

**ЎЗБЕКИСТОН РЕСПУБЛИКАСИ АЛОҚА,
АХБОРОТЛАШТИРИШ ВА ТЕЛЕКОММУНИКАЦИЯ
ТЕХНОЛОГИЯЛАРИ ДАВЛАТ ҚЎМИТАСИ**

**ТОШКЕНТ АХБОРОТ ТЕХНОЛОГИЯЛАРИ УНИВЕРСИТЕТИ
НУКУС ФИЛИАЛИ**

Ахборот технологиялари кафедраси

**Компьютер инжиниринги факультетининг
«Корхоналар сервиси» таълим йўналиши
4-курс талабаси Ярылкапов Дилшадбекнинг**

БИТИРУВ МАЛАКАВИЙ ИШИ

***Мавзу: «Нукус универсал базаси» шўъба корхонаси фаолиятини
автоматлаштириш.***

Илмий раҳбар:

**«Ўзбекистон почтаси»
ОАЖ Қорақалпоғистон
филиали бош ҳисобчиси
К.Рамбергенов**

асс. А.Рысназаров

Кафедра мудири:

т.и.к. Арзымбетов Т.З

МУНДАРИЖА

Кириш.....	3
§1. «Нукус универсал базаси» шўъба корхонаси бизнес-жараёнининг функционал моделини тузиш.	
§2. Клиент-сервер технологиясидаги ахборот тизимлари ва уларни яратувчи дастурлаш тиллари.	
§3. Ахборот тизимининг маълумотлар базасини лойиҳалаш ва уни MySQL да яратиш.	
Хулоса.....	
Фойдаланилган адабиётлар рўйхати.....	

КИРИШ

Компьютер ва алоҳа технологияларининг тезлик билан равожланиши кундаликли турмушумизда асосий истимол товарларини етказиб берувчи савда кархоналарининг мекемелериниң бизнес жараёнларини автоматлаштиришнинг давр талаби эканлигини кўрсатмоқда. Бугунги кунда дунёда электрон тижорат инқилобий даражада ривожланмоқда. Давлатимизда бу соҳада ижобий равожланишлар бўлаётканлиги, унинг савдо ишларида енгилликлар бериб қуймосидан, шарнамаларга кетадиган вақтларни, истимолшиларнинг сайлаш имкониятларинида оширишда ютуқларга эгалиги курунмоқда.

Мамлакатимизда ахборот ва телекоммуникация технологияларини равожлантириш буйича белгиланган мақсад ва вазифалар, ишланаятган ишлар, буларнинг ҳаммаси мамлакатимиздаги соҳаларнинг яқин йилларнинг ишида замон билан ҳамнафас янги ахборот ва телекоммуникация технологиялари тизимларига утишининг гаровидир.

Ири саўда мекемелери электрон тижарат тизимиға өтиўи ушын дэслеп өзиниң ишки бизнес жараёнларини автоматлаштирилган турға утказиши керак бўлади.

Менинг битирув малакавий ишим «Нукус универсал базаси» шўьба корхонаси фаолиятини автоматлаштиришға қаратилган бўлиб, кириш, учта параграф, хулоса, фойдаланилган адабиётлар руйхати ва қўшимча иловалардан туради.

Биринчи параграфда «Нукус универсал базаси» шўьба корхонаси бизнес-жараёнининг урганилиб унинг функционал моделини тузиш масаласи кўрилади.

Еккинчи Клиент-сервер технологиясидаги ахборот тизимлари ва уларни яратувчи дастурлаш тилларидан бўлган РНР тили, ушбу технологияда кенг

кулладиган MySQL маълумотлар базасини бошқариш тизими ва у билан ишлашни енгиллаштирувчи phpMyAdmin дастури ўрганилади.

Учинчи параграфда ахборот тизимининг маълумотлар базасини лойиҳалаш, уни MySQL да яратиш ва тизимнинг модуллари яратлиши ҳал қилинади.

Битирув малакавий ишнинг сўнгида дастур матни ва натижалар келтирилган.

«Нукус универсал базаси» шўъба корхонаси бизнес-жараёнининг функционал моделини тузиш.

Ахборот коммуникация технологияларининг равожланиши куплаган савдо корханаларининг бизнес-жараёнларида янги технологияларни фойдаланишига туртки булмокда. Сабаби савдода рахобат бардашлик катта аҳамиятга эга. Бу улгуржи савдо карханаларинингда иш фаолиятларида янги ахборот комуникация технологияларини куллашнинг давр талаби эканлигини курсатмокда.

Улгуржи савдо - бозор иқтисодиёти шароитида товар харакати жараёнини тезлаштириш ва зарур бўлган интенсивлаштиришни таъминловчи асосий звенодир. Улгуржи савдо товарларни тақсимлаш каналлари бўйича харакатни ташкил этиш билан бирга товарларни ишлаб чиқарилишини ва истеъмолини бир-бирига мос холда ишлашига ёрдам беради.

Хозирги шароит улгуржи савдони қайта қурилишини талаб қилади. Шунинг учун улгуржи савдони ривожлантиришда давлат сиёсатини асосий узоқ муддатли масалаларидан - бу тузилиши бўйича қайта ўзгариши, кичик тадбиркорлик йўналишида улгуржи поғонада щизмат кўрсатишидир.

Оширги йилларда улгуржи операцияларни щажми қисқарди.

Чакана савдо тармоқларини товар билан таъминлаш щам кескин қисқаради.

Шу туфайли, давлат сиёсатини иккинчи босқичи бўлиб улгуржи ташкилотларни ролини қисқарришини тўхтатиш ва улгуржи операцияларини щажмини стабиллаштириш бўлиб щисобланади.

Охирги йилларда мавжуд бўлган улгуржи савдони моддий-техника базаси бир нечта ўн йилликлар давомида вужудга келган. Омбор хўжалиги янгилашни талаб қилади. Бу щолда нафақат уларни янгисини қуриш, балки эскисини реконструкция қилиш ва техник жищатдан қайта жищозлаш, моддий-техника баъзасини ривожлантиришни талаб этади.

Давлат сиёсатини кейинги масалаларидан бири улгуржи савдода рақобатбардошликни ривожлантириш ва улгуржи савдо фаолияти бозорида монополияга чек қўйиш, мащаллий молларни кўпроқ сотиш.

Улгуржи савдони олдига қўйиладиган янги масалалар, янги мақсадлар кўзда тутилади:

- товархаракати каналларини ривожланган структурасини тузиш;
- товар оқимини керакли интенсивлашини қўллаш;
- товархаракати жараёнини молиявий таъминлашни қўшимча манбаларини шакллантириш;
- муомала щаражатларини камайтиришни таъминлаш.

Хўжалик муносабатларининг бозорга йўналган тизимида улгуржи савдо вазифалари ҳам ўзгариб боради. Улгуржи савдони улгуржи харидорларга нисбатан вазифалари қуйидагилар :

- талаб ва таклифни бащолаш;
- ишлаб чиқариш ассортиментини савдо ассортиментига айлантириш;
- товар защираларини тўплаш ва сақлаш;
- товарларни етказиб бериш;
- кредитлаш;
- инщормацион ва консалтинг щизматини амалга ошириш.

Улгуржи савдони товар таъминловчиларга нисбатан вазифалари :

- тижорат фаолиятини бир жойга тўпланиши;
- товарга бўлган мулк шаклини ўзгариш жараёнини қуллаб-қувватлаш;
- товархаракати жараёнини инвестиция ёрдамида таъминлаш;
- тижорат вазифасини камайтириш;
- маркетинг щизмати.

Улгуржи савдони мақсадли ва вазифали муносабатларини қайта йўналтириш нафақат савдода ислошотларни чуқурлаштиришни, балки бутун истеъмол бозорини стратегик мустақамлигини таъминлаб беради.

Улгуржи ташкилотларнинг типлари.

Аввал улгуржи савдо давлат ва кооперативга бўлинар эди. Хозир эса шар хил мулкчилик шаклида тақдим этилади.

Мамлакатда ўтказилаётган улгуржи савдонинг ривожланиш тузилмавий сиёсати узоқ муддатли мақсад қилиб тузилманинг бозор хилма-хиллигини таъминлашни қуяди. Улар улгуржи савдо фаолиятини амалга ошириш билан бирга истеъмол бозорини товар билан тўлдиришига, олиб келади, уларни тўсиқларсиз тақсимлаш қаналлари бўйлаб ҳаракатланишга шароитлар яратиб беради, миллий товар ишлаб чиқарувчиларини кўпайтиради.

Улгуржи савдо ташкилотлари умуммиллий (федерал) ва минтақавий (регион ичидаги)га бўлинади.

Умуммиллий даражасидаги улгуржи ташкилотлар товарларни бутун мамлакат бўйлаб сотади. Улар улгуржи савдонинг ички тармоқ тузилмасини асосий мощияти бўлиб шисобланади. Уларнинг асосий вазифаси бўлиб, хорижий товар ишлаб чиқарувчи ва таъминотчиларига ва мамлакатдаги йирик ишлаб чиқарувчиларга шизмат қилишга мўлжалланган ва товар ҳаракатини зарур бўлган йўллари ташкил этиш шисобланади.

Умуммиллий улгуржи ташкилотларни асосий ташкилий - шуқуқий шакли бўлиб очик типдаги акционер жамиятлари шисобланади, уларнинг бошланғич маблағларида давлат шам иштирок этиши мумкин.

Бундан ташқари давлат концернлари шам бўлади. Товарларнинг улгуржи сотиш жараёнини регионал даражадаги улгуржи ташкилотлар яқунлайди. Улар товарларни бевосита товар ишлаб чиқарувчилардан ва федерал даражасидаги улгуржи ташкилотлардан шарид қилиб олишади ва уларни ўз фаолият доирасида улгуржи шаридорларга еткази.

Уларнинг шизмат кўрсатиш асосий масаласи бўлиб регионал товар бозорларини товар билан таъминлашдир.

Минтақавий (регионал) даражасидаги ҳар хил бирлашмалар ҳам кенг авж олмоқда. Улар бир бири билан боғланган улгуржи савдо компаниялари ёки ихтиёрий улгуржи-чакана занжирлари сифатида вужудга келиши мумкин.

Улгуржи ташкилотларнинг турлари ва вазифалари.

Улгуржи ташкилотларнинг турлари қуйидагиларга бўлинади:

- мустақил бўлган улгуржи савдогарлар - улар сотиб олиш-сотиш операцияларини тўлиқ комплексини амалга оширади, бу шолатда товар улгуржи поғонани мулкига айланади;
- воситачи улгуржи тузилмалар - уларда товарлар ўз мулкига айланмайди (дистрибьюторлар);
- улгуржи айланма ташкилотлари.

Мустақил улгуржи савдогарларнинг вазифаси товар харакатини ўрта поғонасида йирик товар ишлаб чиқарувчиларни ва чакана савдогарларини бозорга чиқариш шарт-шароитларини ташкил этишдан иборатдир. Уларни кўрсатадиган хизмат турлари шар хил бўлиши мумкин.

Воситачи улгуржи тузилмаларда ташкилот-брокер, ташкилот-агент (дистрибьютор) сифатида иштирок этади, ўз клиентини номидан ва унинг ҳисоб рақами орқали фаолият кўрсатади. Уларнинг асосий фаолияти маълумотлар билан таъминлашдир.

Улгуржи айланма ташкилотчилари - бу товар биржалари, улгуржи ярмаркалар, аукционлар, улгуржи озиқ-овқат бозорларидир. Улар - улгуржи инфратузилмасини мушм элементидир ва уларни асосий вазифаси бўлиб улгуржи савдони ташкил этиш учун шароитлар яратиш шисобланади.

Биз «Нукус универсал базаси» шўъба корхонасининг иш жараёнини урганганимизда қуйидаги маълумотларни олдик.

Корхона Нукус шаҳрида жойлашган булиб, битта базада етгита омборханага эга ва асосан улгуржи савдо билан шуғулланади.

База директори ҳар-бир товарнинг туридан омборхоналардаги захираларни белгилайди ва сотиш булими менеджери орқали товар омборхоналардаги захираларни назорат қилади.

Сотиш булими менеджери товарларга боҳаларни белгилайди, харидорлар билан шартномалар тузишни ташкиллаштиради, шартномадаги шартлар буйича пулини тулаган харидорларга курсатилган товрларни омборхонадан олишга хужжат ёзади ва узидаги омборхоналардаги товарларнинг ҳажмидан иш куни оқирида омборхона мудирларининг ҳисобати буйича камайтириб захираларни ҳисоблаб утиради. Ундаги маълумотлар орқали прайст-лист тузилади. Бу прайст-лист иш кунининг бошига реал ҳолатда булади ва бу орқали кархонага келган хориदारлар товарлар ва уларнинг баҳолари билан танишади.

Хоридар (банкада ҳисоб раҳамига эга) даслаб сотиш бўлимига келиб узига керак булган товарларига ҳисоб-фактура ёздиради. Уннан кейин бухгалтерияга бориб кархонага утказган маблағиғи етарли булса, унда бухгалтерядан ҳисоб-варағига белги (имзо) ёздириб, сунг кархона бошлиғига имзо қуйдириб, уннан кейин оқирида омборхонага бориб товарларини олади.

Омборхана мудир омбордаги товарларнинг сақланишига жабовгар булиб, омборханадаги камомадлар унинг зиммасида булади. У келган товарларни узига қабул қилиб олиб, харидорларга ҳисоб-варақаси буйича товарларни бериши, узидаги товарларнинг (ҳар-бир товар буйича алоҳида) карточкаларидан камайтириб бориши, кун оқирида юбарилган товарлар буйича сотиш бўлимига ҳисобат бериши ва ой оқирида складдаги товарларни

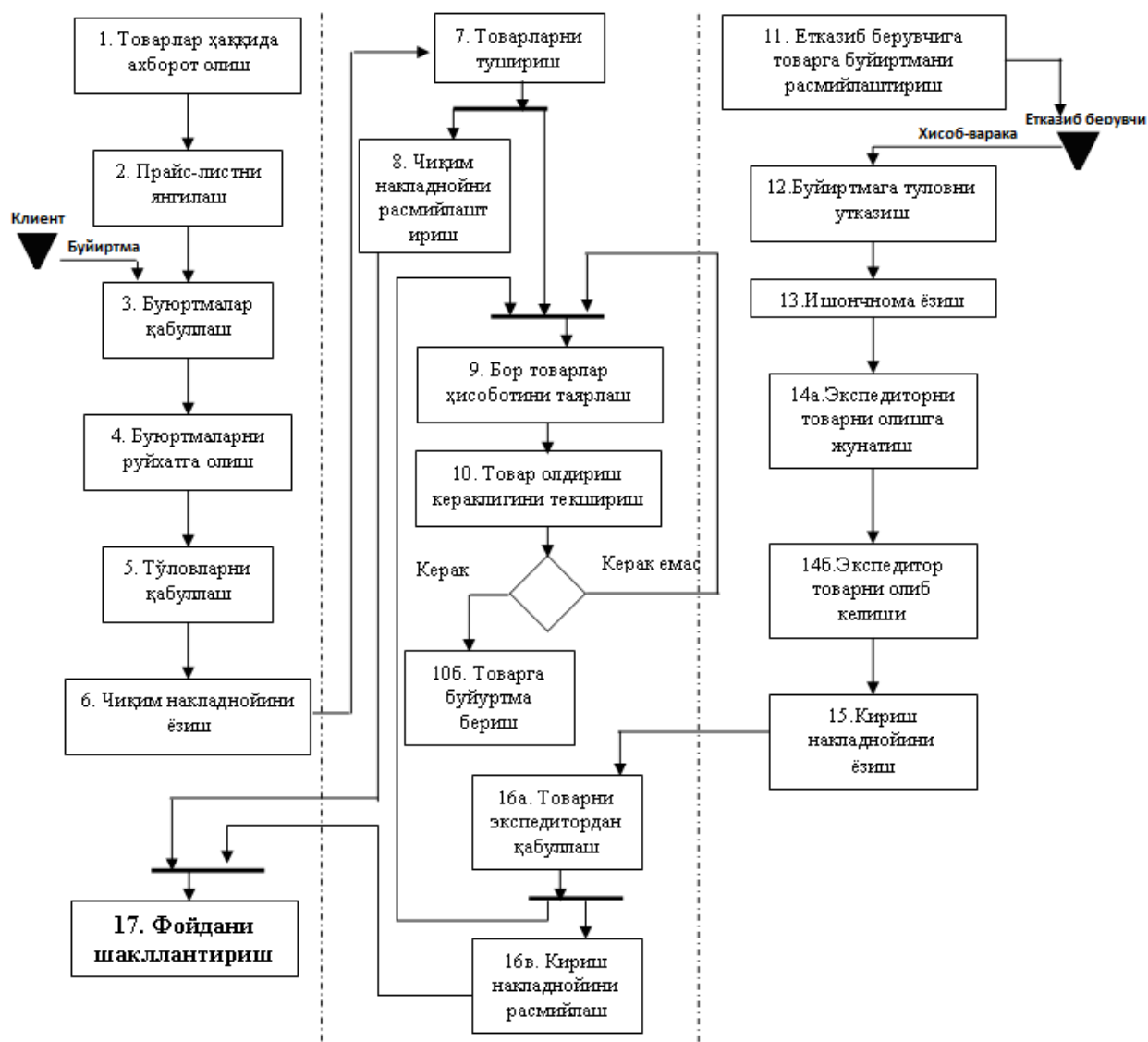
узи ревизия қилиб, бухгалтерияга ҳисобат бериши шарт. Янада омбордаги товарларнинг фойдаоланишга яроқлилик муддатида жавобгар бўлади.

Янада кархонада товарларни етказиб бериш бўйича ходимлар бор бўлиб ўлар экспидиторлар бўлади. Улар кархонанинг бошқа кархоналар ёки ишлаб чиқарувчилардан сотиб олган товарларини етказиб бериш билан шуғулланади. Кархона директори товарларнинг сотилиш динамикаси ва захираларга бўйича қанақа товар олдириш бўйича қарор қабул қилади.

Етказиб берувчилар билан шартномалар тузилади ёки йил бошидан тузилган шартномалар бўйича иш қилинади. Уннан кейин бошлиқ махсус ходим орқали етказиб берувчи кархонага буюртма юбаради. Етказиб берувчи кархона буюртмани қабуллаб, ҳисоб-варақаси ёзади ва кархонамизга тулов учун юбаради. Ходим бу ҳисоб-варақасини олиб бухгалтерияга келтиради, ва бухгалтерия етказиб бериш кархонаси ҳисобига пул маблағини утказди.

Туловдан сунг экспедиторга кархона тамонидан ишончнома ёзилиб у бўйича экспедитор товарни етказиб берувчи кархонадан олиб келади.

Биз энди юқоридагиларни ҳисобга олган ҳолда яратаётган ахборот тизимимиз учун «Нукус универсал базаси» корхонасининг иш фаолиятининг функционал моделини тузамиз.



Тузилган модел асосида тизимни сервер-клиент технологиясида яратишни режаластирсак мақсадга мувафиқ булади. Бунинг учун биз веб технологияларни урганишимиз ва анализлашимиз зарур.

§2. Клиент-сервер технологиясидаги ахборот тизимлари ва уларни яратувчи дастурлаш тиллари.

PHP скриптили дастурлаш тили.

PHP дастурлари икки усулда бажарилиши мумкин: **Web**-сервер томонидан сценарий иловаси ва консол дастури сифатида. Бизнинг мақсадимиз **web** иловаларни дастурлаш бўлгани учун асосан биринчи усулни кўрамиз. **PHP** одатда Интернет билан боғлиқ дастурлар яратиш учун ишлатилади. Лекин **PHP** дан командасатрлар интерпретатори, асосан ***nix** тизимларда фойдаланиш мумкин. Охиргиси **CORBA** ва **COM** интерфейслар ҳамда **PHP-GTK** кенгайтмаси ёрдамида мумкин. Бу холда қуйидаги масалаларни ечиш мумкин:

- Интерактив команда қаторлари ёрдамида иловалар яратиш;
- Кросс-платформали **GUI** иловаларни **PHP-GTK** библиотекаси ёрдамида яратиш;
- **Windows** ва **Linux** учун баъзи масалаларни автоматизация қилиш
Серверга броузернинг мурожжат қилишида ёрдамида **php**-сценарийлари бажарилишини кўриб чиқамиз.

Аввал броузер. **php** кенгайтмали саҳифани сўрайди, сўнгра **web**-сервер дастурни **PHP** машинадан ўтказди ва натижани **html**-код шаклида қайтаради. Агар стандарт **HTML** саҳифани олиб, кенгайтмасини. **php** га ўзгартирилса ва **PHP** машинадан ўтказилса, фойдаланувчига ўзгартирмасдан қайтаради. Буф файлга **PHP** командани қўшиш учун, **PHP** командалани махсус теглар ичига олиш керак. Бу тегларнинг **4** хил шакли мавжуд бўлиб, ихтиёрийсидан фойдаланиш мумкин:

1. XML қайта ишлаш инструкцияси:

`<?php`

`...`

`?>`

2. SGML қайта ишлаш инструкцияси:

```
<?
```

```
...
```

```
?>
```

3. HTML сценарийлари қайта ишлаш инструкцияси:

```
<script language="php">
```

```
...
```

```
</script>
```

4. ASP услубидаги инструкция:

```
<%
```

```
...
```

```
%>
```

Биз XML ёки SGML услубига риоя қиламиз. Хусусан бирор блок ичида PHPдан чиқиш мумкин, фақат кейинчалик яна унинг ичига кириб кодни тугатиш шарти билан, қуйидаги конструкция мумкин:

```
<?
```

```
if(5<3){echo("<p>Hello, world!<p>");
```

```
?>
```

```
<p>Hello!</p>
```

```
// бу қатор PHP коди сифатида қаралмайди
```

```
// ва код блоки бажарилаётган бўлса чиқарилади
```

```
<?
```

```
echo("<p>Hello, world!<p>");
```

```
}
```

```
?>
```

PHP да **echo** командаси **web** – саҳифаларда учрайдиган ҳар қанақа маълумотни(матн, HTML ажратувчи символи, сон) чиқариш учун қўлланади. Унинг маъноси мисолда лўрсатилган.

Изохлар

PHP тилида изохларни жойлаш учун бир неча усуллар мавжуд. энг соддаси иккилик слеш (//) дан фойдаланиш, шундан сўнг PHP сатрлар

охиригача ёзилганни ўтказиб юборади. Бундан ташқари **C** (**/*...*/**) услубидаги кўп қаторли изохлардан фойдаланиш мумкин. Бир қаторли изохлар учун (**#**) символдан фойдаланиш қулай. (**UNIX script** тилларидаги изох).

```
<php echo("<p>Hello</p>"); // изох
echo("<p>Hello</p>"); # изох /*бу ҳам изох */
?>
```

Шуни эсдан чиқармаслик лозимки **PHP** услуби изохлари фақат **PHP** чегаранишлари орасида таъсир қилади. Агар **PHP** бу изохлар символларини чегаранишлари ташқарисида учраса, уларни бошқаматига ўхшаб, **html**-саҳифага жойлаштиради. Масалан:

```
<php
echo("<p>Hello</p>"); // нормал изох?>
// бу изох браузерда кўринади.
<!-- HTML изохи.
```

Бу изох HTML кодда кўринади, браузерда эмас -->

Изохларни фақат оператор охирига эмас, қуйидагича жойлаш ҳам мумкин:

```
<? $a="Hello, world";echo strstr($a, "H");
// бу функцияни кейинчалик қараб чиқамиз
?>
```

Ўзгарувчилар ва константалар

RNR да ўзгарувчилар доллар (**\$**) белгисидан бошланади. Бу символдан ихтиёрий сондаги ҳарф, рақам ва остига чизиқ символлари келиши мумкин, лекин биринчи символ албатта ҳарф бўлиши керак. Шуни есда тутиш керакки, **RNR**да ўзгарувчиларнинг номлари калит сўзлардан фарқли регистрга боғлиқдир. **RNR** да ўзгарувчиларни таърифлаганда ошкора типини кўрсатиш шарт эмас ва дастур давомидан ўзгарувчи ҳар хил типларга эга бўлиши мумкин. Ўзгарувчи унга қиймат берилганда инициализация қилинади ва дастур бажарилгунча мавжуд бўлади. Яъни **web**-саҳифа холида то сўров тугамагунча.

Ташиқи ўзгарувчилар

Клиент сўрови веб-сервер томонидан тахлил қилиниб, **RNR** машинага узатилгандан сўнг, у сўровга тегишли маълумотларни ўз ичига олган ва бажариш давомида мурожаат қилиш мумкин бўлган бир неча ўзгарувчиларни яратади. Олдин **RNR** сизни тизимингиз атроф мухит ўзгарувчиларини олади ва ш у номдаги ва шу қийматдаги **RNR** сценарийси атрофидаги ўзгарувчиларни яратади, токи сервердаги сценарийларга клиент тизими хусусиятлари билан ишлаш мумкин бўлсин. Бу ўзгарувчилар **\$HTTP_ENV_VARS** ассоциатив массивга жойлаштирилади. Табиийки **\$HTTP_ENV_VARS** массиви ўзгарувчилари тизимга боғлиқдир (чунки улар аслида атроф мухит ўзгарувчиларидир). Атроф мухит ўзгарувчилари қийматларини сизни машинангиз учун енв (**Unix**) ёки **set (Windows)** командаси ёрдамида кўришингиз мумкин. Сўнгра **RNR** у **GET**-ўзгарувчиларнинг гуруҳини яратади. Улар сўров сатрини тахлил қилишда яратилади. Сўров сатри **\$QUERY_STRING** ўзгарувчида сақланади ва сўралган **URL** даги "?" символдан кейинги информациядан иборат. **RNR** сўров сатрини **&** символлари бўйича алоҳида элементларга ажратади, ва ҳар бир элементда "=" белгисини қидиради. Агар "=" белгиси топилган бўлса, тенглик чап томонидаги символлардан иборат ўзгарувчи яратади. Қуйидаги формани кўрамиз:

```
<form action="http://localhost/PHP/test. php" method="get">
```

```
HDD: <input type="text" name="HDD"/><br>
```

```
CDROM: <input type="text" name="CDROM"/><br>
```

```
<input type="submit"/>
```

Агар сиз бу формада **HDD** қаторда **"Maxtor"**, **CDROM** қаторда **"Nec"** терсангиз, қуйидаги сўров шаклини хосил қилади:

```
http://localhost/PHP/test. php?HDD=Maxtor&CDROM=Nec
```

Бизнинг мисолимизда **RNR** қуйидаги ўзгарувчиларни яратади:

```
$HDD="Maxtor" va $CDROM="Nec".
```

Сиз ўзингизни **script**ингиздаги (бизда – **test. php**) бу ўзгарувчилар

билан оддий ўзгарувчилар билан ишлагандек ишлашингиз мумкин. Бизнинг мисолимизда улар экранга чиқарилади:

```
<?
echo("<p>HDD is $HDD</p>");
echo("<p>CDROM is $CDROM</p>");
?>
```

Агар саҳифа сўрови **POST** усули ёрдамида бажарилса, **POST-**ўзгарувчиларнинг гуруҳи яратилиб, интерпретация қилинади ва **\$HTTP_POST_VARS** массивга жойлаштирилади.

Константалар

Константалар **RNR ddefine()** функцияси ёрдамида эълон қилинади:

```
define(CONSTANT, value)
```

Бу функция биринчи параметри – констант номи, иккинчиси – унинг қиймати. Константадан фойдаланилганда номи бўйича илова қилинади:

```
<? define(CONSTANT1, 15);
define(CONSTANT2, "\x20"); // kod probela
define(CONSTANT3, "Hello");
echo(CONSTANT1);
echo(CONSTANT2);
echo(CONSTANT3);
?>
```

Одатга кўра константалар номлари юқори регистр ҳарфлари билан ёзилади. Бу фақат одат бўлса ҳам унга риоя қилишни маслаҳат берамиз, чунки яхши одатларга риоя қилмайдиган дастурчилардан ёмон дастурчилар чиқади. Константалар аниқланганлигини **defined()** функцияси ёрдамидекшириш мумкин:

```
<? define(CONSTANT, "Hello");
if(defined("CONSTANT"))
{
    echo("<p>CONSTANT is defined</p>");
}
```


}?>

RNR да маълумотлар типлари. Типларни ўзгартириш

Юқорида айтилганидек **PHP** тилида битта ўзгарувчини дастур бажарилишда вомида сатр ёки сон сифатида ишлатиш мумкин. Шу билан бирга **PHP** тилида ўзгарувчилар билан ишланганда ошкор кўрсатилиши мумкин бўлган асосий маълумотлар типлари тўплами мавжуд:

- **integer;**
- **string;**
- **boolean;**
- **double;**
- **array, object;**

RNR ўзгарувчига тайинлаган типни қайтарувчи **gettype()** функцияси мавжуд:

<?

```
$var="5";$var1=5;
```

```
echo(gettype($var));echo "<br>";
```

```
echo(gettype($var1));
```

?>

Биринчи холда **RNR string** қайтаради, иккинчи холда **integer**. Типни ошкора ўрнатувчи **settype()** функцияси мавжуддир:

<?

```
$var="5";
```

```
echo(gettype($var));
```

```
settype($var, integer);
```

```
echo "<br>";
```

```
echo(gettype($var));
```

?>

Коднинг бу фрагментини бажариш, аввалгисини бажариш билан бир хил натижага олиб келади. **RNR** тилида **settype()** функциясидан ташқари типни ўзгартириш операторидан фойдаланиш мумкин. Типни ўзгартириш қавсларга олинган янги типни кўрсатиш орқали бажарилади: **\$var=(int)\$var;**

RNR қуйидаги кодни бажариш натижасида, **integer** қайтаради:

```
<?
```

```
$var="5"; // string тип
```

```
$var=(int)$var; // int га ўзгартирамиз
```

```
echo(gettype($var));
```

```
?>
```

Операторлар

Арифметик операторлар

+ Қўшиш

- Айириш

* Кўпайтириш

/ Бўлиш

% Модул бўйича қолдиқни ҳисоблаш (масалан: 5 % 2=1)

Разрядли операторлар

Бу операторлар гуруҳи бутун қийматли операторларнинг битли тасвирлари билан ишлайди. Асосан бу операторлар мантиқий байроқлар тўпламини яратиш учун ишлатилади. Бизнинг мисолларимизда бу операторларни кам ишлатамиз, лекин қуйида шу операторлардан фойдаланишга мисол келтирамиз.

& Разрядли кон'юнкция (**AND**)

| Разрядли диз'юнкция (**OR**)

^ Разрядли ажратувчи ЁКИ (**XOR**)

~ Разрядли инкор (**NOT**)

<< Чап бутун қийматли операнд битли тасвирини ўнг бутун қийматли операнд қийматига тенг разрядлар сонига чапга силжитиш

>> Чап бутун қийматли операнд битли тасвирини ўнг бутун қийматли операнд қийматига тенг разрядлар сонига ўнгга

Мисол:

```
<? echo(4<<2); // тенг 16
echo"<br>";echo(5>>1); // тенг 2
echo"<br>";echo(6&5); // тенг 4
echo"<br>";echo(6|5); // тенг 7
echo"<br>";echo(6^5); // тенг 3
>?
```

Биринчи мисолда **4** сонининг иккилик коди **100** га тенг, чапга **2** разряд сурилганда **100** код **10000** га тенг бўлади. Бу ўнлик **16** га тенг бўлади. Бошқа мисолларни ҳам шунга ўхшаб қараб чиқиш мумкин. Чапга n позицияга суриш 2^n га кўпайтиришга тенг, ўнгга суриш 2^n мартакамайтириб, каср қисмини ташлаб юбориш натижасига тенг (шунинг учун $5>>1$ тенг **2**). Бу операторлар асосий вазифаси тез ҳисоблашлар. Улар аппарат даражасида қўллангани учун, бу операторлардан фойдаланиб тузилган алгоритмлар ўта унумдор бўлади.

Мантиқий операторлар

or ili 	ЁКИ
Xor	Рад қилувчи ЁКИ
and ili &&	ВА
!	Инкор

Бу гурухоператорлари разрядли операторлардан фарқли мантиқий (**boolean**) ўзгарувчилар билан ишлайди ва бошқарувчи конструкциялар:цикл лар ва шартларда ишлатилади. Мантиқий ўзгарувчилар ёки тўғрироғи **Boolean** типигаги ўзгарувчилар фақат икки қиймаитга эгалар: **true(пост)** ва **false(ёлғон)**. Ифодаларда**true** ва **false** қийматларини **1(0** дан фарқли ихтиёрий сон) ва **0** гаалмаштириш мумкин. Бу имконият махсус **true** и **false**

қийматларига эга бўлмаган С тилининг меросидир.

Мантикий операторлар **Boolean** типдаги ўзгарувчилар учун оддий ўзгарувчилар учун қўшиш, айириш операторлари ролини ўйнайди. Оператор **if(cond)** аввал **cond** шартни текширади, агар унинг қиймати **true** бўлса **if** дан кейинги фигурали қавс ичидаги код бажарилади, агар **false** бўлса **else** операторидан кейинги код бажарилади. Агар **cond** қиймати **false** бўлса тескарис.

```
<php $flag=true; // Рост
if($flag) { echo "<p> flag ўзгарувчиси true қийматга эга </p> "; }
else { echo "<p> flag ўзгарувчиси false қийматга эга</p>"; }
?>
```

Script ишлаши натижаси қуйидаги кўринишга эга: flag ўзгарувчиси **true** қийматга эга. Агар

```
$flag=true; // Рост қатор
```

```
$flag= false; // Ёлғон
```

Қаторга алмаштирилса тег қайтарадиган қиймат ўзгаради: flag ўзгарувчиси **false** қийматга эга. Яъни биз **else** блока кирдик, чунки **if** операторида шарт **ёлғон**. Шарт битта бўлса, ҳаммаси содда. Шартлар кўпроқ бўлса мантикий операторлар ишга тушади:

```
<php $flag1=true; // Рост $
flag2=true; // Рост
if($flag1 && $flag2) // И $flag1 ва $flag2 рост
{
echo "<p>Шарт: true (Иккала байроқ рост)</p>"; // Ха
}
else
{
echo "<p>Шарт: false (Байроқлардан бири ёлғон)</p>";
// Йўқ, биттаси ёки иккови ёлғон
}
```

?>

Бу **script** қуйидагини чиқаради: Шарт: **true** (Иккала байроқ рост). Агар операторлардан бири **false** бўлса(ёки иккови) – бқуйидагини чиқаради: Шарт: **false** (Байроқлардан бири ёлғон). Албатта шартлар билан ўйнаб кўринг: агарда сиз шартли операторларни яхши билмасангиз ҳеч қайси дастурлаш тилида яхши дастурчи бўлолмайсиз. Агарда сизга шартли оператор керак бўлмаса, демак сизга дастурлаш тили ҳам керак эмас. **&&** (ВА) ни **||** (ЁКИ) гаалмаштирамиз

```
<php $flag1=true;
// Rost $flag2= false; // Rost
if($flag1 || $flag2)
// Агар байроқлардан жуда бўлмасабири ($flag1 $flag2) рост
{
echo "<p>Шарт: true (Байроқлардан бири рост)</p>";
// Ха
} else
{echo "<p>Шарт: false (Икала байроқ ёлғон)</p>";
// Ўқ, Иккови қиймати false
}
?>
```

Натижа: Шарт: **true**. Ва ҳамда Ёки операторларининг икки кўриниши мавжуд бўлиб, уларнинг устиворлиги ҳар хилдир. Мантикий операторлар устиворлиги: **or** > **xor** > **and** > **||** > **&&** > ! Базида шартларда устиворликни ошкор кўрсатиш учун қавслардан фойдаланилади:

```
<php $flag1=true; $flag2=true; $flag3= false; if($flag1 $flag2 || $flag3) {
echo "<p>Шарт рост <p>"; } else{
echo "<p>Шарт ёлғон <p>"; } ?>
```

Натижада «Шарт рост» жумласи чиқарилади, чунки устиворлик асосидааввал **&&** оператори бажарилади. Бунинг сабаби икала байроқ **true** бўлгани учун–**true** қайтаради, сўнггра бу ифода **\$flag3 (true || \$flag3)** билан

солиштиради ва яна **true** қайтаради, чунки солиштирилаётганлардан бири **true** га тенг.

Java Script сценарийларини тузиш тили

Java тили Sun фирмаси томонидан 1995 йилда ишлаб чиқилган. Java–кроссплатформали иловаларни ишлаб чиқиш тили. Кроссплатформали иловалар деганда, платформага боғлиқ бўлмаган иловалар тушунилади. Иловалар Java тилида ёзилади, байт-код деб аталадиган муайян оралиқ кодга компиляция қилинади, (у процессор командалари билан боғлиқ эмас) ва байт-кодни тушунувчи Java виртуал машинаси ёрдамида мижозларнинг компьютерларида бажарилади. Java тилидаги иловалар жуда ихчам ва фойдаланувчи учун хавфсиздир, чунки виртуал машина алоҳида тузилишга эга бўлганлиги туфайли, мижоз компьютерида ишлаш жуда қулайдир.

Кичик Java иловаларини Web-саҳифаларга тўғридан-тўғри ўрнатиш мумкин; уларни кўздан кечириш чоғида браузер иловани автоматик тарзда юклайди ва виртуал машинани ишга туширади. 1996 йилда Netscape компанияси Java негизида JavaScript тилини ишлаб чиқди. Бу тилдаги дастурлар (скриптлар) HTML кодига тўғридан-тўғри ўрнатилиши мумкин эди. JavaScript–Java тилининг скриптли версияси. Java тилидан фарқли ўлароқ, JavaScript компиляция қилинадиган эмас, балки интерпретация қилинадиган тил эди, яъни Java виртуал машинаси компиляция қилиш натижасида олинган байт-кодни эмас, балки JavaScript кодининг ўзинин бажарарди. Шундай қилиб, компиляторга эҳтиёж соқит этилди ва дастурлар ишлаб чиқиш жараёни жадаллаштирилди.

JavaScript–мижоз томонида ҳам, сервер томонида ҳам амал қилувчи ўрнатиладиган иловаларнинг объектга йўналтирилган тили. Тил синтаксиси Java тилининг синтаксисига жуда ўхшаш, шу боис у кўпинча Java-симон тил деб аталади. Мижоз иловалари Web ҳужжатларни кўздан кечириш браузер

томонидан фойдаланувчининг машинасида бажарилади, сервер иловалари эса серверда бажарилади

JavaScript сценарийсини HTML саҳифасига бир неча усул билан ўрнатиш мумкин:

1. HTML тилининг `<SCRIPT>` теги ичидаги тил операторини юклаш.
2. `<SCRIPT>` тегининг SRC параметрида JavaScript кодли файлни кўрсатиш.
3. JavaScript ифодаларидан HTML теглари параметрларининг ифодалари сифатида фойдаланиш.
4. HTML теглар асосида ҳодисаларга ишлов берувчини аниқлаш.

Тег `<SCRIPT>` дан фойдаланиш

HTML саҳифасига JavaScript сценарийсини ўрнатиш учун HTML тилининг спецификациясига `<SCRIPT> ... </SCRIPT>` тег-контейнери киритилди. Бу тег-контейнер ичида JavaScript тилининг операторлари жойлашиши мумкин. Одатда бирон-бир HTML тегларини қўллаб-қувватламайдиган браузерлар уларга эътибор бермайди, бироқ ўтказилувчи тегларнинг мазмунини HTML тили нуқтаи назаридан таҳлил қиладики, бу саҳифани акс эттиришда хатоларга олиб келиши мумкин. Бунга йўл қўймаслик учун JavaScript тил операторини `<!--...-->` контейнерига қуйида кўрсатилган тарзда жойлаштириш лозим:

```
<SCRIPT [LANGUAGE = "JavaScript"]>
```

```
<!--
```

```
JavaScript операторлари
```

```
//-->
```

```
</SCRIPT>
```

LANGUAGE параметри қўлланилувчи сценарийлар тилини юклайди. JavaScript тили билан боғлиқ ҳолда унинг ифодасини юклаш шарт эмас, чунки бу тилни браузерлар ўз-ўзидан қўллайди. VBScript сценарийлари тили

учун бу параметр ифодасини «VBScript» сатри кўринишида юклаш талаб этилади.

Шарҳни ёпувчи тег--> олдидаги (//) символлари JavaScript шарҳининг операторидир. У интерпретатор тўғри ишлаши учун зарур.

JavaScript кодли файлни юклаш

<SCRIPT> теги тузилган сценарийни JavaScript тилидаги дастурий кодга эга бўлган ташқи файл билан боғлаш имконини берувчи SRC параметрига эгадир. Параметр ифодаси сифатида ресурснинг тўлиқ ёки нисбий URL адреси юкланади. Тег ичидаги операторлар юкланган-юкланмаганлигидан қатъи назар, ёпувчи </SCRIPT> тегининг юкланиши мажбурийдир. Коднинг навбатдаги фрагменти HTML ҳужжатида муайян функциялар тўпламидан иборат манба файл билан боғлайди:

```
<SCRIPT SRC = "http://home.bhv.com/functions/jsfuncs.js">
```

```
[JavaScript операторлари]
```

```
</SCRIPT>
```

Боғловчи ташқи файл HTML тегларига эга бўлмаслиги ва JS кенглигига эга бўлиши лозим.

Агар файл саҳифага киритилганида хато рўй берган бўлмаса, масалан, файл топилмаган бўлса, браузер <SCRIPT> тегининг ичидаги операторларга эътибор бермайди.

Ҳодисаларга ишлов берувчилар

Сценарийларнинг тилларига мувофиқлаштириш учун айрим HTML тегларига юзага келувчи ҳодисаларга ишлов бериш махсус параметрлари киритилган. JavaScript тилининг операторлари мазкур параметрларнинг ифодалари бўлиши мумкин. Одатда ифода тариқасида функция номи юкланади. Ҳодисага ишлов бериш параметри билан белгиланган тегишли ҳодиса рўй берганида функция номи чақирилади. Параметрнинг номи *он* олд қўшимчаси билан бошланади, унинг кетидан ҳодиса номи келади. Масалан,

Click ҳодисасига ишлов бериш параметри («сичқон» тугмаси босилади) onClick деган номга эга бўлади.

Ҳодисалар асосан фойдаланувчи HTML шакли элементлари ёрдамида амалга оширувчи ҳаракатлар билан боғлиқ. Шу боис, ҳодисаларни ушлаш ва уларга ишлов бериш кўпинча шакллар элементларининг параметрларида юкланади. Бу киритилган ахборотни CGI-сценарий билан ишлов беришга юборишдан олдин текшириш имконини беради, [21; 98-110].

Функция ёки процедура—операторларнинг номланган кетма-кетлиги бўлиб, у муайян вазифани бажаради ва муайян ифодани қайтариши мумкин. Функция қуйидаги синтаксисга эга бўлган *function* оператори билан белгиланади:

```
function функция_номи ([параметрлар]) {  
    [JavaScript операторлари]  
    [return ифода]}
```

Функция ифодаловчи параметрлар вергуллар (,) билан ажратилади. Функция танаси (фигурали қавс ичига олинган операторлар блоки)даги *return* номажбурий оператори функция қайтарувчи ифодани белгилайди.

Функцияни эълон қилиш ва уни чақириш ўртасидаги фарқни аниқ тушуниш лозим. *Функцияни эълон қилиш* фақат унинг номини юклайди ва функция чақирилганида у нима қилишини белгилайди. Сценарийда функция *чақирилганида* ва унинг ҳақиқий параметрлари узатилганида функциянинг бевосита бажарилиши амалга ошади.

Зарур функцияларни белгилаш <HEAD> тегида амалга оширилиши лозим, чунки унда белгиланган сценарийнинг барча операторлари саҳифа акс этгунга қадар интерпретация қилинади ва бутун саҳифа акс этиши жараёнида маълум бўлади.

Қуйидаги мисол ҳужжат саҳифасини шакллантириш жараёнида функциянинг юкланиши ва унинг чақирилишини намоиш этади.

```
<HTML>  
<HEAD>
```

```

<TITLE> функцияни юклаш </TITLE>
<SCRIPT LANGUAGE = "JavaScript">
<!--// JavaScriptни қўллаб-қувватламаётган браузерлардан сценарийни
яшириш
function square (number) {
alert ("Функцияларни ҳисоблашим ва кейин ҳужжат тузишим керак!");
return number * number; }
//-->
</SCRIPT>
</HEAD>
<BODY>
<P> функцияни ҳисоблаш сценарийси киритилган саҳифа акс эта
бошлайди </P>
<SCRIPT>
<!--
document.write («Ҳисобланган ифода тенг», square (s), “.”);
//-->
</SCRIPT>
<P> саҳифанинг шаклланиши шу билан тугади.
</BODY>
</HTML>

```

Тег <HEAD>да ўз параметри ифодасининг квадратини қайтарувчи, шунингдек маълумот дарчасини акс эттирувчи square () функциясининг тавсифи берилган. Функцияни чақириш HTML ҳужжати танасида жойлашган сценарийда амалга ошади. Бу сценарийда HTML саҳифасини шакллантиришга мўлжалланган *document* объектининг *write* методи қўлланилади

Ушбу ҳужжат Internet Explorer браузерига юкланганида янги абзац акс этади, сўнгга маълумот дарчаси чиқади.

ОК тугмасининг босилиши маълумот дарчасини беркитади ва саҳифани акс эттиришни давом эттиради.

Java Script тилининг умумий тавсифи ва унинг ўзаги

Юқорида қайд этиб ўтилганидек, JavaScript интерпретация қилинадиган тиллар тоифасига киради, яъни Java виртуал машинаси дастурнинг бошланғич кодини ўқийди ва шу заҳоти уни бажаради. Шу билан JavaScript дастурнинг бошланғич коди компилятор дастур ёрдамида процессорнинг командаларига айлантириладиган компиляция қилинувчи тиллардан фарк қилади. Интерпретация қилинаётган тилдаги дастурларни бажарувчи виртуал машина *интерпретатор*, деб аталади.

JavaScript тилидаги дастур билан ишлашга оид кичик бир мисолни кўриб чиқамиз.

```
function somefunction (x, y, z, t)
{var a, b, c;
a = x * y;
b = z / t;
c = a-b;
return c;}
```

Бу ерда формулалар *return* параметрлари сифатида сонларни қабул қилувчи ва улар билан ҳисоблаш амалларини бажарувчи функцияни тавсифлайди.

Маълумки, исталган тилдаги ҳар қандай дастур муайян *ифода*лардан таркиб топади. Ифодалар бу дастур томонидан бажариладиган муайян амаллар тавсифидир. Ифодалар *операторлар* ва *операндлардан* ташкил топади; операторлар бажариш лозим бўлган амални, операндлар эса—тавсифланган амалларни бажариш талаб этилган маълумотларни юклайди. Ўзгарувчилар ва литераллар операндлар бўлиши мумкин.

Маълумотларнинг типлари

Маълумотларга тўғри ишлов бериш ҳар қандай дастурнинг асосий вазифасидир. Маълумотлар дастурга киритилади, муайян қоидаларга мувофиқ қайта тузилади ва фойдаланувчига қулай кўринишда тақдим этилади. Дастурда маълумотларни қайта тузувчи қоидалар тавсифи *алгоритм*, деб аталади.

Аксарият замонавий дастурлаштириш тиллари сингари, JavaScript тили ҳам муайян типга мансуб бўлган маълумотлар билан ишлайди. Маълумотлар типи бу маълумотларнинг муайян тури: сон, матнли сатр ва ҳ.к. демакдир. У маълумотларнинг мумкин бўлган ифодалари ва уларга нисбатан қўлланиладиган операциялар тўпламини белгилайди. JavaScript тушунадиган маълумотларнинг барча типлари жадвалда келтирилган.

JavaScript тили тушунадиган маълумотларнинг типлари

Маълумотлар типи	Тавсифи
Сатрли маълумот	Матннинг одатдаги сатрлари. Қўш ёки якка қавс (апостроф белгилари) ичига олиниши лозим. Бунда қўш қавс ичига олинган матн якка қавсли бўлиши мумкин ва аксинча.
Бутун сонли маълумот	Бутун сонлар. Ўнли, саккизли ёки ўн олтили сонлар бўлиши мумкин.
Сузувчи нуқтали маълумот	Каср сонлар. Нуқта билан ажратилган бутун ва каср қисмлардан иборат; каср қисми бўлмаган ҳолда ҳам нуқта қўйилиши шарт. Экспоненциал шаклда ёзилиши мумкин: {мантисса}Е {тартиб}. Каср сонлар фақат ўнли бўлиши мумкин.
Мантиқий маълумот	«Ҳа-йўқ» қабилидаги катталиклар. Уларнинг ифодаларига true ва false—«ҳақиқат» ва «ёлғон» таянч сўзлари мос келади.
Null	Null маълумотлар типи маълумот йўқлигини

Шуни қайд этиш керакки, мантикий тип Internet Explorer иловасида 4.0 версиясидан бошлаб, Navigatorда эса–3.0 версиясидан бошлаб қўлланила бошланган.

Ифодалар

Ифодалар–ўзгарувчилар, литераллар ва операторлар комбинацияси бўлиб, уларни ҳисоблаш натижасида ягона бир ифода ҳосил бўлади. Бу ифода сонли (бутун ёки моддий), сатрли ёки булли бўлиши мумкин.

Ифодаларда ўзгарувчилар ё var оператори билан, ё қайд этиш оператори (=) билан инициализация қилиниши лозим. Агар ифодани ҳисоблашда инициализация қилинмаган ёки мутлақо белгиланмаган ўзгарувчи учраса, интерпретатор “undefined variable” («ўзгарувчи белгиланмаган») хатосини қайд этади ва унинг HTML саҳифасидаги ўрнини кўрсатади.

Энди JavaScript тилида ифодалар тузишнинг бир неча қоидаларини келтириб ўтамиз:

–ҳар бир JavaScript ифодаси нукта ва вергул “;” билан тамомланиши керак;

–ифода бир сатрни ёки бир неча сатрни эгаллаши мумкин бўлиб, бу ҳолда ҳам нукта ва вергул ифода тугаганлигининг белгиси ҳисобланади;

–ифода эга бўлиши мумкин бўлган операторлар ва операндлар сони чекланмаган.

Бироқ, шуни қайд этиб ўтиш керакки, ҳаддан ташқари мураккаб ифодаларни нафақат бошқа дастурчи, балки интерпретаторнинг ўзи ҳам тушунишга кўпинча қийналади. Шу боис, мураккаб ифодаларни нисбатан содда ифодаларга ажратган маъқул. JavaScript тилидаги ҳар хил ифодаларни ёзишнинг *асосий принципларини* ва бунда қўлланиладиган операторларни кўриб чиқамиз.

Маълумотларга ишлов бериш ифодалари

JavaScript арифметик ифодалари “=” кайд этиш оператори билан ажратилган икки қисмдан ташкил топади. Унинг чап томонида янги ифода берилаётган ўзгарувчи, ўнг томонида эса–мана шу янги ифода кўрсатилади. У литерал, бошқа ўзгарувчи ёки муайян ифодани ҳисоблаш натижаси бўлиши мумкин.

Бунга бир неча мисолларни кўриб чиқамиз:

$a = 127;$

$b = a;$

$c = a + b;$

Бу уч ифода мутлақо тўғри ифодалардир. Улар математик формулаларга ўхшаб кетади, бироқ нисбатан қатъий нотацияда ёзилган.

Арифметик операторлар

Булар, авваламбор, ҳаммамиз яхши биладиган қўшиш (+), айириш (-), кўпайтириш (*) ва бўлиш (/) операторларидир. Улар иккита операндни қабул қилади ва битта натижа беради. Уларга қўшимча тарзда қолдикни олиш (%) оператори ҳам мавжуд бўлиб, у бир сонни бошқа сонга нобутун бўлади ва бўлишдан қолган қолдикни қайтаради.

Математик инверсия оператори операнд белгисини ўзгартиради. Масалан, мусбат сонни манфийга, манфий сонни эса–мусбатга айлантиради.

Инкремент (++) ва декремент (--) операторлари операнднинг ифодасини биттага кўпайтиради ёки камайтиради. Бунда, агар оператор белгиси операнд олдида турган бўлса, аввал операция–инкремент ёки декремент амалга оширилади, сўнгра операнднинг ўзгартирилган ифодаси қайтарилади. Бордию, оператор белгиси операнддан кейин турган бўлса, аввал операнднинг ўзгартирилмаган ифодаси қайтарилади, сўнгра операция амалга оширилади.

$a = b^{++}; c = -^{-}c;$

Сатрларга ишлов бериш оператори

Сатрларга ишлов бериш оператори атиги битта. Бу сатрларни қўшиш оператори (+)дир. У ўзининг арифметик цехдаги «ҳамкасби»га ўхшайди, лекин у икки сатрни бирлаштиради, яъни иккинчи сатрни биринчи сатр охирига жойлаштиради.

```
title = “Java” + “Script”;
```

Маълумотлар типларининг ўзаро мувофиқлиги

Ниҳоят, маъруза охирида яна бир муҳим масала—маълумотлар типларининг ўзаро мувофиқлиги ва бир типни бошқа типга автоматик тарзда айлантириш масаласини кўриб чиқамиз.

Агар иккита сонли ифода қўшилса, нима бўлади? Яна битта сонли ифода ҳосил бўлади. Агар сон билан сатр қўшилса-чи? Бу ерда JavaScript интерпретатори маълумотлар типларининг номувофиқлиги муаммоси билан тўқнаш келади ва улардан бирини иккинчисига айлантириб, бу типларни ўзаро мувофиқлаштиришга ҳаракат қилади. Бизнинг ҳолатда сон сатрга айлантирилади ва кейин олинган икки сатр биттага бирлаштирилади.

Type of оператори

Type of оператори ўзгарувчи ёки ифодани операнд сифатида қабул қилади ва операнд маълумотлари типини тавсифловчи сатрни қайтаради. Агар операнд сонли (бутун ёки сузувчи нуқтали) маълумотга эга бўлса, “number” сатри қайтади, бордию операнд сатрли маълумотга эга бўлса, “string” сатри, мантикий маълумот учун эса—“boolean” сатри қайтади. Агар ўзгарувчи умуман маълумотга эга бўлмаса, (var оператори томонидан эълон қилинган, лекин ифодаси берилмаган бўлса) ёки ифода (типлар муваффақиятсиз ўзгартирилиши натижасида) ноаниқ маълумот қайтарса, “undefined” сатри қайтади.

typeof оператори қуйидаги форматга эга:

```
typeof ({Ифода})
```

Ушбу операторнинг қўлланилиши билан боғлиқ мисолларни кўриб чиқамиз:

```
if typeof (a + 1) == "number"
```

```
f = typeof (somevar);
```

Операторлар устуворлиги

Мураккаб ифодаalar тузишда дастурчи назарда тутиши лозим бўлган яна бир тушунча бу операторлар устуворлиги, яъни уларни бажариш кетма-кетлиги тушунчасидир. Биз, ўрганган барча операторлар устуворлик даражасига қараб келтирилган.

Операторлар устуворлиги

Операторлар	Тавсифи
++ ---typeof	Инкремент, декремент, арифметик ва мантикий инверсия, тип белгиси
* / %	Кўпайтириш, бўлиш, бўлишдан қолган қолдиқ
+ -	Сатрларни қўшиш ёки бирлаштириш, айириш
<< >> >>>	Иккиталик силжишлар
&	Иккиталик ВА
^	Соқит этувчи иккиталик ЁКИ
/	Иккиталик ЁКИ
= {оператор} =	Оддий ёки мураккаб қайд этишлар

Бир хил устуворликка эга бўлган операторлар чапдан ўнгга қараб бажарилади.

Шартли операторлар, цикллар ва шарҳлар

Шартли операторлар маълумотларга ишлов бериш билан эмас, балки дастур ишини бошқариш билан шуғулланганликлари учун улар алоҳида гуруҳга ажратилган. Шартли оператор муайян шарт берилган (ёки

берилмаган) ҳолда коднинг муайян қисмини бажариши (ёки, аксинча, бажармаслиги) мумкин. Мантикий ўзгарувчи ифодаси ёки мантикий ифодани ҳисоблаш натижаси мана шундай шарт бўлиб хизмат қилади.

Бу ерда мантикий ифода тушунчасига таъриф бериб ўтиш жоиз. Мантикий ифода JavaScript тилининг одатдаги ифодасидир, бироқ уни ҳисоблаш натижаси *true* ёки *false* мантикий катталиклари ҳисобланади. Мантикий ифода қуйида кўриб чиқиладиган мантикий таққослаш операторларига эга.

if-else тармоқлаш оператори

Тармоқлаш оператори муайян ҳодиса содир бўлганида коднинг муайян қисмини бажариш имконини беради. Ўзгарувчига муайян ифоданинг берилиши ёки муайян ифодани ҳисоблашнинг у ёки бу тарзда алоҳида муносабат билдирилиши лозим бўлган натижаси мана шундай ҳодиса бўлиб хизмат қилиши мумкин. Тармоқлаш оператори қуйидаги форматга эга:

```
if ({Шарт}) ... “у ҳолда” блоки  
[else  
... “акс ҳолда” блоки]
```

Шарт–мантикий ифода бўлиб, унга мувофиқ интерпретатор қайси блокни бажариш тўғрисида қарор қабул қилади. Агар шарт *true* («ҳақиқат») ифодасига эга бўлса, “у ҳолда” блоки бажарилади. Агар шарт *false* («ёлғон») ифодасига эга бўлса, “акс ҳолда” блоки бажарилади, (агар у мавжуд бўлса). Бордию, «акс ҳолда» блоки мавжуд бўлмаса, дастурнинг навбатдаги ифодаси бажарилади. Агар шартнинг натижаси *null* ёки *undefined* бўлса, тармоқлаш оператори *false*га қандай муносабат билдирган бўлса, унга ҳам шундай муносабат билдиради.

Мана, бунга бир мисол:

```
if (x == 1){f = 3; h = 4;}  
else { f = 33; h = 44; }
```

Бу ерда биз, *x* ўзгарувчининг ифодасини бир (1)га таққослаймиз ва таққослаш натижаларига қараб *f* ва *h* ўзгарувчиларга ҳар хил ифодалар

берамиз. Шартга эътибор беринг—*мантиқий таққослаш оператори* шундай топилади.

Таққослаш операторлари

Оператор	Тавсифи	Оператор	Тавсифи
<	Кичик	&&	Мантиқий ВА
>	Катта	//	Мантиқий ЁКИ
==	Тенг	!	Мантиқий МАС
<=	Кичик ёки тенг	===	Қатъий тенг
>=	Катта ёки тенг		
!=	Тенг эмас	! ==	Қатъий тенг эмас

Дастлабки олти оператор билан ҳеч қандай муаммо туғилмаслиги керак—улар икки ифодани бир-бирига таққослайди ва шарт бажарилган бўлса, true ни, бажарилган бўлмаса—false ни қайтаради.

Кейинги уч оператор мураккаброқ мантиқий ифодалар тузиш имконини беради. Улар икки мантиқий ифода устида ВА, ЁКИ ва ЭМАС мантиқий операцияларини бажаради ва натижани қайтаради.

Цикллар

Цикллар—бир неча марта бажарилувчи блоклар. Циклнинг такрорланиши муайян шарт берилишига қараб тўхтатилади. Бу муайян шарт берилганида коднинг муайян блоки бажариладиган ёки бажарилмайдиган шартли операторлар билан боғлиқ ҳолатга ўхшайди.

JavaScript дастурчиларга циклларнинг бир неча турларини таклиф қилади.

for цикли

Токи муайян шарт ҳақиқий бўлиб қолар экан, for цикли бажарилади, (true қайтарилади). У коднинг муайян фрагментини маълум марта бажариш учун қўлланилади. Бу ҳолда коднинг мазкур фрагменти неча марта бажарилганлигини ҳисобловчи бутун сон *цикл ҳисоблагичи*, деб аталадиган

ўзгарувчига жойлаштирилади. Коднинг циклда бажарилувчи блоки эса *цикл танаси* ҳисобланади.

```
for ({Инициализация ифодаси}; {Шарт};  
    {Инкремент ифодаси}) ... Цикл танаси
```

Инициализация ифодаси ўзгарувчи-ҳисоблагичнинг бошланғич ифодасини беради. Шундан кейин шарт текшириб кўрилади ва агар унинг ифодаси ҳақиқий бўлса, цикл танаси бажарилади. Шундан сўнг, инкремент ифодаси ҳисоблагичнинг ифодасини ўзгартиради ва яна шарт текшириб кўрилади. Токи, шарт ёлғон (false) тус олмагунича бу амал қайта-қайта бажарилаверади. Ҳисоблагич билан боғлиқ бундай цикллар дастурларда жуда кўп учрайди.

```
for (i = 1; i < 11; i++) { a += 3; b = i * 2 + 1; }
```

Оператор , (вергул)

JavaScript асосан, for цикллари инкрементининг ифодаларида қўлланувчи операторни таклиф қилади. Бу , (вергул) операторидир. У инкрементнинг бир у, бир бу ифодасини кетма-кет бажариш имконини беради. У қуйидаги форматга эга:

```
{“Тоқ ифода”}, {“Жуфт ифода”}
```

Циклнинг биринчи ўтишида «тоқ» ифода, иккинчи ўтишида—«жуфт» ифода, учинчи ўтишида эса—яна «тоқ» ифода бажарилади ва ҳ.к.

```
for (i = 0; j < 11; i++, j++) { a = i * 2 + 1; b = j / 2-3; }
```

, (вергул) оператори устуворлик жиҳатидан барча операторлар орасида энг охирги ўринда туради.

do-while цикли

do-while цикли for циклига кўп жиҳатдан ўхшаш, бироқ унда ҳисоблагич қўлланилмайди. Токи, шарт ҳақиқий бўлиб қолар экан, do-while цикли бажарилаверади. Бунда шарт цикл танаси бажарилишидан олдин эмас, балки у бажарилганидан кейин текширилади. Шу боис, ҳатто шарт аввалбошдан

сохта бўлган тақдирда ҳам, do-while цикли ҳеч бўлмаса бир марта бажарилади.

Do

... Цикл танаси

while ({Шарт});

Сиз do-while циклини ҳар хил усулда қўллашингиз мумкин:

Маълумотлар базаси MySQL

Қуйида қандай қилиб клиент дастури MySQL га уланишни кўрамиз. Бу дастур ёрдамида MySQL серверга уланиш, SQL-сўровларни бажариш ва шу сўровлар натижаларини кўриб чиқиш мумкин. Бу қсимни ўрганиш учун компютерингизда утилитига MySQL ўрнатилгаг ва MySQL сервери билан боғланган бўлиши керак. MySQL серверига MySQL дастури ёрдамида боғланиш учун фойдаланувчи номини ва одатда парол киритиш лозим. Агар сервер ва клиент ҳар хил машиналарда жойлашган бўлса, MySQL сервери ишга туширилган хост номини кўрсатиш лозим:

```
shell> MySQL -h host -u user -p
```

Шундан сўнг экранда қуйидаги сўров пайдо бўлади Ентер пассворд:, ва сизга ўз паролингизни киритишингиз керак бўлади. Агар уланиш тўғри амалгаошган бўлса, экранда қуйидаги маълумот ва команда сатри белгиси пайдо бўлади

```
MySQL>:
```

```
Welcome to the MySQL monitor. Commands end with ; or \g.
```

```
Your MySQL connection id is 459 to server version:
```

```
Type 'help' for help.
```

```
MySQL>
```

Қуйидаги MySQL> белгининг пайдо бўлиши MySQL дастури ишга тайёрлигини билдиради. Сервердан ихтиёрий пайтда QUIT командасини териб узилиш мумкин:

```
MySQL> QUIT
```

Сервердан версияси ва вақтни сўрайдиган содда командани қуйидагича:

```
MySQL> SELECT VERSION(), CURRENT_DATE;
```

MySQL нинг бу сўровга жавоби қуйидаги жадвалдан иборат:

```
+-----+-----+
| version() | current_date |
+-----+-----+
1 row in set (0.02 sec)
```

Бу сўровни бажариш мисолида MySQL билан ишлаш асосий хусусиятларини кўриш мумкин:

- Серверга юборилаётган команда, одатда SQL-ифодадан иборат бўлиб, кетидан нуқта вергул келади. Бу қоидадан чекинишлар бор, масалан QUIT командасидан сўнг нуқтавергул қўйилмайди;

- MySQL сўров натижасини жадвал шаклида чиқаради;

- Сўров натижаларидан иборат жадвални чиқаргандан сўнг, MySQL қайтарилган сатрлар сони ва сўров бажариш вақтини кўрсатади. Бу қулай, чунки сервер унумдорлигини ва сўров бажариш эффективлигини баҳолашга имкон беради;

- Сўров натижалари ва бажарилиш вақтини чиқаргандан сўнг, MySQL янги MySQL> сатрни чиқаради, бу эса янги командалар бажаришга тайёрлигини кўрсатади.

MySQL командалари регистрга боғлиқ эмас, шунинг учун қуйидаги сўровлар бир хилдир:

```
MySQL> select version(), current_date;
```

```
MySQL> SELECT VERSION(), CURRENT_DATE;
```

```
MySQL> Select Version(), Current_DATE;
```

MySQL бир сатрга бир неча командаларни жойлаштиришга имкон беради, лекин улар ҳар бири нуқтавергул билан тугаши керак. Масалан:

```
MySQL> SELECT VERSION(); SELECT NOW();
```

Бундай сўровга қуйидаги натижани оламиз:

```
+-----+
```

```
| version() |
+-----+
| 4. 0. 13-nt |
+-----+
1 row in set (0. 00 sec)
+-----+
| NOW() |
+-----+
| 2004-01-25 16:57:00 |
+-----+
1 row in set (0. 03 sec)
```

Лекин ҳамма командаларни бир сатрга жойлаш шарт эмас:

```
MySQL > SELECT USER(),
-> CURRENT_DATE;
```

Натижа:

```
+-----+-----+
| user()      | current_date |
+-----+-----+
| ODBC@localhost | 2004-01-25   |
+-----+-----+
1 row in set (0. 00 sec)
```

Эътибор беринки биз янги сатрга ўтгандан сўнг, команда сатри белгиси MySQL> дан -> гаўзгарди. Бу билан MySQL тугатилган сўров олинмаганлигини ва сўров охирини кутаётганлигини билдиради. Бу белги жуда фойдали, чунки баъзи хатолар олдини олишга имкон беради. Агар сиз сўров охирида нуқия вергул қўйишни унутган бўлсангиз, MySQL бу тўғрида -> белгини чиқариб билдиради:

```
MySQL> select user()
->
```

MySQL дан сода калкулятор сифатида фойдаланиш учун маслан

қуйидаги сўровни киритиш керак:

МББТ MySQL да SQL тилининг реализацияси ва PHP

бошқарилувчи сўров

MySQL маълумотлар базасида ишлатиладиган маълумотлар типлари Бутун сонлар. Маълумотлар типини кўрсатиш умумий формаси:

prefiksINT [UNSIGNED]

Шарт бўлмаган байроқ UNSIGNED иорасиз сонлар (ўнга тенг ёки катта) сонлар сақлаш учун майдон яратишни билдиради.

TINYINT	Диапазони -128 дан 127 гача сонларни сақлаши мумкин
SMALLINT	Диапазони -32 768 дан 32 767 гача сонларни сақлаши мумкин
MEDIUMINT	Диапазони -8 388 608 дан 8 388 607 гача сонларни сақлаши мумкин
INT	Диапазони -2 147 483 648 дан 2 147 483 647 гача сонларни сақлаши мумкин
BIGINT	Диапазони -9 223 372 036 854 775 808 дан 9 223 372 036 854 775 807 гача сонларни сақлаши мумкин

Каср сонлар

MySQL да бутун сонлар бир неча турга ажратилганидек, каср сонлар ҳам бир неча турга ажратилади. Умумий ҳолда улар қуйидагича ёзилади:

ТипНоми[(length, decimals)] [UNSIGNED]

Бу ерда – каср узатилишда жойлашадиган белги жойлари сони (майдон кенглиги). decimals – ўнли нуқтадан сўнг ҳисобга олинувчи рақамлар сони. UNSIGNED – ишорасиз сонларни беради.

FLOAT	Аниқлиги унча катта бўлмаган сузувчи нуқтали сон.
DOUBLE	Иккилик аниқликка эга бўлган сузувчи нуқтали сон.
REAL	DOUBLE учун синоним.
DECIMAL	Сатрлар шаклида сақланувчи каср сон.

MAL	
NUM ERIC	DECIMAL учун синоним.

Сатрлар

Сатрлар символлар массивларидан иборат. Одатда SELECT сўрови бўйича матнли майдонлар бўйича излашда символлар регистри ҳисобгаолинмайди, Яъни "Vasya" ва "VASYA" сатрлари бир хил ҳисобланади. Агар маълумотлар базаси матни жойлаштириш ва ўқишдаавтоматик қайтакодлашга созланган бўлса, бу майдонлар сиз кўрсатган кодлашда сақланади. Олдинига length дан ошмаган символлар сақловчи сатрлар типлари билан танишамиз., length принадлежит диапазону от 1 дан то 255 гача бўлган диапозонда ётади.

VARCHAR (length) [BINARY]

Бу типдаги майдонга бирор қиймат киритилганда ундан охирини кўрсатувчи символлар автоматик равишда қирқиб олинади. Агар BINARY байроғи кўрсатилган бўлса, SELECT сўровда сатр регистрни ҳисобгаолган холда солиштирилади.

VARCHA R	255 дан ортиқ бўлмаган символларни сақлаши мумкин.
TINYTEX T	255 дан ортиқ бўлмаган символларни сақлаши мумкин.
TEXT	65 535 дан ортиқ бўлмаган символларни сақлаши мумкин.
MEDIUMT EXT	16 777 215 дан ортиқ бўлмаган символларни сақлаши мумкин.
LONGTEX T	4 294 967 295 дан ортиқ бўлмаган символларни сақлаши мумкин.

Кўпинча TEXT тпи қўлланади, лекин маълумотлар 65536 символдан ошмаслигига ишонмасангиз, LONGTEXT типидан фойдаланинг.

Бинар маълумотлар

Бинар маълумотлар – TEXT форматидаги маълумотларнинг ўзи, лекин

уларда излашда символлар регистри ҳисобгаолинади.

TINYBLOB	255 дан ошмаган символларни сақлаши мумкин.
BLOB	65 535 дан ошмаган символларни сақлаши мумкин.
MEDIUMBLOB	16 777 215 дан ошмаган символларни сақлаши мумкин.
LONGBLOB	4 294 967 295 дан ошмаган символларни сақлаши мумкин.

BLOB-маълумотлар автоматик қайта кодланмайди, агар ўрнатилган уланиш билан ишлагндадрхол қайта кодлаш имконияти ўрнатилган бўлса.

Сана ва вақт

MySQL сана ва вақтни ҳар хил форматларда сақлаш учун мўлжалланган майдонлар бир неча типларини қўллайди.

DATE	GGGG-MM-DD форматдаги сана
TIME	CHCH:MM:SS форматдаги вақт
DATETIME	GGGG-MM-DD CHCH:MM:SS форматдаги сана ва вақт
TIMESTAMP	тimestамп форматдаги сана ва вақт. Лекин майдон қийматини олишда у тimestамп форматида эмас, GGGGMMDDCHCHMMSS форматдаакс этади, бу эса PHP дан ундан фойдаланиш қиматини анча камайтиради.

Энди APM маълумотлар базасини MySQL базасида қандай қилиб яратишни ўрганиб чиқамиз. Бизга маълумки маълумотлар базаси ўзининг номига эга булиши билан бирга уни ташкил қилувчилари жадваллар бўлади.

MySQL базаси билан ишловчи командалар

MySQL да маълумотлар базаси CREATE DATABASE командаси ёрдамида яратилади. Команда синтаксиси :

CREATE DATABASE database_name

• database_name – Маълумотлар базасига бериладиган номни

Tatu_elib деб белгилайлик. Демак,

CREATE DATABASE Tatu_elib

PHP да маълумотлар базасини яратиш:

```
$SQL="CREATE DATABASE Tatu_elib"; MySQL_query($SQL);
```

Агар маълумотлар базасини ўчирмоқчи бўлсак DROP DATABASE командасидан фойдаланилади. Синтаксис: DROP DATABASE database_name

PHPда маълумотлар базасини ўчириш:

```
$SQL="DROP DATABASE db_test"; MySQL_query($SQL);
```

USE командаси ёрдамида жадваллар билан ишлаш учун MySQL га қайси база билан ишлаш ҳақида маълумот беради.

§3. Ахборот тизимининг маълумотлар базасини лойиҳалаш ва уни MySQL да яратиш.

Юқоридаги урганганлар буйича биз тизимнинг маълумотлар базасининг жадвалларини ишлаб чиқамиз.

1. Кархонанинг лавозимлари жадвали:

Server: localhost Database: nub Table: labozim

MySQL returned an empty result set (i.e. zero rows). (Query took 0.00)

```
SELECT *
FROM `labozim`
LIMIT 0 , 30
```

	Field	Type	Collation	Attributes	Null	Default
☐	id	int(2)			No	None
☐	name	varchar(30)	utf8_general_ci		No	None

Бунда «id» - лавозимнинг идентификация рақами, «name» - лавозимнинг номланиши.

2. Ходимларнинг руйхатини курсатувчи жадвал:

Field	Type	Collation	Attributes	Null	Default	E
id	int(3)			No	None	
surname	varchar(30)	utf8_general_ci		No	None	
name	varchar(30)	utf8_general_ci		No	None	
lastname	varchar(30)	utf8_general_ci		No	None	
id_labozim	int(2)			No	None	

3. Товарларнинг турларининг жадвали

Field	Type	Collation	Attributes	Null	Default
id	int(3)			No	None
tur_nomi	varchar(40)	utf8_general_ci		No	None

4. Товарларнинг номлари руйхатининг жадвали

Field	Type	Collation	Attributes	Null	Default	Extra
id	int(6)			No	None	
nomi	varchar(100)	utf8_general_ci		No	None	
id_turi	int(3)			No	None	

5. Омборханаларнинг жадвали

Field	Type	Collation	Attributes	Null	Default
id	int(2)			No	None
nomi	varchar(100)	utf8_general_ci		No	None
mudir_id	int(3)			No	None
kladovchik_id	int(3)			No	None

6. Етказиб берувчи кархоналар ва харидорлар руйхати жадвли

Field	Type	Collation	Attributes	Null	Default
id	int(3)			No	None
name	varchar(255)	utf8_general_ci		No	None
rekvezit	varchar(255)	utf8_general_ci		No	None
adres	varchar(255)	utf8_general_ci		No	None
prim	varchar(3)	utf8_general_ci		No	None

7. Товарларнинг сотиб олиниши, кархонга киришини курсатувчи жадвал

Field	Type	Collation	Attributes	Null	Default
id	int(6)			No	None
id_tovar	int(4)			No	None
id_eb	int(3)			No	None
ulcham_birl	varchar(15)	utf8_general_ci		No	None
kirish_boqasi	varchar(15)	utf8_general_ci		No	None
sotish_boqasi	int(6)			No	None
olingan_hajmi	int(6)			No	None
sertifikat_fail	varchar(20)	utf8_general_ci		No	None
kirish_vaqti	date			No	None
yaromlilik_muddati	date			No	None

7. Товарларнинг омборханаларда сақланиш жадвали

Field	Type	Collation	Attributes	Null	Default
id_sklad	int(2)			No	None
id_tobar	int(4)			No	None
kirish_hajmi	int(6)			No	None
qoldiq_hajmi	int(6)			No	None

Биз юқорида тизимнинг асосий жадвалларини курсатиб кеттик.

Ушбу маълумотлар базаси ва PHP тили ёрдамида фойдаланувчилар интерфейсини яратамиз.

прр	Накладной раками	Етказиб берувчилар	Жўнатиш санаси	Қабул қилинган сана	Сумма	Куриш
1	ks-2342	"Янгикола" МЧЖ	2014-04-02	2014-04-04	10.767.000,00	Куриш
2	w24	Шарх ширинликлари ХФ	2014-05-05	2014-05-06	100.000.000,00	Куриш
3	sat_1298	ХК "Транс-узб-К"	2014-06-02	2014-06-06	101.000.000,00	Куриш
4	wq	"Янгикола" МЧЖ	2014-06-18	2014-06-21	100.000,00	Куриш
5	2345	Шакар заводи	2014-06-16	2014-06-22	170.000.000,00	Товар киритиш
6						Сақлаш



Яратган тизимимизда асосий учта бўлим бор бўлиб, «Прайс-лист» бўлими барча фойдаланувчилар учун очиқ ва уннан кархонадаги мавжуд товарлар руйхати, боҳаси в.х. билан танишимиз мумкин.

«Сотиш бўлими» эса ёпиқ бўлим бўлиб, фақат рухсат этилган ходимлар уз парол ва логиини орқали кириш имконига эга. Бу бўлимда «Етказиб берувчилар» ишки бўлими орқали етказиб берувчи кархоналарнинг руйхатини куришимиз ва янги ёзув қушишимиз мумкин. «Товарлар номлари» ишки бўлимида омборлардаги товарлар номларини ва янги товар

номини киритишимизга булади. «Келтирилган товарлар» ишки бўлими юқоридаги расмда курсатилган. Унда олдин келтирилган тоарларнинг накладнойлари курсатилади ва янги накладной кушишга булади. Янги кушилган накладнойга екки кун ичида товарларнинг руйхатларини киритишимиз ва омварларга булиштиришимиз мумкун.

«Омборхона мудир» булимигада парол ва логин билан кирилиб унда омбор мудирлари узларига бекитилган омбордаги товарларнин ҳолати ва харидорларга ёзилган хисоб барахасидаги товарларни йўнотганлиги туғрисида белги туширади. Уннан кейин тизимда товар кулами омбор хисобидан чиқарилади.

Бу яратилган тизимга янги модулар кушиш ва узгартиришлар киритиш, имкон дойрасидаги ҳисобатлар таёрлаш модулларини кушиб боришга булади.

Хулоса

Мамлакатимизда ахборот ва телекоммуникация технологияларини равожлантириш буйича белгиланган мақсад ва вазифалар, ишланаятган ишлар, буларнинг ҳаммаси мамлакатимиздаги соҳаларнинг яқин йилларнинг ишида замон билан ҳамнафас янги ахборот ва телекоммуникация технологияларга утишининг гаровидир.

Битирув малакавий ишда улгуржи савдо кархоналари фаолиятини автоматлаштиришда «Нукус универсал базаси» шўъба корхонаси мисолида ахборот тизимини яратиш масалалари урганилиб куйидаги хулосалар олинди.

- «Нукус универсал базаси» шўъба корхонаси бизнес жараёни урганилиб таҳлил этлган ва функционал модели ишлаб чиқилди.
- Ахборот тизимида маълумотлар ва фойдаланувчилар ҳисобга олиниб унинг лойҳаси яратилди.
- Ахборот тизимини яратишда Веб дастурлашда қулай булган MySQL маълумотлар базаси ва PHP скриптили тили танланиб улар урганилиб чиқилди.
- Буларнинг асосида тизимнинг ахборотлар манбайлари учун маълумотлар базаси яратилиб фойдаланувчилар учун интерфейс ишланди.

Нукус универсал базаси шўъба корхонаси фаолиятини автоматлаштириш учун яратилган ўрта ахборот тизими янада ривожлантирилиб масофадан харид қилиш модулларини ва бошқада имконларини қушишга қаратилиб ишланганини ҳисобга олсак уни бошқада ушбу фаолияттаги бошқа кархоналаргада фойдаланишга булади.

Фойдаланилган адабиётлар

1. И. Шапошников
Web-сайт своими руками -: БХВ – Санкт-Петербург, 2000
2. Орлов А.А.
PHP. Полезные приёмы. Издательство: Горячая -Телеком, 2004
3. Кузнецов М.В., Симдянов И.В., Голышев С.В.
PHP5. Практика создания web-сайтов. Изда: БХВ-Петербург, 2005
4. Кухарчик А..
PHP. Обучение на примерах. Издательство: Новое знание, 2004. PDF
5. Луис Аргерих, Ванкиу Чой, Джон Коггсхол.
Профессиональное PHP программирование. Издательство: Символ-Плюс, 2003.DJVU
6. Sh.A. Nazirov, R.V. Qobulov.
MySQL va PHP (qo'llanma). Uzbekistan 2008)
7. <http://www.php.ru/>
8. <http://www.intuit.ru/>
9. <http://www.allday2.com/>
10. <http://www.mysql.ru/docs/man/Connecting-disconnecting.html>