

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Коммерциялық емес акционерлік қоғамы
АЛМАТЫ ЭНЕРГЕТИКА ЖӘНЕ БАЙЛАНЫС УНИВЕРСИТЕТІ

«Компьютерлік технологиялар» кафедрасы

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

(қолы)

« _____ » _____ 2014 ж.

ДИПЛОМДЫҚ ЖОБА

Тақырыбы: «Кафедраның автоматтандырылған жүйесін жобалау және
күру»

5В070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету»
мамандығы бойынша

Орындаған Жексенқұлов Қуаныш Ерболсынұлы тобы: ВТк-10-2

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

Кеңесшілер :

Экономикалық бөлім бойынша :

доцент Боканова Г.Ш.

« 02 » 06 2014ж.

(қолы)

Өмір тіршілігі қауіпсіздігі бойынша:

т.ғ.к., аға оқытушы Муташева Г.С.

« 28 » 05 2014 ж.

(қолы)

Есептеу техникасын қолдану бойынша :

ф.-м.ғ.к., доцент Аманбаев А.А.

« 06 » 06 2014 ж.

(қолы)

Мөлшер бақылаушы:

аға оқытушы Рахимжанова З.М.

« 06 » 09 2014 ж.

(қолы)

Пікір жазушы :

У.А. Жолдасбеков атын. ИМиМ, профессор, т.ғ.д. Баймухаметов А.А.

« 05 » 06 2014 ж.

(қолы)

Алматы 2014

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

Жексенқұлов Қуаныш Ерболсынұлы

Жоба тақырыбы: «Кафедраның автоматтандырылған жүйесін жобалау және құру» ректордың « 24 » қыркүйек № 115 бұйрығы бойынша бекітілген. Аяқталған жұмысты тапсыру мерзімі: « 03 » маусым 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 2010 (ASP.NET) бағдарламалау ортасын пайдалану арқылы оқу орнының сабақтарын автоматтандыру қарастырылды.

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

- Автоматтандырылған жұмыс орнының анықтамасы (арналуы, құрылымы, т.б.).
- Ақпараттық жүйелер, дерекқорлар (ДК), дерекқорларды басқарудың жүйелері (ДҚБЖ) түсініктері.
- 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.

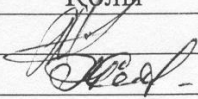

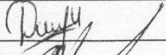
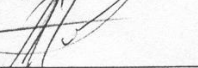
Сызба материалдарының (міндетті түрде дайындалатын сызуларды көрсету) тізімі:

- Дерекқордың ER диаграммасы.
- Жасалған бағдарламалық қосымшаның жұмыс істеу қабілеттілігін мен функционалдығын көрсететін графикалық материал.

Негізгі ұсынылатын әдебиеттер:

1. Герберт Шилдт. C# 4.0: полное руководство = C# 4.0 The Complete Reference.- М.: «Вильямс», 2010. — С. 1056.
2. Кренке Д. Теория и практика построения баз данных: [пер.с англ] / Д. Кренке. - 9 - е изд. - СПб.: Питер, 2005. - 858 с.
3. Ульман, Дж. Основы систем баз данных, **2012**. - 292 с.
4. Айтхожаева Е.Ж. Стандартный язык баз данных SQL: учеб. пособие. – Алматы: АУЭС, 2005. - 48 с.
5. Ибраева Л.К., Сатимова Е.Г. Проектирование баз данных. МУ к выполнению лабораторных работ. – Алматы: АУЭС, 2004.
6. Марк Шпенник, Оррин Следж. Microsoft SQL Server 2000 DBA Survival Guide. – М.: Вильямс, 2001. – 236 с.Никита Культин. Основы программирования в ASP.NET 2010. – Санкт-Петербург, 2010. – 448с.
7. Алексей Вишневикий. SQL Server . Эффективная работа. – Санкт-Петербург, 2009. – 541с.
8. Никита Культин. Основы программирования в ASP.NET 2010. 2-е издание. –Санкт-Петербург, 2010. – 640с.

Жоба тараулары бойынша кеңес берушілер және оның мерзімі:

Бөлім	Кеңесші	Мерзімі	Қолы
Негізгі бөлім	Аманбаев А.А.	10.02.2014 - 27.02.14	
Тіршілік қауіпсіздігі	Муташева Г.С.	12.04.2014 - 14.05.2014	
Экономикалық бөлім	Боканова Г.Ш.		
Норма бақылаушы	Рахимжанова З.М.	08.04.2014 - 09.04.14	
Есептеу техникасын қолдану	Аманбаев А.А.	13.03 - 16.04.2014	

ДИПЛОМ ЖОБАСЫН ДАЙЫНДАУ

КЕСТЕСІ

№ р/с	Тарау аттары, әзірленетін сұрақтардың тізімі	Жетекшіге ұсыну мерзімдері	Ескерту
1	Автоматтандырылған жұмыс орнының анықтамасы (арналуы, құрылымы, т.б.).		
2	Ақпараттық жүйелер, дерекқорлар (ДҚ), дерекқорларды басқарудың жүйелері (ДҚБЖ) түсініктері.		
3	Жүйенің дерекқорын жобалау және жүзеге асыру, жобалау құралдарын таңдау, ER моделін жобалау.		
4	Дерекқорды құру, пайдаланушы интерфейсін жобалау, дерекқормен байланысын орнату.		

Тапсырманың берілген уақыты « _____ » _____ 2014 ж.

Кафедра меңгерушісі

(КОЛЫ)

ф.-м.ғ.д., проф. Құралбаев З.К.

Жоба жетекшісі

(КОЛЫ)

доцент, ф.-м.ғ.к. А.А. Аманбаев

Орындалаты тапсырманы
кабылдаған студент

(КОЛЫ)

Жексенқұлов Қ.Е.

ңдатпа

Бұл дипломдық жобада кафедрасының мұғалімдерінің жүктемесін бөлуді жүзеге асыратын программа құрылады.

Ақпараттық жүйені MS SQL Server 2008 ДҚБЖ (дерекқорларды басқару жүйесінде) жасалған дерекқор және Microsoft Visual Studio 2010 (ASP.NET) бағдарламалық ортасында құрылған клиенттік қосымша құрайды. Дипломдық жобаны жазу барысында ақпараттық жүйенің архитектурасын, ДҚ құрылымын және клиенттік қосымшаны жасау жұмыстары жүргізілген.

Тіршілік қауіпсіздігі бөлімінде жұмыс бөлмесіндегі ауа алмасу (вентиляция) жүйесінің есептері келтірілген.

Экономикалық бөлімде енгізілген жүйеден алынған экономикалық шығындар мен пайданы есептеу жүргізілген.

Аннотация

В данном дипломном проекте рассматривается решение задачи создания программы для распределение нагрузок преподавателей кафедры. Информационная система представляет собой базу данных, разработанную на СУБД MS SQL Server 2008 и клиентское приложение, созданное в среде программирования Microsoft Visual Studio 2010 (ASP.NET).

В процессе написания дипломного проекта были разработаны архитектуры информационной системы, структуры БД и клиентского приложения.

В части по безопасности жизнедеятельности представлены расчеты вентиляции в рабочем помещении.

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

Abstract

In this thesis project is considered the task of creating programs for the distribution of loads CT teachers in the Department . Information system is a database developed by the DBMS MS SQL Server 2008 and the client application created in the medium programmitovaniya Microsoft Visual Studio 2010 (ASP.NET). Also addresses the issue of data security and confidentiality . In the process of writing the thesis project developed information system architecture , database structure and the client application.

As part of Health and Safety provides estimates of ventilation in the working area.In the economic part of a calculation resulting economic costs and benefits of the implemented system.

Мазмұны

Кіріспе.....	8
1 Техникалық тапсырма.....	9
1.1 Тағайындалуы және қолдану аймағы	9
1.2 Сенімділікке қойылатын талаптар.....	9
1.3 Техникалық құрылғыларға қойылатын талаптар.....	9
1.4 Бағдарламалық құжатқа қойылатын талаптар.....	9
1.5 Құру кезеңдері мен сатылары	10
1.6 Кезең бойынша жұмыстардың мазмұны	10
1.7 Тестілеу.....	10
1.8 Автоматтандырылған жұмыс орнының арналуы мен құрылымы.....	12
1.9 АЖО құрылымы.....	15
1.10 АЖО-ға қойылатын жалпы талаптар.....	16
1.11 АЖО-ды жобалау сатылары.....	17
2 Ақпараттық жүйелер. Деректер қорын жобалау.....	18
2.1 Ақпараттық жүйелер туралы қысқаша мағлұматтар.....	228
2.2 Ақпараттық жүйе жобалау сатылары	19
2.3 Ақпараттық жүйенің тіршілік циклі.....	21
2.4 АЖ–ның басты сенімділік түсінігі	432
2.5 Деректер қорын жобалау. Деректер моделі.....	25
2.6 Реляционды деректер қорына қойылатын талаптар	32
2.7 SQL тілі туралы негізгі деректер	343
2.8 Кесте құру және өшіру.....	34
3 Дерекқорды жобалау және оны іске асыру	37
3.1 ДҚ құрудың бастапқы кезеңі.....	37
3.2 Концептуалды жобалау.....	37
3.3 Дерекқор және қосымшаны жобалау құралдарын таңдауы.....	38
3.4 ER моделін құру.....	39
3.5 Бизнес ережелер.....	40
3.6 Реляционды қатынастарды құрастыру.....	42
3.7 Логикалық жобалау.....	45
3.8 Көрсетілімдер және процедуралар.....	47
3.9 Физикалық жобалау. Дерек қорына қосымша құру.....	48
3.10 Бағдарламаны сипаттау.....	50
4 Тіршілік қауіпсіздігі.....	56
4.1 Ауа алмасу (кондиционер) жүйесінің құрылғысы және есебі.....	56
4.2 Бөлменің электр торабы	58
4.3 Бөлменің өрт қауіпсіздігі.....	59
4.4 Бөлмедегі және сырттағы зиянды элементтер.....	59
4.5 Бөлмедегі дыбыстар , яғни шу.....	60
4.6 Бөлменің жарықтандыру көздері	60

4.7	Бөлменің ауа жүйесі.....	60
4.8	Кондиционерлеу және ауаны жаңарту жүйелерін есептеу.....	61
4.9	Температура айырымы нәтижесінде алынатын жылу, жылу жоғалту.....	62
4.10	Адамдардан келетін жылу.....	63
4.11	Ауа алмасуды есептеу.....	66
4.12	Табиғи жарықты есептеу.....	66
5	Техника-экономикалық негізделуі.....	69
5.1	Жоба сипаттамасы.....	69
5.2	Маркетинг–микс қызметтері.....	70
5.3	Бағдарламалық қамтаманың өңделуінің еңбек өнімділігінің есептелуі.....	71
5.4	Бағдарламалық қамтаманы өңдеуге кеткен шығынның есептелуі.....	73
5.5	Қосымша шығындар статьясы.....	74
5.6	Өзіндік құн нәтижесінің кестесі	78
5.7	Бағдарламаның бағасына есептеу.....	78
5.8	Минималды бағаны, кірісті және пайданы есептеу.....	78
5.9	Бағдарламалық өнімді иемденудегі кәсіпорын шығындарын есептеу.....	79
5.10	Қолдану саласындағы жылдық бізмезгілдегі шығындарды есептеу.....	80
5.11	Ақпараттық өнімді енгізуден алынған үнемдеу мөлшері.....	81
5.12	Пайда мен шығындарды есептеу.....	82
5.13	Ақшалай құралдардың қозғалысы.....	82
5.14	Экономикалық тиімділікті есептеу.....	83
5.15	Пайда индексін есептеу (Profitability index, PI).....	84
5.16	Табыстың ішкі нормасын есептеу (Internal rate of return, IRR).....	84
5.17	Өтімділік периодын есептеу (Payback period, PBP).....	85
	Қорытынды.....	87
	Пайдаланылған әдебиеттер тізімі.....	88
	А қосымшасы.....	89
	Ә қосымшасы.....	91

Кіріспе

Бұл дипломдық жобада кафедра мұғалімдерінің сабақтарын бөлу үшін жасалған, дерекқор негізінде жасалған клиенттік программалық жоба.

Бұл бағдарлама университет, колледж, академия т.б. мұғалімдерінің сабақтарын тиімді бөлу үшін жасалған. Оқу орындарында негізінен оқытушылар мен топтар, пәндер, бөлінетін жүктемелер туралы ақпаратты жүйелеумен байланысты болады. Сондықтан, кафедра мұғалімінің жүктемесін бөлу жұмысы өте ыңғайлы, әрі жылдам бөлу үшін, барлық деректерді жүйелі түрде сақтау және қажетті деректі тез алу, сонымен қатар жаңа деректерді ыңғайлы түрде енгізу мүмкіндіктері қарастырылған.

Бағдарламаны құру үшін алдымен зерттелетін облысының талдауы жасалды. Келесі кезеңде оқытушылар мен пәндер, топтар туралы қажетті деректерді сақтайтын логикалық байланысқан кестелер құрылып, олардың негізінде жүйенің инфологиялық моделі алынады. Инфологиялық модель MS SQL Server–жүйесінде жасалады. Кафедрасының дерекқоры MS SQL Server дерекқорларды басқару жүйесінде құрылғаннан кейін, пайдаланушылардың барлық сұраныстарын қанағаттандыратын клиенттік қосымша жобаланады. Клиенттік бағдарлама Microsoft Visual Studio 2010 (ASP.NET) программалық ортасында іске асырылады.

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

1 Техникалық тапсырма

1.1 Тағайындалуы және қолдану аймағы

STYS – 2 ақпараттық жүйесі кафедрасының мұғалімдерінің жүктемесін бөлуге арналған. Программа керекті есептеулерді автоматты түрде жүргізеді, есептеу барысында кететін қателерді азайтады. Сонымен қатар, мынадай бөлімдерді қамтиды:

- пәндер;
- бөлінген жүктемелер;
- бөлінбеген жүктемелер;
- есеп беру;
- анықтама.

1.2 Сенімділікке қойылатын талаптар

Бағдарлама толық сенімді функционалды жұмыс жасауы үшін тапсырыс беруші ұйымдық – техникалық шаралардың жиынтығын қамтамасыз ету керек. Олар төмендегідей болады:

- техникалық құралдарға үзіліссіз қоректену көзін ұйымдастыру;
- лицензиялық бағдарламалық қамтуды пайдалану компьютерлік вирусқа бағдарламалық құралдарды тексеру.

1.3 Техникалық құрылғыларға қойылатын талаптар

Бағдарлама серверде сақталады. Бағдарламамен жұмыс істеу үшін бар болғаны компьютер, браузер керек. Компьютер сипаттамалы төмендегідей:

- процессор Pentium – 2,0 Hz;
- жедел жады кез келген;
- HDD, 10 Гигабайт;
- операционды жүйе Windows 2000 Server , Windows 2003,XP,7,8;
- Браузерлер Opera, Google chrome, Internet explorer, Mozilla т.б.

1.4 Бағдарламалық құжатқа қойылатын талаптар

Бағдарламалық құжаттың құрамы келесідей болу керек:

- техникалық тапсырма;
- бағдарлама және байқаулардың әдістері;
- пайдаланушыға көмек.

1.5 Құру кезеңдері мен сатылары

Құру сатылары.

Құру келесі үш сатыны қамту керек:

- техникалық тапсырманы құру;
- жобалау жұмысы;
- енгізу.

Құру кезеңдері.

Техникалық тапсырмаға сай құру жұмыстары кезеңмен орындалу керек. Құру кезеңдері келесідей болады:

- ақпарат жүйесін құру;
- бағдарламалық құжатты құру;
- мысалмен бағдарламаның тексерілуі.

Бағдарламаны енгізу кезінде құру кезеңдері орындалып және бағдарламаны беру болып табылады.

1.6 Кезең бойынша жұмыстардың мазмұны

Құру кезеңінде келесі жұмыстар орындалу керек:

- Тапсырманың мақсатын анықтау;
- Техникалық құрылғыларға қойылатын талаптарды анықтау;
- Бағдарламаға қойылатын талаптарды анықтау.

1.7 Тестілеу

Тестілеу – бұл ақпараттық жүйеде қателерді іздеу болып табылады.

Қате – бұл бар нәрсенің бір – біріне сәйкес келмеуі.

Сенімділік – бұл әрбір қабылдалмаудың құнын есепке ала отырып есептелінген, белгілі бір уақыт ішіндегі қарсылық жасалмаған жұмыс ықтималдығы.

Тестілеу принциптері.

Тестілеу жақсы жобаланған жүйеде қалған қателерді табу үшін, сонымен қатар жүйенің сенімділігін арттыру үшін және құндылығын арттыру үшін жүргізіледі. Тестілеу арқылы нашар жобаланған жүйенің жоғарғы дәрежедегі сенімділігіне жетуге болмайды.

Егер біз бағдарламаны тестілейтін болсақ, онда қандай да бір жағдайлармен бағдарламаның бағасын көтеру арқылы тестілеуге кеткен шығынды қайтарып алуымыз қажет. Мұны тек бағдарламаның сенімділігін жоғарлату арқылы ғана жасауға болады. Сенімділікті тек үрдіске енгізілген қателерді түзету арқылы жоғарлатуға болады.

Сәтті тест деп, қатені тапқан тестті айтамыз. Егер бірде бір қате табылмаса, онда ол сәтсіз тест деген сөз.

- қателер топтасу қасиетіне ие;

- тестілеуді жеңілдету үшін еш уақытта бағдарламаны өзгертпеңіз;
- бағдарламаны тестілеудің авторымен кездесуден қашу керек. Егер программист бағдарламаны жазуда қате жіберсе, онда оны тестілеуде де дәл сол қатені жіберуі ықтимал;
- тестіні құру – бұл шығармашылық үрдіс, яғни кей жағдайда мидың жұмыс істеуін қажет ететін үрдіс;
- қатені жақсы табатын тест жақсы тест болып табылады;
- бағдарлама неге арналған, яғни тек өзіне ғана тиесілі істі істейді ме, әлде басқа жұмыстарды да істейді ме соны тексеру керек;
- комментариилердің жеткіліксіздігі қателерді іздеуді қиындатады;
- комментариилердің аздығы қатені іздеуді қиындатумен қатар, түсініктеме сөздігінсіз күрделі бағдарламаны талқылау да қиын.

Тестілеуден кейін қатенің жоқ екендігіне кепілдік беруге болмайды, тек белгілі бір деңгейде жүйе жұмысының дұрыстығының сенімділігі жайлы айтуға болады.

Тест – бұл кіріс деректерінің жиынтығы және (немесе нәтижені күтуші қолданушының қызметі) жауап бағдарламалардың іс-әрекеті. Қарапайым бағдарламамен толық, жан-жақты тестілеу жүргізу мүмкін емес, себебі оған уақытта, қажетті ресурстарда жетпейді. Сондықтан тестілеудің қатені табу ықтималдылығына байланысты, тестті құру әдісіне байланысты бірнеше түрі бар.

Тестілеу түрлері:

- құрылымдық тестілеу (ақ жәшік);
- функционалдық (қызметтік) тестілеу (қара жәшік).

Құрылымдық тестілеу.

Берілген тестілеуде бағдарламаның тексті көрініп тұрады. Шешімдердің блоктары, циклдері тестіленеді.

Құрылымдық тестілеудің бірнеше типі бар:

- операторларды жабу;
- шешімдерді жабу;
- шарттарды жабу;
- шарттардың аралас жабулары;
- циклдердің тестіленуі.

Толық түсініктер үшін құрылымдық тестілеудің мысалдарын қарастыруға болады.

Функционалдық (қызметтік) тестілеу.

Берілген тестілеуде бағдарламаның тексті көрінбейді және бағдарлама қара жәшік түрінде, яғни кіріс және шығыс шарттары белгілі, сонымен қоса жұмыстың жалпы схемасы қарастырылады. Бағдарлама оның спецификациясы бойынша тексеріледі.

Функционалдық тестілеудің бірнеше түрі бар:

- эквивалентті кластар;
- шекті мәндерінің анализі;
- белгілі бір мөлшердегі тестілеу;

- белгілі бір көлемдегі тестілеу;
- қорғанысты тестілеу;
- жүйе эксплуатациясы;
- тәжірибелі эксплуатация.

Үлкен түсініктер үшін функционалдық тестілеудің мысалдарын қарастыруға болады.

1.8 Автоматтандырылған жұмыс орнының (АЖО) арналуы мен құрылымы

Автоматтандырылған жұмыс орны (АЖО) немесе шетел терминологияда «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 АЖО – мен және басқа жергілікті есептеу желілерімен т.б. түйіндесетін құрылғылармен толықтырылатын әдетте дербес компьютер болып табылады.

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

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

АЖО – да шешілетін мәселелерді шартты түрде ақпараттық және есептеуіш деп 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 семантикалық 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.

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 прграммалау 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.

1.9 АЖО құрылымы

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

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

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

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

АЖО-ның қамтамасыз ету бөлігі:

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

АҚЕ АЖО бұл құжаттардың, ақпарат массивтерінің (ДҚ) және кодалаудың стандартталған және бәрыңғай формалар жиынтығы.

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

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

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

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

1.10 АЖО-ға қойылатын жалпы талаптар

АЖО – ға қойылатын жалпы талаптар келесідей:

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

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

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

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

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

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

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

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

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

Мұндай жүйеде келесі қамтамасыз етілу керек:

– мамандардың ақпаратты өңдеу машина құралдарына максималды жақындығы;

– диалогты тәртіпте жұмыс жасау;

– эргономика талаптарына сәйкестілік;

– компьютердің жоғары икемділігі;

– ескілік процестердің максималды автоматтандырылуы;

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

– мамандардың өзін-өзі оқыту мүмкіндігі.

1.11 АЖО-ды жобалау сатылары

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

– бар басқару жүйесін зерттеу мен сараптама жасау;

– АЖО құруының техникалық тапсырманы құрстыру;

– АЖО – ды техникалық жобалау;

– АЖО – ды жұмыстық жобалау.

2 Ақпараттық жүйелер. Деректер қорын жобалау

2.1 Ақпараттық жүйелер туралы қысқаша мағлұматтар

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

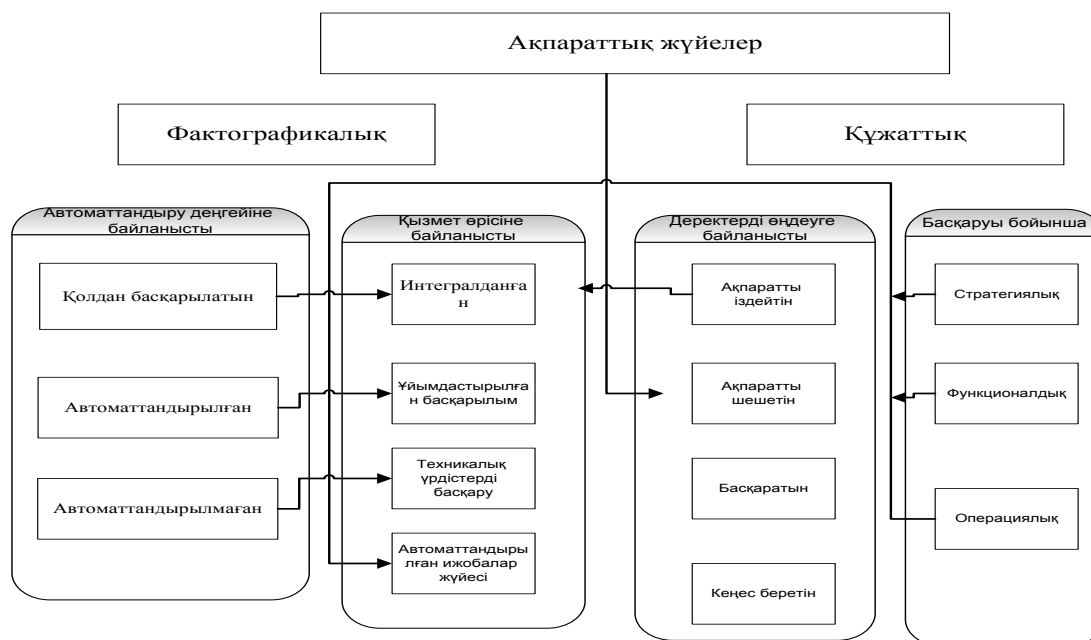
- сақтау;
- ақпарат ауысу;
- ақпаратты түрлендіру.

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

Ақпарат жүйесінің көмегімен шешілетін тапсырмалардың әр түрлі болып келуіне арқасында құрылу жағынан бір – бірінен айырмашылығы бар ақпараттық жүйелердің әр түрлі болып бөлуіне әкеп соқты.

Ақпараттық жүйелерді көптеген белгілері бойынша бөлуге болады. Негізгісі белгісі, функционалдық мүмкіндігі мен құрылу ерекшеліктерімен бөлу болып табылады. Орындайтын тапсырмаға, қолданатын техникалық құралдарға байланысты ақпараттық жүйелер әр түрлі сыныптарға бөлінеді (Сурет 2.1).

Деректерді сақтау бойынша ақпараттық жүйелер фактографикалық және құжаттық болып бөлінеді. Фактографикалық жүйелер сан және мәтін түріндегі деректерді сақтауға және өңдеуге арналған.



Сурет 2.1 – Ақпараттық жүйелердің бөлінуі

2.2 Ақпараттық жүйе жобалау сатылары

Тап осы лекцияда қарастырылған типтік тіршілік цикл бірінші кезеңдері ақпараттық жүйе жасау, жүйеге техникалық тапсырма берумен және өңдеумен аяқталады.

Стандартқа сәйкес ISO/IEC 12207 ақпараттық жүйені жобалау үрдістері үш топқа бөлінген.

Негізгі үрдістер:

- тауып алу;
- жеткізу;
- өңдеу;
- қанау;
- ере жүру.

Қосалқы үрдістер:

- құжаттау;
- кескін үйлесімімен басқару;
- сапасын қамтамасыз ету;
- верификация;
- іскерлік сараптау;
- бірге бағалау;
- тексеру;
- проблемалардың рұқсаты.

Ұйымдық процестер:

- басқару;
- инфражүйе жасау;
- жетілдіру;
- оқу.

Күрделі жүйені жобалау процесі өз алдында шығармашылық жұмыс, ал бұнда қолданылатын технология – ұйымдасқан шығармашылық технология – пәндік облыстағы фундаментальдік зерттеу сатысы мен дайындау және қолдану жүйесі арасындағы процестердің жиынтығы.

Бұндай технологияның бірінші сатысы ақпараттық жүйе құру концепциясымен аяқталатын пәндік облысты зерттеу болып табылады.

Көптеген шынайы жүйе иерархиялық құрылымға ие. Бұл жүйенің ерекшелігі:

- арнайыланған жеке жұмыстарды белгілеу және оны арнайы дайындалған элементтермен бөлек беру;
- координация – жүйенің негізгі мақсатына жетуге бағытталған арнайы элементтің жиынтығын қамтамасыз ету.

Иерархиялық жүйені қолдана отырып жеке шешім қабылдай алатын, иерархиялық элементтің өзара әсерінің жүйесіндегі ұйымды анықтауға болады. Ұйымның өз бағытында қолданып отырған қиындықтарды шешу ережелерінің жиынтығы стратегия деп аталады.

Барлық ақпараттық жүйе қабылдау сатысында жүйелік жақындау қолданылады. Бұнда жүйелік модельдің көмегімен іске асатын әр түрлі саладағы, типтегі,

кластардағы ұйымдардың және пәндік облыстардың зерттелуі мен өңдеуі қарастырылады. Мамандар жүйелік жақындау жоқ деп қабылдау бұл осы жүйеге енудің жақсы жолы деп айтады. Жүйелік жақындау уақытпен тексерілген және қарапайым зерттеу іздеу әдісіне қарама – қарсы келеді.

Бұндай жүйе кәдімгі коммерциялық әдіспен алынбайды. Бәсекелестік бұл жағдайда тек ұсынысты зерттеу және алдын ала өңдеумен ғана шектеледі. Шешуші техникалық және жұмысты жобалау тек бір ғана ұйыммен келісім жасалған соң орындалады. Тапсырыс беруші тапсырманы бекіткеннен кейін нарыққа ұсынысын көрсете алмайды. Сондықтан жүйелік талдау сатысы АЖ жоғарғы мәнге ие. Мысалы, АҚШ – тың қорғаныс министрлігі 1965 жылы №32009 директивасын шығарды, онда құрылым концепция сатысынсыз күрделі жүйені өңдеу туралы келісім қабылданбайды делінген.

Концепция түсінігі – ол анық түсінім әдісі, белгілі бір заттың, құбылыстың, процестің жеткізілуі. Кейбір сатыда бір – бірімен байланысқан білім, факт, тапсырма мен қызығушылықтар жиналып қалады.

Бұндай жағдай адам санасы тез дамып келе жатқан облыстарда көп кездеседі; тез дамып келе жатқан облыстарда бұл үрдіс дағдарысқа алып келеді.

Жүйенің ерекшелігі барлық компоненттердің байланысын анықтайтын модельдің болуы. әсіресе жеке компоненттердің қасиеттерін жүйенің қасиетімен байланыстыратын модель.

Бұл жүйенің шешімі теңдік жүйесін құру арқылы жүзеге асады. Онда түрлі компоненттердің қасиеттерінің бір – біріне және жүйе қасиетіне әсерін көрсетеді. Әр компонент жеке өз қасиетін жақсартып алмайды, себебі ол өзгеріс басқа компоненттерге ісерін тигізеді. Ол ішкі жүйеге кері әсерін тигізеді, тіпті жалпы жүйенің бұзылуына алып соғады.

Осылайша, тапсырма тек теңдік жүйесін құру арқылы ғана шешуге болады. Кейбір компоненттер жобаланады және осы жүйеге арналып жасалады. Барлық осы арнайы компоненттерді қолдану және осылайша түскен табысты компонентті құрғандарға бөліп беруді қамтамасыз ететін нарық болу үшін жүйе жеткілікті үлкен болуы керек.

Тапсырылымның бітетін ақпараттық жүйе құру сатысы рационалды жоспарлау және ұйымды жұмыстардың үйлесімімен ерекшеленеді. Ақпараттық жүйенің тіршілік циклына мына модельдер:

- каскадты;
- аралық бақылаумен жүретін тарауланған модель;
- спиральды модель.

2.3 Ақпараттық жүйенің тіршілік циклі

Ақпараттық жүйенің тіршілік циклін жүйені құру үрдісі және оны пайдалану кезіндегі болатын оқиғалар түрінде көрсетуге болады.

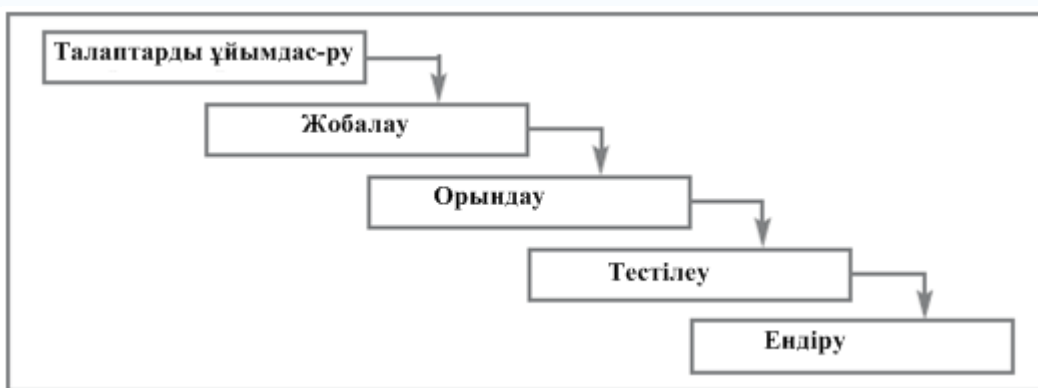
Тіршілік циклі жүйенің әр түрлі күйін көрсетеді. Оның бастапқы қажеттілігенен бастап толық пайдалануынан тыс қалғанша қарастырады.

Қаіргі кезде жоғарыда айтылғандай тіршілік моделінің келесі түрлері қолданылады:

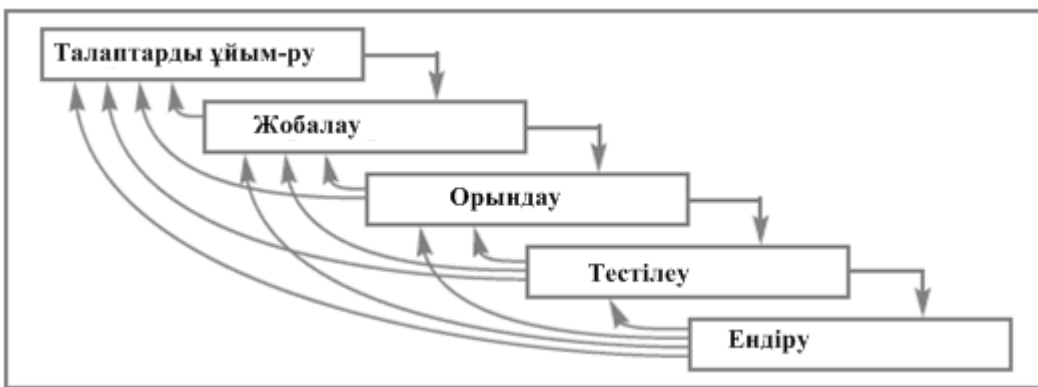
– Каскадты модель (Сурет 2.2) жобаның бекітілген ретімен барлық кезеңдерінің орындалуын қамтамасыз етеді. Келесі кезеңге өту алдындағы кезеңдегі жұмыстын толықтай аяқталуын білдіреді.

– Аралық бақылаумен жүретін тарауланған модель (Сурет 2.3) ақпараттық жүйенің құрылуы, кезеңдер аралығымен кері байланыс циклының итерациясымен басқарылады. Аралық кезең түзетулері.

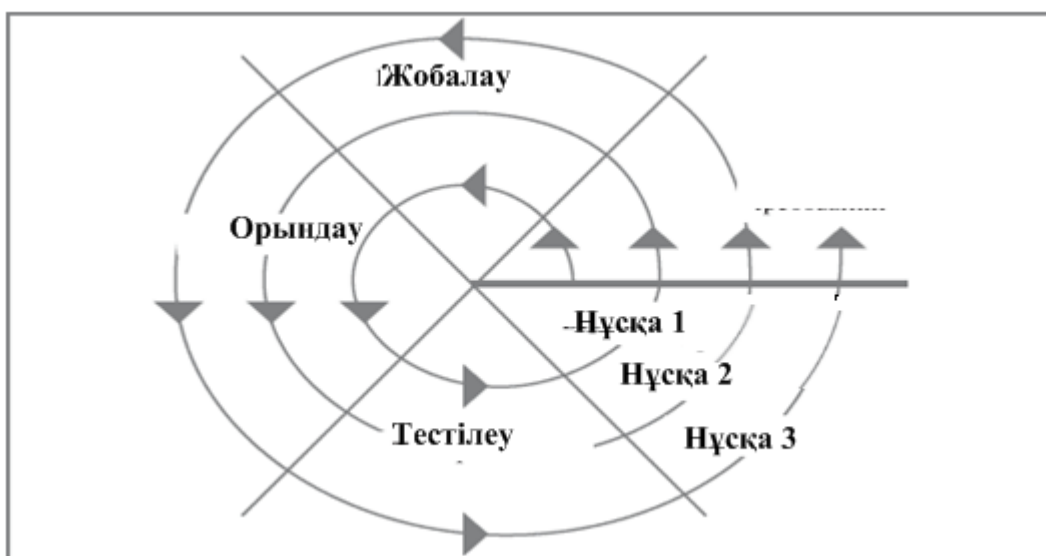
– Спиральды модель (Сурет 2.4) әр спиральдың орамында өнімнің келесі нұсқасы құрылады, жобаға талаптар қарастырылады және оның сапасы анықталады.



Сурет 2.2 – Каскадты модель



Сурет 2.3 – Аралық бақылаумен жүретін тарауланған модель



Сурет 2.4 – Спиральды модель

Тәжірибе жүзінде тіршілік циклының ең таратылған екі түрі қолданылады:

- Каскадты модель (мерзім 1970–1985 ж.);
- Спиральды модель (мерзім 1986 ж.).

2.4 АЖ – ның басты сенімділік түсінігі

АЖ–ның кез келген шығу эталоны тапсырыс берушінің немесе потенциалды қолданушының техникалық тапсырмасы болып табылады (немесе спецификациялық талап ету).

Мұндай құжаттарды қолданушының белгілі шартын және шығу берілгенін алу кезінде, мазмұнын және мәнінің нәтижесінің құрамын құру кезінде алуға болады. Кез келген АЖ – ның шарттарын көрсеткен функционалды нәтижені қайтаруға және соған керек формалды эталонды қателік сияқты квалификация жасауға болады.

Практика жүзінде тапсырыс беруші мен өндірушінің өндіріс процесінде ПЖ – нің шығу талаптары бойынша нақтыланады және детальданады. Базаның мұндай нақтылануы формальданусыз көрсеткіші және мамандар білімі болып табылады, сонымен қатар уақытаралық жоба этабының тексеру нәтижесі болады. Нормативтік – техникалық құжаттарды орнату, бір немесе бірнеше АЖ – ның ережелерімен берілген мінездемелерін орнату қорытындысын АЖ – ның техникалық операциясы тексереді.

Оның қасиетінің бейнеленуі және мүмкіндігінің анықталуы және оны тура белгілеуде эффектті қолдану, кез келген өнім көрсеткіш жиынының сапасын көрсетеді. Өнім күрделі болған сайын, оның қасиетінің мінездемелік саны және оның сапасының адекватты бейнелену көрсеткішінің қажеттігі көп болады. АЖ сапасы және күрделі бағдарламалық құрал жиынтығының көрсеткіші – критериилермен жазылады, олардың әрқайсысын метрикалық және әдіс өлшемін анықтау қажет. Бұл критериилер және метрикалық бағдарламаларының соңғы өнімнің жетістік тәсілінің қатысуынсыз қасиетін бейнелейді.

Анықтау белгілері, АЖ – ның басты функциялары, тапсырыс беруші немесе потенциалды қолданушының техникалық талап етілуі, атрибуттар жиыны түрінде анықталатын, оның функционалды қажеттілігі АЖ – ның сапасына байланысты болады.

Көптеген деңгейде функционалды қажеттілік АЖ сенімділігінен байқалады. Оған қоса, функционалды қажеттілікті көптеген әр түрлі критериилер бейнелейді. Оларды басты сенімділік көрсеткішіне байланысты бөлім критерииі немесе фактор ретінде қарастыруға болады.

АЖ – ның жұмысқа қабілеттілігі немесе олардың ірі компоненттерінен айырылу кедергілерін көрсеткіш сенімділігінде тіркеуге болады. Техникалық құжаттарды талап етуге орнатылған, берілген функция параметрін орындауға қабілетті объект жағдайын сенімділік теориясының жұмысқа қабілеттілігі деп аталады.

Тек уақыт аралығында байқалатын жүйені ішкі қасиеті сенімділік болып табылады.

Сапа критерииі динамикалық және мүлікті стохастикалық, АЖ – ның толық немесе ірі компоненттерінің функционалды мінездемелері бола алады. АЖ – ның интегралды сапа көрсеткіші тәжірибе жүзінде тура бағаланып және анықталып өлшенеді.

Жинап қою және тексеруде қолданылатын шығу берілгендерге АЖ–ның жұмысқа қабілеттілігіне толық сенім береді. Нағыз шығу берілгендерінің мәні берілген техникалық тапсырма және тексеруде қолданылатын бағдарламада басқаша болады. Осындайда шығу берілгендері функционалды бағдарламаларын алдын ала болжау қиын, өйткені олар әр түрлі аномалия, қарсы аяқталулар шақыруы мүмкін.

Нормативтік – техникалық және (немесе) конструкторлық құжаттарды жүйеде орнату, бір немесе бірнеше функцияларды бұзу ең болмаса сапа талабының біреуінде орындалу жағдайын – жауапсыз дейді. Нормативтік –техникалық және (немесе) конструкторлық құжаттама және берілген функцияларды орындау жауапсыздығын анықтауды рұқсат етуге орнатылған, белгі немесе жиын белгілері жауапсыз критериилердің бағалануын жағдайға әкеледі.

Бұзудың «жауапсыздықтан» айрықшалығы уақытша көрсеткіштің ұзақтығын қайта құрудан кейін АЖ – ның функционалды бұзуына қатысты болады, бұзудан кейін жауапсыздан кейінгіге қарағанда қайта құру уақыты аз болады.

Бұзулар кезінде және қайта құрылу қасиетіне тура қатысты АЖ–ның тұрақтылық мінездемелерінің функционалдыққа қарсы жауапсыздыққа қабілетті болады. Қайта құрылу – АЖ–ның функционалды толық қайта құрудан кейін қайта жіберумен (рестарт) мінездемеледі. Ресурстар және уақыт қажеттілігі АЖ–ның функционалды нормаларын жаңартуды қамтамасыз етуді қайта жіберуге болады, сондықтан функционалды қайта құрудың толықтығы және ұзақтығы АЖ–ның бейнелеу сапасынан кейінгі бұзылуды және белгілеуді оның тура мүмкіндікте қолдануға болады.

Басқа өндіріс өніміне қабылданған АЖ–ның сенімді көрсеткіші адекватты аналогиялық мінездеменің мәнділік деңгейі болады. Критерииді кең қолдануға

қарсы өндірудің ұзақтығы – жұмысқа қабілеттілік жүйе жағдайының уақытаралық жауапсызы немесе оның жақсы функционалды жүйеден кейінгі бастамасы болады. Критерии сенімінің әр түрлілігі мүмкіндік мінездемесінің бірнеше форма көлемінде қолданылады.

Мүмкіндікті қайта құрылу жүйесінен алу жұмысқа қабілеттілік жағдайының соңғы уақытында бейнелейтін, мінездеме жауапсызын қайта құру критерии коэффициентін дайындауға өндіріледі. Жауапсыз және қайта құру құрамы коэффициент мәнінің дайындығы үлкен интервалға жететін жұмыс жүйесінің пайдалысы уақыт уысына байланысты болады.

Интегралды бағдарламаның сапа көрсеткішінің жақсы өлшемін, қатар дәлдігін алуға күрделі АЖ–ның сапасын бағалауды рұқсат етуге сенімділік теориясының басты түсінігі қолданылады. Келтірілген критериилер басты ретте АЖ–ны тексеру және комплексті жинап қоюдың аяқталу фазасында қолданылады.

Сұраныста АЖ қолданушыларының шығу ақпаратында сенімді көрсетілімі барлық АЖ типінің интегралдық көрсеткішінің маңыздыларының бірі болып табылады, сонымен қатар бағарламалық–техникалық комплекстердің әр түрлі белгілері болады (P_{nan} – мүмкіндік сенім көрсетілімі).

Қазіргі заманғы, корпоративті АЖ өзінің комплекстілігін көрсетеді:

- бағарламалық–техникалық құрылғылар;
- ақпараттық қамтамасыз ету (берілгендер базасы);
- телекоммуникация құрылғылары;
- оның функционалды қолданылуы үшін өңдеу сұраныстарын және шығу ақпаратының көрсетілімін қамтамасыз етеді.

Автоматтандырылған сұраныс өңдеуімен және эксплуатациялық шартты қолдануды бақылау кезінде алгоритмдік шындалған шығу ақпаратының келісімін көрсетумен және АЖ–ны техникалық қызмет етуімен, АЖ құрылымын қабылдауды қамтамасыз етумен АЖ қолданушыларының шығу ақпаратының сұраныс көрсетілімінің сенімділігімен түсіндіріледі (P_{nan} – мүмкіндік сенім көрсетілімі).

АЖ қолданушысының шығу ақпаратының курстық жұмыста сұраныс көрсетілімінің сенімділігін бағалау келесі модельде қолданылады.

АЖ–ның кез келген уақытында (қолданушының көзқарасымен қарағанда) қатар екі жағдайдың біреуінде болады:

- жұмысқа қабілеттілік;
- жұмысқа қабілетсіздік.

АЖ–ның математикалық уақытты күтудің жұмысқа қабілеттілік жағдайы жауапсыздың орташа өңделіміне тең. АЖ–ның математикалық уақытты күтудің жұмысқа қабілетсіздік жағдайы жұмысқа қабілеттіліктің жауапсыздан кейінгі орташа қайта құру уақытына тең.

Кездейсоқ уақытта шығу ақпараты көрсетілім сұранысын орындайды (экран дисплейінде, файлды магнитті тасымалдаушыда немесе шығарғанда). 3 варианты мүмкін:

– АЖ–ның функционалды t уақытында түскен сұраныс, оның жұмысқа қабілеттілік жағдайы үстінен түседі және сұранысты өңдеу қажеттілігі және оған шығу ақпаратына байланысты берілуі АЖ–ның жағдайында барлық уақытында болады;

– АЖ–ның функционалды t уақытында түскен сұраныс, оның жұмысқа қабілеттілік жағдайы үстінен түседі, бірақ сұранысты өңдеуге қажетті және оған шығу ақпаратына қатысты берілуінде АЖ бұл жағдайда аз уақыт болады;

– АЖ–ның функционалды T уақытында түскен сұраныс, оның жұмысқа қабілетсіздік жағдайы үстінен түседі.

Бірінші жағдайда сенімді ақпараттық көрсетілім болады, ал екінші және үшінші жағдайларда сұратылған ақпарат көрсетілмейді.

Зерттеулер кезінде алынған бағалар:

- жауапсыздыққа орташа өңделу;
- орташа қайта құрылу уақыты;
- шығу ақпаратының орташа уақыт көрсетілімі (АЖ–ға сұраныс уақыты);
- АЖ қолданушыларының шығу ақпаратына сұрастырылатын сенімді көрсетілім есептеулерін өткізу үшін шығу берілгендері болады. Шығу ақпараты көрсетілуі мүмкін барлық сұрастырылған ақпараттар нақты анықталған формада (немесе мазмұнда) типтерге бөлінеді.

2.5 Деректер қорын жобалау. Деректер моделі

Деректер қорын басқару жүйесі (ДҚБЖ) қолдайтын деректердің моделі өзінің логикалық құрылымы болады. Оған келесідей модельдер жатады:

- иерархиялық;
- желілік;
- реляционды.

Сонымен қатар кейінгі уақытта тәжірие жүзінде бастапқы үш модельдерден гөрі белсенді деректер моделі қолданылып жүр. Олар:

- постреляционды;
- көп өлшемді;
- нысанды–бағытталған.

Осы модельдерге негізделе отырып, қазіргі кезде басқа да деректер моделі құрылуда. Олардың қатарына нысанды–реляционды, деуктивті–нысанды–бағытталған, семантикалық, концептуалдық және бағытталған моделдерді жатады.

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

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

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

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

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

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

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 карағанда 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.

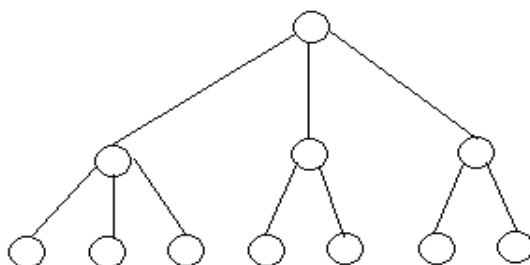
Жай деректер өздігінен компьютерлік формада егер оларға қолжетерлік құрылғылар болмаса, қолданушыға қызығушылық тудырмайды. Деректерге қолжеткізу дерекқорына сұраныс түрінде орындалады. Олар стандартты сұраныс тілдерінде жазылады. Бүгінде көп ДҚБЖ үшін сондай тіл болып SQL саналады.

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

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

Кейбір ДҚБЖ бірнеше деректер моделін қамти алады. Мысалы, ИНТЕРБАЗА жүйесінде желілік деректер қорын басқарылады және пайдаланушы интерфейсінде SQL және QBE тілдері құрылған.

Иерархиялық модельде деректер арасындағы байланыстар графа (ағаш тәріздес) түрінде болады. Иерархиялық модельдің сызбалық көрінісі келесідей болады (Сурет 2.5).

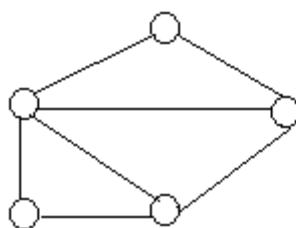


Сурет 2.5 – Иерархиялық модельдегі байланыстар көрінісі

Иерархиялық деректер қорын құру кезде арнайы бағдарламалау тіліндегі ағаш типі қолданылады. Ағаш типі бірнеше подтиптерден құралған болып келеді. Әр ағаштың өзіне сәйкес «тамырлары» (түбірі) болады. Сол тамырдан ббағынатын көптеген құрылымдық ағаш тәріздес типтер де болады.

Иерархиялық модельдің бір артықшылығы, ол ЭЕМ жадысын тиімді қолдануы және деректерге қолданатын операциялардың жылдам орындалуымен сипатталады.

Желілік модель. Желілік модель деректер элементтерінің өзара әр түрлі байланысын көрсетеді. Желілік модельдің сызбалық кескіні келесідей болады (Сурет 2.6).



Сурет 2.6 – Желілік модельдегі байланыстар

Желілік модельдің артықшылығы – иерархиялық модельге қарағанда байланыстардың әр түрлі болып келуі болып табылады.

Реляционды модель. Реляционды модель IBM фирмасының Эдгаром Коддом қызметкерімен ұсынылған болатын. Модельдің негізгі ұғымы қатынас (relation) болып табылады.

Қатынас, ол элементтердің жиынынан (кортеж) тұрады.

Кесте жолдар (жазбалар) мен бағандардан тұрады. Кесте жолдарына кортеж сәйкес келсе, бағанға қатынас атрибуттары сәйкес келеді.

Артықшылығы оның қарапайымдылығында. Осы қарапайымдылығының арқасында пайдаланушы оны кең қолдана алады. Сонымен қатар деректерді өңдеу жеңіл әрі тез орындалады.

Реляционды моделдің мысалы ретінде келесілерді қарастыруға болады: dBaseIII Plus және dBase IV (Ashton–Tate фирмасы), DB2 (IBM), R:BASE (Microrim), FoxPro және FoxBase (Fox Software), Windows (Borland) үшін Paradox және dBASE for, Visual FoxPro және Access (Microsoft), Clarion (Clarion Software), Ingres (ASK Computer Systems), Oracle (Oracle) және Microsoft SQL Server, сонымен қатар ресейлік реляционды ДҚБЖ ПАЛЬМА (ИК АН ҮССР) және HyTech (МИФИ).

Реляционды ДҚБЖ нысанды–бағытталған жүйелердің кейбір қасиеттерін камтиды. Ондай ДҚБЖ көбіне нысанды–реляционды деп атайды. Оның мысалы ретінде Oracle 8.x. алуға болды. Ал, оның алдындағы нұсқасы, яғни Oracle 7.x таза реляционды болып табылады. Microsoft SQL Server 2005, PostgreSQL, сол сияқты Sav Zigzag, IBM Cloudscape, FirstSQL бұлардың бұлардың бәрі нысанды–реляционды болып табылады.

Постреляционды модель. Постреляционды модель реляционды модельдің кеңейтілген түрі болып есептеледі. Ол кестеде сақталған деректерді бөлуге рұқсат етеді және өріс көп мәндік бола алады, яғни бір мәннің ішінде бірнеше мәндер болу мүмкін. Осылардан құралған кестелер негізгі кестеге бағынады.

Артықшылығы, реляционды кестелерді жиынын бір постреляционды кестемен беруге болады. Сонымен қатар, тағы артықшылықтары оның ақпараттық жоғары дәрежеде көрсетілуі мен оны өңдеудің тиімділігінде.

Постреляционды модельдің мысалы ретінде ДҚБЖ uniVers айтсақ болады.

Көп өлшемді модель. Көп өлшемді модель, реляционды модельмен бірге пайда болды. Бұрын ол реляционды модель сияқты кеңінен қолданылмаған, тек 90 жылдардың ортасынан бастап оған қызығушылық артуда. Көп өлшемді модель ақпаратты интерактивті өңдеуге арналған.

Оның артықшылығы, реляционды модельге қарағанда нысандардың өзара күрделі байланыстарын қарастырады.

Көп өлшемді модельдің мысалы ретінде мына жүйелерді қарастыруға болады: Essbase (Arbor Software), Media Multi–matrix (Speedware), Oracle Express Server (Oracle) және Cache (InterSystems).

Нысанды–бағытталған модель. Нысанды–бағытталған модельде деректер өздерінен бөлек жазбаларын идентифициялана алады және де оларды өңдейтін функцияларды анықтай алады. Ал кемшілігі, сұраныстардың жай орындалуында.

90 жылдарды нысанда–бағытталған модельдер тәжірибе жүзінде іске асырылды. Қазіргі кезде олар кеңінен қолданыста. Атап айтатын болсақ: ДҚБЖ: POET (POET Software), Jasmine (Computer Associates), Versant (Versant Technologies), O2 (Ardent Software), ODB–Jupiter («Интелтек Плюс» ғылыми–өндіріс орталығы), Iris, Orion және Postgres (Кесте 2.1).

Кесте 2.1 – SQL тілінің негізгі операторлары

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

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

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

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

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

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

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

Егер осы амалды орындау барысында жаңылысу болса, мысалы, бірінші операция орындалып болғаннан кейін, онда ақша жоғалады. Сәйкесінше, кез келген

дерекқормен байланысты амал тұтастай жасалуы керек немесе мүлдем жасалмау керек. Мұндай амал транзакция деген атқа ие болды.

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

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

2.6 Реляционды деректер қорына қойылатын талаптар

Біз жұмыста реляциялық деректер қорын негізге алатын болғандықтан оған қойылатын талап – деректердің қалыптастырылуы. Қалыптастыру кестелердегі деректердің қайшылықтары және үйлесімсіз болмауын қадағалайды. Қалыптастырудың алты формасы бар, бірақта тәжірибеде деректер алдыңғы үш қалыптың талаптарын қанағаттандыратын болса жеткілікті.

Деректер базасының теориясы қалыптастыру принципіне негізделінеді. Ал, қалыптастыру ұғымы функционалдық тәуелділік ұғымына негізделінеді.

– R – қатынасының A атрибуттарының әрбір мәніне B атрибутының тек бір ғана мәні сәйкес болса, онда $B \rightarrow A$. B атрибуты A атрибутына функционалды тәуелді;

– егер $A \rightarrow B$ -ға, $B \rightarrow A$ тәуелді болса, онда A және B атрибуттары өзара функционалды өзара тәуелді;

– егер кілт емес атрибуттар құрама кілттің бір бөлігіне ғана тәуелді болса, онда ол кілтке жартылай тәуелді деп аталады;

– егер R қатынасының кілт емес B атрибуты. A атрибуттар тобына, яғни құрама кілттің барлық атрибутына тәуелді болса және ешбір жартылай функционалдық тәуелділік болмаса, онда B атрибуты A атрибуттар тобына толық тәуелді;

– егер R қатынасында $A \rightarrow B$ -ға, $B \rightarrow C$ -ға тәуелді болса, онда C атрибуты A атрибутына транзитивті тәуелді.

2.7 SQL тілі туралы негізгі деректер

Деректерге ену мүмкіндігінің реляциялық тәсілі жазбалар тобының амалдарына негізделген. Амалдарды орындауға SQL (Structured Query Language) құрылымдық сұраныс тілінің құралдары қолданылады. ASP.NET қосымшаларында BDE механизмін қодану барысында деректер жиыны ретінде SQL-сұранысты орындауды беретін Query немесе StoredProc компоненттері қолданылады. SQL құралдарын жергілікті және қашықтағы деректер қорымен амалдар орындауға қолдануға болады.

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

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

SQL бағдарламалау тілінің толық функционалдық мүмкіндіктерін қамтымағанмен, ол деректерге ену мүмкіндігіне арналған және оны бағдарламаны жасау құралына қосады. және ASP.NET жүйесіне ендірілген. Сонымен қатар, SQL командаларымен жұмыс істеу үшін сәйкес құралдар және компоненттер бар. ASP.NET-де мұндай компоненттерге Query, SQLQuery және ADOQuery деректер жиыны жатады.

SQL тілінің функциялары.

Статистикалық функциялар:

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

Жолдармен жұмыс істеуге арналған функциялар:

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

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

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

Кестелермен келесі амалдарды орындауға болады:

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

Бұл әрекеттер SQL тілінің инструкцияларының көмегімен орындалады.

Дерекқорында тек физикалық объекттер көрсетілмейді. Ол абстракция, процесстер, құбылыстар – яғни, адам тіршілігінде кездесетін барлық заттар туралы мәліметтерді сақтауға мүмкіндігі бар. Мысалы, дерекқорында складқа құралдарды

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

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

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

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

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

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

2.8 Кесте құру және өшіру

Кестені құру үшін CREATE TABLE инструкциясы қолданылады:

```
CREATE TABLE <Кесте аты>  
(<Өріс аты> <дерек типі>
```

. . .
<өріс аты> <дерек типі>;

Файлдың кеңейтілуі бойынша кестенің форматы автоматты түрде, яғни .db – Paradox кестесі үшін, .dbf dBase кестесі үшін анықталады. Кесте файлы дерек қор псевдонимі көрсетілген, МҚ каталогында орналастыру керек.

SQL тілінің деректер типі және сәйкес Paradox–тың деректер типтері келесі кестеде көрсетілген (Кесте 2.1).

К е с т е 2.1 – SQL тілінің деректер типі

SQL	Paradox
SMALLINT	Short
INTEGER	Long Integer
DECIMAL	BCD
NUMERIC(x,y)	Number
FLOAT(x,y)	Float(x,y)
CHARACTER(n)	Alpha
VARCHAR(n)	Alpha
DATE	Date
BOOLEAN	Logical
BLOB(n,1)	Memo
BLOB(n,2)	Binary
BLOB(n,3)	Formatted memo
BLOB(n,4)	OLE
BLOB(n,5)	Graphic
TIME	Time
TIMESTAMP	Timestamp
MONEY	Money
AUTOINC	Autoincrement
BYTES(n)	Bytes

SQL тілінің көмегімен кесте құру мысалы:

```
CREATE TABLE Student.dbf  
    (Numer Integer  
    Name CHAR(15))
```

Paradox кестесі үшін PRIMARY KEY сипаттамасын және жақшаның ішіне оның өрістерін көрсетіп, негізгі немесе алғашқы кілтті анықтауға болады. Кілттік өрістер тізімінде бірінші болып орналасуы керек. Негізгі кілтті құрумен жаңа кестеге құруға мысал:

```
CREATE TABLE Personal.db  
    (Code AUTOINC)
```

```
Name CHAR(15)
Post CHAR(15)
Oklad Numeric(10,2)
PRIMARY KEY(Code));
```

Кестені өшіру үшін келесі инструкция қолданылады `DROP TABLE <Кесте аты>`.

Мысалы `DROP TABLE <student>`. Дискіден `student` атты кестеге қатысы бар барлық файлдар өшіріледі.

Кесте өрістерінің құрамын өзгерту.

Кесте өрістерінің құрамын өзгерту өрістерді қосу немесе өшіруді білдіреді және оның құрылымын өзгертуге әкеледі, сонымен қатар кестені басқа қосымшалар қолданбауы керек. Кесте өрістерінің құрамын өзгерту `ALTER TABLE` инструкциясының көмегімен орындалады:

```
ALTER TABLE <кесте аты>
    ADD <1-өріс аты> <1-өріс типі>,
    DROP <1-өріс аты>,
    . . .
    ADD <n-өріс аты> <n-өріс типі >,
    DROP <n-өріс аты>;
```

`ADD` операндысы кестеге жаңа өріс қосады, өріс аты және типі `CREATE TABLE` инструкциясындағыдай беріледі, ал `DROP` операндысы көрсетілген атымен кестеден өрісті өшіреді. `ADD` және `DROP` операндылары бір-бірімен байланысты емес және еркін түрде орналасуы мүмкін.

3 Дерекқорды жобалау және оны іске асыру

3.1 ДҚ құрудың бастапқы кезеңі

Жүктемені бөлу жұмысының жалпы сипаттамасы.

STYS – 2 КТ кафедрасының мұғалімдерінің жүктемесін бөлу жұмысының негізін құрайды. Жүктемелерді бөліп, бөлінбеген жүктемелерді көріп, оларға өзгеріс енгізе алады.

Мәселелер және шектеулер:

- құпия сөзсіз кестені өңдеуге және өзгеріс енгізуге мүмкіндіктің болмауы;
- жүктеменің дұрыс бөлінуі;
- есеп беру жүйелі түрде болуы.

Дерекқордағы шешілетін мәселелер.

Мұғалімдердің жүктемесін бөлетін орталықтандырылған деректер қорларын тиімді басқаратын жүйесін құру, бірінші семестр, екінші семестр және жылдық жоспар бойынша сағаттардың қосындысын есептеу, бөлінген сағаттардың санын есептеу, бөлінбеген сағаттарды есептеу, пәндер бойынша, мұғалімдер бойынша, семестр бойынша, оқу формасы бойынша, топ бойынша сұрыптау мүмкіндігін қарастыру.

Дерекқордың функционалдығын сипаттау. Жобаланатын дерекқор және бағдарламалық интерфейс кафедра мұғалімдерінің жүктемесін бөлуге арналған. Бағдарламалық қосымшада келесі функциялар іске асырылған:

- ақпараттық функция, жүктеме, бөлінген және бөлінбеген жүктемелер туралы толық ақпаратты береді;
- енгізу функциясы, дерекқорға жаңа ақпарат енгізуге мүмкіндік береді.

3.2 Концептуалды жобалау. Қажетті ақпарат пайдаланушылардың талабы

Жоба пайдаланушының барлық негізгі талаптарын қанағаттандыру керек. Бұл жүйенің негізгі пайдаланушылары университет мұғалімдері мен студенттері болады. Негізгі талаптар алынған ақпараттың ұсынылатын тиімділік деңгейінде негізделеді. Жүйеге қойылатын негізгі талаптардың тізімі:

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

3.3 Дерекқор және қосымшаны жобалау құралдарын тандаудың негіздемесі

Дерекқорларды басқару жүйесі.

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

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

- орнату және қолданудың жеңілдігі;
- кең таралған;
- қымбат серверлік қуатты орнатуды қажет етпейді;
- өндіргіштігінің жақсы көрсеткіші;
- деректерді сақтау сенімдігі мен қауіпсіздігін қамтамасыз етудің жеке құралдары;

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

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

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

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

SQL компьютерлік дерекқорда сақталатын деректерді өңдеу және оқу үшін арналған құрал болып табылады, және қолданушының дерекқормен өзара қатынас ұйымдастыру үшін қолданылатын программалау тілі болып табылады. Шындығында

SQL тек бір белгілі типті – реляционды дерекқормен жұмыс істейді. SQL ДҚБЖ қолданушыға ұсынатын барлық функционалдық мүмкіншіліктерді жүзеге асыру үшін қолданылады, соның ішінде:

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

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

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

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

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

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

SQL – дерекқорды программалау тілі. Дерекқорға қатынас құру үшін бағдарлаушылар өздерінің бағдарламаларына SQL командаларын қояды. Бұл әдістеме қолданушылар жазған бағдарламаларда да, дерекқорлардың қызметтік бағдарламаларында да қолданылады (есеп беру генераторы және деректерді енгізу құралдарында). Осылайша, SQL адамдарға, бағдарламаларға және есептеуіш жүйелерге реляционды дерекқорлардағы ақпараттарға қатынас құруды қамтамасыз ететін қажетті және мықты құралға айналды.

3.4 ER моделін құру

ДҚ құрудың бастапқы кезеңінде алғашқы нысандар жиынтығын анықтау қажет. Бұл ақырғы пайдаланушылар мен жобалаушы көзқарасы бойынша жүйе объектілері туралы ең маңызды ақпарат болып табылады. Кафедра дерекқоры (Кесте 3.1) көрсетілген нысандарды қолданады.

К е с т е 3.1 – «КТ кафедра АЖО» дерекқорының нысандары

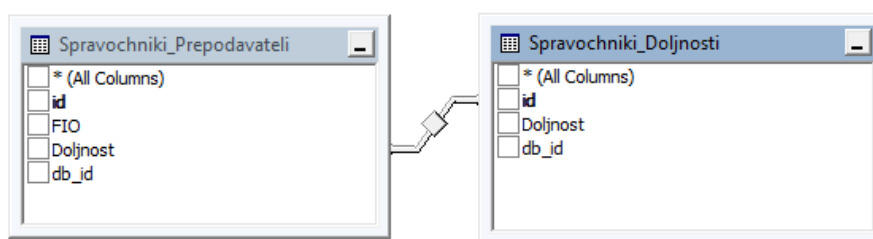
Нысан аты	Сипаттамасы
Polzovatel	Пайдаланушының логині, құпия сөзі сақталады
Spravochniki_Dissiplini	Пәндер
Spravochniki_Gruppi	Топтар жайлы мәліметтер

Spravochniki_Forma_ovuchenia	Оқу формасы
Spravochniki_Prepodavатели	Мұғалімдер
Raspredelenie_Nagruzki	Бөлінген жүктемелер
Spravochniki_Doljnosti	Оқытушылардың лауазымдары
Dissiplini	Бөлінбеген жүктемелер
Name_db	Қолданушының кестелері

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

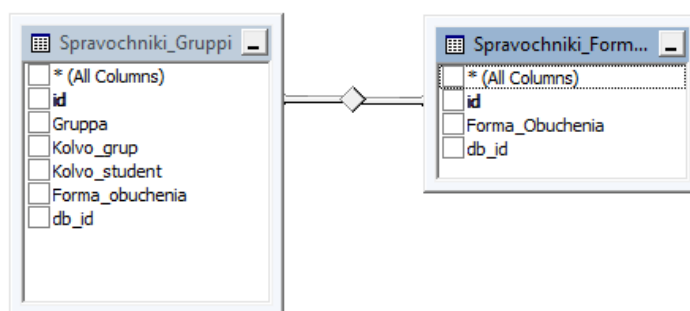
3.5 Бизнес ережелер

Әрбір оқытушының тиесілі лауазымы болады (Сурет 3.1).



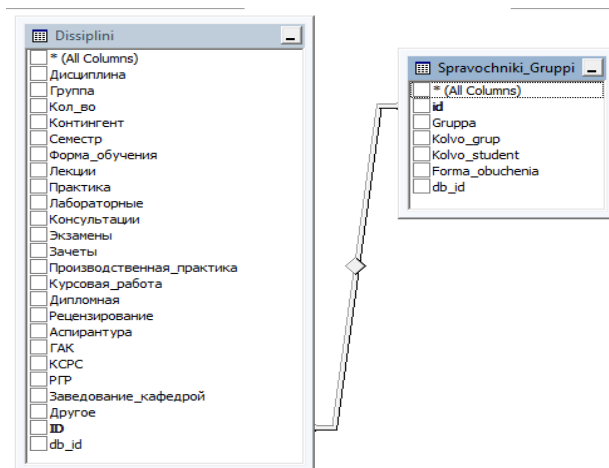
Сурет 3.1 – 1 – бизнес ереже үшін ER сегменті

Әрбір топқа бір ғана оқу формасы тиесілі (Сурет 3.2).



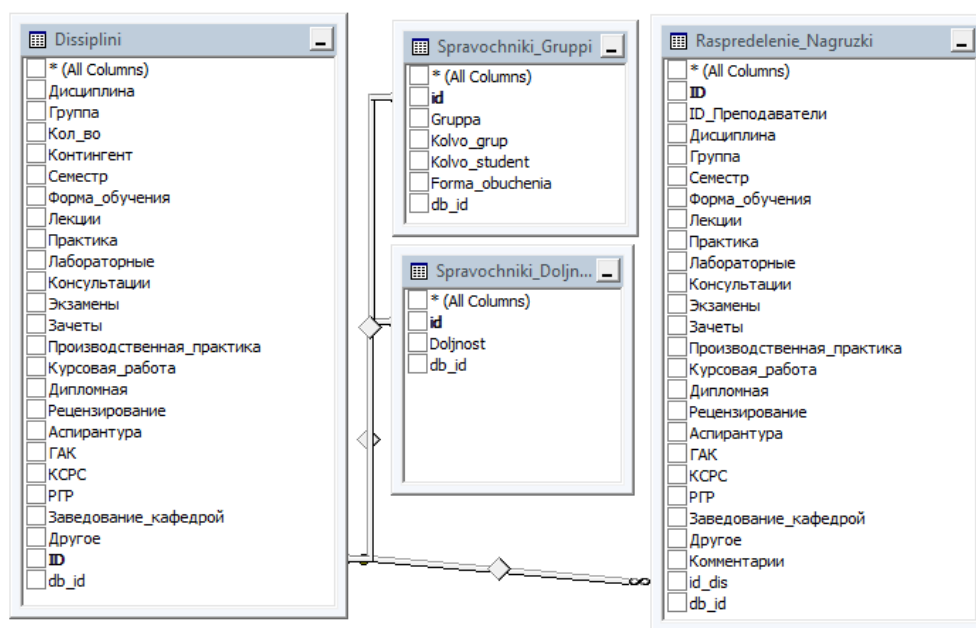
Сурет 3.2 – 2-бизнес ереже үшін ER сегменті

Бір пәнді бірнеше топтар оқиды, сәйкесінше, бір топ бірнеше пәндерді оқиды (Сурет 3.3).



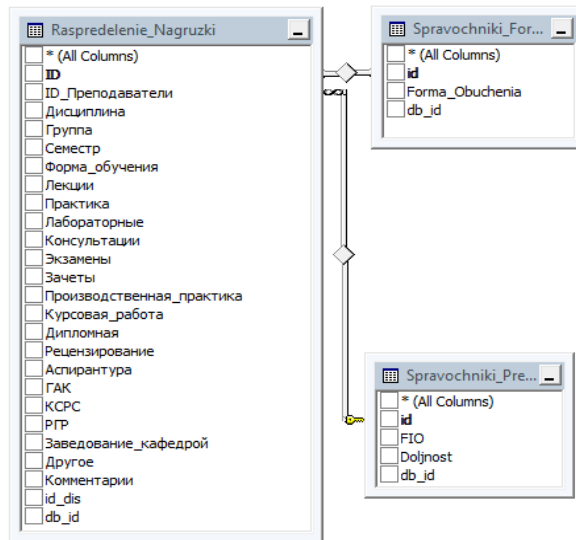
Сурет 3.3 – 3 – бизнес ереже үшін ER сегменті

Топ, жүктеме, бөлінген жүктеме, пән және пән түрі таблицалары бір бірімен байланысқан. Мағынасын ашып қарайтын болсақ, әрбір пәнді қандай топ оқиды, қандай топқа қандай пәннің бөлінуі анықталады. Ең бастысы бұл таблицада ақпараттар қайталанбауы тиіс (Сурет 3.4).

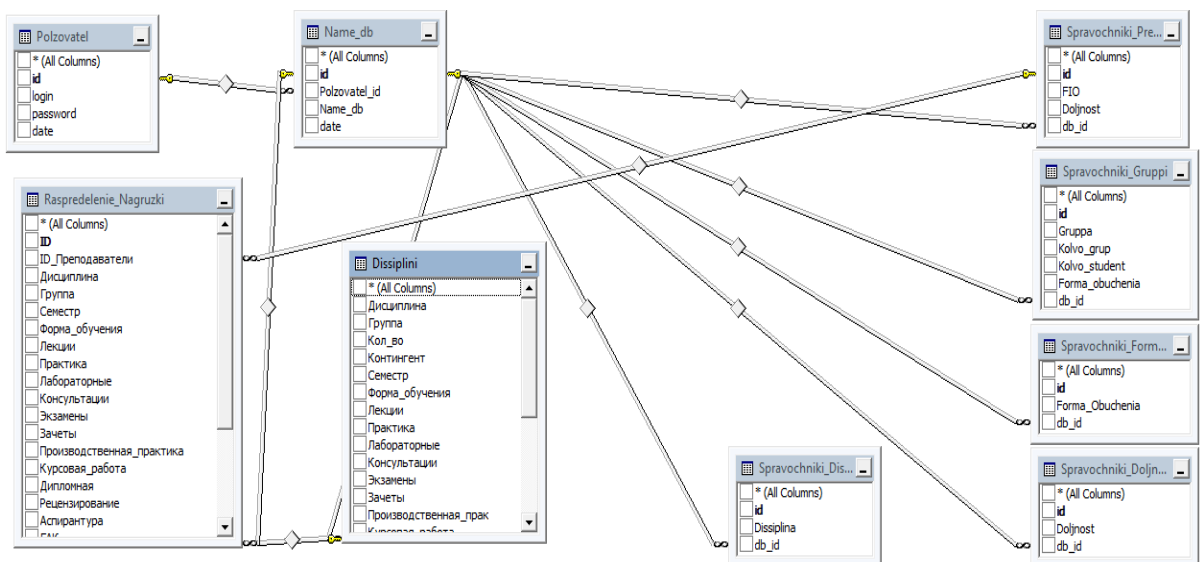


Сурет 3.4 – 4-бизнес ереже үшін ER сегменті

Мұғалімдер қандай пәнді оқытатыны жәнесол мұғалімнің сәкес лауазымы болады (Сурет 3.5).



Сурет 3.5– 5 – бизнес ереже үшін ER сегменті



Сурет 3.6 – ER диаграмма

3.6 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 (Кесте 3.2-3.8) 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).

Кесте 3.2 – Spravochniki_Prepodavately қатынасы

Өріс аты	Атрибут аты	Мәліметтер типі	Ескерту
Мұғалім коды	id	int	Алғашқы кілт
Мұғалім аты-жөні	FIO	varchar(50)	Қажетті өріс
Лауазым коды	Doljnost	varchar(50)	Қажетті өріс
ДҚ коды	db_id	int	Сыртқы кілт

Кесте 3.3 – Spravochniki_Doljnosti қатынасы

Өріс аты	Атрибут аты	Мәліметтер типі	Ескерту
Лауазым коды	id	int	Алғашқы кілт
Лауазым аты	Doljnost	varchar(30)	Қажетті өріс
ДҚ коды	db_id	int	Сыртқы кілт

Кесте 3.4 – Spravochniki_Dissiplini қатынасы

Өріс аты	Атрибут аты	Мәліметтер типі	Ескерту
Пәндер коды	id	Int	Алғашқы кілт
Пән аталуы	Dissiplina	varchar(50)	Қажетті өріс
ДҚ коды	db_id	id	Сыртқы кілт

Кесте 3.5 – Polzovatel қатынасы

Өріс аты	Атрибут аты	Мәліметтер типі	Ескерту
Семестр коды	id	int	Алғашқы кілт
Логин аты	login	varchar(20)	Қажетті өріс
Құпия сөз	password	varchar(20)	Қажетті өріс
Ұақыты	date	datetime	Қажетті өріс

К е с т е 3.6 – Dissiplini қатынасы

Өріс аты	Атрибут аты	Мәліметтер	Ескерту
Пән аты	Дисциплина	varchar(50)	Қажетті өріс
Группа аты	Группа	varchar(15)	Қажетті өріс
Группа саны	Кол_во	int	Қажетті өріс
Контингент	Контингент	int	Қажетті өріс
Семестр	Семестр	int	Қажетті өріс
Оқу түрі	Форма_обучения	varchar(10)	Қажетті өріс
Дәріс	Лекции	int	Қажетті өріс
Тәжірибе	Практика	int	Қажетті өріс
Зертхана	Лабораторные	int	Қажетті өріс
Консультации	Консультации	int	Қажетті өріс
Сынақ	Экзамены	int	Қажетті өріс
Зачет	Зачеты	int	Қажетті өріс
Өндірістік прак.	Производственная_практика	int	Қажетті өріс
Курстық жұмыс	Курсовая_работа	int	Қажетті өріс
Дипломдық	Дипломная	int	Қажетті өріс
Рецензиялау	Рецензирование	int	Қажетті өріс
Аспирантура	Аспирантура	int	Қажетті өріс
ГАК	ГАК	int	Қажетті өріс
СӨЖ	КСРС	int	Қажетті өріс
ЕГЖ	РГР	int	Қажетті өріс
Кафедра	Заведование_кафедрой	int	Қажетті өріс
Басқа	Другое	int	Қажетті өріс
Форма коды	ID	int	Алғашқы кілт
ДҚ коды	db_id	int	Сыртқы кілт

К е с т е 3.7 – Spravochniki_Gruppi қатынасы

Өріс аты	Атрибут аты	Мәліметтер типі	Ескерту
Топ коды	id	int	Алғашқы кілт
Топ аты	Gruppa	varchar(20)	Қажетті өріс

Топтар саны	Kolvo_grup	int	Қажетті өріс
Студенттер саны	Kolvo_student	int	Қажетті өріс
Оқу формасы	Forma_obuchenia	int	Сыртқы кілт
ДҚ коды	db_id	id	Сыртқы кілт

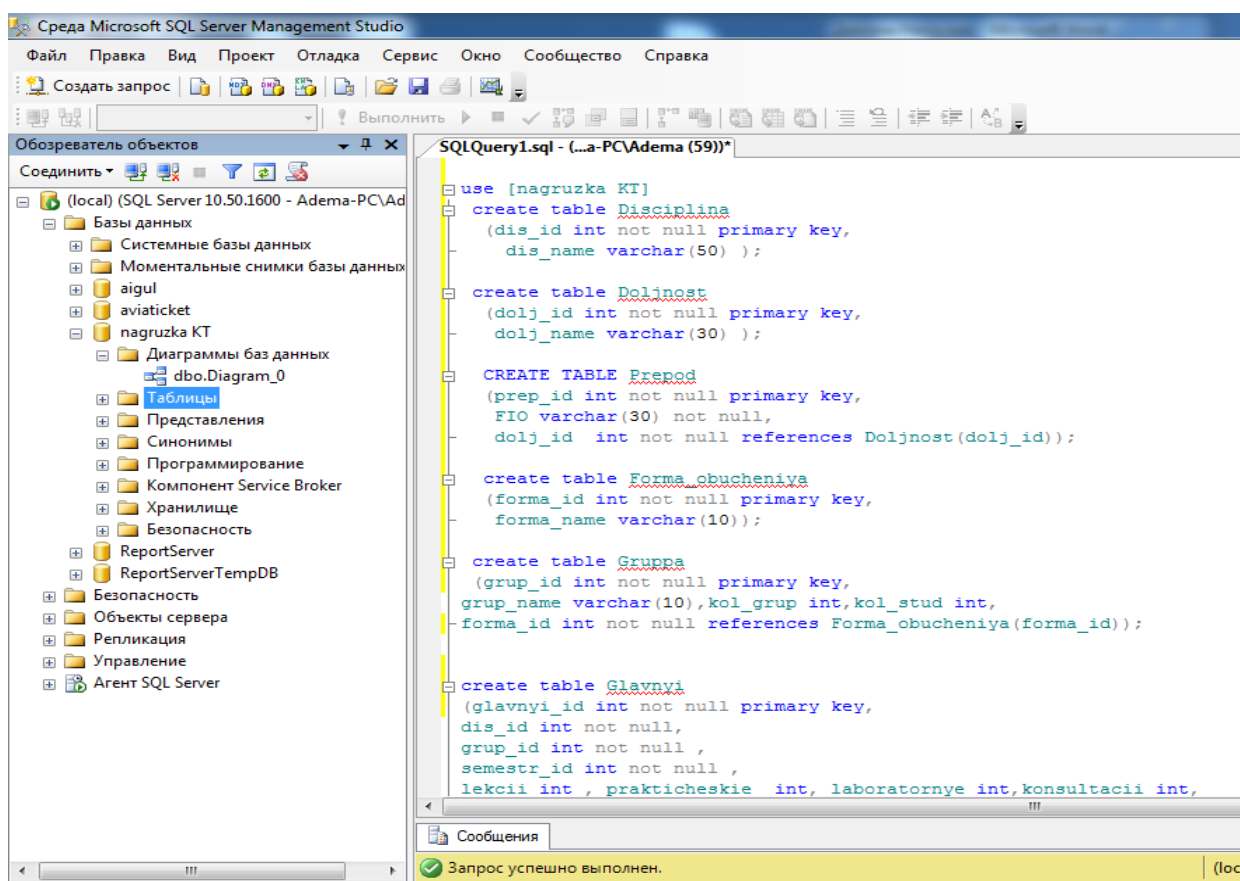
К е с т е 3.8 – Raspredelenie қатынасы

Өріс аты	Атрибут аты	Мәліметтер типі	Ескерту
Пән аты	Дисциплина	varchar(50)	Қажетті өріс
Группа аты	Группа	varchar(15)	Қажетті өріс
Группа саны	Кол_во	int	Қажетті өріс
Контингент	Контингент	int	Қажетті өріс
Семестр	Семестр	int	Қажетті өріс
Оқу түрі	Форма_обучения	varchar(10)	Қажетті өріс
Дәріс	Лекции	int	Қажетті өріс
Тәжірибе	Практика	int	Қажетті өріс
Зертхана	Лабораторные	int	Қажетті өріс
Консультации	Консультации	int	Қажетті өріс
Сынақ	Экзамены	int	Қажетті өріс
Зачет	Зачеты	int	Қажетті өріс
Өндірістік прак.	Производственная_практика	int	Қажетті өріс
Курстық жұмыс	Курсовая_работа	int	Қажетті өріс
Дипломдық	Дипломная	int	Қажетті өріс
Рецензиялау	Рецензирование	int	Қажетті өріс
Аспирантура	Аспирантура	int	Қажетті өріс
ГАК	ГАК	int	Қажетті өріс
СӨЖ	КСРС	int	Қажетті өріс
ЕГЖ	РГР	int	Қажетті өріс
Кафедра меңгеру	Заведование_кафедрой	int	Қажетті өріс
Басқа	Другое	int	Қажетті өріс
Форма коды	ID	int	Алғашқы кілт
ДҚ коды	db_id	int	Сыртқы кілт
Мұғалім коды	ID_Преподаватели	int	Сыртқы кілт
Бөлінбеген	id_dis	int	Сыртқы кілт

3.7 Логикалық жобалау

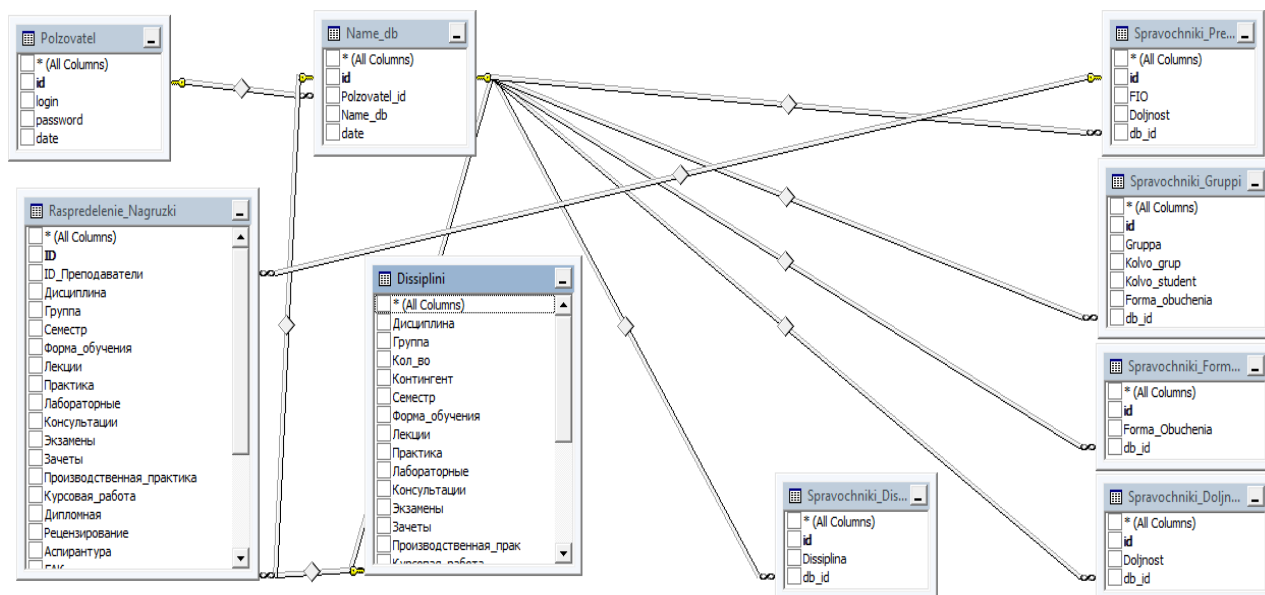
Логикалық жобалау кезеңінде дерекқор моделі жүйені іске асыру кезінде қолданылатын белгілі бір ДҚБЖ (MS SQL Server) үшін жарамды форматқа трансформаланады. Логикалық жобалаудың мақсаты реляционды кестелік құрылымдарды құру болып табылады. Жобаланатын дерекқорда кестелердің және байланыстардың құрылымдарын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

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 скрипті Query Analyzer – де (MS SQL Server) қосу. Кестелерді құруға арналған скрипт А қосымшасында келтірілген (Сурет 3.7).



Сурет 3.7 – Query Analyzer-де кестелерді құру

ДҚ диаграммасы. Концептуалды моделді дерекқордың сәйкес құрылымына енгізілу тәсілі қандай болса да, ДҚ реляционды сұлбасы жобаға сәйкес болу керек (Сурет 3.8).



Сурет 3.8 – ДҚ диаграммасы

3.8 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 (Сурет 3.9).

Сұрыптама жасау процедурасы. Бұл есептуді, төменде келтірілген процедура көмегімен оңай іске асыруға болады (Сурет 3.10).

```

create procedure [dbo].[procedure_Dissiplini]
as
begin
UPDATE Dissiplini SET [Лекции] = NULL WHERE [Лекции]=0
UPDATE Dissiplini SET [Практика]= NULL WHERE [Практика]=0
UPDATE Dissiplini SET [Лабораторные]= NULL WHERE [Лабораторные]=0
UPDATE Dissiplini SET [Консультации]= NULL WHERE [Консультации]=0
UPDATE Dissiplini SET [Экзамены]= NULL WHERE [Экзамены]=0
UPDATE Dissiplini SET [Зачеты]= NULL WHERE [Зачеты]=0
UPDATE Dissiplini SET [Производственная_практика]= NULL WHERE [Производственная_практика]=0
UPDATE Dissiplini SET [Курсовая_работа]= NULL WHERE [Курсовая_работа]=0
UPDATE Dissiplini SET [Дипломная]= NULL WHERE [Дипломная]=0
UPDATE Dissiplini SET [Рецензирование]= NULL WHERE [Рецензирование]=0
UPDATE Dissiplini SET [Аспирантура]= NULL WHERE [Аспирантура]=0
UPDATE Dissiplini SET [ГЭК]= NULL WHERE [ГЭК]=0
UPDATE Dissiplini SET [КПС]= NULL WHERE [КПС]=0
UPDATE Dissiplini SET [РГР]= NULL WHERE [РГР]=0
UPDATE Dissiplini SET [Заведование_кафедрой]= NULL WHERE [Заведование_кафедрой]=0
UPDATE Dissiplini SET [Другое]= NULL WHERE [Другое]=0
end;

```

Сурет 3.9 – Сабактар процедурасы


```

create procedure [dbo].[procedure_Raspredelenie_Nagruzki]
as
begin
UPDATE Raspredelenie_Nagruzki SET [Лекции] = NULL WHERE [Лекции]=0
UPDATE Raspredelenie_Nagruzki SET [Практика]= NULL WHERE [Практика]=0
UPDATE Raspredelenie_Nagruzki SET [Лабораторные]= NULL WHERE [Лабораторные]=0
UPDATE Raspredelenie_Nagruzki SET [Консультации]= NULL WHERE [Консультации]=0
UPDATE Raspredelenie_Nagruzki SET [Экзамены]= NULL WHERE [Экзамены]=0
UPDATE Raspredelenie_Nagruzki SET [Зачеты]= NULL WHERE [Зачеты]=0
UPDATE Raspredelenie_Nagruzki SET [Производственная практика]= NULL WHERE [Производственная практика]=0
UPDATE Raspredelenie_Nagruzki SET [Курсовая работа]= NULL WHERE [Курсовая работа]=0
UPDATE Raspredelenie_Nagruzki SET [Дипломная]= NULL WHERE [Дипломная]=0
UPDATE Raspredelenie_Nagruzki SET [Рецензирование]= NULL WHERE [Рецензирование]=0
UPDATE Raspredelenie_Nagruzki SET [Аспирантура]= NULL WHERE [Аспирантура]=0
UPDATE Raspredelenie_Nagruzki SET [ГЭК]= NULL WHERE [ГЭК]=0
UPDATE Raspredelenie_Nagruzki SET [КСПС]= NULL WHERE [КСПС]=0
UPDATE Raspredelenie_Nagruzki SET [РГР]= NULL WHERE [РГР]=0
UPDATE Raspredelenie_Nagruzki SET [Заведование кафедрой]= NULL WHERE [Заведование кафедрой]=0
UPDATE Raspredelenie_Nagruzki SET [Другое]= NULL WHERE [Другое]=0
end;

```

Сурет 3.10 – Сабактарды бөлу процедурасы

3.9 Физикалық жобалау. Дерек қорына қосымша құру

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

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

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

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 бағыттап отырып, клиент ретінде де болуы мүмкін.

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

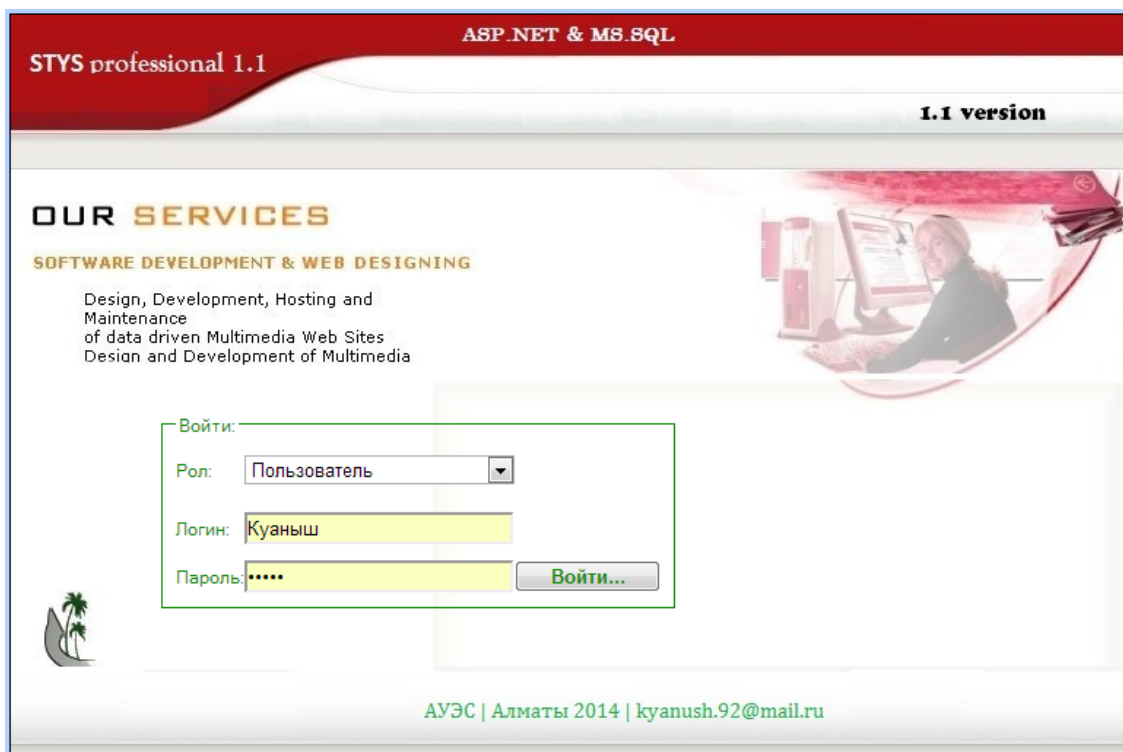
Алғашында ДҚЖБ орталықтанған архитектурасы болды. Онда ДҚБЖ өзі және дерекқормен жұмыс істеген қолданбалы бағдарламалар орталық компьютерде (үлкен ЭЕМ немесе мини – компьютер) функцияланды. Сол жерде дерекқор да орналасты. Орталық компьютерге қолданушылардың жұмыс орны ретінде болған терминалдар да қосылды.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

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

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

3.10 Бағдарламаны сипаттау

Бағдарламаның бөлімдері 4 бөліктен тұрады. 1 – құпия сөз және қолданушы атын енгізу бөлігі, 2- жаңа құжат жасау және өзгерту, 3 – жұмыс істеу бөлігі, 4 – администратор бөлігі (Сурет 3.11).



Сурет 3.11 – Программаның алғашқы терезесі. Құпия сөз және қолданушы атын енгізу бөлігі

Алғашқы терезеде біз екі қолданушы жұмыс істеу алады. Мұғалім және администратор. Төмендегі терезеде администратордың жұмыс істеу терезесі (Сурет 3.13). Ол мұнда жаңа қолданушы енгізеді және өзгерте алады. Үш бөлімнен тұрады Логин, құпия сөз, құжат жасалған уақыты (Сурет 3.12).

Добро пожаловать - а!

Назад...

Логин пользователя	Пароль пользователя	Дата создания пользователя	Edit X
Куаныш	12345	27.12.2013	Edit X
Ербол	12345	26.12.2013	Edit X
			Add

Сурет 3.12 – Администратор құжат құру және өзгерту терезесі

Добро пожаловать - Куаныш!

Назад...

Имя БД	Дата создания	Edit	X
КуанышДБ	27.12.2013	Edit	X
ку	06.01.2014	Edit	X
eee	07.01.2014	Edit	X
		Add	

Выберите БД

КуанышДБ

Далее...

Сурет 3.13 – Қолданушы өз құжатың жаңадан жасай алады және өзгерте алады. Керекті құжатты тандап, ‘Далее..’ батырмасын басу қажет .

Алғашқы жүктемелерді бөлу беті Бұл бет 4 сұрыптау сырғымалы табырмадан, Excel – ге есеп бету батырмасы және ‘Сброс’ батырмасынан тұрады. Жүктеме бөлімінде жана пәндерді енгізуге, өзгертуге, өшіруге, сұрыптауға болады (Сурет 3.14).

Назад... Добро пожаловать - Куаныш! Имя БД - КуанышДБ 1

Дисциплины Распределение Нагрузки Нераспределенная Нагрузка Отчеты Справочники

Все дисциплины Все семестры Все формы Все группы Сброс

Дисциплина	Группа	Кол.гр	Конг	Сем	Форм.обуч.	Лекции	Прак	Лаб	Конс	Экзамены	Зачеты	Пр.Прак	Курс.раб	Дипл	Рец	Асп	ГАК	КСРС	РГР	Зав.каф	Другое	Всего	Edit	X		
Cisco	БВТк-13	2	50	1	до		45			41										5	6	5	7	109	Edit	X
MS SQL	БВТк-10	2	50	1	з.о.	4				5			9	7	9					5		9	121	Edit	X	
Oracle	ЭФ-12	3	96	2	до	87		87		8			54		87		87					87	9	506	Edit	X
C++	БВТк-13	2	50	2	до																	6	6	Edit	X	
Oracle	БВТк-13	2	50	1	до		5																5	Edit	X	
CSS	ЭФ-12	3	96	1	до	1	2				6												12	Edit	X	
Delphi 2010	БИС-12	2	65	2	до	3		3				3	3		3		3			3	3	3	24	Edit	X	
C++	БВТ-11	1	45	1	д.у.о			6		6			6										18	Edit	X	
MS SQL	БИС-12	2	65	2	до	1		9	9					9	9					9			46	Edit	X	
Delphi 2010	ЭФ-13	3	96	1	до	6	6	6				6			6		6						36	Edit	X	
Математика	БВТк-13	2	50	2	до		6	1				6	4					6				6	29	Edit	X	
Php	БИС-12	2	65	2	до	3				3	3											3	15	Edit	X	
CSS	ФРТС-10	2	56	2	до		21		3		3				3						1	21	52	Edit	X	
Java EE	ФРТС-10	2	56	2	до	6			1		2				11								20	Edit	X	
Математика	ФРТС-10	2	56	1	до	3		2			22											1	28	Edit	X	
Delphi 2010	БИС-12	2	65	1	до			11															11	Edit	X	

	Лекции	Прак	Лаб	Конс	Экзамены	Зачеты	Пр.Прак	Курс.раб	Дип	Рец	Асп	ГАК	КСРС	РГР	Зав.каф	Другое	Всего
1 семестр	14	99	8	9	52	28	6	15	12	15	6	6	83	1	16		370
2 семестр	100	27	87	16	18	6	68	7	9	90	23	87	9	13	90	48	698
за год	114	126	95	25	70	34	74	22	21	90	38	93	15	96	91	64	1068

Экспорт

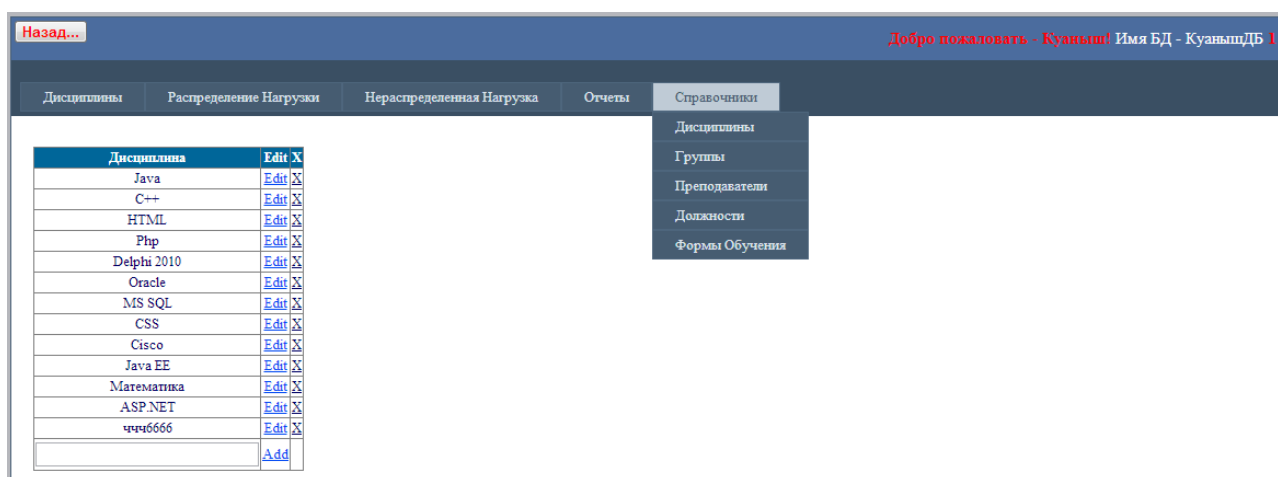
Сурет 3.14 – Жүктемелерді бөлу беті

Бағдарламада Excel-ге шығаруға (Сурет 3.15), жоғары жағында сабақтардың аты жазылған.

№	Дисциплина	Индекс группы	Количество групп	Количество студентов	Семестр	Форма обучения	Лекции	Практические занятия	Лабораторные	Консультации	Экзамены	Зачеты	Производственная практика	Курсовые	Дипломные	Рецензирование	Аспирантура	ГАК	КСРС	РГР	Заведование кафедры	Прочее	Всего	
1	Cisco	БВТк-13	2	50	1	д.о		45			41			5			6			5		7	1	
2	MS SQL	БВТк-10	2	50	1	з.о.	4				5		9	7		9				78		9	1	
3	Oracle	ЭФ-12	3	96	2	д.о	87		87		8		54			87		87				87	9	1
4	C++	БВТк-13	2	50	2	д.о																6	1	
5	Oracle	БВТк-13	2	50	1	д.о		5																1
6	CSS	ЭФ-12	3	96	1	д.о	1	2		3		6												1
7	Delphi 2010	БИС-12	2	65	2	д.о	3			3			3	3			3			3		3	3	1
8	C++	БВТ-11	1	45	1	д.у.о			6		6			6										1
9	MS SQL	БИС-12	2	65	2	д.о	1			9	9				9		9				9			1
10	Delphi 2010	ЭФ-13	3	96	1	д.о	6	6		6			6				6		6					1
11	Математика	БВТк-13	2	50	2	д.о		6		1			6	4					6				6	1
12	Php	БИС-12	2	65	2	д.о	3					3	3								3		3	1
13	CSS	ФРТС-10	2	56	2	д.о		21		3		3				3					1		21	1
14	Java EE	ФРТС-10	2	56	2	д.о	6			1			2				11							1
15	Математика	ФРТС-10	2	56	1	д.о	3		2			22										1		1
16	Delphi 2010	БИС-12	2	55	1	з.о.		41																1
Всего за 1-й семестр							14	99	8	9	52	28	6	15	12		15	6	6	83	1	16	370	
Всего за 2-й семестр							100	27	87	16	18	6	68	7	9	90	23	87	9	13	90	48	698	
Всего за год							114	126	95	25	70	34	74	22	21	90	38	93	15	96	91	64	1068	

Сурет 3.15 – Жүктемелерді Excel – ге экспорттау

Анықтама беті. Таңдау бөлімінде бес түрлі сырғымалы батырма арқылы қажетті кестелерді таңдауға болады. Таңдалып алынған кесте негізгі кестелер бөлінде көрсетіледі. Өңдеу бөлімінде таңдап алынған кестемен жұмыс істейді, яғни жаңа ақпарат қосы, енгізілген өзгерістерді сақтау және қажет емес ақпараттарды өшіруге болады (Сурет 3.16).



Сурет 3.16 – Анықтама бетін көрсететін терезе

Жүктемені бөлу беті. Бұл бөлім жүктемені бөлуге арналған, яғни қандай пәнді қандай оқытушы беретіні, қанша, практикаға, экзаменге және т.б. қанша сағат берумен жұмыс атқарады.

Бұл бет бес бөліктен тұрады: таңдау жасау, сұрыптама жасау, өңдеу, Excel – ге экспорттау, бөлінген пәндер кестесін көрсету және бөлінген сағаттардың, сонымен қатар, жоспар бойынша сағаттар санының қосындысы. Оқытушылар не болмаса пәндерді кім үйретіні бойынша таңдау жасауға болады (Сурет 3.17).

Дисциплины		Распределение Нагрузки		Нераспределенная Нагрузка		Отчеты		Справочники																	
Куралбаев З		0.25 профессор																							
Выбор	Дисциплина	Группа	Сем	Фор.обуч.	Лекции	Прак.	Лаб.	Конс.	Экзамены	Зачеты	Пр.прак.	Курс.раб.	Дипл.	Рец.	Асп	ГАК	КСРС	РГР	Зав.каф.	Другое	Всего	Комм.	Edit	X	
	MS SQL	БВТк-10	1	з.о.	4				5			9	7	9					78	9	121		Edit	X	
	C++	БВТк-13	2	д.о																6	6		Edit	X	
	Php	БИС-12	2	д.о	3					3	3								3	3	15		Edit	X	
	Математика	ФРТС-10	1	д.о	1					17											18		Edit	X	
	Математика	БВТк-13	2	д.о		6															6		Edit	X	
	Oracle	ЭФ-12	2	д.о	87																87		Edit	X	
	Java EE	ФРТС-10	2	д.о																	0		Edit	X	
Итого по преподавателю					Лек.	Прак.	Лаб.	Конс.	Экс.	Зач.	Про.прак.	Курс.раб.	Дипл.	Рец.	Асп	ГАК	КСРС	РГР	Зав.каф.	Другое	Всего				
				1 семестр	5				5	17		9	7	9						78	9	139			
				2 семестр	90	6				3	3									3	9	114			
				за год	95	6			5	20	3	9	7	9						81	18	253			
Дисциплину		Преподаватель	Гр.	Сем.	Фор.обу	Лек.	Прак.	Лаб.	Конс.	Экс.	Зач.	Про.прак.	Курс.раб.	Дипл.	Рец.	Асп	ГАК	КСРС	РГР	Зав.каф.	Другое	Всего			
		Жунусов З.А.	ФРТС-10	1	д.о	2		6			5					5							18		
		Куралбаев З.К.	ФРТС-10	1	д.о	1					17												18		
Распределено					Лек.	Прак.	Лаб.	Конс.	Экс.	Зач.	Про.прак.	Курс.раб.	Дипл.	Рец.	Асп	ГАК	КСРС	РГР	Зав.каф.	Другое	Всего				
				Распределено часов	3		6			22					5								36		
				Часов по плану	3		2			22										1			28		
				Не распределено часов	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8		

Сурет 3.17 – Жүктемені бөлу терезесі

Есеп беру беті. Есеп беру маңызды бөліктерінің бірі болып табылады. Қандай пәнді қандай мұғалімдер беретіні, қанша саған бөлінгені жайлы толық ақпарат алуға болады (Сурет 3.20).

Бұл төрт бөліктен тұрады: таңдау жасау, сұрыптама жасау, Excel – ге экспорттау (Сурет 3.19), бөлінбеген пәндер бойынша есеп беру кестесін көрсету. Оқытушылар не болмаса пәндерді кім үйретіні бойынша таңдау жасауға болады (Сурет 3.18).

Дисциплины	Распределение Нагрузки	Нераспределенная Нагрузка	Отчеты	Справочники																			
Все дисциплины	+ Все семестры	+ Все формы	+ Все группы	Экспорт Обновить Button																			
Преподаватель	Должность	Дисциплина	Сем.	Ф.обу	Группа	Лекции	Прак.	Лаб.	Конс.	Экс.	Зач	Про.прак	Курс.раб	Дипл	Рецен	Аспир.	ГАК	КСРС	РГР	Зав.каф	Другое	Всего	
Аманбаев А.А.	0.25 ассистент	Java EE	2	д.о	ФРТС-10	4																	4
Аманбаев А.А.	0.25 ассистент	Delphi 2010	1	з.о.	БИС-12		41																41
Аманбаев А.А.	0.25 ассистент	CSS	1	д.о	ЭФ-12	1	2																3
Аманбаев А.А.	0.25 ассистент	CSS	2	д.о	ФРТС-10		21		3														24
Байжанова Д.О.	0.25 доцент	MS SQL	2	д.о	БИС-12	1		9	9					9		9			9				46
Бимурзаев З.К.	доцент	Oracle	2	д.о	ЭФ-12	2				2					2						2		8
Бимурзаев З.К.	доцент	Java EE	2	д.о	ФРТС-10	2					2												4
Бимурзаев З.К.	доцент	Oracle	1	д.о	БВТк-13		5																5
Жексенкулов К.Е.	профессор	Delphi 2010	2	д.о	БИС-12	3			3		3	3			3		3			3	3	3	24
Жунусов З.А.	профессор	Математика	1	д.о	ФРТС-10	2		6			5				5								18
Жунусов З.А.	профессор	Oracle	1	д.о	БВТк-13		3					2							1				6
Куралбаев З.К.	0.25 профессор	C++	2	д.о	БВТк-13																	6	6

Отчет по... Преподавателям Диспетчерская

Экспорт

Сурет 3.18 – Есеп беру терезесі

№	Дисциплина	Индекс группы	Количество групп	Количество студентов	Семестр	Форма обучения	Лекции	Практические занятия	Лабораторные	Консультации	Экзамены	Зачеты	Производственная практика	Курсовые	Дипломные	Рецензирование	Аспирантура	ГАК	КСРС	РГР	Заведование кафедр	Прочие	Всего
1	Аманбаев А.А.	0.25 ассистент	Java EE	2	д.о	ФРТС-10	4																4
2	Аманбаев А.А.	0.25 ассистент	Delphi 2010	1	з.о.	БИС-12		41															41
3	Аманбаев А.А.	0.25 ассистент	CSS	1	д.о	ЭФ-12	1	2															3
4	Аманбаев А.А.	0.25 ассистент	CSS	2	д.о	ФРТС-10		21		3													24
5	Байжанова Д.О.	0.25 доцент	MS SQL	2	д.о	БИС-12	1		9	9					9		9				9		46
6	Бимурзаев З.К.	доцент	Oracle	2	д.о	ЭФ-12	2			2						2						2	8
7	Бимурзаев З.К.	доцент	Java EE	2	д.о	ФРТС-10	2					2											4
8	Бимурзаев З.К.	доцент	Oracle	1	д.о	БВТк-13		5															5
9	Жексенкулов К.Е.	профессор	Delphi 2010	2	д.о	БИС-12	3			3		3	3			3		3			3	3	24
10	Жунусов З.А.	профессор	Математика	1	д.о	ФРТС-10	2		6			5					5						18
11	Жунусов З.А.	профессор	Oracle	1	д.о	БВТк-13		3				2									1		6
12	Куралбаев З.К.	0.25 профессор	C++	2	д.о	БВТк-13																	6
13	Куралбаев З.К.	0.25 профессор	MS SQL	1	з.о.	БВТк-10	4							9	7		9				78	9	121
14	Куралбаев З.К.	0.25 профессор	Php	2	д.о	БИС-12	3					3	3								3	3	15
15	Куралбаев З.К.	0.25 профессор	Математика	1	д.о	ФРТС-10	1																18
16	Куралбаев З.К.	0.25 профессор	Математика	2	д.о	БВТк-13		6															6
17	Куралбаев З.К.	0.25 профессор	Oracle	2	д.о	ЭФ-12	87																87
18	Куралбаев З.К.	0.25 профессор	Java EE	2	д.о	ФРТС-10																	0
19	Сатимова Е.Г.	0.25 ассистент	Delphi 2010	1	д.о	ЭФ-13	1																1
20	Шайхин Б.М.	0.25 ассистент	Delphi 2010	1	д.о	ЭФ-13		6		6			6				6		6				30

Сурет 3.19 – Excel-ге экспорттау мұғалім бойынша

№	Ф.И.О. Преподават	Должность и степень	Лекции	Практика	Лабораторные	Консультации	Экзамены	Зачеты	Производствен на практика	Курсовые	Дипломные	Рецензировани	Руководство аспирантов	Участие в ГАК	КСРС	РГР	Заведование кафедр	Другое	Всего
Прогр. на алг.яз			семестр: 1			форма обучения: д.о.			индекс группы: БВТ-11										
1	Бимурзаев С.Б.	профессор	30	150	11	20									12	50			253
Итого за семестр			30	150	11	20									12	50			273
ООПТ			семестр: 2			форма обучения: д.о.			индекс группы: БВТ-10										
1	Бимагамбетов Т.С.	доцент		60		24										30			90
3	Бимурзаев С.Б.	профессор	23	30	9										16	10			88
Итого за семестр			23	90	9	24									16	40			202
Прогр. на алг.яз			семестр: 1			форма обучения: д.о.			индекс группы: БВТк-11										
1	Акижанова З.А.	ассистент			30										10	25			65
2	Куралбаев З.К.	профессор	30			7													37
4	Тлеубаева А.Ж.	ассистент		90												15			105
Итого за семестр			30	120	7	16									10	40			223
Корпор. БД			семестр: 1			форма обучения: д.о.			индекс группы: БИСк-08										
1	Аманбаев А.А.	доцент	22	8	30	3				20					5				88
3	Сатимова Е.Г.	0.5 доцент						8											8
Итого за семестр			22	8	60	3	8			40					5				146

Сурет 3.20 – Excel-ге экспорттау пән бойынша

Бөлінбеген сабақтар беті. Бөлінбеген сабақтар бетінде қанша сабақ бөлінбегенін, артық бөліп қойған сабақтарды көруге болады. Кестегеді 0 – ден кіші сандар, яғни минус сандар көп бөлініп қойған сабақтар. Бұл терезеде де сұрыптауға (Сурет 3.21), Excel – ге шығаруға болады.

Назад... Добро пожаловать - Куаныш Имя БД - КуанышДБ 1

Дисциплины | Распределение Нагрузки | Нераспределенная Нагрузка | Отчеты | Справочники

Все дисциплины | Все семестры | Все формы | Все группы | Сброс | Обновить

Дисциплина	Группа	Форма обучения	Семестр	Лекции	Прак	Лаб	Конс	Экзамены	Зачеты	Пр.прак	Курс.раб	Дипл	Реп	Асп	ГАК	КСРС	РГР	Зав.каф	Другое	Всего
CSS	ЭФ-12	д.о	1				3		6											9
CSS	ФРТС-10	д.о	2						3				3				1		21	28
Dejrn 2010	ЭФ-13	д.о	1	5																5
Java EE	ФРТС-10	д.о	2					1						11						12
Oracle	ЭФ-12	д.о	2	-2		87		6		54			85		87			87	7	411
Математика	ФРТС-10	д.о	1			-4								-5				1		-8
Математика	БВТк-13	д.о	2				1			6	4					6			6	23
Cisco	БВТк-13	д.о	1		45			41					5		6		5		7	109
C++	БВТ-11	д.у.о	1			6		6			6									18

	Лекции	Прак	Лаб	Конс	Экзамены	Зачеты	Пр.Прак	Курс.раб	Дипл	Реп	Асп	ГАК	КСРС	РГР	Зав.каф	Другое	Всего
Распределено часов	111	84	6	21	16	25	16	12	16	2	32		9	91	3	23	467
Часов по плану	114	126	95	25	70	34	74	22	21	90	38	93	15	96	91	64	1068
Не распределено часов	3	42	89	4	54	9	58	10	5	88	6	93	6	5	88	41	601

Экспорт

Сурет 3.21 – Бөлінбеген сабақтар беті

4 Тіршілік қауіпсіздігі

4.1 Ауа алмасу (кондиционер) жүйесінің құрылғысы және есебі

Еңбек жағдайын талдау. Дипломдық жұмыста мен кафедраның автоматтандырылған жүйесін жобалау және құру бағдарламасын жасадым. АЖО құру және жобалау үшін MS SQL дерекқор басқару жүйесі (ДҚБЖ) және Visual Studio бағдарламалық ортасы таңдалған. Қолданыстағы дерекқор негізінде жаңалау көзделеді.

Мекеме бөлмесі төменде (Сурет 4.1) көрсетілген. Алматы қаласындағы KuanyshIT компаниясының жұмыс орнын алдым. Бөлменің ұзындығы 7 м, ені 6 м және биіктігі 3 м, ұзындығы 2,5 м екі терезе бар. Бөлмеде 4 адам жұмыс істейді, жұмыс графигі аптасына 5 күн, күніне 8 сағат.

Дербес компьютер операторының жұмысы ұзақ көру жұмысымен байланысты болғандықтан, мекеме бөлмесіндегі жарықтануды есепке алу қажет. Жарықтану деңгейі психикалық функциялардың күйіне және ағзадағы физиологиялық үрдістерге әсер етеді. Бөлме компьютерлік құрылғылар мен оргтехникамен жабдықталған, сол себептен мекеменің персоналы артық жылулық сәулеленуге шалдығады. Сондықтан персоналдың қолайлы еңбек ету шарттарын қамтамасыз ету үшін микроклимат параметрлерін нормалау қажет. Микроклиматтың бөлек параметрлерінің ұсынылған мәндерінен ауытқуы жұмысшының еңбекке қабілеттілігін төмендетеді, көңіл күйін нашарлатады және кәсіби ауруларға әкелуі мүмкін. ССБТ сәйкес категориясы I а жеңіл физикалық жұмыс үшін қалыпты микроклиматтық шарттар келтірілген (Кесте 4.1).

К е с т е 4.1 – Микроклимат параметрлерінің қалыпты нормалары

Жыл мезгілі	Жұмыс категориясы	Температура, °С	Ауа қозғалысының жылдамдығы, м/с
Салқын	I а	18 – 26	0,1
Жылы	I а	20 – 30	0,2

Мекеме бөлмесіндегі жаз уақыт кезіндегі температура +26°С-ге дейін көтеріледі, ал қыс кезіндегі температура +18–ден +20°С – ге дейін. Қажетті микроклиматтық шарттарды сақтау үшін бөлме кондиционермен жабдықталған.

Бөлменің терезелер арқылы түсетін табиғи жарықтануы, және тәуліктің қараңғы уақытында жұмыс істеу мүмкіндігін беретін жасанды жарықтануы бар. Жасанды жарықталу люминесцентті шамдар арқылы жүзеге асырылады.

Қызмет көрсетушілердің қауіпсіздігін қамтамасыз ету үшін бөлме қызметкерлеріне әсер ететін барлық мүмкін факторларды талдау қажет.

Бөлмеде құрылғылардың мынадай түрлері қолданылады. Дербес компьютерлер саны – 4. Зиян электрмагнитті сәулелердің әсері оларды операторлардан алысырақ орналастырудан және дербес электрондық есептеуіш машина (ДЭЕМ) мониторуна қорғаныс экранын орнатудан төмендейді. Газдылықтың, шаңдылықтың және қондырғының изоляциясынан туындайтын зиян булардың әсері табиғи желденуді қамтамасыз ететін құрылғыларды дұрыс орналастыру есебінен жойылады.

Көрермен залы мен дыбыстық қамтамасыз етудің аппараттық бөлмесі арасындағы әуе шуының изоляция индексі 50 дБ – ден кем болмауы керек. Дыбысты қамтамасыз етудің аппараттық бөлмесінің қабырғалары мен төбесі 500 – 2000 Гц жиіліктер диапазонында дыбысты жұту коэффициенті 0,6 – дан кем болмайтын дыбысты жұтқыш материалдармен қапталуы керек. Дыбысты қамтамасыз ету жүйесінің барлық техникалық аппараттық бөлмелерінің едендері шаң тудырмайтын болмауы және күнделікті ылғалды жинастыру жұмыстарын өткізуге мүмкіндік беретін (метлах тақтасы, линолеум) болуы керек.

Дербес электрондық есептеуіш машина қолданушысының жұмыс орнын ұйымдастыруда келесі негізгі талаптар сақталуы қажет:

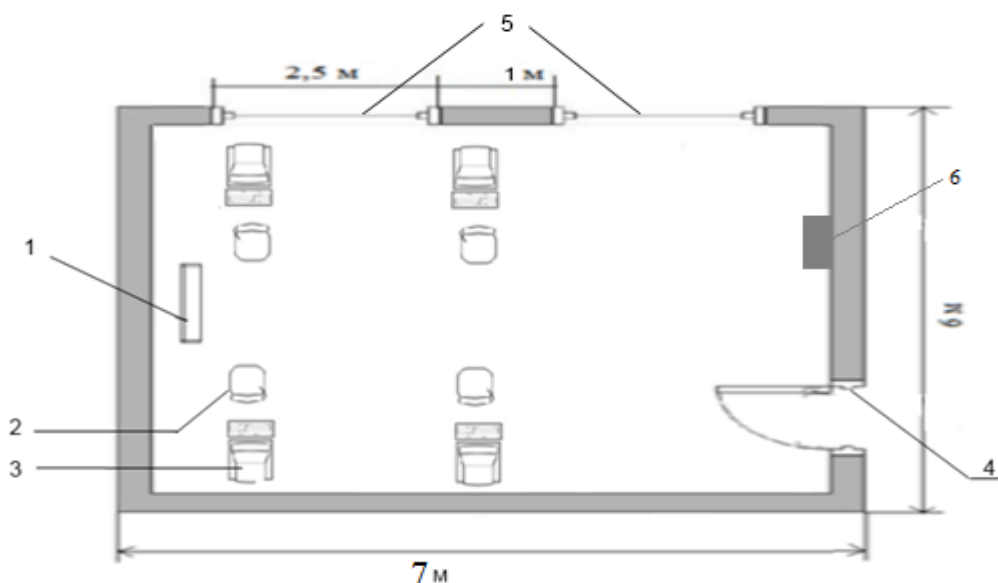
- жұмыс орнының құрамына кіретін құрылғылардың оптималды орналасуы;
- барлық қажет қозғалыстар мен орын ауыстыруларды жүзеге асыруға мүмкіндік беретін жеткілікті жұмыс аймағы;
- қызметтерді іске асыру үшін табиғи және жасанды жарықтандыру қажет;
- акустикалық шудың деңгейі рұқсат етілген мәнінен аспауы керек.

Бөлмеде келесі құрал-жабдық қолданылады. Дербес компьютер – 5 дана

Құрылғының техникалық сипаттамалары:

- Acer core i7 3210/4Gb/500Gb/Combo/DOS дербес компьютері;
- SAMSUNG LS19A100N монитору;
- мөлшерлер 1200x750x1150 мм (дербес компьютер+үстел);
- электрлік қоректену көзі: айнымалы кернеу 220 – 250 В, 50 Гц жиілігі, қуаты 400 Вт.

Сплит-жүйе плазма Panasonic CS/CU – E 18 NKDW– кондиционер, қуаты 5 кВт.



Сурет 4.1 – Мекеме бөлмесі, 1 – кондиционер, 2 – орындық, 3 – үстел және дербес компьютер, 4 – есік, 5 – терезе, 6 – өрт сөндіргіш

Операторлар бөлмесінің микроклиматтық шамалары: жыл мезгілінің суық кездерінде ауа қозғалысының жылдамдығы және салыстырмалы ылғалдылығы 0,1 м/с, 60%, ауа температурасы 18 – 26°C шамасында болады.

Ал жыл мезгілінің жылы кездерінде ауа қозғалысының жылдамдығы және салыстырмалы ылғалдылығы 0,2 м/с, 60 – 70%, келтірілген шамалар адам организміне ыңғайлы нормаларға сай келмейді. Сондықтан операторлар бөлмесінде ауаны кондиционерлеу мәселесі қарастырылған.

4.2 Бөлменің электр торабы

Адамның электр тогынан зақымдану ықтималдығына әсер ететін біздің бөлмеміздің класын анықтайық:

- едендер бір қабатты поливинилхлоридті антистатикалық линолеуммен қапталған, сондықтан ол ток өткізбейтін болып табылады;
- ауаның салыстырмалы ылғалдылығы 60% – дан аспайды, сондықтан бөлме құрғақ;
- ауа температурасы Цельсий бойынша плюс 30 градустан аспайды;
- адамның бір уақытта бір жақтан жермен байланысы бар технологиялық жабдықтардың корпустарымен және басқа жерлендірілген бөліктермен, екінші жақтан электр жабдықтарының металл корпустарымен немесе ток өткізуші бөліктермен жанасу мүмкіндіктерінің болмауы (кернеу 1000 В мәнінен аспағандықтан сымдардың өте жақсы изоляциясында);
- химиялық белсенді заттар жоқ осы бөлмені маңызды қауіпсіз бөлме ретінде классификациялауға болады.

Біздің жағдайымызда электр қауіпсіздігін қамтамасыз ету үшін бойынша жерлендіру мүмкіндігін қарастыру қажет. Біздің жағдайымыздағы кернеу – 220 В, сондықтан жерлендіру мен нөлдеу міндеттелмейді, бірақ ұсынылады

4.3 Бөлменің өрт қауіпсіздігі

Құрылыс конструкцияларын дайындау үшін кірпіш, темір бетон, әйнек, металл және басқа жанбайтын материалдар қолданылады. Сонымен қатар жанбайтын материалдардан жасалған қоршаулар түріндегі өртке қарсы өткелдерді ескеру қажет, олар біздің офистің бөлмелері арасында орнатылады. Ғимараттарда өрт крандары дәлістерде, баспалдақ торларында және кіре беріс аумақтарында орнатылады. Дербес электрондық есептеуіш машинаны қолданушылар бөлмелерінде, архивте және қосымша, қызметтік бөлмелердегі өртті өшіру үшін су қолданылады. Дербес электрондық есептеуіш машина бар бөлмелерде, ақпаратты тасушыларды сақтау бөлмелерінде, қымбат құрылғыларды бұзу немесе толықтай істен шығару қаупінен бақылау–өлшеуіш жабдықтары бар бөлмелерде суды қолдану тек кейбір жағдайларда ғана рұқсат етіледі, мысалы өрт қауіпті ірі көлмеде болғанда. Бірақ судың мөлшері минималды болуы және дербес электрондық есептеуіш машинаны, дыбыстық құрылғыларды брезентпен немесе матамен жауып судан қорғау керек.

Барлық бөлмелерді стационарлы автоматты өрт өшіргіш қондырғылармен жабдықтау қажет. Ауа құрамындағы оттегіні тез азайтатын от өшіргіш газбен бөлмені бірден толтыруға негізделген өртті газбен өшіру қондырғыларын қолданған тиімдірек болып табылады.

4.4 Бөлмедегі және сырттағы зиянды элементтер

Зиян химиялық заттардың деңгейін нормалау. Бөлмені ластану көздері сыртқы ортаның және ғимараттың құрылыс материалдарынан, жиһаздардан, киімнен, аяқ-киімнен бөлінетін жүздеген әрекеттесулердің зиян заттары және адамның биоактивті әрекеттесулері (антропогендер) болып табылады.

Бөлменің сыртқы ортаның зиян заттарымен ластануын қарастыра отырып, ең алдымен ғимараттың орналасқан орнын ескеру қажет, біздің жағдайымызда ол автострадаға жақын орналасқан. Бөлмеге сыртқы ортадан келетін жиі ластағыштар көміртек оксиді, азот диоксиді, күкірт диоксиді, қорғасын, шаң және тағы басқалары болып табылады.

Құрылыс конструкциялары бөлменің радон және торонмен ластануын көзі болып келеді, сонымен қатар ең көбірек концентрация нашар желдетуі бар бетоннан жасалған үйлерде кездеседі.

Жиһаз, киім және аяқ – киімдер минералды талшықты, көмір сутегісі, полиэфир кара майы және тағыбасқа зиянды заттары бар шаңды бөледі. Биоктивті әрекеттесулердің ең маңыздысы көміртек диоксиді, күкірт сутегісі және тағы басқалары болып табылады.

4.5 Бөлмедегі дыбыстар , яғни шу

Дербес электрондық есептеуіш машина қолданушысының, оператордың, жұмыс орнындағы шу көздері – сөйлесіп тұрған адамдар, сыртқы ортаның – компьютердің, принтердің, желдеткіш қондырғының шуы болып табылады. Олар болмашы мәнде шуды тудырады, сондықтан бөлмеде дыбысты жұтқыштарды қолдану жеткілікті.

Ең жақсы дыбысты жұту қасиеті талшықты – ауа көлемді материалдарда: фибролитті плиталарда, жарықталшықтарында, минералды мақтада, полиуретанды поропластта, ауа көлемді поливинилхлоридте және басқаларында болады. Дыбыс жұтқыш материалдарға дыбыс жұту коэффициенті 0,2 мәнінен төмен емес материалдар жатады.

4.6 Бөлменің жарықтандыру көздері

Дербес электрондық есептеуіш машинамен жұмыс істеуге арналған бөлменің тиімді жарықтандырылуы табиғи және жасанды жарықтың болуы есебінен жасалады.

Жеткіліксіз жарықтандыру көздің, адамның тез шаршауына, жақыннан көргіштікке, жұмыс сапасының төмендеуіне, ақаудың көбеюіне соқтырады. Тым жарық көз қабықшасын тітіркендіреді, шағылыстырады, көз тез шаршайды, өндірістік травматизм көбейеді.

Қаралып жатқан бөлмеде жұмыс істеуге қажетті жарықтандыру қалып бойынша $E_n=300$ лк, осылайша бөлмедегі қажетті жарықтандыруды қамтамасыз етеміз.

4.7 Бөлменің ауа жүйесі

Жұмыс бөлмесінде белсенді кондиционерлеу және вентиляция жүйесі жоқ. Операторлар залын талдау барысында жұмыс аумағының қалыпты микроклиматтық шарттарын міндетті түрде қарастыру қажет. Мұндай әмбебап жүйе ретінде автономдық кондиционерлер болып табылады.

Жұмыс істеу аймағының ауасы, жалпы санитарлы-гигиеналық талаптар сәйкес, компьютерлермен жабдықталған бөлмедегі адамдардың жұмысы жеңіл физикалық жұмысқа жатады. Ағзаның энергия жұмсау жұмыстарының категориялары келтірілген (Кесте 4.2).

К е с т е 4.2 – Адам ағзасының энергия жұмсау жұмыстарының категориялары

Жұмыс	Категория	Ағзаның энергия жұмсауы, Ккал/сағат,	Жұмыс сипаттамасы
-------	-----------	--------------------------------------	-------------------

		Дж/с	
Жеңіл	I a	<138	Жұмыс отырып жүргізіледі

4.8 Кондиционерлеу және ауаны жаңарту жүйелерін есептеу

Ауаны технологиялық кондиционерлеудің талаптары өнеркәсіптің әр түрлі салаларындағы технологиялық процестерді жүргізуде, сондай-ақ компьютерлік 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 қамтамасыз 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.

Кондиционерді таңдау үшін алдымен артық жылудың қосындысын, сонымен қатар оған күннің радиациясынан бөлінетін жылу кіреді, өндірістік жарықтануды, жұмыс істейтін адамдар санын, оргтехникаларды және т.б. есептеу қажет. Салқын өндіргіштік бойынша қосындысы сондай немесе шамалы үлкен мәнді, сонымен қатар қажетті ауа алмасу қамтамасыз ететін кондиционер моделі танданылады.

4.9 Температура айырымы нәтижесінде алынатын жылу және жылу жоғалту

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.

Салқын мезгіл үшін есептік сыртқы температура ($t_{\text{сырт.есеп}}$) ең салқын айдың 13 сағатындағы орташа температурасына, жылы период үшін – ең ыстық айдың 13 сағатындағы орташа температурасына сәйкес келеді. Ал ішкі ($t_{\text{іш.есеп}}$) жайлылық шартын және өндірістік процесстерде көрсетілетін технологиялық талаптарын ескере отырып таңдалады формулада берілген.

$$Q_{\text{қоршау}} = V_{\text{бөлме}} X_0 (t_{\text{шыққан}} - t_{\text{келген}}), \text{ Вт.} \quad (4.1)$$

мұнда $V_{\text{бөлме}}$ – бөлменің көлемі, м^3 . $V_{\text{бөлме}} = 7 \times 6 \times 3 = 126 \text{ м}^3$;

X_0 – меншікті жылулық сипаттама, $\text{Вт}/\text{м}^3 \text{ } ^\circ\text{C}$, формулада берілген;

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

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

$t_{\text{сырт.есеп}} = -25^\circ$ – жылдың суық мезгіліне арналған сыртқы есептік температурасы;

$t_{\text{іш.есеп}} = 24^\circ$ – жылдың жылы мезгіліне арналған ішкі есептік температурасы;

$t_{\text{іш.есеп}} = 20^\circ$ – жылдың суық мезгіліне арналған ішкі есептік температурасы.

Жылы мезгіл үшін (4.1).

$$Q_{\text{қоршау}} = 126 \times (27,6 - 24) \times 0,42 = 190,512 \text{ Вт.}$$

Салқын мезгіл үшін (4.1).

$$Q_{\text{қоршау}} = 126 \times 0,42 \times (-25 - 20) = -2381,4 \text{ Вт.}$$

Шынылау арқылы күннің сәулеленуінен келетін жылу. Күннің сәулеленуінен (радиация) келетін жылу терезе арқылы сәуле бөлмеге кіріп, күннен шынылау сәулелену периоды үшін формулада берілген.

$$Q_p = (q_{\text{тура}} + q_{\text{шашыр.}}) K_1^c K_2 \beta_{\text{ж.ө.}} n H_0 B_0, \text{ Вт.} \quad (4.3)$$

Күннің сәулелері терезеден кірмейтін көлеңке периоды үшін (шашыраңқы радиация) формулада берілген.

$$Q_p = q_{\text{шашыр.}} K_1^T K_2 \beta_{\text{ж.ө.}} n H_0 B_0, \text{ Вт.} \quad (4.4)$$

мұнда $q_{\text{тура}}$; $q_{\text{шашыр.}}$ – тура және шашыраңқы радиациядан келетін жылулық ағындар, $\text{Вт}/\text{м}^2$;

$F_0 = n H_0 B_0$ – жарықтық ойықтың ауданы, м^2 (n – терезелердің саны, биіктігі H_0 және ені B_0);

K_1 – қапсырмамен шынылаудың көлеңкелену коэффициенті (K_1^c – сәулеленген ойықтар үшін; K_1^T – көлеңкедегі ойықтар үшін);

K_2 – шынылаудың ластану коэффициенті;

$\beta_{ж.ө.}$ – жылу өткізу коэффициенті.

Алматы қаласындағы ҚуанушIT мекеме бөлмесіндегі шынылаудың ауданы, 44^0 СШ $F_0 = 2 \times 2,5 \times 2 = 10 \text{ м}^2$. Шынылаудың бағыты оңтүстік-шығыс (ОШ). Ішінде жарық перделері бар. $\beta_{ж.ө.} = 0,4$ деп қабылдаймыз. Түске дейін ОШ үшін, яғни сағат 9 – дан 12 – ге дейін 44^0 СШ ендікте тура радиацияның мәні (П) $q_{тура} = 387 \text{ Вт/м}^2$ және шашыраңқы радиацияның мәні (Р) $q_{шашыр.} = 101 \text{ Вт/м}^2$ тең. $44 - 68^0$ СШ ендік диапазонында металды қапсырмалы екі қабатты шынылау үшін $K_1 = K_1^C = 0,72$ егер ойық күнмен сәулеленген болса, яғни 9 – 10 және 13 – 14 сағат аралығындағы период үшін. $K_1 = K_1^T = 1,15$, 14–15 және 19 – 20 сағат аралығындағы период үшін. Әйнектің бірқалыпты ластануы коэффициенті $K_2 = 0,9$ қабылданады.

Тура сәулелену периодында 9 бен 14 сағат аралығында есептелу мына формула арқылы жүреді.

$$Q_p = (387 + 101) \times 0,72 \times 0,9 \times 10 \times 0,4 = 1265 \text{ Вт.}$$

ал көлеңкелену периодында 14 пен 20 сағат аралығында мына формуламен есептелінеді (4.4).

$$Q_p = 22 \times 1,15 \times 0,9 \times 10 \times 0,4 = 91 \text{ Вт.}$$

Максималды есептелу уақыты 9 – 10 сағат, жылу түсу 1265 Вт.

4.10 Адамдардан келетін жылу.

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 (Кесте 4.3). Адамдардың жылу таратуы төмендегі кестелермен сипатталады (Кесте 4.4).

К е с т е 4.3 – Адамның сыртқы ортаға жылу таратуы, Вт

Сыртқы орта температурасы $^{\circ}\text{C}$	Отырғандағы жағдай			Тұрғанда немесе жеңіл қозғалыс			Ауыр жұмыс		
	Анық	Жасырын	Жалпы	Анық	Жасырын	Жалпы	Анық	Жасырын	Жалпы
24	67	35	102	72	60	132	95	154	249
20	82	21	103	92	42	133	140	110	250

К е с т е 4.4 – Адам бөлетін ылғал және көміртегі саны

Параметрлер	Бөлме ауасының температурасындағы мәндер °С				
	15	20	25	30	35
Ылғал г/сағат	40	40	50	75	115

Бөлмеде 4 әйел адам-операторлар отырады. $t = 24$ °С температурада отырған күйде бір ер адам 67 Вт анық жылу, ал жалпы – 102 Вт жылу бөледі. Әйел адам ересек ер адамның жылу бөлу нормасының 85 %-ын, ал кішкентай бала – 75 %-ын бөледі деп саналады. Бөлмедегі адамдардың бөлетін анық жылуы $Q_a^a = 67 \times 4 \times 0,85 = 227,8$ Вт. Ал жалпы жылу $Q_a^ж = 102 \times 4 \times 0,85 = 346,8$ Вт.

$t = 20$ °С температурада бір ер кісі 82 Вт анық жылу және 103 Вт жалпы жылу бөледі. Бөлмедегі адамдардың бөлетін анық жылуы $Q_a^a = 82 \times 4 \times 0,85 = 278,8$ Вт. Ал жалпы жылуы $Q_a^ж = 103 \times 4 \times 0,85 = 350,2$ Вт.

$t = 24$ °С үшін ылғалдылық және көміртегі қышқылының мәндерін кестеден интерполяция жолымен табамыз бір адамнан 50 г/сағат ылғалдылық, 45 г/сағат көміртегі қышқылы бөлінеді. Ал 4 адамның ылғалдылығы $4 \times 50 = 200$ г/сағат, көміртегі қышқылы мөлшері $4 \times 45 = 180$ г/сағат құрайды.

$t = 20$ °С үшін 1 адамнан бөлінетін ылғалдылық – 40 г/сағат, көміртегі қышқылы – 45 г/сағат. 4 адамнан бөлінетін ылғалдылық $4 \times 40 = 160$ г/сағат. 4 адамнан бөлінетін көміртегі қышқылы мөлшері $4 \times 45 = 180$ г/сағат. Бөлмедегі адамдардан бөлінетін зиянды заттардың есептелуінің нәтижелері көрсетілген (Кесте 4.5).

К е с т е 4.5 – Бөлмедегі адамдардан бөлінетін зиянды заттардың есептелуінің нәтижелері

Жыл мезгілі	Температура °С	Жылу, Вт		Ылғалдылық, W г/сағат	CO ₂ г/сағат
		Q_a^a	$Q_a^ж$		
Жылы	24	227,8	346,8	200	180
Салқын	20	278,8	350,2	160	180

Жарықтану аспаптарынан, оргтехникадан және құрылғылардан келетін жылу Шамдардан келетін жылу мына формуламен есептеледі.

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

мұнда η - электр энергиясының жылулыққа ауысу коэффициенті;
люминесцентті шамдарды қолдану кезінде $\eta = 0,5 - 0,6$;
 $N_{\text{жарықтану}}$ – шамдардың орнатылған қуаты 65 Вт/м^2 ;
мекеменің еденінің ауданы $F_{\text{еден}} = 7 \times 6 = 42 \text{ м}^2$.

$$Q_{\text{жарықтану}} = 0,6 \times 65 \times 42 = 1638 \text{ Вт.}$$

Оргтехниканың әсерінен пайда болатын жылу ағыны бір компьютерге орташа есеппен 300 Вт алады. Мекемеде 4 дербес компьютер болғандықтан.

$$Q_{\text{құрал}} = 4 \times 300 = 1200 \text{ Вт.}$$

Орындалған есептеулерден формуласы бойынша мекеме бөлмесіне келетін жылу балансын құрамыз. Жылдың жылы мезгілінде: температура айырымы нәтижесінде келетін жылу $Q_{\text{қоршау}} = 191 \text{ Вт}$; күн радиациясынан $Q_p = 1265 \text{ Вт}$; адамдардан $Q_a^a = 279 \text{ Вт}$; жарықтану аспаптарынан $Q_{\text{жарықтану}} = 1638 \text{ Вт}$; оргтехника мен құрылғылардан $Q_{\text{құрал}} = 1200 \text{ Вт}$. Мекеменің жылулық балансы жазда формулада берілген.

$$Q_{\text{жылу.б}} = Q_{\text{қоршау}} + Q_p + Q_a^a + Q_{\text{жарықтану}} + Q_{\text{құрал}}, \text{ Вт.} \quad (4.6)$$

$$Q_{\text{жылу.б}} = 1265 + 279 + 191 + 1638 + 1200 = 4573 \text{ Вт} = 4,573 \text{ кВт.}$$

$$Q_{\text{жылу.б}} = 4573 \times 3600 = 16462,8 \text{ кДж/сағат құрайды.}$$

Жылдың салқын мезгілінде температура айырымы нәтижесінде жоғалатын жылу $Q_{\text{қоршау}} = -2381 \text{ Вт}$, күн радиациясынан келетін жылу $Q_p = 1265 \text{ Вт}$, адамдардан $Q_a^a = 279 \text{ Вт}$, жарықтану аспаптарынан $Q_{\text{жарықтану}} = 1638 \text{ Вт}$, оргтехника және құрылғылардан $Q_{\text{құрал}} = 1200 \text{ Вт}$. Мекеменің жылулық балансы қыста формулада берілген:

$$Q_{\text{жылу.б}} = Q_{\text{қоршау}} + Q_p + Q_a^a + Q_{\text{жарықтану}} + Q_{\text{құрал}}, \text{ Вт.} \quad (4.7)$$

4.11 Ауа алмасуды есептеу

$Q_{\text{жылу.б}}$ жазда $> Q_{\text{жылу.б}}$ қыста болғандықтан, $Q_{\text{жылу.б}}$ жазда мәнімен ауаның жылу кернеулігін мына формуламен $Q_n > 20 \text{ ккал/м}^3$ болғанда $\Delta t = 8^\circ \text{C}$. мұнда $C = 0,24 \text{ ккал/кг}^\circ \text{C}$ – ауаның жылу сыйымдылығы, $\gamma = 1,206 \text{ кг/м}^3$ – ағынды ауаның сыбағалы массасы.

Барлық артық жылулар $16462,8 \text{ кДж/сағат}$ немесе $16462,8 : 3600 = 4,573 \text{ кВт}$ құрайды. Бөлмеге қажетті ауа мөлшері $L = 1698 \text{ м}^3/\text{сағ} = 28,3 \text{ м}^3/\text{мин}$. Өз таңдауымызды Panasonic CS/CU-E 18 сплит-жүйесі кондиционеріне тоқтатамыз.

Кондиционердің техникалық сипаттамалары:

- салқын 5,30 кВт; жылу 5,90 кВт;
- қорек кернеуі 220В,50 Гц;
- салқынның жұмсайтын қуаты, кВт 1,45;
- жылудың жұмсайтын қуаты, кВт 1,52
- салқын/жылу жұмыс тогы, А 2,4 /3,2;
- EER, А 4,36;
- COP, А 4,41;
- жылдық ток пайдалануы 945 кВт*сағат;
- шудың деңгейі, ішкі (жоғ/орт/төм), дБ(А) 45/36/33;
- шудың деңгейі, сыртқы, дБ(А) 48;
- габаритті өлшемдері, Ш/В/Г, Ішкі, мм 300*1080*250;
- габаритті өлшемдері, Ш/В/Г, сыртқы, мм 690*870*325;
- салмағы, кг 13.

4.12 Табиғи жарықты 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 люминесцентты шамдардан тұрады. Жарық көзінің нормативы төмендегі көрсетілген (Кесте 4.6).

К е с т е 4.6 –Ортақ жарықтандыру жүйесіндегі ұсынылған жарық көздері

Түсті ажыратудағы талаптарға сәйкес көру жұмысының сипаттамасы	Жарықтандыр, лк	Күн көзінің түсу температурасының түс диапазоны $T_c, ^\circ K$	Қолданылатын шамдар түрі
Түсті ажырату	300, 400	3500 – 5500	ЛД, ЛДЦ, ЛБ,

Кесте 4.6 жалғасы

талаптары көп емес объектілердің түстеріндегі ерекшелігі	150, 200	3000 – 4500	ЛБ,(ЛХБ), НЛВД+МТЛ, ДРЛ
--	----------	-------------	----------------------------

Операторлардың жұмыс орындарының бөлмесін жасанды жарықтандыру үшін люминесцентті шамдарды қолданамыз, олардың жарық бергіштігі жоғары (75 им/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 $L = 7$ м, ені $B = 6$ м, биіктігі $H = 3$ м. Жұмыс бетінің еден деңгейінен биіктігі h_{pp} , $h_{pp} = 0,725$ м, терезелер $h_{но}, h_{но} = 0,8$ м биіктігінен басталады, терезелердің биіктігі h_o , $h_o = 1,5$ м. Жұмыс орының бөлмесі IV сағат белдеуі Алматы қаласы (жарықтану белдеуінің климаты – IV 50^0 солтүстік ұзындығы және оңтүстік (Алматы, Карағанды)).

Терезенің толық ауданы S_0 , m^2 , осы формуламен анықтаймыз.

$$100 \cdot \frac{S_0}{S_n} = \frac{e_n \cdot \eta_0}{\tau_0 \cdot r_1} \cdot k_{зд} \cdot k_3 \quad (4.8)$$

$$S_0 = \frac{S_n \cdot e_n \cdot \eta_0}{100 \cdot \tau_0 \cdot r_1} \cdot k_{зд} \cdot k_3 \quad (4.9)$$

мұнда S_n – бөлменің ауданы m^2 ; $S_n = 42 m^2$;

e_n – Ортақ жарықтандыру жарық көздері 4.3- кестесінен тандаймы;

көру жұмыстың биік дәлме-дәлділігі үшін $e_n = 1,2$ қабылдаймыз;

m_N – климаттың жарықтанукоэффициент, 4.3 – кестесінен тандаймыз.

Жарықтану белдеуін ескере отырып, анықтауымыз:

$$e_n^{IV} = e_n \cdot m \cdot c \quad (4.10)$$

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

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

$e_n^{IV} = 1,2 \cdot 0,7 \cdot 0,75 = 0,63$;

Бөлменің типын ескере, осыдан табамыз коэффициенті $k_3 = 1,2$ (жұмыс істеу орнын 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;

$\eta_0 = 10$ кесте алынған.

Бөлмедегі орташа шағылу коэффициент $\rho_{CP} = 0,5$ тек бір жағындағы жарықты қарастырамыз.

$$\frac{l_{pr}}{B} = \frac{0,5}{6} = 0,08 \quad (4.11)$$

Онда $r_1 = 1,05$ $k_{зд}$ – іске асыру кезіндегі жарық көздерінің шаңдануы мен жарамсыз болуын ескеретін қор коэффициенті $k_{зд} = 1$.

Терезелердің толық ауданын есептейміз.

$$S_0 = \frac{42 \cdot 1,35 \cdot 10 \cdot 1 \cdot 0,63}{100 \cdot 0,24 \cdot 1,05} = 14,175 \text{ м}^2 \quad (4.12)$$

Жарық өтуінің ауданы тең $S_{сп} = 14,175 \text{ м}^2$.

Осыған орай жарық өту ауданы ($1,5 \cdot 1,2 \cdot 2 = 3,6 < 14,175$) жұмыс орындағы қажетті еңбек жағдайларын қамтамасыз етпейді. Осы мақсатта еңбектің ең қолайлы с параметрі $7 \times 6 \times 3$ қамтасамыз ететін, $e_N = 0,84$ көру жұмысының III мінездемесі, сонда жұмыс орнында табиғи жарықты және жасанды жарықты қолданады.

5 Техника-экономикалық негізделуі

5.1 Жоба сипаттамасы

Бұл дипломдық жобада Кафедраның автоматтандырылған жүйесін жобалау және құру яғни , сабақтарын бөлу үшін дерекқор негізінде жасалған клиенттік программалық жоба.

Бұл бағдарлама университет, колледж, академия т.б. мұғалімдерінің сабақтарын тиімді бөлу үшін жасалған. Оқу орындарында негізінен оқытушылар мен топтар, пәндер, бөлінетін жүктемелер туралы ақпаратты жүйелеумен байланысты болады. Сондықтан, кафедра мұғалімінің жүктемесін бөлу жұмысы өте ыңғайлы, әрі жылдам бөлу үшін, барлық деректерді жүйелі түрде сақтау және қажетті деректі тез алу, сонымен қатар жаңа деректерді ыңғайлы түрде енгізу мүмкіндіктері қарастырылған.

Бағдарламаны құру үшін алдымен зерттелетін облысының талдауы жасалды.

Келесі кезеңде оқытушылар мен пәндер, топтар туралы қажетті деректерді сақтайтын логикалық байланысқан кестелер құрылып, олардың негізінде жүйенің инфологиялық моделі алынады. Инфологиялық модель MS SQL Server–жүйесінде жасалады. Кафедрасының дерекқоры MS SQL Server дерекқорларды басқару жүйесінде құрылғаннан кейін, пайдаланушылардың барлық сұраныстарын қанағаттандыратын клиенттік қосымша жобаланады.

Тұтас сегмент мен позициялау. Тұтас сегмент – бұл қазіргі техникалық прогрестің мүмкіндігін толық пайдаланып жұмыс істеуді қажет ететін электроэнергетикалық компаниялар жүйесі. Қызметті SWOT – талдау төменде көрсетілген (Кесте 5.1).

К е с т е 5.1 – Қызметті SWOT–талдау

Мықты жақтары	Әлсіз жақтары
Бағдарлама мүмкіншіліктерінің көптігі; Қолжетімділігі; Қызмет бірақ рет орнатылады қосымша жаңартуларды жыл сайын қажет етпейді;	Электроэнергетика саласындағы адамдар үшін пайдалануға арналған. Басқа қызмет иелері үшін тиімсіз болып табылады.
Мүмкіндіктері	Қаупі
Қосымша табыс көзіне қол жеткізу.	Ғаламторда арқылы жұмыс істегендіктен хакерлер, вирустар бағдарламаның істен шығуына әсер етуі мүмкін.

5.2 Маркетинг–микс қызметтері

Маркетинг – микс – Маркетингтік құралдардың жиынтығы, бұл арқылы компания тауарлар немесе қызметтер сұранысына ықпал етуге тырысады.

Маркетинг – микс – бұл белгілі бір тауарға анықталған баға бойынша белгілі бір орында қозғалыс арқылы сұранысты қалыптастыратын жиындардың (стратегия) жүйесі.

Айта кету керек, стратегия тек компанияның белгілі бір мақсатта нарықтың кез келген сегментінде қозғалуына арналған ұмтылысы және стратегия жалпы, бөлінбейтін сипатқа ие. Жағдайды басқару мен оны назардан тыс шығармау үшін тек нақты орындалу уақыты анықталған жиындарды іске асыру керек, осы жиындарға жауапты адамдарды, қажетті ресурстарды (қаржылық, адами және т.б.) басқару қажет. іске асыру қажет. Міне сондықтан таңдалған стратегия үшін маркетинг – микс деп аталатын арнайы жиындар іске асырылады. Осы маркетингтің тактикалық деңгейі болып табылады.

Ереже бойынша, стандартты маркетинг – микс жиындарды төрт құраушы тактикалық іс бойынша сипаттайды, маркетинг жиынтығы (marketing–mix) – бұл фирма біртұтас нарықтың оң жауабын алу үшін қолданатын басқаруға мүмкіндік беретін маркетингтік факторлардың жиыны. Маркетинг-микстің міндеті - біртұтас нарықта потенциалды тұтынушылардың қажеттерін өтеумен ғана шектелмейтін, сонымен қоса ұйымның әсерін жоғарлатуға тырысатын жиынтықтарды (mix) құрастыру.

Продукт (Product) – бұл фирманың тұтас нарыққа ұсынатын «бұйымдар мен тауарлар» жиынтығы.

Баға (Price) – тұтынушылардың тауарды алу үшін төлеуі қажет ақша суммасы. Дистрибуция (Place) – тауар барлық бірыңғай тұтынушыларға қолжетімді болу үшін жүзеге асырылатын мүмкіндігі көп іс.

Қозғалу, даму (Promotion) – фирманың өзінің тауарлары туралы ақпарат таратуы мен тұтынушыларды осы тауарларды сатып алуға үгіттейтін мүмкіндігі зор іс-шаралар (Кесте 5.2).

К е с т е 5.2 – Бағдарламаны сату

Маркетинг–микс элементтері	Қажетті істер мен талдаулар
Product	
Электрэнергетика жүйесіне арналған web–қосымша қызметі	Жалпы тапсырманы шешу келесі бөлімдерден тұрады: 1) Администраторларға қолайлы интерфейс жүйесін ұйымдастыру. 2) Мекеме дәрежесіндегі қолданушыларға бағдарламаны бейімдеу.
Price	
Базалық бағасы	Мекенің қажеттілік ерекшеліктерін ескере отырып ақысы аланады.
Promotion	
	Бағдарлама компаниясында қолданылады және осы салада жұмыс істейтін компанияларға ұсынылады.
Place	
Мекеме 1, Мекеме 2, Мекеме 3, Мекеме 4	Алматы қаласындағы осы мекеме қызметімен айналысатын мекеме администраторларын хабарландыру, келіссөз жүргізу. Программа толығымен орнатылмағанша қызметімізді жалғастыру.

К

е с
т е
5.2
жа
лға
сы

5.3 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 3300–ге тең, атап өткен q коэффициентінің мәнін 1 – кестеден (А қосымшасынан) таңдап алуға болады. c –бағдарлама күрделілігінің коэффициенті 2 – кестеден (А қосымшасынан), ол менде 1,26 – ге тең, себебі менің жобамның бағдарламалық тілі жоғарғы тілге жатады және күрделілік тобы үшінші топ, ал жаналық дәрежесі бойынша В тобын, яғни типтік шешім қолданған бағдарламаларды дайындау тобына жатқандықтан осы мәндердің қиылысуынан арнайы кестеден алынды. Осы әдіспен базалық көрсеткішті табамыз.

$$Q = 3300 * 1,26 = 4158$$

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.

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

$$T_{ПО} = 24 \text{ адам / сағат.}$$

T_O (мақсат сипаттамасы уақыты) келесі формуламен.

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

$$T_O = 4158 * 1,2 / (50 * 0,8) = 124,74 \text{ адам/сағат}$$

мұндағы B – мақсат есебі өзгерісінің коэффициенті (А қосымшасынан), B коэффициенті мақсат күрделілігіне және өзгеріс санына тәуелді – 1,2-ден 1,5 – ке дейін K – бағдарлама жасаушы біліктілігін ескеретін коэффициент (А қосымшасынан), менің берілген салада жұмыс істеу өтілім 2 жылға дейінгі уақытты құрағандықтан, K коэффициенті мәні – 0,8–ге тең болады.

T_A (алгоритм құруға кеткен уақыт) мына формуламен.

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

$$T_A = 4158/50*0,8=103,95 \text{ адам/сағат}$$

T_{BC} (блок – сұлба құруға кеткен уақыт) T_A сияқты 3 формуламен есептеледі.
 T_H (бағдарламаның тілінде жазуға кеткен уақыт) келесі формуламен анықталады 5.4 формулада берілген.

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

$$T_H = 4158*1,5/(50*0,8)=155,925 \text{ адам / сағат.}$$

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

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

$$T_{II} = 4158/50=83,16 \text{ адам / сағат}$$

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

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

$$T_{OT} = 4158*4.2/50*0,8=436,59 \text{ адам/сағат.}$$

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

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

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

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

$$T = 24+124,74 +103,95 +103,95 +155,925 +83,16 +436,59 +24=928,4 \text{ адам / сағат}$$

5.4 Бағдарламалық қамтаманы өңдеуге кеткен шығынның есептелуі

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

Негізгі еңбек құнын анықтау берілген.

$$Z_{\text{осн}} = t_{\Sigma} * TC / (t_{\text{cp}} * 8), \text{ теңге} \quad (5.9)$$

мұндағы t_{Σ} – 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 01.01.2014 ж. бастап ҚР–да 19966 теңгені құрады . ТС – тарифтік қойылым = $19966/8 = 2495,5$ ке тең болды. t_{Σ} – формуласы бойынша 928,4 – ге тең.

Осылайша негізгі еңбек ақы.

$$Z_{осн} = 928,4 * 19966 / (21 * 8) = 110331,76 \text{ теңге}$$

Қосымша еңбек ақы негізгі еңбек ақының 20%-ын құрайды формуласы бойынша анықталады.

$$Z_{доп} = 0,2 * 110331,76 = 22066,35 \text{ теңге}$$

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

$$\text{ФОТ} = Z_{доп} + Z_{осн} \quad (5.10)$$

$$\text{ФОТ} = 22066,35 + 110331,76 = 132398,11 \text{ тг.}$$

5.5 Қосымша шығындар статьясы

Бағдарламалық қамтамасыз етудегі қосымша шығындар: материалдар мен құрамаға кеткен шығындар (құрылғының өз құны, яғни компьютерлер есепке алынбайды), социалды сақтандыруға аударым, жүкқұжатқа 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 $C_{обор} = 194900$ 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% (бұл мәндер бітіру жұмысының экономика бөлімін орындауға әдістемелік нұсқаулардан алынды Алматы: АЭЖБУ, 2014 – 27 б., 10 – шы бет) құрайды және мына формулалар мен есептеледі формулада берілген.

$$C_{МжК} = 0,015 * C_{обор} \quad (5.11)$$

$$C_{\text{Мжк}} = 0,015 * 194900 = 2923,5 \text{ теңге}$$

$$C_{\text{ТО}} = 0,025 \times C_{\text{обор}} \quad (5.12)$$

$$C_{\text{ТО}} = 0,025 * 194900 = 4872,5 \text{ теңге}$$

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, формуласы бойынша есептеледі.

$$A = \frac{C_{\text{обор}} \times H_A \times N}{100 \times 12 \times t} \quad (5.13)$$

мұндағы $C_{\text{обор}}$ – компьютер құны, H_A – амортизация нормасы, N – жұмыс орындалуына кеткен күннің саны.

t – дербес компьютерді қолдануға кеткен жалпы уақыт.

$$N = T/24 \quad (5.14)$$

$$928,4 / 24 = 38,68 \text{ күн}$$

бұл жерде T еңбек шығынына тең, оның мәні – 928,4 адам/сағат тең болған.

Дербес компьютерде жалпы жұмыс істеу уақыты мына формуламен есептеледі.

$$T = T_A + T_{\text{БС}} + T_H + T_{\text{П}} + T_{\text{ОТ}} \quad (5.15)$$

$$T = 103,95 + 103,95 + 155,925 + 83,16 + 436,59 = 883,575 \text{ адам/сағат}$$

Амортизация нормасы формуласы бойынша анықталады.

$$H_A = \frac{C_{\text{обор}} - C_{\text{ликв}}}{T_{\text{норм}} \cdot C_{\text{обор}}} \cdot 100\%, \quad (5.16)$$

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

$$C_{\text{ликв}} = 0,05 * C_{\text{обор}} \quad (5.17)$$

$$C_{\text{ликв}} = 0,05 * 194900 = 9745 \text{ теңге}$$

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

$$H_A = \frac{C_{\text{обор}} - C_{\text{ликв}}}{T_{\text{норм}} \cdot C_{\text{обор}}} \cdot 100\% \quad (5.18)$$

$$H_A = \frac{194900 - 9745}{4 \cdot 194900} \cdot 100\% = 23,75\%$$

$$A = \frac{C_{\text{обор}} \times H_A \times N}{100 \times 12 \times t} \quad (5.19)$$

$$A = 194900 \cdot 23,75 \cdot 23,64 / 100 \cdot 12 \cdot 883,575 = 103,2 \text{ теңге}$$

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

$$C_{\text{ЭЭ}} = M \cdot k_3 \cdot F_{\text{эф}} \cdot C_{\text{кВт-ч}} \quad (5.20)$$

Мұндағы M – ЭВМ қуаты (450 Вт.), k_3 – жүктеу коэффициенті (0,8), $C_{\text{кВт-ч}}$ – 1 кВт–сағат электроэнергияның құны, оның мәні ҚР – да 14,65 теңгеге тең.

$F_{\text{эф}}$ – жұмыс уақытының эффективті фонды, мына формула бойынша есептеледі.

$$F_{\text{эф}} = D_{\text{ном}} \cdot d \cdot (1 - f / 100) \quad (5.21)$$

$$F_{\text{эф}} = 258 \cdot 8 \cdot (1 - 2,6 / 100) = 2010$$

Мұндағы $D_{\text{ном}} = 258$ – бір жылдағы жұмыс күнінің номиналды саны, $d = 8$ – жұмыс күнінің ұзақтығы, $f = 2.6\%$ – ЭВМ – ды жөндеуге кеткен жоспарланған уақыт. Берілген коэффициенттер мен параметрлер бойынша электроэнергияның құны.

$$C_{\text{ЭЭ}} = 0,45 \cdot 0,8 \cdot 2010 \cdot 14,65 = 10601 \text{ теңге}$$

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

$$\text{Э}_3 = t_{\Sigma} \cdot C_{\text{Э}} / F_{\text{эф}}, \text{ теңге} \quad (5.22)$$

мұндағы $C_{\text{Э}} = C_{\text{ЭЭ}} + C_{\text{ТО}} + A_{\text{жыл}}$ – ЭЕМ-ның жалпы жылдық эксплуатациондық шығын құны.

$$C_3 = C_{33} + C_{TO} + A_{жыл} \quad (5.23)$$

$$C_3 = 10601 + 4872,5 + 103,2 = 15576,7 \text{ теңге}$$

мұндағы $F_{эф}$ – жұмыс уақытының эффективті фонды, формуласы бойынша анықталды, – t_{Σ} – ЭВМ-нің формуласы бойынша анықталған есепті шешу үшін жалпы қолданылған уақыты, бірақ бұл жерде тек компьютердегі жұмыс уақытын ескере отырып анықталған.

$$t_{\Sigma} = t_{пр} + t_{отл} + t_d \quad (5.24)$$

$$t_{\Sigma} = 83,16 + 436,59 + 24 = 543,75 \text{ сағат}$$

Сәйкесінше, ЭЕМ-ның жалпы жылдық эксплуатационды шығын құны.

$$\Theta_3 = 543,75 * 15576,7 / 2010 = 4213,84 \text{ теңге}$$

ал уақыт коэффициенті мына формула бойынша есептеледі.

$$w = \frac{\Theta_3}{C_3} \quad (5.25)$$

$$w = \frac{4213,84}{15576,7} = 0,27$$

Осылайша, жалпы эксплуатационды шығынды уақыт коэффициентін ескеріп коррекциялаймыз (дұрыстаймыз).

Электроэнергияға шығын.

$$C_{33}^* = C_{33} \cdot w \quad (5.26)$$

$$C_{33}^* = 10601 * 0,27 = 2862,27 \text{ теңге}$$

Амортизационды аударым.

$$A_{жыл}^* = A_{жыл} \cdot w \quad (5.27)$$

$$A_{жыл}^* = 103,2 * 0,27 = 27,864 \text{ теңге}$$

Сонымен қатар, формуласы бойынша есептелетін жұмыс жалақысына тәуелді шығындар бар. Осындай шығындар қатарына социалды сақтандыруға кететін аударымдар мен жүктеме аударымдар кіреді. Әлеуметтік салыққа кететін аударымдар

бүкіл жалақының 11% құрайды (11%, ҚР ның Салық Кодексі), формуласы бойынша анықталады.

$$CC = (\text{ФОТ} - 0,1*\text{ФОТ})*0,11 \quad (5.28)$$

$$CC = (132398,11 - 0,1*132398,11)*0,11=13107,41 \text{ теңге}$$

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 анықталады.

$$C_{\text{накл}} = 0,5*\text{ФОТ} \quad (5.29)$$

$$C_{\text{накл}} = 0,5*132398,11 = 66199,055 \text{ теңге}$$

5.6 Өзіндік құн нәтижесінің кестесі

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 төменгі кестеде көрсетілген.

К е с т е 5.3 – Өзіндік құн нәтижесінің

Шығын статьялары		Сумма, теңге	Жалпы суммадан(%)
ФЗП	З _о сн	110331,76	49,6
	З _д оп	22066,35	9,9
Жүктеме шығындар, C _{накл}		66199,055	29,8
Әлеуметтік сақтандыру, CC		13107,41	5,9
Эксплуатационды шығындар	* ЭЭ С	2862,27	1,29
	ТО С	4872,5	2,2
	* ЖЫЛ А	27,864	0,01
Материалдар мен комплектілер, C _{МиК}		2923,5	1,3
Барлығы:		222390,709	100

5.7 Бағдарламаның бағасына есептеу

Өнімнің өзіндік құны 222390,709 теңге болды, онда бір енгізу объектісіне арналған жүйенің толық өзіндік құны ($C_{пс}$).

$$C_{пс} = 222390,709 / 4 = 55597,9 \text{ теңге}$$

5.8 Минималды бағаны, кірісті және пайданы есептеу

Өнімнің минималды бағасы келесі формуламен есептеледі.

$$Ц_{\min} = (C_{пс} + Ц_{мд} + Ц_3)(1 + r/100) \quad (5.30)$$

$$Ц_{\min} = (55597,9 + 190 + 140) * 1,21 = 67672,5 \text{ теңге}$$

Мұндағы магнитті диск бағасы, қазіргі нарықтық бағасын қарастырып, 190 теңге деп алынды;

$Ц_3$ – жазба бағасы;

r – Өзіндік құнға қатынасымен алынған тиімділіктің жоспарланған пайызы, ол

$r = 21\%$;

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

$$Ц_{\min_{НДС}} = Ц_{\min} + N_{ндс} Ц_{\min} \quad (5.31)$$

$$Ц_{\min_{НДС}} = 67672,5 + 0,12 * 67672,5 = 75793,2 \text{ теңге}$$

Кіріс келесі формуламен есептеледі.

$$V_{\text{реал}} = Ц_{\min_{НДС}} * V_{\text{объем}} \quad (5.32)$$

$$V_{\text{реал}} = 75793,2 * 4 = 303172,8 \text{ теңге}$$

мұндағы $V_{\text{объем}}$ – бағдарламаны жүзеге асыру көлемі; Жалпы пайда келесі формуламен есептеледі.

$$П = V_{\text{реал}} - V_{\text{реал}} * N_{ндс} - C_{\text{разраб}} \quad (5.33)$$

$$П = 303172,8 - 36380,7 - 222390,709 = 44401,4 \text{ теңге}$$

мұндағы НДС – қосылған құн салығы (12%). Таза пайда келесі формуламен есептеледі.

$$П_{\text{чист}} = П - Н_{\text{прибыль}} \quad (5.34)$$

$$П_{\text{чист}} = 44401,4 - 0,2 * 44401,4 = 35521,12 \text{ теңге}$$

мұндағы $N_{\text{прибыль}}$ – пайдаға салынатын салық (ҚР да $N_{\text{прибыль}}=20\%$).

5.9 Бағдарламалық өнімді иемденудегі кәсіпорындардың бірмезгілдегі шығындарын есептеу

Кәсіпорындардың бірмезгілдегі шығындары келесі бөлімдерден тұрады:

- жүйе құны;
- көлік шығындары;
- оқуға кететін төлемдер.

Жүйе құны 75793,2 теңге. Көлік шығындары жүйе құнының 20% құрайды $75793,2 * 0,2 = 15158,64$ теңге.

Жұмысшыларды жүйемен жұмыс істеуге оқыту шамамен екі сағатта консалтингті фирманың маманы көмегімен сағатына 2500 – 3000 теңгедей болып табылады (бұл мән бітіру жұмысының экономика бөлімін орындауға әдістемелік нұсқаулардан алынды, Алматы: АЭЖБУ, 2014 – 27 б., 11 бет).

Соған сәйкес оқу бағасы $3000 * 2 = 6000$ теңге.

Кесте 5.4 – кәсіпорынның бірмезгілдегі шығындарын есептеулерінің нәтижесі көрсетілген.

Кесте 5.4 – Ақпаратты жүйені енгізуге арналған бірмезгілдегі шығындардың есептелу нәтижесі

К е с т е 5.4 – Шығын түрлері

Шығындар түрлері	Сумма, теңге
Жүйе құны	75793,2
Көлік шығындары	15158,64
Жұмысшыларды оқыту бағасы	6000
Қорытынды:	96951,84

5.10 Қолдану саласындағы жылдық бірмезгілдегі шығындарды есептеу

Негізгі шығындар түрлері: жұмысшы еңбегіне төленетін ақша қоры, әлеуметтік салық, басқада шығындар.

Есептеу жылына бір рет ПК операторымен 2500 теңге/сағат (жоғарыда берілген) сағаттық мөлшерде үш сағат көлемінде жүргізіледі $2500 * 3 = 7500$ теңге.

Әлеуметтік салыққа жұмсалатын аударымдар (ҚР да әлеуметтік салық мөлшері 11% жұмысшы еңбегіне төленетін ақша қорынан, ҚР Салық Кодексі) $(7500 - 7500 * 0,1) * 0,11 = 742,5$ теңге . Электроэнергияға кететін шығындар $2,4 * 14,65 = 35,16$ теңге. Мұндағы 2,4 – жұмсалған қуат мөлшері, кВт – сағат, 14,65 тг. ҚР 1 кВт/с электроэнергияның орташа құны.

Бағдарламаны енгізу арқылы қағаз бастылықты азайтады. Бір кафедра есеп беру үшін 10 парағын (А4) жұмсайды. Нарықта бір парақтың құны 2,5 теңге . Сонда бізде $10 * 2,5 = 25$ теңге болады бір рет сабақтарды бөлгенде. Кәсіпорынның жылдық бірмезгілдегі шығындарын есептеу нәтижесі кестеге енгіземіз (Кесте 5.5).

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

Шығындар түрлері	Сумма, теңге
Еңбекке төленетін қор	7500
Әлеуметтік салыққа аударымдар	742,5
Электроэнергияға төлем	35,16
Қағазға жұмсалған ақша	25
Қорытынды:	8302,66

5.11 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 бірмезгілдік 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 1600 теңге/сағат (ұйымның қойып отырған талабы) мөлшерінде жұмыс істейтін мамандандырылған

инженер қажет. Өткізу қабілетін есептеу үшін инженер 5 күн жұмсайды, бір күнде 8 сағаттан жұмыс жасайды. Қолмен есептеу барысында еңбектің жылдық төлем қоры $ФОТ = 5 \cdot 8 \cdot 1600 = 64000$ теңге

Әлеуметтік салыққа жұмсалатын аударымдар (ҚР да әлеуметтік салық мөлшері 11% жұмысшы еңбегіне төленетін ақша қорынан, ҚР Салық Кодексі) $(64000 - 64000 \cdot 0,1) \cdot 0,11 = 6336$ теңге.

Бір кафедра есеп беру үшін 60 парағын (А4) жұмсайды. Нарықта бір парақтың құны 2,5 теңге. Сонда бізде $60 \cdot 2,5 = 150$ теңге болады бір рет сабақтарды бөлгенде. Шығындар түрлерін (Кесте 5.6) енгіземіз.

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

Шығындар түрлері	Сумма, теңге
Еңбекке төленетін қор	64000
Әлеуметтік салыққа аударымдар	6336
Қағазға жұмсалған ақша	150
Қорытынды:	70486

Ақпараттық жүйені қолданғандағы жылдық шығындар жоғарыда есептелді және 8302,66 теңге болды. Осыған сәйкес, шығындардың шартты үнемделуі мынаған тең болады $\Delta \mathcal{E}_k = 70486 - 8302,66 = 62183,34$ теңге.

5.12 Пайда мен шығындарды есептеу

К е с т е 5.7 – Бір рет есептегендегі пайда мен шығындарды есептеу

Аты	Жыл
Шығындардың шартты үнемделуі, теңге	62183,34
Ағымдағы шығындар, теңге	8302,66
Шартты пайда, теңге	44401,4
Пайдаға салынатын салық, теңге	8880,28
Таза пайда, теңге	35521,12

5.13 Ақшалай құралдардың қозғалысы

К е с т е 5.8 – Ақшалай тәсілдердің қозғалысы, теңге

Аты	Жылдар				
	2014	2015	2016	2017	2018
Бірмезгілдік шығындар, КВ мың, теңге	96951,84				

Жеке кәсіпкерлік пайда		35521,12	35521,12	35521,12	35521,12
Шығындарды үнемдеу		62183,34	62183,34	62183,34	62183,34
Ағымдағы шығындар		8302,66	8302,66	8302,66	8302,66
Пайдаға салынатын салық		12436,668	12436,668	12436,668	12436,668

Кес т е 5.8 жалғасы

Таза табыс	-96951,84	49746,672	49746,672	49746,672	49746,672
Дисконттау коэффициенті (20% мөлшерінде)	1	0,833	0,694	0,578	0,482
Таза дисконтталған табыс (ТДТ)	-96951,84	41438	34524	28753	23977
ТДТ өспелі нәтижесімен	-96951,84	-55513,84	-20989,84	7763,16	31740,16

Дисконттау коэффициентің формуласы

$$K_{д2} = 1/(1+\alpha)^1 = 1/(1+0,2)^1 \quad (5.35)$$

$$K_{д3} = 1/(1+\alpha)^2 = 1/(1+0,2)^2 \quad (5.36)$$

$$K_{д4} = 1/(1+\alpha)^3 = 1/(1+0,2)^3 \quad (5.37)$$

$$K_{д5} = 1/(1+\alpha)^4 = 1/(1+0,2)^4 \quad (5.38)$$

$$K_{д2} = 0,833 \quad K_{д3} = 0,694 \quad K_{д4} = 0,578 \quad K_{д5} = 0,482$$

мұндағы α – инфляция коэффициенті, K_d – дисконттау коэффициенті

Таза дисконтталған табыс (ТДТ) = Таза табыс* Дисконттау коэффициенті (5.39)

Таза дисконтталған табыс (ТДТ) =

$$49746,672 * 0,833 = 41438 \text{ теңге} \quad 49746,672 * 0,694 = 34524 \text{ теңге}$$

49746,672*0,578=28753 теңге 49746,672*0,482=23977 теңге

$$ТДТ_2 \text{ (өспелі нәтижесімен)} = ТДТ_1 + \text{Таза дисконтталған табыс (ТДТ)}_2 \quad (5.40)$$

5.14 Экономикалық тиімділікті есептеу. Таза ағымдағы құндылықты есептеу (Net present value, NPV)

Шығындары бізмезгілде тек жобаның басында күрделі салымдар (C_0) ретінде іске асатын жобалар үшін NPV келесі формуламен есептеледі.

$$NPV = \sum_{i=1}^n \frac{B_i}{(1+r)^i} - C_0. \quad (5.41)$$

мұндағы B_i – i -ші жылдағы жобадан алынатын пайда, r – дисконттеу мөлшері.

$$NPV(20\%) = \left(\frac{49746.672}{(1+0.2)^1} + \frac{49746.672}{(1+0.2)^2} + \frac{49746.672}{(1+0.2)^3} + \frac{49746.672}{(1+0.2)^4} \right) - 96951,84 = 31830$$

теңге > 0 сондықтан ұсынылып отырған жоба табысты. Сол себептен жобаны қажет және жобаны талдау мен оған баға беруді жалғастыру керек.

5.15 Пайда индексін есептеу (Profitability index, PI)

PI жобаның салыстырмалы пайдасын көрсетеді. Бұл көрсеткіш келесі формуламен есептеледі.

$$PI = \frac{NPV}{C_0} \quad (5.42)$$

$$PI = (31830 / 222390,709) * 100\% = 14.3 \%$$

5.16 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.

$IRR = r$, мұндағы $NPV = 0$.

$$IRR = r_1 + \frac{f(r_1)}{f(r_1) - f(r_2)} * (r_2 - r_1) \quad (5.43)$$

$$r_b = 25 \% \quad (5.44)$$

$$PV_1 = 49746,672 / (1 + 0,25) = 39800 \text{ тенге}$$

$$PV_2 = 49746,672 / (1 + 0,25)^2 = 31840 \text{ тенге}$$

$$PV_3 = 49746,672 / (1 + 0,25)^3 = 25470 \text{ тенге}$$

$$PV_4 = 49746,672 / (1 + 0,25)^4 = 20380 \text{ тенге}$$

$$NPV(25,0\%) = (39800,896 + 31840 + 25470 + 20380) - 96951,84 = 20540 \text{ тенге}$$

$$NPV(20\%) = 31830 \text{ тенге} \quad NPV(25,0\%) = 20540 \text{ тенге}$$

$$IRR = r_a + (r_b - r_a) * NPV_a / (NPV_a - NPV_b) \quad (5.45)$$

$$IRR = 20 + (25 - 20) * 31830 / (31830 - 20540) = 14,098\%$$

5.17 Өтімділік периодын есептеу (Payback period, PBP)

Өтімділік периоды мына формуламен есептеледі.

$$T_{ок} = t + \frac{C_0 - \sum_{i=1}^n B_i}{B_{i+1}} \quad (5.46)$$

мұндағы C_0 – жоба басындағы салым; B_i – i -ші жылдағы жобадан алынатын пайда; Сонда біздің өтімділік периодымыз.

$$T_{ок} = 3 + \frac{96951,8 - (41438 + 34524)}{28753} = 3 \text{ жыл } 7 \text{ ай}$$

Бұл жобаның өтімділік периоды 3 жыл 7 ай тең, яғни жобамыз осы уақытта өз-өзін ақтайды.

К е с т е 5.9 - q коэффициентінің мәні

Тапсырма түрлері	Коэффициенттің өзгеру аралығы
Септеу тапсырмалары	1400 ден 1500
Оперативті басқару тапсырмалары	1500 ден 1700
Жоспарлау тапсырмалары	3000 ден 3500
Көп вариантты	4500 ден 5000

Комплекстік тапсырма	5000 ден 5500
----------------------	---------------

К е с т е 5.10 – Еңбек сыйымдылығын есептейтін коэффициент

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

К е с т е 5.11 – Бағдарлама жасаушы білектілігін ескеретін коэффициент

Жұмыс тәжірибиесі	Білектілік коэффициенті
Екі жылға дейін	0,8
2-3 жыл	1
3-5 жыл	1,1 – 1,2
5-7 жыл	1,3 – 1,4
7 жылдан көп	1,5 – 1,6

К е с т е 5.12 – NPV мәніне байланысты жоба бойынша есептеу нұсқалары

Көрсеткіш мәнінің нұсқалары	Жоба сипаттамасы
$NPV < 0$	Құрдымға кететін жоба
$NPV = 0$	Құрдымға кету үстіндегі жоба
$NPV > 0$	Табысты жоба

К е с т е 5.13 - IRR және r мәндеріне сәйкес жоба бойынша шешім қабылдау нұсқалары

Көрсеткіш мәнінің нұсқалары	Жоба сипаттамасы	Шешім қабылдау нұсқалары
-----------------------------	------------------	--------------------------

$IRR < r$	Құрдымға кететін жоба	Жобадан бас тарту керек немесе жобаның мәндерін өзгертіп, қайта есептеу жүргізу керек.
$IRR = r$	Құрдымға кету үстіндегі жоба	Жобаны қабылдау қажет және жобаны талдау мен оған баға беруді жалғастыру керек.
$IRR > r$	Табысты жоба	Жобаны қабылдау қажет және жобаны талдау мен оған баға беруді жалғастыру керек.

Қорытынды

Бұл дипломдық жобада барлық қойылған мақсаттар іске асырылды. КТ кафедрасының мұғалімдерінің жүктемесін бөлуге арналып дерекқор құрылды. Дерекқорды құру барысында барлық ерекшеліктер ескерілді.

Дерекқор MS SQL Server дерекқорларды басқару жүйесінде жасалды, жаңа түскен ақпаратқа байланысты деректерді тек құпия сөз арқылы енгізе алады. Мұндай қатынау принципі пайдаланушылар үшін онша ыңғайлы емес, сондықтан олар үшін ASP.NET программалық ортасында клиенттік қосымша жасалды. Сонымен қатар, құпия сөз қолдану арқылы деректердің қауіпсіздік пен құпиялылық мәселелері қарастырылды.

Бұл бағдарламалық өнімді кез келген жоғары оқу орнындағы кез келген кафедрада пайдалануға болады.

Тіршілік қауіпсіздігі KyanushIT компаниясының бөлмесінің ауа алмасу (кондиционер) жүйесін есептедім. Мұнда сыртқы және ішкі микроклимат параметрлерін, температура, ылғалдылық және ауаның қозғалыс жылдамдығы, адам мен компьютерден бөлінетін жылу, зиянды химиялық заттар және радиацияны, бөдме көлемін ескере отырып кафедра бөлмесіне плазмалық Panasonic CS/CU-E 18 NKDW сплит-жүйесі кондиционері таңдалып алынды.

Техника-экономикалық негізделуі нәтижесі бойынша бұл программаны құру ақшалай қаржыландыруды қажет етеді. Бағдарламаның өзіндік құны қосымша құн салығынсыз 222390,709 теңгені қамтиды. Шығындардың көп бөлігі 132398.11 (59.5 %) жалақы төлеуге жұмсалады.

Берілген бағдарламаны қолдану мынадай мүмкіндіктер береді:

- мұғалімдер жүктемесін бөлу барысын жеңілдетеді;
- керекті есептеулерді автоматты түрде жүргізе отырып, уақытты үнемдейді;
- есептеу барысында кететін қателерді азайтады.

Кафедраға арналған бағдарламаны құру қиын әрі ұзақ уақытты қажет етуіне қарамастан мұғалімдер жүктемесін бөлудегі таптырмас бағдарлама болып табылады.

Пайдаланылган әдебиеттер тізімі

- 1 Никита Культин. Основы программирования в ASP.NET 2010. – Санкт-Петербург, 2010. – 448с.
- 2 Алексей Вишневский. SQL Server . Эффективная работа. – Санкт-Петербург, 2009. – 541с.
- 3 Никита Культин. Основы программирования в ASP.NET 2010. 2-е издание. – Санкт-Петербург, 2010. – 640с.
- 4 Полякова Л.Н. Основы SQL. Курс лекций: учеб. пособие. – Москва, 2004. – 368с.
- 5 Глушаков С.В., Ломотько Д.В. Базы данных: Учебный курс. – М.: ООО «Издательство АСТ», 2000. – 504с.
- 6 Ибраева Л.К., Сатимова Е.Г. Проектирование баз данных. МУ к выполнению лабораторных работ. – Алматы: АИЭС, 2004.
- 7 Айтхожаева Е.Ж. Стандартный язык баз данных SQL: Учебное пособие. – Алматы: АИЭС, 2005. - 48 с.
- 8 Дунаев В.В. Базы данных. Язык SQL. – СПб.: БХВ – Петербург, 2006.
- 9 Кандзюба С.П., Громов В.Н. ASP.NET 6/7. Базы данных и приложения. – СПб: ООО «ДиаСофт», 2002.
- 10 Хакимжанов Т.Е. Расчет аспирационных систем. Дипломное проектирование. Для студентов всех форм обучения всех специальностей. – Алматы: АИЭС, 2002. – 30 с.
- 11 Ломакин В.К. Мировая экономика: Учебник для вузов. – М.: Финансы, ЮНИТИ, 2008. – 727 с.
- 12 Борисов, Е. Ф. Основы экономики: Учебное пособие / Е. Ф. Борисов. – М.: Юрайт – Издат, 2009. – 316 с.
- 13 Современная экономика: Учебное пособие /Под ред. О. Ю. Мамедова. – Ростов-на-Дону: Феникс, 2011. – 456 с.
- 14 Экономика: Учебник / Под ред. Р. П. Колосовой. – М.: Норма, 2011. – 345 с.
- 15 Экономика: Учебное пособие /Под ред. А.С. Булатова. – М.: Юристъ, 2009. – 896 с.


```

CREATE TABLE [dbo].[Dissiplini](
    [Дисциплина] [varchar](50) NULL,
    [Группа] [varchar](15) NULL,
    [Кол_во] [int] NULL,
    [Контингент] [int] NULL,
    [Семестр] [int] NULL,
    [Форма_обучения] [varchar](10) NULL,
    [Лекции] [int] NULL,
    [Практика] [int] NULL,
    [Лабораторные] [int] NULL,
    [Консультации] [int] NULL,
    [Экзамены] [int] NULL,
    [Зачеты] [int] NULL,
    [Производственная_практика] [int] NULL,
    [Курсовая_работа] [int] NULL,
    [Дипломная] [int] NULL,
    [Рецензирование] [int] NULL,
    [Аспирантура] [int] NULL,
    [ТАК] [int] NULL,
    [КСРС] [int] NULL,
    [РГР] [int] NULL,
    [Заведование_кафедрой] [int] NULL,
    [Другое] [int] NULL,
    [ID] [int] IDENTITY(1,1) NOT NULL,
    [db_id] [int] NULL)

CREATE TABLE [dbo].[Name_db](
    [id] [int] IDENTITY(1,1) NOT NULL,
    [Polzovatel_id] [int] NULL,
    [Name_db] [nvarchar](50) NULL,
    [date] [nvarchar](50) NULL)

CREATE TABLE [dbo].[Polzovatel](
    [id] [int] IDENTITY(1,1) NOT NULL,
    [login] [varchar](50) NULL,
    [password] [varchar](50) NULL,
    [date] [varchar](50) NULL)

CREATE TABLE [dbo].[Spravochniki_Dissiplini](
    [id] [int] IDENTITY(1,1) NOT NULL,
    [Dissiplina] [varchar](50) NULL,
    [db_id] [int] NULL)

CREATE TABLE [dbo].[Spravochniki_Doljnosti](
    [id] [int] IDENTITY(1,1) NOT NULL,
    [Doljnost] [varchar](30) NULL,
    [db_id] [int] NULL)

CREATE TABLE [dbo].[Spravochniki_Forma_ovuchenia](
    [id] [int] IDENTITY(1,1) NOT NULL,
    [Forma_Obuchenia] [varchar](10) NULL,
    [db_id] [int] NULL)

```

А қосымшасының соңы

```

CREATE TABLE [dbo].[Spravochniki_Gruppi] (
[id] [int] IDENTITY(1,1) NOT NULL,
[Gruppa] [varchar](10) NULL,
[Kolvo_grup] [int] NULL,
[Kolvo_student] [int] NULL,
[Forma_obuchenia] [varchar](10) NULL,
[db_id] [int] NULL)

```

```

CREATE TABLE [dbo].[Spravochniki_Prepodavateli] (
[id] [int] IDENTITY(1,1) NOT NULL,
[FIO] [varchar](50) NULL,
[Doljnost] [varchar](50) NULL,
[db_id] [int] NULL)

```

```

CREATE TABLE [dbo].[Raspredelenie_Nagruzki] (
[ID] [int] IDENTITY(1,1) NOT NULL,
[ID_Преподаватели] [int] NULL,
[Дисциплина] [varchar](50) NULL,
[Группа] [varchar](15) NULL,
[Семестр] [int] NULL,
[Форма_обучения] [varchar](10) NULL,
[Лекции] [int] NULL,
[Практика] [int] NULL,
[Лабораторные] [int] NULL,
[Консультации] [int] NULL,
[Экзамены] [int] NULL,
[Зачеты] [int] NULL,
[Производственная_практика] [int] NULL,
[Курсовая_работа] [int] NULL,|
[Дипломная] [int] NULL,
[Рецензирование] [int] NULL,
[Аспирантура] [int] NULL,
[ГАК] [int] NULL,
[КСРС] [int] NULL,
[РРР] [int] NULL,
[Заведование_кафедрой] [int] NULL,
[Другое] [int] NULL,
[Комментарии] [varchar](100) NULL,
[id_dis] [int] NULL,
[db_id] [int] NULL,

```

Ә қосымшасы

Admin.aspx

```
using System;using System.Data;using System.Collections;using System.Configuration;
using System.Web;using System.Web.Security;using System.Web.UI;using
System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;using System.Web.UI.HtmlControls;
using System.Data.SqlClient;using System.Collections.Generic;using System.Linq;
namespace stys2
{
    public partial class Admin : System.Web.UI.Page
    {
        string conn = ConfigurationManager.ConnectionStrings["connectionDB"].ToString();
        protected void Page_Load(object sender, EventArgs e)
        {
            if (Session["admin_Login"] == null)
            {
                Response.Redirect("~/index.aspx");
            }
            else
            {
                Lb_login.Text = "Добро пожаловать-"+Session["admin_Login"].ToString() + "!";
                if (!Page.IsPostBack)
                {
                    select();
                }
            }
        }
        public void select()
        {
            SqlConnection con = new SqlConnection(conn);
            con.Open();
            SqlDataAdapter da = new SqlDataAdapter("select * from [Polzovatel]", con);
            DataSet ds = new DataSet();
            da.Fill(ds);
            GridView_1.DataSource = ds;
            GridView_1.DataBind();
        }
        protected void GridView_1_RowDeleting(object sender, GridViewDeleteEventArgs e)
        {
            string sql = "Delete Polzovatel Where id=" +
Convert.ToInt32(GridView_1.DataKeys[e.RowIndex].Values[0].ToString());
            SqlConnection con = new SqlConnection(conn);
            con.Open();
            SqlCommand cmd = new SqlCommand(sql, con);
            cmd.ExecuteNonQuery();
            con.Close();
            con.Dispose();

```

Ә қосымшасының жалғасы

```
select();
```

```

    }
    protected void GridView_1_RowCancelingEdit(object sender, GridViewCancelEventArgs
e)
    {
        GridView_1.EditIndex = -1;
        select();
    }
    protected void GridView_1_RowEditing(object sender, GridViewEditEventArgs e)
    {
        GridView_1.EditIndex = e.NewEditIndex;
        select();
    }
    protected void GridView_1_RowUpdating(object sender, GridViewUpdateEventArgs e)
    {
        TextBox txtLogin =
(TextBox)GridView_1.Rows[GridView_1.EditIndex].FindControl("txtLogin");
        TextBox txtPassword =
(TextBox)GridView_1.Rows[GridView_1.EditIndex].FindControl("txtPassword");
        TextBox txtDate =
(TextBox)GridView_1.Rows[GridView_1.EditIndex].FindControl("txtDate");

        string sql = "UPDATE Polzovatel SET login=" + txtLogin.Text + ", password = " +
txtPassword.Text +
        ", date = " + txtDate.Text + " Where id=" +
Convert.ToInt32(GridView_1.DataKeys[e.RowIndex].Values[0].ToString());
        SqlConnection conn1 = new SqlConnection(conn);
        conn1.Open();
        SqlCommand cmd = new SqlCommand(sql, conn1);
        cmd.ExecuteNonQuery();
        conn1.Close();
        conn1.Dispose();
        GridView_1.EditIndex = -1;
        select();
    }
    protected void GridView_1_RowCommand(object sender, GridViewCommandEventArgs e)
    {
        try
        {
            if (e.CommandName.Equals("AddNew"))
            {
                TextBox txtNewLogin =
(TextBox)GridView_1.FooterRow.FindControl("txtNewLogin");
                TextBox txtNewPassword =
(TextBox)GridView_1.FooterRow.FindControl("txtNewPassword");

```

Ә қосымшасының жалғасы

```

TextBox txtNewDate = (TextBox)GridView_1.FooterRow.FindControl("txtNewDate");

```

```

        string sql = "Insert Into [Polzovatel] Values ('" + txtNewLogin.Text + "', '" +
txtNewPassword.Text + "', '" + txtNewDate.Text + "')";
        SqlConnection c = new SqlConnection(conn);
        c.Open();
        SqlCommand cmd = new SqlCommand(sql, c);
        cmd.ExecuteNonQuery();
        c.Close();
        c.Dispose();
        select();
    }
}
catch (System.Exception ex)
{
    string message = "Ошибка:" + ex;
    System.Text.StringBuilder sb = new System.Text.StringBuilder();
    sb.Append("<script type = 'text/javascript'>");
    sb.Append("window.onload=function(){"");
    sb.Append("alert(");
    sb.Append(message);
    sb.Append(")");
    sb.Append("</script>");
    ClientScript.RegisterClientScriptBlock(this.GetType(), "alert", sb.ToString());
}
}
protected void Next_Click(object sender, EventArgs e)
{
    Response.Redirect("~/index.aspx");
    Session["admin_Login"] = null;
}
}
}
}

```

Dissiplina.aspx

```

using System;using System.Data;using System.Collections;using System.Configuration;
using System.Web;using System.Web.Security;using System.Web.UI;using
System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;using System.Web.UI.HtmlControls;
using System.Data.SqlClient;using System.Collections.Generic;using System.Linq;
using System.IO;using System.Data.Linq;using NPOI.HSSF.UserModel;using
NPOI.SS.UserModel;
using NPOI.SS.Util;using NPOI.HSSF.Util;using NPOI.POIFS.FileSystem;
using NPOI.HPSF;using System.Drawing;using Winnovative.ExcelLib;
namespace stys2
{
    public partial class Dissiplina : System.Web.UI.Page
        Ә қосымшасының жалғасы
    {
        string conn = ConfigurationManager.ConnectionStrings["connectionDB"].ToString();

```

```

protected void Page_Load(object sender, EventArgs e)
{
    if (Session["db_name"] == null && Session["db_id"] == null &&
Session["db_Plzovotel_Name"] == null)
    {
        Response.Redirect("~/index.aspx");
    }
    else
    {
        if (!Page.IsPostBack)
        {
            SqlConnection con_sorting = new SqlConnection(conn);
            string com_sorting = "select * from Spravochniki_Dissiplini where db_id = " +
Session["db_id"].ToString();
            SqlDataAdapter adpt_sorting = new SqlDataAdapter(com_sorting, con_sorting);
            DataTable dt_sorting = new DataTable();
            adpt_sorting.Fill(dt_sorting);
            DropDownList1.DataSource = dt_sorting;
            DropDownList1.DataBind();
            DropDownList1.DataTextField = "Dissiplina";
            DropDownList1.DataValueField = "Dissiplina";
            DropDownList1.DataBind();
            DropDownList1.Items.Insert(0, new ListItem("Все дисциплины", "Все
дисциплины"));
            SqlConnection con_form = new SqlConnection(conn);
            string com_form = "select * from Spravochniki_Forma_ovuchenia where db_id = " +
Session["db_id"].ToString();
            SqlDataAdapter adpt_form = new SqlDataAdapter(com_form, con_form);
            DataTable dt_form = new DataTable();
            adpt_form.Fill(dt_form);
            DropDownList3.DataSource = dt_form;
            DropDownList3.DataBind();
            DropDownList3.DataTextField = "Forma_Obuchenia";
            DropDownList3.DataValueField = "Forma_Obuchenia";
            DropDownList3.DataBind();
            DropDownList3.Items.Insert(0, new ListItem("Все формы", "Все формы"));
            SqlConnection con_grup = new SqlConnection(conn);
            string com_grup = "select * from Spravochniki_Gruppi where db_id = " +
Session["db_id"].ToString();
            SqlDataAdapter adpt_grup = new SqlDataAdapter(com_grup, con_grup);
            DataTable dt_grup = new DataTable();
            adpt_grup.Fill(dt_grup);
            DropDownList4.DataSource = dt_grup;
            DropDownList4.DataBind();
            DropDownList4.DataTextField = "Gruppa";
            DropDownList4.DataValueField = "Gruppa";
            DropDownList4.DataBind();
            DropDownList4.Items.Insert(0, new ListItem("Все группы", "Все группы"));
        }
    }
}

```

Ә қосымшасының жалғасы

```

        FillCustomerInGrid();
    }
}
private void FillCustomerInGrid()
{
    DataTable dtCustomer = Fetch();
    GridView_1.DataSource = dtCustomer;
    GridView_1.DataBind();
    GridView1.DataSource = dtCustomer;
    GridView1.DataBind();
    if (dtCustomer.Rows.Count > 0)
    {
        GridView_1.DataSource = dtCustomer;
        GridView_1.DataBind();
        sortirovka();
    }
    else
    {
        SqlConnection con_sem_1 = new SqlConnection(conn);
        con_sem_1.Open();
        SqlDataAdapter da_sem_1 = new SqlDataAdapter(@"SELECT '' as [Дисциплина]
,' ' as [Группа] ,'' as [Кол_во] ,'' as [Контингент] ,'' as [Семестр] ,'' as [Форма_обучения]
,' ' as [Лекции]
,' ' as [Практика] ,'' as [Лабораторные] ,'' as [Консультации] ,'' as [Экзамены]
,' ' as [Зачеты] ,'' as [Производственная_практика] ,'' as [Курсовая_работа]
,' ' as [Дипломная] ,'' as [Рецензирование] ,'' as [Аспирантура]
,' ' as [ГАК] ,'' as [КСРС] ,'' as [РГР] ,'' as [Заведование_кафедрой]
,' ' as [Другое] ,'' as [Всего] ,'' as [ID] ,'' as [db_id]", con_sem_1);
        DataSet ds_sem_1 = new DataSet();
        da_sem_1.Fill(ds_sem_1);
        GridView_1.DataSource = ds_sem_1;
        GridView_1.DataBind();
    }
}
protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
{ sortirovka();}
protected void DropDownList2_SelectedIndexChanged(object sender, EventArgs e)
{sortirovka();}
protected void DropDownList3_SelectedIndexChanged(object sender, EventArgs e)
{sortirovka();}

```

Ә қосымшасының жалғасы

```

protected void DropDownList4_SelectedIndexChanged(object sender, EventArgs e)
{sortirovka();}
public void sortirovka()
{

```

```

string sum_1_2 = ",sum(ISNULL(Лекции,0)+ ISNULL([Практика],0)
+ISNULL([Лабораторные],0)+
ISNULL([Консультации],0)+ISNULL([Экзамены],0)+ISNULL([Зачеты],0)+ISNULL([Производствен
ная_практика],0)+ISNULL([Курсовая_работа],0)+ISNULL([Дипломная],0)+ISNULL([Рецензирова
ние],0)+ISNULL([Аспирантура],0)+ISNULL([ГАК],0)+ISNULL([КСРС],0)+ISNULL([РГР],0)+ISNULL(
[Заведование_кафедрой],0)+ISNULL([Другое],0)) as [Всего]";
string sql0 = "SELECT * ,ISNULL(Лекции,0)+ ISNULL([Практика],0)
+ISNULL([Лабораторные],0)+
ISNULL([Консультации],0)+ISNULL([Экзамены],0)+ISNULL([Зачеты],0)+ISNULL([Производствен
ная_практика],0)+ISNULL([Курсовая_работа],0)+ISNULL([Дипломная],0)+ISNULL([Рецензирова
ние],0)+ISNULL([Аспирантура],0)+ISNULL([ГАК],0)+ISNULL([КСРС],0)+ISNULL([РГР],0)+ISNULL(
[Заведование_кафедрой],0)+ISNULL([Другое],0) as [Всего],ID From Dissiplini where db_id=" +
Session["db_id"].ToString() + " and [Дисциплина]=";
string sql = null, sql1 = null, sql2 = null, sql3 = null;
string sem_1 = null; string sem_2 = null; string sem_3 = null;
if (DropDownList1.SelectedValue.ToString() != "Все дисциплины")
{
    sql = sql0 + " " + DropDownList1.SelectedValue.ToString() + " ";
    sem_1 = " SELECT '1 семестр' as ' ',sum([Лекции]) as 'Лекции' ,sum([Практика]) as
'Прак' ,sum([Лабораторные])as 'Лаб' ,sum([Консультации])as 'Конс' ,sum([Экзамены])as
'Экзамены' ,sum([Зачеты]) as 'Зачеты' ,sum([Производственная_практика]) as 'Пр.Прак'
,sum([Курсовая_работа])as 'Курс.раб' ,sum([Дипломная])as 'Дип' ,sum([Рецензирование])as
'Рец' ,sum([Аспирантура])as 'Асп' ,sum([ГАК])as 'ГАК' ,sum([КСРС])as 'КСРС'
,sum([РГР])as 'РГР' ,sum([Заведование_кафедрой])as 'Зав.каф' ,sum([Другое])as 'Другое'" +
sum_1_2 + " From Dissiplini where Семестр=1 and [Дисциплина]=" +
DropDownList1.SelectedValue.ToString() + " and db_id = " + Session["db_id"].ToString();
    sem_2 = " SELECT '2 семестр' as ' ',sum([Лекции]) as 'Лекции' ,sum([Практика]) as
'Прак' ,sum([Лабораторные])as 'Лаб' ,sum([Консультации])as 'Конс' ,sum([Экзамены])as
'Экзамены' ,sum([Зачеты]) as 'Зачеты' ,sum([Производственная_практика]) as 'Пр.Прак'
,sum([Курсовая_работа])as 'Курс.раб' ,sum([Дипломная])as 'Дип' ,sum([Рецензирование])as
'Рец' ,sum([Аспирантура])as 'Асп' ,sum([ГАК])as 'ГАК' ,sum([КСРС])as 'КСРС'
,sum([РГР])as 'РГР' ,sum([Заведование_кафедрой])as 'Зав.каф' ,sum([Другое])as 'Другое'" +
sum_1_2 + " From Dissiplini where Семестр=2 and [Дисциплина]=" +
DropDownList1.SelectedValue.ToString() + " and db_id = " + Session["db_id"].ToString();
    sem_3 = " SELECT 'за год' as ' ',sum([Лекции]) as 'Лекции' ,sum([Практика]) as 'Прак'
,sum([Лабораторные])as 'Лаб' ,sum([Консультации])as 'Конс' ,sum([Экзамены])as 'Экзамены'
,sum([Зачеты]) as 'Зачеты' ,sum([Производственная_практика]) as 'Пр.Прак'
,sum([Курсовая_работа])as 'Курс.раб' ,sum([Дипломная])as 'Дип' ,sum([Рецензирование])as
'Рец' ,sum([Аспирантура])as 'Асп' ,sum([ГАК])as 'ГАК' ,sum([КСРС])as 'КСРС'
,sum([РГР])as 'РГР' ,sum([Заведование_кафедрой])as 'Зав.каф' ,sum([Другое])as 'Другое'" +
sum_1_2 + "

```

Ә қосымшасының жалғасы

```

From Dissiplini where [Дисциплина]=" + DropDownList1.SelectedValue.ToString() + " and
db_id = " + Session["db_id"].ToString();
}
if (DropDownList1.SelectedValue.ToString() == "Все дисциплины")
{
    sql = sql0 + " [Дисциплина] ";

```



```

sem_1 = " SELECT '1 семестр' as ' ',sum([Лекции]) as 'Лекции' ,sum([Практика]) as
'Прак' ,sum([Лабораторные])as 'Лаб' ,sum([Консультации])as 'Конс' ,sum([Экзамены])as
'Экзамены' ,sum([Зачеты]) as 'Зачеты' ,sum([Производственная_практика]) as 'Пр.Прак'
,sum([Курсовая_работа])as 'Курс.раб' ,sum([Дипломная])as 'Дип' ,sum([Рецензирование])as
'Рец' ,sum([Аспирантура])as 'Асп' ,sum([ГАК])as 'ГАК' ,sum([КСРС])as 'КСРС'
,sum([РГР])as 'РГР' ,sum([Заведование_кафедрой])as 'Зав.каф' ,sum([Другое])as 'Другое'" +
sum_1_2 + " From Dissiplini where Семестр=1 and db_id = " + Session["db_id"].ToString();
sem_2 = " SELECT '2 семестр' as ' ',sum([Лекции]) as 'Лекции' ,sum([Практика]) as
'Прак' ,sum([Лабораторные])as 'Лаб' ,sum([Консультации])as 'Конс' ,sum([Экзамены])as
'Экзамены' ,sum([Зачеты]) as 'Зачеты' ,sum([Производственная_практика]) as 'Пр.Прак'
,sum([Курсовая_работа])as 'Курс.раб' ,sum([Дипломная])as 'Дип' ,sum([Рецензирование])as
'Рец' ,sum([Аспирантура])as 'Асп' ,sum([ГАК])as 'ГАК' ,sum([КСРС])as 'КСРС'
,sum([РГР])as 'РГР' ,sum([Заведование_кафедрой])as 'Зав.каф' ,sum([Другое])as 'Другое'" +
sum_1_2 + " From Dissiplini where Семестр=2 and db_id = " + Session["db_id"].ToString();
sem_3 = " SELECT 'за год' as ' ',sum([Лекции]) as 'Лекции' ,sum([Практика]) as 'Прак'
,sum([Лабораторные])as 'Лаб' ,sum([Консультации])as 'Конс' ,sum([Экзамены])as 'Экзамены'
,sum([Зачеты]) as 'Зачеты' ,sum([Производственная_практика]) as 'Пр.Прак'
,sum([Курсовая_работа])as 'Курс.раб' ,sum([Дипломная])as 'Дип' ,sum([Рецензирование])as
'Рец' ,sum([Аспирантура])as 'Асп' ,sum([ГАК])as 'ГАК' ,sum([КСРС])as 'КСРС'
,sum([РГР])as 'РГР' ,sum([Заведование_кафедрой])as 'Зав.каф' ,sum([Другое])as 'Другое'" +
sum_1_2 + " From Dissiplini where id=id and db_id = " + Session["db_id"].ToString();

```

```

}
if (DropDownList2.SelectedValue.ToString() != "Все семестры")
sql1 = " and [Семестр]=" + DropDownList2.SelectedValue.ToString();
if (DropDownList2.SelectedValue.ToString() == "Все семестры")
sql1 = " and [Семестр]=[Семестр]";
if (DropDownList3.SelectedValue.ToString() != "Все формы")
sql2 = " and [Форма_обучения]=" + DropDownList3.SelectedValue.ToString() + """;
if (DropDownList3.SelectedValue.ToString() == "Все формы")
sql2 = " and [Форма_обучения]=[Форма_обучения]";
if (DropDownList4.SelectedValue.ToString() != "Все группы")
sql3 = " and [Группа]=" + DropDownList4.SelectedValue.ToString() + """;
if (DropDownList4.SelectedValue.ToString() == "Все группы")
sql3 = " and [Группа]=[Группа]";
SqlConnection con = new SqlConnection(conn);
con.Open();
SqlDataAdapter da = new SqlDataAdapter(sql + sql1 + sql2 + sql3, con);

```

Ә қосымшасының жалғасы

```

DataSet ds = new DataSet();
da.Fill(ds);
GridView_1.DataSource = ds;
GridView_1.DataBind();
GridView1.DataSource = ds;
GridView1.DataBind();
con.Close();
SqlConnection con_sem_1 = new SqlConnection(conn);
con_sem_1.Open();

```

```

        SqlDataAdapter da_sem_1 = new SqlDataAdapter(sem_1 + sql1 + sql2 + sql3 + " UNION
ALL " + sem_2 + sql1 + sql2 + sql3 + " UNION ALL " + sem_3 + sql1 + sql2 + sql3, con_sem_1);
        DataSet ds_sem_1 = new DataSet();
        da_sem_1.Fill(ds_sem_1);
        GridView2.DataSource = ds_sem_1;
        GridView2.DataBind();
        con_sem_1.Close();
    }
    protected void GridView_1_RowCommand(object sender, GridViewCommandEventArgs e)
    {
        if (e.CommandName.Equals("AddNew"))
        {
            DropDownList cmbNewДисциплина =
            (DropDownList)GridView_1.FooterRow.FindControl("cmbNewДисциплина");
            DropDownList cmbNewГруппа =
            (DropDownList)GridView_1.FooterRow.FindControl("cmbNewГруппа");
            TextBox txtNewКол_во =
            (TextBox)GridView_1.FooterRow.FindControl("txtNewКол_во");
            TextBox txtNewКонтингент =
            (TextBox)GridView_1.FooterRow.FindControl("txtNewКонтингент");
            DropDownList cmbNewСеместр =
            (DropDownList)GridView_1.FooterRow.FindControl("cmbNewСеместр");
            TextBox cmbNewФорма_обучения =
            (TextBox)GridView_1.FooterRow.FindControl("txtNewФорма_обучения");
            TextBox txtNewЛекции =
            (TextBox)GridView_1.FooterRow.FindControl("txtNewЛекции");
            TextBox txtNewПрактика =
            (TextBox)GridView_1.FooterRow.FindControl("txtNewПрактика");
            TextBox txtNewЛабораторные =
            (TextBox)GridView_1.FooterRow.FindControl("txtNewЛабораторные");
            TextBox txtNewКонсультации =
            (TextBox)GridView_1.FooterRow.FindControl("txtNewКонсультации");
            TextBox txtNewЭкзамены =
            (TextBox)GridView_1.FooterRow.FindControl("txtNewЭкзамены");
            TextBox txtNewЗачеты =
            (TextBox)GridView_1.FooterRow.FindControl("txtNewЗачеты");

```

Ә қосымшасының жалғасы

```

            TextBox txtNewПроизводственная_практика =
            (TextBox)GridView_1.FooterRow.FindControl("txtNewПроизводственная_практика");
            TextBox txtNewКурсовая_работа =
            (TextBox)GridView_1.FooterRow.FindControl("txtNewКурсовая_работа");
            TextBox txtNewДипломная =
            (TextBox)GridView_1.FooterRow.FindControl("txtNewДипломная");
            TextBox txtNewРецензирование =
            (TextBox)GridView_1.FooterRow.FindControl("txtNewРецензирование");
            TextBox txtNewАспирантура =
            (TextBox)GridView_1.FooterRow.FindControl("txtNewАспирантура");

```

```

        TextBox txtNewГАК =
(TextBox)GridView_1.FooterRow.FindControl("txtNewГАК");
        TextBox txtNewКСРС =
(TextBox)GridView_1.FooterRow.FindControl("txtNewКСРС");
        TextBox txtNewПГР = (TextBox)GridView_1.FooterRow.FindControl("txtNewПГР");
        TextBox txtNewЗаведование_кафедрой =
(TextBox)GridView_1.FooterRow.FindControl("txtNewЗаведование_кафедрой");
        TextBox txtNewДругое =
(TextBox)GridView_1.FooterRow.FindControl("txtNewДругое");

        Insert(
            cmbNewДисциплина.SelectedValue,
            cmbNewГруппа.SelectedValue,txtNewКол_во.Text, txtNewКонтингент.Text,
            cmbNewСеместр.SelectedValue,cmbNewФорма_обучения.Text,
            txtNewЛекции.Text,NewПрактика.Text txtNewЛабораторные.Text,
            txtNewКонсультации.Text,txtNewЭкзамены.Text,
            txtNewЗачеты.Text, txtNewПроизводственная_практика.Text,txtNewКурсовая_работа.Text,
            txtNewДипломная.Text, txtNewРецензирование.Text,
            txtNewАспирантура.Text, txtNewГАК.Text, txtNewКСРС.Text,
            txtNewПГР.Text,txtNewЗаведование_кафедрой.Text,
            txtNewДругое.Text, Session["db_id"].ToString()); О_clear();sortirovka();} }
        protected void cmbГруппа_SelectedIndexChanged(object sender, EventArgs e)
        {
            DropDownList cmbГруппа =
(DropDownList)GridView_1.Rows[GridView_1.EditIndex].FindControl("cmbГруппа");
            TextBox txtКол_во =
(TextBox)GridView_1.Rows[GridView_1.EditIndex].FindControl("txtКол_во");
            TextBox txtКонтингент =
(TextBox)GridView_1.Rows[GridView_1.EditIndex].FindControl("txtКонтингент");
            TextBox txtФорма_обучения =
(TextBox)GridView_1.Rows[GridView_1.EditIndex].FindControl("txtФорма_обучения");

            SqlConnection con = new SqlConnection(conn);
            con.Open();
            SqlCommand cmd = new SqlCommand("select * from Spravochniki_Gruppi where
Группа="

                Ә қосымшасының жалғасы

            + cmbГруппа.SelectedValue.ToString() + " and db_id=" + Session["db_id"].ToString(),
            con);

            SqlDataReader dr = cmd.ExecuteReader();
            while (dr.Read())
            {
                txtКол_во.Text = dr["Kolvo_grup"].ToString();
                txtКонтингент.Text = dr["Kolvo_student"].ToString();
                txtФорма_обучения.Text = dr["Forma_obuchenia"].ToString();
            }
            con.Close();

```

```

    }
    protected void cmbNewГруппа_SelectedIndexChanged(object sender, EventArgs e)
    {
        DropDownList cmbNewГруппа =
        (DropDownList)GridView_1.FooterRow.FindControl("cmbNewГруппа");

        TextBox txtNewКол_во =
        (TextBox)GridView_1.FooterRow.FindControl("txtNewКол_во");
        TextBox txtNewКОНТИНГЕНТ =
        (TextBox)GridView_1.FooterRow.FindControl("txtNewКОНТИНГЕНТ");
        TextBox txtNewФорма_обучения =
        (TextBox)GridView_1.FooterRow.FindControl("txtNewФорма_обучения");

        SqlConnection con = new SqlConnection(conn);
        con.Open();
        SqlCommand cmd = new SqlCommand("select * from Spravochniki_Gruppi where
        Группа=" + cmbNewГруппа.SelectedValue.ToString() + " and db_id=" + Session["db_id"].ToString(),
        con);

        SqlDataReader dr = cmd.ExecuteReader();
        while (dr.Read())
        {
            txtNewКол_во.Text = dr["Kolvo_grup"].ToString();
            txtNewКОНТИНГЕНТ.Text = dr["Kolvo_student"].ToString();
            txtNewФорма_обучения.Text = dr["Forma_obuchenia"].ToString();
        }
        con.Close();
    }
    protected void GridView_1_RowDataBound(object sender, GridViewRowEventArgs e)
    {
        if (e.Row.RowType == DataControlRowType.DataRow)
        {
            DropDownList cmbType = (DropDownList)e.Row.FindControl("cmbДисциплина");

            if (cmbType != null)
            {
                cmbType.DataSource = select_Dissiplini();
                Ә қосымшасының жалғасы

                cmbType.DataBind();
                SqlConnection con = new SqlConnection(conn);
                con.Open();
                SqlCommand cmd = new SqlCommand("Select * From Dissiplini where id=" +
                GridView_1.DataKeys[e.Row.RowIndex].Values[0].ToString() + " and db_id=" +
                Session["db_id"].ToString(), con);
                SqlDataReader dr = cmd.ExecuteReader();
                while (dr.Read())
                {
                    cmbType.SelectedValue = dr["Дисциплина"].ToString();
                }
                con.Close();
            }
        }
    }

```



```

{Delete(Convert.ToInt32(GridView_1.DataKeys[e.RowIndex].Values[0].ToString()));
  sortirovka();}
// UPDATE
protected void GridView_1_RowUpdating(object sender, GridViewUpdateEventArgs e)
{

```

```

        DropDownList                cmbДисциплина                =
(DropDownList)GridView_1.Rows[GridView_1.EditIndex].FindControl("cmbДисциплина");
        DropDownList                cmbГруппа                =
(DropDownList)GridView_1.Rows[GridView_1.EditIndex].FindControl("cmbГруппа");
        TextBox                    txtКол_во                =
(TextBox)GridView_1.Rows[GridView_1.EditIndex].FindControl("txtКол_во");
        TextBox                    txtКонтингент            =
(TextBox)GridView_1.Rows[GridView_1.EditIndex].FindControl("txtКонтингент");
        DropDownList                cmbСеместр                =
(DropDownList)GridView_1.Rows[GridView_1.EditIndex].FindControl("cmbСеместр");
        TextBox                    txtФорма_обучения        =
(TextBox)GridView_1.Rows[GridView_1.EditIndex].FindControl("txtФорма_обучения");
        TextBox                    txtЛекции                =
(TextBox)GridView_1.Rows[GridView_1.EditIndex].FindControl("txtЛекции");
        TextBox                    txtПрактика                =
(TextBox)GridView_1.Rows[GridView_1.EditIndex].FindControl("txtПрактика");
        TextBox                    txtЛабораторные            =
(TextBox)GridView_1.Rows[GridView_1.EditIndex].FindControl("txtЛабораторные");
        TextBox                    txtКонсультации            =
(TextBox)GridView_1.Rows[GridView_1.EditIndex].FindControl("txtКонсультации");
        TextBox                    txtЭкзамены                =
(TextBox)GridView_1.Rows[GridView_1.EditIndex].FindControl("txtЭкзамены");
        TextBox                    txtЗачеты                =
(TextBox)GridView_1.Rows[GridView_1.EditIndex].FindControl("txtЗачеты");
        TextBox                    txtПроизводственная_практика =
(TextBox)GridView_1.Rows[GridView_1.EditIndex].FindControl("txtПроизводственная_практика");

```

Ә қосымшасының жалғасы

```

        TextBox                    txtКурсовая_работа        =
(TextBox)GridView_1.Rows[GridView_1.EditIndex].FindControl("txtКурсовая_работа");
        TextBox                    txtДипломная                =
(TextBox)GridView_1.Rows[GridView_1.EditIndex].FindControl("txtДипломная");
        TextBox                    txtРецензирование            =
(TextBox)GridView_1.Rows[GridView_1.EditIndex].FindControl("txtРецензирование");
        TextBox                    txtАспирантура                =
(TextBox)GridView_1.Rows[GridView_1.EditIndex].FindControl("txtАспирантура");
        TextBox                    txtГАК                    =
(TextBox)GridView_1.Rows[GridView_1.EditIndex].FindControl("txtГАК");
        TextBox                    txtКСРС                    =
(TextBox)GridView_1.Rows[GridView_1.EditIndex].FindControl("txtКСРС");
        TextBox                    txtРГР                    =
(TextBox)GridView_1.Rows[GridView_1.EditIndex].FindControl("txtРГР");

```

```

        TextBox                txtЗаведование_кафедрой                =
(TextBox)GridView_1.Rows[GridView_1.EditIndex].FindControl("txtЗаведование_кафедрой");
        TextBox                txtДругое                            =
(TextBox)GridView_1.Rows[GridView_1.EditIndex].FindControl("txtДругое");

```

```

        Update(Convert.ToInt32(GridView_1.DataKeys[e.RowIndex].Values[0].ToString()),
            cmbДисциплина.SelectedValue,
            cmbГруппа.SelectedValue,
            txtКол_во.Text,
            txtКонтингент.Text,
            cmbСеместр.SelectedValue,
            txtФорма_обучения.Text,
            txtЛекции.Text,
            txtПрактика.Text,
            txtЛабораторные.Text,
            txtКонсультации.Text,
            txtЭкзамены.Text,
            txtЗачеты.Text,
            txtПроизводственная_практика.Text,
            txtКурсовая_работа.Text,
            txtДипломная.Text,
            txtРецензирование.Text,
            txtАспирантура.Text,
            txtГАК.Text,
            txtКСРС.Text,
            txtРГР.Text,
            txtЗаведование_кафедрой.Text,
            txtДругое.Text);
        GridView_1.EditIndex = -1;
        O_clear();
        sortirovka();
    }

```

Ә қосымшасының жалғасы

```

public void Update(int id, string Дисциплина, string Группа, string Кол_во, string
Контингент, string Семестр, string Форма_обучения,
    string Лекции, string Практика, string Лабораторные, string Консультации, string Экзамены,
string Зачеты, string Производственная_практика,
    string Курсовая_работа, string Дипломная, string Рецензирование, string Аспирантура, string
ГАК, string КСРС, string РГР,
    string Заведование_кафедрой, string Другое)
    {
        string sql = @"UPDATE Dissiplini SET
            Дисциплина="" + Дисциплина
+ "", Группа = "" + Группа
+ "", Кол_во="" + Кол_во
+ "", Контингент = "" + Контингент
+ "", Семестр = "" + Семестр
+ "", Форма_обучения="" + Форма_обучения

```

```

+ "", Лекции = "" + Лекции
+ "", Практика = "" + Практика
+ "", Лабораторные="" + Лабораторные
+ "", Консультации = "" + Консультации
+ "", Экзамены = "" + Экзамены
+ "", Зачеты="" + Зачеты
+ "", Производственная_практика = "" + Производственная_практика
+ "", Курсовая_работа = "" + Курсовая_работа
+ "", Дипломная="" + Дипломная
+ "", Рецензирование = "" + Рецензирование
+ "", Аспирантура = "" + Аспирантура
+ "", ГАК="" + ГАК
+ "", КСРС = "" + КСРС
+ "", РГР = "" + РГР
+ "", Заведование_кафедрой="" + Заведование_кафедрой
+ "", Другое = "" + Другое
+ "" Where id=" + id;
SqlConnection con = new SqlConnection(conn);
con.Open();
SqlCommand cmd = new SqlCommand(sql, con);
cmd.ExecuteNonQuery();
con.Close();
con.Dispose();
}

```

```

protected void Zvros_Click(object sender, EventArgs e)
{

```

```

    DropDownList1.SelectedIndex =
DropDownList1.Items.IndexOf(DropDownList1.Items.FindByText("Все дисциплины"));
    DropDownList2.SelectedIndex =
DropDownList2.Items.IndexOf(DropDownList2.Items.FindByText("Все семестры"));
    DropDownList3.SelectedIndex =
DropDownList3.Items.IndexOf(DropDownList3.Items.FindByText("Все формы"));
        Ә қосымшасының жалғасы
    DropDownList4.SelectedIndex =
DropDownList4.Items.IndexOf(DropDownList4.Items.FindByText("Все группы"));
    FillCustomerInGrid();

```

```

}
protected void Export_Click(object sender, EventArgs e)
{
    ExcelWorkbookFormat workbookFormat = ExcelWorkbookFormat.Xlsx_2007;
    ExcelWorkbook workbook = new ExcelWorkbook(workbookFormat);
    ExcelWorksheet worksheet = workbook.Worksheets[0];
    worksheet.Name = "Дисциплины";

```

```

#region SET CELLS WITH VARIOUS VALUES AND FORMATTING
int sun = 6;

```



```
string[] address = { "A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L", "M", "N", "O", "P", "Q", "R", "S", "T", "U", "V", "W", "X" };
```

```
for (int i = 0; i < address.Length; i++)
{
```

```
    worksheet[address[i] + "2:" + address[i] + "5"].Merge();
    switch (address[i])
    {
        case "A": worksheet["A2"].Value = "№"; break;
        case "B": worksheet["B2"].Value = "Дисциплина"; break;
        case "C": worksheet["C2"].Value = "Индекс группы"; break;
        case "D": worksheet["D2"].Value = "Количество групп"; break;
        case "E": worksheet["E2"].Value = "Количество студентов"; break;
        case "F": worksheet["F2"].Value = "Семестр"; break;
        case "G": worksheet["G2"].Value = "Форма обучения"; break;
        case "H": worksheet["H2"].Value = "Лекции"; break;
        case "I": worksheet["I2"].Value = "Практические занятия"; break;
        case "J": worksheet["J2"].Value = "Лабораторные"; break;
        case "K": worksheet["K2"].Value = "Консультации"; break;
        case "L": worksheet["L2"].Value = "Экзамены"; break;
        case "M": worksheet["M2"].Value = "Зачеты"; break;
        case "N": worksheet["N2"].Value = "Производственная практика"; break;
        case "O": worksheet["O2"].Value = "Курсовые"; break;
        case "P": worksheet["P2"].Value = "Дипломные"; break;
        case "Q": worksheet["Q2"].Value = "Рецензирование"; break;
        case "R": worksheet["R2"].Value = "Аспирантура"; break;
        case "S": worksheet["S2"].Value = "ГAK"; break;
        case "T": worksheet["T2"].Value = "КСРС"; break;
        case "U": worksheet["U2"].Value = "РГР"; break;
        case "V": worksheet["V2"].Value = "Заведование кафедры"; break;
        case "W": worksheet["W2"].Value = "Прочее"; break;
        case "X": worksheet["X2"].Value = "Всего"; break;
```

Ә қосымшасының жалғасы

```
}
```

```
ExcelRange quartersTitle = worksheet[address[i] + "2"].MergeArea;
quartersTitle.Style.Alignment.Orientation = 90;
quartersTitle.Style.Alignment.WrapText = true;
quartersTitle.Style.Alignment.HorizontalAlignment =
ExcelCellHorizontalAlignmentType.Center;
quartersTitle.Style.Alignment.VerticalAlignment =
ExcelCellVerticalAlignmentType.Center;
quartersTitle.Style.Font.Size = 9;
quartersTitle.Style.Font.Bold = true;
quartersTitle.ColumnWidthInChars = 4;
quartersTitle.RowHeightInPoints = 23;
```

```

        quartersTitle.Style.Borders[ExcelCellBorderIndex.Bottom].LineStyle =
ExcelCellLineStyle.Thin;
        quartersTitle.Style.Borders[ExcelCellBorderIndex.Top].LineStyle =
ExcelCellLineStyle.Thin;
        quartersTitle.Style.Borders[ExcelCellBorderIndex.Left].LineStyle =
ExcelCellLineStyle.Thin;
        quartersTitle.Style.Borders[ExcelCellBorderIndex.Right].LineStyle =
ExcelCellLineStyle.Thin;

        sun = 6;
        for (int ii = 0; ii <= GridView1.Rows.Count - 1; ii++)
        {

                worksheet[adress[i]+sun].Style.Borders[ExcelCellBorderIndex.Bottom].LineStyle =
ExcelCellLineStyle.Thin;
                worksheet[adress[i] + sun].Style.Borders[ExcelCellBorderIndex.Top].LineStyle =
ExcelCellLineStyle.Thin;
                worksheet[adress[i] + sun].Style.Borders[ExcelCellBorderIndex.Left].LineStyle =
ExcelCellLineStyle.Thin;
                worksheet[adress[i] + sun].Style.Borders[ExcelCellBorderIndex.Right].LineStyle =
ExcelCellLineStyle.Thin;
                sun++;
        }
    }

    worksheet["A2"].ColumnWidthInChars = 2;
    worksheet["B2"].ColumnWidthInChars = 13;
    worksheet["C2"].ColumnWidthInChars = 10;
    int sum = 6;
    for (int i = 0; i <= GridView1.Rows.Count - 1; i++)
    {
        /*Мәндерді тазалау*/
        for (int p = 6; p <= 23; p++)
            Ә қосымшасының жалғасы

        {
            if (p == 22) { continue; }
            if (GridView1.Rows[i].Cells[p].Text == "&nbsp;")
                GridView1.Rows[i].Cells[p].Text = "";
        }
        /****** Excel-дегі мәндерді алу *****/
        worksheet["A" + sum].Value = i + 1;

        int o = 0;
        for (int t = 0; t < adress.Length; t++)
        {
            if (adress[t] == "A") continue;
            if (o == 22) o++;

            worksheet[adress[t] + sum].Value = GridView1.Rows[i].Cells[o].Text;

```

```

        o++;
    }

    sum++;
}

/*****1-2
шығару*****/
for (int p = 1; p <= 17; p++)
{
    if (GridView2.Rows[0].Cells[p].Text == "&nbsp;")
        GridView2.Rows[0].Cells[p].Text = "";
    if (GridView2.Rows[1].Cells[p].Text == "&nbsp;")
        GridView2.Rows[1].Cells[p].Text = "";
    if (GridView2.Rows[2].Cells[p].Text == "&nbsp;")
        GridView2.Rows[2].Cells[p].Text = "";
}

int sum1 = sum+1;
int cel = 1;
for (int t = 7; t < adress.Length; t++)
{
    worksheet[adress[t] + sum1].Value = GridView2.Rows[0].Cells[cel].Text;
    cel++;
}

int sum2 = sum1 + 1;
int cel2 = 1;
for (int t = 7; t < adress.Length; t++)
{
    worksheet[adress[t] + sum2].Value = GridView2.Rows[1].Cells[cel2].Text;
    cel2++;
}

```

Ә қосымшасының жалғасы

```

int sum3 = sum2 + 1;
int cel3 = 1;
for (int t = 7; t < adress.Length; t++)
{
    worksheet[adress[t] + sum3].Value = GridView2.Rows[2].Cells[cel3].Text;
    cel3++;
}

int y = 1;

for (int r = sum + 1; r <= sum + 3; r++)
{
    for (int t = 7; t < adress.Length; t++)
    {

```

```

        worksheet[adress[t]
r].Style.Borders[ExcelCellBorderIndex.Bottom].LineStyle = ExcelCellLineStyle.Thin;
        worksheet[adress[t] + r].Style.Borders[ExcelCellBorderIndex.Top].LineStyle
= ExcelCellLineStyle.Thin;
        worksheet[adress[t] + r].Style.Borders[ExcelCellBorderIndex.Left].LineStyle
= ExcelCellLineStyle.Thin;
        worksheet[adress[t]
r].Style.Borders[ExcelCellBorderIndex.Right].LineStyle = ExcelCellLineStyle.Thin;
        //worksheet[adress[t] + r].Value = GridView2.Rows[k].Cells[p].Text;
    }
    worksheet["B" + r + ":G" + r].Merge();
    ExcelRange titleRange = worksheet["B" + r].MergeArea;
    switch (y)
    {
        case 1: worksheet["B" + r].Text = "Всего за 1-й семестр"; break;
        case 2: worksheet["B" + r].Text = "Всего за 2-й семестр"; break;
        case 3: worksheet["B" + r].Text = "Всего за год"; break;
    }
    y++;
    worksheet["B" + r + ":G"
r].Style.Borders[ExcelCellBorderIndex.Bottom].LineStyle = ExcelCellLineStyle.Thin;
    worksheet["B" + r + ":G"
r].Style.Borders[ExcelCellBorderIndex.Top].LineStyle = ExcelCellLineStyle.Thin;
    worksheet["B" + r + ":G"
r].Style.Borders[ExcelCellBorderIndex.Left].LineStyle = ExcelCellLineStyle.Thin;
    worksheet["B" + r + ":G"
r].Style.Borders[ExcelCellBorderIndex.Right].LineStyle = ExcelCellLineStyle.Thin;
    worksheet["B" + r + ":G" + r].Style.Font.Bold = true;
    }

#endregion

```

Ә қосымшасының жалғасы

```

System.Web.HttpResponse httpResponse = System.Web.HttpContext.Current.Response;
httpResponse.Clear();
httpResponse.ContentType = "Application/vnd.openxmlformats-
officedocument.spreadsheetml.sheet";
httpResponse.AddHeader("Content-Disposition", String.Format("attachment;
filename={0}", "Дисциплины.xlsx"));
workbook.Save(httpResponse.OutputStream);
workbook.Close();
httpResponse.End();
}

public DataTable O_clear()
{
    string sql = "exec procedure_Dissiplini";

```

```

        SqlDataAdapter da = new SqlDataAdapter(sql, conn);
        DataTable dt = new DataTable();
        da.Fill(dt);
        return dt;
    }

    public DataTable Fetch()
    {
        string sql = "SELECT * ,ISNULL(Лекции,0)+ ISNULL([Практика],0)
+ISNULL([Лабораторные],0)+
ISNULL([Консультации],0)+ISNULL([Экзамены],0)+ISNULL([Зачеты],0)+ISNULL([Производствен
ная_практика],0)+ISNULL([Курсовая_работа],0)+ISNULL([Дипломная],0)+ISNULL([Рецензирова
ние],0)+ISNULL([Аспирантура],0)+ISNULL([ГАК],0)+ISNULL([КСРС],0)+ISNULL([ПТР],0)+ISNULL(
[Заведование_кафедрой],0)+ISNULL([Другое],0) as [Всего],ID From Dissiplini where db_id=" +
Session["db_id"].ToString();
        SqlDataAdapter da = new SqlDataAdapter(sql, conn);
        DataTable dt = new DataTable();
        da.Fill(dt);
        return dt;
    }

    public void Delete(int id)
    {
        string sql = "Delete Dissiplini Where id=" + id;
        SqlConnection con = new SqlConnection(conn);
        con.Open();
        SqlCommand cmd = new SqlCommand(sql, con);
        cmd.ExecuteNonQuery();
        con.Close();
        con.Dispose();
    }

```

Ә қосымшасының жалғасы

```

    public DataTable select_Dissiplini()
    {
        string sql = "Select Distinct [Dissiplina] From [Spravochniki_Dissiplini] where db_id=" +
Session["db_id"].ToString();
        SqlDataAdapter da = new SqlDataAdapter(sql, conn);
        DataTable dt = new DataTable();
        da.Fill(dt);
        return dt;
    }

    public DataTable select_Gruppa()
    {
        string sql = "Select Distinct Gruppo From [Spravochniki_Gruppi] where db_id=" +
Session["db_id"].ToString();
        SqlDataAdapter da = new SqlDataAdapter(sql, conn);

```

```

        DataTable dt = new DataTable();
        da.Fill(dt);
        return dt;
    }
    public void Insert(string Дисциплина, string Группа, string Кол_во, string Контингент, string
Семестр, string Форма_обучения, string Лекции, string Практика, string Лабораторные, string
Консультации, string Экзамены, string Зачеты, string Производственная_практика, string
Курсовая_работа, string Дипломная, string Рецензирование, string Аспирантура, string ГАК, string
КСРС, string РГР, string Заведование_кафедрой, string Другое, string db_id)
    {
        string sql = "Insert Into Dissiplini Values (" + Дисциплина + ", " + Группа + ", " +
Кол_во + ", " + Контингент + ", " + Семестр + ", " + Форма_обучения + ", " + Лекции + ", " +
Практика + ", " + Лабораторные + ", " + Консультации + ", " + Экзамены + ", " + Зачеты + ", " +
Производственная_практика + ", " + Курсовая_работа + ", " + Дипломная + ", " + Рецензирование
+ ", " + Аспирантура + ", " + ГАК + ", " + КСРС + ", " + РГР + ", " + Заведование_кафедрой + ",
" + Другое + ", " + db_id + ")";
        SqlConnection con = new SqlConnection(conn);
        con.Open();
        SqlCommand cmd = new SqlCommand(sql, con);
        cmd.ExecuteNonQuery();
        con.Close();
        con.Dispose();
    }
}
}
}

```

index.aspx

using System;

Ә қосымшасының жалғасы

```

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

namespace stys2
{
    public partial class index : System.Web.UI.Page
    {
        string conn = ConfigurationManager.ConnectionStrings["connectionDB"].ToString();

        protected void Page_Load(object sender, EventArgs e)

```

```
{  
}
```

```
protected void Button1_Click(object sender, EventArgs e)
```

```
{
```

```
    string p_id = null;  
    string p_Login = null;  
    string p_Password = null;
```

```
    if (DrRol.SelectedValue == "Пользователь")
```

```
    {
```

```
        SqlConnection con = new SqlConnection(conn);  
        con.Open();
```

```
        SqlCommand cmd = new SqlCommand("select * from Polzovatel where login='" +  
Login.Text + "' and password='" + Password.Text + "'", con);
```

```
        SqlDataReader dr = cmd.ExecuteReader();  
        while (dr.Read())
```

```
        {
```

```
            p_id = dr["id"].ToString();  
            p_Login = dr["login"].ToString();  
            p_Password = dr["password"].ToString();
```

```
        }
```

```
        con.Close();
```

```
        if (p_Login != null && p_Password != null && p_id != null)
```

```
        {
```

```
            Session["Login"] = p_Login;  
            Session["id"] = p_id;  
            Response.Redirect("~/PolzovotelNameDB.aspx");
```

```
        }
```

```
    else
```

Ә қосымшасының жалғасы

```
{
```

```
    string message = "Кіруге рұқсат жоқ . Отказано в доступе !";  
    System.Text.StringBuilder sb = new System.Text.StringBuilder();
```

```
    sb.Append("<script type = 'text/javascript'>");
```

```
    sb.Append("window.onload=function(){");
```

```
    sb.Append("alert('");
```

```
    sb.Append(message);
```

```
    sb.Append("');");
```

```
    sb.Append("< /script>");
```

```
    ClientScript.RegisterClientScriptBlock(this.GetType(), "alert", sb.ToString());
```

```
    }
```

```
    }
```

```
    if (DrRol.SelectedValue == "Администратор")
```

```
    {
```

```
        if (Login.Text == "a" || Password.Text == "a")
```

```
        {
```

```
            Session["admin_Login"] = Login.Text;
```

```

        Response.Redirect("~/Admin.aspx");
    }
}
}
}
}

```

Neraspredelennaia_Nagruzka.aspx

```

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

```

```

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

```

```

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

```

```

using System.IO;
using System.Data.Linq;
using NPOI.HSSF.UserModel;
using NPOI.SS.UserModel;

```

Ә қосымшасының жалғасы

```

using NPOI.SS.Util;
using NPOI.HSSF.Util;
using NPOI.POIFS.FileSystem;
using NPOI.HPSF;

```

```

using System.Drawing;
using Winnovative.ExcelLib;

```

```

namespace stys2
{
    public partial class Neraspredelennaia_Nagruzka : System.Web.UI.Page
    {
        string conn = ConfigurationManager.ConnectionStrings["connectionDB"].ToString();
    }
}

```



```

protected void Page_Load(object sender, EventArgs e)
{
    if (Session["db_name"] == null && Session["db_id"] == null &&
Session["db_Plzovotel_Name"] == null)
    {
        Response.Redirect("~/index.aspx");
    }
    else
    {
        if (!Page.IsPostBack)
        {
            //Сортировка дисциплины
            SqlConnection con_sorting = new SqlConnection(conn);
            string com_sorting = "select * from Spravochniki_Dissiplini where db_id = " +
Session["db_id"].ToString();
            SqlDataAdapter adpt_sorting = new SqlDataAdapter(com_sorting, con_sorting);
            DataTable dt_sorting = new DataTable();
            adpt_sorting.Fill(dt_sorting);
            Sortyrovka_Deissiplini.DataSource = dt_sorting;
            Sortyrovka_Deissiplini.DataBind();
            Sortyrovka_Deissiplini.DataTextField = "Dissiplina";
            Sortyrovka_Deissiplini.DataValueField = "Dissiplina";
            Sortyrovka_Deissiplini.DataBind();
            Sortyrovka_Deissiplini.Items.Insert(0, new ListItem("Все дисциплины", "Все
дисциплины"));

            //Сортировка семестры ол html кода

                Ә қосымшасының жалғасы

            //Сортировка Форм обучения
            SqlConnection con_form = new SqlConnection(conn);
            string com_form = "select * from Spravochniki_Forma_ovuchenia where db_id = " +
Session["db_id"].ToString(); ;
            SqlDataAdapter adpt_form = new SqlDataAdapter(com_form, con_form);
            DataTable dt_form = new DataTable();
            adpt_form.Fill(dt_form);
            Sortyrovka_Forma_ovuchenie.DataSource = dt_form;
            Sortyrovka_Forma_ovuchenie.DataBind();
            Sortyrovka_Forma_ovuchenie.DataTextField = "Forma_Obuchenia";
            Sortyrovka_Forma_ovuchenie.DataValueField = "Forma_Obuchenia";
            Sortyrovka_Forma_ovuchenie.DataBind();
            Sortyrovka_Forma_ovuchenie.Items.Insert(0, new ListItem("Все формы", "Все
формы"));

            //Сортировка группы
            SqlConnection con_grup = new SqlConnection(conn);

```

```

        string com_grup = "select * from Spravochniki_Gruppi where db_id = " +
Session["db_id"].ToString(); ;
        SqlDataAdapter adpt_grup = new SqlDataAdapter(com_grup, con_grup);
        DataTable dt_grup = new DataTable();
        adpt_grup.Fill(dt_grup);
        Sortyrovka_Gruppi.DataSource = dt_grup;
        Sortyrovka_Gruppi.DataBind();
        Sortyrovka_Gruppi.DataTextField = "Gruppa";
        Sortyrovka_Gruppi.DataValueField = "Gruppa";
        Sortyrovka_Gruppi.DataBind();
        Sortyrovka_Gruppi.Items.Insert(0, new ListItem("Все группы", "Все группы"));

        sortirovka();
    }
}
}

```

```

protected void Sortyrovka_Deissiplini_SelectedIndexChanged(object sender, EventArgs e)
{
    sortirovka();
}

```

Ә қосымшасының жалғасы

```

protected void Sortyrovka_Semestr_SelectedIndexChanged(object sender, EventArgs e)
{
    sortirovka();
}

protected void Sortyrovka_Forma_ovuchenie_SelectedIndexChanged(object sender,
EventArgs e)
{
    sortirovka();
}

protected void Sortyrovka_Gruppi_SelectedIndexChanged(object sender, EventArgs e)
{
    sortirovka();
}

public void sortirovka()
{

```

```
string disciplina = null, semestr = null, forma_obuchanie = null, grupa = null, db_id = Session["db_id"].ToString();
```

```
if (Sortyrovka_Deissiplini.SelectedValue.ToString() != "Все дисциплины")  
    disciplina = " " and D.[Дисциплина]=" " +  
Sortyrovka_Deissiplini.SelectedValue.ToString() + "";
```

```
if (Sortyrovka_Deissiplini.SelectedValue.ToString() == "Все дисциплины")  
    disciplina = " and D.[Дисциплина]=D.[Дисциплина]";
```

```
if (Sortyrovka_Semestr.SelectedValue.ToString() != "Все семестры")  
    semestr = " and D.[Семестр]=" + Sortyrovka_Semestr.SelectedValue.ToString();
```

```
if (Sortyrovka_Semestr.SelectedValue.ToString() == "Все семестры")  
    semestr = " and D.[Семестр]=D.[Семестр]";
```

```
if (Sortyrovka_Forma_ovuchenie.SelectedValue.ToString() != "Все формы")  
    forma_obuchanie = " " and D.[Форма_обучения]=" " +  
Sortyrovka_Forma_ovuchenie.SelectedValue.ToString() + "";
```

```
if (Sortyrovka_Forma_ovuchenie.SelectedValue.ToString() == "Все формы")  
    forma_obuchanie = " and D.[Форма_обучения]=D.[Форма_обучения]";
```

```
if (Sortyrovka_Gruppi.SelectedValue.ToString() != "Все группы")  
    grupa = " and D.[Группа]=" + Sortyrovka_Gruppi.SelectedValue.ToString() + "";
```

```
if (Sortyrovka_Gruppi.SelectedValue.ToString() == "Все группы")  
    grupa = " and D.[Группа]=D.[Группа]";
```

```
string sql = @"SELECT D.Дисциплина,d.[Группа],D.Форма_обучения as 'Форма  
обучения',D.Семестр,
```

Ә қосымшасының жалғасы

```
CASE WHEN (ISNULL(D.[Лекции],0) - ISNULL(sum(R.[Лекции],0))=0  
THEN NULL ELSE (ISNULL(D.[Лекции],0) - ISNULL(sum(R.[Лекции],0)) END AS 'Лекции',
```

```
CASE WHEN (ISNULL(D.[Практика],0) - ISNULL(sum(R.[Практика],0))=0  
THEN NULL ELSE (ISNULL(D.[Практика],0) - ISNULL(sum(R.[Практика],0)) END as  
'Прак',
```

```
CASE WHEN (ISNULL(D.[Лабораторные],0) - ISNULL(sum(R.[Лабораторные],0))=0  
then null else (ISNULL(D.[Лабораторные],0) - ISNULL(sum(R.[Лабораторные],0)) end as  
'Лаб',
```

```
CASE WHEN (ISNULL(D.[Консультации],0) - ISNULL(sum(R.[Консультации],0))=0  
THEN NULL ELSE (ISNULL(D.[Консультации],0) - ISNULL(sum(R.[Консультации],0)) END  
as 'Конс',
```

```
CASE WHEN (ISNULL(D.[Экзамены],0) - ISNULL(sum(R.[Экзамены],0))=0 THEN NULL  
ELSE  
(ISNULL(D.[Экзамены],0) - ISNULL(sum(R.[Экзамены],0)) END as 'Экзамены',
```

```
CASE WHEN (ISNULL(D.[Зачеты],0) - ISNULL(sum(R.[Зачеты],0))=0
```

```

THEN NULL ELSE (ISNULL(D.[Зачеты],0) - ISNULL(sum(R.[Зачеты]),0)) END as 'Зачеты',

CASE WHEN (ISNULL(D.[Производственная_практика],0) - ISNULL(sum(R.[Производственная_практика]),0))=0 THEN NULL ELSE ISNULL(D.[Производственная_практика],0) - ISNULL(sum(R.[Производственная_практика]),0) END as 'Пр.прак',

CASE WHEN (ISNULL(D.[Курсовая_работа],0) - ISNULL(sum(R.[Курсовая_работа]),0))=0 THEN NULL ELSE ISNULL(D.[Курсовая_работа],0) - ISNULL(sum(R.[Курсовая_работа]),0) END as 'Курс.раб',

CASE WHEN (ISNULL(D.[Дипломная],0) - ISNULL(sum(R.[Дипломная]),0))=0 THEN NULL ELSE ISNULL(D.[Дипломная],0) - ISNULL(sum(R.[Дипломная]),0) END as 'Дипл',

CASE WHEN (ISNULL(D.[Рецензирование],0) - ISNULL(sum(R.[Рецензирование]),0))=0 THEN NULL ELSE ISNULL(D.[Рецензирование],0) - ISNULL(sum(R.[Рецензирование]),0) END as 'Рец',

CASE WHEN (ISNULL(D.[Аспирантура],0) - ISNULL(sum(R.[Аспирантура]),0))=0 THEN NULL ELSE ISNULL(D.[Аспирантура],0) - ISNULL(sum(R.[Аспирантура]),0) END as 'Асп',

```

Ә қосымшасының жалғасы

```

CASE WHEN (ISNULL(D.[ГАК],0) - ISNULL(sum(R.[ГАК]),0))=0 THEN NULL ELSE ISNULL(D.[ГАК],0) - ISNULL(sum(R.[ГАК]),0) END as 'ГАК',

CASE WHEN (ISNULL(D.[КСПС],0) - ISNULL(sum(R.[КСПС]),0))=0 THEN NULL ELSE ISNULL(D.[КСПС],0) - ISNULL(sum(R.[КСПС]),0) END as 'КСПС',

CASE WHEN (ISNULL(D.[ПГР],0) - ISNULL(sum(R.[ПГР]),0))=0 THEN NULL ELSE ISNULL(D.[ПГР],0) - ISNULL(sum(R.[ПГР]),0) END as 'ПГР',

CASE WHEN (ISNULL(D.[Заведование_кафедрой],0) - ISNULL(sum(R.[Заведование_кафедрой]),0))=0 THEN NULL ELSE ISNULL(D.[Заведование_кафедрой],0) - ISNULL(sum(R.[Заведование_кафедрой]),0) END as 'Зав.каф',

CASE WHEN (ISNULL(D.[Другое],0) - ISNULL(sum(R.[Другое]),0))=0 THEN NULL ELSE ISNULL(D.[Другое],0) - ISNULL(sum(R.[Другое]),0) END as 'Другое',

ISNULL(CASE WHEN (ISNULL(D.[Лекции],0) - ISNULL(sum(R.[Лекции]),0))=0 THEN NULL ELSE (ISNULL(D.[Лекции],0) - ISNULL(sum(R.[Лекции]),0)) END,0)+

```

```

ISNULL(CASE WHEN (ISNULL(D.[Практика],0) - ISNULL(sum(R.[Практика],0))=0
THEN NULL ELSE (ISNULL(D.[Практика],0) - ISNULL(sum(R.[Практика],0)) END,0) +

ISNULL(CASE WHEN (ISNULL(D.[Лабораторные],0) - ISNULL(sum(R.[Лабораторные],0))=0
then null else (ISNULL(D.[Лабораторные],0) - ISNULL(sum(R.[Лабораторные],0)) end,0)

+ ISNULL(CASE WHEN (ISNULL(D.[Консультации],0) - ISNULL(sum(R.[Консультации],0))=0
THEN NULL ELSE (ISNULL(D.[Консультации],0) - ISNULL(sum(R.[Консультации],0))
END,0)+

ISNULL(CASE WHEN (ISNULL(D.[Экзамены],0) - ISNULL(sum(R.[Экзамены],0))=0 THEN
NULL ELSE
(ISNULL(D.[Экзамены],0) - ISNULL(sum(R.[Экзамены],0)) END,0)

+ISNULL(CASE WHEN (ISNULL(D.[Зачеты],0) - ISNULL(sum(R.[Зачеты],0))=0
THEN NULL ELSE (ISNULL(D.[Зачеты],0) - ISNULL(sum(R.[Зачеты],0)) END,0)

+ISNULL(CASE WHEN (ISNULL(D.[Производственная_практика],0) - ISNULL(sum(R.[Производственная_практика],0))=0
THEN NULL ELSE (ISNULL(D.[Производственная_практика],0) - ISNULL(sum(R.[Производственная_практика],0))
END,0)

        Ә қосымшасының жалғасы

THEN NULL ELSE (ISNULL(D.[Производственная_практика],0) - ISNULL(sum(R.[Производственная_практика],0))
ISNULL(sum(R.[Производственная_практика],0)) END,0)

+ISNULL(CASE WHEN (ISNULL(D.[Курсовая_работа],0) - ISNULL(sum(R.[Курсовая_работа],0))=0
THEN NULL ELSE (ISNULL(D.[Курсовая_работа],0) - ISNULL(sum(R.[Курсовая_работа],0))
END,0)

+ISNULL(CASE WHEN (ISNULL(D.[Дипломная],0) - ISNULL(sum(R.[Дипломная],0))=0
THEN NULL ELSE (ISNULL(D.[Дипломная],0) - ISNULL(sum(R.[Дипломная],0))
END,0)

+ISNULL(CASE WHEN (ISNULL(D.[Рецензирование],0) - ISNULL(sum(R.[Рецензирование],0))=0
THEN NULL ELSE (ISNULL(D.[Рецензирование],0) - ISNULL(sum(R.[Рецензирование],0))
END,0)

+ISNULL(CASE WHEN (ISNULL(D.[Аспирантура],0) - ISNULL(sum(R.[Аспирантура],0))=0
THEN NULL ELSE (ISNULL(D.[Аспирантура],0) - ISNULL(sum(R.[Аспирантура],0))
END,0)

+ISNULL(CASE WHEN (ISNULL(D.[ГАК],0) - ISNULL(sum(R.[ГАК],0))=0 THEN NULL
ELSE (ISNULL(D.[ГАК],0) - ISNULL(sum(R.[ГАК],0))
END,0)

+ISNULL(CASE WHEN (ISNULL(D.[КСПС],0) - ISNULL(sum(R.[КСПС],0))=0 THEN NULL
ELSE (ISNULL(D.[КСПС],0) - ISNULL(sum(R.[КСПС],0))
END,0)

```

```

ISNULL(sum(R.[KCPC]),0) END,0)

+ISNULL(CASE WHEN (ISNULL(D.[ПГП],0) - ISNULL(sum(R.[ПГП]),0))=0 THEN NULL
ELSE ISNULL(D.[ПГП],0) -
ISNULL(sum(R.[ПГП]),0) END,0)

+ISNULL(CASE WHEN (ISNULL(D.[Заведование_кафедрой],0)
ISNULL(sum(R.[Заведование_кафедрой]),0))=0 THEN
NULL ELSE ISNULL(D.[Заведование_кафедрой],0)
ISNULL(sum(R.[Заведование_кафедрой]),0) END,0)

+ISNULL(CASE WHEN (ISNULL(D.[Другое],0) -
ISNULL(sum(R.[Другое]),0))=0 THEN NULL ELSE ISNULL(D.[Другое],0)
ISNULL(sum(R.[Другое]),0) END,0)
as 'Всего'

FROM Dissiplini D JOIN [Распределение_Nagruzki] R ON (D.ID = R.id_dis and D.db_id=" +
Session["db_id"].ToString() +

```

Ә қосымшасының жалғасы

```

@" and R.db_id=" + Session["db_id"].ToString() + disciplina + группа + semestr +
forma_obuchanie + @" )

```

```

GROUP BY D.Дисциплина, D.Форма_обучения, D.Семестр, d.[Группа]
,D.[Лекции] ,D.[Практика] ,D.[Лабораторные] ,D.[Консультации] ,D.[Экзамены]
,D.[Зачеты] ,
D.[Производственная_практика] ,D.[Курсовая_работа] ,D.[Дипломная]
,D.[Рецензирование] ,D.[Аспирантура] ,D.[ГАК]
,D.[KCPC] ,D.[ПГП] ,D.[Заведование_кафедрой] ,D.[Другое]
Having (ISNULL(D.[Лекции],0) - ISNULL(sum(R.[Лекции]),0))>0 or
(ISNULL(D.[Практика],0) - ISNULL(sum(R.[Практика]),0)
)>0 or (ISNULL(D.[Лабораторные],0) - ISNULL(sum(R.[Лабораторные]),0) )>0 or
(ISNULL(D.[Консультации],0) - ISNULL(sum(R.[Консультации],
0) )>0 or (ISNULL(D.[Экзамены],0) - ISNULL(sum(R.[Экзамены],0))>0 or
(ISNULL(D.[Зачеты],0) - ISNULL(sum(R.[Зачеты]),0) )>0
or (ISNULL(D.[Производственная_практика],0)
ISNULL(sum(R.[Производственная_практика]),0) )>0 or (ISNULL(D.[Курсовая_работа],0) -
ISNULL(sum(R.[Курсовая_работа]),0) )>0 or (ISNULL(D.[Дипломная],0) -
ISNULL(sum(R.[Дипломная]),0) )>0 or (ISNULL(D.[Рецензирование],0)
- ISNULL(sum(R.[Рецензирование]),0) )>0 or (ISNULL(D.[Аспирантура],0) -
ISNULL(sum(R.[Аспирантура]),0) )>0 or (ISNULL(D.[ГАК],0) -
ISNULL(sum(R.[ГАК]),0) )>0 or (ISNULL(D.[KCPC],0) - ISNULL(sum(R.[KCPC]),0) )>0
or (ISNULL(D.[ПГП],0) - ISNULL(sum(R.[ПГП]),0) )>0
or (ISNULL(D.[Заведование_кафедрой],0) - ISNULL(sum(R.[Заведование_кафедрой]),0))>0
or (ISNULL(D.[Другое],0) - ISNULL(sum(R.[Другое]),0) )>0";

```

```

string sql1 = @"union all select
D.Дисциплина,d.[Группа],D.Форма_обучения,D.Семестр,D.[Лекции],D.[Практика],D.[Лабораторны

```

```

e] ,D.[Консультации],D.[Экзамены],D.[Зачеты],D.[Производственная_практика]
,D.[Курсовая_работа],D.[Дипломная] ,D.[Рецензирование] ,D.[Аспирантура] ,D.[ГАК]
,D.[КСПС] ,D.[ПГР],D.[Заведование_кафедрой] ,D.[Другое],ISNULL(D.Лекции,0)+
ISNULL(D.[Практика],0) +ISNULL(D.[Лабораторные],0)+
ISNULL(D.[Консультации],0)+ISNULL(D.[Экзамены],0)+ISNULL(D.[Зачеты],0)+ISNULL(D.[Произ
водственная_практика],0)+ISNULL(D.[Курсовая_работа],0)+ISNULL(D.[Дипломная],0)+ISNULL(D.
[Рецензирование],0)+ISNULL(D.[Аспирантура],0)+ISNULL(D.[ГАК],0)+ISNULL(D.[КСПС],0)+ISN
ULL(D.[ПГР],0)+ISNULL(D.[Заведование_кафедрой],0)+ISNULL(D.[Другое],0) as 'Всего'
FROM Dissiplini D where d.db_id = " + Session["db_id"].ToString() +
@" and D.id not in (select R.[id_dis] from [Распределение_Nagruzki] R where R.db_id = "
+ Session["db_id"].ToString() + ")" + disciplina + группа + semestr + forma_obuchanie;

```

```

SqlConnection con = new SqlConnection(conn);
con.Open();
SqlDataAdapter da = new SqlDataAdapter(sql+sql1, con);
DataSet ds = new DataSet();

```

Ә қосымшасының жалғасы

```

da.Fill(ds);
GridView1.DataSource = ds;    }    }}

```

HTML- коды

```

<%@ Page Title="Добро пожаловать !" Language="C#" AutoEventWireup="true"
CodeBehind="index.aspx.cs" Inherits="stys2.index" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

```

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head id="Head1" runat="server">
<title></title>
<style type="text/css">
.style1
{
width:79%;
}
.style2
{
width: 79%;
height: 21px;
}
.style3
{
height: 40px;
width: 47px;
}
.style4
{
width: 47px;
}
.style5

```

```

    {
      height: 178px;
      width: 50px;
    }

    .style6
    {
      height: 178px;
      width: 382px;
    }
  </style>
</head>
<body style="font-size: .80em;
font-family: Helvetica Neue, Lucida Grande, Segoe UI, Arial, Helvetica, Verdana, sans-serif;
margin: 0px;
padding: 0px;

```

Ә қосымшасының жалғасы

```

background-image: url('Picture/bg.jpg');
"
>
<center>
  <form id="form1" runat="server">
    <div>
      <br />
      <table border="0" cellpadding="0" cellspacing="0" style="border:1px solid #000000">
        <tr>
          <td colspan="3">
            
          </td>
        </tr>
        <tr>
          <td colspan="3" style="background-image: url(Picture/MySamples_2x1.jpg); height:
197px">
            <table border="0" cellpadding="0" cellspacing="0">
              <tr>
                <td style="width: 100px; height: 178px">
                </td>
                <td class="style5">
                  <table border="0" cellpadding="0" cellspacing="1" style="width: 234px">
                    <tr>
                      <td align="left" style="font-size: 9pt; width: 106px; font-family:
'Microsoft Sans Serif';
height: 13px">
                    <fieldset style="width: 250px; border:1px solid #0E8A06; color:#0E8A06">
                    <legend id="legend">Войти:</legend>
                    <table border="0" cellpadding="0" cellspacing="0">
                    <TR>
                    <TD class="style3">Пол:</TD>
                    <TD class="style2"><asp:DropDownList ID="DrRol" runat="server" Width="185px">

```



```

        <asp:ListItem>Пользователь</asp:ListItem>
        <asp:ListItem>Администратор</asp:ListItem>
        </asp:DropDownList></TD>
</TR>

<TR>
    <TD class="style3">Логин:</TD>
    <TD
        class="style2"><asp:TextBox
            ID="Login"
            runat="server"
            Width="180px"></asp:TextBox></TD>
</TR>

<TR>
    <TD class="style4">Пароль:</TD>
    <TD
        class="style1"><asp:TextBox
            ID="Password"
            runat="server"
            Width="180px"
            TextMode="password"></asp:TextBox></TD>
</TR>

```

Ә қосымшасының жалғасы

```

        <td align="right"><asp:Button ID="Boiti" runat="server" Text="Войти..."
            ForeColor="#0E8A06" Font-Bold="true" Width="100px" onclick="Button1_Click"/></td>
</TR>
</table>
</fieldset>

        </td>
        </tr>

        </table>
        </td>
        <td class="style6">
        </td>
    </tr>
</table>
<br />
</td>
</tr>
<tr>
    <td colspan="3" style="background-image: url(Picture/MySamples_3x1.jpg); height:
63px" align="right">
        &nbsp;
    </td>
</tr>
</table>
</div>
</form>
</center>
</body>
</html>

```

```

Inherits="stys2.Dissiplina" %>

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

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

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

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

        <script language="Javascript" type="text/javascript">
            Ә қосымшасының жалғасы

            function EnsureNumeric() {
                var key = window.event.keyCode;
                if (key < 48 || key > 57)
                    window.event.returnValue = false;
            }
        </script>

        <table>
        <tr>
        <td>

            <asp:DropDownList ID="DropDownList1" runat="server" Height="22px" Width="142px"
                AutoPostBack="true"
                onselectedindexchanged="DropDownList1_SelectedIndexChanged" ></asp:DropDownList> +

            <asp:DropDownList ID="DropDownList2" runat="server" Height="22px" Width="142px"
                AutoPostBack="true"
                onselectedindexchanged="DropDownList2_SelectedIndexChanged" >
                <asp:ListItem>Все семестры</asp:ListItem>
                <asp:ListItem>1</asp:ListItem>
                <asp:ListItem>2</asp:ListItem>
            </asp:DropDownList> +
            <asp:DropDownList ID="DropDownList3" runat="server" Height="22px" Width="142px"
                AutoPostBack="true"
                onselectedindexchanged="DropDownList3_SelectedIndexChanged"></asp:DropDownList> +
            <asp:DropDownList ID="DropDownList4" runat="server" Height="22px" Width="142px"
                AutoPostBack="true"
                onselectedindexchanged="DropDownList4_SelectedIndexChanged"></asp:DropDownList>

            <asp:Button ID="Zvros" runat="server" Text="Сброс" style="height: 25px"
                width="65px" onclick="Zvros_Click"/>

        </td>

```

```

</tr>
<tr><td>
<asp:Panel runat="server" ID="MainPanel" ScrollBars="Auto" Height="315" Width="105%">

    <asp:GridView ID="GridView_1" runat="server" AutoGenerateColumns="False"

ShowFooter="True" DataKeyNames="id,db_id"

OnRowEditing="GridView_1_RowEditing"
OnRowCancelingEdit="GridView_1_RowCancelingEdit"
OnRowUpdating="GridView_1_RowUpdating"
OnRowDataBound="GridView_1_RowDataBound"
                Ә қосымшасының жалғасы

OnRowCommand="GridView_1_RowCommand"
OnRowDeleting="GridView_1_RowDeleting" style="margin-bottom: 104px"
>

<Columns>
<asp:TemplateField HeaderText="Дисциплина">
<EditItemTemplate>
<asp:DropDownList ID="cmbДисциплина" Width="80px" runat="server"
DataTextField="Dissiplina" DataValueField="Dissiplina">
</asp:DropDownList>
</EditItemTemplate>
<ItemTemplate>
<asp:Label ID="Дисциплина" runat="server" Text='<%=# Eval("Дисциплина") %>'></asp:Label>
</ItemTemplate>
<FooterTemplate>
<asp:DropDownList ID="cmbNewДисциплина" Width="80px" runat="server"
DataTextField="Dissiplina" DataValueField="Dissiplina">
</asp:DropDownList>
</FooterTemplate>
</asp:TemplateField>

<asp:TemplateField HeaderText="Группа">
<EditItemTemplate>
<asp:DropDownList ID="cmbГруппа"
onselectedindexchanged="cmbГруппа_SelectedIndexChanged" Width="50px" AutoPostBack="true"
runat="server" DataTextField="Gruppa" DataValueField="Gruppa">
</asp:DropDownList>
</EditItemTemplate>
<ItemTemplate>
<asp:Label ID="Группа" runat="server" Text='<%=# Eval("Группа") %>'></asp:Label>
</ItemTemplate>
<FooterTemplate>
<asp:DropDownList ID="cmbNewГруппа"
onselectedindexchanged="cmbNewГруппа_SelectedIndexChanged" Width="50px" AutoPostBack="true"
runat="server" DataTextField="Gruppa" DataValueField="Gruppa">

```

```
</asp:DropDownList>
</FooterTemplate>
</asp:TemplateField>
```

```
<asp:TemplateField HeaderText="Кол.гр" SortExpression="Кол.групп">
  <EditItemTemplate> <asp:TextBox OnKeyPress="EnsureNumeric()" Enabled="false"
Width="30px" ID="txtКол_во" runat="server" Text='<%=# Eval("Кол_во") %>'> </asp:TextBox>
</EditItemTemplate>
  <FooterTemplate> <asp:TextBox OnKeyPress="EnsureNumeric()" Enabled="false"
Width="30px" ID="txtNewКол_во" runat="server"> </asp:TextBox>
</FooterTemplate>
```

Ә қосымшасының жалғасы

```
<ItemTemplate> <asp:Label ID="Label2" runat="server" Text='<%=# Bind("Кол_во") %>'>
</asp:Label> </ItemTemplate>
</asp:TemplateField>
```

```
<asp:TemplateField HeaderText="Конт">
  <EditItemTemplate>
  <asp:TextBox OnKeyPress="EnsureNumeric()" Enabled="false" Width="30px"
ID="txtКонтингент" runat="server" Text='<%=# Bind("Контингент") %>'></asp:TextBox>
  </EditItemTemplate>
  <FooterTemplate>
  <asp:TextBox OnKeyPress="EnsureNumeric()" Enabled="false" Width="30px"
ID="txtNewКонтингент" runat="server" ></asp:TextBox>
  </FooterTemplate>
  <ItemTemplate>
  <asp:Label ID="Контингент" runat="server" Text='<%=# Bind("Контингент") %>'></asp:Label>
  </ItemTemplate>
</asp:TemplateField>
```

```
<asp:TemplateField HeaderText="Сем">
  <EditItemTemplate>
  <asp:DropDownList Width="40px" ID="cmbСеместр" runat="server" SelectedValue='<%=#
Eval("Семестр") %>'>
  <asp:ListItem Value="1" Text="1"></asp:ListItem>
  <asp:ListItem Value="2" Text="2"></asp:ListItem>
  </asp:DropDownList>
  </EditItemTemplate>
  <ItemTemplate>
  <asp:Label ID="Семестр" runat="server" Text='<%=# Eval("Семестр") %>'></asp:Label>
  </ItemTemplate>
  <FooterTemplate>
  <asp:DropDownList Width="40px" ID="cmbNewСеместр" runat="server" >
  <asp:ListItem Selected="True" Text="1" Value="1"></asp:ListItem>
  <asp:ListItem Text="2" Value="2"></asp:ListItem> </asp:DropDownList>
  </FooterTemplate>
</asp:TemplateField>
```

```

<asp:TemplateField HeaderText="Форм.обуч.">
  <EditItemTemplate>
    <asp:TextBox      OnKeyPress="EnsureNumeric()"      Enabled="false"      Width="50px"
ID="txtФорма_обучения" runat="server" Text='<%# Bind("Форма_обучения") %>'></asp:TextBox>
  </EditItemTemplate>
  <FooterTemplate>
    <asp:TextBox      OnKeyPress="EnsureNumeric()"      Enabled="false"      Width="50px"
ID="txtNewФорма_обучения" runat="server" ></asp:TextBox>

```

Ә қосымшасының жалғасы

```

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

```

```

<asp:TemplateField HeaderText="Лекции">
  <EditItemTemplate>
    <asp:TextBox OnKeyPress="EnsureNumeric()" Width="35px" ID="txtЛекции" runat="server"
Text='<%# Bind("Лекции") %>'></asp:TextBox>
  </EditItemTemplate>
  <FooterTemplate>
    <asp:TextBox OnKeyPress="EnsureNumeric()" Width="35px" ID="txtNewЛекции"
runat="server" ></asp:TextBox>
  </FooterTemplate>
  <ItemTemplate>
    <asp:Label ID="Лекции" runat="server" Text='<%# Bind("Лекции") %>'></asp:Label>
  </ItemTemplate>
</asp:TemplateField>

```

```

<asp:TemplateField HeaderText="Прак">
  <EditItemTemplate>
    <asp:TextBox OnKeyPress="EnsureNumeric()" Width="30px" ID="txtПрактика" runat="server"
Text='<%# Bind("Практика") %>'></asp:TextBox>
  </EditItemTemplate>
  <FooterTemplate>
    <asp:TextBox OnKeyPress="EnsureNumeric()" Width="30px" ID="txtNewПрактика"
runat="server" ></asp:TextBox>
  </FooterTemplate>
  <ItemTemplate>
    <asp:Label ID="Практика" runat="server" Text='<%# Bind("Практика") %>'></asp:Label>
  </ItemTemplate>
</asp:TemplateField>

```

```

<asp:TemplateField HeaderText="Лаб">
  <EditItemTemplate>
    <asp:TextBox OnKeyPress="EnsureNumeric()" Width="20px" ID="txtЛабораторные"
runat="server" Text='<%# Bind("Лабораторные") %>'></asp:TextBox>

```

```

</EditItemTemplate>
<FooterTemplate>
<asp:TextBox OnKeyPress="EnsureNumeric()" Width="20px" ID="txtNewЛабораторные"
runat="server" ></asp:TextBox>
</FooterTemplate>
<ItemTemplate>
<asp:Label ID="Лабораторные" runat="server" Text='< %# Bind("Лабораторные")
%>'></asp:Label>

```

Ә қосымшасының жалғасы

```

</ItemTemplate>
</asp:TemplateField>

<asp:TemplateField HeaderText="Конс">
<EditItemTemplate>
<asp:TextBox OnKeyPress="EnsureNumeric()" Width="30px" ID="txtКонсультации"
runat="server" Text='< %# Bind("Консультации") %>'></asp:TextBox>
</EditItemTemplate>
<FooterTemplate>
<asp:TextBox OnKeyPress="EnsureNumeric()" Width="30px" ID="txtNewКонсультации"
runat="server" ></asp:TextBox>
</FooterTemplate>
<ItemTemplate>
<asp:Label ID="Консультации" runat="server" Text='< %# Bind("Консультации")
%>'></asp:Label>
</ItemTemplate>
</asp:TemplateField>

<asp:TemplateField HeaderText="Экзамены">
<EditItemTemplate>
<asp:TextBox OnKeyPress="EnsureNumeric()" Width="30px" ID="txtЭкзамены" runat="server"
Text='< %# Bind("Экзамены") %>'></asp:TextBox>
</EditItemTemplate>
<FooterTemplate>
<asp:TextBox OnKeyPress="EnsureNumeric()" Width="30px" ID="txtNewЭкзамены"
runat="server" ></asp:TextBox>
</FooterTemplate>
<ItemTemplate>
<asp:Label ID="Экзамены" runat="server" Text='< %# Bind("Экзамены") %>'></asp:Label>
</ItemTemplate>
</asp:TemplateField>

<asp:TemplateField HeaderText="Зачеты">
<EditItemTemplate>
<asp:TextBox OnKeyPress="EnsureNumeric()" Width="50px" ID="txtЗачеты" runat="server"
Text='< %# Bind("Зачеты") %>'></asp:TextBox>
</EditItemTemplate>
<FooterTemplate>
<asp:TextBox OnKeyPress="EnsureNumeric()" Width="50px" ID="txtNewЗачеты" runat="server"
></asp:TextBox>

```

```

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

```

```

<asp:TemplateField HeaderText="Пр.прак">

```

Ә қосымшасының жалғасы

```

<EditItemTemplate>
<asp:TextBox OnKeyPress="EnsureNumeric()" Width="50px"
ID="txtПроизводственная_практика" runat="server" Text='< %# Bind("Производственная_практика")
%>'></asp:TextBox>
</EditItemTemplate>
<FooterTemplate>

```

```

<asp:TextBox OnKeyPress="EnsureNumeric()" Width="50px"
ID="txtNewПроизводственная_практика" runat="server" ></asp:TextBox>
</FooterTemplate>
<ItemTemplate>

```

```

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

```

```

<asp:TemplateField HeaderText="Курс.раб">

```

```

<EditItemTemplate>
<asp:TextBox OnKeyPress="EnsureNumeric()" Width="50px" ID="txtКурсовая_работа"
runat="server" Text='< %# Bind("Курсовая_работа") %>'></asp:TextBox>
</EditItemTemplate>
<FooterTemplate>

```

```

<asp:TextBox OnKeyPress="EnsureNumeric()" Width="50px" ID="txtNewКурсовая_работа"
runat="server" ></asp:TextBox>
</FooterTemplate>
<ItemTemplate>

```

```

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

```

```

<asp:TemplateField HeaderText="Дипл">

```

```

<EditItemTemplate>
<asp:TextBox OnKeyPress="EnsureNumeric()" Width="20px" ID="txtДипломная" runat="server"
Text='< %# Bind("Дипломная") %>'></asp:TextBox>
</EditItemTemplate>
<FooterTemplate>

```

```

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

```

```

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

```

```

</ItemTemplate>
</asp:TemplateField>

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

```

Ә қосымшасының жалғасы

```

<asp:TextBox OnKeyPress="EnsureNumeric()" Width="20px" ID="txtРецензирование"
runat="server" Text='<%=# Bind("Рецензирование") %>'></asp:TextBox>
</EditItemTemplate>
<FooterTemplate>
<asp:TextBox OnKeyPress="EnsureNumeric()" Width="20px" ID="txtNewРецензирование"
runat="server" ></asp:TextBox>
</FooterTemplate>
<ItemTemplate>
<asp:Label ID="Рецензирование" runat="server" Text='<%=# Bind("Рецензирование")
%>'></asp:Label>
</ItemTemplate>
</asp:TemplateField>

<asp:TemplateField HeaderText="Асп">
<EditItemTemplate>
<asp:TextBox OnKeyPress="EnsureNumeric()" Width="20px" ID="txtАспирантура"
runat="server" Text='<%=# Bind("Аспирантура") %>'></asp:TextBox>
</EditItemTemplate>
<FooterTemplate>
<asp:TextBox OnKeyPress="EnsureNumeric()" Width="20px" ID="txtNewАспирантура"
runat="server" ></asp:TextBox>
</FooterTemplate>
<ItemTemplate>
<asp:Label ID="Аспирантура" runat="server" Text='<%=# Bind("Аспирантура")
%>'></asp:Label>
</ItemTemplate>
</asp:TemplateField>

<asp:TemplateField HeaderText="ГAK">
<EditItemTemplate>
<asp:TextBox OnKeyPress="EnsureNumeric()" Width="20px" ID="txtГAK" runat="server"
Text='<%=# Bind("ГAK") %>'></asp:TextBox>
</EditItemTemplate>
<FooterTemplate>
<asp:TextBox OnKeyPress="EnsureNumeric()" Width="20px" ID="txtNewГAK" runat="server"
></asp:TextBox>
</FooterTemplate>
<ItemTemplate>
<asp:Label ID="ГAK" runat="server" Text='<%=# Bind("ГAK") %>'></asp:Label>
</ItemTemplate>
</asp:TemplateField>

```



```

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

```

Ә қосымшасының жалғасы

```

<asp:TextBox OnKeyPress="EnsureNumeric()" Width="30px" ID="txtNewКСРС" runat="server"
></asp:TextBox>
</FooterTemplate>
<ItemTemplate>
<asp:Label ID="КСРС" runat="server" Text='<%=# Bind("КСРС") %>'></asp:Label>
</ItemTemplate>
</asp:TemplateField>

```

```

<asp:TemplateField HeaderText="ПГР">
<EditItemTemplate>
<asp:TextBox OnKeyPress="EnsureNumeric()" Width="20px" ID="txtПГР" runat="server"
Text='<%=# Bind("ПГР") %>'></asp:TextBox>
</EditItemTemplate>
<FooterTemplate>
<asp:TextBox OnKeyPress="EnsureNumeric()" Width="20px" ID="txtNewПГР" runat="server"
></asp:TextBox>
</FooterTemplate>
<ItemTemplate>
<asp:Label ID="ПГР" runat="server" Text='<%=# Bind("ПГР") %>'></asp:Label>
</ItemTemplate>
</asp:TemplateField>

```

```

<asp:TemplateField HeaderText="Зав.каф">
<EditItemTemplate>
<asp:TextBox OnKeyPress="EnsureNumeric()" Width="35px" ID="txtЗаведование_кафедрой"
runat="server" Text='<%=# Bind("Заведование_кафедрой") %>'></asp:TextBox>
</EditItemTemplate>
<FooterTemplate>
<asp:TextBox OnKeyPress="EnsureNumeric()" Width="35px"
ID="txtNewЗаведование_кафедрой" runat="server" ></asp:TextBox>
</FooterTemplate>
<ItemTemplate>
<asp:Label ID="Заведование_кафедрой" runat="server" Text='<%=#
Bind("Заведование_кафедрой") %>'></asp:Label>
</ItemTemplate>
</asp:TemplateField>

```

```

<asp:TemplateField HeaderText="Другое">
<EditItemTemplate>
<asp:TextBox OnKeyPress="EnsureNumeric()" Width="40px" ID="txtДругое" runat="server"
Text='<%=# Bind("Другое") %>'></asp:TextBox>
</EditItemTemplate>

```

```

    <FooterTemplate>
    <asp:TextBox OnKeyPress="EnsureNumeric()" Width="40px" ID="txtNewДругое"
runat="server" ></asp:TextBox>
    </FooterTemplate>
    <ItemTemplate>

```

Ә қосымшасының жалғасы

```

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

```

```

    <asp:TemplateField HeaderText="Bcero">
    <EditItemTemplate>
    <asp:TextBox OnKeyPress="EnsureNumeric()" Width="35px" Enabled="false" ID="txtBcero"
runat="server" Text='< %# Bind("Bcero") %>'></asp:TextBox>
    </EditItemTemplate>
    <FooterTemplate>
    <asp:TextBox OnKeyPress="EnsureNumeric()" Width="35px" Enabled="false"
ID="txtNewBcero" runat="server" ></asp:TextBox>
    </FooterTemplate>
    <ItemTemplate>
    <asp:Label ID="Bcero" runat="server" Text='< %# Bind("Bcero") %>'></asp:Label>
    </ItemTemplate>
    </asp:TemplateField>

```

```

    <asp:TemplateField HeaderText="Edit" ShowHeader="False">
    <EditItemTemplate>
    <asp:LinkButton ID="LinkButton1" runat="server" CausesValidation="True"
CommandName="Update" Text="Update"></asp:LinkButton>
    <asp:LinkButton ID="LinkButton2" runat="server" CausesValidation="False"
CommandName="Cancel" Text="Cancel"></asp:LinkButton>

```

```

    </EditItemTemplate>
    <FooterTemplate>

```

```

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

```

```

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

```

```

    </ItemTemplate>
    </asp:TemplateField>

```

```

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

```

</Columns>

<FooterStyle BackColor="White" ForeColor="#000066" />
<HeaderStyle BackColor="#006699" Font-Bold="True" ForeColor="White" />
<PagerStyle BackColor="White" ForeColor="#000066" HorizontalAlign="Left" />

Ә қосымшасының жалғасы

<RowStyle ForeColor="#000066" />
<SelectedRowStyle BackColor="#669999" Font-Bold="True" ForeColor="White" />
<SortedAscendingCellStyle BackColor="#F1F1F1" />
<SortedAscendingHeaderStyle BackColor="#007DBB" />
<SortedDescendingCellStyle BackColor="#CAC9C9" />
<SortedDescendingHeaderStyle BackColor="#00547E" />

</asp:GridView>

<asp:GridView ID="GridView1" runat="server" Visible="false" Width="16px">
</asp:GridView>

</asp:Panel></td></tr>

<tr><td> <div style="margin-left:236px">

<asp:GridView ID="GridView2" runat="server" BackColor="White"
BorderColor="#CCCCCC" BorderStyle="None" BorderWidth="1px" CellPadding="3">

<FooterStyle BackColor="White" ForeColor="#000066"/>
<HeaderStyle BackColor="#006699" Font-Bold="True" ForeColor="White" />
<PagerStyle BackColor="White" ForeColor="#000066" HorizontalAlign="Left" />
<RowStyle ForeColor="#000066" />
<SelectedRowStyle BackColor="#669999" Font-Bold="True" ForeColor="White" />
<SortedAscendingCellStyle BackColor="#F1F1F1" />
<SortedAscendingHeaderStyle BackColor="#007DBB" />
<SortedDescendingCellStyle BackColor="#CAC9C9" />
<SortedDescendingHeaderStyle BackColor="#00547E" />

</asp:GridView>

</div></td></tr>

</table>

</ContentTemplate>

</asp:UpdatePanel>

<asp:Button ID="Export" runat="server" Text="Экспорт"
style="height: 25px" width="65px" onclick="Export_Click"/>

</asp:Content>

<%@ Page Title="Администратор" Language="C#" AutoEventWireup="true"
CodeBehind="Admin.aspx.cs" Inherits="stys2.Admin" %>

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

Ә қосымшасының жалғасы

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title></title>
</head>

<body>
<center>
  <form id="form1" runat="server">
    <div>
      <h4><asp:Label ID="Lb_login" runat="server" Text="Label" ForeColor="Red" Font-
Bold="true"></asp:Label></h4>
      <p>
        <asp:Button ID="Next" runat="server" Text="Назад..."
        onclick="Next_Click" ForeColor="#0E8A06" Font-Bold="true" />
      </p>
      <asp:GridView ID="GridView_1" runat="server" AutoGenerateColumns="False"
ShowFooter="True" DataKeyNames="id"

OnRowDeleting="GridView_1_RowDeleting"
OnRowCancelingEdit="GridView_1_RowCancelingEdit"
OnRowEditing="GridView_1_RowEditing"
OnRowUpdating="GridView_1_RowUpdating"
OnRowCommand="GridView_1_RowCommand"
>
<Columns>

  <asp:TemplateField HeaderText="Логин пользователя" SortExpression="Логин пользователя">
    <EditItemTemplate> <asp:TextBox Width="200px" ID="txtLogin" runat="server" Text='<%#
Eval("login") %>'> </asp:TextBox> </EditItemTemplate>
    <FooterTemplate> <asp:TextBox Width="200px" ID="txtNewLogin" runat="server">
</asp:TextBox> </FooterTemplate>
    <ItemTemplate> <asp:Label ID="Label1" runat="server" Text='<%# Bind("login") %>'>
</asp:Label> </ItemTemplate>
  </asp:TemplateField>

  <asp:TemplateField HeaderText="Пароль пользователя" SortExpression="Пароль
пользователя">
    <EditItemTemplate> <asp:TextBox Width="200px" ID="txtPassword" runat="server"
Text='<%# Eval("password") %>'> </asp:TextBox> </EditItemTemplate>
    <FooterTemplate> <asp:TextBox Width="200px" ID="txtNewPassword" runat="server">
</asp:TextBox> </FooterTemplate>
```

```

    <ItemTemplate> <asp:Label ID="Label2" runat="server" Text='< %# Bind("password") %>'>
</asp:Label> </ItemTemplate>
</asp:TemplateField>

```

Ә қосымшасының жалғасы

```

<asp:TemplateField HeaderText="Дата создания пользователя" SortExpression="Дата создания
пользователя">
    <EditItemTemplate> <asp:TextBox Width="200px" ID="txtDate" runat="server" Text='< %#
Eval("date") %>'> </asp:TextBox> </EditItemTemplate>
    <FooterTemplate> <asp:TextBox Width="200px" ID="txtNewDate" runat="server">
</asp:TextBox> </FooterTemplate>
    <ItemTemplate> <asp:Label ID="Label3" runat="server" Text='< %# Bind("date") %>'>
</asp:Label> </ItemTemplate>
</asp:TemplateField>

```

```

<asp:TemplateField HeaderText="Edit" ShowHeader="False">
<EditItemTemplate>
    <asp:LinkButton ID="LinkButton1" runat="server" CausesValidation="True"
CommandName="Update" Text="Update"></asp:LinkButton>
    <asp:LinkButton ID="LinkButton2" runat="server" CausesValidation="False"
CommandName="Cancel" Text="Cancel"></asp:LinkButton>

```

```

</EditItemTemplate>
<FooterTemplate>

```

```

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

```

```

</FooterTemplate>
<ItemTemplate>
    <asp:LinkButton ID="LinkButton1" runat="server" CausesValidation="False"
CommandName="Edit" Text="Edit"></asp:LinkButton>
</ItemTemplate>
</asp:TemplateField>
<asp:CommandField HeaderText="X" DeleteText="X" ShowDeleteButton="True"
ShowHeader="True" />
</Columns>

```

```

    <FooterStyle BackColor="White" ForeColor="#000066" />
    <HeaderStyle BackColor="#006699" Font-Bold="True" ForeColor="White" />
    <PagerStyle BackColor="White" ForeColor="#000066" HorizontalAlign="Left" />
    <RowStyle ForeColor="#000066" />
    <SelectedRowStyle BackColor="#669999" Font-Bold="True" ForeColor="White" />
    <SortedAscendingCellStyle BackColor="#F1F1F1" />

```

```

        <SortedAscendingHeaderStyle BackColor="#007DBB" />
        <SortedDescendingCellStyle BackColor="#CAC9C9" />
        <SortedDescendingHeaderStyle BackColor="#00547E" />
    </asp:GridView>
</div>

```

Ә қосымшасының жалғасы

```

</form>
</center>
</body>
</html>
<%@ Page Title="Нераспределенная Нагрузка" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="Neraspredelennaia_Nagruzka.aspx.cs"
Inherits="stys2.Neraspredelennaia_Nagruzka" %>
<asp:Content ID="Content1" ContentPlaceHolderID="HeadContent" runat="server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="server">
    <asp:ScriptManager ID="myScriptManager" runat="server">
</asp:ScriptManager>

<asp:UpdatePanel ID="JokeUpdatePanel" runat="server">
    <ContentTemplate>
        <table>
            <tr>
                <td>
                    <asp:DropDownList ID="Sortyrovka_Deissiplini" runat="server" Height="22px" Width="142px"
                        AutoPostBack="true"
                        onselectedindexchanged="Sortyrovka_Deissiplini_SelectedIndexChanged"
></asp:DropDownList> +
                    <asp:DropDownList ID="Sortyrovka_Semestr" runat="server" Height="22px" Width="142px"
                        AutoPostBack="true"
                        onselectedindexchanged="Sortyrovka_Semestr_SelectedIndexChanged" >
                        <asp:ListItem>Все семестры</asp:ListItem>
                        <asp:ListItem>1</asp:ListItem>
                        <asp:ListItem>2</asp:ListItem>
                    </asp:DropDownList> +
                    <asp:DropDownList ID="Sortyrovka_Forma_ovuchenie" runat="server" Height="22px"
Width="142px"
                        AutoPostBack="true"
                        onselectedindexchanged="Sortyrovka_Forma_ovuchenie_SelectedIndexChanged"></asp:DropDownList>
+
                    <asp:DropDownList ID="Sortyrovka_Gruppi" runat="server" Height="22px" Width="142px"
                        AutoPostBack="true"
                        onselectedindexchanged="Sortyrovka_Gruppi_SelectedIndexChanged"
></asp:DropDownList>
                    <asp:Button ID="Sbros" runat="server" Text="Сброс"
                        style="height: 25px" width="75px" onclick="Sbros_Click" />

                    <asp:Button ID="Update" runat="server" Text="Обновить"
                        style="height: 25px" width="75px" onclick="Update_Click" />

```

Ә қосымшасының соңы

```
</td>
</tr>
<tr><td>

<asp:Panel runat="server" ID="MainPanel" ScrollBars="Auto" Height="315" Width="100%">
<asp:GridView ID="GridView1" runat="server" BackColor="White" BorderColor="#CCCCCC"
BorderStyle="None" BorderWidth="1px" CellPadding="3">

    <FooterStyle BackColor="White" ForeColor="#000066"/>
    <HeaderStyle BackColor="#006699" Font-Bold="True" ForeColor="White" />
    <PagerStyle BackColor="White" ForeColor="#000066" HorizontalAlign="Left" />
    <RowStyle ForeColor="#000066" />
    <SelectedRowStyle BackColor="#669999" Font-Bold="True" ForeColor="White" />
    <SortedAscendingCellStyle BackColor="#F1F1F1" />
    <SortedAscendingHeaderStyle BackColor="#007DBB" />
    <SortedDescendingCellStyle BackColor="#CAC9C9" />
    <SortedDescendingHeaderStyle BackColor="#00547E" />
</asp:GridView>
</asp:Panel>

<tr><td> <div style="margin-left:236px">
<asp:GridView ID="GridView2" runat="server" BackColor="White"
BorderColor="#CCCCCC" BorderStyle="None" BorderWidth="1px" CellPadding="3">

    <FooterStyle BackColor="White" ForeColor="#000066"/>
    <HeaderStyle BackColor="#006699" Font-Bold="True" ForeColor="White" />
    <PagerStyle BackColor="White" ForeColor="#000066" HorizontalAlign="Left" />
    <RowStyle ForeColor="#000066" />
    <SelectedRowStyle BackColor="#669999" Font-Bold="True" ForeColor="White" />
    <SortedAscendingCellStyle BackColor="#F1F1F1" />
    <SortedAscendingHeaderStyle BackColor="#007DBB" />
    <SortedDescendingCellStyle BackColor="#CAC9C9" />
    <SortedDescendingHeaderStyle BackColor="#00547E" />
</asp:GridView>
</div></td></tr>
</table>
</ContentTemplate>
</asp:UpdatePanel>
<asp:Button ID="Export" runat="server" Text="Экспорт" style="height: 25px"
width="75px" onclick="Export_Click" />

</asp:Content>
```