

**O'ZBEKISTON RESPUBLIKASI OLIY VA O'RTA TA'LIM
VAZIRLIGI**

BUXORO DAVLAT UNIVERSITETI

Fizika-matematika fakulteti

“Axborot texnologiyalari” kafedrası

Hayitova Durdona Alijonovnaning

**“Universitet informatsion tizimida fan dasturlardagi o'quv adabiyotlar
monitoringi modulini yaratish”**

**“Amaliy matematika va informatika” ta'lim yo'nalishi bo'yicha bakalavr
darajasini olish uchun**

BITIRUV MALAKAVIY ISHI

**“Ish ko'rildi va himoyaga ruxsat
berildi”**

**Ilmiy rahbar ____ Shirinov Z.Z
«__» _____ 2016 y.**

**Kafedra mudiri
____ dots.T.B.Boltayev**

**Taqrizchi____
«__» _____ 2016 y.**

«__» _____ 2016 y.

«Himoya qilishga ruxsat berildi»

Fakutet dekani prof.Sh.M.Mirzayev

«__» _____ 2016 y.

Buxoro-2016

Mundarija:

Kirish.....

I.BOB. Web sayt tayyorlashda ishlatiladigan zaruriy dasturiy vositalar.

1.1. HTML tili, strukturasi, tag(teg)lari.....

1.2.ASP texnologiyasi.....

1.3. SQL tili, SQL server berilgan bazasini boshqarish tizimlari.....

II.BOB. Universitet informatsion tizimida fan dasturlardagi o'quv adabiyotlar monitoringi modulini yaratish.

2.1. O'quv adabiyotlar monitoringi modulini yaratish spetsifikatsiyasi.....

2.2. Dastur loyihasi.....

2.2.1. ER modeli.....

2.2.2. Foydalanuvchi grafik interfeysi.....

2.2.3. Asosiy modullari.....

2.3. O'quv adabiyotlar monitoringi modulini yaratishni amalga oshirish.....

Xotima.....

Foydalanilgan adabiyotlar.....

KIRISH

Taraqqiyotning tez rivojlanayotgan hozirgi davridako'pgina sohalarni kompyuterlarsiz tasavvur qilib bo'lmaydi. Kompyuterlar xalq xo'jaligining barcha sohalariga kirib keldi va keng qo'llanilmoqda. Oldingi yaratilgan hisoblash mashinalari faqat hisoblash ishlarini bajargan bo'lsa, hozirgi kompyuterlarning imkoniyatlari juda katta va bir necha yuzlab amallarni bajaradi. Hozirda kompyuterlarni qo'llashda ko'pgina foydalanuvchilar uchun yagona axborot makonini ta'riflovchi tarmoqlarni tashkil etish muhim ahamiyatga ega. Buni butun dunyo kompyuter tarmog'i hisoblanmish internet misolida yaqqol ko'rish mumkin.

Insonning barcha sohalardagi faoliyati axborot bilan to'liq ta'minlanganligi va axborotlardan samarali foydalanishiga bog'liq bo'lmoqda. Katta hajmdagi axborotlar oqimidan kerakli bo'lganini tanlab olish, ularni qayta ishlash va foydalanish uchun har bir mutaxassis shaxsiy kompyuter va aloqa vositalaridan unumli foydalanishi kerakdir.

Foydalanuvchilar kundan-kun kengayib boradigan, ko'p marta foydalaniladigan dasturlar olamiga oson kirish imkoniyatiga ega bo'lishni istaydilar. Hozirgi kunda aynan Java tili dasturlovchilar jamiyatiga shunday afzalliklarni taklif eta oladi.

Dunyoda kechayotgan globallashuv jarayonlari davlat va jamiyat qurilishi tizimiga, davlat va jamiyatning turli sohalariga zamonaviy axborot va kommunikatsiya texnologiyalarini keng joriy etishni talab etmoqda. Bu o'z navbatida soha faoliyatining ochiqligi va oshkoraligini ta'minlash, axborotlarni yetkazish va interaktiv xizmatlarni taqdim etishni yo'lga qo'yish, milliy ma'naviyatni yuksaltirish, aholini foydali axborot bilan ta'minlash, hamda yuqori iqtisodiy natijalarga erishish bilan bir qatorda, mahsulot va xizmatlarni jahon bozoriga olib chiqish uchun keng imkoniyat yaratib bermoqda.

O'zbekiston Respublikasi Prezidentining «O'zbekiston Respublikasi Milliy axborot-kommunikatsiya tizimini yanada rivojlantirish chora-tadbirlari to'g'risida» 2013 yil 27 iyundagi PQ-1989-sonli qaroriga muvofiq, unda davlat hokimiyati va

boshqaruvi organlari faoliyatiga zamonaviy axborot kommunikatsiya texnologiyalarini keng joriy etish, ular tomonidan interaktiv davlat xizmatlari ko'rsatish, elektron tijorat va elektron to'lovlar tizimini yanada rivojlantirish, tijorat faoliyati va tadbirkorlikda axborot texnologiyalaridan samarali foydalanishni rivojlantirish, Internet tizimi orqali kommunal va boshqa to'lovlarni yo'lga qo'yish, mahalliy tovarlar va xizmatlarni tashqi bozorda realizatsiya qilish maqsadida elektron tijorat infratuzilmasini yaratish borasida ustuvor vazifalar belgilab berilgan. Bundan tashqari O'zbekistonda o'qitish texnologiyalarini zamonaviylashtirish, jadallashtirish rivojlangan iqtisodiyotli mamlakatlarga qaraganda yanada dolzarb ahamiyatga ega. Chunki hozirgi kunda milliy ta'lim tizimining salohiyatli tizimli rivojlanishi yanada yuqori pog'onaga ko'tarildi.

Mavzuning dolzarbligi: Yaratiladigan o'quv adabiyotlar monitoringi kutubxonaning zamonaviy shakllaridan biri hisoblanadi. Ta'limda adabiyotlarning ahamiyati beqiyos hisoblanadi. O'quv adabiyotlar monitoringi adabiyotlardan unumli foydalanishga undaydi.

Bitiruv malakaviy ishimizning maqsadi: PHP web dasturlash texnologiyasi imkoniyatlaridan foydalangan holda fan dasturlardagi o'quv adabiyotlar monitoringi modulini yaratish.

Bitiruv malakaviy ishimizning vazifalari: Malakaviy ish maqsadidan kelib chiqqan holda quyidagi vazifalar belgilandi. PHP web dasturlash texnologiyasini imkoniyatlarini o'rganish, fan dasturlardagi o'quv adabiyotlar monitoringi modulini yaratishda PHP texnologiyasidan foydalanib uning imkoniyatlarini ko'rsatish.

Bitiruv malakaviy ishimizning o'rganilganlik darajasi: Bitiruv malakaviy ishimni qilish jarayonida juda ko'p ma'lumotlarni o'rganib oldim. Ya'ni dasturni tuzish jarayonida PHP texnologiyasini, SQL tili, web dasturlash tillarini va Universitetda mavjud moodle tizimi haqida oz bo'lsa ham bilimga ega bo'ldim.

Bitiruv malakaviy ishimizning predmeti: HTML, PHP texnologiyasi, SQL tili.

**Bitiruv malakaviy ishimizning ob`ekti:**Fan dasturlardagi o`quv adabiyotlar.

Bitiruv malakaviy ishimizning ilmiy farazi:Fan dasturlardagi o`quv adabiyotlar monitoringi modulini yaratishda PHP texnologiyasi va SQL tillarini qo`llash.

Bitiruv malakaviy ishimizning yangiligi:Fan dasturlardagi o`quv adabiyotlar monitoringi modulini Universitet informatsion tizimida qo`llash.

Bitiruv malakaviy ishimizning amaliy ahamiyati: Talabalarning adabiyot izlash jarayonini osonlashtirish, vaqtdan unumli foydalanish, adabiyotlarni tezkor topa olish va ulardan istalgan vaqtda foydalana olish va qulaylikni oshirish.

Bitiruv malakaviy ishimizning metodologik asoslari:WEB arxitektura va SQL tili.

Bitiruv malakaviy ishimizning metodlari:WEB serverni dasturlashda PHP texnologiyasidan va ma'lumotlar bazasi ma'lumotlarini qayta ishlashda SQL tilidan foydalanildi.

Bitiruv malakaviy ishimizning tarkibi va hajmi:Ushbu bitiruv malakaviy ishi ___betdan iborat bo`lib, kirish, 2 ta bob, xotima va foydalanilgan adabiyotlar ro`yxatidan iborat.

I. Web sayt tayyorlashda ishlatiladigan zaruriy dasturiy vositalar.

1.1 HTML tili, strukturasi, tag(teg)lari.

HTML tili buyruqlari teg (tag) deb ataluvchi maxsus elementlar yordamida beriladi, ya'ni uning asosini teglar tashkil etadi. Teglar <> qavs orasida berilib, ular brouzerda ko'rinmaydi, balki qulay ko'rinishga keltirish uchun xizmat qiladi. Odatda, ko'pchilik teglar ikki marta takrorlanib, jufti bilan beriladi, ya'ni "ochilib-yopiladi". Masalan, < BODY>, < / BODY>.

HTML tili tanasida ajratib ko'rsatish uchun teglar katta harflar bilan yoziladi va < HTML > bilan boshlanib, </ HTML > bilan tugaydi. Bunda katta va kichik harflar farqlanmaydi. Shuningdek, HTML tili andozasi bo'yicha hujjatga <HEAD> va <BODY> teglarini kiritish tavsiya etiladi. Brauzer HTML hujjatni o'qiganida, ularning borligi hujjat bo'limlarini aniq ko'rsatadi. Biroq ular bo'lmasa ham, brauzer HTML — hujjatni to'g'ri o'qiydi, lekin hujjat bo'limlari bir-biridan ajralib turmaydi.

Shunday qilib, to'g'ri tuzilgan HTML — hujjat quyidagi tuzilishga ega:

< HTML>

< HEAD>

Sarlavhaga oid ma'lumot

< / HEAD>

< BODY>

Hujjatning mazmuni

< / BODY>

</ HTML>

Bunda < HEAD>, </ HEAD> orasida joylashgan sarlavhaga oid ma'lumot qismida, odatda, foydalanuvchiga e'tiborli bo'lmagan, lekin brouzer uchun lozim ma'lumot beriladi.

< BODY>, </ BODY> orasiga esa uning operatorlari to'laligicha ketma-ketlikda joylashtiriladi.

HTML tili asosiy operatorlari va ularning tasnifi:

HTML tili operatorlardan tashkil topadi. Ularning asosiylarini ko'rib chiqamiz.

— izoh. Shu belgi orasiga joylashtirilgan ixtiyoriy matn izoh deb qaraladi.

<A>... — hujjatga gipermurojaat o'rnatish.

Atributlari (qo'shimcha yordamchi operatorlari, bular <A>... orasida ishlatiladi): HREF, NAME, TARGET, _blank, _top, _parent, _self, TITLE, TYPE, CHARSET, HREFLANG.

-**HREF** atributi — gipermurojaatga olib boruvchi URL ta'riflovchisi vazifasini bajaradi. Masalan:

O'zMU . Agar yo'nalish ichma-ich joylashgan papkalarda joylashgan bo'lsa, “/” belgisi yordamida ko'rsatiladi. Agar murojaatni shu sahifadagi biror belgiga o'rnatish lozim bo'lsa, “#” belgisidan keyin ko'rsatiladi.

- **NAME** — agar gipermurojaat shu sahifaning o'ziga o'rnatilgan bo'lsa, o'tishni ushbu atribut yordamida ko'rsatish mumkin.

-**TARGET** — gipermurojaat qayerda ochilishi kerakligini ko'rsatadi:

blank — gipermurojaat natijasini yangi darchaga ochib beradi.

top — natijani brouzerdagi barcha darchalarga ochish (maxsus FRAME lar uchun).

parent — natijani brouzerdagi bosh sahifa darchasiga ochish (maxsus FRAME lar uchun).

self — natijani joriy darchada ochish (maxsus FRAME lar uchun).

-**TITLE** — bu hujjat darchasining sarlavhasini ifodalash uchun.

-**TYPE** — gipermurojaat o'rnatilgan hujjatning MIME — turini aniqlaydi.

-**CHARSET** — gipermurojaat o'rnatilgan hujjatning yozuv kodini aniqlaydi.

-**HREFLANG** — gipermurojaat o'rnatilgan hujjatning yozuv tilini aniqlaydi.

... — matnni to'q-qalin shrift bilan tasvirlaydi.

<BASEFONT>...</BASEFONT> — hujjatda avvaldan qabul qilingan shriftning o'lchami, turi va rangini ko'rsatish uchun ishlatiladi. Odatda, matnda shrift o'lchamini bir xilda, ta'kidlashsiz ko'rsatish uchun ishlatiladi. Shrift o'lchami 1 dan (eng kichik) 7 gacha (eng katta). Ta'kidlanmasa, avtomatik ravishda 3 o'lchovida yoziladi.

BGSOUND - Sahifa foniga ovozni ulash. *.mid, *.wav, *.au turdagi ovozli fayllarni qabul qiladi.

-**SCR** — orqali ovozli fayl joylashgan joy ko'rsatiladi.

- **LOOP** — fonli ovozning takrorlanish sonini ko'rsatish, “-1” qiymati yoki “INFINITE” ni tanlash orqali cheksiz qilib qo'yish mumkin.

Masalan: <big> Katta o'lchamli matn </big>.

<BODY>...</BODY> — veb-sahifani to'ldiruvchi matn, deskriptorlar va boshqa ma'lumotlarni aniqlaydi, ya'ni asosiy ma'lumotlar ushbu teglar orasiga kiritiladi. Atributlari:

-**BACKGROUND** — fon tasviri yoki rasmning manzilini ko'rsatish.

- **BGCOLOR** — fon rangini ko'rsatish, bunda rang o'mida tasvir yoki rasmlardan ham foydalanish mumkin.

-**TEXT** — matnning asosiy rangi.

-**LINK** — gipermurojaatli matn rangi.

Masalan: <body background = «rasml.gif» bgcolor = «red» text = «black» link = «blue»... >

-**VLINK** — gipermurojaatli matn tanlangandan keyingi rangi.

Masalan: <body background= «rasml.gif» bgcolor= «red» text=«black» link= «blue» vlink= «olive»... >

- **ALINK** — joriy gipermurojaat rangi, kursor yordamida tanlangan paytda shu rang ko'rinadi.

BR - Satrni avtomatik bo'lib, keyingi satrga o'tkazish.

DD - Matn chap chegara bo'yicha tekislanadi. Ta'riflar ro'yxati (DL) ichida ishlatiladi.

<DL>...</DL> — ta'riflar ro'yxatini ko'rsatadi. Ichida <DT> teg orqali aniqlanayotgan termin, <DD> teg bilan esa termin ta'rif beriladi.

EMBED - Sahifadagi bajariluvchi obyektlar uchun zarur bo'lgan vositalar joylashgan joyni ko'rsatish yoki avtomatik bajariladigan qilish. Masalan:

Flash, VRML, QuickTime, Adobe Acrobat va boshqa vositalar orqali.

SCR — orqali obyekt (fayl) ning joylashgan joyi yo'lini ko'rsatiladi.

... — shrift parametrlarini ko'rsatadi, uning atributlari:

-**SIZE** — matn o'lchamini ko'rsatish;

-**COLOR** — matn rangini ko'rsatish;

- **FACE** — shrift nomini ko'rsatish, shuningdek bir nechta shriftni ham o'rnatish mumkin.

Masalan:
qizil rang, o'lchami uch

H1 - Matn qismi yoki holati.

<H1>...</H1> — birinchi pog'ona sarlavhalari (eng kattasi).

Atributlari:

-**ALIGN** — Tekislashni aniqlash.

Masalan: <h1 align= «center» ... > ... </h1>

-**FONT** — berilgan matn qismi shriftini ko'rsatish.

Masalan: <h1 align = «center» font = «Verdana» ... > ... </h1>

-**SIZE** — shrift o'lchamini ko'rsatish.

Masalan: <h1 align = «center» font = «Verdana» size = «3» > ... </h1>

<H2>...</H2> — ikkinchi pog'ona sarlavha. Umuman olti xil

sarlavha mavjud. Ularning qolgan to'rttasi <H3>,<H4>,<H5>,<H6> (eng kichigi) bilan belgilanadi.

<HR> — gorizontaal chiziq (chizg'ich) qo'yadi.

<I>...</I> — matnni yozma shrift bilan tasvirlaydi.

 — sahifada rasm joylashtirishda ishlatiladi.

Masalan: , bu yerda photol — sizning veb-sahifangizdagi fayl bilan bitta katalogda turgan rasm nomi.

Atributlari:

-**SCR** — rasm joylashgan joyni ko'rsatish;

- **ALT**— rasm ochilguncha yoki ochilmay qolganda tegishli matnli satr kiritish;

-**ALIGN** — tasvirda matnning holatini ko'rsatish:

top — yuqori chegara bo'yicha; bottom — pastki chegara bo'yicha; left — chap chegara bo'yicha; right — o'ng chegara bo'yicha; center — markaz bo'yicha.

-**WIDTH** — rasm kengligi;

-**HEIGHT** — rasm balandligi;

-**HSPACE** — gorizontaal bo'yicha tasvirgacha bo'lgan bo'sh maydonni aniqlash;

- **VSPACE** — vertikal bo'yicha tasvirgacha bo'lgan bo'sh maydonni aniqlash;

-**BORDER** — tasvir atrofidagi chiziq o'lchamini ko'rsatish;

-**NAME** — tasvir nomini aniqlaydi.

Masalan:

...() — ro'yxatdagi har bir element boshlanishini aniqlaydi (odatda, ... yoki ... ro'yxat teglari orasida ishlatiladi).

-**VALUE** — ro'yxat yozilish tartibini (nomer boshini) aniqlaydi.

-**TYPE** — ro'yxatdagi nomerlash turini ko'rsatadi.

... — to'liq tartiblangan ro'yxatni aniqlaydi.

Atributlari:

-**TYPE** — tartiblangan ro'yxat turini ko'rsatish;

-**1** — arab raqamlari yordamida tartiblash;

-**A** — katta harflarda tartiblash;

-**a** — kichik harflarda tartiblash;

-**I** — rim raqamlarida tartiblash;

-**i** — kichik rim raqamlarida tartiblash;

- **START=n** — nechadan boshlanishi;

-**COMPACT** — ro'yxatni ixcham ko'rinishda tasvirlash.

<P>...</P> — abzats (xat boshi)ni aniqlash: ALIGN — tekislashni aniqlash; left — chap chegara bo'yicha; right — o'ng chegara bo'yicha; center — markaz bo'yicha; justify — kengligi bo'yicha (eniga).

<PRE>...</PRE> — oldindan formatlangan matnni aniqlaydi, matn holatini saqlaydi.

-**SCRIPT** - Java Script kodini aniqlaydi.

-**SUB** - Matnni quyi indeks kabi ifodalash.

-**SUP** - Matnni yuqori indeks kabi ifodalash.

-**TABLE** - Jadval yaratishda ishlatiladi.<TABLE> ... </TABLE>

<TABLE ... > ichida atributlari yoziladi, ular:

-**BORDER** — jadval chegarasi chiziq kengligini ko'rsatish.

- **ALIGN** — gorizontal bo'yicha tekislash, o left — chap chegara bo'yicha tekislash; o right — o'ng chegara bo'yicha tekislash; o center - markaz bo'yicha tekislash.

Masalan: <table border = «2» align = «center»... > ... </table>

-**WIDTH, HEIGHT** — kengligi va balandligi, piksel yoki foizlarda.

Masalan: <table border = «2» align = «center» width = «300» height = «100%»... > ... </table>

- **CELLSPACING** — Qo'shni yacheykalar orasining kengligini ko'rsatish uchun ishlatiladi, piksel yoki foizlarda beriladi.

-**CELLPADDING** — yacheykalardagi ma'lumot va chiziq chegarasi orasining kengligini ko'rsatish, piksel yoki foizlarda beriladi.

-**FRAME** — jadvalda qaysi chegaralarni ko'rsatish va qaysilarini ko'rinmaydigan qilish.

-**BGCOLOR,BORDERCOLOR**— jadval foni vajadval chegarasi rangini ko'rsatish.

-**BACKGROUND** — jadval foni sifatida tasvirdan foydalanish.

<TD>...</TD> — jadval satrida alohida yacheykani ramkaga oladi.

<TH>...</TH> — jadval sarlavha yacheykasi uchun ishlatiladi.

<TITLE>...</TITLE> — sarlavhani tashkil etadi. Ushbu teglar

orasida browserda hujjat nomini ko'rsatish mumkin.

<TR>...</TR> — jadvalda satrning boshi va oxiri. Jadval satrini aniqlaydi. Odatda, jadvalda <TH> yoki <TD> teglari bilan ishlatiladi.

<U>...</U> — matnni ostki qismi chizilgan holda tasvirlaydi.

... — Markerli ro'yxat yaratish. yordamida beriladi.

Atributlari:

-**TYPE** — marker turini ko'rsatish;

-**disc** — bo'yalgan doira;

-**circle** — bo'yalmagan doira;

-**square** — kvadrat;

-**COMPACT** — matn ko'rinishini ixchamlashtirish.

Forma bu bir nechta elementlarni o'z ichiga oluvchi umumiy soha hisoblanadi. Web saytlarda formalar juda ko'p joyda ishlatiladi. Misol uchun, **registratsiyada, so'rovlarda, avtorizatsiyada, kommentariyalarda** va hokazo. Demak, zamonaviy saytlarni formalarsiz tasavvur qilib bo'lmas ekan. Html orqali faqat formani tashqi ko'rinishini qilish mumkin, undagi ma'lumotlarni qayta ishlash veb dasturlash(**php, asp,...**) orqali amalga oshiriladi.

1.2. ASP texnologiyasi.

ASP (Active Server Pages). Bu WWW sahifalarni tezkor va oson ishlab chiqish uchun Microsoft firmasi tomonidan ishlab chiqilgan kuchli texnologiyadir. ASP Windows NT va IIS (Internet Information Server), platformalarida ishlaydi.

ASP bu shunday ichki texnologiyaki, uning yordamida Web sahifalarga dasturlarni bog'lash mumkin. ASP ning yutug'i skriptli tillar (Visual Basic Script yoki Java Script) va tashqi COM komponentlarini qo'llay olishi hisoblanadi.

VBScriptda strukturalangan dasturlash uchun barcha konstruktsiyalar(if, while, case, etc) mavjud. Undan tashqari bu tilda o'zgaruvchilar (qaysi tipga tegishli ekanligini ko'rsatmasa ham bo'ladi) bor. Ob'ektlarni qo'llab quvvatlaydi. Ular bilan ishlash oddiygina Object.Property, Object.Method. Yana bir nechta

qo'shimcha ob'ektlar ham mavjud (Request, Response, Session, Server, Connection, Recordset). Undan tashqari dasturchining o'zi ham kerakli komponentlarni o'rnatishi mumkin masalan elektron pochta bilan ishlayotganda "Ko'chirib olish, sotib olish... ". [5]

ASP texnologiyasida chiqarish buyrug'i.

Foydalanuvchiga ko'rsatilishi kerak bo'lgan ma'lumotlar HTML tili yordamida chiqariluvchi potokka tashlanadi. Foydalanuvchi brauzeri bu HTML potokni interpretatsiyalaydi. Ma'lumot chiqarishni yengillashtirish uchun Response. Obyekti mavjud. Chiqarishni esa Write metodi yordamida amalga oshiriladi. Masalan:

```
Response.Write("<h2> Salom dunyo </h2>").
```

Bu yo'l bilan Response obyektining ichki buferiga yozish amali bajariladi. Skript ishini yakunlagandan so'ng, bufer to'laligicha klientga beriladi. Shuni aytib o'tish kerakki, mijoz "top toza" HTML ni oladi, va asosiysi buning natijasida ASP dasturlari mijozning dasturiy ta'minotiga umuman bog'liq bo'lmaydi. Response obyektining boshqa metod va xususiyatlari chiqarishni boshqarishni ta'minlab beradi. Masalan Response.Buffer metodi mijoz ma'lumotlarni Response obyekti yordamida olyaptimi yoki sahifa yakuniga ko'ra hammasini o'qish o'qimasligini boshqarib turadi. Response.Redirect metodi mijoz brauzerini boshqa sahifaga yo'llaydi. Bu metoddan foydalanilganda, ungacha sahifada Response.Write metodi qo'llamaslik kerak.

ASP texnologiyasida kiritish buyrug'i

ASP da tuzilgan dastur foydalanuvchidan to'g'ridan to'g'ri biror nima haqida so'ray olmaydi. U ma'lumotlarni boshqa sahifalardan yoki URL manzilidan olishi mumkin. Beriladigan ma'lumotlar chiqariluvchi potokka joylashtiriladi va Request obyekti yordamida qabul qilinadi. Var o'zgaruvchisining qiymatini test.asp dasturiga berish uchun quyidagi jumlanı yozish kerak bo'ladi:

```
test.asp?var=abc
```

Dasturdan o`zgaruvchining qiymatini olish uchun quyidagi kodni yozishimiz kerak:

```
var = Request("var")
```

Agar o`zgaruvchilar bittadan ko`p bo`lsa, ular & belgisi yordamida ajratiladi:

```
test.asp?var1=abc&var2=def
```

ASP.NET MVC da fayllarni uzatish.

File Result sinfi mijozga fayllarni uzatadi. Ammo bu sinf abstraktli, aslida bizlar merosxo`rlar bilan ishlaymiz:

-FileContentResult: fayldan o`qilgan, baytlar massivini mijozga uzatadi.

-File Path Result: faylni to`g`ridan to`g`ri serverdan uzatadi.

-File StreamResult: bu sinf oqimni yaratadi – System.IO.Stream ob`yekti orqali faylni o`qib chiqadi va faylni mijozga uzatadi.

Ko`rsatilgan uchta usulda File metodi ishlatiladi, qaysisi File Result ob`yektini qaytaradi. Faqat tanlangan usulga qarab, bu metodning tegishli qayta yuklangan versiyasi ishlatiladi.

Kim phpda ishlagan bo`lsa agar, ular s – server ob`yektini va o`zgaruvchan qatorini biladilar. Bular orqali so`rov haqida ma`lumot olish mumkin: brauzer haqida, IP-manzili haqida va izoh httpContext ob`yektida ASP.NET.MVC bunday harakatlarni bajarish imkonini beradi.

1.3. SQL tili, SQL server berilganlar bazasini boshqarish tizimlari

Ma`lumotlar bazasi dunyosi tobora yagona bolib bormoqda. Bu jarayon har xil kompyuter muhitlarida faoliyat ko`rsatuvchi axborot tizimlarini hosil qilishda qo'llanuvchi yagona standart til yaratishni talab qildi.

Buyruqlar to'plamini bilgan foydalanuvchilarga standart til, ular shaxsiy kompyuter tarmoq ishchi stansiyasida yoki katta EHM da ishlashidan qat'iy nazar, ma`lumot yaratish, izlash va uzatishga imkon beradi.

SQL (Structured Query Language, odatda “sikvel” deyiladi), ma`nosi — Tarkiblangan so'rovlar tili. Bu relyatsion ma`lumotlar bazalarida ishlashga imkon

beradigan tildir. Bu til ifodalarining xususiyati shundan iboratki, ular ma'lumotlarni qayta ishlash protseduralariga emas, natijalariga yo'naltirilgandir.

SQL o'zi ma'lumotlar qayerda joylashgani, indekslar qandayligini va, hatto, amallarni qanday izchillikda qo'llasa bo'lishini aniqlaydi, bu tafsilotlarni ma'lumotlar bazasiga so'rovlarda ko'rsatish kerak emas. SQL tili IBM kompaniyasida MBBT DB2 yaratish jarayonida ishlab chiqilgan va keng ko'lamda RISC protsessorli mashinalarda UNIX tizimlar asosida hamda meynfreymlarda, super kompyuterlar asosida qurilgan katta hisoblash tizimlarida qo'llanilgan. Shu bilan birga, u mustaqil bo'lmasdan, PL/SQL va Transact-SQL kabi ichki dasturlash tillariga inkapsulyatsiya qilinadi. 1986-yilda ANSI (American National Standart Institute) SQL tilining rasmiy standartini ishlab chiqdi, 1992- yilda bu standart kengaytirildi. Butun til 30 ga yaqin operatorlarga ega bo'lib, ba'zi versiyalarida sal ko'proq, ba'zilarida sal kamroqdir. Har qanday MB har xil ob'ektlarga, ya'ni jadvallar, protseduralar, funksiyalar, tasavvurlar, ketma-ketliklar va hokazolarga ega.

“Kliyent-Server” texnologiyasiga ko'ra, foydalanuvchi EHM (Kliyent) lar so'rovlari maxsus ma'lumotlar serverlarida qayta ishlanadi, foydalanuvchi EHM larga faqat so'rovni qayta ishlash natijalari qaytariladi. Tabiiyki, Server bilan muloqot qilish uchun yagona til kerak va bunday til sifatida SQL tanlangan.

Shuning uchun hamma zamonaviy relyatsion MBBT versiyalarida (DB2, Oracle, Ingres, Informix, Sybase, Progress, Rdb) va, hattoki, norelyatsion MBBT versiyalarida (masalan, Adabas) “Klient-Server” texnologiyasi va SQL tilidan foydalaniladi. SQL tilida ma'lumotlarni jadval ko'rinishda tasvirlashga yo'naltirilgan amallar konsepsiyasi ko'p bo'lmagan (30 dan kam) ifodalardan iborat kompakt til yaratishga imkon berdi. Ikki xil: interaktiv va joylashtirilgan SQL mavjud. Ko'p hollarda ikkala forma bir xil ishlaydi, lekin ikki xil foydalaniladi. Interaktiv SQL ma'lumotlar bazasining o'zida faoliyat ko'rsatadi va buyurtmachi foydalanishi uchun chiqish hosil qilishda ishlatiladi. SQLning bu formasida buyruq kiritilsa, u darrov bajariladi va darhol natijani (agar u mavjud bo'lsa) ko'rish

mumkin. Joylashtirilgan SQL boshqa tilda yaratilgan dasturga joylashtirilgan SQL buyruqlardan iborat.

SQLning interaktiv va joylashtirilgan formalarida ko'p sonli guruhlar yoki subbolimlar mavjud. Ular ANSI tomonidan e'tiborga olingan va konseptual darajada foydali, lekin ko'pchilik SQL dasturlar ularni alohida qayta ishlamaydi, shuning uchun ular aslida SQL buyruqlarining funksional kategoriyalaridir.

- DDL (Ma'lumotlarni Ta'riflash Tili) — ANSI da sxemani ta'riflash tili, obyektlarni (jadvallar, indekslar, tasavvurlar va hokazo) yaratuvchi buyruqlardan iborat.

- DSL (Ma'lumotlarni O'zgartirish Tili) — bu jadvallarda qanday qiymatlar saqlanishini istalgan daqiqada aniqlovchi buyruqlar majmuasi.

- DCL (Ma'lumotlarni Boshqarish Tili) - foydalanuvchiga ma'lum obyektlar ustida malum ta'sir o'tkazishga ruxsat berish yoki bermaslikni aniqlovchi vositalardan iborat. SQL Standarti ANSI tomonidan aniqlangan va hozir ISO (Standartlashtirish bo'yicha xalqaro tashkilot) tomonidan qabul qilingan. Lekin kommersial ma'lumotlar bazalari dasturlari ANSI ni ogohlantirmasdan SQL ni kengaytiradilar, ya'ni o'zlari foydali deb hisoblagan har xil xossalarni qo'shadilar.

SQL so'rov tili — ma'lum talablar asosida berilganlar bazasiga murojaat qilib, undan so'rov talablariga javob beradigan natijalarni olish tili. Hozirgi paytda turli berilganlar bazasi va ularning sistemalari yaratilgan. Ammo relyatsion tipdagi barcha berilganlar bazasiga murojaat qilib, ulardan tegishli ma'lumotlarni olish SQL tilida ishlab chiqilgan. Bu tilda so'rovlar qanday vositalar yordamida tashkil qilinishiga to'xtalib o'tamiz. Shuni aytish joizki, ba'zi DBMS larda (masalan, Accessda) uni ishlatmasdan tanlashga so'rov yoki maxsus namuna bo'yicha so'rov blanki sifatida murojaat qilib tegishli ma'lumotlar olinadi, bunday so'rovlarni tashkil qilish uchun esa Базаданных — Создать (Berilganlar bazasi yaratish) buyrug'idan foydalaniladi. So'rov natijasini javob sifatida jadval ko'rinishida olish mumkin.

SQL — interaktiv so'rovlar tili. Foydalanuvchilar SQL komandalari asosida ishlovchi ma'lumotlarni o'qish va ularni ekranga chiqarishga mo'ljallangan

interaktiv dastur orqali berilganlar bazasi ma'lumotlaridan foydalanadilar. Bu esa maxsus so'rovlarni hosil qilishning qulay usulidir.

SQL — berilganlar bazasini dasturlash tili. Berilganlar bazasi bilan ishlashda dasturchilar o'zlarining dasturlarida SQL komandalaridan foydalanadilar. Bu uslub foydalanuvchi tomonidan yozilgan dasturda qo'llanilganidek berilganlar bazasining xizmatchi dasturlarida ham ishlatiladi.

SQL — berilganlar bazasida administratorlik qilish tilidir. Berilganlar bazasining administratori SQL tilidan berilganlar bazasi strukturasi aniqlashda va ulardan foydalanish huquqini boshqarishda foydalanadi.

SQL — klient — server texnologiyasi asosida dasturlarni tuzish tilidir. Shaxsiy kompyuterlar uchun tuzilgan dasturlarda SQL tili lokal tur orqali berilganlar bazasi serveri bilan aloqani tashkil etish uchun qo'llaniladi. Ko'pgina yangi dasturlarda klient — server arxitekturasida tarmoq trafigidan minimum darajada foydalanish hamda shaxsiy kompyuterlar singari berilganlar bazasi serverlari tezligini oshirish uchun ishlatiladi.

SQL — taqsimlangan berilganlar bazasi tili. Taqsimlangan berilganlar bazasi sistemalarini boshqarishda SQL bir necha o'zaro aloqa qiluvchi hisoblash sistemalari o'rtasida ma'lumotlarni taqsimlashga yordam beradi. Har bir sistemaning dasturiy ta'minoti SQL dan foydalanib boshqa sistemalar bilan aloqa qiladi, ularga so'rovlarni uzatadi.

SQL — berilganlar bazasida shlyuzlar o'rnatish imkonini beruvchi tildir. Hisoblash tarmoqlarida SQL dan bir BBBT dan boshqa turdagi BBBT bilan bog'lash uchun foydalanadilar.

SQL tili birinchi navbatda so'rovlarni bajarishga mo'ljallangan. SQL so'rovlarini ko'rish uchun qo'llaniladigan SELECT operatori boshqa SQL operatorlariga nisbatan ancha keng imkoniyatli hisoblanadi. SELECT operatori berilganlar bazasidan ma'lumotlarni o'qiydi va so'rov natijalarini jadval ko'rinishida qaytaradi. SELECT operatori MB jadvallardan natijaviy to'plam olish uchun mo'ljallangan ifodadir. Biz SELECT operatori yordamida so'rov beramiz, u

bo'lsa ma'lumotlar natijaviy to'plamini qaytaradi. Bu ma'lumotlar jadval shaklida qaytariladi. Bu jadval keyingi SELECT operatori tomonidan qayta ishlanishi mumkin va hokazo. SELECT operatori oddiy so'rovlar uchun ikkita zaruriy predlojeniyadan iborat bo'ladi. SELECT predlojeniyasida talab qilingan ustun nomlari keltiriladi. FROM predlojeniyasida esa bu ustunlarni saqlovchi jadvallar ko'rsatiladi. So'rov mantiqiy darajada FROM predlojeniyasida ko'rsatilgan jadvallarni satrma - satr ko'rib chiqish orqali bajariladi. Jadvalning har bir satridan natija sifatida chiqariladigan ustunlarning qiymatlari olinadi va so'rov natijasining bitta satri tuziladi. Shunday qilib oddiy so'rov natijalari jadvali berilganlar bazasi jadvalining har bir yozuvi uchun bitta berilganlar satri hosil qiladi. SELECT ifodasi o'z ichiga albatta FROM ifodasini olishi kerak. Qolgan ifodalar kerak bo'lsa ishlatiladi.

SELECT ifodasidan so'ng so'rovda qaytariluvchi ustunlar ro'yxati yoziladi. FROM ifodasidan so'ng so'rovni bajarish uchun jadvallar nomi yoziladi. WHERE ifodasidan so'ng agar ma'lum satrlarni qaytarish lozim bo'lsa, izlash sharti yoziladi. GROUP BY ifodasi guruhlariga ajratilgan natijaviy so'rov yaratishga imkon beradi. HAVING ifodasidan guruhlarni qaytarish sharti yoziladi va GROUP BY bilan birga ishlatiladi. ORDER BY ifodasi ma'lumotlar natijaviy to'plamini tartiblash yo'nalishini aniqlaydi. OFFICES jadvalidagi hamma yozuvlarni qaytaruvchi sodda so'rov ko'ramiz:

SELECT * FROM OFFICES

SELECT operatori albatta "qaytariluvchi ustunlar ro'yxati"ni o'z ichiga olishi kerak, ya'ni:

SELECT FIELD1, FIELD2, FIELD3 ... FROM ...

FIELD1, FIELD2, FIELD3 qaytariluvchi ustunlar ro'yxati bo'lib, ma'lumotlar ketma-ketligi shu tartibda qaytariladi. Ya'ni "qaytariluvchi ustunlar ro'yxati" hisoblanuvchi ustunlar va konstantalarni o'z ichiga olishi mumkin.

SELECT FIELD1, (FIELD2 - FIELD3) "CONST" ... FROM ...

FROM jumlası "jadval spetsifikatorlari ", ya'ni so'rovni tashkil qiluvchi jadvallar nomini o'z ichiga oladi. Bu jadvallar so'rov asosini tashkil qiluvchi jadvallar deyiladi.

Misol: Hamma xizmatchilarning nomlari, ofislari va ishga olish sanalari ro'yxatini hosil qilish.

SELECT NAME, REP_OFFICE, HIRE_DATE FROM SALESREPS

SELECT operatori qaytaruvchi ustunlar ixtiyoriysi hisoblanuvchi, ya'ni natijada mustaqil ustun sifatida tasvirlanuvchi matematik ifoda bo'lishi mumkin.

Misol: Har bir ofis uchun shaharlar, regionlar va sotuvlar rejasi qanchaga o'rtig'i yoki kami bilan bajarilganligi ro'yxati.

SELECT CITY, REGION, (SALES-TARGET) FROM OFFICES

Har bir xizmatchi uchun rejadagi sotuvlar hajmini haqiqiy sotuvlar hajmining 3% iga oshirish.

SELECT NAME, QUOTA, (QUOTA + ((SALES/100)*3))

FROM SALESREPS

Ba'zida ustunlardan biri izlash shartiga bog'liq bo'lmagan qiymat qaytarishi kerak bo'ladi.

Masalan: Har bir shahar uchun sotuvlar hajmlari ro'yxatini chiqaring.

SELECT CITY, 'Has sales of', SALES FROM OFFICES

'Has sales of' bu konstantalar ustunidir.

Ba'zida ma'lumotlarni tanlashda qaytariluvchi qiymatlar hosil bo'ladi. Bu hol yuz bermasligi uchun DISTINCT operatoridan foydalanish lozim.

Masalan: **SELECT DISTINCT MGR FROM OFFICES**

Endi WHERE ifodasidan foydalanib ba'zi so'rovlarni ko'rib chiqamiz:
Sotuvlar haqiqiy hajmi rejadani oshgan ofislarni ko'rsating.

**SELECT CITY, SALES, TARGET FROM OFFICES WHERE SALES
> TARGET**

Identifikatori 105 ga teng bo'lgan xizmatchi nomi haqiqiy va rejadagi
sotuvlar hajmini ko'rsating:

**SELECT SALES, NAME, QUOTA FROM SALESREPS WHERE
EMPL_NUM = 105**

Agar izlash sharti TRUE bo'lsa qator natijaviy to'plamga qo'shiladi, agar
izlash sharti FALSE bo'lsa, qator natijaviy to'plamga qo'shilmaydi, agar NULL
bo'lsa ham natijaviy to'plamdan chiqariladi. O'z ma'nosiga ko'ra WHERE
keraksiz yozuvlarni chiqarib, kerakligini qoldiruvchi filtr sifatida ishlatiladi.

Asosiy izlash shartlari - "predikatlar" beshta. Ularni ko'rib chiqamiz:

- Solishtirish, ya'ni bir shart natijasi ikkinchisi bilan solishtiriladi, birinchi
so'rov kabi.

- Qiymatlar diapazoniga tegishlilikni tekshirish. Masalan berilgan qiymat
diapazonga kiradimi yo'qmi.

- To'plam elementligini tekshirish. Masalan, ifoda qiymati to'plamdagi
biror qiymat bilan ustma - ust tushadimi.

- Shablonga moslikni tekshirish. Ustundagi satrli qiymat shablonga mos
keladimi.

- NULL qiymatga tenglikka tekshirish.

Solishtirish amallari maydon va konstantalarni solishtirish amallarini o'z
ichiga olishi mumkin: 1988 yilgacha ishga olingan hamma xizmatchilar
nomlarini toping.

SELECT NAME FROM SALESREPS

WHERE HIRE_DATE < TO_DATE('01.06.1988','DD/MM/YYYY')

TO_DATE('01.06.1988','DD/MM/YYYY') - PL/SQL Oracle sana bilan ishlash standart funksiyasi. Yoki arifmetik ifodalarni o'z ichiga olishi mumkin: Haqiqiy sotuvlar hajmi rejaning 80 foizidan kam bo'lgan ofislar ro'yxatini chiqaring.

SELECT CITY, SALES, TARGET FROM OFFICES

WHERE SALES < (0.8 * TARGET)

Ko'phollarda izlash birlamchi kaliti bo'yicha konstantalar bilan solishtirish so'rovlaridan foydalaniladi, masalan shahar telefon tarmog' iabonenti, axir ikkita bir xil nomerlar mavjud emas.

Har bir foydalanuvchi SQL ma'lumotlar bazasida nima qilish mumkinligini ko'rsatuvchi imtiyozlarga egadir. Bu imtiyozlar vaqt o'tishi bilan o'zgarishi, ya'ni eskilari o'chirilib yangilari qo'shilishi mumkin. SQL imtiyozlari bu ob'ekt imtiyozlaridir. Bu shuni bildiradiki, foydalanuvchi berilgan komandani ma'lumotlar bazasining biror ob'ekti ustida bajarishi mumkin. Ob'ekt imtiyozlari bir vaqtning o'zida foydalanuvchilar va jadvallar bilan bog'liq. Ya'ni imtiyoz ma'lum foydalanuvchiga ko'rsatilgan jadvalda, asos jadvalda yoki tasavvurda beriladi. Ixtiyoriy turdagi jadvalni yaratgan foydalanuvchi shu jadval egasidir. Bu shuni bildiradiki foydalanuvchi bu jadvalda hamma imtiyozlarga ega va imtiyozlarini shu jadvalning boshqa foydalanuvchilariga uzatishi mumkin. Foydalanuvchiga tayinlash mumkin bo'lgan imtiyozlar:

SELECT - bu imtiyozga ega foydalanuvchi jadvallarda so'rovlar bajarishi mumkin.

INSERT -bu imtiyozga ega foydalanuvchi jadvalda INSERT komandasini bajarishi mumkin.

UPDATE - bu imtiyozga ega foydalanuvchi jadvalda UPDATE komandasini bajarishi mumkin. Bu imtiyozni jadvalning ayrim ustunlari uchun cheklab qo'yish mumkin.

DELETE -bu imtiyozga ega foydalanuvchi jadvalda DELETE komandasini bajarishi mumkin.

REFERENCES -bu imtiyozga ega foydalanuvchi jadvalning ustunidan (yoki ustunlaridan) ajdod kalit sifatida foydalanuvchi tashqi kalit aniqlashi mumkin.

SQL tilida quyidagi asosiy ma'lumotlar turlari ishlatilib, ularning formatlari har xil MBBT lar uchun farq qilishi mumkin:

INTEGER - butun son (odatda 10 tagacha qiymatli raqam va ishora).

SMALLINT – “qisqa butun” (odatda 5 tagacha qiymatli raqam va ishora).

DECIMAL(p, q) - o'nli son, p raqam va ishoradan iborat ($0 < p < 16$). O'nli nuqtadan so'ng raqamlar soni q orqali beriladi ($q < p$, agar $q = 0$ bo'lsa tashlab yuborilishi mumkin).

FLOAT - haqiqiy son 15 ta qiymatli raqam va butun darajadan iborat.

DarajaMBBT tipi bilan aniqlanadi (masalan, 75 yoki 307).

CHAR(n) - uzunligi o'zgarmas, n ga teng bo'lgan simvolli qator ($0 < n < 256$).

VARCHAR(n) - uzunligi o'zgaruvchi, n simvoldan oshmagan simvolli qator ($n > 0$ va har xil MBBT larda har xil lekin 4096 dan kam emas).

DATE - maxsus komanda orqali aniqlanuvchi formatdagi sana, sana maydonlari bizning eramizdan oldin bir necha mingyilliklardan boshlanuvchi va bizning eramiz beshinchi- o'ninchi mingyilligi bilan cheklangan haqiqiy sanalarni o'z ichiga olishi mumkin.

TIME -maxsus komanda orqali aniqlanuvchi formatdagi vaqt (ko'zda tutilgan bo'yicha hh.mm.ss).

DATETIME - sana va vaqt kombinatsiyasi.

Jadvallar CREATE TABLE komandasi bilan yaratiladi. Bu komanda qatorlarsiz bo'sh jadval yaratadi. CREATE TABLE komandasi jadval nomini va jadval o'zini ma'lum tartibda ko'rsatilgan ustunlar nomlari ketma – ketligi ta'rifi ko'rinishida aniqlaydi. U ma'lumotlar tiplari va ustunlar o'lchovini aniqlaydi. Har bir jadval juda bo'lmaganda bitta ustunga ega bo'lishi kerak.

CREATE TABLE komandasi sintaksisi:

CREATE TABLE <table-name >
(<column name><data type>[(<size>)],
<column name><data type>[(<size>)], ...);

Argument qiymati kattaligi ma'lumot turiga bog'liqdir. Agar siz maxsus ko'rsatmasangiz, tizim avtomatik qiymatni o'rnatadi.

Jadvalni o'chirish imkoniga ega bo'lish uchun, jadval egasi (ya'ni yaratuvchisi) bo'lishingiz kerak. Faqat bo'sh jadvalni o'chirish mumkin. Qatorlarga ega bo'lgan, to'ldirilgan jadvalni o'chirish mumkin emas, ya'ni jadval o'chirishdan oldin tozalangan bo'lishi kerak. Jadvalni o'chirish komandasi quyidagi ko'rinishga ega:

DROP TABLE < table name >;

Masalan: **DROP TABLE Salepeople;**

Jadvalni o'zgartirish uchun ALTER TABLE komandasidan foydalaniladi. Bu komanda jadvalga yangi ustunlar qo'shish, ustunlarni o'chirish, ustunlar kattaligini o'zgartirish, hamda cheklanishlarni qo'shish va olib tashlash imkoniyatlariga ega. Bu komanda ANSI standarti qismi emas, shuning uchun har xil tizimlarda har xil imkoniyatlarga ega.

Jadvalga ustun qo'shish uchun komandaning tipik sintaksisi:

ALTER TABLE <table name> ADD <column name><data type><size>;

Masalan: ALTER TABLE Salepeople ADD Phone CHAR(7);

Hamma satrlar SQLda INSERT komandasi yordamida kiritiladi. INSERT quyidagi formatlar biriga ega bo'lishi mumkin:

INSERT INTO<table name | view name> [(column [,column] ...)]

VALUES (<value> [,<value>] ...); yoki

INSERT INTO<table name | view name> [(column [,column] ...)]

Masalan, sotuvchilar jadvaliga satr kiritish uchun quyidagi shartdan foydalanish mumkin:

INSERT INTO Salepeople VALUES (11, 'Peel', 'London', .12);

Nom kiritish uchun ustunlar ko'rsatish mumkin. Bu nomlarni ixtiyoriy tartibda kiritishga imkon beradi. Masalan:

INSERT INTO Salepeople (Sname, Comm, SNum)

VALUES ('Peel', .12, 11);

INSERT komandasidan bir jadvaldan qiymat tanlab, so'rov bilan ishlatish uchun, ikkinchisiga joylashishda foydalanish mumkin. Buning uchun VALUES ifodasini mos so'rovga almashtish kerak:

INSERT INTO Londonstaff

SELECT * FROM Salespeople

WHERE City = 'London';

Satrlarni jadvaldan DELETE komandasi bilan o'chirish mumkin. U aloxida qiymatlarni emas faqat satrlarni o'chiradi. DELETE quyidagi formatga ega:

DELETE FROM<table name | view name>

[WHERE search-condition];

Masalan, Sotuvchilar jadvalidagi hamma satrlarni o'chirish uchun, quyidagi shartni kiritish mumkin:

DELETE FROM Salepeople;

Ma'lum satrlarni o'chirish uchun predikatdan foydalaniladi. Masalan, jadvaldan Axelrod sotuvchini o'chirish uchun:

DELETE FROM Salepeople WHERE SNum = 13;

Maydon qiymatlarini o'zgartirish UPDATE komandasi yordamida bajariladi. Bu komandada UPDATE ifodasidan so'ng jadval nomi va SETifodasidan so'ng ma'lum ustun uchun o'zgartirish ko'rsatiladi. UPDATE ikki formatga ega. Ulardan birinchisi:

UPDATE <table name | view name>

SET column = expression [, column = expression] ...

[WHERE search-condition]

bu yerda expression - bu ustun | ifoda | konstanta| o'zgaruvchi.

Ikkinchi variant:

UPDATE <table name>SET column = expression, ...

[FROM table-list][WHERE search-condition]

Masalan, hamma buyurtmachilar bahosini 200 ga o'zgartirish uchun quyidagini kiritish mumkin:

UPDATE Customers SET Rating = 200;

Ma'lum satrlarni o'zgartirish uchun DELETE dagi kabi predikatdan foydalanish kerak. Masalan Peel (SNum=11) sotuvchining hamma buyurtmachilari uchun bir xil o'zgartirishni quyidagicha kiritish mumkin:

UPDATE Customers SET Rating = 200 WHERE SNum = 11;

SETvergul bilan ajratilgan ixtiyoriy sondagi ustunlarga qiymat tayinlashi mumkin. Ixtiyoriy jadval satrlari uchun qiymat tayinlanishi mumkin, lekin bir vaqtning o'zida faqat bitta satrga qiymat tayinlanadi. Masalan:

UPDATE Salepeople SET SName = 'Gibson', City = 'Boston', Comm = .10 WHERE SNum = 14;

UPDATE komandasiningSET jumlasida skalyar ifodalardan o'zgartirilayotgan maydon ifodasiga qo'shgan holda foydalanish mumkin.Masalan: **UPDATE Salepeople SET Comm = Comm * 2;**

SQL yordamida so'rovlarni bir - birining ichiga joylashtirish mumkin. Odatda, ichki so'rov qiymat hosil qiladi va bu qiymat tashqi predikat tomonidan tekshirilib, to'g'ri yoki noto'g'riligi tekshiriladi.

Misol: bizga sotuvchi nomi ma'lum: Motika, lekin biz SNum maydoni qiymatini bilmaymiz va buyurtmachilar jadvalidan hamma buyurtmalarni ajratib olmoqchimiz. Buni quyidagichaamalgaoshirish mumkin:

SELECT * FROM Orders

**WHERE SNum =(SELECT SNum FROM Salepeople
WHERE SName = 'Motika');**

Avval ichki so'rov bajariladi, so'ngra uning natijasi tashqi so'rovni hosil qilish uchun ishlatiladi(SNum ostki so'rov natijasi bilan solishtiriladi).

Ostki so'rov bitta ustun tanlashi lozim, bu ustun qiymatlari tipi predikatda solishtiriladigan qiymat tipi bilan bir xil bo'lishi kerak. Ba'zi hollarda ostki so'rov bitta qiymat hosil qilishi uchun DISTINCT operatoridan foydalanish mumkin.

Misol: Hoffman (CNum=21) ga xizmat ko'rsatuvchi sotuvchilar hamma buyurtmalarini topish lozim bo'lsin.

SELECT * FROM Orders

**WHERE SNum = (SELECT DISTINCT SNum FROM Orders
WHERE CNum = 21);**

Bu holda ostki so'rov faqat bitta 11 qiymat chiqaradi, lekin umumiy holda bir necha qiymatlar bo'lishi mumkin va ular ichidan DISTINCT faqat bittasini tanlaydi.

Ixtiyoriy sondagi satrlar uchun avtomatik ravishda bitta qiymat hosil qiluvchi funksiya turi - agregat funksiyabo'lib, undan ostki so'rovda foydalanish mumkin.

Masalan, siz summasi 4 oktyabrdagi bajarilishi lozim bo'lgan buyurtmalar summasi o'rta qiymatidan yuqori bo'lgan hamma buyurtmalarni ko'rmoqchisiz:

SELECT * FROM Orders

**WHERE AMT >(SELECT AVG (AMT) FROM Orders
WHERE ODate = '1990/10/04');**

Shuni nazarda tutish kerakki guruhlangan agregat funksiyalar GROUP BY ifodasi terminlaridaaniqlangan agregat funksiyalar bo'lsa ko'p qiymatlar hosil qilishi mumkin.

Agar ostki so'rovda IN operatoridan foydalanilsa, ixtiyoriy sondagi satrlar hosil qilish mumkin.

Misol: Londondagi sotuvchilar uchun hamma buyurtmalarni ko'rsatish.

SELECT * FROM Orders

**WHERE SNum IN (SELECT SNum FROM Salepeople
WHERE City = 'London');**

Bu natijani jamlanmaorqali hosil qilish mumkin. Lekin odatda ostki so'rovli so'rovlar tezroq bajariladi. Ostki so'rov SELECT jumlasida ustunga asoslangan ifodadan foydalanish mumkin. Bu relyatsion operatorlar yordamida yoki IN yordamida amalga oshirilishi mumkin. Ostki so'rovlarni HAVING ichida ishlatish mumkin. Bu ostki so'rovlar agar ko'p qiymatlar qaytarmasa xususiy agregat funksiyalaridan yoki GROUP BY yoki HAVING operatorlaridan foydalanishi mumkin.

**Misol: SELECT Rating, COUNT (DISTINCT CNum) FROM Customers
GROUP BY Rating HAVING Rating >(SELECT AVG (Rating) FROM
Customers WHERE City = 'San Jose');**

Bu komanda San Jose dagi baholari o'rtachadan yuqori bo'lgan buyurtmachilarni aniqlaydi.

SQL tilida ostki so'rovlardan foydalanilganda tashqi so'rov FROM qismidagi ichki so'rovga mutanosib so'rov yordamida murojaat qilish mumkin. Bu holda ostki so'rov asosiy so'rov har bir satri uchun bir martadan bajariladi.

Misol: 3- oktyabrda buyurtma bergan hamma buyurtmachilarni toping.

**SELECT * FROM Customers a WHERE '1990/10/03' IN
(SELECT ODate FROM Orders b WHERE a.CNum = b.CNum);**

Bu misolda tashqi so'rovning CNum maydoni o'zgargani uchun ichki so'rov tashqi so'rovning har bir satri uchun bajarilishi kerak. Ichki so'rov bajarilishini talab qiladigan tashqi so'rov satri joriy satr -kandidat deyiladi. Mutanosib ostki so'rov bilan bajariladigan baholash protsedurasi quyidagicha:

- Tashqi so'rovda nomlangan jadvaldan satrni tanlash. Bu kelajak satr -kandidat.

- Tashqi so'rov FROM jumlasida nomlangan psevdonimda bu satr -kandidat qiymatlarini saqlab qo'yish.

- Ostki so'rovni bajarish. Tashqi so'rov uchun berilgan psevdonim topilgan hamma joyda joriy satr-kandidat qiymatidan foydalanish. Tashqi so'rov satr-kandidatlari qiymatlaridan foydalanish tashqi ilova deyiladi.

- Tashqi so'rov predikatini 3 qadamda bajariluvchi ostki so'rov natijalari asosida baholash. U chiqarish uchun satr-kandidat tanlanishini belgilaydi.

- Jadval keyingi satr-kandidatlari uchun protsedurani qaytarish va shu tarzda toki hamma jadval satrlari tekshirilib bo'lmaguncha.

Ba'zida xatolarni topish uchun maxsus yaratilgan so'rovlardan foydalanish kerak bo'ladi.

Misol: Quyidagi so'rov buyurtmachilar jadvalini ko'rib chiqib, SNum va CNum mos kelishini tekshiradi va mos bo'lmagan satrlarni chiqaradi.

SELECT * FROM Orders main WHERE NOT SNum =

(SELECT SNum FROM Customers WHERE CNum = main.CNum);

Asosiy so'rov asoslangan jadvalga asoslanuvchi mutanosib so'rovdan foydalanish mumkin.

Misol: sotib olishlar buyurtmachilari uchun o'rta qiymatdan yuqori bo'lgan hamma buyurtmalarni topish.

**SELECT * FROM Orders a WHERE AMT > (SELECT AVG (AMT)
FROM Orders b WHERE b.CNum = a.CNum);**

Standart SQL tilida ma'lumotlarni kiritish INSERT buyrug'i asosida amalga oshiriladi. Bu buyruq bitta satrni kiritishga imkon beradi, keyingi satrlaming kiritilishi INSERT buyrug'ining qaytarilishi yordamida hosil bo'ladi:

INSERT INTO factory (Name, Dob, Pol, Los, Dept, Salary, Telno) Values (Paul F. 05.07.1962, 'm', 12 'worker', 2250.75, '420027');

Barcha satrli o'zgaruvchilar apostroflarga kiritilishi lozim. Agar biz biror o'zgaruvchining ma'nosini bilmasak, kiritishda uning ma'nosini tushirib qoldirish mumkin, EHM uning o'rnini avtomatik ravishda bo'shliq bilan to'ldiradi.

Misol: INSERT INTO factory (Name, Dob, Pol, Dept, Salary)

Values ('Clint E', 12. 12.1970, 'm\ 'tecknic', 2400);

Bunday holatda Telno va Los o'rinlari, toki ular tartibli aniqlanmaguncha, EHM yordamida probel va nullar bilan to'ldiriladi, bu jarayon korrekt yaqinlangunicha davom ettiriladi. Alohida qiymatlar tushib qolgan bo'lsa, ularni ham NUll bilan to'ldirish mumkin.

Select buyrug'i

Bu bo'limda foydalanuvchining jadvallar bilan ishlash usullari ko'rib chiqiladi. Select buyrug'i SQL tilining asosiy buyruqlaridan biri hisoblanadi. Bu buyruq barcha amallarni ma'lumotlar qatori bilan ta'minlaydi. Quyida select buyrug'ining asosiy imkoniyatlarini ko'rib chiqamiz.

a) Barcha ma'lumotlarni ko'rish .

```
Select* from tactory;
```

(F) belgisi factory jadvalining barcha ustunlarini tanlash kerakligini bildiradi.

Bu natijani Select buyrug'idan keyin barcha ustunlar nomini berish bilan olsa bo'ladi:

```
Select Name, Dob, Pol, Los, Dept, Salary, Telno  
from factory;
```

Natijada ekranda 1-jadvalning hamma ustunlari paydo bo'ladi.

Ustunlar tartibi Select buyrug'ida belgilangandek bo'ladi:

b) ustunlarni tanlabchaqirish.

Ustunlarni alohida chaqirish uchun Select buyrug'ida ko'rsatish kerak:

```
Select Name, Dept from factory;
```

Natijada bir Name va Dept dagi jadvallarni olamiz:

d) Satrlarni tanlab ko'rish . Ma'lumotlar bilan ishlaganda ko'pincha jadvaldan faqat aniq bir satrni ko'rish lozim bo'lgan hollar uchray turadi. Bunday holda Select buyrug'ining umumiy ko'rinishi quyidagicha bo'ladi:

Select — ustunlar nomi;

from — jadvallar nomi;

where — satr tanlash sharti.

Misol: 1-jadvaldan barcha xodimlar, ishchilarni tanlash uchun quyidagini kiritish kerak:

Select name, dept

from factory

Where Dept—dmik; buyrug'ini berish lozim.

Natijada ekranda quyidagilar hosil bo'ladi:

e) Solishtirish opera torlarining qo'llanilishi.

Awalgi misolda satr tanlash shartini tasvirlashda <<tenglik>> belgisi qo'llangan edi.

<<Tenglik>> belgisidan tashqari yana solishtirish shartini tanlashda qo'llanilishi mumkin bo'lgan 7 ta eng oddiy solishtirish operatorlari mavjud.

Bular:

!= — teng emas;

> — katta;

Name

Rustam V.

Joan A .

admin

admin;

Dept

<> — teng emas;

! > — berilgandan katta emas;

< — berilgandan kichik;

! < — berilgandan kichik emas;

>= — katta yoki teng;

<= — kichik yoki teng.

Quyidagi misol solishtirish operatorlarining qo'llanish imkoniyatlarini ko'rsatadi.

1-misol: Factory jadvalidan ish staji 10 yildan ortiq bo'lgan ishchilarning ismlarini tanlash.

Select Name, Los

from factory

where Los > 10;

Natijada ekranda

NAME Los

Rustam V. 22

Barot V. 21

Azamat T. 17

Dolly S. 14

Paul F. 12

Joan A. 11 chiqadi.

2-misol: Factory jadvalidan 12.12.1970 dan keyin tug'ilganlarni tanlash.

Select Name, Dob

from factory

where Dob>12.12.1970;

Ekranda hisobot quyidagicha bo'ladi:

NAME Dob

Barot V. 02.07.1975

Maria G. 20.11.1972

ViancaA. 14.03.1975

JaneS. 01.02.1979

Yuqorida tasvirlangan solishtirish operatorlaridan tashqari, quyidagi operatorlarni qo'llash orqali ham qiyoslash mumkin.

1) between... and

2) is null

3) like

4) in

Between operatori aniq bir sohani ajratishga mo'ljallangan.

Misol:

select Name, Salary

from factory

where Salary between 2150 and 2350;

Ushbu misol factory jadvalida oylik maoshi 2150—2350 oralig'ida

bo'lgan xodimlar ro'yxatini ko'rsatadi.

NAME Los

Rustam V. 22

Barot V. 21

Azamat T. 17

Dolly S. 14

Name Salary

Azamat T. 2300

Barot V. 2250

Raul T. 2250

Don T. 2250

Donoron A. 2250

SQL not between birikmasining qo'llanishiga yo'l qo'yadi.

Misol:

```
select Name, Salary
```

```
from factory
```

```
where Salary not between 2150 and 2350;
```

Bu ekranda oylik maoshi 2150—2350 oralig'idan tashqarida bo'lgan xodimlarning ro'yxatini ko'rsatadi.

Is null operatori axborot bo'lmagan satrlarni tanlash imkonini beradi.

Misol: Factory jadvalidan telefoni bo'lmagan ishchilar ro'yxatini tanlash.

```
Select Name, Telno
```

```
from factory
```

```
where Telno. Is null;
```

Natijada quyidagi jadvalga ega bo'lamiz;

```
Name Telno
```

```
Jane S.
```

```
DonT.
```

Like operatori. Like operatori satrli o'zgaruvchilarni solishtirishni ta'minlaydi. EHM satrli o'zgaruvchilarni shablon bo'yicha solishtiradi.

Bu holda satrli o'zgaruvchini solishtirishda shablondan keyin <<%>> belgisi turadi, o'zgaruvchining o'zi qavs ichida yoziladi.

Misol: Factory jadvalidagi Name ustunida kimning nomi <<DO>> dan boshlansa, o'shani ekranga chiqarish.

```
select Name from factory
```

```
where Name like, 'DO%';
```

Natijada ekranda quyidagilar hosil bo'ladi:

Name

Dolly S.

Don T.

Donowan A.

Bu operator satrli kattaliklarni alohida harflari aniq bo'lganda tanlashni ta'minlaydi.

Misol uchun:

```
select Name from factory
```

```
where Name like <<AN%>>;
```

Chizish AN dan oldin har qancha miqdor bo'lish imkoni borligini anglatadi.

Natijada quyidagilar hosil bo'ladi:

Name

Bianca A.

Jane S.

Joan A.

Donovan A.

Bo'limda muhokama qilinadigan eng oxirgi operator In operatori bo'ladi. U tanlov asosida boshqa ma'lumotlarga mos bo'lgan ma'lumotlarni yig'ish imkonini beradi.

Masalan:

```
select Name, Dept
```

```
from factory
```

```
where Dept in ('admin', 'account');
```

Berilgan buyruq asosida EHM factory jadvalidan administrativ (ma'muriy) bo'limda va buxgalteriyada ishlayotgan xodimlarning ro'yxatini keltiradi.

Name Dept

Rustam V. admin

Edwin E. Account

Dolly S. Account

(I — IV) punktlarda tasvirlangan barcha operatorlar NOT operatori bilan ham ishlashi mumkin;

f) shartlarni o'zaro almashtirish .

Hozirgacha biz jadvaldan ma'lumotlarni tanlab olishda faqat bitta shartdan foydalandik. SQL tili bitta iborada ikki yoki undan ortiq shartlarning qo'llanishiga imkon beradi. Bu holatda shartlar bir-biridan AND yoki OR biriktiruvchi kreditlar yordamida ajratiladi. Quyida biz shartlarni almashtirishni qo'llash imkoniyatini ko'rsatadigan ikkita misol keltiramiz.

1-misol. Factory jadvalidan worker bo'limiga tegishli va staji 10 yildan ortiq bo'lgan xodimlarning ro'yxatini tanlash.

Select Name, Los, Dept

from factory

where Dept='worker' and Los> 10;

Natijada ekranda quyidagilar hosil bo'ladi:

Name Los Dept

Erkin J. 17 worker

Paul T. 12 worker

2-misol. Factory jadvalidan 01.01.1960 dan keyin tug'ilgan, oylik maoshi 2050 dan kam yoki 2450 dan ortiq, telefon raqami bor xodimlarning ro'yxati, tug'ilgan sanasi, oylik maoshi va telefon raqamlarini tanlash.

Select Name, Dob, Salary, Telno

from factory

where salary < 2050 and salary > 2450 and Dob>01.01.1960;

Arifmetik operatsiyalar. SQL faqat tanlashni emas, balki ma'lumotlar bilan arifmetik amallarni bajarishni ham ta'minlaydi.

Masalan:

```
Select Name, Salary*1.2
```

```
from factory
```

```
where Dept= 'service';
```

Natijada ekranda quyidagilar hosil bo'ladi:

```
Name Salary* 1.2
```

```
Bianca A. 2340
```

```
Jane S. 2280
```

Agar oxirgi ifodada shart bo'lmasa, EHM barcha xodimlarning oylik maoshini 20% ga oshiradi. Shuni aytib o'tish kerakki, factory jadvalining o'zida Bianca A. va Jane S. ning oylik maoshi o'zgarishsiz qoladi. Jadval ma'lumotlarini o'zgartirish uchun maxsus buyruqlar kerak, ular haqida keyinroq suhbatlashamiz. Ma'lumotlar bazasi bilan ishlaganda shu kundagi sana va vaqtni bilish kerak. Shu maqsadda SQLda quyidagi operatorlar nazarda tutilgan.

SYSTIME — soat, minut, sekund formatida shu kungi vaqtni belgilaydi.

SYS DATE — kun, oy, yil formatida shu kungi sanani belgilaydi.

Misol:

```
Select Name, Sysdate,systime
```

```
from factory
```

```
where Salary = 2700 or Salary =3000
```

Natijada quyidagiga ega bo'lamiz.

```
Name Sysdate Systime
```

```
Rustam B. 14.01.1997. 14.02.27.
```

Shuningdek, sysdate va systime operatorlari bilan arifmetik operatsiyalarni bajarish mumkin.

Masalan:

```
Select Name, Sysdate+31
```

```
from factory
```

where Name like 'Rustam%';

Berilgan buyruq asosida, shu kungi sanaga EHM 31 kunni qo'shib qo'yadi.

Natijada quyidagilar hosil bo'ladi:

Name Sysdate+31

Rustam B. 14.02.97.

SQL sana va vaqt bilan bo'lgan arifmetik operatsiyalardagi literallar kabi, sana va vaqtning har xil konstantalarini qo'llashga imkon beradi. Shu maqsadda SQL ga yil(lar), oy(lar), kun(lar), soat(lar), minut(lar), sekund(lar), mikrosekund(lar), YEAR(S), month(S), Day(S), HOUR(S), MINUTE(S), SECOND(S), MICROSECUND(S) lar kiritiladi.

Quyidagi misol mazkur konstantalarni qo'llash usulini ko'rsatadi.

Select Name, Dob, Dob+5 years

from factory

where Name like 'Dob%';

Display ekranida quyidagilar hosil bo'ladi:

Name Dob Dob+5 years

DonT. 11.04.1966 11.04.1971

Donovan A. 14.05.1967 14.05.1972

II.BOB. Universitet informatsion tizimida fan dasturlardagi o'quv adabiyotlar monitoringi modulini yaratish.

2.1. O'quv adabiyotlar monitoringi modulini yaratish spetsifikatsiyasi.

O'quv adabiyotlar monitoringi bu talim muassasasidagi o'qituvchi va talabalarning adabiyot qo'shish va yuklab olish jarayonini o'z ichiga oladi.

Bu monitoringda o'qituvchining adabiyotlarni istalgan joydan va istalgan vaqtda qo'shish imkoniyatini yaratsa, talabalarga esa bu adabiyotlardan istalgan joydan va istalgan vaqtda foydalanish imkoniyatini yaratadi.

Adabiyotlar nomi va muallifi ham keltiriladi, ularning qaysi fanga tegishli ekanligi, bu fan esa qaysi mutaxassislikka kirishi dasturimizda ko'zda tutilgan.

Har bir o'qituvchi o'ziga biriktirilgan fan yuzasidan o'quv adabiyotlarni qo'shgan bo'lsa, u holda universitet bazasidan bu adabiyotlar o'rin olgan bo'ladi. Talabalar tizimga murojaat qilganida bu adabiyotlarning ro'yxati ularga ko'rinadi. Demak, talabalar va o'qituvchilar oynalari o'zaro bog'langan bo'ladi. Bu oynalarga bosh menyudan kirish mumkin bo'ladi.

Tizim Universitet bazasiga ulangan va bu tizimga har bir foydalanuvchi o'z login va paroli bilan kirganida shu foydalanuvchiga tegishli bo'lgan fanlar va adabiyotlar hosil bo'ladi.

2.2. O'quv adabiyotlar monitoringi modulinining dasturi loyihasi.

Biz yaratgan o'quv adabiyotlar monitoringi dasturiy tizimining loyihasi quyidagilardan iborat:

- Oldindan mavjud tizimga o'qituvchi, talaba, fan, mutaxassislik va adabiyot haqidagi ma'lumotlar oldindan kiritib qo'yiladi. Bu oldindan kiritilgan ma'lumotlar shu ma'lumotlarni kiritishga mas'ul xodimlar tomonidan kiritiladi.

- Bu kiritilgan ma'lumotlarni o'quv bo'limi ko'rib chiqadi hamda tasdiqlaydi.

- Bu tayyor ma'lumotlar va yaratilgan rollar asosida o'qituvchilar tizimga o'zlariga tegishli fanlarga oid o'quv adabiyotlarni kiritish imkoniyatiga ega bo'ladilar. Kiritilgan o'quv adabiyotlar ichidan kerak bo'lmaganlarini o'chirish imkoniyati ham mavjud bo'ladi.

- O'quv adabiyotlarni olish uchun talabalar o'z parol va loginlari bilan tizimga kirib bu adabiyot(adabiyot, o'quv qo'llanma, uslubiy qo'llanma)lardan foydalana oladilar.

O'qituvchi va talabalarning tizimga kirishlari login va parol orqali himoyalab qo'yilgan bo'ladi.

O'qituvchi adabiyot qo'shadi:

```
namespace adabiyotlarMyBMI.Uqituvchi{
```

```

public partial class addAdabiyot : System.Web.UI.Page {
    SqlConnection cnn;
    int fan, uqt, fan1;
    protected void Page_Load(object sender, EventArgs e) {
        fan = int.Parse(Request.QueryString["fan"].ToString());
        uqt = int.Parse(Request.QueryString["uqt"].ToString());
        cnn = new SqlConnection(ConfigurationManager.ConnectionStrings["uquv-
2ConnectionString"].ConnectionString);
        cnn.Open();
        SqlCommand cmd = new SqlCommand() {
            Connection = cnn,
            CommandText = "select fan from mut_fanlari where id=" + fan,
            CommandType = CommandType.Text };
        SqlDataReader str = cmd.ExecuteReader();
        if (str.HasRows) {
            str.Read();
            fan1 = int.Parse(str["fan"].ToString()); }
        str.Close();
        cnn.Close(); }
    protected void Button1_Click(object sender, EventArgs e) {
        string fileName = "";
        HttpPostedFile postedFile = FileUpload1.PostedFile;
        if (postedFile.ContentLength > 0) {
            fileName = System.IO.Path.GetFileName(postedFile.FileName);
            postedFile.SaveAs(Server.MapPath(@"\kitoblar\" + fileName)); }
        cnn.Open();
        SqlCommand cmd = new SqlCommand() {
            Connection = cnn,
            CommandText = "insert into
adabiyotlar(tili,turi,nomi,fan_id,uqituvchi,faylname,avtor) values(" +

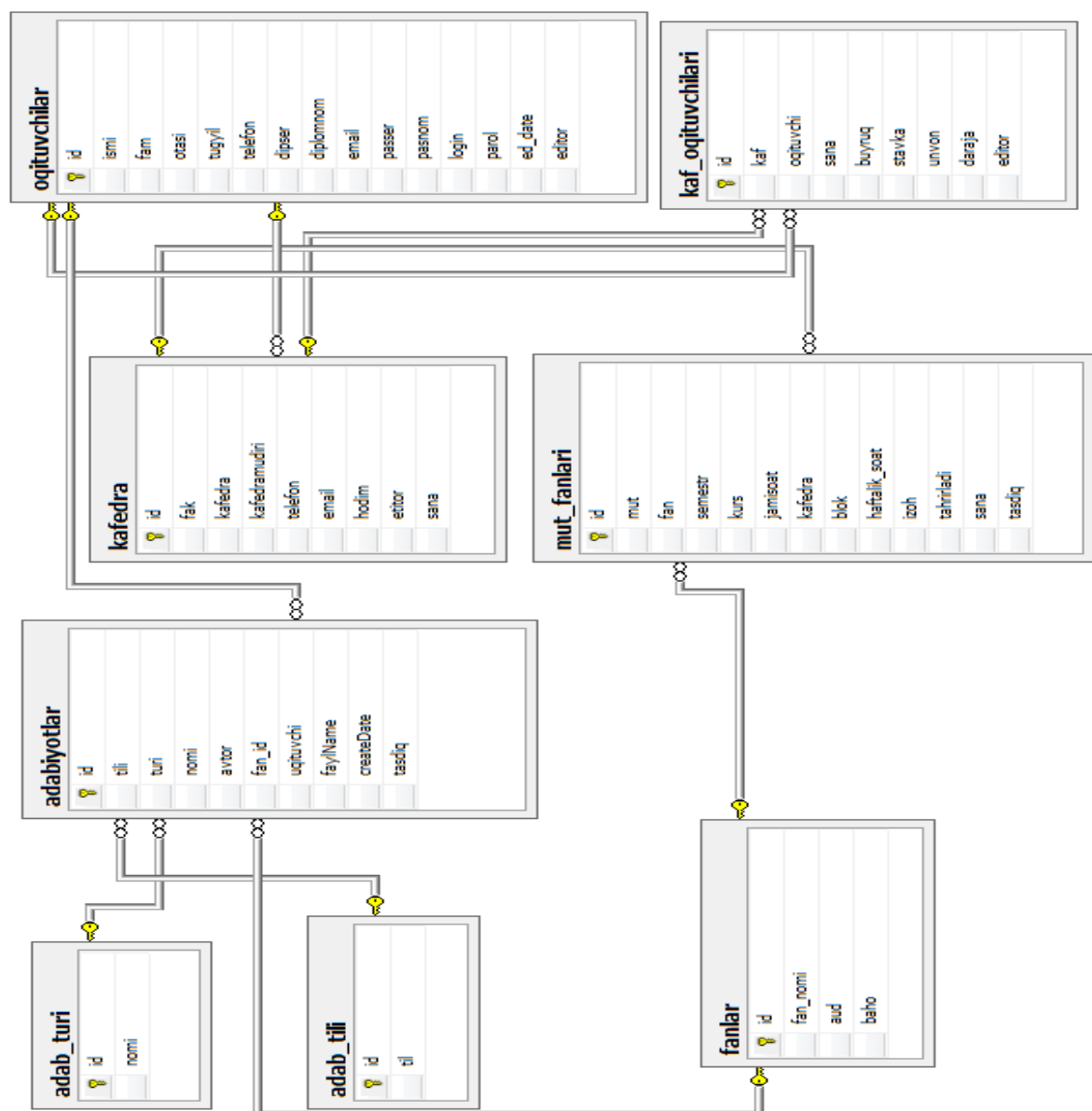
```

```

DropDownTili.SelectedValue + "," + DropDownTuri.SelectedValue + "," +
txtName.Text + "," +
fan1 + "," + uqt + "," + fileName + "," + txtName0.Text + ")",
CommandType = CommandType.Text      };
cmd.ExecuteNonQuery();
cnn.Close();
Response.Redirect("Default.aspx?uqt=" + Session["uqt"]);    }    }}

```

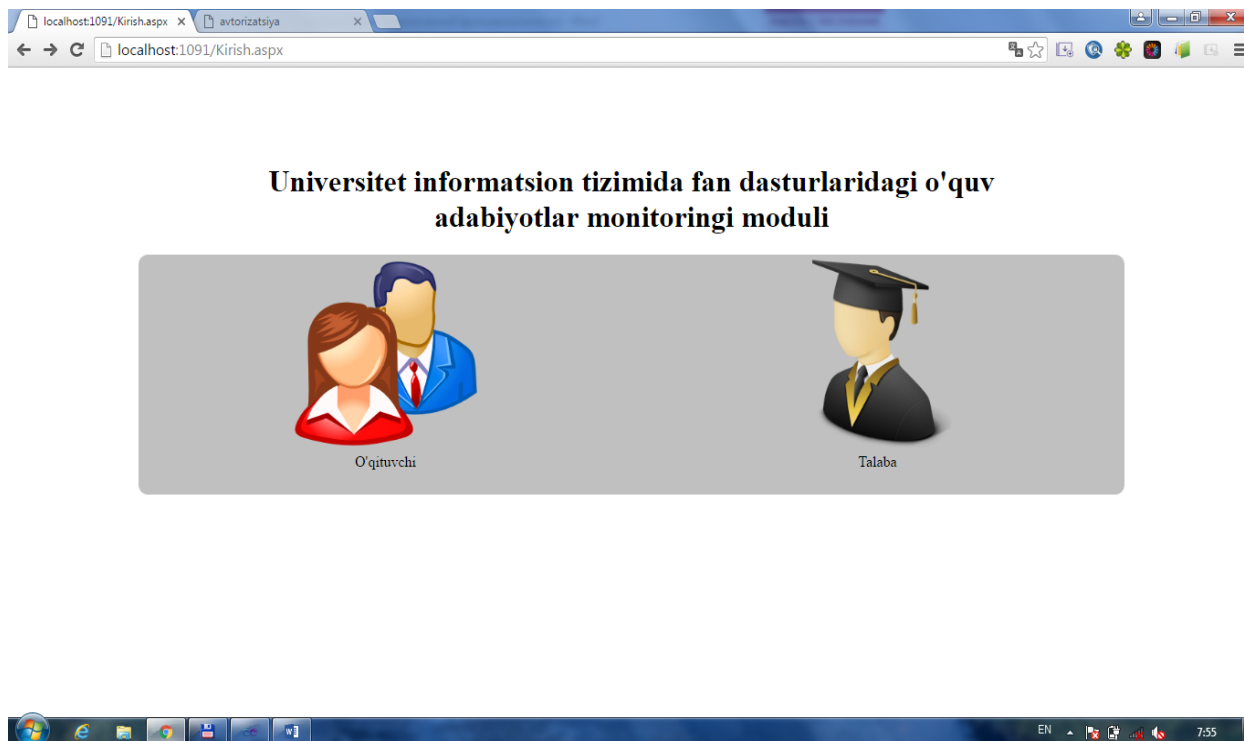
2.2.1. ER modeli.



2.1-chizma. Ma`lumotlarining ER modeli.

2.2.2. Foydalanuvchi grafik interfeysi.

Dasturga kirilganda foydalanuvchi oynasi ochiladi. Bu oynadan foydalanuvchi o'zi uchun kerakli bo'limni tanlaydi: agar bu foydalanuvchi talaba bo'ladigan bo'lsa, u "talabalar" ni, o'qituvchi bo'lsa, u holda "o'qituvchilar" ni tanlaydi va keyingi oynaga o'tiladi.



2.2 – chizma. Foydalanuvchilar oynasi.



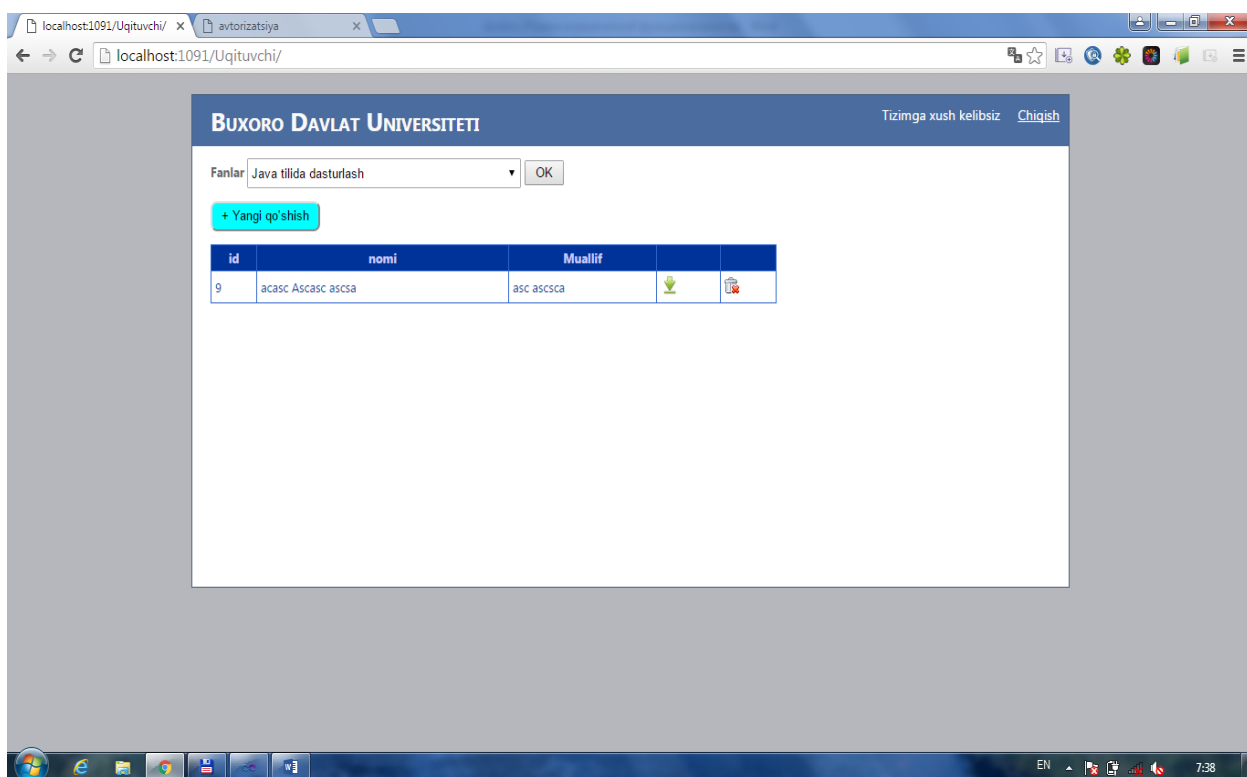
2.3 – chizma. O'qituvchilar oynasiga kirish.

Agar foydalanuvchi o'qituvchi bo'lsa shu oyna orqali o'z bo'limiga o'tadi. Bu oynada o'qituvchi o'z login va parolini kiritadi va “kirish” tugmasi bosish orqali keyingi oynaga kiriladi. Foydalanuvchi uchun “chiqish” tugmasi ham berilgan bo'lib, bu tugma bosilganda 2.2 – chizmadagi oynaga qaytiladi.



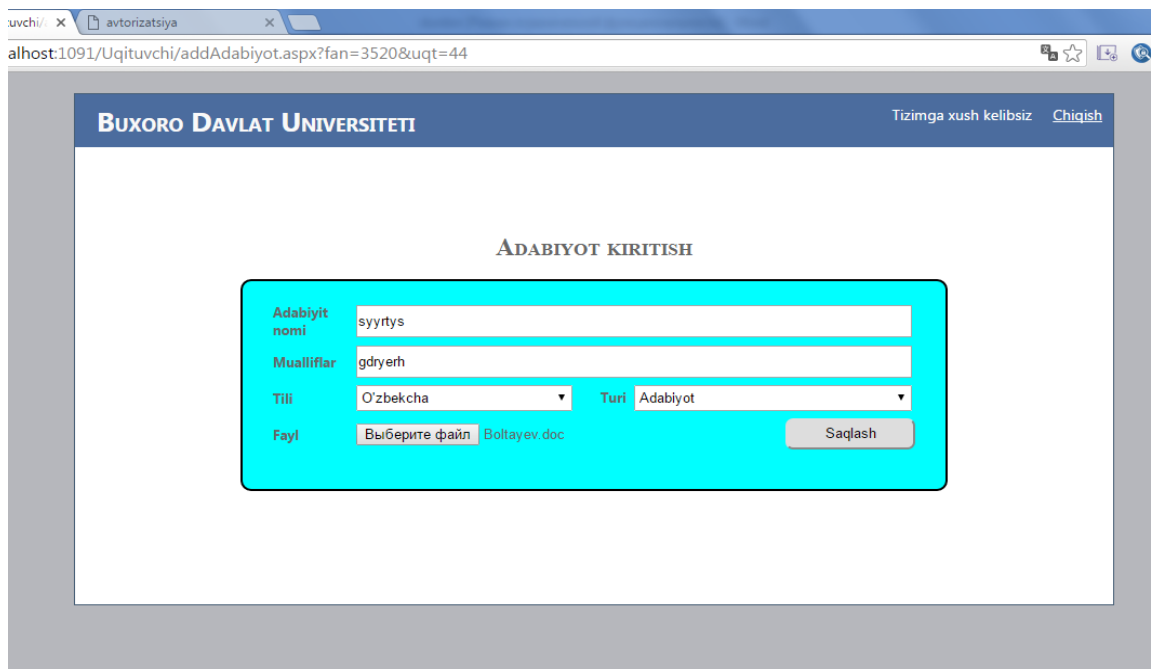
2.4 – chizma. O’qituvchining o’z login va parolini kiritish jarayoni.

Keyingi oyna o’qituvchining o’ziga tegishli bo’lgan oyna bo’lib, unda o’qituvchiga biriktirilgan fanlar yacheykasidan kerakli fan tanlanadiva“ok” tugmasi bosilganda bu fanga oid barcha adabiyotlar chiqadi.



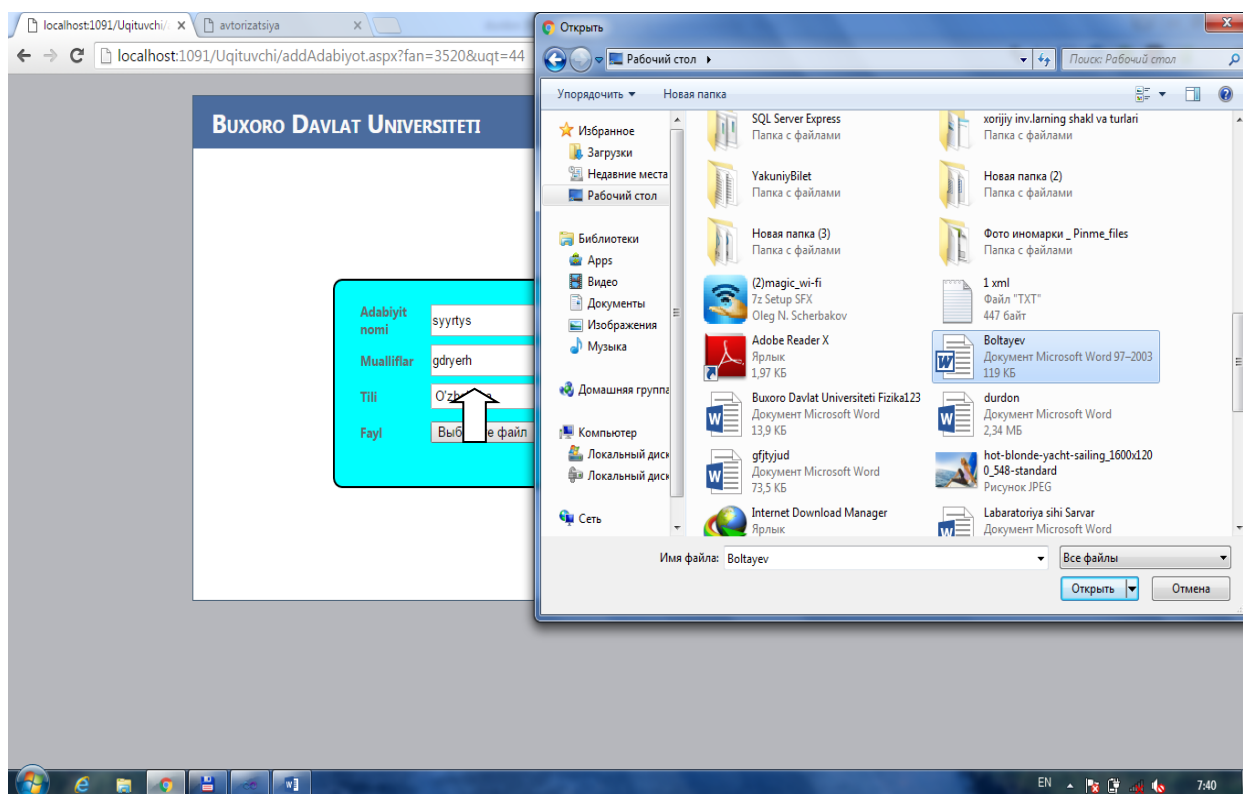
2.5 – chizma. O’qituvchi oynasining ko’rinishi.

Oynada mavjud adabiyotlarni o’chirish va yuklab olish mumkin. Qo’shilgan adabiyotlarning ro’yxati mavjud. Bu ro’yxat “id”, “nomi”, “muallif” ustunlaridan iborat. Adabiyot qo’shish uchun “yangi qo’shish” tugmasi bosiladi va adabiyot qo’shish oynasiga murojaat bo’ladi.



2.6 – chizma. Adabiyot kiritish oynasi..

Bu oynadan kerakli adabiyotning nomini, muallifini, tilini, turini tanlaymiz va “выберите файл” tugmasi bosiladi va yuklash jarayonini boshlaymiz.



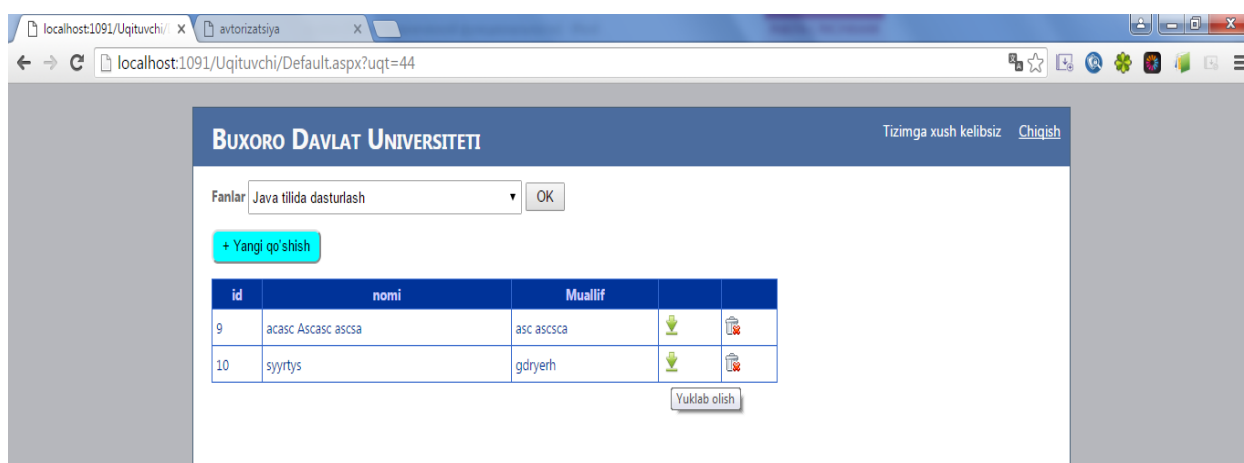
2.7 – chizma. Adabiyotni yuklash.

Bu oynadan “открыть” tugmasi bosiladi vaadabiyot kiritish oynasidagi saqlash tugmasi bosiladi.

ADABIYOT KIRITISH

Adabiyot nomi	syirtys		
Mualliflar	gdryerh		
Tili	O'zbekcha	Turi	Adabiyot
Fayl	Выберите файл Boltayev.doc		Saqlash

2.8 – chizma. Saqlash jarayoni.



2.9 – chizma. Adabiyotning ro'yxatga qo'shilishi.

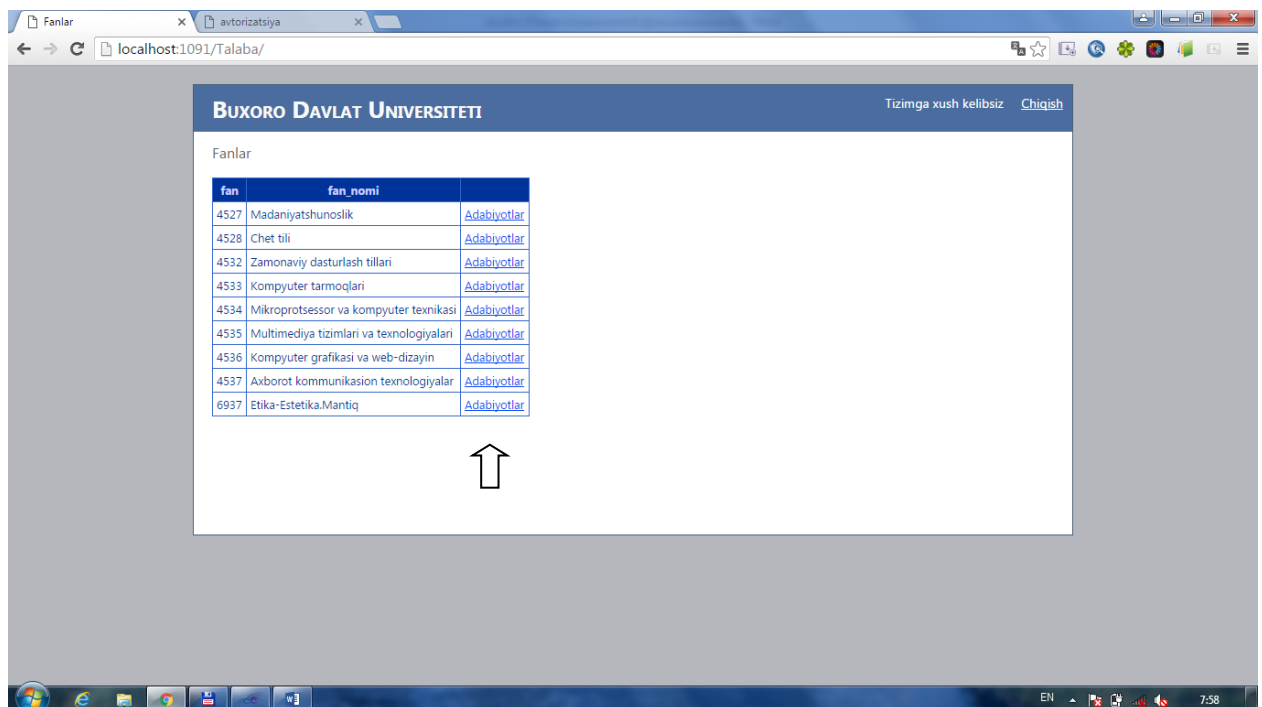


2.10 – chizma. Talaba oynasiga kirish.

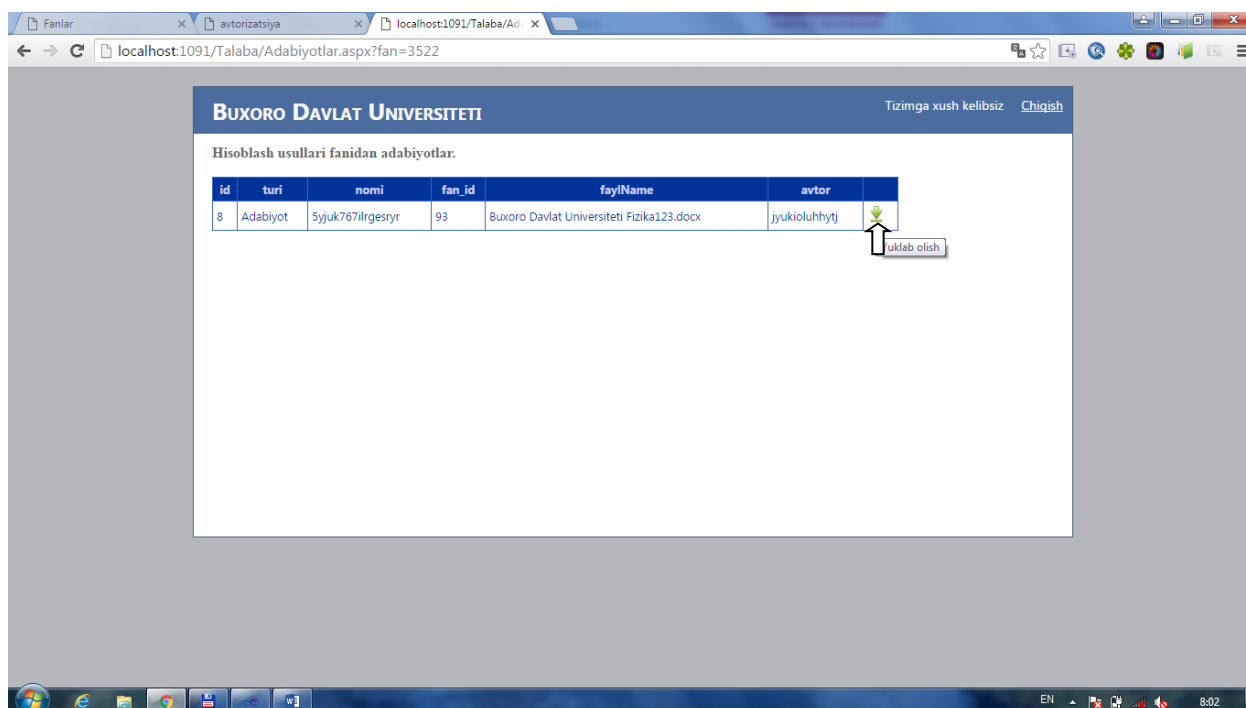


2.11 – chizma. Talabalarning tizimga kirish jarayoni.

Talabalar tizimga kirish uchun login va parolini kiritishi kerak. Tizimga kirilganda hosil bo’ladigan oynada fanlar ro’yxati turadi. Kerakli fan uchun “adabiyotlar”ni bosganimizda shu fanga oid barcha kiritilgan adabiyotlarni chiqarib beradi.



2.12 – chizma. Talabalar oynasining ko’rinishi.



2.13 – chizma. Yuklab olish jarayoni.

Talaba o'zi uchun kerakli bo'lgan adabiyotni topganidan so'ng, o'ng tomondagi "yuklash" tugmasi bosiladi. Shuning bilan dasturning ishi tugaydi, foydalanuvchi yana boshqa adabiyotlarni yuklab olishi mumkin.

2.2.3. Asosiy modullari.

O'quv adabiyotlar monitoringi 2 ta asosiy moduldan tashkil topgan bular:

- O'qituvchi;
- Talaba;

Talaba adabiyotlarni ko'radi va yuklab oladi:

```
namespace adabiyotlarMyBMI.Talaba{
public partial class Adabiyotlar : System.Web.UI.Page {
SqlConnection cnn;
int fan1, fan;
protected void Page_Load(object sender, EventArgs e) {
fan = int.Parse(Request.QueryString["fan"]);
cnn = new SqlConnection(ConfigurationManager.ConnectionStrings["uquv-
2ConnectionString"].ConnectionString);
```

```

if (!IsPostBack) {
    filtr(); }
private void filtr() {
    cnn.Open();
    SqlCommand cmd = new SqlCommand() {
        Connection = cnn,
        CommandText = "select fan from mut_fanlari where id=" + fan,
        CommandType = CommandType.Text };
    SqlDataReader str = cmd.ExecuteReader();
    if (str.HasRows) {
        str.Read();
        fan1 = int.Parse(str["fan"].ToString()); }
    str.Close();
    cnn.Close();
    SqlDataSource1.SelectCommand = "SELECT [id], [tili], [turi], (select nomi from
    adab_turi where adabiyotlar.turi=id) as turi_nomi, [nomi], [fan_id], [faylName],
    [uqituvchi], [avtor] FROM [adabiyotlar] where fan_id=" + fan1;
    SqlDataSource1.DataBind();
    GridView1.DataBind();
    cnn.Open();
    cmd.CommandText = "select fan_nomi from fanlar where id=" + fan1;
    SqlDataReader str1 = cmd.ExecuteReader();
    if (str1.HasRows) {
        if(str1.Read())
        Label1.Text = str1["fan_nomi"].ToString() + " fanidan adabiyotlar."; }
    str1.Close();
    cnn.Close(); }
protected void ImageButton1_Click(object sender, ImageClickEventArgs e) {
    using (GridViewRow row = (GridViewRow)
    ((ImageButton)sender).Parent.Parent) {

```

```

Label lblFayl = (row.Cells[6].FindControl("faylName") as Label);
string fileName = lblFayl.Text;
string sssfile = fileName;
string fileExtension = sssfile.Substring(sssfile.LastIndexOf('.'));
// Set Response.ContentType
Response.ContentType = GetContentType(fileExtension);
// Append header
Response.AppendHeader("Content-Disposition", "attachment; filename=" +
fileName);
// Write the file to the Response
Response.TransmitFile(Server.MapPath("~/kitoblar/" + fileName));
Response.End(); } }
private string GetContentType(string fileExtension){
if (string.IsNullOrEmpty(fileExtension))
return string.Empty;
string contentType = string.Empty;
switch (fileExtension) {
case ".htm":
case ".html":
contentType = "text/HTML";
break;
case ".txt":
contentType = "text/plain";
break;
case ".doc":
case ".rtf":
case ".docx":
contentType = "Application/msword";
break;
case ".xls":

```

```

case ".xlsx":
contentType = "Application/x-msexcel";
break;
case ".jpg":
case ".jpeg":
contentType = "image/jpeg";
break;
case ".gif":
contentType = "image/GIF";
break;
case ".pdf":
contentType = "application/pdf";
break; }
return contentType; } }

```

2.3. O'quv adabiyotlar monitoringi modulini yaratishni amalga oshirish

Dastur ASP texnologiyasi orqali yaratildi. Asosiy sahifaning ko'rinishi quyidagi kodda aniqlangan:

```

<%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true"
CodeBehind="addAdabiyot.aspx.cs"
Inherits="adabiyotlarMyBMI.Uqituvchi.addAdabiyot" %>
<asp:Content ID="Content1" ContentPlaceHolderID="HeadContent"
runat="server">
<style type="text/css">.style1 {
width: 190px; } .style2 {
width: 222px; } .style3 {
width: 73px;
font-weight: bold; } .style4 {
font-size: x-large;

```

[illegible]

```

ConnectionString="<%%$ ConnectionStrings:uquv-2ConnectionString %>"
SelectCommand="SELECT [id], [til] FROM
[adab_tili]"></asp:SqlDataSource></td>
    <td colspan="2">
        <strong>Turi</strong>&nbsp;
        <asp:DropDownList ID="DropDownTuri" runat="server" Height="25px"
Width="257px"      DataSourceID="SqlDataSource2" DataTextField="nomi"
DataValueField="id"></asp:DropDownList>
<asp:SqlDataSource ID="SqlDataSource2" runat="server"
    ConnectionString="<%%$ ConnectionStrings:uquv-2ConnectionString %>"
SelectCommand="SELECT [id], [nomi] FROM
[adab_turi]"></asp:SqlDataSource></td>
</tr><tr>
    <td valign="middle" class="style3">Fayl </td>
    <td colspan="2"><asp:FileUpload ID="FileUpload1" runat="server" /></td>
    <td align="right" class="style1">
<asp:Button ID="Button1" runat="server" Text="Saqlash" Style="cursor: pointer;
border-radius: 7px;
    height: 30px; width: 120px;" onclick="Button1_Click" />
</td></tr></table></div></div></asp:Content>

<%@ Page Title="Fanlar" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true"
CodeBehind="Default.aspx.cs"
Inherits="adabiyotlarMyBMI.Talaba.Default" %>

<asp:Content ID="Content1" ContentPlaceHolderID="HeadContent"
runat="server">
    </asp:Content>

```

```

        <asp:Content ID="Content2" ContentPlaceHolderID="MainContent"
runat="server">
            <asp:Label ID="Label1" runat="server" Text="Fanlar" Font-
Size="Medium"></asp:Label><br />
<br />
<asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
BackColor="White"
        BorderColor="#3366CC" BorderStyle="None" BorderWidth="1px"
CellPadding="4" DataKeyNames="id"
        DataSourceID="SqlDataSource1" EmptyDataText="Ma'lumot yo'q"
ShowHeaderWhenEmpty="True">
    <Columns>
        <asp:TemplateField HeaderText="id" SortExpression="id" Visible="false">
        <ItemTemplate>
            <asp:Label ID="id" runat="server" Text='<%=# Bind("id") %>'></asp:Label>
        </ItemTemplate>
        </asp:TemplateField>
        <asp:TemplateField HeaderText="fio" SortExpression="fio" Visible="false">
        <ItemTemplate>
            <asp:Label ID="fio" runat="server" Text='<%=# Bind("fio") %>'></asp:Label>
        </ItemTemplate>
        </asp:TemplateField>
        <asp:BoundField DataField="guruh" HeaderText="guruh"
SortExpression="guruh" Visible="false" />
        <asp:BoundField DataField="fan" HeaderText="fan" SortExpression="fan"
/>
        <asp:BoundField DataField="fan_nomi" HeaderText="fan_nomi"
SortExpression="fan_nomi" />
        <asp:BoundField DataField="semestr" HeaderText="semestr"
SortExpression="semestr"

```

```

Visible="false" />
<asp:TemplateField>
<ItemTemplate>
<asp:LinkButton ID="LinkButtonAdabiyot" runat="server"
OnClick="LinkButtonAdabiyot_Click">Adabiyotlar</asp:LinkButton>
</ItemTemplate>
</asp:TemplateField>
</Columns>
<FooterStyle BackColor="#99CCCC" ForeColor="#003399" />
<HeaderStyle BackColor="#003399" Font-Bold="True"
ForeColor="#CCCCFF" />
<PagerStyle BackColor="#99CCCC" ForeColor="#003399"
HorizontalAlign="Left" />
<RowStyle BackColor="White" ForeColor="#003399" />
<SelectedRowStyle BackColor="#009999" Font-Bold="True"
ForeColor="#CCFF99" />
<SortedAscendingCellStyle BackColor="#EDF6F6" />
<SortedAscendingHeaderStyle BackColor="#0D4AC4" />
<SortedDescendingCellStyle BackColor="#D6DFDF" />
<SortedDescendingHeaderStyle BackColor="#002876" />
</asp:GridView>
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
ConnectionString="<%%$ ConnectionStrings:uquv-2ConnectionString %>"
SelectCommand="SELECT DISTINCT talaba.id, talaba.fam + ' ' + talaba.ismi + '
' + talaba.otasi as fio, guruh_fan_domlo.guruh, guruh_fan_domlo.fan,
fanlar.fan_nomi, mut_fanlari.semestr FROM guruh_fan_domlo INNER JOIN
mut_fanlari ON guruh_fan_domlo.fan = mut_fanlari.id INNER JOIN fanlar ON
mut_fanlari.fan = fanlar.id INNER JOIN guruhlar ON guruh_fan_domlo.guruh =
guruhlar.id INNER JOIN talaba ON guruhlar.id = talaba.guruh WHERE (talaba.id
= @Param1) AND (mut_fanlari.semestr = @Param2)">

```

```
<SelectParameters>  
<asp:SessionParameter Name="Param1" SessionField="tal" />  
<asp:Parameter DefaultValue="10" Name="Param2" />  
</SelectParameters>  
</asp:SqlDataSource>  
</asp:Content>
```

Bu bobda asosan dasturning yaratilishi haqidagi ma'lumotlar o'rin egallagan. Birinchi bo'limda o'quv adabiyotlar monitoringi yaratilishida e'tibor berilishi kerak bo'ladigan ma'lumotlar keltirilgan. Ikkinchi bo'limda esa dasturning loyihasi keltiriladi. Uchinchi bo'limda ER modeli haqidagi ma'lumotlar berilgan. Keyingi bo'limda dastur interfeysi haqidagi ma'lumotlar keltirilgan. Unda adabiyotlarni qo'shish va yuklab olish sahifasi va qanday ishlar ketma-ketligini bajarish ko'rsatib o'tilgan.

Xotima

Ushbu bitiruv malakaviy ishining mavzusi "Universitet informatsion tizimida fan dasturlardagi o'quv adabiyotlar monitoringi modulini yaratish" deb nomlanib, u kirish qismi, II bob, xotima va foydalanilgan adabiyotlardan iborat.

I bob o'quv adabiyotlar monitoringini yaratishda asosiy kerak bo'ladigan texnologiyalar haqidagi bilimlarni o'z ichiga olgan. Birinchi bo'limda html tilining asosiy teg(tag)lari haqidagi ma'lumotlar keltirilgan. Teglarga izohlar berilgan va ular misollar yordamida ham yoritilgan. Ikkinchi bo'limda esa PHP texnologiyasi haqidagi ma'lumotlar keltirilgan. PHP texnologiyasida sahifa tuzilishi, funksiyalarning yozilishi, o'zgaruvchilar bilan ishlash borasidagi ma'lumotlar keltirilgan va ular misollar yordamida tushuntirilgan. Uchinchi bobda esa SQL tili haqidagi ma'lumotlar keltirilgan. Jadval yaratish, yaratilgan jadvallarni ma'lumotlar bilan to'ldirish, ortiqcha ma'lumotlarni o'chirish. Bobda dastur yaratilishida kerak bo'ladigan barcha texnologiyalarning asosiy qismlari keltirilgan.

II bobdan asosan dasturning yaratilishi haqidagi ma'lumotlar o'rin egallagan. Birinchi bo'limda o'quv adabiyotlar monitoringi yaratilishida e'tibor berilishi kerak bo'ladigan ma'lumotlar keltirilgan. Ikkinchi bo'limda esa dasturning loyihasi keltiriladi. Uchinchi bo'limda ER modeli ko'rsatib o'tilgan. Keyingi bo'limda dastur interfeysi haqidagi ma'lumotlar keltirilgan. Unda o'qituvchi sahifasi va talaba sahifasi haqida tushunchalar berilgan.

Hozirgi davrda daqiqa sayin o'zgarib borayotgan texnika – texnologiyalar, ilm – fan tizimi, qisqa qilib aytganda jamiki borliqda vaqtdan unumli foydalanish ehtiyoji tobora oshib bormoqda. Ushbu bitiruv malakaviy ishi ham ma'lum miqdorda bu ehtiyojni qondira oladi deb o'ylayman.

Foydalanilgan adabiyotlar

1. I.A.Karimov “Yuksak ma'naviyat yengilmas kuch”. Toshkent, 2008, 176 b.
2. M.Aripov, B.Begalov va boshqalar “Axborot texnologiyalari” Noshir Toshkent 2009

3. Кухарчик А. “PHP обучение на примерах” – Мн.: Новое знание, 2004 237 с.
4. Sayfiyev J.F. “SQL tiligakirish” uslubiy qo’llanma. Buxoro 2005 yil. 76 bet.
5. Ицки Бен – Ган Microsoft SQL Server 2008. Основы T-SQL: Пер. с англ. – СПб.: БХВ – Петербург 2009. -432 с.
6. Zakirova T.A., Ibragimov E.U., Masharipov A.K., Musaeva M.A. “Webdasturlash”. O’quv qo’llanma – Toshkent 2010 yil. 184 bet.
7. Мальчук Е.В. HTML и CSS. Самоучитель – М.: Издательский дом «Вильямс», 2008. – 416 с.: ил.
8. Мержевич В.В. HTML и CSS на примерах. – СПб.: БХВ-Петербург, 2005. – 448 с.: ил.
9. Гудман, Дэнни, Моррисон, Майкл. JavaScript. Библия пользователя, 5-е издание.: Пер с англ. – М.: ООО «И.Д. Вильямс», 2006. – 1184 с.: ил. – Парал. тит. англ.
10. Кристина Пейтон, Андре Миллер. PHP 5 & MySQL 5. Москва «Бином», 2007.

Internet manzillar:

www.ziyonet.uz

www.wikipedia.org