

**O`ZBEKISTON RESPUBLIKASI ALOQA, AXBOROTLASHTIRISH
VA TELEKOMMUNIKATSIYA TEXNOLOGIYALARI DAVLAT
KO`MITASI**

**TOSHKENT AXBOROT TEXNOLOGIYALARI UNIVERSITETI
NUKUS FILIALI**

Axborot texnologiyalari kafedrası

**Komp`yuter injiniringi fakul`tetining
Informatika va axborot texnologiyalari yo`nalishining
talabasi Yuldashev Abatning**

BITIRUV MALAKAVIY ISHI

***Mavzusi:* “Avtosalonlarning ish jarayonlarini avtomatlashtirish”**

**Ilmiy rahbar:_____ «O`zbektelekom» AK
Qoraqalpog`iston filiali
etakchi injineri Artiqbaeva.,
ass. Besinbaev J.**

Kafedra mudiri:_____ t.f.n. Arzimbetov T.Z.

MUNDARIJA

KIRISH.....	6
1. MA'LUMOTLAR BAZASINI TASHKIL ETISH ASOSLARI	
1.1. Ma'lumotlar bazasini tashkil qilish prinsiplari	8
1.2. Ma'lumotlar bazasi arxitekturasini.....	11
1.3. Relyatsion ma'lumotlar bazasi texnologiyasi.....	14
1.4. Masalani qo'yilishi.....	17
2. INSTRUMENTAL VOSITALAR VA ULARNI TANLASH	
2.1. Ma'lumotlar bazasini yaratish bosqichlari.....	19
2.2. Ma'lumotlar bazasini yaratish.....	22
2.3. Microsoft Access ma'lumotlar bazasini boshqarish tizimi.....	30
2.4. C# dasturi va uning imkoniyatlari.....	37
3. «AVTOSALONLARNING» ISH JARAYONINI AVTOMATLASHTIRISH TIZIMINI YARATISH	
3.1. «Avtosalonlarning» ish jarayonini avtomatlashtirish tizimini yaratish va strukturasini ishlab chiqish.....	47
3.2. Foydalanuvchi uchun qo'llanma.....	49
XULOSA.....	53
FOYDALANGAN ADABIYOTLAR.....	55
ILOVA.....	57

KIRISH

O'zbekiston Respublikasi Prezidenti Islom Karimovning 2011 yilning asosiy yakunlari mamlakatimizni ijtimoiy-iqtisodiy rivojlantirish yakunlari va 2012 yilda O'zbekistonni ijtimoiy-iqtisodiy rivojlantirishning ustuvor yo'nalishlariga bag'ishlangan O'zbekiston Respublikasi Vazirlar Mahkamasining majlisidagi "2012 yil Vatanimiz taraqqiyotini yangi bosqichga ko'taradigan yil bo'ladi" mavzusidagi ma'ruzasida yurtboshimiz tomonidan ilgari surilgan o'ta muhim va dolzarb fikrlar, g'oyalar, amaliy taklif va tavsiyalarni hamda mamlakatimizni ijtimoiy-iqtisodiy rivojlantirish borasida 2012 yil uchun belgilab berilgan ustuvor yo'nalishlar, maqsad-vazifalarni talaba yoshlar o'rtasida keng targ'ib etish, ularning ijtimoiy-iqtisodiy, siyosiy-huquqiy ongu tafakkurini yuksaltirish, tasavvur doirasini kengaytirishga qaratilgan. Shuningdek, yurtimizda olib borilayotgan tub ijtimoiy-iqtisodiy islohotlar jarayonlariga qiziqadigan keng kitobxonlar foydalanishi mumkin.

Informatsion texnologiyalarning rivojlanishi va axborot oqimlarining tobora ortib borishi, ma'lumotlarning tez o'zgarishi kabi xolatlar insoniyatni bu ma'lumotlarni o'z vaqtida qayta ishlash choralarini qidirib topishga undaydi. Ma'lumotlarni saqlash, uzatish va qayta ishlash uchun ma'lumotlar bazasi (MB) ni yaratish, so'ngra undan keng foydalanish bugungi kunda dolzarb bo'lib qolmoqda.

Darxaqiqat, xozirgi kunda inson xayotida MBda kerakli axborotlarni saqlash va undan oqilona foydalanish juda muxim rol o'ynaydi. Sababi: jamiyat taraqqiyotining qaysi jabxasiga nazar solmaylik o'zimizga kerakli ma'lumotlarni olish uchun, albatta, MBga murojaat qilishga majbur bo'lamiz. Demak, MBni tashkil qilish axborot almashuv texnologiyasining eng dolzarb xal qilinadigan muammolaridan biriga aylanib borayotgani davr taqozasidir

Axborot tizimlari va texnologiyalari yildan-yilga inson faoliyatining turli soxalarida keng qo'llanimoqda. Ularni yaratish, ishga tushirish va xayotga keng tadbiq etishdan maqsad – jamiyat va inson butun xayot faoliyatini axborotlashtirish borasidagi muammolarini xal etishdir.

Axborot resurslarini oqilona tashkil etish va foydalanish jarayonida ular mexnat, moddiy va energetik resurslar ekvivalenti sifatida namoyon bo'ladi. Ayni paytda axborot boshqa barcha resurslardan oqilona va samarali foydalanish xamda ularni asrab-avaylashga ko'maklashuvchi yagona resurs turidir.

Qaralayotgan bitiruv malakaviy ishi Avtosalonlarning ish jarayonini avtomatlashtirish mavzusida bulib, u kirish bo`limida, uch bobdan, xulosa bo`limidan va foydalanilgan adabiyotlar ro`yhatidan iborat.

Ushbu bitiruv malakaviy ishini ishlab chiqishda «Avtosalonlarning» ish jarayonini avtomatlashtirish tizimi va MBning dasturi yaratildi. Dasturni yaratishda hozirgi zamon talablariga javob bera oladigan C# dastur tilidan foydalanildi va dastur uchun ishlatilgan Microsoft Access ma'lumotlar bazasini boshqarish tizimida ma'lumotlar bazasi ishlab chiqildi.

Bitiruv malakaviy ishining birinsh bobida quydagi masalalar ko'rip shiqidi:

- MBning ta'rifi, umumiy tushunchalari va tashkiliy qismlari keltirildi;
- Ma'lumotlar bazasi arxitekturasini va tushinchasi yoritildi;
- MBni tashkil etuvchi relyatsion ma'lumotlar bazasi texnologiyasidagi makrosalar orqali makrobuyruqlar tashkil qilindi;
- Batafsil yuqoridagilar taxlil qilinib masalani qo'yilishi keltirildi.

Ikkinshi bobida dasturni yaratishda kerak bo'ladigan va tanlab olingan instrumental vositalar haqida, ularni qulayligi va ishlash jarayoni misollar tariqasida keltirildi.

Ushichi bobda Avtosalonlarning ish jarayonlarini avtomatlashtirish tizimini loyixalashning ish rejasi belgilanib MB strukturasi keltirildi. Shuningdek korxona axborot-ma'lumot bazasini dasturiy vositalari va ta'minoti ishlab chiqildi. Foydalanish uchun batafsil bulgan yo'riqnoma keltirildi. Dasturiy taminotga qo'yiladigan apparat, ya'ni texnik hamda dasturiy ta'minotiga qo'yiladigan talablar ko'rsatildi.

Ishning xulosa bo'limida olingan natijalar so'z etiladi.

1. MA'LUMOTLAR BAZASINI TASHKIL ETISH ASOSLARI

1.1. Ma'lumotlar bazasini tashkil qilish prinsiplari

Ma'lumotlar bazasi – bu o'zaro bog'langan va tartiblangan ma'lumotlar majmuasi bo'lib, u ko'rilayotgan ob'ektlarning xususiyatini, xolatini va ob'ektlar o'rtasidagi munosabatni ma'lum soxada tavsiflaydi.

MBni yaratishda ikkita muxim shartni hisobga olmoq zarur:

Birinchidan, ma'lumotlar turi, ko'rinishi, ularni qo'llaydigan dasturlarga bog'lik bo'lmasligi lozim, ya'ni **MB**ga yangi ma'lumotlarni kiritganda yoki ma'lumotlar turini o'zgartirganda, dasturlarni o'zgartirish talab etilmasligi lozim.

Ikkinchidan, **MB**dagi kerakli ma'lumotni bilish yoki izlash uchun biror dastur tuzishga xojat qolmasin.

Shuning uchun xam **MB**ni tashkil etishda ma'lum qonun va qoidalarga amal qilish lozim. Bundan buyon **axborot** so'zini **ma'lumot** so'zidan farklaymiz, ya'ni **axborot** so'zini umumiy tushuncha sifatida qabul qilib, **ma'lumot** deganda aniq bir belgilangan narsa yoki xodisa sifatlarini nazarda tutamiz.

Masalan, biror Oliy o'quv yurtining aniq fakultetida taxsil olayotgan biror gurux talabalari to'g'risidagi ma'lumotlar bitilgan quyidagi jadvalni ko'raylik:

1.1. Jadval. Talabalar haqida ma'lumot

Familiyasi	Ismi	Tug'ilgan sanasi	Guruxi	Turar joyi	Qiziqqan Fani
Bozorov	Alisher	2.05.1990	214-08	S-4-5	Matem.
G'ulomov	Farxod	2.12.1992	213-08	I.Sino,1	Adabiyot
Akmalov	San'at	3.6.1990	217-08	S-3-9	Tarix

Bu misolda **3ta yozuv** bo'lib, ularning xar biri **6ta maydondan** iborat. Mazkur maydonlarning har biri mos ravishda «**Familiyasi**», «**Ismi**», «**Tug'ilgan sanasi**», «**Guruxi**», «**Turar joyi**» va «**Qiziqqan fani**» deb nomlangan. Demak, **yozuvdagi maydonlar soni yozuvga** kiritiladigan ma'lumotlar xajmiga bog'lik. Fayldagi bu **yozuvar birlamchi** hisoblanadi. Chunki biror **yozuvdagi** ixtiyoriy

ma'lumotni boshqa **yozuv**dagi ma'lumotlar bilan taqqoslab aniqlash mumkin emas. Shuning uchun ham bizga kerakli bo'ladigan ikkilamchi yozuvlarni esa faqat amaliy dasturlar yordamida olish mumkin bo'ladi. Modomiki shunday ekan, **MB** tashkil qilish, ularga qo'shimcha ma'lumotlarni kiritish va mavjud **MB**dan foydalanish uchun maxsus **MB**lar bilan ishlaydigan **programmalar** zarur bo'ladi. Bunday dasturlar majmui **ma'lumotlar bazasini boshqarish tizim (MBBT)** deb yuritiladi. Aniqroq qilib aytganda, **MBBT**—bu ko'plab foydalanuvchilar tomonidan **MB**ni yaratish, unga qo'shimcha ma'lumotlarni kiritish va **MB**ni birgalikda ishlatish uchun zarur bo'lgan dasturlar majmuidir. **MBBT**ning tarkibida asosiy komponenti—bu **ma'lumotlar** bo'lsa, boshqa komponenti—**foydalanuvchilar, Hardware- texnik va Software**-dasturiy ta'minoti hisoblanadi. **Hardware** tashki qo'shimcha xotiradan (disk, magnit lentasi) iborat bo'lsa, dastur qismi esa **MB** bilan foydalanuvchi o'rtasidagi muloqotni tashkil qilishni amalga oshiradi. **MB**ning tuzilishi o'rganilayotgan obuektning ma'lumotlari ko'rinishi, ma'nosi, tuzilishi va xajmiga bog'lik bo'ladi.

Odatda, foydalanuvchilar quyidagi kategoriyalarga bo'linadilar:

- Foydalanuvchi - dastur tuzuvchi,
- tizimli dastur tuzuvchi,
- ma'lumotlar bazasi administratori.

Bunda dastur tuzgan **foydalanuvchi MBBT** uchun yozgan dasturiga javob beradi, **tizimli dastur tuzuvchi** esa butun sistemaning ishlashi uchun javobgar hisoblanadi. U xolda **MB administratori** sistemaning saqlanish xolatiga va ishonchliligiga javob beradi.

MBBT quyidagicha tavsiflanadi:

- **Ispolnimost`**-Bajarilishlik, foydalanuvchi so'roviga hozirjavoblik bilan muloqotga kirishish
- **Minimalnaya povtoryaemost`** - Minimal takrorlanishlik

MBdagi ma'lumot iloji boricha kam takrorlanishi lozim, aks xolda ma'lumotlarni izlash susayadi.

- **Yaxlitlik** –axborotni **MBda** saqlash iloji boricha ma'lumotlar orasidagi bog'liqlikni asragan xolda bo'lgani, ayni muddao.
- **Bezopasnost`**–Xavfsizlik **MB** ruxsat berilmagan kirishdan ishonchli ximoya qilingan bo'lishi lozim. Faqat foydalanuvchi va tegishli tashkilotgina ma'lumotlarga kira olish va foydalanish xuquqiga egalik qilishi mumkin.
- **Migratsiya**–ba'zi bir ma'lumotlar foydalanuvchilar tomonidan tez ishlatilib turiladi, boshqalari esa faqat talab asosida ishlatiladi. Shuning uchun ma'lumotlarni tashki xotiralarda joylashtiriladi va uni shunday tashkil qilish kerakki, eng ko'p ishlatiladigan ma'lumotlarga murojaat qilish qulay bo'lsin.

Ma'lumotlar bazasini boshqarish tizimida xar bir **MB modeli** quyidagi xususiyatlari bo'yicha tavsiflanadi:

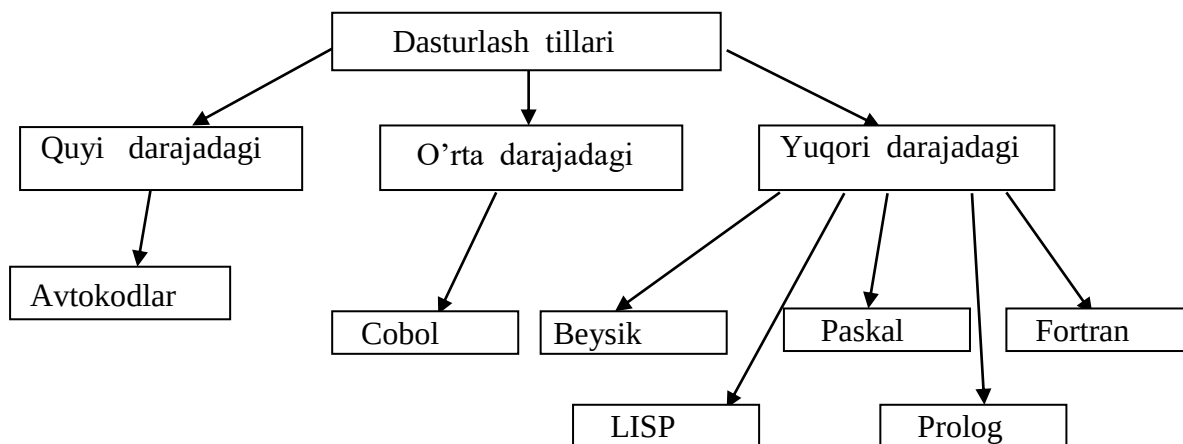
1. Ma'lumotlar tuzilmalarining turi.
2. Ma'lumotlar ustida bajariladigan amallar.
3. Butunlikning cheklanganligi.

Bu xususiyatlarni e'tiborga olgan xolda ma'lumotlar bazasi modellari quyidagi turlarga bo'linadi:

- Daraxtsimon (ierarxik) modellar.
- Tarmoqli (turli) modellar.
- Relyatsion modellar.

Yana shu narsani ta'kidlash lozimki, ma'lumotlar bazasi modellarining faqatgina yuqorida qayd qilingan modeli mavjud deyish, noto'g'ri. Chunki, bulardan tashqari yana ma'lumotlar bazasining binar munosabatlar modeli, ER – modellari, semantik model kabi boshqa turlari ham mavjud. Lekin amalda asosan dastlabki ta'kidlangan 3 turli modellar ko'prok qo'llanilib kelinmoqda. Shuning uchun ham biz ushbu modellarga qisqacha to'xtalib o'tamiz.

Daraxtsimon (ierarxik) modelda obuektlar yozuvlar ko'rinishida ifodalanadi. Masalan,



1.1-rasm. Ierarxik model

Ierarxik modelda ikki yarusdagi elementlar bog'langan bo'lsa, unday ma'lumotlar **tarmoqli (turli)** modelda ifodalangan deyiladi. **Tarmoqli** modellarda ham obuektlar **daraxtsimon** modellardagi kabi yozuvlar ko'rinishida tasvirlanadi. Obuektlarning o'zaro aloqalari **yozu**vlar o'rtasidagi aloqalar sifatida tavsiflanadi.

Relyatsion modellarda esa obuektlar va ularning o'zaro aloqalari ikki o'lchovli jadval ko'rinishida tasvirlanadi. Ma'lumotlarning bunday ko'rinishda tasvirlanishi obuektlarning o'zaro aloqalarini yaqqol tasvirlanishiga asos bo'ldi.^[3]

1.2. Ma'lumotlar bazasi arxitekturas

MBBT aloxida olingan modullardan tashkil topgan:

- **MB boshqarish bloki** disklardagi ma'lumotlar bilan foydalanuvchi dasturi va tizimning so'rovi (**guery**) orasidagi interfeysni aniqlaydi.
- **Fayl menedjeri**-ma'lumotlar tuzilmasi bilan disklar o'rtasidagi bog'lanishni boshqaradi.
- **Guery-protssessor** ingliz tilida yozilgan **guery** gaplarini **MBni** boshqarish bloki tushunadigan tilga o'tkazadi.

- **Prekompilyator DML**(Data Manipulation Language) **ma'lumotlar** bilan manipulyatsiya qiladigan til bo'lib, u quyidagi operatsiyalarga javob beradi:
 1. **MB**dan ma'lumotlarni ajratib olish,
 2. **MB**ga ma'lumotlarni kiritish,
 3. **MB**dan ma'lumotlarni olib tashlash,
 4. **MB**ni modifikatsiya (o'zgartirishlar) qilish.
- **Kompilyator DDL**(Data Definition Language)-**MB** tilini, uning tuzilmasini va tashqi xotiralardagi axborot turini aniqlaydi. **MB**ning tuzilmasi ko'pincha jadval shaklida bo'ladi.

Shuni takidlash lozimki, xozirgi vaqtda deyarli barcha **MBBT**lar asosan relyatsion modellar asosida tashkil qilinmoqda. Shuni nazarda tutgan **Microsoft Office** korporatsiyasi ham eng ommalashgan dastur vositalarga ega, bu dastur vositalari ixtiyoriy soxada yuqori darajadagi professional xujjatlar tayyorlash imkonini beradi. Shulardan biri **MB**lar bilan ishlashga mo'ljallangan **Microsoft Access** dasturi bo'lib, bu programma **Visual Basic for Application** dasturlash muxitida **makros**lar yaratish va boshqa bir qancha imkoniyatlarga egaki, bu foydalanuvchiga har tomonlama mukammal bo'lgan xujjatlar tayyorlashga yordam beradi.^[4]

Microsoft Office ning o'zbek tilidagi varianti yo'qligi, uning faqat **ingliz** va **rus** tilida yaratilgan versiyalaridangina foydalanish imkoniyatiga ega ekanligimizdan mazkur fikrlar **Microsoft Access** ning ruscha versiyasiga tayanib yozilgan. **Microsoft Access** dasturi xam **relyatsion modellar** asosiga qurilgan bo'lib, unda tashkil qilinadigan **MB**lar jadval ko'rinishida aks etadi. Bunday jadvaldagi ustunlar **maydon** deb, satrlar esa **yozuv** deb ataladi.

Maydon – ma'lumotlarni tashkil etishning oddiy birligi bo'lib, ma'lumotning aloxida, bo'linmas birligiga egaligi rekvizitga mos keladi.

Yozuv–mantikiy bog'langan rekvizitlarga mos keluvchi maydonlar yig'indisidir. YOzuvning tuzilishi o'z tarkibiga mos har bir oddiy ma'lumotga ega maydonlar tarkibi va ketma-ketligi bilan belgilanadi.

Demak, **maydon MB**ning asosiy tuzilmali elementi bo'lib quyidagi parametrlar bilan ifodalanadi:

- **uzunligi** (belgi simvollarda ifodalanib baytlarda o'lchanadi),
- **nomi** (maydonning o'ziga hos aloxida xususiyati),
- **podpis** - imzo (ustun sarlavxasi xaqida ma'lumot).

Maydonlar xususiyatiga va tarkibiga qarab quyidagi turlarga bo'linadi:

1. Matnli maydon.

2. Sonli maydon.

3. Vakt va sanani ifodalovchi maydon.

4. Mantiqiy maydon ($1 \cap 0$; Ha yoki yo'k; rost yoki yolg'on kabi mantiqiy birliklar bilan ifodalanadi).

5. Pul birliklarida ifodalangan maydon (raqamlar pul birliklari bilan birgalikda ifodalanadi)

6. OLE maydoni (shakl, tasvir, rasm, musiqiy kliplar va videoyozuvlar shaklida ifodalanadi)

MEMO maydoni- matn uzunligi **256** simvoldan uzun bo'lgan maydonda faqat matnning qaerdaligini ifodalovchi ko'rsatqich turadi. Bu holda har bir maydonda **65 535** simvol saqlanishi mumkin.

Schetchik (sanovchi) **maydoni** - maydonda turgan ifoda avtomatik ravishda sanalib o'zgaradi.

Endi, keng foydalanuvchilar ommasi uchun mo'ljallangan va eng qulay bo'lgan **relyatsion MB**ni tashkil qilish xaqida biroz to'xtalib o'tamiz.

1.3. Relyatsion ma'lumotlar bazasi texnologiyasi

Agarda **MB**da ishtirok etadigan jadvallar bir-biri bilan bog'langan bo'lsa, bunday **MB**ni **relyatsion MB** deb atash qabul qilingan. Bunda jadvallarni bir-biri bilan bog'lash uchun umumiy xususiyatga ega bo'lgan **unikal maydon** tushunchasi kiritilgan. Ushbu tushuncha ba'zan **MB**ning **kalitli** maydoni deb xam ataladi. Jadvalning bir-biri bilan bog'lanish strukturasi **bog'lanish sxemasi** deyiladi. **MB** doimo o'zgarib turadi: unga yangi **yozuvlar**, borlariga esa yangi

elementlar qo'sqiladi. Relyatsion ma'lumotlar bazasi quyidagi parametrlar bilan baholanadi:

1. **Prostota** -Soddalik
2. **Gibkost`** -Moslanuvchanlik
3. **Tochnost`** - Aniqlilik matematik aniq usullar bilan **MB** manipulyatsiya qilinadi,
4. **Sekretnost`** -Maxfiylik
5. **Svyazannost`** -Bog'liqlik
6. **Nezavisimost`** -Ma'lumotlar bog'liqligi yo'qligi,
7. Ma'lumotlar bilan manipulyatsiya qilish tili.

Ba'zan, **MB** ishlatilishi samaradorligini oshirish maqsadida uning tuzilishi xam o'zgartirilib turiladi. Bu xolda **MB**ning **ierarxik va tarmoqli** modellari vujudga keladi. **MB**ni tashkil qilish, uni to'ldirish, nusxasini olish kabi vazifalarni bajarish uchun maxsus dastur ta'minoti bo'lish lozim. Bunday dastur ta'minoti **MBBT** (yuqorida qayd qilganimizdek) deyiladi. Mazkur tizimlar bir vaqtning o'zida bir necha foydalanuvchiga xizmat ko'rsata oladi, ya'ni ma'lumotlardan bir vaqtda bir necha kishining foydalanishi mumkin. Bunday **MBBT**larni tashkil qilishda yuqori darajadagi dasturlash tillari: **Clipper, Paradox, FoxPro** lar mavjud. Bunday **MBBT**lardan **Windows** muxitida ishlash imkoniyatiga ega **Microsoft Works 3.0**, yangi texnologiya asosida ishlay oladigan «**klient – server**»-**SQL Windows Solo** kabilarni keltirish mumkin. Ammo, bu tillarda ishlab chiqilgan **MBBT** juda qimmat bo'lgani uchun **Microsoft** firmasi **Microsoft Offise** tarkibida (kichik va o'rta biznes xodimlari uchun juda qulay bo'lgan va birmuncha arzon) **Microsoft Access (Access 2.0 va Access-9x)** ni ishlab chiqib amaliyotga tadbiq qildi. **Access** dasturi **Visual Basic** dasturlash muxitida ijro qilingan. **Access** ning yana bir qo'shimcha qulayligi shundaki, bu dastur **Microsoft Excel 9x, Word 9x** va boshqa programmalar bilan **integratsiyalangan**. Shuning uchun xam u yoki bu dasturdagi ma'lumotlarni **import** yoki **eksport qilish** imkoni mavjud.^[6]

Munosabatlar yordamida qurilgan ma'lumotlar bazasi yassi (ikki o'lchovli) ma'lumotlar elementlarining to'plamidan quriladi.

Munosabat yoki jadval - bu kartejlar to'plami. Agar kartejlar n -o'lchovli bo'lsa, ya'ni agar jadval n ta ustunga ega bo'lsa, munosabat n -darajali munosabat deyiladi, 2-darajali munosabat binarli, 3-darajali - ternarli, n - darajali - n -arli munosabat deyiladi.

Bir turdagi ma'lumotlar elementlarining qiymatlari to'plami ya'ni jadvalning bir ustuni domen deyiladi. j raqamli ustun j -li munosabat domeni deyiladi.

Matematikada R (Relation) berilgan n -ta ko'p $S_1, S_2, S_3, \dots, S_n$ (shart emas xar xil bo'lishi) munosabatlar bilan aniqlanadi, agar u kartejlar to'plamini taqdim etsa, shunda xar bir kartejning birinchi elementi S_1 dagi, ikkinchisi S_2 dagi va hokazo.

Bunday munosabatlarni tasvirlash va ular ustida operatsiya qilishda aniq matematik belgilar mavjud, munosabatlar algebrasiga yoki hisoblab chiqiladigan munosabatlarga asoslangan. Ma'lumotlar relyatsion asoslarning ayrim afzallik (ustunlik)larini sanab o'tamiz:

Oddiyliги. Ko'pchilik ma'lumotlar tuzilishini taqdim etishda ikki o'lchovli jadvallardan foydalanish uncha tayyor bo'lmagan yoki tajribasiz foydalanuvchining ma'lumotlar asoslari bilan ishlashda - eng oddiy usullardan biri.

Ihchamligi. Proeksiyalash va bog'lash operatsiyalari munosabatlarni kesmoq va yopishtirmoqqa yo'l qo'yadiki, unda amaliy dasturlashtiruvchilar har hil fayllarni kerakli formada olishi mumkin.

Muayyanligi. Ko'zlangan maqsad bog'lanishlari asoslarida odatdagi xodisa bo'lishi mumkin qadar tushiriladi. Munosabatlar o'z tabiatiga ko'ra muayyan ma'noga ega bo'ladilar va matematik muayyan usullar bilan manipulyatsiya qilish, shunday vositalarni qo'llash: munosabatlar algebrasi va hisoblab chiqiladigan munosabatlarga asoslangan.

Maxfiyligi. Maxfiylik nazorati soddalashtiriladi. Xar bir munosabat uchun kirish imkoniyati xaqililigi beriladi. O'ziga xos maxfiylik ma'nosida ko'rsatqichlarning kirish imkoniyatini tekshirish xuquqi talab qilinadi; agar kirish imkoniyati xuquki buzilmagan bo'lsa, ko'rsatqich amaliy dasturlarni va fayllarni

kiritish, ishlov berish qiymatlari kuchli o'sib ketadi; amaliy dasturlar soni o'sishi bilan, ularni kiritish xarajatlari xam juda katta bo'lib ketadi.

Bog'langanligi. Relyatsion tasavvurlar turli munosabatlar va fayllarning atributlarini o'zaro aloqadorligi to'g'risida aniq ko'rinish beradi.

Oddiy boshqarilishi. Ikki o'lchovli jadvallarni fizikaviy joylashtirish mumkin sodda bo'ladi, boshqa daraxt ko'rinishli va tarmoqli tuzilmalarga ko'ra. Xotiraning fizikaviy tashkillashtirishning yangi vositalarini ixtiro qilish natijasida joylashtirishning yangi optimal imkoniyatlari paydo bo'lmoqda.

Ma'lumotlar mustaqilligi. Qoidaga ko'ra, asoslar tuzilmasi (tuzilishi) o'sish imkoniyatiga ruxsat bermog'i kerak, ya'ni yangi atributlar va munosabatlarning ko'sqilishi. Ma'lumotlardan foydalanish usullari xam o'zgaruvchan. Yangi kortejlar qo'sqilishi va eskilari chiqarib yuborilishi mumkin. Xuddi shu narsa ma'lumotlar elementiga xam tegishli. Ma'lumotlar bazasini normallashtirilgan formada tashkil etishda mustaqil programmali ta'minoti bilan ma'lumotlarni restruktrizatsiya qilish amaliy programmalarini o'zgartirilishini talab qilmaydi. Bu ayniqsa muxim, agarda ma'lumotlar asoslari o'sishi bilan ma'lumotlarning bo'linmas elementi bo'ladi.

1.4. Masalani qo'yilishi

«Avtosalon» dagi ish jarayonini avtomatlashtirishda masalani qo'yilishi quydagilardan iborat:

- ma'lumotlar bazasini tashkil qilish prinsiplari o'rganish;
- ma'lumotlar bazasini yaratishda Microsoft Access texnologiyasi bilan tanishish;
- C# dasturlash tili va uning imkoniyatlari bilan tanishish;
- «Avtosalon» ish jarayonini avtomatlashtirish tizimini yaratish va strukturasini ishlab chiqish;
- Foydalanuvchi va dasturchiga qulay bo'lgan qo'llanma tuzish;
- Dastur ishlashi uchun dasturiy va texnik ta'minotlarga qo'yiladigan talablarini ishlab chiqish;

Bu rejalashtirilgan ishlarni amalga oshirishda quydagi asosiy yo'nalishlarga e'tibor qaratishimiz zarur dir:

- Avtomashinalar to'g'risidagi keng va batafsil ma'lumotlarni aks ettirish;
- Avtomashinalar va oldi-sotti to'g'risidagi ma'lumotni rezume sifatida bosmadan chiqarish imkoniyatiga ega bo'lish;
- Dasturdan foydalanishda qulay interfeysga ega bo'lishligi;
- MBni ma'lumotlar to'ldirishi yoki tuzatish qulayligi;
- Dastur modullari aloqasining ta'minlanganligi;
- Dasturda turli xil parametrlar bo'yicha qidiruv tashkil etilishi;
- Kerakli ko'rinishdagi hisobotlarni tayyorlash imkoniyati borligi;
- MBni taxrirlash, qo'shish, o'chirish va chop etish imkoniyatlari mavjudligi;

«Avtosalon» ish jarayonini avtomatlashtirish tizimi – ishlab chiqarishdagi xizmatchilarni asosan zamonaviy axborot texnologiyalardan foydalanib mexnat unumdorligini oshirish va vaqtni tejashni ta'minlashdan iborat dir.

Tizimni asosiy afzalik tomoni:

olingan axborotni ishlov berishda avtomatlashtirish;
avtomashinalar haqidagi ma'lumotlarni markazlashtirib saqlash va shu bilan birga ularni yo'qolib ketishiga yo'l qo'ymaslik va ulardan foydalanish imkoniyatini qulayligini mumkin qadar oshirishni ta'minlash;

tezkor kerakli axborotni qidirish;

ma'lumotlarni taxlil qilish imkoniyati, ya'niy tizimni oldi-sotdi ishlarini hisob-kitobini olishga imkoniyat mavjudligi.

foydalanuvchiga qulay interfeys yaratish. Bu jarayon qidiruv filtrlar yordamida amalga oshirilib, bunda parametrlarni shunday sozlash mumkin bo'ladiki, kerakli axborotni topish uchun ortiqcha ma'lumotarni chiqarib tashlash imkonini beradi. Topilgan ma'lumotlar esa hisobot tariqasida taqdim etiladi.

2. INSTRUMENTAL VOSITALAR VA ULARNI TANLASH

2.1. Ma'lumotlar bazasini yaratish bosqichlari

Ma'lumotlar bazasini yaratish ob'ekt sohasini o'rganishdan boshlanadi. Va natijada to'planadigan va foydalanuvchiga taqdim etiladigan axborotning quyidagi xususiyatlariga alohida e'tibor berildi:

1. To'laqonlik. Axborot to'laqonligi ob'ekt faoliyatining u yoki bu tomonlarining miqdoriy va sifat parametrlarini aniq belgilash hamda mos qarorlarni ishlab chiqarishda ifodalaniladi. Axborotning noto'laqonligi qarorlar qabul qilishda xatolarga olib kelishi mumkin.

2. Ishonchlilik qabul qilinadigan qarorlar samaradorligi saqlanadigan yetib kelgan va natijaviy axborotlarda muayyan darajada buzilishlarga yo'l qo'yadi.

3. Axborotni qabul qilishning **bemalolli**gi vaqt birligida ma'lumotlarni qabul qilish tezligi bilan belgiladi. Shu bois ham ma'lumotlar ko'proq jadval shaklida beriladi, u nafaqat axborot mazmunini ochib beradi, balki yengil qabul qilinadi ham.

4. Ma'lumotlarning **dolzarbli**gi muayyan vaqt mobaynida aniq vazifani amalga oshirish uchun yaroqligini ifodalaydi. Shu bois ham dolzarblik, hozirjavoblik va tezkorlik axborotga xos xususiyatlardir.

5. Kechikmaslik axborotning qulay yoki belgilangan vaqtda kelib tushishini anglatadi. Bu talabni buzish axborotni qadrsizlantiradi.

6. Aniqlik uning to'g'riligini anglatadi. Axborotning aniqligi uning barcha iste'molchilar tomonidan bir xil qabul qilinishini ta'minlaydi.

7. Tezkorlik vaqt o'tgach axborot eskirishi va dolzarbligini yo'qo-tishini aks ettiradi.

Axborotning o'z vaqtida qabul qilinmasligi qaror qabul qilishni kechiktiradi, oqibatda qabul qilinayotgan qarorlar o'zgaruvchan sharoitda talabga javob bermaydi. Axborot qanchalik tezkor bo'lsa, u shunchalik qimmatli bo'ladi.

Axborotning qimmati aniqlik darajasi oshgani sayin yoki xabar qilinayotgan va aniq natijalar o'rtasidagi farq kamayganda tez ko'tariladi. To'liqroq va ishonchli axborot to'g'ri qarorni qabul qilishni ta'minlaydi.

Ma'lumotlar bazasiga kiritiladigan ma'lumotlarni aniqlangandan keyin bu ma'lumotlar bazasini tashkil etishda foydalaniladigan ma'lumotlar bazasini boshqarish tizimini (MBBT) tanlash lozim.

MBBT o'z tasniflanishining muhim belgilaridan biri bo'lgan ma'lumotlar modeli turlaridan birini (tarmoqli, ierarxik yoki relyatsion) ta'minlaydi. MBBT ma'lumotlar bazalarining ko'p maqsadli tavsifini, ma'lumotlarni himoyalash va qayta tiklashni amalga oshiradi. Rivojlangan muloqot vositalari va yuqori darajali talablar tilining mavjudligi MBBTni oxirgi foydalanuvchi uchun oson vositaga aylantiradi.

Xususan MBBT – bu foydalanuvchining aslahaviy qobig'i hisoblanadi. MBBT tarkibida dasturlash tilining mavjudligi aniq masalalarni va aniq foydalanuvchiga mo'ljallangan ma'lumotlarga ishlov berishning murakkab tizimlarini yaratish imkonini beradi.

Ma'lumotlar bazalarini loyihalashtirishda quyidagi tavsiflarini solishtirish va tahlil qilishga asoslangan MBBTni asoslab tanlab olish muhim vazifa hisoblanadi:

- dasturiy texnik bazasi (ShK turi va modeli, hisoblash vositalari konfiguratsiyasiga qo'yiladigan talablar, OT versiyasi);
- ma'lumotlar bazalarining turlari (amaliy, predmet, lokal, integrellashgan, taqsimlangan);
- foydalanuvchilarning malakalari (MBBT bilan ishlash uchun maxsus tayyorgarlikka ega bo'lmagan foydalanuvchi, mutaxassisligi dasturchi bo'lmagan darajadagi MB bilan ishlashga tayyorgarligi bo'lgan predmet sohasining mutaxassisi – foydalanuvchi, amaliy dasturchi, MB larning administratori);
- ma'lumotlar bazalari bilan foydalanuvchilarning muloqat qilish vositalari (dasturlash tillarini o'z ichiga oluvchi ma'lumotlar ustida ish olib borish va tasvirlash tili);

- ma'lumotlarni qayta ishlash rejimi (paketli, interaktiv, tarmoqli);
- ma'lumotlarni mantiqiy va fizik mustaqilligi;
- ma'lumotlar bazalari axborot strukturalarining asosiy xossalari (mantiqiy strukturasi-MBBT vositalari orqali amal qiladigan va tashkiliy tuzilmasini o'zgartirmasdan uni modifikatsiya qilish imkoniyati, ma'lumotlar turini kengaytirgan holda qarshiliksiz ishlov berishda);
- havfsizlik darajasini ta'minlash va ma'lumotlarning to'laqligi;
- xizmat ko'rsatishning standart vositalari mavjudligi (ma'lumotlar bazalarini kuzatishni doimiy dasturiy modullarini va ma'lumotlar lug'atini, ma'lumotlar bazalarini yengillashtiradigan (yukini tushiradigan), qayta tashkil etadigan va qayta strukturalashtiradigan, tiklaydigan jurnalni yuritish, kiritish va hisobotlar generatorlari va h.k.);
- ishlatish tavsiflari (loyihachilar haqida, sarmoya egalari haqida, moddiy-texnik ta'minotga bo'lgan talablar, tarqatish shakli).

MBBTning asosiy vositalari quyidagilar:

- ◆ ma'lumotlar bazalari tuzilmalariga topshiriq berish (tasvirlash) vositalari;
- ◆ ma'lumotlarni kiritish, ko'rish va muloqotlar rejimida ishlashga mo'ljallangan ekran shakllarini loyihalash vositalari;
- ◆ berilgan sharoitlarda ma'lumotlarni tanlash uchun talablar yaratish, shuningdek, ularni ishlash bo'yicha operatsiyalar bajarish vositalari;
- ◆ foydalanuvchiga qulay ko'rinishda ishlov natijalarini bosmaga chiqarish uchun ma'lumotlar bazasidan hisobot yaratish vositalari;
- ◆ til vositalari – makroslar, qurilgan algoritmik til (Dbase, Visual Basic, Microsoft Access yoki boshalar), talablar tili (QBE- Query Example, SQL, C#) va h.k. Ular ma'lumotlarni ishlashning nostandart algoritmlarini, shuningdek foydalanuvchi topshiriqlaridagi voqealarni ishlash protseduralarini bajarish uchun qo'llaniladi .^[7]

2.2. Ma'lumotlar bazasini yaratish.

Ushbu malakaviy bitiruv ishini ishlab chiqishda avvalam bor mavjud bo'lgan ma'lumotlar bazasini yaratish usullarini ko'rib chiqamiz va kerakli amallarni belgilab olamiz.

Ma'lumotlar modeli – o'zaro bog'langan ma'lumotlar tuzilishi va ular ustida bajariladigan operatsiyalar majmuidir. Model shakli va unda foydalaniladigan ma'lumotlar tuzilishining turi MBBT (ma'lumotlar bazalarini boshqarish tizimi) yordamchi model yoki ma'lumotlarni ishlashning amaliy dasturi yaratiladigan dasturlash tilida qo'llanuvchi ma'lumotlarni tashkil etish va ishlash konsepsiyasini aks ettiradi.

Fayllar modeli ma'lumotlarining asosiy tuzilmalari(strukturalari) – maydon, yozuv, fayl hisoblanadi. Yozuv ma'lumotlarini ishlashning asosiy tuzilma birligi operativ va tashqi xotira o'rtasidagi almashuv birligi hisoblanadi.

Maydon – ma'lumotlarni tashkil etishning oddiy birligi bo'lib, axborotning alohida, bo'linmas birligi rekvizitga mos keladi.

Yozuv – mantiqan bog'langan rekvizitlarga mos keluvchi maydonlar yig'indisidir. Yozuvning tuzilishi o'z tarkibiga kiruvchi har bir oddiy ma'lumotga ega maydonlar tarkibi va ketma-ketligi bilan belgilanadi.

Fayl – alohida maydonlarda mazmunga ega bo'lgan bir xil tuzilishdagi ko'plab yozuv nusxalaridir. Yozuv nusxasi maydonlarning konkret mazmunini o'z ichiga olgan yozuvlarni aks ettiradi. Fayl yozuvi tuzilishida maydon yagona mazmunga ega va guruhli ma'lumotlar mavjud emas (quyida qarang: ierarxik va tarmoqli modellar haqidagi bo'lim) bo'ladi. Har bir yozuv nusxasida yagona yozuv kaliti bir xil bo'ladi. Umumiy hollarda yozuv kalitlari ikki xil ko'rinishda bo'ladi: **dastlabki (birlamchi)** va **ikkilamchi** kalit.

Dastlabki kalit (DK) – yozuvni ma'no jihatidan bir xillashtiruvchi bir yoki bir necha maydonlar. Agar dastlabki kalit bir maydondan iborat bo'lsa u **oddiy** deyiladi, agar bir necha maydonli bo'lsa – turli **tarkibli** kalit hisoblanadi.

Ikkilamchi kalit (IK) – dastlabkidandan farqli o'laroq, shunday maydonki, uning mazmuni faylning bir necha yozuvlaridan takrorlanadi, ya'ni u yagona

bo'lmaydi. Agar dastlabki kalitning mazmuni bo'yicha faqat bitta yozuv nusxasi topilsa, ikkinchi kalit bo'yicha bir necha nusxa topilishi mumkin.

Ma'lumotlarning sanab o'tilgan tuzilishi bir qator MBBTda qo'llaniladi. Bu esa ushbu tushunchani ma'lum ma'noda universallashtiradi.

Indeksatsiyalash. Kalit bilan fayl yozuvlariga kirishning samarali vositasi indeksatsiyalashdir. Indeksatsiyalashda indeksli qo'shimcha fayl yaratiladi. U ma'lumotlar fayli kalitining barcha mazmunini tartiblashtirib o'zida saqlaydi. Indeksli faylda har bir kalit mazmuni uchun ma'lumotlar faylining tegishli yozuviga mo'ljallangan ko'rsatkich bo'ladi. Hajmi asosiy faylda kichik indeksli fayl mavjud bo'lganda berilgan kalit bo'yicha qidirilayotgan yozuv tez topiladi. Ma'lumotlar faylida yozuv ko'rsatkichi yordamida ushbu yozuvga bevosita yo'l ochiladi. Indeksatsiyalash faqat dastlabki emas, balki ikkilamchi kalit bo'yicha ham amalga oshirilishi mumkin.

Fayllar modeli ma'lumotlarini mantiqiy tashkil etishni tasvirlash. Ma'lumotlarni mantiqiy tashkil etishni tasvirlashda har bir faylga yagona nom beriladi va uning yozuvlar tuzilmasi tasvirlanadi. YOzuvlar tuzilmalarini tasvirlash undagi maydonlar va ularning yozuv ichidagi joylashuv tartibini o'z ichiga oladi. Har bir maydon uchun qisqartma ko'rsatkich fayl nomi (yozuv ichidagi maydon identifikatori), maydon hajmi saqlanayotgan ma'lumot turi, maydon uzunligi va raqamli ma'lumotlarning aniqligi belgilab olinadi. Yozuvning yagona dastlabki kaliti vazifasini o'tovchi maydonlar uchun kalit belgisi ko'rsatiladi. 1–jadvalda «Yetkazib berish» fayli yozuvining tasvirlanishi misol qilib ko'rsatilgan. Bu yerda yozuvning dastlabki kiliti turli tarkiblidir, chunki rekvizit belgilar NPS, KTOV, DATP yig'indisi muayyan yetkazib berishni aniqlaydi. Tovarlarni yetkazib berishning miqdoriy tavsifi rekvizit asoslar KOLT, ST bilan tasvirlanadi .

Ma'lumotlar bilan amal bajarishning ilk tizimlari axborot ishlashning an'anaviy usullariga asoslanib tuzilgan edi. Har bir muayyan holat uchun tashqi foydalanuvchining o'z mantiqi ishlab chiqiladi. U axborot tuzilmasi, tanlash operatsiyasi, axborotni qo'shish va yo'q qilish kabi tushunchalarni o'z ichiga oladi. Ma'lumotlar va dastur o'rtasidagi o'zaro bog'liqlik yuzaga keladi:

ma'lumotlarni o'zgartirishda, yo dasto'zni almashtirish yoki ma'lumotlarni qaytadan tuzish zaruriyati yuzaga keladi.

Murakkab axborotlarni ishlab chiquvchilar duch kelgan bu va boshqa qiyinchiliklar ma'lumotlar bilan amal bajarish uchun tizimlarga nisbatan standart talablarning shakllanishiga olib keldi. Asosiy talablardan biri – ma'lumotlarning iloji boricha mustaqil yoki axborot tuzilmasini fizik tushunchalardan alohida qilish kerak. Bunda hamma ma'lumotlar ko'p foydalanuvchilar kirishi mumkin bo'lgan holda ba'zi standart ichki tuzilishli qilib saqlanadi.^[8]

Ma'lumotlar bazasi – axborotlar tizimlarining eng muhim tarkibiy qismi hisoblanadi. Oxirgi foydalanuvchi va ma'lumotlar bazasi administratorining ishini yengillashtirish uchun MBBT yaratilgan edi. Bu tizimlar ma'lumotlar bazasini amaliy dasturlardan ajratadi. MBBT dastur va apparat vositalarining murakkab kompleksi bo'lib, foydalanuvchi shu tufayli faqat ma'lumotlar bazasini mantiqiy tashkil etishnigina tasavvur qiladi. Ma'lumotlar bazasini mantiqiy tashkil etish uni fizik amalga oshirishdan (ya'ni tashkil etish va fayllarni ishlashdan) sezilarli farq qilishi mumkin. Foydalanuvchilarning ixtiyorida talablar tili bo'lib ular yordamida foydalanuvchilar ma'lumotlarni tanlashi va o'zgartirishi mumkin.

Yuqorida ishonch hosil qilganimizdek, bu xotiralarning qo'shimcha sarflanish, tanlash va xizmat ko'rsatish vaqtining ko'payishini talab qiladi. Bundan tashqari, tuzilmalarni qayta tashkil etish xato qilish xususiyatiga ega bo'lgan dasturlar bilan bajariladi. Tizimli dasturlar foydalanuvchilarga oson bo'lmaganligi sababli, bunday xatolarni, ma'lumotlarni tasdiqlash operatsiyalari orqaligina aniqlash mumkin. Bu turli xil axborot tuzilmalarini samarali ta'minlaydigan tizimlarni loyihalashni qiyinlashtiradi. Bu qiyinchiliklarni yengish uchun zamonaviy MBBTlar quyidagi talablar javob berishi lozim:

- ◆ ma'lumotlarning mustaqilligi;
- ◆ talablarning kuchli tili;
- ◆ javobning qisqa vaqti;

◆ ma'lumotlar va kataloglarni qayta tashkil etishni qisqartirish yoki ulardan voz kechish.

Ma'lumotlarning mustaqilligi – MBBTga asosiy talab, talablarning kuchli tili esa foydalanuvchining talablarini qondirishning muhim shartidir. Bu tillar assotsiativ manzillashtirish va ma'lumotlar to'plami bilan amallar bajarish vositalariga ega. Bu esa o'z navbatida EHMLardan samarali foydalanishga sharoit yaratadi.

Ma'lumotlar bazasini boshqarish tizimi. Bu quyidagilarga imkon beruvchi dasturiy vositalarning to'plamidir.

A) foydalanuvchilarni ma'lumotlarni aniqlash va amallar bajarish tili vositalari bilan ta'minlaydi. Bunday vositalarga ma'lumotlarni aniqlash tili (MAT) va ma'lumotlar bilan amallar bajarish(MABT) kiradi. Ma'lumotlar tili atamasi aytib o'tilganlarining har ikkalasini yoki ulardan birini anglatadi. Ma'lumotlar so'zi ma'lumotlar tilini SQQ, Paskal va h.k. kabi tillar turidan farqlaydi. Lekin ma'lumotlar tili universal tilga, masalan SQQ, Paskalga kiritilishi mumkin. Bunday holda dasturlashning universal tili va ma'lumotlar tili tegishli ravishda (kirituvchi) til va ma'lumotlarning til osti deb ataladi.

B) foydalanuvchi ma'lumotlarining modelini qo'llab-quvvatlashni ta'minlaydi. Ma'lumotlar modeli – ba'zi ilovaga tegishli fizik ma'lumotlarning mantiqiy taqdim etilishini aniqlash vositasidir.

V) aniqlash, yaratish va mantiqiy ma'lumotlar bilan amallar bajarish (ya'ni tanlash, yangilash, kiritish va yo'q qilish)ga imkon beruvchi MAT va MABT funksiyalarini amalga oshiruvchi dasturlarni ta'minlaydi.

G) ma'lumotlarning himoyasi va yaxlitligini ta'minlaydi. Tizimdan foydalanish faqat shunga huquqi bo'lgan foydalanuvchigagina ruxsat etiladi. Foydalanuvchilar ma'lumotlar ustida operatsiya bajarayotganlarida saqlanayotgan ma'lumotlarning muvofiqligi (yaxlitligi) ta'minlanadi. Gap shundaki, MBBT ko'plab foydalanuvchilar ishi jamoa rejimiga asosan tuziladi.

Relyatsion model (inglizcha relation – munosabat) o'tgan asrning 70- yillari boshida Kodd tomonidan yaratilgan. Modelning soddaligi va egiluvchanligi

ma'lumotlar bazalari tuzuvchilarini e'tiborini o'ziga tortdi. 80-yillarga kelib keng taraqqiy eta boshladi va relyatsion MBBTlar sanoat standarti darajasiga ko'tarildi.

Model relyatsion algebrasi tushunchalari tizimiga tayanadi. Bu tushunchalarning eng muhimlari jadval, satr, ustun, munosabat va birlamchi kalitlar sanaladi, bajariladigan amallar esa jadvallar bilan ish olib borishda aks etadi.

Relyatsion modellarda axborot to'g'ri to'rtburchakli jadvallar ko'rinishida aks ettiriladi. Har bir jadval ustun va satrlardan tashkil topgan bo'ladi va ma'lumotlar bazalari ichida o'zining takrorlanmas nomiga ega.

Jadval real olam axborotini – mohiyatini aks ettiradi, uning har bir satri (yozuvi) esa ob'ektning aniq bir nusxasini – nusxa mohiyatini aks ettiradi. Jadvalning har bir ustuni ushbu jadvalda o'ziga xos nomga ega. Jadval kamida bir ustunga ega bo'lishi kerak.

Ma'lumotlarning relyatsion modeli yuqorida ko'rib chiqilgan tarmoqli va ierarxiklardan foydalanuvchi uchun qulay bo'lgan jadvalli tasavvurlar va ma'lumotlarga kirishning oddiy tuzilishi bilan farq qiladi. Ma'lumotlarning relyatsion modeli oddiy ikki o'lchamli jadval – ***munosabat (model ob'ektlari)***larning yig'indisidir. Relyatsion modeldagi relyatsion bog'langan ikki jadvallar orasidagi mantiqiy bog'langan aloqalar jadval munosabatlariga tegishli bir xil atributlarning mazmun jihatidan tengligiga ko'ra o'rnatiladi.

Jadval-munosabat relyatsion modellarning universal ob'ekti hisoblanadi. Bu relyatsion modelni ta'minlovchi turli MBBTlardagi ma'lumotlarni birxillashtirish imkonini beradi. Relyatsion modellarni ishlash operatsiyalari munosabatlar algebrasi va relyatsion hisob-kitoblarning universal apparatidan foydalanishga asoslangan.

Jadval relyatsion model ma'lumotlari (ob'ekti)ning asosiy turi hisoblanadi. Jadvalning tuzilishi ustunlarning yig'indisi bilan belgilanadi. Jadvalning har bir satrida tegishli ustunga mos keluvchi bittadan mazmun joylashgan bo'ladi. Jadvalda ikkita bir xil satr bo'lishi mumkin emas. Satrlarning umumiy soni chegaralanmagan.

Ustun ma'lumotlarining ba'zi tarkibiy qismi – **atributga** mos keladi. Atribut ma'lumotlarning eng oddiy tuzilmasidir. Jadvalda yuqorida ko'rib o'tilgan tarmoqli va ierarxik modellardagi kabi ko'p tarkibiy qismlari gu-ruh yoki takrorlanuvchi guruh kabilar belgilanishi mumkin emas. Jadvalning har bir ustini ma'lumotlar tegishli tarkibiy qismi (atribut)ning nomiga ega bo'lishi kerak. Ma'nosi jadval satriga teng bo'lgan bir yoki bir nechta atributlar jadvalning **kaliti** hisoblanadi.

Ma'lumotlar bazasini tuzishda relyatsion yondashuv munosabatlar nazariyasining terminologiyasi qo'llaniladi. Eng oddiy ikki o'lchamli jadval munosabat sifatida belgilanadi. Tegishli atribut mazmuniga ega bo'lgan jadval ustuni **domen** deyiladi. Turli atributlar mazmuniga ega bo'lgan satrlar esa **kortej** deb ataladi.

Relyatsion jadval-munosabat. R relyatsion jadval munosabatining ko'rinishi berilgan. R munosabat (relyatsion jadval)ning formal ta'rifi uning domenlari D_i (ustunlari), kortejlari K_i (satrlari) haqidagi tushunchaga tayanadi. Ko'plab domenlar $\{D_i\}$ belgilangan R munosabat deb, $D_1 * D_2 * D_3 * \dots * D_n$ domenlarini **dekart (bevosita) ishlab chiqaruvchi ko'plikka** aytiladi.

Jadval-munosabat ma'lumotlar tarkibiy qismi atributlar ($A_1, A_2, \dots$) nomiga ega bo'lgan ustunlarni o'z ichiga olgan d atributlarning mazmuni jadvalning asosiy qismida joylashgan bo'lib satrlar va ustunlarni tashkil qiladi. Bir ustunda atributlar mazmunining ko'pligi **domen D_i ni** hosil qiladi. Bir satrda atributlar mazmunining ko'pligi bir **kortej K_j ni** hosil qiladi. R munosabat ko'plab tartibga solingan kortejlar orqali hosil bo'ladi: $R = \{ K_j \}, j = 1-m \ K_j = \{d_{1j}, d_{2j}, \dots, d_{nj}\}$

n – munosabat domenlarining soni; munosabatlarning ko'lamini belgilaydi.

J – kortej nomeri;

k – munosabatdagi kortejlarning umumiy soni bo'lib, munosabat koordinata soni deyiladi.

Jadval-munosabatning kaliti. Kortejlar jadval-munosabat ichida takrorlanmasligi zarur va ular tegishli yagona identifikator – dastlabki (birlamchi) kalitga ega bo'lishi kerak.

Dastlabki (birlamchi) kalit atributdan tashkil topgan bo'lsa **oddiy**, bir necha atributdan tashkil topganda esa **turli tarkibli** deb ataladi. Munosabatda dastlabki kalitdan tashqari ikkilamchi kalit ham bo'lishi mumkin.

Ikkilamchi kalit – mazmuni turli satr – kortejlarda takrorlanishi mumkin bo'lgan kalitdir. Ular bo'yicha ikkinchi kalitning bir xil mazmuni satrlar guruhi izlab topiladi.

Satrlar ustunlardan farqliroq o'z nomlariga ega emas, ularning jadvalda joylashish tartibi aniqlanmagan va satrlar soni mantiqan chegaralanmagan bo'ladi. Satrni tartib raqamiga ko'ra tanlab olib bo'lmaydi. Faylda har bir satr o'z raqamiga ega bo'lsa ham, bu narsa satrni tavsiflamaydi. Bu raqam satr jadvaldan olib tashlanganda o'zgaradi. Mantiqan satrlar o'rtasida “birinchi” va “oxirgi” degan tushuncha yo'q.

2.2.1. jadval. Relyatsion munosabatning tasviri

$R \text{ munosabat} = \{K1, K2, \dots\}$

A1	A2	A3	A4	A5	A6
d11	d21	d31	d41	d51	d61
d12	d22	K2 korteji			d62
d13	d23	d33	d43	d	d63

} Atributlar (ustunlar)
nomi

Kortej
 $K2 = \{d12, d22, d32, d42, d52, d62\}$

....				O	
....				m	
....				e	
....				n	
D1	D2	D3	D4	D5	D6

Domen
 $D5 = \{d51, d52, d53, d54, \dots\}$

R – relyatsion jadval munosabatning tasviri

Relyatsion tizimlarning qo'llanilishi murakkab navigatsiya zaruriyatni bartaraf qildi. Chunki ma'lumotlar endi bir fayl ko'rinishida emas, balki mustaqil naborlar asosida tuzilib, ma'lumotlarni tanlab olish uchun amaliy to'plamlar nazariyasi – relyatsion algebra amallari qo'llaniladi.

Jadvalning har bir satrini bir ma'noli nomlovchilari bo'lgan ustun yoki ustunlar to'plami relyatsion model jadvalida mavjud bo'lishi kerak. Ushbu ustun yoki ustunlar to'plami jadvalning **birlamchi kaliti** deb ataladi.

Agar jadval birlamchi kalitning unikalligi talablarini qanoatlantirsa, u holda **munosabat** mavjud deyiladi. Relyatsion modellarda barcha javdallar munosabatlarga o'zgartirilgan bo'lishi kerak. Relyatsion model munosabatlari biri biri bilan bog'langan bo'ladi. Bog'lanishlar tashqi kalitlar bilan ta'minlanadi.

Tashqi kalit – boshqa munosabatning (jadvalni) birlamchi kaliti qiymatini bir ma'noda tavsiflovchi ustun (ustunlar to'plami). Boshqacha aytganda, tashqi kaliti aniqlangan munosabat ustunlari to'plami birlamchi kalit bo'lib xizmat qiluvchi boshqa mos munosabatga murojaat qilishi demakdir.

Relyatsion jadval (munosabat) sxemasi jadval yozuvini tashkil etuvchi maydonlar nomi to'plamini o'zida aks ettiradi:

JADVAL NOMI (Maydon 1, Maydon 2, ...Maydon n).

Relyatsion modellarning hozirgi zamon MBBTlarida ustunlikka erishishi quyidagi omillar bilan aniqlanadi:

- 1) rivojlangan nazariyaning mavjudligi (relyatsion algebrani);
- 2) ma'lumotlarni boshqa modellarini relyatsion modellarga keltirish apparatining mavjudligi;
- 3) axborotga ruxsatli kirishni tezlashtirishni maxsus vositalarini mavjudligi;

4) tashqi xotirada MB aniq fizik tashkil etish bilimiga ega bo'lmay ular bilan ish olib borish imkoniyatini yaratadigan MBga nisbatan standartlashgan yuqori darajadagi so'rovlar tilining mavjudligi.[9]

2.3. Microsoft Access ma'lumotlar bazasini boshqarish tizimi

Access foydalanuvchilariga ma'lumotlar bazasi bilan ishlash uchun ikkita avtomatizasiya vositasini taqdim etadi: makroslar tili va Visual Basic for application (VBA) tili. Bu vositalar mashaqqatli operatsiyalar ketma-ketligini qaytarilishni, tugmani yoki tugmalar kombinatsiyasini bosish yoki menyu komandasini aktivizasiya qilishdan iborat oddiy proseduraga aylantirishga imkon beradi.

VBA ning imkoniyatlari makroslar tiliga nisbatan beqiyosdir. Bunga qaramasdan makroslar tilidan ma'lumotlar bazasini boshqarish bo'yicha ko'pgina operatsiyalarni avtomatizatsiyalash uchun foydalanish mumkin.

Makroslarni yaratish texnikasi. Makroslar oynasi ma'lumotlar bazasi oynasidagi **Makrosy** qo'shimcha varaqasidagi **Sozdat** (yaratish) yoki **Konstruktor** tugmasini bosishda ochiladi. Oyna to'rtta ustunni o'z ichiga oladi

1. Imya makrosa (Makros nomi)
2. Yslovie (Shart)
3. Makrokomanda (Makrokomanda)
4. Primechanie (Izox)

O'rnatilgan bo'yicha yangi makros yaratilayotganda faqatgina Makrokomanda va Izox ustunlari aks etadi. Qolgan ustunlarni namoyishi makroslar nomi opsiyasi vositasi bo'yicha va **Vid** menyusidan shart bo'yicha o'rnatiladi. Agar oyna bir necha makroslarni o'z ichiga olsa, berish lozim bo'lgan makros nomini **Imya makrosa** ustunida ko'rsatiladi. Makroslar nomlari ko'rsatilayotganda qaytarishlari bo'lmasligi kerak. **Usloviet** ustunida makrosni faqat bir qismi bajarilishi uchun shart kiritish (mantiqiy ifoda) amalga oshiriladi. **Makrokomanda** ustunida bajarish lozim bo'lgan xarakatlar (makrokomandalar) kerakli ketma-ketlikda sanalib chiqiladi. Dasturga sharx saqllovchi **Primechaniya** ustuni makros bajarilganda dastur tomonidan e'tiborga olinmaydi, biroq uni

tuldirish tavsiya etiladi, chunki bunday xolda makros matni tushunarliroq.

Makrosni qiyin bo'lmagan prosesslarni avtomatizasiyalash uchun ishlab chiqarish qulaydir, xususan bir necha forma yoki hisobotlarni ochilishi va yopilishi, bir necha xujjatlarni ekranga yoki bosmaga bosib chiqarish va boshqalar.

Makroslarni yaratilishi va qo'llanilishi o'rgangan holda, makros yordamida ma'lumotlari bazasida bir necha obuektlarni ochilish misolini ko'rib chiqamiz. Aniq operatsiyalarni avtomatizasiyalash uchun mo'ljallangan ma'lumotlar bazasi ko'pgina jadvallar, formalar, so'rovlar va hisobotlardan iborat bo'ladi. Odatda shunday ma'lumotlar bazasida operator ko'p bo'lmagan bir xil miqdordagi obuektlar bilan ishlaydi. Har bir baza bilan ishlash seans boshida qo'shimcha kerakli obuektlarni ochish uchun qo'shimcha vaqt ketadi. Bu jarayonni tezlatishga xarakat qilamiz: kerakli xujjatlarni ochuvchi va aniq tartibda ekranga joylashtiruvchi makros yaratamiz:

- Yangi ma'lumotlar bazasi oynasini oching.

- **Makrosy** qo'shimcha saxifasiga uting va **Sozdat`** tugmasini bosing, buning natijasida makrosni Konstruktor oynasi ochiladi.

- Ekranda xam ma'lumotlar bazasi oynasi, xam makros oynasi aks etishi uchun **Okno** menyusidagi **Sleva naprava** (chapdan o'ngga) buyrug'ini amalga oshiring.

- Ma'lumotlar bazasi oynasida **Formy** qo'shimcha saxifasiga o'ting.

O'qituvchilar bo'yicha soatlar miqdori formasiga belgi qo'ying, sichqoncha yordamida makros oynasiga tashib o'ting va **Makrokomanda** ustunini birinchi yacheykasiga joylashtiring. Maydonchada **Otkryt formu**(formani ochisi) makrokomandasi paydo bo'ladi. Shunday qilib, makrosda Kol-vo chasov po prepodavatelyam (O'qituvchilar bo'yicha soatlar miqdori) formasini ochish operatsiyasi qo'shiladi.

- **Primechaniya** ustunida xuddi usha qatorga shunday matn kiriting:

“ Kol-vo chasov po prepodavatelyam ” formasini oching.

- [Enter] klavishi yordamida makrokomanda ustunining ikkinchi qatoriga

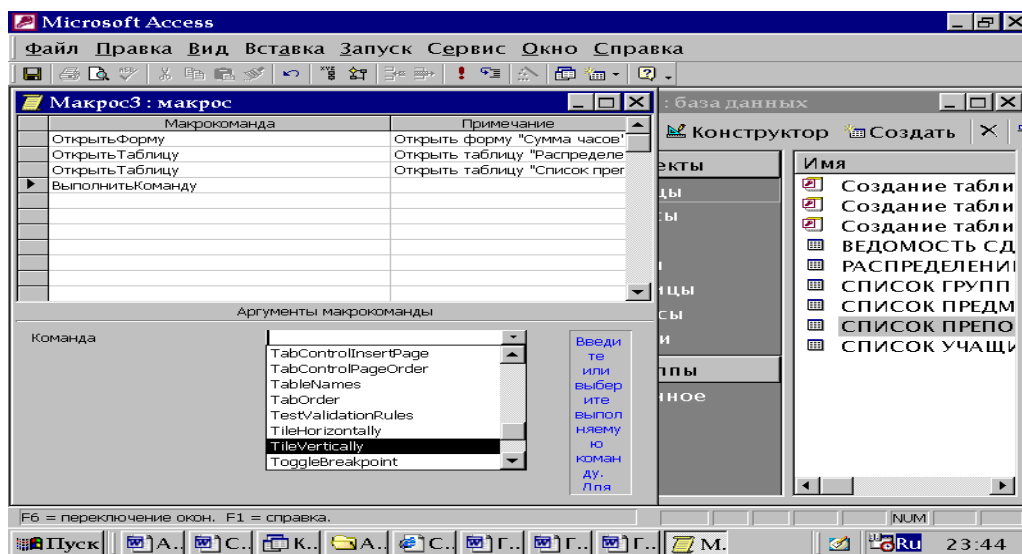
o'ting

- Shu xarakatni o'qilishi kerak bo'lgan hamma obyektlar (misol uchun, raspredelenie predmetov (Darslarni taqsimlash), Spisok prepodavateley (O'qituvchilar ro'yxati) jadvallari) uchun qaytaring.

- Makrokomanda ustunining keyingi bo'sh yacheykasini bosib va kirish mumkin bo'lgan makrokomandalar ro'yxatini oching. **Выполнит команду** (buyruqni bajarish) buyrug'iga belgi qo'ying.

- *Areumenty makrokomandy* soxasida Komanda maydonchasini ishga solamiz va ro'yxatdan Tile Vertically elementini tanlaymiz. Shunday qilib, agar bir necha oyna ochilsa, makrokomanda ekranni bo'lish operatsiyasini uzi ichiga oladi.

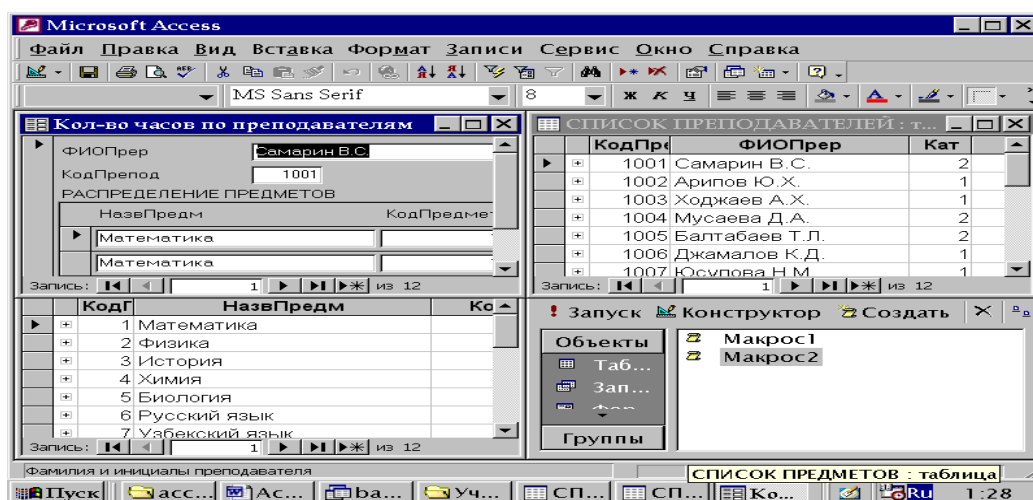
- **Файл** menyusidan **Сохранит`**(saqlash) buyrug'ini chaqiring va makrosni



Razmeshenie na ekrane nomi ostida saqlab qo'ying.

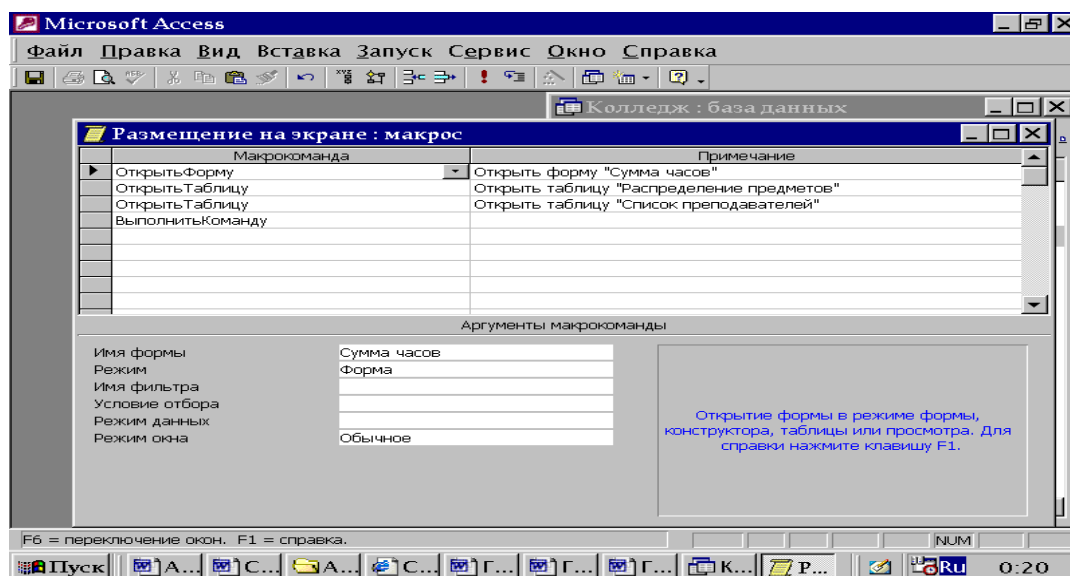
2.3.1-rasm. Makroslar oynasi

Razmeshenie na ekrane(ekranda joylashtirish) makrosini ishga tushirilgandan so'ng hamma kerakli ma'lumotlar bazasining obuektlari yuklanadi va kerakli xolda joylanadi.



2.3.2-rasm. Fanlarni taqsimlash oynasi

Xar bir makrokomandani amalga oshirish uning argumentlarining qiymatiga bog'liq (ba'zi makrokomandalar, misol uchun *Razvernut`* (yoyish) argumentlarga ega emas). Foydalanuvchi makrokomandalarning sintaksisini yodlab olishi kerak emas, barcha argumentlar buning uchun maxsus makros oynasini pastki qismida joylashgan maydonchasiga kiritiladi. Agar ruxsat etilgan qiymatlar to'plami ba'zi argumentlar uchun chegaralangan bo'lsa, Access ularni ro'yxat sifatida



rasmiylashtiradi. Argumentlarni klaviatura yordamida kiritish mumkin, biroq, mavjud bo'lmagan qiymatlarni bermaslik uchun ro'yxatdan tanlagan yaxshirokdir. Argumentlarni makros oynasining pastki qismida qanday joylashgan bo'lsa, shunday berish tavsiya etiladi.

2.3.3-rasm. Makroslarni taqsimlash

Makrosni ishlab chiqishni biz obuektlarni ochish, ya'ni ma'lumotlar bazasining oynasidan mos ravishdagi obuektlarni makrosning Konstruktor oynasi yacheykalarga tashlab o'tish yo'li bilan makrokomandalarni kiritishni boshladik. Shu bilan birga Access qaysi obuekt xaqida gap borayotganini avtomatik o'zi tanib, mos ravishda makrokomandalarni tanlaydi: forma uchun *Otkryt`* (ochish) *formu* ochish yoki jadval uchun *Otkryt` tablitsu* (jadvalni ochish). Tashib o'tilgan obuektning nomi makrokomandaning argumentlar soxasida *Imya* (nom) parametri

qiymati sifatida paydo bo'ladi.^[10]

Makrokomandada obuekt nomi ochilishi argumentlar soxasiga klaviatura orqali kiritish mumkin. Asosiysi makros bajarayotganda ochilayotgan obuekt mavjud bo'lishi kerak, aks xolda dastur xatolik xaqida xabar beradi.

Rejim maydonchasiga kiritilgan Makrokomanda ochilish obuektining keyingi argumenti ekranda aks etish rejimini aniqlaydi. Ushbu argumentning ruxsat etilgan qiymati *Rejim* menyusidagi operatsiyalarga mos keladi.

2.3.1. Jadval. *Rejim* menyusidagi operatsiyalari

Rejim argumenti qiymati	Qo'llanish	Ta'rif
Pechat`	Hisobotni	Hisobotni bosmaga chiqarishni beradi
Forma	Formani	Forma rejimini aktivlashtiradi
Tablitsa	Jadval, so'rov, formani	To'ldirish va o'zgartirish rejimini aktivlashtiradi
Konstruktor	Jadvalning, so'rovning, xisobotning, formaning	Konstruktor rejimini aktivlashtiradi
Prosmotr	Jadvalni, so'rovni, hisobotni, formani	Saxifani ko'rib chiqish rejimini o'rnatadi

Vypolnit` komandu* makrokomanda yordamida Access menyusidagi ko'pgina buyruqni bajarishni topshirish mumkin. Bajarilayotgan buyruqning nomi *Komanda* maydonchasida argument sifatida ko'rsatiladi. Menyu qatorlari tarkibi aktiv obuekt axvoli va tipiga bog'liq. *Vypolnit` komandu makrokomandasidan foydalanilganda, qaysi obuekt oxirgi bo'lib aktivizasiya qilingan va qaysi rejimda joylashganligiga e'tibor berish kerak. Bundan menyu komandasining ochiqligi va bajarilish tug'riligiga bog'liq.

2.3.2. Jadval. Microsoft Access Makrokomandalari

Kategoriya	Vazifasi	Makrokomanda
Forma va xisobotlarda ma'lumotlar bilan ishlash	Ma'lumotlarni tanlash	Primenit fil`tr (ApplyFilter)
	Ma'lumotlar bo'yicha kuchish	Sleduyushaya zapis` (FindNext) Nayti zapis` (FindRecord) Elementu upravleniya(GoToControl) Na stranitsu (GoToPage) Na zapis` (GoToRecord)
	Ma'lumotlarni yoki ekranni yangilash	Obnovlenie (Requery) Pokazat` vse zapisi (ShowAllRecords)
Bajarish	Komandani bajarish	Vypolnit` komandu (Run Comraand)
	Makros, prosedura yoki surovni bajarish	Zapusk makrosa(Run Macro) Zapusk Programmy (Run Code) Otkryt` zapros (Open Query) Zapusk zaprosa SQL (Run SQL)
	Boshqa ilovani bajarish	Zapusk prilojeniya (RunApp)
	Bajarishni uzish	Otmenit sobytie(Cancel Event) Ostanovit` vse Makrosy (Stop All Macros) Ostanovit Makros (Stop Macro)
	Microsoft Access dan chiqish	Vыход (Quit)
Import va eksport	Microsoft Access Obuektlarini boshqa ilovalarga uzatish	Vывести v formate(Output To) Otpравit` Ob`ekt (Send Object)

	Ma'lumotlar formatini uzgartirish	Preobrazovat` Bazu Dанных (Transfer Database) Perenos Bazy Dанных SQL (Transfer SQL Database) Preobrazovat` Elektronnuyu Tablitsu (Transfer Spreadsheet) Preobrazovat` Tekst (Transfer Text)
Obuektlar bilan ishlash	Obuektdan nusxa olish, Obuekt nomini uzgartirish va obuektni saqlash	Kopirovat` Ob`ekt(CopyObject) Kopirovat` Fayl Bazy Dанных (Copy Database File) Pereimenovat` (Rename) Soxranit` (Save)
O'zgalar	Maxsus yoki xos instrumentlar panelini ekranga chiqarish yoki berkitish Tovush signalini berish	Panel` Instrumentov (ShowToolbar) Signal (Veer)

2.4. C# dasturi va uning imkoniyatlari

Dasturlash tillarida buyruqlar va amallar ma'lum kodlar (raqamlar) bilan ifodalangan bo'lib, ular ShK qurilmalari adreslari bilan bevosita ishlashga mo'ljallangan va mashina tili deb ham yuritiladi. Dasturlash tillarida ko'rsatmalar inson tiliga yaqin bo'lgan so'zlar va dastur tuzish uchun juda qulay.

Axborot texnologiyalarining jadal sur'atlar bilan rivojlanishi, Internet ning paydo bo'lishi, yangidan yangi va ma'lum maqsadlarga mo'ljallangan Dasturlash tillari va muhitlari yaratildi va yaratilmoqda. Ularga misol sifatida dBase, KARAT, LISP, FoxPro, Simula, HTML, Java, Java Script, Delphi, Visual Basic, C++, C# kabi tillar va dasturlash muhitlarini keltirish mumkin. Hozirgi kunga kelib ob'ektga yo'naltirilgan va vizual dasturlash texnologiyalari keng tarqalmoqda.

C# universal dasturlash tili bo'lib, xar xil darajadagi masalalar uchun yechim topish mumkin. C# tilining asosiy tushunchalaridan biri bu klasslardir. Klass bu – foydalanuvchi tomonidan yaratilgan (ifodalangan) til. C# tilida C va C++ tillarning deyarli barcha imkoniyatlari saqlangan. C va C++ tillarda tayyor xoldagi dasturlarga qayta o'zgarish kiritganda ham C# kompilyatori dastur matnidan xatoliklar topmaydi. Ya'ni dasturni xar ikkala tilda xam foydalanib tuzish mumkin.

Dasturlashda o'zgaruvchilardan foydalanmasdan dastur ishlash mumkin emas. O'zgaruvchilar o'zida qiymatlarni saqlaydi, bu qiymatlar vaqtinchalik operativ xotirada saqlanib turadi. O'zgaruvchilarning nomlari bo'ladi, nomlar lotin alifbosining **a** dan **z** gacha bo'lgan oraliqdagi harflar bilan belgilanadi. Bundan tashqari o'zgaruvchi nomlarini faqat bitta harf emas bir necha harflar ketma-ketligi bilan ya'ni so'zlar bilan belgilash mumkin.

Sonlar bilan harflar orqali belgilash mumkin, lekin harflardan oldin son kelishi mumkin emas. O'zgaruvchining asosiy turlari sonli o'zgaruvchi, satriy o'zgaruvchi, simvolli o'zgaruvchi, mantiqiy o'zgaruvchilar bo'lib bo'linadi. Sonli o'zgaruvchilarga faqat sonlar kiradi, 0, 5, -10, -5.6, 4995 va h. Satriy o'zgaruvchilarga esa satrlar kiradi. Simvolli o'zgaruvchilar qiymati faqat bitta simvoldan iborat bo'ladi. Mantiqiy o'zgaruvchilarga faqat ikkita qiymat beriladi rost, yolg'on ya'ni true, false..^[11]

C# tilida butun sonli o'zgaruvchilarda 8 ta tip ishlatiladi, ular quyidagi jadvalda berilgan.

2.4.1.jadval. C# tilida butun sonli o'zgaruvchilar

Tip nomi	Baytlar soni	Qiymatlar diapazoni
sbyte	1	-129 dan 128 gacha
byte	1	0 dan 255 gacha
short	2	-32,768 dan 32,767 gacha
ushort	2	0 dan 65,535 gacha
int	4	-2,147,483,647 dan 2,147,483,647

		gacha
uint	5	0 dan 4,294,967,295 gacha
long	8	-9,223,371,036,854,775,808 dan 9,223,372,036,854,775,807 gacha
ulong	8	0 dan 18,446,744,073,709,551,615 gacha

Quyida esa C# tilida kasr sonli o'zgaruvchi tiplari keltirilgan.

2.3.2. jadval C# tilida kasr sonli o'zgaruvchi tiplari

Tip nomi	Baytlar soni	Qiymatlar diapazoni
float	4	$\pm 1.5 \cdot 10^{-45}$ dan $\pm 3,4 \cdot 10^{38}$ gacha
double	8	$\pm 5 \cdot 10^{-324}$ dan $\pm 1.7 \cdot 10^{308}$ gacha
decimal	12	$\pm 1 \cdot 10^{-10}$ dan $\pm 7.9 \cdot 10^{28}$ gacha

float tipidagi o'zgaruvchiga qiymat berganda qiymat oxiriga "f" belgisini qo'yish, decimal tipidagi o'zgaruvchiga qiymat berganda "m" belgisini qo'yish lozim.

Birinchi ifoda bu o'zgaruvchini e'lon qilish. Bu yerda count o'zgaruvchi nomi, int(integer) tipi va count o'zgaruvchisiga 1 soni qiymat qilib olinadi. int tipi da faqat butun sonlardan foydalaniladi. Keyingi qatorda " " belgilari orasiga olingan satr shu holiday chop qilinadi, count esa o'zgaruvchining qiymatini chop qiladi. Uchinchisi esa "Satr" so'zini chop qiladi. Kod yozganda chekinishlardan foydalangan ma'qul. Chunki keyin kerakli qatorni, kerakli kodni topib olishda qulay bo'ladi, bir necha bo'sh qatorlarni qoldirib ketish, qator boshidan bo'sh joy qoldirib ketish tushuniladi. C# tilida chekinishlarga tabulyatsiya simvoli, probellar, bo'sh qatorlar kiradi.

Qisqartirilgan operatorlar. Bu operatorlar dastur kodini yozish jarayonini tezlashtirish uchun qo'llaniladi. Bu operatorlar quyidagilar:

2.3.3. jadval. Qisqartirilgan operatorlar

Operator	Tarifi
<code>+=</code>	qo'shish
<code>- =</code>	ayirish
<code>*=</code>	ko'paytirish
<code>G' =</code>	bo'lish

```
Int a = 1;
```

```
a += 3;
```

Bu kodda **a** ning qiymati 4 ga teng bo'ladi, chunki **a** ga 3 ni qo'shish ko'rsatilgan. **a = a +3** va **a += 3** ifodalari natijani bir xil chiqaradi. Qolgan operatorlar ham shunga o'xshash.^[12]

SHART OPERATORLARI

if..else operatori

if operatorining qo'llanilishi quyida ko'rsatilgan:

```
if (shart)
```

```
{
```

```
  kod
```

```
}
```

```
else
```

```
{
```

```
  kod
```

```
}
```

ko'rib turganingizdek birinchi bo'lib **if** operatori yoziladi, keyin qavs ichiga shart yoziladi. Shart mantiqiy ifoda ko'rinishida yoziladi. Shart yozilgandan keyin "{“ va “}" belgilari qo'yiladi, shartdagi ifoda qiymati **True**(rost) bo'lganida shu qavslar ichidagi kod ishlaydi, agar ifoda qiymati **False** (yolg'on) bo'lsa u holda to'g'ri o'tib ketadi yoki **else** operatori qo'yilgan bo'lsa **False** operatori qavslari ichidagi kod ishlaydi. Shart operatorlarini izma-iz qo'yish ham mumkin, buning uchun bir

shart operatori qavslari ya'ni kod yoziluvchi joyga boshqa shart operatorini qo'yish kerak, bu yo'l bilan bir necha shart operatorlarini qo'yish mumkin.

Case operatori

Case operatori ham shart operatori bo'lib hisoblanadi, uning qo'llanilishi quyidagicha:

```
int a = 3;
switch (a)
{
case 1:
System.Console.WriteLine(a);
break;
case 2:
System.Console.WriteLine(a);
break;
case 3:
System.Console.WriteLine(a);
break;
default:
System.Console.WriteLine("Xato");
break;
}
```

Yuqorida ko'rinib turganingizdek case operatori switch kalit so'zidan boshlangan. Undan keyin kod case larga bo'lindan, bu case lar switch kalit so'zi ichida ya'ni "{“ va “}” belgilari orasida joylashadi. Har bir case ning qiymati berilgan, agar yuqorida switch dan keyingi qavs ichiga yozilgan o'zgaruvchi qiymati biron bir case ning qiymatiga teng bo'lsa u holda shu case ichidagi kod ishlaydi. Agar teng bo'lmasa u holda default ichidagi kod ishlaydi. Case o'z qiymatiga yega bo'ladi va shu case ga tegishli kodning tamomlanganligini bildirish uchun kod oxiriga break kalit so'zi qo'yiladi, break; defaultda ham ishlatiladi. Case qiymati yozilgandan keyin (:) ikki nuqta qo'yiladi.

SIKL OPERATORLARI

While

While operatori sikl operatori bo'lib hisoblandi, dasturlashda keng **operatori** qo'llaniladi. Operatorning qo'llanilishi quyida ko'rsatilgan:

while (shart)

```
{  
kod  
}
```

Operator **While** so'zidan boshlanadi, izidan qavslar ichiga shart qo'yiladi. Agar shu shart qiymati **true** bo'lsa u holda operator ichiga ya'ni "{“ va “}” belgilari orasidagi kod ishlaydi. Masalan:

```
int a = 0;
```

```
while (a<10)
```

```
{  
a++;  
System.Console.WriteLine(a);  
}
```

Bu kod ishlaganda ekranda 1 dan 10 gacha bo'lgan sonlarni chop qiladi. Shart a 10 dan kichik bo'lsa deb qo'yilgan, birinchi siklda aq0 bo'ladi, shart o'rinli, operator kodi ishlaydi, a soniga birni qo'shadi(aQQ;) va ekranda 1 sonini chop qiladi, yana boshiga qaytadi, yana shart o'rinli chunki aq1, a<10 sharti o'rinli yana kod ishlaydi, shu ketishda sikl ishlayveradi, agar a soni 10 ga teng yoki 10 dan katta (a>q10) bo'lsa sikldan chiqib ketadi.

do (While operatori

do (While operatori sikl operatori, bu operator ham **while** operatoriga o'xshab ketadi, qo'llanilishi quyida ko'rsatilgan:

```
int a = 1;
```

```
do (
```

```
{  
System.Console.WriteLine(a);
```

```
a++;  
}
```

```
while (a<=10);
```

a o'zgaruvchisiga 1 qiymati beriladi va sikl birinchi marta ishlaydi, ekranda a o'zgaruvchisi qiymatini chop qiladi va a soniga birni qo'shadi. **While** qatoriga kelganda shart tekshiriladi, agar shart o'rinli bo'lsa u holda sikl qaytadan ishlaydi. Birinchi sikldayoq shart o'rinlanmay qolganda ham sikl bir marta ishlab qoladi. Chunki shart koddan keyin yoziladi.

For operatori

For sikl operatori dasturlashda boshqa sikl operatorilariga qaraganda ko'p qo'llaniladi. Bu operatorning qo'llanilishi quyida ko'rsatilgan:

```
for (int i = 1; i <= 10; i++){ System.Console.WriteLine(i);}
```

Sikl **for** so'zidan boshlangan, keyin qavs ochilgan, qavs ichiga yoziluvchi kodlar (;) belgisi bilan ajratilgan uch bo'lakka bo'lingan, qavs ichida birinchi bo'lakka siklni boshlovchi o'zgaruvchi va qiymati qo'yiladi, keyin siklning oxirini bildiruvchi o'zgaruvchi qiymati beriladi, uchinchi bo'lakka sikl o'zgaruvchisining o'sish yoki kamayish ifodasi yoziladi. Undan keyin **for** operatoriga tegishli "{", "}" qavslari ochiladi. Bu qavslar ichiga sikl aylanganda o'rinlanuvchi kod yoziladi. Yuqoridagi kodda i o'zgaruvchisi **int** tipidagi o'zgaruvchi qilib e'lon qilingan va qiymatini 1 qilib olingan, sikl oxiri i 10 dan kichik yoki teng bo'lsa deb olingan, ifodada ko'rinib turibdiki har sikl aylanganda o'zgaruvchiga bir soni qo'shib boriladi.

Bu dastur ishlaganda ekranda 1 dan 10 gacha bo'lgan sonlarni chop qiladi.

For operatorini qo'yish uchun oldin **for** so'zi yoziladi iziga qavs qo'yiladi.

Break buyrug'i

Bu buyruq **do** (**while**, **while**, **for** operatorlarida siklni to'xtatish uchun qo'llaniladi. Agar sikl ishlayotgan paytda sikldan chiqib ketish kerak bo'lsa **break** buyrug'idan foydalaniladi. Masalan:

```
int a = 1;  
do (
```

```

{
if (a==6)
{
break;
}
System.Console.WriteLine(a);
a++;
}
while (a<=10);

```

`goto` buyrug'i bir qatordan boshqa qatorga ya'ni `Label` ga o'tishda qo'llaniladi. `Label` ni qo'yish uchun uning nomini yozib oxiriga (:) ikki nuqta qo'yish lozim. Keyin shu `Label` nomini `goto` buyrug'idan keyin yozish kerak, shunda dastur `Label` qo'yilgan joydan boshlab ishlab ketadi.^[13] Masalan:

```

int a = 1;
myLabel:
a++;
if (a<10)
{
goto myLabel;
}

```

Sinflar va ob'ektlar

Atrofimizdagi predmetlarning hammasi ob'ekt. Masalan, avtomobil – bu ob'ekt, velosiped - bu ob'ekt, uy – bu ham ob'ekt. Bu ob'ektlarni sinflarga yig'ishimiz mumkin. Masalan, avtomobil boshqa avtomobillarga ayrim jihatlaridan o'xshab ketadi, ularda dvigatel, g'ildiraklar, rul, pedallar bor. Bu avtomobillarni umumiy avtomobillar sinf qilib olishimiz mumkin. Bundan boshqa faqat bir xil rusumli avtomobillar, masalan, Mercedes-Benz rusumiga tegishli avtomobillarni bir sinfga yig'ishimiz mumkin. Ob'ektlar o'z xarakterlariga yega bo'ladi, masalan, avtomobil rangi, ot kuchi, rusumi va h.

Sinflarni e'lon qilish

Sinflarni ob'ektlarni ornatuvchi shablon deb olsak ham bo'ladi. C# tilida ob'ekt xarakterlari o'zgaruvchilarda saqlanadi ya'ni kengliklarda, modellashtirishni metodlar ishlaydi. Metod va kengliklar sinf komponentlari deb aytiladi. Metodga misol Main() metodi. Sinflarni e'lon qilish quyida ko'rsatilgan:

```
[kirish darajasi] class sinf_nomi { kod }
```

Kirish darajasi bu shu sinfga tashqaridan kirish imkoniyati darajasi. Odatda sinflar public KI (kirish imkoniyati, dostup) bilan ochiladi. Bu KI ochiq ekanligini bildiradi. Endi bir necha kenglik va metodlarga ega sinf yarataylik:

```
public class Char
{
    public string make;
    public string model;
    public void Start()
    {
        System.Console.WriteLine(model + " o't oldi");
    }
    public void Stop()
    {
        System.Console.WriteLine(model + " to'xtadi");
    }
}
```

Yuqoridagi kodda “ Char ” nomli klass e'lon qilingan. Bu sinf make va model nomli ikkita kenglikga yega. Start (), Stop () metodlari ko'rsatilgan. Bu metodlar avtomobil o't olishini va to'xtashini ekranda chop qilib turadi. Kengliklar string ya'niy satriy o'zgaruvchi ko'rinishida e'lon qilingan, bu kengliklarda avtomobil rusumi va model nomi saqlanadi. Ko'rib turganingizdek kengliklar public ki bilan e'lon qilingan, endi bu kengliklar qiymati tashqaridan turib olishimiz mumkin. Metodlar ham public, tashqaridan metodlarni ishlatishimiz

mumkin. Bu metodlar void kalit so'zi bilan e'lon qilingan, endi ular qiymat bermaydi.^[14]

Ob'ekt yaratish

Sinf e'lon qilingandan keyin ob'ektlarni yaratishimiz mumkin. Quyidagi kodda **new** operatori orqali **Char** sinfiga tegishli **myChar** ob'ektini yaratadi:

```
Char myChar;
```

```
myChar = new Char();
```

Kenglikni ob'ektga chaqirish uchun (.) nuqtadan foydalaniladi.

```
myChar.make = "Toyota";
```

```
myChar.model = "MR2";
```

Ekranida ob'ekt kengliklarini chop qilish uchun quyidagicha kod yoziladi:

```
System.Console.WriteLine(myChar.make);
```

Metodlarni chaqirish uchun ham (.) nuqta qo'llaniladi:

```
myChar.Start();
```

```
myChar.Stop();
```

Yangi ob'ekt yaratish uchun quyidagicha yoziladi:

```
Char redPorsche = new Char();
```

Har bir ob'ekt uchun klass tegishli kengliklardan o'zida nusxasi bo'ladi:

```
redPorsche.make = "Porsche";
```

```
redPorsche.model = "Boxter";
```

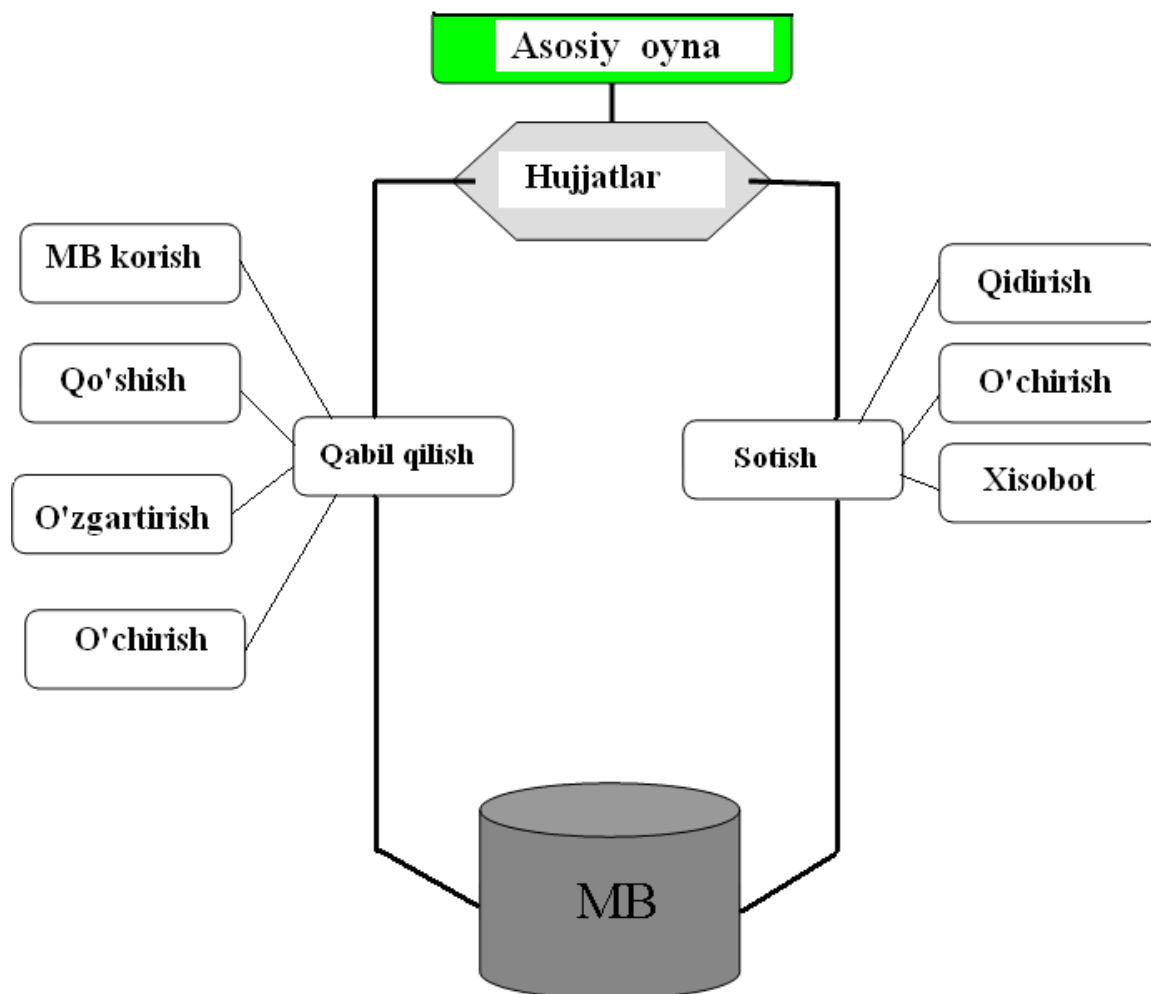
3. «AVTOSALONLARNING» ISH JARAYONINI AVTOMATLASHTIRISH TIZIMINI YARATISH

3.1. «Avtosalonlarning» ish jarayonini avtomatlashtirish tizimini yaratish va strukturasini ishlab chiqish

Quyida biz yaratgan «Avtosalonlarning» ish jarayonini avtomatlashtirish tizimini yaratish turli komponentalar bilan bog'liqlik strukturaviy sxemasi keltirilgan (3.1.1-rasm).

Avtomatlashtirilgan tizim strukturasini bir nechta komponentdan tashkil topgan:

- ma'lumotlar bazasi
- qabul qilish(prixod): qo'shish, o'zgartirish, o'chirish va h.k.
- sotish (rasxod): qidirish, o'chirish, hisobot.



3.1.1-rasm. «Avtosalon» ish jarayonini avtomatlashtirish tizimi strukturasini

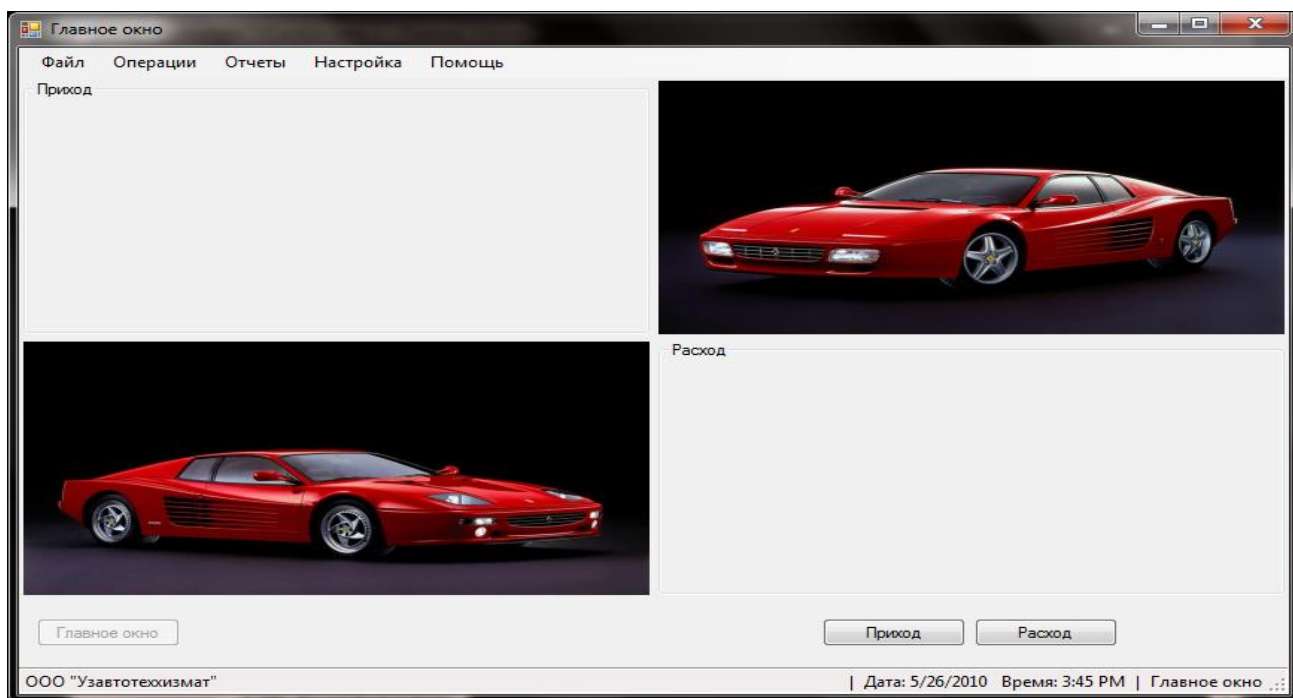
Axborotlar bu dasturiy ta'minotda bir necha qismlardan iborat. Ular **Bosh menyu, Hujjatlar , MB, Qabul qilish, Sotish, Tozalash , O'zgartirish , Qidiruv , Hisobot , Bosma , O'chirish , Chiqish va Yuklash** dasturni qismlari bir-birlari orqali bog'langan. Bular quyidagi vazifani bajaradi:

- **Bosh menyu** – dasturni ishga tushiradi.
- **Xujjatlar** – Ma'lumotlar bazasi, MB kiritish, Qo'shish, Tozalash, O'zgartirish, Qidirish, Hisobot, Vosma, O'chirish, Chiqish larni ishga tushiradi.
- **Qabul qilish** – quyidagi ma'lumotlar kiritiladi (**Avtomashinalar, Ularni turi, Ishlab chiqargan korxonalar, Nakladnoy sanasii, Nakladnoy №, Schyot faktura №, Schyot faktura sanasi, Modeli, Modifikatsiyasi, Tov.№, Vin №, Dvigatel №, Rangi, Narxi v.k.**) kiritiladi.
- **Sotish** – soyilgan avtomashinalar to'g'risidagi ma'lumotlarni **MBga** qo'shadi.
- **Tozalash** – **MB kiritish** dagi menyuda saqlab qolingan eski ma'lumotlarni o'chirib, yangi ma'lumot kiritish uchun sahifani tozalaydi.
- **O'zgartirish** – **MB** ga kiritilgan ma'lumotlarniga qo'shimcha va o'zgartirishlar kiritishni ta'minlaydi.
- **Qidiruv** – **MB** dagi ma'lumotlardan (**Avtomashinalar, Ularni turi, Modeli, Modifikatsiyasi, Nakladnoy sanasi**) bo'yicha tez va qulay holda qidiruvni amalga oshiradi.
- **Hisobot** – **MB** dagi (**Avtomashinalar, Ularni turi, Modeli, Modifikatsiyasi, Nakladnoy sanasi**) ma'lumotlardan hohlaganini tanlab hisobot tayyorlanadi.
- **Bosma** – **MB** dagi har bir Avtomashinaga oid ma'lumotlarni exe ko'runishida saqlaydi va bosmaga chiqarish imkoniyatini beradi.
- **O'chirish** – **MB** dagi Avtomashinalar haqidagi ma'lumotlarni o'chiradi.

- **Chiqish – dasturga** kirgandan so'ng dasturdan chiqish uchun xizmat qiladi va kerakli ma'lumotlarni foydalanuvchiga Microsoft Exell dasturiga yuboradi va foydalanuvchi o'ziga mos xolatda bosmadan chiqaradi.

3.2. Foydalanuvchi uchun qo'llanma

Asosiy oyna ko'rinishi. Ushbu dasturda ikkita jarayon sodir bo'lib ulardan birinchisi moshinalarni sotish uchun qabul qilib olish jarayoni (prixod) va omborda turgan moshinalarni sotish jarayoni (Rasxod) hisoblanadi. Ushbu jarayonlarning ikkisi ham ushbu dasturda joylashgan bo'lib, ikkisini ham asosiy oynadan ochish imkoniyati mavjud, ya'niy dasturning quyi qismida joylashgan «Prixod» va «Rasxod» tugmalari yordamida amalga oshirish mumkin.



3.2.1-rasm. Bosh oyna

Qabul qilib olish oynasi (3.2.2-rasm) ikkita bo'limdan iborat bo'lib birinchi bo'limida moshinalar kirish hujjati bo'lib hisoblanmish yo'riqnoma haqidagi ma'lumotlarni olish mumkin.

Ikkinchi bo'limida tanlangan yo'riqnomada oynasi (3.2.3-rasm) qanday mashinalar mavjudligini va ular haqida ma'lumotlar berish uchun xizmat qiladigan bo'lim hisoblanadi. Ushbu oynada biz ma'lumotlarni qabul qilib olingan sanasi bo'yicha qidirish imkoniyati mavjud bo'ladi. Bundan ma'lumotlarni aniq bir qabul qilish sanasi bo'yicha qidirib uni topish imkoniyati mavjud hisoblanadi. Bundan tashqari ushbu ishchi oynasida ma'lumotlarni qanday turga tegishli ekanligini bilgan holda ularni saralash imkoniyati mavjud. Ushbu jarayon «Tip avtomobilya» bo'limida amalga oshiriladi. Ushbu oynada qo'shish tugmasi bosgan holda yangi yo'riqnoma qo'shishimiz mumkin.

№	Дата накладной	№ накладной	Дата счет фактуры	№ счет фактуры	От кого	Получатель	Тип автомобиля
1	24.07.2009	87	24.07.2009	6767	ООО "Фаргона..."	Автосалон "Рох..."	Авто. нал.
2	24.07.2009	45	24.07.2009	466	ЗАО "GM Uzbeki..."	Автосалон "Рох..."	Авто. СКВ
3	02.06.2010	89	02.06.2010	21321122	ООО "BAYER G..."	Автосалон "Рох..."	Авто. нал.

№	Модель	Модификация	Опция	Тов. номер	VIN номер	Номер двигателя	Цвет	Отп. цена пред.	Отп. цена
1	Damas D2	NBB	B0	125847	WXXV125452185575	5223696522	95L	9480787	11990408
2	Damas D2	VAN	B0	022584	WXXV125452185574	5223696526	95L	8509021	10761409
3	Lacetti	CDX A/T	B3 GB HD 71 m2 r1	022584	WXXV125452185573	5223696525	95L	25479572	32224164
4	Lacetti	CDX A/T	B3 GB HD W1 71 m2 r1	057478	WXXV125452185572	5223696528	95L	25513164	32266647

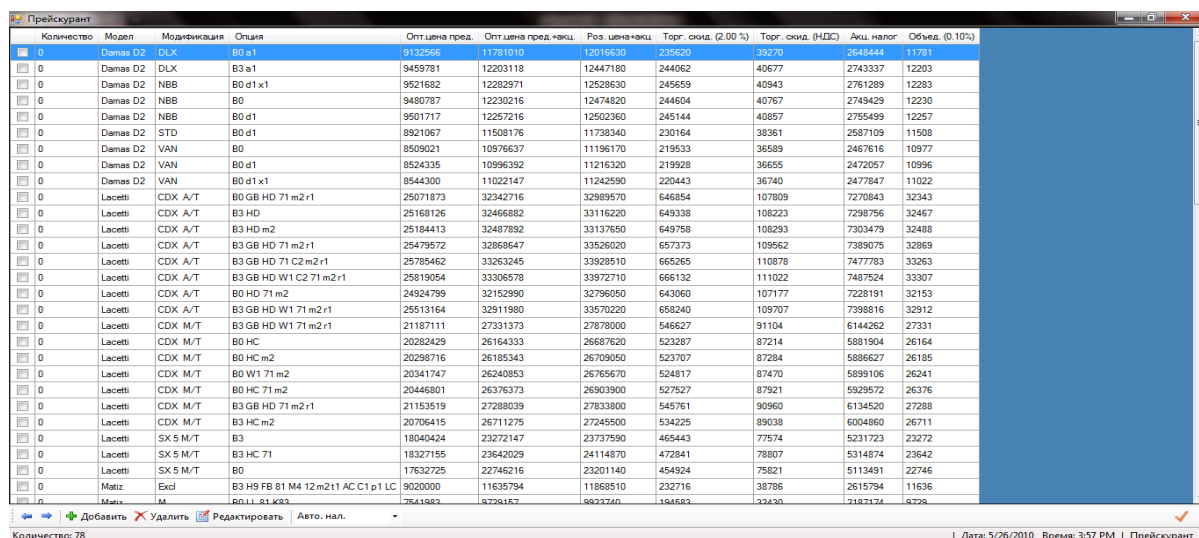
3.2.2-rasm. Qabul qilish oynasi

Yo'riqnoma qo'shish ishchi oynasi.

№	Модель	Модификация	Опция	Тов. номер	VIN номер	Номер двигателя
1	Damas D2	NBB	B0	125847	WXXV125452185575	5223696522
2	Damas D2	VAN	B0	022584	WXXV125452185574	5223696526
3	Lacetti	CDX A/T	B3 GB HD 71 m2 r1	022584	WXXV125452185573	5223696525
4	Lacetti	CDX A/T	B3 GB HD W1 71 m2 r1	057478	WXXV125452185572	5223696528

3.2.3-rasm. Yo'riqnoma qo'shish ishchi oynasi

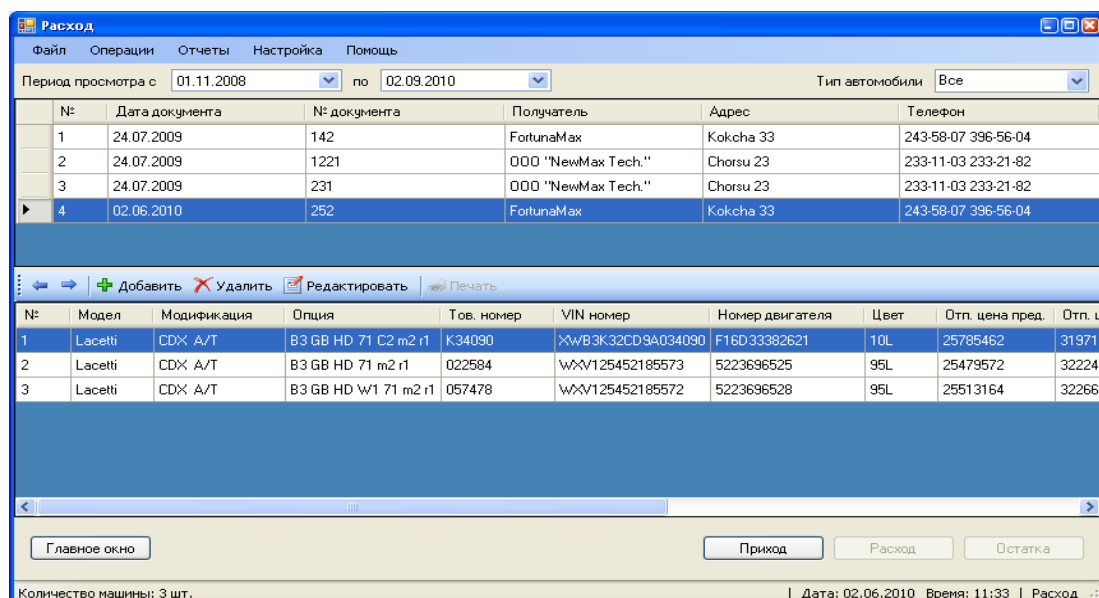
Ushbu Nakladnoy (3.2.3-rasm) oynada yo'riqnomaning nomeri va boshqa ma'lumotlarini kiritgan holda unga tegishli bo'lgan mashinalarni kiritish uchun qo'shish tugmasini bosamiz. Shundan so'ng bizda quyidagi ishchi oynasi ochiladi.



Количество	Модель	Модификация	Опция	Опт. цена пред.	Опт. цена пред. «экз.»	Роз. цена «экз.»	Торг. оид. (2.00 %)	Торг. оид. (HDC)	Акц. налог	Объед. (0.10%)
0	Demas D2	DLX	B3 a1	9133555	11791810	12016530	235520	30275	2648444	11721
0	Demas D2	NBB	B0 d1 x1	9459781	12203118	12447180	244062	40677	2743337	12203
0	Demas D2	NBB	B0	9521682	12282971	12528630	245659	40843	2761289	12283
0	Demas D2	NBB	B0 d1	9480787	12230216	12474820	244604	40767	2749429	12230
0	Demas D2	STD	B0 d1	9501717	12557216	12502360	245144	40857	2755499	12257
0	Demas D2	VAN	B0	8921067	11508176	11738340	230164	38361	2587109	11508
0	Demas D2	VAN	B0 d1 x1	8509021	10976637	11196170	219533	36589	2467616	10977
0	Demas D2	VAN	B0 d1	8543335	10996392	11216320	219928	36555	2472057	10996
0	Demas D2	VAN	B0 d1 x1	8543300	11022147	11242590	220443	36740	2477847	11022
0	Lacetti	CDX A/T	B0 GB HD 71 m2 r1	25071873	32342716	32989570	646854	107809	7270843	32343
0	Lacetti	CDX A/T	B3 HD	25168126	32466882	33116220	649338	108223	7296796	32467
0	Lacetti	CDX A/T	B3 HD m2	25184413	32487892	33137650	649758	108293	7303479	32488
0	Lacetti	CDX A/T	B3 GB HD 71 m2 r1	25479572	32669647	33586020	657373	109562	7389075	32669
0	Lacetti	CDX A/T	B3 GB HD 71 C2 m2 r1	25785462	33263245	33928510	665265	110878	7477783	33263
0	Lacetti	CDX A/T	B3 GB HD W1 C2 71 m2 r1	25819054	33305578	33972710	666132	111022	7487524	33307
0	Lacetti	CDX A/T	B0 HD 71 m2	24924789	32152990	32796050	643060	107177	7228191	32153
0	Lacetti	CDX A/T	B3 GB HD W1 71 m2 r1	25513164	32911980	33570220	658240	109707	7398816	32912
0	Lacetti	CDX M/T	B3 GB HD W1 71 m2 r1	21167111	27331373	27878000	546627	91104	6144262	27331
0	Lacetti	CDX M/T	B0 HC	20282429	26164333	26687620	523287	87214	5881904	26164
0	Lacetti	CDX M/T	B0 HC m2	20289716	26185343	26708050	523707	87284	5886627	26185
0	Lacetti	CDX M/T	B0 W1 71 m2	20341747	26240853	26765670	524817	87470	5899106	26241
0	Lacetti	CDX M/T	B0 HC 71 m2	20446801	26376373	26903900	527527	87521	5929572	26376
0	Lacetti	CDX M/T	B3 GB HD 71 m2 r1	21153519	27288039	27833800	545761	90960	6134520	27288
0	Lacetti	CDX M/T	B3 HC m2	20706415	26711275	27245500	534225	89038	6004860	26711
0	Lacetti	SX 5 M/T	B3	18840424	23272147	23737590	465443	77574	5231723	23272
0	Lacetti	SX 5 M/T	B3 HC 71	18327155	23642029	24114870	472841	78807	5314874	23642
0	Matiz	Excl	B0	17632725	22746216	23201140	454924	75821	5113491	22746
0	Matiz	Excl	B3 H9 FB 81 M4 12 m211 AC C1 p1 LC	9020000	11635794	11868510	232716	38785	2615794	11636
0	Matiz	Excl	B0	7541983	9228167	9823730	194583	32430	2187174	9228

3.2.4-rasm. Preyskurant oynasi

Ushbu oyna preykurant (3.2.4-rasm) deb atalib, ushbu hujjat yordamida biz mashinalar turlari va ularning parametrlari bo'yicha yo'riqnomaga qoshishimiz mumkin.



№	Дата документа	№ документа	Получатель	Адрес	Телефон
1	24.07.2009	142	FortunaMax	Kokcha 33	243-58-07 396-56-04
2	24.07.2009	1221	ООО "NewMax Tech."	Chorsu 23	233-11-03 233-21-82
3	24.07.2009	231	ООО "NewMax Tech."	Chorsu 23	233-11-03 233-21-82
4	02.06.2010	252	FortunaMax	Kokcha 33	243-58-07 396-56-04

№	Модель	Модификация	Опция	Тов. номер	VIN номер	Номер двигателя	Цвет	Опт. цена пред.	Опт. цена
1	Lacetti	CDX A/T	B3 GB HD 71 C2 m2 r1	K34090	XXWB3K32CD9A034090	F16D 33382621	10L	25785462	319711
2	Lacetti	CDX A/T	B3 GB HD 71 m2 r1	022584	XXV125452185573	5223696525	95L	25479572	32224
3	Lacetti	CDX A/T	B3 GB HD W1 71 m2 r1	057478	XXV125452185572	5223696528	95L	25513164	32266

3.2.5-rasm. Rasxod oynasi

Sotish, ya'ni ombordagi mashinani sotish uchun qo'yidagi ishchi oynasidan (3.2.5-rasm) foydalanamiz. Ushbu oynada mashinani sotish uchun qo'shish tugmasini bosgan holda yangi hujjat rasmiylashtiriladi.

Документ

Документ № документа: Дата: 5/26/2010

Подразделение: Автосалон "Рохат"

Тип авто: Авто. нал.

№ счет фак.: Дата: 5/26/2010

Получатель:

№ доверенности:

Дата доверенности: 5/26/2010

Паспорт по доверенности:

Договор № договора:

Дата договора: 5/26/2010

Кредит банка:

Дата оплаты: 5/26/2010

Дата отгрузки: 5/26/2010

Пластиковая карточка Тип пласт. карточки:

№ пласт. карточки: -XXXX-XXXX-

Дата выдачи: 5/26/2010

Дата окончание: 5/26/2010

Курс доллара: 1474.45

OK Отмена

№	Модел	Модификация	Опция	Тов. номер	VIN номер	Номер двигателя	Цвет	Отп. цена пред.	Отп. цена пред. +
[Empty table body]									

Колчество машины: 0 шт. | Дата: 5/26/2010 Время: 4:00 PM | Документ ...

3.2.6-rasm. Hujjatlar oynasi

Ushbu oynaga rasmiylashtirilayotgan hujjat haqidagi turli ma'lumotlar kiritiladi. Haridorni qo'shish uchun tugmasini bosiladi. Shundan so'ng bizda quyidagicha oyna ochiladi (3.2.7-rasm).

Получатели

№	ФИО	Адрес	Телефон	МФО	ОКОНК	ИНН
1	Zokirov Valmatilla Xorujbaevich	Ташкент Kolscha 33	243-58-07 396-56-04	00416	71500	200625047
2	ООО "NewMax Tech."	Ташкент Chotshi 23	233-11-03 233-21-82	00873	14341	200244767
3	Раджапов Жасур Турғибаевич	Ташкент ул.Юнусова 4 дом	993-77-74 - -	00723	75331	20100723
4	МЧУ "Sifat Kafolati Sertifikat"	Ташкент ул.Фароҳия 72 дом	228-29-22 - -	00915	72331	24564451
5	Гафур Абдулов Шайхонвич	Хоразм ул. Махмудов 14 дом	774-34-55 - -	00126	73445	27554231

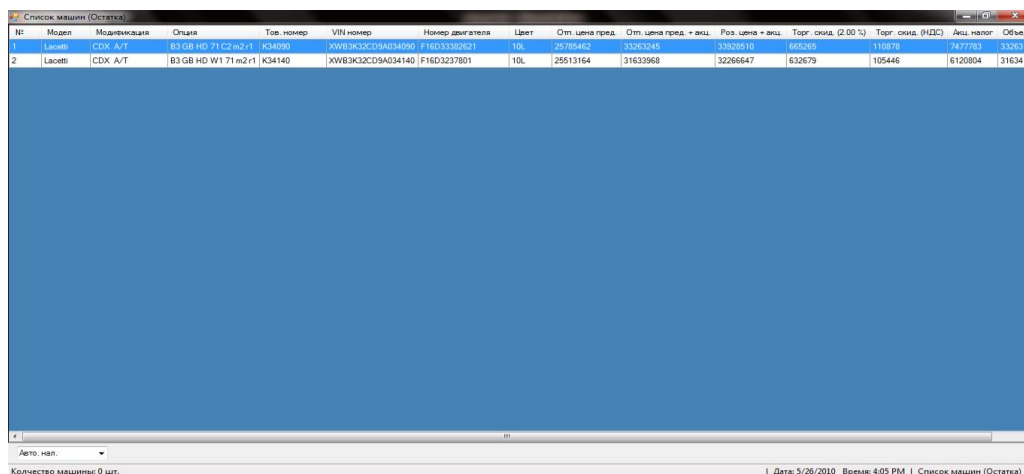
Колчество получателей: 0

Дата: 19.05.2011. Время: 6:40 | Получатели

3.2.7-rasm. Xaridor xaqidagi oyna

Bu ishchi oynasida haridorlar ro'yxati joylashgan bo'lib, haridorni tanlashimiz yoki yangi haridor qo'shish imkoniyati mavjud.

Hujjat rasmiylashtirish oynasida  **Добавить** tugmasini bosgan holda omborxonadagi mashinalardan tanlashimiz va ularni sotish uchun tanlashimiz mumkin. Bunda quyidagi (3.2.8-rasm) oynadan foydalaniladi.



№	Модель	Модификация	Опция	Тов. номер	VIN-номер	Номер двигателя	Цвет	Отп. цена пред.	Отп. цена пред. + авт.	Роз. цена + авт.	Торг. цена (2.00 %)	Торг. цена (H2C)	Авт. налог	Объем
1	Lacetti	CDX A/T	B3 GB HD W1 C2 m2 r1	K34090	XWB3K3C2DA034090	F16D1332821	10L	25785462	33263245	33502010	865255	110878	7477703	33263
2	Lacetti	CDX A/T	B3 GB HD W1 71 m2 r1	K34140	XWB3K3C2C09A034140	F16D3237801	10L	25513164	31633968	32266647	632679	105446	6120804	31634

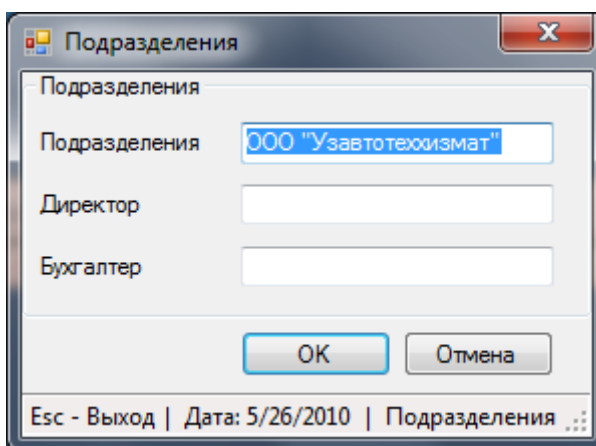
Автом. нал. | Дата: 5/26/2010 | Времени: 4:05 PM | Список машин (Остаток)

3.2.8-rasm. Qolgan mashinalar ro'yxati oynasi

Ushbu oynada (3.2.8-rasm) omborxonada sotish uchun ruxsat bo'lgan mashinalar ro'yxati chiqadi.

Yuqoridagilarni amalga oshirish uchun biz dasturni sozlab olishimiz darkor bo'ladi. Buning uchun bizda mahsus menyu, ya'ni sozlash (Nastroyka) menyusidan foydalanishimiz mumkin.

Tashkilot haqidagi ma'lumotlarni «Informatsiya o dillere» bandida olishimiz va ularni o'zgartirishimiz mumkin:



Подразделения

Подразделения: ООО "Узавтотехизмат"

Директор: _____

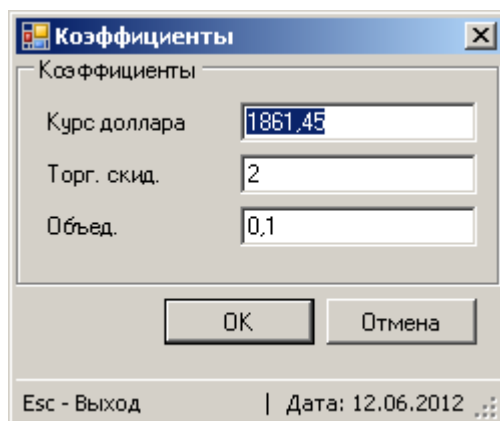
Бухгалтер: _____

OK Отмена

Esc - Выход | Дата: 5/26/2010 | Подразделения

3.2.9-rasm. Tashkilot xaqida ma'lumot oynasi

Dollar kursi va AvtoPROM uchun topshiriladigan koeffitsient qiymatlarini «Koeffitsienty» bandida o'rnatishimiz mumkin (3.2.10-rasm).



Кoeffициенты	
Курс доллара	1861.45
Торг. скид.	2
Объед.	0.1

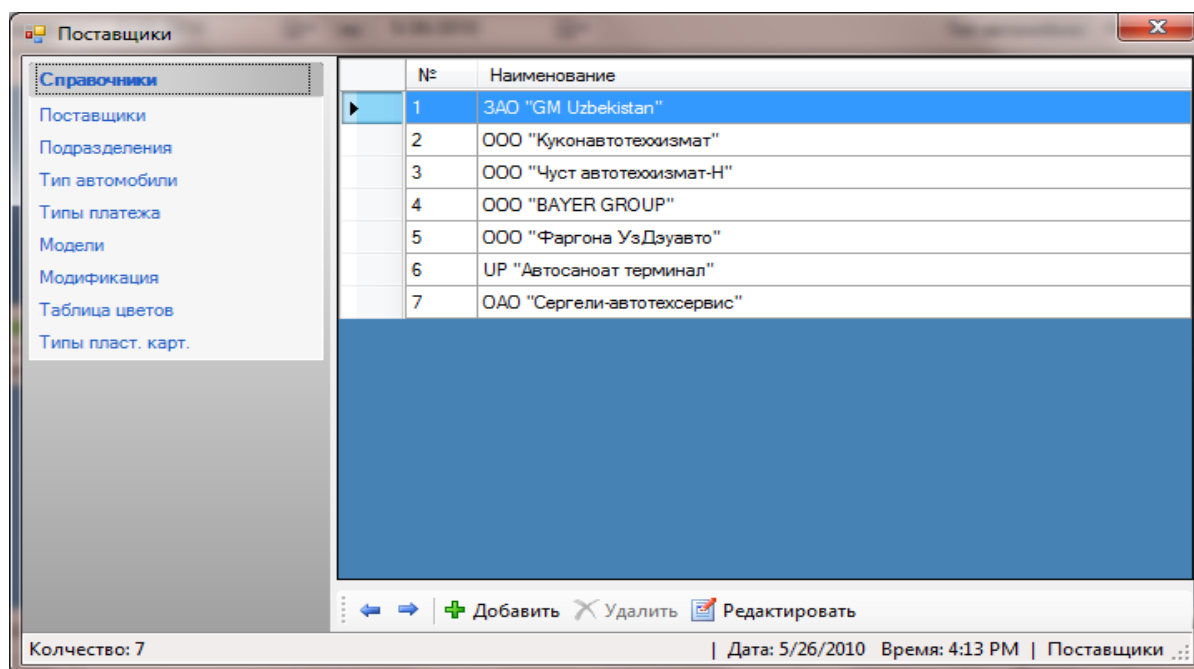
OK Отмена

Esc - Выход | Дата: 12.06.2012

3.2.10-rasm. Dollar kursi oynasi

Haridorlar ro'yxatini va preykurantlarni ham shu menyu orqali o'zgartirishimiz mumkin.

«Spravochnik» (3.2.11-rasm) bandida biz mashinalar turlarini, tashkilotlarni, yetkazib beruvchilarni, to'lov turini, mashinalar modifikatsiyasini, mashinalar ranglari xaqidagi ma'lumotlarni o'rnatishimiz, qo'shishimiz va o'chirishimiz mumkin.



№	Наименование
1	ЗАО "GM Uzbekistan"
2	ООО "Куконавотехизмат"
3	ООО "Чуст автотехизмат-Н"
4	ООО "BAYER GROUP"
5	ООО "Фаргона УзДэуавто"
6	УР "Автосаноат терминал"
7	ОАО "Сергели-автотехсервис"

Колчество: 7 | Дата: 5/26/2010 | Время: 4:13 PM | Поставщики

3.2.11-rasm. Spravochnik oynasi

X U L O S A

Ushbu bitiruv malakaviy ishini ishlab chiqishda «Avtosalonlarning» ish jarayonini avtomatlashtirish tizimi va MBning dasturi yaratildi. Dasturni yaratishda hozirgi zamon talablariga javob bera oladigan C# dastur tilidan foydalanildi va dastur uchun ishlatilgan Microsoft Access ma'lumotlar bazasini boshqarish tizimida ma'lumotlar bazasi ishlab chiqildi.

Birinchi bobida asosiy xulosa va natijalar sifatida quyidagilarni keltiramiz:

- MBning ta'rif, umumiy tushunchalari va tashkiliy qismlari keltirildi;
- Ma'lumotlar bazasi arxitekturasini va tushunchasi yoritildi;
- MBni tashkil etuvchi relyatsion ma'lumotlar bazasi texnologiyasidagi makroslar orqali makrobuyruqlar tashkil qilindi;
- Batafsil yuqoridagilar taxlil qilinib masalani qo'yilishi keltirildi.

Ikkinchi bobida dasturni yaratishda kerak bo'ladigan va tanlab olingan instrumental vositalar haqida, ularni qulayligi va ishlash jarayoni misollar tariqasida keltirildi.

Uchinchi bobida Avtosalonlarning ish jarayonlarini avtomatlashtirish tizimini loyixalashning ish rejasi belgilanib MB strukturasi keltirildi. Shuningdek korxona axborot-ma'lumot bazasini dasturiy vositalari va ta'minoti ishlab chiqildi. Foydalanish uchun batafsil bulgan yo'riqnoma keltirildi. Dasturiy taminotga qo'yiladigan apparat, ya'ni texnik hamda dasturiy ta'minotiga qo'yiladigan talablar ko'rsatildi.

Bitiruv malakaviy ishini ishlab chiqish jarayonida quyidagi muhim vazifalar amalga oshirildi:

- «Avtosalonlarning» ish jarayonini avtomatlashtirish tizimini MBsi yaratildi;
- MBning strukturasi ishlab chiqildi;
- Foydalanuvchiga qulay bo'lgan interfeysi ishlab chiqildi;
- MBdan ma'lumotlarni bosmaga chiqarib berish imkonini beruvchi hisobotlar tuzildi;
- MB uchun ma'lumotlar bazasidagi operatsiyalarni: saqlash, o'chirish,

taxrirlash, ma'lumotlarni yozish kabi amallarni bajaruvchi dasturiy ta'minot yaratildi.

Ishlab chiqilgan dastur korxona raxbariyatiga ayrim xizmatlar va butun korxonaning rentabilliligi xaqida kerakli ma'lumotni taqdim etuvchi analitik hisobotni shakillantirishga va ish vaqtini tejash imkonini beradi.

FOYDALANGAN ADABIYOTLAR

1. 2011 yilning asosiy yakunlari va 2012 yilda O'zbekistonni ijtimoiy-iqtisodiy rivojlantirishning ustuvor yo'nalishlariga bag'ishlangan O'zbekiston Respublikasi Vazirlar Mahkamasining majlisidagi "2012-yil vatanimiz taraqqiyotini yangi bosqichga ko'taradigan yil bo'ladi" mavzusidagi ma'ruzasini. "O'qituvchi" NMIU. Toshkent 2012.
2. Karimov I.A.«Yuksak bilimli va intellektual rivojlangan avlodni tarbiyalash – mamlakatni barqaror taraqqiy ettirish va modernizatsiya qilishning eng muhim sharti» mavzusidagi xalqaro konferensiyadagi nutq // Xalq so'zi, 2012 yil 18 fevral.
3. A Konnoli T., Bregk K. Базы данных, проектирование, реализация и сопровождение, теория и практика, Университет Пеysli, Шотландия, изд. М.-СПБ.- Киев, 2003.
Axborot tizimlari va texnologiyalari S.S. G'ulomov, va boshqalar T: «Sharq» 2000y.-342s.
4. A. Xomokeno i dr. "Baza dannix" Uchebnik Sank-Peterburg 2004g.
5. Bajenova I.Yu. Samouchitel programmista. – M.: 2003y.- 448s
6. Ma'lumotlar bazalari. B.A. Begalov, E.E.Inog'omjonov, Sh. Nasriddinova. Toshkent – 2008y. -124s
7. Bobrovskiy S. Uchebno'y kurs. – SPb.:Piter. 2004y. – 736s
8. V. Vasvani. MySQL: ispol'zovanie i administrirovanie MySQL Database Usage & Administration. — M.: «Piter», 2011.
9. .A. Xomokenno "Microsoft Access 2007" Ekspress kurs Peterburg 2009y.- 47s.
10. Begalov B.A., Abdullaev i dr. Vvedenie v bazu dannix i znaniy. Tashkent, 2007. -129s.
- 11.Labor Vladimir Vladimirovich "S# Sozdanie prilozheniy dlya Windows" 2007 y.
- 12.Pettsol'd Charl`z "Programmirovaniye dlya Microsoft Windows na S#"
- 13.Tom 1-2. Moskva 2007.

- 14.C# dlya professionalov / K. Neygel, B. Iv`en, D. Glinn, K. Uotson. 2008.
- 15.AyvorXortonMicrosoftVisual C++ 2005: bazovyy kurs - BeginningVisual C++ 2005. — M.: «Dialektika», 2008.
- 16.16. Xayot faoliyati xavfsizligi fanidan ma'ruzalar matni. Q.Yo. Yormatov. Toshkent - 1999 y.- 109s.
- 17.Konnoli T., Bregk K. Bazy dannых, proektirovanie, realizatsiya i soprovjdeniya, teoriya i praktika, Universitet Peysli, Shotlandiya, izd. M.-SPB.- Kiev, 2003.
- 18.Krenkyo D.. Teoriya i praktika postroeniya baz dannых. M-SPb,-2003.
- 19.Golitsyna O.L.,Maksimov N.V.Popov I.I. Bazy dannых: Uchebnoe posobie.-M.:Forum:Infra-M, 2005. 352 s
- 20.www.Google.com.uz

ILOVA

Salon dasturining listiningi

```
using System;
using System.Reflection;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.OleDb;
using System.Drawing;
using System.Text;
using System.IO;
using System.Windows.Forms;

namespace Storehouse
{
    public partial class MainForm : Form
    {
        public MainForm()
        {
            InitializeComponent();

            #region Variables
            OleDbConnection con = new
OleDbConnection(@"Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=|DataDirectory|\base\rohat.mdb;Persist Security Info=True");
            OleDbDataAdapter dataAdapter = new OleDbDataAdapter("",
@"Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=|DataDirectory|\base\rohat.mdb;Persist Security Info=True");
            string currentState = "";
            #endregion

            /// <summary>
            /// Not finished Buttons Excel
            /// </summary>

            private void PrintButton_Click(object sender, EventArgs e)
            {
                Microsoft.Office.Interop.Excel.ApplicationClass excel = new
Microsoft.Office.Interop.Excel.ApplicationClass();
                switch (currentState)
                {
                    case "naklad":
                        //
                        break;
                    case "selldoc":
                        #region Table create for excel
                        System.Data.DataTable table = new
System.Data.DataTable("table");
                        table.Columns.Add("Модель");
                        table.Columns.Add("Номер кузова (VIN номер)");
                        table.Columns.Add("Отпускная цена");
                        table.Columns.Add("Розничная цена");
                        table.Columns.Add("Розничная цена (долл. США)");
                        table.Columns.Add("№ пласт. Карточки");
                        table.Columns.Add("Дата отпуска автомобиля");
                        table.Columns.Add("Ф.И.О. покупателя");
                        table.Columns.Add("Адрес");
                        table.Columns.Add("Тел.");
                        #endregion

```

```

        dataAdapter.SelectCommand.CommandText = "select * from
[auto] where [status]=1";
        System.Data.DataTable dt = new
System.Data.DataTable("dt");
        dataAdapter.Fill(dt);
        System.Data.DataTable dtCustomer = new
System.Data.DataTable("customer");
        //string inn = "";
        for (int i = 0; i < dt.Rows.Count; i++)
        {
            dtCustomer.Clear();
            dataAdapter.SelectCommand.CommandText = "select *
from customer where inn=" + dt.Rows[i]["inn"].ToString();
            dataAdapter.Fill(dtCustomer);
            table.Rows.Add(
                dt.Rows[i][1].ToString() + " " +
dt.Rows[i][2].ToString(),
                dt.Rows[i][3].ToString(),
                dt.Rows[i][7].ToString(),
                dt.Rows[i][8].ToString(),
                "$$$$ ",
                "4790 XXXX XXXX 92",
                "06.06.2009",
                dtCustomer.Rows[0][2].ToString() + " " +
dtCustomer.Rows[0][3].ToString() + " " + dtCustomer.Rows[0][4].ToString(),
                dtCustomer.Rows[0][6].ToString(),
                dtCustomer.Rows[0][7].ToString());
        }

#region Opening Excel
excel.Workbooks.Open(Directory.GetCurrentDirectory() +
@"\xls\rasxod.xls",
        Missing.Value, Missing.Value, Missing.Value,
        Missing.Value, Missing.Value, Missing.Value,
Missing.Value,
        Missing.Value, Missing.Value, Missing.Value,
Missing.Value,
        Missing.Value, Missing.Value, Missing.Value);
int ColumnIndex = 0;
int rowIndex = 7;
#endregion

#region Filling data into Excel
foreach (DataRow row in table.Rows)
{
    rowIndex++;
    ColumnIndex = -1;

    #region New option adding
    foreach (DataColumn col in table.Columns)
    {
        ColumnIndex++;
        if (ColumnIndex < 11)
        {
            excel.Cells[rowIndex, ColumnIndex + 3] =
row[ColumnIndex].ToString();
        }
        else
        {
            ColumnIndex++;
        }
    }
}
#endregion
}

```



```

        #endregion

        excel.Visible = true;
        Microsoft.Office.Interop.Excel.Worksheet worksheet =
(Microsoft.Office.Interop.Excel.Worksheet)excel.ActiveSheet;
        worksheet.Activate();
        this.TopMost = false;
        worksheet = null;
        excel = null;
        break;
    }
}

private void buttonOstatka_Click(object sender, EventArgs e)
{
    Microsoft.Office.Interop.Excel.ApplicationClass excel = new
Microsoft.Office.Interop.Excel.ApplicationClass();

    #region Table create for excel
    System.Data.DataTable table = new System.Data.DataTable("table");
    table.Columns.Add("От кого получено");
    table.Columns.Add("Модель");
    table.Columns.Add("Мод-я");
    table.Columns.Add("Опции");
    table.Columns.Add("Цвет");
    table.Columns.Add("Номер кузова (VIN номер)");
    table.Columns.Add("Номер двигателя");
    table.Columns.Add("Документ по приходу № накл");
    table.Columns.Add("Документ по приходу Дата");
    table.Columns.Add("Отпускная цена в сумах");
    table.Columns.Add("Розничная цена в сумах");
    #endregion

    dataAdapter.SelectCommand.CommandText = "select * from [auto]
where [status]=0";
    System.Data.DataTable dt = new System.Data.DataTable("dt");
    System.Data.DataTable dtNaklad = new
System.Data.DataTable("naklad");
    dataAdapter.Fill(dt);

    for (int i = 0; i < dt.Rows.Count; i++)
    {
        dtNaklad.Clear();
        dataAdapter.SelectCommand.CommandText = "select * from naklad
where no_naklad='" + dt.Rows[i][10].ToString() + "'";
        dataAdapter.Fill(dtNaklad);
        table.Rows.Add(
            dtNaklad.Rows[0][0].ToString(),
            dt.Rows[i][1].ToString(),
            dt.Rows[i][2].ToString(),
            dt.Rows[i][3].ToString(),
            dt.Rows[i][6].ToString(),
            dt.Rows[i][4].ToString(),
            dt.Rows[i][5].ToString(),
            dt.Rows[i][10].ToString(),
            dtNaklad.Rows[0][4].ToString(),
            dt.Rows[i][7].ToString(),
            dt.Rows[i][8].ToString());
    }

    #region Opening Excel
    excel.Workbooks.Open(Directory.GetCurrentDirectory() +
@"\xls\ostatok.xls",
        Missing.Value, Missing.Value, Missing.Value,

```

```

Missing.Value, Missing.Value, Missing.Value,
Missing.Value, Missing.Value, Missing.Value,
Missing.Value, Missing.Value, Missing.Value);
int ColumnIndex = 0;
int rowIndex = 9;
//string model = "";
#endregion

#region Filling data into Excel
//excel.Cells[3, 4] = "(действуют с " +
DateTime.Now.ToShortDateString() + " г.)";
foreach (DataRow row in table.Rows)
{
    rowIndex++;
    ColumnIndex = -1;

    #region New option adding
    foreach (DataColumn col in table.Columns)
    {
        ColumnIndex++;
        //Microsoft.Office.Interop.Excel.Range celModif =
(Microsoft.Office.Interop.Excel.Range)excel.Cells[rowIndex, ColumnIndex];
        //if (ColumnIndex == 2 || ColumnIndex == 3)
        //{
        //    celModif.Font.Bold = true;
        //}
        //else
        //{
        //    celModif.Font.Bold = false;
        //}
        //celModif.Font.Size = 10;
        if (ColumnIndex < 11 )
        {
            excel.Cells[rowIndex, ColumnIndex+2 ] =
row[ColumnIndex ].ToString();
        }
        else
        {
            ColumnIndex++;
        }
    }
    #endregion
}
#endregion

excel.Visible = true;
Microsoft.Office.Interop.Excel.Worksheet worksheet =
(Microsoft.Office.Interop.Excel.Worksheet)excel.ActiveSheet;
worksheet.Activate();
this.TopMost = false;
worksheet = null;
excel = null;
}

private void IncomeReportMenuItem_Click(object sender, EventArgs e)
{
    //
}

private void OutcomeReportMenuItem_Click(object sender, EventArgs e)
{
    //

```

```

    }

    private void BalanceMenuItem_Click(object sender, EventArgs e)
    {
        //
    }

    /// <summary>
    ///
    //////////////////////////////////////
    /// </summary>

    private void FSELLDOCLOAD()
    {
        dataGridCome.Rows.Clear();
        dataGridAuto.Rows.Clear();

        #region Selldoc read
        switch (comboTypeAuto.Text)
        {
            case "Bce":
                dataAdapter.SelectCommand.CommandText = @"select * from "
+ currentState;
                break;
            default:
                dataAdapter.SelectCommand.CommandText = @"select * from "
+ currentState + " where type_auto_name='" + comboTypeAuto.Text + "'";
                break;
        }
        System.Data.DataTable dtSelldoc = new
System.Data.DataTable(currentState);
        dataAdapter.Fill(dtSelldoc);
        #endregion

        #region Customer read
        dataAdapter.SelectCommand.CommandText = "select * from customer";
        System.Data.DataTable dtCustomer = new
System.Data.DataTable("customer");
        dataAdapter.Fill(dtCustomer);
        #endregion

        #region Selldoc add
        string fio = "";
        string address = "";
        string telefon = "";
        for (int i = 0; i < dtSelldoc.Rows.Count; i++)
        {
            fio = "";
            address = "";
            telefon = "";

            #region Find Customer
            for (int j = 0; j < dtCustomer.Rows.Count; j++)
            {
                if (dtSelldoc.Rows[i]["id_customer"].ToString() ==
dtCustomer.Rows[j]["id_customer"].ToString())
                {
                    fio = dtCustomer.Rows[j]["first_name"].ToString() + "
" + dtCustomer.Rows[j]["second_name"].ToString() + " " +
dtCustomer.Rows[j]["last_name"].ToString();
                    address = dtCustomer.Rows[j]["address"].ToString();
                    telefon = dtCustomer.Rows[j]["telephone"].ToString();
                    break;
                }
            }
        }
    }

```

```

    }
    #endregion

    if (dateCompare(dateStart.Text,
dtSelldoc.Rows[i] ["date_doc"].ToString(), dateEnd.Text))
    {
        dataGridCome.Rows.Add
        (
            i + 1,
            dtSelldoc.Rows[i] ["date_doc"].ToString(),
            dtSelldoc.Rows[i] ["no_doc"].ToString(),
            fio,
            address,
            telefon,
            dtSelldoc.Rows[i] ["id_selldoc"].ToString()
        );
    }
}
#endregion

EditStatus();
}

private void EditStatus()
{
    if (dataGridCome.Rows.Count != 0)
    {
        DeleteButtonItem.Enabled = true;
        EditButtonItem.Enabled = true;
    }
    else
    {
        DeleteButtonItem.Enabled = false;
        EditButtonItem.Enabled = false;
    }
}

private void FNAKLADLOAD()
{
    #region Naklad read
    switch (comboTypeAuto.Text)
    {
        case "Bce":
            dataAdapter.SelectCommand.CommandText = @"select * from "
+ currentState;
            break;
        default:
            dataAdapter.SelectCommand.CommandText = @"select * from "
+ currentState + " where type_auto_name='" + comboTypeAuto.Text + "'";
            break;
    }
    System.Data.DataTable dtNaklad = new
System.Data.DataTable(currentState);
    dataAdapter.Fill(dtNaklad);
    #endregion

    #region Add row
    for (int i = 0; i < dtNaklad.Rows.Count; i++)
    {
        if (dateCompare(dateStart.Text,
dtNaklad.Rows[i] ["date_naklad"].ToString(), dateEnd.Text))
        {
            dataGridCome.Rows.Add
            (

```

```

        i + 1,
        dtNaklad.Rows[i]["date_naklad"].ToString(),
        dtNaklad.Rows[i]["no_naklad"].ToString(),
        dtNaklad.Rows[i]["date_fact"].ToString(),
        dtNaklad.Rows[i]["no_fact"].ToString(),
        dtNaklad.Rows[i]["supplier_name"].ToString(),
        dtNaklad.Rows[i]["shop_name"].ToString(),
        dtNaklad.Rows[i]["type_auto_name"].ToString(),
        dtNaklad.Rows[i]["id_naklad"].ToString()
    );
    }
}
#endregion

EditStatus();
}

private void FLOAD()
{
    DataTable dtAuto = new DataTable("auto");
    dtAuto.Clear();
    dataAdapter.Fill(dtAuto);

    #region Rates read
    System.Data.DataTable dtRates = new
System.Data.DataTable("rates");
    dataAdapter.SelectCommand.CommandText = "select * from rates
order by id_rate asc";
    dataAdapter.Fill(dtRates);
    #endregion

    double a = Convert.ToDouble(dtRates.Rows[1][2].ToString());
    double b = Convert.ToDouble(dtRates.Rows[2][2].ToString());
    double[] Variables = new double[7];

    for (int i = 0; i < dtAuto.Rows.Count; i++)
    {
        #region Calculation
        Variables[0] =
Convert.ToInt32(dtAuto.Rows[i]["otp_price"].ToString());
        Variables[2] =
Convert.ToInt32(dtAuto.Rows[i]["roz_price"].ToString());
        Variables[1] = Math.Round((100 * Variables[2]) / (100 + a));
        Variables[3] = Math.Round((a * Variables[2]) / (100 + a));
        Variables[4] = Math.Round(Variables[3] / 6);
        Variables[5] = Variables[1] - Variables[0];
        Variables[6] = Math.Round((Variables[1] * b) / 100,
MidpointRounding.AwayFromZero);
        #endregion

        #region AddCode
        dataGridAuto.Rows.Add
        (
            i + 1,
            dtAuto.Rows[i]["model_name"].ToString(),
            dtAuto.Rows[i]["modif_name"].ToString(),
            dtAuto.Rows[i]["option"].ToString(),
            dtAuto.Rows[i]["no_prod"].ToString(),
            dtAuto.Rows[i]["no_body"].ToString(),
            dtAuto.Rows[i]["no_engine"].ToString(),
            dtAuto.Rows[i]["color_name"].ToString(),
            Variables[0],
            Variables[1],
            Variables[2],

```

```

        Variables[3],
        Variables[4],
        Variables[5],
        Variables[6]
    );
    #endregion
}
StatusLabel.Text = "Количество машины: " +
dataGridViewAuto.Rows.Count.ToString() + " шт.";
}

private bool dateCompare(string dStart, string dOriginal, string
dEnd)
{
    if (Convert.ToInt32(dStart.Substring(6, 4)) <
Convert.ToInt32(dOriginal.Substring(6, 4)) &&
        Convert.ToInt32(dOriginal.Substring(6, 4)) <
Convert.ToInt32(dEnd.Substring(6, 4)))
    { return true; }
    else
    {
        if (Convert.ToInt32(dStart.Substring(6, 4)) ==
Convert.ToInt32(dOriginal.Substring(6, 4)) &&
            Convert.ToInt32(dOriginal.Substring(6, 4)) ==
Convert.ToInt32(dEnd.Substring(6, 4)))
        {
            if (Convert.ToInt32(dStart.Substring(3, 2)) <
Convert.ToInt32(dOriginal.Substring(3, 2)) &&
                Convert.ToInt32(dOriginal.Substring(3, 2)) <
Convert.ToInt32(dEnd.Substring(3, 2)))
            { return true; }
            else
            {
                if (Convert.ToInt32(dStart.Substring(3, 2)) ==
Convert.ToInt32(dOriginal.Substring(3, 2)) &&
                    Convert.ToInt32(dOriginal.Substring(3, 2)) ==
Convert.ToInt32(dEnd.Substring(3, 2)))
                {
                    if (Convert.ToInt32(dStart.Substring(0, 2)) <=
Convert.ToInt32(dOriginal.Substring(0, 2)) &&
                        Convert.ToInt32(dOriginal.Substring(0, 2)) <=
Convert.ToInt32(dEnd.Substring(0, 2)))
                    { return true; }
                    else
                    { return false; }
                }
            }
            else
            {
                if (Convert.ToInt32(dStart.Substring(3, 2)) ==
Convert.ToInt32(dOriginal.Substring(3, 2)))
                {
                    if (Convert.ToInt32(dStart.Substring(0, 2))
<= Convert.ToInt32(dOriginal.Substring(0, 2)))
                    { return true; }
                    else
                    { return false; }
                }
            }
            else
            {
                if (Convert.ToInt32(dEnd.Substring(3, 2)) ==
Convert.ToInt32(dOriginal.Substring(3, 2)))
                {
                    if (Convert.ToInt32(dEnd.Substring(0, 2))
>= Convert.ToInt32(dOriginal.Substring(0, 2)))

```

```

        { return true; }
        else
        { return false; }
    }
    else
    { return false; }
}
}
}
else
{
    if (Convert.ToInt32(dStart.Substring(6, 4)) ==
Convert.ToInt32(dOriginal.Substring(6, 4)))
    {
        if (Convert.ToInt32(dStart.Substring(3, 2)) <=
Convert.ToInt32(dOriginal.Substring(3, 2)) &&
Convert.ToInt32(dStart.Substring(0, 2)) <=
Convert.ToInt32(dOriginal.Substring(0, 2)))
        { return true; }
        else
        { return false; }
    }
    else
    {
        if (Convert.ToInt32(dEnd.Substring(6, 4)) ==
Convert.ToInt32(dOriginal.Substring(6, 4)))
        {
            if (Convert.ToInt32(dEnd.Substring(3, 2)) >=
Convert.ToInt32(dOriginal.Substring(3, 2)) &&
Convert.ToInt32(dEnd.Substring(0, 2)) >=
Convert.ToInt32(dOriginal.Substring(0, 2)))
            { return true; }
            else
            { return false; }
        }
        else
        { return false; }
    }
}
}
}

private void MainForm_Load(object sender, EventArgs e)
{
    splitIn.Panel2Collapsed = true;
    splitIn.Panel1Collapsed = false;
    buttonMain.Enabled = false;

    #region type_auto Load
    dataAdapter.SelectCommand.CommandText = "select type_auto_name
from type_auto order by id_type_auto asc";
    System.Data.DataTable dtTAuto = new
System.Data.DataTable("type_auto");
    dataAdapter.Fill(dtTAuto);
    comboTypeAuto.Items.Add("Bce");
    comboTypeAuto.Text = comboTypeAuto.Items[0].ToString();
    for (int i = 0; i < dtTAuto.Rows.Count; i++)
    {
        comboTypeAuto.Items.Add(dtTAuto.Rows[i][0].ToString());
    }
    #endregion

    #region Temp Auto Clear

```

```

        OleDbCommand cmd = new OleDbCommand("update [auto] set status=0
where status=2", con);
        con.Open();
        cmd.ExecuteNonQuery();
        con.Close();
        #endregion

        DateTimeLabel.Text = "Дата: " + DateTime.Now.ToShortDateString()
+ "    Время: " + DateTime.Now.ToShortTimeString();
        Vaqt.Enabled = true;

        dateEnd.Text = DateTime.Now.ToShortDateString();
        dateStart.Text = "01." + DateTime.Now.Month.ToString() + "." +
DateTime.Now.Year.ToString();
    }

    private void buttonCome_Click(object sender, EventArgs e)
    {
        splitIn.Panel2Collapsed = false;
        splitIn.Panel1Collapsed = true;
        currentState = (sender as
System.Windows.Forms.Button).Tag.ToString();
        StatusLabel.Text = "Количество машины: 0 шт.";
        dataGridCome.Columns.Clear();
        dataGridAuto.Rows.Clear();
        switch (currentState)
        {
            case "naklad":
                buttonIncome.Enabled = false;
                buttonOutcome.Enabled = true;
                buttonMain.Enabled = true;
                this.Text = "Приход";
                FormNameLabel.Text = "Приход";

                #region Column add
                dataGridCome.Columns.Add("column0", "№");
                dataGridCome.Columns.Add("column1", "Дата накладной");
                dataGridCome.Columns.Add("column2", "№ накладной");
                dataGridCome.Columns.Add("column3", "Дата счет фактуры");
                dataGridCome.Columns.Add("column4", "№ счет фактуры");
                dataGridCome.Columns.Add("column5", "От кого");
                dataGridCome.Columns.Add("column6", "Получатель");
                dataGridCome.Columns.Add("column7", "Тип автомобиля");
                dataGridCome.Columns.Add("id_naklad", "id_naklad");
                dataGridCome.Columns[0].AutoSizeMode =
DataGridViewAutoSizeColumnMode.ColumnHeader;
                dataGridCome.Columns[1].AutoSizeMode =
DataGridViewAutoSizeColumnMode.DisplayedCells;
                dataGridCome.Columns[2].AutoSizeMode =
DataGridViewAutoSizeColumnMode.DisplayedCells;
                dataGridCome.Columns[3].AutoSizeMode =
DataGridViewAutoSizeColumnMode.DisplayedCells;
                dataGridCome.Columns[4].AutoSizeMode =
DataGridViewAutoSizeColumnMode.DisplayedCells;
                dataGridCome.Columns[5].AutoSizeMode =
DataGridViewAutoSizeColumnMode.Fill;
                dataGridCome.Columns[6].AutoSizeMode =
DataGridViewAutoSizeColumnMode.Fill;
                dataGridCome.Columns[7].AutoSizeMode =
DataGridViewAutoSizeColumnMode.DisplayedCells;
                dataGridCome.Columns[8].Visible = false;
                #endregion

                FNAKLADLOAD();
            }
        }
    }

```



```

        break;
    case "selldoc":
        buttonOutcome.Enabled = false;
        buttonIncome.Enabled = true;
        buttonMain.Enabled = true;
        this.Text = "Расход";
        FormNameLabel.Text = "Расход";

        #region Column add
        dataGridCome.Columns.Add("column0", "№");
        dataGridCome.Columns.Add("column1", "Дата документа");
        dataGridCome.Columns.Add("column2", "№ документа");
        dataGridCome.Columns.Add("column3", "Получатель");
        dataGridCome.Columns.Add("column4", "Адрес");
        dataGridCome.Columns.Add("column5", "Телефон");
        dataGridCome.Columns.Add("id_selldoc", "id_selldoc");
        dataGridCome.Columns[0].AutoSizeMode =
DataGridViewAutoSizeColumnMode.ColumnHeader;
        dataGridCome.Columns[1].AutoSizeMode =
DataGridViewAutoSizeColumnMode.Fill;
        dataGridCome.Columns[2].AutoSizeMode =
DataGridViewAutoSizeColumnMode.Fill;
        dataGridCome.Columns[3].AutoSizeMode =
DataGridViewAutoSizeColumnMode.Fill;
        dataGridCome.Columns[4].AutoSizeMode =
DataGridViewAutoSizeColumnMode.Fill;
        dataGridCome.Columns[5].AutoSizeMode =
DataGridViewAutoSizeColumnMode.Fill;
        dataGridCome.Columns[6].Visible = false;
        #endregion

        FSELDOCLOAD();
        break;
    case "main":
        splitIn.Panel1Collapsed = false;
        splitIn.Panel2Collapsed = true;
        buttonMain.Enabled = false;
        buttonIncome.Enabled = true;
        buttonOutcome.Enabled = true;

        #region Dealer Load
        dataAdapter.SelectCommand.CommandText = "select dealer
from dealer";

        DataTable dt = new DataTable("dt");
        dataAdapter.Fill(dt);
        #endregion

        StatusLabel.Text = dt.Rows[0][0].ToString();
        this.Text = "Главное окно";
        FormNameLabel.Text = "Главное окно";
        break;
    }
}

private void dataGridCome_CellClick(object sender,
DataGridViewCellEventArgs e)
{
    dataGridAuto.Rows.Clear();
    if (dataGridCome.Rows.Count > 0)
    {
        if (dataGridCome.SelectedRows[0].Index <
dataGridCome.Rows.Count)
        {

```

```

        string id =
dataGridCome.Rows[dataGridCome.SelectedRows[0].Index].Cells["id_" +
currentState].Value.ToString();
        dataAdapter.SelectCommand.CommandText = "select * from
[auto] where id_" + currentState + "=" + id;
        FLOAD();
    }
    else
    {
        StatusLabel.Text = "Количество машины: 0 шт.";
    }
}

}

#region FULLLOAD Method

private void FULLLOAD()
{
    dataGridCome.Rows.Clear();
    dataGridAuto.Rows.Clear();
    switch (currentState)
    {
        case "naklad":
            FNAKLADLOAD();
            break;
        case "selldoc":
            FSELDOCLOAD();
            break;
    }
}

private void AddButtonItem_Click(object sender, EventArgs e)
{
    switch (currentState)
    {
        case "naklad":
            NakladForm NakladF = new NakladForm();
            NakladF.ShowDialog();
            break;
        case "selldoc":
            SellDocument SelldocF = new SellDocument();
            SelldocF.ShowDialog();
            break;
    }
    FULLLOAD();
}

private void DeleteButtonItem_Click(object sender, EventArgs e)
{
    if (dataGridCome.Rows.Count != 0)
    {
        if (DialogResult.OK == MessageBox.Show("Are you sure ?",
"Info", MessageBoxButtons.OKCancel, MessageBoxIcon.Information))
        {
            OleDbCommand cmd = new OleDbCommand("", con);
            con.Open();
            cmd.CommandText = "delete from " + currentState + " where
id_" + currentState + "=" + dataGridCome.SelectedRows[0].Cells["id_" +
currentState].Value.ToString();
            cmd.ExecuteNonQuery();
            cmd.CommandText = "update [auto] set [status]=0,
id_selldoc=null, id_customer=null where id_selldoc=" +
dataGridCome.Rows[dataGridAuto.SelectedRows[0].Index].Cells["id_selldoc"].Val
ue.ToString();

```

```

        cmd.ExecuteNonQuery();
        con.Close();
    }
    FULLLOAD();
    EditStatus();
}

private void EditButtonItem_Click(object sender, EventArgs e)
{
    switch (currentState)
    {
        case "naklad":
            NakladForm NakladF = new NakladForm();
            NakladF.update = true;
            NakladF.groupDoc.Tag =
dataGridCome.SelectedRows[0].Cells["id_naklad"].Value.ToString();
            NakladF.ShowDialog();
            break;
        case "selldoc":
            SellDocument SelldocF = new SellDocument();
            SelldocF.update = true;
            SelldocF.groupDoc.Tag =
dataGridCome.SelectedRows[0].Cells["id_selldoc"].Value.ToString();
            SelldocF.ShowDialog();
            break;
    }
    FULLLOAD();
}

private void date_ValueChanged(object sender, EventArgs e)
{
    bool change = false;

    #region StartEndDate
    if (Convert.ToInt32(dateStart.Text.Substring(6, 4)) >
Convert.ToInt32(dateEnd.Text.Substring(6, 4)))
    {
        change = true;
    }
    else
    {
        if (Convert.ToInt32(dateStart.Text.Substring(6, 4)) ==
Convert.ToInt32(dateEnd.Text.Substring(6, 4)))
        {
            if (Convert.ToInt32(dateStart.Text.Substring(3, 2)) >
Convert.ToInt32(dateEnd.Text.Substring(3, 2)))
            {
                change = true;
            }
            else
            {
                if (Convert.ToInt32(dateStart.Text.Substring(3, 2))
== Convert.ToInt32(dateEnd.Text.Substring(3, 2)))
                {
                    if (Convert.ToInt32(dateStart.Text.Substring(0,
2)) > Convert.ToInt32(dateEnd.Text.Substring(0, 2)))
                    {
                        change = true;
                    }
                }
            }
        }
    }
}

```

```

        if (change)
        {
            switch ((sender as DateTimePicker).Name)
            {
                case "dateStart":
                    dateEnd.Text = dateStart.Text;
                    break;
                case "dateEnd":
                    dateStart.Text = dateEnd.Text;
                    break;
            }
        }
        #endregion

        FULLLOAD();
    }

    private void comboTypeAuto_TextChanged(object sender, EventArgs e)
    {
        FULLLOAD();
    }

    #endregion

    private void time_Tick(object sender, EventArgs e)
    {
        DateTimeLabel.Text = "Дата: " + DateTime.Now.ToShortDateString()
+ " Время: " + DateTime.Now.ToShortTimeString();
    }

    private void comboTypeAuto_KeyPress(object sender, KeyPressEventArgs
e)
    {
        e.Handled = true;
    }

    #region MenuItemClicks

    private void priceMenuItem_Click(object sender, EventArgs e)
    {
        PriceApp.PriceForm PriceF = new PriceApp.PriceForm();
        PriceF.ShowDialog();
    }

    private void ReferencesMenuItem_Click(object sender, EventArgs e)
    {
        little.LittleForm ReferenceF = new little.LittleForm();
        ReferenceF.ShowDialog();
    }

    private void IncomeMenuItem_Click(object sender, EventArgs e)
    {
        buttonCome_Click(buttonIncome, e);
    }

    private void OutcomeMenuItem_Click(object sender, EventArgs e)
    {
        buttonCome_Click(buttonOutcome, e);
    }

    private void CurrencyRateMenuItem_Click(object sender, EventArgs e)
    {
        Pricelist.RatesForm RateF = new Pricelist.RatesForm();
        RateF.ShowDialog();
    }

```

```

    }

    private void InfoDealerMenuItem_Click(object sender, EventArgs e)
    {
        DealerForm DealerF = new DealerForm();
        DealerF.ShowDialog();
    }

    private void CustomersMenuItem_Click(object sender, EventArgs e)
    {
        CListForm CListF = new CListForm();
        CListF.ShowDialog();
    }

    private void ExitMenuItem_Click(object sender, EventArgs e)
    {
        Application.Exit();
    }

    #endregion
}

```