

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ
Коммерциялық емес акционерлік қоғамы
АЛМАТЫ ЭНЕРГЕТИКА ЖӘНЕ БАЙЛАНЫС УНИВЕРСИТЕТІ
«Компьютерлік технологиялар» кафедрасы

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

_____ « _____ » _____ 2014 ж.
(қолы)

ДИПЛОМДЫҚ ЖОБА

Тақырыбы: « АҚ «Қазақтелеком» қызметкерлерінің шифрланған деректер қорын құрастыру»

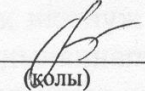
5В070400 – Есептеу техникасы және бағдарламалық қамтамасыз ету мамандығы бойынша

Орындаған: ВТк-10-2 Айдаркулова М.М.

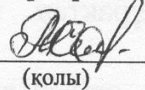
Жетекші: ф.-м.ғ.д., проф. Құралбаев З.Қ.

Кеңесшілер :

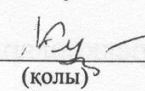
Экономикалық бөлім бойынша :

 « 26 » _____ 2014 ж.
(қолы) _____ доцент Боканова Г.Ш.

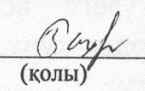
Өмір тіршілігі қауіпсіздігі бойынша:

 « 20 » _____ 2014 ж.
(қолы) _____ т.ғ.к., аға оқытушы Муташева Г.С.

Есептеу техникасын қолдану бойынша :

 « 26 » _____ 2014 ж.
(қолы) _____ ф.-м.ғ.д., профессор Құралбаев З.Қ.

Мөлшер бақылаушы:

 « 23 » _____ 2014 ж.
(қолы) _____ аға оқытушы Рахимжанова З.М.

Пікір жазушы :

_____ « _____ » _____ 2014 ж.
(қолы) _____ ҚазҰТУ, п.ғ.д., проф. Сатыбалдиев О.С.

Алматы 2014

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ
Коммерциялық емес акционерлік қоғамы
АЛМАТЫ ЭНЕРГЕТИКА ЖӘНЕ БАЙЛАНЫС УНИВЕРСИТЕТІ

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

жобаны орындауға берілген

ТАПСЫРМА

Студент Айдаркулова Меруерт Муратовна

Жоба тақырыбы «АҚ «Қазақтелеком» қызметкерлерінің шифрланған деректер қорын құрастыру»
ректордың « ___ » _____ № ___ бұйрығы бойынша бекітілген.

Аяқталған жұмысты тапсыру мерзімі: «27» мамыр 2014 ж.

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

MS SQL Server 2008 дерекқорларды басқару жүйесін және MS Visual Studio бағдарламалау ортасын пайдалану арқылы компанияның қызметкерлері туралы дерекқор көзі шифрланған ақпаратты ұсынатын қосымша құрылды.

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

– Автоматтандырылған қызмет көрсету орнының анықтамасы (арналуы, құрылымы, т.б.).

– Ақпараттық жүйелер, дерекқорлар (ДҚ), дерекқорларды басқарудың жүйелері (ДҚБЖ) түсініктері. TDE шифрлау.

– Жүйенің дерекқорын жобалау және жүзеге асыру, жобалау құралдарын таңдау, ER моделін жобалау, дерекқорды құру, пайдаланушы интерфейсін жобалау, дерекқормен байланысын орнату.

– Техникалық-экономикалық негізделуі.

– Өндірістегі еңбек қорғау мен қауіпсіздік шараларын ұйымдастыру.

Сызба материалдарының (міндетті түрде дайындалатын сызуларды көрсету) тізімі:

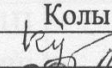
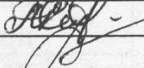
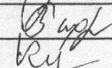
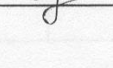

– Дерекқордың ER диаграммасы.

– Жасалған бағдарламалық қосымшаның жұмыс істеу принципі мен функционалдығын көрсететін графикалық материал.

Негізгі ұсынылатын әдебиеттер

- 1 Бондарь А.Г. MS SQL Server 2012. Создание баз данных и разработка программ. –БХВ-Петербург, 2013. – 608с.
- 2 Д. Петкович Microsoft SQL Server 2012. Руководство для начинающих. –БХВ-Петербург, 2013. – 816с.
- 3 Джо Майо: Самоучитель Microsoft Visual Studio 2010. – BHV, 2011г.– 464с.
- 4 Мамаев Е. MS SQL Server. Проектирование и реализация баз данных. Сертификационный экзамен. – СПб.: BHV, 2004. – 416с.
- 5 Марк Шпенник, Оррин Следж. Microsoft SQL Server 2000 DBA Survival Guide. – М.: Вильямс, 2001. – 236 с.

Жоба бойынша бөлімшелерге қатысты белгіленген кеңесшілер

Бөлім	Кеңесші	Мерзімі	Қолы
Негізгі бөлім	Құралбаев З.Қ.		
Тіршілік қауіпсіздігі	Муташева Г.С.		
Экономикалық бөлім	Боканова Г.Ш.	26.05.14	
Норма бақылаушы	Рахимжанова З.М.		
Есептеу техникасын қолдану	Құралбаев З.Қ.		

ДИПЛОМ ЖОБАСЫН ДАЙЫНДАУ

КЕСТЕСІ

№ р/с	Тарау аттары, әзірленетін сұрақтардың тізімі	Жетекшіге ұсыну мерзімдері	Ескерту
1	Автоматтандырылған қызмет көрсету орнының анықтамасы (арналуы, құрылымы, т.б.).	11.03.14	
2	Ақпараттық жүйелер, дерекқорлар (ДҚ), дерекқорларды басқарудың жүйелері (ДҚБЖ) түсініктері.	26.03.14	
3	Жүйенің дерекқорын жобалау және жүзеге асыру, жобалау құралдарын таңдау, ER моделін жобалау.	3.04.14	
4	Дерекқорды құру, пайдаланушы интерфейсін жобалау, дерекқормен байланысын орнату.	17.04.14	
5	Шифрлау, деректерді мөлдір шифрлау (TDE) түсініктері мен принциптері.	23.04.14	
6	Жүйенің дерекқорын шифрлау түрін таңдау, жүзеге асыру	14.05.14	

Тапсырманың берілген уақыты « 25 » ақпан 2014ж.

Кафедра меңгерушісі _____ ф.-м.ғ.д., проф. Құралбаев З.Қ.
(қолы)

Жоба жетекшісі _____ ф.-м.ғ.д., проф. Құралбаев З.Қ.
(қолы)

Орындалатын тапсырманы қабылдаған студент _____ Айдаркулова М.М.
(қолы)

Аннотация

В данном дипломном проекте анализированы различные методы шифрования. Была разработана база данных для сотрудников АО «Қазақтелеком» и для сохранности информации была зашифрована. Пользователь может получить всю нужную информацию о сотрудниках из базы данных.

База данных информационной системы разрабатывалась в СУБД MS SQL Server 2008 и клиентское приложение на Visual Studio 2010. Важно отметить что актуальность создания этой системы заключается в безопасности и удобстве программы для пользователя.

В дипломном проекте рассмотрены такие вопросы, как разработка архитектуры информационной системы, структура БД и создание приложения.

В разделе «Безопасность жизнедеятельности» проведен анализ условий труда, рабочего помещения. На основании полученных данных сделан расчет системы кондиционирования и искусственного освещения.

В разделе «Технико-экономического обоснования разработки» произведен расчет полученной экономической выгоды от внедряемой системы

Андатпа

Дипломдық жобада түрлі шифрлану әдістері талданған. АҚ «Қазақтелеком» компаниясының қызметкерлеріне арналған деректер қоры құрылды және ақпарат қауіпсіздігін қамтамасыз ету үшін деректер қоры түрлі тәсілмен шифрланды. Деректер қорынан пайдаланушы барлық қызметкерлер жайында қажет ақпаратты алуға мүмкіндігі бар.

Ақпараттық жүйенің дерек қоры ДҚБЖ MS SQL Server 2008 және клиенттік қосымшасы Visual Studio 2010 бағдарламалық құралында жасалған. Пайдаланушы үшін осы жүйені құрудың өзекті мәселесі ретінде бағдарламаның қауіпсіздігі мен оңтайлылығын айтып өту өте маңызды.

Дипломдық жобада ақпараттық жүйе архитектурасы, ДҚ құрылымы және қызметкерлер бағдарламасының қосымшасын жасау мәселелері қарастырылған.

«Тіршілік қауіпсіздігі» бөлімінде еңбек жағдайларына, жұмыс орнына талдау жасалынды. Жұмыс орнында кондиционерлеу мен жасанды жарықтандыру бойынша есептеулер жүргізілді.

«Техника-экономикалық негізделуі» бөлімде енгізілген жүйеге кеткен шығынды және одан алынған экономикалық пайданы есептеу жүргізілді.

МАЗМҰНЫ

Кіріспе.....	7
1 Автоматтандырылған қызмет көрсету орнының АҚКО тағайындалуы мен құрылымы.....	8
1.1 АҚКО құрылымы.....	9
1.2 АҚКО-ға қойылатын жалпы талаптар.....	10
1.3 АҚКО-ның жобалау сатылары.....	11
2 Ақпараттық жүйелер. Дерекқорлар (ДҚ). Дерекқорларды басқарудың жүйелері (ДҚБЖ).....	12
2.1 Компьютерлік қауіпсіздіктің негізгі түсінігі.....	22
2.2 SQL тілі.....	24
2.3 ДҚ шифрлау. Transparent Data Encryption (TDE).....	25
3 Дерекқорды жобалау және оны іске асыру.....	27
3.1 ДҚ құрудың бастапқы кезеңі.....	27
3.1.1 Қолдану облысы және тағайындалуы.....	27
3.1.2 Дерекқордағы шешілетін мәселелер.....	27
3.2 Концептуалды жобалау.....	27
3.2.1 Пайдаланушының талабы.....	27
3.2.2 Техникалық құралдардың параметрлеріне қойылатын талаптар..	27
3.2.3 Дерекқор және қосымшаны жобалау құралдарын таңдаудың негіздемесі.....	28
3.2.4 ER моделін құру.....	29
3.2.5 Бизнес ережелер.....	29
3.2.6 Реляционды қатынастарды құрастыру.....	31
3.3 Логикалық жобалау.....	32
3.3.1 Кестелерді құру.....	32
3.3.2 ДҚ диаграммасы.....	34
3.3.3 Ақпаратты құпияландыру. TDE жүзеге асыру.....	34
3.4 Физикалық жобалау.....	36
3.4.1 Көрсетілімдер және процедуралар.....	36
3.4.2 Триггерлер.....	36
3.5 Қосымша интерфейсін және оның функционалдығын сипаттау. Қосымшамен жұмыс істеу.....	37
4 Тіршілік қауіпсіздігі.....	42
4.1 Инженердің жұмыс жағдайын талдау.....	42
4.2 Есептеу бөлімі.....	45
4.2.1 Кондиционерлеу және кондиционерді таңдап, есептеу.....	45
4.2.2 Жарықтандыруды есептеу.....	49
5 Техничко-экономикалық негізделуі.....	52
5.1 Жұмысты сипаттау және қажеттілігін негіздеу.....	52
5.2 Нарықты саралау және орнын анықтау.....	52
5.3 Бағдарламаның SWOT-анализі.....	53
5.4 Маркетинг-микс элементі.....	53

5.5 Бағдарламалық қамтаманың өңделуінің еңбек өнімділігінің есептелуі.....	53
5.6 Бағдарламаның бағасын есептеу.....	59
5.7 Ақшалай құралдардың қозғалысы.....	62
5.8 Экономикалық тиімділікті есептеу.....	62
Қорытынды.....	65
Әдебиеттер тізімі.....	66
ҚОСЫМША А.....	67
ҚОСЫМША Ә.....	71

Кіріспе

Берілген бітіру жұмысында «Қазақтелеком» компаниясының қызметкерлеріне арналған деректер қорын құрып, шифрлау арқылы ақпарат қауіпсіздігін қамтамасыз ететін бағдарлама жобаланды.

Берілген өнімді әзірлеудің мақсаты деректерді енгізу, сақтау, өңдеу және тасымалдау үшін қолданылатын ақпарат пен қорлардың тұтастығын, қол жеткізулік оңтайлығын және жасырындылығын қолдау болып табылады. Ақпараттың сыртқа кетуінің, оны ұрлаудың, жоғалтудың, рұқсатсыз жоюдың, өзгертудің, маңызына тимей түрлендірудің, рұқсатсыз көшірмесін жасаудың, бұғаттаудың алдын алу үшін бағдарламалық қамтама жасау қазіргі заманда өте маңызды. Осы бағдарламалық өнім көмегімен құнды ақпараттардың қауіпсіздігін қамтамасыз етеміз.

Дерекқор құру үшін алдымен зерттелетін облыстың талдауы жасалады. Инфологиялық модель MS SQL Server-жүйесінде жасалады. Қызметкерлердің дерекқоры MS SQL Server дерекқорларды басқару жүйесінде құрылғаннан кейін, пайдаланушылардың барлық сұраныстарын қанағаттандыратын қосымша жобаланады. Бағдарлама MS Visual Studio бағдарламалық ортасында іске асырылады.

Бағдарлама жасау нәтижесінде, қызметкерлер деректер қорымен жұмыс жасаудың тиімділігі, жылдамдығы және ыңғайлылығы артады және шифрлану арқасында ақпараттың қауіпсіздігі қамтамасыз етіледі.

1 Автоматтандырылған қызмет көрсету орнының (АҚКО) тағайындамасы мен құрылымы

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

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

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

АҚКО-да шешілетін мәселелерді шартты түрде ақпараттық және есептеуіш деп бөлуге болады.

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

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

1.1 АҚКО құрылымы

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

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

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

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

АҚКО-ның қамтамасыз ету бөлігі:

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

АҚЕту, АҚКО бұл құжаттардың, ақпарат массивтерінің (ДК) және кодалаудың стандартталған және бәрыңғай формалар жиынтығы.

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

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

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

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

1.2 АҚКО-ға қойылатын жалпы талаптар

АҚКО-ға қойылатын жалпы талаптар келесідей:

- жүйелілік;
- икемділік;
- тұрақтылық;
- тиімділік.

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

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

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

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

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

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

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

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

Мұндай жүйеде келесі қамтамасыз етілу керек:

– мамандардың ақпаратты өңдеу машина құралдарына максималды жақындығы;

– диалогты тәртіпте жұмыс жасау;

– эргономика талаптарына сәйкестілік;

– компьютердің жоғары икемділігі;

– ескілік процестердің максималды автоматтандырылуы;

– жекелік жағдайда мамандардың творчестволық белсенділігін арттыратын, ал кейін жүйенің дамуына әкелетін мамандардың еңбек жағдайларына моральды түрде қанағат болуы;

– мамандардың өзін-өзі оқыту мүмкіндігі.

1.3 АҚКО-ның жобалану сатылары

АҚКО-ын дұрыс ұйымдастыру, құрастыру және енгізу мақсатында оның жобалау процесін келесі сатыларға жіктейді:

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

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

Көптеген мәселелер шешімінің негізінде ақпаратты өңдеу жатыр. Ақпарат өңдеуін ыңғайлату үшін ақпараттық жүйелер (АЖ) құрылады. Автоматтандырылған ақпараттық жүйелер бұл техникалық құралдарды, әсіресе ЭЕМ-ды қолданатын жүйелер. Ақпараттық жүйелердің көпшілігі автоматтандырылған болып табылады, сондықтан қысқарту үшін оларды жай АЖ деп атайды.

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

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

Ақпараттық жүйенің эффективтілігі келесі үштіктен тұрады:

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

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

Қазіргі кезде клиент-сервер сәулеті перспективті болып табылады.

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

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

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

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

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

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

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

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

– қолданба бұл ДҚ-ды қолданатын және кейбір пән облысынан ақпараттың өңдеуін автоматтандырылуын қамтамасыз ететін программа немесе программалар кешені;

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

Қолданушылар дерекқормен жұмыс істейтін бағдарламалар қосымшалар деп аталады. Жалпы жағдайда бір дерекқормен әртүрлі бірнеше қосымшалар жұмыс істей алады. Мысалы, егер дерекқор бір кәсіпорынды үлгілесе, онда оның жұмысы үшін кадрлардың есепке алу ішкі жүйесін қамтамасыз ететін қосымша құрылуы мүмкін, басқа қосымша қызметкерлердің жалақысын есептеуге арналады, үшінші қосымша складтың есепке алу ішкі жүйесі ретінде жұмыс атқарады, төртінші қосымша өндірістік процесстің жоспарлануына арналған. Бір дерекқормен жұмыс істейтін қосымшаларды қарастырғанда олар бір-бірімен параллельді түрде және бір-бірінен тәуелсіз де жұмыс істей алады деп болжайды, және дәл ДҚБЖ көптеген қосымшалардың жұмысын бір дерекқормен әрқайсысы дұрыс, бірақ басқа қосымшалардың енгізген өзгерістерін қабылдайтындай қамтамасыз етуге арналған.

"Клиент-сервер" технологиясы

Файл-серверлердің кемшіліктері:

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

– масштабталудың болмауы.

Құрылымы:

– клиент-сервер архитектурасы желінің және үлестірілген дерекқордың (корпоративті ДҚ және дербес ДҚ) бар болуын болжайды;

– КДҚ сервер-компьютерде орналасады;

– ДДҚ корпоративті ДҚ-ның клиенті болып табылатын бөлімшенің қызметкерлерінің компьютерлерінде орналасады.

Анықтауыш:

– компьютерлік желіде белгілі бір ресурстың сервері деп осы ресурсты басқаратын компьютер (бағдарлама) аталады;

– клиент – осы ресурсты қолданатын компьютер (бағдарлама);

– компьютерлік желінің ресурсы ретінде, мысалға, дерекқор, файлдық жүйелер, баспа қызметтері, почта қызметтері бола алады.

Артықшылықтары:

– орталықтандырылған сақтау, қызмет көрсету және жалпы корпоративті ақпаратқа коллективті қол жеткізудің дербес ақпаратпен жеке жұмыстың сәтті қосылуы. корпоративті (серверлік) ДҚ ДҚ серверінің басқаруымен құрылады, қолданады және функцияланады;

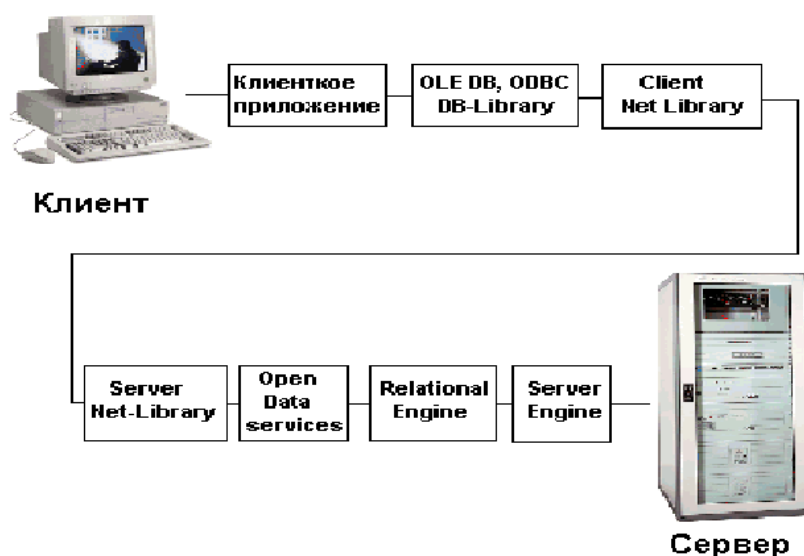
– осы технологияны қолданумен деректерді өңдеудегі барлық шамадан тыс еңбек толығымен серверге жүктеледі. Машина-клиент сұраныс жібереді, ал сервер оны орындайды және жауаптарын клиентке жібереді.

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

– корпоративті және дербес қорларды құрайтын компьютер-сервер;

– компьютер-сервер және ддқ бар дербес компьютерлер;

– бірнеше компьютер-серверлер және ддқ бар дербес компьютерлер.



2.1-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.

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

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

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

Егер жобаланып жатқан ақпараттық жүйе (АЖ) "клиент-сервер" технологиясы бойынша құрылады деп 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. Осыған сәйкес кез келген қосымшада келесі логикалық компоненттер бөлінеді:

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

Деректер үлгісі.

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

- иерархиялық;
- желілік;
- реляциондық.

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

- постреляционды;
- көпөлшемді;
- объектілі-нысандалған.

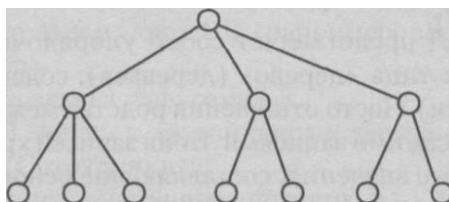
Белгілі үлгілерді кеңейтіп, басқа да деректер үлгісінде негізделген мүмкін жүйелер әзірленуде. Оның қатарына объектілі-реляционды, дедуктивті-объектілі-нысандалған, семантикалық, концептуалдық және бағыттауыш үлгілерді жатқызуға болады. Бұл үлгілердің кейбіреуі дерекқордың, білім қорының және программалау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

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

1) Иерархиялық үлгі.

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

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



2.2-сурет – Байланыстардың иерархиялық түрде көрсетілімі

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

«Ағаш» типінің анықтамасының сәйкесімен ата-аналар мен олардың ұрпақтарының арасында байланыстың тұтастығын тексеру автоматты түрде болатынын қорытындылауға болады. Тұтастықты тексерудің негізгі ережесі келесі түрде тұжырымдалады: ұрпақ ата-анасыз бола алмайды, ал кейбір ата-аналарда ұрпақ болмауы да мүмкін. Өртүрлі ағаштар жазбаларының арасындағы байланыстың тұтастығын қолдау механизмі мүлдем болмайды.

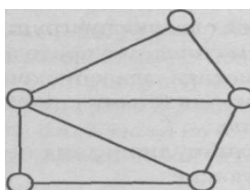
Деректердің иерархиялық үлгілерінің артықшылықтарына ЭЕМ жадын тиімді қолдану және деректердің негізгі операциясын орындауға кететін жаман емес уақыт көрсеткіші жатады. Деректердің иерархиялық үлгісі иерархиялық реттелген ақпараттармен жұмыста өте қолайлы.

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

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

2) Желілік үлгі.

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



2.3-сурет – Желілік үлгідегі байланыстардың көрсетілімі

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

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

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

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

3) Реляционды үлгі.

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

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

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

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

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

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

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

4) Постреляционды үлгі.

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

Деректердің постреляционды үлгісі кесте жазбаларында сақталатын деректердің бөлінбейтіндігіне шектеулікті шешетін кеңейтілген реляционды үлгіні ұсынады. Деректердің постреляционды үлгісі мәндері ішкі мәннен тұратын көпмәнді жолдардан тұрады. Көпмәнді жолдардың мәндерін теру негізгі кестеге кіргізілген өзіндік кесте болып саналады.

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

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

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

5) Көпөлшемді үлгі.

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

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

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

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

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

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

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

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

Деректердің көпөлшемді үлгісін қолдайтын жүйе мысалы болып Essbase (Arbor Software), Media Multi-matrix (Speedware), Oracle Express Server (Oracle) және Cache (InterSystems) табылады. Кейбір бағдарламалық өнімдер, мысалы, Media/MR (Speedware) біруақытта көпөлшемді және реляционды ДҚ-мен жұмыс істеуге мүмкіндік береді. Ішкі үлгісі көпөлшемді үлгі болып табылатын Cache ДҚБЖ-де деректерге қол жеткізу үш әдіспен орындалады: тура (көпөлшемді массивтер түйінінің деңгейінде), объектілі және реляционды.

б) Объектілі-нысандалған үлгі.

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

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

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

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

Реляционды дерекқордың негізгі түсініктері.

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

Дерекқорында тек физикалық объектілер көрсетілмейді. Ол абстракция, процесстер, құбылыстар – яғни, адам тіршілігінде кездесетін барлық заттар

туралы мәліметтерді сақтауға мүмкіндігі бар. Мысалы, дерекқорында складқа құралдарды әкелуге тапсырыс туралы ақпаратты сақтауға болады. "Тапсырыс" нысанының атрибуттары әкелінетін құралдың аты, құралдың саны, әкелушінің аты, мерзімі т.б. болады. Шынайы әлемнің объектітері бір-бірімен көптеген қиын тәуелділіктермен байланысты. Дерекқорында тек маңызды байланыстарды сақтау керектігін айта кету жөн.

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

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

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

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

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

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

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

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

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

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

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

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

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

Тұтастық шектеуліктерінің бірнеше типі бар. Мысалы, кесте бағанасы мәндері тек сәйкес домендерден таңдалынсын деп талап етіледі. Практикада одан да қиын шектеуліктер қояды, мысалы, сілтеме бойынша шектеуліктер (referential integrity). Оның мәні сыртқы кілт кестеде жоқ бағанаға көрсеткіш бола алмайтындығында. Тұтастық шектеулігі арнайы құрылғылардың – ереже (rules), триггерлер (triggers) және домендер көмегімен жасалады.

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

2.1 Компьютерлік қауіпсіздіктің негізгі түсінігі

Әртүрлі категориядағы адамдар үшін «қауіпсіздік» сөзі әралуан ассоциация келтіреді. Адамның әрбір қызмет саласымен байланысты бұл термин бірнеше мағынада қолданады. Мысалы, мемлекеттік қауіпсіздік немесе тіршілік әрекет қауіпсіздігі, политикалық және құрылыс қауіпсіздігі. Негізінде мұндай түрлер өте көп, және басқадан айыратын, әр жағдайда бұл теминге нақты қызмет саласына жататын спецификалық талап қойылады. Жалпыланған «қауіпсіздік» терминінің анықтамасын былай айтуға болады: қауіпсіздік – орындалуы қажетсіз салдарға алып келетін, тыйым салынған әрекеттің алдын алуға бағытталған талаптамалар мен құралдар жиынтығы.

«Ақпараттық қауіпсіздік» термині салыстырмалы түрде, есептеуіш техника мен ЭЕМ-ның дамуна байланысты жақын уақыт аралығында пайда болды. Ақпараттық қауіпсіздік құрамына тек пайдаланатын бағдарламалық қамтамасыз етудің қауіпсіздігі кірмейді, сол сияқты аппаратты құралдардың қауіпсіздігі, байланыс арнасының қауіпсіздігі және тағы көптеген.

Операциялық жүйе қауіпсіздігі түсінігіне операциялық жүйенің орнатылған өз ядросының қауіпсіздігінен басқа оған орнатылған программалық қамтамасыз ету қауіпсіздігі кіреді. Басқа сөзбен айтқанда, операциялық жүйе қауіпсіздігі – қолданушы немесе басқа бағдарламамадан, операциялық жүйенің қалыпты функционалдауына зақым келуіне алып келетін әрекететінің, алдын алатын шаралар жиынтығы.

Тыйым салу бар болса, оны бұзғысы келген адам пайда болады! Ақпараттық технологиялар мен компьютердің барлық қызмет аясына енгізілуіне байланысты «кракер» (ағылшын тілінен cracker – «компьютерлік желі мен бағдарламаны бұзғыш») атты адамдар пайда болды. Олар әртүрлі қулық және стандартты емес тәсілдермен, өзіне пайда түсіру үшін оларға арналмаған, рұқсат етілмеген ақпаратқа мүмкіндік алғысы келеді. Кей кезде ол ақпарат иесін ауыр жағдайға алып келеді. Сол кезде «ақпараттық зиянкестермен» күреске қажеттілік пайда болады. Ақпараттық қауіпсіздікті күшейтетін және кракерлермен күресуге арналған бүтін комплекстер жасалып шығарылды. Сол себепті кез келген электронды-есептеуіш комплексті арнаулы операциялық жүйе басқаратындықтан, жүйенің қауіпсіздігін қамтамасыз ету үшін, операциялық жүйенің қауіпсіздігінің қамын ойлау қажет.

Әрбір өндіруші операциялық жүйе қауіпсіздігіне өзіндік сипатпен қарады. Нәтижесінде кейбір жүйелер айтарлықтай қорғалған, басқаларының қорғанысын айналып өту оңайшылыққа түспеді, ал үшіншілері іс жүзінде бұзушылар алдында әлсіз болып қалды. ОЖ Linux қорғаныс дәрежесі, UNIX тәрізді тағы бір нақты уақыттағы ОЖ, осы жұмыста қаралады.

ОЖ қауіпсіздігін зерттей отырып, компьютерлік қауіпсіздік теориясы туралы көз шолмау мүмкін емес. Компьютерлік қауіпсіздік теориясы үш негізгі түсінікпен стратегиялық әрекет жасайды: қауіп-қатер, әлсіздік, шабуыл.

Компьютерлік жүйенің қауіпсіздік қауіп-қатері – жүйенің өзіне (қайта қотарлау сияқты) және ол жерде жатқан ақпаратқа (ошіру, файлдарды бұзу және сол сияқты) кері әсерін тигізетін потенциалды мүмкін жағдай. Компьютерлік жүйенің әлсіздігі – қауіп-қатердің пайда болуына мүмкіндік беретін сәтсіз және орынсыз сипаты.

Әлсіздікке ақпараттық жүйенің әтүрлі жағдайларын жатқызуға болады.

Пайдаланатын бағдарламалық қамтамасыз етудің жетілдірілмегені. Бағдарламалық қамтамасыз ету адаммен жазылған, ал адам қателік жасауға бейім. Бағдарламалық өнімді құрастыру қадамында көп адам жұмыс істесе де, оның байланыстары мен мүмкін қателерін аңдып табу мүмкін емес. Бағдарламалық өнімді құрушылар бағдарламалық кодты қаншалықты оптимизациялап, оны қателіктен босатып жатса да, барлық жағдайларды қарастыру мүмкін емес. Мұндай аяқтап істелмеулер кейде өз бағдарлама немесе мәліметтер үшін ешқандай да сыни зардаптарды алып жүрмейді, бірақ олар өз мақсатында бағдарламаны пайдалануға рұқсат бергенін қалайды, басқалармен салыстырғанда ол үшін бастапқы жасалмаған, өте маңызды зардаптармен шығарып салынады. Бағдарламада мұндай аяқтап істелмеулер әдетте «тесіктер» деп аталады.

Компьютерлік жүйеге шабуыл – әлсіздіктерді іздеу және қауіп-қатердің іске асыру мақсаты мүмкін болатын әрекеттер алгоритмі.

Қауіп-қатердің үш негізгі түрі бар:

- ашылу қауіп-қатері;
- бүтінділік қауіп-қатері;
- қызмет көрсетудің тоқтап қалу қауіп-қатері.

Ашылу қауіп-қатерінің түсінігі білуі тиіс емес адамға ақпараттың мәлім болуы. Компьютерлік қауіпсіздік терминында ашылу қауіп-қатері бірнеше кұпия ақпаратқа, есептеуіш жүйеде сақталатын немесе бір жүйеден келесіге берілетін, қол жеткізу кезінде орын алады. "Ашылу" сөз орынына кейде "ұрлық" немесе "ағып кету" терминдерді пайдаланады.

2.2 SQL тілі

SQL тілі компьютердің деректер қорында сақталған деректерді оқуға және өңдеуге арналған құрал болып табылады. Атынан көрініп тұрғандай, SQL тілі қолданушыны деректер қорымен байланыстыру үшін қолданылатын программалау тілі болып табылады. Шынында, SQL тек белгілі бір типті, яғни реляционды деректер қорымен жұмыс істейді.

Егер қолданушыға деректер қорынан деректерді оқу қажет болса, онда ол SQL арқылы деректерді ДҚБЖ-дан сұрастырады. ДҚБЖ сұраныстарды өңдейді және қажетті деректерді тауып, оларды қолданушыға жібереді. Деректерді сұрату және нәтижені қабылдау процесі деректер қорының сұранысы деп аталады. Сұраныстың құрылымдық тілі деген аты осыдан шыққан.

Бірақ бұл аты ақиқатқа сәйкес келмейді. Біріншіден, SQL сұраныстарды құрайтын жай ғана құрал болып табылмайды, басында сол үшін ғана арналған болғанымен. Қазіргі кезде SQL тілі барлық функционалдық мүмкіндіктерді орындау үшін қолданылады. Олар:

- деректерді ұйымдастыру. SQL пайдаланушыға деректерді көрсету құрылымын өзгерту мүмкіндігін береді. Сонымен қатар деректер қорының элементтері арасында қатынастарды құрады;

- деректерді оқу. SQL пайдаланушыға немесе қолданбаға деректер қорында сақталған деректерді оқу мүмкіндігін береді;

- деректерді өңдеу. SQL пайдаланушыға немесе қолданбаға деректерді өзгерту мүмкіндігін береді. Олар: қорға жаңа деректерді қосу, өшіру немесе бар базаны жаңарту;

- қол жеткізуді басқару. SQL көмегімен пайдаланушының деректерді оқу, өзгерту мүмкіндігін шектеуге болады. Сонымен қатар оларды рұқсатсыз қол жеткізуден қорғауға болады;

- деректерді қатар, бірге қолдану мүмкіндігі. SQL параллель жұмыс жасап отырған пайдаланушыларға, бір-біріне кедергі жасамайтындай, бір уақытта істеуіне мүмкіндік береді;

- деректердің тұтастығы. SQL деректер қорының тұтастығын қамтамасыз етеді. Яғни, деректер қорын күтпеген өзгерістерден немесе жүйенің істен шығуынан қорғайды.

Осылайша, SQL ДҚБЖ-мен істейтін мықты тіл болып табылады. Екіншіден, SQL тілі COBOL, FORTRAN немесе C типті толық емес компьютер тілі. SQL отызға жуық операторлары бар деректер қоры болып табылады. SQL операторлары базалық тілге қондырылады. Мысалы, COBOL, FORTRAN немесе C тілдеріне және деректер қорына қол жеткізу мүмкіндігін береді.

Сонымен қатар С сияқты тіл арқылы SQL операторына функцияны шақыру интерфейсін қолдану арқылы ДҚБЖ-ны айқын түрінде жіберуге болады.

SQL — деректер қорын программалау тілі. Деректер қорына рұқсатты алу үшін программисттер өздерінің SQL бағдарламасының командаларын қояды. Бұл әдіс пайдаланушылардың өздері жазған бағдарламаларда және деректер қорының қызметші бағдарламаларында қолданылады.

Осылайша, SQL пайдаланушы адамдарға, бағдарламаларға және есептеу жүйелеріне реляционды деректер қорында сақталған мәліметтерге қол жеткізу мүмкіндігін беретін пайдалы және қуатты құралға айналды.

Microsoft SQL Server – бұл клиент-серверлік реляционды дерекқорды басқаратын жүйе.

SQL Server негізі - осы программалау тілі SQL болып табылады. Бұл тіл 1970 жылдардың басында IBM компаниясымен жасалған. Оның бастапқыда SEQVEL (Structured English Query Language) деп аталды. SQL Server тілінің негізі ретінде SQL тілінің T-SQL (Transact – SQL) деген түрі енді. Microsoft SQL Server 2008 құрамына келесілер кіреді:

- 1) Deployment Wizard – серверде сақталған ақпаратты шығару мастері;
- 2) SQL Server Installation Center – SQL Server 2008 орнату ортасы;
- 3) Reporting Services Configuration Manager – есеп беру қызметінің менеджері;
- 4) SQL Server Configuration Manager – серверді жөндеу менеджері;
- 5) SQL Server Error and Usage Reporting – қателер бойынша есеп беру және сервер жұмысын протоколдау;
- 6) Microsoft Samples Overview – Microsoft корпорациясының парақшасына сілтеме;
- 7) SQL Server Books Online – Microsoft SQL Server 2008 бойынша толық анықтамалық жүйе;
- 8) SQL Server Tutorials – сервермен жұмыс істеу оқулықтары;
- 9) Data Profile Viewer – деректермен жұмыс істеу профильдерің қарау;
- 10) Execute Package Utility – деректерді кішірейту құралы;
- 11) Database Engine Tuning Advisor
- 12) SQL Server Profiler – деректермен жұмыс істеу профильдері;
- 13) Import and Export Data – деректердің импорты мен экспорты;
- 14) SQL Server Business Intelligence Development Studio.

2.3 ДҚ шифрлау. Transparent Data Encryption (TDE)

Microsoft SQL Server 2008-де жақсы криптографиялық қолдау жүзеге асырылған. Ұсынылған құралдар арқылы деректерді симметриялық та, ассиметриялық алгоритмдерді қолдана шифрлауға мүмкіндік бар. Хэширование және электронды қолтаңба операцияларын қолдану мүмкіндігі орнатылған. Кілттермен басқару жүйесі жақсы жүзеге асырылған.

Негізінен деректерді шифрлау еш қиындық көрсетпейді. Шифрлау алгоритмдері жақсы белгілі және көбісі операционды жүйеде жүзеге

асқан(мысалы, AES). Қиыны сол деректер шифрланған кілтті қорғау болып келеді. Себебі егер кілтті сол шифрланған дерекқор жанына сақтасақ, берік қорғау емес, жай ғана обфускацияға келеміз. Осы мәселені орындау үшін SQL Server арнайы кілттер иерархиясын қолданады.

Transparent Data Encryption (TDE) арқылы шифрланған әрбір дерек үшін арнайы кілт құрылады – Database Encryption Key (DEK). Бұл кілт дерекқорды шифрлауға қолданылады.

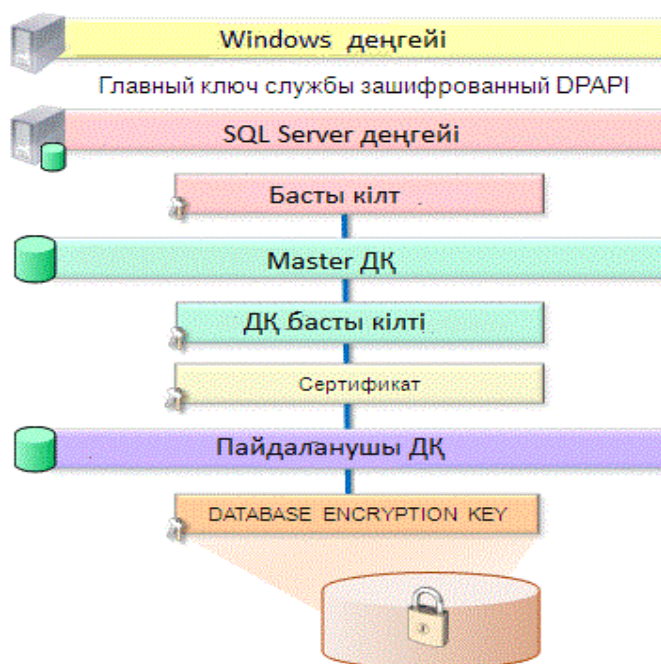
Database Encryption Key (DEK) – БД master-де құрылған сертификат арқылы шифрланады.

Әрі қарай қалыпты:

- бұл сертификат БД master басты кілті арқылы шифрланады.
- БД master басты кілтті қызмет басты кілтімен (Service Master Key немесе SMK) шифрланады.

- қызмет басты кілтті (SMK) DPAPI арқылы шифрланады.

Бұл иерархия толығымен келесі суретте келтірілген:



2.4-сурет – TDE иерархиясы

Егер бөгде адам қорғалған деректерге қол жеткізсе, онда Transparent Data Encryption (TDE) мүмкіндіктері оны ашуға мүмкіндік бермейді. Себебі деректер ұрланған кезде де, ұрланған ақпарат ішіне кірмеген басты кілтсіз ашу мүмкін емес. Басты кілт деректермен бірге ұрланса да, оны ашар ДҚ сертификатсыз ашылмайды. Шифрланған дерекқор басқа компьютерде жай ғана белгісіз символдарға толы бір файл болып қалады.

Сонымен бұл көптеген нормалық және басқарушылық талаптарды қанағаттандырады.

3 Дерекқорды жобалау және оны іске асыру

3.1 ДҚ құрудың бастапқы кезеңі

3.1.1 Қолдану облысы және тағайындалуы

Бұл жүйе автоматтандырылған қызмет көрсету орны болып табылады. Яғни барлық қызметкерлер, олардың регионы, қай отдел, логины мен паролі және де өзге жеке ақпараттары туралы ақпарат алу және қажет болған жағдайда өзгерту, ақпаратты шифрлау арқылы ағып кетуінен қорғау. Бұл жүйе келесідей мәліметтерді жүйелеп, қолданушының талаптарына сай болуы қажет:

- аймақ, бөлімше, қызмет (лауазым) жайлы мәліметтер;
- логин мен паролін анықтап, тіркеу.

3.1.2 Дерекқордағы шешілетін мәселелер

Берілген өнімді әзірлеудің мақсаты болып деректерді енгізу, сақтау, өңдеу және тасымалдау үшін қолданылатын ақпарат пен қорлардың тұтастығын, қол жеткізулік оңтайлығын және жасырындылығын қолдау болып табылады.

3.2 Концептуалды жобалау

3.2.1 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 жанартулар болмау керек.

3.2.2 Техникалық құралдардың параметрлеріне қойылатын талаптар

Техникалық құралдардың құрамына мынадай сипаттамалары бар дербес компьютер кіреді:

- процессор Pentium-4 кем емес;
- жедел жады көлемі 2 Гбайттан кем емес;
- HDD, 250 Гбайт аз емес;
- Windows 7 операцияндық жүйесі;

- MS SQL Server.
- MS Visual Studio

3.2.3 Дерекқор және қосымшаны жобалау құралдарын таңдаудың негіздемесі

Дерекқорларды басқару жүйесі.

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

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

- 1) орнату және қолданудың жеңілдігі;
- 2) кең таралған;
- 3) қымбат серверлік қуатты орнатуды қажет етпейді;
- 4) өндіргіштігінің жақсы көрсеткіші;
- 5) деректерді сақтау сенімділігі мен қауіпсіздігін қамтамасыз етудің жеке құралдары;
- 6) коммерциялық емес қолдану үшін өнімді сызғышта арзан түрлерінің болуы.

Бағдарламалық өнімді құру үшін дерекқор ретінде Microsoft SQL Server 2008 нұсқасы таңдалынды. SQL Server 2008 (Microsoft Corporation) - өте ірі американдық корпорация, әлемнің ең ірі серверлік құралдармен жабдықтырушы компания әрібағдарламалық қосымша жасаумен айналысатын ірі мекеме. Компания 1970 жылы құрылған. 145-тен астам елде бөлімшесі бар. 2011 жылғы санаққа сәйкес 108 мың адам жұмыс жасайды. Штаб-бөлмесі Калифорния штатының Сан-Франциско қаласына жақын аймақта орналасқан. Microsoft SQL Server 2008 компьютерлік дерекқорда сақталатын деректерді өңдеу және оқу үшін арналған құрал болып табылады, және қолданушының дерекқормен өзара қатынас ұйымдастыру үшін қолданылатын программалау тілі болып табылады. Microsoft SQL Server ДҚБЖ қолданушыға ұсынатын барлық функционалдық мүмкіншіліктерді жүзеге асыру үшін қолданылады, соның ішінде:

- деректерді оқу. Қолданушыға немесе қосымшаға дерекқордың ішіндегі деректерді оқуға және олармен қолдануға мүмкіндік береді.
- деректерді өңдеу. Қолданушыға немесе қосымшаға дерекқорды өзгертуге, яғни оған жаңа деректер енгізуге, оған қоса бұрыннан бар деректерді өшіруге немесе жанартуға мүмкіндік береді.
- деректерді ұйымдастыру. Қолданушыға деректердің көрсетілімі құрылымын өзгертуге, сонымен қатар дерекқордың элементтері арасында қатынас құруға мүмкіндік береді.
- деректердің тұтастығы.

3.2.3 ER моделін құру

ДҚ құрудың бастапқы кезеңінде алғашқы нысандар жиынтығын анықтау қажет. Бұл ақырғы пайдаланушылар мен жобалаушы көзқарасы бойынша жүйе объектілері туралы ең маңызды ақпарат болып табылады. Дерекқор 3.1-кестеде көрсетілген нысандарды қолданады.

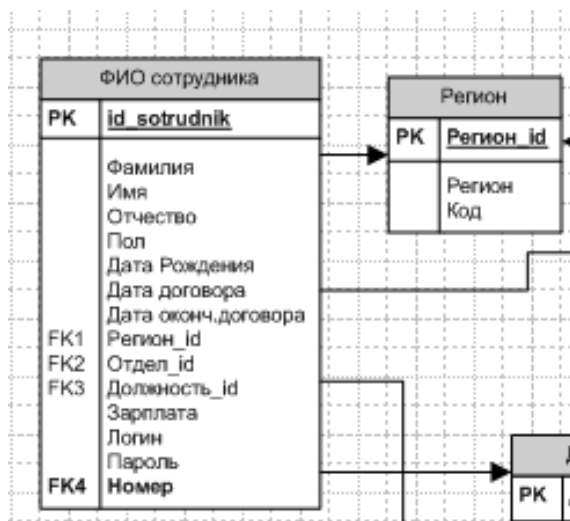
К е с т е 3.1 - «Қазақтелеком» компаниясының қызметкерлеріне арналған шифрланған дерекқорының нысандары

Нысан аты	Сипаттамасы
Қызметкерлер	Қызметкерлер туралы мәліметтер
Регион	Аймақтар туралы мәліметтер
Отдел	Бөлім туралы мәліметтер
Должность	Лауазым туралы мәліметтер
Банковская карта	Карталар тізімі
Архив	Қызметкерлер туралы мәліметтер
Карта	Қызметкерлер жеке карта мәліметі

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

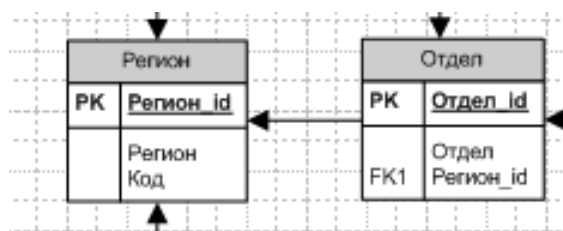
3.2.4 Бизнес ережелер

1) Әрбір қызметкер бір аймаққа жатады, бірақ бір аймақта көп қызметкерлер бар.



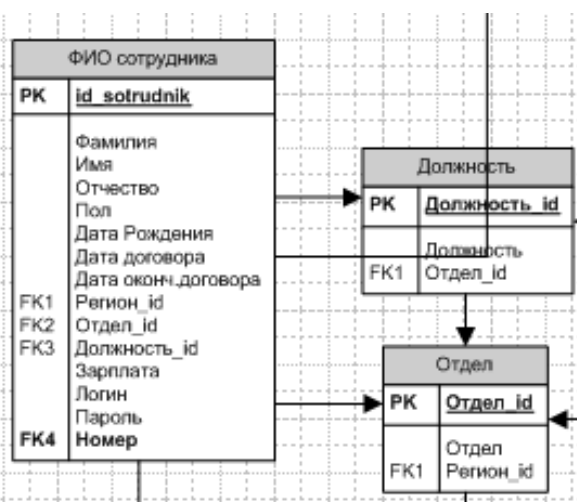
3.1-сурет – 1-бизнес ереже үшін ER сегменті

2) Әрбір бөлім жекелеген аймақтарда орналасады, бір аймақта бірнеше бөлімше болуы мүмкін.

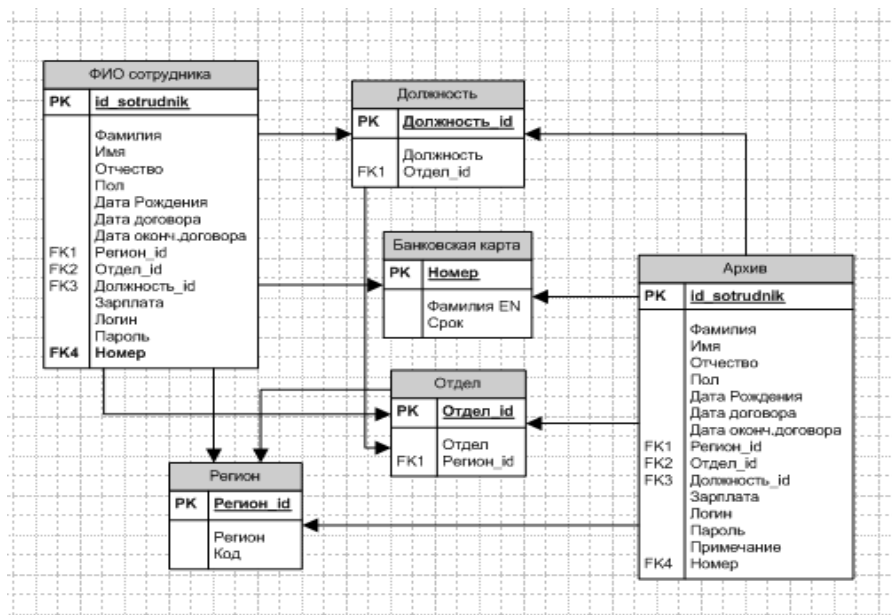


3.2-сурет – 2-бизнес ереже үшін ER сегменті

3) Әрбір лауазым жекелеген бөлімшелерде орналасады, әр лауазым бір қызметкерге тән.



3.3-сурет – 3-бизнес ереже үшін ER сегменті



3.4-сурет – ER диаграмма

3.2.5 Реляционды қатынастарды құрастыру

Әрбір реляционды қатынас бір нысанға сәйкес келеді және оған нысанның барлық атрибуттары енгізіледі. Әрбір қатынас үшін алғашқы кілт және сыртқы кілттерді (егер бар болса) анықтау қажет. Қатынастар 3.2-3.7-кестелерінде келтірілген.

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

К е с т е 3.2 – FIO қатынасы

Әріс аты	Атрибут аты	Мәліметтер типі	Ескерту
Қызметкер коды	id_sotrudnik	int	Алғашқы кілт
Фамилия	Familiya	varchar(50)	Қажетті өріс
Аты	Imya	varchar(50)	Қажетті өріс
Әкесінің аты	Otchestvo	varchar(50)	Қажетті өріс
Жынысы	Pol	varchar(1)	Қажетті өріс
Туылған мезгілі	dob	datetime	Қажетті өріс
Келісім басталу мезгілі	dog_date	datetime	Қажетті өріс
Келісім аяқталу мезгілі	dog_ist	datetime	Қажетті өріс
Жалақы	zarplata	char(6)	Қажетті өріс
Аймақ	region_id	int	Сыртқы кілт
Бөлімше	otdel_id	int	Сыртқы кілт
Лауазым	dolzh_id	int	Сыртқы кілт

К е с т е 3.3 – Region қатынасы

Әріс аты	Атрибут аты	Мәліметтер типі	Ескерту
Аймақ	region_id	int	Алғашқы кілт
Аймақ атауы	region_name	varchar(50)	Қажетті өріс
Аймақ теру код	kod	int	Қажетті өріс

К е с т е 3.4 – Otdel қатынасы

Әріс аты	Атрибут аты	Мәліметтер типі	Ескерту
Бөлім	otdel_id	int	Алғашқы кілт
Бөлім аты	otdel_name	varchar(50)	Қажетті өріс
Аймақ	region_id	int	Сыртқы кілт

К е с т е 3.5 – dolzhnost қатынасы

Әріс аты	Атрибут аты	Мәліметтер типі	Ескерту
Лауазым	dolzh_id	int	Алғашқы кілт
Лауазым аты	dolzh_name	varchar(50)	Қажетті өріс

К е с т е 3.6 – Kazkom қатынасы

Өріс аты	Атрибут аты	Мәліметтер типі	Ескерту
Номер	nomer	int	Алғашқы кілт
Фамилия (EN)	fam_en	varchar(50)	Қажетті өріс
Срок	srok	date	Қажетті өріс

К е с т е 3.7 – Arhiv қатынасы

Өріс аты	Атрибут аты	Мәліметтер типі	Ескерту
Қызметкер коды	id_sotrudnik	int	Алғашқы кілт
Фамилия	Familiya	varchar(50)	Қажетті өріс
Аты	Imya	varchar(50)	Қажетті өріс
Әкесінің аты	Otchestvo	varchar(50)	Қажетті өріс
Жынысы	Pol	varchar(1)	Қажетті өріс
Туылған мезгілі	dob	datetime	Қажетті өріс
Келісім басталу мезг	dog_date	datetime	Қажетті өріс
Келісім аяқталу мезг	dog_ist	datetime	Қажетті өріс
Жалақы	zarplata	char(6)	Қажетті өріс
Аймақ	region_id	int	Сыртқы кілт
Бөлімше	otdel_id	int	Сыртқы кілт
Лауазым	dolzh_id	int	Сыртқы кілт
Қосымша	dopolnenie	varchar(500)	Қажетті өріс

3.3 Логикалық жобалау

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

3.3.1 Кестелер құру

Кестелер CREATE TABLE командасымен құрылады. CREATE TABLE кесте атын және бағандар аттарын анықтайды. Сонымен қатар ол деректердің типін және бағандардың өлшемін анықтайды. Ең ыңғайлы бұл генерацияланған скриптті SQL Server қосу. Кестелерді құруға арналған скрипт А қосымшасында келтірілген.

The screenshot shows the SQL Server Enterprise Manager interface with a query window open. The query contains the following SQL statements:

```
create table FIO
(id_sotrudnik int not null primary key,
familiya varchar(25),
imya varchar(25),
otchestvo varchar(15),
pol varchar(2),
data date,
id_region int,
id_otdel int,
id_dolz int );|
go

create table region
(id_region int not null primary key,
region_name varchar (30),
kod varchar(15));
go

ALTER TABLE FIO
ADD CONSTRAINT PK_FIO_region FOREIGN KEY (id_region) REFERENCES region (id_region) ;

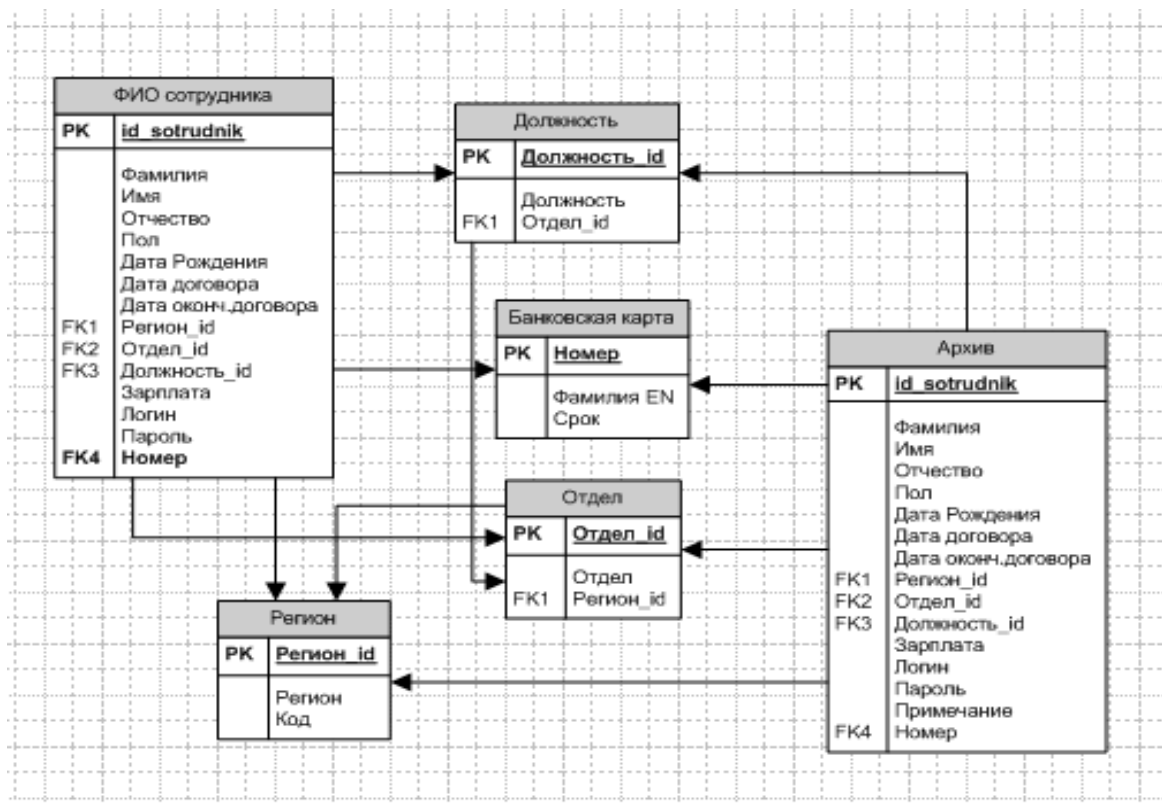
insert into region (id_region,region_name,kod)
values ('1','Тараз','7262'),
('2','с.Сарыкемер','72637'),
('3','с.В.Момышулы','72635'),
('4','с.Кордай','72636'),
('5','г.Каратау','72644'),
('6','с.Кулан','72631'),
('7','с.Толе би','72638'),
('8','г.Жанатас','72634'),
('9','с.Мерке','72632'),
('10','с.Аса','72633'),
```

The status bar at the bottom indicates "Connected. (1/1)".

3.5- сурет – Sql-де кестелерді құру

3.3.3 ДҚ диаграммасы

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



3.6-сурет – ДҚ диаграммасы

3.3.4 Ақпаратты құпияландыру. TDE жүзеге асыру

Transparent Data Encryption (TDE) арқылы деректерді шифрлау үшін келесі қадамдарды орындау керек:

- БД master басыт кілтін құру (егер бастапқыда құрылған болмаса);
- сертификат құру, жабық кілті жаңағы БД master басты кілтімен шифрлану керек;
- әрі қарай шифрланатын дерекқорда Database Encryption Key (DEK) кілтін құрамыз. DEK сертификатпен шифрланады.

Осы тұста сертификаттың резерттік көшірмесін жасау керек. Себебі ол жоғалған таңда ДҚ ашу мүмкін болмайды. Бұл қадамдар келесі 3.7- суретте көрсетілген:

```

SQLQuery06ggg.sql...ser-I IK\User (64)) | SQLQdb.sql - (loc...User-I IK\User (62)) | SQLdfng.sql - (loc...User-I IK\User (61)) encryption
use master

create master key encryption by password = 'Aidarkulova';

create certificate TDE_certificate with subject = 'TDE_Sertificate';

use master

backup certificate TDE_certificate to file = 'C:\mika\diplomka\TDE\TDE_certificate.cert'
with private key(
file='C:\mika\diplomka\TDE\TDE_certificate',
encryption by password = 'Aidarkulova18Meruert08Muratovna91');

USE Baza
GO

CREATE DATABASE ENCRYPTION KEY
WITH ALGORITHM = AES_256
ENCRYPTION BY SERVER CERTIFICATE TDE_certificate;
GO

ALTER DATABASE Baza
SET ENCRYPTION ON;
GO

```

3.7-сурет - TDE жүзеге асыру

Шифрлануды қосқаннан кейін оның параметрлерін тексеріп, дерекқордың резервтік көшірмесін жасаймыз:

```

SELECT *
FROM sys.dm_database_encryption_keys

use master
go
SELECT * FROM sys.certificates where [name] = 'TDE_certificate'
GO

BACKUP DATABASE [Baza] TO DISK = N'C:\mika\diplomka\Резервная копия базы\Baza_Encrypted.bak' WITH NOFORMAT, NOINIT,
NAME = N'Baza-Full Database Backup', SKIP, NOREWIND, NOUNLOAD, STATS = 10
GO

use master
go
SELECT * FROM sys.certificates where [name] = 'TDE_certificate'
GO

```

name	certificate_id	principal_id	pvt_key_encryption_type	pvt_key_encryption_type_desc	is_active_for_begin_dialog	issuer_name	cert_serial_number	sid
TDE_certificate	260	1	MK	ENCRYPTED_BY_MASTER_KEY	1	TDE_Sertificate	49f8 4e a0 98 c4 cc 85 41 bd 55 90 2e f3 3d 9f	0x0106000000

string_sid	subject	expiry_date	start_date	thumbprint	attested_by	pvt_key_last_backup_date	
2C282FF5...	S-1-9-1-333127740-741588933-2180329256-719591458...	TDE_Sertificate	2015-05-12 ...	2014-05-12 ...	0x3C20DB13C5BF332C282FF5812218E42AB0834B0B	NULL	2014-05-12 17:48:59.580

3.8-сурет – Шифрланған ДҚ параметрлері

3.4 Физикалық жобалау

3.4.1 Көрсетілімдер және процедуралар

Көрсетілімдер бұл кестелер, бірақ оның құрамы басқа кестелерден алынады. Олар сұраныстарда дәл негізгі кестелер сияқты жұмыс істейді, бірақ оларда өзінің жеке деректері жоқ. Дерекқорда келесі процедуралар бар:

- `zr_otdel` – бір бөлімшедегі барлық қызметкерлердің жалақысын есептеу;
- `obwee_otdel` – бір бөлімшедегі барлық қызметкерлердің санын есептеу;
- `poisk_all` – барлық қызметкерлер тізімінен Фамилиясын енгізу арқылы шығару;
- `poisk1` – бір аймақты таңдау арқылы сондағы барлық қызметкерлердің тізімін шығару;
- `poisk2` – бір бөлімді таңдау арқылы сондағы барлық қызметкерлердің тізімін шығару;
- `poisk3` – бір лауазымды таңдау арқылы қызметкерлердің тізімін шығару.

Дерекқорда келесі көрсетілімдер бар:

- `obwee` – барлық қызметкерлер туралы ақпаратты толық, әрі түсінікті түрде шығару.

3.4.2 Триггерлер

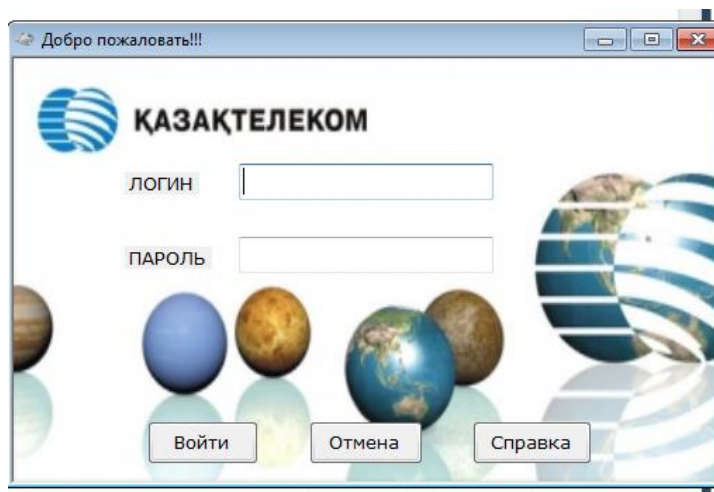
Триггер — бұл серверде автоматты түрде орындалатын сақталған процедура түрі. Триггер – ДҚ деректерінің тұтастылығын жақтаушы SQL-сервердің ерекше құралы.

Қате немесе дерекқор тұтастылығын бұзушылық жағдайда транзакция шегінуі болады. Өзгерту енгізу тыйым салынады. Дерекқорда келесі триггерлер бар:

- `otdel` – бір қызметкерді бірнеше бөлімшеге енгізуге тыйым салу;
- `dolzhnost` – бір қызметкерге бірнеше лауазым беруге тыйым салу;
- `region` – бір қызметкерді бірнеше аймаққа енгізуге тыйым салу;
- `kazkom` – бір карта тек бір қызметкерге тән;
- `date_dog` – келісім-шарт жасалған күн оның күші бітер күннен кейін болмау керек және керісінше;
- `zr` – әр лауазымның жалақысы белгілі бір аралықта болады(мысалы, директор жалақысы оператордан кем болуы мүмкін емес)

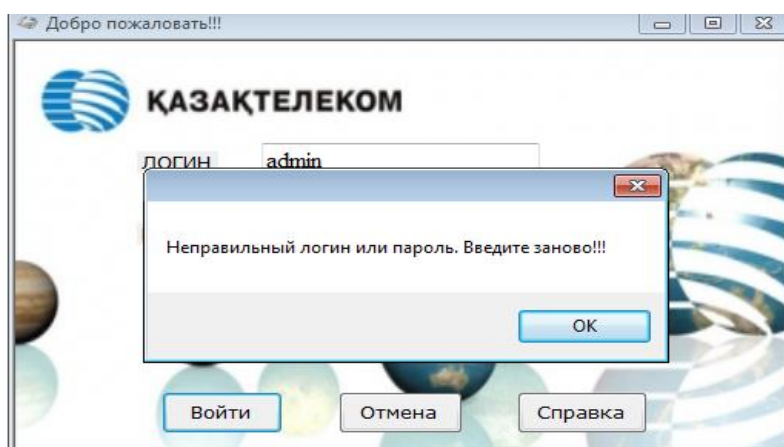
3.5 Қосымша интерфейсін және оның функционалдығын сипаттау. Қосымшамен жұмыс істеу

Жоғарыда айтылғандай, жобаның интерфейсі Visual Studio 2010 объектілі-нысандаған бағдарламалау ортасында орындалды. Қосымшаның бағдарламалық коды Б қосымшасында келтірілген. Бағдарламаны қосу кезінде, алдымен бағдарламаның алғашқы беті ашылады.

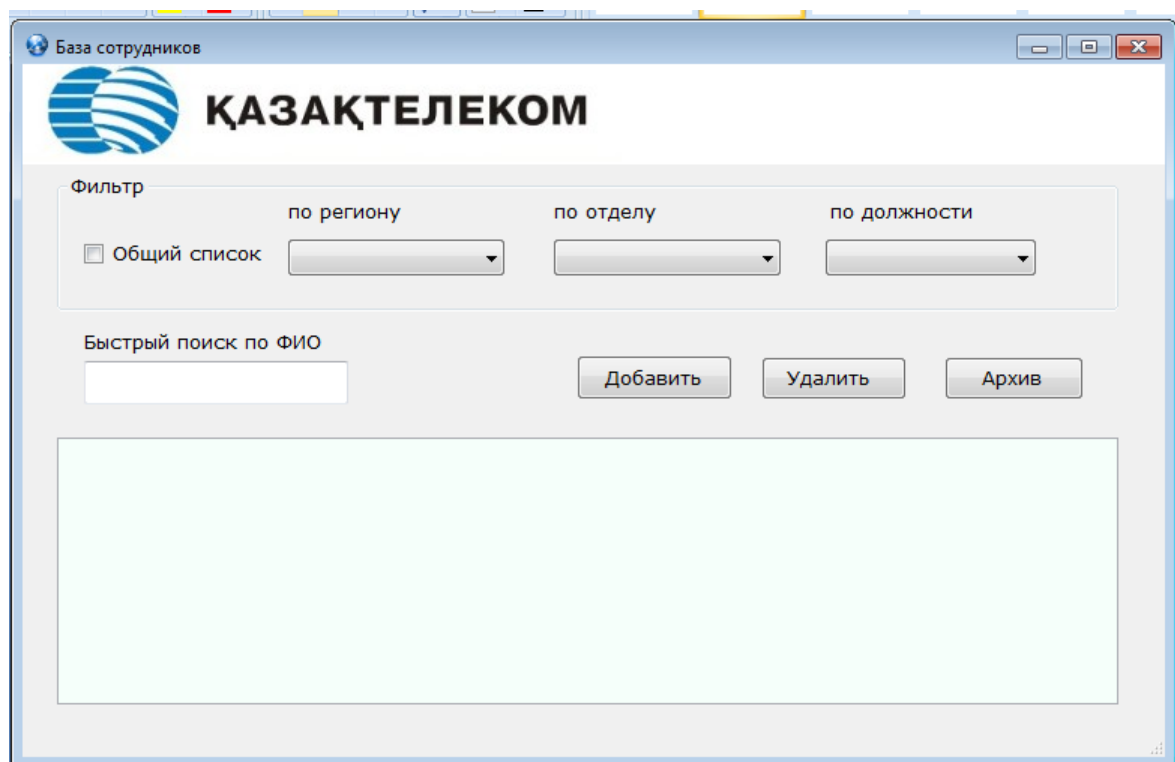


3.9-сурет – Бағдарламаның негізгі аутентификация терезесі

«Абонент» терезесінде (3.9-сурет) сәйкесінше оператор өзінің кілттік сөзін тереді. Аутентификация арқылы оператордың деректер қорында осындай оператор бар немесе жоқ екендігі тексеріледі. Осындай оператор бар болған жағдайда, әрі қарай процестің орындалуына қолжетімділік рұқсат етіледі. Терезеде ақпараттық қауіпсіздік негіздері ескерілген. Кілттік сөз дұрыс енгізілбеген жағдайда ескерту терезесі шығады.



3.10-сурет – Ескерту терезесі



3.11-сурет – Басты терезе

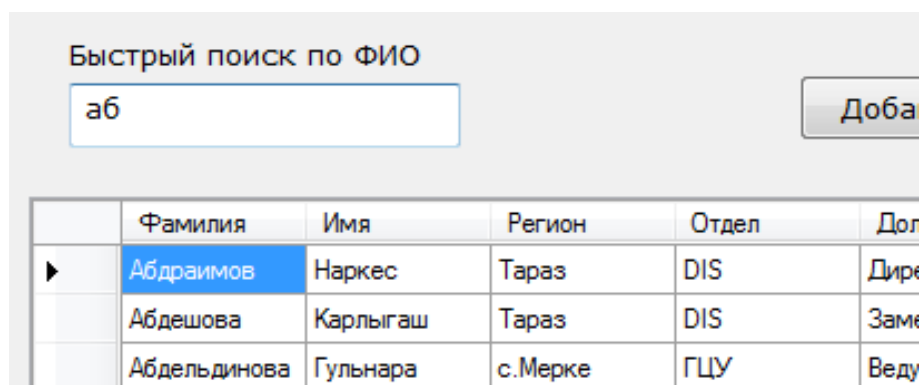
Басты терезеде пайдаланушы өзі іздейтін қызметкерді оңтайлы фильтр параметрлерді енгізу арқылы табады:

«Общий список»-ті басқан жағдайда барлық қызметкерлер тізімі шығады;
 «По региону» тізімінен аймақты таңдап ондағы барлық қызметкерлер тізімі шығады;

«По отделу» тізімінен бөлімшені таңдап ондағы барлық қызметкерлер тізімі шығады;


«По должности» тізімінен лауазымды таңдап сол лауазымға ие барлық қызметкерлер тізімі шығады;

«Быстрый поиск по ФИО» Фамилияның бастыпқы әріптерін енгізсе де қызметкерлер тізімі шығады:



3.12-сурет – Фамилия бойынша Іздеу

Добавить нового сотрудника /*регион/*отдел/*должность



ҚАЗАҚТЕЛЕКОМ

СОТРУДНИК

Фамилия

Имя

Отчество

Пол М Ж

Дата Рождения

Регион

Отдел

Должность

ДОГОВОР

с по

Зарплата

Логин

Пароль

Прочее

РЕГИОН

Название г. / с.

Код

ОТДЕЛ

Название


ДОЛЖНОСТЬ

Название

3.13-сурет – «Жаңа қызметкер қосу» терезесі

Жаңа қызметкер қосуды (3.13-сурет) берілген аймақтардың барлығын толтыру арқылы жүзеге асырамыз. Одан басқа қызметкерер тізімінен жұмыстан шыққан қызметкерлерді қордан шығару керек.

Удаление сотрудника из базы



ҚАЗАҚТЕЛЕКОМ

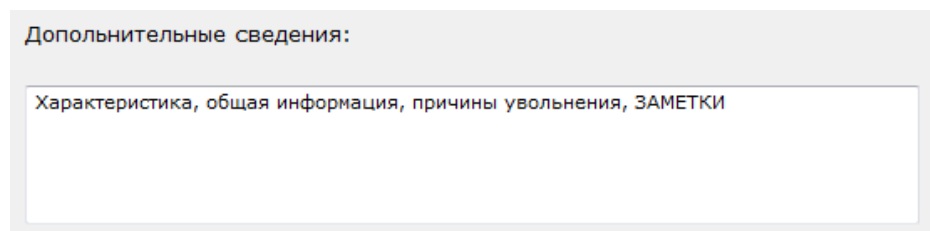
Фамилия

	Фамилия	Имя	Отчество	Регион	Отдел	Договор от	до
▶	Абдраимов	Наркес	Абуталипович	Тараз	DIS	18.04.2007	18.04.2014

Дополнительные сведения:

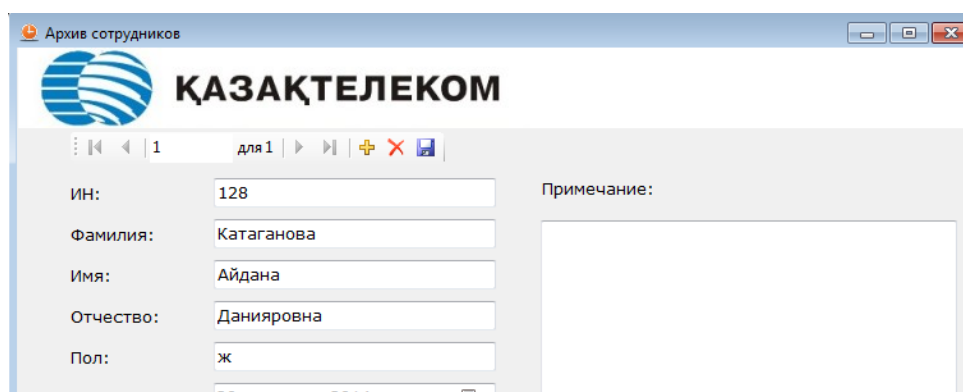
Характеристика, общая информация, причины увольнения, ЗАМЕТКИ

3.14-сурет – «Қызметтен шығару» терезесі



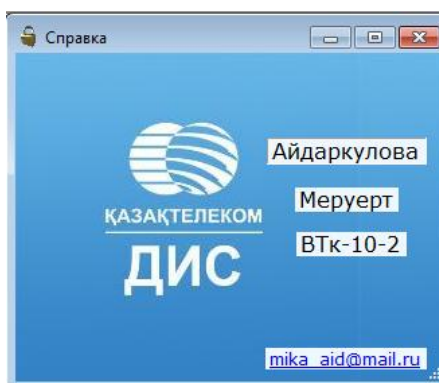
3.15-сурет – «Қызметтен шығару» терезесінің бөлігі

Қызметтен шығарғаннан кейін кейбір жағдайларда бұрын жұмыс жасағандардың тізімін қайта қарау үшін Архивті қолданамыз (3.16-сурет).



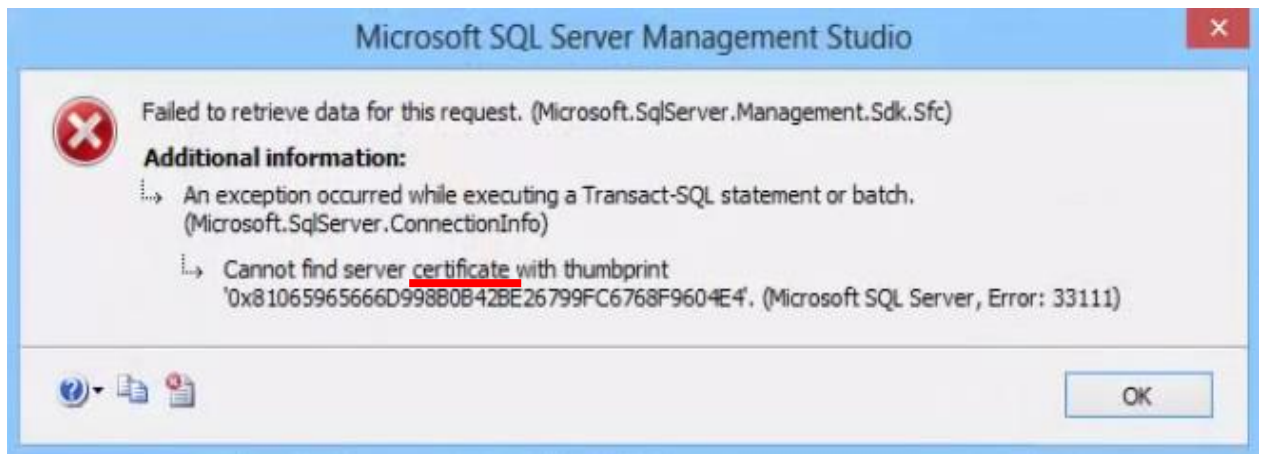
3.16-сурет – «Архив» терезесі

«Архив» терезесіндегі «Примечание» деген терезесінде (3.15-сурет) «Қызметтен шығару» терезесінің бөлігінде енгізілген әр түрлі ақпарат көрсетіледі.



3.17-сурет – «Справка» терезесі

Нарық қызған шақта, кез-келген компаниялар бір орында тұрмауы тиіс. Заман дамыған сайын адам жұмысты өзіне оңтайлы етеді. Бірақ қазіргі таңда бәсекелес компаниялардың ақпараттарын ұрлау өрге жүзуде, сондықтан дерекқордың қауәпсәздәгән қамтамасыз ету керек. Бұл мақсатта әлемдегі ең жақсы деген шфирлану тілі қатарына жататын AES-256 қолданамыз. Оны MS SQL Server-дегі TDE (Transparent Data Encryption) арқылы жүзеге асырылды.



3.18-сурет – Шифрланган дерекқор

4 Тіршілік қауіпсіздігі

4.1 Инженердің жұмыс жағдайын талдау

Берілген бітіру жұмысының тақырыбы – АҚ«Қазақтелеком» қызметкерлерінің деректер қорын құрып, шифрлану арқылы ақпарат қауіпсіздігін қамтамасыз ету.

Инженердің жұмыс орны деп іскердің өндірістік процесті жүргізу және бақылау мақсаты үшін жабдықталған кеңістік бөлігін айтады. Сонымен қатар, жұмыс орыны болып, оның еңбек қызметін іске асыратын және басқару органы мен қосымша жабдықпен, ақпаратты көрсету құралдарымен жабдықталған жүйедегі орны саналады. Инженердің жұмыс орнын құрастырғанда келесі шарттардың маңызы зор.

Инженердің жұмыс орнын ұйымдастырғанда жұмыс бөлмесі ауданының маңызы зор. Бұл жобада қарастырылған жұмыс бөлмесінің ұзындығы 9м, ені 6м, биіктігі 3м, бөлмеде 2 терезе бар, оның көлемі 2х3.

Қамтама орнатылатын қала-Алматы.

Максималды қол жету аймағын инженерден оң жағына құжаттамалық құралдарын орналастырамыз, ал сол жағына- байланыс құралдарын, оң қолды жазудан босату үшін орналастырамыз.

Бұл жобада телекоммуникация тақырыбындағы ақпараттық деректер қоры қарастырылған. Жұмысшының негізгі жұмысы ақпаратты қабылдау немесе енгізу, бағдарламалар бойынша ЭЕМ-да тапсырманың есептелуін түзету және бақылау, машина тоқтап немесе істен шыққан жағдайда дер кезінде шара қолдану, сонымен қатар тапсырыс берушіге қортындыны жібергені туралы ақпарат беру. Міне осыған орай, инженер организмнің энергия шығынына байланысты жұмыс категориясын таңдаймыз. ГОСТ 12.1.005-88 ССБТ «Жұмыс орнының ауасы, санитарлық-гигиенаның жалпы талаптары» 4.1 – кестеде келтірілген. Инженердің жасайтын еңбегі бірінші категорияға жатады (Ia).

К е с т е 4.1 – Организмнің энергошығыны бойынша жұмыс категориясы

Жұмыс категориясы	Категория	Адам организмнің энергия шығыны, Ккал/сағ.	Жұмыс сипаттамасы
Жеңіл физикалық жұмыс I a	I a	138-172	Жұмыс отыру күйінде өтеді

Өрт – бұл арнайы жасалмаған, материалдық шығындарымен қатарласып жүретін, кейде адам өлімі болуы мүмкін, бақылана алмайтын ошақ көзі. Бөлмеде түгіндік өрт хабарлайтын құрылғы орнатылған. Одан басқа көмірқышқылды ОУ-8 өрт сөндіргіш баллоны бар. Нөлдеу контур-шинаны нөлдік қосқышпен таратушы күш щитінде қосу арқылы ұйымдастырылған.

Жұмыс бөлмесінде белсенді кондиционерлеу және вентиляция жүйесі жоқ. Жұмыс бөлмесін талдау барысында жұмыс аумағының қалыпты

микроклиматтық шарттарын міндетті түрде қарастыру қажет. Мұндай әмбебап жүйе ретінде автономдық кондиционерлер болып табылады. Кондиционерді орнату алдында арнайы есептеулер жасау керек. Сол есептеулердің қорытындысымен сипаттамасы және талаптарына сай кондиционер таңдалады. Бөлмені кондиционерлеу төменде келтірілген.

Микроклиматтың күйін қадағалау инженерлар залында қолайлы жағдайды орнатуға көмектеседі. Ал жұмыс орнының қолайлы жағдайларында адамдардың жұмыс істеу қабілеттері жақсарады, жұмысшылардың ауруға шалдығуы азаяды сонымен қатар, еңбек өндірісі артады. Жұмыс орнындағы нормаланған климаттық параметрлер 4.2 кестеде келтірілген.

К е с т е 4.2 – Температураның, ылғалдылық пен ауа қозғалысының нормалары МЕСТ 12.1.055.88ССБТ

Жыл мезгілі	Ауа температурасы, °С			Салыстырмалы ауа ылғалдылығы, %		Ауа қозғалысының жылдамдығы, м/с	
	Қолайлы	Рұқсат етілген		Қолайлы	Рұқсат етілген	Қолайлы	Рұқсат етілген
		Жоғарғы	Төменгі				
Суық	21–23	24	20	40–60	75	0,1	<0,1
Жылы	22–24	28	22	40–60	60 - 70	0,2	0,1–0,3

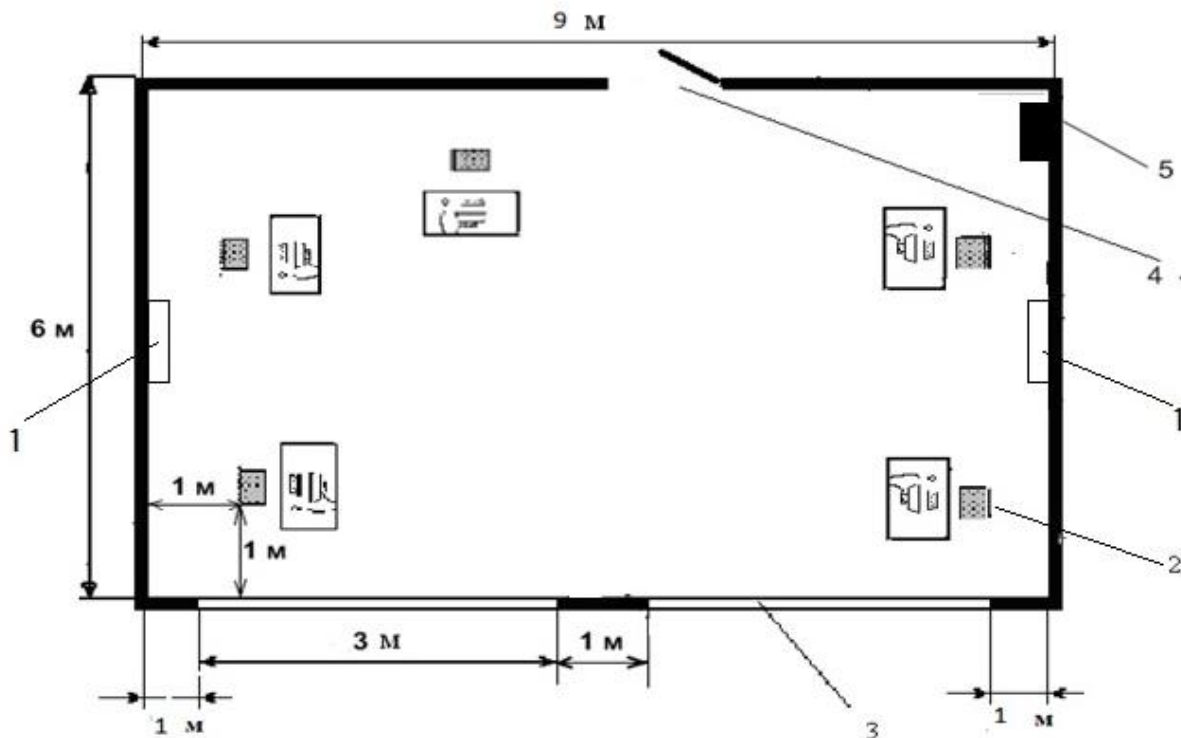
Жазғы микроклиматтың қысқыға қарағанда, ауа баптауы сәл жоғарырақ. Кесте бойынша, ауа температурасы 23-25° ылғалдылығы 30–50%, ауа қозғалысының жылдамдығы 0,2м/с. 55% дейінгі ауа ылғалдылығы құрғақ деп есептеледі, 56-70% - деңгейінде орташа құрғақ, 71-85% - орташа ылғалды және 85% жоғары – тым ылғалды. Ең үлкен ылғалдылық пен салыстырмалы ылғалдылық арасындағы айырым қанығу тапшылығы деп аталады. Қанығу тапшылығы қаншалықты үлкен болса, соншалықты ылғалды ауа жұтады. Ауаның температурасы 23-24°С – де ауа қозғалысының жылдамдығы 0,15 м/с дейін болған жөн. Адамның өмір тіршілігіне байланысты ауа құрамының өзгеруі, оған берілген көмір қышқыл газымен есептеледі – CO₂. Бөлмедегі CO₂ шоғырлануы 4.3 – кестеде келтірілген.

К е с т е 4.3 – Бөлме ауасындағы көмір қышқыл газының (CO₂) шоғырлану шектері

Бөлме	CO ₂ шоғырлану шегі	
	Салмағы бойынша, г/м ³	Көлеміне % қатысы
Балалар мен аурулар болу үшін	1,3	0,07
Адамдардың ұзағырақ болуы үшін	1,86	0,1
Адамдардың кезеңдік болуы үшін	2,32	0,125
Адамдардың қысқа уақыт болуы үшін	3,72	0,2

CO₂ жұтатын ауада артық не кем болуы ағзаға біркелкі зиянды. CO₂, жетімді шоғырлануы 0,03% кем болса, онда аталған органдардың жұмысы бұзылады, CO₂ > 1,5%-тен артық болса, онда наркотикалық әсері болады, бас ауырады және т.б. Егер дем алатын ауада CO₂= 0,5-1,5% мәнінде болса, онда ағзаға оның елеулі әсері болмайды. Ал ең қолайлы шоғырлану шамасы CO₂= 0,04-0,5% сәйкес келеді. Қолданыстағы санитарлық нормалар бойынша бір адамға 20-60 м³/сағ таза ауа қажет.

Келтірілген шамалар адам организміне ыңғайлы нормаларға сай келмейді. Сондықтан инженерлар бөлмесінде ауаны кондиционерлеу мәселесі қарастырылған.



4.1-сурет – Инженер залының жоспары

мұндағы

- 1 – кондиционерлер орны;
- 2 – инженерлардың жұмыс орындары;
- 3 – терезе жақтаулары;
- 4 – есік жақтауы;
- 5 – өрт сөндіргіш баллоны.

4.2 Есептеу бөлімі

4.2.1 Кондиционерлеу және кондиционерді таңдап, есептеу

Ауаны технологиялық кондиционерлеудің талаптары өнеркәсіптің әр түрлі салаларындағы технологиялық процестерді жүргізуде, сондай-ақ радиоэлектрондық жабдықтың, басқа құралдар мен аспаптардың және т.б. жұмыс қабілеттілігін қамтамасыз ету үшін ауа ортасының белгілі бір параметрлерін (ауаның температурасын, ылғалдылығын және қозғалысын) өндіріске сай қолдауға негізделеді. Өзінің тағайыны бойынша кондиционер жүйесі қолайлы және технологиялық болып бөлінеді. Қолайлы жүйелер үйлесімді санитарлық-гигиеналық талаптарға жауап беретін ауаның температурасын, ылғалдылығын, тазалығы мен қозғалыс жылдамдығын жасау және автоматты қолдау үшін тағайындалады. Кондиционердің технологиялық жүйелері белгілі бір өндірістік және технологиялық процесс талаптарына басым дәрежеде жауап болатын ауа параметрлерін қамтамсыз ету үшін тағайындалады.

Аумағы 15-тен 140 м² дейінгі тұрғын және қоғамдық бөлмелерде сплит-жүйе кондиционерлері кең таралуда. Олар сыртқы блоктан (компрессорлы-конденсаторлы) және ішкі блоктан (буландырғыш) тұрады. Сыртқы блок ғимарат қабырғасына, шатырға немесе шатыр астына, қосалқы бөлмеге, балконға, яғни ыстық конденсатор төменірек температурасы бар атмосфера ауасымен салқындатылатын жерге орнатылуы тиіс. Ішкі блок тікелей кондиционерленетін бөлмеге орнатылады және ауаны салқындату немесе жылыту, сүзу және бөлмедегі ауаның қажетті қозғалысын жасау үшін тағайындалады. Ішкі блоктар берілген температураны ұстайды және бөлмедегі ауаның тең бөлінуін қамтамасыз етеді және шусыз жұмыс істейді (шу деңгейі 35-38 дБ). Кондиционерленген ғимараттың жылулық және ылғал теңгерімін белгілі әдістермен орындалады. Мұнда ғимараттың ауа ортасының қалпы өзгеруіне әкеп соғатын, барлық факторлар есепке алынуы керек.

Температура айырымы нәтижесіндегі жылудың түсуі және жоғалуы.

Жылудың саны $Q_{\text{коршау}}$ мына формуламен анықталады:

$$Q_{\text{коршау}} = V_{\text{ғимарат}} \cdot X_0 (t_{\text{шыққан}} - t_{\text{келген}}), \text{ Вт} \quad (4.1)$$

мұндағы $V_{\text{бөлме}} = 6 \times 9 \times 3 = 162 \text{ м}^3$ – бөлме көлемі;

$X_0 = 0,42 \text{ Вт/м}^3 \text{ } ^\circ\text{C}$ – меншікті жылулық сипаттама;

$t_{\text{сырт.есеп}} = 27,6 \text{ } ^\circ$ - жылдың жылы мезгіліне арналған сыртқы есептік температурасы;

$t_{\text{сырт.есеп}} = - 10 \text{ } ^\circ$ - жылдың суық мезгіліне арналған сыртқы есептік температурасы

$t_{\text{іш.есеп}} = 19 \text{ } ^\circ$ - жылдың суық мезгіліне арналған ішкі есептік температурасы.

Жылы жыл мезгілі үшін:

$$Q_{\text{коршау}} = 162 \times 0,42 \times (27,6 - 22) = 381,02 \text{ Вт}$$

Суық жыл мезгілі үшін:

$$Q_{\text{коршау}} = 162 \times 0,42 \times (-10 - 19) = -1973,16 \text{ Вт}$$

Әйнек арқылы күннің радиациясынан түсетін жылу

Күннен бөлінетін жылу әйнектің түріне байланысты 90%-ға дейін бөлме ортасымен жұтылады, қалған бөлігі шағылысады. Ең үлкен жылу жүктемесі тура және шашырай түсетін күн сәулесінің ең үлкен деңгейінде алынады. Сәуле түсу қарқыны жергілікті кеңдікке, жыл мезгіліне және тәулік уақытына байланысты.

Күн шағылысуынан бөлінетін жылу (радиация) мынадай формуламен анықталады:

$$Q_p = (q^1 \cdot F_o^1 + q^{11} \cdot F_o^{11}) \beta_{K.K} \quad (4.2)$$

мұндағы q^1, q^{11} - тура және шашырай түскен күн радиациясының жылулық ағыны, Вт/м²;

F_o^1, F_o^{11} - тура күн радиациясымен сәулеленетін және сәулеленбейтін жарық ойықтары ауданы, м²;

$\beta_{K.K}$ - жылу өткізгіш коэффициенті.

Сырт көлеңкелейтін күнқағарлар, жақтаулар және т.б. болмағанда, күн сәулесінің терезеден бөлмеге түсуі, $F_o^1 = F_o$; $F_o^{11} = 0$.

$$Q_p = q^1 \cdot F_o \cdot \beta_{K.K} = (q_{\text{тік}} + q_{\text{жайыл}}) \cdot K_1^c \cdot K_2 \cdot \beta_{\text{Е.К}} \cdot n \cdot H_o \cdot V_o, \text{ Вт} \quad (4.3)$$

Күн сәулесі терезеден түспеген көлеңке жағдайындағы (шашыраған радиация), $F_o^1 = 0$; $F_o^{11} = 0$.

$$Q_p = q^{11} \cdot F_o \cdot \beta_{K.K} = q_{\text{жайыл}} \cdot K_1^T \cdot K_2 \cdot \beta_{K.K} \cdot n \cdot H_o \cdot V_o, \text{ Вт} \quad (4.4)$$

мұндағы $q_{\text{тік}}, q_{\text{жайыл}}$ - тура және шашыраған радиациялардың жылулық ағындары, Вт/м²;

$F_o = n \cdot H_o \cdot V_o$ - жарық ойығы ауданы, м² (n - терезелер саны, биіктік H_o және ені V_o);

K_1 - шыны жақтаулармен көлеңкеленген коэффициент

(K_1^c - түскен сәуле ойықтары үшін, K_1^T - көлеңкедегі ойықтар үшін)

K_2 - кірленген терезелердің коэффициенті

Бұл бөлмеде жоғарыда аталғандай, 2 терезе - 2 x 3 метр, солтүстікке бағытталған («С»):

$$F_o = 2 \times 2 \times 3 = 12 \text{ (м}^2\text{)};$$

Ашық жалюздер үшін жылу өткізу коэффициенті:

$$\beta_{K.K} = 0,15$$

«С» үшін түскі уақытқа дейін, яғни сағат 9-дан 12-ге дейін 44° с. е. (Алматы қаласы) тура радиацияның мәні (Т):

$$q_{\text{вп}} = 0 \left(\frac{\text{Вт}}{\text{м}^2} \right)$$

Шашыраған радиация (Ш):

$$q_{вр} = 64 \left(\frac{Вт}{м^2} \right)$$

Түстен кейін «С» бағыты үшін, сағат 12-ден 13-ке дейін:

$$q_{вп} = 0 \left(\frac{Вт}{м^2} \right) \qquad q_{вр} = 59 \left(\frac{Вт}{м^2} \right)$$

Сағат 9-10-нан 13-14-ке дейінгі ойық жарығының қаранғылауын есептеу үшін қолданылатын K_1 коэффициенті:

$$K_1 = K_1^C = 0,72$$

Ал, сағат 14-15-тен 19-20-ға дейінгі уақытта төмендегідей мәнге ие:

$$K_1 = (K_1)^T = 1.15$$

Терезенің ластануын есепке алатын коэффициент:

$$K_2 = 0.9$$

Сағат 9-дан 14-ке дейінгі уақытта жылу бөлу 4.3 – формула бойынша анықталады:

$$Q_{P1} = q_{\hat{A}D} K_1^C K_2 \beta_{KK} n H_0 B_0$$

$$Q_{P1} = 64 * 0,72 * 0,9 * 0,15 * 12 = 74,65 \text{ Вт}$$

Сағат 14-тен 20-ға дейінгі уақыттағы жылу бөлу де 4.3 – формула бойынша анықталады:

$$Q_{P2} = 59 * 0,72 * 0,9 * 0,15 * 12 = 68,82 \text{ Вт}$$

Максималды есептеу сағаты 9-дан 10-ға дейінгі уақыт болады. Себебі, бұл уақытта күн радиациясы әсерінен болатын жылу бөлу 74,65 Вт.

Адамдардан жылудың берілуі.

Адамдардан түсетін жылу қоршаған ауа параметрлеріне және орындалатын жұмыс қарқынына байланысты. Адам бөлетін жылу ауаға конвекция арқылы сезілетін және өкпеден, теріден бөлінетін байқалмайтын жылудан тұрады. Адамдардың жылу таратуы 4.4 – кестемен сипатталады:

К е с т е 4.4 – Адамның сыртқы ортаға жылу таратуы, Вт

Сыртқы орта температурасы °С	Отырғандағы жағдай			Тұрғанда немесе жеңіл қозғалыс			Ауыр жұмыс		
	Анық	Жасырын	Жалпы	Анық	Анық	Жасырын	Жалпы	Жасырын	Анық
22	76	26	102	84	76	26	102	132	76

Жұмыс залының қалыпты температурасы 22°С тең. Бір уақытта бөлмеде 6 адам отырады. 22°С температура кезінде бір ер адам отырған күйінде 76 Вт

анық және 102 Вт жалпы жылу бөледі. Олай болса, залдағы ер адамдардың анық жылу бөлуі төмендегідей есептеледі:

$$Q_a^a = 6 \times 76 = 456 \text{ Вт.}$$

Жалпы жылу:

$$Q_a^ж = 6 \times 102 = 612 \text{ Вт.}$$

Адамдардан бөлінетін көмірқышқылгаз және ылғал мөлшері.

22° температурада бір адам бөлетін көміртегі мен ылғал мәні:

$$CO_2 = 45 \text{ г/сағ,} \quad W = 45 \text{ г/сағ.}$$

Осы мәндерден алты адам бөлетін көміртегі мен ылғалды табамыз:

$$CO_2 = 6 \times 45 = 270 \text{ г/сағ,} \quad W = 6 \times 45 = 270 \text{ г/сағ.}$$

Жарықтандырғыш құралдардан және басқа да аспаптардан бөлінетін жылу.

Шамдардан түсетін жылу мына формуламен анықталынады:

$$Q_{\text{жарықтану}} = \eta N_{\text{жарықтану}} F_0, \text{ Вт} \quad (4.5)$$

мұндағы η — электрлік энергияның жылу энергиясына өтуі коэффициенті;

$N_{\text{жарықтану}}$ — шамдардың берілген қуаты;

F_0 — бөлме еденінің ауданы.

Люминесценттік шамдар үшін $\eta = 0,5-0,6$. Жақсы жарықтанған ғимараттар үшін қабылданған $N_{\text{жарықтану}} = 50-100 \text{ Вт/м}^2$; $F_0 = 6 \times 9 = 54 \text{ м}^2$.

$$Q_{\text{жарықтану}} = 0,5 \times 50 \times 54 = 1350 \text{ Вт}$$

Өндірістік жабдықпен бөлінетін жылу келесі формула арқылы есептеледі:

$$Q_{\text{құралдар}} = N_{\text{орнату}} \times K \quad (4.6)$$

Оргтехника есебінен туатын жылу ағымдары, 1 компьютерге толық жиынында немесе жабдықтың 30% қуатында орташа $K=300 \text{ Вт}$ алынады. Шамамен бір уақытта 6 компьютер жұмыс істейді деп аламыз:

$$Q_{\text{құралдар}} = 6 \times 300 = 1800 \text{ Вт.}$$

Осы есептеулердің негізінде инженер залындағы жылу берілу байланысын құраймыз.

$$Q = Q_{\text{қоршау}} + Q_p + Q_{\text{жарықтану}} + Q_{\text{адам}} + Q_{\text{құралдар}}$$

Жылдың жылы кезеңінде:

$$Q = 381,02 + 74,65 + 1350 + 456 + 1800 = 4061,67 \text{ Вт.}$$

Жылдың суық кезеңінде:

$$Q = -1973,16 + 74,65 + 1500 + 456 + 1800 = 1707,49 \text{ Вт.}$$

Жазда $Q_{\text{изб.}} > Q_{\text{изб.}}$ қыстан болғандықтан, онда $Q_{\text{изб.}}$ жаздағы мәні бойынша ауадағы жылу кернеуін келесі формуламен анықтаймыз:

$$Q_n = (Q_{\text{изб.}} \cdot 860) / V_{\text{помещ.}} = (3,954 \cdot 860) / 162 = 20,99 \text{ ккал/м}^3,$$

$$Q_{\text{изб.}} > 20 \text{ ккал/м}^3 \text{ болғанда } \Delta t = 8^\circ\text{C.}$$

Бөлмеге кіретін қажетті ауаның шығынын анықтау.

$$L = (Q_{\text{изб.}} \cdot 860) / (C \cdot \Delta t \cdot \gamma) = (3,954 \cdot 860) / (0,24 \cdot 1,206 \cdot 8) = 1468,54 \text{ м}^3 / \text{сағ}$$

мұндағы, $C = 0,24 \text{ ккал/кг } ^\circ\text{C}$ – ауаның жылу сыйымдылығы;

$\gamma = 1,206 \text{ кг/м}^3$ – ауаның меншікті массасы.

Берілген жағдайда қуаттырақ кондиционер орнату қажет. Сондықтан біздің типті SAMSUNG сплит – жүйесінің кондиционерін орнаттық.

Кондиционерді таңдау және оның сипаттамасы
 $L = 1469 \text{ м}^3/\text{ч}$ ауадағы шығынды жеткізіп тұру үшін төменгі ауа ағынды
 Neoclima NS12АНС/NU12АНС фирмасының $654 - 826 \text{ м}^3/\text{сағ}$ ауадағы
 шығынды жеткізе алатын екі кондиционерді аламыз. Оның негізгі
 сипаттамалары:

- ауа шығыны – $654 - 826 \text{ м}^3/\text{сағ}$;
- ең көп суық ауа беруі – $3,9 - 5,2 \text{ кВт}$;
- компрессор қуаты – $1,8 \text{ кВт}$;
- энергия жылытқыш – $3,3 \text{ кВт}$;
- ылғандандырғыш – $1,8 \text{ кВт}$;
- бу шығыны – $2,0 \text{ кг}/\text{сағ}$;
- өлшемдері – $745 \text{ мм} * 210 \text{ мм} * 250 \text{ мм}$;
- массасы – $42,5 \text{ кг}$

4.2.2 Жарықтандыруды есептеу

Табиғи жарық толығымен жұмыс істеу уақытысын да жарықты қанағаттандырмайды, тағыда ауа райдын өзгеруін, кешкі уақыттағы жұмысты ескеру қажет. Соған байланысты жасанды жарықты бөлмеде қолданамыз, ол люминесцентті шамдардан тұрады. Жарық көзінің нормативы төмендегі 4.5-кестеде көрсетілген.

Кесте 4.5 – Ортақ жарықтандыру жүйесіндегі ұсынылған жарық көздері

Түсті ажыратудағы талаптарға сәйкес көру жұмысының сипаттамасы	Жарықтандыру, лк	Күн көзінің түсу температурасының түс диапазоны $T_c, \text{ }^\circ\text{K}$	Қолданылатын шамдар түрі
Түсті ажырату талаптары көп емес объектілердің түстеріндегі ерекшелігі	300, 400	3500 - 5500	ЛД, ЛДЦ, ЛБ,
	150, 200	3000 - 4500	ЛБ, (ЛХБ), НЛВД+МТЛ, ДРЛ

Операторлардың жұмыс орындарының бөлмесін жасанды жарықтандыру үшін люминесцентті шамдарды қолданамыз, олардың жарық бергіштігі жоғары ($75 \text{ им}/\text{Вт}$ -қа дейін және одан жоғары), жарамдылық мерзімінің ұзақ ($10\ 000$ сағатқа дейін), шағылысу бетінің жарықтығы аз және жақсы жарық беруді қамтамасыз ететін сәулелендіргіш жарықтың табиғиға жақын спектральді құрамы.

Қолдануға ең ыңғайлы шамдар люминесцентті ЛБ (ақ түсті) және ЛТБ (жылы ақ түсті) қуаты $20,40$ немесе 80 Вт шамдар.

Операторлық залды жарықтандыру үшін екі ЛТБ-40 люминесцентті шамдары бар УСП-35 типті шамшырақтар қарастырылған.

Бөлменің өлшемдері: ұзындығы $L=9\text{ м}$, ені $B=6\text{ м}$, биіктігі $H=3\text{ м}$. Жұмыс бетінің еден деңгейінен биіктігі $h_{\text{рп}}$, $h_{\text{рп}}=0,725\text{ м}$, терезелер $h_{\text{но}}, h_{\text{но}}=0,8\text{ м}$ биіктігінен басталады, терезелердің биіктігі h_o , $h_o=2\text{ м}$. Жұмыс орының бөлмесі IV сағат белдеуі – в г. Алматы (жарықтану белдеуінің климаты- IV 50^0 солтүстік ұзындығы және оңтүстік (Алматы, Караганда)).

Терезенің толық ауданы S_0 , м^2 , осы формуламен анықтаймыз:

$$100 \cdot \frac{S_0}{S_n} = \frac{e_H \cdot \eta_0}{\tau_0 \cdot \tau_1} \cdot k_{\text{зд}} \cdot k_3, \quad (4.10)$$

$$S_0 = \frac{S_n \cdot e_H \cdot \eta_0}{100 \cdot \tau_0 \cdot \tau_1} \cdot k_{\text{зд}} \cdot k_3. \quad (4.11)$$

S_n – бөлменің ауданы м^2 ;

$$S_n = 36 \text{ м}^2;$$

e_H – Ортақ жарықтандыру жарық көздері.

Көру жұмыстың биік дәлме-дәлділігі үшін $e_H = 1,2$ қабылдаймыз.

m_N - климаттың жарықтанукоэффициент.

Жарықтану белдеуін ескере отырып, анықтауымыз : $m_N=0,9$.

$$e_H^{\text{IV}} = e_H \cdot m \cdot c \quad (4.12)$$

$m = 0,7$; $c = 0,75$ (қабырғаның сыртқы жағы);

$e_H = 1,2$ жұстағы үлкен дәлдіктер III разрядтағы көрініс жұмысы;

$$e_H^{\text{IV}} = 1,2 \cdot 0,7 \cdot 0,75 = 0,63.$$

Бөлменің типін ескеріп, осыдан $k_3 = 1,2$ коэффициенті табамыз (жұмыс істеу орны қарастырылады).

τ_0 - жарық өткізудің толық коэффициенті тең $\tau_0 = \tau_1 \cdot \tau_2 \cdot \tau_3 \cdot \tau_4$;

$\tau_1 = 0,5$ (бос орынды шынылы блок);

$\tau_2 = 0,6$ (екі қабатты ағаштан жасалған тоқыма);

$\tau_3 = 0,8$ (темірбетонды аркалар);

$\tau_4 = 1$ (жиналатын жалюзилер және шторлар);

$$\tau_0 = 0,5 \cdot 0,6 \cdot 0,8 \cdot 1 = 0,24;$$

η_0 -терезенің жарықтану мнездемесі.

Анықтаймыз η_0 :

$$l = B - 1,$$

$$l = 6 - 1 = 5 \text{ м},$$

$$\frac{L}{l} = \frac{L}{B-1} = \frac{9}{5} = 1,8,$$

$$h_{\text{расч}} = h_{\text{но}} + h_o - h_{\text{рп}}, \quad (4.13)$$

$$h_{\text{расч}} = 0,8 + 2 - 0,72 = 2,08,$$

$$\frac{B}{h_{\text{расч}}} = \frac{6}{2,08} = 2,88,$$

$\eta_0 = 10$ кесте алынған.

Бөлмедегі орташа шағылу коэффициент $\rho_{\text{СР}} = 0,5$, тек бір жағындағы жарықты қарастырамыз:

$$\frac{l_{\text{рт}}}{B} = \frac{0,5}{6} = 0,08$$

онда $r_1 = 1,05$,

$k_{\text{зд}}$ – іске асыру кезіндегі жарық көздерінің шаңдануы мен жарамсыз болуын ескеретін қор коэффициенті $k_{\text{зд}} = 1$.

Терезелердің толық ауданын есептейміз:

$$S_0 = \frac{36 \cdot 1,2 \cdot 10 \cdot 1 \cdot 0,63}{100 \cdot 0,24 \cdot 1,05} = 10,8 \text{ м}^2$$

Жарық өтуінің ауданы тең $S_{\text{сн}} = 10,8 \text{ м}^2$.

5 Техника – экономикалық негізделуі

5.1 Жұмысты сипаттау және қажеттілігін негіздеу

Берілген бітіру жұмысының тақырыбы – АҚ «Қазақтелеком» компаниясының қызметкерлерінің деректер қорын құрып, шифрлану арқылы ақпарат қауіпсіздігін қамтамасыз ету.

Берілген өнімді әзірлеудің мақсаты болып деректерді енгізу, сақтау, өңдеу және тасымалдау үшін қолданылатын ақпарат пен қорлардың тұтастығын, қол жеткізулік оңтайлығын және жасырындылығын қолдау болып табылады. Ақпараттың сыртқа кетуінің, оны ұрлаудың, жоғалтудың, рұқсатсыз жоюдың, өзгертудің, маңызына тимей түрлендірудің, рұқсатсыз көшірмесін жасаудың, бұғаттаудың алдын алу үшін бағдарламалық қамтама жасау қазіргі заманда өте маңызды. Осы бағдарламалық өнім көмегімен құнды ақпараттардың қауіпсіздігін қамтамасыз етеміз.

Берілген бөлімде осы жобаның жүзеге асырылуының экономикалық құрамының қарастырылуы келтіріледі. Ол уақыттық, еңбектік және қаржылай шығындарды көрсетеді. Бұл бағдарлама :

- 1) жаңа қызметкерлер туралы ақпаратты қағаз жүзінде емес, бағдарламаға оңай енгізіледі.
- 2) пайдаланушыда қызметкерлер деректер қоры архив сақталады
- 3) барлық жеке ақпарат бірнеше кілтті және сертификатты үздік шифрлану - TDE (AES) арқылы құпияланады.

5.2 Нарықты саралау және орнын анықтау

Бағдарламалық қамтаманың басты мақсаты мекемеде орнатылған деректер қорының административтік және қауіпсіздік жағынан істеу барысын басқару.

Біздің өнімнің экономикалық көрсеткіштерін нарықта үш категорияға бөлуге болады:

- 1) өнім (Product) – бұл шифрланған деректер қорын қолданатын бағдарламалық қамтама.
- 2) құн (Price)– бұл бағдарламалық қамтаманы пайдаланатын мекемеге қойылған ақша сомасы.
- 3) дистрибуция - бағдарламалық қамтама жеткізілетін ұсыныс берген мекеме.
- 4) қозғалу, даму (Promotion) — мекеменің алдымен тәжірибе жүзінде бірнеше елді-мекенге бағдарламалық қамтаманы қоюы, кейіннен өзге мекендерге ақпарат таратуы мен сатып алуға үгіттейтін мүмкіндігі зор іс-шаралар.

5.3 Бағдарламаның SWOT-анализі

Кесте 5.1 – Бағдарламаның SWOT-анализі кестесі

Артықшылықтары	Кемшіліктері
Бұл қызмет түрі елдегі телекоммуникация жүйелерінің әлі еш біреуінде қолданылмауы; АҚШ үкіметі қолдаған әдемдегі ең үздік құпиялау әдісінің қолданылуы;	Қымбат баға
Мүмкіншілігі	Қауіпі
Заманауи талаптарға сай барлық мекемелерінде қолданылу мүмкіндігі	XSL-шабуылдар

5.4 Маркетинг-микс элементі

Кесте 5.2 – Бағдарламаның маркетинг-микс кестесі

Маркетинг-микс элементтері	Қажетті істер мен талдаулар
Product	
Мекеме қызметкерлерінің шифрланған деректер қоры	Жоспар бойынша бірнеше компоненттер пайдаланылады: 1) Техникалық өнім; 2) Бағдарлама интерфейсі C# тілдерінде орындалмақ
Price	
Базалық бағасы	Техникалық құралдың түріне қарай бағдарламалық қамтаманың бағасы анықталады.
Promotion	
Жарнама	Жарнама жоқ, сатылам тек тапсырыс бойынша
Place	
Фирма 1, фирма 2, фирма 3	Қызметкерлер саны көп, ақпарат құпиялығының мәні зор барлық мекемелер

5.5 Есептеу бөлімі

5.5.1 Бағдарламалық қамтаманың өңделуінің еңбек өнімділігінің есептелуі

Еңбекке кеткен шығынның базалық көрсеткіштері мына формула бойынша есептеледі:

$$Q=q*c \quad (5.1)$$

Мұндағы q- бағдарламалық өнімдегі (бастапқы команда) операторлар саны, менің жұмысымда операторлар саны 2500-ге тең, c-бағдарлама

күрделілігінің коэффициенті, ол менде 1.26 ге тең [16]. Осы әдіспен базалық көрсеткішті табамыз:

$$Q=2500*1.26=3150$$

Ары қарай бағдарламалық өнімді әзірлеуге кететін уақытты есептеу керек. 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 сипаттамасының уақыты) келесі формуламен анықталады:

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

(5.2)

$$T_O=3150*1,5/50*0,8=118,125 \text{ адам/сағ}$$

мұндағы B – мақсат есебі өзгерісінің коэффициенті, B коэффициенті мақсат күрделілігіне және өзгеріс санына тәуелді – 1,2-ден 1,5-ке дейін ;

K – бағдарлама жасаушы біліктілігін ескеретін коэффициент, менің берілген салада жұмыс істеу өтілім 2 жылға дейінгі уақытты құрағандықтан, K коэффициенті мәні – 0,8-ге тең болады.

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

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

$$T_A = 3150/(50*0,8) = 78,75 \text{ адам/сағ}$$

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:

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

$$T_H=3150*1,5/(50*0,8) = 118,125 \text{ адам/сағ}$$

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

$$T_{II} = Q / 50 = 63 \text{ адам/сағ} \quad (5.5)$$

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

$$T_{OT} = Q \times 4,2/50 \times K = 3150*4,2/50*0,8 = 330,75 \text{ адам/сағ} \quad (5.6)$$

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:

$$T = T_{\text{ПО}} + T_{\text{ТО}} + T_{\text{А}} + T_{\text{БС}} + T_{\text{Н}} + T_{\text{П}} + T_{\text{ОТ}} + T_{\text{Д}}. \quad (5.7)$$
$$T = 24 + 118,125 + 78,75 + 78,75 + 118,125 + 63 + 330,75 + 24 = 835,5 \text{ адам/сағ.}$$

5.5.2 Еңбекақыға жұмсалатын шығындар

Еңбек құны екі құрамнан құралады: негізгі еңбек құны және қосымша еңбек құны.

Негізгі еңбек құны:

$$Z_{\text{осн}} = t_{\Sigma} * TC / (t_{\text{cp}} * 8), \text{ теңге} \quad (5.8)$$

мұндағы t_{Σ} – (5.7) формуласы бойынша есептелетін жалпы еңбек шығыны;

t_{cp} – айдағы орташа күн саны, 21 күнге тең, жұмыс күніндегі 8 сағатқа көбейтіледі.

Тарифтік мөлшерлеме минималды өлшемдегі еңбек ақы 01.01.2014ж бастап ҚР-да 19966 теңгені құрады.

TC – тарифтік қойылым = 19966 -ке тең болды.

t_{Σ} – (5.7) формуласы бойынша 473 –ге тең.

Осылайша негізгі еңбек ақы:

$$Z_{\text{осн}} = 835,5 * 19966 / (21 * 8) = 99295,196 \text{ теңге.}$$

Қосымша еңбек ақы негізгі еңбек ақының 20%-ын құрайды. Мына формула бойынша анықталады:

$$Z_{\text{доп}} = 0,2 * Z_{\text{осн}} = 19859,039 \text{ теңге.} \quad (5.9)$$

Жалпы еңбек ақысы (еңбекті төлеу фонды) негізгі және қосымша еңбек ақының қосындысы ретінде анықталады (5.10):

$$\text{ФОТ} = Z_{\text{доп}} + Z_{\text{осн}} = 99295,196 + 19859,039 = 119154,235 \text{ теңге.} \quad (5.10)$$

5.5.3 Қосымша шығындар статьясы

Бағдарламалық қамтамасыз етудегі қосымша шығындар: материалдар мен құрамаға кеткен шығындар (құрылғының өз құны, яғни компьютерлер есепке алынбайды), социалды сақтандыруға аударым, жүккүжатка кеткен шығындар, амортизациялық аударым, техникалық қондырғылар қызметіне кеткен шығын, компьютермен жұмыс істеу кезінде кеткен электроэнергиялық шығындар).

Қондырғының құны бағдарламалық қамтама өңделуінің өзіндік құнына кірмегенімен, қосымша шығындар статьясындағы есептеулерде қолданылады. ЭЕМ - на бағдарлама жазу кезінде құрылғы ретінде дербес компьютер

қолданылады. Құны $C_{обор} = 150000$ теңге (дербес компьютердің қазіргі уақыттағы нарықтағы теңгемен берілген құны алынды)

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% [11] құрайды және мына формулалар мен есептеледі:

$$\begin{aligned} C_{МжК} &= 0,015 \times C_{обор} = 0,015 * 150000 = 2250 \text{ теңге,} \\ C_{ТО} &= 0,025 \times C_{обор} = 0,025 * 150000 = 3750 \text{ теңге.} \end{aligned} \quad (5.11)$$

Амортизациялық аударым өнімнің физикалық және моральдық жарамсыздығына байланысты сол өнімді толық қалпына келтіру үшін жиналатын ақша қоры болып есептеледі. Амортизациялық аударым амортизация нормасы бойынша орнатылады, (5.12)-формуласы бойынша есептеледі:

$$A = \frac{C_{обор} \times H_A \times N}{100 \times 12 \times t} \quad (5.12)$$

мұндағы $C_{обор}$ – компьютер құны;

H_A – амортизация нормасы;

N – жұмыс орындалуына кеткен күннің саны;

t – дербес компьютерді қолдануға кеткен жалпы уақыт.

$$N = T/24 = 835,5/24 = 34,812 \text{ күн}$$

бұл жерде T еңбек шығынына тең, оның мәні – 473 адам/сағ тең болған.

Дербес компьютерде жалпы жұмыс істеу уақыты мына формуламен есептеледі:

$$t = T_A + T_{БС} + T_H + T_{П} + T_{ОТ} = 78,75 + 78,75 + 118,125 + 63 + 330,75 = 669,375 \text{ адам/ сағ}$$

Амортизация нормасы (5.13)-формуласы бойынша анықталады:

$$H_A = \frac{C_{обор} - C_{ликв}}{T_{норм} \cdot C_{обор}} \cdot 100\%, \quad (5.13)$$

мұндағы $C_{ликв}$ – таратылым құны, жабдықтың құнынан 5% құрайды:

$$C_{ликв} = 0,05 * C_{обор} = 0,05 * 150000 = 7500 \text{ теңге.} \quad (5.14)$$

$T_{норм}$ – нормативті қызмет ету өтілі (дербескомпьютер үшін ҚР да $T_{норм} = 4$ жыл). Осыдан:

$$H_A = \frac{C_{обор} - C_{ликв}}{T_{норм} \cdot C_{обор}} \cdot 100\% = \frac{150000 - 7500}{4 \cdot 150000} \cdot 100\% = 23,75\%,$$

$$A = \frac{C_{обор} \times H_A \times N}{100 \times 12 \times t} = 150000 \cdot 23,75 \cdot 669,379 / 100 \cdot 34,812 \cdot 12 = 57084,304 \text{ теңге.}$$

Бағдарламалық қамтаманы жазу кезіндегі электроэнергияның шығынын қоса есептеу керек. Электроэнергияның құны мына формула бойынша есептеледі (5.15):

$$C_{ЭЭ} = M \cdot k_3 \cdot F_{эф} \cdot C_{кВт-ч} \quad (5.15)$$

мұндағы M – ЭВМ қуаты (450 Вт);

k_3 – жүктеу коэффициенті (0.8);

$C_{кВт.ч}$ – 1 кВт-сағ электроэнергияның құны, оның мәні ҚР – да 14,36 теңгеге тең;

$F_{эф}$ – жұмыс уақытының эффективті фонды, мына формула бойынша есептеледі (5.16):

$$F_{эф} = D_{ном} \cdot d \cdot (1 - f / 100) = 258 \cdot 8 \cdot (1 - 2,2 / 100) = 2019 \text{ сағ.} \quad (5.16)$$

мұндағы $D_{ном} = 258$ – бір жылдағы жұмыс күнінің номиналды саны;

$d = 8$ – жұмыс күнінің ұзақтығы [сағ];

$f = 2.2\%$ – ЭВМ – ды жөндеуге кеткен жоспарланған уақыт.

Берілген коэффициенттер мен параметрлер бойынша электроэнергияның құны:

$$C_{ЭЭ} = 0,45 \cdot 0,8 \cdot 2019 \cdot 14,36 = 10437 \text{ теңге.}$$

Дегенмен, алынған амортизациялық аударым мен электроэнергияға шығын – жылдық шығынның мәні, оны жалпы жылдық эксплуатациондық шығынмен анықталатын уақыт коэффициентімен дұрыстау керек (5.17):

$$\mathcal{E}_3 = t_{\Sigma'} \cdot C_3 / F_{эф}, \text{ теңге} \quad (5.17)$$

мұндағы $C_3 = C_{ЭЭ} + C_{ТО} + A_{жыл}$ – ЭЕМ-ның жалпы жылдық эксплуатационды шығын құны:

$$C_3 = C_{ЭЭ} + C_{ТО} + A_{жыл} = 10437 + 3750 + 57084,307 = 71271,307 \text{ теңге.}$$

мұндағы $F_{эф}$ – жұмыс уақытының эффективті фонды, (5.16) формуласы бойынша анықталды, $t_{\Sigma'}$ – ЭВМ-нің (5.7) формуласы бойынша анықталған есепті шешу үшін жалпы қолданылған уақыты, бірақ бұл жерде тек компьютердегі жұмыс уақытын ескере отырып анықталған:

$$t_{\Sigma} = t_{\text{пр}} + t_{\text{отл}} + t_{\text{д}} = 78,75 + 330,75 + 24 = 433,5 \text{ сағ.}$$

Сәйкесінше, ЭЕМ-ның жалпы жылдық эксплуатационды шығын құны:

$$\Theta_3 = 433,5 * 71571,307 / 2019 = 15302,68 \text{ теңге}$$

Әлеуметтік салыққа кететін аударымдар бүкіл жалақының 11% құрайды [11%, ҚР ның Салық Кодексі], (5.19) формуласы бойынша анықталады:

$$CC = (\text{ФОТ} - 0,1 * \text{ФОТ}) * 0,11 = (119154,235 - 0,1 * 119154,235) * 0,11 = 11796,269 \text{ теңге} \quad (5.19)$$

Жүктеме шығындар қызмет көрсету мен басқаруға қатысты, құрылғының эксплуатациясы мен басқа да өндіріс процесін қамтамасыз ететін қосымша шығындар болып табылады, еңбек жалақысы фондының 50% құрайды, (5.20) формуласы бойынша анықталады:

$$C_{\text{накл}} = 0,5 * \text{ФОТ} = 0,5 * 119154,235 = 59577,117 \text{ теңге} \quad (5.20)$$

5.5.4 Өзіндік құн нәтижесінің кестесі мен диаграммасы

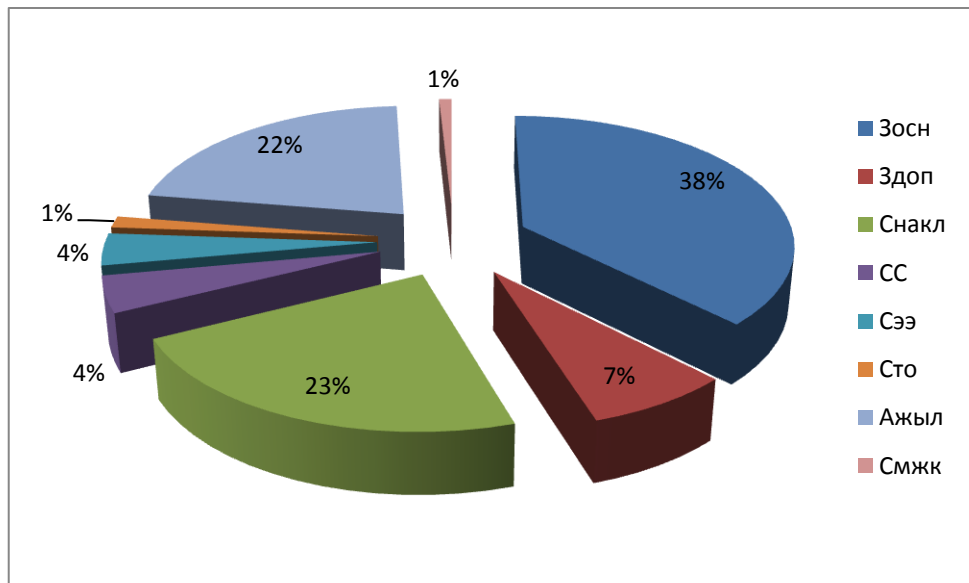
Бағдарламалық қамтаманы өңдеуге кеткен жалпы шығын еңбек жалақысы фондының, эксплуатационды шығынның, социалды сақтандыру, жүктеме шығындар мен материал және құрамалардың қосындысына тең болады.

Бағдарламалық қамтаманы өңдеудің қорытынды құны 5.3 кестеде көрсетілген.

К е с т е 5.3 – Өзіндік құн нәтижесінің кестесі

Шығын статьялары		Сомма, теңге	Жалпы соммадан, %
ФЗП	$Z_{\text{осн}}$	99295,196	38
	$Z_{\text{доп}}$	19859,039	7,5
Жүктеме шығындар, $C_{\text{накл}}$		59577,117	22,5
Әлеуметтік сақтандыру, CC		11796,269	4,4
Эксплуатационды Шығындар	$C_{\text{ээ}}^*$	10437	3,9
	$C_{\text{то}}$	3750	1,4
	$A_{\text{жыл}}^*$	57084,304	21,6
Материалдар мен комплектілер, $C_{\text{мик}}$		2250	1
Барлығы:		264048,928	100

Өзіндік құнының диаграммасы:



Сурет-5.1 – Өнімнің өзіндік құнының диаграммасы

5.6 Бағдарламаның бағасына есептеу

Өнімнің өзіндік құны 264048,928 теңге (жоғарыда кесте 5.3) болды. Өнімнің минималды бағасы келесі формуламен есептеледі:

$$Ц_{\min} = (C_{\text{пс}} + C_{\text{мд}} + C_{\text{з}})(1 + r/100) = (264048,928 + 200 + 150) * 1,21 = 319922,703 \text{ тг.} \quad (5.21)$$

мұндағы $C_{\text{мд}}$ - магнитті диск бағасы, қазіргі нарықтық бағасын қарастырып, 200 теңге деп алынды;

$C_{\text{з}}$ – жазба бағасы;

r - Өзіндік құнға қатынасымен алынған тиімділіктің жоспарланған пайызы=21%;

НДС ті есептегендегі минималды баға келесі формуламен есептеледі (ҚР да НДС 12%):

$$Ц_{\min_{\text{НДС}}} = Ц_{\min} + N_{\text{ндс}} \quad Ц_{\min} = 319922,703 + 0,12 * 319922,703 = 358313,43 \text{ тг.} \quad (5.22)$$

Сонда бағдарламаның құны 360000 теңге.

5.6.1 Бағдарламалық өнімді иемденудегі кәсіпорындардың бірмезгілдегі шығындарын есептеу

Кәсіпорындардың бірмезгілдегі шығындары келесі бөлімдерден тұрады:

– жүйе құны;

- көлік шығындары;
 - оқуға кететін төлемдер;
- Жүйе құны 360000 теңге (5.22 формула).
Көлік шығындары жүйе құнының 20% құрайды:

$$C_{тр} = 360000 * 0,2 = 72000 \text{ теңге}$$

Жұмысшыларды жүйемен жұмыс істеуге оқыту шамамен екі сағатта консалтингті фирманың маманы көмегімен сағатына 2000-3000 теңгедей болып табылады [11]. Соған сәйкес оқу бағасы:

$$C_{обуч} = 2000 * 8 = 16000 \text{ теңге}$$

Кәсіпорынның бірізгідегі шығындарын есептеулерінің нәтижесі 5.4-кестеде көрсетілген.

К е с т е 5.4 – Ақпаратты жүйені енгізуге арналған бірізгідегі шығындардың есептелу нәтижесі

Шығындар түрлері	Сомма, теңге
Жүйе құны	360000
Көлік шығындары	72000
Жұмысшыларды оқыту бағасы	16000
Қорытынды:	448000

5.6.2 Қолдану саласындағы жылдық шығындарды есептеу

Негізгі шығындар түрлері:

- жұмысшы еңбегіне төленетін ақша қоры;
- әлеуметтік салық;
- басқада шығындар.

Дұрыстау ПК операторымен 2500 теңге/сағ сағаттық мөлшерде сегіз сағат көлемінде жүргізіледі.

$$C_{опер} = 2500 * 8 = 20000 \text{ теңге.}$$

Әлеуметтік салыққа жұмсалатын аударымдар (ҚР да әлеуметтік салық мөлшері 11% жұмысшы еңбегіне төленетін ақша қорынан, ҚР Салық Кодексі)

$$CC = (20000 - 20000 * 0,1) * 0,11 = 1980 \text{ теңге.}$$

К е с т е 5.5 – Кәсіпорынның жылдық бірізгідегі шығындарын есептеу нәтижесі

Шығындар түрлері	Сомма, теңге
Еңбекке төленетін қор	20000
Әлеуметтік салыққа аударымдар	1980
Қорытынды:	21980

5.6.3 Ақпараттық өнімді енгізуден алынған үнемдеу мөлшері мен табысты есептеу

Ақпараттық өнімді құрастыратын фирма үшін табыс көзі ретінде тапсырыс берушіге бағдарламалық өнімді сату болып табылады. Фирманың шығындары – ол құрастыруға кететін шығындар және жүйенің көбейтілуіне кететін шығындар. Қаржыландыру көзі ретінде құрастырушы-фирманың өзінің ақша қоры алынады.

Тапсырыс беруші-кәсіпорын үшін үнемдеу көзі есептеу уақытын біршама қысқартатын, қол еңбегін алмастыратын машина болып табылады. Кәсіпорын шығындары ақпараттық жүйені иемденудегі біртегіздік шығындардан, ақпараттық жүйені тасымалдау және енгізу шығындарынан, және де жүйеге талдау жүргізу мен бақылауға кететін шығындардан тұрады.

Жүйені енгізу қолмен есептеу шараларын автоматтандырылған шараға ауыстыру нәтижесінде өндірістің қорын біраз үнемдеуге алып келеді деп жоспарланады.

Үнемдеу мөлшерін есептейік. Ол үшін өткізу қабілетін есептеуге кететін кәсіпорынның шығындарын қолмен есептеу керек.

Қол еңбегімен келесі шығындар түрлері есептеледі:

- жұмысшы еңбегіне төленетін ақша қоры;
- әлеуметтік салық;
- басқа да шығындар.

Қолмен есептеу барысында еңбек өнімділігін талдау үшін 1500 теңге/сағ (ұйымның қойып отырған талабы) мөлшерінде жұмыс істейтін мамандандырылған инженер қажет. Өткізу қабілетін есептеу үшін инженер 21 күн жұмсайды. Қолмен есептеу барысында еңбектің жылдық төлем қоры:

$$EA=21*8*1500=252000 \text{ теңге.}$$

Әлеуметтік салыққа жұмсалатын аударымдар (ҚР да әлеуметтік салық мөлшері 11% жұмысшы еңбегіне төленетін ақша қорынан, ҚР Салық Кодексі):

$$CC=(252000-252000*0,1)*0,11=24948 \text{ теңге.}$$

К е с т е 5.6 – Ақпараттық жүйені қолданбай еңбек өнімділігін талдауға кететін шығындарды есептеу

Шығындар түрлері	Сомма, теңге
Еңбекке төленетін қор	252000
Әлеуметтік салыққа аударымдар	24948
Қорытынды:	276948

Ақпараттық жүйені қолданғандағы жылдық шығындар жоғарыда есептелді және 21980 теңге болды.

Осыған сәйкес, шығындардың шартты үнемделуі мынаған тең болады:

$$C_{\text{экон}} = 276948 - 21980 = 254968 \text{ теңге.}$$

5.7 Ақшалай құралдардың қозғалысы

К е с т е 5.7 – Ақшалай тәсілдердің қозғалысы, теңге

Аты	Жыл				Барлығы
	2014	2015	2016	2017	
Бірмезгілдік шығындар	448000				448000
Операциялық кәсіпкерліктен пайда		254968	254968	254968	764904
Дисконттау коэффициенті (21% мөлшерінде)	1	0,826	0,683	0,564	
Таза дисконтталған табыс (ТДТ)	-448000	210603,564	174143,144	143801,952	
ТДТ өспелі нәтижесімен	-448000	-237396,436	-63253,292	80548,656	

Таза дисконтталған табыс (ТДТ) = Таза табыс* Дисконттау коэффициенті (5.23)

$$\text{ТДТ}_1 = 254968 * 0.826 = 210603,564 \text{ теңге}$$

$$\text{ТДТ}_2 = 254968 * 0,683 = 174143,144 \text{ теңге}$$

$$\text{ТДТ}_3 = 254968 * 0,564 = 143801,952 \text{ теңге}$$

5.8 Экономикалық тиімділікті есептеу

5.8.1 Таза ағымдағы құндылықты есептеу (Net present value, NPV)

Шығындары бірмезгілде тек жобаның басында күрделі салымдар (C_0) ретінде іске асатын жобалар үшін NPV (5.24) формуламен есептеледі:

$$NPV = \sum_{i=1}^n \frac{B_i}{(1+r)^i} - C_0. \quad (5.24)$$

мұндағы V_i – i -ші жылдағы жобадан алынатын пайда;
 r – дисконттеу мөлшері

$$NPV(21\%) = \left(\frac{254968}{(1+0.21)^1} + \frac{254968}{(1+0.21)^2} + \frac{254968}{(1+0.21)^3} \right) - 448000 = 80548,656 \text{тг} > 0$$

сондықтан ұсынылып отырған жоба табысты. Сол себептен жобаны қажет және жобаны талдау мен оған баға беруді жалғастыру керек.

5.8.2 Пайда индексі есептеу (Profitability index, PI)

PI жобаның салыстырмалы пайдасын көрсетеді. Бұл көрсеткіш келесі (5.25) формуламен есептеледі:

$$PI = \frac{NPV}{C_0} \quad (5.25)$$

$$PI = (528548,656/448000) = 1,17\%$$

5.8.3 Табыстың ішкі нормасын есептеу (Internal rate of return, IRR)

Табыстың ішкі нормасы (ТИН немесе IRR) r пайыздық мөлшерінің мәні болып табылады, мұндағы $NPV = 0$. Бұл нүктеде r^* қосылған шығындардың дисконтталған ағыны қосылған дисконтталған табыс ағындарына тең.

$$IRR = r_1 + \frac{f(r_1)}{f(r_1) - f(r_2)} * (r_2 - r_1) \quad (5.26)$$

$$r_b = 33\%: \quad PV_1 = 254968 / (1 + 0,33) = 191735,936 \text{ теңге,}$$

$$PV_2 = 254968 / (1 + 0,33)^2 = 144056,92 \text{ теңге,}$$

$$PV_3 = 254968 / (1 + 0,33)^3 = 108361,4 \text{ теңге.}$$

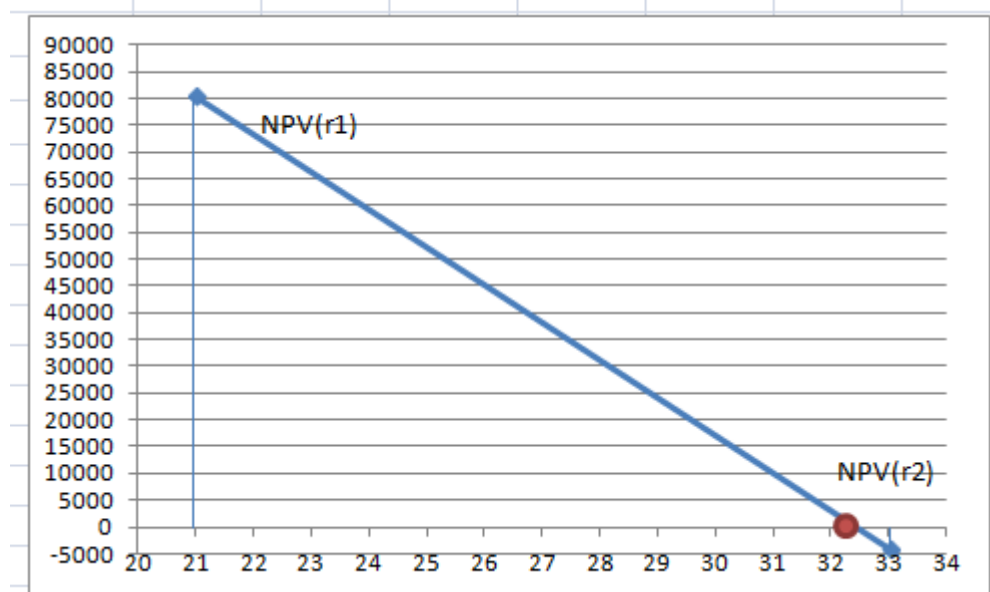
$$NPV(33\%) = (191735,936 + 144056,92 + 108361,4) - 448000 = -3845,744 \text{ теңге}$$

$$NPV(21\%) = 80548,656 \text{ теңге,}$$

$$NPV(33\%) = -3845,744 \text{ теңге,}$$

$$IRR = r1 + \frac{NPV(r1)}{NPV(r1) - NPV(r2)} \times (r2 - r1) = 21 + \frac{80548,656}{80548,656 - (-3845,744)} * (33 - 21) = 32\%$$

Табыстың ішкі нормасы сурет 5.2-де көрсетілген:



Сурет-5.2 - Іріктеу әдісі арқылы IRR табу

5.8.4 Өтімділік периодын есептеу (Payback period, PBP)

Өтімділік периоды мына формуламен есептеледі:

$$T_{ок} = t + \frac{C_0 - \sum_{i=1}^n B_i}{B_{i+1}} \quad (5.27)$$

мұндағы C_0 – жоба басындағы салым;

B_i – i -ші жылдағы жобадан алынатын пайда.

Сонда біздің өтімділік периодымыз:

$$T_{ок} = 2 + \frac{448000 - 384746,709}{143801,952} = 2,44 = 2 \text{ жыл } 5 \text{ ай.}$$

Бұл жобаның өтімділік периоды, 2 жыл 5 айға тең, яғни жобамыз осы уақытта өз-өзін ақтайды.

Қорытынды

Бұл дипломдық жобада барлық қойылған мақсаттар іске асырылды. Қызметкерлер дерекқоры құрылды. Дерекқорды құру барысында барлық нюанстар ескерілді. Беріліп отырған бағдарламалық өнім барлық үрдісті автоматтандыруға мүмкіндік береді. Ақпараттық ортадағы өнімнің артықшылығы қауіпсіздік мәселесі болып табылады.

Дерекқор SQL Server 2008 нұсқасы дерекқоры басқару жүйесінде жасалды. Жана 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 есептеулер нәтижесінде кондиционердің жыл бойына жұмысын қамтамасыз ету үшін таңдауымызды DELONGHI фирмасының кондиционеріне тоқтаттық, себебі бұл фирманың кондиционерлері салқын жыл мезгілінде «жылу насосы» тәртібінде тиімді жұмыс істейді.

Техника-экономикалық негізделуі бөлімінде жасалған есептеулер нәтижесінде бағдарламалық өнімді жүзеге асыруға кеткен шығын 480000 теңгені құрады. Бұл бағдарламалық өнім кең қолданысқа ие ДҚБЖ MS SQL Server-де жазылған. Ол әсіресе ірі мекемелерде кең таралған, сондықтан жүзеге асырылуының жоғары бағасына қарамастан, берілген өнім экономикалық тиімді болып саналады.

Әдебиеттер тізімі

- 1 Бондарь А.Г. MS SQL Server 2012. Создание баз данных и разработка программ. –БХВ-Петербург, 2013. – 608с.
- 2 Д. Петкович Microsoft SQL Server 2012. Руководство для начинающих. –БХВ-Петербург, 2013. – 816с.
- 3 Джо Майо: Самоучитель Microsoft Visual Studio 2010. – БХВ, 2011 г.– 464с.
- 4 Сатимова Е.Г. Проектирование баз данных. Методические указания к выполнению лабораторных работ для студентов специальностей 050704 – Вычислительная техника и программное обеспечение и 050703 – Информационные системы. – Алматы: АИЭС, 2009
- 5 Марк Шпеник, Оррин Следж. Microsoft SQL Server 2000 DBA Survival Guide. – М.: Вильямс, 2001. – 236 с.
- 6 Мамаев Е. MS SQL Server. Проектирование и реализация баз данных. Сертификационный экзамен. – СПб.: BHV, 2004. – 416с.
- 7 Плю Р., Стефенс Р., Райан К. Освой самостоятельно SQL за 24 часа. – М.: Вильямс, 2000.
- 8 Глушаков С.В., Ломотько Д.В. Базы данных: учеб. курс. – М.: ООО «Издательство АСТ», 2000. – 504с.
- 9 Айтхожаева Е.Ж. Стандартный язык баз данных SQL: учеб. пособие. – Алматы: АИЭС, 2005. - 48 с.
- 10 Дунаев В.В. Базы данных. Язык SQL. – СПб.: БХВ – Петербург, 2006.
- 11 З.Д. Еркешева, Г.Ш. Боканова, 5B070400 – Есептеу техникасы және бағдарламамен қамтамасыз ету мамандығының студенттері үшін диплом жұмысының экономика бөлімін орындауға әдістемелік нұсқаулар. – Алматы: АЭЖБУ, 2014 – 27 б.
- 12 Базылов К.Б., Алибаева С.А. Методические указания по выполнению экономического раздела выпускной работы бакалавров для студентов всех форм обучения – Алматы: АИЭС, - 2009
- 13 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.
- 14 Хакимжанов Т.Е. Расчет аспирационных систем. Дипломное проектирование. Для студентов всех форм обучения всех специальностей. – Алматы: АИЭС, 2002. – 30 с.

Қосымша А

Бағдарламаның MS SQL Server-дегі листингі

```
create table FIO
(id_sotrudnik int not null primary key,
familiya varchar(25),
imya varchar(25),
otchestvo varchar(15),
pol varchar(2),
data date,
id_region int,
id_otdel int,
id_dolzh int );

go

create table region
(id_region int not null primary key,
region_name varchar (30),
kod varchar(15),
);

go

alter table fio
add constraint pk_fio_region foreign key (id_region) references region (id_region) ;

insert into region (id_region,region_name,kod)
values ('1','Тараз','7262'),
      ('2','с.Сарыкемер','72637'),
      ('3','с.Б.Момышулы','72635'),
      ('4','с.Кордай','72636'),
      ('5','г.Каратау','72644'),
      ('6','с.Кулан','72631'),
      ('7','с.Толе би','72638'),
      ('8','г.Жанатас','72634'),
      ('9','с.Мерке','72632'),
      ('10','с.Аса','72633'),
      ('11','с.Акколь','72641'),
      ('12','с.Мойынқум','72642'),
      ('13','г.Шу','72643');

insert into FIO (id_sotrudnik,familiya,imya,otchestvo,pol,data,id_region,id_otdel,id_dolzh)
values ('1','Абдраимов', 'Наркес','Абуталипович','м','1961-04-12','1','1','1'),
      ('2','Абдешова', 'Карлыгаш','Сеитазимовна','ж','1982-11-15','1','1','2'),
      ('3','Нурмаханова', 'Гульзира',' ','ж','1987-08-05','1','2','4'),
      ('4','Дворецкая', 'Татьяна',' ','ж','1975-04-28','13','3','6'),
      ('5','Алиев', 'Дмитрий',' ','м','1987-08-05','13','4','6'),
      ('6','Абдельдинова', 'Гульнара','Толековна','ж','1969-09-19','9','5','5'),
      ('7','Садыков', 'Талгат',' ','м','1987-08-05','9','6','5');
```

А қосымшасының жалғасы

```
create table otdel
(id_otdel int not null primary key,
otdel_name varchar(25));

insert into otdel (id_otdel,otdel_name)
values ('1','DIS'),
       ('2','Абонентский модуль'),
       ('3','Технический учет'),
       ('4','Монитор(ЦБР)'),
       ('5','ГЦУ'),
       ('6','ЦП'),
       ('7','ЦРС');

alter table fio
add constraint pk_fio_otdel foreign key (id_otdel) references otdel (id_otdel );

create table dolzhnost
(id_dolzh int not null primary key,
dolzh_name varchar(25));

insert into dolzhnost (id_dolzh,dolzh_name)
values ('1','Директор'),
       ('2','Заместитель директора'),
       ('3','Главный специалист'),
       ('4','Специалист Абон.отдела'),
       ('5','Ведущий Специалист АО'),
       ('6','Администратор'),
       ('7','Siebel Administrator '),
       ('8','ERM Guest'),
       ('9','Optimizer group'),
       ('10','Test Position');

alter table fio
add constraint pk_fio_dolzhnost foreign key (id_dolzh) references dolzhnost (id_dolzh ) ;

create table arhiv (
id_sotrudnik int,
familiya varchar(25),
imya varchar(25),
otchestvo varchar(25),
pol varchar(2) ,
data date,
id_region int,
id_otdel int,
id_dolzh int,
date_dog date,
date_ist date,
zarplata int)
```

А қосымшаның жалғасы

```
INSERT INTO arhiv(id_sotrudnik , familiya, imya,otchestvo,pol,data,id_region,id_otdel
id_dolz ,date_dog ,date_ist ,zarplata )
SELECT TOP 1 id_sotrudnik ,familiya,imya,otchestvo,pol ,data, id_region ,id_otdel ,
id_dolz , date_dog , date_ist ,zarplata FROM FIO;
```

```
alter table arhiv
add constraint fk_dolzhnost_arhiv foreign key (id_dolz) references dolzhnost (id_dolz)
```

```
alter table arhiv
add constraint fk_region_arhiv foreign key (id_region) references region (id_region)
```

```
alter table arhiv
add constraint fk_otdel_arhiv foreign key (id_otdel) references otdel (id_otdel)
```

```
create table kazkom
(nomer int not null primary key,
id_sotrudnik int,
familiya_en varchar(25),
srok date);
```

```
alter table kazkom
add constraint fK_FIO_kazkom foreign key (id_sotrudnik)references FIO (id_sotrudnik)
```

```
alter table kazkom
add constraint fK_arhiv_kazkom foreign key (id_sotrudnik)references arhiv (id_sotrudnik)
```

```
create master key encryption by
password='AidarkulovaMeruert';
go
```

```
create certificate proba01
with subject='proba';
go
```

```
create symmetric key NOMERKARTY
with algorithm=triple_des
encryption by certificate proba01;
go
```

```
alter table sales
add encryptedccnumber varbinary(128);
go
```

```
open symmetric key NOMERKARTY
decryption by certificate proba01;
```

```
update nomer
set encryptedccnumber =
ENCRYPTBYKEY( key_guid(' NOMERKARTY'),ccnumber);
```

Қосымша А-ның соңы

```
close symmetric key NOMERKARTY;  
go
```

```
create certificate TDE_certificate with subject = 'TDE_Certificate';
```

```
use master
```

```
backup certificate TDE_certificate to file = 'C:\mika\diplomka\TDE\TDE_certificate.cert'  
with private key(  
file='C:\mika\diplomka\TDE\TDE_certificate',  
encryption by password = 'Aidarkulova18Meruert08Muratovna91');
```

```
use baza
```

```
go
```

```
create database encryption key  
with algorithm = aes_256  
encryption by server certificate tde_certificate;  
go
```

```
alter database baza
```

```
set encryption on;
```

```
go
```

```
select *  
from sys.dm_database_encryption_keys
```

```
use master
```

```
go
```

```
select * from sys.certificates where [name] = 'tde_certificate'
```

```
go
```

```
backup database [baza] to disk = n'c:\mika\diplomka\дәңгәліәй êîrëÿ áàçû\baza_encrypted.bak' with  
noformat, noinit,  
name = n'baza-full database backup', skip, norewind, nounload, stats = 10  
go
```

Қосымша Ә

Бағдарламаның MS Visual Studio-дегі листингі

Form 1

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

namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            textBox1.Clear();
            textBox2.Clear();
        }

        private void button3_Click(object sender, EventArgs e)
        {
            new Form3().ShowDialog();
        }

        private void button1_Click_1(object sender, EventArgs e)
        {
            if (textBox1.Text == "admin" && textBox2.Text == "admin")
                new Form2().ShowDialog();
            else
                MessageBox.Show("Неправильный логин или пароль. Введите заново!!!");
        }
    }
}
```

Form 2

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
```

Ә қосымшаның жалғасы

```
using System.Text;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace WindowsFormsApplication1
{
    public partial class Form2 : Form
    {
        public Form2()
        {
            InitializeComponent();

            var reg = (from i in Program.db.region select i.region_name);
            foreach (var r in reg)
            {
                regionComboBox.Items.Add(r);
            }

            var otd = (from i in Program.db.otdel select i.otdel_name);
            foreach (var o in otd)
            {
                otdelComboBox.Items.Add(o);
            }

            var dolzh = (from i in Program.db.dolzhnost select i.dolzh_name);
            foreach (var d in dolzh)
            {
                dolzhComboBox.Items.Add(d);
            }
        }

        private void textBox1_TextChanged(object sender, EventArgs e)
        {
            string constring = "Data Source=(local);Initial Catalog=Baza;Integrated Security=True";
            SqlConnection condatabase = new SqlConnection(constring);
            SqlCommand cmddatabase = new SqlCommand(
                "select f.familiya as Фамилия,f.imya as Имя,r.region_name as Регион, o.otdel_name as
                Отдел,d.dolzh_name as Должность, f.date_dog as Начало,f.date_ist as Окончание,f.zarplata as
                Зарплата from FIO f join region r on f.id_region=r.id_region join otdel o on o.id_otdel=f.id_otdel
                join dolzhnost d on f.id_dolzh=d.id_dolzh where( f.familiya like" + textBox1.Text + "%) ",
                condatabase);

            try
            {
                SqlDataAdapter sda = new SqlDataAdapter();
                sda.SelectCommand = cmddatabase;
                DataTable dbdataset = new DataTable();
                sda.Fill(dbdataset);
                BindingSource bsource = new BindingSource();
                bsource.DataSource = dbdataset;
            }
        }
    }
}
```

Ә қосымшаның жалғасы

```
dataGridView2.DataSource = bsource;
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}

string cn, a, a1;

public void connect()
{
    SqlConnection cn = Program.conn;
    a = a1;
    DataSet ds = new DataSet();
    SqlDataAdapter da = new SqlDataAdapter(a, cn);
    da.Fill(ds);
    dataGridView2.DataSource = ds.Tables[0];
}

private void Form2_Load(object sender, EventArgs e)
{
    // TODO: данная строка кода позволяет загрузить данные в таблицу
    "bazaDataSet1.obwee.obwee". При необходимости она может быть перемещена или удалена.
    this.obweeTableAdapter.Fill(this.bazaDataSet1.obwee.obwee);
    // TODO: данная строка кода позволяет загрузить данные в таблицу
    "bazaDataSet.otdel". При необходимости она может быть перемещена или удалена.
    this.otdelTableAdapter.Fill(this.bazaDataSet.otdel);
    // TODO: данная строка кода позволяет загрузить данные в таблицу
    "bazaDataSet.region". При необходимости она может быть перемещена или удалена.
    this.regionTableAdapter.Fill(this.bazaDataSet.region);
}

private void regionComboBox_SelectedIndexChanged(object sender, EventArgs e)
{
    // a1 = "SELECT * FROM region where region_name = " + regionComboBox.Text + """;
    a1 = "select r.region_name as Регион, f.familiya as Фамилия, f.imya as Имя, o.otdel_name
as Отдел, d.dolzh_name as Должность from FIO f join region r on f.id_region=r.id_region join
otdel o on o.id_otdel=f.id_otdel join dolzhnost d on f.id_dolzh=d.id_dolzh where region_name = "
+ regionComboBox.Text + """;
    connect();
}

private void otdelComboBox_SelectedIndexChanged(object sender, EventArgs e)
{
    a1 = "select o.otdel_name as Отдел, f.familiya as Фамилия, f.imya as Имя, r.region_name as
Регион, d.dolzh_name as Должность from FIO f join region r on f.id_region=r.id_region join otдел
o on o.id_otdel=f.id_otdel join dolzhnost d on f.id_dolzh=d.id_dolzh where otдел_name = "
+ otdelComboBox.Text + """;
}
```


Ә қосымшаның жалғасы

```
connect();
}

private void button1_Click(object sender, EventArgs e)
{
    new Form4 ().ShowDialog();
}

private void checkBox1_CheckedChanged(object sender, EventArgs e)
{
    string constring = "Data Source=(local);Initial Catalog=Baza;Integrated Security=True";
    SqlConnection condatabase = new SqlConnection(constring);
    SqlCommand cmddatabase = new SqlCommand(
        "select f.familiya as Фамилия ,f.imya as Имя,r.region_name as Регион, o.otdel_name as
Отдел, d.dolzh_name as Должность, f.date_dog as Начало,f.date_ist as Окончание,f.zarplata as
Зарплата from FIO f join region r on f.id_region=r.id_region join otdel o on o.id_otdel=f.id_otdel
join dolzhnost d on f.id_dolzh=d.id_dolzh ", condatabase);

    try
    {
        SqlDataAdapter sda = new SqlDataAdapter();
        sda.SelectCommand = cmddatabase;
        DataTable dbdataset = new DataTable();
        sda.Fill(dbdataset);
        BindingSource bsource = new BindingSource();
        bsource.DataSource = dbdataset;
        dataGridView2.DataSource = bsource;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void delete_btn_Click(object sender, EventArgs e)
{
    new Form5().ShowDialog();
}

private void arhiv_btn_Click(object sender, EventArgs e)
{
    new Form6().ShowDialog();
}

private void dolzhComboBox_SelectedIndexChanged(object sender, EventArgs e)
{
    a1 = "select d.dolzh_name as Должность, f.familiya as Фамилия,f.imya as
Имя,r.region_name as Регион, o.otdel_name as Отдел from FIO f join region r on
```

Ә қосымшаның жалғасы

```
f.id_region=r.id_region join otdel o on o.id_otdel=f.id_otdel join dolzhnost d on
f.id_dolzh=d.id_dolzh where dolzh_name = "" + dolzhComboBox.Text + "";
    connect();
    }}}
}
```

Form 4

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

namespace WindowsFormsApplication1
{
    public partial class Form4 : Form
    {
        public Form4()
        {
            InitializeComponent();
            var reg = (from i in Program.db.region select i.region_name);
            foreach (var r in reg)
            {
                comboBox1.Items.Add(r);
            }
            var otd = (from i in Program.db.otdel select i.otdel_name);
            foreach (var o in otd)
            {
                comboBox2.Items.Add(o);
            }
            var dolzh = (from i in Program.db.dolzhnost select i.dolzh_name);
            foreach (var d in dolzh)
            {
                comboBox3.Items.Add(d);
            }
        }

        private void button1_Click(object sender, EventArgs e)
        {
            int region_id = (from i in Program.db.region where i.region_name == comboBox1.Text
            select i.id_region).SingleOrDefault();
            int otel_id = (from i in Program.db.otdel where i.otdel_name == comboBox2.Text select
            i.id_otdel).SingleOrDefault();
```

Ә қосымшаның жалғасы

```
int dolzh_id = (from i in Program.db.dolzhnost where i.dolzh_name == comboBox3.Text
select i.id_dolzh).SingleOrDefault();
var dog_s1 = date_dog.Value;
var dog_s2 = date_ist.Value;
var dog_dob = dob.Value;
string constring=@"Data Source=(local);Initial Catalog=baza;Integrated Security=True";
string Query = "INSERT INTO FIO (familiya,
imya,otchestvo,zarplata,id_region,id_otdel,id_dolzh,data,date_dog,date_ist,login1,passw) VALUES
('" + fam_txt.Text + "','" + imya_txt.Text + "','" + otch_txt.Text + "','" + zarplata_txt.Text + "','" +
region_id + "','" + otdel_id + "','" + dolzh_id + "','" + dog_dob + "','" + date_dog + "','" + date_ist +
 "','" + login_txt.Text + "','" + parol_txt.Text + "')";
SqlConnection conDataBase=new SqlConnection(constring);
SqlCommand cmdDataBase=new SqlCommand(Query, conDataBase);
SqlDataReader myReader;

try{
    conDataBase.Open();
    myReader=cmdDataBase.ExecuteReader();
    MessageBox.Show("Новые данные сохранены");
    while(myReader.Read()){
    }
} catch(Exception ex){
    MessageBox.Show(ex.Message);
}

private void button4_Click(object sender, EventArgs e)
{
    Form4.ActiveForm.Close();
}

private void button1_Click_1(object sender, EventArgs e)
{
    string constring = @"Data Source=(local);Initial Catalog=baza;Integrated Security=True";
    string Query = "INSERT INTO region(region_name,kod) VALUES ('" +
newregion_txt.Text + "','" + newkod_txt.Text + "')";
    SqlConnection conDataBase = new SqlConnection(constring);
    SqlCommand cmdDataBase = new SqlCommand(Query, conDataBase);
    SqlDataReader myReader;

    try
    {
        conDataBase.Open();
        myReader = cmdDataBase.ExecuteReader();
        MessageBox.Show("Новые регион добавлен...");
        while (myReader.Read())
        {
        }
    }
    catch (Exception ex)
```

Ә қосымшаның жалғасы

```
{
    MessageBox.Show(ex.Message);
}
}

private void button2_Click(object sender, EventArgs e)
{
    string constring = @"Data Source=(local);Initial Catalog=baza;Integrated Security=True";
    string Query = "INSERT INTO otdel (otdel_name) VALUES ('" + newotdel_txt.Text + "'
)";
    SqlConnection conDataBase = new SqlConnection(constring);
    SqlCommand cmdDataBase = new SqlCommand(Query, conDataBase);
    SqlDataReader myReader;

    try
    {
        conDataBase.Open();
        myReader = cmdDataBase.ExecuteReader();
        MessageBox.Show("Новый отдел добавлен...");
        while (myReader.Read())
        {
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void button3_Click(object sender, EventArgs e)
{
    string constring = @"Data Source=(local);Initial Catalog=baza;Integrated Security=True";
    string Query = "INSERT INTO dolzhnost (dolzh_name) VALUES ('" + newdolzh_txt.Text
+ "' )";
    SqlConnection conDataBase = new SqlConnection(constring);
    SqlCommand cmdDataBase = new SqlCommand(Query, conDataBase);
    SqlDataReader myReader;

    try
    {
        conDataBase.Open();
        myReader = cmdDataBase.ExecuteReader();
        MessageBox.Show("Новая должность добавлена...");
        while (myReader.Read())
        {
        }
    }
    catch (Exception ex)
    {
    }
}
```

Ә қосымшаның жалғасы

```
MessageBox.Show(ex.Message);  
    }  
  
    }  
    }  
    }
```

Form 5

```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Windows.Forms;  
using System.Data.Sql;  
using System.Data.SqlClient;  
  
namespace WindowsFormsApplication1  
{  
    public partial class Form5 : Form  
    {  
        public Form5 ()  
        {  
            InitializeComponent();  
            var fio = (from i in Program.db.FIO select i.familiya);  
            foreach (var fam in fio)  
            {  
                comboBox1.Items.Add(fam);  
            }  
        }  
  
        string cn, a, a1;  
  
        public void connect()  
        {  
            SqlConnection cn = Program.conn;  
            a = a1;  
            DataSet ds = new DataSet();  
            SqlDataAdapter da = new SqlDataAdapter(a, cn);  
            da.Fill(ds);  
            dataGridView1.DataSource = ds.Tables[0];  
        }  
  
        private void button1_Click(object sender, EventArgs e)  
        {  
            var primechanie = textBox1.Text;  
            string constring = @"Data Source=(local);Initial Catalog=baza;Integrated Security=True";
```

Ә қосымшаның жалғасы

```
string Query= "INSERT INTO arhiv
(id_sotrudnik,imya,otchestvo,data,id_region,id_otdel,id_dolzh,dog_date,ist_date,zarplata,pol,login1
,passw,primechanie) select top (1) from FIO where familiya= '' + comboBox1.Text + ''";
SqlConnection conDataBase=new SqlConnection(constring);
SqlCommand cmdDataBase=new SqlCommand(Query, conDataBase);
SqlDataReader myReader;

try{
    conDataBase.Open();
    myReader=cmdDataBase.ExecuteReader();
    MessageBox.Show("Данные перемещены в архив");
    while(myReader.Read()){
    }
} catch(Exception ex){
    MessageBox.Show(ex.Message);
}

private void Form5_Load(object sender, EventArgs e)
{
    // TODO: данная строка кода позволяет загрузить данные в таблицу
    "bazaDataSet1.FIO". При необходимости она может быть перемещена или удалена.
    this.fIOTableAdapter.Fill(this.bazaDataSet1.FIO);
}

private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    a1 = "select f.familiya as 'Фамилия',f.imya as 'Имя',f.otchestvo as
'Отчество',r.region_name as 'Регион', o.otdel_name as 'Отдел',f.date_dog as 'Договор от'
,f.date_ist as 'до' from FIO f join region r on f.id_region=r.id_region join otdel o on
o.id_otdel=f.id_otdel where f.familiya = '' + comboBox1.Text + ''";
    connect();
}
}}
```

Form 6

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

namespace WindowsFormsApplication1
{
    public partial class Form6 : Form
    {
        public Form6()
```

Ә қосымшаның соңы

```
{
    InitializeComponent();
}

private void arhivBindingNavigatorSaveItem_Click(object sender, EventArgs e)
{
    this.Validate();
    this.arhivBindingSource.EndEdit();
    this.tableAdapterManager.UpdateAll(this.bazaDataSet3);
}

private void Form6_Load(object sender, EventArgs e)
{
    // TODO: данная строка кода позволяет загрузить данные в таблицу
    "bazaDataSet3.arhiv". При необходимости она может быть перемещена или удалена.
    this.arhivTableAdapter.Fill(this.bazaDataSet3.arhiv);
}

private void button1_Click(object sender, EventArgs e)
{
    Form6.ActiveForm.Close();
}}}
```