

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Коммерциялық емес акционерлік қоғамы  
АЛМАТЫ ЭНЕРГЕТИКА ЖӘНЕ БАЙЛАНЫС УНИВЕРСИТЕТІ

«Компьютерлік технологиялар» кафедрасы

«Қорғауға жіберілді»  
Кафедра меңгерушісі  
ф.-м.ғ.д., проф. Құралбаев З.Қ.

(КОЛЫ)

« \_\_\_\_\_ » \_\_\_\_\_ 2014 ж.

ДИПЛОМДЫҚ ЖОБА

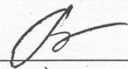
Тақырыбы: «Айырбастау пунктіндегі ақпараттар қозғалысын бақылау  
жүйесін құру»  
5В070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету»  
мамандығы бойынша

Орындаған Дайрабаева Малика Серікқызы тобы: ВТк-10-1

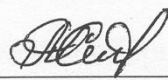
Жетекші доцент, ф.-м.ғ.к. Аманбаев А.А.

Кеңесшілер :

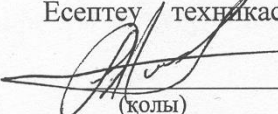
Экономикалық бөлім бойынша :

 доцент Боканова Г.Ш.  
« 20 » 05 2014ж.  
(КОЛЫ)

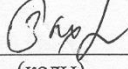
Өмір тіршілігі қауіпсіздігі бойынша:

 т.ғ.к., аға оқытушы Муташева Г.С.  
« 23 » 04 2014 ж.  
(КОЛЫ)

Есептеу техникасын қолдану бойынша :

 доцент, ф.-м.ғ.к. Аманбаев А.А.  
« 22 » 05 2014 ж.  
(КОЛЫ)

Мөлшер бақылаушы:

 аға оқытушы Рахимжанова З.М.  
« 28 » 05 2014 ж.  
(КОЛЫ)

Пікір жазушы :

АТУ, АТ кафедрасының меңгерушісі : т.ғ.д., проф. Заурбеков Н.С.  
« \_\_\_\_\_ » \_\_\_\_\_ 2014 ж.  
(КОЛЫ)

Алматы 2014

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Коммерциялық емес акционерлік қоғамы  
АЛМАТЫ ЭНЕРГЕТИКА ЖӘНЕ БАЙЛАНЫС УНИВЕРСИТЕТИ

«Ақпараттық технологиялар» факультеті  
«Есептеу техникасы және бағдарламалық қамтамасыз ету» мамандығы  
«Компьютерлік технологиялар» кафедрасы

жобаны орындауға берілген

ТАПСЫРМА

Студент

Дайрабаева Малика Серікқызына

Жоба тақырыбы «Айырбастау пунктіндегі ақпараттар қозғалысын бақылау жүйесін құру»

ректордың «24» қыркүйек №115 бұйрығы бойынша бекітілген.

Аяқталған жұмысты тапсыру мерзімі: «26» мамыр 2014 ж.

Жобаға бастапқы деректер (талап етілетін жоба нәтижелерінің параметрлері және нысанның бастапқы деректері):

MS SQL Server дерекқоры арқылы басқару жүйесін  
және MS Visual Studio бағдарламалық орта-  
сын пайдалану арқылы айырбастау нүктесін  
дегі ақпараттар қозғалысын бақылау  
жүйесін құру қарастырылған.

Диплом жобасындағы әзірленуі тиіс сұрақтар тізімі немесе диплом жобасының қысқаша мазмұны:

- Автоматтастырылған нүктені орнатуға асықпалық
- Автоматтық оқиды, дерекқорлар (DQ), дерекқорлар-  
ды басқаруды жүйесі (DQ БЖ) түсініктері
- Нүктені дерекқоры арқылы және жүзеге асыру  
жобасын құрудың маңызы, пайдаланушы ин-  
терфейсі арқылы; дерекқормен байланысты  
орнату
- Техникалық - жобаның маңызы
- Әдіс-тәсілдерді сұрақтар мен қысқаша мазмұны  
жүйесін құру.

Сызба материалдарының (міндетті түрде дайындалатын сызуларды көрсету) тізімі:

- *Функциональ ЕК-диаграммасы*  
*насанға бағдарлы әрекеттің қосымшасы*  
*нұсқа істеу қабілетімен, функционал-*  
*дағи көрсететін графикалық нәтижелерін*  
*дағ.*

Негізгі ұсынылатын әдебиеттер:

1. Алексей Вишневыский. SQL Server . Эффективная работа. – Санкт-Петербург, 2009. – 541с.
2. Полякова Л.Н. Основы SQL. Курс лекций: учеб. пособие. – Москва, 2004. – 368с.
3. Глушаков С.В., Ломотько Д.В. Базы данных: Учебный курс. – М.: ООО «Издательство АСТ», 2000. – 504с.
4. Агапов В.П. Основы программирования на языке C# . учеб.пособие.- Москва,2012.-128с.
5. Голицына О.Л., Попов И.И.Основы алгоритмизации и программирования. учеб.пособие.-Форум 2008.-432с.

Жоба тараулары бойынша кеңес берушілер және оның мерзімі:

Бөлім	Кеңесші	Мерзімі	Қолы
Негізгі бөлім	Аманбаев А.А.		
Тіршілік қауіпсіздігі	Муташева Г.С.		
Экономикалық бөлім	Боканова Г.Ш.		
Норма бақылаушы	Рахимжанова З.М.		
Есептеу техникасын қолдану	Аманбаев А.А.		



КЕСТЕСІ

№ р/с	Тарау аттары, әзірленетін сұрақтардың тізімі	Жетекшіге ұсыну мерзімдері	Ескерту
1	Автоматтандырылған жүйелі қимылдың алгоритмі (Ақпарат, құрылым, т.б.)	8.10.2013.	
2	Ақпараттық жүйелер, директори (ДЖ), директориға басқаруға жүйелері (ДЖБЖ) түйіндері.	12.12.2013.	
3.	Жүйелік директориға мабалау және жүйелі алуға, мабалау, директориға БЖ мабалау мабалау.	22.01.2014	
4.	Директориға құру, пайдалануға итерділік мабалау, директориға байланысты ермату.	14.03.2014	

Тапсырманың берілген уақыты « 5 » Қараша 2014 ж.

Кафедра меңгерушісі

ф.-м.ғ.д., проф. Құралбаев З.Қ.

Жоба жетекшісі

доцент, ф.-м.ғ.к. Аманбаев А.А.

Орындалатын тапсырманы қабылдаған студент

Дайрабаева Малика Серікқызы

Андатпа

Бұл дипломдық жұмыста айырбастау пунктіндегі ақпараттар қозғалысының бақылау жүйесін автоматтандыру, дерекқор негізінде клиенттік программасын жоспарлау қарастырылады. Жобаның басты мақсаты – айырбастау пунктіндегі ақпараттар қозғалысын бақылау жүйесін құру. Ақпараттық жүйені MS SQL 2008 Server ДҚБЖ (дерекқорларды басқару жүйесінде) жасалған дерекқор және MS Visual Studio 2012 бағдарламалық ортасында құрылған клиенттік қосымша құрайды.

Өміртіршілік қауіпсіздігі бөлімінде жұмыс бөлмесіндегі жасанды жарықтандыру есептері келтірілген.

Экономикалық бөлімде енгізілген жүйеден алынған экономикалық пайданы есептеу жүргізілген.

### **Аннотация**

В данном дипломном проекте рассматривается автоматизация решение наблюдение потока информации в обменном пункте, на основе база данных планирование клиентской программы. Цель данного проекта - разработка системы наблюдения информационного потока обменного пункта. Информационная система представляет собой базу данных, разработанную на СУБД MS SQL2008 Server и клиентское приложение, созданное в среде программирования MS Visual Studio 2012.

В части по безопасности жизнедеятельности представлены расчеты искусственного освещения в рабочем помещении.

В экономической части произведен расчет полученной экономической выгоды от внедряемой системы.

### **Annotation**

In this degree project automation the decision supervision of a flow of information in exchange point, on a basis a database planning of the client program is considered. The purpose of this project - development of the system of supervision of information stream of exchange point. Information system represents the database developed on MS SQL2008 Server DBMS and the client application, created in the environment of MS Visual Studio 2012 programming.

In part on health and safety calculations of artificial lighting in the working room are presented.

In economic part calculation of the received economic benefit from introduced system is made.

## Мазмұны

Кіріспе .....	8
1 Автоматтандырылған жұмыс орнының (АЖО) арналуы мен құрылымы .....	9
1.1 Автоматтандырылған жұмыс орнының(АЖО)құрылымы .....	10
1.2 АЖО-ға қойылатын жалпы талаптар .....	11
1.3 АЖО-ды жобалау сатылары.....	12
2 Ақпараттық жүйелер. Дерекқорлар (ДҚ). Дерекқорларды басқарудың жүйелері (ДҚБЖ).....	13
2.1 Ақпараттық жүйелер туралы қысқаша мағлұматтар.....	13
2.2 Деректер қорын жобалау .....	17
2.3 SQL тілі туралы негізгі деректер.....	22
3 Дерекқорды жобалау және оны іске асыру .....	28
3.1 ДҚ құрудың бастапқы кезеңі .....	28
3.1.1 Айырбастау пунктiнiң дерек қоры жұмысының жалпы сипаттамасы.....	28
3.1.2 Мәселелер мен шектеулер.....	28
3.1.3 Дерекқордағы шешiлетiн мәелелер.....	28
3.1.4 Дерекқордағы функционалдығын сипаттау .....	28
3.2 Концептуалды жобалау .....	28
3.2.1 Қажеттi ақпарат пайдаланушылардың талабы.....	29
3.2.2 Дерекқор және қосымшаны жобалау құралдарын таңдаудың негiздемесi. Дерекқорларды басқару жүйесi.....	29
3.2.3 ER моделiн құру .....	30
3.2.4 Бизнес ережелер .....	31
3.2.5 Реляционды қатынастарды құрастыру.....	33
3.3 Логикалық жобалау.....	35
3.3.1 Кестелер құру .....	35
3.3.2 ДҚ диаграммасы.....	36
3.4 Физикалық жобалау .....	37
3.4.1 Қосымша интефейсiн және оның функционалдығын сипаттау .....	37
4 Тiршiлiк қауiпсiздiгi.....	44
4.1 Жасанды жарықтандыруды жүйесiн есептеу.....	44
4.1.1 Ақша айырбастау пунктiнiң еңбек ету шартының талдауы .....	44
4.2 Жарық көзiн таңдау.....	44
4.3 Жасанды жарықтандыруды есептеу.....	49
4.4 Меншiктi қуат әдiсiмен жарықтандыруды есептеу.....	51

4.5 Өнеркәсіптегі жасанды жарықтандыру .....	51
5 Техника-экономикалық негізделуі .....	53
5.1 Жоба сипаттамасы және қажеттілігінің негіздеу.....	53
5.2 Маркетингтік стратегия.....	53
5.3 Бағдарламамен қамтамасыз етудегі еңбек сыйымдылығын есептеу.....	54
5.4 Бағдарламалық қамсыздандыру шығынының есебі .....	56
5.5 Бағдарлама өнімін сатып алуға кеткен бір жолғы шығындар есебі .....	60
5.6 Игеру саласындағы жылдық бір жолғы шығындар есебі .....	61
5.7 Ақпараттық жүйе енгізуден үнем мен табыс мөлшерінің есебі.....	62
5.8 Салыстырмалы экономикалық тиімділіктің көрсеткіштерін есептеу .....	63
5.9 Динамикалық көрсеткіштер негізінде жобаны өткізуде экономикалық тиімділігін бағалау .....	64
Қорытынды .....	67
Әдебиеттер тізімі.....	68
А қосымшасы.....	69
Б қосымшасы .....	71

## **Кіріспе**

Бұл дипломдық жұмыста айырбастау пунктіндегі ақпараттар қозғалысының бақылау жүйесін автоматтандыру, дерекқор негізінде клиенттік программасын жоспарлау қарастырылады.

Бұл программа ақша айырбастау пункттеріне көмек көрсету үшін арналған. Жалпы айырбастау пункттері үшін де, қолданушыларға да ыңғайлы уақытты үнемдеу мақсатымен, жұмыс барысында мүмкін болатын қателерді азайту үшін қолданылады.

Айырбастау пункттеріндегі жұмыс негізінен администратор мен кассир, қолданушы туралы ақпаратты жүйелеумен байланысты болады. Сондықтан, бұл бағдарлама жұмысы ыңғайлы, әрі тез болу үшін, барлық деректерді жүйелі түрде сақтау, қажетті ақпаратқа тез қатынас алу, сонымен қатар жаңа деректерді ыңғайлы түрде енгізу мүмкіндіктері қарастырылған болу керек.

Дерекқор құру үшін алдымен зерттелетін облысының талдауы жасалады. Келесі кезеңде айырбастау орны үшін администратор ,кассирлерге арналған қажетті деректерді сақтайтын логикалық байланысқан кестелер құрылып, олардың негізінде жүйенің инфологиялық моделі алынады. Инфологиялық модель MS SQL Server-жүйесінде жасалады. Айырбастау пунктiнiң дерекқоры MS SQL Server дерекқорларды басқару жүйесінде құрылғаннан кейiн, пайдаланушылардың барлық сұраныстарын қанағаттандыратын клиенттік қосымша жобаланады. Клиенттік бағдарлама Microsoft Visual Studio 2012 ultimate программалық ортасында iске асырылады.

Бұл бағдарлама ыңғайлы әрі уақытты үнемді пайдалануға мүмкіндік береді, сонымен қатар, аз уақыт ішінде бағдарламаны игеріп, оңай пайдалануға

мүмкіндік береді.



## **1 Автоматтандырылған жұмыс орнының (АЖО) арналуы мен құрылымы**

Автоматтандырылған жұмыс орны (АЖО) немесе шетел терминологияда «жұмыс станция» (work-station) бұл анықталған функцияларды автоматтандыру үшін қажетті құралдармен жабдықталған белгілі бір мамандықтағы қолданушы маманның орны. Мұндай құралдар, керек жағдайда басқа қосалқы электронды құрылғылармен, атап айтқанда, дискілік жинақтауыштармен, басу құрылғылармен, оптикалық оқитын құрылғылармен немесе штрих кодасын оқитын, графика құрылғыларымен, басқа АЖО-мен және басқа жергілікті есептеу желілерімен т.б. түйіндесетін құрылғылармен толықтырылатын әдетте дербес компьютер болып табылады.

Таралған ИС концепциясына сәйкес олардың ішінде ақпарат көлемінің төменнен жоғары қарай берілуі ИС дың ұйымдық иерархияда жоғарыдағы деңгейлердегі оның қаншалықты қажеттілігіне байланысты болады. Бұл жерде ақпаратты өңдеу нәтижелерінің басым бөлігі және бастапқы деректер деңгейлердің жергілікті дерекқорларда (ДБ) сақталған жөн. Осы таралған басқару идеясын жүзеге асыруда әрбір ИС деңгейі үшін мамандардың автоматтандырылған жұмыс орындарын құруды талап етті.

АЖО-ның негізгі арналуы деп жұмыс орындарындағы ақпараттың орталықтандырылмаған өңдеуді, «өзінің» сәйкес келетін дерекқорларын қолданумен қатар жергілікті АЖО-мен ДК желілеріне, ал кейде қуатты ЭЕМ-лар бар ауқымды есептеу тораптарына ену мүмкіндігімен қолдануды атауға болады.

АЖО-да шешілетін мәселелерді шартты түрде ақпараттық және есептеуіш деп жіктеуге болады.

Ақпараттық мәселелерге ақпаратты кодтау, классификациялау (жіктеу), жинау, құрылымдық ұйымдастыру, түзету, сақтау жатады. Ақпараттық мәселелер жиі түрде арифметикалық пен мәтіндік сипаттағы және байланыстағы қарапайым есептеу мен логикалық процедуралардан тұрады. Ақпараттық мәселелер әдетте күрделі болып табылады және мамандардың жұмыс уақытының көп бөлігін алады.

Есептеу мәселелер формальдандырылатын және жартылай формальдандырылатын болады. Формальдандырылатын мәселелер формальды алгоритм негізінде шешіледі және екі топқа бөлінеді: тура есептеу мәселелер мен математикалық модельдер негізіндегі мәселелер. Тура есептеу мәселелер қарапайым алгоритмдер көмегімен шешіледі. Одан күрделі мәселелерді шешу үшін әртүрлі математикалық модельдерді қолдану талап етіледі. Соңғы кезде жартылай формальданатын мәселелерді шешуге арналған сематикалық деп аталатын амалдарды жетілдіруге аса назар аударылады. Мұндай мәселелер жиі түрде экономикалық объектілерді оперативті басқару барысында, әсіресе толық емес ақпарат жағдайында шешімді қабылдау кезінде туады.

## 1.1 Автоматтандырылған жұмыс орнының (АЖО) құрылымы

АЖО бұл ұйымдастыру, функционалды және қамтамасыз ету бөліктерінен тұратын кешенді жүйе.

Ұйымдастыру бөлігі АЖО қолданушылардың функцияларын бөліп таратуға арналған ұйымдастыру құрлымынан, сонымен қатар мамандарды даярлау, дамыту мен әкімшілдеу тәсілдерінің жиынтығынан тұрады. Әкімшілдеуге жұмысты жоспарлау, тіркеу, бақылау, сараптау, реттеу, АЖО қолданушыларының құқықтарын мен міндеттерін құжатты түрде рәсімдеуді жатқызады.

АЖО-дың функционалды бөлігі бұл оперативті және перспективалық жоспарлау мәселелердің шешімін, техника-экономикалық көрсеткіштердің тіркеуін мен сараптауын қамтамасыз ететін экономика-математикалық тәсілдер кешені. Кешен құрылымы АЖО ның анықталған функцияларын жүзеге асыратын жүйенің біршама дербес ішжүйеден тұрытыны болжамдалады.

Ішжүйелердің ерекшеленуі жұмыс орынның функционалды специализациясы мен ұйымдық құрылымдық есеппен функционалды ұйымдық белгісі бойынша жүргізіледі.

АЖО-ның қамтамасыз ету бөлігі:

- функционалды ішжүйелердің ақпараттық қамтамасыз етуі (АҚЕту);
- техникалық құралдар кешені (ТҚК);
- ішжүйелер функцияларын жүзеге асырудың программалық құралдары.

АҚЕту АЖО бұл құжаттардың, ақпарат массивтерінің (ДҚ) және кодалаудың стандартталған және бірыңғай формалар жиынтығы.

АЖО-да қолданылатын ақпарат тиянақтылық, оның арналуы және қозғалу бағыты бойынша жіктеледі. Тұрақтылық бойынша ақпарат шартты тұрақты және айнымалы тұрақты, арналуы бойынша нормативті-анықтамалық, жоспарлау, есептемелік, оперативті-өндірістік (директивті-оперативті және оперативті) деп бөлінеді. Қозғалу бағыты бойынша ақпарат кіріс, шығыс және аралық деп бөлінеді.

АЖО-ның техникалық құралдар кешеніне (ТҚК) ақпаратты жинау, жеткізу, өңдеу және сақтау процестердің жүзеге асырылуын қамтамасыз ететін құрылғылар жиынтығы кіреді.

Біртекті операцияларды орындайтын құрылғылар техникалық құралдардың функционалды топтарын құрайды. Ішжүйелердің функцияларды жүзеге асырудың программалық құралдарына біріншіден адам-машина АЖО жүйесінде қатынасудың программалық тір құралдарына арнап, «жұмыс ортасын» қалыптастыратын операциялық жүйелерді жатқызуға болады. АЖО прогораммалық тіл құралдары келесімен қамтамасыз етеді:

- АЖО-ның функционалды мәселелерін шешу;
- АЖО қолданушылар жағынан мәселелердің шешімін ыңғайлы және түсінікті басқару – мәселелер интерфейстері.

## 1.2 АЖО-ға қойылатын жалпы талаптар

АЖО-ға қойылатын жалпы талаптар келесідей:

- жүйелілік;
- икемділік;
- тұрақтылық;
- эффективтілік (тиімділік).

Жүйелілік. АЖО жұмыс орынның функционалды арналуымен анықталатын құрылымы бар жүйе деп қарастырғаны жөн.

Икемділік. Элементтердің стандартталуы мен құрылудың модульдігінің арқасында жүйе мүмкін болатын қайта құруларға икемді болу керек.

Тұрақтылық. Ішкі мен сыртқы қоздыратын факторлардың әсеріне қарамастан жүйе негізгі функцияларды орындауы керек.

Эффективтілік. АЖО-ды жүйенің құрылуы мен қолдануына кеткен шығындарына апаратын жоғарыда келтірілген қағидалардың жүзеге асырылу деңгейінің интегралды көрсеткіші де қарастырғаны жөн.

АЖО бұл әрдайым арнаулы сипатты жүйе. Нақты бір маманға әкімші, экономист, инженер, конструктор, жобалаушы, сәулетші, дизайнер, дәрігер, ұйымдастырушы, зерттеуші, кітапханашы, мұражай қызметкері және т.б. арналған техникалық құралдар мен программалық қамтамасыз етудің жиынтығы.

АЖО-ды ендіру компьютер ұйытқы болатын ақпаратты өңдеудің машиналық құралдар мен адам арасындағы функциялар мен жүктеменің дұрыс бөліп тарату жағдайында қалаған эффектті беруі мүмкін.

АЖО бұл тек еңбектің өнімділігін мен басқарудың эффективтілігін жоғарлатуының құралы ғана емес, сонымен қатар мамандардың қоғамдық қолайшылығының құралы.

АЖО жеке (дербес) , топтық, коллективті болуы мүмкін. Топтық пен коллективті АЖО-ға қатысты олардың ДҚ-дің жүйе ретінде эффективті функционалдау мақсатында мамандарға (коллективке) әкімшілдеудің нақты функцияларын анықтау қажет. АЖО адам машиналық жүйе ретінде ашық, икемді, үнемі даму мен жетілдірілуіне бейімді болу қажет.

Мұндай жүйеде келесі қамтамасыз етілу керек:

- мамандардың ақпаратты өңдеу машина құралдарына максималды жақындығы;
- диалогты тәртіпте жұмыс жасау;
- эргономика талаптарына сәйкестілік;
- компьютердің жоғары икемділігі;
- ескілік процестердің максималды автоматтандырылуы;
- жекелік жағдайда мамандардың творчестволық белсенділігін арттыратын, ал кейін жүйенің дамуына әкелетін мамандардың еңбек жағдайларына моральды түрде қанағат болуы;
- мамандардың өзін-өзі оқыту мүмкіндігі.

### **1.3 АЖО-ды жобалау сатылары**

АЖО-ды дұрыс ұйымдастыру, құрастыру және енгізу мақсатында оның жобалау процесін келесі сатыларға жіктейді:

- бар басқару жүйесін зерттеу мен сараптама жасау;
- АЖО құруының техникалық тапсырманы құрстыру;
- АЖО-ды техникалық жобалау;
- АЖО-ды жұмыстық жобалау.

## **2 Ақпараттық жүйелер. Дерекқорлар (ДҚ). Дерекқорларды басқарудың жүйелері (ДҚБЖ)**

### **2.1 Ақпараттық жүйелер туралы қысқаша мағлұматтар**

Көптеген мәселелер шешімінің негізінде ақпаратты өңдеу жатыр. Ақпарат өңдеуін ыңғайлату үшін ақпараттық жүйелер (АЖ) құрылады.

Ақпараттық жүйе – қандай да бір адам қызметіне арналған ақпараттық моделі болып табылады. Ол жүйе төмендегідей ақпараттық үдерістерді қамтуы керек:

- сақтау;
- ақпарат ауысу;
- ақпаратты түрлендіру.

Қазіргі заманда күннен күнге ақпараттық жүйелер, қызметтердің әр саласына маңыздылығы артуда.

Автоматтандырылған ақпараттық жүйелер бұл техникалық құралдарды, әсіресе ЭЕМ-ды қолданатын жүйелер. Ақпараттық жүйелердің көпшілігі автоматтандырылған болып табылады, сондықтан қысқарту үшін оларды жай АЖ деп атайды.

Ақпарат жүйесін қашанда дамып отырған жүйе болғандықтан оған сәйкес ақпарат жүйесі де дамып, жетілдіріп отыруы тиіс.

Жүйенің даму барысы зерттеу, талдау, жобалау, қолдануға ендіру және қолдау сатыларынан тұады. Бұл сатылар кейде, яғни прототипті әдіс қолданылғанда, бір уақытта қатар жүргізілуі де мүмкін.

Кең түсініктемеде АЖ анықтамасына ақпаратты өңдеудің кез келген жүйесі сәйкес келеді. Қолдану облысы бойынша АЖ-ді өндірісте, бітім беру саласында, денсаулық сақтау, ғылым, әскери істе, қоғамдық салада, саудада және т.б. қолданылатын жүйелер деп бөдуге болады. Мақсат функциялары бойынша АЖ-ді шартты түрде келесі негізгі топтарға бөлуге болады: басқару, ақпаратты-анықтамалық, шешімдерді қабылдауды қолдау тобы.

Кейде АЖ түсініктемесінің ең тар тұжырымдамасы қолданылады, яғни белгілі бір қолданбалы мәселені шешу үшін жұмысқа қатысатын аппараттық-программалық құралдар жиынтығы ретінде қарастырылады. Мысалы, ұйымда келесі тапсырмаларды орындауға арналған ақпараттық жүйелер бола алады: кадрлар мен материалдық-техникалық құралдарды есептеу, тапсырыс беруші мен орындаушылар арасындағы есеп, бухгалтерлік есеп.

Ақпараттық жүйенің эффективтілігі келесі үштіктен тұрады:

- дерекқорды жобалау және оны жүзеге асыру;
- қолданбалыларды жобалау мен іске асыру;
- әкімшілдік процедуралар.

Бұдан басқа ақпараттық жүйенің жұмыс жасау эффективтілігі көбінесе оның сәулетіне тәуелді: файл-серверлік және клиент-серверлік сәулетті ажыратады.

Қазіргі кезде клиент-сервер сәулеті перспективті болып тыбылады.

Бір немесе бірнеше дерекқорларға ұйымдастырылған өңделетін ақпараттың орталықтанған сақтау мен жинау функциялары жүзеге асырылған ақпараттық жүйенің түрі болып деректер банкі болып табылады.

Жалпы жағдайда деректер банкі келесі құрама бөліктерден тұрады: дерекқорлар (бірнеше), дерекқорларын басқару жүйелері, деректер сөздігі, әкімші, есептеу жүйесі мен қызмет ететін персонал.

Деректер банкінің ең маңызды бөлігі бұл қарастырылатын пән аумағындағы объектілердің күйін мен арабайланыстарын көрсететін есептеу жүйесінің жадында сақталатын арнайы ұйымдастырылған деректер жиынтығы болып келетін деректер қоры (ДҚ) болып табылады.

ДҚБЖ дерекқорларды құру шығару және қолдануға арналған қолданбалы программалардың дестесі мен тіл құралдардың жиынтығы болып келеді.

Дерекқормен жұмыс істеу үшін көп жағдайда тек ДҚБЖ-нің құралдарын қолдануға болады (мысалы, сұрау салу мен есептеме құрғанда). Қолданбалыларды әдетте квалификациясы жоқ қолданушының дерекқормен жұмысын ыңғайлатуын қамтамасыз ету үшін немесе ДҚБЖ интерфейсі қолданушыға ұнамаған жағдайында құрылады.

Іс жүзінде ДҚБЖ қолданушы мен ДҚ арасындағы делдал рөлін атқарады, ал қолданушының сұрау алуларын орындауға арналған осы қолданушы сұрау салуларын күрделі кодаға түрлендіреді. ДҚБЖ ДҚ-қолданатын қолданбалы программалардан өзінің күрделі ішкі құрылымын жасырайды. Қолданбалы программалар программистпен белгілі бір программалау тілінде жазылуы мүмкін (мысалы COBOL) немесе ДҚБЖ-нің сервистік программалар көмегімен жазылады.

ДҚБЖ бұл сізбен алынған коммерциялық программалық өнім екенін естен шығармаған жөн және оған қандай да өзгерістерді енгізуге мүмкіндігіңіз болмайды. Сондықтан ДҚ жобалауы туралы айтқанда, ДҚБЖ-нің программалық қамтамасыз етуін жобалауды емес, деректерді сақтау мен басқару үшін қолданылатын ДҚ құрылымының жобалауын қарастырамыз.

Дерекқор жобалауын аяқтағаннан кейін ДҚБЖ компьютерге ыңғайлы формаға келтірілген жобалаушының деректер құрылымының бейнеленуін қажетті барлық күрделі процедураларын басқаруды өз қолына алады.

Қолданба бұл ДҚ-ды қолданатын және кейбір пән облысынан ақпараттың өңдеуін автоматтандырылуын қамтамасыз ететін программа немесе программалар кешені.

Қолданбалар ДҚБЖ-нің ортасында да, одан тыс та ДҚ-ға қатынас құру құралдарын қолданатын, мысалы, Delphi немесе MS Visual Studio программалау жүйесі көмегімен құрастыра алады.

Қолданушылар дерекқормен жұмыс істейтін бағдарламалар қосымшалар деп аталады. Жалпы жағдайда бір дерекқормен әртүрлі бірнеше қосымшалар жұмыс істей алады. Мысалы, егер дерекқор бір кәсіпорынды үлгілесе, онда оның жұмысы үшін кадрлардың есепке алу ішкі жүйесін қамтамасыз ететін қосымша құрылуы мүмкін, басқа қосымша қызметкерлердің жалақысын



есептеуге арналады, үшінші қосымша складтың есепке алу ішкі жүйесі ретінде жұмыс атқарады, төртінші қосымша өндірістік процесстің жоспарлануына арналған. Бір дерекқормен жұмыс істейтін қосымшаларды қарастырғанда олар бір-бірімен параллельді түрде және бір-бірінен тәуелсіз де жұмыс істей алады деп болжайды, және дәл ДҚБЖ көптеген қосымшалардың жұмысын бір дерекқормен әрқайсысы дұрыс, бірақ басқа қосымшалардың енгізген өзгерістерін қабылдайтындай қамтамасыз етуге арналған.

"Клиент-сервер" технологиясы.

Файл-серверлердің кемшіліктері:

- есептеулер клиенттің машинасында жүргізілуі – өндірістік төмен;
- сервер мен клиент арасында файлдар мен деректердің жіберілуі – желіні жүктейді;
- қорғаныс төмен деңгейде;
- масштабталудың болмауы.

Құрылымы:

- клиент-сервер архитектурасы желінің және үлестірілген дерекқордың (корпоративті ДҚ және дербес ДҚ) бар болуын болжайды;
- КДҚ сервер-компьютерде орналасады;
- ДДҚ корпоративті ДҚ-ның клиенті болып табылатын бөлімшенің қызметкерлерінің компьютерлерінде орналасады.

Анықтауыш:

- компьютерлік желіде белгілі бір ресурстың сервері деп осы ресурсты басқаратын компьютер (бағдарлама) аталады;
- клиент – осы ресурсты қолданатын компьютер (бағдарлама);
- компьютерлік желінің ресурсы ретінде, мысалға, дерекқор, файлдық жүйелер, баспа қызметтері, почта қызметтері бола алады.

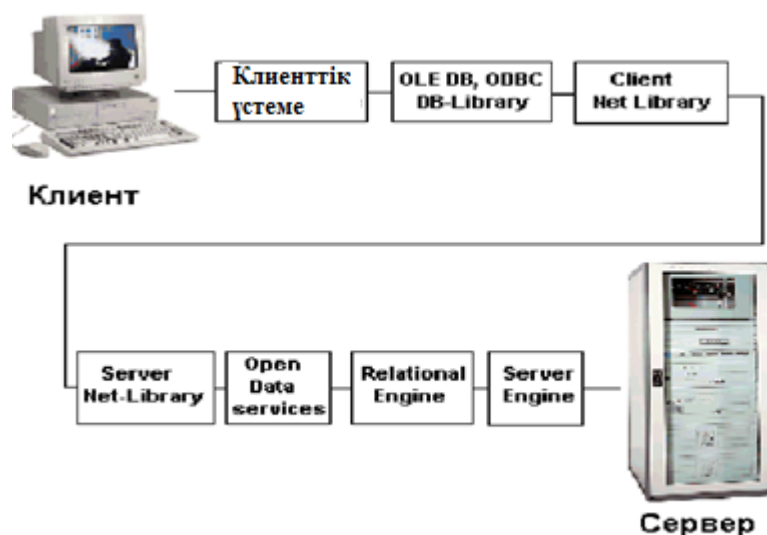
Артықшылықтары:

- орталықтандырылған сақтау, қызмет көрсету және жалпы корпоративті ақпаратқа коллективті қол жеткізудің дербес ақпаратпен жеке жұмыстың сәтті қосылуы. Корпоративті (серверлік) ДҚ ДҚ серверінің басқаруымен құрылады, қолданады және функцияланады;
- осы технологияны қолданумен деректерді өңдеудегі барлық шамадан тыс еңбек толығымен серверге жүктеледі. Машина-клиент сұраныс жібереді, ал сервер оны орындайды және жауаптарын клиентке жібереді.

Пішінүйлесім. Ұйымның өлшемдеріне байланысты ақпараттық жүйесінде келесі пішінүйлесімдердің біреуі болуы мүмкін:

- корпоративті және дербес қорларды құрайтын компьютер-сервер;
- компьютер-сервер және ДДҚ бар дербес компьютерлер;
- бірнеше компьютер-серверлер және ддқ бар дербес компьютерлер.

Берілген 2.1-суретте жалпы клиент пен сервер арасындағы құрлымы көрсетілген.



Сурет 2.1 – Клиент-сервер архитектурасы

"Клиенттік-сервер" – бұл желідегі компьютерлердің өзара әрекеттесу үлгісі. Ереже бойынша, компьютерлердің бәрі тең құқықты емес. Олардың әрқайсысының өзіндік бір қасиеті, арнауы, ролі болады. Желідегі кейбір компьютерлер ақпараттық-есептеу ресурстарымен, яғни процессорлар, файлдық жүйелер, почта қызметі, баспа қызметі, дерекқормен басқарады және пайдаланады. Басқалары біріншісінің қызмет көрсетуін пайдаланып, бұл қызметке жолығуға мүмкіндігі бар. Осы және басқа да ресурстарды басқаратын компьютерді осы ресурстың сервері деп атайды, ал оны қолданғысы келген компьютер – клиент деп саналады. Нақты сервер өзі басқаратын ресурстың түрімен анықталады. Сонымен, егер ресурс болып дерекқор табылса, онда дерекқордың сервері туралы сөз қозғалғаны, қызметі – деректерді өңдеумен байланысты клиенттердің сұранысына қызмет көрсету; егер ресурс файлдық жүйе болса, онда файлдық сервер немесе файл-сервер туралы айтылғаны т.б.

Желіде бір компьютер клиенттің де, сервердің де ролін атқара алады. Мысалы, UNIX басқаруымен дербес компьютері, үлкен ЭЕМ және мини-компьютері бар, соңғысы дерекқор сервері ретінде болуы мүмкін, ақпараттық жүйеде клиент-дербес компьютерлердің сұраныстарына қызмет етеді, немесе үлкен ЭЕМ-ге сұраныстарды бағыттай отырып, клиент ретінде де болуы мүмкін.

Бұл принцип бағдарламалардың өзара әрекеттесулеріне де таралады. Егер біреуі кейбір функцияларды басқаларына сәйкес қызмет түрлерін ұсына отырып орындаса, онда мұндай бағдарлама сервер ретінде қарастырылады. Бұл қызмет көрсетулермен қолданатын бағдарламалар клиент деп аталады. Сонымен, реляционды SQL-бағытталған ДҚБЖ ядросын әдетте дерекқордың сервері немесе SQL-сервер, ал деректерді өңдеу қызметтері бойынша жолығатын бағдарламаны SQL-клиент деп атайды.

Алғашында ДҚЖБ орталықтанған архитектурасы болды. Онда ДҚБЖ өзі және дерекқормен жұмыс істеген қолданбалы бағдарламалар орталық

компьютерде (үлкен ЭЕМ немесе мини-компьютер) функцияланды. Сол жерде дерекқор да орналасты. Орталық компьютерге қолданушылардың жұмыс орны ретінде болған терминалдар да қосылды. Деректерді өңдеумен байланысты барлық процестер: қолданушымен орындалған теруді қолдау, қалыптастыру, сұраныстарды оптимизациялау және орындау, сыртқы жадының құрылғыларымен алмасу және т.б., орталық компьютерде орындалды, бұл оның өндірістігіне қатал талаптар әкелді. Бірінші буынды ДҚБЖ ерекшеліктері тікелей үлкен ЭЕМ және мини-компьютердің архитектурасымен байланысты, және олардың бар артықшылықтары мен кемшіліктерін анық көрсетіп тұрады.

Қазіргі уақытта көпқолданбалы ДҚБЖ үшін стандарт болып "клиент-сервер" архитектурасы атанды.

Егер жобаланып жатқан ақпараттық жүйе (АЖ) "клиент-сервер" технологиясы бойынша құрылады деп талданса, онда қолданбалы бағдарламалар үлестірілген сипатқа ие болады. Басқаша айтқанда, қолданбалы бағдарламаның (жай айтқанда, қосымшалар) функцияларының бір бөлігі клиент-бағдарламада, қалғаны – сервер-бағдарламада шынайыланады.

"Клиент-сервер" технологиясының негізгі принципі стандартты интерактивті қосымшаның функциясының әртүрлі табиғаты бар төрт топқа бөлінуінде қорытындыланады. Бірінші топ – бұл енгізу және деректерді шығару функциялары. Екінші топ тек берілген пәндік аймаққа тән қолданбалы функцияларды біріктіреді. Үшінші топқа ақпараттық ресурстарды (деректер базасы, файлдық жүйелер т.б.) сақтау және басқару бастапқы функциялары жатады. Сонымен, төртінші топ функциялары – қызметтік, алдыңғы үш топтың арасындағы байланыстардың ролін ойнайды. Осыған сәйкес кез келген қосымшада келесі логикалық компоненттер бөлінеді:

- көрсетілім компоненті, бірінші топтың функцияларын шынайылайды;
- қолданбалы компонент, екінші топтың функцияларын қолдайды;
- ақпараттық ресурстарға қол жеткізу компоненті, үшінші топ функцияларын қолдайды;
- өзара әрекеттесу хаттамасы.

## **2.2 Деректер қорын жобалау**

Дерекқорында сақталғандардың белгілі бір логикалық құрылымы бар – басқаша айтқанда, ДҚБЖ қолдайтын кейбір деректер көрсетілімдері үлгісімен (деректер үлгісімен) сипатталады. Классикалық қатарына келесі деректер үлгісі жатады:

- иерархиялық;
- желілік;
- реляциондық.

Сонымен қатар, соңғы жылдары келесі деректер үлгілері шығып, практикаға белсенді түрде сіңуде:

- постреляциондық;
- көпөлшемді;

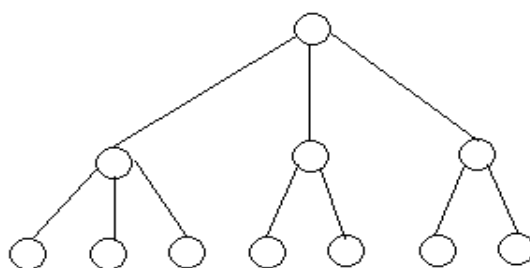
– объектілі-нысандалған.

Белгілі үлгілерді кеңейтіп, басқа да деректер үлгісінде негізделген мүмкін жүйелер әзірленуде. Оның қатарына объектілі-реляционды, дедуктивті-объектілі-нысандалған, семантикалық, концептуалдық және бағыттауыш үлгілерді жатқызуға болады. Бұл үлгілердің кейбіреуі дерекқордың, білім қорының және программалау тілінің интеграциясы үшін қызмет етеді.

Кейбір ДҚБЖ-да бір уақытта бірнеше деректер үлгісі болады. Мысалы, ИНТЕРҚОР жүйесінде қосымшалар үшін желілік деректермен манипуляциялау тілін қолданады, ал қолданушылардың интерфейсінде SQL және QBE тілдері шынайыланған.

1) Иерархиялық үлгі.

Иерархиялық үлгіде деректер арасындағы байланыстарды жүйеленген график көмегімен сипаттауға болады. Деректер арасындағы байланыстар көрсетілімі иерархиялық үлгіде қысқаша 2.2 - суретте көрсетілген.



Сурет 2.2 – Байланыстардың иерархиялық түрде көрсетілімі

Кейбір программалау тілдерінде иерархиялық ДҚ-ның құрылымын (схемасын) сипаттау үшін ағаш деректер типі қолданылады. Ағаш типі құралған болып табылады. Ол әрқайсысы «ағаш» типі болып табылатын ішкі типтен («ішкі ағаш») құралады. Әр «ағаш» типі бір «түбірлі» типтен және бағынышты типтердің реттелген жиынтығынан тұрады. Түбірлі деп бағынышты типі бар және өзі ішкі тип болмайтын тип аталады. Бағынышты тип ол үшін ата-анасының ролінде болған типке ұрпақ болып табылады. Бір типтің ұрпақтары бір-біріне егіздер болып табылады.

Иерархиялық ДҚ «жазба» типінің үлгілерін құрайтын «ағаш» типінің деректер үлгісінің реттелген жиынтығынан тұрады. Типтер арасындағы туысқандық қатынас көбінесе жазбалардың өзара қатынастарына көшеді. Жазбалар жолын ДҚ-ның негізгі мазмұнын құрайтын сандық және символдық мәндер сақтайды. Иерархиялық ДҚ-ның барлық элементтерін қарап өту әдетте жоғарыдан төмен және солдан оңға қарай өтеді.

«Ағаш» типінің анықтамасының сәйкесімен ата-аналар мен олардың ұрпақтарының арасында байланыстың тұтастығын тексеру автоматты түрде болатынын қорытындылауға болады. Тұтастықты тексерудің негізгі ережесі келесі түрде тұжырымдалады: ұрпақ ата-анасыз бола алмайды, ал кейбір ата-

аналарда ұрпақ болмауы да мүмкін. Әртүрлі ағаштар жазбаларының арасындағы байланыстың тұтастығын қолдау механизмі мүлдем болмайды.

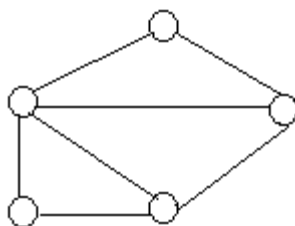
Деректердің иерархиялық үлгілерінің артықшылықтарына ЭЕМ жадын тиімді қолдану және деректердің негізгі операциясын орындауға кететін жаман емес уақыт көрсеткіші жатады. Деректердің иерархиялық үлгісі иерархиялық реттелген ақпараттармен жұмыста өте қолайлы.

Иерархиялық үлгінің кемшіліктері болып оның ақпаратты жеткілікті қиын логикалық байланыстармен өңдеуі үшін үлкендігі, сонымен қатар қарапайым қолданушыға қиындығы болып табылады.

Деректердің иерархиялық үлгісінде салыстырмалы түрде санаулы ДҚБЖ саны негізделген, оның бірнеше жүйелерін атап өтуге болады: IMS, PC/Focus, Team-Up және Data Edge, сонымен қатар Ресейдің жүйелері Ока, ИНЭС және МИРИС.

## 2) Желілік үлгі.

Деректердің желілік үлгісі деректердің элементтерінің нешетүрлі өзара байланыстарын графа түрінде суреттеуге мүмкіндік береді, сөйтіп деректердің иерархиялық үлгісін жалпылайды (2.3-сурет). Желілік ДҚ-ның толық концепциясы алғаш КОДАСИЛ (KODASYL) тобының ұсыныстарында көрсетілді.



Сурет 2.3 – Желілік үлгідегі байланыстардың көрсетілімі

Желілік ДҚ жазбалар жиынтығынан және сәйкес байланыстар жиынтығынан құралады. Байланыстарды қалыптастыруға арнайы шектеуліктер қойылмайды. Егер иерархиялық құрылымда ұрпақ-жазбада тек бір ата-ана-жазбасы болса, онда деректердің желілік үлгісінде ұрпақ-жазба ретсіз ата-ана – жазбасы санын иелене алады.

Деректердің желілік үлгісінің артықшылығы болып жады және оперативтілік шығындарының көрсеткіші бойынша тиімді шынайылауға мүмкіншілік табылады. Иерархиялық үлгімен салыстырғанда желілік үлгі ретсіз байланыстардың айналуына жол беруіне көп мүмкіншілік береді.

Деректердің желілік үлгісінің кемшіліктері болып соның негізінде құрылған ДҚ схемаларының қиындығы және қаталдығы, сонымен қатар қарапайым қолданушының түсінуіне және ДҚ-да ақпаратты өңдеуді орындауға қиыншылығы табылады. Одан басқа, деректердің желілік үлгісінде жазбалар арасындағы ретсіз байланыстарды орнатуға қол жеткізуге байланысты байланыстардың тұтастығын тексеруі әлсіреді.

Желілік үлгі негізіндегі жүйелер практикада кең таралған жоқ. Біршама белгілілері болып келесі ДҚБЖ табылады: IDMS, db\_VistaIII, СЕТЬ, СЕТОР және КОМПАС.

### 3) Реляционды үлгі.

Деректердің реляционды үлгісі IBM фирмасының қызметкері Эдгар Кодпен ұсынылған және қатынас (relation) түсінігімен негізделеді.

Қатынас кортеж деп аталатын көптеген элементтерден тұрады. Деректердің реляционды үлгісі туралы толық теориялық негізі келесі бөлімде қарастырылады. Қатынастың көрсетілімінің сыртқы үлгісі адам түйсігіне үйреншікті екі өлшемді кесте болып табылады.

Кестеде жолдар (жазбалар) және бағаналар бар. Әр жол бірдей құрылымнан тұрады және аймақтардан тұрады. Кестенің жолдарына кортеж, ал бағаналарға қатынас атрибуттары сәйкес келеді.

Деректердің реляционды үлгісінің артықшылықтары: қарапайымдылығы, түсініктілігі және ЭЕМ-де физикалық шынайылауындағы ыңғайлылығы. Олардың кең қолданылуына себеп болып қолданушыға қарапайымдылығы және түсініктілігі табылды. Ал осы типтің деректерді өңдеу тиімділігімен байланысты мәселелер техникалық түрде шешіледі.

Реляционды үлгінің негізгі кемшіліктері болып келесілер табылады: бөлек жазбалардың стандартты ұқсастық құралының болмауы және иерархиялық және желілік байланыстарын сипаттаудың қиындығы.

ПЭЕМ үшін реляционды ДҚБЖ-ның мысалдары болып келесілер табылады: dBaselll Plus және dBase IY (Ashton-Tate фирмасы), DB2 (IBM), R:BASE (Microim), FoxPro алғашқы түрлері және FoxBase (Fox Software), Paradox және dBASE for Windows (Borland), FoxPro соңғы түрлері, Visual FoxPro және Access (Microsoft), Clarion (Clarion Software), Ingres (ASK Computer Systems), Oracle (Oracle) и Microsoft SQL Server, сонымен қатар реляционды типті ресейлік ДҚБЖ жүйелері: ПАЛЬМА (ИК АН УССР), HyTech (МИФИ) жүйесі.

Реляционды ДҚБЖ-лердің соңғы түрлері объектілі-нысандаған жүйелердің кейбір қасиеттеріне ие. Мұндай ДҚБЖ объектілі-реляционды деп аталады. Мұндай жүйенің мысалы ретінде Oracle 8.x айтуға болады. Oracle-ге дейінгі жүйелердің ерте түрлерін тек реляционды деп санайды. Сонымен қатар, объектілі-реляционды деп Microsoft SQL Server, PostgreSQL, Sav Zigzag, IBM Cloudscape, FirstSQL саналады.

### 4) Постреляционды үлгі.

Классикалық реляционды үлгі кесте жазбаларында сақталған деректердің бөлінбейтіндігін болжайды. Бұл кестедегі ақпараттың бірінші қалыпты формада екенін білдіреді. Бұл шектеулік қосымшаның тиімді шынайылануына кедергі келтіретін жағдайлар болады.

Деректердің постреляционды үлгісі кесте жазбаларында сақталатын деректердің бөлінбейтіндігіне шектеулікті шешетін кеңейтілген реляционды үлгіні ұсынады. Деректердің постреляционды үлгісі мәндері ішкі мәннен



тұратын көпмәнді жолдардан тұрады. Көпмәнді жолдардың мәндерін теру негізгі кестеге кіргізілген өзіндік кесте болып саналады.

Постреляционды үлгінің артықшылығы – байланысқан реляциондық кестелердің жиынтығын бір постреляциондық кесте түрінде көрсетуге болады. Бұл ақпаратты көрсетуде жоғары анықтылық және оның өңделу тиімділігін жоғарлатуды қамтамасыз етеді.

Постреляционды үлгінің кемшілігі болып сақталған деректердің тұтастығы мен қарама-қайшылығын қамтамасыз ету мәселесін шешу қиындығы болып табылады.

Біз қарастырған деректердің постреляционды үлгісі uniVers ДҚБЖ-н қолданады. Деректердің постреляциондық үлгісінде негізделген ДҚБЖ қатарына Vubba және Dasdb жүйелері кіреді.

#### 5) Көпөлшемді үлгі.

Қордағы деректер көрсетіліміне көпөлшемді жол бір уақытта реляциондымен пайда болды, бірақ расында жұмыс істейтін көпөлшемді ДҚБЖ біздің уақытқа дейін өте аз болды. 90-жылдардың ортасынан бастап оларға деген қызығушылық көбейді.

Көпөлшемді ДҚБЖ ақпаратты интерактивті аналитикалық өңдеуге арналған тар мамандандырылған ДҚБЖ болып табылады. Осы ДҚБЖ-де қолданылатын негізгі түсініктерді көрсетейік: деректердің агрегирленуі, тарихтылық және болжамдылығы.

Деректердің агрегирленуі ақпаратты оның жалпылауының әртүрлі деңгейінде қарастыруды білдіреді. Ақпараттық жүйелерде ақпаратты көрсетудің толықтық деңгейі қолданушылар үшін олардың өзінің деңгейіне байланысты болады: аналитик, қолданушы-оператор, басқарушы, жетекші.

Деректердің тарихтылығы деректердің және олардың өзара байланыстарының статикалығының (тұрақтылық) жоғары деңгейін, сонымен қатар деректердің уақытқа қосу міндеттілігін қамтамасыз етуді болжайды.

Деректердің статикалығы оларды өңдеу кезінде жүктеу, сақтау, индекстеу және таңдаудың мамандандырылған әдістерін қолдануға рұқсат береді.

Деректердің болжамдылығы болжам жасау функциясы тапсырмасын және олардың әртүрлі уақыттық интервалдарда қолдануын айтады.

Деректердің көпөлшемді үлгісінің негізгі артықшылығы болып уақытпен байланысты үлкен көлемді деректерді аналитикалық өңдеудің ыңғайлылығы мен тиімділігі табылады. Реляциондық үлгі негізінде деректерді аналогты өңдеуді ұйымдастыру кезінде ДҚ өлшеміне және индекстеуге оперативті жадының шығынының өсуіне байланысты операцияның еңбексыйымдылығының сызықсыз өсуі болады.

Деректердің көпөлшемді үлгісінің кемшілігі болып ақпаратты жай оперативті өңдеу туралы қарапайым есептер үшін ебедейлігі табылады.

Деректердің көпөлшемді үлгісін қолдайтын жүйе мысалы болып Essbase (Arbor Software), Media Multi-matrix (Speedware), Oracle Express Server (Oracle) және Cache (InterSystems) табылады. Кейбір бағдарламалық өнімдер, мысалы, Media/MR (Speedware) біруақытта көпөлшемді және реляционды ДҚ-мен

жұмыс істеуге мүмкіндік береді. Ішкі үлгісі көпөлшемді үлгі болып табылатын Cache ДҚБЖ-де деректерге қол жеткізу үш әдіспен орындалады: тура (көпөлшемді массивтер түйінінің деңгейінде), объектілі және реляционды.

б) Объектілі-нысандалған үлгі.

Объектілі-нысандалған үлгісінде деректерді көрсеткенде қордың жазбаларын ұқсастыру мүмкіндігі туады. Дерекқордың жазбалары мен функцияларының арасында олардың өңделуі программалаудың объектілі-нысандалған тілдеріндегі құралдарға ұқсас механизмдердің көмегімен өзара байланыстар қондырылады.

Деректердің объектілі-нысандалған үлгісінің реляциондымен салыстырғандағы негізгі артықшылығы қиын өзарабайланысты объект туралы ақпараттың көрсетілуі болып табылады. Деректердің объектілі-нысандалған үлгісі дерекқордың жазбасын ұқсастыруға және олардың өңдеу функциясын анықтауға мүмкіндік береді.

Объектілі-нысандалған үлгісінің кемшілігі түсінігі қиын, деректерді өңдеу ыңғайсыз және сұраныстарды орындау жылдамдығы төмен.

90-жылдары ДҚ басқарудың объектілі-нысандалған жүйелерінің сараптауыш түп тұлғалары болды. Қазіргі уақытта осындай жүйелер кең таралған, оның ішінде келесі ДҚБЖ: POET (POET Software), Jasmine (Computer Associates), Versant (Versant Technologies), O2 (Ardent Software), ODB-Jupiter (ғылыми-өндірістік орталық «Интелтек Плюс»), сонымен қатар Iris, Orion және Postgres.

Реляционды дерекқордың негізгі түсініктері.

Қысқаша тоқталғанда, дерекқор – жұмысқа қажетті деректер жиыны. Деректер – нақты әлемнің объектерінің бейнесі. Әдеттегі терминологияда мәліметтері дерекқорында сақталатын шынайы әлемнің объектері нысан – entities, ал олардың нысандары – атрибут (attributes) деп аталады. Нақты объекттің әр белгісі атрибуттың мәні болып табылады.

Дерекқорында тек физикалық объекттер көрсетілмейді. Ол абстракция, процесстер, құбылыстар – яғни, адам тіршілігінде кездесетін барлық заттар туралы мәліметтерді сақтауға мүмкіндігі бар. Мысалы, дерекқорында складқа құралдарды әкелуге тапсырыс туралы ақпаратты сақтауға болады. "Тапсырыс" нысанының атрибуттары әкелінетін құралдың аты, құралдың саны, әкелушінің аты, мерзімі т.б. болады. Шынайы әлемнің объекттері бір-бірімен көптеген қиын тәуелділіктермен байланысты. Дерекқорында тек маңызды байланыстарды сақтау керектігін айта кету жөн.

Сонымен, ашып айтатын болсақ, дерекқор – бұл шынайы әлемнің объектерінің суреттеуінің және олардың арасындағы нақты қосымша аймақ үшін маңызды байланыстар жиынтығы.

Нысан, атрибуттар және байланыстар құрылымға айналатын әдіс деректер үлгісімен анықталады.

Қазіргі күні ең көп таралған реляционды ДҚБЖ болып есептеледі. Олар кәсіптік стандартқа айналды. Деректердің реляционды үлгісін қысқаша қарастырып шығайық.

Ол 1969-70 жылдары математикалық қатынас теориясы негізінде және түсінік жүйесіне сүйене отырып Коддпен құрылған. Маңыздысы кесте, қатынас, жол, бағана, алғашқы кілт, сыртқы кілт болып табылады.

Реляционды деректер базасы деп барлық деректер қолданушы үшін үшбұрышты деректер мәні кестесі түрінде көрсетілген, және барлық операциялар кестелермен манипуляцияға қосылатын база аталады. Кесте жолдар мен бағаналардан тұрады және дерекқор ішінде өзінің қайталанбас аты болады. Кесте шынайы әлемнің объектінің типін (нысанын), ал әр бағана нақты объектті көрсетеді.

Атрибуттардың мәні көптеген рұқсат етілген домен (domain) деп аталатын мәндерден таңдалады.

Әр бағананың әдетте кестенің үстіңгі бөлігінде жазылатын аты болады. Оның қайталанбас аты болу керек, бірақ әртүрлі кестелердің бағаналарының аты бірдей болуы мүмкін. Кез келген кестеде кем дегенде бір бағана болу керек; бағаналар кестеде оларды құру кезіндегі реттікке сәйкес орналасады. Бағаналарға қарағанда жолдардың аты болмайды; кестеде олардың реті анықталмаған, ал саны шектелмеген.

Жолдар кестеде реттелмегендіктен, бағананы оның позициясы бойынша таңдау мүмкін емес. Сонымен қатар, жолдың нөміріне жалғасу көпқолданбалы ДҚБЖ-да дұрыс емес. Кез келген кестеде мәндері әр жолмен теңесетін бір немесе бірнеше бағанадан тұрады. Мұндай бағана (немесе бағаналар комбинациясы) бірінші кілт деп аталады (primary key). Егер кесте осы талаптарды қанағаттандырса, онда ол қатынас (relation) деп аталады.

Кестелердің өзарабайланысы деректердің реляционды үлгісінің маңызды элементі болып табылады. Ол сыртқы кілтпен қолданады (foreign key).

Егер дерекқорында «деректер туралы деректер» болмаса, кестені сақтауға немесе өңдеуге болмайды, мысалы, кестені, бағананы сипаттауыш т.б. оларды әдетте метадеректер деп атайды. Метадеректер кестелік түрде көрсетілген және деректер сөздігінде сақталған (data dictionary).

Кестелерден бөлек дерекқорында басқа объекттер – экранды форма, есептер (reports), көрсетілімдер (views), сонымен бірге дерекқормен жұмыс жасайтын қосымша программалар да сақталады.

Ақпараттық жүйенің қолданушысына дерекқордың тек шынайы әлемнің объектерін көрсетіп тұруы жеткіліксіз. Мұндай бейненің бірауызды болуы маңызды. Мұндай жағдайда дерекқор тұтастық шартын (integrity) қанағаттандырды деп айтады.

Деректердің дұрыстығын кепілдеу үшін дерекқорына тұтастық шектеуліктері (data integrity constraints) деп аталатын кейбір шектеуліктер қойылады.

Тұтастық шектеуліктерінің бірнеше типі бар. Мысалы, кесте бағанасы мәндері тек сәйкес домендерден таңдалынсын деп талап етіледі. Практикада одан да қиын шектеуліктер қояды, мысалы, сілтеме бойынша шектеуліктер (referential integrity). Оның мәні сыртқы кілт кестеде жоқ бағанаға көрсеткіш

бола алмайтындығында. Тұтастық шектеулігі арнайы құрылғылардың – ереже (rules), триггерлер (triggers) және домендер көмегімен жасалады.

Жай деректер өздігінен компьютерлік формада егер оларға қолжетерлік құрылғылар болмаса, қолданушыға қызығушылық тудырмайды. Деректерге қолжеткізу дерекқорына сұраныс түрінде орындалады. Олар стандартты сұраныс тілдерінде жазылады. Бүгінде көп ДҚБЖ үшін сондай тіл болып SQL саналады.

### 2.3 SQL тілі туралы негізгі деректер

Бұл тілдің дерекқорына қол жеткізудің сипаттау құралы ретінде пайда болуы және дамуы реляционды дерекқордың теориясы құрылуымен байланысты. SQL тілінің сипаты 1970 жылы System/R жобасының ғылыми-зерттеуінің нәтижесінде шықты, жұмыс IBM фирмасының Санта-Тереза зертханасында өтті. Қазірде SQL – бұл реляционды ДҚБЖ интерфейсі стандарты. Оның атақтылығы сонша, реляционды емес ДҚБЖ (мысалы, Adabas немесе Betrieve) құрушылары өздерінің жүйелерін SQL-интерфейстерімен қамтайды.

SQL тілі ANSI/ISO стандартына кіреді. Көп ДҚБЖ құрушылар осы стандартты таңдайды, алайда деректерді өңдеудің жаңа мүмкіншіліктерін орындау үшін оны жиі кеңейтеді. Деректерді басқарудың жаңа механизмдері тек арнайы SQL операторлары арқылы қолданылуы мүмкін, жалпы жағдайда стандартқа кірмейтін тілдер.

Бағдарламалаудың процедуралық тілінен айырмашылығы: есептеу процестерін басқару инструкциясы (циклдер, тармақталу, басқа операторға өту) және енгізу/шығару құралдары жоқ.

Microsoft Access, Visual FoxPro немесе Paradox сияқты ДҚБЖ, SQL–де сұраныстарды бағдарламалаумен байланысты әрекеттерді өздері орындайды, мысалы Query By Example (QBE) – үлгі бойынша сұраныс, пайдаланушыға сұранысты визуальды құру құралдарын білдіреді.

SQL бағдарламалау тілінің толық функционалдық мүмкіндіктерін қамтымағанмен, ол деректерге ену мүмкіндігіне арналған және оны бағдарламаны жасау құралына қосады. Сонымен қатар, SQL командаларымен жұмыс істеу үшін сәйкес құралдар және компоненттер бар.

SQL тілінің функциялары.

Статистикалық функциялар:

- AVG() – орташа мән;
- MAX() – ең үлкен мән;
- MIN() – ең кіші мән;
- SUM() – қосынды;
- COUNT() – мәндердің саны;
- COUNT(\*) – нөлдік емес мәндердің саны.

Жолдармен жұмыс істеуге арналған функциялар:

- UPPER(Str) – Str символдық жолды жоғарғы регистрге айыстыру;
- LOWER(Str) – Str символдық жолды төменгі регистрге айыстыру;
- TRIM(Str) – Str жолының бастапқы және соңғы бос орындарын өшіру;
- SUBSTR (Str FROM TO) – Str жолынан, өзі символдардан тұратын, n1 жолынан бастап, n2 жолымен аяқталатын ішкі жолды бөліп алу;
- CAST(<Expression> AS <Type>) – Expression өрнегін Type типіне келтіру.

Ай-күн (Дата) және уақытпен жұмыс істейтін функция:

- EXTRACT(<Элемент> FROM <Өрнек>) – дата және уақыт мәндерінен тұратын өрнектен, сәйкес көрсетілген элементтен мәндер алу; дата және уақыт элементі ретінде YEAR, MONTH, MINUTE және SECOND элементтерін көрсетуге болады.

Кестелермен келесі амалдарды орындауға болады:

- жаңа кесте құру;
- кестені өшіру;
- кесте өрістерінің құрамын өзгерту;
- индексті құру және өшіру.

Бұл әрекеттер SQL тілінің инструкцияларының көмегімен орындалады

SQL әдеттегі көрсетілімді программалау тілі емес. Онда бағдарлама жазылмайды, ал дерекқорына сұраныстар жазылады. Сондықтан SQL – декларативті тіл. Бұл дегеніміз оның көмегімен қажетті алынатын затты нысандауға болады, бірақ оның қалай істелінетін көрсетуге болмайды. Жалпы процедуралық программалау тілдеріне (Си, Паскаль, Ада) қарағанда, SQL тілінде if...then...else, for, while операторлары болмайды, бірақ SQL кеңейтілімінде сақталатын процедуралар және триггерлер үшін (SQL/PTL - SQL/Procedure And Trigger Language) олар бар екенін көрсеткен жөн.

SQL тілінде сұраныс бір-бірінің артынан еріп отыратын және нүктелі үтірмен бөлінетін бір немесе бірнеше операторлардан тұрады.

Төменде 2.1-кестеде ANSI/ISO SQL стандартына кіретін маңызды операторлар көрсетілген.

К е с т е 2.1 – SQL тілінің негізгі операторлары

Оператор синтаксисі	Орындалатын амал
SELECT	Дерекқорынан деректі таңдау
INSERT	Кестеге деректі енгізу
DELETE	Кестеден деректі өшіру
UPDATE	Кестедегі деректерді өзгерту
GRANT	Объекттің амалдарына құқықты беру
REVOKE	Объекттің амалдарынан құқықты алып кету
COMMIT	Транзакцияны құптау
ROLLBACK	Транзакциядан секіріп өту
CREATE	Дерекқор объектін құру
DROP	Дерекқор объектін өшіру

SQL тілінің сұраныстарында дерекқордың объектітерін теңестіретін аттар қолданылады. Жайларымен қатар, қиын аттар да қолданылады – мысалы, бағананың квалификациялық аты (qualified column name) өзі жататын бағана мен жолдың атын анықтайды.

Кез келген бағана кестеде анықталған бір типтің деректерін сақтайды. Базалық деректер типі – символдар жолы өлшенген ұзындықта, бүтін және нақты сандар, және қосымша деректер типі - айнымалы қзындықты бағананың жолдары, ақшалық бірліктер, уақыт және мерзім, логикалық деректер (екі мән - "РАС" және "ЖАЛҒАН") деп екіге бөледі. SQL тілінде сандық, жолдық, символдық тұрақтылар және «уақыт» және «мерзім» типті тұрақтыларды қолдануға болады.

Кестелерге тез қолжетімді қамтамасыз ететін құралдың бірі индекс болып табылады. Индекс – бұл кестенің нақты жолына көрсеткіш болатын дерекқордың құрылымы. Индекс кестенің нақты жолының бір немесе бірнеше бағандарынан алынған мәнді және осы жолға сілтемені құрайды. Индексте міндер жинақталған, ол ДҚБЖ кестеде тез іздеу салуды қамтамасыз етеді.

Егер кесте үшін индекстер болмаса, сұранысты орындау үшін ДҚБЖ барлық кестені қарап шығуы керек. Үлкен кесте үшін мұндай сұраныс өте ұзақ орындалады.

Егер алдын-ала бағана бойынша WHERE сұранысы шарттарына кіретін индекс құрылған болса, онда кестеден іздеу уақыты біршама азаяды. Индекс SQL CREATE INDEX (ИНДЕКС ҚҰРУ) операторымен құрылады.

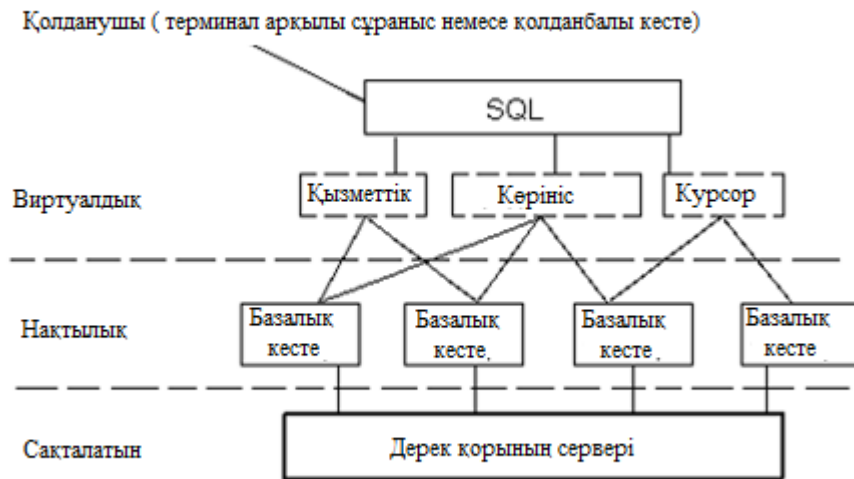
ДҚБЖ қолданушысы үшін қызығушылықты SQL тілінің операторлары емес, ал олардың тұтастай әшекейленген және мәні бар кейбір тізбегі тудырады. SQL тілінің әрбір осындай операторлар тізбегі деректер базасымен анықталған амал жүргізеді. Ол бірнеше қадаммен орындалады, әр қадамда дерекқордың кестелерімен бірнеше операциялар орындалады. осылай банк жүйесінде белгілі сумманың аударымы бірнеше операция арқылы орындалады. Оның ішінде қысқамерзімді есептен сумманы шығару, ұзақмерзімді есепке қосу.

Егер осы амалды орындау барысында жаңылысу болса, мысалы, бірінші операция орындалып болғаннан кейін, онда ақша жоғалады. Сәйкесінше, кез келген дерекқормен байланысты амал тұтастай жасалуы керек немесе мүлдем жасалмау керек. Мұндай амал транзакция деген атқа ие болды.

Транзакцияны өңдеу транзакцияны өту және дерекқордың жағдайын қайтару үшін қолданатын журналға сүйенеді.

Әлі күнге дейін «кесте» ұғымы шынайы немесе базалық кесте, яғни машинаның физикалық жадында сақталатын, әр жол үшін шынында егізі бар кестемен байланыстырылды (2.4-сурет). Алайда, SQL виртуалды кестелерді құрады және қолданады: көрсетілімдер, курсорлар және аты жоқ жұмыс кестелері, оларда сұраныстардың нәтижелері нысандалады. Бұл кестелер дерекқорында жоқ, бірақ қолданушы көзқарасымен олар бар.





Сурет 2.4 – Қолданушы түйсігіндегі дерекқор

### **3 Дерекқорды жобалау және оны іске асыру**

#### **3.1 ДҚ құрудың бастапқы кезеңі**

##### **3.1.1 Айырбастау пунктiнiң дерек қоры жұмысының жалпы сипаттамасы**

Администратор және кассир ақша айырбастау пунктi жұмысының негiзiн қалайды. Ақша айырбастау пунктiнiң администраторы сол күннiң ақша курсы енгізедi және валюта өзгерiсiн көре алады. Администратор кассир бөлiмiн бақылай алады және жаңа кассир енгiзiп, логин, пароль бередi. Ал кассир администратор берген, логин парольмен кiрiп клиенттерге қызмет көрсете алады.

##### **3.1.2 Мәселелер және шектеулер**

Айырбастау пункттерi үшiн қойылған мәселелер мен шектеулер:

- қазiргi кезде айырбастау пункттерiнде автоматтандырылған бағдарлама жетiспеуi және жоқтығы;
- бағдарлама тек администратор мен кассир үшiн арналған;
- бағдарламаны кез келген айырбастау пункттерiнде қолдануға тиiмдi, қолжетiмдi етiп жасалынған.

##### **3.1.3 Дерекқордағы шешiлетiн мәселелер**

Ақша айырбастау пунктiндегi дерек қорында администратор мен кассир бөлiмi қарастырылады. Администратор бөлiмi кассирлер бөлiмiн бақылайтын орталықтандырылған деректер қорларын тиiмдi басқаратын жүйесiн құру, валютаны енгізуге байланысты ағымдағы валюта өзгерiсiн бақылау, кассир бөлiмi клиент үшiн ақпараттар бере алуы және валюта айырбастау соңында жүргiзiлген қызмет үшiн чек бередi. Осы мәлiметтердiң барлығыда администратор бөлiмдегi деректер қорына енгiзiледi.

##### **3.1.4 Дерекқордың функционалдығын сипаттау**

Жобаланатын дерекқор және бағдарламалық интерфейсi ақша айырбастау пункттерi үшiн арналған. Бағдарламалық қосымшада келесi функциялар iске асырылған:

- ақпараттық функция, кассир, ақша айырбастау операциясы туралы толық ақпаратты бередi;
- енгiзу функциясы, дерекқорға жаңа ақпарат енгізуге мүмкiндiк бередi;

## **3.2 Концептуалды жобалау**

### **3.2.1 Қажетті ақпарат пайдаланушылардың талабы**

Жоба пайдаланушының барлық негізгі талаптарын қанағаттандыру керек. Бұл жүйенің негізгі пайдаланушылары ақша айырбастау пункттері болып табылады. Негізгі талаптар алынған ақпараттың ұсынылатын тиімділік деңгейінде негізделеді. Жүйеге қойылатын негізгі талаптардың тізімі:

- жүйе қолданыста қарапайым болу керек. интерфейс ыңғайлы болу керек;
- жүйе қауіпсіздікті қамтамасыз ету керек;
- жүйе дерекқордың тұтастығын қамтамасыз ету керек;
- жүйе толығымен интеграциялану керек, яғни мәліметтердің артылуы және артық жаңартулар болмау керек.

### **3.2.2 Дерекқор және қосымшаны жобалау құралдарын таңдаудың негіздемесі. Дерекқорларды басқару жүйесі**

Барлық талаптарды қамтамасыз ететін информациондық жүйені жүзеге асыру үшін бірінші кезекте реляциондық дерекқорын және серверлі архитектура клиентін қолданудың анық шешіміне әкелетін орталық деректер сақтаудың әдісін таңдау қажет.

Қазіргі кезде қойылған тапсырмалар талаптарын ДҚБЖ мүмкіншілігіне қарай және тапсырыс берушінің таңдауына байланысты таңдауға болатын бірнеше ДҚБЖ көрсетілген:

- орнату және қолданудың жеңілдігі;
- кең таралған;
- қымбат серверлік қуатты орнатуды қажет етпейді;
- өндіргіштігінің жақсы көрсеткіші;
- деректерді сақтау сенімдігі мен қауіпсіздігін қамтамасыз етудің жеке құралдары;
- коммерциялық емес қолдану үшін өнімді сызғышта арзан түрлерінің болуы.

Microsoft SQL Server төмен өндіргіштік серверлік қуаттарға есептелген, сонымен қатар қолданушылық жұмыс станциясында қолданушыға өндіргіштіктің төмендеуінің білінуінсіз жұмыс істеуге мүмкіндігі бар. Сонымен қатар кең таралғандық критерийларына және деректерді сақтау сенімдігі мен қауіпсіздігін қамтамасыз етудің жеке құралдарының бар екендігіне жауап береді.

SQL компьютерлік дерекқорда сақталатын деректерді өңдеу және оқу үшін арналған құрал болып табылады, және қолданушының дерекқормен өзара қатынас ұйымдастыру үшін қолданылатын программалау тілі болып табылады. Шындығында SQL тек бір белгілі типті – реляционды дерекқормен жұмыс

істейді. SQL ДҚБЖ қолданушыға ұсынатын барлық функционалдық мүмкіншіліктерді жүзеге асыру үшін қолданылады, соның ішінде:

– деректерді ұйымдастыру. SQL қолданушыға деректердің көрсетілімі құрылымын өзгертуге, сонымен қатар дерекқордың элементтері арасында қатынас құруға мүмкіндік береді;

– деректерді оқу. SQL қолданушыға немесе қосымшаға дерекқордың ішіндегі деректерді оқуға және олармен қолдануға мүмкіндік береді;

– деректерді өңдеу. SQL қолданушыға немесе қосымшаға дерекқорды өзгертуге, яғни оған жаңа деректер енгізуге, оған қоса бұрыннан бар деректерді өшіруге немесе жаңартуға мүмкіндік береді;

– қатынас құруды басқару. SQL-дің көмегімен қолданушының деректерді оқу және өзгерту және рұқсат етілмеген қатынас құрудан қорғауға мүмкіндігін шектеуге болады.

– деректерді бірге қолдану. SQL параллельді түрде жұмыс істейтін қолданушыларға бір-біріне кедергі келтірмес үшін деректерді бірге қолдануын қадағалайды;

– деректердің тұтастығы. SQL дерекқордың тұтастығын қамтамасыз етіп, оны келісімсіз өзгерістерден немесе жүйенің бас тартуынан қираудан қорғауға рұқсат береді.

SQL – дерекқорды бағдарламалау тілі. Дерекқорға қатынас құру үшін бағдарлаушылар өздерінің бағдарламаларына SQL командаларын қояды. Бұл әдістеме қолданушылар жазған бағдарламаларда да, дерекқорлардың қызметтік бағдарламаларында да қолданылады (есеп беру генераторы және деректерді енгізу құралдарында).

Осылайша, SQL адамдарға, бағдарламаларға және есептеуіш жүйелерге реляционды дерекқорлардағы ақпараттарға қатынас құруды қамтамасыз ететін қажетті және мықты құралға айналды.

### 3.2.3 ER моделін құру

ДҚ құрудың бастапқы кезеңінде алғашқы нысандар жиынтығын анықтау қажет. Бұл ақырғы пайдаланушылар мен жобалаушы көзқарасы бойынша жүйе объектілері туралы ең маңызды ақпарат болып табылады обода «Айырбастау пунктіндегі ақпараттар қозғалысын бақылау жүйесін құру» дерекқоры 3.1-кестеде көрсетілген нысандарды қолданады.

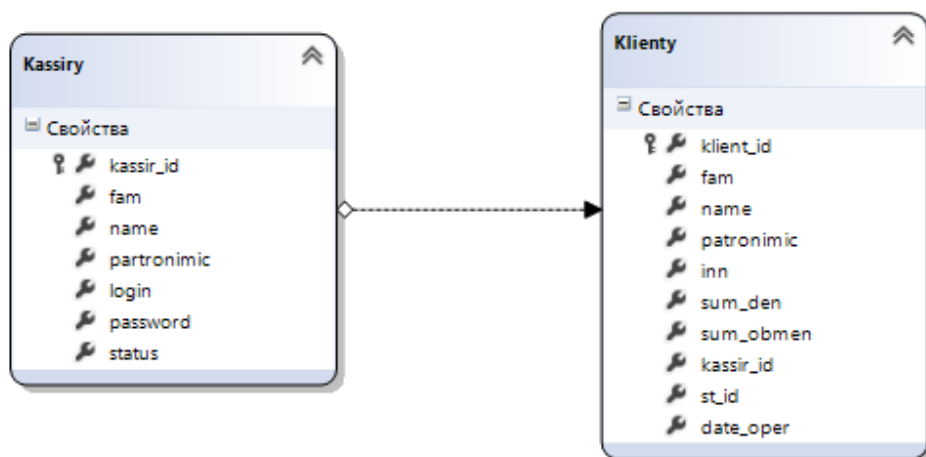
К е с т е 3.1 - «Айырбастау пунктіндегі ақпараттар қозғалысын бақылау жүйесін құру» дерекқорының нысандары

Нысан аты	Сипаттамасы
Kassiry	Кассир туралы мәліметтер
Klienty	Клиент туралы мәліметтер
Oper_type	Операция типі(сату,сатып алу)
Valute_status	Ақша бірлітері
Valute	Ақша бірліктері

Жобалаушы және ақырғы пайдаланушы нысандарды анықтау кезінде келісімге келу керек. Жобалаушы нысандар арасындағы байланыстарды операциялардың сипаттамасына негізделіп анықтайды. Нысандар арасындағы байланыстар операциялардың толық сипаттамасы негізінде құрылған бизнес-ережелерге негізделеді.

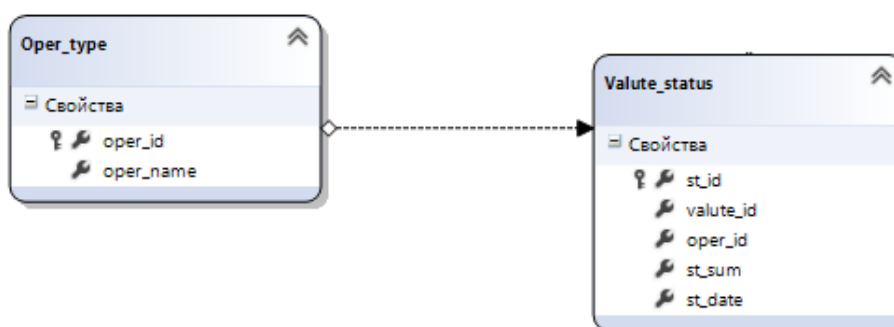
### 3.2.4 Бизнес ережелер

1) Кассир мен клиент туралы ақпараттар мына таблицалардың байланысынан тұрады. (3.1-сурет)



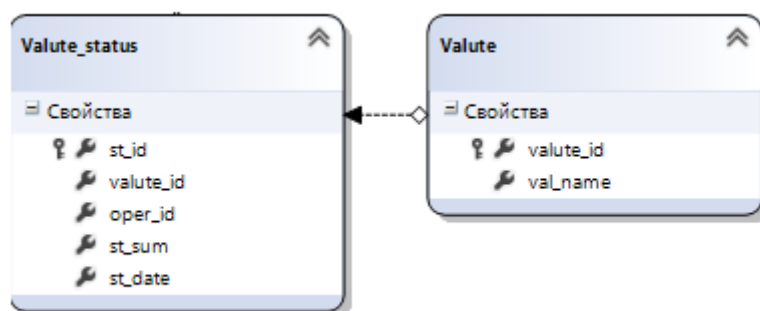
Сурет 3.1 – 1-бизнес ереже үшін ER сегменті

2) Операция типі және валюта туралы толық ақпарат мына таблицалардың байланысынан тұрады.(3.2-сурет)



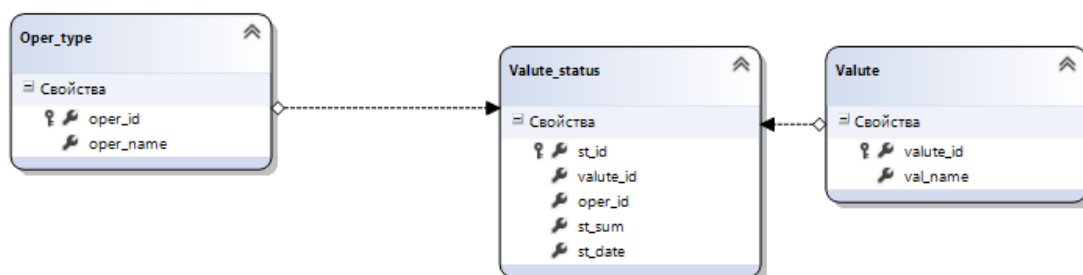
Сурет 3.2 – 2-бизнес ереже үшін ER сегменті

3) Ақша бірліктері туралы ақпараттармен байланысқан таблицалар.(3.3-сурет)



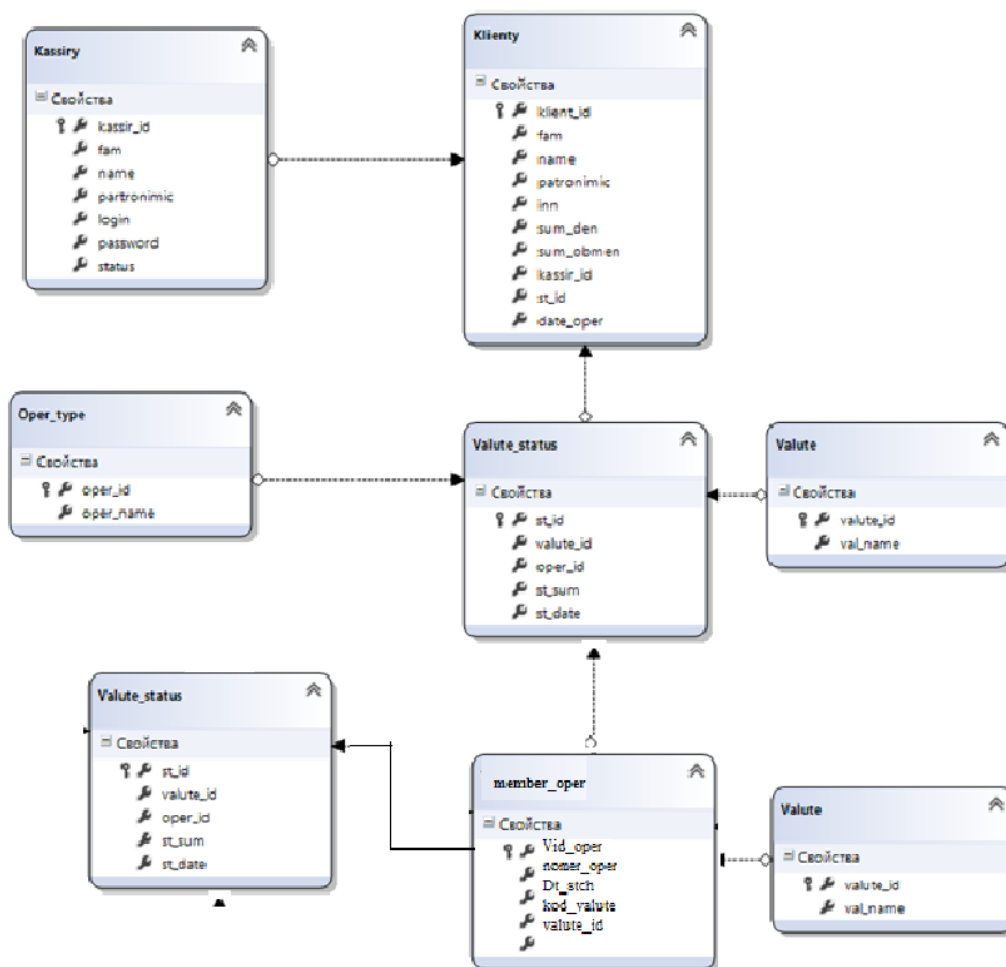
Сурет 3.3 – 3-бизнес ереже үшін ER сегменті

4) Операция типі мен ақша бірліктер таблица арасындағы қатынас таблица байланысы.(3.4-сурет)



Сурет 3.4 – 4-бизнес ереже үшін ER сегменті

5) «Айырбастау пунктіндегі ақпараттар қозғалысын бақылау жүйесін құру» ДҚ.(3.5-сурет)



Сурет 3.5 – ER диаграмма

### 3.2.5 Реляционды қатынастарды құрастыру

Әрбір реляционды қатынас бір нысанға сәйкес келеді және оған нысанның барлық атрибуттары енгізіледі. Әрбір қатынас үшін алғашқы кілт және сыртқы кілттерді (егер бар болса) анықтау қажет. Қатынастар 3.2-3.6-кестелерінде келтірілген.

Әрбір қатынас үшін ішкі аты, типі мен ұзындығы бар атрибуттар көрсетіледі Мәліметтер типі келесідей белгіленеді: int – сандық, char – символдық, Datetime – мезгіл (оның ұзындығы ДҚБЖ-не байланысты әртүрлі болады, сондықтан ол көрсетілмейді).

К е с т е 3.2 - Kassiry қатынасы

Әріс аты	Атрибут аты	Мәліметтер	Ескерту
Кассир коды	kassir_id	int	Алғашқы
Кассир фамилиясы	fam	varchar(MAX)	Қажетті

Кесте 3.2-нің соңы

Кассир аты	name	varchar(MAX)	Қажетті өріс
Кассир тегі	otchec	varchar(MAX)	Қажетті өріс
Кассир логині	login	varchar(MAX)	Қажетті өріс
Кассир паролі	password	varchar(MAX)	Қажетті өріс
Кассир статус	status	int	Қажетті өріс

К е с т е 3.3 - Klienty қатынасы

Өріс аты	Атрибут аты	Мәліметтер типі	Ескерту
Клиент коды	klient_id	int	Алғашқы кілт
Клиент фамилиясы	fam	varchar(MAX)	Қажетті өріс
Клиент аты	name	varchar(MAX)	Қажетті өріс
Клиент тегі	otchec	varchar(MAX)	Қажетті өріс
Клиент ЖНС	inn	bigint	Қажетті өріс
Ауыстыратын ақша	sum_den	real	Қажетті өріс
Ауысқан ақша	sum_obmen	real	Қажетті өріс
Кассир коды	kassir_id	int	Сыртқы кілт
Ақша бірлігі коды	st_id	int	Сыртқы кілт
Күні	date_oper	datetime	Қажетті өріс

К е с т е 3.4 - Oper\_type қатынасы

Өріс аты	Атрибут аты	Мәліметтер типі	Ескерту
Операция	oper_id	int	Алғашқы кілт
Операция	oper_name	varchar(MAX)	Қажетті өріс

К е с т е 3.5 - Valute қатынасы

Өріс аты	Атрибут аты	Мәліметтер типі	Ескерту
Валюта коды	valute_id	int	Алғашқы кілт
Валюта аты	val_name	varchar(MAX)	Қажетті өріс



К е с т е 3.6 - Valute\_status қатынасы

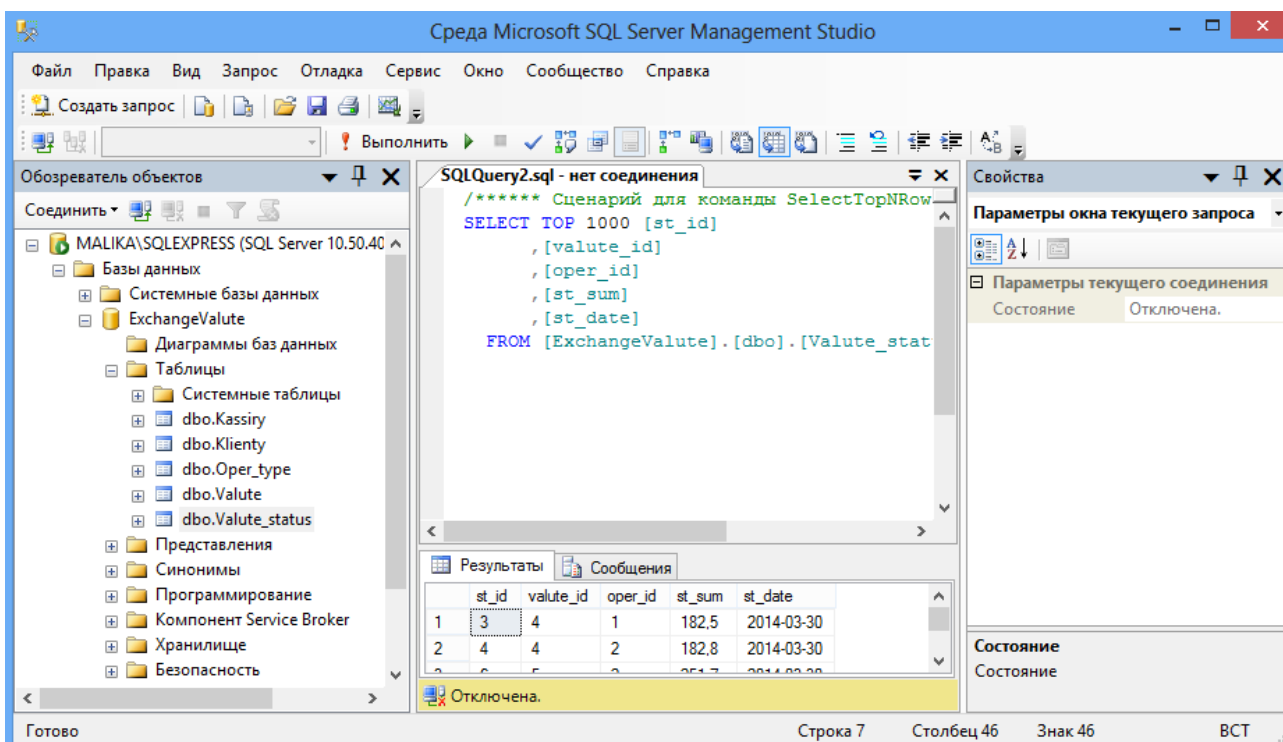
Өріс аты	Атрибут аты	Мәліметтер типі	Ескерту
Ақша бірлігі коды	st_id	int	Алғашқы кілт
Валюта коды	valute_id	int	Сыртқы кілт
Операция коды	oper_id	int	Сыртқы кілт
Валюта суммасы	st_sum	real	Қажетті өріс
Күні	st_date	date	Қажетті өріс

### 3.3 Логикалық жобалау

Логикалық жобалау кезеңінде дерекқор моделі жүйені іске асыру кезінде қолданылатын белгілі бір ДҚБЖ (MS SQL Server) үшін жарамды форматқа трансформаланады. Логикалық жобалаудың мақсаты реляционды кестелік құрылымдарды құру болып табылады. Жобаланатын дерекқорда кестелердің және байланыстардың құрылымдарын MS SQL Server көмегімен жасаймыз.

#### 3.3.1 Кестелер құру

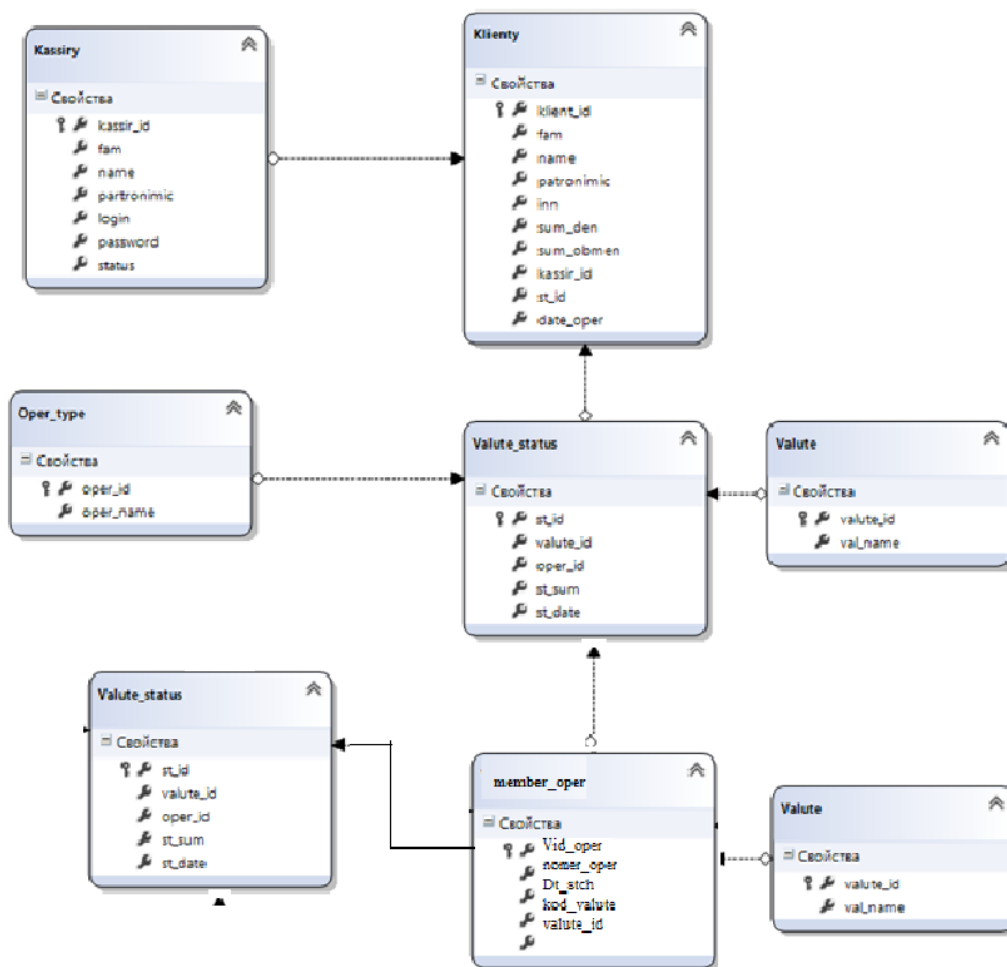
Кестелер CREATE TABLE командасымен құрылады. CREATE TABLE кесте атын және бағандар аттарын анықтайды. Сонымен қатар ол деректердің типін және бағандардың өлшемін анықтайды. Ең ыңғайлы бұл генерацияланған скриптті Query Analyzer-де (MS SQL Server) қосу. Кестелерді құруға арналған скрипт Б қосымшасында келтірілген.(3.6-сурет)



Сурет 3.6 – Query Analyzer-де кестелерді құру

### 3.3.2 ДҚ диаграммасы

Концептуалды моделді дерекқордың сәйкес құрылымына енгізілу тәсілі қандай болса да, ДҚ реляционды сұлбасы жобаға сәйкес болу керек. (3.7-сурет)

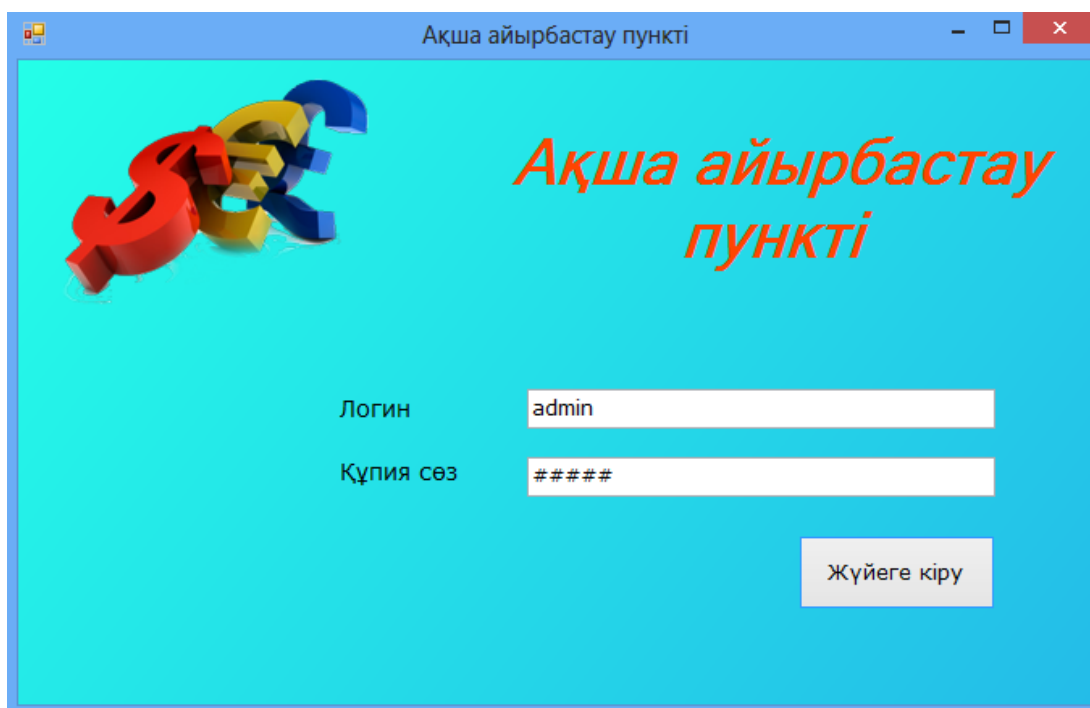


Сурет 3.7 – ДҚ диаграммасы

### 3.4 Физикалық жобалау

#### 3.4.1 Қосымша интерфейсін және оның функционалдығын сипаттау.

Жобаның интерфейсі Visual Studio 2012 объектілі-нысандалған бағдарламалау ортасында орындалды. Қосымшаның бағдарламалық коды Б қосымшасында келтірілген.(3.8-сурет) Бағдарламаны қосу кезінде, алдымен бағдарламаның администратор алғашқы беті ашылады.

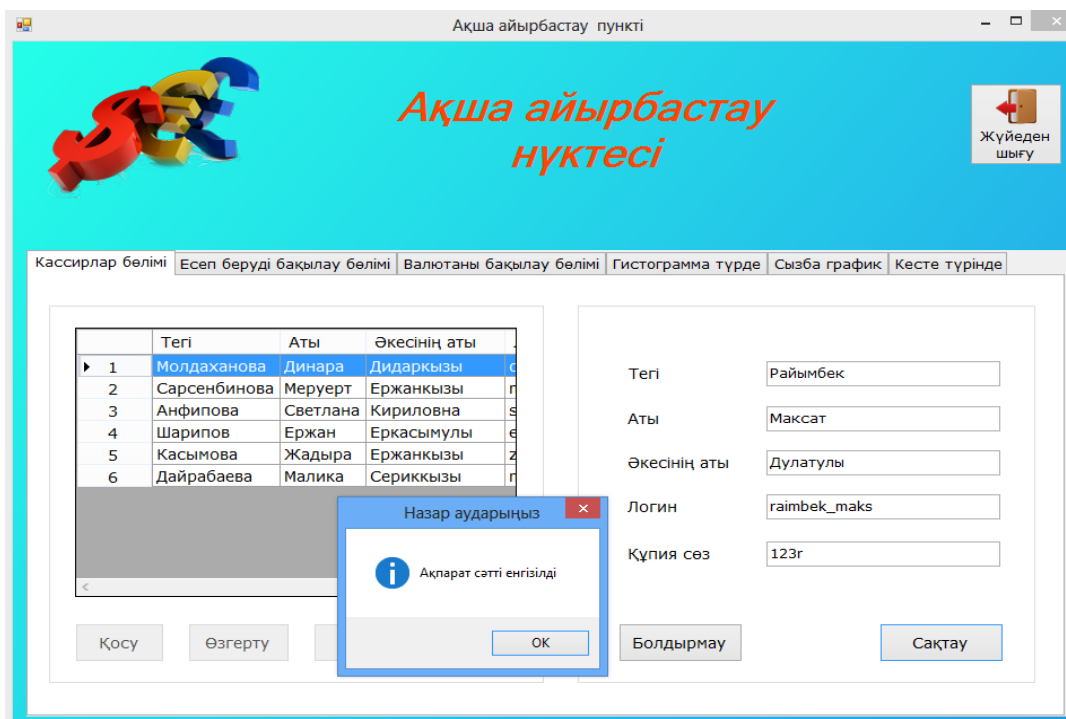


Сурет 3.8 – Бағдарламаның администратор бөлімі

Қолданушылар:

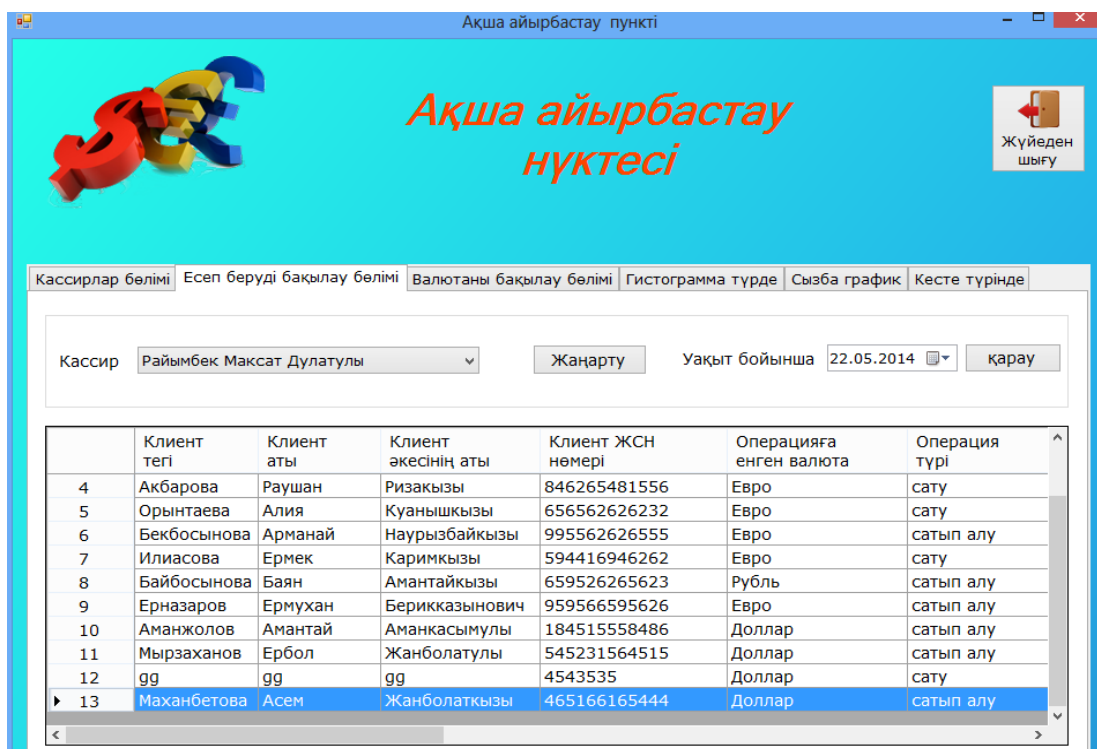
- админ;
- кассир.

Администратор және әр кассирдің логині және құпия сөзі бар. «Администратор» бөлімінде сәйкесінше админ өзінің логині мен құпия сөзін тереді. Администратор бөлімінде админ кассир бөлімін басқарады. Яғни жаңа кассир қосу, өшіру, қол жетімділігін ашу, жабуға болады. (3.9-сурет)

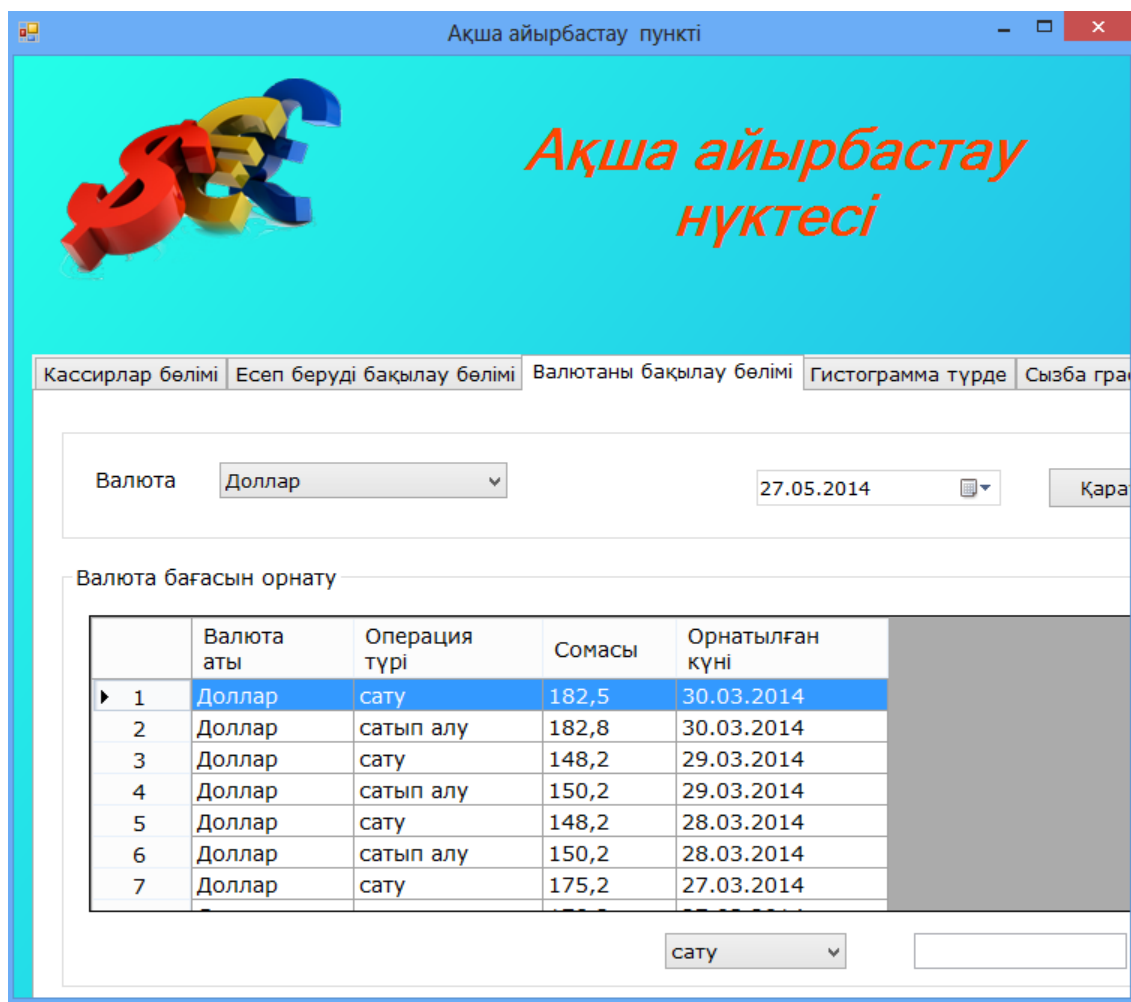


Сурет 3.9 – Бағдарламаның администратор бөліміндегі кассирлерді енгізу бөлімі

Администратор бөлімінде есеп беруді бақылау бөлімі бар. Онда кассирлердің жалпы жұмысын уақыты бойынша қарауға мүмкіндік бар.



Сурет 3.10 – Бағдарламаның администратор бөліміндегі есеп беруді бақылау бөлімі

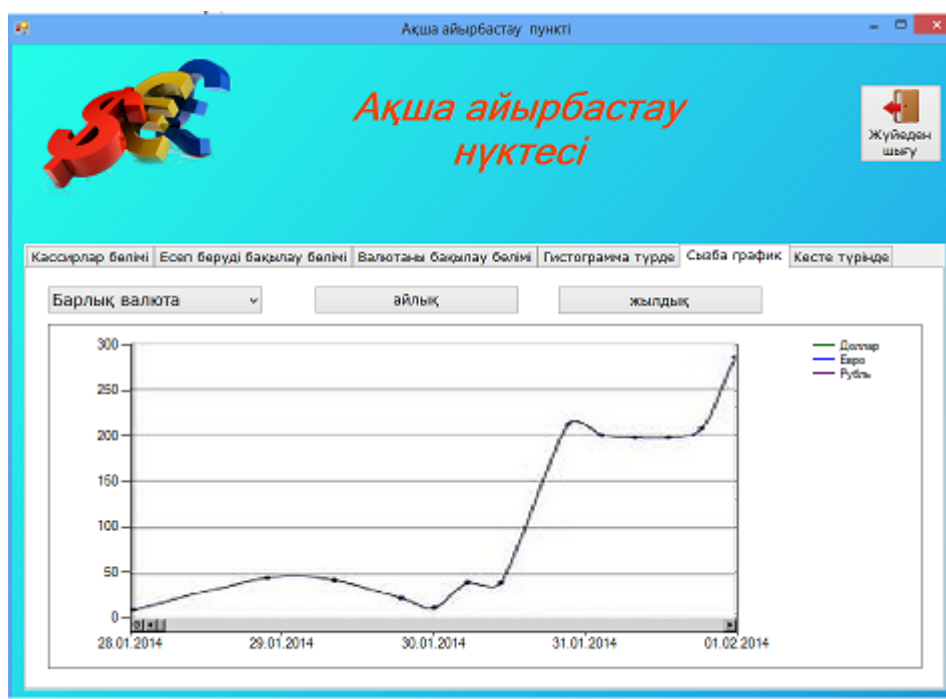


Сурет 3.11 – Бағдарламаның администратор бөліміндегі валютаны бақылау бөлімі

Администратор бөлімінде 3.11-3.18-суреттер бойынша валютаны бақылау бөлімі қарастырылды. Яғни администратор валюта бағасын енгізеді, өткен күндердің бағасын қарай алады. Егер баға қате енгізіген жағдайда өзгерте алады. Валюта бағасын енгізе отырып, администратор оларды гистограмма, график, кесте түрінде қарай алады. Бақылау жыл, ай, күн бойынша қарастырылған.



Сурет 3.12 – Валютаны бақылауды гистограмма түрінде қарау



Сурет 3.13 – Валютаны бақылауды график түрінде қарау

Ақша айырбастау пункті

**Ақша айырбастау нүктесі**

Жүйеден шығу

Кассирлар бөлімі | Есеп беруді бақылау бөлімі | Валютаны бақылау бөлімі | Гистограмма түрде | Сызба график | Кесте түрінде

Доллар | Орташа бағасы 160,31 тг.

	Валюта	Операция түрі	Орта бағасы (тг.)	Айы
1	Доллар	сату	123	қаңтар
2	Доллар	сатып алу	123,6	қаңтар
3	Доллар	сату	167	ақпан
4	Доллар	сатып алу	164,2	ақпан
5	Доллар	сату	153,16	наурыз
6	Доллар	сатып алу	154,96	наурыз
7	Доллар	сату	136,5	сәуір
8	Доллар	сатып алу	141,8	сәуір
9	Доллар	сату	177,75	мамыр
10	Доллар	сатып алу	178,47	мамыр

Сурет 3.14 – Валютаны бақылауды кесте түрінде қарау

Ақша айырбастау пункті

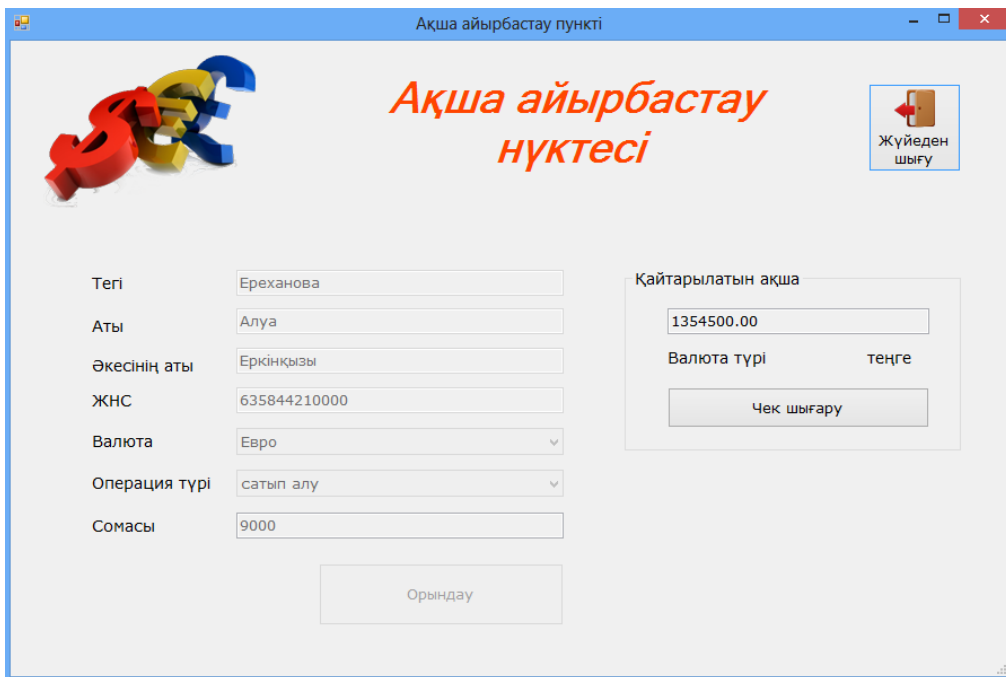
**Ақша айырбастау нүктесі**

Логин

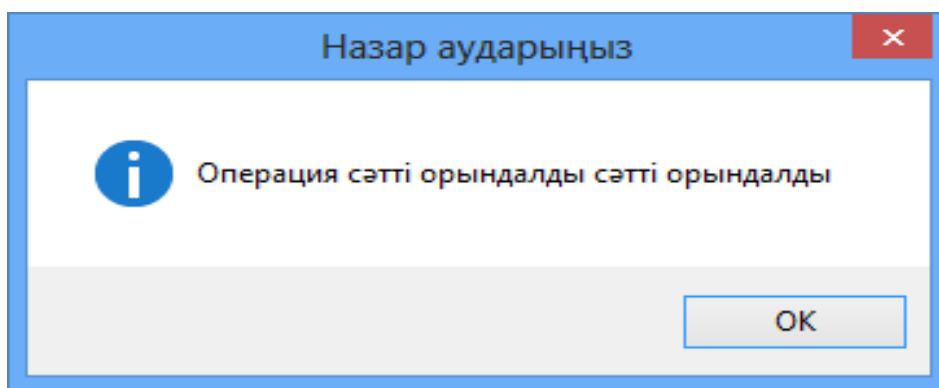
Құпия сөз

Жүйеге кіру

Сурет 3.15 – Бағдарламаның кассир бөлімі

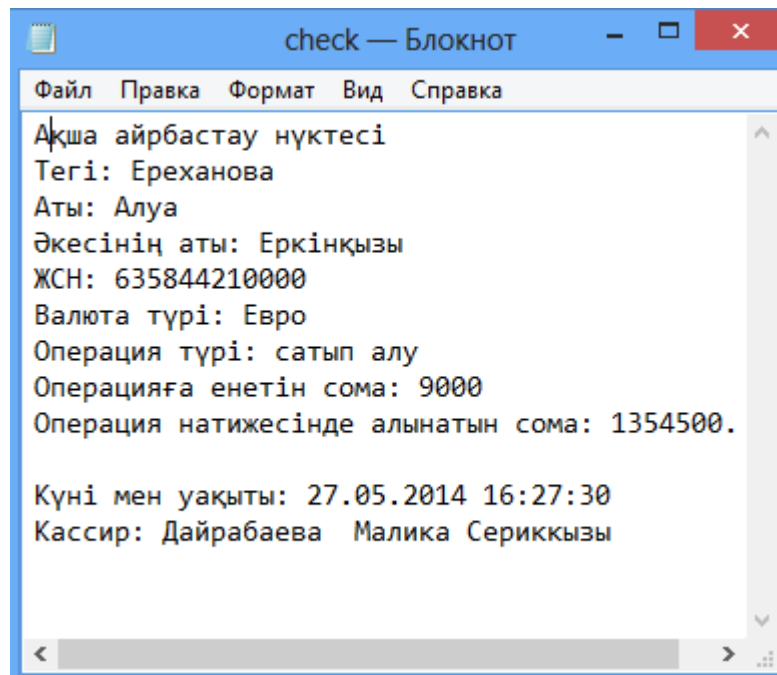


Сурет 3.16 – Бағдарламаның кассир бөлімі



Сурет 3.17 – Бағдарламаның кассир бөлімінің операцияны сәтті орындау хабарлама терезесі





Сурет 3.18 – Орындалған операция бойынша чек шығару

## **4 Тіршілік қауіпсіздігі**

### **4.1 Жасанды жарықтандыру жүйесін есептеу**

#### **4.1.1 Ақша айырбастау пунктiнiң еңбек ету шарттарының талдауы**

Бұл дипломдық жұмыста айырбастау пунктіндегі ақпараттар қозғалысының бақылау жүйесін автоматтандыру, дерекқор негізінде клиенттік программасын жоспарлау қарастырылады.

Жобаны жасау барысында компьютер негізгі рөл атқарады. Сондықтан, жұмысшылардың компьютермен жұмыс істеген кездегі еңбектің қауіпсіздігімен жұмысқа қабілеттілігін сақтауымыз қажет. Еңбек қорғау саласы бойынша бұл технологияға қатысты жасанды жарықтандырудың сипаттамалары мен әдістерін, түрлерін қысқаша сипаттап жазып есеп жүзінде дәлелдеуге тырыстым. Адам ағзасы жұмыс жасағандағы шартты түрдегі негізгі фактор болып жарықтандыру болып табылады. Қоршаған орта арқылы түсетін ақпараттың 90% көру мүшесі арқылы адам ағзасы ала алады. Еңбек қауіпсіздігі көз- қарасы бойынша көру мүмкіндігі мен көруге арналған жайлылық адам ағзасы үшін өте маңызды болып табылады. Жақсы жарықтандырудың болмауы еңбек шартының жағдайы жақсы болмауына кепілдік береді. Сондықтан да жарықты еңбек шартымызға ыңғайлы етіп пайлануымыз қажет. Жұмысшылардың еңбек жұмысы кезінде немесе құрылғылар арқылы қызмет көрсету кезіндегі қанағаттандырмайтын жарықтандыру немесе қателіктер, жарақаттану, сондай-ақ жұмыс сапасы мен еңбек өнімділігінің төмендеуіне алып келеді. Өндіріс орындарын жарықтандыру кезінде географиялық ендікке байланысты, жыл мезгілі және тәулік , атмосфераның бұлттылығы мен мөлдірлігіне тәуелді, тікелей күн көзі және көкжиектен түсетін жарықтандыруды табиғи жарықтандыру деп атаймыз, ал жасанды жарықтандыру деп электр көзінен түсетін жарықтандыруды атаймыз; шамасы бойынша жасанды жарықтандыру жетіспеген жағдайда жасанды жарықтандырумен толықтыратын жарықтандыруды біріктірілген жарықтандыру деп атаймыз.

Жасанды жарықтандыру түрлері. Жасанды жарықтандыру конструктивтік жасалу бойынша үш түрге бөлінеді-жалпы, жергілікті (локальді) және біріккен. Жалпы жарықтандыру жүйесін бөлменің барлық аумағын (сварка, гальваникалық цехтар) қамтитын жұмыс барысында қолданылатын жарықтандыруды айтамыз, сондай-ақ административтік, конторлық, складтық бөлмелер жатады. Жалпы бірқалыпты жарықтандыру (жұмыс орнының орналасуына қарамастан бөлменің бүкіл ауданына бірқалыпты болып таралады) және жалпы жергілікті жарықтандыру (жұмыс орнының орналасуын ескере отырып таралатын ) деген түрге бөлінеді. Нақты көру жұмыстарымен айналысатын (мысалы, темір ұстасы), сондай-ақ құрылғылар қараңғы, көлеңкелі жерлерде орналасуына байланысты жарықтандыру

ұйымдастырылуын жергілікті жарықтандыру деп атаймыз. Жергілікті және жалпы жарықтандыру қосындысын біріккен жарықтандыру деп атайды. Менің жұмысым бойынша жасанды жарықтандырудың барлық түрлерін қолдануға болады, менің ойымша біріккен жарықтандыруды қолдану керек. Ал жұмыс орнында жергілікті жарықтандыруды қолданылуына жол берілмейді, өйткені ол өткір көлеңке туғызып соның салдарынан көру қабілеті тез шаршатып өндіріс қабілеттілікті төмендетуге алып келеді

Жұмыс мақсаты-тапсырмаға байланысты жекелеген өндіріс бөлмесінің жасанды жарықтандырылуын орындап есеп жүзінде дәлелдеу,  $E_{\min}$  қажетті жарық шамдарының санын анықтап нормаланған жарықтандыруды есептеу. Жалпы айырбастау орындарын салыстыра отырып, мынадай өлшемдер алдым.

Бөлменің өлшемдері келесі: Ұзындығы  $L = 4$  м, ені  $B = 3$  м, биіктігі  $H = 2,50$  м. Бөлме ішінде 2 жұмыс орны және 1 шкаф, 2 терезе, 1 өрт сөндіргіш бар. Бөлме жоспары 4.1-суретте көрсетілген. Бөлмеде 2 адам жұмыс істейді: 1-администратор, 1-кассир. жұмыс графигі - аптасына бес күн, күніне сегіз сағат.

Қызмет көрсетушілердің қауіпсіздігін қамтамасыз ету үшін станцияның қызметкерлеріне әсер ететін барлық мүмкін факторларды талдау қажет. Жұмыс бөлмесінің бір жақтық екі терезе ретіндегі әр қайсысы 1,1м табиғи жарықтандыруы бар.

Қаралып жатқан бөлмеде жұмыс істеуге қажетті жарықтандыру қалып бойынша  $E_n=200$ лк, осылайша бөлмедегі қажетті жарықтандыруды қамтамасыз етеміз.

Бөлмеде құрылғылардың жұмысын жүзеге асыру үшін 2 дербес компьютер қолданылады.

Құрылғының техникалық көрсеткіштері:

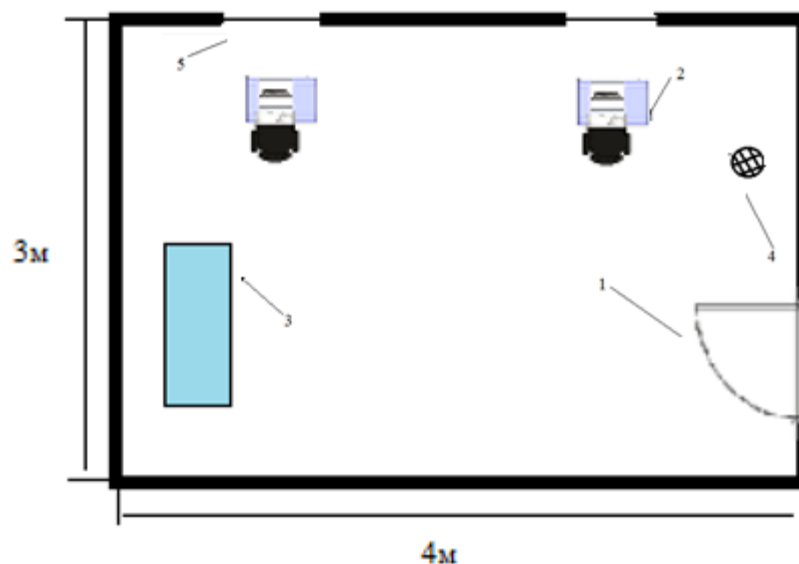
– дербес компьютер Intel Core i5-3317U Processor E7500 2.60 GHz (SIS751+SB+VGA)/2GB DDR II/160 Gb SATA Seagate/SP/CD 400W;

– монитор 17" LG LI720 0.26dpi;

– электрқоректенуі – айнымалы ток 220-250 В, жиілігі 60 Гц, қуаты 400

Вт.

ГОСТ 12.1.005–88 стандартына байланысты ССБТ "Жұмыс аумағының ауасы, жалпы санитарлық гигиеналық талаптар", бөлмедегі жұмысшылардың жұмыс істеу категориясы 1а жеңіл физикалық жұмыс категориясына жатады, өйткені құрылғыны аралықтан компьютер арқылы басқару болады. Адам организміне жұмыс категориясы бойынша энергия шығару 4.1-кестеде көрсетілген.



Сурет 4.1 – Бөлме жоспары: 1 – есік, 2 – компьютер, 3-шкаф, 4-өрт сөндіргіш, 5 – терезе;

Кесте 4.1 – Адам организмінің жұмыс категориясы бойынша энергия шығыны

Жұмыс категориясы	Категория	Адам организмінің энергия шығыны, Ккал/сағ.	Жұмыс сипаттамасы
Жеңіл физикалық жұмыс 1а	1 а	138-172	Жұмыс отыру күйінде өтеді

Жұмыс орнындағы нормаланған климаттық параметрлер 4.2 -кестеде келтірілген.

Кесте 4.2 – Температураның, ылғалдылық пен ауа қозғалысының нормалары ГОСТ 12.1.055.88ССБТ

Жұмыс мезгілі	Ауа температурасы, °С	Салыстырмалы ауа ылғалдылығы, %	Ауа қозғалысының жылдамдығы, $\frac{м}{с}$
Суық	22 - 24	40–60	0,1
Жылы	23 - 25	30–50	0,2

Жазғы микроклиматтың қысқыға қарағанда, ауа баптауы сәл жоғарырақ. Кесте бойынша, ауа температурасы 23-25°, ылғалдылығы 30–50%, ауа

қозғалысының жылдамдығы 0,2м/с. 55% дейінгі ауа ылғалдылығы құрғақ деп есептеледі, 56-70% - деңгейінде орташа құрғақ, 71-85% - орташа ылғалды және 85% жоғары – тым ылғалды. Ең үлкен ылғалдылық пен салыстырмалы ылғалдылық арасындағы айырым қанығу тапшылығы деп аталады. Қанығу тапшылығы қаншалықты үлкен болса, соншалықты ылғалды ауа жұтады. Ауаның температурасы 23-24<sup>0</sup>С – де ауа қозғалысының жылдамдығы 0,15 м/с дейін болған жөн.

Кәсіпорындық ғимараттарды жарықтанудың жоғары болуы жұмыс жағдайына дұрыс әсерін тигізеді.

Авариялық жарықтану, жалпы жарықтанудан 5%-ке аз жарықтануды қамтамасыз етуі керек, бірақ ғимарат ішінде оның деңгейі 2 лк-дан аз болмауы керек.

Эвакуациялық жарықтану, адамның қауіпті деген өтетін жерлеріне орналасады, ол өтетін жолдардың еденінде, баспалдақтарда орналасады, оның деңгейі ғимарат ішінде 0,1лк, ал ашық территорияда 0,2 лк болуы керек.

Жұмыс орнындағы нормаланған жарықтандыру параметрлері 4.3 - кестеде келтірілген

К е с т е 4.3 – Жарықтану нормасы

Жұмыс орны	Жарықтандыру беті, еденнен биіктігі	Жарықтану нормасы, лк		K <sub>п</sub> , % артық емес
		Комбинациялық жарықтану	Жалпы жарықтану	
Машиналық зал	Г – 0,8	750	400	15
ЭЕМ эксплуатациялық зал	Г – 0,8	750	400	15
Техникалық қызмет көрсету залы	Г – 0,8	750	400	15
Ақпарат тасығыш құралдарды сақтау залы	–	–	300	20

Біздің жағдайымызда электр қауіпсіздігін қамтамасыз ету үшін ГОСТ 12.1.030-81 бойынша жерлендіру мүмкіндігін қарастыру қажет. Біздің жағдайымыздағы кернеу 220В, сондықтан жерлендіру мен нөлдеу міндеттелмейді, бірақ ұсынылады. Жерлеуді электрқауіпсіздігі мақсатымен қолданады. Ал өрт қауіпсіздігі үшін бөлмеде өрт сөндіргіш қойылған.

Бөлмедегі шуды азайту үшін шу көзін жауып, қашықтатып отыруға болады. Шуды көп бөлетін технологиялық құрылғыларды айырбастау мүмкін

болмағанда дыбыс шағылғыштар, дыбыс тұтқыштар қолданады. Менің алынған жобам бойынша шу шығаратын зат ол дербес компьютерлер. Ал есептеу орталықтарының өндірістік бөлмелерінде пайда болатын шу көздерін азайтуды қоршайтын конструкцияларда дыбысты оқшаулауды арттыра отырып, қол жеткізуге болады.

#### 4.1 Жұмыс орынның жасанды жарықталуы

Өндірістік кәсіпорындарындағы жасанды жарықталуының шарттары адамның көру жұмыс қабілеттігіне, физикалық және моральдық күйіне үлкен әсер етеді, ал бұл еңбек өнімділігіне және өнімнің сапасына әсер етуіне әкеледі. Еңбектің жағымды шарттарын құру үшін өндірістік жарықталуы келесі талаптарға сай болу керек:

- жұмыс орындағы жарықталуы тазалық нормаға сәйкес болу керек;
- жұмыс бетіндегі және қоршаған кеңістіктің шектегі жарық мүмкіндігінше біркелкі таратылу керек;
- жұмыс бетіндегі кенет көлеңкенің болуы жарықтың біркелкі емес таратылуына әкеледі, сондықтан оларды мүмкіндігінше жою керек;
- көру алаңында жарқылдық (тура немесе шағылысқан) болмау керек.

#### 4.2 Жарықтың көзін таңдау

Жасанды жарық үшін қызу және газоразрядтық шамдар қолданылады. Жалпы жұмыстық қыздыру шамдардың типтерін белгілеу шартындағы келесі әріптер мынаны білдіреді: В- вакуумдық, Г- газбен толтырылған, Б- биспиральдық, БК- биспиральдық криптондық. Шамдардың жарықтық ағымдары, жалпы жұмыстық қыздырушылық және олардың қуаты 4.3-кестеде келтірілген.

К е с т е 4.3 – Қыздырушылық шамның қуаты мен жарықтық ағым

Шамның типі	Қуат Вт	220	200-235	Шамның типі	Қуат Вт	200	200-235
		кернеуіндегі жарықтық ағым				кернеуіндегі жарықтық ағым	
В	15	105	85	Б	150	2100	1840
В	25	220	190	Г	200	2800	-
Б	40	400	300	Б	200	2920	2540
БК	40	460	-	Г	300	4600	4000
Б	60	716	550	Г	500	8300	7200
БК	60	790	-	Г	750	13100	-
Б	100	1350	1090	Г	1000	18600	-
БК	100	1450	-	Г	1500	2900	-
Г	150	2000	-				

Кіші қысымды газоразрядтық шамдар ең үнемді болып саналады (ЛД, ЛДЦ, ЛХБ және т.б.). Газоразрядтық шамдардың кейбір техникалық сипаттамалары 4.4- кестеде келтірілген.

Кесте 4.4 – Кіші қысымды газоразрядтық шамдардың кейбір сипаттамалары

Номиналдық қуат, Вт	Номиналды жарықтық ағым Лм., шамдар типі					Шамның өлшемі, мм.	
	ЛДЦ	ЛД	ЛХБ	ЛТБ	ДБ	Диаметр	Штырьдың ұзындығы
15	500	590	675	700	760	27	451,6
20	820	920	935	975	1060	40	604,0
30	1450	1640	1720	1720	2100	27	908,8
40	2100	2340	3000	3000	3120	40	1213,6
65	3050	3570	3820	3980	4650	40	1514,2
80	3740	4070	4440	4440	5220	40	1514,2

### 4.3 Жасанды жарықтандыруды есептеу

Жарықтандыру табиғи және жасанды болуы мүмкін. Жасанды жарықтандыру екі жүйелі болуы мүмкін: жалпы және аралас. Аралас жарықтандыру кезінде жалпыға жергілікті жарықтандыру қосылады.

Жасанды жарықтандыруды есептеу үшін пайдалану коэффициенті және нүктелік әдістер қолданылады. Пайдалану коэффициенті әдісі үлкен көлеңке түсіретін заттар болмаған жағдайдағы біркелкі горизонталь беттердегі жарықтандыруды есептеуге арналған. Бұл әдіс арқылы  $\eta$  коэффициентінің мәні анықталады. Ол есептелінетін бетке түсетін жарық ағынының жарықтандыру құралының толық ағынына қатынасы болып табылады.

Біздің есептейтін бөлменің өлшемдері мынадай:

- габариттері: 4x3x2,5;
- шағылу коэффициенттері –  $\rho_{\text{төбе}} = 70\%$ ,  $\rho_{\text{кабырға}} = 50\%$ ,  $\rho_{\text{еден}} = 30\%$ ;
- жарықтандырудың бірқалыпсыздық коэффициенті  $\lambda = 1,1 \div 1,2$ ;
- еденнен жұмыс орнына дейінгі биіктік:  $h_{\text{еден}} = 0,8$  м.

Нормаланған жарық қабырғалары, төбесі әктелген, терезелері жабылмаған бөлме үшін  $E=200$  лк. Жарықтандыру жүйесіне люминесцентті ЛД шамдарын қолданамыз, оның қуаты 40 Вт, ал жарықтық ағыны 2340 лм.

Ілінудің есептік ұзындығын анықтаймыз. Жұмыс орнының еденнен биіктігі  $h_{\text{еден}} = 0,8$  м, ал шамның іліну биіктігі  $h_{\text{іліну}} = 0,2$  м болғандықтан, ол келесі шамаға тең.

$$h = h - (h_{\text{еден}} + h_{\text{іліну}}), \quad (4.1)$$

$$h = 2,50 - (0,8 + 0,2) = 1,5 \text{ м.}$$

Шырақтар арасындағы ең қолайлы қашықтық төмендегі өрнекпен анықталады.

$$z = \lambda \cdot h, \quad (4.2)$$

$$z = 1,2 \cdot 1,5 = 1,8 \text{ м.}$$

Шырақтардың қатарлар санын санайық.

$$n = \frac{B}{L}, \quad (4.3)$$

мұндағы  $B$  – бөлме ені,  $B = 3 \text{ м}$ ;

$L$  – шырақтар арасындағы қашықтық,  $L = 1,8 \text{ м}$ .

Осыдан

$$n = \frac{3}{1,8} = 1,7 \approx 2.$$

Сәйкесінше шырақтарды бір немесе екі қатарға орналастыруға болады.

Бөлмеге 2 қатар шырақ орнатамыз, қатарларының ара қашықтығы 1,8 м, ал қабырға мен шырақтар арасындағы қашықтық 1,2 м.

Бөлме индексін анықтаймыз

$$i = \frac{S}{h(A+B)}, \quad (4.4)$$

мұндағы  $A$  – бөлме ұзындығы;

$B$  – бөлменің ені;

$S = A \cdot B = 12 \text{ м}^2$  – ауданы.

$$i = \frac{12}{1,5 \cdot (4 + 3)} = 1,14.$$

Кесте бойынша пайдалану коэффициенті  $\eta = 45\%$ , қор коэффициенті  $K_k = 1,5$ .

Қажетті шамдар санын анықтаймыз.

$$N = \frac{E \cdot K_k \cdot S \cdot Z}{n \cdot \Phi_l \cdot \eta}, \quad (4.5)$$

мұндағы  $E$  – берілген жарықтық;

$K_k$  – қор коэффициенті;

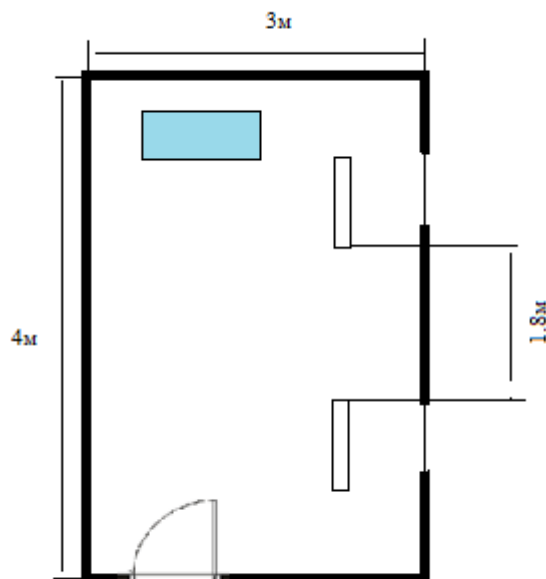
$S$  – бөлменің ауданы,  $\text{м}^2$ ;

$Z$  – жарықтандырудың бірқалыпсыздық коэффициенті.

$$N = \frac{200 \cdot 1,5 \cdot 12 \cdot 1,1}{2 \cdot 2340 \cdot 0,45} \approx 2 \text{ шам.}$$

Әр қатарға шам орнатылады, олардың арасындағы қашықтық (шамның ұзындығын ескереміз) 1,8 м, ал қабырға мен шырақтар арасындағы қашықтық 1,2 м. Шамдардың бөлме ішінде орналасу сұлбасы 4.2-суретте келтірілген.





Сурет 4.2 – Жасанды жарықтандыру сұлбасы

Нормаланған 200 лк жарықтандыруды қамтамасыз ету үшін қуаты 40Вт болатын ЛД40-4 типті екі шам қажет. ЛД40-4 типті шамдар қажетті есептелінген жарық ағынын (2340 лм) қамтамасыз етеді. өзге шамдарға қарағанда құрылымы мен сыртқы көрінісі желінің номиналды кернеуіне (220В) және бөлме ішіндегі қоршаған орта шарттарын орындай алады. Бұл түрдегі шамның пайдалы әсер коэффициенті 85% - ке тең.

#### 4.4 Меншікті қуат әдісімен жарықтандыруды есептеу

Меншікті қуат әдісімен жарықтандыруды шешудің маңызы, шырақтардың түрі мен оны орнатудың орнына байланысты, жұмыс бетінен ілгіштің биіктігі, горизонтальды беттегі жарықтандыру мен бөлменің ауданымен меншікті қуат мәні анықталаумен қорытындыланады.

ЛДР шырақтарын орнатуды қабылдаймыз және параллель екі қатарға, бір-бірден орналыстырамыз.

$$N = 2 * 1 = 2 \text{ дана.}$$

Шырақтың меншікті қуаты  $\omega = 40 \text{ Вт/м}^2$ .

Бір шырақтың қуаты:

$$\rho = \frac{\omega \cdot S}{N} = \frac{40 \cdot 12}{2} = 240 \text{ Вт.} \quad (4.6)$$

Екі шырақты орнатамыз. Әр шамның қуаты:

$$\rho_g = \frac{240}{2} = 120 \text{ Вт}$$

Сәйкесінше, екі әдіспен келтірілген есептер негізінде екі шырақты тандап аламыз, олардың әр қайсысы 120 Вт тұрады.

#### 4.5 Өнеркәсіптегі жасанды жарықтандыру

Өнеркәсіптегі жарықтандыру адамның көру жұмыс әрекетіне, физикалық және моралді жағдайына, яғни жұмыс өнімділігіне, сапасына және өнеркәсіптік жарақаттануға үлкен әсерін тигізеді.

Еңбек етудің қолайлы жағдайын жасау үшін өнеркәсіп бөлмелері келесі талаптарға жауап беруі керек:

- жарықтану гигиеналық нормаларға сай болу керек;
- жұмыс бетіндегі және айналадағы жарық бірқалыпты болу керек;
- көлеңкеге өтуі күрт болмауы тиіс;
- жұмыс бетінен шағылысулар болмау керек;
- дұрыс жарық берілу үшін жарықтың қажетті спектралді құрамы қамтамасыз етілуі керек.

Өндіріс бөлмесін құру жарық көзін таңдаумен, оларды орналастырумен, жарықты-техникалық есептеулермен және жарықтандыру құрылғыларын анықтаумен сипатталады.

Сызықты – аппарат бөлмесіне ең қолайлысы ЛД типті люминесцентті шам болып табылады. Төбеге бекітілетін шамдар шаңдануды аздату үшін төбеден 50 мм-ден аспай ілінуі керек. Шам қалпақшалары өткізу коэффициенті 0,7-ден кем болмайтын сәуле шағылыстырушы материалдан жасалады.

Бөлменің жарықтану нормасы ( $E_{\min}$ ) сол бөлмедегі минималды объектке қатысты жасалынатын жұмысқа қатысты көру жұмысының разрядына байланысты.

## 5 Техника-экономикалық негізделуі

### 5.1 Жоба сипаттамасы және қажеттілігінің негіздеуі

Жұмысымының мақсаты - айырбастау пунктіндегі ақпараттар қозғалысын бақылау жүйесін құру болып табылады. Бұл бағдарлама ыңғайлы әрі уақытты үнемді пайдалануға мүмкіндік береді, сонымен қатар, жұмыс барысында мүмкін болатын қателерді азайтады.

Бағдарламаның тиімділігіне мыналарды жатқызуға болады:

- аз уақыт ішінде бағдарламаны игеріп, оңай пайдалануға мүмкіндік береді;
- айырбастау орындарында бақылау жұмысын тездігін қамтамасыз етеді;
- бағдарлама Windows XP, Windows 7 және Windows8 операциялық жүйелерінде жұмыс істейді.

Жұмысты жазу барысында деректер қорының құрылымын және клиенттік қосымшаны жасау жұмыстары жүргізілген және енгізілген жүйеден алынған экономикалық пайданы есептеу жүргізілген. Бұл бағдарламада мекеме үшін ақпараттық жүйесін жобалауға өте қолайлы. Бұл интеллектуалды еңбек пен қатар өте көптеген қағаздармен жұмысты қысқартады және экономикалық жағынан да өте тиімді.

### 5.2 Маркетингтік стратегия

Маркетингтік стратегияның мақсаты – бағдарламалық қамтамасыз ету арқылы пайда табу.

К е с т е 5.1 - Жоспарланып отырған пайда мың.теңге

Жылдар	2014	2015	2016	Барлығы,мың тг
Жоспарланған табыс, барлығы,мың.тг	95148	190296	285444	570888

Бағдарламаның сатылым жоспары

Біздің стратегия «Айырбастау пунктіндегі ақпараттар қозғалысын бақылау жүйесін құру» ДҚ-ын автоматандырылған жүйеге келтіру болып табылады. Бұл бағдарлама толығымен қолданылса, айырбастау пунктерінің жұмысын жеңілдетеді. Бағдарлама валюта өзгерісін бақылап, кассир жұмысын анықтап және тұрақты тұтынушыларды қамтамасыз етеді. Қазіргі таңда осындай бағдарламаларды қолдану жұмыс көзін жылдам, әрі тез дамытуға септігін тигізеді.

Бірінші жылы тауарлық белгіні кең түрде тарату және тұрақты тұтынушылардың санын арттыру мақсатында 570888 теңге қаржы жұмсауды жоспарлап отырмыз.

### 5.3 Бағдарламамен қамтамасыз етудегі еңбек сыйымдылығын есептеу.

Айырбастау пунктіндегі ақпараттарды бақылау жұмыстарын автоматтандыру бағдарламасы– интеллектуалды, техникалық және қаржылық шығындардан тұратын қиын және көп еңбекті қажет ететін процесс. Сондықтан өнімді жүзеге асыру шығындарын есептеу қажет. Еңбек шығыны құрамдасын есептеудегі базалық көрсеткіш мына формуламен есептелінеді:

$$Q = q \times c, \quad (5.1)$$

$q$  коэффициентінің мәні менің тапсырмам бойынша- ол есептеу тапсырмалары болып табылады, яғни  $q=1405$  деп аламын (А-қосымшасынан)

Ал 2-кесте бойынша «с» коэффициентін,яғни жобамның күрделілігімен жаналығы бойынша  $c=1,08$  деп алдым.(А-қосымшасынан)

$$Q = q \times c = 1405 * 1.08 = 1517,4$$

Ары қарай бағдарламалық өнімді әзірлеуге кететін уақытты есептеу керек.

1)  $T_{\text{ПО}}$  (мақсат сипатын дайындау уақыты), нақтылы деректер бойынша алынады және келесі мәнге тең деп алынады (3-тен 5 күнге дейін, 8 сағаттан):

$$T_{\text{ПО}} = 24 \text{ адам/сағ.}$$

2)  $T_{\text{O}}$  (мақсат сипаттамасы уақыты) келесі формуламен анықталады:

$$T_{\text{O}} = Q \times B / (50 \times K), \quad (5.2)$$

мұндағы  $B$  – мақсат есебі өзгерісінің коэффициенті,  $B$  коэффициенті мақсат күрделілігіне және өзгеріс санына тәуелді – 1,2-ден 1,5-ке дейін (2- кестені қара (А-қосымшасынан));

$K$  – бағдарлама жасаушы білектілігін ескеретін коэффициент  $K=0,8$  (А-қосымшасынан).

$$T_{\text{O}} = 1405 \times 1,5 / (50 \times 0,8) = 52,68 \text{ адам/сағ.}$$

3)  $T_{\text{A}}$  (алгоритм құруға кеткен уақыт) мына формуламен есептейміз:

$$T_A = Q / (50 \times K), \quad (5.3)$$

$$T_A = 1517,4 / (50 \times 0,8) = 37,9 \text{ адам/сағ.}$$

4)  $T_{BC}$  (блок – сұлба құруға кеткен уақыт)  $T_A$  сияқты 5.3 формуламен есептеледі.

5)  $T_H$  (бағдарламаның тілінде жазуға кеткен уақыт) келесі формуламен анықталады:

$$T_H = Q \times 1,5 / (50 \times K), \quad (5.4)$$

$$T_H = 1517,4 \times 1,5 / (50 \times 0,8) = 56,90 \text{ адам/сағ.}$$

6)  $T_{II}$  (бағдарлама теру уақыты) келесі формуламен анықталады:

$$T_{II} = Q / 50, \quad (5.5)$$

$$T_{II} = 1405 / 50 = 28,1 \text{ адам/сағ.}$$

7)  $T_{OT}$  (бағдарламаны реттеу және тестілеу уақыты) келесі формуламен анықталады:

$$T_{OT} = Q \times 4,2 / 50 \times K, \quad (5.6)$$

$$T_{OT} = 1405 \times 4,2 / 50 \times 0,8 = 147,5 \text{ адам/сағ}$$

8)  $T_D$  (құжаттарды рәсімдеу уақыты), нақтылы деректер бойынша алынады және құрылады (3-тен 5 күнге дейін, күніне 8 сағат):

$$T_D = 24 \text{ адам/сағ.} \quad (5.7)$$

Еңбек шығындарының сомасы еңбек шығынының құрама сомасы ретінде 5.7-формуламен есептеледі:

$$T = T_{II} + T_O + T_A + T_{BC} + T_H + T_{II} + T_{OT} + T_D, \quad (5.8)$$

$$T = 24 + 52,68 + 37,9 + 37,9 + 56,90 + 28,1 + 147,5 + 24 = 408,98 \text{ адам / сағ.}$$

## 5.4 Бағдарламалық қамсыздандыру шығынының есебі

Айырбастау пунктіндегі ақпараттар жүйесін құру ДҚ-ын автоматтандыру бағдарламасы– интеллектуалды, техникалық және қаржылық шығындардан тұратын қиын және көп еңбекті қажет ететін процесс. Сондықтан бағдарламалық қамсыздандыру шығыны ішіне еңбек ақы шығыны да, еңбек ақыдан аударылымдар, амортизациялық және тағы да басқа шығындар кіреді, олар мынандай формуламен анықталады:

$$C = EAK + O_{\text{нег}} + A + C_{\text{ЭЭ}} + C_{\text{МЖК}} + C_{\text{ТО}} + C_{\text{ПР}} + C_{\text{Н}}, \quad (5.9)$$

Еңбек ақы екі жасаушыдан құрылады: негізгі еңбек ақы және қосымша еңбек ақы сомасы (немесе еңбек ақы қоры, *EAK*) негізгі еңбек ақы және қосымша еңбек ақы сомасы мына формуламен есептеледі:

$$EAK = Z_{\text{нег}} + Z_{\text{кос}}, \quad (5.10)$$

$$EAK = 122694 + 24538,8 = 147232,8 \text{ тг.}$$

Негізгі еңбек ақы төмендегідей анықталады:

$$Z_{\text{нег}} = T \times TC / t_{\text{орт}}, \quad (5.11)$$

$$Z_{\text{нег}} = 408,98 \times 2800 \times 2,25 / 21 = 122694 \text{ тг.}$$

ТС-тарифтік мөлшерлеме (А-қосымшасынан).

Қосымша еңбек ақы негізгі еңбек ақының 20 % құрайды және келесі формуламен есептеледі.

$$Z_{\text{кос}} = 0,2 \times Z_{\text{нег}}, \quad (5.12)$$

$$Z_{\text{кос}} = 0,2 \times 122694 = 24538,8 \text{ тг}$$

Әлеуметтік салық *EAK* 11 % құрайды (ҚР СК 358 б. 1-тарау) жұмыскердің табысынан, мынандай формуламен есептеледі:

$$O_{\text{СН}} = (EAK - ZA) \times 11\%, \quad (5.13)$$

$$O_{\text{СН}} = (147232,8 - 14723,28) \times 0,11 = 14576,04 \text{ тг.}$$

мұндағы *ZA* – зейнетақы аударылымдар, *EAK*-нан 10% құрайды және әлеуметтік салықпен міндеттелмейді:

$$3A = EAK \times 10\%, \quad (5.14)$$

$$3A = 147232,8 \times 0,1 = 14723,28 \text{ тг.}$$

Амортизациялық аударылымдар амортизацияның тағайынды шамаларымен орындалады, пайыздармен жабдықтың баланстық құнына және мына формуламен есептеледі:

$$A = \frac{B_{\text{бас}} \times A_{\text{ш}} \times N}{100 \times 12 \times t}, \quad (5.15)$$

$$A = \frac{150000 \times 0,23 \times 17,04}{100 \times 12 \times 308,3} = 1,58$$

К е с т е 5.2 – Амортизациялық шығындары

Құрылғы атауы	Саны	Бірлік үшін бағасы, (НДС-сіз),тенге
Ноутбук Acer Aspire V5 Core i 5-2450M/4Gb/750Gb	1	150000
МФУ Canon Принтер/Копир/Сканер 30К,128MB,GDI USB	1	250000
Барлығы	2	175000

Амортизация шамалары ( $A_{\text{ш}}$ ), мына формуламен есептеледі:

$$A_{\text{ш}} = \frac{B_{\text{бас}} - K_{\text{тар}}}{T_{\text{норм}} \cdot B_{\text{бас}}} \times 100\%, \quad (5.16)$$

$$A_{\text{ш}} = \frac{150000 - 8750}{4 \cdot 150000} \times 100\% = 23,54 \%$$

мұндағы  $K_{\text{тар}}$  – таратылым құны, жабдықтың құнынан 5% құрайды.

$T_{\text{норм}}$  – жабдықтың нормативтік қызмет ету мерзімі (дербес компьютер үшін – 4 жыл).

Дербес компьютерде жалпы жұмыс істеу уақыты мына формуламен есептелегенді:

$$T = T_A + T_{\text{БС}} + T_{\text{Н}} + T_{\text{П}} + T_{\text{От}}, \quad (5.17)$$

$$T = 37,9 + 37,9 + 56,90 + 28,1 + 147,5 = 308,3 \text{ адам/сағ.}$$

Электрэнергия шығындары мына формуламен есептеледі:

$$C_{\text{ЭЭ}} = K \times k_3 \times T \times C_{\text{кВт-сағ}}, \quad (5.18)$$

$$C_{\text{ЭЭ}} = 0,45 \times 0,8 \times 19,37 \times 8 = 55,78 \text{ тг.}$$

мұндағы  $K$  – ЭЕМ қуаты (450 Вт);

$k_3$  – жүтеме коэффициенті (0,8);

$C_{\text{кВт.с}}$  – 1 кВт-сағ электрэнергиясының құны (А-қосымшасынан);

$T$  – жұмыс уақыты, сағ.

Материалдар мен көмекші бөлшектер шығыны, бағдарламалық өнімді жазу барысында қолданылды ( $C_{\text{МжК}}$ ), сонымен қатар техникалық қызмет көрсету шығыны ( $C_{\text{ТО}}$ ), жабдықтың құнынан 1,5% және 2,5% құрайды және мына формулалар мен есептеледі (5.19 – 5.20):

$$C_{\text{МжК}} = 0,015 \times C_{\text{обор.}} = 0,015 \times 150000 = 2250 \text{ тг} \quad (5.19)$$

$$C_{\text{ТО}} = 0,025 \times C_{\text{обор.}} = 0,025 \times 150000 = 3750 \text{ тг} \quad (5.20)$$

Басқару мен қызмет көрсетуге байланысты үстеме шығындар, сондай-ақ жабдықты пайдалану кезіндегі және де кәсіпорын үдерістері мен айналымдарынан қосымша шығындар еңбек ақы қорынан 50% құрайды және де мына формуламен есептеледі:

$$C_{\text{Н}} = 0,5 \times \text{ЕАҚ} = 0,5 \times 147232,8 = 73616,4 \text{ тг} \quad (5.21)$$

К е с т е 5.2– Өзіндік құнның қорытынды кестесі

Шығын бабы атауы		Сомасы, теңге	Әр баптың үлесі, %
ЕАҚ	ЕА <sub>нег</sub>	122694	46,46
	ЕА <sub>қос</sub>	24538,8	9,29
Үстеме шығындар, $C_{\text{Накл}}$		73616,4	27,88
Әлеуметтік салық шығыны, ӘС		1822,01	0,69
Пайдалану шығындары	$C_{\text{ЭЭ}}$	55,78	0,02
	$C_{\text{ТО}}$	3750	1,42
	$A_{\text{жыл}}$	35310	13,37

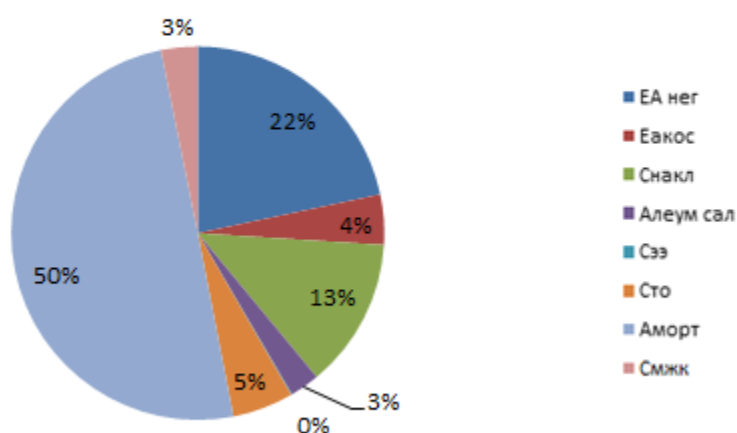


Кесте 5.2-нің соңы

Материалдар және көмекші, $C_{МжК}$	2250	0,85
Барлығы:	264036,99	100

Бағдарламалық өнімнің өзіндік құнының есебінің жиынтық нәтижелері диаграмма түрінде:

### Өзіндік құнның нәтижесі



Сурет 5.1– Бағдарламалық өнімнің өзіндік құнының есебінің жиынтық нәтижелері

Бағдарламалық өнімді жүзеге асыру бағасы оның құны мен таза кірістің қосындысынан тұрады:

$$Ц = C + П, \quad (5.22)$$

мұндағы  $C$  – өнім бағасы;

$П$  – таза кіріс.

Бастапқы бағаны анықтауда бағдарламалық өнімді жүзеге асыру үшін керекті рентабельдік деңгейін анықтау қажет (20%):

$$Ц_{п} = C \cdot \left(1 + \frac{P}{100}\right), \quad (5.23)$$

мұндағы,  $P$  – рентабельдік (20%).

$$Ц_{п} = 264036,99 * (1 + 20/100) = 316844,388 \text{ тг.}$$

Бағдарламаның орындалу бағасы келесі формула арқылы табылады.

$$Ц_{р} = Ц_{п} + ҚҚС, \quad (5.24)$$

Қазіргі таңда ҚР-да НДС 12% құрайды:

$$ҚҚС = Ц_{п} \cdot 12\%, \quad (5.25)$$

$$ҚҚС = 316844,388 * 12\% = 38021,32 \text{ тг.}$$

$$Ц_{р} = 316844,388 + 38021,32 = 354865,71 \text{ тг.}$$

### 5.5 Бағдарлама өнімін сатып алуға кеткен бір жолғы шығындар есебі

Бағдарлама өнімін сатып алуға және оны өндіріске енгізу шығындары келесі шығындардан тұрады:

$$\Sigma З = C_C + C_{TP} + C_O \quad (5.26)$$

$$\Sigma З = C_C + C_{TP} + C_O = 354865,71 + 88716,42 + 86400 = 529982,13 \text{ тг}$$

Бағдарламаның бағасы:  $C_C = 354865,71 \text{ тг}$

$C_{TP}$  – көлік шығыны, жүйе құнынан - 25 %, мың тенге;

$$C_{TP} = 0,25 * C_C = 354865,71 * 0,25 = 88716,42 \text{ тг}$$

Жүйе құны үстінде есептелінді, ал қалғандары келесі түрде есептеледі. Өнімді игеруге деген маманды оқыту шығыны, оқытуға кеткен уақыт пен оған деген консалтингті фирмадағы мөлшерлемеден тұрады:

$$C_O = T \times C_{OP}, \quad (5.27)$$

$$C_O = T \times C_{OP} = 4 * 8 * 2700 = 86400 \text{ тг}$$

мұндағы  $T$  – оқытуға кеткен уақыт, сағ.;

$C_{OP} = 2700 \text{ тг.}$  Себебі қазіргі таңда сағатына 2500-3000 тг аралығында.

К е с т е 5.3– Ақпарат жүйелерін енгізуге керекті бір жолымғы шығындар есебінің жиынтығы

Шығын бабы атауы	Сомасы, мың тенге
------------------	-------------------

Жүйенің құны	354865.71
<i>Кесте 5.3-нің соңы</i>	
Көлік шығыны	88716,42
Жүйені оқуға кеткен шығыны	86400
Барлығы:	529982,13

### 5.6 Игеру саласындағы жылдық бір жолғы шығындар есебі.

Ақпараттық технологияларын қолданған кездегі жылдық шығындары келесі баптардан тұрады:

- жылдық ЕАҚ;
- әлеуметтік салық аударым;
- басқа да шығындар.

Осының барлығысын енді формула арқылы жазайық:

$$C_{\Sigma} = 3\Pi + O_{CH} + \Pi_p, \quad (5.28)$$

$$C_{\Sigma} = 14112000 + 1397088 + 4233600 = 19742688 \text{ тг}$$

АТ енгізгеннен кейінгі мамандардың жылдық еңбек ақы шығындары келесі формуламен есептеледі:

$$3\Pi = (O_C \times t \times K_p) \times Ч \times (1 + K_d), \quad (5.29)$$

$$3\Pi = (3000 \times 8 \times 245) \times 2 \times (1 + 0,2) = 14112000 \text{ тг.}$$

$$O_{CH} = (EAK - 3A) \times 11\% = (14112000 - 1411200) \times 0.11 = 1397088 \text{ тг.}$$

$$3A = EAK \times 10\% = 14112000 \times 0.1 = 1411200 \text{ тг.}$$

мұндағы  $O_C$  – маманның сағаттық ақысы, 3000 мың теңге/сағ;

$t$  – жұмыс күнінің ұзақтығы, 8 сағ;

$K_p$  – жылдағы жұмыс күндер саны, 245 күн;

$Ч$  – үдеріске қатысы бар мамандар саны, адам;

$K_d$  – қосымша еңбек ақы коэффициенті, 20 %.

Басқа да шығындар – материалға деген шығындар, үстеме шығындар жылдық еңбек ақы шығындарынан 30 % құрайды және келесі формуламен есептеледі:

$$\Pi_p = 3\Pi \times 0,3 = 14112000 \times 0,3 = 4233600 \text{ тг} \quad (5.30)$$

К е с т е 5.4– Ақпараттық технологияларын қолданған кездегі жылдық бір жолғы шығындар

Шығын баптары атауы	Сомасы, мың теңге
Жылдық ЕАҚ	14112000
Әлеуметтік салық аударым	1397088
Басқа да шығындар	4233600
Барлығы:	19742688

### 5.7 Ақпараттық жүйе енгізуден үнем мен табыс мөлшерінің есебі.

Ақпараттық жүйе енгізген ұйымда үнем көзі, оны енгізуден кейін түскен пайда немесе шығын үнемделенетіні болып табылады. Ұйымда ақпараттық жүйе енгізу барысында деректерді өңдеуге, пайдалануға уақыт азайып, еңбек өнімділігі өсіп, адамдар саны азайатын үнемділікке кез келеді. Еңбек өнімділігі өсуінен түскен үнемділікті келтірінді шығындардың базалық ( $C_0$ ) және ұсынылған ( $C_1$ ) нұсқалар айырмасы ретінде шығарып алуға болады.

$$\Delta_{\text{ут}} = C_0 - C_1, \quad (5.31)$$

$$\Delta_{\text{ут}} = C_0 - C_1 = 39455376 - 19742688 = 19712688 \text{ тг}$$

$$C_0 = 3\Pi + O_{\text{сн}} + \Pi_p = 28224000 + 2794176 + 8467200 = 39455376 \text{ тг} \quad (5.32)$$

$$3\Pi = (3000 \times 8 \times 245) \times 4 \times (1 + 0,2) = 28224000 \text{ тг} \quad (5.33)$$

$$O_{\text{сн}} = (\text{ЕАҚ} - 3A) \times 11\% = (28224000 - 2822400) \times 0,11 = 2794176 \text{ тг} \quad (5.34)$$

$$3A = \text{ЕАҚ} \times 10\% = 28224000 \times 0,1 = 2822400 \text{ тг} \quad (5.35)$$

$$\Pi_p = 3\Pi \times 0,3 = 28224000 \times 0,3 = 8467200 \text{ тг} \quad (5.36)$$

К е с т е 5.5– Ақпараттық технологияларын қолданған кездегі жылдық бір жолғы шығындар

Шығын баптары атауы	Сомасы, мың теңге
Жылдық ЕАҚ	28224000

Әлеуметтік салық аударым	2794176
Басқа да шығындар	8467200
Барлығы:	39455376

### 5.8 Салыстырмалы экономикалық тиімділіктің көрсеткіштерін есептеу.

АТ енгізудегі жылдық экономикалық әсерінің болжалды мәні келесі формуламен есептеледі:

$$\mathcal{E}_2 = \mathcal{E}_{yz} - K \times E_n, \quad (5.37)$$

$$\mathcal{E}_2 = \mathcal{E}_{yz} - K \times E_n = 19712688 - 39455376 \times 0.33 = 6692413,92 \text{ тг}$$

Нормативтік күрделі қаржы салымының экономикалық тиімділігінің коэффициенті келесі формуламен есептеледі:

$$E_n = \frac{1}{T_n} = \frac{1}{3} = 0,33$$

(5.38)

$$E_p = \frac{\mathcal{E}_{yz}}{K} = \frac{19712688}{39455376} = 0,49 \quad (5.39)$$

мұндағы  $E_p$  – есептік күрделі қаржы салымының экономикалық тиімділігі;  
Есептік күрделі қаржы салымын өтелу мерзімі:

$$T_p = \frac{1}{E_p} = \frac{1}{0,49} = 2,04 \quad (5.40)$$

Дисконтты есептемегенде өтелу мерзімі 2 жыл 1 ай.

К е с т е 5.6– Бағдарлама өнімін енгізудің салыстырмалы экономикалық тиімділігінің көрсеткіштері

Көрсеткіштер атауы	Мәні
Шартты жылдық шығынды үнемдеу, мың тенге	19712688

Күрделі қаржы салымының экономикалық тиімділігінің коэффициенті ( $E_p$ )	0,49
Күрделі қаржы салымын өтелу мерзімі ( $T_p$ ), жыл	2,04

### 5.9 Динамикалық көрсеткіштер негізінде жобаны өткізуде экономикалық тиімділігін бағалау.

Таза дисконттық табыс ( $ЧДД$ ) келесі формуламен есептеледі:

$$ЧДД = \sum_{t=0}^T (P_t - Z_t) \times \frac{1}{(1+E)^t} - K, \quad (5.41)$$

$$\Theta_t = (P_t - Z_t) = \Theta_{yr} = 19712688 \text{ мың теңге.}$$

Есеп аралығы – есептеу кезеңіндегі уақыттың бір бөлігі (жыл, квартал, ай және т. б.).

$$T_{ок} = t + \frac{K - (d\Pi_1 + d\Pi_2)}{d\Pi_3} = 2.04 + \frac{39455376 - 30141606.6}{11407805.55} = 2,856 = 2 \text{ жыл } 10 \text{ ай}$$

Егер инвестициялық жобадан таза дисконттық табыс ( $ЧДД$ ) оң болса, жоба тиімді (берілген дисконттау мөлшерінде).

$E$  – тұрақты дисконт мөлшері, 20 % .

Онда барлық есеп шегін ескергендегі таза дисконттық табыс сомасы:

$$NPV = ЧДД = \Theta_1 \frac{1}{1+E} + \Theta_2 \frac{1}{(1+E)^2} + \Theta_3 \frac{1}{(1+E)^3} - K;$$

$$ЧДД = \Theta_1 \frac{1}{1+E} + \Theta_2 \frac{1}{(1+E)^2} + \Theta_3 \frac{1}{(1+E)^3} - K = \frac{19712688}{1,2} + \frac{19712688}{1,2^2} + \frac{19712688}{1,2^3} - 39455376 = 2094036,15 \text{ тг}$$

Табыстық индексі ( $ИД$ ) келтірілген әсерлердің сомасының күрделі қаржы салымына қатынасы. Ол келесі формуламен есептеледі:

$$ИД = \frac{1}{K} \sum_{t=1}^T (P_t - Z_t) \cdot \frac{1}{(1+E)^t},$$

$$ИД = \frac{41524412,15}{39455376} = 1,03. \quad (5.42)$$

$$E_1 = 0,15.$$

$$ЧДД_1 = \frac{19712688}{1,15} + \frac{19712688}{1,15^2} + \frac{19712688}{1,15^3} - 39455376 = 5553554,48 \text{ тг}$$

$$E_2 = 0,2.$$

$$ЧДД_2 = \frac{19712688}{1,2} + \frac{19712688}{1,2^2} + \frac{19712688}{1,2^3} - 39455376 = 2069036,15 \text{ тг}$$

$$ВНД = 0,15 + \frac{5553554,48}{5553554,48 + 2069036,15} \cdot (0,2 - 0,15) = 0,186$$

Сонымен, дисконттау мөлшері келесі аралықта 15 %...18,6 %.

К е с т е 5.7– Бағдарлама өнімін әзірлеуінің және енгізуінің экономикалық пайдалылығының көрсеткіштері

Көрсеткіштер атауы	Мәні
Бағдарлама өнімін әзірлеуге және енгізуге шығын, мың теңге	39455376
Бағдарлама өнімін енгізгеннен кейінгі болжалды үнем, мың теңге	19712688
Таза дисконттық табыс, мың теңге	2069036,15
Табыстық индекс	1,03
Ішкі табыстық мөлшері	0,186
Дисконтталған өтелу мерзімі, жыл	2жыл 10ай
Моральдық ескіру мерзімі, жыл	3

### Ақшаның жылжуы (қозғалысы)

К е с т е 5.8 - Ақшаның жылжуы

Атауы	Жылдар			
	2014	2015	2016	2017
Біреттік шығын,тг	39455376			





Кесте 5.8 –нің соңы

Үнемдеуден шыққан пайда		19712688	19712688	19712688
Пайдаға салық		3942537,6	3942537,6	3942537,6
Таза пайда	-39455376	15770150,4	15770150,4	15770150,4
Ағымдығы шығын		39455376	39455376	39455376
Дисконттау еселігі (20% мөлшерлеменен)	1	0,83	0,694	0,58
Таза дисконтталған пайда	-39455376	16361531,04	13680605,47	11433359,04
Жалпы өсім бойынша ТДП	-39455376	-23093844,96	9413239,49	2020119,55

## Қорытынды

Қазіргі таңда, Қазақстандағы ақпараттық технологиялардың дамуына байланысты кез келген мекеме орындары және кәсіпорындарының құжаттамасы және құжатайналымы қағаз түрінен электрондық түрге ауыса бастады. Яғни барлығы автоматтырылған жүйеге ауысуда. ХХІ ғасыр — ғылым мен техника, ақпарат ғасыры десек, өркениеттің көшінен қалмау үшін біз білім сапасын арттырып, адам ресурсын байытуымыз қажет. Мұндай өзгерістердің құндылығы зор: басқару, көшіру және өзгерту техникаларының оңайлылығына қоса, қазіргі заманғы адамдардың еңбек ету өнімділігі мен жылдамдығы артады.

Бұл дипломдық жобада барлық қойылған мақсаттар іске асырылды. «Ақша айырбастау пунктеріндегі ақпараттар қозғалысының бақылау жүйесін құру» дерекқоры құрылды. Дерекқорды құру барысында барлық ережелер ескерілді.

Дерекқор MS SQL Server дерекқорларды басқару жүйесінде жасалды, жаңа түскен ақпаратқа байланысты деректерді тек администратор енгізе алады. Мұндай қатынау принципі пайдаланушылар үшін онша ыңғайлы емес, сондықтан пайдаланушының барлық сұраныстарын қанағаттандыратын клиенттік қосымшасы Microsoft Visual Studio 2012 ultimate программалық ортасында іске асырылады.

Visual Studio бағдарламалық ортасында клиенттік қосымша жасалды. Сонымен қатар, авторизация қолдану арқылы деректердің қауіпсіздік пен құпиялылық мәселелері қарастырылды.

Бұл бағдарламалық өнімді кез келген ақша айырбастау пункттерінде пайдалануға болады.

Тіршілік қауіпсіздігі бөлімінде өндірістік жарықтануды есептеудің нәтижесінде кафедрада 65 Вт қуатты, екі люминесцентті ЛХБ шамдары бар төрт ЛПО-02 типті шырақ қондырылды. Табиғи жарықтануды нормаға сәйкес келмейді, сондықтан күндіз қосымша жасанды жарықтануды қолдану ұсынылады.

Техника-экономикалық негізделуі бөлімінде жасалған есептеулер нәтижесінде бағдарламалық өнімнің дисконталған өтелу мерзімі 2 жыл 10 ай, ал бағдарлама өнімін әзірлеуге және енгізуге шығын 39455376 мың теңгені құрады.

Жалпы бағдарлама қолданушыға тиімді, қызметкерге айырбастау валюталарының ақпараттарын бақылау үшін оңтайлы қамтама болып табылады.

## ӘДЕБИЕТТЕР ТІЗІМІ

- 1 Соснов А. Основы проектирование информационных систем. – М.: ДМК Пресс, 2002. – 1020 с.
- 2 Марк Шпеник, Оррин Следж. Microsoft SQL Server 2000 DBA Survival Guide. – М.: Вильямс, 2001. – 236 с.
- 3 Мамаев Е. MS SQL Server. Проектирование и реализация баз данных. Сертификационный экзамен. – СПб.: BHV, 2004. – 416с.
- 4 Плю Р., Стефенс Р., Райан К. Освой самостоятельно SQL за 24 часа. – М.: Вильямс, 2000.
- 5 Глушаков С.В., Ломотько Д.В. Базы данных: учеб. курс. – М.: ООО «Издательство АСТ», 2000. – 504с.
- 6 Ибраева Л.К., Сатимова Е.Г. Проектирование баз данных. МУ к выполнению лабораторных работ. – Алматы: АИЭС, 2004.
- 7 Айтхожаева Е.Ж. Стандартный язык баз данных SQL: учеб. пособие. – Алматы: АИЭС, 2005. - 48 с.
- 8 Дунаев В.В. Базы данных. Язык SQL. – СПб.: БХВ – Петербург, 2006.
- 9 Алексей Вишнеvский. SQL Server . Эффективная работа. – Санкт-Петербург, 2009. – 541с.
- 10 Абдимуратов Ж.С., Мананбаева С.Е. Безопасность жизнедеятельности. Методические указания к выполнению раздела «Расчет производственного освещения» в выпускных работах для всех специальностей. Бакалавриат. - Алматы: АИЭС, 2009. – 20 с.
- 11 Хакимжанов Т.Е. Расчет аспирационных систем. Дипломное проектирование. Для студентов всех форм обучения всех специальностей. – Алматы: АИЭС, 2002. – 30 с.
- 12 Еркешева З.Д., Боканова Г.Ш. Методические указания к выполнению экономической части дипломной работы.5В070400-Вычислительная техника и программное обеспечение . –Алматы:АУЭС,2013 –40с.
- 13 Резникова Н.П. Маркетинг в телекоммуникациях. – М.:ЭКО-ТРЭНДЗ, 1998.
- 14 Хавронская А.М. Оценка технико-экономической эффективности программных средств. – Алматы: КазНТУ, 2000.
- 15 Линаев В.В., Потапов А.И. Оценка затрат на разработку программных средств. – М.: Финансы и статистика, 1998.

## А қосымшасы

### Экономикалық бөлім керекті мәліметтер

К е с т е 5.1 – q коэффициентінің мәні

Тапсырма түрлері	Коэффициенттің өзгеру аралығы
Есептеу тапсырмалары	1400 ден 1500
Оперативті басқару тапсырмалары	1500 ден 1700
Жоспарлау тапсырмалары	3000 ден 3500
Көп вариантты	4500 ден 5000
Комплекстік тапсырма	5000 ден 5500

К е с т е 5.2– Еңбек сыйымдылығын есептейтін коэффициент

Бағдарлама тілі	Күрделік тобы	Жаңалықтық дәрежесі				B коэффициенті
		A	Б	В	Г	
Жоғарғы деңгей	1	1,38	1,26	1,15	0,69	1,2
	2	1,30	1,19	1,08	0,65	1,35
	3	1,20	1,10	1,00	0,60	1,5
Төменгі деңгей	1	1,58	1,45	1,32	0,79	1,2
	2	1,49	1,37	1,24	0,74	1,35
	3	1,38	1,26	1,15	0,69	1,5

К е с т е 5.3– Бағдарлама жасаушы білектілігін ескеретін коэффициент

Жұмыс тәжірибиесі	Білектілік коэффициенті
Екі жылға дейін	0.8
2-3 жыл	1

*Кесте 5.3-нің соңы*

3-5 жыл	1.1 – 1.2
5-7 жыл	1.3 – 1.4
7 жылдан көп	1.5 – 1.6

**4 Бірыңғай тарифтік сеткадан көшірме (БТС)**

К е с т е 5.4– Бірыңғай тарифтік сеткадан көшірме (БТС)

Еңбек ақы разряды	1	2	3	4	5	6	7	8	9	10	11	12
Тарифтік коэффицие нт	1,0	1,07	1,15	1,24	1,33	1,43	1,54	1,66	1,78	1,91	2,05	2,25

5. <http://egov.kz>

6. <http://www.inform.kz/> сайты бойынша алматы қаласында 2014 жылығы электр энергиясының тұтыну бағасы 1 кВт үшін 14,36 теңге болса, екінші деңгейде ол сағатына 1 кВт үшін - 19,44 теңгені, үшінші деңгейде - сағатына 1 кВт үшін 24,32 теңгені құрады. Орташа есеппен 19,37 теңге болды.

## Б қосымшасы

### Бағдарлама листингі

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace ExchangeAdmin
{
    public partial class AdminControl : Form
    {
        private static int rec;
        private static string[] kas_fam, kas_name, kas_patronymic;
        public AdminControl()
        {
            InitializeComponent();
            a1 = "select k.kassir_id, k.fam as 'Тегі', k.name as 'Аты', k.patronymic as 'Әкесінің аты', k.login as 'Логин', k.password as 'Құпия сөз' from Kassiry k";
            connect();
            dataGridView1.Columns["kassir_id"].Visible = false;
            for (int i = 0; i < dataGridView1.RowCount; i++)
            {
                var p = (from ie in Program.db.Kassiries where ie.kassir_id ==
                    Convert.ToInt32(dataGridView1.Rows[i].Cells[0].Value) select ie.status).SingleOrDefault();
                if (p == 1)
                {
                    dataGridView1.Rows[i].DefaultCellStyle.BackColor = Color.Red;
                }
            }
            groupBox2.Enabled = false;

            c1 = "select k.fam as 'Клиент тегі', k.name as 'Клиент аты', k.patronymic as 'Клиент әкесінің аты',
                k.inn as 'Клиент ЖСН нөмері', v.val_name as 'Операцияға енген валюта', v.oper_name as
                'Операция түрі', v.st_sum as 'Сол уақыттағы валюта сомасы', v.st_date as 'Валюта сомасы
                енгізілген уақыт', k.sum_den as 'Клиент сомасы', k.sum_obmen as 'Клиентке берілетін сома',
                ks.fam as 'Кассир тегі', ks.name as 'Кассир аты', ks.patronymic as 'Кассир әкесінің аты',
                k.date_oper as 'Операция жасалған уақыт' from Klienty k inner join valute_oper v on v.st_id =
                k.st_id inner join Kassiry ks on ks.kassir_id = k.kassir_id";
            connect3();
            kas_fam = (from i in Program.db.Kassiries select i.fam).ToArray();
        }
    }
}
```

## Б қосымшасының жалғасы

```
kas_name = (from i in Program.db.Kassiries select i.name).ToArray();
kas_patronimic = (from i in Program.db.Kassiries select i.partronimic).ToArray();
for (int i = 0; i < kas_fam.Length; i++)
{
    comboBox1.Items.Add(kas_fam[i] + " " + kas_name[i] + " " + kas_patronimic[i]);
}
comboBox4.Items.Add("Барлық валюта");
comboBox5.Items.Add("Барлық валюта");
var queryVal = from i in Program.db.Valutes select i.val_name;
foreach (var v in queryVal)
{
    comboBox4.Items.Add(v);
    comboBox5.Items.Add(v);
    comboBox6.Items.Add(v);
    comboBox2.Items.Add(v);
}
groupBox5.Enabled = false;
dateTimePicker2.Enabled = false;
button8.Enabled = false;
textBox6.Enabled = false;
var typeO = from i in Program.db.Oper_types select i.oper_name;
foreach (var s in typeO)
{
    comboBox3.Items.Add(s);
}
panel2.Visible = false;
button11.Enabled = false;
button12.Enabled = false;
}
private void dataGridView1_RowPrePaint(object sender, DataGridViewRowPrePaintEventArgs e)
{
    int index = e.RowIndex;
    string indexStr = (index + 1).ToString();
    object header = this.dataGridView1.Rows[index].HeaderCell.Value;
    if (header == null || !header.Equals(indexStr))
        this.dataGridView1.Rows[index].HeaderCell.Value = indexStr;
}

private void dataGridView2_RowPrePaint(object sender, DataGridViewRowPrePaintEventArgs e)
{
    int index = e.RowIndex;
    string indexStr = (index + 1).ToString();
    object header = this.dataGridView2.Rows[index].HeaderCell.Value;
    if (header == null || !header.Equals(indexStr))
        this.dataGridView2.Rows[index].HeaderCell.Value = indexStr;
}
```

## *Б қосымшасының жалғасы*

```
private void dataGridView3_RowPrePaint(object sender, DataGridViewRowPrePaintEventArgs e)
{
    int index = e.RowIndex;
    string indexStr = (index + 1).ToString();
    object header = this.dataGridView3.Rows[index].HeaderCell.Value;
    if (header == null || !header.Equals(indexStr))
        this.dataGridView3.Rows[index].HeaderCell.Value = indexStr;
}
```

```
private void dataGridView4_RowPrePaint(object sender, DataGridViewRowPrePaintEventArgs e)
{
    int index = e.RowIndex;
    string indexStr = (index + 1).ToString();
    object header = this.dataGridView4.Rows[index].HeaderCell.Value;
    if (header == null || !header.Equals(indexStr))
        this.dataGridView4.Rows[index].HeaderCell.Value = indexStr;
}
```

```
string a, a1, b, b1, c, c1, d, d1, cn;
public void connect()
{
    SqlConnection cn = Program.conn;
    a = a1;
    DataSet ds = new DataSet();
    SqlDataAdapter da = new SqlDataAdapter(a, cn);
    da.Fill(ds);
    dataGridView1.DataSource = ds.Tables[0];
}
```

```
public void connect2()
{
    SqlConnection cn = Program.conn;
    b = b1;
    DataSet ds1 = new DataSet();
    SqlDataAdapter da1 = new SqlDataAdapter(b, cn);
    da1.Fill(ds1);
    dataGridView2.DataSource = ds1.Tables[0];
}
```

```
public void connect3()
{
    SqlConnection cn = Program.conn;
    c = c1;
    DataSet ds2 = new DataSet();
    SqlDataAdapter da2 = new SqlDataAdapter(c, cn);
    da2.Fill(ds2);
    dataGridView3.DataSource = ds2.Tables[0];
}
```



## *Б қосымшасының жалғасы*

```
public void connect4()
{
    SqlConnection cn = Program.conn;
    d = d1;
    DataSet ds3 = new DataSet();
    SqlDataAdapter da3 = new SqlDataAdapter(d, cn);
    da3.Fill(ds3);
    dataGridView4.DataSource = ds3.Tables[0];
}
private void button2_Click(object sender, EventArgs e)
{
    this.Close();
}
private void button1_Click(object sender, EventArgs e)
{
    groupBox1.Enabled = false;
    groupBox2.Enabled = true;
    rec = 0;
}

private void button3_Click(object sender, EventArgs e)
{
    groupBox1.Enabled = false;
    groupBox2.Enabled = true;
    rec = 1;
    textBox1.Text = dataGridView1.CurrentRow.Cells[1].Value.ToString();
    textBox2.Text = dataGridView1.CurrentRow.Cells[2].Value.ToString();
    textBox3.Text = dataGridView1.CurrentRow.Cells[3].Value.ToString();
    textBox4.Text = dataGridView1.CurrentRow.Cells[4].Value.ToString();
    textBox5.Text = dataGridView1.CurrentRow.Cells[5].Value.ToString();
}
private void button4_Click(object sender, EventArgs e)
{
    var kas_id = dataGridView1.CurrentRow.Cells[0].Value;
    SqlConnection sqlConnection1 = Program.conn;
    sqlConnection1.Open();
    SqlCommand cmd = new SqlCommand();
    cmd.CommandType = System.Data.CommandType.Text;
    cmd.CommandText = "update Kassiry set status = 1 where kassir_id = " + kas_id;
    cmd.Connection = sqlConnection1;
    cmd.ExecuteNonQuery();
    sqlConnection1.Close();
    MessageBox.Show("Пайдаланушыға жүйеге ену жабылды", "Назар аударыңыз",
    MessageBoxButtons.OK, MessageBoxIcon.Information);
    a1 = "select k.kassir_id, k.fam as 'Тегі', k.name as 'Аты', k.partronimic as 'Әкесінің аты', k.login as
    'Логин', k.password as 'Құпия сөз' from Kassiry k";
    connect();
    dataGridView1.Columns["kassir_id"].Visible = false;
    for (int i = 0; i < dataGridView1.RowCount; i++)
    {
```

## Б қосымшасының жалғасы

```
var p = (from ie in Program.db.Kassiries where ie.kassir_id ==
Convert.ToInt32(dataGridView1.Rows[i].Cells[0].Value) select ie.status).SingleOrDefault();
if (p == 1)
{
dataGridView1.Rows[i].DefaultCellStyle.BackColor = Color.Red;
}
}

}

private void button5_Click(object sender, EventArgs e)
{
if (rec == 0)
{
if ((string.IsNullOrEmpty(textBox1.Text) || string.IsNullOrEmpty(textBox2.Text) ||
string.IsNullOrEmpty(textBox3.Text) || string.IsNullOrEmpty(textBox4.Text) ||
string.IsNullOrEmpty(textBox5.Text)))
{
MessageBox.Show("Барлық жолдарды толтырыңыз!", "Назар аударыңыз",
MessageBoxButtons.OK, MessageBoxIcon.Information);
}
else
{
SqlConnection sqlConnection1 = Program.conn;
sqlConnection1.Open();
SqlCommand cmd = new SqlCommand();
cmd.CommandType = System.Data.CommandType.Text;
cmd.CommandText = "insert into Kassiry(fam, name, partronimic, login, password, status)
values('" + textBox1.Text + "','" + textBox2.Text + "','" + textBox3.Text + "','" + textBox4.Text +
 "','" + textBox5.Text + "', 0)";
cmd.Connection = sqlConnection1;
cmd.ExecuteNonQuery();
sqlConnection1.Close();
MessageBox.Show("Ақпарат сәтті енгізілді", "Назар аударыңыз", MessageBoxButtons.OK,
MessageBoxIcon.Information);
textBox1.Clear();
textBox2.Clear();
textBox3.Clear();
textBox4.Clear();
textBox5.Clear();
groupBox2.Enabled = false;
groupBox1.Enabled = true;
a1 = "select k.kassir_id, k.fam as 'Тегі', k.name as 'Аты', k.partronimic as 'Әкесінің аты', k.login as
'Логин', k.password as 'Құпия сөз' from Kassiry k";
connect();
dataGridView1.Columns["kassir_id"].Visible = false;
for (int i = 0; i < dataGridView1.RowCount; i++)
{

```

## Б қосымшасының жалғасы

```
var p = (from ie in Program.db.Kassiries where ie.kassir_id ==
Convert.ToInt32(dataGridView1.Rows[i].Cells[0].Value) select ie.status).SingleOrDefault();
if (p == 1)
{
dataGridView1.Rows[i].DefaultCellStyle.BackColor = Color.Red;
}
}

}

else if (rec == 1)
{
if ((string.IsNullOrEmpty(textBox1.Text) || string.IsNullOrEmpty(textBox2.Text) ||
string.IsNullOrEmpty(textBox3.Text) || string.IsNullOrEmpty(textBox4.Text) ||
string.IsNullOrEmpty(textBox5.Text)))
{
MessageBox.Show("Барлық жолдарды толтырыңыз!", "Назар аударыңыз",
MessageBoxButtons.OK, MessageBoxIcon.Information);
}
else
{
var kas_id = dataGridView1.CurrentRow.Cells[0].Value;
SqlConnection sqlConnection1 = Program.conn;
sqlConnection1.Open();
SqlCommand cmd = new SqlCommand();
cmd.CommandType = System.Data.CommandType.Text;
cmd.CommandText = "update Kassiry set fam = " + textBox1.Text + ", name = " + textBox2.Text
+ ", patronimic = " + textBox3.Text + ", login = " + textBox4.Text + ", password = " +
textBox5.Text + " where kassir_id = " + kas_id;
cmd.Connection = sqlConnection1;
cmd.ExecuteNonQuery();
sqlConnection1.Close();
MessageBox.Show("Ақпарат сәтті жаңартылды", "Назар аударыңыз", MessageBoxButtons.OK,
MessageBoxIcon.Information);
textBox1.Clear();
textBox2.Clear();
textBox3.Clear();
textBox4.Clear();
textBox5.Clear();
roupBox2.Enabled = false;
groupBox1.Enabled = true;
a1 = "select k.kassir_id, k.fam as 'Тегі', k.name as 'Аты', k.patronimic as 'Әкесінің аты', k.login as
'Логин', k.password as 'Құпия сөз' from Kassiry k";
connect();
dataGridView1.Columns["kassir_id"].Visible = false;
for (int i = 0; i < dataGridView1.RowCount; i++)
{
```

## Б қосымшасының жалғасы

```
var p = (from ie in Program.db.Kassiries where ie.kassir_id ==
Convert.ToInt32(dataGridView1.Rows[i].Cells[0].Value) select ie.status).SingleOrDefault();
if (p == 1)
{
dataGridView1.Rows[i].DefaultCellStyle.BackColor = Color.Red;
}
}
}
}
private void button6_Click(object sender, EventArgs e)
{
textBox1.Clear();
textBox2.Clear();
textBox3.Clear();
textBox4.Clear();
textBox5.Clear();
groupBox2.Enabled = false;
groupBox1.Enabled = true;
}
private void button13_Click(object sender, EventArgs e)
{
var kas_id = dataGridView1.CurrentRow.Cells[0].Value;
SqlConnection sqlConnection1 = Program.conn;
sqlConnection1.Open();
SqlCommand cmd = new SqlCommand();
cmd.CommandType = System.Data.CommandType.Text;
cmd.CommandText = "update Kassiry set status = 0 where kassir_id = " + kas_id;
cmd.Connection = sqlConnection1;
cmd.ExecuteNonQuery();
sqlConnection1.Close();
MessageBox.Show("Пайдаланушыға жүйеге ене алады", "Назар аударыңыз",
MessageBoxButtons.OK, MessageBoxIcon.Information);
a1 = "select k.kassir_id, k.fam as 'Тегі', k.name as 'Аты', k.partronic as 'Әкесінің аты', k.login as
'Логин', k.password as 'Күпия сөз' from Kassiry k";
connect();
dataGridView1.Columns["kassir_id"].Visible = false;
for (int i = 0; i < dataGridView1.RowCount; i++)
{
var p = (from ie in Program.db.Kassiries where ie.kassir_id ==
Convert.ToInt32(dataGridView1.Rows[i].Cells[0].Value) select ie.status).SingleOrDefault();
if (p == 1)
{
dataGridView1.Rows[i].DefaultCellStyle.BackColor = Color.Red;
}
}
}
}
```

### *Б қосымшасының жалғасы*

```
private void button7_Click(object sender, EventArgs e)
{
    c1 = "select k.fam as 'Клиент тегі', k.name as 'Клиент аты', k.patronymic as 'Клиент әкесінің аты',
    k.inn as 'Клиент ЖСН нөмері', v.val_name as 'Операцияға енген валюта', v.oper_name as
    'Операция түрі', v.st_sum as 'Сол уақыттағы валюта сомасы', v.st_date as 'Валюта сомасы
    енгізілген уақыт', k.sum_den as 'Клиент сомасы', k.sum_obmen as 'Клиентке берілетін сома',
    ks.fam as 'Кассир тегі', ks.name as 'Кассир аты', ks.patronymic as 'Кассир әкесінің аты',
    k.date_oper as 'Операция жасалған уақыт' from Klienty k inner join valute_oper v on v.st_id =
    k.st_id inner join Kassiry ks on ks.kassir_id = k.kassir_id";
    connect3();
}
private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    int index = comboBox1.FindString(comboBox1.Text);
    c1 = "select k.fam as 'Клиент тегі', k.name as 'Клиент аты', k.patronymic as 'Клиент әкесінің аты',
    k.inn as 'Клиент ЖСН нөмері', v.val_name as 'Операцияға енген валюта', v.oper_name as
    'Операция түрі', v.st_sum as 'Сол уақыттағы валюта сомасы', v.st_date as 'Валюта сомасы
    енгізілген уақыт', k.sum_den as 'Клиент сомасы', k.sum_obmen as 'Клиентке берілетін сома',
    ks.fam as 'Кассир тегі', ks.name as 'Кассир аты', ks.patronymic as 'Кассир әкесінің аты',
    k.date_oper as 'Операция жасалған уақыт' from Klienty k inner join valute_oper v on v.st_id =
    k.st_id inner join Kassiry ks on ks.kassir_id = k.kassir_id where ks.fam = '' + kas_fam[index] + ''";
    connect3();
}
private void button14_Click(object sender, EventArgs e)
{
    c1 = "select k.fam as 'Клиент тегі', k.name as 'Клиент аты', k.patronymic as 'Клиент әкесінің аты',
    k.inn as 'Клиент ЖСН нөмері', v.val_name as 'Операцияға енген валюта', v.oper_name as
    'Операция түрі', v.st_sum as 'Сол уақыттағы валюта сомасы', v.st_date as 'Валюта сомасы
    енгізілген уақыт', k.sum_den as 'Клиент сомасы', k.sum_obmen as 'Клиентке берілетін сома',
    ks.fam as 'Кассир тегі', ks.name as 'Кассир аты', ks.patronymic as 'Кассир әкесінің аты',
    k.date_oper as 'Операция жасалған уақыт' from Klienty k inner join valute_oper v on v.st_id =
    k.st_id inner join Kassiry ks on ks.kassir_id = k.kassir_id where st_date = '' +
    dateTimePicker1.Value + ''";
    connect3();
}
private void comboBox2_SelectedIndexChanged(object sender, EventArgs e)
{
    b1 = "select v.st_id, v.val_name as 'Валюта аты', v.oper_name as 'Операция түрі', v.st_sum as
    'Сомасы', v.st_date as 'Орнатылған күні' from valute_oper v where v.val_name = '' +
    comboBox2.Text + ''";
    connect2();
    dataGridView2.Columns["st_id"].Visible = false;
    groupBox5.Enabled = true;
    dateTimePicker2.Enabled = true;
    label10.Text = "Бүгінгі күнге " + comboBox2.Text + " бағасы орнатылған. Оны өзгертіңіз келе
    ма?";
}
private void button10_Click(object sender, EventArgs e)
{

```

## Б қосымшасының жалғасы

```
b1 = "select v.st_id, v.val_name as 'Валюта аты', v.oper_name as 'Операция түрі', v.st_sum as 'Сомасы', v.st_date as 'Орнатылған күні' from valute_oper v where v.val_name = '" +
comboBox2.Text + "' and st_date = '" + dateTimePicker2.Value + "'";
connect2();
dataGridView2.Columns["st_id"].Visible = false;
}

private void button8_Click(object sender, EventArgs e)
{
var val_id = (from i in Program.db.Valutes where i.val_name == comboBox2.Text select
i.valute_id).SingleOrDefault();
var oper_id = (from i in Program.db.Oper_types where i.oper_name == comboBox3.Text select
i.oper_id).SingleOrDefault();
int[] zapr = (from i in Program.db.Valute_status where i.valute_id == val_id && i.oper_id ==
oper_id && i.st_date == DateTime.Today select i.st_id).ToArray();
if (zapr.Length == 0)
{
if (string.IsNullOrEmpty(textBox6.Text))
{
MessageBox.Show(comboBox2.Text + " валюта бағасын енгізіңіз", "Назар аударыңыз",
MessageBoxButtons.OK, MessageBoxIcon.Information);
}
else
{
try
{
SqlConnection sqlConnection1 = Program.conn;
sqlConnection1.Open();
SqlCommand cmd = new SqlCommand();
cmd.CommandType = System.Data.CommandType.Text;
cmd.CommandText = "insert into Valute_status(valute_id, oper_id, st_sum, st_date) values ('" +
val_id + "','" + oper_id + "','" + textBox6.Text + "','GETDATE())";
cmd.Connection = sqlConnection1;
cmd.ExecuteNonQuery();
sqlConnection1.Close();
MessageBox.Show("Жаңа баға орнатылды", "Назар аударыңыз", MessageBoxButtons.OK,
MessageBoxIcon.Information);
b1 = "select v.st_id, v.val_name as 'Валюта аты', v.oper_name as 'Операция түрі', v.st_sum as 'Сомасы', v.st_date as 'Орнатылған күні' from valute_oper v where v.val_name = '" +
comboBox2.Text + "'";
connect2();
dataGridView2.Columns["st_id"].Visible = false;
comboBox3.Items.Clear();
var typeO = from i in Program.db.Oper_types select i.oper_name;
foreach (var s in typeO)
{
comboBox3.Items.Add(s);
}
button8.Enabled = false;
}
}
}
```

## Б қосымшасының жалғасы

```
textBox6.Enabled = false;
textBox6.Clear();
}
catch (Exception)
{
    MessageBox.Show("Деректер қоры жабық", "Назар аударыңыз", MessageBoxButtons.OK,
    MessageBoxIcon.Information);
}
}
}
else
{
    if (string.IsNullOrEmpty(textBox6.Text))
    {
        MessageBox.Show(comboBox2.Text + " валюта бағасын енгізіңіз", "Назар аударыңыз",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    else
    {
        panel2.Visible = true;
    }
}
}

private void comboBox3_SelectedIndexChanged(object sender, EventArgs e)
{
    button8.Enabled = true;
    textBox6.Enabled = true;
}
private void button15_Click(object sender, EventArgs e)
{
    panel2.Visible = false;
}
private void button9_Click(object sender, EventArgs e)
{
    try
    {
        var val_id = (from i in Program.db.Valutes where i.val_name == comboBox2.Text select
        i.valute_id).SingleOrDefault();
        var oper_id = (from i in Program.db.Oper_types where i.oper_name == comboBox3.Text select
        i.oper_id).SingleOrDefault();
        SqlConnection sqlConnection1 = Program.conn;
        sqlConnection1.Open();
        SqlCommand cmd = new SqlCommand();
        cmd.CommandType = System.Data.CommandType.Text;
        cmd.CommandText = "update Valute_status set st_sum = " + textBox6.Text + " where valute_id =
        " + val_id + " and oper_id = " + oper_id + " and st_date = " + DateTime.Today + """;
        cmd.Connection = sqlConnection1;
        cmd.ExecuteNonQuery();
    }
}
```

## Б қосымшасының жалғасы

```
sqlConnection1.Close();
MessageBox.Show("Жаңа баға орнатылды", "Назар аударыңыз", MessageBoxButtons.OK,
MessageBoxIcon.Information);
b1 = "select v.st_id, v.val_name as 'Валюта аты', v.oper_name as 'Операция түрі', v.st_sum as
'Сомасы', v.st_date as 'Орнатылған күні' from valute_oper v where v.val_name = '" +
comboBox2.Text + "'";
connect2();
dataGridView2.Columns["st_id"].Visible = false;
comboBox3.Items.Clear();
var typeO = from i in Program.db.Oper_types select i.oper_name;
foreach (var s in typeO)
{
comboBox3.Items.Add(s);
}
button8.Enabled = false;
textBox6.Enabled = false;
textBox6.Clear();
panel2.Visible = false;
}
catch (Exception)
{
MessageBox.Show("Деректер қоры жабық", "Назар аударыңыз", MessageBoxButtons.OK,
MessageBoxIcon.Information);
panel2.Visible = false;
}
}
private void comboBox4_SelectedIndexChanged(object sender, EventArgs e)
{
button11.Enabled = true;
button12.Enabled = true;
if (comboBox4.Text == "Барлық валюта")
{
chart1.Series.Clear();
var queryVal = from i in Program.db.Valutes select i.val_name;
foreach (var v in queryVal)
{
chart1.Series.Add(v);
chart1.Series[v].ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.StackedColumn;
}
chart1.ChartAreas[0].CursorX.IsUserEnabled = true;
chart1.ChartAreas[0].CursorX.IsUserSelectionEnabled = true;
chart1.ChartAreas[0].AxisX.ScaleView.Zoomable = true;
chart1.ChartAreas[0].AxisX.ScrollBar.IsPositionedInside = true;
int dolId = (from i in Program.db.Valutes where i.val_name == "Доллар" select
i.valute_id).SingleOrDefault();
int evroId = (from i in Program.db.Valutes where i.val_name == "Евро" select
i.valute_id).SingleOrDefault();
```



## Б қосымшасының жалғасы

```
int rublId = (from i in Program.db.Valutes where i.val_name == "Рубль" select
i.valute_id).SingleOrDefault();
float[] sumDol = (from i in Program.db.Valute_status where i.valute_id == dolId orderby i.st_date
select i.st_sum).ToArray();
float[] sumEvro = (from i in Program.db.Valute_status where i.valute_id == evroId orderby
i.st_date select i.st_sum).ToArray();
float[] sumRubl = (from i in Program.db.Valute_status where i.valute_id == rublId orderby i.st_date
select i.st_sum).ToArray();
DateTime[] dDol = (from i in Program.db.Valute_status where i.valute_id == dolId orderby
i.st_date select i.st_date).ToArray();
DateTime[] dEvro = (from i in Program.db.Valute_status where i.valute_id == evroId orderby
i.st_date select i.st_date).ToArray();
DateTime[] dRubl = (from i in Program.db.Valute_status where i.valute_id == rublId orderby
i.st_date select i.st_date).ToArray();
for (int i = 0; i < sumDol.Length; i++)
{
chart1.Series[0].Points.AddXY(dDol[i], sumDol[i]);
chart1.Series[1].Points.AddXY(dEvro[i], sumEvro[i]);
chart1.Series[2].Points.AddXY(dRubl[i], sumRubl[i]);
}
}
else if (comboBox4.Text == "Доллар")
{
chart1.Series.Clear();
chart1.Series.Add("Доллар");
chart1.Series[0].Points.Clear();
chart1.Series[0].ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.StackedColumn;
int dolId = (from i in Program.db.Valutes where i.val_name == "Доллар" select
i.valute_id).SingleOrDefault();
float[] sumDol = (from i in Program.db.Valute_status where i.valute_id == dolId orderby i.st_date
select i.st_sum).ToArray();
DateTime[] dDol = (from i in Program.db.Valute_status where i.valute_id == dolId orderby
i.st_date select i.st_date).ToArray();
for (int i = 0; i < sumDol.Length; i++)
{
chart1.Series[0].Points.AddXY(dDol[i], sumDol[i]);
}

chart1.ChartAreas[0].CursorX.IsUserEnabled = true;
chart1.ChartAreas[0].CursorX.IsUserSelectionEnabled = true;
chart1.ChartAreas[0].AxisX.ScaleView.Zoomable = true;
chart1.ChartAreas[0].AxisX.ScrollBar.IsPositionedInside = true;
}
else if (comboBox4.Text == "Евро")
{
chart1.Series.Clear();
chart1.Series.Add("Евро");
chart1.Series[0].Points.Clear();
```

## Б қосымшасының жалғасы

```
chart1.Series[0].ChartType
Windows.Forms.DataVisualization.Charting.SeriesChartType.StackedColumn;
int evroId = (from i in Program.db.Valutes where i.val_name == "Евро" select
i.valute_id).SingleOrDefault();
float[] sumEvro = (from i in Program.db.Valute_status where i.valute_id == evroId orderby
i.st_date select i.st_sum).ToArray();
DateTime[] dEvro = (from i in Program.db.Valute_status where i.valute_id == evroId orderby
i.st_date select i.st_date).ToArray();
for (int i = 0; i < sumEvro.Length; i++)
{
chart1.Series[0].Points.AddXY(dEvro[i], sumEvro[i]);
}
chart1.ChartAreas[0].CursorX.IsUserEnabled = true;
chart1.ChartAreas[0].CursorX.IsUserSelectionEnabled = true;
chart1.ChartAreas[0].AxisX.ScaleView.Zoomable = true;
chart1.ChartAreas[0].AxisX.ScrollBar.IsPositionedInside = true;
}
else if (comboBox4.Text == "Рубль")
{
chart1.Series.Clear();
chart1.Series.Add("Рубль");
chart1.Series[0].Points.Clear();
chart1.Series[0].ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.StackedColumn;
int rublId = (from i in Program.db.Valutes where i.val_name == "Рубль" select
i.valute_id).SingleOrDefault();
float[] sumRubl = (from i in Program.db.Valute_status where i.valute_id == rublId orderby i.st_date
select i.st_sum).ToArray(); ;
DateTime[] dRubl = (from i in Program.db.Valute_status where i.valute_id == rublId orderby
i.st_date select i.st_date).ToArray();
for (int i = 0; i < sumRubl.Length; i++)
{
chart1.Series[0].Points.AddXY(dRubl[i], sumRubl[i]);
}
chart1.ChartAreas[0].CursorX.IsUserEnabled = true;
chart1.ChartAreas[0].CursorX.IsUserSelectionEnabled = true;
chart1.ChartAreas[0].AxisX.ScaleView.Zoomable = true;
chart1.ChartAreas[0].AxisX.ScrollBar.IsPositionedInside = true;
}
}
private void button11_Click(object sender, EventArgs e)
{
label12.Text = "Айлық";
if (comboBox4.Text == "Барлық валюта")
{
chart1.Series.Clear();
var queryVal = from i in Program.db.Valutes select i.val_name;
foreach (var v in queryVal)
{
```

## Б қосымшасының жалғасы

```
chart1.Series.Add(v);
chart1.Series[v].ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.StackedColumn;
}
chart1.ChartAreas[0].CursorX.IsUserEnabled = true;
chart1.ChartAreas[0].CursorX.IsUserSelectionEnabled = true;
chart1.ChartAreas[0].AxisX.ScaleView.Zoomable = true;
chart1.ChartAreas[0].AxisX.ScrollBar.IsPositionedInside = true;
int dolId = (from i in Program.db.Valutes where i.val_name == "Доллар" select
i.valute_id).SingleOrDefault();
int evroId = (from i in Program.db.Valutes where i.val_name == "Евро" select
i.valute_id).SingleOrDefault();
int rublId = (from i in Program.db.Valutes where i.val_name == "Рубль" select
i.valute_id).SingleOrDefault();
SqlConnection sqlConnection1 = Program.conn;
sqlConnection1.Open();
for (int i = 0; i < DateTime.Now.Month; i++)
{
int mm = i + 1;
DateTime date2 = new DateTime(DateTime.Now.Year, mm, 1);
SqlCommand cmd = new SqlCommand();
cmd.CommandType = System.Data.CommandType.Text;
cmd.CommandText = "SELECT SUM(st_sum)/COUNT(st_sum) FROM Valute_status WHERE
valute_id = " + dolId + " and DATEPART ( MM, st_date ) = " + mm + " AND DATEPART (
YYYY , st_date ) = " + DateTime.Now.Year;
cmd.Connection = sqlConnection1;
SqlCommand cmd2 = new SqlCommand();
cmd2.CommandType = System.Data.CommandType.Text;
cmd2.CommandText = "SELECT SUM(st_sum)/COUNT(st_sum) FROM Valute_status WHERE
valute_id = " + evroId + " and DATEPART ( MM, st_date ) = " + mm + " AND DATEPART (
YYYY , st_date ) = " + DateTime.Now.Year;
cmd2.Connection = sqlConnection1;
SqlCommand cmd3 = new SqlCommand();
cmd3.CommandType = System.Data.CommandType.Text;
cmd3.CommandText = "SELECT SUM(st_sum)/COUNT(st_sum) FROM Valute_status WHERE
valute_id = " + rublId + " and DATEPART ( MM, st_date ) = " + mm + " AND DATEPART (
YYYY , st_date ) = " + DateTime.Now.Year;
cmd3.Connection = sqlConnection1;
var dollar = cmd.ExecuteScalar();
var evro = cmd2.ExecuteScalar();
var rubl = cmd3.ExecuteScalar();
chart1.Series[0].Points.AddXY(date2, dollar);
chart1.Series[1].Points.AddXY(date2, evro);
chart1.Series[2].Points.AddXY(date2, rubl);
}
sqlConnection1.Close();
}
else if (comboBox4.Text == "Доллар")
{
```

## Б қосымшасының жалғасы

```
chart1.Series.Clear();
chart1.Series.Add("Доллар");
chart1.Series[0].Points.Clear();
chart1.Series[0].ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.StackedColumn;
int dolId = (from i in Program.db.Valutes where i.val_name == "Доллар" select
i.valute_id).SingleOrDefault();
SqlConnection sqlConnection1 = Program.conn;
sqlConnection1.Open();
chart1.ChartAreas[0].CursorX.IsUserEnabled = true;
chart1.ChartAreas[0].CursorX.IsUserSelectionEnabled = true;
chart1.ChartAreas[0].AxisX.ScaleView.Zoomable = true;
chart1.ChartAreas[0].AxisX.ScrollBar.IsPositionedInside = true;
for (int i = 0; i < DateTime.Now.Month; i++)
{
int mm = i + 1;
DateTime date2 = new DateTime(DateTime.Now.Year, mm, 1);
SqlCommand cmd = new SqlCommand();
cmd.CommandType = System.Data.CommandType.Text;
cmd.CommandText = "SELECT SUM(st_sum)/COUNT(st_sum) FROM Valute_status WHERE
valute_id = " + dolId + " and DATEPART ( MM, st_date ) = " + mm + " AND DATEPART (
YYYY , st_date ) = " + DateTime.Now.Year;
cmd.Connection = sqlConnection1;
var dollar = cmd.ExecuteScalar();
chart1.Series[0].Points.AddXY(date2, dollar);
}

sqlConnection1.Close();
}
else if (comboBox4.Text == "Евро")
{
chart1.Series.Clear();
chart1.Series.Add("Евро");
chart1.Series[0].Points.Clear();
chart1.Series[0].ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.StackedColumn;
chart1.ChartAreas[0].CursorX.IsUserEnabled = true;
chart1.ChartAreas[0].CursorX.IsUserSelectionEnabled = true;
chart1.ChartAreas[0].AxisX.ScaleView.Zoomable = true;
chart1.ChartAreas[0].AxisX.ScrollBar.IsPositionedInside = true;
chart1.Series[0].ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.StackedColumn;
int evroId = (from i in Program.db.Valutes where i.val_name == "Евро" select
i.valute_id).SingleOrDefault();
SqlConnection sqlConnection1 = Program.conn;
sqlConnection1.Open();

for (int i = 0; i < DateTime.Now.Month; i++)
{
```

```
int mm = i + 1;
DateTime date2 = new DateTime(DateTime.Now.Year, mm, 1);
SqlCommand cmd2 = new SqlCommand();
cmd2.CommandType = System.Data.CommandType.Text;
cmd2.CommandText = "SELECT SUM(st_sum)/COUNT(st_sum) FROM Valute_status WHERE
valute_id = " + evroId + " and DATEPART ( MM, st_date ) = " + mm + " AND DATEPART (
YYYY , st_date ) = " + DateTime.Now.Year;
cmd2.Connection = sqlConnection1;
var evro = cmd2.ExecuteScalar();
chart1.Series[0].Points.AddXY(date2, evro);
}
sqlConnection1.Close();
}
else if (comboBox4.Text == "Рубль")
{
chart1.Series.Clear();
chart1.Series.Add("Рубль");
chart1.Series[0].Points.Clear();
chart1.Series[0].ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.StackedColumn;
chart1.ChartAreas[0].CursorX.IsUserEnabled = true;
chart1.ChartAreas[0].CursorX.IsUserSelectionEnabled = true;
chart1.ChartAreas[0].AxisX.ScaleView.Zoomable = true;
chart1.ChartAreas[0].AxisX.ScrollBar.IsPositionedInside = true;
int rublId = (from i in Program.db.Valutes where i.val_name == "Рубль" select
i.valute_id).SingleOrDefault();
SqlConnection sqlConnection1 = Program.conn;
sqlConnection1.Open();
for (int i = 0; i < DateTime.Now.Month; i++)
{
int mm = i + 1;
DateTime date2 = new DateTime(DateTime.Now.Year, mm, 1);
SqlCommand cmd3 = new SqlCommand();
cmd3.CommandType = System.Data.CommandType.Text;
cmd3.CommandText = "SELECT SUM(st_sum)/COUNT(st_sum) FROM Valute_status WHERE
valute_id = " + rublId + " and DATEPART ( MM, st_date ) = " + mm + " AND DATEPART (
YYYY , st_date ) = " + DateTime.Now.Year;
cmd3.Connection = sqlConnection1;
var rubl = cmd3.ExecuteScalar();
chart1.Series[0].Points.AddXY(date2, rubl);
}
sqlConnection1.Close();
}
}
```