

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Коммерциялық емес акционерлік қоғамы  
АЛМАТЫ ЭНЕРГЕТИКА ЖӘНЕ БАЙЛАНЫС УНИВЕРСИТЕТІ

Инфокоммуникациялық технологиялар кафедрасы

«Қорғауға жіберілді»

Кафедра меңгерушісі

Т.Ғ.К., доцент Чежимбаева К.С.  
(аты-жөні, ғылыми дәрежесі, атағы)

«\_\_» \_\_ 20\_\_ ж.  
(колы)

ДИПЛОМДЫҚ ЖОБА

Тақырыбы: NoSQL технологиясы қолданумен деректерді  
өңдеуді талдау

5B071900–Радиотехника, электроника және телекоммуникациялар мамандығы бойынша

Орындаған Оспан Алтынбай Абдулманнанов  
(аты - жөні) (тобы)

Жетекші аға оқытушы Ахметов М  
(аты-жөні, ғылыми дәрежесі, атағы, колы)

Кенесшілер:

Экономикалық бөлім бойынша:

А.З.Қ. профессор М.Ж.Жолдасов  
(ғылыми дәрежесі, атағы, аты-жөні)  
Ахметов М « 03 » 06 2016 ж.  
(колы)

Өмір тіршілігі қауіпсіздігі бойынша:

Б.З.К. аға оқытушы Мустафин М.З.  
(ғылыми дәрежесі, атағы, аты-жөні)  
Мустафин М.З. « 17 » 05 2016 ж.  
(колы)

Есептеу техникасын қолдану бойынша:

А.З.К. аға оқытушы Ахметов М  
(ғылыми дәрежесі, атағы, аты-жөні)  
Ахметов М « 10 » 06 2016 ж.  
(колы)

Мөлшер бақылаушы:

аға оқытушы Ахметов М  
(ғылыми дәрежесі, атағы, аты-жөні)  
Ахметов М « 10 » 06 2016 ж.  
(колы)

Пікір жазушы:

Алматы Энергетика және байланыс университеті  
(ғылыми дәрежесі, атағы, аты-жөні)  
меңгеруші Чежимбаева К.С. « 2 » 06 2016 ж.  
(колы)

Алматы 2016

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Коммерциялық емес акционерлік қоғамы  
АЛМАТЫ ЭНЕРГЕТИКА ЖӘНЕ БАЙЛАНЫС УНИВЕРСИТЕТІ

Радиотехника және байланыс факультеті  
Радиотехника, электроника және телекоммуникациялар мамандығы  
Инфокоммуникациялық технологиялар кафедрасы

жобаны орындауға берілген

ТАПСЫРМА

Студент Асан Айтнай Абдимаханұлы  
(аты - жөні)

Жоба тақырыбы NoSQL технологиясына қатысты деректерді  
құрастыру талдау

ректордың «19» қаңтар 2015ж № 148 бұйрығы бойынша бекітілген.

Аяқталған жұмысты тапсыру мерзімі: «    » 20 16 ж.

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

«Достанов Сатыр» мекемесінің қызметінде атқарыла-  
тын жұмыстар туралы мақалалар мен  
NoSQL деректер қорына қатысты жұмыс құруға  
арналған құрылғылар зерттеу

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

- Кіріспе
1. Деректер қоры құрудың негізгі принциптері
    - 1.1. Қоспақтың автоматтандырылған ақпарат жүйесі
    - 1.2. Қоспақтың қызметкердің автоматтандырылған жұмыс орындары
  2. Деректер қоры құру құрылғылары
  3. Деректер қорына қатысты жұмыс құру
  4. Ақпараттық қорының құрылымы
  5. Бизнес-жоспар
- Қорытынды



Сызба материалдарының (міндетті түрде дайындалатын сызуларды көрсету) тізімі:

Дайын қосымшаның деректер қорымен жұмысқа  
арналған құралдардың жалпы құрамы  
Локальдер деректер қорымен жұмыс кезіндегі  
бір қайырымдылық архитектурасы  
Жүйелі-сервер архитектурасы  
Рунет DGT PC-ге мәліметтерді сақтау архи-  
тектурасы  
DGT диаграммасы  
Жүйелі администратор арқылы кіру терезесі  
Бағдарламаның негізгі терезесі  
Жүйелі қолданушы ретінде кіру  
Барлық қолданушылар жалпы мәлімет  
Жүйелі бөлімдеріне қолданушылар жалпы  
мәлімет

#### Негізгі ұсынылатын әдебиеттер

1. А. Кригел, Б. Трухманов. SQL. 2-е издание. Фидель  
пользователи. 2. Microsoft SQL Server. 3. Рязанская С.Ю.  
База данных. Издательство SQL. 4. Душев В.В. База данных.  
Издательство SQL. 5. Рязанская Н.Т. Маркетинг в телеком-  
муникациях. 6. Рязанская Е.А., Мизурьская Т.М.  
Исследования связи. - М.: Радио и связь, 1999.  
7. Душев В.В., Петер Мюллер. Глава 7. Самоуни-  
тель - Санкт-Петербург, 2013.

#### Жоба бойынша бөлімшелерге қатысты белгіленген кеңесшілер

бөлімшелер	кеңесші	мерзімі	колы
Жоғары	Жоғары	14.4 - 03.06.16	Жоғары
Ақпараттық қоры	Ақпараттық	12.04-22.05.16	Ақпараттық
Техн. бөлім	Техн. бөлім	20.02 - 31.05.16	Техн. бөлім
Қосымша бөлім	Қосымша бөлім	15.06.16	Қосымша бөлім
Мәлімет бөлімдері	Мәлімет бөлімдері	10.06.16	Мәлімет бөлімдері

КЕСТЕСІ

№ p/c	Тарау аттары, әзірленетін сұрақтардың тізімі	Жетекшіге ұсыну мерзімдері	Ескерту
	<u>Кіріспе</u>	15.12.2015	орындалды
	1. Деректер қоры құрудың негізгі принциптері	5.01.2016	орындалды
	1.1 Жәкіптің автоматтандырылған ақпараттық жүйесі	20.01.2016	орындалды
	1.2 Бақылаушы қызметкердің автоматтандырылған жүйесі	1.02.2016	орындалды
	2 Деректер қоры құру құралдары	15.02.2016	орындалды
	2.1 SQL тілі деректер қорының стандартты тілі ретінде	25.02.2016	орындалды
	2.2 SQL тілі	2.03.2016	орындалды
	3. Деректерді тоқалау және оны іске асыру	10.03.2016	орындалды
	3.1 Бағалау объектісі және тағайындау	15.03.2016	орындалды
	3.2 Деректерді шешетін мәселелер	20.03.2016	орындалды
	4 Ақпараттық қауіпсіздігі	27.04.2016	орындалды
	5 Бизнес - жоспар	05.06.2016	орындалды
	Қорытынды	16.05.2016	орындалды
	Әдебиеттер тізімімен жұмыс істеу	16.05.2016	орындалды

Тапсырманың берілген уақыты «10» желтоқсан 2015 ж.

Кафедра меңгерушісі \_\_\_\_\_ т.ғ.к., доцент Чежимбаева К.С.  
(қолы) (аты-жөні, ғылыми дәрежесі, атағы)

Жоба жетекшісі \_\_\_\_\_ аға оқытушы Ахметов Т.  
(қолы) (аты-жөні, ғылыми дәрежесі, атағы)

Орындалатын тапсырманы қабылдаған студент \_\_\_\_\_ Оспан А.А.  
(қолы) (аты - жөні)



## **Аннотация**

В данном дипломном проекте анализируется система управления индивидуального предпринимателя. В этой программе создается автоматизированная информационная система предприятия, которая состоит из взаимосвязанных функциональных подсистем, обеспечивающих управленческий аппарат необходимой информацией.

База данных информационной системы разрабатывалась в СУБД NoSQL и клиентское приложение на приложении ASP.NET программного средства Visual Studio 2010.

В разделе «Безопасность жизнедеятельности» сделаны расчеты освещения в помещении естественным и искусственным методом. В разделе «Технико-экономического обоснования разработки» произведен расчет полученной экономической выгоды от внедряемой системы.

## **Abstract**

In this degree project the control system of the individual entrepreneur is analyzed. In this program the automated information system the enterprise which consists of the interconnected functional subsystems providing administrative personnel with necessary information is created.

The database of information system was developed in NoSQL DBMS and the client application on the ASP.NET appendix of a software of Visual Studio 2010.

In the section "Health and safety" lighting calculations indoors to natural and artificial methods are made. In the section "Feasibility study on development" calculation of the received economic benefit from introduced system is made.

## **Аңдатпа**

Дипломдық жобада жеке кәсіпкерлікті басқару жүйесі талданған. Бұл бағдарламада кәсіпорындағы басқару аппаратын қажетті ақпаратпен қамтамасыз ететін өзара байланысқан функционалды жүйелерден тұратын автоматтандырылған ақпараттық жүйе құрылады.

Ақпараттық жүйенің дерек қоры клиент-сервер технологиясына негізделіп, NoSQL ДҚБЖ-де және клиенттік қосымшасы Visual Studio 2010 бағдарламалық құралының ASP.NET қосымшасында жасалған.

«Тіршілік қауіпсіздігі» бөлімінде жұмыс орнындағы жарықтандырудың табиғи және жасанды әдістері бойынша есептеулер жүргізілді. Экономикалық бөлімде енгізілген жүйеге кеткен шығынды және одан алынған экономикалық пайданы есептеу жүргізілген.

## Мазмұны

Кіріспе.....	8
1 Деректер қорын құрудың негізгі принциптері.....	9
1.1 Кәсіпорынның автоматтандырылған ақпараттық жүйесі.....	9
1.2 Басқарушы қызметкердің автоматтандырылған жұмыс орындары	10
1.3 Автоматтандырылған жұмыс орнының тілдік құралдары.....	12
1.4 Автоматтандырылған жұмыс орнының жіктелуі.....	14
1.5 Деректер қорын ұйымдастырудың қанағаттандыратын талаптары.....	16
1.6 Көпжақты байланыс орнату.....	17
1.7 Өнімділігі.....	17
1.8 Минималды шығындары.....	17
1.9 Минималды артықтығы.....	18
1.10 Іздеу мүмкіндігі.....	18
1.11 Бүтіндігі.....	18
1.12 Қауіпсіздігі және құпиялығы.....	18
1.13 Өткенмен байланыс.....	18
1.14 Болашақпен байланыс.....	19
1.15 Пайдалану қарапайымдылығы.....	19
2 Деректер қорын құру құралдары.....	19
2.1 SQL тілі деректер қорының стандартты тілі ретінде.....	19
2.2 SQL тілі.....	20
2.3 SQL-дің артықшылықтары.....	21
2.4 SQL операторлары.....	24
2.5 Деректер қорының архитектурасы.....	26
2.6 Локальды деректер қоры және «файл-сервер» архитектурасы.....	27

2.7 Жойылған деректер қоры және «клиент-сервер» архитектурасы.....	28
2.8 NoSQL ДҚБЖ.....	32
2.9 NoSQL жүйелері.....	32
2.10 Биллинг принциптері.....	39
3 Дерекқорды жобалау және оны іске асыру.....	41
3.1 Қолдану облысы және тағайындалуы.....	41
3.2 Дерекқордағы шешілетін мәселелер.....	42
3.3 Концептуалды жобалау.....	42
3.4 ER моделін құру.....	43
3.5 Логикалық жобалау.....	44
3.6 Физикалық жобалау.....	45
3.7 Бағдарламаны сипаттау.....	45
4 Өмір тіршілік қауіпсіздігі және еңбек қорғау.....	66
4.1 Өндірістік ғимараттағы еңбек шартын талдау.....	66
4.2 Есептеу техникасы бөлмесіндегі өндірістік ортаның микроклиматы..	67
4.3 Жұмыс орнының жарықтандыру жүйесі.....	67
4.4 Электр тоғының адам ағзасына әсері және тоқпен зақымданудан қорғану шаралары.....	68
4.5 Жұмыс орнына шудың әсері.....	68
4.6 Табиғи жарықтандыру әдісімен есептеу.....	70
4.7 Пайдалану коэффициенті әдісімен жасанды жарықтануды есептеу....	72
4.8 Жасанды жарықтануды нүктелік әдіспен есептеу.....	74
4.9 «Тіршілік қауіпсіздігі» бөлімі бойынша қорытынды.....	75
5 Бизнес-жоспар.....	76
5.1 Жоба сипаттамасы.....	76

5.2 Маркетингтік стратегия.....	77
5.3 Бағдарламамен қамтамасыз етудегі еңбек сыйымдылығын есептеу....	77
5.4 Бағдарламалық қамсыздандыру шығынының есебі.....	79
5.5 Бағдарлама өнімін сатып алуға кеткен бір жолғы шығындар есебі.....	84
5.6 Игеру саласындағы жылдық бір жолғы шығындар есебі.....	85
5.7 Ақпараттық жүйе енгізуден үнем мен табыс мөлшерінің есебі.....	86
5.8 Салыстырмалы экономикалық тиімділіктің көрсеткіштерін есептеу...	87
5.9 Динамикалық көрсеткіштер негізінде жобаны өткізуде экономикалық тиімділігін бағалау.....	88
Қорытынды.....	92
Әдебиеттер тізімі.....	93
А қосымшасы Экономикалық бөлім.....	94
Б қосымшасы ДЖ электронды нұсқасы және бейнематериал(CD-R).....	
В қосымшасы Үлестірмелі материалдар (А4 пішімі – 13 бет).....	
Г қосымшасы Антиплагиат туралы анықтама.....	



## **Кіріспе**

Қазіргі уақытта жаңа ақпараттық технологияларға негізделген тиімділігі жоғары басқару жүйесі болмаса өндірістің дамуы мүмкін емес. Үнемі өзгеріп тұратын нарық талаптары, ғылыми-техникалық, технологиялық және маркетингтік сипаттамалары, үлкен ақпарат ағындары кәсіпорынның жоғары дәрежелі технологиялармен даму әдісіне және стратегиясына жауапты кәсіпорын қызметшісінен аз шығын шығара отырып, көп табыс табуға бағытталған шешімді тез және нақты қабылдауын талап етеді. Шығындарды оңтайландыруды, тұтынушылардың барлық талаптарына сәйкес өндірістің реактивтілігін жоғарылатуды қатал нарықтық бәсекелестік шартында ойша жасалған қорытындыларға негіздеп жасауға болмайды. Кәсіпорындағы шығын орталықтарының барлығына жан-жақты бақылау жасау, талдаудың күрделі математикалық әдістері пайдалану, болжам жасау, параметрлер мен белгілердің есебіне және жинақтардың үйлесімді жүйесіне негізделген жоспарлау, қор жинау және ақпараттарды өңдеу қажет. Бұл мәселенің басқару аппаратының шамадан тыс артуымен байланысты экстенсивті шешу жолдары, тіпті оның жұмысы өте жақсы ұйымдастырылса да оң нәтиже бере алмайды. Және мұнда тек бір ғана жол бар – басқарудың компьютерлік ақпараттық жүйесін құру және барлық жерге орнату.

Берілген дипломдық жобаның негізгі мақсаты кәсіпорындарда NoSQL технологиясы көмегімен деректер қорын құра отырып, соларды талдау болып табылады.

Қойылған мақсатқа жету үшін бұл жұмыста келесідей тапсырмалар қойылады:

- ақпараттық жүйелерді жобалаудың жалпы сұрақтарын зерттеу;
- кәсіпорын шешетін типтік тапсырмалардың талдауын орындау;
- кәсіпорынды басқару жүйесінің жобасын өңдеу.

Ақпараттың көп мөлшерін сақтау және өңдеу, жинақтаудың тез және тиімді жүзеге асуы қазіргі заманғы ұйымдардың, мекемелердің және кәсіпорындардың табысты қызмет көрсетуінің басты шарты болды. Қоғамның ақпараттануының өсу қарқыны басқару процесіндегі есептеу техникасының мәнін арттырады. Ақпаратты өңдеу процесін автоматтандыру үшін қазіргі есептеу техникаларының мүмкіндіктерін пайдалану, жұмыс өнімділігін көбейтуге, құжаттармен жұмыс істеудің тиімділігін көтеруге және басқаруға арналған ақпараттармен алмасуды тездетуге мүмкіндік береді.

Соңғы біршама жылда ақпараттар қорымен басқару жүйесінің тұтынушылық сапасының деңгейі, яғни функциялардың әртүрлілігі, пайдаланушыға ыңғайлы интерфейс, бағдарламалық өніммен, кей жағдайларда басқа ДҚБЖ-мен біріктірілуі желіде жұмыс жасауға арналған мүмкіндіктер артты. ДҚБЖ әр түрлі ақпараттарды (электрондық кестелер, басқа дерек қорлары) біріктіруге мүмкіндік береді және қажетті ақпаратты жылдам тауып, оны айналадағыларға есептеулер, сызбалар және кестелер

көмегімен жеткізуге көмектеседі.

# **1 Деректер қорын құрудың негізгі принциптері**

## **1.1 Кәсіпорынның автоматтандырылған ақпараттық жүйесі**

Кәсіпорынның автоматтандырылған ақпараттық жүйесінің материалды-техникалық базасын көрсететін есептеу техникасының және байланыс құралдарының негізінде кәсіпорынды басқарудың формалары мен әдістерінің жетілдірілуі прогрестің маңызды факторы болып табылады. Ол басқарудың объектілері мен субъектілері арасындағы байланыстырушы бөлімше қызметін атқарады және келесі функцияларды орындайды: пайдаланушы енгізген негізгі мәліметтер мен сұраныстарды қабылдайды, жүйеге енгізілген және сақталған мәліметтерді анықталған алгоритмдерге сәйкес өңдейді және қажетті шығатын ақпаратты құрады.

Кәсіпорында басқару аппаратын қажетті ақпаратпен қамтамасыз ететін өзара байланысқан функционалды жүйелерден тұратын автоматтандырылған ақпараттық жүйе құрылады. Негізгі функционалды жүйелер өндірістің техникалық дайындығының, перспективті жоспарлауының және өндіріс дамуының болжамының, маркетингтік зерттеудің, материалды, еңбектік және қаржылық ресурстарды оперативті басқарудың, дайын өнімнің сатылуының және іске асуының, кәсіпорынның бухгалтерлік есебінің және шаруашылық қызметтер талдауының шешімін қамтамасыз етеді. Шешілетін тапсырмалар жиынтығы бойынша кәсіпорынның автоматтандырылған ақпараттық жүйесі 80-ші жылдардағы кәсіпорынды басқарудың автоматты жүйесіне ұқсайды, бірақ басқа сапалы идеологиялық және техникалық базада. Автоматтандырылған ақпараттық базада әрбірі өңделетін мәліметтің аса спецификалық құрылымдық ерекшеліктеріне ие жүйелер немесе «бизнес-процестер» қарастырылады. Бизнес-процесс негізінде кәсіпорынның өзара әрекет ететін бөлімдері арқылы өтетін және тұтынушы тапсырысын (тауарды дайындау немесе қызмет көрсету) орындауға бағытталған материалды ақпараттық, қаржылық ағындардың немесе жұмыстық ағындардың өзара байланысқан жиынтығы түсіндіріледі. Автоматтандырылған ақпараттық жүйе бизнес-процеске өзгеріс енгізу жөнінде уақытында шешім қабылдау мақсатында экономикалық объектілердің жағдайын тікелей көрсетуге, бизнес-процестердің тиімділігін болжауға және олардың ұйымдастырылуы бойынша сипаттама беруге міндетті.

Қазіргі заманғы автоматтандырылған ақпараттық жүйенің сипаттамалық ерекшелігі – кәсіпорынның өзара байланысқан мәліметтер қорын және бірыңғай ақпараттық жүйенің білімін пайдалану.

Кәсіпорынның автоматтандырылған ақпараттық жүйесі техникалық жағынан байланысқан локальды есептеу желілерінің (ЛЕЖ) жиынтығын көрсетеді. Ірі корпорацияларға, бірлестіктерге, холдингтерге арнап мыңдаған ЭЕМ-ді санайтын және күрделі құрылымы бар үлкен корпоративті желілер құрылады. Корпоративті желілер локальды есептеу желілерін және глобальді есептеу желілерін (ГЕЖ) қосады. Соңғылары әр түрлі қалаларда, елдерде және

тіпті әр түрлі континеттерде болуы мүмкін территориялық бөлінген ЭЕМ-дерді біріктіреді. Масштабына байланысты бөлімдердің желілері мен жұмыстық топтардың желілері ажыратылады.

Жұмыстық топтардың желілері әдетте 10-нан 20-ға дейінгі ПЭЕМ қосады. Мұндай желілердің қарапайымдылық және біртектілік қасиеттері бар. Бөлімшелердің желісі 100-150 ПЭЕМ-ді қамтиды. ПЭЕМ басқарушы қызметкердің жұмыс орнына орнатылады және олардың қазметтерін автоматтандыру үшін, кәсіпорынның кеңейтілген ақпараттық базасына оперативті рұқсатты қамтамасыз ету үшін пайдаланылады. Мұндай жұмыс орны автоматтандырылған деп аталады және кәсіпорынның автоматтандырылған ақпараттық басқаруының элементі болып табылады.

## **1.2 Басқарушы қызметкердің автоматтандырылған жұмыс орындары**

Автоматтандырылған жұмыс орны (АЖО) ретінде есептеу желілерінің құрамындағы секілді тәуелсіз жұмыс жасайтын, соңғы пайдаланушының жұмыс орнында орнатылған ЭЕМ негізіндегі оның кәсіби қызметінің тапсырмасын шешудің техникалық және ұйымдастырушылық-әдістемелік қамтамасыз етілуін қосқанда, соңғы пайдаланушының аспаптық құралдарының жиынтығы түсіндіріледі. Орындалатын функцияларына байланысты автоматтандырылған жұмыс орнының үш түрі қарастырылады:

- басқарушының АЖО;
- маманның АЖО;
- техникалық және қосалқы қызметкердің АЖО.

Басқарушының автоматтандырылған жұмыс орнына келесі талаптар ұсынылады:

- үнемі оперативті және ең анық ақпараттармен толтырылып отыратын бөлінген мәліметтер қоры мен ақпараттың болуы;
- жеке мәліметтер қорына және ақпаратқа немесе олардың фрагменттеріне тек шектелген ортаның тұлғаларының, ал жеке элементтерге тек басқарушының ғана рұқсаты болуы мүмкін;
- ақпарат басқарушының психологиялық сипаттамасына бағытталған формадағы көрнекілігі;
- ақпаратты оперативті іздеуді қамтамасыз ету;
- басқарушылық шешім қабылдаудың қамтамасыз бағдарламалық құралдарының болуы;
- жұмыстың қарапайымдылығы;
- басқарушылық шешім негізінде тәжірибе жинау мүмкіндігін қамтамасыз ету;
- кәсіпорынның немесе оның бөлімшелерінің ұйымдастырушылық құрылымы шеңберінде формацияның басқа көздерімен оперативті байланысын қамтамасыз ету.

Бұл типтің автоматтандырылған жұмыс орындарында ақпаратты енгізудің тілдік және сезімталдық мүмкіндіктері жиі пайдаланылады.

Маманның (жоспарлаушының, қаржыгердің, нормалаушының, технологтың, маркетингтің, есепшінің және т.б.) автоматтандырылған



жұмыс орны локальды мәліметтер қоры мен ақпараттар сияқты мәліметтер қоры мен қолданбалы бағдарламалық қамтаманың бөлінуі негізіндегі кәсіби қызмет тапсырмаларын шешуді қамтамасыз етеді.

Техникалық қызметкердің автоматтандырылған жұмыс орны мына функцияларды жүзеге асырады:

- ақпарат енгізу;
- мәліметтер қорын басқару;
- кіріс және шығыс құжаттарын өңдеу;
- атқарушылық қызметтерді басқару.

Жеке автоматтандырылған жұмыс орындарының арасындағы ақпарат алмасу есептеу желілерінің байланыс каналы бойынша жүзеге асады.

Локальді есептеу желілерінде қысқа ара қашықтық үшін мәліметтер жіберудің қарапайым әдісін пайдалануға, мәліметтер алмасудың жоғары жылдамдығына (100 Мбит/с-қа дейін) мүмкіндік беретін қымбат жоғары сапалы байланыс желілері пайдаланылуы мүмкін. Осы қызметтерге байланысты жіберілетін локальды есептеу желілері кең алуан түрлілігімен ерекшеленеді және әдетте, on-line режимінде қосымшаның іске асырылуын қарастырады.

Егер кәсіпорында (бірлестік, корпорация) қаланың әр жерлерінде, аймақтарды, елдерде және континенттерде территориялық оңашаланған бөлімшелер болса, онда жеке автоматтандырылған жұмыс орнының арасында ақпарат алмасу корпоративті есептеу желілерінің байланыс каналы арқылы жүзеге асырылады (КЕЖ).

Жойылған локальды есептеу желілері мен корпоративті есептеу желілеріндегі жеке компьютерлерді жалғау үшін әр түрлі телекоммуникациялық құралдар, оның ішінде телефон каналдары, радиоканалдар, спутникті байланыс пайдаланылады.

Мұндай күрделі және үлкен көлемдегі желілердің құрамында компьютерлердің әр түрлі типтерінің – үлкен ЭЕМ-нен бастап (оларды мейнфреймдар деп атау қабылданған) персоналды ЭЕМ-ге дейін, көптеген операциялық жүйелер және қосымшалар болуы оның өзіндік ерекшелігі болып табылады. Корпоративті желінің біртекті емес бөліктері пайдаланушыға барлық қажетті ресурстарға мүмкіндігінше деректер алуға айқын рұқсат бере отырып, бүтін желі ретінде жұмыс жасауы керек.

Корпоративті желілерде және локальды есептеу желілерінде есептеу желілеріндегі ақпарат – видеосурет, суреттер, дауыстар өңделе бастады. Мұндай мультимедиялық ақпараттарды желі арқылы жіберу оның мәліметтер дестесін жіберу кезіндегі кідіріске сезімталдығымен байланысты. Кідірістер, әдетте, желінің соңғы түйінінде мұндай ақпараттың бұрмалануына – «эха» эффектісінің туындауына, кейбір сөздердің талдануыны мүмкіндігінің болмауыны, суреттердің дірілдеуіне және т. б. алып келеді. Сондықтан корпоративті желілерде жаңа желілік технологиялар және жаңа құрал-жабдықтар есебінен мәліметтерді жіберу жылдамдығы локальды есептеу желілерінің дәстүрлі желілеріндегі жылдамдығына жақындайды (frame relay

желілерінде — 2 М бит/с, АТМ - 622 М бит/с).

### **1.3 Автоматтандырылған жұмыс орнының тілдік құралдары**

Автоматтандырылған жұмыс орнының тілдік құралдары, ең алдымен, пайдаланушы әрекеті мен ПЭЕМ реакциясының мағыналық жағынан сәйкес келуі үшін қажет. Оларсыз оқу процесі, диалог ұйымдастыру, қатені табу және жою мүмкін емес. Мұндай тілдерді өңдеудің қиындығы олардың әсіресе процедуралық емес болуы керектігінде болып табылады. Егер процедуралық тіл берілген жағдай қалай орындалатынын көрсетсе, онда процедуралық емес – нақтылаусыз орындау керектігін, ол үшін қандай әрекеттер қажет екендігін көрсетеді. Соңғы пайдаланушылар ақпараттық қажеттілікті жүзеге асыру процесін нақты білмейді және білуге болмайтындықтан, АЖО-ның интеллектуалдығы неғұрлым жоғары болса, оның тілдерінде соншалықты процедуралық емес мүмкіндіктер болмауы керек.

АЖО тілдері пайдаланушыға бағытталған, сонымен қатар кәсіби бағытталған болуы керек. Бұл тек кәсіби құрал-жабдықтар емес, қызметтік жағдайдың иерархиясы, оқыту деңгейі, қолданылатын мәліметтер деңгейі және т. б. бойынша бөлінетін пайдаланушы классификациясындағы өзгешеліктермен байланысты. Табиғи тілді пайдалану, мұндай тәсілдің қарапайымдылығына қарамастан, кейде қиын емес нәтиже алу үшін пернетақта арқылы үлкен конструкция енгізудің қажеттілігінен біршама белгілі артықшылықтарды бере алмайтындығын ескерген жөн.

Барлық тілдердегі сияқты, АЖО тілдерінің негізін алдын ала анықталған терминдер, сонымен қатар жаңа терминдер орнату көмегімен бар нәрселерді алмастырып немесе толықтыра отырып әдістерді сипаттау арқылы құру керек. Бұл АЖО жоспарлау кезінде АЖО-ның терминологиялық негізін анықталған түрде жіктеу қажеттілігіне алып келеді, яғни тілдің барлық негізгі синтаксистік құрылымын және терминдер мен олардың жиынтығы арасындағы семантикалық қарым-қатынасты анықтайды. Осыған байланысты АЖО-ның қарапайым жіктелуінде қажеттілік туындауы мүмкін, мысалы, мәліметтердің өңдеудің пайдаланушылық режимінде көріністерінің мүмкіндіктері бойынша: сандық, мәтіндік, аралас. Күрделі жағдайларда АЖО-ның жіктелуі деректер қорының ұйымдастырылуымен анықталуы мүмкін. Көбіне тілдің мүмкіндіктері, пайдаланушы ақпараттық қажеттіліктің іске асуымен сәйкес келетін формальды конструкциялар тұрғызуы мүмкін болатын ережелер тізімін анықтайды. Мысалы, кейбір АЖО-да барлық мәліметтер және конструкциялар кестелік түрде (кестелік АЖО) немесе арнайы түрдің операторлары (функционалды АЖО) түрінде бекітіледі.

Пайдаланушы тілдерін АЖО диалог түрі бойынша бөледі. Соңғы есепте диалогты қолдау құралдары пайдаланушыға қажетті тілдік конструкцияларды анықтайды.

АЖО-ны өңдеу кезінде болатын диалогтардың арасында көбірек пайдаланылатыны: ПЭЕМ әсер ететін диалог, жаттықпаған пайдаланушы диалогы және ақпараттың бекітілген кадрларының көмегімен диалог. ПЭЕМ әсер ететін диалог кезінде АЖО пайдаланушысы мнемониканы және тіл

конструкциясын зерттеуден толық босайды. Бұл әдістің модификациясының бірі – ұсынылған ПЭЕМ нұсқаларының бірі немесе бірнешеуі таңдалатын мәзір әдісі болып табылады.

ПЭЕМ әсер ететін формаларды толтыру диалогы кезінде пайдаланушы олардың келесі талдауымен және өңделуімен дисплейде арнайы таңдалған формаларды толтырады. Гибридті диалог пайдаланушыға да, ПЭЕМ-ға жа әсер етуі мүмкін.

Жаттықпаған пайдаланушымен диалог кезінде оған не істеу керектігіне байланысты пайдаланушыда күмән қалдырмауы мүмкін ПЭЕМ жауаптарының толық анықтығы қамтамасыз етілуі керек.

Ақпараттың бекітілген кадрларының көмегімен диалог кезінде ПЭЕМ бар болған тізімнен жауап таңдайды. Бұл жағдайда пайдаланушы тек өте қысқа жауаптарды ғана енгізеді, ал негізгі ақпарат автоматты түрде беріледі.

Сонымен қатар, диалог типі АЖО-ның жіктелуін анықтауы мүмкін, мысалы жаттықпаған пайдаланушының диалогтық құралдары АЖО. Осындай белгілері бойынша АЖО-ның жіктелуі пайдаланушының кәсібм бағыты бойынша жіктелуімен байланысты. Егер автоматтандырылған жұмыс орнын оларды іске асыратын бағдарламалық құралдар тұрғысынан қарастырса, онда АЖО-ның түрленуі өте кең тараған болуы мүмкін. Олар бағдарламалау тілі бойынша, пайдаланушыға бағдарламалаудың процедуралық құралдарын көрсету мүмкіндігі бойынша, эксплуатация процесі кезінде бағдарламалық жүйенің құрылуын аяқтау мүмкіндігі бойынша, деректер қорын басқару жүйесінің, пайдаланушы тілі бар интерпретатордың немесе транслятордың, қателерді табу және түзету құралдарының болуы бойынша және т.б. жіктелуі мүмкін. АЖО-да қолданылатын қолданбалы бағдарламалардың дестесі жүйенің нақты қосымшаға қосылуын қамтамасыз ету үшін параметрленуі мүмкін. Қолданбалы бағдарламалар дестесінің генераторлары пайдаланылуы мүмкін.

АЖО құрамына міндетті түрде негізгі есептік функцияларды және диалогтың ұйымдастырылуын қамтамасыз ететін әр түрлі бағдарламалық компоненттер, сонымен қатар деректер қорын басқару жүйесі, трансляторлар, анықтамалық жүйелер, негізгі мәліметтерден тұратын деректер қоры, диалог сценарийі, параметрді басқаратын нұсқаулықтар, қателер тізімі және т.б. АЖО-ның негізгі компоненттері оның құрамын анықтайды және әр түрлі белгілері бойынша АЖО-ның жіктелу мүмкіндігін қамтамасыз етеді.

АЖО шеңберінде пайдаланылуына байланысты АЖО-ның соңғы пайдаланушы арқылы дамуын қамтамасыз ететін АЖО құралдардың шеңберінде пайдалануға байланысты АЖО-ны екі үлкен класқа бөлеміз: қызмет ететін және интеллектуалды. Бұл екі класс та әр түрлі пайдаланушыларға арналуы мүмкін. Бірақ сонымен қатар, екі АЖО-ның да пайдаланушысы бола алмайтындығын алдын ала айтуға болатын пайдаланушылар да болады. Мысалы, қызмет көрсетуші тұлға (іс жүргізушілер, хатшылар) олар арқылы орындалатын функциялардың ерекшеліктеріне байланысты интеллектуалды АЖО-ны (өзінің тікелей қызметтерінде) қажет етпейді.

Қызмет көрсететін АЖО ұйымдастырушылық басқаруда былай бөлінуі мүмкін:

- ақпараттық-анықтамалық;
- есептік;
- мәтін құратын.

Интеллектуалды АЖО-ны ең алдымен мәліметтерге бағытталған және білімге бағытталған (дatalogиялық және фактологиялық) деп бөлуге болады. Ақпараттық-анықтамалық АЖО қандай да бір басқару процесіне қызмет етеді. Есептік АЖО өзінің мазмұны бойынша әр түрлі және пайдаланушының бірнеше категорияларымен қабылдануы мүмкін. Олардың көмегімен бір-бірімен байланысты және байланысты емес ұйымдастырушылық-экономикалық тапсырмалар, АЖО құру процесі кезінде анықталатын немес алдын ала анықталған мәліметтерді іздеу және өңдеу тапсырмалары қойылуы және шешілуі мүмкін. Мәтін құратын АЖО әр түрлі құрылымдағы мәтіндік ақпараттың құрылуына және өңделуіне арналған.

#### **1.4 Автоматтандырылған жұмыс орнының жіктелуі**

АЖО жеке, топтық, ұжымдық болуы мүмкін. ЭЕМ жүйесін тиімді құру мақсатында топтық және ұжымдық АЖО-ға байланысты – мамандарға (ұжымға) АЖО жұмысын ұйымдастыру талаптарын күшейту мұндай жүйедегі әкімшілік жүргізу функцияларын нақты анықтау керек. «Адам-машына» болып табылатын АЖО жүйесі ашық, икемді, үнемі пайдалануға және жетілдіруге бейімделген болуа керек. Мұндай жүйеде келесі аталғандар қамтамасыз етілуге тиіс:

- ақпаратты өңдеудің машыналық құралдарына мамандардың максималды жақындауы;
- диалогтық режимдегі жұмыс;
- АЖО-ның эргономика талаптарына сәкес жабдықталуы;
- компьютердің жоғары өнімділігі;
- бір типті процестердің максималды автоматтандырылуы;
- мамандардың өзін-өзі оқыту мүмкіндігі.

АЖО-да шешілетін тапсырмаларды шартты түрде ақпараттық және есептік деп бөлуге болады.

Ақпараттық тапсырмаларға ақпаратты кодтау, жіктеу, жинақтау, құрылымдық ұйымдастыру, түзету, сақтау, іздеу және беру жатады. Көбіне ақпараттық тапсырмалар арифметикалық және мәтіндік сипаттаманың және қатынастың (байланыстың) күрделі емес есептік және логикалық процедураларын енгізеді. Ақпараттық тапсырмалар, ереже бойынша, еңбекті көп қажет ететін болып табылады және маманның жұмыс уақытының үлкен бөлігін алады.

Есептік тапсырмалар толығымен рәсімделмеген болса да, рәсімделген болып табылады. Рәсімделген тапсырмалар формалды алгоритмдер базасында шешіледі және екі топқа бөлінеді: тура есеп тапсырмасы және математикалық модель негізіндегі тапсырмалар. Тура есеп тапсырмасы қарапайым алгоритмдер көмегімен шешіледі. Күрделірек тапсырмалар үшін әр



түрлі математикалық модельдерді қолдану талап етіледі.

Соңғы уақыттарда семантикалық деп аталатын толығымен рәсімделмеген тапсырмаларды шешу құралдарына үлкен мән беріледі. Мұндай тапсырмалар экономикалық объектілерді оперативті басқару кезінде, яғни толық емес ақпарат шартында шешім қабылдау кезінде жиі туындайды. АЖО құрылымы – оның жүйелері мен элементтерінің жиынтығы. Қамтамасыз ететін жүйелерге ең алдымен мыналар жатады: техникалық, ақпараттық, бағдарламалық және ұйымдастырушылық. Сонымен қатар, жүйелердің толық қатары бар.

Техникалық қамтамасыз ету маман жұмысын делдалсыз (бағдарламашылар, операторлар және т.б.) қарастыратын кәсіби персоналды компьютер қызмет ететін техникалық құралдардың жиынтығы жиынтығын білдіреді. Топтық АЖО-да мұндай компьютерлерді 4-6 адам пайдалана алады. Кәсіби персоналды компьютер жиынтығына процессор, дисплей, пернетақта, ақпараттың магнитті жинағышы жазу құралдары және графкүрушылар жатады. Техникалық құралдар жиынтығына желідегі әр түрлі АЖО-ны байланыстыруға арналған коммуникация құралдарын, сонымен қатар телефондық байланыс құралдарын жатқызуға болады.

Ақпараттық қамтамасыз ету – локальды деректер қорында сақталатын ақпараттар массиві. Ақпарат негізінен магнитті дискілерде ұйымдастырылады және сақталады. Оны басқару ақпаратты жазуды, ақпараттық тапсырмаларды іздеуді, оқуды, түзетуді және шешуді жүргізетін деректер қорын басқарудың бағдарламалық жүйесінің көмегімен жүзеге асырылады. АЖО-да бірнеше деректер қоры болуы мүмкін.

Ұйымдастырушылық қамтамасыз ету АЖО-ның жұмыс істеуін, жетілдірілуін және дамуын ұйымдастырудың, сонымен қатар кадр біліктілігін дайындаудың және арттырудың әдістері мен құралдарын енгізеді. Топтық және ұжымдық АЖО үшін ұйымдастырушылық қамтамсыз ету жүйесіне АЖО әкімшілік іс жүргізу функциялары енгізіледі: жобалау, жоспарлау, есептеу, басқару, талдау, реттеу, инфражүйесі бар ұйымдастырушылық байланыстар және т.б. Ұйымдастырушылық қамтамасыз ету АЖО пайдаланушыларының құқықтары мен міндеттерін анықтауды және құжаттық рәсімдеуді қарастырады.

Бағдарламалық қамтамасыз ету жүйелік және қолданбалы бағдарламалық қамтамасыз етуден тұрады. Жүйелік қамтамасыз етудің негізі операциялық жүйе және бағдарламалау жүйесі, мысалы Basic алгоритмдік тілі болып табылады. Жүйелік бағдарламалар ақпаратты өңдеудің рационалды технологиясын қамтамасыз етеді. АЖО желісіндегі ақпараттық байланысты және әр түрлі канал бойынша АЖО байланысын қамтамасыз ету үшін жүйелік бағдарламалауға жататын бағдарламалық құралдар қолданылады.

### **1.5 Деректер қорын ұйымдастырудың қанағаттандыратын талаптары.**

Жеке кәсіпкерлік жалдамалы еңбекті пайдаланып, не пайдаланбай жүзеге асыралады, сонымен қатар заң актілерімен шектелмеген шаруашылық қызметтің барлық түрін де жүргізеді.

Жеке кәсіпкердің мынадай құқықтары бар:

- мемлекеттік кәсіпорындардың және басқадай меншік түрлеріне негізделген кәсіпорындардың мүліктерін толық немесе жартылай иемдену;
- шаруашылық жүргізуші субъектілер қызметіне өз мүлкімен қатысу;
- келісімшарт негізінде жұмыскерлерді жалдау және жұмыстан

босату;

- еңбекақы нысандарын, жүйесін, мөлшерін өздерінше белгілеу;
- шаруашылық қызметтің бағдарламасын қалыптастыру;
- жабдықтаушылар мен тұтынушыларды таңдау;
- бағалар мен тарифтерді белгілеу;
- сыртқы экономикалық қатынастарда болу.

Жеке кәсіпкерлік өзінің құқықтық иелігіне жататын мүлік базасында өз бетінше қызмет ететін бір ғана жеке тұлғадан құрылады. Некеде тұратын жеке тұлғаның жеке кәсіпкерлікті жүзеге асыруы үшін жұбайынан келісім алуының қажеті жоқ. Тек жеке кәсіпкерлікті іске асыру үшін жеке тұлға жұбайы екеуінің ортақ мүлкін пайдаланар болса ғана, екіншісінің келісімін алуы қажет болады.

Жеке кәсіпкерліктің артықшылықтары туралы:

- адамның еңбекпен қамтамасыз етілу және ақша табу проблемаларын шешеді;

- қымбат емес әрі тез арада тіркеуге болады;
- мемлекеттік реттеуге аз ұшырайды;
- тым көп құжаттар жүргізуді қажет етпейді.

Жеке кәсіпкерліктің кемшіліктері:

- мұраға қалмайды;
- алғашқы капитал жинау қиындығы;
- кәсіпкер жеке мүлкін тәуекелге тігеді.

### **1.6 Көпжақты байланыс орнату**

Әр түрлі бағдарламашыларға әр түрлі логикалық файлдар қажет болады. Бұл файлдар бір деректер жиынтығынан алынады. Деректерді есте сақтайтын элементтер арасында әртүрлі байланыстар болуы мүмкін. Кейбір деректер қоры өзара байланыстың күрделі шиеленісуінен тұрады. Деректерді ұйымдастыру әдісі осы өзара байланысты және оған енгізілген тез келісімді ыңғайлы көрсету мүмкіндігі қамтамасыз етілетіндей болуы тиіс. Деректер қорын басқару жүйесі деректер мен олардың арасындағы байланыстың қажетті логикалық файлын алу мүмкіндігін қамтамасыз етуі керек. Логикалық

файлдың қолданбалы бағдарламадағы бейнесі мен деректерді физикалық сақтау әдісі арасында біраз ұқсастықтық болуы керек [7].

### **1.7 Өнімділігі**

Терминал операторы арқылы пайдалану үшін арнайы жасалған деректер қоры адамдар диалогы үшін қанағаттандырылатын жауап уақытын – терминалды қамтамасыз етеді. Одан басқа, деректер қорын басқару жүйесі сәйкес өткізу мүмкіндігін қамтамасыз етуі тиіс. Сұраныстың шағын ағынына есептелген жүйелерде өткізу мүмкіндігі деректер қорының құрылымына аз ғана шектеу қояды. Сұраныстың үлкен ағыны бар жүйелерде, мысалы авиабилеттерді резервтеу жүйесінде өткізу мүмкіндігі деректердің физикалық сақталуын ұйымдастыруды таңдауға шешуші әсер етеді.

Тек дестелік өңдеуге арналған жүйелерде жауап уақыты аса маңызды емес және физикалық ұйымдастыру әдісі тиімді дестелік өңдеуді қамтамасыз ету шартынан таңдалуы мүмкін [7].

### **1.8 Минималды шығындары**

Деректер қорын құруға және пайдалануға кеткен шығындарды азайту үшін ішкі жадыға талапты азйтатын ұйымдастыру әдістері таңдалады. Осы әдістерді пайдалану кезінде жадыдағы деректердің физикалық көрінісі қолданбалы бағдарламашы пайдаланатын көріністен қатты ерекшеленеді. Бір көріністің екіншісіне айналуы бағдарламалық қамтамасыз етуді немесе, егер мүмкін болса, аппараттық немесе микробағдарламалық құралдарды жүзеге асырады. Мұндай жағдайларда түрлендіру алгоритміне кеткен шығындар мен жадыны үнемдеу арасында таңдау жасауға тура келеді [7].

### **1.9 Минималды артықтығы**

Деректер қорын басқару жүйесін қолдануға дейін пайда болған өңдеу жүйелерінде ақпараттық қорлар артықшылықтың өте жоғары деңгейіне ие болды. Ленталық кітапханалардың басым бөлігінде көптеген артық мәліметтер бар. Тіпті біріктірілген деректер қорымен байланыстырылған ақпараттың өсуіне байланысты деректер қорының қолданылуы кезінде артық деректердің пайда болуының потенциалды мүмкіндігі үнемі біртіндеп артады. Жолдың артық деректері жадыдан көбірек орын алса, жаңарудың бірден көп операциясын талап етеді. Деректер қорын ұйымдастыру мақсатында артық деректер жойылуы керек [7].

### **1.10 Іздеу мүмкіндігі**

Деректер қорын пайдаланушы оған сақталған деректер бойынша әр түрлі сұрақтармен баруына болады. Қазіргі заманғы коммерциялық қосымшалардың көпшілігінде сұраныстар типі алдын ала анықталған, және деректердің физикалық ұйымдастырылуы оларды қажетті жылдамдықпен өңдеу үшін жетілдіріледі. Жүйеге қойылатын өспелі талаптар осындай сұраныстарды өңдеуді қамтамасыз етуге немесе алдын ала жоспарланбаған осындай жауаптардың құрылуына негізделеді [7].

### **1.11 Бүтіндігі**

Егер деректер қоры көптеген пайдаланушы қолданатын деректерден тұрса, деректер элементтері мен олардың арасындағы байланыстар бұзылмауы өте маңызды. Қателер мен әр түрлі кездейсоқ бұзылулардың туындау мүмкіндігін ескеру керек. Деректерді сақтау, оларды жаңарту, деректерді қосу процедуралары, жүйе бұзылулардың пайда болу жағдайында деректер жойылмастан қайта құрыла алатындай болуы керек. Есептеу жүйесі онда сақталған деректердің бүтіндігіне кепілдік беруі керек [7].

### **1.12 Қауіпсіздігі және құпиялығы**

Деректер қоры жүйесіндегі деректер құпияда сақталуы керек. Есте сақталатын ақпарат кейде оны пайдаланатын мекеме үшін өте маңызды. Ол жоғалуға және ұрлануға тиісті емес. Деректер қорындағы ақпараттың төзімділігін арттыру үшін оны аппараттық немесе бағдарламалық бұзылулардан, катастрофалық және криминалдық жағдайлардан, компонентті емес немесе оны дұрыс пайдаланбайтын арам ниетті қолданушылардан қорғау маңызды.

Деректердің қауіпсіздігі ретінде оған құқығы жоқ тұлғалардың кездейсоқ немесе алдын ала жоспарланбаған қолжетімділіктен, деректердің авторластырылмаған модификациясынан немесе олардың жойылуынан деректердің қорғалуын қарастырады.

### **1.13 Өткенмен байланыс**

Белгілі уақыт аралығында деректерді өңдеу жүйесін эксплуатациялайтын кәсіпорындар бағдарламаларды, процедураларды жазуға және деректерді сақтауды маңызды құралдарды жұмсайды. Фирма есептік орнатуда деректер қорын басқарудың жаңа бағдарламалық қамтамасын пайдалана бастаған кезде, ол осы орнатудағы бағдарламамен жұмыс жасай алады. Мұндай шарт бағдарламалық және ақпараттық үйлесімділіктің болуын талап етеді, және оның болмауы жаңа деректер қорын басқару жүйесіне өту кезінде тоқтататын негізгі фактор болуы мүмкін. Бірақ өткенмен байланыс мәселесі деректер қорын басқару құралдарының дамуын тоқтатпағаны маңызды [7].

### **1.14 Болашақпен байланыс**

Болашақпен байланыс ерекше маңызды болып саналады. Болашақта мәліметтер және олардың сақталу ортасы көптеген бағыт бойынша өзгереді. Кез келген коммерциялық ұйым уақыт өте өзгерістерге ұшырайды. Әсіресе, деректерді өңдеу жүйесінің пайдаланушысы үшін бұл өзгерістер қымбат болады. Өте қарапайым өзгерістерді іске асыруға қажетті үлкен шығындар осы жүйелердің дамуын қатты тежейді. Бұл шығындар деректердің қайта құрылуына, қайта жазылуына өзгерістер енгізу нәтижесінде пайда болған қолданбалы бағдарламаны ретке келтіруге жұмсалады. Уақыт өте мекемедегі



қолданбалы бағдарламалардың саны артады, және сондықтан барлық бағдарламаларды қайта жазудың болашағы іске аспайды. Деректер қорын өңдеу кезінде ең маңызды тапсырмалардың бірі – деректер қорын оның өзгерісін қолданбалы бағдарламалардың модификациясына орныдауға болатындай жоспарлау [7].

### **1.15 Пайдалану қарапайымдылығы**

Деректерді жалпы логикалық сипаттаудың көрінісі үшін қолданылатын құралдар қарапайым және көрнекі болуы керек.

Бағдарламалық қамтаманың интерфейсі соңғы пайдаланушыға бағытталуы керек және пайдаланушы деректер қоры теориясы бойынша қажет етпейтін мүмкіндікті ескереді [7].

## **2 Деректер қорын құру құралдары**

### **2.1 SQL тілі деректер қорының стандартты тілі ретінде**

SQL таңымалдығының қарқынды өсуі қазіргі заманғы компьютерлік өнеркәсіпте өте маңызды үрдіс болып табылады. Соңғы бірнеше жылда SQL деректер қорының жалғыз тілі болып келді. Қазіргі кезде SQL-ді жоғарыдан үлкен ЭЕМ-дерде сияқты персоналды компьютерлерде жұмыс жасайтын жүздеген ДҚБЖ қолдап отыр. Қабылданды, содан соң SQL-ға ресми халықаралық стандарт толтырылды. SQL тілі барлық жетекші бағдарламалық өніммен қамтамасыз етушілер шығаратын деректер қорын басқару жүйесінің архитектурасындағы маңызды бөлімше болып табылады және Microsoft компаниясының өнімінің деректер қоры саласындағы стратегиялық бағыты қызметін атқарады. IBM компаниясының қосымша зерттеу жобасын орындау нәтижесінде туындаған SQL бүгінгі күнде мықты нарықтық фактор ретінде де кеңінен танымал.

### **2.2 SQL тілі**

SQL компьютерлік деректер қорындағы мәліметтерді өңдеуге және оқуға арналған құрал болып табылады. SQL – құрылымдық сұраныстар тілінің қысқартылған атауы (Structured Query Language). Аты айтылғандай, SQL пайдаланушының деректер қорымен өзара әрекетін ұйымдастыру үшін пайдаланылатын бағдарламалау тілі. Шын мәнінде SQL реляциялық типтегі деректер қорымен жұмыс жасайды. Қабылданған схемаға сәйкес, есептеу жүйесінде маңызды ақпарат сақталатын мәліметтер қоры болады. Егер есептеу жүйесі бизнес саласына қатысты болса, онда деректер қорында шығарылатын өнімнің материалдық құндылықтары, сатылым көлемі және жалақысы жайлы ақпарат сақталуы мүмкін. Персоналды компьютердағы деректер қорында жазып шығарылған чектар, телефондар және мекен-жайлар туралы ақпарат немесе үлкенірек есептеу жүйесінен шығарылған ақпараттар сақталуы мүмкін. Деректер қорын басқаратын компьютерлік бағдарлама деректер қорын басқару жүйесі немесе ДҚБЖ деп де атайды.

Егер пайдаланушыға деректер қорынан мәліметтерді оқу қажет болса, ол

оны ДҚБЖ-нен SQL көмегімен сұрайды. ДҚБЖ сұранысты өңдеп, қажетті мәліметтерді табады және оларды пайдаланушыға жібереді. Мәліметтерді сұрау процесі және нәтиже алу деректер қорына сұраныс деп аталады: осыдан оның атауы – құрылымданған сұраныс тілі.

Бірақ бұл атау шындыққа аса сәйкес келмейді. Біріншіден, қазір SQL қарапайым сұраныс құру құралына қарағанда үлкен мәнге ие, алайда ол бастапқы осы үшін жасап шығарылған. Ештеңеге қарамастан, мәліметтерді оқу бұрынғыша SQL-дің маңызды функцияларының бірі болып қала береді, қазір бұл тіл ДҚБЖ-нің пайдаланушыға мүмкіндік туғызатын барлық функционалдық мүмкіндіктерін іске асыру мақсатында пайдаланылады, олардың ішінде:

- мәліметтерді ұйымдастыру. SQL пайдаланушыға мәліметтер көрінісінің құрылымын өзгертуге, сонымен қатар деректер қоры элементтерінің арасында қарым-қатынас орнатуға мүмкіндік береді.

- мәліметтерді оқу. SQL пайдаланушыға немесе қосымшаға деректер қорындағы мәліметтерді оқуға және оны пайдалануға мүмкіндік береді.

- мәліметтері өңдеу. SQL пайдаланушыға немесе қосымшаға деректер қорын өзгертуге, яғни оған жаңа мәліметтер қосуға, сонымен қатар ондағы бар мәліметтерді жоюға және жаңартуға мүмкіндік береді.

- рұқсат етуді басқару. SQL-дің көмегімен пайдаланушының оқу және өзгерту мүмкіндіктерін шектеуге және оларды мүмкіндік берілмеген рұқсат етуден қорғауға болады.

- мәліметтерді бірлесіп пайдалану. SQL пайдаланушы мен жұмысшының мәліметтерді бірлесіп пайдалануын бір-біріне кедергі жасамас үшін, қатар ұйымдастырады.

- мәліметтердің бүтіндігі. SQL деректер қорын үйлестірілмеген өзгерістердің әсерінен бұзылудан және жүйенің істен шығуынан қорғай отырып, оның бүтіндігін қамтамасыз етеді.

Сондықтан, SQL ДҚБЖ-мен өзара әрекет үшін мықты тіл болып табылады.

Екіншіден, SQL – COBOL, FORTRAN немесе С сияқты толыққанды компьютерлік тіл емес. SQL-де шартты тексеруге арналған IF операторы, өтулерді ұйымдастыруға арналған GOTO операторы және DO операторы немесе циклдарды құруға арналған FOR операторы жоқ. SQL операторлары базалық тілде құрылады, мысалы COBOL, FORTRAN немесе С деректер қорына рұқсат алуға мүмкіндік береді. Одан басқа, С сияқты тілден SQL операторлары функцияның шақыру интерфейстерін пайдалана отырып, ДҚБЖ-сін анық түрде жіберуге болады.

Соңында, SQL – бұл С және Pascal сияқты мықты құрылымданған тілдермен салыстырғанда әлсіз құрылымданған тіл. SQL операторлары ағылшынша сөйлемді еске түсіреді және оператордың мағынасына әсер етпейтін, бірақ оның оқылуын жеңілдететін «бос сөздерден» тұрады. SQL-да қисынсыздық жоқ, сонымен қатар дұрыс сияқты көрінгенімен, мағынасы жоқ SQL операторларының құрылуын болдырмайтын арнайы ережелер атары бар.

Тура емес аталуына қарамастан, қазіргі кезде SQL реляциялық деректер қорымен жұмыс жасауға арналған жалғыз стандартты тіл болып табылады.

SQL – мықты және үйренуге айтарлықтай жеңіл тіл [8].

### 2.3 SQL-дің артықшылықтары

SQL – бұл түсінуге жеңіл, сонымен қатар мәліметтерді басқаратын әмбебап бағдарламалық құрал. SQL тілінің жетістігі оның келесідей артықшылықтарын тудырды:

- нақты ДҚБЖ-нен тәуелсіздігі;
- бір есептеу жүйесінен екіншісіне ауысу қабілеті;
- стандарттарының болуы;
- IBM компаниясымен мақұлдануы (ДҚБЖ DB2);
- Microsoft компаниясының қолдауы (ODBC хаттамасы);
- реляционды негіз;
- ағылшын тілін еске салатын жоғары деңгейлі құрылым;
- арнайы интерактивті сұраныстарды толтыру мүмкіндігі;
- деректер қорына бағдарламалық рұқсатты қамтамасыз ету;
- мәліметтердің әр түрлі көрінісінің мүмкіндігі;
- деректер қорымен жұмыс жасауға арналған тіл сияқты толыққандылығы;
- мәліметтердің динамикалық анықталу мүмкіндігі;
- клиент/сервер архитектурасын қолдау.

Жоғарыда аталған барлық факторлар SQL-дің персоналды компьютерда, мини-компьютерда және үлкен ЭЕМ-дердегі мәліметтерді басқаруға арналған стандартты құрал болуының себебі болды. Төменде бұл факторлар толығырақ қарастырылған [2].

Нақты ДҚБЖ-дан тәуелсіздігі. ДҚБЖ-нің барлық жетекші жеткізушілері SQL-ді қолданады, және SQL қолдамаған бірде-бір жаңа ДҚБЖ жетістікке жете алмайды. Реляциялық деректер қорын, онымен жұмыс жасайтын бағдарламаны бір ДҚБЖ-дан екіншісіне минималды жетілдіру арқылы және қызметкердің қайта дайындауы арқылы ауыстыруға болады. Сұраныстарды, есептеу генераторларын және қосымшаның генераторларын құруға арналған бағдарламалар сияқты персоналды компьютерлерге арналған ДҚБЖ-ның құрамына кіретін бағдарламалық құралдар көптеген типтердің реляциялық деректер қорымен жұмыс жасайды. Сондықтан, SQL оның танымалдығының маңызды себебі болып табылатын нақты ДҚБЖ-нен тәуелсіздігін қамтамасыз етеді.

Бір есептеу жүйесінен екіншісіне өту қабілеті. ДҚБЖ-сін жеткізушілер әр түрлі есептеу жүйелеріне арналған бағдарламалық өнімдерді ұсынады: персоналды компьютерлерден және жұмыс станцияларынан локалды желілерге, мини-компьютерлерге және үлкен ЭЕМ-дерге дейін. SQL көмегімен құрылған және бір тұтынушылық жүйеге есептелген қосымшалар өзінің өсу көлемі бойынша үлкенірек жүйеге ауысуы мүмкін. Корпоративті реляциялық деректер қорынан алынған ақпарат деректер қорында жеке бөлімшелерге немесе жеке деректер қорына жүктелуі мүмкін. Соңында, реляциялық деректер қорына арналған қосымшалардың алдымен үнемді

персоналды компьютерлерде үлгісін жасап алып, сосын қымбат көп пайдаланылатын жүйеге ауыстыруға болады.

SQL тілінің стандарттары. SQL тілінің ресми стандарты Америкалық ұлттық стандарттар институты (American National Standards Institute — ANSI) және Халықаралық стандарттар ұйымы арқылы (International Standards Organization — ISO) 1986 жылы жарияланды және 1992 жылы айтарлықтай кең тарады. Сонымен қатар, SQL АҚШ-тың ақпаратты өңдеу бойынша федералды стандарты (FIPS — Federal Information Processing Standard) болып табылады және оған сәйкес келуі есептеу техникасы саласына қатысты үлкен үкіметтік келісім-шартта болатын негізгі талаптардың бірі болып табылады. Еуропада UNIX операциялық жүйесінің негізіндегі бағдарламалаудың ауыспалы ортасына арналған X/OPEN стандарты SQL-ді деректер қорына рұқсат беруге арналған стандарт ретінде көрсетеді. SQL Access Group — компьютерлік құралдардың және деректер қорының жеткізушілерінің консорциумы — SQL үшін Microsoft компаниясының ODBC хаттамасының негізі болып табылатын және X/OPEN стандартына кіретін функцияларды шақырудың стандартты интерфейсін анықтады. Бұл стандарттар SQL-ді мақұлдайтын ресми мөр ретінде қызмет етті және олармен нарықты жаулап алуды тездетті [8].

SQL-дің IBM компаниясымен мақұлдануы (СУБД DB2). SQL-ді IBM компаниясының ғылыми қызметкерлері ойлап тапты және бағдарламалық қамтамасыз етудің дестелерінің жиынтығындакеңінен пайдаланылады. Оны растаушы ретінде IBM компаниясының флагмандық СУБД DB2 қызмет етеді. IBM компаниясының компьютерлерінің барлық негізгі жиынтығы SQL-ді қолдайды: персоналды компьютерлерге арналған PS/2 жүйесі, AS/400 орталық деігей жүйесі, UNIX негізіндегі RS/6000 жүйесі, сонымен қатар MVS операциялық жүйелері және үлкен ЭЕМ-нің VM. M компаниясының SQL-ді қолдауы оның мойындалуын жеңілдетті.

ODBC хаттамасы және Microsoft компаниясы. Microsoft компаниясы өзінің Windows операциялық жүйесінің мағызды бөлігі ретінде деректер қорына рұқсат етуді қарастырады. Деректер қорына рұқсат етуді қамсыздандыру бойынша бұл компанияның стандарты ODBC (Open Database Connectivity — ашық деректер қорымен өзара әрекеттесу) — SQL-ға негізделген бағдарламалық интерфейс болып табылады. ODBC хаттамасы Microsoft компаниясының өзі сияқты басқа да жетекші жеткізушілер жасап шығарған Windows-тың кең тараған қосымшалары арқылы жұмыс жасайды. ODBC-ның жұмыс жасауын барлық жетекші реляциялық деректер қоры қамтамасыз етеді. Сонымен қатар, ODBC SQL Access Group жеткізушілерінің консорциумымен мақұлданған стандарттарға негізделеді [2].

Реляционды негіз. SQL реляционды деректер қорының тілі болып табылады, сондықтан ол мәліметтер көрінісінің реляционды моделінің танымал болған кезінде танымал болды. Реляциялық деректер қорының кестелік құрылымы пайдаланушыға түсінікті, сондықтан SQL тілі қарапайым және оқып білу үшін жеңіл болып табылады. Реляционды модельде



реляционды деректер қорының эволюциясы мен іске асырылуы негізделген көрнекті теориялық іргетас бар. Реляциялық модельдің табыстылығы бойынша танымалдық толқынында SQL реляциялық деректер қорына арналған жалғыз тіл болды [4, 7].

Ағылшын тілін еске салатын жоғары деңгейлі құрылым. SQL операторлары олардың оқылуын және түсіндірілуін оңайлататын қарапайым ағылшынша сөйлемдер сияқты көрінеді. Бұл SQL операторының мәліметтерді іздеу тәсілін анықтайтын емес, алуға қажетті мәліметтерді сипаттауымен анықталған. Реляциялық деректер қорындағы кестелер мен бағандарда ұзақ сипаттамалық атаулар болып мүмкін. Нәтижесінде SQL операторларының көпшілігі олардың атауларына толық сәйкес келуін білдіреді, сондықтан оларды қарапайым түсінікті сөйлемдер ретінде оқуға болады.

Интерактивті сұраныстар. SQL пайдаланушылардың мәліметтерге жылдам рұқсат алуын қамтамасыз ететін интерактивті сұраныстар тілі болып табылады. SQL-дің көмегімен пайдаланушы өте күрделі сұраныстарға санаулы минуттарда немесе секундтарда интерактивті режимде жауап ала алады, ал бағдарламашыға пайдаланушы үшін сәйкес бағдарламаны жазу үшін күндер немесе апталар қажет болуы мүмкін. SQL жылдам сұраныстар жіберетіндіктен, мәліметтер қолжетімді бола бастады және оларды дәйекті ете отырып шешім қабылдауға жәрдемдесуі мүмкін [4].

Деректер қорына бағдарламалық рұқсат. Бағдарламашылар деректер қорының сұранысынан тұратын қосымшаларды жазу үшін SQL тілін пайдаланады. Сол бұрынғы SQL операторлары интерактивті сияқты бағдарламалық рұқсат ету ретінде қолданылады, сондықтан мәліметтер қорына сұраныстан тұратын бағдарлама бөлімдерін алдымен интерактивті режимде тестілеп, содан соң бағдарламада құруға болады. Бағдарламалық рұқсат етуге арналған дәстүрлі деректер қорында тек бағдарламалық құралдар қолданылады, ал жылдам сұраныстарды орындау үшін – осы екі рұқсат ету режимінің арасында қандай да бір байланысы жоқ басқалары [7].

Мәліметтердің әр түрлі көріністері. SQL-дің көмегімен деректер қорын құрушы деректер қорының әр түрлі пайдаланушылары оның құрылымының және мазмұнының әр түрлі көріністерін көре алатындай жасауы мүмкін. Мысалы, деректер қорын әрбір пайдаланушы деректер қорының бөлімдеріне немесе сауда аймағына байланысты мәліметтерді көре алатындай етіп жобалауға болады. Одан басқа, деректер қорының әр түрлі бөлімдеріндегі мәліметтер араластырылған және пайдаланушыға бір қарапайым кесте ретінде көрсетілуі мүмкін. Осыған сәйкес, көріністерді деректер қорының қорғанысын күшейту үшін және оның талаптарын жеке пайдаланушының нақты талаптары ретінде пайдалануға болады [8].

Деректер қорымен жұмыс жасауға арналған толыққанды тіл. Бастапқыда SQL интерактивті сұраныстар тілі ретінде ойлап табылған, бірақ қазір ол деректерді оқу шеңберінен алшақтап кетті. SQL деректер қорын құруға, оның қорғанысын басқаруға, оның мазмұнын өзгертуге, деректерді оқуға және мәліметтерді қатар жұмыс жасайтын бірнеше пайдаланушымен бірігіп пайдалануға арналған толыққанды және логикалық тіл болып табылады.

Тілдің бір бөлімін оқу кезінде меңгерілген тәсілдер кейінірек пайдаланушының жұмыс өнімділігін жоғарылататын басқа командаларда пайдаланылуы мүмкін [2].

Мәліметтердің динамикалық анықталуы. SQL-дің көмегімен пайдаланушы оның мазмұнын қолданып жатқан кезде де деректер қорының құрылымын динамикалық тұрғыдан өзгертуге және кеңейтуге болады. Бұл деректер қорының құрылымының өзгеруі кезінде деректер қорына рұқсатты болдырмайтын деректердің статикалық анықтамасының тіліндегі үлкен артықшылық. Сондықтан, SQL деректер қорына нақты уақыт көлемінде орындалатын қосымшаның жұмысын тоқтатпастан өзгергіш талаптарға бейімделу мүмкіндігін бергендіктен, максималды икемділікті қамтамасыз етеді [7].

## 2.4 SQL операторлары

SQL тілі сұраныстарды құруда, сонымен қатар реляциялық мәліметтер қорын жаңарту және басқару үшін қолданылады. SQL толық бағдарламалау тілі болып табылады, оған тек сұраныс операциялары ғана емес, DDL – Data Definition Language – мәліметтерді сипаттау тіліне сәйкес келетін операторлар да қатысады. Бұдан басқа, бұл тілде мәліметтер қорын басқаруға арналған операторлар да бар.

Объектілерді құру және өзгерту үшін мәліметтерді анықтау операторлары DDL қолданылады:

- CREATE TABLE – кесте құру. Мәліметтер қорында жаңа кесте құрады;

- DROP TABLE – кестені жою. Мәліметтер қорындағы кестені жояды;

- ALTER TABLE – кестені өзгерту. Қолданыстағы кестенің құрылымын немесе берілген кестенің бүтіндігінің шегін өзгертеді;

- CREATE VIEW – көрініс құру. Кейбір SQL-сұраныстарға сәйкес келетін виртуалды кестелерді құрады;

- DROP VIEW – бұрын құрылған көріністі жояды;

- ALTER VIEW – көріністі өзгерту. Бұрын құрылған көріністі өзгертеді;

- CREATE INDEX – индекс құру. Кейбір кестелерге арнап индекстарға кірген атрибуттарға тез қолжетімділікті қамтамасыз ету үшін индекс құрады;

- DROP INDEX – бұрын құрылған индексті жояды.

Мәліметтерді өзгерту үшін мәліметтерді манипуляциялайтын операторлар пайдаланылады:

- DELETE – қатарларды өшіру. Базалық кестедегі фильтрациялау шартымен сәйкес келетін бір немесе бірнеше қатарларды өшіреді. Оператордың қолданылуы бүтіндікті қолдау принципімен келіседі, сондықтан бұл оператор тіпті ол синтаксистік жағынан дұрыс жазылса да, әрқашан дұрыс орындалмауы мүмкін;

- INSERT – қатар қою. Базалық кестеге бір қатар қояды. Бірден бірнеше қатар бір кестеден немесе сұраныстан базалық кестеге ауысуы мүмкін болғанда, оператор модификациясы болады; UPDATE – қатарды жаңарту. Фильтрациялау шартымен сәйкес келетін бір немесе бірнеше қатардағы бір

немесе бірнеше бағандардың мәндерін жаңартады.

Мәліметтер талдауын өңдеу үшін сұраныстар тілі Data Query Language (DQL) қолданылады:

– SELECT – қатар тандау. Реляционды алгебраның барлық операторларын алмастыратын және сұранысқа сәйкес келетін нақты қатынасты құруға мүмкіндік беретін оператор;

– SELECT – кілттік сөз. Барлық сұраныстар осы сөзден басталып, бұдан соң бос орын қалдырылады. Одан соң сұрыптау әдісі - дубликаты өшірілген (DISTINCT) немесе өшірілмеген (ALL, үнсіз келісім бойынша) болады. Содан соң кестеден сұраныспен таңдалатын үтір арқылы жазылған бағандар тізімі немесе барлық қатардытаңдау үшін \* (жұлдызша) символы орналасады. Бұл жерде аталмаған кез келген кестелер команданың орындалуына сәйкес келетін нақты қатынасқа қосылмайды. Бұл, әрине, олар жойылады немесе олардың ақпараттары кестеден өшіріледі деген сөз емес, өйткені сұраныс кестелердегі ақпаратқа әсер етпейді – ол тек мәліметтерді көрсетеді;

– FROM – әр сұраныста болуы тиіс SELECT-ке ұқсас кілттік сөз. Ол бос орын және содан соң ақпарат көзі ретінде пайдаланылатын кестелердің атымен құрылады. Егер бірден көп кесте аты көрсетілген жағдайда, аталған кестелермен декарттық туынды операциясы іске асатыны анық байқалмайды;

– WHERE – соңынан предикаты – сұрыптауға түсу үшін ол қанағаттандыруы тиіс кестедегі жазбаларға қойылатын шарты болатын кілттік сөз. Бұл бөлімде нәтиже қатарының сұрыптау шарты және негізгі кестені біріктіру шарыт қойылады;

– WHERE бөлімінің шартын көрсетуде келесі предикаттар пайдаланылуы мүмкін:

Дәстүрлі мәні бар салыстыру предикаттары { =, <, >, <=, >=, <> }.

Between A and B предикаты A және B арасындағы мәнді береді.

Салыстырылатын мән диапазон шекарасын қоса берілген диапазонға түсетін болса, предикат шындық. Сонымен қатар, стандартта салыстырылатын мән диапазон шекарасын қоса берілген диапазонға түспейтін болса, шындық болатын қарама-қарсы Not Between A and B предикаты берілген.

Жиынтыққа кіру предикаты IN салыстырылатын мән берілген мәннің жиынтығына кіргенде, шындық болады. Бұдан мәннің жиынтығы қарапайым аударыммен немесе құрылған сұраныстармен берілуі мүмкін. Сонымен қатар, салыстырылатын мән берілген жиынтыққа кірмеген жағдайда шындық болатын қарама-қарсы NOT IN (жиынтық) предикаты бар.

LIKE және NOT LIKE түріндегі салыстыру предикаттары. LIKE предикаты берілген мән салыстырылатын шаблонның берілуін талап етеді, егер салыстырылатын мән шаблонға сәйкес келсе, предикат шындық, және қарама-қарсы жағдайда жалған. NOT LIKE предикаты қарама-қарсы мағынаға ие. IS NULL анықталмаған мәні бар салыстыру предикаты. Анықталмаған мән уақыттың берілген сәтінде анықталмаған мән ретінде реляциялық моделде интерпретацияланады. Бұл мән қосымша ақпарат пайда болғанда уақыттың кез келген сәтінде кейбір нақты мәндерге алмастырылуы мүмкін. Анықталмаған мәндердә салыстыру кезінде салыстырудың стандартты

ережелері әсер етпейді: бір анықталмаған мән ешқашан басқа анықталмаған мәнге тең болып есептелмейді. Кейбір атрибут мәнінің теңдігін анықтау үшін анықталмаған ретінде арнайы стандартты предикаттар пайдаланылады: <атрибут аты> IS NULL және <атрибут аты> IS NOT NULL. Егер берілген қатарда көрсетілген атрибут анықталмаған мәнге ие болса, IS NULL предикаты «Шындық» (TRUE) мәнін қабылдайды, ал IS NOT NULL предикаты – «Жалған» (FALSE), қарама-қарсы жағдайда IS NULL предикаты «Жалған» мәнін қабылдайды, ал IS NOT NULL предикаты «Шындық» мәнін қабылдайды.

GROUP BY бөлімінде топтау аймағының тізімі беріледі.

HAVING бөлімінде шарттары әрбәр группаға қойылатын предикаттар беріледі.

ORDER BY бөлімінде нәтижені реттеу аймақтарының тізімі, яғни соңғы қатынасында сұрыптау ретін анықтайтын аймақтар қатары беріледі.

## **2.5 Деректер қорының архитектурасы**

Деректер қорын ұйымдастырудың әдістерін қарастыру үшін бірнеше түсініктерді анықтау керек.

Деректер қорының ядросы ішкі жадының мәліметтерін басқаруға, оперативті жадының буферін басқаруға, транзакцияны және журналдауды басқаруға жауап береді. Сәйкес ядроның мұндай компоненттерін (дегенмен, кейбір желілерде бұл компоненттер анық байқалады) мәліметтер менеджері, буферлер менеджері, транзакция менеджері ретінде бөліп қарастыруға болады. Деректер қорының ядросы пайдаланушыға тікелей қолжетімсіз және бағдарламады қолданатын жеке интерфейске ие. Деректер қорының ядросы ДҚБЖ-ның негізгі резиденттік бөлігі болып табылады. «Клиент-сервер» архитектурасын пайдалану кезінде ядро жүйенің серверлік бөлігінің негізгі құрушысы болып табылады.

Дайын қосымшаның деректер қорымен жұмыс жасауына қажетті құралдардың жалпы құрамы 2.1 суретте көрсетілген. Бұл жалпы сұлбаға сәйкес, мынадай тізбек аламыз: қосымша —> деректер қорының ядросы —> деректер қоры. Қосымшаның құрылымында мынадай тізбек болады: визуалды емес компоненттер —> визуалды компоненттер. Визуалды емес компоненттер бағдарламашыға деректер қорының ядросымен, сонымен қатар деректердің өзімен басқару бойынша кейбір функцияларға мүмкіндік береді. Визуалды компоненттердің көмегімен мәліметтер экранда көрінеді (кестелер, тізімдер, түскен тізімдер, графиктер және т. б.). ДҚ ядросының және деректер қорының орналасқан жері бұл тізбекте көрсетілмеген.

ДҚ ядросының және деректер қорының орналасқан жері пайдаланылған архитектураға байланысты болады. Деректер қорының архитектурасының үш түрі бар:

- локальды деректер қоры және «файл-сервер» архитектурасы;
- «клиент-сервер» архитектурасы;
- көп бөлімді (үш бөлімді N-tier немесе multi-tier) архитектурасы.

Бұл архитектуралардың қолданылуы қосымша жұмысының идеологиясына, қосымшадағы бағдарламалық кодқа, қосымшаларда

қолданылатын деректер қорымен жұмыс жасауға арналған компоненттер құрамына үлкен таңба қояды (ең алдымен бұл визуалды емес компоненттерге қатысты) [4, 15].



2.1 сурет– Дайын қосымшаның деректер қорымен жұмысына арналған құралдардың жалпы құрамы

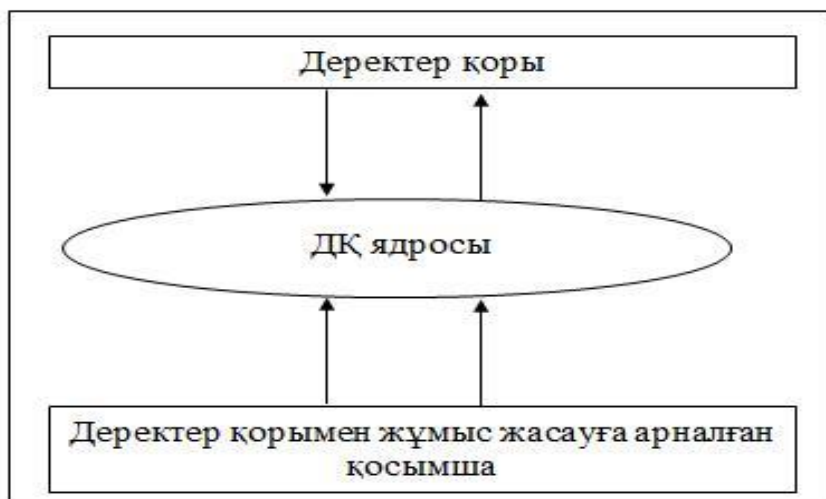
## 2.6 Локальды деректер қоры және «файл-сервер» архитектурасы

Локальды деректер қорымен жұмыс кезінде деректер қоры оған рұқсатты жүзеге асыратын қосымшалар орналасқан еомпьютерде орналасқан. Деректер қорымен жұмыс бір пайдаланушылық режимде орындалады. Деректер қорының ядросы пайдаланушы компьютерінде орналасады. Қосымша деректер қорының бүтіндігін сақтауға және деректер қорына сұраныстардың орындалуына жауапты. Бір пайдаланушылық архитектураның жалпы сұлбасы 2.2 суретте көрсетілген.

«Файл-сервер» архитектурасындағы жұмыс кезінде деректер қоры мен қосымша желінің файлдық серверінде орналасқан (мысалы, Novell NetWare). Әрбір пайдаланушы өз компьютерінен желілік серверде орналасқан қосымшаны іске қосса, бір деректер қорында көп пайдаланушылық жұмыс болуы мүмкін.

Сонда пайдаланушы компьютерінде қосымшаның көшірмесі іске қосылады. Қосымшадағы деректер қорына әрбір сұраныс бойынша ДҚ кестелеріндегі мәліметтер сұранысты орындау үшін қанша мәлімет қажет екендігіне қарамастан, пайдаланушы компьютеріне көшіріледі. Әрбір пайдаланушы деректер қорына енгізетін өзгерістер кезінде белгілі кезеңге дейін басқа пайдаланушыларға нақты деректер қорынан алынған пайдаланушы компьютеріндегі мәліметтерінің систематикалық жаңаруының тапсырмасын маңызды ететіндігі көрінбеуі мүмкін. Пайдаланушылардың бірі өзгертетін жазбалардың блоктануы басқа маңызды тапсырма болып табылады:

бұл сол уақытта басқа пайдаланушы сол мәліметтерге өзгерістер енгізбеуі үшін қажет. «Файл-сервер» архитектурасында деректер қорына сұраныстың орындалу ауыртпалығы, деректер қорының бүтіндігін басқару пайдаланушының қосымшасына жүктеледі. Сервердегі деректер қоры пассивті мәліметтер көзі болып табылады. «Файл-сервер» архитектурасының жалпы сұлбасы 2.1 кестеде көрсетілген.



2.2 сурет– Локальды деректер қорымен жұмыс кезіндегі бір пайдаланушылық архитектура

Архитектура тұрғысынан қарағанда бір пайдаланушылық архитектура мен «файл-сервер» архитектурасының арасында маңызды артықшылықтар жоқ. Екі жағдайда да ДҚБЖ ретінде Paradox, dBase және т. б. сияқты «персоналды» (немесе «локальды») ДҚБЖ пайдаланылады. Бұл жағдайда деректер қоры кестелер жиынтығын, индексті файлдарды, түсіндірме аймақтарының файлдарын және басқа да дискіде бір каталогта жеке файлдар түрінде сақталған файлдарды көрсетеді [5].

## 2.7 Жойылған деректер қоры және «клиент-сервер» архитектурасы

SQL – клиент/сервер қосымшасын жүзеге асыруға арналған табиғи құрал. Бұл рөлде SQL пайдаланушымен өзара әрекет жасайтын клиенттік жүйе мен әрбір жүйеге өз функцияларының орындалуын жинақтауға мүмкіндік бере отырып, деректер қорын басқаратын серверлік жүйе арасындағы байланыстырушы бөлімше қызметін атқарады. Сонымен қатар, SQL желілік серверге және үлкен ЭЕМ-дерде орнатылған көлемді деректер қорына қатынасына байланысты персоналды компьютерлерге клиент ретінде қызмет атқаруға мүмкіндік береді; бұл персоналды компьютерлерде жұмыс жасайтын қосымшалардағы корпоративті мәліметтерге рұқсат алуға мүмкіндік береді [7, 8].

«Файл-сервер» архитектурасы екі жағдайда тиімсіз:

Файлдық серверде орналасқан деректер қорына сұранысты орындау кезінде, шын мәнінде, пайдаланушы компьютеріндегі мәліметтердің локальды

көшірмесіне сұраныс болады. Сондықтан, сұранысты орындамас бұрын локальды көшірмедегі мәліметтер нақты деректер қорынан жаңартылады. Мәліметтер толығымен жаңартылады. Сонымен, егер деректер қорының кестесі 1000 жазбадан тұрса, ал сұранысты орындау үшін нақты 10 жазба қажет, бәрібір барлық 1000 жазба көшіріледі. Сондықтан, пайдаланушылардың және олардың сұраныстарының тым көп болғаны қажет емес.

Деректер қорының бүтіндігін қамтамасыз ету қосымшаларда жүзеге асырылады. Бұл деректер қорының физикалық және логикалық бүтіндігін бұзатын қателердің потенциалды көзі, себебі әр түрлі қосымшалар деректер қорының бүтіндігін әр түрлі өзара бірін-бірі жоққа шығаратын әдістермен бақылай жүргізуі мүмкін, немесе мұндай бақылауды мүлде жүргізбеуі мүмкін. Әр түрлі қосымшадан және әр түрлі заңмен басқаруға қарағанда деректер қорын бір жерден және бірыңғай заңмен басқару тиімдірек (бәр қосымшаның қалай жазылғанына байланысты). Сондықтан «файл-сервер» архитектурасындағы жұмыс кезінде қауіпсіздік жоғары және үнемі анықталмағандық элементі қатысады. «Файл-сервер» архитектурасындағы деректер қорымен жұмыс кезінде жасырындылық пен құпиялылық деректер қоры сақталған желілік сервер каталогына рұқсатты қамтамасыз етеді, деректер қорының кестесін кез келген әдіспен өзгертуі, оларды көшіруі, алмастыруы мүмкін [3]. «Клиент-сервер» архитектурасы пайдаланушының (клиент деп аталатын) және сервердің қосымшасының функцияларын бөліп қарастырады.

Қосымша-клиент деректер қоры орналасқан серверге сұранысты SQL сұраныстарының құрылымдық тілінде қалыптастырады. Жойылған сервер сұраныстарды қабылдайды және оны деректер қорының SQL-серверіне қайта жолдайды. SQL-сервер – бұл жойылған деректер қорын басқаратын арнайы бағдарлама. SQL-сервер сұраныстың түсіндірілуін, оның деректер қорында орындалуын, сұраныстың орындалу нәтижесін құруын және оның қосымша-клиентке берілуін қамтамасыз етеді. Сондықтан клиенттік компьютердің ресурстары сұраныстың физикалық орындалуына қатыспайды; клиенттік компьютер тек серверлік деректер қорына сұранысты жібереді және оны қажетті әдіспен түсіндірген соң және пайдаланушыға көрсеткен соң нәтижесін алады. Клиенттік қосымшаға тек сұраныстың орындалу нәтижесі жіберілгендіктен, желі бойымен тек клиентке қажетті мәліметтер ғана «саяхаттайды». Нәтижесінде желіге түсетін күш төмендейді. Сұраныстардың орындалуы деректер сақталған жерде (серверде) болатындықтан, мәліметтердің үлкен дестесін жіберудің қажеттілігі жоқ. Сонымен қатар, SQL-сервер, егер бұл мүмкін болса, алынған сұранысты аздаған шығынмен ең аз уақытта орындалатындай етіп оңтайландырады.

Мұның барлығы жүйенің жылдам әрекет етуін жоғарылатады және сұраныстың нәтижесін күту уақытын төмендетеді.

Сервермен сұраныстарды орындау кезінде мәліметтердің қауіпсіздік деңгейі жоғарылайды, себебі мәліметтер бүтіндігінің ережесі сервердегі деректер қорында анықталады және осы деректер қорын пайдаланатын барлық



қоысмшалар үшін бірыңғай болып табылады. Сондықтан, бүтіндікті қолдаудың қарама-қайшы ережелерін анықтау мүмкіндігі алынып тасталады. SQL-серверлер қолдайтын транзакцияның мықты аппараты әр түрлі пайдаланушылардың мәліметтерді бір уақытта өзгертуін алып тастауға мүмкіндік береді және деректер қорына апатты жағдайда біткен өзгерістерді енгізу кезіндегі бастапқы мәніне шегіну мүмкіндігін береді. Сондықтан, қосымша-клиенттің функциялары мыналар болып табылады:

- серверге сұраныстарды жіберу;
- серверден алынған сұраныстардың нәтижелерін түсіндіру, және оларды пайдаланушыға қажетті формада көрсету;
- пайдаланушы интерфейсін іске асыру.

SQL-сервер – желілік сервер компьютерінде орналасқан бағдарлама. SQL-сервер сұранысты клиенттен қабылдаған сәтте жүктелуі қажет. Деректер қорының функциялары мыналар болып табылады:

- қосымша-клиенттен сұранысты қабылдау, сұраныстарды түсіндіру, деректер қорында сұраныстарды орындау, сұраныстың орындалу нәтижесін қосымша-клиентке жіберу;
- деректер қорының бүтіндігін басқару, қауіпсіздік жүйесін қамтамасыз теу, қосымша-клиенттің бұрыс әрекетін блоктау; жиі пайдаланылатын сұраныстардың бизнес-ережесін түсіндірілген түрде сақтау;
- дәл сол мәліметтермен қауіпсіз және бас тартуға тұрақты көп пайдаланушылық жұмыспен бір уақытта қамтамасыз ету.

«Клиент-сервер» архитектурасында «жойылған» (немесе «өнеркәсіптік») деп аталатын ДҚБЖ пайдаланылады.

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 (корпоротивті ДҚ және дербес ДҚ) бар болуын болжайды;
- ДҚ сервер-компьютерде орналасады;
- ДДҚ корпоротивті 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:

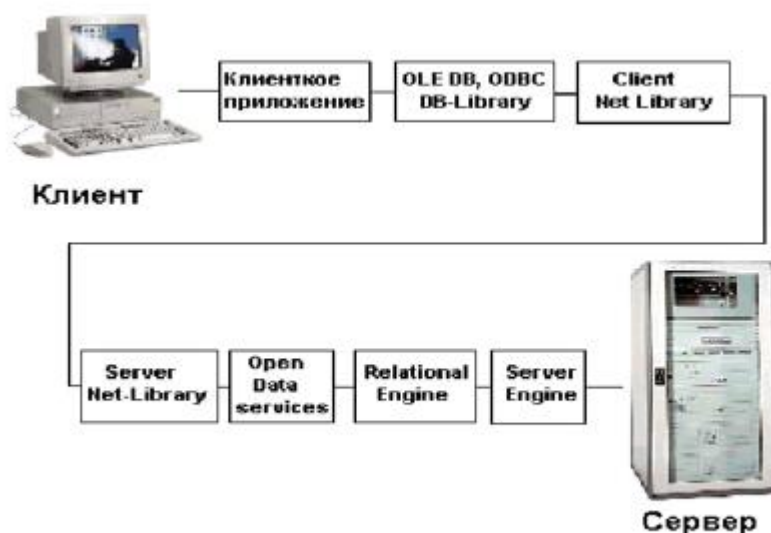
- орталықтандырылған сақтау, қызмет көрсету және жалпы корпоротивті ақпаратқа коллективті қол жеткізудің дербес ақпаратпен жеке жұмыстың сәтті қосылуы. корпоративті (серверлік) ДҚ серверінің басқаруымен

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.



2.3 сурет– Клиент-сервер архитектурасы

"Клиент-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.

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

Егер жобаланып жатқан ақпараттық жүйе (АЖ) "клиент-сервер" технологиясы бойынша құрылады деп болжанса, 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.

## **2.8 NoSQL ДҚБЖ**

Ұялылық дәуірінде, мәліметтер көздерінің саны және олардың көлемдері өсе отырып, ақпараттар бұлттық сервистер және әлеуметтік желі сайттарының аясына түсе отырып, ИТ департаменттерінің бақылауынан шыға бастағанда NoSQL класындағы ДҚБЖ ерекше маңыздылыққа ие болады.

Интернеттің біздің өміріміздің барлық саласына енуі, мәліметтер көзінің айтарлықтай мөлшерде өсуі, сақтау, өңдеу және құрылымсыз ақпаратқа операция жасауға қиындық туғызуда. Нәтижесінде коммерциялық компаниялар мен OpenSource ұйымы жаңа құралдар өндіре бастады – қосымшаларды ашуға және өндірісті оңтайландыратын және талаптары бойынша ұсынылатын, көптұтынушылық сервистерді құруға мүмкіндік беретін NoSQL жүйелері немесе « кілт-мән» қоймасы. NoSQL класты ДҚБЖ [nosql-database.org](http://nosql-database.org) ақпараттары бойынша және квазикұрылымдық және құрылымды емес өте үлкен көлемдегі мәліметтермен жұмыс жасайтын қосымшалар үшін қажет. Қазіргі таңда, әр түрлі типтегі тұтынушылардың талаптарына жауап беретін кем дегенде 150 NoSQL ДҚБЖ бар. 15 танымал мәліметтер базасының ішінен қайталап таныстырмалы іріктеме мұндай ДҚБЖ– нің негізгі артықшылықтары туралы түсінік алуға мүмкіндік береді.

## **2.9 NoSQL жүйелері**

Олардың иілгіштік деңгейінің тәуелділігіне байланысты NoSQL базаларын екі топқа бөлуге болады. Біріншісіне, кластерге жаңа түйіндерді тұтынушылық қосымшаларға қызмет көрсетуді үзбей қосуға мүмкіндік беретін, шынында да иілгіштері кіреді. Екіншісіне, кластерге жаңа түйіндерді қосу барысында айтарлықтай тұрып қалулармен сипатталатын, BigTable типті ДҚБЖ кіреді. Түйіндерді кластерге қосу немесе оларды шығару барысында мәліметтердің тұрақты қол жетімділігі мәліметтер фрагменттерінің орын ауыстыру уақытын таңдауды оңтайландыру алгоритімдерімен және маршрутизация механизмдерімен іске асырылады; мысалы, мәліметтерді қайта қосылған түйіндерге тасымалдау қажет болған жағдайда, олар бастапқы орынынан алынады, ал жаңа түйінде мәліметтердің өзекті нұсқасы пайда болған кезде маршруттаушы процесс сұраныстарды осы түйінге жібере бастайды.

### *Cassandra*

ДҚБЖ Apache Cassandra, жылдам әрекетінің құлдырауынсыз жоғары дайындығын және жақсы масштабталуын қамтамасыз етеді. Бұлтты инфроқұрылымдарда стандартты құрылғыларда жұмыс жасау барысында бас тартуға төзімділік және желілік масштабталу бұл ДҚБЖ-ін критикалық маңызды мәліметтер үшін оңтайлы платформа етеді. Cassandra бірнеше ЦОД арасында тираждауды іске асыруға мүмкіндік береді және аймақтық жүйелердің бас тартулары кезінде, мәліметтерге қол жетімділікті қамтамасыз ету құралына ие. Cassandra-да қолданылатын ColumnFamily ақпараттық моделі, қас қағым сәттік түсірілімдерді қолдайтын қуатты жүйені және дамыған қондырылған кэтеу механизімін, жоғары жылдамдықты әрекетті журналды-құрылымдық (logstructured) жазу механизімі бар, ыңғайлы бағаналы индексті ұсынады.

Cassandra ДҚБЖ-ін пайдаланатындардың арасында үлкен белсенді интернет-базаларда сұрау жүргізу үшін, Netflix, Twitter, Cisco, OpenX, CloudKick, Ooyala сияқты компаниялар жатады, ал ең үлкен Cassandra кластері қазіргі таңда, 400 түйінде 300 Тбайттан көп мәлімет сақтайды.

### *BigTable*

BigTable ДҚБЖ Google-да өндірілді және жолдық және бағаналық кілтке сәйкес жаппай байттар массивін қояды, соныменқатар уақытқа белгі қояды. ДҚБЖ құрылымы шағын, таратылған, көп шекті реттелген сәйкестік тізімін ұсынады. ДҚБЖ петабайт деңгейге дейінгі масштабтау және жаңаларды қарапайым қосуға болатын мүмкіндігі бар бірнеше машиналар арасында тарату есебімен құрылған болатын. Мәліметтер базасының көлемі берілген шекке жақындаған кезде, «таблеттер» (tablets — BigTable мәліметтер құрылымдары) BMDiff алгоритімі көмегімен және Snappy ашық коды бар копмпесся кітапханасы көмегімен (LZ77 алгоритімінің нұсқасы сығу деңгейі бойынша оңтайландырылмаған бірақ, түп нұсқасына қарағанда тез іске асыратын) сығылады. BigTable-ден анықталған жолдарды алу үшін, тұтынушы ағаштардың барлық деңгейлерімен байланысу қажет сонымен қатар, жоғары деңгейде алынған ақпараттар кэштеледі, ал осыған дейін пайдаланылған таблеттерде сақталған мәліметтерге алдағы сұраныстар тікелей ағаштың соңғы деңгейінде іске асырылады.

### *HBase*

Ашық коды бар HBase ДҚБЖ GoogleBigTable үлгісі бойынша құрылған көпнұсқалық қолдануы бар таратылған бағаналық қоймасын ұсынады. Негізінде, ол оның Hadoop таратылған файлдық жүйесінің үстіндегі шынайы уақытта құрылымдалған мәліметтер базасын іске асыратын көшірмесі. HBase, үлкен мәліметтер массивіне жазу үшін және де дәл солай оқу үшін жедел жаппай қол жеткізуді талап ететін қосымша үшін жарайды. ДҚБЖ стандартты құрылымдағы кластерлерде орналасатын миллиондаған бағаналар мен миллиардтаған жолдары бар өте үлкен кесте хостингіне есептелген. Hbase желілік және модульдік масштабтауға ие, келісілген оқу және жазу операцияларын, кестелерді автоматты және конфигурацияланатын сегменттеу

(базаларды көлденең тарауларға бөлу) және бас тарту кезінде аймақтық серверлер арасындағы жүктеме баланысын автоматы түрде келтіруді қолдайды. Одан басқа, Hadoop/MapReduce тапсырмасы үшін, шынайы уақыт сұранысын қолдау үшін Блум сүзгілері ( көптеген элементтерді шағын сақтауға мүмкіндік беретін және олардың көптеген қажеттіліктерін тексеретін мәліметтердің ықтималды құрылымы ), сондай-ақ серверлік сүзгілер көмегімен сұраныстарды қайта өңдеу және блоктық кэш, Java-тұтынушылар үшін қарапайым API базалық классы қолданады. Жүйе JRuby негізінде кеңейтілетін бұлтқа ие және JavaManagementExtensions көмегімен метрик экспотрының мүмкіндігін қолдайды.

### *MongoDB*

MongoDB – ашық коды бар NoSQL-дың құжатты бағытталған қолданылмайтын сұлбалар. Бұл MapReduce іске асырылуға ие және қиын сұраныстар тілін тұрақты сақтаумен қолдайтын бас тартуға тұрақты жоғары өнімді ДҚБЖ. MongoDB артықшылықтары:

Құжатты бағытталған сақтаудың қарапайым және қуатты қозғалтқышы (JSON форматы және динамикалық сұлбалары қолданылады);

Индексацияны толық қолдауы ( кез – келген атрибут индекстене алады);

Мәліметтердің қол жетімділігі ( жергілікті және жаһандық желілері бойынша шағылысу қолданады);

Функционалдығының жоғалтпай көлденең масштабталатын автоматты сегменттену механизмінің бар болуы.

ДҚБЖ сонымен қатар, құжаттардың күрделі ұсынысы, операцияның қақтығыссыздығын қамтамасыз ететін атомарлық модификаторлар және мәліметтерді өңдеу қолданады, сондай-ақ, стектың қиындығысыз үлкен көлемдегі файлдарды сақтауға мүмкіндік беретін GridFS жеке файлдық жүйесі қолданады.

### *Pnuts*

ДҚБЖ Pnuts - көптеген қосымшаларды бір уақытта қолдай алатын жаппай – масштабтайтын орталықтан басқарылатын мәліметтер базасы. ( суретте көрсетілген). Жүйе Yahoo компаниясында, веб – қосымшаларға мәліметтерді жеткізу үшін қолданылады. Pnuts мәліметтерді басқаруды сервис түрінде іске асырады, ол бағдарламалаушыларға мәліметтерді басқару үшін жеке масштабтау шешімін құру қажет болмағандықтан, қосымшаны өндіру уақытын айтарлықтай қысқартады.

Көптеген қосымшалардың бір сервиске қосылуы операциялық шығындарды жаңартуға және мәліметтерді басқарудың жалпы оңтайлы шешімін қолдануға мүмкіндік береді. Одан басқа, ортақ сервис қорларды резервте ( серверлер, жинақтаушылар және т.б )ұстауға және олардың қатты қажетірек болған жағдайында олардың қосымшаларын жылдам белгілеуге мүмкіндік береді.



2.4 сурет– Pnuts ДҚБЖ-де мәліметтерді сақтау архитектурасы

### *Hypertable*

Hypertable – сенімділікті, масштабтылықты және максимал жылдамдықты қажет ететін қосымша үшін өндірілген ашық коды бар жоғары жылдамдықты таратылған бұлттық NoSQL ДҚМЖ. Hypertable, ағымдағы уақытта онлайн – қосымшалар үшін мәліметтер ағынын өңдейтін ұйымдар үшін пайдалы. BigTable бейнесі бойынша құрылған Hypertable, стандартты серверлердегі таратылған кластерлерде орындалу үшін есептелген және жинақтаушының және компьютердің бас тартуына тұрақтылығымен сипатталады.

### *CouchDB*

Apache Couch DB – параллельді таратылған жүйе құру үшін оңтайландырылған, Erlang функционалды бағдарламалау тілінде жазылған құжатталған база. Couch DB ДҚБЖ MapReduce тапсырмаларын бағдарламалау үшін JavaScript қолдану мүмкіндігін ұсынады. Қақтығыстарды шешу және екі бағыттағы оқуы бар инкременталды тираждауды қолдайды. Couch DB, JSON форматында нәтижелерді беруі бар REST-API-ді ұсынады, сондай-ақ бағдарламалау интерфейсіне жүгінуге HTTP-сұранысты қолдайтын кез келген жүйеден болады. Бағдарламалау тілін таңдауды жеңілдететін көптеген орнатылған тұтынушылық кітапханалар бар. Орнатылған веб-консоль әкімшілік базамен HTTP-сұраныс көмегімен тікелей сөйлеседі.

### *Voldemort*

Voldemort ДҚБЖ– бұл таратылған қойма «кілт – мән ». оның әрбір түйіні тәуелсіз және де координацияның орталық пункті және бас тартудың бір келкі нүктесі жоқ. Жүйе, кәштеуші қабатта қажеттілікті болдырмауға мүмкіндік беретін, жеткілікті түрде жылдам әрі қарапайым қойма ретінде құрылған. Voldemort архитектурасы put, get және delete операцияларынан тұратын күрделі емес интерфейсті бірге іске асыратын бірнеше деңгейден тұрады. Әрбір деңгей белгілі бір функцияға жауап береді – мысалы, TCP/TP бойынша байланыс, маршрутизацияны немесе қатығыстарды шешу.

Voldemort-ты қолдану барысында серверлерде тек қана барлық мәліметтердің бөлігі ғана қалатындай етіп, мәліметтер бірнеше серверге автоматты түрде тираждалады және секцияланады. Серверлердің бас тартуы тұтынушы үшін мөлдір түрде өңделеді. Қосылатын мәліметтер сериализаторы қарастырылған, соның арқасында қиын кілттер мен мәндерді қолдануға болады мысалы: аталған жолдары бар кортеждермен тізбектер. Сериализация кітапханасын қолдайтындар санына ProtocolBuffers, Thrift, Avro және Java Serialization кіреді.

Жүйенің басқа да артықшылықтары:

Бас тартулар тұтастығын жүйенің дайындығын жоғалтусыз сақтап қалу үшін мәліметтердің элементтерінің нұсқалылығын қолдау;

Бір түйіннің, компьютерге, желіге, дискалық жүйешеге және тираждау операциясын орындауын жүктелуіне байланысты секундына 10-20 мың операцияны орындау қабілеті

Мәліметтердің орын ауыстыру стратегиясын өзгерту мүмкіндігі – мысалы, географиялық өзара бір-бірінен алыстатылған ЦОД тарату мүмкіндігі.

Voldemort, жоғары масштабты мәліметтерді сақтау тапсырмасын қолдау үшін LinkedIn желісінде қолданылады. Әзірге жүйе өнделуде сондықтан арасында қателіктер туралы хабарлаМалар болып тұрады. ДҚБЖ–нің бастапқы коды Apache 2.0. лицензиясы бойынша қол жетімді.

#### *Infinispan*

Infinispan – Java-да жазылған ашық коды бар жадыдағы платформада жұмыс жасайтын. ДҚБЖ масштабтылығымен, дайындылығымен ерекшеленеді және замануи микропроцессорлық/көп ядролық архитектураларда орындау үшін оңтайландырылған « кілт-мән» қоймасын ұсынады. ДҚБЖ java.util.Map кеңейтетін кэштеудің жеке интерфейсін ұсынады және жоғары әсерлі жүйе таратуды қамтамасыз ететін бір рангтық желілік архитектураны қолдануға мүмкіндік береді.

#### *Dynomite*

Dynomite ДҚБЖ жоғарғы дайындылық пен масштабтауды қамтамасыз ететін « кілт-мән» қарапайым моделін өндірушілерге ұсына отырып, мәліметтерді таратумен сақтауды іске асырады. Dynomite-тің тарату және масштабтау мүмкіндіктерін мәліметтер моделін оңай вариациялай отырып, әр түрлі үйлесімділікте комбинациялауға болады. Dynomite Erlang тілінде өндірілуде және AmazonDynamo ДҚБЖ құрылымында негізделген келісіліп таратылған NoSQL қоймасын ұсынады. Жүйе векторлық сағат, Меркл ағашы және келісілген хэштау алгоритмдерін іске асырады. Басқа артақшылықтарына: кворум әдісі бойынша шешім қабылдау (tunablequorumsensing), сақтау қозғалтқыштарына қосылатын тараулар синхронизациясына және түйіндерге қажеттіліктерді анықтау үшін «сплетен» (gossiping) протоколын қолдану, (thrift) жылдам қол жеткізу интерфейсі және мәліметтерді үлестірмелік көрсетуді қолдауымен веб-консоль.



### *Redis*

Redis (RemoteDictionaryServer) – барлық кілттер жедел жадыда сақталатын, ашық кодты «кілт-мән» мәліметтер базасын қолдануға оңай. Кілттер жолдарды, хэштарды, тізімдерді, көптіктерді және реттелген көптіктерді құрау мүмкін болғандықтан, бұл ДҚБЖ-н мәліметтер құрылымының сервері деп атайды. Redis түйіндері арасындағы мәліметтермен алмасу протоколы өткізу қабілеті бойынша екілік. Мән базада 512 Мб-тан көп орын ала алмайды, ал базаның өзі жедел жүйенің виртуалды жадысын қолдамайды.

### *Xeround*

Xeround ДҚБЖ-нде, солтүстік Америкада және Европада ЦОД AmazonWebServices-те іске асырылатын хостинг, MySQL негізіндегі мәліметтер базасының ыңғайлы жеке сервисі ұсынылды. Xeround хостинг-сервиске бекітілмеген сондықтан, оны бұлт операторлары арасында тасымалдауға болатын болды. Екі деңгейлік архитектура қол жеткізу түйінінен және мәліметтер түйінінен тұрады. Қол жеткізу түйіндері қосымшалардан сұраныс алады, мәліметтер түйіндерімен байланысады, есептеуді іске асырады да нәтижесін береді, ал мәліметтер түйіндері ақпартты сақтайды. Xeround мәліметтерді төменгі деңгейдегі аппараттық инфроқұрылымға бекітілмеген виртуалды тарауларда сақтады. Әрбір тару бөлек серверлерде орналасқан әртүрлі мәліметтер түйініне тираждалды, осының есебінен жоғары дайындық және толық бас тартуға тұрақтылық қамтамасыз етілді. XeroundBasic нұсқасы 500 Мбайтқа дейінгі көлемдегі бір ауқытта 40-қа дейін бір уақыттағы байланысты қолдайды (Pro нұсқасы – 4,8 мыңға дейін байланыс) және 8Мбайт/с-қа дейінгі максималды өткізу қабілетін қамтамасыз етеді. Өндірушілер бұлттық провайдерлерге өз сервисін ұсынуды жоспарлады, қоса алғанда GoGrid және Rackspace бар еді. Бірақ 2013 жылы бұл ДҚБЖ-нің дамуы тоқтады.

### *SimpleDB*

AmazonWebServices типтік қолдануға есептелген жеке SimpleDB мәліметтер базасының бұлттық сервисімен орналастырады және кішкентай көлем кезінде, қолдану тегін. SimpleDB бастапқы коды Erlang-та жазылған соның есебінен архитектураның ыңғайлылығы, кеңейтілу және масштабтаудың қарапайымдылығы қамтамасыз етіледі.

### *Dynamo*

AmazonDynamo ДҚБЖ жылдам әрекетімен, экономикалық тиімділігімен және үйлесімділігімен келіспеушіліктер арасындағы қатаң бақылауды қажет ететін және сенімділігіне жоғары талапты ұсынатын сервистердің жұмысқа қабілеттілігін қолдауға көмектеседі. ДҚБЖ тек біріншілік кілтінде ғана негізделген қарапайым қол жеткізу интерфейсін

ұсынады. Дайындылық пен масштабталуды қамтамасыз ету үшін тексерілген технологиялардың үйлесімділігі қолданылады: мәліметтер абстрактталады және сәйкесінше объекттердің көпнұсқалылығын қолданылуымен тираждалады. Орталықтандырылмаған үйлесімділік көшірмелері жаңару кезінде, кворум принципіне негізделетін синхронизация протоколы есебінен қол жеткізіледі. Орталықтандырылмаған Dynamo құрылымы әкімшілік шығындарды минимумға жеткізуге мүмкіндік береді – сақтау түйіндері, әкімшіліктің араласуын қажет етпей-ақ автоматты түрде алынып, қосылады.

Amazon электронды комерциясының платформасының бір қатар кілттік сервистері үшін Dynamo базалық болып табылады. Жүйе шектік жүктеме уақытында сөнбестен масштабтала алады, мысалы, мейрам алды қарқынды сауда кезінде; әдетте, сатып алушының себетіне қызмет көрсетуші сервис, тәулігіне 3 миллионнан аса сатып алуды қолдауды қамтамасыз ете отырып, миллиондаған сұраныстарға қызмет көрсетуге қабілетті, ал сеанс жағдайын сақтаушы сервис бір уақытта мыңдаған белсенді сеанстарға қызмет көрсетеді.

#### *ClearDB*

ClearDB ДҚБЖ MySQL негізіндегі реляциялық мәліметтер базасында құрылған және жоғары дайындығымен, апатқа тұрақтылығымен және жылдам әрект етушілігімен ерекшеленеді. Жүйе желі және жинақтаушы бас тартқан кезде жоғары дайындықты қамтамасыз ете отырып, оқу/жазу операциясының мультиаймақтық шағылысуын қолдайды. Жүйе, бұлтқа қатысы бойынша әмбебап болып келеді – ол гибридітік конфигурацияларда таратылған ЦОД-тарда және бірнеше әр түрлі бұлттарды қамти алады.

#### Мүмкіндіктерін салыстыру

Айтылған ДҚБЖ-н негізгі екі сипаттамасы бойынша салыстырамыз.

BigTable ДҚБЖ және Cassandra және Hbase сияқты жүйелер жаңарту кезінде дискіде қайта жазуды болдырмау үшін енгізу-шығаруды тізбектей іске асырады. Жаңарту алдымен буфер жадына жазылады да, содан кейін толығымен тізбектей дискіде сақталады. Бір жазбаның бірнеше жаңаруы әр түрлі уақытта дискінің әр түрлі аймақтарына шашылып тасталуы мүмкін. Нәтижесінде, т жазбаны оқу үшін, BigTable типті NoSQL базалары әр түрлі жаңаруларды біріктіру және алып тастау үшін көптеген енгізк-шығару операцияларын іске асыруы қажет. Жазу тізбектей іске асырылатын болғандықтан, оқу операциясы керісінше оңтайландырылмағанда ол жылдам жүреді.

Дәстүрлі пул буферлер сұлбасы бойынша орындалған Pnuts архитектурасы олардың жаңарылулары кезінде қайта жазылады. Жаңару енгізу-шығарудың жаппай операциясын талап ететін болғандықтан, ол BigTable және де сол сияқты жүйелерге қарағанда бірнеше баяу орындалады, бірақ бір операцияда ең жаңа жазбаны толығымен алып тастауға болатындықтан мәліметтерді оқу жылдам жүреді.

HBase-де операцияның алдын алу журналын жүргізуді сөндіру мүмкіндігі бар, ол мәліметтерді жаңарту барысында кедергілерді төмендетуге мүмкіндік береді. Бұл мүмкіндік HBase-тің мақсатымен үйлеседі – олардың

тұтастығына кепілдік бермей мәліметтердің дестелік анализін орындаумен. Осындай жүйелерде мәліметтерді жоғары оқу жылдамдығы жолында сенімділік құрбан болады.

Pnuts-та операция журналы әдеткі қалпы бойынша, әр қашан дискіде жүргізіледі.

Cassandra және Pnuts ассинхронды тираждауды қолдайды, яғни базаның жаңару процедурасы үшін, айтарлықтай кедергілерсіз үлкен қашықтықтарға тираждау. Бұл жүйелер жаппай режимде жазба орындауға мүмкіндік береді, ал операциялардың қақтығыстары олардың орындалуынан кейін рұқсат етіледі. HBase-де және Cassandra-да көшірмелердің жаңалылығын қамтамасыз ету үшін синхронды тираждау қолданылады.

Баған бойынша сақтау, әрбір сұраныста танымал көптік бағанаға қол жеткісуді іске асыру қажет қосымшаларға басымдылық береді. Мұндай әрбір топ физикалық түрде басқаларынан бөлек сақталады. Бірақ егер сұраныстарда көбінесе мәліметтердің толық жолы немесе олардың жаппай көптігі қажет, ондай жағдайда әрбір жол физикалық түрде жадының аймағын алатын бөлу сұлбасын қолданған жақсы. Pnuts-та бұл тұтас жолды сақтау есебінен қол жеткізіледі, ал Cassandra-да және басқа да бағаналық қоймаларда – жалғыз топты/бағаналар отбасын қолдану жолымен.

Қысқасын айтқанда, get, put және delete функциялары Hypertable және Voldemort сияқты «кілт-мән» жүйелерінде жақсырақ қолданады, мәліметтер агрегациясын NoSQL жүйесінің бағанасының көмегімен орындау оңайырақ (мысалы, Cassandra немесе BigTable). Өз кезегінде мәліметтерді қою үшін MongoDB сияқты NoSQL-дың құжаттық базасы көбірек сәйкес келеді.

## **2.10 Биллинг принциптері**

Cassandra ДҚБЖ әрбір шақыру кезінде тұтынушыға ұзақ мерзімді сақтаудың энерго тәуелді аймағында жазуды іске асырудың қажеттілігін айқын көрсетуге рұқсат етеді, ал AmazonDynamoDB тігіссіз масштабтауды іске асыру мүмкіндігін береді. Жаңа кестені құрғаннан кейін жүйе сұранысқа қызмет көрсету құралдарының жұмысн үзбестен және өнімділігін төмендетпестен масштабтауға мүмкіндік береді. AmazonElasticComputeCloud сервисі тұтынушыларға әкімшілік артықшылықтары бар қол жеткізулер береді, бірақ оны қолданғаны үшін есептеу болмаған жағдайдың өзіндеде ақы төлеуді талап етеді.

AmazonSimpleDB тұтынушылары әр айдың бастапқы машиналы 25 сағатына және алғашқы гигабайттық сақтау кеңістігіне ақы төлемейді. AmazonDynamoDB тұтынушылары алғашқы 100 Мбайт сақтауға ақы төлемейді, сол сияқты егер жүгіну жиілігі секундына оқу операциясы оннан және жазу операциясы бестен аспаса ақы төленбейді.

MongoDB тарифтері: базалық жазылу 2500 долл. Стандартты серверге жазылу – 5000 долл. Копоротивтік -7,500 долл.

Жедел жады – жылдам әрекетті жоғарылату үшін қарқынды түрде қолданылатын басты ресурстардың бірі. Көптеген серверлердің қатысы бар

NoSQL жүйелері көлденең масштабтаудың қарапайымдылығын қамтамасыз етеді сонымен қатар қайтлаудың есебінен бас тартуға төзімділікті және жоғары дайындықты қамтамасыз етеді. NoSQL ДҚБЖ көбісі ашық кодтарда өндірілетін болғандықтан ақша үнемдеуге мүмкіндік береді. Негізінді тәжірибеиелік түрде аталған базалардың барлығы шектелген функцияналдылығы бар тегін нұсқаға немесе ашық кодтарға ие.

NoSQL базаларын енгізуді жоспарлайтын ұйымдарда, бағалардан бөлек, ұзақ мерзімді жоспарларын және сақтау кеңістігінің талаптарын ескерген жөн. NoSQL-ға өтудің пайдалы дәлелдері келесідей: құрылымсыз, таратылған мәліметтердің өсіп жатқан және үлкен көлемдерін өңдеу мүмкіндігі; қайталау алдындағы жылдам әрекет пен дайындықты жөн көруі.

Тағы да маңызды дәлелдерінің бірі – экономикалық тиімділік, экспоненциалды өсіп жатқан транзакция және мәліметтер көлемдерімен басқару үшін қолданылатын арзан стандартты кластерлердің шартты қолданылуымен.

Кестеде талқыланған ДҚБЖ-нің негізгі сипаттамалары келтірілген. Сол немесе басқа ДҚБЖ-н таңдау барысында екі негізгі факторды ескерген жөн:

Оқу және жазу операцияларын орындауға арналған оңтайландырылған жүйелерде олардың орындалуын салыстыру жолымен жұмыс тапсырмаларын талап ету;

Кішкене кедергімен сенімділік арасындағы қақтығыс (мысалы, егер өндірушілер жазу операциясының бөлігінің жоғалуы мүмкін болса, онда дискіде синхронизациясыз тізбектей жазуды қолдануға болады).

ДҚБЖ ұялы құрылғылардың таралу және мәліметтер көзінің көлемінің қарқынды өсу дәуірінде ерекше маңыздылыққа ие болды. Мәліметтер ИТ департаменттерінің бақылауынан шығып, бұлттық сервистердің, ұялы құрылғылармен әлеуметтік желі сайттарының билігіне өтті. Бұлттардың таралуы мен дамуы сенімді бұлттық мәліметтер базасының қажеттілігін туғызады. Мақалада ұсынылған ДҚБЖ-нің ішінде ең үлкен танымалдылыққа, NoSQL әлеміндегі ең әдетткі өкілдері деп айтуға болатын, Cassandra, HBase и MongoDB ие болды. ACID (Atomicity — «атомарлық», Consistency — «үйлесімділік», Isolation — «оқшаулау», Durability — «сақтаудың ұзақ мерзімділігі») дәстүрлі принциптерімен бірге көптеген бұлттық базалар BASE (Basically Available, Soft State, Eventually consistent — базалық қол жетімділік, тұрақсыз жағдай, нәтижесіндегі үйлесімділік) принципін қолдайтындығын айта кеткен жөн.

## 2.1 кесте– NoSQL жүйесіндегі ДҚБЖ-нің негізгі артықшылықтары

ДҚБЖ	Сақтау түрі/ платформа	Лицензия түрі	Бағдарламалау тілі
Cassandra	Бағандық	Ашық код	Java

## 2.1 кестенің жалғасы

ДҚБЖ	Сақтау түрі/ платформа	Лицензия түрі	Бағдарламалау тілі
BigTable	Бағандық	Жекеменшік	Си
HBase	Бағандық	Ашық код	Java
MongoDB	Ақпараттық	Ашық код/GPL	C++
Pnuts	Бағандық	Жекеменшік	Java
Hypertable	Кілт-мән	GPL/ Ашық код	C++
CouchDB	Ақпараттық	Ашық код	Erlang
Voldemort	Кілт-мән	Ашық код	Java
Infinispan	Data Grid	Ашық код	Java
Dynomite	Кілт-мән	Ашық код	Erlang
Redis	Кілт- мән/кортеж	Ашық код	Си
Xeround	MySQL негізінде	GPL/ Ашық код	Си/C++
SimpleDB	Ақпараттық	Жекеменшік	Erlang
Dynamo	Кілт-мән	Ашық код	Erlang
ClearDB	MySQL негізінде	Ашық код	Си

## 3 Дерекқорды жобалау және оны іске асыру

### 3.1 Қолдану облысы және тағайындалуы

Бұл жүйе автоматтандырылған қызмет көрсету орны болып табылады. Яғни жеке кәсіпкерлікте атқарылып жатқан жұмыстар туралы ақпарат алу, оңтайлысын таңдау және қажет болған жағдайда ақпарат енгізу. Бұл жүйе келесідей мәліметтерді жүйелеп, қолданушының сұрақтарына жауап беруі қажет:

– кәсіпкерлік жұмысы жайлы мәліметтер;

– жылдық жоспарлық есебін анықтау.

### **3.2 Дерекқордағы шешілетін мәселелер**

Қолданушылардың тиімді және ыңғайлы қызмет түрін таңдауға көмек беретін орталықтандырылған деректер қорларын тиімді басқаратын жүйесін құру, қызметтер туралы қажетті ақпаратты құнын есептеу арқылы салыстыру, Excel бағдарламасы арқылы гистограммамен көрнекі түрде көрсету.

### **3.3 Концептуалды жобалау**

Қажетті ақпарат: пайдаланушылардың талабы

Жоба пайдаланушының барлық негізгі талаптарын қанағаттандыру керек. Бұл жүйенің негізгі пайдаланушылары кез келген қарапайым қолданушы. 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.

Дерекқор және қосымшаны жобалау құралдарын таңдаудың негіздемесі.

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 [2]:

– 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 төменөндіргіштік 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 программалау тілі. Дерекқорға қатынас құру үшін бағдарлаушылар өздерінің бағдарламаларына SQL командаларын қояды. 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.

### **3.4 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. Электрондық веб-журналдың дерекқоры 3.1-кестеде көрсетілген нысандарды қолданады.



### 3.1 к е с т е– «Жеке кәсіпкерлік» дерекқорының нысандары

Нысан аты	Сипаттамасы
Gostinica	Қонақ үй туралы мәліметтер
Kassakafe	Кафе кассасы туралы мәліметтер
Obslsto	ТЖО қызметтері туралы мәліметтер
Prodbenz	Жанармай сату туралы мәліметтер
Spisokdol	Қызметтер тізімі
Sotrudniki	Қызметкерлер туралы мәліметтер
Vihodnarab	Жұмысқа шығу туралы мәліметтер
Matersto	ТЖО материалдары туралы мәліметтер

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.

### 3.5 Логикалық жобалау

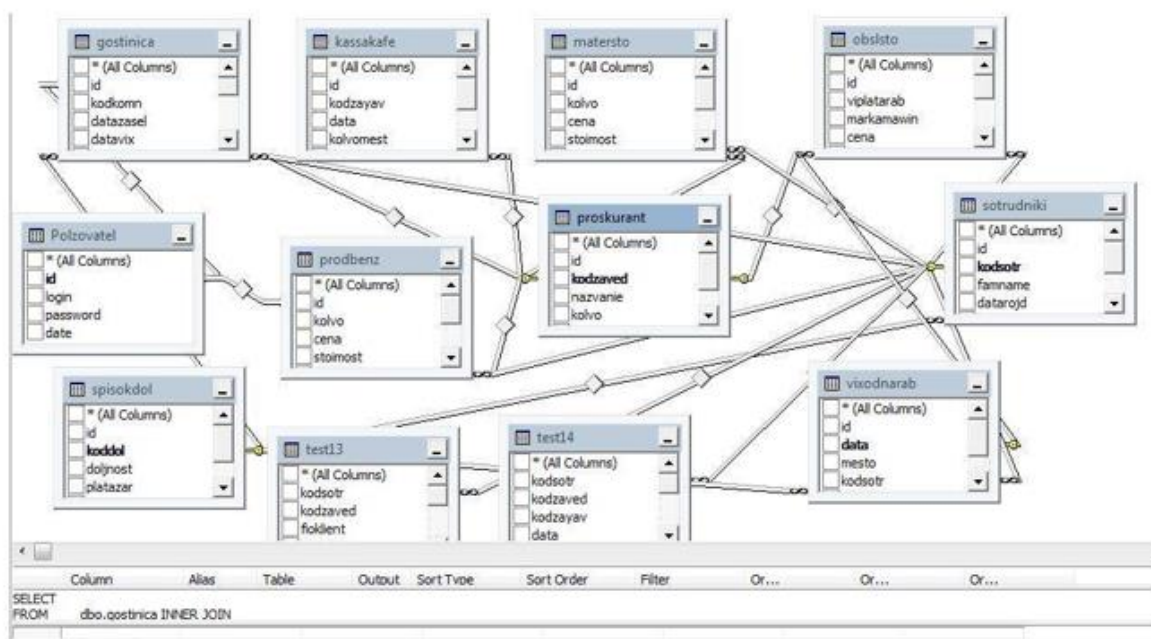
Логикалық жобалау кезеңінде дерекқор моделі жүйені іске асыру кезінде қолданылатын белгілі бір ДҚБЖ (NoSQL Server) үшін жарамды форматқа трансформаланады. Логикалық жобалаудың мақсаты реляционды кестелік құрылымдарды құру болып табылады. Жобаланатын дерекқорда кестелердің және байланыстардың құрылымдарын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

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 (NoSQL Server) қосу. Кестелерді құруға арналған скрипт В қосымшасында келтірілген.

ДҚ диаграммасы.

Концептуалды моделді дерекқордың сәйкес құрылымына енгізілу тәсілі қандай болса да, ДҚ реляционды сұлбасы жобаға сәйкес болу керек.



3.1 сурет– ДҚ диаграммасы

### 3.6 Физикалық жобалау

Тапсырманың қойылуы.

Мәліметтер қорымен жұмыс істеу қосымшасын жасау әдістемесі кәдімгі программаны құру әдістемесінен ешқандай айырмашылығы жоқ. Формаға керекті компоненттер енгізіледі, компоненттердің қасиеттерінің мәндері қойылады, керекті оқиғаны өңдеу процедурасы жасалады.

Мәліметтер қорымен жұмыс істеу қосымшасы мәліметтерге енуді қамтамасыз ететін компоненттерден, өрістердің мәліметтерін қарап шығу және редакциялау мүмкіндіктерінен тұрады. Мәліметтерге кіруді қамтамасыз ететін компоненттер Data Access компоненттер палитрасы қосымшасында, ал мәліметтерді бейнелеу компоненттері - Data Controls қосымшасында болады [1,5,9].

Database компоненті мәліметтер қорын кестелер жиынтығы ретінде, ал Table компоненті – мәліметтер қорының бір кестесі. DataSource (мәліметтер көзі) компоненті мәліметтерді бейнелеу-редакциялау компоненті (мысалы DBGrid компонентін) мен мәліметтер көзі (Table компоненті немесе SQL-сұраныс нәтижесі) арасындағы байланысты қамтамасыз етеді. DataSource компоненті мәліметтер көзін тез таңдауды, сол компоненттердің біреуін пайдаланып, мысалы DBGrid кестеден мәліметтерді көру үшін немесе осы кестеге SQL-сұранысты орындау нәтижесін көруге мүмкіндік береді [2,6].

### 3.7 Бағдарламаны сипаттау

Жоғарыда айтылғандай, жобаның интерфейсі Visual Studio 2010 объектілі-нысандалған бағдарламалау ортасында орындалды. Қосымшаның бағдарламалық коды төменде көрсетілген.

## Visual C# коды

```
using System; using
System.Data;
using System.Collections;

using System.Configuration; using
System.Web;
using System.Web.Security; using
System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;

using System.Collections.Generic; using
System.Linq;

using System.IO; using
System.Drawing;

namespace diplomka
{
    public partial class WebForm2 : System.Web.UI.Page
    {
        string conn = ConfigurationManager.ConnectionStrings["connectionDB"].ToString();
        protected void Page_Load(object sender, EventArgs e)
        {
            if (Session["db_kodzaved"] == null && Session["db_id"] == null &&
Session["db_Plzovotel_kodzaved"] == null)
            {
                Response.Redirect("~/index.aspx");
            }
            else
            {
                if (!Page.IsPostBack)
                {
                    SqlConnection con_sorting = new SqlConnection(conn);
                    string com_sorting = "select * from sotrudniki where db_id = " +
Session["db_id"].ToString();
                    SqlDataAdapter adpt_sorting = new SqlDataAdapter(com_sorting, con_sorting);
                    DataTable dt_sorting = new DataTable();
                    adpt_sorting.Fill(dt_sorting);
                    DropDownList1.DataSource = dt_sorting;
                    DropDownList1.DataBind();
                    DropDownList1.DataTextField = "kodsotr";
                    DropDownList1.DataValueField = "kodsotr";
                    DropDownList1.DataBind();
                    DropDownList1.Items.Insert(0, new ListItem("kodsotr", "kodsotr"));
                    SqlConnection con_form = new SqlConnection(conn); string com_form
                    = "select * from test14 where db_id = " +
Session["db_id"].ToString();
                    SqlDataAdapter adpt_form = new SqlDataAdapter(com_form, con_form); DataTable
```

```

        dt_form = new DataTable();
        adpt_form.Fill(dt_form); DropDownList3.DataSource =
        dt_form; DropDownList3.DataBind();
        DropDownList3.DataTextField = "kolvomest";
        DropDownList3.DataValueField = "kolvomest";
        DropDownList3.DataBind();
        DropDownList3.Items.Insert(0, new ListItem("kolvomest", "kolvomest"));

        SqlConnection con_grup = new SqlConnection(conn);
        string com_grup = "select * from kassakafe where db_id = " +
Session["db_id"].ToString();
        SqlDataAdapter adpt_grup = new SqlDataAdapter(com_grup, con_grup); DataTable
        dt_grup = new DataTable();
        adpt_grup.Fill(dt_grup); DropDownList4.DataSource =
        dt_grup; DropDownList4.DataBind();
        DropDownList4.DataTextField = "kodzayav";
        DropDownList4.DataValueField = "kodzayav";
        DropDownList4.DataBind();
        DropDownList4.Items.Insert(0, new ListItem("kodzayav", "kodzayav"));
        FillCustomerInGrid();
    private void FillCustomerInGrid()
    {
        DataTable dtCustomer = Fetch();

        GridView_1.DataSource = dtCustomer;
        GridView_1.DataBind();

        GridView1.DataSource = dtCustomer;
        GridView1.DataBind();
        if (dtCustomer.Rows.Count > 0)
        {
            GridView_1.DataSource = dtCustomer;
            GridView_1.DataBind();
            sortirovka();
        }
        else
        {
            SqlConnection con_sem_1 = new SqlConnection(conn);
            con_sem_1.Open();
            SqlDataAdapter da_sem_1 = new SqlDataAdapter(@"SELECT '' as [kodsotr] , '' as
[kodzayav]
, '' as [kodzaved] , '' as
[data]
, '' as [Семестр] , '' as
[kolvomest]
, '' as [первое_блюдо] , '' as
[второе_блюдо]
, '' as [салат] , ''
as [сусын] , '' as
[Всего] , '' as [ID]
, '' as [db_id]", con_sem_1);

```

```

        DataSet ds_sem_1 = new DataSet();
        da_sem_1.Fill(ds_sem_1);
        GridView_1.DataSource = ds_sem_1;
        GridView_1.DataBind();
    }
}
protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
{
    sortirovka();
}

protected void DropDownList2_SelectedIndexChanged(object sender, EventArgs e)
{
    sortirovka();
}
protected void DropDownList3_SelectedIndexChanged(object sender, EventArgs e)
{
    sortirovka();
}
protected void DropDownList4_SelectedIndexChanged(object sender, EventArgs e)
{
    sortirovka();
}
public void sortirovka()
{
    string sum_1_2 = ",sum(ISNULL(первое_блюдо,0)+ ISNULL([второе_блюдо],0)
+ISNULL([салат],0)+ ISNULL([сусын],0)) as [Всего]";
    string sql0 = "SELECT * ,ISNULL(первое_блюдо,0)+ ISNULL([второе_блюдо],0)
+ISNULL([салат],0)+ ISNULL([сусын],0) as [Всего],ID From test14 where db_id=" +
Session["db_id"].ToString() + " and [kodsotr]=";
    string sql = null, sql1 = null, sql2 = null, sql3 = null;
    string sem_1 = null;
    string sem_2 = null;
    string sem_3 = null;
    if (DropDownList1.SelectedValue.ToString() != "kodsotr")
    {
        sql = sql0 + " '" + DropDownList1.SelectedValue.ToString() + "' ";
        sem_1 = " SELECT 'Базалык' as ' ',sum([первое_блюдо]) as 'первое_блюдо'
,sum([второе_блюдо]) as 'второе_блюдо' ,sum([салат])as 'салат' ,sum([сусын])as 'сусын'" +
sum_1_2 + " From test14 where Семестр=1 and [kodsotr]='" +
DropDownList1.SelectedValue.ToString() + "' and db_id = " + Session["db_id"].ToString();
        sem_2 = " SELECT 'VIP стол' as ' ',sum([первое_блюдо]) as 'первое_блюдо'
,sum([второе_блюдо]) as 'второе_блюдо' ,sum([салат])as 'салат' ,sum([сусын])as 'сусын'" +
sum_1_2 + " From test14 where Семестр=2 and [kodsotr]='" +
DropDownList1.SelectedValue.ToString() + "' and db_id = " + Session["db_id"].ToString();
        sem_3 = " SELECT 'Барлығы' as ' ',sum([первое_блюдо]) as 'первое_блюдо'
,sum([второе_блюдо]) as 'второе_блюдо' ,sum([салат])as 'салат' ,sum([сусын])as 'сусын'" +
sum_1_2 + " From test14 where [kodsotr]='" + DropDownList1.SelectedValue.ToString() + "' and
db_id = " + Session["db_id"].ToString();
    }
}

```

```

if (DropDownList1.SelectedValue.ToString() == "kodsotr")
{
    sql = sql0 + " [kodsotr] ";
    sem_1 = " SELECT 'Базалык' as ' ',sum([первое_блюдо]) as 'первое_блюдо'
,sum([второе_блюдо]) as 'второе_блюдо' ,sum([салат])as 'салат' ,sum([сусын])as
'сусын'" + sum_1_2 + "      From test14 where Семестр=1 and db_id = " +
Session["db_id"].ToString();
    sem_2 = " SELECT 'VIP стол' as ' ',sum([первое_блюдо]) as 'первое_блюдо'
,sum([второе_блюдо]) as 'второе_блюдо' ,sum([салат])as 'салат' ,sum([сусын])as
'сусын'" + sum_1_2 + "      From test14 where Семестр=2 and db_id = " +
Session["db_id"].ToString();
    sem_3 = " SELECT 'Барлыгы' as ' ',sum([первое_блюдо]) as 'первое_блюдо'
,sum([второе_блюдо]) as 'второе_блюдо' ,sum([салат])as 'салат' ,sum([сусын])as 'сусын'" +
sum_1_2 + " From test14 where id=id and db_id = " + Session["db_id"].ToString();
}
if (DropDownList2.SelectedValue.ToString() != "Тапсырыс")
    sql1 = " and [Семестр]=" + DropDownList2.SelectedValue.ToString(); if
(DropDownList2.SelectedValue.ToString() == "Тапсырыс")
    sql1 = " and [Семестр]=[Семестр]";
if (DropDownList3.SelectedValue.ToString() != "kolvomest")
    sql2 = " and [kolvomest]=" + DropDownList3.SelectedValue.ToString() + """; if
(DropDownList3.SelectedValue.ToString() == "kolvomest")
    sql2 = " and [kolvomest]=[kolvomest]";
if (DropDownList4.SelectedValue.ToString() != "kodzayav")
    sql3 = " and [kodzayav]=" + DropDownList4.SelectedValue.ToString() + """; if
(DropDownList4.SelectedValue.ToString() == "kodzayav")
    sql3 = " and [kodzayav]=[kodzayav]";
SqlConnection con = new SqlConnection(conn);
con.Open();
SqlDataAdapter da = new SqlDataAdapter(sql + sql1 + sql2 + sql3, con);
DataSet ds = new DataSet();
da.Fill(ds); GridView_1.DataSource
= ds; GridView_1.DataBind();
GridView1.DataSource = ds;
GridView1.DataBind();
con.Close();
SqlConnection con_sem_1 = new SqlConnection(conn);
con_sem_1.Open();
SqlDataAdapter da_sem_1 = new SqlDataAdapter(sem_1 + sql1 + sql2 + sql3 + "
UNION ALL " + sem_2 + sql1 + sql2 + sql3 + " UNION ALL " + sem_3 + sql1 + sql2 + sql3,
con_sem_1);
DataSet ds_sem_1 = new DataSet();
da_sem_1.Fill(ds_sem_1);
GridView2.DataSource = ds_sem_1;
GridView2.DataBind();
con_sem_1.Close();
}
protected void GridView_1_RowCommand(object sender, GridViewCommandEventArgs
{
    if (e.CommandName.Equals("AddNew"))
    {

```

```

        DropDownList cmbNewkodsotr =
        (DropDownList)GridView_1.FooterRow.FindControl("cmbNewkodsotr");
        DropDownList cmbNewkodzayav =
        (DropDownList)GridView_1.FooterRow.FindControl("cmbNewkodzayav");
        TextBox txtNewkodzaved =
        (TextBox)GridView_1.FooterRow.FindControl("txtNewkodzaved");
        TextBox txtNewdata = (TextBox)GridView_1.FooterRow.FindControl("txtNewdata");
        DropDownList cmbNewСеместр =
        (DropDownList)GridView_1.FooterRow.FindControl("cmbNewСеместр");
        TextBox cmbNewkolvomest =
        (TextBox)GridView_1.FooterRow.FindControl("txtNewkolvomest");
        TextBox txtNewпервое_блюдо =
        (TextBox)GridView_1.FooterRow.FindControl("txtNewпервое_блюдо");
        TextBox txtNewвторое_блюдо =
        (TextBox)GridView_1.FooterRow.FindControl("txtNewвторое_блюдо");
        TextBox txtNewсалат =
        (TextBox)GridView_1.FooterRow.FindControl("txtNewсалат");
        TextBox txtNewсусын =
        (TextBox)GridView_1.FooterRow.FindControl("txtNewсусын");
        Insert(
            cmbNewkodsotr.SelectedValue,
            cmbNewkodzayav.SelectedValue,
            txtNewkodzaved.Text,
            txtNewdata.Text,
            cmbNewСеместр.SelectedValue,
            cmbNewkolvomest.Text,
            txtNewпервое_блюдо.Text,
            txtNewвторое_блюдо.Text,
            txtNewсалат.Text,
            txtNewсусын.Text,
            Session["db_id"].ToString());

        // O_clear();
        sortirovka();
    }
}
protected void cmbkodzayav_SelectedIndexChanged(object sender, EventArgs e)
{
    DropDownList cmbkodzayav =
    (DropDownList)GridView_1.Rows[GridView_1.EditIndex].FindControl("cmbkodzayav");
    TextBox txtkodzaved =
    (TextBox)GridView_1.Rows[GridView_1.EditIndex].FindControl("txtkodzaved");
    TextBox txtdata =
    (TextBox)GridView_1.Rows[GridView_1.EditIndex].FindControl("txtdata");
    TextBox txtkolvomest =
    (TextBox)GridView_1.Rows[GridView_1.EditIndex].FindControl("txtkolvomest");

    SqlConnection con = new SqlConnection(conn);
    con.Open();
    SqlCommand cmd = new SqlCommand("select * from kassakafe where kodzayav=" +
        cmbkodzayav.SelectedValue.ToString() + " and db_id=" +

```



```

Session["db_id"].ToString(), con); SqlDataReader dr
    = cmd.ExecuteReader(); while (dr.Read())
    {
        txtkodzaved.Text = dr["kodzaved"].ToString();
        txtdata.Text = dr["data"].ToString(); txtkolvomest.Text
        = dr["kolvomest"].ToString();
    }
    con.Close();
}
protected void cmbNewkodzayav_SelectedIndexChanged(object sender, EventArgs e)
{
    DropDownList cmbNewkodzayav =
    (DropDownList)GridView_1.FooterRow.FindControl("cmbNewkodzayav");

    TextBox txtNewkodzaved =
    (TextBox)GridView_1.FooterRow.FindControl("txtNewkodzaved");
    TextBox txtNewdata = (TextBox)GridView_1.FooterRow.FindControl("txtNewdata");
    TextBox txtNewkolvomest =
    (TextBox)GridView_1.FooterRow.FindControl("txtNewkolvomest");

    SqlConnection con = new SqlConnection(conn);
    con.Open();
    SqlCommand cmd = new SqlCommand("select * from kassakafe where kodzayav=" +
    cmbNewkodzayav.SelectedValue.ToString() + " and db_id=" + Session["db_id"].ToString(),
    con);
    SqlDataReader dr = cmd.ExecuteReader();
    while (dr.Read())
    {
        txtNewkodzaved.Text = dr["kodzaved"].ToString();

        txtNewdata.Text = dr["data"].ToString(); txtNewkolvomest.Text =
        dr["kolvomest"].ToString();
    }
    con.Close();
}
protected void GridView_1_RowDataBound(object sender, GridViewRowEventArgs e)
{
    if (e.Row.RowType == DataControlRowType.DataRow)
    {
        DropDownList cmbType = (DropDownList)e.Row.FindControl("cmbkodsotr");

        if (cmbType != null)
        {
            cmbType.DataSource = select_test14();
            cmbType.DataBind();
            SqlConnection con = new SqlConnection(conn);
            con.Open();
            SqlCommand cmd = new SqlCommand("Select * From test14 where id=" +
            GridView_1.DataKeys[e.Row.RowIndex].Values[0].ToString() + " and db_id=" +
            Session["db_id"].ToString(), con);

```

```

        SqlDataReader dr = cmd.ExecuteReader();
        while (dr.Read())
        {
            cmbType.SelectedValue = dr["kodsotr"].ToString();
        }
        con.Close();
    }
}
if (e.Row.RowType == DataControlRowType.Footer)
{
    DropDownList cmbNewType =
(DropDownList)e.Row.FindControl("cmbNewkodsotr");
    cmbNewType.DataSource = select_test14();
    cmbNewType.DataBind(); cmbNewType.Items.Insert(0,
    new ListItem(" ", " "));
}
//*****
*****
    if (e.Row.RowType == DataControlRowType.DataRow)
    {
        DropDownList cmbType = (DropDownList)e.Row.FindControl("cmbkodzayav"); if
        (cmbType != null)
        {
            cmbType.DataSource = select_kodzayav();
            cmbType.DataBind();
            SqlConnection con = new SqlConnection(conn);
            con.Open();
            SqlCommand cmd = new SqlCommand("Select * From test14 where id=" +
GridView_1.DataKeys[e.Row.RowIndex].Values[0].ToString() + " and db_id=" +
Session["db_id"].ToString(), con);
            SqlDataReader dr = cmd.ExecuteReader();
            while (dr.Read())
            {
                cmbType.SelectedValue = dr["kodzayav"].ToString();
            }
            con.Close();
        }
    }
    if (e.Row.RowType == DataControlRowType.Footer)
    {
        DropDownList cmbNewType =
(DropDownList)e.Row.FindControl("cmbNewkodzayav");
        cmbNewType.DataSource = select_kodzayav();
        cmbNewType.DataBind(); cmbNewType.Items.Insert(0,
        new ListItem(" ", " "));
    }
}
//----- Cancel
protected void GridView_1_RowCancelingEdit(object sender,
GridViewCancelEventArgs e)
{

```

```

        GridView_1.EditIndex = -1;
        sortirovka();
    }
    //----- Update
    protected void GridView_1_RowEditing(object sender, GridViewEditEventArgs e)
    {
        GridView_1.EditIndex = e.NewEditIndex;
        sortirovka();
    }
    protected void GridView_1_RowDeleting(object sender, GridViewDeleteEventArgs e)
    {
        Delete(Convert.ToInt32(GridView_1.DataKeys[e.RowIndex].Values[0].ToString()));
        sortirovka();
    }
    // UPDATE
    protected void GridView_1_RowUpdating(object sender, GridViewUpdateEventArgs e)
    {
        DropDownList cmbkodsotr =
        (DropDownList)GridView_1.Rows[GridView_1.EditIndex].FindControl("cmbkodsotr");
        DropDownList cmbkodzayav =
        (DropDownList)GridView_1.Rows[GridView_1.EditIndex].FindControl("cmbkodzayav");
        TextBox txtkodzaved =
        (TextBox)GridView_1.Rows[GridView_1.EditIndex].FindControl("txtkodzaved");
        TextBox txtdata =
        (TextBox)GridView_1.Rows[GridView_1.EditIndex].FindControl("txtdata");

        DropDownList cmbСеместр =
        (DropDownList)GridView_1.Rows[GridView_1.EditIndex].FindControl("cmbСе
местр");
        TextBox txtkolvomest =
        (TextBox)GridView_1.Rows[GridView_1.EditIndex].FindControl("txtkolvomest");
        TextBox txtпервое_блюдо =
        (TextBox)GridView_1.Rows[GridView_1.EditIndex].FindControl("txtпервое_блюдо");
        TextBox txtвторое_блюдо =
        (TextBox)GridView_1.Rows[GridView_1.EditIndex].FindControl("txtвторое_блюдо");
        TextBox txtсалат =
        (TextBox)GridView_1.Rows[GridView_1.EditIndex].FindControl("txtсалат");
        TextBox txtцусын =
        (TextBox)GridView_1.Rows[GridView_1.EditIndex].FindControl("txtцусын");
        Update(Convert.ToInt32(GridView_1.DataKeys[e.RowIndex].Values[0].ToString()),
            cmbkodsotr.SelectedValue,
            cmbkodzayav.SelectedValue,
            txtkodzaved.Text,
            txtdata.Text,
            cmbСеместр.SelectedValue,
            txtkolvomest.Text,
            txtпервое_блюдо.Text,
            txtвторое_блюдо.Text,
            txtсалат.Text, txtцусын.Text
        );
        GridView_1.EditIndex = -1; //

```

```

        O_clear();
        sortirovka();
    }

    public void Update(int id, string kodsotr, string kodzayav, string kodzaved, string data,
string string kolvomest,
string первое_блюдо, string второе_блюдо, string салат, string сусын)
    {
        string sql = @"UPDATE test14 SET
            kodsotr=" + kodsotr
        + ", kodzayav = " + kodzayav
        + ", kodzaved=" + kodzaved
        + ", data = " + data
        + ", kolvomest=" + kolvomest
        + ", первое_блюдо = " + первое_блюдо
        + ", второе_блюдо = " + второе_блюдо
        + ", салат=" + салат
        + ", сусын = " + сусын
        + " Where id=" + id;

        SqlConnection con = new SqlConnection(conn);
        con.Open();
        SqlCommand cmd = new SqlCommand(sql, con);
        cmd.ExecuteNonQuery();
        con.Close();
        con.Dispose();
    }

    protected void Zvros_Click(object sender, EventArgs e)
    {
        DropDownList1.SelectedIndex =
DropDownList1.Items.IndexOf(DropDownList1.Items.FindByText("kodsotr"));
        DropDownList2.SelectedIndex =
DropDownList2.Items.IndexOf(DropDownList2.Items.FindByText("Прих_пах"));
        DropDownList3.SelectedIndex =
DropDownList3.Items.IndexOf(DropDownList3.Items.FindByText("kolvomest"));
        DropDownList4.SelectedIndex =
DropDownList4.Items.IndexOf(DropDownList4.Items.FindByText("kodzayav"));
        FillCustomerInGrid();
    }

    // public DataTable O_clear()
    //{
    // string sql = "exec procedure_test14";
    // SqlDataAdapter da = new SqlDataAdapter(sql, conn);
    // DataTable dt = new DataTable();
    // da.Fill(dt);
    // return dt;
    // }

    public DataTable Fetch()
    {
        string sql = "SELECT * ,ISNULL(первое_блюдо,0)+ ISNULL([второе_блюдо],0)
+ISNULL([салат],0)+ ISNULL([сусын],0) as [Всего],ID From test14 where db_id=" +
Session["db_id"].ToString();

        SqlDataAdapter da = new SqlDataAdapter(sql, conn);

```

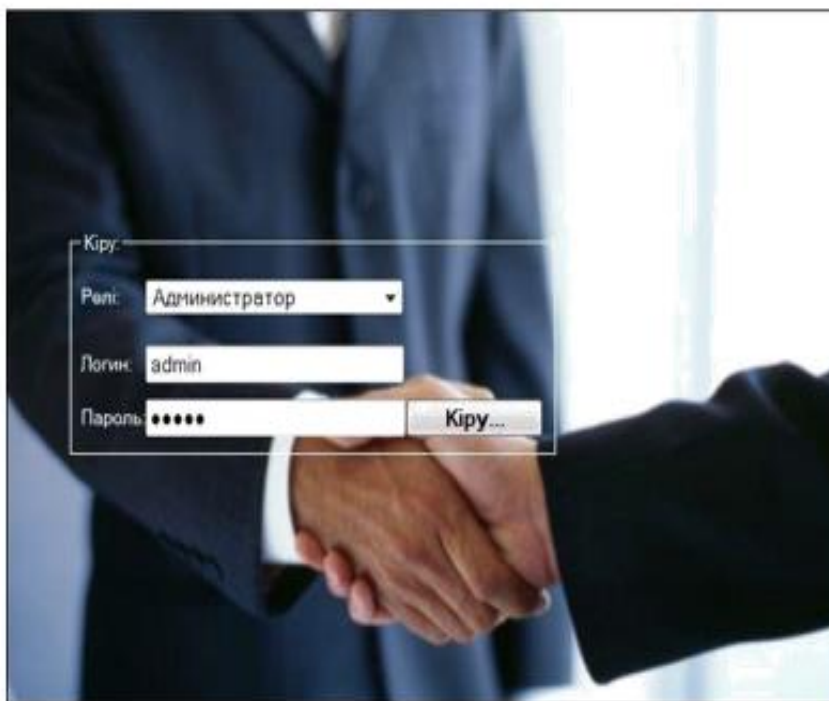
```

        DataTable dt = new DataTable();
        da.Fill(dt); return
        dt;
    }
    public void Delete(int id)
    {
        string sql = "Delete test14 Where id=" + id;
        SqlConnection con = new SqlConnection(conn);
        con.Open();
        SqlCommand cmd = new SqlCommand(sql, con);
        cmd.ExecuteNonQuery();
        con.Close();
        con.Dispose();
    }
    public DataTable select_test14()
    {
        string sql = "Select Distinct [kodsotr] From [sotrudniki] where db_id=" +
Session["db_id"].ToString();
        SqlDataAdapter da = new SqlDataAdapter(sql, conn);
        DataTable dt = new DataTable();

        da.Fill(dt); return dt;
    }
    public DataTable select_kodzayav()
    {
        string sql = "Select Distinct kodzayav From [kassakafe] where db_id=" +
Session["db_id"].ToString();
        SqlDataAdapter da = new SqlDataAdapter(sql, conn);
        DataTable dt = new DataTable(); da.Fill(dt);
        return dt;
    }
    public void Insert(string kodsotr, string kodzayav, string kodzaved, string data, string string
kolvomest, string первое_блюдо, string второе_блюдо, string салат, string сусын, string db_id)
    {
        string sql = "Insert Into test14 Values (" + kodsotr + ", " + kodzayav + ", " + kodzaved +
", " + data + ", " + kolvomest + ", " + первое_блюдо + ", " + второе_блюдо + ", " + салат +
", " + сусын + ", " + db_id + ")";
        //TextBox2.Text="Insert Into test14 Values (" + kodsotr + ", " + kodzayav + ", " +
kodzaved + ", " + data + ", " + kolvomest + ", " + первое_блюдо + ", " + второе_блюдо +
", " + салат + ", " + сусын + ", " + db_id + ")";
        SqlConnection con = new SqlConnection(conn);
        con.Open();
        SqlCommand cmd = new SqlCommand(sql, con);
        cmd.ExecuteNonQuery();
        con.Close();
        con.Dispose();
    }

```

Бағдарламаны қосу кезінде, алдымен бағдарламаның алғашқы беті ашылады.



3.2 сурет– Жүйеге администратор арқылы кіру терезесі

Қолданушылар:

- администратор;
- пайдаланушы.

Мұнда негізгі мәзір, кафе, ТЖО, қонақ үй, жанармай бекеті жайлы мәліметтер бар. Бұл жерде пайдаланушы өзіне қажетті операторды таңдай алады.

Қош келдіңіз - Kasiet!

артқа...

Қолданушы логины	Қолданушы паролі	Қолданушы ны тіркеген күні	Edit	X
Ulpan	1234	19.04.14	Edit	X
Kasiet	123	20.04.14	Edit	X
Waripa	1569	01.05.14	Edit	X
ssop	9876	06.05.14	Edit	X
uuuu	1653	07.05.14	Edit	X
			Add	

3.3 сурет– Бағдарламаның негізгі терезесі

Жүйеге администратор ретінде кіру арқылы жасалып жатқан жұмыстарды басқаруға болады. Ал қолданушы терезесі арқылы кіріп ақпараттарды қарауға және жаңа ақпараттар енгізуге болады.



3.4 сурет– Жүйеге қолданушы ретінде кіру

Жеке кәсіпкерлікте атқарылып жатқан жұмыстарды қарауға болады.

ЖЕКЕ КӘСІПКЕР						
<div>Басты бөлім    Кафе    ТЖО    Қонақ үй    Жанармай_бекет    Табыс кестесі    Кафе кассасы    Қызметкерлер</div>						
Қызметкер коды	ФИО	Туған кун	Жұмысқа қабылданған уақыт	Қызмет коды	Өзгерту	Жою
1	Азамат Тимуров	25.06.87	29.04.14	2	<a href="#">Өзгерту</a>	<a href="#">Жою</a>
2	Ерсултан Дауленулы	23.08.88	19.03.14	1	<a href="#">Өзгерту</a>	<a href="#">Жою</a>
3	Nurlan Sarsenali	03.08.90	18.04.14	1	<a href="#">Өзгерту</a>	<a href="#">Жою</a>
4	Нурдаулет Барат	13.10.92	17.04.14	3	<a href="#">Өзгерту</a>	<a href="#">Жою</a>
5	Магжан Мелдехан	23.05.90	29.10.14	4	<a href="#">Өзгерту</a>	<a href="#">Жою</a>
6	Марат Елен	11.07.86	09.07.13	5	<a href="#">Өзгерту</a>	<a href="#">Жою</a>
7	Нурболсын Лесбек	01.08.88	12.08.13	7	<a href="#">Өзгерту</a>	<a href="#">Жою</a>
8	Жалғас Пернебек	02.08.89	02.08.13	8	<a href="#">Өзгерту</a>	<a href="#">Жою</a>
9	Ансар Мансуров	09.06.92	18.03.13	5	<a href="#">Өзгерту</a>	<a href="#">Жою</a>
10	Диас Каратаев	09.07.92	23.08.13	4	<a href="#">Өзгерту</a>	<a href="#">Жою</a>
11	Бағлан Сарманов	29.03.92	13.06.13	1	<a href="#">Өзгерту</a>	<a href="#">Жою</a>
12	Данияр Бапахов	19.09.84	13.07.13	1	<a href="#">Өзгерту</a>	<a href="#">Жою</a>
				1	<a href="#">Өзгерту</a>	<a href="#">Жою</a>

Охрана  
Официант  
Повар  
Қызметшілер  
Тазалаушылар  
Механик  
Бармен  
Болме тазалаушы

3.5 сурет– Барлық қызметкерлер жайлы мәлімет

ЖЕКЕ КӘСПКЕР					
Басты бөлім	Кафе	ТЖО	Қонақ үй	Жанармай_бекет	Табыс кестесі
Кафе кассасы					
Қызметкер коды	ФИО	Туған күн	Жұмысқа қабылданған уақыты	Қызмет коды	
2	Ерсултан Дауленулы	23.08.88	19.03.14	1	
3	Nurlan Sarsenali	03.08.90	18.04.14	1	
11	Бағлан Сарманов	29.03.92	13.06.13	1	
12	Данияр Бапахов	19.09.84	13.07.13	1	

Охрана  
 Официант  
 Повар  
 Қызметшілер  
 Тазалаушылар  
 Механик  
 Бармен  
 Болме тазалаушы

3.6 сурет– Күзет бөліміндегі қызметкерлер жайлы мәлімет

ЖЕКЕ КӘСПКЕР

Басты бөлім

Кафе

ТЖО

Қонақ үй

Жанармай\_бекет

Табыс кестесі

Кафе кассасы

Қызметкерлер

Қызметкер коды

Пайда\_шығын

Бағасы

Жұмыс орны коды

Алып төстөу

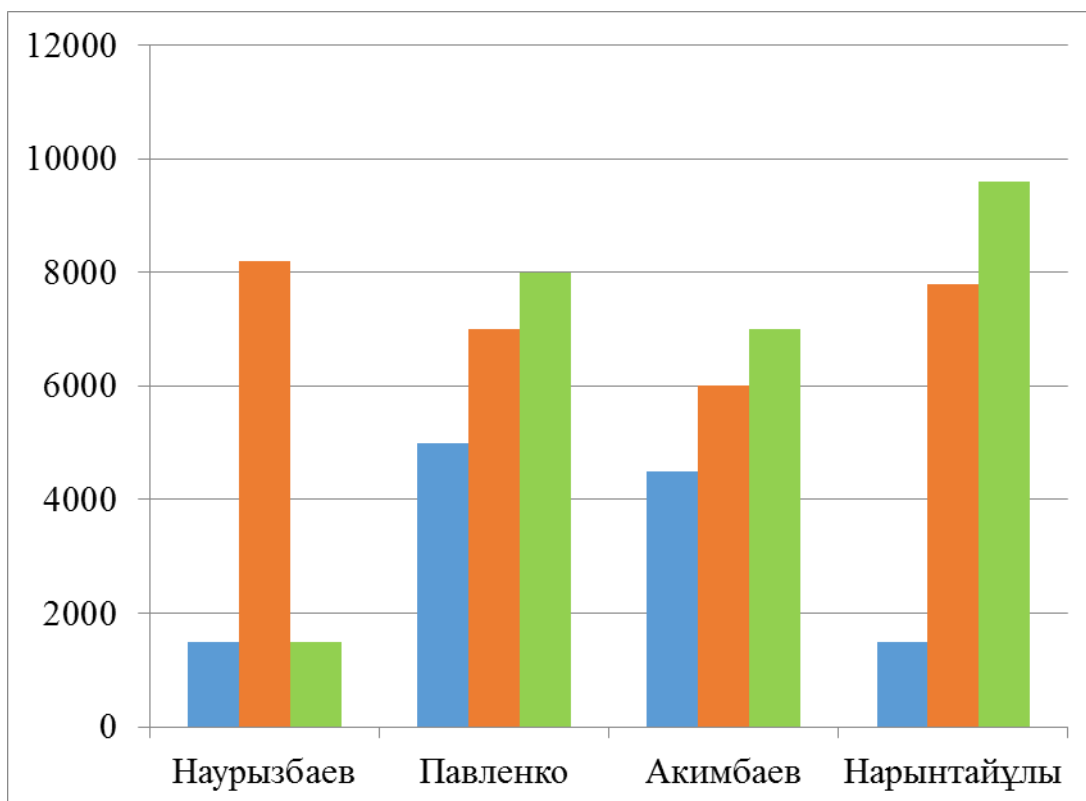
Қызметкер коды	Жұмыс орны коды	Тұтынушы коды	Жеке куәлік	Прих_расх	Бағасы	Тапсырыс1	Тапсырыс1	Тапсырыс1	Тапсырыс4	Барлығы
2	3	Наурызбаев	152635	1	5500	1500	8200	1500	4600	15800
1	2	Павленко	123065	1	4500	45450	4500	6320	8000	64270
3	1	Нарынтайұлы	45602	1	1200	1500	7800	9600	4560	23460
4	4	Акимбаев	896420	2	5000	4500	9000	7800	6000	27300
5	1	Нарынтайұлы	45602	1	1200	8897	4556	48596	6988	69037
4	3	Наурызбаев	152635	1	5500	4500	6000	7800	9000	27300
5	4	Акимбаев	896420	1	5000	4500	6000	7000	8000	25500
6	2	Павленко	123065	1	4500	5000	7000	8000	4500	24500
				1						

	Тапсырыс1	Тапсырыс2	Тапсырыс3	Тапсырыс4	Всего
Пайда	71347	44056	88816	45648	249867
Шығын	4500	9000	7800	6000	27300
Жыл бойы	75847	53056	96616	51648	277167

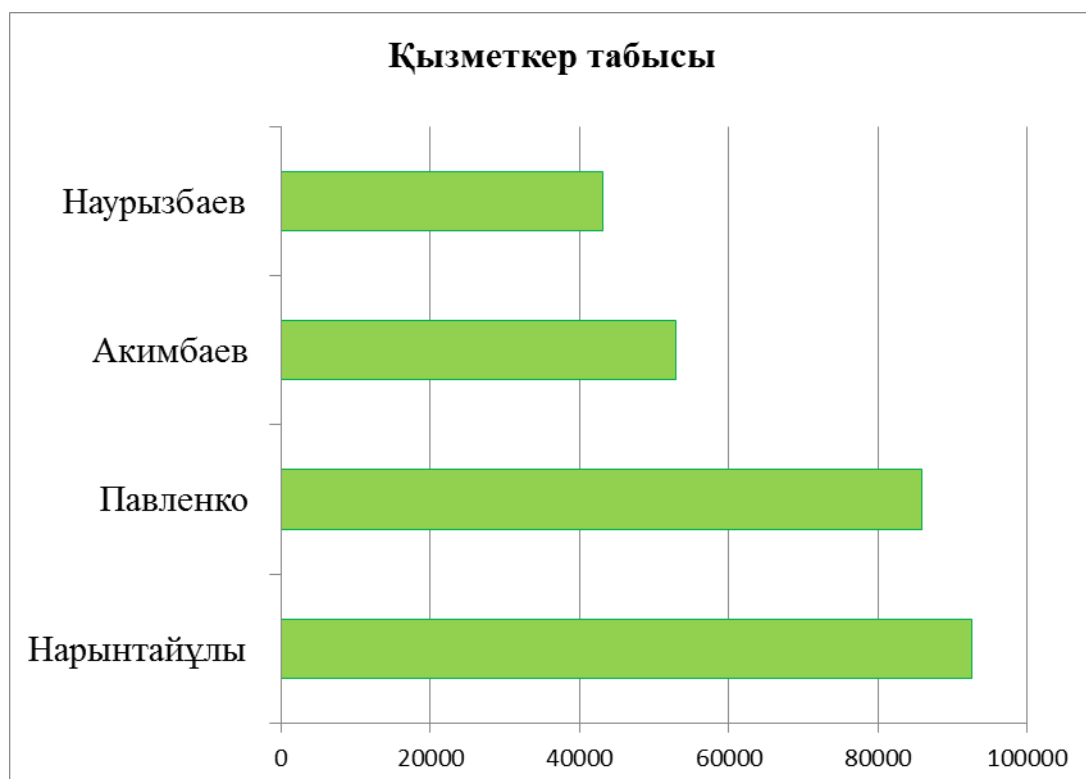
Excel-ға шығару

3.7 сурет – Жүйенің есептік бөлімі





3.8 сурет – Өр тапсырыс кезіндегі кіріс санының өзгерісі



3.9 сурет – Қызметкерлердің бір күндік тапсырыстарының табысы

ЖЕКЕ КӘСІПКЕР

Басты бөлімКафеТЖОҚонақ үйЖанармай бекетТабыс кестесіКафе кассасыҚызметкерлер

Қызметкер коды + Тапсырыс + Орын саны + Тапсырыс коды Алып тастау

Қызметкер коды	Тапсырыс коды	Мекеме коды	Уақыты	Тапсырыс	Орын саны	Бірінші ас	Екінші ас	салат	сусын	Всего	Өзгерту	Жою
1	1202	1	21.04.14	1	4	500	600	300	500	1900	Өзгерту	Жою
2	1203	1	21.04.14	1	10	300	200	300	500	1300	Өзгерту	Жою
2	1205	1	22.04.14	1	9	600	200	300	200	1300	Өзгерту	Жою
1	1208	1	24.04.14	2	4	1000	600	400	500	2500	Өзгерту	Жою
2	1209	1	24.04.14	1	8	800	900	500	600	2800	Өзгерту	Жою
1	1210	1	25.04.14	2	4	800	1000	300	600	2700	Өзгерту	Жою
2	1203	1	21.04.14	1	10	600	900	400	500	2400	Өзгерту	Жою
1	1213	1	28.04.14	1	8	1500	1500	500	500	4000	Өзгерту	Жою
3	1211	1	26.04.14	1	5	500	900	300	600	2300	Өзгерту	Жою
				1							Енгізу	

	Бірінші ас	Екінші ас	салат	сусын	Всего
Базалық	4800	5200	2600	3400	16000
VIP стол	1800	1600	700	1100	5200
Барлығы	6600	6800	3300	4500	21200

3.10 сурет– Кафе кассасының есептік бөлімі

Касса кафе бетінің ASP.NET коды

```
<%@ Page Title="Кафе кассасы" Language="C#" MasterPageFile="~/Site.master"
AutoEventWireup="true" CodeBehind="WebForm2.aspx.cs"
Inherits="diplomka.WebForm2" %>
```

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

```
<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="server">
```

```
<asp:ScriptManager ID="myScriptManager" runat="server"> </asp:ScriptManager>
```

```
<asp:UpdatePanel ID="JokeUpdatePanel" runat="server"> <ContentTemplate>
```

```
<script language="Javascript" type="text/javascript"> function EnsureNumeric() {
```

```
var key = window.event.keyCode; if (key < 48 || key > 57)
```

```
window.event.returnValue = false;
```

```
}
```

```
</script>
```

```
<table>
```

```
<tr>
```

```
<td>
```

```
<asp:DropDownList ID="DropDownList1" runat="server" Height="22px" Width="142px"
AutoPostBack="true"
```

```
onselectedindexchanged="DropDownList1_SelectedIndexChanged" ></asp:DropDownList> +
```

```
<asp:DropDownList ID="DropDownList2" runat="server" Height="22px" Width="142px"
AutoPostBack="true"
```

```
onselectedindexchanged="DropDownList2_SelectedIndexChanged" >
```

```
<asp:ListItem>Тапсырыс</asp:ListItem> <asp:ListItem>1</asp:ListItem>
```

```

<asp:ListItem>2</asp:ListItem> </asp:DropDownList> +
<asp:DropDownList ID="DropDownList3" runat="server" Height="22px" Width="142px"
AutoPostBack="true"
onselectedindexchanged="DropDownList3_SelectedIndexChanged"></asp:DropDownList> +
<asp:DropDownList ID="DropDownList4" runat="server" Height="22px" Width="142px"
AutoPostBack="true"
onselectedindexchanged="DropDownList4_SelectedIndexChanged"></asp:DropDownList>
<asp:Button ID="Zvros" runat="server" Text="Сõпор" style="height: 25px"
width="65px" onclick="Zvros_Click"/>
</td>
</tr>
<tr><td>
<asp:Panel runat="server" ID="MainPanel" ScrollBars="Auto" Height="315" Width="105%">
<asp:GridView ID="GridView_1" runat="server" AutoGenerateColumns="False"
ShowFooter="True" DataKeyNames="id,db_id"
OnRowEditing="GridView_1_RowEditing"
OnRowCancelingEdit="GridView_1_RowCancelingEdit"
OnRowUpdating="GridView_1_RowUpdating"
OnRowDataBound="GridView_1_RowDataBound"
OnRowCommand="GridView_1_RowCommand" OnRowDeleting="GridView_1_RowDeleting"
>
<Columns>
<asp:TemplateField HeaderText="Кызметкер коды"> <EditItemTemplate>
<asp:DropDownList ID="cmbkodsotr" Width="80px" runat="server" DataTextField="kodsotr"
DataValueField="kodsotr">
</asp:DropDownList>
</EditItemTemplate>
<ItemTemplate>
<asp:Label ID="kodsotr" runat="server" Text='<%# Eval("kodsotr") %>'></asp:Label>
</ItemTemplate>
<FooterTemplate>
<asp:DropDownList ID="cmbNewkodsotr" Width="80px" runat="server"
DataTextField="kodsotr" DataValueField="kodsotr"> </asp:DropDownList>
</FooterTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="Тапсырыс коды"> <EditItemTemplate>
<asp:DropDownList ID="cmbkodzayav"
onselectedindexchanged="cmbkodzayav_SelectedIndexChanged" Width="50px"
AutoPostBack="true" runat="server" DataTextField="kodzayav" DataValueField="kodzayav">
</asp:DropDownList>
</EditItemTemplate>
<ItemTemplate>
<asp:Label ID="kodzayav" runat="server" Text='<%# Eval("kodzayav") %>'></asp:Label>
</ItemTemplate>
<FooterTemplate>

```

```

<asp:DropDownList ID="cmbNewkodzayav"
onselectedindexchanged="cmbNewkodzayav_SelectedIndexChanged" Width="50px"
AutoPostBack="true" runat="server" DataTextField="kodzayav" DataValueField="kodzayav">
</asp:DropDownList>
</FooterTemplate>
</asp:TemplateField>

<asp:TemplateField HeaderText="Мекеме коды" SortExpression="kodzaved">
<EditItemTemplate> <asp:TextBox OnKeyPress="EnsureNumeric()" Enabled="false"
Width="30px" ID="txtkodzaved" runat="server" Text='< %# Eval("kodzaved") %>'>
</asp:TextBox> </EditItemTemplate>

<FooterTemplate> <asp:TextBox OnKeyPress="EnsureNumeric()" Enabled="false"
Width="30px" ID="txtNewkodzaved" runat="server"> </asp:TextBox>
</FooterTemplate>
<ItemTemplate> <asp:Label ID="Label2" runat="server" Text='< %#
Bind("kodzaved")
%>'> </asp:Label> </ItemTemplate>

</asp:TemplateField>

<asp:TemplateField HeaderText="Уақыты"> <EditItemTemplate>

<asp:TextBox OnKeyPress="EnsureNumeric()" Enabled="false" Width="30px" ID="txtdata"
runat="server" Text='< %# Bind("data") %>'></asp:TextBox>

</EditItemTemplate>
<FooterTemplate>

<asp:TextBox OnKeyPress="EnsureNumeric()" Enabled="false" Width="30px" ID="txtNewdata"
runat="server" ></asp:TextBox>

</FooterTemplate>
<ItemTemplate>

<asp:Label ID="data" runat="server" Text='< %# Bind("data") %>'></asp:Label>
</ItemTemplate>

</asp:TemplateField>

<asp:TemplateField HeaderText="Тапсырыс"> <EditItemTemplate>

<asp:DropDownList Width="40px" ID="cmbСеместр" runat="server" SelectedValue='< %#
Eval("Семестр") %>'>

<asp:ListItem Value="1" Text="1"></asp:ListItem> <asp:ListItem Value="2"
Text="2"></asp:ListItem> </asp:DropDownList>

</EditItemTemplate>
<ItemTemplate>

<asp:Label ID="Семестр" runat="server" Text='< %# Eval("Семестр") %>'></asp:Label>
</ItemTemplate>

<FooterTemplate>

<asp:DropDownList Width="40px" ID="cmbNewСеместр" runat="server" > <asp:ListItem
Selected="True" Text="1" Value="1"></asp:ListItem> <asp:ListItem Text="2"
Value="2"></asp:ListItem> </asp:DropDownList> </FooterTemplate>

</asp:TemplateField>

<asp:TemplateField HeaderText="Орын саны"> <EditItemTemplate>

<asp:TextBox OnKeyPress="EnsureNumeric()" Enabled="false" Width="50px"
ID="txtkolvomest" runat="server" Text='< %# Bind("kolvomest") %>'></asp:TextBox>

</EditItemTemplate>
<FooterTemplate>

```

```

<asp:TextBox OnKeyPress="EnsureNumeric()" Enabled="false" Width="50px"
ID="txtNewkolvomest" runat="server" ></asp:TextBox>

</FooterTemplate>
</ItemTemplate>

<asp:Label ID="kolvomest" runat="server" Text='<%# Bind("kolvomest") %>'></asp:Label>
</ItemTemplate>

</asp:TemplateField>

<asp:TemplateField HeaderText="первое_блюдо"> <EditItemTemplate>

<asp:TextBox OnKeyPress="EnsureNumeric()" Width="35px" ID="txtпервое_блюдо"
runat="server" Text='<%# Bind("первое_блюдо") %>'></asp:TextBox> </EditItemTemplate>

<FooterTemplate>

<asp:TextBox OnKeyPress="EnsureNumeric()" Width="35px" ID="txtNewпервое_блюдо"
runat="server" ></asp:TextBox>

</FooterTemplate>
</ItemTemplate>

<asp:Label ID="первое_блюдо" runat="server" Text='<%# Bind("первое_блюдо")
%>'></asp:Label>

</ItemTemplate>
</asp:TemplateField>

<asp:TemplateField HeaderText="второе_блюдо"> <EditItemTemplate>

<asp:TextBox OnKeyPress="EnsureNumeric()" Width="30px" ID="txtвторое_блюдо"
runat="server" Text='<%# Bind("второе_блюдо") %>'></asp:TextBox> </EditItemTemplate>
<FooterTemplate>

<asp:TextBox OnKeyPress="EnsureNumeric()" Width="30px" ID="txtNewвторое_блюдо"
runat="server" ></asp:TextBox>

</FooterTemplate>
</ItemTemplate>

<asp:Label ID="второе_блюдо" runat="server" Text='<%# Bind("второе_блюдо")
%>'></asp:Label>

</ItemTemplate>

</asp:TemplateField>

<asp:TemplateField HeaderText="салат"> <EditItemTemplate>

<asp:TextBox OnKeyPress="EnsureNumeric()" Width="20px" ID="txtсалат" runat="server"
Text='<%# Bind("салат") %>'></asp:TextBox>

</EditItemTemplate>
<FooterTemplate>

<asp:TextBox OnKeyPress="EnsureNumeric()" Width="20px" ID="txtNewсалат" runat="server"
></asp:TextBox>

</FooterTemplate>
</ItemTemplate>

<asp:Label ID="салат" runat="server" Text='<%# Bind("салат") %>'></asp:Label>
</ItemTemplate>

</asp:TemplateField>

<asp:TemplateField HeaderText="сусын"> <EditItemTemplate>

<asp:TextBox OnKeyPress="EnsureNumeric()" Width="30px" ID="txtсусын" runat="server"

```

```

Text='< %# Bind("сусын") %>'></asp:TextBox>

</EditItemTemplate>
<FooterTemplate>

<asp:TextBox OnKeyPress="EnsureNumeric()" Width="30px" ID="txtNewсусын"
runat="server" ></asp:TextBox>

</FooterTemplate>
<ItemTemplate>

<asp:Label ID="сусын" runat="server" Text='< %# Bind("сусын") %>'></asp:Label>
</ItemTemplate>

</asp:TemplateField>

<asp:TemplateField HeaderText="Бсөро"> <EditItemTemplate>

<asp:TextBox OnKeyPress="EnsureNumeric()" Width="35px" Enabled="false" ID="txtБсөро"
runat="server" Text='< %# Bind("Бсөро") %>'></asp:TextBox>

</EditItemTemplate>
<FooterTemplate>

<asp:TextBox OnKeyPress="EnsureNumeric()" Width="35px" Enabled="false"
ID="txtNewБсөро" runat="server" ></asp:TextBox>

</FooterTemplate>
<ItemTemplate>

<asp:Label ID="Бсөро" runat="server" Text='< %# Bind("Бсөро") %>'></asp:Label>
</ItemTemplate>

</asp:TemplateField>

<asp:TemplateField HeaderText="Өзгөртү" ShowHeader="False"> <EditItemTemplate>

<asp:LinkButton ID="LinkButton1" runat="server" CausesValidation="True"
CommandName="Update" Text="Жаңарту"></asp:LinkButton> <asp:LinkButton
ID="LinkButton2" runat="server" CausesValidation="False" CommandName="Cancel"
Text="Болдырмай"></asp:LinkButton>

</EditItemTemplate>
<FooterTemplate>

<asp:LinkButton ID="LinkButton2" runat="server" CausesValidation="False"
CommandName="AddNew" Text="Еңгізү"></asp:LinkButton> </FooterTemplate>
<ItemTemplate>

<asp:LinkButton ID="LinkButton1" runat="server" CausesValidation="False"
CommandName="Edit" Text="Өзгөртү"></asp:LinkButton>

</ItemTemplate>
</asp:TemplateField>

<asp:CommandField HeaderText="Жою" DeleteText="Жою" ShowDeleteButton="True"
ShowHeader="True" />

</Columns>

<FooterStyle BackColor="White" ForeColor="#000066" />

<HeaderStyle BackColor="#239FE2" Font-Bold="True" ForeColor="White" /> <PagerStyle
BackColor="White" ForeColor="#000066" HorizontalAlign="Left" /> <RowStyle
ForeColor="#000066" />

<SelectedRowStyle BackColor="#669999" Font-Bold="True" ForeColor="White" />
<SortedAscendingCellStyle BackColor="#F1F1F1" /> <SortedAscendingHeaderStyle
BackColor="#007DBB" /> <SortedDescendingCellStyle BackColor="#CAC9C9" />

```

```

<SortedDescendingHeaderStyle BackColor="#00547E" />
</asp:GridView>
<asp:GridView ID="GridView1" runat="server" Visible="false" Width="16px">
</asp:GridView>
</asp:Panel></td></tr>
<tr><td> <div style="margin-left:236px">
<asp:GridView ID="GridView2" runat="server" BackColor="White" BorderColor="#CCCCCC"
BorderStyle="None" BorderWidth="1px" CellPadding="3">
<FooterStyle BackColor="White" ForeColor="#000066"/>
<HeaderStyle BackColor="#006699" Font-Bold="True" ForeColor="White" /> <PagerStyle
BackColor="White" ForeColor="#000066" HorizontalAlign="Left" /> <RowStyle
ForeColor="#000066" />
<SelectedRowStyle BackColor="#669999" Font-Bold="True" ForeColor="White" />
<SortedAscendingCellStyle BackColor="#F1F1F1" /> <SortedAscendingHeaderStyle
BackColor="#007DBB" /> <SortedDescendingCellStyle BackColor="#CAC9C9" />
<SortedDescendingHeaderStyle BackColor="#00547E" />
</asp:GridView>
</div></td></tr>
</table>
</ContentTemplate>
</asp:UpdatePanel>
</asp:Content>

```

Дисциплины – Excel (Своей активацией продукта)

ФАЙЛ ГЛАВНАЯ ВСТАВКА РАЗМЕТКА СТРАНИЦЫ ФОРМУЛЫ ДАННЫЕ РЕЦЕНЗИРОВАНИЕ ВИД Нагрузочный тест ABBYY FineReader 11 Па

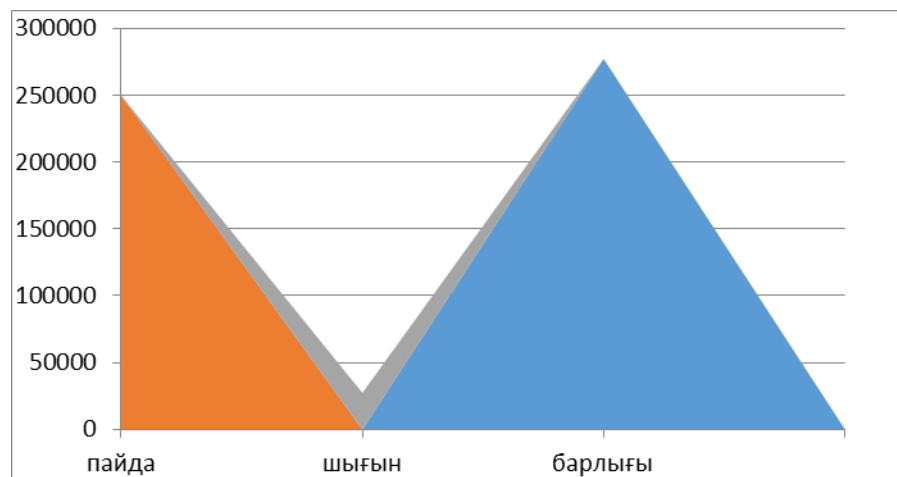
Буфер обмена Вставить Вставить Удалить Формат Ячейки

Шрифт Выравнивание Число Стили

В17 Жыл бойы

	A	B	C	D	E	F	G	H	I	J	K	L
1												
2		Жыл бойы	Жыл бойы	Жыл бойы	Жыл бойы	Жыл бойы	Жыл бойы	Жыл бойы	Жыл бойы	Жыл бойы	Жыл бойы	Жыл бойы
3		Жыл бойы	Жыл бойы	Жыл бойы	Жыл бойы	Жыл бойы	Жыл бойы	Жыл бойы	Жыл бойы	Жыл бойы	Жыл бойы	Жыл бойы
4		Жыл бойы	Жыл бойы	Жыл бойы	Жыл бойы	Жыл бойы	Жыл бойы	Жыл бойы	Жыл бойы	Жыл бойы	Жыл бойы	Жыл бойы
5		Жыл бойы	Жыл бойы	Жыл бойы	Жыл бойы	Жыл бойы	Жыл бойы	Жыл бойы	Жыл бойы	Жыл бойы	Жыл бойы	Жыл бойы
6	1	2	3	Наурызба	152635	1	5500	1500	8200	1500	4600	2
7	2	1	2	Павленко	123065	1	4500	45450	4500	6320	8000	2
8	3	3	1	Нарынтай	45602	1	1200	1500	7800	9600	4560	2
9	4	4	4	Акимбаев	896420	2	5000	4500	9000	7800	6000	2
10	5	5	1	Нарынтай	45602	1	1200	8897	4556	48596	6988	2
11	6	4	3	Наурызба	152635	1	5500	4500	6000	7800	9000	2
12	7	5	4	Акимбаев	896420	1	5000	4500	6000	7000	8000	2
13	8	6	2	Павленко	123065	1	4500	5000	7000	8000	4500	2
14												
15		Пайда					71347	44056	88816	45648	249867	
16		Шығын					4500	9000	7800	6000	27300	
17		Жыл бойы					75847	53056	96616	51648	277167	

3.11 сурет– Excel-ға экспорттау



3.12 сурет – Жыл бойындағы тапсырыстардың пайдасы мен шығыны

Қазіргі технологиялардың дамыған заманында барлық жұмысты бір орталықтан басқару адамдардың жұмысын жеңілдетті. Бұл жоба клиент-сервер технологиясына негізделіп жасалғандықтан кәсіпорын жұмысын оңтайландыруға аса ыңғайлы. Деректер қоры NoSQL-да жасалып, интерфейс ретінде Visual Studio пайдаланылды.

## 4 Өмір тіршілік қауіпсіздігі және еңбек қорғау

### 4.1 Өндірістік ғимараттағы еңбек шартын талдау

Жұмыс орны – белгілі бір жұмыспен немесе операциямен айналысатын орындаушы немесе орындаушылар тобының еңбектік қызметі жүзеге асатын жұмыс орны. Жұмыс орны қажетті жағдайда негізгі технологиялық құралдармен, технологиялық жабдықпен, ұйымдастыру жабдықтарымен, жұмыс жиһаздарымен, бақылауыш-өлшегіш құралдармен, сигнал беру және байланыс құралдарымен жабдықталады. Қарастырылып отырған жұмыс ғимараты Оңтүстік Қазақстан облысында орналасқан.

Жұмыс ғимаратының сипаттамасы:

- жұмыс ғимараты бір қабатты ғимараттың бірінші қабатында орналасқан;
- ғимарат типі: кеңсе;
- жұмыс ғимаратының өлшемі: ұзындығы 4 м, ені 3 м, биіктігі 3 м;
- ғимаратты шынылау – екі есе (бір терезенің өлшемі 2000x2000 мм);
- жасанды жарықтандыру – шамдар: 3 шам, әрқайсысында 2 люминесценцияға негізделген лампалар;
- қабырғалардың ішкі өңделуі – жарық;
- танып білудің кіші объектісі 1-ден 5 мм-ге дейін болғандықтан, ғимарат жұмыстың көру шарты бойынша V разрядқа жатады;
- жұмыс түрі – компьютерде бағдарламалық қамтаманы өңдеу;



- жұмысшы саны – 2;
- жұмыс жасау уақыты – 8 сағаттан 5 күн;
- жұмыс категориясы – жеңіл.

## **4.2 Есептеу техникасы бөлмесіндегі өндірістік ортаның микроклиматы**

Персоналды компьютерде жұмыс жасау кезінде ғимаратта температура жоғарылайды және ауаның қатыстық ылғалдылығы төмендейді, ауаның иондық және сапалық құрамы нашарлайды: ауаның құрамында органикалық заттар және көміртегінің қостотығы артады. Осы себептерге байланысты микроклиматтың 4.1 – кестеде келтірілген оптималды параметрлерін сақтау керек. Сонымен қатар ауа райы жағдайына байланысты жұмыс орнын желдетіп отыру керек, ұзақтығы 10 минуттан кем болмауы керек.

Қауіпті еңбек шартының ұйымдастыру факторлары еңбекті және жұмыс орнын дұрыс ұйыдастырмаумен және қорғаныс құралдарын пайдаланудың ережелерін және нұсқауларын қажетті мөлшерде сақтамаумен байланысты.

### **4.1 к е с т е – Жұмыс орнындағы микроклиматтың оптималды нормасы**

Жыл мезгілі	Жұмыс категориясы	Ауа температурасы °С	Ауаның қатыстық ылғалдылығы, %	Ауаның қозғалыс жылдамдығы, м/с
Суық	Жеңіл – 1а	22-24	40-60	0,1
Жылы	Жеңіл – 1а	23-25	40-60	0,1

Техникалық факторлар құрал-жабдықтардың жетілдірілмеуімен, құрал-жабдықтардың жарамсыздығымен, еңбекті қауіпсіз жүргізудің техникалық құралдарының болмауымен (нашар оқшаулау жағдайының, қорғалған жерге тұйықтаудың болмауы, қоршау және блоктау) байланысты.

Санитарлы-гигиеналық факторлар жарықтанудың және микроклиматтың санитарлық нормасын сақтамаумен, өндірістік ортада сиянды заттарды пайдаланумен, шудың және жоғары жиілікті сәулеленудің болуымен, статикалық электрлеудің жоғары деңгейімен, жоғары кернеуі бар электр қондырғыларын пайдаланумен байланысты.

## **4.3 Жұмыс орнының жарықтандыру жүйесі**

Есептеу техникасы бөлмесінің жарықтануы компьютермен жұмыс жасау кезінде үлкен мәнге ие. Ол көбіне жарықтық және желілік жағдайда анықталады.

Жарықтың жұтылуын азайту үшін төбе мен панельден жоғары қабырғалар (1,5-1,7 м), егер олар дыбыс азайтатын материалмен қапталмаса, ақ эмульсиялық бояумен боялады (шағылысу коэффициенті 0,7-ден кем болмауы керек). Панель қабырғаларын бояу үшін ашық түсті бояуларды тағндаған жөн.

Персоналды электронды есептеу машиналарында жұмыс жасау жарықтандырудың келесі түрлерінде жүзеге асуы мүмкін:

– видеомониторлар бөлменің периметрі бойынша немесе қарсы жаққа

қараған экранды кластардың ұзындығы бойынша жұмыс орнының екі қатарында орналасуы кезіндегі жалпы люминесцентті жарықтандыру;

– экран мен жұмыс үстелінің беткі қабаты жарық тасымалдаушы қабырғаға перпендикуляр орналасқанда жұмыс орнының тек бір немесе үш қатарында орналасуы кезіндегі қатар қолданылатын жарықтандыру (табиғи + жасанды).

Табиғи жарықтандыру персоналды электронды есептеу машинасы бар жұмыс орындары терезесі бар қабырғадан 0,8 – 1,0 м қашықтықта бөлменің ұзындығы бойынша бір қатарда орналасқанда, және экрандар осы қабырғаға перпендикуляр орналасқанда болады.

Табиғи жарықтың негізгі ағыны сол жақтан болуы керек. Табиғи жарықтың негізгі жарықтық ағынының бағыты компьютерде жұмыс жасаған адамға оң жақтан, артқы жақтан және алдынан түсуге рұқсат етілмейді.

#### **4.4 Электр тоғының адам ағзасына әсері және тоқпен зақымданудан қорғану шаралары**

Бағдарламашының жұмысы электр құрылғыларымен тікелей байланысты болғандықтан, оларды дұрыс пайдаланып, электр тоғынан зақымданудан қорғану шараларын сақтау керек.

Электр тоғынан сақтанудың бірнеше шаралары бар. Солардың бірі – қорғаныштық жерге қосу. Қорғаныштық жерге қосу – электрді жермен немесе оның металдық тоқ өткізбейтін бөлігінің эквивалентімен арнайы тұйықтау.

Қорғаныштық жерге қосудың тағайындалуы – корпусқа немесе электр құрылғыларының басқа да тоқ өткізетін бөліктеріне жақындаған жағдайда тоқпен зақымданудың қауіптілігін жояды. Қорғаныштық жерге қосу әрекетінің принципі – корпусқа немесе басқа себептерге қысқа тұйықталуымен анықталған қадамның және жақындау кернеуінің қауіпсіз мәніне дейін төмендетеді.

Сонымен қатар электр тғымен зақымданудан қорғанудың тағы бір шарасы нөлдеу болып табылады. Нөлдеу – металдық тоқ өткізбейтін бөліктің нөлдік қорғаныштық өткізгішімен арнайы электрлік қосылуы. Сонымен қатар электр тоғымен зақымданудан қорғанудың әр түрлі электрлік қорғаныс құралдары бар. Қорғаныс құралдары шартты түрде 3 топқа бөлінуі мүмкін: изоляциялайтын, қоршайтын және алдын ала қорғайтын. Изоляциялайтын – адамды тоқ өткізетін немесе жерге қосылған бөліктерден, сонымен қатар жерден изоляциялайды.

#### **4.5 Жұмыс орнына шудың әсері**

Шу адам ағзасына зиянды әсерін тигізе отырып, жұмыс жағдайын нашарлатады. Адамға шу ұзақ уақыт әсер еткен кезде жағымсыз жағдайлар туындайды: көз көру, есту қабілеті нашарлайды, қан қысымы көтеріліп, көңіл бөлу нашарлай түседі. Қатты ұзақ шу жүрек-қан қысымы және жүйке жүйесі қызметінің бұзылуына себебін тигізеді.

Жұмыс орнындағы шу көзі болып техникалық құралдар табылады, олар компьютерлер, серверлер, желдеткіш қондырғылар, сонымен қатар сыртқы шу. Тәжірибеде шамасы 20-30 дБ шу зиянсыз деп есептеледі, ең жоғары шу

мөлшері 80 дБ-ден аспауы тиіс. Ал 130 дБ-ға жеткен кезде адам жайсыз сезінеді.

Шу деңгейін шу өлшегіштермен бақылайды. Ал қарапайым жағдайда сөйлеушіден 1,5 метр жерде тұрып тыңдағанда сөзі анық естілетін – естілмейтіндігін тексереді. Қазақстан өкіметінің территориясында қабылданған регламент бойынша аурухана мен санаторийларда 35 децибел, тұрғын үй кварталдары, класс бөлмелері және оқу аудиториялары үшін – 40, стадиондар мен вокзалдар үшін – 60 децибел. Сонымен бірге транспорт құралдары үшін сыртқы шудың шекті нормасы белгіленген – 82-85 децибел. Бөлмедегі шуды азайту үшін шу көзін жауып, қашықтатып отыруға болады. Шуды көп бөлетін технологиялық құрылғыларды айырбастау мүмкін болмағанда дыбыс шағылғыштар, дыбыс тұтқыштар қолданады. Мысалы, төбені және қабырғаны дыбыс жұтатын затпен қаптап тастаса, шу деңгейін 6-10 дБ төмендетуге болады. Шудың зиянды әсерінен қорғайтын жеке сақтық құралдары да бар.

МЕСТ 12.2.003-76 сәйкес жұмыс орнында және одан тыс жерде пайда болатын шудан қорғану үшін келесі әдістерді қолдану керек:

- шу шығатын көздерде оны азайту;
- ұжымдық және жеке қорғаныс құралдарын қолдану;
- жұмыс бөлмелерін ұтымды жоспарлау және акустикалық өңдеуден өткізу.

Бағдарламалық өнімді жасау кезінде темір жол торабынан, әуежайдан алшақ орналасқан ғимарат ішіндегі жұмыс орны қарастырылған, сондықтан жұмыс процесіне әсер ететін сыртқы шу көздері жоқ.

Пайдаланылған құрал-жабдықтардың сипаттамасы:

1) Персоналды компьютер:

- Intel® Core 2 Duo E7500 (2,93GHz, 3Mb, FSB800), 2048Mb DDR2 800MHz PC-6400, HDD Seagate Barracuda 7200.12 320Gb;

- монитор 19" TFT Samsung SyncMaster 943N;

- габариті: 1600x700x1050 мм (персоналды компьютер + үстел + орындық);

- электр энергиясы: айнымалы кернеу 220-250 В, жиілігі 50 Гц, қуаты 350 Вт;

- саны – 1 дана.

2) Модем:

- D-Link DSL-2540U/BR, ADSL 2+, 4 портты коммутаторы 10/100 Мбит/с;

- саны – 1 дана.

3) Бір терезелік Samsung SMG4590 кондиционері:

- салқындату режимі – 2400 Вт;

- жылыту режимі – 2400 Вт;

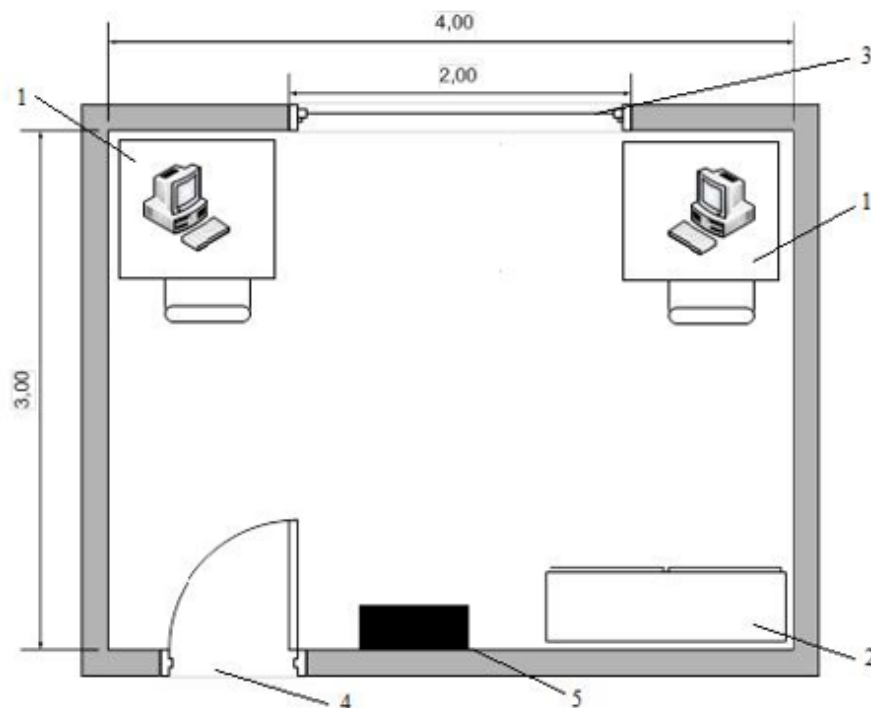
- ағын – 404 м<sup>3</sup>/сағ;

- шу деңгейі – 31 Дб;

- орнату режимі – терезелік;

- габариті - 754x272x176 мм;

- салмағы – 39 кг; - ауданы  $21 \text{ м}^2$  ғимаратқа есептелген.



- мұндағы 1 – операторлардың жұмыс орны;  
 2 – кондиционер орны;  
 3 – терезе жақтауы;  
 4 – есік жақтауы;  
 5 – өрт сөндіргіш балоны.

4.1 сурет– Жұмыс орнының жоспары

#### 4.6 Табиғи жарықтандыру әдісімен есептеу

Ғимараттағы жұмыс орнындағы нормаланған жарықтандыруды құруға қажетті қабырғадағы жарық ойықтардың ауданын есептеу.

Ғимараттың өлшемі: ұзындығы  $a = 4 \text{ м}$ , ені  $b = 4 \text{ м}$ , биіктігі  $h = 3 \text{ м}$ . Жұмыс аймағының еден деңгейімен салыстырғандағы биіктігі –  $0,7 \text{ м}$ , терезе  $0,8 \text{ м}$  биіктіктен басталады, терезе биіктігі  $2 \text{ м}$ . Жұмыс ғимараты IV сағаттық белдеуде орналасқан. Барлық жағында да көлеңкелейтін ғимараттар жоқ.

Жұмыс орны терезелік ойықтарды жоспарлап отырған ғимараттың сыртқы қабырғасынан  $0,5 \text{ м}$  қашықтықта. Минималды жарықтандыру терезелік ойықтан  $2,5 \text{ м}$  қашықтықтағы нүктеде болады.

Терезенің жалпы қажетті ауданын  $S_0, \text{ м}^2$  мына формуламен анықтаймыз:

$$S_0 = \frac{S_n \cdot e_n \cdot \eta_0 \cdot k_{зд} \cdot k_3}{100 \cdot \tau_0 \cdot r_1}, \quad (4.1)$$

мұндағы  $S_n$  – ғимарат ауданы,  $m^2$ ;

$k_3$  – қор коэффициенті;

$k_3 = 1,2$  (оқу ғимараттары, зертханалар, конструкторлық бюро);

$e_n$  – табиғи жарықтандыру коэффициентінің нормаланған мәні;

$$S_n = a \cdot b = 4 \cdot 3 = 12 (m^2).$$

$$e_n^{IV} = e_n \cdot m \cdot c, \quad (4.2)$$

мұндағы  $m = 0,8$ ;

$c = 0,75$  – IV сағаттық белдеуге арналған.

$e_n = 1,2$  V разрядтың орта дәлдікті жұмысына арналған;

$$e_n^{IV} = 1,2 \cdot 0,8 \cdot 0,75 = 0,72.$$

4.2 кесте –  $m$  коэффициентінің мәндері (ҚР ҚНЖЕ 2.04-05-2002)

IV 50° Солтүстік ендіктен және оңтүстікке қарай	Ғимараттың сыртқы қабырғаларында	Тікбұрышты және трапеция пішінді шамдарда	Шед түріндегі шамдарда	Зенит шамдарда
	0.75	0.8	0.8	0.7

$\tau_0$  – жарық жіберудің ортақ коэффициенті тең:

$$\tau_0 = \tau_1 \cdot \tau_2 \cdot \tau_3 \cdot \tau_4$$

Жарық өткізгіш материал ретінде мыналарды пайдаланамыз: терезелік шыны,  
екі есе:

$$\tau_1 = 0,8 ;$$

тыстың түрі – екі еселенген бөлінген:

$$\tau_2 = 0,6 ;$$

салмақ түсіретін конструкцияның түрі – темір бетонды формалар:

$$\tau_3 = 0,8 ;$$

күннен қорғайтын құралдар – жиналатын реттелетін жалюзде:

$$\tau_4 = 1 ;$$

жарық жіберудің ортақ коэффициенті:

$$\tau_0 = 0,8 \cdot 0,6 \cdot 0,8 \cdot 1 = 0,384 ;$$

$\eta_0$  – терезелердің жарықтық сипаттамасы.

Бөлме ұзындығының терезеден алшағырақ орналасқан нүктесінің дәл ортасына қатынасы  $\frac{4}{2,5} = 1,6$  тең. Ғимараттың енінің жұмыс аймағынан терезенің жоғарғы жағына дейінгі биіктігіне қатынасы  $\frac{3}{2,1} = 1,43$  тең. Осыдан

$\eta_0=10,5$ ,  $\Gamma_1$ – ғимараттың жоғары жағынан және ғимаратқа қосылған астына төселген қабаттан шағылысқан жарықтың әсерінен болатын қабырғалық жарықтану кезіндегі ТЖК-нің жоғарылауын ескеретін коэффициент;  $\frac{a}{b} = \frac{4}{3}=1.333$  қатынасы, ғимараттағы шағылысудың орташа коэффициенті

$\rho_{\text{ср}}=0,5$ , бір жақты қабырғалық жарықтануды аламыз. Сонда  $\Gamma_1=1,3$ .  $k_{\text{зд}}$ - ғимаратқа қарсы тұрған терезенің көлеңкелеуін ескеретін коэффициент. Жақын маңда көлеңкелейтін ғимарат жоқ болғандықтан,  $k_{\text{зд}} = 1$ .

Терезенің жалпы ауданын есептейміз:

$$S_0 = \frac{12 \cdot 0,72 \cdot 10,5 \cdot 1,2}{100 \cdot 0,384 \cdot 1,3} = 2,18 (\text{м}^2).$$

Бөлмедегі терезенің жалпы ауданы  $2 \times 2 = 4 \text{ м}^2$  болғандықтан, олар жұмыс ғимаратының табиғи жарықтану нормативіне сәйкес келеді.

#### 4.7 Пайдалану коэффициенті әдісімен жасанды жарықтануды есептеу

- көру жұмысының разряды – V;
- нормаланған жарықтану – 400 лк.

4.3 кестеде көрсетілгендей параметрлері бар люминесцентті шамдармен ЛБ жалпы жарықтану жүйесін пайдаланамыз.

#### 4.3 к е с т е – Газоразрядты лампалардың ЛБ техникалық сипаттамалары

Номиналды қуат, Вт	ЛБ типті лампалардың номиналды жарықтық ағыны, лм	Лампалардың өлшемі, мм	
		Диаметрі	Ұзындығы
40	3120	40	1213,6

Жұмыс аймағының үстінен шамның іліну биіктігін есептейміз:

$$H = h - h_p - h_c, \quad (4.4)$$

мұндағы  $h_c$  – шамнан үстіңгі жабынға дейінгі қашықтық,  $h_c = 0,05 \text{ м}$ ;  
 $h_p$  – еден үстінен жұмыс аймағының биіктігі,  $h_p = 0,7 \text{ м}$ ;

$h$  – ғимарат биіктігі,  $h = 3$  м;

$$H = 3 - 0,7 - 0,05 = 2,25(м).$$

Шамдар арасындағы тиімдірек қашықтық былай анықталады:

$$L = \lambda \cdot H ,$$

$$L = 1,2 \cdot 2,25 = 2,7 (м).$$

мұндағы  $\lambda = 1,2 \div 1,4$ ;

Жұмыс қабырға жанында жүргізілмесе, қабырғадан жақын тұрған шамға дейінгі қашықтықты мына формуламен анықтаймыз:

$$l_1 = (0,4 \div 0,5) \cdot L ,$$

$$l_1 = 0,4 \cdot 2,7 = 1,08 (м).$$

Ғимарат индексін анықтаймыз:

$$i = \frac{a \cdot b}{H \cdot (a + b)} ,$$

$$i = \frac{4 \cdot 3}{2,25 \cdot (4 + 3)} = 0,76 .$$

Қабырғалардың төбесінен және еденнен шағылысу коэффициенттері сәйкес:

$$\begin{aligned} - \rho_{ТОБЕ} &= 70\% , \\ - \rho_{КАБ} &= 50\% ; \\ - \rho_{ЕДЕН} &= 30\% . \end{aligned}$$

Пайдалану коэффициенті бұл жағдайда  $\eta = 40\%$  , қор коэффициенті  $k_K = 1,2$  . Люминесцентті лампалардың санын мына формуламен анықтаймыз:

$$N = \frac{E \cdot k_K \cdot SOC \cdot Z}{n \cdot \Phi_L \cdot \eta} \quad (4.5)$$

мұндағы SOC – ғимарат ауданы;

$k_K$  – қор коэффициенті,  $k_K = 1,2$ ;

$E$  – берілген минималды жарықтану,  $E = 400$  лк.;

$Z$  – жарықтанудың бірқалыпсыздық коэффициенті,  $Z = 1,1$ ;  $n$  – шамдағы лампалар саны;

Фл – таңдалған лампаның жарық ағыны,  
 Фл=3120 лм;  $\eta$  – пайдалану коэффициенті;  
 $\eta = 40\%$ ;

$$N = \frac{400 \cdot 1,2 \cdot 12 \cdot 1,1}{2 \cdot 3120 \cdot 0,4} = 2,54 \approx 3$$

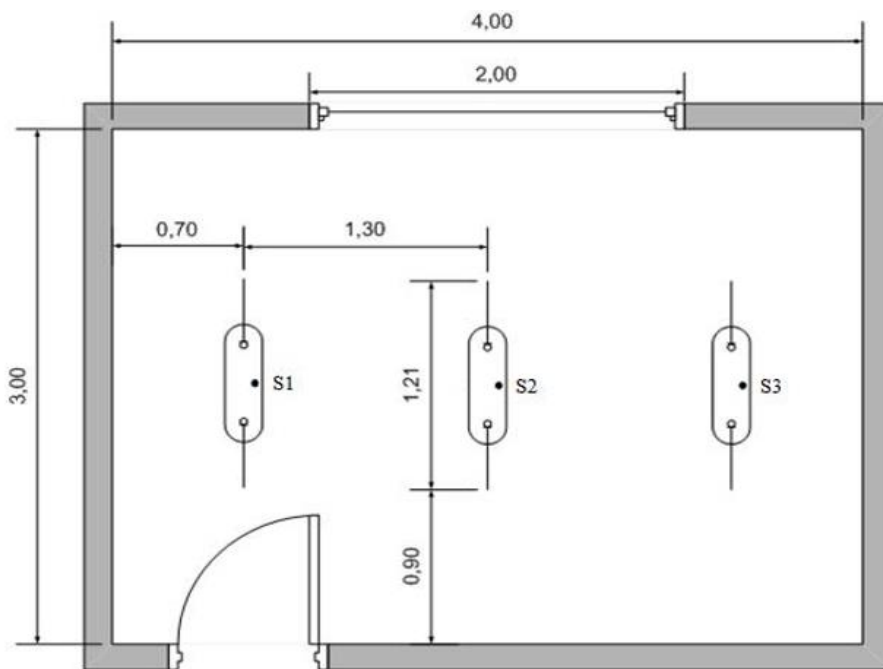
Нормаланған жарықтануды 400 лк құру үшін барлығы ЛБ сериялы 3 шамда орналасқан 6 люминесцентті лампа қажет, әр лампаның қуаты шындыққа сәйкес келетін 40 Вт-тан аспауы қажет, демек қазіргі алынған жарықтану санитарлық нормаға сәйкес келу үшін жеткілікті.

#### 4.8 Жасанды жарықтануды нүктелік әдіспен есептеу

Жасанды жарықтануды есептеудің пайдалы коэффициент әдісімен қатар нүктелік әдіспен де есептелуін қарастырамыз.

- көру жұмысының разряды – V;
- нормаланған жарықтану – 400 лк.

Жарықтанатын аймақтың үстінен шамның іліну биіктігі  $H = 2,25$  м, қор коэффициенті  $k_3 = 1,2$ . Жарықтану схемасы 4.2 суретте көрсетілген.



4.2 сурет– Жарықтану схемасы

Соңғы нүкте А-ны белгілейміз. Ол үшін барлық шамдардың шартты жарықтануының қосындысын келесі түрде анықтаймыз:

Төбедегі А нүктесінен шамға дейінгі қашықтықтың проекциясын табамыз - d. Содан соң төбе мен d түзуінің арасындағы бұрышты анықтаймыз. Бұл бұрыш бойынша шартты жарықтануды табамыз.



#### 4.4 кесте– Шамдардың жарықтық бөлінуі

Шам түрі	Жарық күші $I_\alpha$ , кд $\alpha$ бұрышының бағытына										
	0	5	15	25	35	45	55	65	75	85	90
ПВЛМ-1x40	139	135	132	115	104	84	63	44	22	6	0

Соңғы нүкте №2 лампаның астында болсын.  $d_i$  – соңғы нүктеден  $i$ -ші лампаға дейінгі қашықтық деп көрсетеміз.

$$\alpha = \arctg \left( \frac{d_i}{H} \right); e_i = \frac{I \cdot \cos^3(\alpha)}{H^2}$$

1-ші лампа  $d_2 = 0$  м ;

$$\alpha = 0^0;$$

$$e_i = \frac{139 \cdot 2 \cdot \cos^3(0)}{2,25^2} = 54,9$$

1, 3-ші лампалар  $d_1 = d_3 = 1,3$  м;

$$\alpha = 30^0;$$

$$e_i = \frac{110 \cdot 2 \cdot \cos^3(30)}{2,25^2} = 28,22$$

Шартты жарықтанудың қосындысы тең:

$$\sum E = 54,9 + 2 \cdot 28,22 = 111,34$$

Жарықтанудың қосындысы тең:

$$E_{AG} = \frac{\mu \cdot F_{\text{л}}}{1000 \cdot K} \cdot \sum_3 E = \frac{1,1 \cdot 2 \cdot 3120}{1000 \cdot 1,2} \cdot 111,34 \approx 640$$

#### 4.9 «Тіршілік қауіпсіздігі» бөлімі бойынша қорытынды

Бұл бөлімде берілген офистік ғимараттағы еңбек шартының талдауы, оның ішінде табиғи және жасанды жарықтанудың есептеулері келтірілген. Еңбек шартының деңгейі мүмкін деп танылды және есептеулерден алынған

мәліметтер өмір тіршілігі қауіпсіздігі стандарттарының талаптарын қанағаттандырады.

Есептеулер табиғи жарықтандыру үшін ауданы  $4 \text{ м}^2$  болатын бір ғана терезе жеткілікті болатынын көрсетті. Қажетті жарықтануды қамтамасыз ету үшін күндізгі уақытта жасанды жарықтандыру пайдаланылады. Жұмыс орнын жарықтандыру үшін әрбір шамда әрқайсысының сәулеленуінің жарық ағыны 3120 лм болатын 3 шам арқылы жалпы жарықтандыру жеткілікті, сондықтан бұл ғимаратта жарық емес тәулік уақытында да жұмыс жасауға болады.

Сонымен қатар қорытынды ретінде келесілерді атап өтуге болады:

Нүктелік әдіспен есептеу номиналды жарықтандыру деңгейінде есептеу талдауын жасауға мүмкіндік береді, және бұл әдістің басты кемшілігі шамдардың қаншалықты тиімді қолданылатындығын айта алмаймыз.

Пайдалану коэффициенті әдісімен есептеу кері көріністі береді. Ол сол немесе басқа шамдарды пайдалану қаншалықты тиімді және үнемді екенін анықтауға мүмкіндік береді, яғни номиналды қуатты анықтауға мүмкіндік береді.

## **5 Бизнес-жоспар**

### **5.1 Жоба сипаттамасы**

Қазіргі кезеңде адамзат баласының барлық қызмет салаларында ғылыми-техникалық прогрестің дамуы және сонымен қатар ғылыми-зерттеу жобалары белгілі бір көлемде есептеу техникасының енуіне тәуелді.

Компьютер базасында құрылған ақпараттық жүйелер үлкен көлемді ақпараттарды өңдеуге және ауқымды ақпараттарды өңдеуді қажет ететін салаларда басқаруды шешуге рұқсат етеді. Енді бұл өз кезегінде бағдарламалық қамтамасыз етуді құруды қажет етеді. Дипломдық жобаның негізгі мақсаты Оңтүстік Қазақстан облысындағы “Досжанов Сабыр” жеке кәсіпкерлігінің деректер қорын құру қарастырылады. Ақпараттық жүйені NOSQL Server ДҚБЖ (дерекқорларды басқару жүйесінде) жасалған дерекқор және Visual Basic for Applications (VBA), Java бағдарламалық ортасында құрылған клиенттік қосымша құрайды. Жұмыстың нәтижесі кәсіпорындағы құрылымдық өзгерістерді реттеу үшін пайдаланылады. Сонымен қатар, деректердің қауіпсіздік пен құпиялылық мәселелері қарастырылған.

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. Сондықтан еңбек етудің негізгі құралы болып – дерекқор орналасқан компьютер есептелінеді. Біз жұмыс орны ретінде жеке кәсіпорынның бөлмесін қарастырамыз. Қарастырылып отырған бөлмеде екі қызметкер жұмыс істейді, олардың әрқайсысының өз жұмыс орны бар.

Қазіргі таңда, Қазақстандағы ақпараттық технологиялардың дамуына байланысты кез келген оқу орны және кәсіпорындардың құжаттамасы және

құжатайналымы қағаз түрінен электрондық түрге ауыса бастады. Электрондық құжаттардың көбісі ресми түрге ауысып, физикалық түпнұсқаларын алмастыруда. Мұндай өзгерістердің құндылығы зор: ұйымдастыру, көшіру және өзгерту техникаларының оңайлылығына қоса, қазіргі заманғы коммуникация желілерін қосатын болсақ, құжатайналым процессінің жылдамдығы артады.

## **5.2 Маркетингтік стратегия**

Маркетингтік стратегияның мақсаты – бағдарламалық қамтамасыз ету арқылы пайда табу.

### **5.1 Кесте – Жоспарланып отырған пайда мың теңге**

Жылдар	2016	2017	2018	Барлығы
Жоспарлаған табыс, барлығы мың теңге	422184,357	844368,714	1266553,07	2533106,141

Бағдарламаның сатылым жоспары.

Біздің стратегия Оңтүстік Қазақстан облысындағы “Досжанов Сабыр” жеке кәсіпкерлігінің деректер қорын құру қарастырылады. Деректер қорын жобалауға арналған бұл бағдарлама жеке кәсіпкерліктің ақпараттармен жұмыс жасауын жақсартатын бағдарлама. Бағдарлама деректер қорына қызмет түрлерін енгізу, олар жайлы мәліметтер алу, жұмыс барысын бақылау, қызметкерлерді енгізу және т.б мәліметтерді алып отыруға мүмкіндік береді. Қазіргі таңда жеке кәсіпкерлік жұмысын басқаруға қажетті бағдарлама болып табылады. Бағдарлама жеке кәсіпкерліктің жұмысын ұтымды, тиімді және жеңіл етеді. Интернет желісінде де қолданылатын болады. Бағдарламаны қолдану арқылы қызметкердің уақытын ұтымды пайдалануға болады, ақпараттарды оңай ала-аласыз, деректерді енгізудегі кателіктердің алдын аласыз. Бұл бағдарлама арқылы жеке кәсіпкерліктің жұмысын бір орталықтан басқаруға мүмкіндік болады.

Бұл бағдарлама арнайы тапсырыс арқылы жасалғандықтан бірінші жылы тауарлық белгіні кең түрде тарату және тұрақты тұтынушылардың санын арттыру мақсатында 19742688 мың қаржы жұмсауды жоспарлап отырмыз.

## **5.3 Бағдарламамен қамтамасыз етудегі еңбек сыйымдылығын есептеу**

Еңбек шығыны құрамдасын есептеудегі базалық көрсеткіш мына формуламен есептелінеді:

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

мұндағы  $Q$  – шартты командалар саны;

$q$  – есеп түріне қарай шартты командалар санын ескеретін коэффициент;

$c$  – бағдарламаның қиындығы мен жаңалығын ескеретін коэффициент.

$$Q = q * c = 1490 * 1.26 = 1877.$$

Атап өткен  $q$  коэффициентінің мәнін А1 кестеден (А қосымшасынан) тандап алуға болады.

Атап өткен «с» коэффициенті А2 кестеден (А қосымшасынан) анықталады, ол күрделік тобы бағанасы мен жаңалықтық дәрежесі бағанасының қиылысуы.

Бағдарламаның күрделілік коэффициенті  $c = 1,26$ . Осы әдіспен базалық көрсеткішті табамыз:  $Q = 1877$  Ары қарай бағдарламалық өнімді әзірлеуге кететін уақытты есептеу керек.

Жалпы бағдарлама жасауға кеткен уақыт әртүрлі компоненттен тұрады. Бағдарламалық өнімді жасауға кеткен жалпы уақыт құрамы А3 кестеде көрсетілген.

Уақыт адам-сағатпен есептеледі, ал  $T_d$  нақты істелген уақытпен алынады, ал қалған кезеңнің уақыты  $Q$  командасының шартты санына байланысты есептік жолмен анықталады,

Бағдарламалық өнімін дайындауға кеткен әр кезеңнің уақытын анықтаймыз:

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

$$T_{ПО} = 24 \text{ (адам / сағ.)}$$

2)  $T_O$  (мақсат сипаттамасы уақыты) келесі формуламен анықталады:

$$T_O = Q \times B / (50 \times K), \quad (5.2)$$

$$T_O = Q \times B / (50 \times K) = 1877 * 1,2 / (50 * 0,8) = 56,31 \text{ (адам / сағ.)}$$

мұндағы  $B$  – мақсат есебі өзгерісінің коэффициенті,  $B$  коэффициенті мақсат күрделігіне және өзгеріс санына тәуелді – 1,2-ден 1,5-ке дейін (А2-кестені қара).

$K$  – бағдарлама жасаушы білектілігін ескеретін коэффициент.

$K$  коэффициенті мәнін 4-кестеден (А қосымшасынан) алуға болады.

3)  $T_A$  (алгоритм құруға кеткен уақыт) мына формуламен есептейміз:

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

$$T_A = Q / (50 \times K) = 1877 / (50 \times 0.8) = 46,9 \text{ (адам / сағ)}.$$

4)  $T_{BC}$  (блок – сұлба құруға кеткен уақыт)  $T_A$  сияқты 3 формуламен есептеледі.

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

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

$$T_H = Q \times 1,5 / (50 \times K) = 1877 \times 1.5 / (50 \times 0.8) = 70,39 \text{ (адам / сағ)}.$$

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

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

$$T_{II} = Q / 50 = 1877 / 50 = 37,54 \text{ (адам / сағ)}.$$

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

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

$$T_{OT} = Q \times 4,2 / 50 \times K = 1877 \times 4.2 / 50 \times 0.8 = 197,085 \text{ (адам / сағ)}.$$

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

$$T_D = 24 \text{ (адам / сағ)}.$$

Еңбек шығындарының сомасы еңбек шығынының құрама сомасы ретінде 7 формуламен есептеледі:

$$T = T_{PO} + T_{TO} + T_A + T_{BC} + T_H + T_{II} + T_{OT} + T_D, \quad (5.7)$$

$$T = T_{PO} + T_{TO} + T_A + T_{BC} + T_H + T_{II} + T_{OT} + T_D = 24 + 56,31 + 46,9 + 46,9 + 70,39 + 37,54 + 197,085 + 24 = 503,125 \text{ (адам / сағ)}.$$

#### **5.4 Бағдарламалық қамсыздандыру шығынының есебі**

Бағдарламалық қамсыздандыру шығыны ішіне еңбек ақы шығыны да, еңбек ақыдан аударылымдар, амортизациялық және тағы да басқа шығындар кіреді, олар мынандай формуламен анықталады:

$$: \Phi OT + O_{CH} + A + C_{ЭЭ} + C_{МЖК} + C_{ТО} + C_{ПР} + C_N, \quad (5.8)$$

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

$$\Phi OT = Z_{осн} + Z_{доп}, \quad (5.9)$$

Негізгі еңбек ақы төмендегідей анықталады:

$$Z_{осн} = T \times TC / t_{орт}, \quad (5.10)$$

$$Z_{осн} = T \times TC / t_{орт} = 503,125 \times 2800 \times 2,25 / 21 = 150937,5 \text{ (тенге)}.$$

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

$$Z_{доп} = 0,2 \cdot Z_{осн}, \quad (5.11)$$

$$Z_{доп} = 0,2 \times Z_{осн} = 0,2 \times 150937,5 = 30187,5 \text{ (тенге)}.$$

Жалпы еңбек ақысы (еңбек төлеу фонды) негізгі және қосымша еңбек ақының қосындысы ретінде анықталады (9):

$$\Phi OT = Z_{осн} + Z_{доп},$$

$$\Phi OT = 150937,5 + 30187,5 = 181125 \text{ (тенге)}.$$

Әлеуметтік салық ЕАҚ 11 % құрайды (ҚР СК 358 б. 1-тарау) жұмыскердің табысынан, мынандай формуламен есептеледі [1] (А қосымшасы)

$$O_{CH} = (\Phi OT - 3A) \times 11\%, \quad (5.12)$$

$$O_{CH} = (\Phi OT - 3A) \times 11\% = (181125 - 18112,5) \times 0,11 = 17931,38 \text{ (тенге)}.$$

мұндағы 3А – зейнетақы аударылымдар, ЕАҚ-нан 10% құрайды және әлеуметтік салықпен міндеттелмейді [2]:

$$3A = EAK \times 10\%, \quad (5.13)$$

$$3A = EAK * 10\% = 181125 * 0,1 = 18112,5 \text{ (тенге).}$$

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

$$A = \frac{B_{\text{бас}} \times A_{\text{ш}} \times N}{100 \times 12 \times t}, \quad (5.14)$$

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

$$H_A = \frac{B_{\text{бас}} - K_{\text{тар}}}{T_{\text{норм}} \cdot B_{\text{бас}}} \cdot 100\%, \quad (5.15)$$

$$H_A = \frac{B_{\text{бас}} - K_{\text{тар}}}{T_{\text{норм}} \cdot B_{\text{бас}}} \cdot 100\% = \frac{88110 - 4406}{4 \cdot 88110} \cdot 100\% = 23,75\%.$$

мұндағы  $K_{\text{тар}}$  – таратылым құны, жабдықтың құнынан 5% құрайды (нұсқа бойынша);

$K_{\text{тар}} = 88110 * 0,05 = 4406$  тенге;

Құны  $B_{\text{бас}} = 88110$  тенге;

Компьютердің қазіргі уақыттағы құны 65490 тенге [3];

Принтердің құны 22620 тенге [4];

Дербес компьютерде жалпы жұмыс істеу уақыты мына формуламен есептеледі:

$$T = T_A + T_{\text{БС}} + T_H + T_{\text{П}} + T_{\text{ОТ}}, \quad (5.16)$$

$$T = T_A + T_{\text{БС}} + T_H + T_{\text{П}} + T_{\text{ОТ}} = 46,9 + 46,9 + 70,39 + 37,54 + 197,085 = 398,815 \text{ (адам / сағ).}$$

Амортизациялық аударылымдар:

$$A = \frac{88110 \cdot 0,2375 \cdot 21,87}{100 \cdot 12 \cdot 398,815} = 0,9563.$$

Электрэнергия шығындары мына формуламен есептеледі:

$$C_{\text{ЭЭ}} = K \times k_3 \times T \times C_{\text{кВт-сағ}}, \quad (5.17)$$

$$C_{\text{ЭЭ}} = K \times k_3 \times T \times C_{\text{кВт-сағ}} = 450 * 0,8 * 19,37 * 8 = 55,786 \text{ (тенге).}$$

1 кВт-сағ электрэнергиясының құны 19,37 тенге [5]

Материалдар мен көмекші бөлшектер шығыны, бағдарламалық өнімді жазу барысында қолданылды ( $C_{МЖК}$ ), сонымен қатар техникалық қызмет көрсету шығыны ( $C_{ТО}$ ), жабдықтың құнынан 1.5% және 2.5% құрайды және мына формулалар мен есептеледі (18 – 19):

$$C_{МЖК} = 0,015 \times C_{обор}, \quad (5.18)$$

$$C_{МЖК} = 0,015 \times C_{обор} = 0,015 \times 88110 = 1321,7 \text{ (тенге)}.$$

$$C_{ТО} = 0,025 \times C_{обор}, \quad (5.19)$$

$$C_{ТО} = 0,025 \times C_{обор} = 0,025 \times 88110 = 2202,8 \text{ (тенге)}$$

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:

$$C_H = 0,5 \times EAK, \quad (5.20)$$

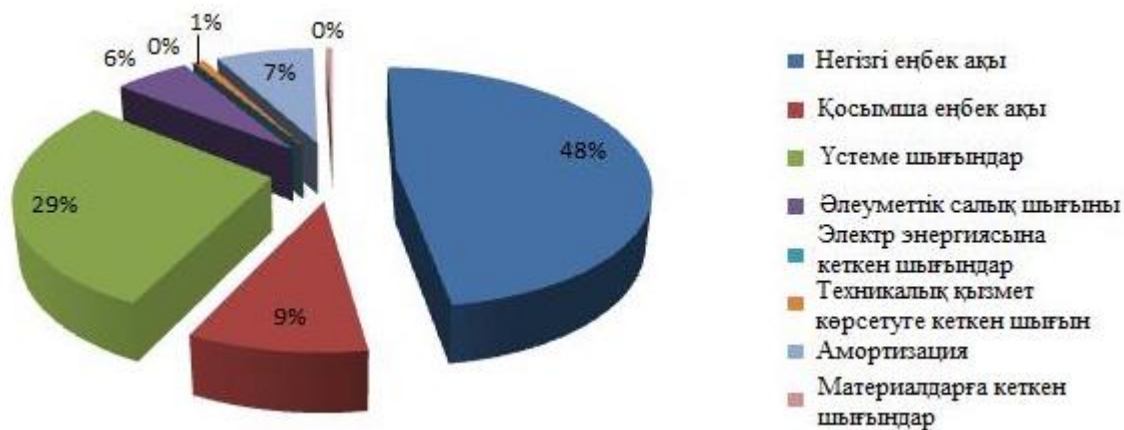
$$C_H = 0,5 \times EAK = 0,5 \times 181125 = 90562,5 \text{ (тенге)}.$$

Бағдарламалық өнімнің өзіндік құнының есебінің жиынтық нәтижелерін кесте түрінде ұсыну керек, шығын статьясын атап, және оның ортақ құндағы сыбағаларын пайызбен есептеп, бағдарламалық қамсыздандырудың өзіндік құнының құрылымының диаграммасын сызу керек.

5.2 кесте– Өзіндік құнның қорытынды кестесі

Шығын бабы атауы		Сомасы, теңге	Әр баптың үлесі, %
ЕАҚ	ЕА <sub>нег</sub>	150937,5	48,0501
	ЕА <sub>қос</sub>	30187,5	9,61
Үстеме шығындар,	С <sub>накл</sub>	90562,5	28,8301
Әлеуметтік салық шығыны,	ӘС	17931,38	5,70835
Пайдалану шығындары	С <sub>ээ</sub>	55,786	0,0178
	С <sub>то</sub>	2202,8	0,70125
	А <sub>жыл</sub>	20926,1	6,6617
Материалдар және көмекші, $C_{МЖК}$		1321,7	0,9938
Барлығы:		314125,266	100





5.1 сурет– Бағдарламалық өнімнің өзіндік құнының есебінің жиынтық нәтижелері

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

$$Ц = C + П, \quad (5.21)$$

мұндағы  $C$  – өнім бағасы,  
 $П$  – таза кіріс.

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

$$Ц_{п} = C \cdot \left(1 + \frac{P}{100}\right), \quad (5.22)$$

мұндағы  $P$  – рентабельдік (20%).

$$Ц_{п} = 314125,266 \cdot (1 + 20/100) = 376950,319 \text{ (теңге)}.$$

Бағдарламаның орындалу бағасы келесі формула арқылы табылады.

$$Ц_p = Ц_{п} + НДС, \quad (5.23)$$

Қазіргі таңда ҚР-да НДС 12% құрайды:

$$НДС = Ц_{п} \cdot 12\%, \quad (5.24)$$

$$НДС = 376950,319 \cdot 12\% = 45234,038 \text{ (теңге)},$$

$$\Pi_p = 376950,319 + 45234,038 = 422184,357 \text{ (теңге)}.$$

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

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

$$\Sigma Z = C_C + C_{TP} + C_O, \quad (5.25)$$

Бағдарламаның бағасы:  $C_C = 422184,357$  теңге ,

$C_{TP}$  – көлік шығыны, жүйе құнынан - 25 %, мың теңге.

$$C_{TP} = 0,25 * C_C,$$

$$C_{TP} = 422184,357 * 0,25 = 105546,089 \text{ (теңге)}.$$

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

$$C_O = T \times C_{OP}, \quad (5.26)$$

$$C_O = T \times C_{OP} = 4 * 8 * 2600 = 83200 \text{ (теңге)}.$$

мұндағы  $T$  – оқытуға кеткен уақыт, сағ;

$C_{OP} = 2600$  теңге. Себебі қазіргі танда сағатына 2500-3000 теңге аралығында.

Бағдарлама өнімін сатып алуға кеткен бір жолғы шығындар есебін А6 кестеге келтіру керек.

$$\Sigma Z = C_C + C_{TP} + C_O,$$

$$\Sigma Z = 422184,357 + 105546,089 + 83200 = 610930,446 \text{ (теңге)}.$$

5.3 кесте – Ақпарат жүйелерін енгізуге керекті бір жолғы шығындар есебінің жиынтығы

Шығын бабы атауы	Сомасы, мың теңге
Жүйенің құны	422184,357
Көлік шығыны	105546,089
Жүйені оқуға кеткен шығыны	83200
Барлығы:	610930,446

## 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 барлығын енді формула арқылы жазайық:

$$C_{\Sigma} = 3П + O_{CH} + Пр, \quad (5.27)$$

$$C_{\Sigma} = 3П + O_{CH} + Пр = 14112000 + 1397088 + 4233600 = 19742688 \text{ (тенге).}$$

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

$$3П = (O_C \times t \times K_P) \times Ч \times (1 + K_D), \quad (5.28)$$

$$3П = (3000 * 8 * 245) * 2 * (1 + 0,2) = 14112000 \text{ (тенге),}$$

$$O_{CH} = (ФОТ - 3А) \times 11\%,$$

$$O_{CH} = (14112000 - 1411200) * 0,11 = 1397088 \text{ (тенге),}$$

$$3А = ЕАҚ * 10\%,$$

$$3А = 14112000 * 0,1 = 1411200 \text{ (тенге).}$$

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

$$Пр = 3П \times 0,3, \quad (5.29)$$

$$Пр = 3П \times 0,3 = 14112000 * 0,3 = 4233600 \text{ (тенге).}$$

Ақпараттық технологияларын қолданған кездегі жылдық бір жолғы шығындарды келесі кестеге сомасын және әр баптарын көрсетіп толтырыңыз (А7 кестені қара).



5.4 кесте– 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

Еңбек өнімділігі өсуінен түскен үнемділікті келтірінді шығындардың базалық ( $C_0$ ) және ұсынылған ( $C_1$ ) нұсқалар айырмасы ретінде шығарып алуға болады.

$$\Delta_{yt} = C_0 - C_1, \quad (5.30)$$

Ұйымда жүйе енгізілмей тұрғанда (қол еңбегімен айналысқанда) жылдық келтірінді шығындар келесі баптардан тұрады:

- еңбек ақы қоры;
- әлеуметтік салық;
- басқада шығындар.

Осының барлығын енді формула арқылы жазайық:

$$C_{\Sigma} = 3П + O_{CH} + П_r, \quad (5.31)$$

$$C_{\Sigma} = 3П + O_{CH} + П_r = 14112000 + 1397088 + 4233600 = 19742688 \text{ (теңге)}.$$

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

$$3П = (O_C \times t \times K_p) \times Ч \times (1 + K_d), \quad (5.32)$$

$$3П = (O_C \times t \times K_p) \times Ч \times (1 + K_d) = (3000 \times 8 \times 245) \times 3 \times (1 + 0,2) = 21168000 \text{ (теңге)},$$

$$O_{CH} = (ФОТ - 3А) \times 11\%,$$

$$O_{CH} = (21168000 - 2116800) \times 0,11 = 2095632 \text{ (теңге)},$$

$$3A = EAK * 10\%,$$

$$3A = 21168000 * 0,1 = 2116800 \text{ (тенге).}$$

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

$$Pr = 3П \times 0,3, \quad (5.33)$$

$$Pr = 3П \times 0,3 = 21168000 * 0,3 = 6350400 \text{ (тенге),}$$

$$\mathcal{E}_{yr} = 29614032 - 19742688 = 9871344 \text{ (тенге).}$$

5.5 кесте– Ақпараттық технологияларын қолданған кездегі жылдық бір жолғы шығындар

Шығын баптары атауы	Сомасы, мың теңге
Жылдық ЕАҚ	21168000
Әлеуметтік салық аударым	2095632
Басқа да шығындар	6350400
Барлығы:	29614032

## 5.8 Салыстырмалы экономикалық тиімділіктің көрсеткіштерін есептеу

Нормативтік күрделі қаржы салымының экономикалық тиімділігінің коэффициенті келесі формуламен есептеледі:

$$E_n = \frac{1}{T_n}, \quad (5.34)$$

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

мұндағы  $T_n$  – нормативтік күрделі қаржы салымын өтелу мерзімі, жыл. ( $T_n = 1, 2, 3 \dots n$ ) бағдарлама өнімдерінің өтеу мерзімі 3 жыл.

Есептік күрделі қаржы салымының экономикалық тиімділігінің коэффициенті:

$$E_p = \frac{\mathcal{E}_{yr}}{K}, \quad (5.35)$$

$$E_p = \frac{\Xi_{yr}}{K} = \frac{9871344}{19742688} = 0,5.$$

Есептік күрделі қаржы салымын өтелу мерзімі:

$$T_p = \frac{1}{E_p}, \quad (5.36)$$

$$T_p = \frac{1}{E_p} = \frac{1}{0,5} = 2.$$

Дисконтты есептемегенде өтелу мерзімі 2 жыл.

5.6 кесте – Бағдарлама өнімін енгізудің салыстырмалы экономикалық тиімділігінің көрсеткіштері

Көрсеткіштер атауы	Мәні
Шартты жылдық шығынды үнемдеу	9871344
Күрделі қаржы салымының экономикалық тиімділігінің коэффициенті( $E_p$ )	0,5
Күрделі қаржы салымын өтелу мерзімі( $T_p$ )	2

## 5.9 Динамикалық көрсеткіштер негізінде жобаны өткізуде экономикалық тиімділігін бағалау

Таза дисконттық табыс (ЧДД) келесі формуламен есептеледі:

$$\text{ЧДД} = \sum_{t=0}^T (P_t - 3_t) \times \frac{1}{(1+E)^t} - K, \quad (5.37)$$

$$\Xi_t = (P_t - 3_t) = \Xi_{yr} = 9871344 \text{ (мың теңге)}.$$

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

$$T_{OK} = t + \frac{K - (d\Pi + d\Pi)}{d\Pi_3},$$

$$T_{OK}=2+\frac{19742688-15081220}{5712583}=2,816=2 \text{ жыл } 10 \text{ ай.}$$

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

$E$  – тұрақты дисконт мөлшері, 20 %.[6]

Онда барлық есеп шегін ескергендегі таза дисконттық табыс сомасы:

$$\text{ЧДД}=\mathfrak{A}_1\frac{1}{1+E}+\mathfrak{A}_2\frac{1}{(1+E)^2}+\mathfrak{A}_3\frac{1}{(1+E)^3}-K,$$

$$\text{ЧДД}=\frac{9871344}{1,2}+\frac{9871344}{1,2^2}+\frac{9871344}{1,2^3}-1974688=1051115,33.$$

Таза дисконттық табыстың оң мәні  $\text{ЧДД} > 0$ , инвестицияның тиімділігін көрсетеді, ақпараттық жүйелер керекті көлемде пайда алып келеді.

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

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

мұндағы  $K$  – күрделі қаржы салымы немесе инвестицияның құны.

$$\text{ИД}=\frac{20793803,33}{19742688}=1,05.$$

Егер  $\text{ИД} > 1$ , жоба тиімді, егер  $\text{ИД} < 1$  – тиімді емес.

Ішкі табыстық мөлшері (ВНД немесе IRR) - келтірілген әсер, келтірілген күрделі қаржы салымдарға тең болғандағы дисконттау мөлшерін ( $r_{\text{ВН}}$ ) көрсетеді.

Бұл көрсеткіш  $r_{\text{ВН}}$  (ВНД) келесі теңдеуден шығады:

$$\sum_{t=0}^T \frac{P_t - \mathfrak{Z}_t}{(1+r_{\text{ВН}})^t} = \sum_{t=0}^T \frac{K}{(1+r_{\text{ВН}})^t} \quad (5.38)$$

Егер  $r_{\text{ВН}}$  мәні инвесторға керек табыс мөлшері салынған капиталдан көп не Ішкі табыстық мөлшері (ВНД):

егер  $E1 \rightarrow \text{ЧДД}1 > 0$ ;

$E2 \rightarrow \text{ЧДД}2 < 0$ ;



$$BHD = E_1 + \frac{ЧДД_1}{ЧДД_1 - ЧДД_2} \cdot (E_1 - E_2)$$

егер  $E_1 \rightarrow ЧДД_1 > 0$ ,

$E_2 \rightarrow ЧДД_2 > 0$ ,

$$BHD = E_1 + \frac{ЧДД_1}{ЧДД_1 + ЧДД_2} \cdot (E_2 - E_1) ;$$

$E_1 = 0,15$ .

$$ЧДД_1 = \frac{9871344}{1,15} + \frac{9871344}{1,15^2} + \frac{9871344}{1,15^3} - 19742688 = 2795812,566 ;$$

$E_2 = 0,20$ .

$$ЧДД_2 = \frac{9871344}{1,2} + \frac{9871344}{1,2^2} + \frac{9871344}{1,2^3} - 19742688 = 1051115,33 ;$$

$E_1 \rightarrow ЧДД_1 > 0$ ;

$E_2 \rightarrow ЧДД_2 > 0$ ;

$$IRR = 0,15 + \frac{2795812,566}{2795812,566 + 1051115,33} \cdot (0,20 - 0,15) = 0,1863 .$$

Сонымен, дисконттау мөлшері келесі аралықта 15 %....18,63 %.

IRR=18,63% Егер мына шарт орындалса, формула дұрыс:  $r_a < IRR < r_b$   
және  $NPV_a > 0 > NPV_b$

5.7 кесте – Бағдарлама өнімін әзірлеуінің және енгізуінің экономикалық пайдалылығының көрсеткіштері

Көрсеткіштер атауы	Мәні
Бағдарлама өнімін әзірлеуге және енгізуге шығын, мың теңге	19742688
Бағдарлама өнімін енгізгеннен кейінгі болжалды үнем, мың теңге	9871344
Таза дисконттық табыс, мың теңге	1051115,33
Табыстық индекс	1,05
Ішкі табыстық мөлшері	0,1863
Дисконтталған өтелу мерзімі, жыл	2
Моральдық ескіру мерзімі, жыл	3

Есептеп шығарылған көрсеткіштер бағдарлама өнімін енгізудің экономикалық тиімді екенін көрсетеді. Сондықтан бағдарлама өнімін әзірлеу және енгізу өте экономикалық негізделеді және пайдалы.

Келесі диаграммадан материалдық шығындардың өзгерісін көре

аласыздар.

Ол әсіресе ірі мекемелерде кең таралған, сондықтан жүзеге асырылуының жоғары бағасына қарамастан, берілген өнім экономикалық тиімді болып саналады.



5.2 сурет– Материалдық шығындардың өзгерісі

Ақшаның жылжуы (қозғалысы)

5.8 кесте – Ақшаның жылжуы

Атауы	Жылдар			Барлығы, тенге.
	2016	2017	2018	
Бірреттік шығын,кВ мың,тенге	610930,446			610930,446
Үнемдеуден шыққан		9871344	9871344	19742688
Пайдаға салық		1974268,8	1974268,8	3948537,6
Таза пайда	-610930,446	7897075,2	7897075,2	15183219,954
Ағымдағы шығын		19742688	19742688	39485376
Дисконттау еселігі (15% мөлшерлеменен)	1	0,87	0,76	

Жалпы өсім бойынша ТДП	-610930,446	6259524,978	12261302,128	178909896,66
------------------------	-------------	-------------	--------------	--------------

### Қорытынды

Бұл дипломдық жобада барлық қойылған мақсаттар іске асырылды. Дерекқорды құру барысында барлық нюанстар ескерілді. Беріліп отырған бағдарламалық өнім барлық үрдісті автоматтандыруға мүмкіндік береді. Ақпараттық ортадағы өнімнің артықшылығы пайдаланушыға кәсіпкерлік жұмысын оңай басқаруға, қажетті ақпаратты жылдам табуға мүмкіндік береді.

Дерекқор NoSQL дерекқоры басқару жүйесінде жасалды. Жаңа түскен ақпаратқа байланысты деректерді тек пайдаланушы енгізе алады. Мұндай қатынау принципі пайдаланушылар үшін онша ыңғайлы емес, сондықтан олар үшін Visual Studio 2010 бағдарламалық ортасында клиенттік қосымша жасалды. Сонымен қатар, авторизация қолдану арқылы деректердің қауіпсіздік пен құпиялылық мәселелері қарастырылды.

«Тіршілік қауіпсіздігі» бөлімінде берілген офистік ғимараттағы еңбек шартының талдауы, оның ішінде табиғи және жасанды жарықтанудың есептеулері келтірілген. Еңбек шартының деңгейі мүмкін деп танылды және есептеулерден алынған мәліметтер өмір тіршілігі қауіпсіздігі стандарттарының талаптарын қанағаттандырады. Есептеулер табиғи жарықтандыру үшін ауданы  $4 \text{ м}^2$  болатын бір ғана терезе жеткілікті болатынын көрсетті.

Техника-экономикалық негізделуі бөлімінде жасалған есептеулер нәтижесінде бағдарламалық өнімді жүзеге асыруға кеткен шығын 314125,265 теңгені құрады. Бұл бағдарламалық өнім кең қолданысқа ие ДҚБЖ NoSQL-де жазылған. Ол әсіресе ірі мекемелерде кең таралған, сондықтан жүзеге асырылуының жоғары бағасына қарамастан, берілген өнім экономикалық тиімді болып саналады.

### Әдебиеттер тізімі

- 1 А.Кригель, Б.Трухнов. SQL. 2-е издание. Библия пользователя. – Москва – Санкт-Петербург - Киев, 2010.
- 2 Microsoft SQL Server. – Санкт-Петербург, 2013.
- 3 М. Сухарев. Золотая книга Delphi. – Санкт-Петербург, 2010.
- 4 Ржеуцкая С.Ю. База данных. Язык SQL. - Вологда: ВоГТУ, 2010.
- 5 Дирк Льюкс, Петер Мюллер. Java 7. Самоучитель. – Санкт-Петербург, 2013.
- 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.
- 8 Дунаев В.В. Базы данных. Язык SQL. – СПб.: БХВ – Петербург, 2006.
- 9 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.
- 10 Хакимжанов Т.Е. Расчет аспирационных систем. Дипломное проектирование. Для студентов всех форм обучения всех специальностей. – Алматы: АИЭС, 2002. – 30 с.
- 11 Резникова Н.П. Маркетинг в телекоммуникациях. – М.:ЭКО-ТРЭНДЗ, 1998.
- 12 Хавронская А.М. Оценка технико-экономической эффективности программных средств. – Алматы: КазНТУ, 2000.
- 13 Голубицкая Е.А., Жигульская Г.М. Экономика связи. – М.: Радио и связь, 1999.
- 14 <http://www.tfa.kz/ru/v-pomoshch-bukhgalteru/86-v-pomoshch-bukhgalteru/115-stavki-nalogov-na-2014-god>
- 15 <http://uchet.kz/news/detail.php?EID=138493>
- 16 [http://www.alser.kz/internet-magazin/noutbuk\\_asus\\_k53s\\_667-core\\_i3\\_-asus/id=120;classId=58](http://www.alser.kz/internet-magazin/noutbuk_asus_k53s_667-core_i3_-asus/id=120;classId=58)
- 17 <http://alfa.kz/peripherals/printers>
- 18 <http://www.inform.kz/kaz/article/2619716>
- 19 <http://uchet.kz/news/detail.php?EID=138493>

## А қосымшасы

### Экономикалық бөлім

А1 кесте – q коэффициентінің мәні

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

А2 кесте - Еңбек сыйымдылығын есептейтін коэффициент

Бағдарлама тілі	Күрделік тобы	Жаңалықтық дәрежесі				В коэффициенті
		А	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

А3 кесте – Бағдарламалық өнімді жасауға жалпы уақыт құрамы

Кезең №	Дәл кезеңдегі уақыт белгісі	Кезеңнің мазмұны
1	T <sub>по</sub>	Мақсат сипатын дайындау

2	$T_O$	Мақсат сипаттамасы
3	$T_A$	Алгоритм құру
4	$T_{BC}$	Алгоритмнің блок-схемасын құру
5	$T_H$	Бағдарламаны.....тілде жазу
6	$T_{II}$	Бағдарламаны теру
7	$T_{OP}$	Бағдарламаны реттеу және тестілеу
8	$T_D$	Құжаттарды рәсімдеу, пайдаланушыға нұсқаулар және түсіндірмелер жазу

А4 кесте – Бағдарлама жасаушы білектілігін ескеретін коэффициент

Жұмыс тәжірибесі	Біліктілік коэффициенті
Екі жылға дейін	0.8
2-3 жыл	1
3-5 жыл	1.1-1.2

*А4 кестенің жалғасы*

Жұмыс тәжірибесі	Біліктілік коэффициенті
5-7 жыл	1.3-1.4
7 жылдан көп	1.5-1.6

А5 кесте – Бірыңғай тарифтік сеткадан көшірме(БТС)

Еңбек ақы разряды	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Тарифтік коэффициенті	1,0	1,07	1,15	1,24	1,33	1,43	1,54	1,66	1,78	1,9	2,0	2,2	2,3	2,5	2,7

										1	5	5	7	5	4
--	--	--	--	--	--	--	--	--	--	---	---	---	---	---	---

*А5 кестенің жалғасы*

Еңбек ақы разряды	16	17	18	19	20	21
Тарифтік коэффициенті	2,95	3,17	3,41	3,67	3,94	4,24