз қилинади ва сўнг очилади.

Юқорида келтирилган мисол асосида берилганлар базаси яратилса, у қўйидаги характеристикаларга эга бўлади:

- Янги берилганлар базаси TEST деб номланади.
- Янги берилганлар базасидаги SYSTEM жадвалий фазоси бир дона 10 MB ҳажмга эга бўлган TEST_SYSTEM деб номланган берилганлар файлида жойлашади.
- Янги берилганлар базаси, ҳар бири иккита 500 KB ҳажмга эга бўлган қисмлардан таркиб топган иккита онлайн қайд журнали гуруҳларидан иборат.
- Янги берилганлар базаси параметрлар файлида аниқланган мавжуд бўлган назорат файлларини устидан ёзолмайди.

Эслатма: Берилганлар базасини яратиши жараёнида бир нечта чегаралар ўрнатишингиз мумкин. Ушбу чегараларнинг айримлари операцион тизим имкониятларига боғлиқ бўлиб бир-бирига ўзаро таъсир кўрсатади. Масалан, агар MAXDATAFILES параметр қийматини ўрнатган бўлсангиз, Oracle тизими назорат файлида MAXDATAFILES файл номларини сақлаш учун етарлича жой ажратиш қўяди; ҳаттоки берилганлар базаси дастлаб битта берилганлар файлига эга бўлса ҳам. Назорат файлининг максимал ўлчови чегараланган бўлганлиги ва операцион тизимга боғлиқлиги, сабабли CREATE DATABASE барча параметрларини максимал назарий қийматларини ўрнатолмайсиз.

Берилганлар базасини яратишда пайдо бўладиган қийинчиликлар

Агарда айрим сабабларга кўра берилганлар базасини яратиш жараёни муваффақиятсиз тугаса, нусхани ёпинг ва CREATE DATABASE жумласи ёрдамида яратилган барча файлларни ўчириш (уларни қайта яратишга уринишдан олдин).

Берилганлар базасини ўчириш

Берилганлар базасини ўчириш учун, унинг берилганлар файлларини, қайд журнали файлларини ва мос бошқа барча файлларни (назорат файллари, параметрлар файллари, архив журнали файллари) чиқариб ташланг.

Берилганлар базасининг берилганлар файллари ва қайд журнали файлларининг номларини V\$DATAFILE ва V\$LOGFILE кўринишлардан сўров орқали ўқиб олишингиз мумкин.

Параметрлар

Ушбу бандда берилганлар базасини яратишда керак бўладиган қўйидаги параметрлар тавсифланади:

- DB_NAME ва DB_DOMAIN
- CONTROL_FILES
- DB_BLOCK_SIZE
- PROCESSES
- ROLLBACK_SEGMENTS
- Лицензия параметрлари
- DB_BLOCK_BUFFERS
- LICENSE_MAX_SESSIONS ва LICENSE_SESSIONS_WARNING
- LICENSE_MAX_USERS

DB_NAME ва DB_DOMAIN

Берилганлар базасининг глобал номи (номи ва тармоқ тузилмасидаги жойлашуви) DB_NAME ва DB_DOMAIN параметрларини ўрнатиш орқали берилганлар базасини яратишдан олдин қўйилади. Берилганлар базасининг номини яратгандан сўнг, уни ўзгартириш осон бўлмайди. The DB_NAME параметри берилганлар базаси номининг локал номи компонентасини аниқлайди, шу билан бирга DB_DOMAIN параметри тармоқ тузилмасидаги доменини (мантқиқий жойлашуви) аниқлайди. Масалан, The combination of the settings for these two parameters should form a database name that is unique within a network.

For example, to create a database with a global database name of TEST.US.ACME.COM глобал номли берилганлар базасини яратиш учун янги параметрлар файлидаги параметрларни қўйидагича тахрирланг:

```
DB_NAME = TEST
```

```
DB_DOMAIN = US.ACME.COM
```

DB_NAME матний қатор бўлиб, узунлиги саккизта белгилардан ошмаслиги лозим. Берилганлар базасини яратиш жараёнида During database creation, the name provided for DB_NAME параметридаги ном, берилганлар базасининг берилганлар файлларига, қайд журнали файлларига ва назорат файлига ёзилади. Агарда нусхани юклатиш вақтида DB_NAME параметрининг қиймати (параметрлар файлидаги) ва назорат файлидаги берилганлар базасининг номи айнан бир хил бўлмаса, берилганлар базаси ишга туширилмайди.

DB_DOMAIN матний қатор бўлиб, берилганлар базаси яратилган тармоқ доменини аниқлайди. Агар сиз яратмоқчи бўлган берилганлар базаси тақсимланган берилганлар базаси тизимининг қисми бўладиган бўлса, берилганлар базасини яратишдан олдин ушбу параметрлар қийматларига махсус эътибор беринг.

CONTROL_FILES

CONTROL_FILES параметрини янги параметрлар файлига қўшиб қўйинг ва унинг қийматини назорат файллар номлари рўйхатига (янги берилганлар базаси томонидан фойдаланиш учун) ўрнатиш. Берилганлар базасининг назорат файлини яратишда янги операцион тизими файлини яратишини хоҳласангиз, CONTROL_FILES параметрида келтирилган файл номлари тизимингизда мавжуд бўлган жорий файл номлари билан устма-уст тушмаслигига қатъий ишонч ҳосил қилинг. Берилганлар базасининг назорат файлини яратишда мавжуд бўлган операцион тизими файлидан фойдаланишни хоҳласангиз, CONTROL_FILES параметрида келтирилган файл номлари тизимингизда мавжуд бўлган жорий файл номлари билан устма-уст тушишигига қатъий ишонч ҳосил қилинг.

Агар CONTROL_FILES параметрида файллар номлари келтирилмаган бўлса, унда Oracle олдиндан аниқланган файл номини ўзи ўрнатади.

Oracle корпорацияси ҳар бир берилганлар базаси учун алоҳида жисмоний дискларида сақланадиган камида иккита назорат файлларидан фойдаланишингизни қатъий равишда тавсия этади. Шу сабабли, янги параметрлар файлида CONTROL_FILES параметрини аниқлашда, қўйидаги қоидаларга риоя қилишингиз лозим:

- CONTROL_FILES параметри учун камида иккита файл номларини келтиринг.

- ҳар бир назорат файлини алоҳида жисмоний дискларда жойлаштиринг.

Эслатма: Назорат файллари учун файл аниқланишии операцион тизимга боғлиқ. Операцион тизимга боғлиқ бўлмаган равишда, ҳамма вақт назорат файлларингиз учун файл номларини тўлиқ аниқланг.

CREATE DATABASE жумласини бажаришда, параметрлар файлидаги CONTROL_FILES параметрида келтирилган назорат файллари яратилади.

DB_BLOCK_SIZE

Одатда Oracle берилганлар блокиннинг ўлчовини олдиндан 2K ёки 4K ҳажмида аниқлаб қўяди. Умуман айтганда, кўпинча ушбу ўлчов талабга жавоб беради. Аммо баъзида каттароқ ҳажмга эга бўлган берилганлар блоки кўпроқ самара беради. Бу ҳолатлар кўйидагиларни ўз ичига олади:

- Oracle катта хотирага ва тезкор дискларга эга бўлган компьютер тизимида ишлайди. Масалан, мейнфрейм туридаги компьютерлар бошқарадиган берилганлар базасида одатда берилганлар блокиннинг ўлчови 4K ёки ундан катта қилиб олинади.
- Одатда Oracle ишлаётган операцион тизими кичик ўлчовли блокдан фойдаланади. Oracle тизими самарали ишлаши учун имкон борича берилганлар базаси блокиннинг ўлчови операцион тизими блокиннинг ўлчовига қаррали бўлиши лозим.

Ҳар бир берилганлар базасининг блок ўлчови берилганлар базасининг яратиш пайтида DB_BLOCK_SIZE параметри орқали берилади. Блок ўлчовини берилганлар базасини яратишдан кейин ўзгартариш мумкин эмас. Агар берилганлар базасидаги блок ўлчови операцион системасидаги блок ўлчовидан фарқ қилса, унда берилганлар базасидаги блок ўлчовини операцион системасидаги блок ўлчовига қаррали қилиб танланг.

Масалан, операцион системасидаги блок ўлчови 2K (2048 байт) бўлса, унда DB_BLOCK_SIZE параметрининг қўйидаги қийматини ўрнатиш тўғри бўлади:

DB_BLOCK_SIZE=4096

DB_BLOCK_SIZE параметри тизимнинг глобал соҳасидаги (SGA) кеш-буферда берилганлар базасининг буферлари ўлчовини ҳам аниқлайди.

DB_BLOCK_BUFFERS

Ушбу параметр тизимнинг глобал соҳасидаги (SGA) кеш-буферада берилганлар базасининг буферлари сонини аниқлайди. Одатда амалиётда минимал буферлар сони 1000 дан 2000 гача бўлиши мумкин.

PROCESSES

Ушбу параметр Oracle тизимига рақобат билан уланадиган операцион тизимнинг максимал жараёнлар сонини аниқлайди. Ушбу параметрнинг қиймати 5та тескари жараёнлар ва битта ҳар бир фойдаланувчи жараёнини ўз ичига олиши лозим. Масалан, агар сиз 50та рақобат билан уланадиган фойдаланувчиларни режлаштирмоқчи бўлсангиз, унда ушбу параметр қийматини камида 55 қилиб ўрнатинг.

ROLLBACK_SEGMENTS

Ушбу параметр берилганлар базасини юклатганда Oracle тизими нусхасида талаб этиладиган қайтариш сегментлари рўйхатини белгилайди.

Диққат: *Инсталляциядан сўнг, ихтиёрий схема юбъектини яратиш мумкинлигини таъминлаш учун SYSTEM қайтариш сегментида қўшимча тарзда SYSTEM жадваллар фазосида камида битта қайтариш сегментини яратишингиз лозим.*

LICENSE_MAX_SESSIONS ва LICENSE_SESSIONS_WARNING

Муайян компьютердаги берилганлар базасига уланадиган рақобатли сессиялар сонини чегаралаб қўйишингиз мумкин. Муайян нусха учун рақобатли сессиялар максимал сонини нусхани юклатадиган параметрлар файлидаги LICENSE_MAX_SESSIONS ва LICENSE_SESSIONS_WARNING параметрлари орқали қўйидаги мисолда кўрсатилганидек ўрнатишингиз мумкин:

`LICENSE_MAX_SESSIONS = 80`

LICENSE_MAX_USERS

Берилганлар базасида яратилиши мумкин бўлган фойдаланувчилар сонини чегаралаб қўйишингиз мумкин. Шу мақсадда LICENSE_MAX_USERS параметри ишлатилади:

`LICENSE_MAX_USERS = 200`

19-§ ББ юклатиш ва тўхтатиш

Ушбу бобда Oracle берилганлар базасини ишга тушириш ва ишни якунлаш жараёнлари тавсифланади. Боб қўйидаги бўлимлардан иборат:

- ББ ишга тушириш
- ББ билан ишлашни таъминлаш
- ББ билан ишлашни якунлаш
- Параметр файлларидан фойдаланиш.

ББ ишга тушириш

Ушбу бўлим қўйидаги қисмлардан иборат:

- Нусхани ишга тушириш учун тайёрлаш
- Нусхани ишга тушириш: Сценариялар

ББ ёки нусхани командалар чизигидан фойдаланиб ишга тушириш учун, Oracle тизимига администратор имтиёзи билан SQL*Plus орқали уланиб STARTUP командасини бериш лозим. STARTUP ва SHUTDOWN командаларини «Recovery Manager» орқали ҳам ижро этиш мумкин.

Нусха ва ББ ҳар хил усулларда юклатишингиз мумкин:

- ББ ўрнатмасдан нусхани ишга тушириш
- Нусхани ишга тушириш ва ББ ўрнатиб ёпиқ ҳолатда сақлаш
- Нусхани ишга тушириш, ББ ўрнатиб ва қўйидаги усуллар билан уни очиш:
 - чегараланмаган усулда (барча фойдаланувчилар ишлаш имконига эга)
 - чегараланган усулда (фақат администраторлар ишлаш имконига эга)

Нусхани ишга тушириш учун тайёрлаш

Нусхани ишга туширишдан олдин, бир қатор ишларни амалга оширишингиз лозим.

1. ББ уланмасдан қўйидаги команда орқали SQL*Plus ишга туширинг:

```
sqlplus /nolog
```

2. Oracle тизимига SYSDBA бўлиб уланинг:

```
connect username/password as sysdba
```

3. STARTUP командасини киритишда ББ номини аниқлаб параметрлар файлига тўлиқ йўлини кўрсатинг:

```
STARTUP database_name PFILE=myinit.ora
```

Агар сиз PFILE параметри қийматини аниқламасангиз, Oracle стандарт параметрлар файлидан фойдаланади. Агар сиз ББ номини аниқламасангиз, Oracle параметрлар файлидаги DB_NAME параметрининг қийматини ишлатиб нусхани юклатади.

Нусхани юклатиш: Сценариялар

Қўйидаги сценариялар нусхани юклатиш усулларини тавсифлаб беради.

ББ ўрнатмасдан нусхани юклатиш

ББ ўрнатмасдан нусхани юклатишингиз мумкин:

```
STARTUP NOMOUNT;
```

Нусхани юклаш ва ББ ўрнатиш

Нусхани юклаб ва ББ очмасдан туриб ўрнатишингиз мумкин. Қўйидаги вазифаларни бажаришда ББ ўрнатилган бўлиб, лекин очилмаган бўлиши лозим:

- берилганлар файлларини қайта номлаш
- тиклаш журнали файлларини ўчириш, қайта номлаш ва янги файл қўшиш
- тиклаш журналини архив учун опциясини ишчи ёки ишсиз ҳолатга келтириш
- берилганлар базасини тўлиқ тиклашини амалга ошириш

Нусхани юклатинг ва берилганлар базасини ўрнатинг, аммо уни ёпиқ ҳолатда сақланг:

```
STARTUP MOUNT;
```

Нусхани юклаш ва берилганлар базасини ўрнатиш, ҳамда очиш

Берилганлар базаси устида нормал ҳолатда амалларни бажариш учун аввал нусхани юклаб, сўнг берилганлар базасини ўрнатиш ва очиш лозим:

```
STARTUP;
```

ББ юклатишда чегараланган режимни ўрнатиш

Фақат администраторлар таркибига ишлашга рухсат этилган чегараланган режимда нусхани юклатишингиз, берилганлар базасини ўрнатишингиз ва очишингиз мумкин.

Бундай режим қўйидаги вазифалардан бирини бажаришда лозим бўлади:

- индексларни қайта тузишдек, ББ тузилмаси билан ишлаганда
- Ббдаги берилганларни экспорт ва импорт қилишда
- SQL*Loader дан фойдаланганда
- Фойдаланувчиларни берилганлардан фойдалинишини маън қилиш мақсадида

Ушбу жараён қўйидаги команда орқали амалга оширилади:

`STARTUP RESTRICT;`

Берилганлар базаси билан ишлашни тугатиш

Берилганлар базаси билан ишлашни қўйидаги режимлардан биридан фойдаланиб тугатиш мумкин:

- [NORMAL](#) режимда берилганлар базаси билан ишлашни тугатиш
- [IMMEDIATE](#) режимда берилганлар базаси билан ишлашни тугатиш
- [TRANSACTIONAL](#) режимда берилганлар базаси билан ишлашни тугатиш
- [ABORT](#) режимда берилганлар базаси билан ишлашни тугати

Берилганлар базаси билан ишлашни тугатиш учун SQL*Plus муҳитидан SHUTDOWN командасини беринг. Ушбу жараён кечаётганда ББ уланишга уринаётган фойдаланувчилар қўйидаги хабар оладилар:

ORA-01090: shutdown in progress - connection is not permitted

Берилганлар базаси ва нусха билан ишлашни тугатиш учун, аввал SYSOPER ёки SYSDBA бўлиб ББ уланинг.

NORMAL режимда берилганлар базаси билан ишлашни тугатиш

Normal режимда берилганлар базаси билан ишлашни тугатишда:

- Янги уланишга йўл қўйилмайди.
- Берилганлар базаси билан ишлашни тугатишдан олдин, Oracle барча уланган жорий фойдаланувчиларни ББ узишини кутади.
- Навбатдаги ББ юклатишда тиклаш процедуралари талаб этилмайди.

Normal режимда берилганлар базаси билан ишлашни тугатиш учун қўйидаги командани беринг:

`SHUTDOWN NORMAL;`

IMMEDIATE режимида берилганлар базаси билан ишлашни

тугатиш

Immediate режимида берилганлар базаси билан ишлашни тугатишда:

- Тугатилмаган транзакциялар қайтарилади.
- Oracle уланган фойдаланувчиларни ББ узишини кутмайди; Oracle ошкормас тарзда актив транзакцияларни қайтаради ва барча уланган фойдаланувчиларни ББ узади.

IMMEDIATE режимида берилганлар базаси билан ишлашни тугатиш учун қўйидаги командани беринг:

`SHUTDOWN IMMEDIATE;`

TRANSACTIONAL режимида берилганлар базаси билан ишлашни

тугатиш

Актив транзакциялар ўз ишини якунлашини кутиб, сўнг ББ билан ишлашни тугатмоқчи бўлсангиз қўйидаги командадан фойдаланинг:

`SHUTDOWN TRANSACTIONAL;`

ABORT режимида берилганлар базаси билан ишлашни тугатиш

Abort режимида берилганлар базаси билан ишлашни тугатиш, қўйидаги натижаларга олиб келади:

- Oracle томонидан бажарилаётган жорий мижознинг SQL жумласи дарҳол тугатилади.
- Commit қилинмаган транзакциялар қайтарилмайди.
- Oracle ошкормас тарзда барча уланган фойдаланувчиларни ББ узади.

ABORT режимида берилганлар базаси билан ишлашни тугатиш учун қўйидаги командадан фойдаланинг:

`SHUTDOWN ABORT;`

Параметрлар файлларидан фойдаланиш

Параметрлар файлларидан фойдаланиш усулларини келтирамыз:

- [Содда параметрлар файли](#)
- [Бир нечта параметрлар файллари](#)

Нусхани юклатиш учун Oracle параметрлар файлини ўқиши лозим. Унда нусханинг параметрлари рўйхати сақланади. Одатда ушбу файл INIT.ORA ёки INITsid.ORA, (бу ерда *sid* операцион тизимни аниқлайди).

Содда параметрлар файли

Содда параметрлар файли (INIT.ORA ёки INITsid.ORA) Oracle дистрибутив тўпламида мавжуд бўлади. Аммо сиз Oracle тизими билан маълум вақт ишлагандан сўнг, параметрлар файлидаги қийматларни ўзгартиришни хоҳлаб қолишингиз мумкин.

Бир нечта параметрлар файллари

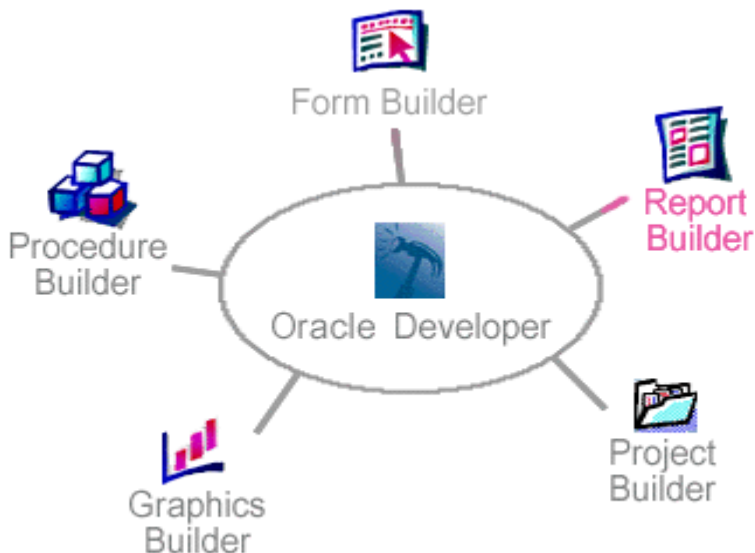
Ҳар бир Oracle ББ камида битта параметрлар файлига эга. Бунда берилганлар базасини аниқловчи DB_NAME ва CONTROL_FILES каби параметрлар берилади. Бир дона берилганлар базаси учун бир нечта параметрлар файлларига ҳам эга бўлиши мумкин. Масалан ҳар хил ҳолатларда ББ турли усулларда оптимал бошқариш учун бир нечта параметрлар файлларига эга бўлишингиз мумкин.

20-§ Oracle Developer

Oracle Developer Тезкор Амалий дастурлар тўплами ҳисобланиб, «приложения»-лар (амалий дастур) қуриш учун асосий қурол бўлиб ҳисобланади. Oracle Developer берилганлар базасини тасвирлаш учун жуда қулай муҳит ҳисобланади. Бу муҳитда сиз:

- **Берилганлар базаси учун юқори сифатли, чиройли интерфейсларни тез яратиш;**
- **Хажми жиҳатидан катта, яхши бошқариладиган дастурлар тузишингиз мумкин.**

Oracle Developerнинг **Form Builder, Report Builder** и **Graphics Builder** каби асосий маҳсулотлари ёрдамида сиз юқори сифатли график, интеграллашган интерфейслар тузишингиз мумкин.

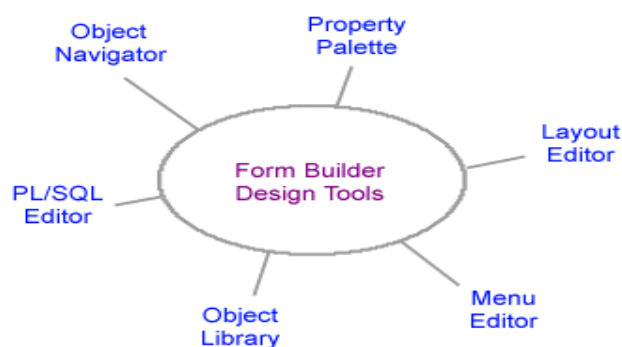


Булардан ташқари, Oracle Developer визуал интерфейсни қўлайдиган қўшимча қурилмаларни ҳам ўз ичига олади. Бу қурилмалар, жараёнларни автоматлаштириш ва бошқариш мақсадида «приложения»-лар қуриш учун мўлжалланган ва улар қуйидагилар: **Project Builder, Procedure Builder, Schema Builder, Query Builder** ва **Translation Builder**.

21-§ Form Builder.

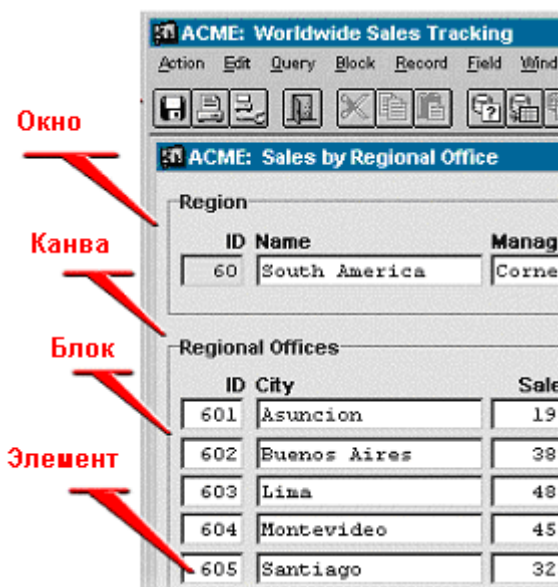
Form Builder - бу мижоз сервер «приложения»-ларини тузиш учун кучли ва қулай дастурлаш қурилмасидир.

Form Builder дан фойдаланиб сиз берилганлар базасид сақланадиган ахборотни



фойдаланувчи ишлатиши учун «приложения» яратишингиз мумкин. Охирги фойдаланувчи сизнинг «приложения»-нгиздан базадаги ахборотни кўриш, янги ахборот киритиш, ўзгартириш ва бу киритилган ёки ўзгартирилган ахборотни берилганлар базасига ёзиш учун фойдаланади. Form Builder - Oracle Developer нинг қурилмаларидан бири бўлиб, у Oracle Developer нинг бошқа қурилмаларини ишлашини тўлиқ қўллаб қувватлайди. Формалар билан ишлашда икки типдаги файллар фарқланади – булар *.fmb ва *.fmх кенгайтмалари

файллардир. Form Builder да «приложения» *.fmb кенгайтмали файл кўринишда сақланади, лекин форма ишга туширилганда *.fmb файли *.fmх файли(бажарилувчи файл)га автоматик равишда генерацияланади, ва фойдаланувчига фойдаланиши учун тавсия қилинади. *.fmх файллар Forms Runtime орқали ишлатилади. Form Builder да приложениялар турли синфларнинг объектлари ёрдамида курилади(тузилади). Буларга: модуллар, блоклар, элементлар ва бошқалар киради.



Дастурлаш жараёнининг асосий босқичлари.

Form Builder да оддий «приложения» куриш(тузиш) тўрт босқичли жараёни ўз ичига олади. Булар:

1. Янги форма модулини тузиш;
2. Берилганлар базаси жадвалларидан сўров олиш учун блоклар тузиш;
3. Канвада шу блок элементларини жойлаштириш;
4. «Приложения»-ни компиляциялаш, ишга тушириш ва текшириб кўриш.

Форма модулини тузиш

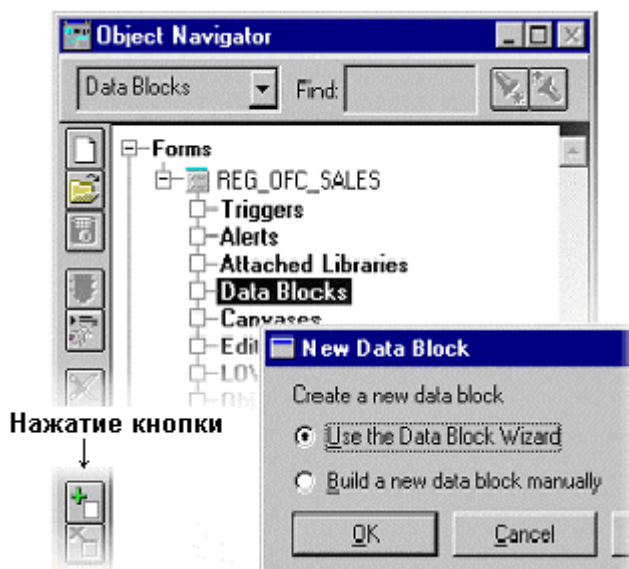
Form Builderда «приложения» тузишда биринчи кадам - бу янги форма модулини яратиш. Модуль – бу Form Builder нинг энг юкори сатҳли объекти. Хар бир модуль ўз навбатида қуйи сатҳли бир неча объектлардан тузилган. Сиз модулни уч хил усул билан тузишингиз мумкин:

- Welcome to the Form Builder (**Form Builderra хуш келибсиз**) мулоқот ойнаси ёрдамида. Бу мулоқот ойнаси Form Builder ни ишга туширганда пайдо бўлади.
- Object Navigator (**Объектларни бошқарувчи**) ёрдамида. Бунинг учун **Object Navigator** дан «Forms» ни танлаб,  «Яратиш» тугмаси босилади.
- Менюдан **File->New->Form** ни танлаб.

Берилганлар блоки тузувчисини(мастерини) чақириш

Янги форма модулини тузгандан сўнг, навбатдаги қадам берилганлар базаси жадвалларига асосланган берилганлар блокини тузиш. Берилганлар блоки – бу берилганлар базасидаги жадвал ёки кўриниш билан боғлиқ бўлган ёки процедуралар тўпламини кўрсатадиган мантиқий тузилма. Сиз берилганлар блокини тузишда жадвал ёки кўриниш устунларини интерфейс объектлари билан боғлайсиз. Бунинг учун **Data Block Wizard** (*Берилганлар Блоки Тузувчиси*) ишлатилади. **Берилганлар Блоки Тузувчисини** чақиришнинг уч хил усули мавжуд:

- **Welcome to Form Builder** мулоқот ойнаси ёрдамида;
- **Object Navigator** ёрдамида;
- **menu** ёрдамида. (Меню. Юқоридаги асосий менюдан **Tools** бўлими танланиб, меню остидан – **Data Block Wizard** танланади, ёки бу бўлимни сичқонча ўнг тугмасини босиш билан пайдо бўладиган контекстли менюдан танлаш мумкин).



Берилганлар Блоки Тузувчиси сизга «приложения»-нгизда ишлатиладиган берилганлар блокини тез ва осон тузиш ёки ўзгартириш имконини беради. Бу мулоқот ойнаси бир неча саҳифаларни ўз ичига олган бўлиб, унда сиз берилганлар манбааси типи (жадвал ёки сакланадиган процедура), жадвал устунлари, ташкил этаётган блоклар, блоклараро муносабатларни (агар керак булса) танлашингиз керак бўлади.

Блокнинг асосий тузилмасини тузгандан(қургандан) сўнг, сиз блокда берилганлар базаси элементларининг визуал кўринишини тузишни бошлашингиз мумкин. Бу ишни **Layout Wizard(Макет тузувчиси)** ёрдамида тез ва осон бажаришингиз мумкин.

Макет тузувчисини қўллаш

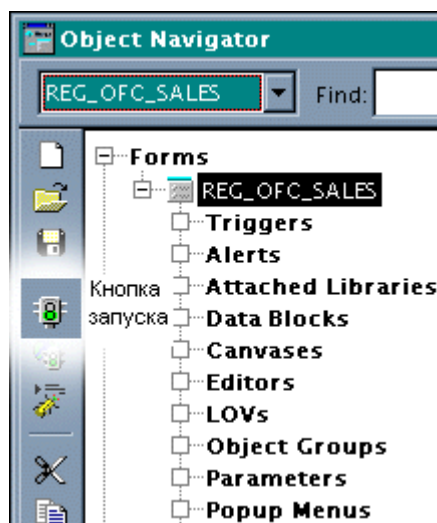
Макет тузувчиси блокдаги базанинг ҳар бир элементлари учун интерфейс объектларини яратади ва уни сиз олдиндан берган канва рамкаси ичида жойлаштиради. Интерфейс объектлари (Items – элементлар)-фойдаланувчи учун ахборотни акслантиради ва унга «приложения» билан ишлашга имконият яратади. Канва – бу орқа план объекти бўлиб, унда интерфейс объектларини ва графикавий элементларни жойлаштиришингиз мумкин.

Макет тузувчиси мулоқот ойнаси ўзига қатор саҳифаларни ўз ичига оладики, унда сиз канва типи ва номини кўрсатишингиз, блок элементларини

жойлаштиришингиз, блок номини ва канвада кўрсатилиши керак бўлган хоҳлаган элементни беришингиз мумкин, ҳамда интерфейс элементларининг номи, типи ва ўлчамини, шу билан бирга стилни(форма ёки табуляр) ҳамда сатрлар сони ва улар орасидаги масофани беришингиз мумкин.

Приложенияни ишлатиш

Блокни тузиб ва унинг элементларини канвада жойлаштиргандан сўнг, сиз «приложения»-ни ишлатишингиз ва текшириб кўришингиз мумкин.



Приложенияни ишлатганингиздан сўнг Form Builder берилганлар базасидан берилганларни автоматик чиқармайди(кўрсатмайди), уларни кўриш учун берилганлар базасидан сўровни бажариш керак, бунинг учун инструментлар панелидан **Execute Query**  тугмасини босиш керак бўлади.

Хўш кейинчи?

Приложения имкониятларини кенгайтириш, берилганларни нисбатан тулиқ намоён қилиш учун Form Builder бир-бири билан боғланган(яъни берилганлар манбааси бўлмиш бир блок бош жадвал, кейинги блок эса бўйсинувчи жадвал) берилганларни бир канвада икки ва ундан ортиқ блокларда қуриш имкониятини беради.

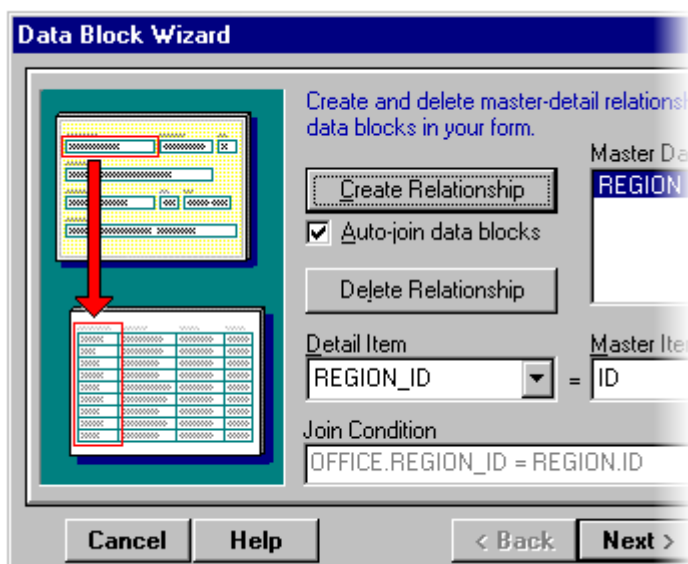
Бундай приложениялар master-detail applications деб, бу блоклар орасидаги боғланиш(муносабат) эса - master-detail relationship деб номланади. Бу муносабатлар берилганлар базасидаги таблицалар орасидаги бирламчи калит ҳамда иккиламчи калит алоқаси(муносабати) негизиди ётади.

REGION table		OFFICE table		
ID	NAME	ID	CITY	SALES
10	Europe	504	San Francisco	415
20	North America	505	Seattle	566
30	South America	601	Asuncion	197
		602	Buenos Aires	389
		603	Lima	483
		604	Montevideo	451

Region ID	60	Name	South America
Manager	Cornejo		

Office ID	City	Weekly Sales
601	Asuncion	197
602	Buenos Aires	389
603	Lima	483
604	Montevideo	451

Шу каби приложениялар тузиш учун янги блок тузиш вақтида **Master-Detail** дан фойдаланиш кифоя:



Шундан сўнг Form Builder автоматик тарзда блоклар орасида боғланишларни кўяди.

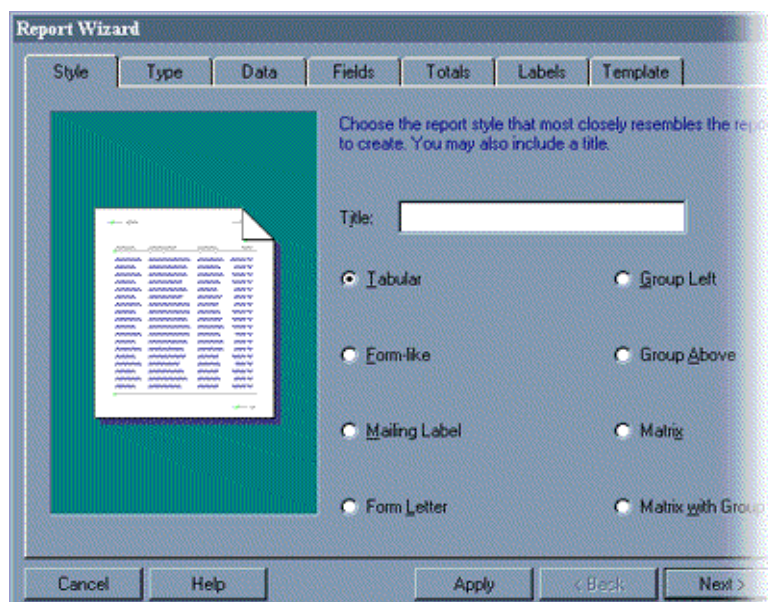
22-§ Report Builder

Report Builder берилганлар базасидан ҳисобот олиш учун ишлатиладиган махсус дастур ҳисобланиб, керакли дизайнда, турли хил форматда чоп этиш учун ишлатилади. Report Builder ни ишлатар экансиз, сиз гуруҳланган, жадвал кўринишига эга бўлган ва бошқа хил ихтиёрий форматдаги ҳисобот чиқаришингиз мумкин. Report Builder янги ҳисобот тузишда ҳисоботни *.rdf кенгайтмасидаги файл кўринишда хотирада сақлайди. Агар хоҳласангиз Reports Runtime да ишлатиш учун *.rdf кенгайтмасидаги файлни генерацияланган *.rpt кенгайтмасидаги файл кўринишида хотирада сақлашингиз мумкин.

Ҳисобот тузишнинг асосий босқичлари.

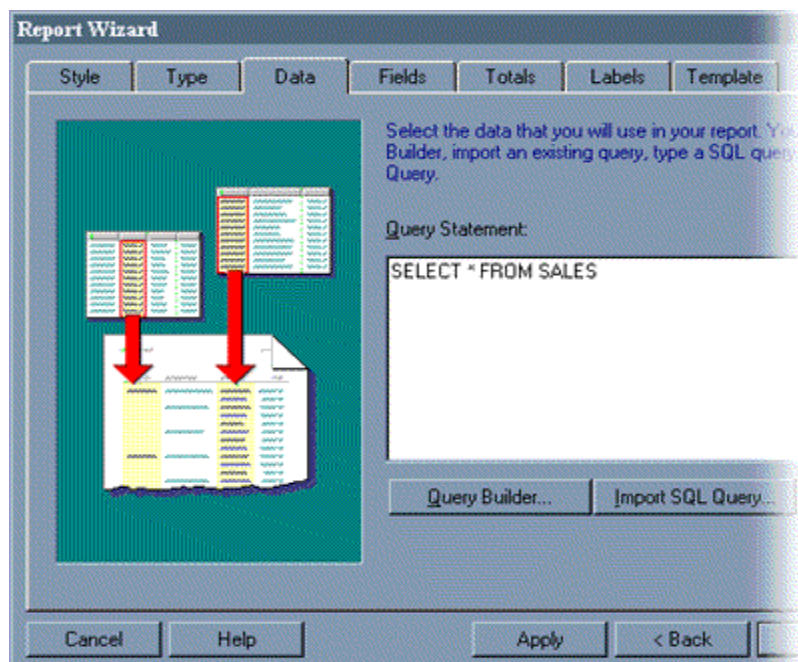
Тез ва осон ҳисоботлар тузиш учун сиз **Report Wizard** (Ҳисоботлар тузувчиси)дан фойдаланишингиз мумкин. **Report Wizard** босқичма-босқич сизнинг ишларингизни йўналтириб туради. Сўнгра у объектларни боғлаб, сизга ҳисоботни тайёр кўринишда беради. Ҳисобот тузувчи мулоқот ойнаси бир нечта саҳифани ўз ичига олади:

1) Ҳисобот шаклини танлаш. Ҳисобот шакли унинг асосий тузилмасини белгилайди. Сиз саккиз шаклдан ихтиёрий биттасини танлашингиз мумкин.



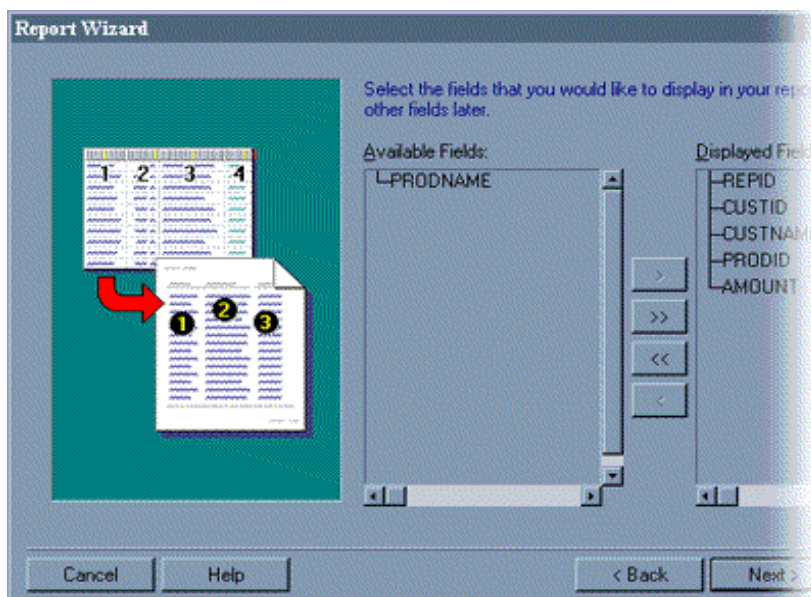
2) Сўровни аниқлаш. Сиз ҳисобот учун берилганлар асосида сўров берасиз. Сўров – бу берилганлар базасидан берилганларни қайтарадиган SQL тилидаги

ифода. Сиз сўровни қўлда ёки Query Builderдан фойдаланиб тузишингиз



мумкин.

3) Майдонларни танлаш. Хисобот майдонлари сўровингизга боғлиқ ҳолда сиз танлаган жадвал устунлари билан боғлиқ бўлади. Ҳисобот олишда сиз жадвалнинг ҳамма устунларини ёки айрим яъни керакли бўлган устунларини



танлашингиз мумкин.

4) Жамини(натижани) аниқлаш. Жами – бу устун ёзувлари орасидаги ҳисоблашлар (масалан: сумма(йиғинди), ўрта қиймат, максимум, минимум). Бу функцияларнинг ҳар бири учун мос тугма мавжуд, шу тугмани босганда

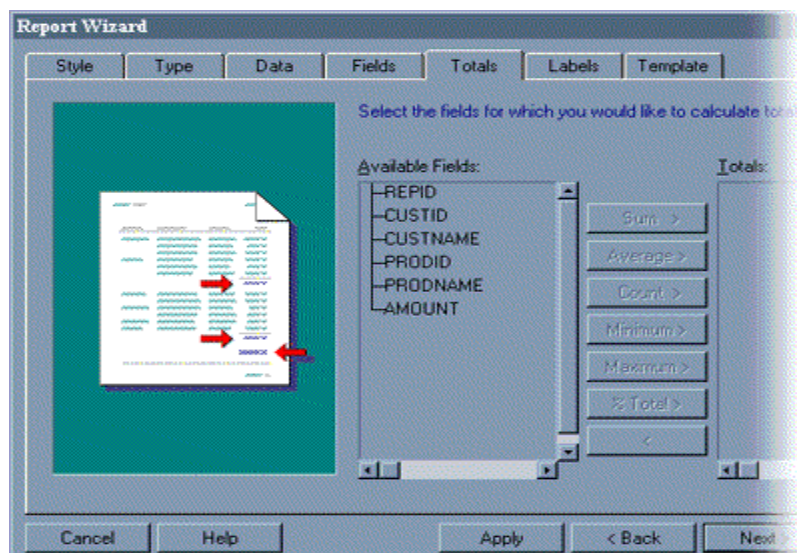
ҳисоботда

керакли

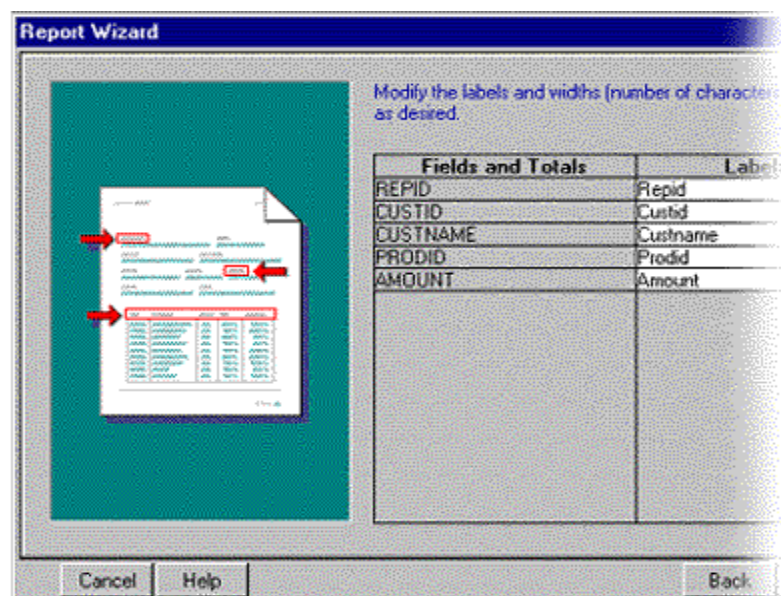
элемент

пайдо

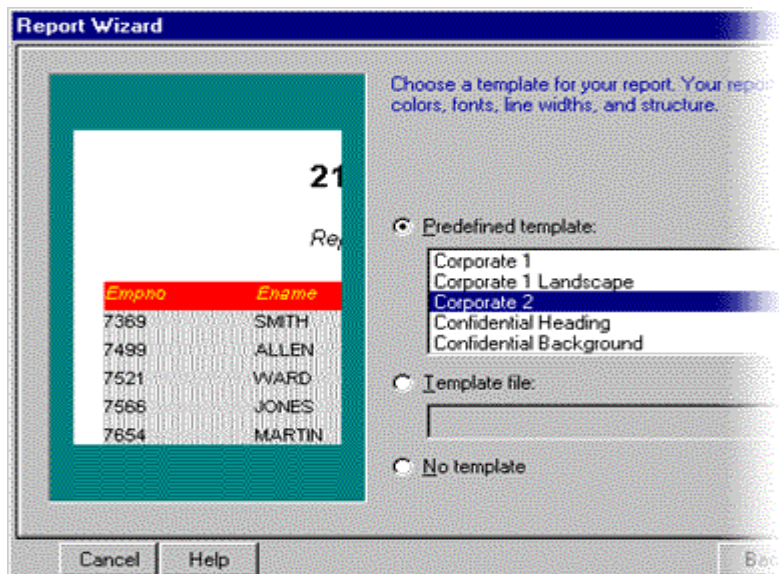
бўлади:



5) Нишонларни аниқлаш. Нишон – бу ҳисоботда ҳар бир майдон учун тузиладиган сарлавҳа. Report Builder нишонларни устун номи билан аниқлайди, лекин сиз уни ўзингиз хоҳлаган ном билан ўзгартиришингиз мумкин.



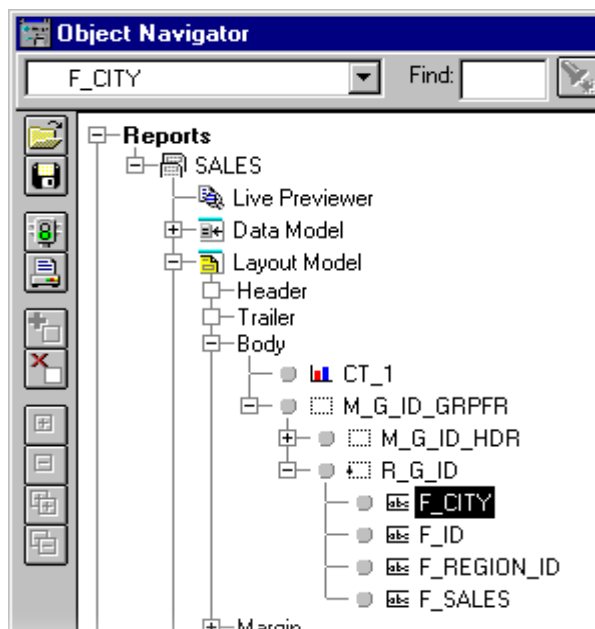
6) Қолип(шаблон)ни танлаш. Қолиплар – олдиндан аниқланган форматлар бўлиб, унга сиз кетма-кетлик ва моҳирона кўриниш бериб ўз ҳисоботингизда уни қўллашингиз мумкин. Сиз қолипни стандарт қолиплар тўпламидан олишингиз, ёки шахсан ўзингиз тузган (*.TDF форматдаги) қолипни ишлатишингиз мумкин, ёхуд умуман қолипдан фойдаланмаслигингиз мумкин.



Сўнгра Finish тугмасини босиш керак. Report Builder ҳисобот ва унинг барча объектларини ўзи лойиҳалаб, ишлатади ва экранда намоён қилади. Бу ерда ҳам сиз ҳисобот устида иш олиб боришингиз мумкин.

Object Navigator

Агар сиз ҳисобот устида иш олиб боришни давом эттиришни хоҳласангиз, бу учун Report Builder сизга қулай қурилма Object Navigator ни тавсия қилади. У ҳисобот ва унинг объектларининг жойлашишини аниқлаш ва улар билан ишлаш

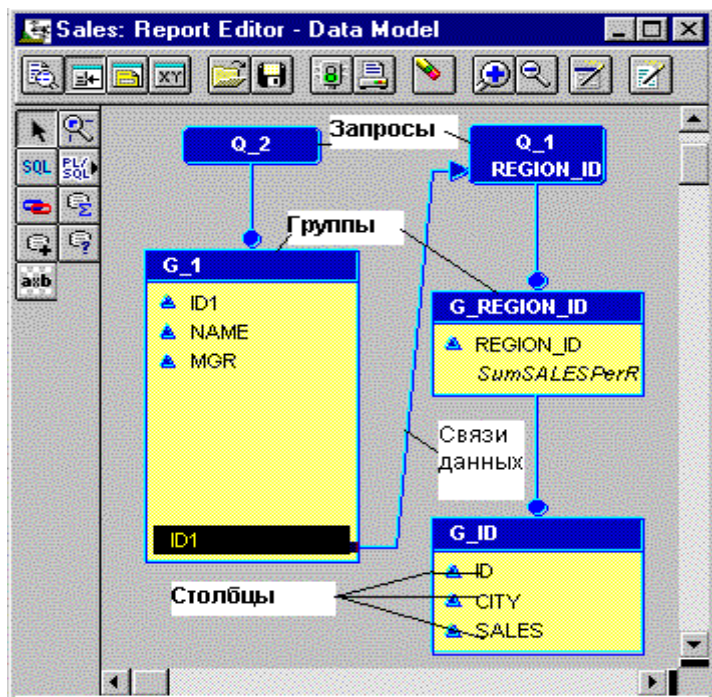


учун юқори сатхли излаш қурилмасига эга.

Ишнинг навбатдаги босқичи ва ҳисоботни ўзгартириш икки модел босқичида олиб борилади. Булар: берилганлар модели(data model) ва макет модели (layout model).

Data Model

Берилганлар модели ҳисоботда чиқариладиган берилганларни аниқлайдиган объектлардан иборат. Берилганлар модели объектлари: сўровлар, устунлар, гуруҳлар, параметрлар (фойдаланувчи томонидан ҳисоботни ишлатилиш олдидан берилладиган ўзгарувчилар) ва берилганлар боғлиқлиги (бош ва бўйсинувчи жадваллар орасидаги муносабати)ни ўз ичига олади:



Layout Model

Макет модели берилганларнинг ва ҳисоботнинг бошқа объектларининг пайдо бўлиши ва жойлашишини аниқлайди.



Модел макети объеклари қуйидагиларни ўз ичига олади:

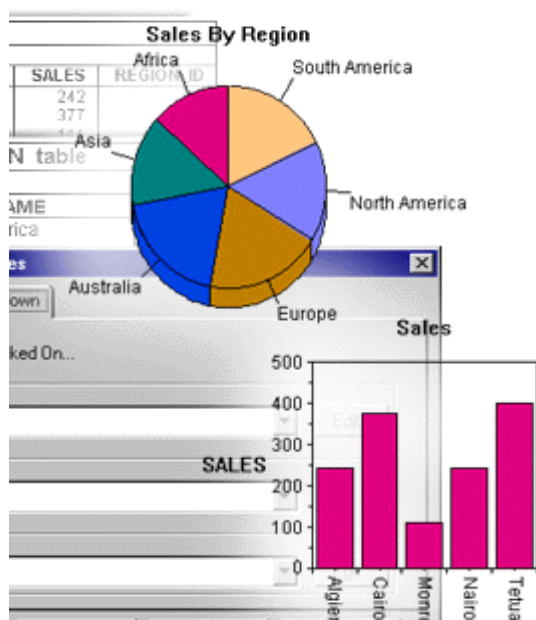
- 1) Такрорланадиган рамкалар. Улар бир гуруҳ чегарасида қайтариладиган берилганлар сатрини кўрсатиш учун ишлатилади.
- 2) Рамкалар. Улар макет объекти тартибини сақлаш ва ишлатиш вақтида рамкалар такрорланганда ёпилиши олдини олиш учун ишлатилади.
- 3) Майдон. Улар ҳисоботда чиқадиган устунларни аниқлайди (масалан, валюта ва сана(число) формати).
- 4) Шаблон(қолип) объеклари. Бу объектлар ҳар доим ҳисоботни ишлатганда пайдо бўлади. Бу объектларга майдонлар нишони (кўрсатгичи), график объеклари мисол бўла олади.
- 5) OLE2 (Object Linking and Embedding, V2) объеклари – бу OLE2 приложения объеклари (масалан, ҳужжатлар, расмлар).

Улар ҳам статик, ҳам динамик бўлиши мумкин (яъни, улар ҳисоботнинг берилганларини акс эттириши мумкин).

- 6) Тугмалар(Кнопки) – бу объектлар online ҳисоботида ишлатиш мумкинлигини. Бу тугмаларни босганда видео клиплар, расмлар, мусиқий файллар ёки PL/SQL дастурлари ишга тушади.

23-§ Graphics Builder

Graphics Builder берилганларни визуаллаштириш ва интерактив график «приложения»-ларини яратиш учун хизмат қилади.



Graphics Builder ёрдамида тузилган приложения, **display** деб номланади. Display приложенияларда ишлатиладиган барча бирликларни(компоненталарни) ўз ичига олади, масалан берилганлар манбаасини аниқлаш, визуал ва мантиқий элементлар). Хар бир Display график объектларни акслантирадиган макетни ўз ичига олади. Макет бир ёки ундан юқори сатҳлар(босқичлар)ни ўз ичига олади. Хар бир сатҳ алохида визуал элементларни ўз ичига олади. М: диаграммалар, графиклар ва матнлар. Приложения

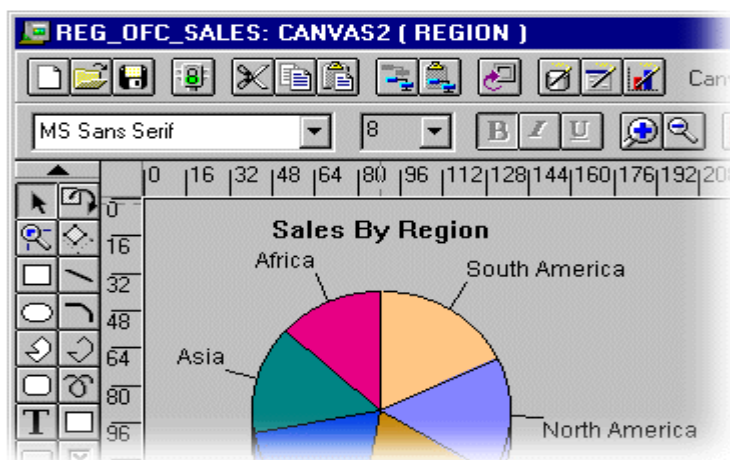
таркибига кирувчи сўровлар ва PL/SQL конструкциялари ҳам display нинг бир қисми ҳисобланади. Display *.ogd кенгайтмали файл кўринишда сақланади.

Object Navigator.

Display ни ташкил этувчи барча объектлар, Object Navigator да иерархик акслантирилади. Object Navigator да сиз displayда объектлар қандай ишлатилишини кўришингиз мумкин. Object Navigatorда объектлар хусусиятларини (хоссаларини) қуриш ёки қайта ишлаш учун фойдаланишингиз мумкин; бунинг учун объект тасвири устига икки марта сичқончани чиккилатишингиз кифоя.

Layout Editor

Макет мухарририда сиз график объектларини тузиш ва қайта ишлашингиз мумкин. Graphics Builder расм чизиш ва таҳрирлаш ҳамда турли график форматлардаги файлларни импорт ва экспорт қилиш учун барча имкониятларни яратади. Инструментлар панелида тез-тез фойдаланиладиган буйруқлардан фойдаланиш мумкин.



Display тузишнинг асосий босқичлари.

Graphics Builder ни ишга туширганда Display автоматик тузилади. Кейинроқ сиз янги display тузишингиз мумкин, Object Navigator да Display ни танлаб, сўнгра «Create» («Яратиш»)  тугмасини босиш орқали.

Янги диаграмма(графика) тузишда инструментлар панелидаги «Chart»  қурилмасидан фойдаланишингиз мумкин. Макетнинг бўш қисмига диаграмма ўлчовигача тузинг. Ҳосил бўлган мулоқот ойнасида диаграммада тасвирланадиган берилганларни аниқлашингиз керак. Агар сизнинг display сўровларни ўз ичига олмаса Graphics Builder автоматик тарзда сизга берилганлар манбаасини сизга таклиф қилади. Сўров хусусиятларини Query property (Сўров хусусиятлари) саҳифасида аниқланг ва берилганларни кўриш учун «Execute» тугмасини босинг. Сўнгра диаграммани турини танлашингиз керак бўлади.

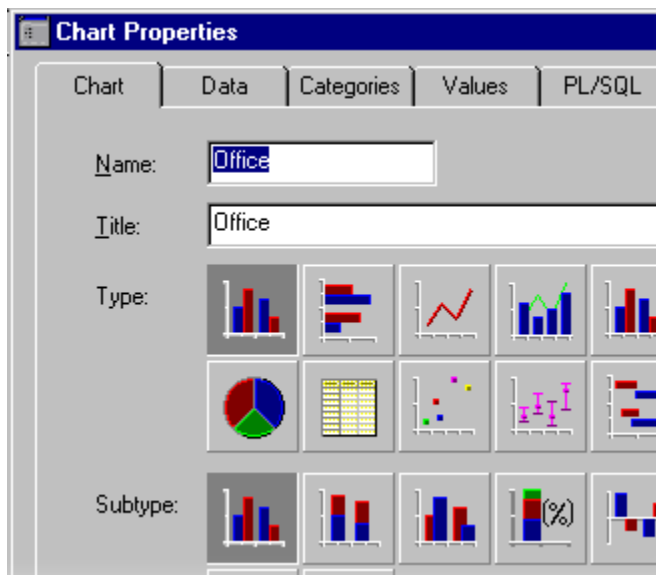


Chart хусусиятлари саҳифаси кўпгина диаграмма хусусиятларини ўзгартиришга имконият яратади. Керакли типни танлагандан сўнг, диаграмманинг(графика) ҳар бир координата ўқи

учун сўровдан устунларни аниқланг. Диаграмма, берилганларнинг динамик комбинацияси ҳисобланади ва ўзгаришлар янгиланиб, акслантирилади.

Формада график объектларни қўллаш.

График объектларни формада қўллашнинг икки усули мавжуд. Chart Wizard (Диаграмма тузувчиси) орқали, ёки диаграммани ўзингиз қўлда тузиб, уни формага боғлаб қўйишингиз мумкин. Диаграммани қўлда бирлаштиришда OG.PLL кутубхонаси таркибига кирувчи бир қатор қурилган процедурани ишлатиш ва Form Builder да мос триггер учун кодлаштириш талаб қилинади.

Chart Wizardни қўллаш.

Сиз Form Builder да оддий диаграмма тузишда Chart Wizardдан фойдаланишингиз мумкин.

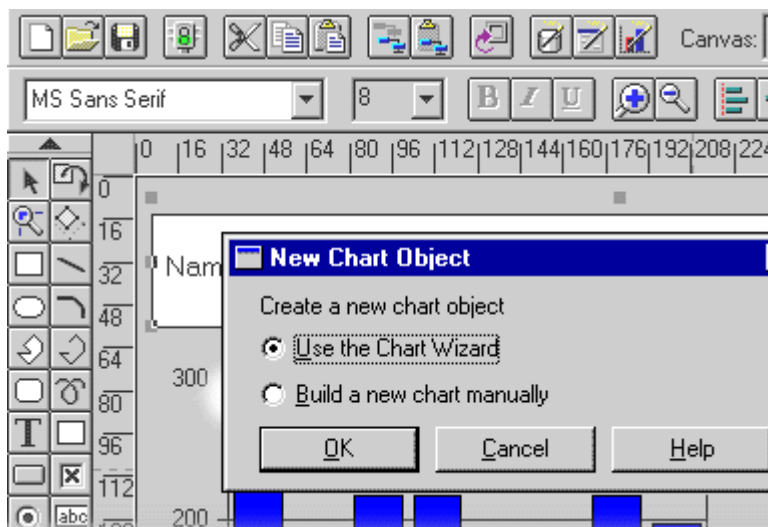


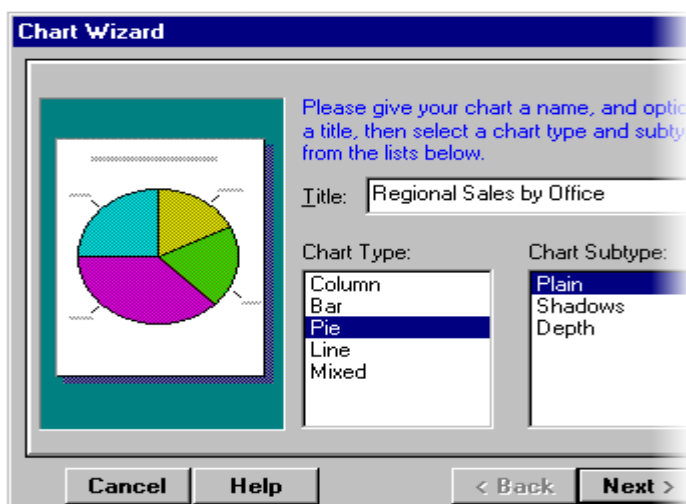
Chart Wizard диаграмма тузиш жараёни ҳар бир босқичида сизга ёрдамлашиб туради. Chart Wizard ёрдамида диаграмма тузиш жараёни беш босқичда амалга оширилади:

1. Янги график элемент тузиш. Бунинг учун Form Builder макет

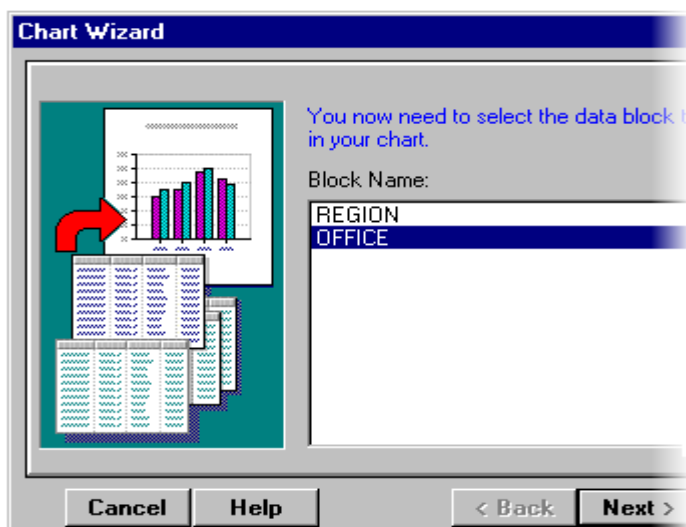
муҳарририда (Layout Editor)  тугмасини босиш керак.

Сўнгра бу график объектни канвада қўйилиши керак бўлган соҳасига чўзиш керак (Chart Wizard автоматик пайдо бўлади).

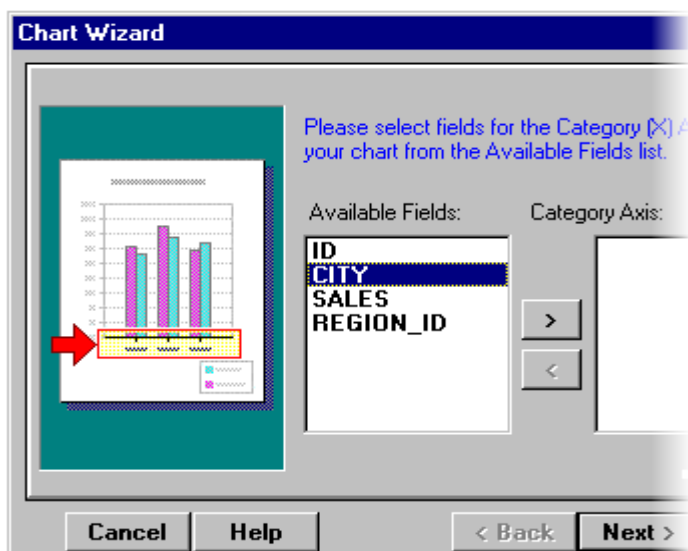
2. Диаграмма сарлавҳаси, тури ва тур остини танлаш. Бу саҳифада диаграмма тузувчиси ёрдамида сарлавҳа ва диаграмма турлари орасидан тегишли(мос) турни танланг. Сўнг тур ости саҳифасидан шу турга мансуб бўлган диаграмма турини танлашингиз мумкин.



3. Диаграмманинг берилганлари манбаасини аниқлаш. Бу саҳифада диаграмманинг берилганлари манбаасини ишлатиш учун тегишли блокни танлашингиз мумкин.



4. Диаграмма X ва Y ўқларини аниқлаш. Бу саҳифада сиз X ўқи ва Y ўқи бўйича аниқланадиган берилганлар блоки элементларини



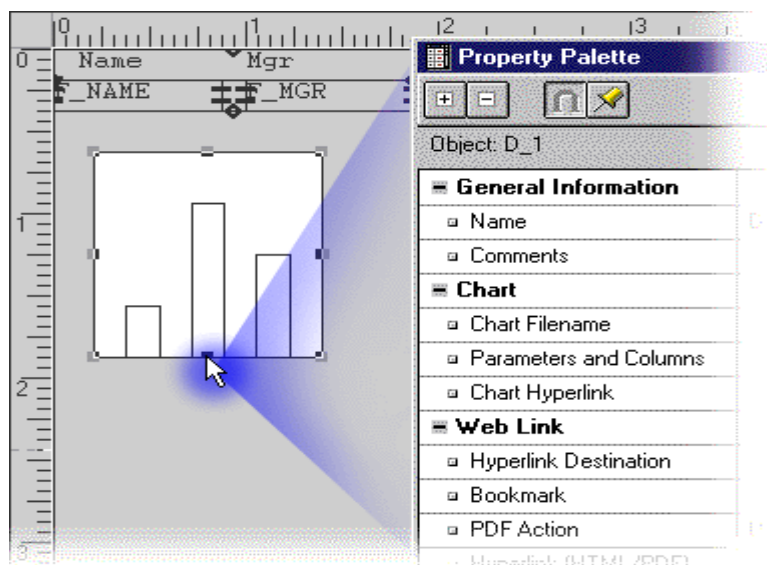
5. Диаграмма сақланадиган файл номини аниқлаш. Бу саҳифада сиз тузган диаграмма қандай ном билан сақланиши керак бўлса, шу файл номини ёзиш керак.

Хисоботларда график объектларни ишлатиш.

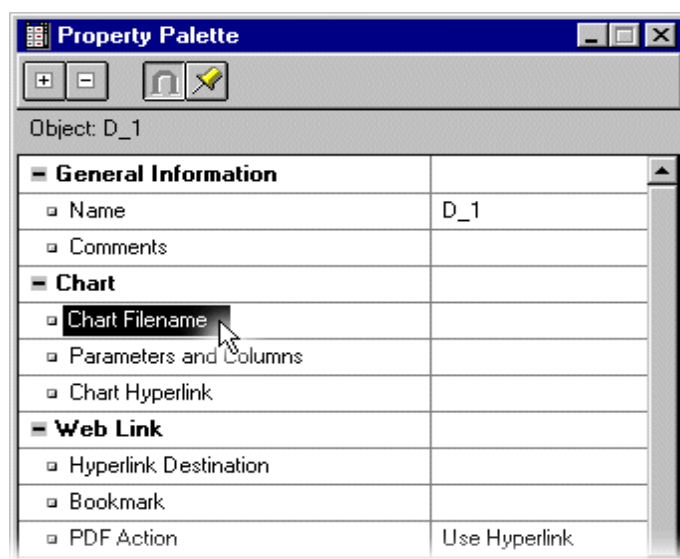
Сиз график объектлар ёрдамида ҳисоботларингизда берилганларни сезиларли даражада яққол намоён қилишингиз мумкин. Формада бўлганидек, бу ерда ҳам график объектларни қўллашни икки усули мавжуд. Chart Wizard (Диаграмма тузувчиси) ёрдамида ёки ўзингиз мустақил диаграмма тузишингиз ва уни ҳисоботга улашингиз мумкин. Chart Wizard – ҳисоботларда график объектларни қуриш учун жуда содда ва қулай йўл бўлиб, юқорида қайд этилган барча босқичлардан ўтиш керак бўлади.

Олдиндан тайёр бўлган график объектларни ҳисоботда ишлатиш учун Report Builder Макет муҳарририда график соҳасини қўлда тузиб ва унинг хусусиятларини Property Palette

(хусусиятлар панели)да беришингиз мумкин.



Бу ерда сиз қуйидагиларни қилишингиз зарур: боғланадиган файл номини ва унга бўлган йўлни(адресини) беришингиз, берилганлар узатиладиган сўров номини аниқлаш, display учун ишлатиладиган параметрларни аниқлаш; ҳисоботда гуруҳни аниқлашингиз, аниқланган сўровингизга тегишли устунлар ва параметрларни танлашингиз мумкин; координата ўқларини аниқлашингиз мумкин.



Санаб ўтилган хусусиятлардан ташқари, юқорида қайд килинган Oracle Developer маҳсулотлари ёрдамида Web дизайнда (саҳифани безашда) приложения ва ҳисобот тузиш мумкин. Oracle Developer ёрдамида тузилган ихтиёрий файлни html форматида сақлаш мумкин.

Oracle маълумотлар базаси бошқариш тизими фани бўйича намунавий тест саволлари

1. Oracle Сервери - бу ...

- A. барча жавоблар тўғри
- B. oracle берилганлар базаси нусхасидан таркиб топган
- C. объект-реляцион берилганлар базасини бошқарув тизими
- D. oracle сервери нусхасидан таркиб топган

2. Oracle нусхаси деб нимага айтилади?

- A. фон жараёнлари ва хотира буферлари комбинациясига
- B. битта ва ундан кўп жадваллар фазосига
- C. берилганлар базасининг ахбороти учун фойдаланадиган хотира соҳасига
- D. битта ва ундан кўп назорат файллар мажмуасига

3. Сервер –

- A. берилганлар устида амалларни бажаради
- B. берилганларни таҳлил қилиб, экранда кўрсатади
- C. тақсимланган жараён муҳитларига эга дастурдир
- D. аҳолига хизмат кўрсатиш жараёнидир

4. Oracle -

- A. очик турдаги тизимдир
- B. ёпик турдаги тизимдир

- C. ҳисоблаш тизими
- D. тўғри жавоб берилмаган

5. Қандай тузилмага эга берилганлар базасига тақсимланган берилганлар базаси дейилади?

- A. берилганлар яхлитлиги таъминланган, ҳар хил жойларда жойлашган (тармоқга уланган) берилганлар базаларини мантиқан битта берилганлар базаси сифатида қараш имконияти мавжуд
- B. маълумотлари ёпиқ турдаги тизимда жойлашган
- C. берилганлар яхлитлиги таъминланмаган
- D. тўғри жавоб берилмаган

6. SQL дастурлаш тили бўлиб, ...

- A. берилганлар базасини аниқлаш ва бошқариш учун хизмат қилади
- B. берилганлар базаси нусхасидан таркиб топган
- C. маълумотларни классик ва замонавий моделларини яратади
- D. тўғри жавоб берилмаган

7. Қайси жавобда SQL командалари берилган?

- A. DDL командалари ва DML командалари
- B. DCL командалари ва DPL командалари
- C. DPL командалари ва DML командалари
- D. DBL командалари ва DPL командалари

8. Oracle берилганлар базаси қандай тузилмалардан иборат?

- A. жисмоний ва мантиқий
- B. математик ва мантиқий
- C. жисмоний ва юридик
- D. ҳуқуқий ва мантиқий

9. Oracle берилганлар базасининг жисмоний тузилмаси нималарни ўз ичига олади?

- A. берилганлар базасини ташкил қилувчи операцион тизим файллари
- B. берилганлар базасини ташкил қилувчи операцион тизим каталогларини
- C. битта ва ундан кўп назорат файллари
- D. тўғри жавоб берилмаган

10. Oracle сервери -

- A. жараёни берилганлар базасининг сервери ва мижозлар орасида ташкил қилишга имконият яратади
- B. маълумотларни классик ва замонавий моделларини яратади
- C. берилганлар устида амалларни бажаради
- D. берилганларни таҳлил қилиб, экранда кўрсатади

11. Oracle берилганлар базаси қандай турдаги файллардан иборат?

- A. барча жавоблар тўғри
- B. иккита ва ундан кўп "redo log" файлларидан
- C. битта ва ундан кўп назорат файлларидан
- D. битта ва ундан кўп берилганлар файлларидан

12. Oracle берилганлар базаси мантиқий тузилмаси нималардан таркиб топган?

- A. битта ва ундан кўп жадваллар фазоси ва берилганлар базаси схемаси
- B. берилганлар базасини ташкил қилувчи операцион тизим каталоглари
- C. берилганлар базасини ташкил қилувчи операцион тизим файллари
- D. тўғри жавоб берилмаган

13. Схема объектлари қандай тузилмалардан иборат?

- A. барча жавоблар тўғри

- B. жадваллар, кўринишлар, синонимлар
- C. кетма-кетликлар, сақланадиган процедуралар, индекслар
- D. кластерлар ва берилганлар базасининг алоқалари

14. Oracle нусхаси икки турдаги жараёнларга эга. Улар қайсилар?

- A. фойдаланувчи жараёнлари ва Oracle жараёнлари
- B. ишлаб чиқариш жараёнлари ва Oracle жараёнлари
- C. фойдаланувчи жараёнлари ва ишлаб чиқариш жараёнлари
- D. фақат битта фойдаланувчи жараёнига эга

15. Net8 - ...

- A. стандарт алоқа протоколларига Oracle нинг интерфейси бўлиб,
компьютерлар орасидаги берилганлар алмашинувини таъминлайди
- B. фон жараёнлари ва хотира буферлари комбинацияси
- C. дастурлаш тили
- D. тўғри жавоб берилмаган

16. Жадвалий фазолар -

- A. мантиқий тузилмаларни бир гуруҳга йиғиш учун ишлатилади
- B. жисмоний тузилмаларни бир гуруҳга йиғиш учун ишлатилади
- C. мантиқий ва жисмоний тузилмаларни бир гуруҳга йиғиш учун
ишлатилади
- D. тўғри жавоб берилмаган

17. Жадвалий фазолар қандай ҳолатларда бўлиши мумкин?

- A. актив (кириш мумкин) ёки автоном (кириш мумкин эмас)
- B. фақат актив (кириш мумкин)
- C. фақат автоном (кириш мумкин эмас)
- D. барча жавоблар тўғри

18.(System Global Area) (SGA) нима ?

- A. Тизимнинг глобал соҳаси
- B. Тизимнинг устун соҳаси
- C. Тизимнинг йопиқ соҳаси
- D. Тизимнинг йакка соҳаси

19.Қайд журнали буферидаги маълумотлар қайд журнали файлига ёзилиб, берилганлар базасини тиклаш лозим бўлганда ишлатилади?

- A. -«online»
- B. -«хеш»
- C. -«бўш»
- D. -«глобал»

20.....– бу махсус жумлалар билан боғланган хотира номидир?

- A. курсор
- B. умумий пул
- C. қайд журнали буфери
- D. берилганлар базасининг буферлари

21.Program Global Area (PGA) бу...

- A. Программа глобал соҳаси
- B. Тизимнинг глобал соҳаси
- C. Тизимнинг устун соҳаси
- D. Программа устун соҳаси

22.Oracle сервери икки турдаги жараёнларга эга...

- A. фойдаланувчи жараёнлари ва Oracle жараёнлари
- B. программа глобал соҳаси ва Тизимнинг глобал соҳаси

- C. сервер ва мижоз
- D. очик ва йопик

23.Oracle нусхасини нима ташкил қилади

- A. SGA ва фон жараёнлари мажмуаси
- B. PGA ва фон жараёнлари мажмуаси
- C. фон жараёнлари мажмуаси
- D. SGA ва PGA мажмуаси

24.DB_WRITER_PROCESSES – инициализация параметри нимани аниқлайди

- A. DBWn жараёнлар сонини аниқлайди
- B. DBWn жараёнлар ёзувини аниқлайди
- C. DBWn жараёнлар тизимини аниқлайди
- D. DBWn жараёнлар НОЛЛларни аниқлайди

25.Берилганлар базасининг ёзувчиси бу...

- A. Database Writer
- B. LGWR
- C. SGA
- D. PGA

26.Берилганлар базаси хеш буферидаги ўзгартирилган блокларни берилганлар файлларига ёзади...

- A. Database Writer
- B. LGWR
- C. SGA
- D. PGA

27..... қайд журналини дискга ёзади. Қайд журналининг берилганлари SGA – нинг қайд журнали буфери ишлаб чиқилади.

- A. LGWR
- B. SGA
- C. PGA
- D. Database Writer

28.“online” қайд журнали файлларидан (агар улар тўлган бўлса) нусха олиб архивда сақлайди бу ?

- A. Архиваторъ
- B. Тикловчи
- C. Диспетчер
- D. Қулфлаш

29.Таксимланган берилганлар базасида, тизим ёки турнинг ишдан чиқиши муносабати билан юзага келган тақсимланган транзакция муаммосини ҳал қилиш учун ишлатилади?

- A. Тикловчиъ
- B. Архиватор
- C. Диспетчер
- D. Қулфлаш

30.Жисмоний берилганлар базасининг тузилмаси ... дан иборат.

- A. берилганлар файллари, тиклаш журнали файллари, назорат файллари
- B. Тиклаш, индекс, вақтинчалик сегментлар
- C. Математик ва мантиқий файллар
- D. Тиклаш ва назорат каталоглари

31.Берилганлар файлларига берилган характеристикалардан қай бири хато?

- A. Берилганлар файли фақат битта берилганлар базаси билан боғлиқ бўлмайди
- B. Берилганлар файли фақат битта берилганлар базаси билан боғлиқ бўлади
- C. Берилганлар базасининг файллари автоматик тарзда, берилганлар базаси тўлиб кетганда, кенгайиш хусусиятига эга бўлиши мумкин.
- D. Битта ёки ундан кўп берилганлар файллари жадвал фазоси деб номланувчи мантиқий бирлик сақловчисини ташкил этади.

32.SGA – да сақланаётган маълумотлар бир нечта турдаги хотира тузилмаларига бўлинади, булар:

- A. берилганлар базасининг буферлари, қайд журнали буфери, умумий пул (shared pool)
- B. Фиксерланган ва фиксерланмаган буфферлар
- C. Актив ва автоном буфферлар
- D. Тўғри жавоб берилмаган

33.... ўзгартирилган, ҳамда ўзгартирилмаган блокларни ўз таркибида сақлайди

- A. хеш буфери
- B. Тиклаш сегенти
- C. Берилганлар базаси
- D. Қайд журнали буфери

34.SGA қайд журнали буфери қандай вазифани бажаради?

- A. Берилганлар базасида вужудга келган ўзгаришларни сақлайди
- B. Бу махсус жумлалар билан боғланган хотира

- C. Махсус хотира буфери
- D. «online» қайд журнали файлига ёзилганларни ўчиради

35.SGA қайд журнали буфери қандай ўлчовга эга?

- A. Статик (ўзгармас)
- B. Динамик
- C. Махсус ҳолатларда статик, оддий ҳолатларда динамик
- D. Мантиқий

36.Қандай тузилмага эга берилганлар базасига тақсимланган берилганлар базаси дейилади?

- A. берилганлар яхлитлиги таъминланган, ҳар хил жойларда жойлашган (тармоқга уланган) берилганлар базаларини мантиқан битта берилганлар базаси сифатида қараш имконияти мавжуд
- B. маълумотлари ёпиқ турдаги тизимда жойлашган
- C. берилганлар яхлитлиги таъминланмаган
- D. тўғри жавоб берилмаган

37.Қайд журнали буферидаги маълумотлар қачон ишлатилади?

- A. Берилганлар базасини тиклаш лозим бўлганда
- B. Берилганлар базасини тиклаш иложи бўлмаганда
- C. Берилганлар базасида вужудга келган ўзгаришларни бекор қилишда
- D. «online» қайд журнали файлига ёзилганларни ўчиришда

38.Oracle нусхаси бу -

- A. фон жараёнлари ва хотира буферлари комбинацияси
- B. битта ва ундан кўп жадваллар фазоси
- C. берилганлар базасининг ахбороти учун фойдаланадиган хотира соҳаси
- D. битта ва ундан кўп назорат файллар мажмуаси

39.Курсор – ...

- A. бу махсус жумлалар билан боғланган хотира номи
- B. берилганлар базасида вужудга келган ўзгаришларни сақлайди
- C. сичқонча кўрсатгичи
- D. махсус хотира буфери

40.Қайд журнали буферидаги маълумотлар қаерга ёзилади?

- A. «online» қайд журнали файлига
- B. Махсус хотира буферига
- C. Асосий хотирага
- D. Берилганлар базаси блокларига

41.Программа глобал сохаси хотира буфери бўлиб, ...

- A. Сервер жараёни учун берилганлар ва назорат ахборотларни сақлайди
- B. Берилганлар базасида вужудга келган ўзгаришларни сақлайди
- C. «online» қайд журнали файлига ёзилганларни ўчиради
- D. Махсус жумлалар билан боғланган хотира қисми ҳисобланади

42.Net8 – бу

- A. стандарт алоқа протоколларига Oracle нинг интерфейси бўлиб,
компьютерлар орасидаги берилганлар алмашинувини таъминлайди
- B. фон жараёнлари ва хотира буферлари комбинацияси
- C. дастурлаш тили
- D. тўғри жавоб берилмаган

43.Oracle сервери икки турдаги жараёнларга эга:

- A. фойдаланувчи жараёнлари ва Oracle жараёнлари
- B. локал жараёнлар ва глобал жараёнлар
- C. мантикий жараёнлари ва Oracle жараёнлари.

D. PL SQL жараёнлари ва Oracle жараёнлари.

44.Фойдаланувчи жараёнлари сервер жараёнлари билан қайси дастури ёрдамида алоқа қилади?

- A. интерфейс дастури ёрдамида
- B. Oracle дастури ёрдамида
- C. MY SQL дастури ёрдамида
- D. таржимон дастурлари ёрдамида

45.Oracle нусхаси ишлатиши мумкин бўлган фон жараёнлари номларини кўрсатинг.

- A. барча жавоблар тўғри
- B. Dnnn, LCKn, SNPn, QMNn
- C. SMON, PMON, ARCH, RECO
- D. DBWn, LGWR, CKPT

46.SGA ва фон жараёнлари мажмуаси ... ташкил қилади.

- A. Oracle нусхасини
- B. Сервер жараёнини
- C. Oracle жараёнини
- D. Жисмоний берилганлар базасини

47.Берилганлар базасининг ёзувчиси қандай вазифани амалга оширади?

- A. берилганлар базаси хеш буферидаги ўзгартирилган блокларни берилганлар файлларига ёзади
- B. Сервер жараёни учун берилганлар ва назорат ахборотларни сақлайди
- C. Берилганлар базасида вужудга келган ўзгаришларни сақлайди
- D. Қайд журналинни дискга ёзади.

48.Журнал ёзувчисининг вазифаси?

- A. қайд журналини дискга ёзади
- B. берилганлар базаси хеш буферидаги ўзгартирилган блокларни берилганлар файлларига ёзади
- C. сервер жараёни учун берилганлар ва назорат ахборотларни сақлайди
- D. берилганлар базасида вужудга келган ўзгаришларни сақлайди

49.Маълум вақт интервалларида SGA – даги берилганлар базасининг барча ўзгартирилган буферлари берилганлар файлларига DBWN томонидан ёзилиш ҳодисаси ... дейилади.

- A. текширув нуқтаси
- B. Жараён монитори
- C. Архиватор
- D. Тикловчи

50.... - “online” қайд журнали файлларида (агар улар тўлган бўлса) нусха олиб архивда сақлайди.

- A. архиватор
- B. жараён монитори
- C. текширув нуқтаси
- D. тикловчи

51.Транзакция «commit» қилинганда нима рўй беради?

- A. ўзгаришлари фиксирланади
- B. ўзгаришларни кўради
- C. қайтиш сегментида ёзилади
- D. ўзгаришларни тўғри кетма-кетликда акс этиради

52.Қулфлаш механизми ниям?

- A. берилганларга параллел равишда мурожаат қилувчи

фойдаланувчилар орасидаги ўзаро бузувчи таъсир кўрсатиш олдини олишдир.

- B. максимал самарадорликга эга бўлиш учун берилганлар базаси тизимининг аксарият фойдаланувчиларидан юқори савияда ишлашда.
- C. берилганлар базасидаги берилганларни ёзувчилари, айнан ўша берилганларни ўқувчиларини кутмаслиги кафолатланиши
- D. кўп-фойдаланувчи тизимнинг берилганларга кириш конкуренцияси максималлаштирилиши

53.Қулфлаш берилганлар базасининг мақсади нима?

- A. Биргаликдалиги, яхлитлик
- B. Биргаликдалиги, яхлитлик ва параллеллик
- C. Яхлитлик ва параллеллик
- D. Биргаликдалиги

54.Тақсимланган жараёнлар нима учун ишлатилади?

- A. биттадан кўп бўлган процессорларни, жараённи ўзаро боғланган топшириқлар тўпламига бўлиш учун ишлатилади
- B. жараённи ўзаро боғланган топшириқлар тўпламига бўлиш учун ишлатилади
- C. фойдаланувчи кўраётган ёки ўзгартираётган берилганлар, у ўз ишини берилганлар билан тугатгунга қадар ўзгартирмаслиги учун
- D. берилганларга параллел киришни назорат қилиш учун

55.... Oracle дастурий таъминотини ишлатади ва берилганларга параллел , умумий кириш қўйилган талаб функцияларини ишлов беради.

- A. сервер қисми
- B. мижоз қисми

- C. тақсимланган берилганлар
- D. архитектура берилганлар

56.... тақсимланган берилганлар базасининг ҳар бир берилганлар базаси алоҳида, бошқа берилганлар базаларига боғлиқ бўлмаган равишда бошқариш мумкинлиги тушунилади.

- A. тугун автономлиги
- B. location transparency
- C. тақсимланган берилганларни бошқаруви
- D. жойлашиш шаффофлиги

57.Бу ерда қандай амал бажарилляпти

SELECT * FROM [emp@sales](#);

- A. EMP жадвалга сўров бериш
- B. SALES жадвалга сўров бериш
- C. узоқланган берилганлар базасидаги жадвалга мурожаат
- D. [emp@sales](#) жадвалига сўров бериш

58.икки-фазавий «Commit» механизми нима?

- A. транзакция тақсимланган берилганлар базасининг барча тугунларида яхлит бирлик сифатида ё «Commit» қилинади ёки «Rollback» қилиниши
- B. транзакция тақсимланган берилганлар базасининг барча тугунларида яхлит бирлик сифатида «Rollback» қилиниши
- C. транзакция тақсимланган берилганлар базасининг барча тугунларида яхлит бирлик сифатида «Commit» қилиниши
- D. Тақсимланган берилганлар базаси тизимида локал фойдаланувчилар томонидан кўп сўров берадиган узоқланган жадвал учун нусха олиб қўйилиши

59.RECO фон жараёни қачон рўй беради?

- A. «Commit» узилиб қолса
- B. «Rollback» узилиб қолса
- C. Тизим ёки тур носозлигини автоматик тарзда тиклашда
- D. автомат тарзда носозлик тузатилганда тиклашда

60.Юклатиш ва ёпиш амалларини ким амалга оширади?

- A. Администратор
- B. Фойдаланувчи
- C. Oracle сервери
- D. Net8

61.Берилганлар базасини юклатиш ва уни бошқалар ишлатишга яроқли холга келтириш учун кадамлари...

- A. Берилганлар базасини ўрнатиш
Берилганлар базасини очиш
- B. Берилганлар базасини узиш (Dismount the database)
Oracle сервер нусхасини ёпиш
- C. Oracle сервери нусхасини ишга тушириш
Берилганлар базасини очиш
- D. Берилганлар базасини очиш
Oracle сервер нусхасини ёпиш

62.Схема ...

- A. берилганлар базаси объектларининг мантикий мажмуасидир
- B. фойдаланувчи номи/пароли комбинациясини тўғрилиги
- C. тизим махфийлиги ва берилганлар махфийлиги
- D. тўғри жавоб берилмаган

63.Тизим махфийлиги қуйидагиларни ўз ичига олади:...

- A. фойдаланувчи номи/пароли комбинациясини тўғрилиги;
фойдаланувчи схемаси объектлари учун ажратилган дискдаги жой микдори
фойдаланувчи учун ресурс чегараси
- B. фойдаланувчи берилганлар базага уланишга ҳуқуқи борлигини
фойдаланувчи қандай тизимий амалларни бажариши мумкинлигини
- C. берилганлар базасига кириш ва ундан фойдаланишни
- D. тизим даражасида назорат килишни

64.” фойдаланувчи SCOTT EMP жадвали устида SELECT ва INSERT жумлаларни бажариб билади, аммо DELETE жумласини бажаролмайди” шу жумла қачон ўринли бўлади?

- A. берилганлар махфийлигида
- B. ҳема ҳар бир объекти учун бажариладиган амалларни текширишда
- C. берилганлар базасини текшириш актив ҳолатдалигида
- D. ўринли эмас

65.Имтиёзлар, роллар, жойлашув параметрларини қўйиш ва квоталар нима учун ишлатилади?

- A. махфийлигини бошқариш учун
- B. фойдаланувчи берилганлар базага уланиши учун
- C. схема объектларига киришлари учун
- D. тизимий амалларни бажариш учун

66.Махфийлик домени нима?

- A. ҳамма жавоб ўринли
- B. сри иш вақти
- C. фойдаланувчига бериладиган жадвалий фазодаги квоталар

D. фойдаланувчи бажарилиши мумкин бўлган ҳатти-ҳаракатлар

67.... бу SQL жумлаларини бажариш ҳуқуқидир

- A. Имтиёзлар
- B. Жойлашув параметрларини қўйиш ва квоталар
- C. Аудит қилиш
- D. Мажбурият

68.Ўзингизни схемангизда жадвал очиш ҳуқуқи ...

- A. имтиёз
- B. мажбурият
- C. аудит қилиш
- D. ресурс чегаралари

69. EMP жадвалига қатор киритиш имтиёзини SCOTT деган фойдаланувчига ошкор тарзда бериш мумкинми?

- A. мумкин
- B. мумкин фақат имтиёзда
- C. мумкин эмас
- D. мумкин фақат ролда

70.Имтиёзларни динамик бошқариш...

- A. фойдаланувчилар гуруҳи имтиёзларини ўзгартирилиши лозим бўлса, фактаггина рол имтиёзларини ўзгартириш кифоя бўлади
- B. фойдаланувчиларга берилган ролларни танланган ҳолда ишлатиш учун яроқли ёки яроқсиз ҳолга келтириш мумкин, бу эса фойдаланувчиларга берилган имтиёзларни ихтиёрий ҳолатда назорат қилишга имкон беради
- C. берилганлар базаси дастурини шундай тузиш мумкинки,

фойдаланувчи дастурни ишлатишга уринганда танланган роллар автоматик тарзда яроқли ёки яроқсиз ҳолга келтирилади

- D. айнан бир хил имтиёзлар тўпламини кўп фойдаланувчиларга тайинлаш ўрнига, берилганлар базасининг администратори ўзаро боғланган фойдаланувчилар гуруҳига мўлжалланган имтиёзларни ролга тайинлаши мумкин

71.Имтиёзларни танлаб тайинлаш

- A. Фойдаланувчиларга берилган ролларни танланган ҳолда ишлатиш учун яроқли ёки яроқсиз ҳолга келтириш мумкин. Бу эса фойдаланувчиларга берилган имтиёзларни ихтиёрий ҳолатда назорат қилишга имкон беради
- B. Фойдаланувчилар гуруҳи имтиёзларини ўзгартирилиши лозим бўлса, фактаггина рол имтиёзларини ўзгартириш кифоя бўлади.
- C. Берилганлар базаси дастурини шундай тузиш мумкинки, фойдаланувчи дастурни ишлатишга ўринганда танланган роллар автоматик тарзда яроқли ёки яроқсиз ҳолга келтирилади
- D. Айнан бир хил имтиёзлар тўпламини кўп фойдаланувчиларга тайинлаш ўрнига, берилганлар базасининг администратори ўзаро боғланган фойдаланувчилар гуруҳига мўлжалланган имтиёзларни ролга тайинлаши мумкин

72.....-берилганлар базаси дастурини шундай тузиш мумкинки, фойдаланувчи дастурни ишлатишга ўринганда танланган роллар автоматик тарзда яроқли ёки яроқсиз ҳолга келтирилади

- A. Дастурлар (приложениялар) учун имтиёзлар
- B. Имтиёзларни динамик бошқариш
- C. Имтиёзлар тайинлаш жараёнини қисқартиради
- D. Имтиёзларни танлаб тайинлаш

73.DBA....

- A. берилганлар базасининг администратори
- B. берилганлар базаси дастурини шундай тузиш мумкинки,
фойдаланувчи дастурни ишлатишга ўринганда танланган роллар
автоматик тарзда яроқли ёки яроқсиз ҳолга келтирилади
- C. дастур ролига дастурни ишга тушириш учун зарур бўладиган барча
имтиёзлар
- D. ишчи жадвалий фазо

74.Вақтинчалик жадвалий фазо қачон ишлатилади?

- A. Фойдаланувчи вақтинчалик сегментларни очишни талаб қиладиган
SQL жумласини бажарса
- B. Профил фойдаланувчига тайинланмаса
- C. Фойдаланувчи ўрнатилиши мумкин бўлган параллел сессиялар сони
- D. Фойдаланувчи жадвал, индекс ёки кластер очмоқчи бўлиб, унда
уларни жисмонан сақлаш учун жадвалий фазо кўрсатилмаган бўлса

75.Имтиёз даражасида текширув...

- A. схема объектларини аниқ кўрсатилмаган ҳолда тизимий
имтиёзларни назорат қилиш
- B. берилганлар базасининг администратори
- C. фойдаланувчига боғлиқ бўлмаган равишда схема айрим объектларига
киришни назорат қилиш
- D. берилганлар базаси дастурини шундай тузиш мумкинки,
фойдаланувчи дастурни ишлатишга ўринганда танланган роллар
автоматик тарзда яроқли ёки яроқсиз ҳолга келтирилади

76.Фойдаланувчига боғлиқ бўлмаган равишда схема айрим

объектларига киришни назорат қилиш...

- A. схема объектлари даражасидаги текширув
- B. берилганлар базасини тиклаш
- C. жумла даражасидаги текширув
- D. имтиёз даражасида текширув

77.Берилганлар базасини хатолик содир бўлишидан олдинги вақтга қайтаришни талаб қилиши...

- A. фойдаланувчи хатолиги
- B. жумлани таҳлил қилиб ишлов бериш
- C. oracle rmon фон жараёни
- D. нусха бузилиши

78.Тўлиқ тиклашни амалга ошириш учун Oracle тизими бир нечта тузилмалари...

- A. қайд журнали, «rollback» сегменти, назорат файли, ва берилганлар базасини зарурий лахзагача орқага қайтариш
- B. «rollback» сегменти
- C. назорат файли, ва берилганлар базасини зарурий лахзагача орқага қайтариш
- D. қайд журнали

79.LGWR фон жараёни...

- A. берилганлар базасидаги барча «commit» қилинган ўзгартиришларни ёзади
- B. архивланган қайд журнали
- C. хотирадаги ўзгартирилган берилганлар базасидаги берилганларини сақлаш учун ишлатилади
- D. ишчи қайд журналида

80. Тўлиб кетган ишчи қайд журнали файлларини даврий тарзда қайтадан фойдаланиш учун, олдин улар архивлаш

- A. ARCHIVELOG» вазият
- B. «NOARCHIVELOG» вазият
- C. Назорат файллари
- D. Архивланган қайд журнали

81.....бу жадваллар, кўринишлар, индекслар ва шу каби объектлар бўлиб берилганлар базасидаги берилганларни сақлаш ёки уларга кириш учун ишлатилади

- A. Тузилмалар
- B. Амаллар
- C. Яхлитлик қоидалари
- D. Реляцион берилганлар базасини бошқарув тизими

82.....– бу содда тарзда аниқланган ҳаракат бўлиб, улар ердамида фойдаланувчилар берилганлар базасидаги тузилмалар ва берилганларни бошқарадилар

- A. Амаллар
- B. Тузилмалар
- C. Яхлитлик қоидалари
- D. Реляцион берилганлар базасини бошқарув тизими

83. ...қонун бўлиб, берилганлар базасидаги берилганлар ва тузилмалар устида бажарилиши мумкин бўлган амалларни аниқлайди

- A. Яхлитлик қоидалари

- B. Амаллар
- C. Тузилмалар
- D. Реляцион берилганлар базасини бошқарув тизими

84.Схема объектларига нималар киради

- A. Хаммаси тўғри
- B. берилганлар базаси боғланишлар
- C. сақланадиган процедуралар
- D. синонимлар, индекслар

85..... жадвалларга асосланган сақланадиган сўров деб тасаввур қилиш мумкин

- A. Кўриниш
- B. Кетма-кетликлар
- C. Синонимлар
- D. Дастур бирликлари

86..... жадвал, кўриниш, кетма-кетлик ёки бошқа дастур бирлиги учун ҳамроҳ бўлиб ҳисобланади.

- A. Синонимлар
- B. Кетма-кетликлар
- C. Кўриниш
- D. Дастур бирликлари

87._____ жадвал устунларини тўлдириш учун уникал бўлган сонлар кетма-кетлигини ишлаб чиқаради

- A. Кетма- кетликлар
- B. Кўриниш
- C. Синонимлар

D. Дастур бирликлари

88.Реляцион берилганлар базасини бошқарувчи тизим учун стандарт тил....

- A. SQL
- B. Oracle
- C. Acses
- D. PL/SQL

89.DML.....

- A. берилганларни бошқариш тили
- B. берилганларни аниқлаш тили
- C. тизимни назорат қилувчи жумлалари
- D. ички қўрилган SQL жумлалари

90. DDL...

- A. берилганларни аниқлаш тили
- B. тизимни назорат қилувчи жумлалари
- C. берилганларни бошқариш тили
- D. ички қўрилган SQL жумлалари

91.Сўров бериш, янги қиймат қўйиш, қийматни ўзгартириш, ва жадвал қаторларини ўчириш кандай амаллар?

- A. DML
- B. DDL
- C. SQL
- D. PL/SQL

92.COMMIT, ROLLBACK, ва SAVEPOINT кандай амаллар?

- A. Транзакцияни назорат қилувчи жумлалар
- B. Ички қурилган SQL жумлалари
- C. Сессияни назорат қилувчи жумлалари
- D. Тизимни назорат қилувчи жумлалари

93.ALTER SYSTEM кандай амаллар?

- A. Тизимни назорат қилувчи жумлалари
- B. Ички қурилган SQL жумлалари
- C. Сессияни назорат қилувчи жумлалари
- D. Транзакцияни назорат қилувчи жумлалар

94.OPEN, CLOSE, FETCH, ва EXECUTE кандай амаллар?

- A. Ички қурилган SQL жумлалари
- B. Транзакцияни назорат қилувчи жумлалар
- C. Сессияни назорат қилувчи жумлалари
- D. Тизимни назорат қилувчи жумлалари

95.ALTER SESSION ва SET ROLE кандай амаллар?

- A. Сессияни назорат қилувчи жумлалари
- B. Ички қурилган SQL жумлалари
- C. Транзакцияни назорат қилувчи жумлалар
- D. Тизимни назорат қилувчи жумлалари

96. _____ ягона фойдаланувчи томонидан бажариладиган битта ёки ундан кўп SQL жумлаларидан иборат ишнинг мантиқий бирлигидир?

- A. Транзакция
- B. Меткалар
- C. Сессияни назорат қилувчи жумлалари

D. Тизимни назорат қилувчи жумлалари

97. Келгусида оний нуқтадан белгиланган нуқтагача ‘rollback’

қилиш имконига эга бўлишингизни нима таъминлайди?

- A. rollback
- B. транзакция
- C. меткалар
- D. commit

98. _____ схема объектларини аниқлаш, сақлаш ва керак

бўлмаганда йўқотиш учун ишлатилади?

- A. DDL
- B. DML
- C. SQL
- D. Oracle

99. _____ берилганлар базасидаги берилганларни бошқаради?

- A. DML
- B. DDL
- C. SQL
- D. Oracle

100. _____ қилиш транзакциядаги барча SQL жумлалари

томонидан амалга оширилган ўзгаришларни фиксирлайди?

- A. Commit
- B. Rollback
- C. DDL
- D. DML

101. _____ транзакциядаги ишингизни ихтиёрий жойини белгилаб қўйиш?

- A. Оралиқ метка
- B. Savepoints
- C. «Commit»
- D. 1 ва 2

102. Олдинги ҳолатга қайтиш учун зарур бўладиган берилганларни вақтинча сақлаш учун ... дан фойдаланилади.

- A. тиклаш сегменти
- B. индекс сегменти
- C. вақтинчалик сегмент
- D. берилганлар сегменти

103. Жисмоний берилганлар базасининг тузилмаси ... дан иборат.

- A. берилганлар файллари, тиклаш журнали файллари, назорат файллари
- B. тиклаш, индекс, вақтинчалик сегментлар
- C. математик ва мантиқий файллар
- D. тиклаш ва назорат каталоглари

104. Берилганлар файлларига берилган характеристикалардан қай бири хато?

- A. Берилганлар файли фақат битта берилганлар базаси билан боғлиқ бўлмайди
- B. Берилганлар файли фақат битта берилганлар базаси билан боғлиқ бўлади
- C. Берилганлар базасининг файллари автоматик тарзда, берилганлар базаси тўлиб кетганда, кенгайиш хусусиятига эга бўлиши мумкин
- D. Битта ёки ундан кўп берилганлар файллари жадвал фазоси деб

номланувчи мантикий бирлик сақловчисини ташкил этади

105. SGA қайд журнали буфери қандай вазифани бажаради?

- A. Берилганлар базасида вужудга келган ўзгаришларни сақлайди
- B. Бу махсус жумлалар билан боғланган хотира
- C. Махсус хотира буфери
- D. «online» қайд журнали файлига ёзилганларни ўчиради

106. SGA қайд журнали буфери қандай ўлчовга эга?

- A. Статик (ўзгармас)
- B. Динамик
- C. Махсус ҳолатларда статик, оддий ҳолатларда динамик
- D. Мантикий

107. Қандай тузилмага эга берилганлар базасига тақсимланган берилганлар базаси дейилади?

- A. берилганлар яхлитлиги таъминланган, ҳар хил жойларда жойлашган (тармоқга уланган) берилганлар базаларини мантиқан битта берилганлар базаси сифатида қараш имконияти мавжуд
- B. маълумотлари ёпиқ турдаги тизимда жойлашган
- C. берилганлар яхлитлиги таъминланмаган
- D. тўғри жавоб берилмаган

108. Қайд журнали буферидаги маълумотлар қачон ишлатилади?

- A. Берилганлар базасини тиклаш лозим бўлганда
- B. Берилганлар базасини тиклаш иложи бўлмаганда
- C. Берилганлар базасида вужудга келган ўзгаришларни бекор қилишда
- D. «online» қайд журнали файлига ёзилганларни ўчиришда

109. Oracle нусхаси бу -

- A. фон жараёнлари ва хотира буферлари комбинацияси
- B. битта ва ундан кўп жадваллар фазоси
- C. берилганлар базасининг ахбороти учун фойдаланадиган хотира соҳаси
- D. битта ва ундан кўп назорат файллар мажмуаси

110. Курсор – ...

- A. Бу махсус жумлалар билан боғланган хотира номи
- B. Берилганлар базасида вужудга келган ўзгаришларни сақлайди
- C. Сичқонча кўрсатгичи
- D. Махсус хотира буфери

111. Қайд журнали буферидаги маълумотлар қаерга ёзилади?

- A. «online» қайд журнали файлига
- B. махсус хотира буферига
- C. асосий хотирага
- D. берилганлар базаси блокларига

112. Net8 – бу

- A. стандарт алоқа протоколларига Oracle нинг интерфейси бўлиб,
компьютерлар орасидаги берилганлар алмашинувини таъминлайди
- B. фон жараёнлари ва хотира буферлари комбинацияси
- C. дастурлаш тили
- D. тўғри жавоб берилмаган

113. Oracle сервери икки турдаги жараёнларга эга:

- A. фойдаланувчи жараёнлари ва Oracle жараёнлари
- B. локал жараёнлар ва глобал жараёнлар.
- C. мантикий жараёнлари ва Oracle жараёнлари.

D. PL SQL жараёнлари ва Oracle жараёнлари.

114. Фойдаланувчи жараёнлари сервер жараёнлари билан қайси дастури ёрдамида алоқа қилади?

- A. Интерфейс дастури ёрдамида
- B. Oracle дастури ёрдамида
- C. MY SQL дастури ёрдамида
- D. Таржимон дастурлари ёрдамида

115. Oracle нусхаси ишлатиши мумкин бўлган фон жараёнлари номларини кўрсатинг?

- A. Барча жавоблар тўғри
- B. Dnnn, LCKn, SNPn, QMNn
- C. SMON, PMON, ARCH, RECO
- D. DBWn, LGWR, CKPT

116. SGA ва фон жараёнлари мажмуаси ... ташкил қилади.

- A. oracle нусхасини
- B. сервер жараёнини
- C. oracle жараёнини
- D. жисмоний берилганлар базасини

117. Берилганлар базасининг ёзувчиси қандай вазифани амалга оширади?

- A. берилганлар базаси хеш буферидаги ўзгартирилган блокларни берилганлар файлларига ёзади
- B. сервер жараёни учун берилганлар ва назорат ахборотларни сақлайди
- C. берилганлар базасида вужудга келган ўзгаришларни сақлайди
- D. қайд журналини дискга ёзади

118. Журнал ёзувчисининг вазифаси?

- A. қайд журналини дискга ёзади
- B. берилганлар базаси хеш буферидаги ўзгартирилган блокларни берилганлар файлларига ёзади.
- C. сервер жараёни учун берилганлар ва назорат ахборотларни сақлайди
- D. берилганлар базасида вужудга келган ўзгаришларни сақлайди

119. ... - “online” қайд журнали файлларидан (агар улар тўлган бўлса) нусха олиб архивда сақлайди.

- A. Архиватор
- B. Жараён монитори
- C. Текширув нуқтаси
- D. Тикловчи

120. ... таксимланган берилганлар базасида, тизим ёки турнинг ишдан чиқиши муносабати билан юзага келган тақсимланган транзакция муаммосини ҳал қилиш учун ишлатилади.

- A. Тикловчи
- B. Диспетчер
- C. Иш навбати
- D. Навбат монитори

121. Тикловчи нима мақсадда ишлатилади?

- A. Таксимланган берилганлар базасида, тизим ёки турнинг ишдан чиқиши муносабати билан юзага келган тақсимланган транзакция муаммосини ҳал қилиш учун ишлатилади
- B. Фон жараёнлари ҳисобланиб, кўп-жараёнли сервер (multi-threaded server) конфигурацияси учун ишлатилади
- C. Oracle параллел серверида нусхалараро муносабат учун ишлатилади

D. “Online” қайд журнали файлларидан (агар улар тўлган бўлса) нусха олиб архивда сақлаш учун ишлатилади

122. ... жараёнлари (LCK0, . . . , LCK9) Oracle параллел серверида нусхалараро муносабат учун ишлатилади.

- A. Қулфлаш
- B. Диспетчер
- C. Тикловчи
- D. Иш навбати

123. Қулфлаш жараёнлари қандай ҳолатларда ишлатилади?

- A. Oracle параллел серверида нусхалараро муносабат учун ишлатилади
- B. Фон жараёнлари бўлиб ҳисобланиб, кўп-жараёнли сервер (multi-threaded server) конфигурацияси учун ишлатилади
- C. Таксимланган берилганлар базасида, тизим ёки турнинг ишдан чиқиши муносабати билан юзага келган тақсимланган транзакция муаммосини ҳал қилиш учун ишлатилади
- D. “Online” қайд журнали файлларидан (агар улар тўлган бўлса) нусха олиб архивда сақлаш учун ишлатилади

124. ... Oracle AQ опцияси хабар навбатларини назорат қилувчи фон жараёнидир.

- A. Навбат монитори
- B. Тиклаш
- C. Қулфлаш
- D. Иш навбати

125. Навбат монитори – бу

- A. Oracle AQ опцияси хабар навбатларини назорат қилувчи фон

жараёнидир

- B. Фон жараёнлари бўлиб ҳисобланиб, кўп-жараёнли сервер (multi-threaded server) конфигурацияси учун ишлатилади
- C. Таксимланган берилганлар базасида, тизим ёки турнинг ишдан чиқиши муносабати билан юзага келган тақсимланган транзакция муаммосини ҳал қилиш учун ишлатилади
- D. “Online” қайд журнали файлларидан (агар улар тўлган бўлса) нусха олиб архивда сақлаш учун ишлатилади

126. Фойдаланувчи жараёни ва сервер жараёни ўртасида алоқа ўрнатувчи механизм?

- A. Дастур интерфейси
- B. Интернет
- C. Компютер
- D. Берилганлар базаси

127. Ҳар бир берилганлар базаси, маъмурий вазифаларни бажариш учун камида нечта кишини талаб этади?

- A. 1
- B. 2
- C. 3
- D. 4

128. Oracle сервери ва дастурий ускуналарни ўрнатиш ва янгилаш, берилганлар базаси тизими талаб этадиган фазони жойлаштириш ва келгуси ривож талабларини режалаштириш вазифалари ким бажаради?

- A. Берилганлар базаси администратори
- B. Берилганлар базаси администратор фойдаланувчилари
- C. Берилганлар базасининг фақат битта фойдаланувчиси
- D. Берилганлар базасининг иккита фойдаланувчиси

129. Дастур ишлаб чиқувчилар ўзларининг дастурларини яратганларидан сўнг, берилганлар базасининг дастлабки сақлагич тузилмаларини (жадвалий фазолар) ташкил қилиш вазифалари ким бажаради

- A. Берилганлар базаси администратори
- B. Берилганлар базаси фойдаланувчилари
- C. Берилганлар базасининг фақат битта фойдаланувчиси
- D. Берилганлар базасининг иккита фойдаланувчиси

130. Фойдаланувчиларга ролларни тайинлаш ва тизим махфийлигини таъминлаш, oracle лицензияси шартларини кафолатлаш, фойдаланувчиларнинг берилганлар базасига кириш имкониятларини назорат қилиш вазифалари ким бажаради?

- A. Берилганлар базаси администратори
- B. Берилганлар базаси фойдаланувчилари
- C. Берилганлар базасининг фақат битта фойдаланувчиси
- D. Берилганлар базасининг иккита фойдаланувчиси

131. Oracle тизими нечта дастур администраторларига эга бўлади?

- A. Битта ёки ундан кўп
- B. Фақат битта
- C. Учта ёки ундан кўп
- D. Камида учта

132. Oracle нинг Net8 каби тармоқ билан ишлайдиган ва махсулотларини бошқариш учун масъул бўлишлари мумкин булар... ?

- A. тармоқ администраторлари
- A. дастур ишлаб чиқувчилар администратори

- B. хавфсизлигини таъминловчи администраторлар
- C. берилганлар базаси фойдаланувчилари

133. Берилганлар базасини яратишда автоматик тарзда иккита фойдаланувчи ташкил этилади ва уларга DBA роли тайинланади. Булар қайсилар?

- A. SYS , SYSTEM
- B. SYSTEM ва SYSDBA
- C. S YS ва SYSOPER
- D. SYSDBA ва SYSOPER

134. Фойдаланувчилар қайси калит сўз ёрдамида локал ёки узоқланган базаларга уланишлари лозим?

- A. CONNECT
- B. STARTUP
- C. SHUTDOWN
- D. ALTER DATABASE

135. ADMIN OPTION билан барча тизимий имтиёзлар ва OSOPER ролини ўз ичига оладиган CREATE DATABASE ва вақтга асосланган қайта тиклашни бажаришга имкон берадиган операцион тизим ролини топинг?

- A. OSDBA
- B. OSOPER
- C. SYS
- D. SYSOPER

136. OSDBA ролни актив ҳолатга келтириш муваффақиятсиз бўлса қайси ролни актив ҳолга келитиришга ўтади?

- A. OSOPER
- B. қайтадан OSDBA ролни актив ҳолатга келтиради
- C. уланиш содир бўлмайди
- D. grant

137. Агар иккала OSOPER ва OSDBA роллари билан бирин кетин боғланишга уриниш муваффақиятсиз бўлса уланишда қайси ролни актив қилади?

- A. уланиш содир бўлмайди
- B. қайтадан osdba ролни актив ҳолатга келтиради
- C. қайтадан osoper ролни актив ҳолатга келтиради
- D. нар иккала osoper ва osdba роллари билан

138. Қайси утилит ёрдамида пароллар файлини ташкил қилинади?

- A. ORAPWD
- B. PASSWORD
- C. EXCLUSIVE
- D. OSOPER

139. ORAPWD FILE=*filename* PASSWORD=*password* ENTRIES=*max_users*

ушбу мисол қандай натижа оланади?

- A. пароллар файлини ташкил қилинади
- B. инициализация параметрини PASSWORD ўрнатилади
- C. SYSDBA имтиёзи фойдаланувчига OSDBA орқали бажарилиши мумкин бўлган амалларни бажаришга имкон беради
- D. барча жавоблар тўғри

140. Пароллар файлига берилганлар базасининг администратори функциясини бажарувчи фойдаланувчиларни SQL ёрдамида қандай қўшилади?

- A. GRANT SYSDBA TO scott
GRANT SYSOPER TO scott
- B. ORAPWD SYSDBA TO scott;
ORAPWD SYSOPER TO scott;
- C. PASSWORD SYSDBA TO scott;
PASSWORD SYSOPER TO scott;
- D. GRANT SYSDBA TO scott;
ORAPWD SYSOPER TO scott;

141. CONNECT scott/tiger@acct.hq.com AS SYSDBA

Қандай жараён сўдир бўлади?

- A. имтиёзи бор фойдаланувчи берилганлар базасига уланади
- B. имтиёзсиз фойдаланувчи берилганлар базасига уланади
- C. имтиёзи бор фойдаланувчи берилганлар базасидан парол файли очади
- D. имтиёзсиз фойдаланувчи берилганлар базасидан парол файли очади

142. SYSDBA ёки SYSOPER тизимий имтиёзларни фойдаланувчилардан чақириб олиш учун қайси командасидан фойдаланилади?

- A. REVOKE
- B. GRANT
- C. ORAPWD
- D. ENTRIES

143. Oracle тизимида имтиёзланган фойдаланувчи сифатида махфий бўлмаган усулда уланиш учун қандай шартларни

учратасиз?

- A. барча жавоблар тўғри
- B. уланишингиз лозим бўлган сервер пароллар файлига эга бўлиши лозим
- C. сиз sysoper ёки sysdba тизимий имтиёзга эга бўлишингиз лозим
- D. сиз номингиз ва паролнингиз орқали уланишингиз лозим

144. SELECT * FROM product_component_version;

ушбу сўрўв нима учун ишлатилади?

- A. Oracle тизими ва унинг қайси компоненталаридан фойдаланаётганлигини кўриш
- B. Тизим фойдаланувчиларига тегишли ролларни тайинлаш
- C. Берилганлар базаси мантикий тузилмасини кўриш
- D. Берилганлар базасини яратиш ва очиш

145. Берилганлар базасини яратиш жараёни қайси амалларни ўз ичига олади?

- A. Барча жавоблар тўғри
- B. Янги берилганлар файлларини яратиш ёки мавжуд бўлган олдинги берилганлар файлларидан фойдаланиш
- C. Oracle талаб этадиган берилганлар базасидан фойдаланиш учун зарур бўладиган маълумот тузилмаларини (берилганлар лугати) яратиш
- D. Берилганлар базаси учун назорат ва redo log файлларини яратиш ва инициализация қилиш

146. Янги берилганлар базасини яратиш учун нималарга эга бўлишингиз лозим?

- A. Барча жавоблар тўғри
- B. Берилганлар базаси администраторининг тўлиқ вазифасини бажариш

учун зарур бўладиган операцион тизим имтиёзлари

- C. Oracle нусхасини ишлатиш учун зарур бўладиган етарли хотира
- D. Oracle тизими ишлаётган компьютер дискларида режалаштирилган берилганлар базасини сақлаш учун етарли сақлаш жойи

147. CREATE DATABASE test

DATAFILE 'test_system' SIZE 10M

LOGFILE GROUP 1 ('test_log1a', 'test_log1b') SIZE 500K,

GROUP 2 ('test_log2a', 'test_log2b') SIZE 500K;

Ушбу мисолда янги берилганлар базаси нима деб номланади?

- A. TEST
- B. TEST_SYSTEM
- C. TEST_LOG2A
- D. TEST_LOG2B

148. CREATE DATABASE test

DATAFILE 'test_system' SIZE 10M

LOGFILE GROUP 1 ('test_log1a', 'test_log1b') SIZE 500K,

GROUP 2 ('test_log2a', 'test_log2b') SIZE 500K;

Ушбу мисолда янги берилганлар базаси, ҳар бири иккита қанча ҳажмга эга бўлган қисмлардан таркиб топган иккита онлайн қайд журнали гуруҳларидан иборат?

- A. 500 KBayt
- B. 250 KB
- C. 100 KB
- D. 1000 KB

149. Берилганлар базаси номини қайси параметри тармоқ тузилмасидаги доменини (мантиқий жойлашуви) аниқлайди?

- A. DB_DOMAIN
- B. The DBNAME
- C. The DATAFILE
- D. The NAME

150. Қайси параметри орқали муайян компьютердаги берилганлар базасига уланадиган рақобатли сессиялар сонини чегаралаб қўйишингиз мумкин?

- A. LICENSE_MAX_SESSIONS
- B. LICENSEMAXUSERS
- C. DB_DOMAIN
- D. The DBNAME

151. Қайси параметри орқали берилганлар базасида яратилиши мумкин бўлган фойдаланувчилар сонини чегаралаб қўйишингиз мумкин?

- A. LICENSE_MAX_USERS
- B. DBDOMAIN
- C. The DB_NAME
- D. LICENSEMAXSESSIONS

Адабиётлар

1. **Кайт Т.** Oracle для профессионалов. Книга 1. Архитектура и основные особенности: Пер. с англ. — К.: 0 0 0 «ТИД ДС», 2003. — 662 с. ТК.
2. **Кайт Т.** Oracle для профессионалов. Книга 2. Расширение возможностей и защита: Пер. с англ. К.: 0 0 0 «ТИД ДС», 2003. 848 с.
3. **С.Фейерштейн Б.Прибыл.** Oracle PL/SQL. Для профессионалов. 6-е изд. — СПб.: Питер, 2015. — 1024 с.: ил. — (Серия «Бестселлеры О'Reilly»).
4. **В.С. Лаврентьев** Освоение SQL и PL/SQL Oracle. Лабораторные работы: учебное пособие.- М.:МИФИ, 2009.-105с.
5. **МакДональд Кониор.** Oracle PL/SQL для профессионалов: практические решения / Коннор МакДональд, Хаим Кац, Бек Кристофер и др. ; Пер. с англ. - СПб. : ООО «ДиаСофтЮП», 2005. — 560 с.
6. **Аруп Нанда, Стивен Фейерштейн** Oracle PL/SQL для администраторов баз данных. — Пер. с англ. — СПб: Символ-Плюс, 2008. — 496 с., ил.

Оглавление

SQL ORACLE	3
§1. ORACLE/SQL тилига кириш.	3
1.1. SQL тарихи.....	3
1.2. SQL стандартлари	3
1.3. SQL тилининг ижобий томонлари.	4
1.4. Ички қурилган SQL	5
1.5. Лексик келишувлар.....	5
2-§. SQL элементлари	5
2.1. Схема объектлари	6
2.2. Объектларни номлаш ва унинг қисмлари.....	6
2.3. Объектларга мурожаат қилиш қоидалари ва тавсиялар.	8
2.4. Литераллар.....	9
2.5. Берилганлар турлари.....	10
2.7. Берилганларни солиштириш қоидалари.....	13
2.8. Берилганларни турини ўзгартириш.	13
2.9. Бўш қиймат.....	14
2.10. Псевдоустунлар.	14
§ 3. Амаллар	15
3.1. Амаллар бажариш тартиби.....	16
3.2. Арифметик амалларга мисоллар.....	16
3.3. Белгили амалларга мисол.....	16
3.4. Солиштириш амаллари.....	17
3.5. Мантикий амаллар.....	18
3.6. Тўпламлар устида амаллар.....	18
§ 4. Функциялар	19
4.1. Сонли функциялар	19
4.2. Белгили функциялар.....	20
4.3. Сана функциялари	22
4.4. Турларни ўзгартириш функциялари.....	23
4.5. Бошқа функциялар.....	24
4.6. Гуруҳли функциялар.....	25
§ 5. Формат маскалари.....	26
5.1. NUMBER форматининг элементлари.....	26
5.2. DATE форматининг элементлари.	27
§ 6. Ифода (expr параметри).....	29
6.1. Формалар.....	29
6.2. Ифодани декодлаштириш.	31
6.3. Ифодалар рўйхати.	31
§ 7. Шарт	31
§ 8. SQL командалари.....	34
8.1. Берилганларнинг аниқлаш тили командалари:	34
8.2. Берилганларнинг манипуляция қилиш тили командалари:	51
8.3. Транзакцияларни бошқариш командалари:.....	58
8.4. Сеансни бошқариш командалари:	58
8.5. Системани бошқариш командалари.....	59
8.6. Ички қурилган SQL командалари.	59
PL/SQL ORACLE	59

1. PL/SQL АСОСЛАРИ.....	59
Белгилар тўплами.....	60
Лексемалар.....	60
Чеклагичлар.....	61
Идентификаторлар	62
Литераллар.....	64
Изоҳлар.....	64
2. Берилганлар турлари	65
Number турлари.....	66
LOB турлари	70
Фойдаланувчи томонидан аниқланган турлар.....	71
Берилганлар турларини ўзгартириш.....	73
Ўзгарувчи ва ўзгармасларни эълон қилиш	73
3. Номлаш қоидалари	76
Таъсир ва кўриниш соҳалари.....	79
Қиймат бериш оператори	81
Ифода ва солиштириш	82
Амалларни бажариш тартиби	82
4. Бул ифодалар	85
NULL қиймати.....	86
5. Функциялар	87
6. Назорат тузилмалари.....	89
Шартли назорат: IF жумлалари.....	90
Итератив назорат: LOOP ва EXIT жумлалари	93
Итерация схемалари	97
Динамик интервал	97
Таъсир қилиш соҳаси	98
Кетма-кет бошқарув: GOTO ва NULL жумлалари.....	100
7. Коллекциялар ва ёзувлар	105
Ичма-ич жойлашган жадваллар	106
Массивлар	107
Коллекцияларни аниқлаш ва эълон қилиш.....	108
Коллекцияларни инициализация қилиш ва унга ҳавола этиш.....	112
Коллекцияларга қиймат бериш ва солиштириш	117
Коллекцияларни бошқариш.....	120
8. PL/SQL орқали Oracle билан ўзаро мулоқот қилиш	128
SQL қўллаб қувватлаш.....	129
Берилганларни бошқариш.....	129
Транзакцияни бошқариш.....	129
SQL функциялари.....	129
SQL Pseudocolumns	130
SQL операторлари	131
Солиштириш операторлари.....	131
Тўплам операторлари	132
Қатор операторлари	132
9. PL/SQLда курсорлар билан ишлаш.....	132
Курсорларни бошқариш	132

Ошкор курсорлар.....	132
Ошкормас курсорлар	136
Курсорларни пакетларда бирлаштириш	136
FOR Loops курсоридан фойдаланиш	136
Курсор ўзгарувчиларидан фойдаланиш.....	138
Курсор ўзгарувчиларини бошқариш	140
Курсор атрибутларидан фойдаланиш.....	148
10. Хатоликларга ишлов бериш	152
Хатоликлар.....	153
Олдиндан аниқланган хатоликлар.....	155
Фойдаланувчи томонидан аниқланган мустаснолар	160
Мустасно қандай содир этилади	163
Мустаснолар қандай тарқалади.....	164
Мустаснони қайта содир этиш	167
Содир этилган мустасноларга ишлов бериш	167
11. PL/SQLда объект турлари ва улар билан ишлаш	174
Объект турлари	174
Атрибутлар	179
Методлар.....	179
Объект турларини аниқлаш	184
Объектларни эълон қилиш ва қиймат бериш.....	192
Атрибутларга мурожаат қилиш.....	194
Конструкторни чақириш.....	195
Методларни чақириш	195
Объектлардан умумий тарзда фойдаланиш	196
Объектларни бошқариш.....	199
1-§ Oracle сервер концепцияси	207
2-§ Берилганлар базасининг тузилмаси ва фазони бошқариш	211
3-§ Хотира ва жараёнлар тузилмаси	216
4-§ Берилганлар биргаликдалиги ва конкуренцияси	223
5-§ Таксимланган жараёнлар ва таксимланган берилганлар базалари	225
6-§ Юклатиш ва ёпиш амаллари	228
7-§ Берилганлар базаси махфийлиги	229
8-§ Берилганлар базасини тиклаш	236
9-§ Берилганлар базасини бошқариш учун объектли-реляцион модел	239
10-§ Берилганларга кириш	244
11-§ Oracle берилганлар базасининг администратори.....	250
12-§ Берилганлар базаси администраторининг имтиёзлари ва махфийлиги	253
13-§ Берилганлар базасининг администратори уланишини назорат қилиш методлари	254
14-§ Пароллар файлини бошқариш	257
15-§ Берилганлар базаси администратори учун мўлжалланган утилиталар.....	264
16-§ Берилганлар базаси администраторининг приоритетлари.....	264
17-§ Берилганлар базасини яратишдан олдин қилинадиган ишлар	268

18-§ Oracle берилганлар базасини яратиш	270
19-§ ББ юклатиш ва тўхтатиш	278
20-§ Oracle Developer.....	282
21-§ Form Builder.	283
22-§ Report Builder.....	289
23-§ Graphics Builder.....	295
Адабиётлар.....	341