

**ЎЗБЕКИСТОН РЕСПУБЛИКАСИ ОЛИЙ ВА ЎРТА МАХСУС ТАЪЛИМ
ВАЗИРЛИГИ
АБУ РАЙХОН БЕРУНИЙ НОМИДАГИ ТОШКЕНТ ДАВЛАТ ТЕХНИКА
УНИВЕРСИТЕТИ**

*Автоматика ва Бошқариш Факултет
Бошқаришда ахборот технологиялари кафедраси*

КОДИРОВ ЖАСУР

МАВЗУ «JAVA JDBC базаси модулини автоматлаш»

05.11.13 – «Ахборот буюмларни назорат қилиш асбоблари ва усуллари»

бакалаврият даражасини олиш учун тақдим этилган
диссертация

Илмий раҳбар: т.ф.д, профессор Соликов Я.Ю

Расмий оппонентлар: Зарипов О

ТОШКЕНТ – 2012

Мундарижа

Кириш

1. Тизмли таҳлил

- 1.1. JDBC JAVA базасини автоматлаш ҳақида.
- 1.2. Замонавий дастурий таъминот яратиш ҳақида.
- 1.3. Масалани қўйилиши.

2. Асосий қисм

- 2.1. Маълумотлар базаси ҳақида асосий тушинчалар.
- 2.2. Маълумотлар моделлари.
- 2.3. Маълумотлар базасини лойihalаштириш.
- 2.4. JAVA Визуал компоненталар библиотекази.
- 2.5. Фойдалануфчига йўриқнома.

3. Мехнат муҳофазаси ва техника хавфсизлиги

ЭХМ операторининг чарчасини олдини олиш.

Хулоса

Фойдаланилган адабиётлар

Маълумотлар базаси ҳақида асосий тушинчалар

Информацион технологияларнинг ривожланиши ва ахборот оқимларининг тобора ортиб бориши, маълумотларнинг тез ўзгариши каби ҳолатлар инсониятни бу маълумотларни ўз вақтида қайта ишлаш чораларининг янги усуллари кидириб топишга ундамоқда. Маълумотларни сақлаш, узатиш ва қайта ишлаш учун МБни яратиш, сўнгра ундан кенг фойдаланиш бугунги кунда долзарб бўлиб қолмоқда. Молия, ишлаб чиқариш, савдо-сотиқ ва бошқа корхоналар ишларини маълумотлар базасисиз тасаввур қилиб бўлмайди.

Маълумки, МБ тушунчаси фанга кириб келгунга қадар, маълумотлардан турли кўринишда фойдаланиш жуда қийин эди. Дастур тузувчилар маълумотларини шундай ташкил қилар эдиларки, у фақат қаралаётган масала учунгина ўринли бўларди. Ҳар бир янги масалани ҳал қилишда маълумотлар қайтадан ташкил қилинар ва бу ҳол яратилган дастурлардан фойдаланишни қийинлаштирар эди.

Ҳар қандай ахборот тизимининг мақсади реал муҳит объектлари ҳақидаги маълумотларга ишлов беришдан иборат. Кенг маънода маълумотлар базаси - бу қандайдир бир предмет соҳасидаги реал муҳитнинг аниқ объектлари ҳақидаги маълумотлар тўпламидир. Предмет соҳаси деганда автоматлаштирилган бошқаришни ташкил қилиш учун ўрганилаётган реал муҳитнинг маълум бир қисми тушинилади. Масалан, корхона, завод, илмий текшириш институти, олий ўқув юрти ва бошқалар.

Шуни қайд қилиш лозимки, **МБ**ни яратишда иккита муҳим шартни ҳисобга олмоқ зарур:

Биринчидан, маълумотлар тури, кўриниши, уларни қўллайдиган программаларга боғлиқ бўлмаслиги лозим, яъни **МБ**га янги маълумотларни киритганда ёки маълумотлар турини ўзгартирганда, программаларни ўзгартириш талаб этилмаслиги лозим.

Иккинчидан, МБдаги керакли маълумотни билиш ёки излаш учун бирор программа тузишга ҳожат қолмасин.

Шунинг учун ҳам **МБ**ни ташкил этишда маълум қонун ва қоидаларга амал қилиш лозим. Бундан буён **ахборот** сўзини **маълумот** сўзидан фарқлаймиз, яъни **ахборот** сўзини умумий тушунча сифатида қабул қилиб, **маълумот** деганда аниқ бир белгиланган нарса ёки ҳодиса сифатларини назарда тутамиз.

Маълумотлар базасини яратишда, фойдаланувчи ахборотларни турли белгилар бўйича тартиблашга ва ихтиёрий белгилар бирикмаси билан танлангани тез олишга интилади. Буни фақат маълумотлар тизимлаштирилган ҳолда бажариш мумкин.

Тизимлаштириш—бу маълумотларни тасвирлаш усуллари ҳақидаги қелишувни киритишдир. Агар маълумотларни тасвирлаш усули ҳақида қелишув бўлмаса, у ҳолда улар тизимлаштирилмаган дейилади. Тизимлаштирилмаган маълумотларга мисол сифатида матн файлига ёзилган маълумотларни кўрсатиш мумкин.

Маълумотлар базаси—бу маълум бир предмет соҳасига оид тизимлаштирилган(структуралаштирилган) маълумотларнинг номланган тўпламидир.

Маълумотлар базаси тушунчаси майдон, ёзув, файл (жадвал) каби элементлар билан ҳамбарчас боғлиқ.

Майдон—бу маълумотларни мантиқий ташкил этишни элементар бирлиги бўлиб, у ахборотни энг кичик ва бўлинмас бирлиги бўлган реквизитга мос келади. Майдонни тасвирлаш учун қуйидаги тавсифлардан фойдаланилади:

Майдон номи, масалан, фамилияси, исми, туғилган сана, лавозими, иш стажи, мутахассислиги.

Майдон тури, масалан, сон (числовой), символ (символьный), сана/вақт (дата/время), мантиқий (логический).

Майдон узунлиги (ўлчами), масалан, энг кўп символлар сифими;

Майдон аниқлиги, (сон типдаги маълумотлар учун) масалан, соннинг ўнлик қисмини акс эттириш учун ўнлик рақамдан тўртта.

Ёзув –бу мантиқий боғланган майдонлар тўплами. Ёзув тузилиши учун унинг таркибига кирувчи майдонлар таркиби ва жойлашиши кетма-кетлиги билан аниқланиб, уларни ҳар бири ичида элементар **ёзувларнинг нусхаси** деб аталади. Ёзув объектнинг бирор бир элементи ҳақида тўлиқ маълумотни ифодалайди.

Файл (жадвал) -бу бир хил тузилмага эга бўлган ёзувнинг нусхалар тўпламидир. У ўзича ҳар бир майдонда қийматга эга.

Ҳар бир МБ жадвали ўзининг бирламчи калитига эга бўлиши мумкин. **Бирламчи калит** деганда ёзувлар қайтарилмаслигини таъминловчи майдон (поля) ёки майдонлар гуруҳи тушинилади. Бирламчи калит сифатида ишлатиладиган майдон ёки майдонлар гуруҳи, бир хил ёзувга эга бўлмаслик шартини бажариши керак. Бирламчи калит қисқа ва сонли майдонлардан ташкил топиши мақсадга мувофиқдир. МБ жадвалига бирламчи калитни киритишдан мақсад, жадвалдаги маълумотларни излаш, тартиблаштириш ва танлаб олишда қулайликни беради. Бирламчи калит киритиш ёки киритмаслик фойдаланувчи тамонидан МБ жадвали структурасини ташкил қилишда аниқланади.

Бош жадвал ёрдамида қарам жадвалдаги мос маълумотларни чақиришни таъминлаш учун қарам жадвалда ташқи калит ташкил қилинади. “Битта-кўпга” боғланиш ҳолатида ташқи калит бош жадвалда ташкил қилинади. Биринчи ва иккинчи калитларни аниқлашда МББТ автоматик равишда жадвалда индексларни куради. Индекслар киритиш механизми МБ жадвалидаги маълумотларни тез топиш ва ажратиб олиш каби имкониятларни беради.

Бундан хулоса қилиб айтганда жадвалларда “индексация” киритиш жадвал маълумотларини излаш, бир нечасини танлаб олиш ва ажратиш каби ишларда қулайликни туғдиради. Бу эса ўз навбатида жадвал устида ҳар хил операциялар олиб боришни фойдаланувчига осонлаштиради. Ҳақиқатда

индекслашни ташкил этиш анча қийин бўлиб, у МБни лойихалаш жараёнида унинг структурасини ташкил этиш вақтида аниқланади.

МБдан фойдаланишда, яъни ундаги маълумотлар устида ҳар хил ишлар бажаришда бир неча усуллар мажуд. Бу усулларга база маълумотларидан фойдаланишга мурожаат (доступ) усуллари дейилади. Қуйидаги база маълумотлардан фойдаланиш усуллари кўриб чиқамиз.

JAVA ва Dbase МББСда индекслар бошқа файлда сақланади. Oracle, SyBase, InterBase ва SqrServer МББСда эса индекслар биргаликда база файлининг ўзида сақланади.

Маълумотларга ишлов бериш технологияси бўйича маълумотлар базаси марказлаштирилган ва тақсимланган базаларга бўлинади.

Марказлаштирилган маълумотлар базаси бир ҳисоблаш тизимининг хотирасида сақланади. Агар бу ҳисоблаш тизими компьютерлар тармоғининг компоненти бўлса, у ҳолда бундай базага тармоқ орқали кириш учун рухсат берилиши мумкин. Маълумотлар базасидан бундай фойдаланиш усули компьютерларнинг локал тармоқларида кўп учирайди.

Тақсимланган маълумотлар базаси ҳисоблаш тармоғининг турли компьютерларида сақланувчи бир-неча қисмлардан иборат бўлиб, улар кесишуви, хатто бир-бирини такрорлаши мумкин. Бундай база билан ишлаш маълумотларни тақсимланган базасини бошқариш тизимининг ёрдами билан амалга оширилади.

Маълумотлар базасидаги маълумотлардан фойдаланиш ҳуқуқи бўйича улар локал кириш ҳуқуқига эга бўлган маълумотлар базаси ва узоқ масофадан (тармоқдан) кириш ҳуқуқига эга бўлган маълумотлар базасига бўлинади.

Тармоқдан кириш ҳуқуқига эга бўлган маълумотлар базасини марказлаштирилган тизими бу каби тизимларнинг турли архитектурасини назарда тутаяди:

- Файл-сервер;

- Клиент-сервер.

Файл-сервер. Ушбу концепцияда тармоқдан кириш ҳуқуқига эга бўлган маълумотлар базаси тизимнинг архитектураси марказий ЭХМ (файллар сервери) сифатида тармоқ компьютерларидан бирини ажратиб кўрсатишини назарда тутди. Бундай компьютерда умумий фойдаланишга мўлжалланган марказий маълумотлар базаси сақланади. Тармоқдаги бошқа ҳамма компьютерлар ишчи станциялари функциясини бажаради. Уларнинг ёрдами билан фойдаланувчи тизимдан марказий маълумотлар базасига кириши таъминланади. Маълумотлар базаси файллари фойдаланувчи сўровларига мос равишда иш станцияларига юборилади. Маълумотларни қайта ишлаш асосан иш станцияларида амалга оширилади. Маълумотлар базасига кириш интенсивлиги катта бўлганда ахборот тизимнинг унумдорлиги пасаяди. Фойдаланувчилар иш станцияларида локал маълумотлар базаси яратишлари ва улардан яқка тартибда фойдаланишлари ҳам мумкин.

Клиент-сервер. Бу концепцияда, марказий маълумотлар базаси махсус компьютерда (Маълумотлар базаси серверида) сақланиши билан бирга, маълумотларни қайта ишлаш масалаларининг асосий қисми бажарилишини таъминлаши зарур.

Клиент (иш станцияси) томонидан маълумотларга берилган сўров сервердаги маълумотларни қидириш ва топишга олиб келади. Олинган маълумотлар (аммо файллар эмас) тармоқ бўйича сервердан клиентга узатилади. Клиент-сервер архитектурасининг ўзига хос хусусиятларидан бири SQL сўровлар тилидан фойдаланиш ҳисобланади.

Маълумотлар базаси - ахборот тизимларининг энг асосий таркибий қисми бўлиб ҳисобланади. Маълумотлар базасидан фойдаланиш учун фойдаланувчи ишини енгиллаштириш мақсадида маълумотлар базасини бошқариш тризимлари яратилган. Бу тизимлар маълумотлар базасини амалий дастурлардан ажратади.

Маълумотлар базасини бошқариш тизими (МББТ) -бу дастурий ва аппарат воситаларининг мураккаб мажмуаси бўлиб, улар ёрдамида фойдаланувчи маълумотлар базасини яратиш ва шу базадаги маълумотлар устида иш юритиши мумкин.

Жуда кўп турдаги МББТ мавжуд. Улар ўз махсус дастурлаш тилларига ҳам эга бўлиб, бу тилларга СУБД буйрукли дастурлаш тиллари дейилади. МББТга JAVA, Oracle, Clipper, Paradox, FoxPro, Access ва бошқаларни мисол келтириш мумкин.

Маълумотлар базасини бошқариш тизими—бу маълумотлар базасини яратиш, уларни долзарб ҳолатини таъминлаш ва ундаги зарур ахборотни топиш ишларини ташкил этиш учун мўлжалланган дастурлар мажмуи ва тил воситасидир.

Маълумотлар моделлари

Машина муҳитида маълумотларни ташкил этиш икки поғонадан иборат бўлиб, **мантикий ва физик** поғоналар билан характерланади. Маълумотларни бевосита физик ташкил этишда уларни машина "ташувчиси"да жойлаштириш усули аниқлаб олинади. Замонавий амалий дастурлар воситаларида маълумотларни ташкил этишнинг бу поғонаси автоматик равишда фойдаланувчи аралашувисиз таъминланади. Одатда, фойдаланувчи амалий дастурлар воситаларининг маълумотларни мантикий ташкил этиш хақидаги тушунчалар билан операциялар бажаради. Машина "ташувчиси"да маълумотларни мантикий ташкил этиш, фойдаланилаётган дастурий воситалардан ва машина муҳитида маълумотлар билан юритишдан боғлиқ. Маълумотларни ташкил этишнинг мантикий усули фойдаланилаётган маълумотлар тузилишининг тури ва дастурий воситалар орқали кулланиладиган *моделнинг шакли* аниқланади.

Маълумотларнинг модели - бу маълумотлар узаро боғланган тузилишлари ва улар устида бажариладиган операциялар тўпламидир. Моделнинг шакли ва унда фойдаланиладиган маълумотлар тузилишининг тури

дастурлаш тизими тилида фойдаланган маълумотларни ташкил этиш ва ишлов бериш концепциясини акс эттиради.

Маълумки, айнан бир ахборотни машина ички муҳитида жойлаштириш учун маълумотларнинг турли хил тузилишлари ва моделларидан фойдаланиш мумкин. Улардан қайси бирини танлаш ахборотлар базасини яратаётган фойдаланувчининг зиммасига юклатилган бўлиб, у қўплаб омиллардан боғлиқ. Бу омиллар қаторига мавжуд техник ва дастурий таъминотлар, ҳамда автоматлаштирилаётган масалаларнинг мураккаблиги ва ахборотнинг хажми кабилар киради.

Маълумотлар модели куйидаги таркибий қисмдан иборат:

1. Фойдаланувчининг маълумотлар базасига муносабатини намоиш этишга мулжалланган маълумотлар тузилмаси.

2. Маълумотлар тузилишида бажарилиш мумкин бўлган операциялар. Улар кўриб чиқиладиган маълумотлар модели учун маълумотлар тилининг асосини ташкил этади. Яхши маълумотлар тузилмасининг узигина етарли эмас. Маълумотларни аниқлаш тили (МАТ) ва маълумотлар билан амаллар бажариш тилининг (МАБТ) турли операциялари ёрдамида бу тузилма билан ишлаш имконига эга бўлиш зарур.

3. Яхлитликни назорат қилиш учун чеклашлар. Маълумотлар модели унинг яхлитлигини сақлаш ва ҳимоя қилишга имкон берувчи воситалар билан таъминланган бўлиши лозим.

Маълумотларнинг иерархик ва тармоқ моделлари. Машина муҳитидаги маълумотларнинг мураккаброқ моделлари - тармоқли ва иерархик моделлар бўлиб ҳисобланади. Бу моделлар уларнинг ўзларига хос турдаги маълумотлар базасини бошқариш тизимида ишлатилади. МББТда маълумотларни мантиқий ташкил этиш усули маълумотларнинг тармоқли ёки иерархик моделига мос ҳолда кўрсатилади. Бундай модел ўзаро боғлиқ объектларнинг мажмуидир. Икки объектнинг алоқаси уларнинг бир-бирига тобелигини акс эттиради. Тармоқли ёки иерархик моделида объект бўлиб, МББТ киритилган маълумотлар тузилмасининг асосий турлари ҳисобланади. Турли МББТларда

бу турдаги маълумотларнинг тузилмаси турлича аниқланиши ва номланиши мумкин.

Моделларда маълумотларнинг тузилмалари. маълумотларнинг намунавий тузилмаларига қуйидагилар киради: маълумотларнинг элементи, маълумотларнинг агрегати, ёзув, маълумотлар базаси ва ҳ.к. бу элементлар ва агрегатлаи ўзаро алоқада бўлган тузилма билан тавсифланади. шунинг учун ёзувнинг тузилмаси иерархик характерга эга бўлиши мумкин. бир хил тузилмага эга бўлган ёзув нусхалари тўпламининг ҳаммаси ёзув турини ташкил этади.

Маълумотларнинг элементи - бу маълумотлар тузилмасининг номланган минимал бирлиги (файлли тизимлардаги майдоннинг ўхшаши).

Маълумотлар агрегати - бу маълумотлар элементларнинг қуйи тўплами ёки ёзувлар ичидаги бошқа агрегатларнинг номланган қуйи тўплами.

Ёзув - умумий ҳолда агрегат бўлиб, у бошқа агрегатларнинг таркибига кирмайдиган таркибли агрегатдан иборат.

Объектларнинг моделлардаги алоқаси. Маълумотлар модели бир неча туридаги ёзувларни (объектларни) ўз ичига олиши мумкин. Маълумот модели объектлар ўртасида алоқалар ўрнатади. Қандайдир бир предмет соҳаси учун моделнинг ўзаро боғланган муайян объектлар тўплами маълумотлар базасини ташкил қилади.

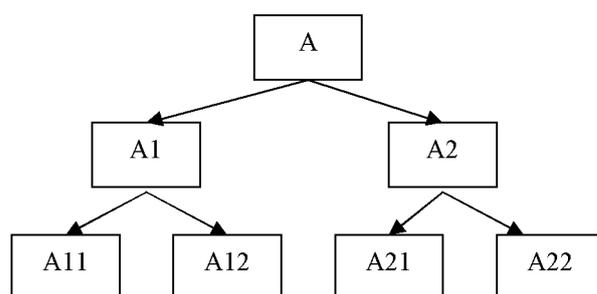
Икки турдаги ёзувларнинг (модел объектлари) ўртасидаги алоқалар, уларнинг нусхалари ўртасидаги гуруҳ муносабатлари билан аниқланади. Гуруҳ муносабати - бу икки турдаги ёзувлар ўртасидаги катъий иерархик муносабат бўлиб, улар асосий ёзувлар тўплами ва тобе ёзувлар тўпамидан иборатдир.

Иерархик моделларда калит бўйича бевосита кириш одатда, фақат бошқа объектларга тобе бўлмаган энг юқори поғонадаги объектгагина мумкин. Бошқа объектларга кириш моделнинг энг юқори поғонасидаги объектдан алоқалар бўйича амалга оширилади. Тармоқли моделларда эса калит бўйича бевосита ихтиёрий объектга кириш (унинг моделда жойлашган поғонасидан қатъий

назар) таъминланиши мумкин.. Шунингдек, алоқалар бўйича ҳар қандай нуқтадан кириш ҳам мумкин. Тармоқли моделларда объект (ёзув, файл)нинг тузилмаси кўпинча чизикли ва камроқ ҳолларда эса иерархик бўлади. Қуйи поғонадаги маълумотларнинг тузилмаси ҳам ўз хусусиятга ва номига эга бўлиши мумкин. Масалан, атрибут бу маълумотлар элементининг аналогидир. Чизикли тузилмага эга бўлган объект фақат оддий ва калит атрибутлардан иборат. Иерархик моделлардаги объект (ёзув, сегмент) тузилмаси иерархик ёки чизикли бўлиши мумкин.

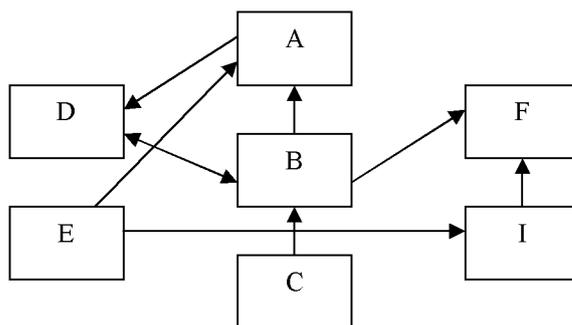
Турли предмет соҳалари учун маълумотларнинг тармоқли модели иерархик моделига нисбатан машинанинг иш муҳитида ахборот тузилмаларини акс эттирувчи умумий восита ҳисобланади. Кўплаб предмет соҳаларининг маълумотлари ўртасидаги алоқалар тармоқли кўринишга эга. Бу эса маълумотларнинг иерархик моделига эга бўлган МББТдан фойдаланишни чеклаб қўяди. Тармоқли моделлар, маълумотларнинг иерархик алоқасини ҳам акс эттиришга имкон беради. Бундан ташқари, тармоқли моделлар билан ишлаш технологияси фойдаланувчи учун қулайдир, чунки маълумотларга киришни амалга оширишда ҳеч қандай чеклашлар йўқ ва бевосита ихтиёрий поғонадаги объектларга кириш имкони мавжуд.

Иерархик маълумотлар базасида - маълумотлар иерархия (дарахт) кўрнишида сақланади. Унинг кўринишини қуйидагича тасвирлаш мумкин.



Масалан, бу ерда A12 тугунидаги маълумотни олиш учун, олдин МБдан А тугун, кейин А1 тугун ва ундан кейин А12 топилади.

Тармоқ маълумотлар базаси - ички маълумотлар структураси, бири-иккинчисига боқлиқ равишда бўлади. Унинг кўринишини қуйидагича тасаввур қилиш мумкин.



Маълумотларнинг тармоқ модели

Иерархик ва тармоқ моделларида маълумотлар тасвирининг мураккаблиги ва бу маълумотлар орасидаги алоқани МБни лойиҳалашда аниқлаш керак бўлиб, бу эса МБга сўров берилганда реляцион МБ жадваллари орасида алоқа ўрнатишни тامينлаб беради.

Маълумотларнинг реляцион модели. Реляцион МБ кучли назарий фундаментга эга бўлиб, у математик муносабатлар (отношения) назариясига асосланган. Маълумотларнинг реляцион модели концепцияси 1970 йилда Е.Ф.Кодд томонидан таклиф қилинган бўлиб, у маълумотларни тавсифлаш ва тасвирлашнинг амалий дастурларидан боғлиқ бўлмаслигини таъминлаш масаласини ҳал қилиш учун хизмат қилади.

Маълумотларнинг реляцион модели асосида «муносабат» тушунчаси ётиб, у инглизча relation сўзидан олинган. Баъзи бир қоидаларга амал қилган ҳолда муносабатларни икки ўлчовли жадвал кўринишда тасвирлаш мумкин. Жадвал ҳар қандай одамга тушунарли ва қулайдир.

Реал дунё объектлари хақидаги маълумотларини ЭХМ хотирасида сақлаш ва улар орасидаги муносабатларни моделлаштириш учун муносабатлар (жадвал) тўпламидан фойдаланиш мумкинлигини Е.Ф.Кодд исботлаб берди.

Маълумотларнинг реляцион базаси - бу ўзаро боғланган муносабатлар, яъни жадваллар тўпламидир. Ҳар қандай муносабат (жадвал) компьютерларнинг хотирасида файл кўринишда жойлаштирилади. Уларнинг орасида қуйидаги мослик мавжуд:

| Файл | Жадвал | Муносабат | Мазмуни |
|--------|--------|-----------|--------------------|
| Ёзув | Сатр | кортеж | мазмуннинг нуسخаси |
| Майдон | Устун | атрибут | атрибут |

Жадвал ҳамма учун жуда қулай бўлиши билан бир қаторда маълумотларни манипуляция қилишнинг асосий уч операциясини бажариш учун ноқулайдир:, яъни тартиблаш, индексларнинг қийматлари бўйича гурухлаш ва дарахт кўринишидаги параметрлар билан ишлаш.

Жадвалда ушбу уч операция бир-бири билан чамбарчас боғланган. Бу эса баъзи бир операцияларни бажаришда маълум бир қийинчиликларга олиб келади. Масалан, маълумотларни бир параметр асосида тартиблаш иккинчи бир параметр бўйича тартиблашни бузиб юбориши туфайли зарур маълумотларни излаб топиш операцияси бир параметр бўйича осонлашса, бошқалари бўйича қийинлашади.

Кодд таклиф қилган усулининг оригиналиги шундан иборатки, у муносабатларга (жадвалларга) тадбиқ қилиш учун жуда чиройли қурилган операциялар тизимини ишлаб чиқди. Уларни амалга ошириш натижасида бир муносабатни бошқа муносабат орқали ҳисоблаб чиқиш имконияти пайдо бўлди. Бу ахборотларни сақланадиган ва сақланмайдиган (ҳисобланадиган) қисмларга ажратиш, ҳамда компьютер хотирасини тежаш зарур бўлган пайтда ахборотларнинг сақланмайдиган қисмини сақланадиганлар асосида ҳисоблаб чиқиш имкониятини беради.

Маълумотларнинг реляцион базасидаги муносабатлар устида бажариладиган асосий операциялар саккизта бўлиб, улар куйидагилардан иборат:

-тўпламлар устидаги ананавий (традицион) операциялар, яъни тўпламларнинг бирлашмаси (йиғиндиси), кесишмаси (кўпайтмаси), тўлдирувчиси (айирмаси), декарт кўпайтмаси, бўлишмаси;

-махсус реляцион операциялар, яъни проекциялаш, боғланиш (кўшилиш), бирлаштириш (улаб кўйиш) ва танлаш.

Ҳар бир маълумотлар базасини бошқариш тизимининг самарадорлиги ушбу операцияларнинг борлиги ва уларни бажариш воситаларининг қанчалик қулайлиги билан аниқланади. Реляцион МББТда муносабатлар устида операциялар бажариш учун мўлжалланган тилларини икки синфга ажратиш мумкин: реляцион алгебра тили (РАТ) ва реляцион ҳисоб тили (РХТ).

РАТ реляцион алгебрага (Кодд алгебрасига, α -алгебрага) асосланган. Маълум тартиб муносабатлар устида операцияларни кетма-кет ёзиш асосида хоҳлаган натижага эришиш мумкин. Шунинг учун РАТни процедурали тил дейишади.

РХТ предикатларни ҳисоблаб чиқишнинг классик усулига асосланган. Улар фойдаланувчиларга сўровларни ёзиш учун маълум қоидалар тўпламини беради. Бундай сўровларда фақат хоҳлаган натижа ҳақидаги ахборотлар бўлади холос. Ушбу сўров асосида МББТ янги муносабатлар ҳосил қилиш йўли билан автоматик тарзда зарур натижани беради. Шунинг учун РХТни процедуралимас тил дейишади.

Маълумотлар базасини лойиҳалаштиришда реляцион модел билан ишлаш анча ноқулайликларга олиб келади. Шу сабаб маълумотлар базасини лойиҳалашда ҳар хил семантик моделлар ҳам ишлатилади. Улардан энг кўп тарқалганларидан бирига - **ER** модели дейилади. Бу модел инглизча “**Entity-relation**” дейилиб, маъноси “**Моҳият-боғланиш**” демакдир.

Бу модел 1976 йил Питер Чен тамонидан киритилган бўлиб у ўзига бир катор график диаграммаларини олувчи бир неча ҳар хил турдаги

компоненталарни бирлаштирган. Питер Чен моҳиятлар тўплами ва улар орасида боғланиш сифатида реляцион маълумотлар структурасини интерпритация қилишни таклиф қилди.

ER -моделининг асосий компоненталари моҳият, боғланиш ва атрибут (хосса) бўлиб ҳисобланади.

Моҳият -бу маълумотлари маълумотлар базасида сақланиши керак бўлган бирор реал ёки тасаввур қилинган объектдир. ER модели диаграммасида моҳият одатда тўртбурчак шаклида тасвирланиб, унинг ичига моҳият номи қўйилади.

Моҳият аниқ маънога эга бўлган номга эга бўлиб, у ягона бўлиши керак. Моҳият турини унинг нусхаси билан фарқ қилиш керак. Моҳият номи унинг нусхасига эмас, турига берилади. Моҳият нусхаси -бу аниқ бир хил турдаги нарсалар, ходисалар ва бошқалардир.

Масалан, юқоридаги “Ўқувчи” моҳиятида “Ўқувчи” моҳият турининг номи, моҳият нусхаси эса аниқ бир ўқувчидир. Масалан, Ахмедов, Тошматов ва бошқа.

Боғланиш -бу икки ёки бир неча моҳиятлар бирикмасидир. Боғланиш фақат иккита ҳар хил моҳиятлар орасида мавжуд бўлади. Охириги боғланишга рекурсив дейилади.

“Моҳият-боғланиш” диаграммаларини ишлаб чиқувчи ҳар хил турдаги стандарт метаологиялар мавжуд. Масалан, IDEFIX, IE, DM. Бу усуллар ҳар қайсисининг моҳият-боғланишни тасвирлаш учун ўз белгилари бор.

Атрибут (хосса) -моҳиятни характерловчи номлардир. У ўзида ягона муракаб бўлмаган структурани тасвирлаб, моҳият ҳолатини характерлайди. Масалан, “Ўқувчи” моҳияти атрибути -код, фамилия, исм, манзил, ёш ва бошқалардир.

Моҳият атрибутлар тўплами чексиздир. У ахборот тизимлари билан ишлайдиган фойдаланувчи талабига ва ечиладиган масалага боғлиқдир.

Маълумотлар базаси жадваллари орасидаги реляцион боғланиш.

МБинг икки ва ундан ортиқ жадваллари орасида бири иккинчисига боғлиқ бўлиши мумкин. Агар иккинчи жадвал биринчи жадвалга қарам бўлса биринчи жадвалга бош жадвал, иккинчи жадвалга эса қарам жадвал дейилади. Бош жадвалдаги битта ёзувга қарам жадвалда унга мос бир неча ёзув мавжуд бўлиши мумкин.

МБ жадваллари орасида учта ҳар хил алоқа бўлиши мумкин: “битта-кўпга”; “битта-биттага”; “кўп-кўпга”.

“Битта-кўпга” боғланиш. “Битта-кўпга” боғланиш бўлади, қачонки бош жадвалдаги битта ёзув қарам жадвалдаги бир неча ёзувга алоқаси бўлса.

Реляцион МБ учун “битта-кўпга” боғланиш ҳолати энг кўп ишлатилади.

“Битта-биттага” боғланиш. “Битта-биттага” боғланиш бўлади, қачонки бош жадвалдаги битта ёзув қарам жадвалдаги фақат битта ёзувга алоқаси бўлса. “Бирга-бир” боғланиш қаттиқ ёки юмшоқ бўлиши мумкин. Агар бош жадвалдаги битта ёзувга қарам жадвалдан ҳамма вақт фақат битта ёзув тўғри келса қаттиқ боғланиш бўлади. Агар бош жадвалдаги битта ёзувга қарам жадвалда битта ёзув бўлиш ёки бўлмаслик шарти бўлса у ҳолда боғланиш юмшоқ бўлади.

“Кўп-кўпга” боғланиш. “Кўп-кўпга” боғланиш қуйидаги ҳолларда бўлиши мумкин.

а) бош жадвалдаги ёзувга қарам жадвалда биттадан ортиқ ёзув тўғри келса.

б) қарам жадвалдаги ёзувга бош жадвалда биттадан ортиқ ёзув тўғри келса.

Битта жадвалдаги ёзувлар ҳам бир-бири билан алоқада бўлиши мумкин. куйидаги мисолни караймиз. Реляцион МБ куйидаги дарахт кўринишдаги структурага эга бўлсин.

Маълумотлар базасини лойиҳалаштириш

Маълумотлар базасини (МБ) ишлаб чиқиш (лойиҳалаш)нинг асосий мақсади унинг мантикий тузилишини аниқлашдан иборатдир. МБни ишлаб чиқиш предмет соҳасини тавсифлаш асосида амалга оширилади. Бу тавсифлаш МБга кирувчи ҳамма маълумотларни ўз ичига оловчи хужжатлар мажмуини ва предмет соҳасини ифодаловчи объект ва жараёнлар ҳақидаги бошқа маълумотларни ўз ичига олади.

МБни яратишни уни лойиҳалашдан бошлаш лозим. Лойиҳалаш натижасида реляцион базанинг тузилиши, яъни реляцион жадваллар таркиби, уларнинг тузилиши ва мантикий алоқадорлиги аниқланади. Реляцион жадвалнинг тузилиши эса унинг устунлари таркиби, уларнинг кетма-кетлиги, устун маълумотларининг тури ва ўлчами, шунингдек жадвал калити билан аниқланади.

Мантикий лойиҳалаш.

МБ яратишнинг энг зарурий ва масъулятли босқичларидан бири - бу мантикий лойиҳалаштиришдир. Унинг асосий масаласи танланган МББТ учун мўлжалланган ҳолда МБ мантикий схемасини ишлаб чиқишдан иборат. Мантикий лойиҳалаштириш босқичи концептуал лойиҳалаштиришдан фарқли равишда, у компьютернинг дастурий воситасини тўлиқ ҳисобга олган ҳолда амалга оширилади. Иш мазмуни бўйича мантикий лойиҳалаштириш ахборот тизимини ва уни ташкил этувчи қисмларни реал МББТига мос шаклда моделлаштиришдан иборат.

Мантикий лойиҳалаштириш жараёни куйидаги босқичлардан иборат:

1. Аниқ бир МББТни танлаш
2. Концептуал схемани мантикий схемага ўтказиш
3. Зарур калитларни танлаш

4.Сўров тилини тавсифлаш

Аниқ бир МББТни танлаш процедурасини батафсил қараймиз. МББТнинг амалга ошириш учун МББТни танлаш жуда катта масъулият талаб қилади. Бу бир томондан МББТларнинг жуда кўплиги бўлса, иккинчи томондан кўп сонли характеристикалар бўйича МББТни баҳолаш ва уларнинг орасидан айнан шундай тизимни танлаш керакки, у фойдаланиувчи ва ишлаб чиқарувчилар талабларини тўлиқ қаноатлантириши мумкин бўлсин. Чунки МББТда ахборотдан фойдаланиш ва ишлов бериши самарадорлиги МББТнинг қанчалик тўғри танлашдан боғлиқ бўлади.

МББТни танлашнинг асосий меъёрларидан бири-бу маълумотларни ишлатадиган ички моделининг концептуал схемасини тавсифлаш учун қанчалик самарадор эканлигини баҳолашдан иборат. Шахсий компьютерлар учун мўлжалланган МББТларнинг кўпчилиги, одатда маълумотларнинг реляцион ёки тармоқ моделига таянган ҳолда ишлайди. Замонавий МББТларнинг жуда катта қисми реляцион модел асосида яратилган. Агар реляцион тизим танланган бўлса, у ҳолда МББТнинг концептуал схемасини реляционга акслантириш (ўтказиш) олдинда турибди.

Ишнинг мазмуни бўйича маълумотнинг танланган модели (реляцион, тармоқ ва иерархик) маълумотлар тизилмасини тавсифлаш учун восита беради. Процедуралар МББТ ядросига қирадиган маълумотларни тавсифлаш тилида бажарилади.

МББТнинг иккинчи таркибий қисми маълумотларни манипуляция қилиш тилдан иборат. Ундан МББТ турли тадбиқлар учун ишлатишда фойдаланилади. Кўп ҳолларда маълумотларни манипуляция қилиш тили (ММТ) дастурлаштириш тилга ўрнатилган (киритилган) бўлади. ММТ турли имкониятларга эга бўлиши мумкин: қуйи поғонадаги тил ва юқори поғонадаги тил. Одатда қуйи поғонадаги тил процедурали, юқори поғонадагиси эса декларатив тил бўлади. Процедурали тиллардан фойдаланиш маълум тайёргарликни талаб қилади, декларатив тил бўлса кўпроқ профессионал бўлмаган фойдаланувчилар учун яроқли. Шунинг учун маълум ММТга эга

МББТни танлаш махсус тайёргарлиги бўлмаган фойдаланувчи учун жуда муҳимдир. Бундан ташқари МББТга сервис дастурлар ва амалий масалаларни ечиш учун воситалар киради.

MainMenu - дастур бош менюси. Компонента мураккаб иерархик структурали меню яратиш учун хизмат қилади.

JDBCMenu - ёрдамчи ёки локал менюси. Бу меню ойнада сичқонча ўнг тугмасини босиш билан чиқади.

Label - метка (белги). Бу компонента форма ойнасига унча узун бўлмаган бир қаторли ёзувни чиқаришда ишлатилади ва унинг пиктограммаси панелда “А” кўринишда берилган.

Edit - киритиш қатори. Форма ойнасида матнли қатор киритиш ва тахрирлашда ишлатилади.

Memo - кўпқаторли матн мухаррири. Кўпқаторли матнларни киритиш ёки чиқаришда ишлатилади.

Button - буйруқ тугмаси (Обработчик события OnClick). Бу компонента дастурчи тамонидан берилган бир неча буйруқларни бажаришда ишлатилади.

CheckBox - боғлиқ бўлмаган танлаш тугмаси (переключатель). Дастурда бу компонента асосий мантикий хоссаси (**Checked**) ўзгартирилади.

RadioButton - боғлиқ бўлган танлаш тугмаси (переключатель). Янги тутанлаш тугмаси босилганда, олдин танланган тугма автоматик равишда озод этади.

ListBox - рўйхатдан танлаш. Рўйхат вариантларини тақдим этади ва танлаш имконини яратади.

ComboBox – киритиш қаторига эга (комбинированный) рўйхатдан танлаш. Рўйхатдан комбинация қилиб танлаш

ScrollBar - йўлчали бошқариш. Windows ойнаси четларида горизонтал ёки вертикал йўлча ташкил этади.

GroupBox - элементлар гуруҳи. Маъно бўйича бир неча боғлиқ компоненталарни груҳлашда ишлатилади.

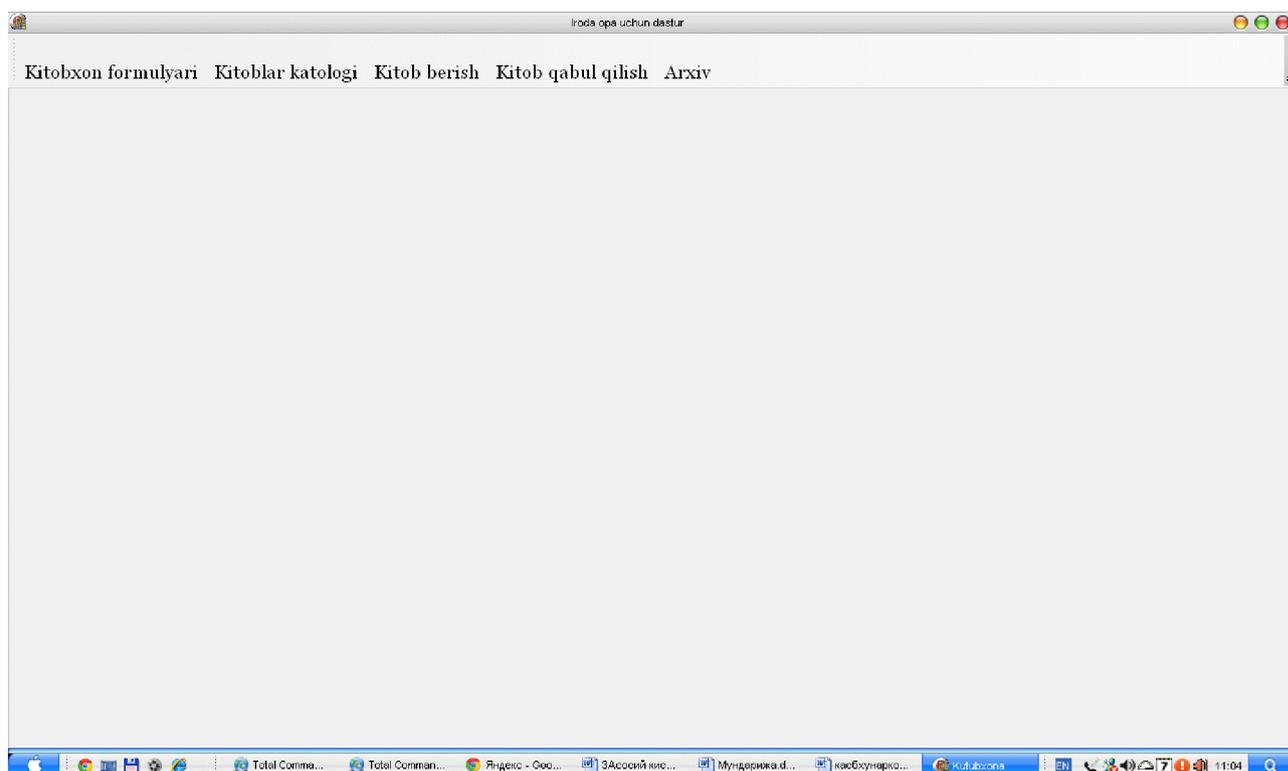
RadioGroup - боғлиқ гуруҳланган танлаш тугмалари (ўчириб ёқувчи тугмалар). Бир неча боғлиқ танлаш тугмалари хоссаларини сақлайди.

Panel - панел. Бу компонента, худди GroupBoxга ўхшаб бир неча компоненталарни бирлаштириш учун хизмат қилади.

Actionlist - таъсир қилиш рўйхатлари. Фойдаланувчи дастурга марказлашган ҳолда таъсир қилиши учун ишлатилади.

Фойдаланувчига йўриқнома

Дастурни таҳлил қилувчи дастурий таъминот JAVA дастурлаш тилида ишлаб чиқилди. Дастурий таъминотда талабаларнинг ҳар бир дарсдаги давомати киритилади. Дастур киритилган маълумот асосида талабаларга хайфсан, қаттиқ хайфсан бериш ёки талабалар сафидан чиқариш тўғрисида маълумот беради. Дастурнинг умумий кўриниши қуйидагича:



Дастурга кириш хавфсизлигини таъминловчи кириш қисми, махфий код киритилиб дастурга кирилади ва қуйидагидек мулоқот ойнаси пайдо бўлади.

Хабар дарчаси дастур дастлаб ишга тушган вақтида чиқади ва хайфсан, қаттиқ хайфсан олган талабалар ҳамда талабалар сафидад чиқарилиши керак бўлган талабалар ҳақида ахборот беради.

```
package JDBC.component.permission;
```

```
import java.util.Date;
```

```
import java.util.HashSet;
```

```
import java.util.Iterator;
```

```
import java.util.Properties;
```

```
import java.util.Set;
import java.util.logging.Level;

import net.mysrc.util.ObjectPool;

public class PermissionComponent extends Component {

    private DbsComponent mDbs;

    private boolean doOutput = false;

    private ObjectPool cachedRights;
    private int cacheSize = 0;

    public PermissionComponent(Properties settings) throws CfgException {
        super(settings);
        cacheSize = Integer.parseInt(settings.getProperty("cacheSize"));
        cachedRights = new ObjectPool(cacheSize);
    }

    public void init() throws CfgException {
        try {
            mDbs = (DbsComponent) getBroker().getComponent("Dbs");
            mDbs.registerFactory(new Right.Factory());
        } catch (DbsException exc) {
            getLogger().log(
                Level.WARNING,
                "RightFactory could not be registered.",
                exc);
            throw new CfgException(
                this,
                "RightFactory could not be registered.\n" + exc);
        }
    }

    public void removeUser(AbstractUser toRemove) {
        removePermission(toRemove.getInitialId());
    }

    public void addUser(AbstractUser toAdd) {
```

```

    }

    public void stop() throws CfgException {
    }

    public void start() throws CfgException {
    }

    protected boolean isAllowed(long objectId, long userId, String right) {

        long startTime = (new Date()).getTime();
        int usedFromCache = 0;
        boolean returnValue = false;
        boolean initialValue = true;
        Storable[] rights = null;
        Set idsToSearch = new HashSet();

        UserComponent userComp =
            (UserComponent) CfgComponent.getBroker().getComponent("User");
        try {
            // get the user or group
            AbstractUser currentUser = userComp.getAbstractUserById(userId);
            // get the memberships
            idsToSearch = currentUser.getMemberships(false);
        } catch (Exception e) {
            getLogger().log(
                Level.WARNING,
                "Couldn't get user with id " + userId + " or memberships.",
                e);
        }

        idsToSearch.add(new Long(userId));

        Iterator ids = idsToSearch.iterator();
        DbConnection dbsCon = null;
        Properties searchPattern = new Properties();
        // set object id to search for
        searchPattern.setProperty("objectId", "" + objectId);
        // set right to search for

```

```

searchPattern.setProperty("right", right);
// go through all ids (user or group and memberships)
while (ids.hasNext()) {
    long currentId = ((Long) ids.next()).longValue();
    // set current id to search for
    searchPattern.setProperty("userId", "" + currentId);
    long hc = currentId ^ objectId;
    Integer key =
        new Integer(((int) (hc ^ (hc >>> 32))) ^ right.hashCode());
    Right currentRight = null;
    if (!cachedRights.contains(key)) {
        // search db if not in cache
        try {
            dbsCon =
                ((DbComponent) CfgComponent
                    .getBroker()
                    .getComponent("Dbs"))
                    .connect();
            rights = dbsCon.get(Right.class, searchPattern);
            if (rights != null && rights.length > 0) {
                currentRight = (Right) rights[0];
                addToCache(currentRight);
            } else {
                cachedRights.put(key, null);
            }
        } catch (Exception e) {
            getLogger().log(
                Level.WARNING,
                "Error in method isAllowed().",
                e);
        } finally {
            try {
                dbsCon = dbsCon.release();
            } catch (Exception e) {
                // catch the DbException and the possible NPE
            }
        }
    } else {
        // get from cache
        currentRight = getFromCache(key);
        usedFromCache++;
    }
}

```

```

        if (currentRight != null) {
            if (initValue) {
                returnValue = currentRight.getValue();
                initValue = false;
            } else {
                returnValue = returnValue && currentRight.getValue();
            }
            if (returnValue == false) {
                // if returnValue is false no other ids
                // need to be searched
                break;
            }
        }
    }
    if (doOutput)
        System.out.println(
            "*** PermissionComponent.isAllowed() took "
                + ((new Date()).getTime() - startTime)
                + "("
                + usedFromCache
                + ")");
    return returnValue;
}

protected void setAllowed(
    long objectId,
    long userId,
    String right,
    boolean value) {
    DbConnection dbCon = null;
    try {
        dbCon =
            ((DbComponent) CfgComponent.getBroker().getComponent("Db"))
                .connect();
        Properties searchPattern = new Properties();
        searchPattern.put("objectId", "" + objectId);
        searchPattern.put("right", right);
        searchPattern.put("userId", "" + userId);
        Storable[] rights = dbCon.get(Right.class, searchPattern);
        // if there is already that right for this object and user
        // update it!
    }
}

```

```

        if (rights != null && rights.length > 0) {
            Right currentRight = (Right) rights[0];
            long actualId = CfgComponent.getBroker().generateId(this);
            currentRight.setValue(value);
            currentRight.setParentId(currentRight.getActualId());
            currentRight.setActualId(actualId);
            dbsCon.update(currentRight);
            addToCache(currentRight);
        }
        // else make a new one and add it!
    } else {
        Right currentRight =
            new Right(CfgComponent.getBroker().generateId(this));
        currentRight.setObjectId(objectId);
        currentRight.setUserId(userId);
        currentRight.setRight(right);
        currentRight.setValue(value);
        dbsCon.add(currentRight);
        addToCache(currentRight);
    }
} catch (Exception e) {
    getLogger().log(Level.WARNING, "Error in method setAllowed().", e);
} finally {
    try {
        dbsCon = dbsCon.release();
    } catch (Exception e) {
        // catch the DbsException and the possible NPE
    }
}
}

protected void removePermission(long id) {
    DbsConnection dbsCon = null;
    try {
        UserComponent userComp =
            (UserComponent) CfgComponent.getBroker().getComponent("User");
        dbsCon =
            ((DbsComponent) CfgComponent.getBroker().getComponent("Dbs"))
                .connect();
        Storable[] rightsToDelete;
        Properties searchPattern = new Properties();
        if (userComp.isUser(id) || userComp.isUserGroup(id)) {

```

```

        searchPattern.put("userId", "" + id);
    } else {
        searchPattern.put("objectId", "" + id);
    }
    Storable[] rights = dbsCon.get(Right.class, searchPattern);
    if (rights != null && rights.length > 0) {
        for (int i = 0; i < rights.length; i++) {
            dbsCon.remove(rights[i]);
            removeFromCache((Right) rights[i]);
        }
    }
} catch (Exception e) {
    getLogger().log(
        Level.WARNING,
        "Error in method removePermission(id).",
        e);
} finally {
    try {
        dbsCon = dbsCon.release();
    } catch (Exception e) {
        // catch the DbsException and the possible NPE
    }
}
}

```

```

protected void removePermission(long objectId, long userId) {
    DbsConnection dbsCon = null;
    try {
        dbsCon =
            ((DbsComponent) CfgComponent.getBroker().getComponent("Dbs"))
                .connect();
        Storable[] rightsToDelete;
        Properties searchPattern = new Properties();
        searchPattern.put("userId", "" + userId);
        searchPattern.put("objectId", "" + objectId);
        Storable[] rights = dbsCon.get(Right.class, searchPattern);
        if (rights != null && rights.length > 0) {
            for (int i = 0; i < rights.length; i++) {
                dbsCon.remove(rights[i]);
                removeFromCache((Right) rights[i]);
            }
        }
    }
}

```

```

        }
    } catch (Exception e) {
        getLogger().log(
            Level.WARNING,
            "Error in method removePermission(object,user).",
            e);
    } finally {
        try {
            dbsCon = dbsCon.release();
        } catch (Exception e) {
            // catch the DbsException and the possible NPE
        }
    }
}
}

```

```

protected void removePermission(long objectId, String right) {
    DbsConnection dbsCon = null;
    try {
        dbsCon =
            ((DbsComponent) CfgComponent.getBroker().getComponent("Dbs"))
                .connect();
        Storable[] rightsToDelete;
        Properties searchPattern = new Properties();
        searchPattern.put("right", right);
        searchPattern.put("objectId", "" + objectId);
        Storable[] rights = dbsCon.get(Right.class, searchPattern);
        if (rights != null && rights.length > 0) {
            for (int i = 0; i < rights.length; i++) {
                dbsCon.remove(rights[i]);
                removeFromCache((Right) rights[i]);
            }
        }
    } catch (Exception e) {
        getLogger().log(
            Level.WARNING,
            "Error in method removePermission(object,right).",
            e);
    } finally {
        try {
            dbsCon = dbsCon.release();
        } catch (Exception e) {

```

```

        // catch the DbsException and the possible NPE
    }
}

protected void removePermission(long objectId, long userId, String right) {
    DbsConnection dbsCon = null;
    try {
        dbsCon =
            ((DbsComponent) CfgComponent.getBrokerer().getComponent("Dbs"))
                .connect();
        Storable[] rightsToDelete;
        Properties searchPattern = new Properties();
        searchPattern.put("userId", "" + userId);
        searchPattern.put("objectId", "" + objectId);
        searchPattern.put("right", right);
        Storable[] rights = dbsCon.get(Right.class, searchPattern);
        if (rights != null && rights.length > 0) {
            for (int i = 0; i < rights.length; i++) {
                dbsCon.remove(rights[i]);
                removeFromCache((Right) rights[i]);
            }
        }
    } catch (Exception e) {
        getLogger().log(
            Level.WARNING,
            "Error in method removePermission(object,user,right).",
            e);
    } finally {
        try {
            dbsCon = dbsCon.release();
        } catch (Exception e) {
            // catch the DbsException and the possible NPE
        }
    }
}

// internal cache methods

private void removeFromCache(Right toRemove) {
    long hc = toRemove.getUserId() ^ toRemove.getObjectId();

```

```

        Integer key =
            new Integer(
                ((int) (hc ^ (hc >>> 32))) ^ toRemove.getRight().hashCode());
        if (cachedRights.contains(key))
            cachedRights.remove(key);
    }

    private Right getFromCache(Integer key) {
        if (cachedRights.contains(key)) {
            if (doOutput)
                System.out.println(
                    "*** PermissionComponent.getFromCache found key");
            Object toReturn = cachedRights.get(key);
            if (toReturn != null) {
                return (Right) toReturn;
            } else {
                return null;
            }
        } else {
            return null;
        }
    }

    private void addToCache(Right toAdd) {
        if (doOutput)
            System.out.println(
                "*** PermissionComponent.addToCache actual size");
        long hc = toAdd.getUserId() ^ toAdd.getObjectId();
        Integer key =
            new Integer(
                ((int) (hc ^ (hc >>> 32))) ^ toAdd.getRight().hashCode());
        cachedRights.put(key, toAdd);
    }
}

```

ЭҶМ операторининг чарчашини олдини олиш.

Компьютерда ишлаш вақтида инсоннинг чарчаш сабаблари.

Замонавий автоматлаштирилган ишлаб чиқаришда инсон операторнинг психология ва физиологияси асосий ролни эгаллайди. Ишлаб чиқаришда меҳнат шaroитини яхшилаш ва илмий асосда аниқлаш, меҳнатни тўғри ишлаш маромини таъминлаш, меҳнат тартиби ва дам олишни ташкил қилиш зарур.

Компьютер билан ишлаш вақтида инсон куйидаги факторлардан чарчайди:

- Экранни ёруғлиги;
- Контраст ва фон ўртасидаги аниқлиги;
- Компьютерда ишлаш пайтидаги иссиқликдан нурланиши;
- Компьютерда нурланишнинг инсонга таъсири;
- Компьютер бузуклиги.

Дисплей билан ишлайдиган ЭҶМ операторларида асосан, бош оғриши, бел оғриқлари, елка, орқа оғриши, кўз чарчашини кузатилади.

Меҳнат шaroитини яхшилаш чора-тадбирлари.

Меҳнат шaroитини яхшилаш мақсадида ташкилий, гигиеник, техникавий чора-тадбирлар ишлаб чиқилади ва ишчи-хизматчилар орасида меҳнат гигиенаси нормалари, қоидаларига риоя этиш бўйича ташвиқот ишлари олиб бoрилади.

Ишлаб чиқариш санитарияси санитария-технологик, ташкилий тадбирларни фойдалайди ва ишлаб чиқаришда соғлом меҳнат шaroитларини таъминлайди. Шунинг мақсадида ишчи-хизматчиларнинг саломатлигига таъсир қилувчи технологик жараёнлар ва ускуналардаги камчиликларни йўқотиш йўллари ишлаб чиқарди. Бунинг учун саноат корхоналарида техника таракқиёти ютуқларидан унумли фойдаланишни, жараёнларни олисдан бошқариш ва ишчиларни зарарли муҳитда ишлашларининг олдини олишни, ускуналарни, қурилмаларни очиқ майдонда жойлаштиришни, хавфсиз таркибини текшириб туришни, қўл меҳнатини талаб қиладиган ишларда имкон борича механизация воситалари ва замонавий ускуналарни қўллашни, химоват воситаларидан фойдаланишни зарур деб ҳисоблайди.