



**Коммерциялық емес
акционерлік қоғам**

**АЛМАТЫ
ЭНЕРГЕТИКА ЖӘНЕ
БАЙЛАНЫС
УНИВЕРСИТЕТІ**

Ақпараттық
жүйелер
кафедрасы

МӘЛІМЕТТЕР ҚОРЛАРЫНЫҢ ЖҮЙЕСІ

5В070300 – Ақпараттық жүйелер
мамандығының студенттері үшін
зертханалық жұмыстарды орындауға арналған әдістемелік нұсқаулықтар

Алматы 2014

ҚҰРАСТЫРУШЫЛАР: Б.К. Каирбаева, Н.А. Водолазкина. Мәліметтер қорларының жүйесі. 5B070300 – Ақпараттық жүйелер мамандығының студенттеріне арналған зертханалық жұмыстарды орындауға арналған әдістемелік нұсқаулар. – Алматы: АЭЖБУ, 2014. –38 бет.

Әдістемелік нұсқаулар «Мәліметтер қорларының жүйесі» пәні бойынша компьютерлік сыныптарда зертханалық жұмыстарды орындауға арналған.

Әрбір зертханалық жұмыс оны орындауға арналған, нақты мысалдармен бекітілген және қажетті түсіндірмелер, сонымен қатар тапсырмалар нұсқалары мен бақылау сұрақтары көрсетілген, теориялық мағлұматтар берілген.

Әдістемелік нұсқаулар 5B070300 – Ақпараттық жүйелер мамандығының студенттеріне арналған.

Без. – 32, кесте – 1, әдебиеттер – 9 атау.

Пікір беруші: Р.Қ. Қойлыбаева

«Алматы энергетика және байланыс университеті» коммерциялық емес акционерлік қоғамының 2014 жылғы баспа жоспары бойынша басылады.

© «Алматы энергетика және байланыс университеті» КЕАҚ, 2014 ж.

Мазмұны

Кіріспе	4
1 Зертханалық жұмыс № 1. Database Desktop - пен жұмыс, деректер қорларының бүркеме атауларын (псевдоним) құру	5
2 Зертханалық жұмыс № 2. Деректер қорында рұқсат және басқару компоненттерімен жұмыс. Мәтінді енгізу және редакциялау	13
3 Зертханалық жұмыс №3. Деректер қорында жазбалар іріктемесі. Деректер қорында жазбаларды іріктеу және іздеу	21
4 Зертханалық жұмыс № 4. Жазбаларды филтрлеу	26
5 Зертханалық жұмыс № 5. Кестелер арасында байланыс орнату	28
6 Зертханалық жұмыс № 6. Есептемелер құру	32
А қосымшасы	37
Әдебиеттер тізімі	38

..

Кіріспе

Деректер базасы жобасын құрғаннан кейін оның нақты *деректер қорлары жүйесінде* (ДҚЖ) іске асырылу жұмыстарына кіріссе болады. Заманауи ақпаратты басқару технологиялар толқыны *реляциялық* деректер қорларын басқару жүйелерін қолдануға негізделген.

Деректер қорларын жобалау және құру деректермен тиімді жұмыс жасау үшін жеткіліксіз: берілген деректерге жүгінуге, оларды толықтыруға және өзгертуге мүмкіндік болуы қажет. Клиенттік қосымша осыған – қолданушыға ыңғайлы формада деректермен жұмыс жасауға мүмкіндік беретін арнайы бағдарламаға – негізделген. Жалпы бір деректер қорымен көптеген әртүрлі қосымшалар жұмыс жасай алады. Бір деректер қорымен жұмыс жасайтын қосымшаларды қарастырғанда, олар бір-бірінен тәуелсіз және қатарлас жұмыс жасай алады деп қаралады. ДҚЖ бірыңғай деректер қорымен көптеген қосымшалардың жұмысын және деректер қорларындағы басқа қосымшалармен енгізілген барлық өзгерістерін ескереуге және олардың әр қайсысы дұрыс жұмыс жасау жұмысын қамтамасыз етуге арналған [1].

Delphi бағдарламалау ортасы аяқталған Windows қосымшаларын салыстырарлықтай жеңіл және тез құруға рұқсат береді. Сондықтан, ол RAD деген атауға ие болды (Rapid Application Development – қосымшаларды тез құру ортасы).

Delphi бағдарламалау ортасында кірістірілген компилятор бар, ол компьютер процессорына түсінікті, бағдарлама мәтінін машиналық код түріне айналдырады. Қазіргі кезде бұл компилятор ең тез болып келеді және Windows қосымшаларын құру үшін қажетті бағдарламалау ортасының жоғары өнімділігін қамтамасыз етуге мүмкіндік береді. Delphi бағдарламалау ортасының компиляторы дербес exe-файлдарын (атқарылатын файлдар) құрады. Сондықтан Delphiде құрылған қосымшалар Delphi ортасы орнатылмаған компьютерлерде жұмыс жасайды.

Ұсынылған зертханалық жұмыстарды орындауға арналған әдістемелік нұсқаулықтар студенттермен Delphi ортасында деректер қорларын құру негіздерін игеру үшін құрылған. Әр зертханалық жұмысты орындау нұсқаулығы жұмысты орындауға қажетті ілеспелі курстан ақпараттан және ұқсас тапсырманы орындау мысалдарынан тұрады.

Әр зертханалық жұмыс бойынша студент есеп беруді ұсынуға және жұмысты қорғауға міндетті.

1 Зертханалық жұмыс № 1. Database Desktop - пен жұмыс, деректер қорларының бүркеме атауларын (псевдоним) құру

Жұмыс мақсаты:

- 1) Database Desktop утилитасымен жұмыс.
- 2) Database Desktop көмегімен кестелерді құру және редакциялау процессін оқып білу.

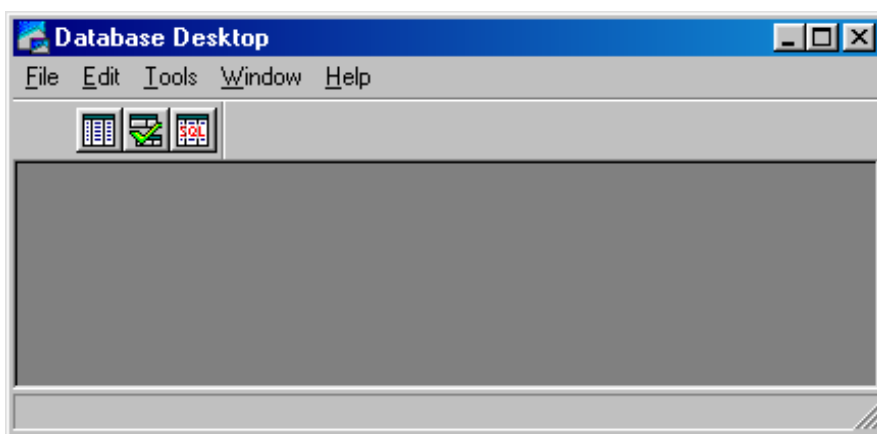
Жалпы ақпарат

Database Desktop.

Database Desktop – Delphi-мен бірге ұсынылатын утилита. Ол локалді деректер қорларының әр түрлі форматты кестелермен – Paradox және dBase, сонымен қатар InterBase SQL-серверлі деректер қорлары, Oracle, Informix, Sybase (SQL Links қолданып) – интерактивті жұмыс жасау үшін беріледі. Утилитаның орындалатын файлы DBD32.EXE деп аталады. Database Desktop орындау үшін оның иконкасына екі рет басу қажет [1].

Database Desktop іске қосу:

- 1) Delphi-ді қосу.
- 2) Delphi мәзірінде Tools бөлімін таңдау.
- 3) Пайда болған тізімде Database Desktop жолын таңдау.
- 4) Пуск батырмасын басу.
- 5) Бас мәзірден Программы жолын таңдау.
- 6) Пайда болған тізімнен Delphi 7 жолын таңдау.
- 7) Келесі тізімде Database Desktop жолын таңдау.
- 8) Database Desktop ашылғаннан кейін экранда терезе пайда болады (1 сурет).

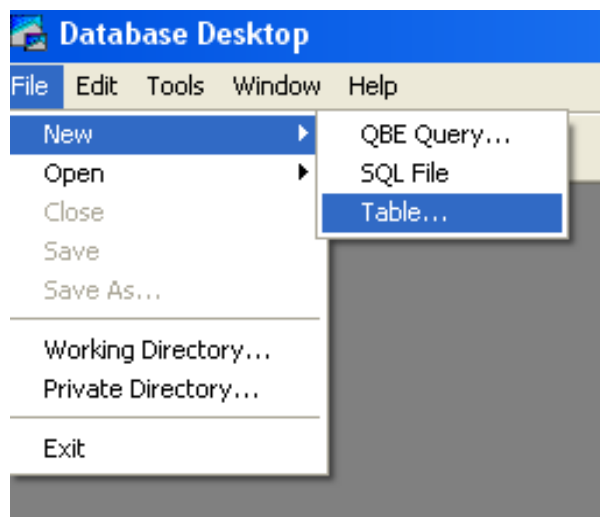


1 сурет – Database Desktop терезесі

Database Desktop-те кесте құру:

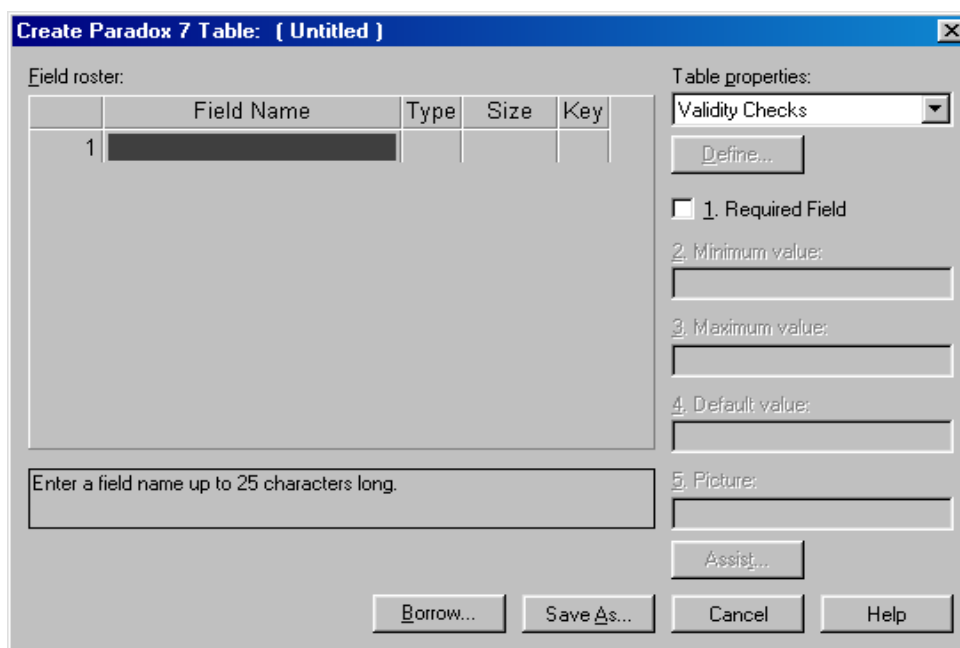
Database Desktop бағдарламасын қосу;

Пайда болған терезеде: File ► New ► Table таңдау (2 сурет).



2 сурет – Database Desktop терезесі

Кесте форматы ретінде Paradox таңдаймыз. Осыдан кейін кесте құру терезесі пайда болады. Оның ішінде кесте өрістерін (бағандарын) және олардың типтерін анықтауға болады (3 сурет) [1].



3 сурет – Paradox кестесін құру

Терезені ашқаннан кейін бірден оның оң жағында Table properties тізімінде Validity Checks пункті таңдалған. Ол өрістер (бағандар) құрамын бақылауға мүмкіндік береді. Required Fields жалаушасы көмегімен жаңа жазу енгізгенде міндетті түрде өрісті толтыруды талап етуге болады. Сонымен қатар Minimum Value және Maximum Value жолдарында сандық өрістің минималды және максималды мәнін бақылауға болады. Default Value жолында өрістің бірден қабылданған (по умолчанию) мәнін көрсетуге болады – жаңа жазу енгізген кезде мән бұл өріске BDE орнатады. Picture жолы көмегімен өріс мәнін автоматты форматтау үлгісін (шаблонын) беруге

болады. Мысалы, егер (###)###-#### үлгісі берілсе және өріске 9054005647 жолы енгізілсе, ол автоматты түрде (905)400-5647 түріне келтіріледі.

Батырмалар:

- 1) Borrow... – кесте құрылымын басқа кестеден көшіруді іске асырады;
- 2) Save as... – кесте құрылымындағы өзгерістерді сақтайды;
- 3) Cancel – сақтаусыз шығып кету;
- 4) Help – анықтаманы шақыру.

Кесте құрылымын анықтау үшін бұл терезеде келесі графаларды толтыру қажет:

1) Field Name – Өріс атауы.

2) Type – Өріс типі. Тышқанның оң жақ батырмасын басу немесе аралық (пробел) клавишін басу арқылы мүмкін болатын типтер тізімін шақырады.

3) Size - Өлшем. Өріс өлшемін анықтайды. Өрістер типтерінің барлығында бірдей өлшемдер бола бермейді. Типтердің көпшілігінде өзгертілмейтін стандартты өлшемдер болады. Негізінен өлшемдер (Alpha), бинарлы (Binary) және тағы басқа жолдық типтерде өзгереді.

4) Key – Кілт. Тышқан батырмасын екі рет басу кілттік өрісті анықтайды. Тек қана бірінші өрістер (баған) кілттік бола алады. Екінші өріс (баған) біріншімен бірге ғана кілттік бола алады [2].

1 кесте – Paradox форматындағы өрістер типтері

Alpha	1-255 байт ұзындықты жол, кез келген басылып енгізілетін символдардан тұрады
Number	8 байт ұзындықты сандық өріс, мәні оң немесе теріс бола алады. Сандар диапазоны – 15 орынды сандармен 10-308 бастап 10308 дейін
\$ (Money)	Сандық өріс, мәні оң немесе теріс бола алады. Бірден қабылданған (по умолчанию), ондық нүктені және қаржы белгісін бейнелеу үшін форматталған болып келеді
Short	2 байт ұзындықты сандық өріс, -32768-дан 32767-ға дейінгі диапазондағы тек қана бүтін сандардан тұра алады
Long Integer	4 байт ұзындықты сандық өріс, -2147483648-дан 2147483648-ға дейінгі диапазондағы бүтін сандардан тұра алады
# (BCD)	сандық өріс, BCD (Binary Coded Decimal) форматындағы деректерден құралады. Басқа сандық форматтарға қарағанда есептеу жылдамдығы сәл азырақ, дегенмен нақтылығы әлде қайда жоғары. Ондық нүктеден кейін 0-32 цифрлардан тұра алады
Date	4 байт ұзындықты күнтізбе өрісі, біздің дәуірімізге дейінгі 9999 жылдың 1 қаңтарнан бастап – біздің дәуіріміздің 9999 жылының 31 желтоқсанына дейінгі күндерден құрыла алады. Толы жылдарды (високосные года) дұрыс өңдейді

	және күндердің дұрыстығын тексеретін кірістірілген механизмі бар
Time	4 байт ұзындықты уақыт өрісі, түн жарымынан бастап миллисекундтармен берілген уақыттан құрылған және 24 сағатпен шектелген
@ (Timestamp)	8 байт ұзындықты жалпыланған күнтізбе өрісі – күндерден және уақыттан құрылған
Memo	Қосынды ұзындығы 255 байттан асатын символдарды сақтауға арналған өріс. Кез келген ұзындыққа ие бола алады. Бірақ, кестені құру кезінде көрсетілетін өлшем кестеде сақталатын символдар санын білдіреді (1-240) – қалған символдар .MV кеңейтуі бар бөлек файлда сақталады
Formatted Memo	Мемо-ға ұқсас өріс, мәтін шрифтін беру мүмкіндігі қосылған. Сонымен қатар кез келген өлшемді бола алады. Бұл жағдайда, кестені құру кезінде көрсетілетін өлшем кестеде сақталатын символдар санын білдіреді (1-240) – қалған символдар «.MV» кеңейтуі бар бөлек файлда сақталады. Бірақ, стандартты қамсыздандырудағы Delphi-де Formatted Memo өрістерімен жұмыс жасау мүмкіндіктері жоқ
Graphic	Графикалық ақпаратты өріс. Кез келген өлшемді бола алады. Өлшем мәні Formatted Memo-дағы тәрізді. Database Desktop Graphic типті өрістер құрай алады, бірақ оларды толтыру тек қана қосымшаларда іске асырылады
OLE	OLE-деректерінен құрылған өріс (Object Linking and Embedding) – бейнелер, дыбыс, видео, документтер – өздерінің өңделуі үшін өздерін құрған қосымшаларды шақырады. Кез келген өлшемді бола алады. Өлшем мәні Formatted Memo-дағы тәрізді. Database Desktop OLE типті өрістер құрай алады, бірақ оларды толтыру тек қана қосымшаларда іске асырылады. Delphi “турасынан” OLE-өрістерімен жұмыс жасай алмайды, бірақ бұл ағындарды қолдану жолымен жеңіл іске асырылады
Logical	1 байт ұзындықты өріс, тек қана екі мән қабылдай алады – Т (true, ақиқат) немесе F (false, жалған). Бас әріптер және кіші әріптер рұқсат етіледі
+ (Autoincrement)	4 байт ұзындықты өріс, long integer типті редакцияланбайтын (read-only) мәннен құралған. Бұл өрістің мәні автоматты түрде (1-ден бастап) 1 мәнді қадаммен ұлғаяды – бұл бірегей жазу идентификаторын құру үшін өте ыңғайлы (жазудың физикалық нөмірі оның идентификаторы бола алмайды, өйткені Paradox-та ондай

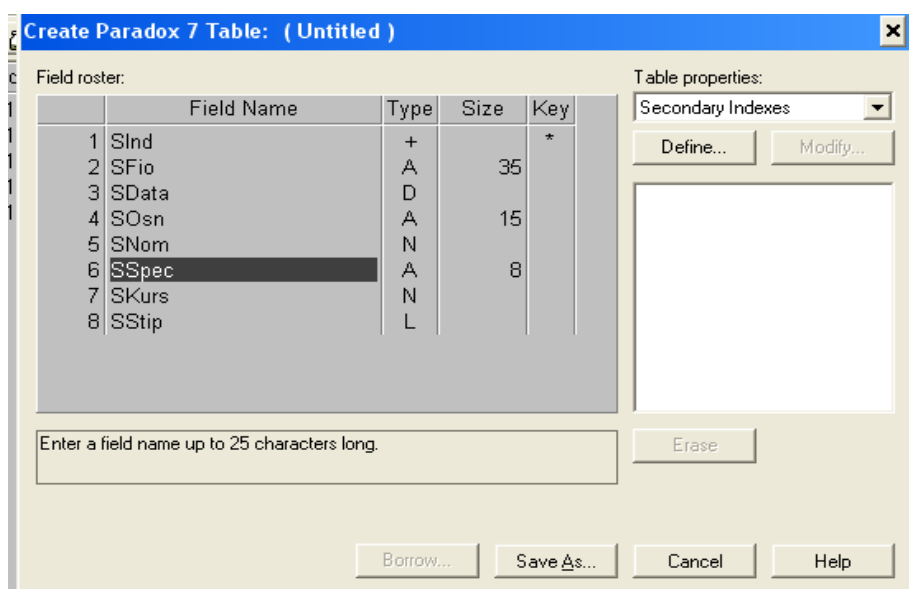
	жоқ. InterBase-те де жазулардың физикалық нөмірлері жоқ, бірақ Autoincrement өрісі де қарастырылмаған. Оны, бәрінен де триггерлерде қолдануға ыңғайлы, Gen_id кірістірілген функциясы сәтті алмастырады)
Binary	Кез келген екілік ақпараттан құралған өріс. Сонымен қатар кез келген өлшемді бола алады. Бұл жағдайда, кестені құру кезінде көрсетілетін өлшем кестеде сақталатын символдар санын білдіреді (1-240) – қалған символдар «.MB» кеңейтуі бар бөлек файлда сақталады. Бұл InterBase-тегі BLOB өрісінің толықтай аналогы
Bytes	1-255 байт ұзындықты цифрлар жолы, кез келген деректерден құралады

А, Б тапсырмалары

Керекті нұсқаны таңдап, ДҚ бүркеме атауларын (псевдоним) құру (А қосымшасы), 2 кестеден құру және оларды редакциялау (Paradox форматы). Студенттер барлық тапсырмаларды екі нұсқа бойынша орындаулары қажет (А, Б).

Әдістемелік нұсқаулықтар

Студенттер туралы ақпараттары бар кесте құру. Келесі ақпараттарды енгізу қажет: ТАЖ (ФИО), дүниеге келу күні, студенттің университетке қабылдану бұйрығы, сынақ кітапшасы нөмірі, мамандығы, курсы, стипендия алу/алмауы. Барлық мәтіндік өрістерде өлшемдерін көрсету қажет (4 сурет) [2].



4 сурет – Кесте құру

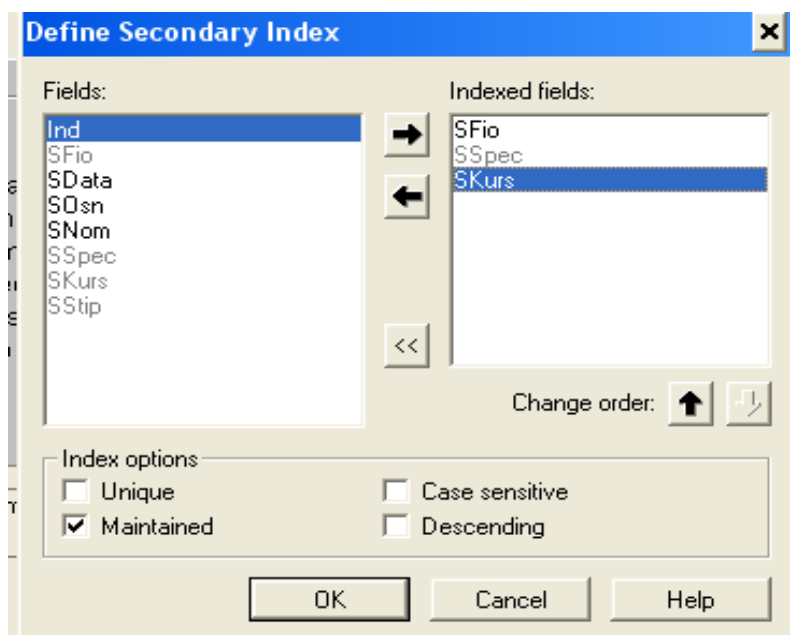
Құрылған кестені Student.db атымен сақтаймыз және кесте құру терезесін жабамыз. Кейбір жағдайда құрылған кестені (өрістерді қосу, өзгерту, өшіру немесе кесте қасиеттерін өзгерту үшін) редакциялау қажет болуы мүмкін.

Кестені редакциялау

Екінші индекстерді анықтау:

Student.db кестесін ашу (File►Open►Table).

Table мәзірінен Restructure пунктін таңдау. Кесте өрістерін редакциялау терезесі ашылады. Table properties құлама тізімінен Secondary Indexes таңдау және Define батырмасын басы. Define Secondary Index терезесінде екінші индекстер анықталады (5 сурет).



5 сурет – Define Secondary Index терезесі

Index options тобының жалаушаларының көмегімен индекстің келесі ерекшеліктерін анықтауға болады:

Unique – индекс құрамында бірегей мағына болады;

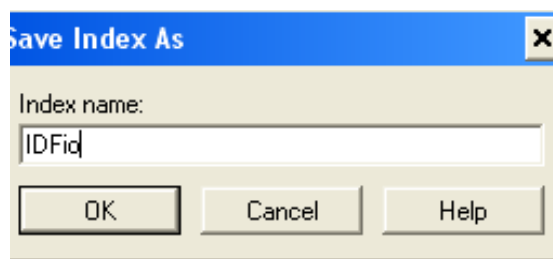
Maintained – индексті өрістер мағыналардың өсуімен іріктеледі;

Case sensitive – индекс мәтіндік өрістердегі әріп регистріне сезімтал;

Descending – индексті өрістер мағыналардың кемуімен іріктеледі.

Fields тізімінен «SFio» таңдау және оң жаққа бейнеленген көрсеткіші бар батырманы басы. Indexed fields тізімінде (индекстелген өріс) «SFio» пайда болады. Дәл соны «SSpec» және «SKurs» өрістерімен жасау. Define Secondary Index терезесін жабу.

Пайда болған терезеде IDFio индексі атын енгізу және «OK» басы (6 сурет) [2].



6 сурет – IDFio индексі атын енгізу

Енді, факультетте тек екі мамандық қана бар болғандықтан, «SSpec» өрісі типі қайта тағайындауға болады. Өріс типі Logical болса, жұмысқа ыңғайырақ болады. Тілтуірді «SSpec» өріс типіне апару және «L» өріс типін жазу.

Тілдік драйверді анықтау. Оны орыс тіліндегі мәтінді дұрыс бейнелеу үшін жасау қажет. Student.db кестесі өрістерін редакциялау терезесін ашу.

Table properties құлама тізімінде Table Language таңдау және Modify батырмасын басу. Пайда болған терезеде тізімнен Pdox ANSI Cyrillic таңдау.

Кестені сақтау. Логикалық тип үшін бірден қабылданған мән False береміз [2].

Student.db кестесінде деректермен толтыру:

- 1) Кестені ашу (File ► Open ► Table).
- 2) Table ► Edit data таңдау немесе F9 клавишін басу.
- 3) Бірнеше жазуларды құру.

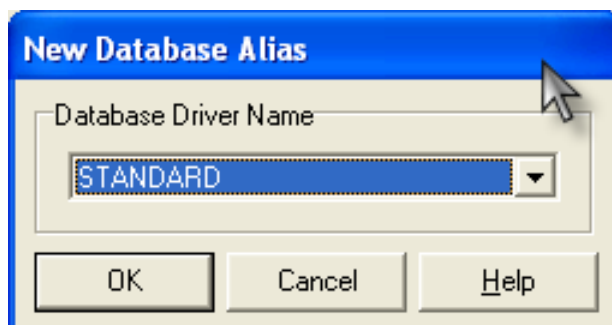
Бүркеме атауларын (псевдоним) құру:

Бүркеме атау ДҚ файлдарының орналасу жерін көрсетеді және каталогты белгілеу үшін арнайы атау болып келеді. Бүркеме атауларды қолдану ДҚ файлдарының басқа компьютерлердегі басқа каталогтарға орын ауыстырылуын әлдеқайда жеңілдетеді. Бұл жағдайда ДҚ кестелеріне рұқсатты іске асыратын қосымшаларды өзгерту талап етілмейді. Егер қосымшада кестенің орналасуы бүркеме атау көмегімен көрсетілсе, ДҚ орын ауыстырғаннан кейін қосымшаның жұмысын қамтамасыз ету үшін бүркеме атау көрсететін бағытты (жолды) өзгертсе жеткілікті. Егер қосымшада ДҚ-на бағыт айқын түрде, бүркеме атаусыз, көрсетілсе ДҚ орын ауыстырғаннан кейін қосымшаның өзін орын ауыстыру – бастапқы код ішіне өзгерістер енгізіп, оны қайтадан трансляциялау қажет [3].

Бүркеме атауды тіркеу:

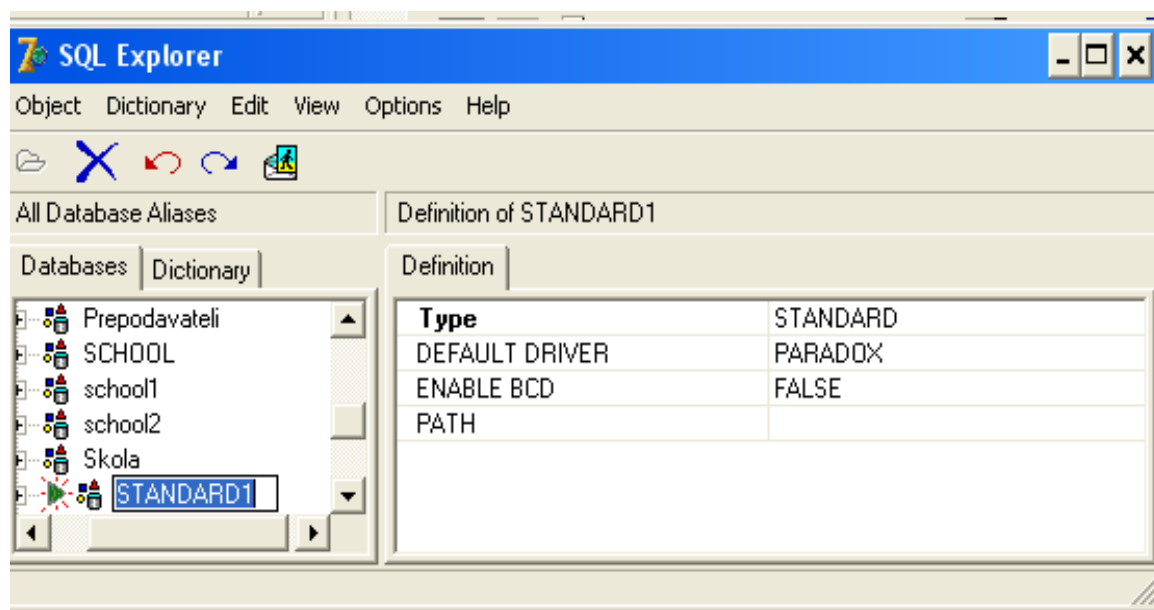
Database ► Explore командаларымен қосылатын SQL Explorer қосымшасын қолданамыз. Терезенің сол жақ бөлігінде BDE жүйесінде тіркелген барлық деректер қорларының тізімі, оң жағында – тізімде таңдалған ағымдағы қордың қасиеттері келтірілген.

Деректер қорына бүркеме атау құрамыз. Бұл үшін Object►New командасын орындаймыз және драйвер таңдау терезесінде Standart мәнін көрсетеміз (7 сурет).



7 сурет – драйвер таңдау терезесі

«OK» батырмасын басқаннан кейін тізімде жасыл үшбұрышпен белгіленген, жаңа элемент пайда болады (8 сурет).



8 сурет – Жасыл үшбұрышпен белгіленген, жаңа элемент

Бірден қабылдану бойынша Standard1 деректер қорының аты құрылады. Default Driver қасиеттерінде (Бірден қабылданған драйвер) Paradox мәні болуы қажет. Path қасиеттерінде, біздің кесте сақталған, каталогты көрсетеміз.

Енді BDE жүйесінде тіркелген базаны сақтаймыз, ол үшін Student объектінің контекстті мәзірінде Apply таңдаймыз. Өзгерістерді сақтау қажеттігі туралы сұраққа Yes басамыз. Осыдан кейін кесте Student атауымен BDE ортасынан қолжетімді [3].

Sql Explorer-ді жабыңыз.

Бақылау сұрақтары.

1. Database Desktop утилитасымен жұмыс.
2. Database Desktop қосу.
3. Бүркеме атауды құру.
4. Бүркеме атауды тіркеу.
5. Кестені мәліметтермен қалай толтыруға болады?
6. Paradox кестенің қандай өрістерін таңдайды?
7. Кестені редакциялау.
8. Индекстермен жұмыс.

2 Зертханалық жұмыс № 2. Деректер қорында рұқсат және басқару компоненттерімен жұмыс. Мәтінді енгізу және редакциялау

Жұмыс мақсаты:


- 1) Delphi ортасында деректер қорларымен жұмыс жасау үшін қосымшаларды құрудың бастапқы этаптарын оқып қарастыру:
 - ДҚ рұқсат беру компоненттерімен жұмыс: TTable, TDataSource;
 - ДҚ басқару компоненттерімен жұмыс: TDBGrid, TDBNavigator.
- 2) ДҚ мәтінді енгізіп және өңдеп үйрену.

Деректер қорларымен жұмыс жасау үшін қосымшаларды құру әдісінің қарапайым бағдарламалар құру әдісінен ешқандай айырмашылығы жоқ: формаға керекті компоненттер қосылады, компоненттердің қасиеттерінің мәне орнатылады, оқиғаларды өңдеудің керекті процедуралары құрылады.

Деректер қорларымен жұмыс жасайтын қосымша мәліметтерге қолжетімділікті қамтамасыз ететін, өрістердің құрамын қарауға және өңдеуге мүмкіндік беретін компоненттерден құрылуы керек. Деректерге рұқсат беретін компоненттер Data Access қосымша бетінде және BDE компоненттер палитрасында, ал деректерді бейнелеу компоненттері – Data Controls қосымша бетінде орналасқан [3].

Деректер қорларына рұқсат және басқарудың негізгі компоненттері.

Деректер қорларына рұқсат компоненттері:

TTable  - ДҚ кестесімен қарым-қатынасты қамтамасыз етеді. TTable компоненті мәліметтерді қайдан алу қажет екенін және мәліметтер жиынын қандай өрістер құрайтынын көрсетеді. TTable компонентінің келесі негізгі қасиеттері бар:


- 1) DatabaseName – деректер қоры.
- 2) TableName – кесте аты.
- 3) Active – кестені активтеу (True мәні оны активтейді).

DatabaseName қасиеті кесте орналасқан деректер қорын анықтайды.

Бұл қасиет келесіден тұра алады:


- 1) бүркеме атау (псевдоним);
- 2) жергілікті ДҚ үшін бағыт (жол);

- 3) Local InterBase үшін деректер қорының файлының жолы және атауы;
 - 4) TDatabase компоненті арқылы анықталған жергілікті бүркеме атау.
- TableName қасиеті* деректер қоры кестесінің атын анықтайды [3].

TDataSource  - деректер қоры мен деректерді басқару компоненттері арасындағы байланысты анықтайды. TDataSource компоненті нақты деректер қорымен байланысқан Table1 компоненті мен, қолданушы кестемен қарым-қатынас жасайтын, DBGrid1 және DBNavigator1 визуалды компоненттері арасында аралық буын болып келеді.

Көп жағдайда DataSource жасайтын нәрсе – ол DataSet қасиеттерінде қажетті Ttable көрсету. Содан кейін DBGrid немесе DBNavigator визуалды компоненттерінде DataSource қасиеттерінде қазіргі уақытта қолданылатын TdataSource көрсетіледі.

Data Contorls палитрасынан мәліметтерді басқару компоненттері:

TDBGrid  – ДҚ кестесінің құрамын тор түрінде бейнелейді. Онда бағандар кесте өрістеріне, ал жолдар кесте жазуларына сәйкес.

Компонент келесі қасиеттерге ие:

Data Source – мәліметтер көзі болып табылатын TDataSource типті компонентіне сілтемесі бар;

Editor Mode – Егер құрамы true болса, қолданушы F2 немесе Enter батырмаларын басқаннан кейін ұяшықты өңдей алады. Егер Option қасиеті goEditing немесе goAlwaysShowEditor мәндеріне ие болса, елемейді;

Option – компоненттің түрін және әрекетін анықтайды;

dgEditing – мәліметтер жиынын өзгертуге рұқсат;

dgAlwaysShowEditor – автоматты түрде бағанды ерекшелеу кезінде редакторлау режиміне көшіреді;

dgTitles – бағандар тақырыптарын көрсетеді;

dgIndicator – ең сол жақтағы белгіленген бағанда ағымдағы жолдың индикаторын көрсетеді;

dgColumnResize – қолданушыға бағандардың енін қолмен өзгертуге рұқсат етеді;

dgColLines – бөлуші вертикальді сызықтарды көрсетеді;

dgRowLines – бөлуші горизонтальді сызықтарды көрсетеді;

dgTabs – Tab батырмасы көмегімен бағаннан бағанға өтуге мүмкіндік береді;


dgRowSelect – таңдалған жолды толықтай түспен ерекшелеуге мүмкіндік береді;

dgAlwaysShowSelection – егер компонент енгізу фокусын жоғалтса ағымдағы жолдың ерекшеленуі сақталады;

dgConfirmDelete – жолдың өшірілуі дәлелденуі қажет;

dgCancelOnExit – егер қолданушы бос жолды қойып және одан шығып кетсе, ол мәліметтер жиынына орнатылмайды;

dgMultiSelect – жолдардың көптік таңдауына мүмкіндік береді.

TDBNavigator  – жазулардың орынауыстырылуы және өңделуі іске асырылады (батырмалардың түрі және қызметі DBNavigator пунктінде көрсетілген). DataSource қасиет көмегімен компонент TDataSource мәліметтердің керекті көзімен байланысады – бұл дұрыс жұмыс жасау үшін керекті нәрсенің бәрі. ConfirmDelete қасиеті жазуларды өшіруді растау өтінішімен диалогты терезені шығаруды басқарады (бұл қасиеттің True мәні терезені шығарады) [3].

А, Б тапсырмалары

Керекті нұсқаны таңдап, Delphi бағдарламалық ортасында формаға екі кестеден орнату (А қосымшасы), деректер қорларына рұқсат және басқару компоненттерін қолданып формалар құру. Студенттер барлық тапсырмаларды екі нұсқа бойынша орындаулары қажет (А, Б).

Әдістемелік нұсқаулықтар

Студенттер қосымшасын құру:

- 1) Delphi-ді қосу.
- 2) Caption қасиетінде форма атын Студенттер деп өзгерту.
- 3) Формаға TTable компонентін орнату.

Table1 компонентінің келесі қасиеттерін анықтау:

- 1) Бүркеме атауды анықтау – объектілер инспекторының DatabaseName қасиетінде «Student» бүркеме атауын таңдау.
- 2) Кесте атын беру – TableName қасиетінде Student кестесін таңдау.
- 3) Кестені жандандыру – Active қасиетінде true мәнін орнату (бұл келесі пункттерді орындағаннан кейін мүмкін болады).

Формаға Data Access бетбелгісі бар DataSource компонентін орнату және объектілер инспекторының DataSet қасиетінде Table1 компонентін таңдау.

Формада Data Controls бетбелгісі бар DBGrid компонентін орнату және DataSource қасиетінде DataSource1 таңдау [4].

Егер бірнеше жазу енгізілген болса Студенттер формасының түрі төмендегідей болады (9 сурет).

Ind	SFio	SData	SDsn
11	Болдырев Арслан Николаевич	23.05.1983	по пр №342342
12	Болдырев Виктор Анатольевич	02.03.2005	по пр№2344
13	Сангаджиев Алексей Владимирович	05.11.1985	по пр №456234
14	Ангаджиев	02.03.2005	по пр

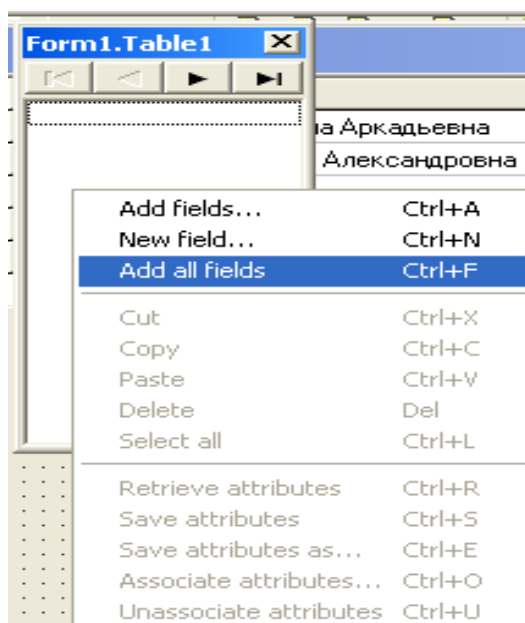
9 сурет – Студенттер формасы

Өрістер редакторы:

Кестедегі мәліметтерді көрсетуді басқару үшін арнайы өрістер редакторы қолданылады – ол Editor Field.

Editor Field шақыру үшін:

- 1) Table1 екі рет басу қажет.
- 2) Ашылған терезеге контексті мәзірді шақыру және Add All Field пунктін таңдау, егер қажет болса кестенің барлық өрістерін қосу.
- 3) Add Field жеке өрісті таңдау үшін (10 сурет).

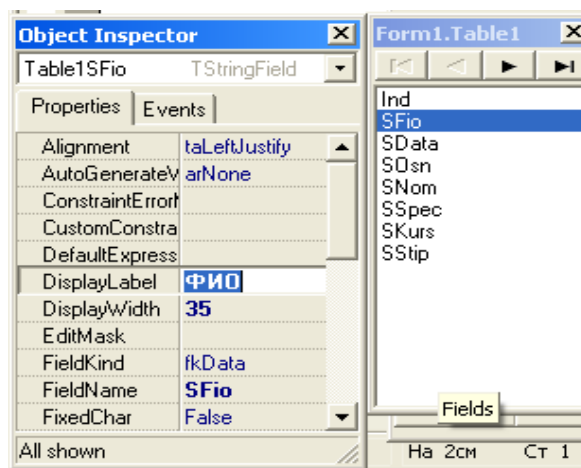


10 сурет – Add All Field пунктін таңдау

Өрістер редакторының келесі қасиеттері бар:

DisplayLabel – өріске ат береді;

DisplayWidth – өріске енгізілетін символдар санын анықтау (11 сурет)

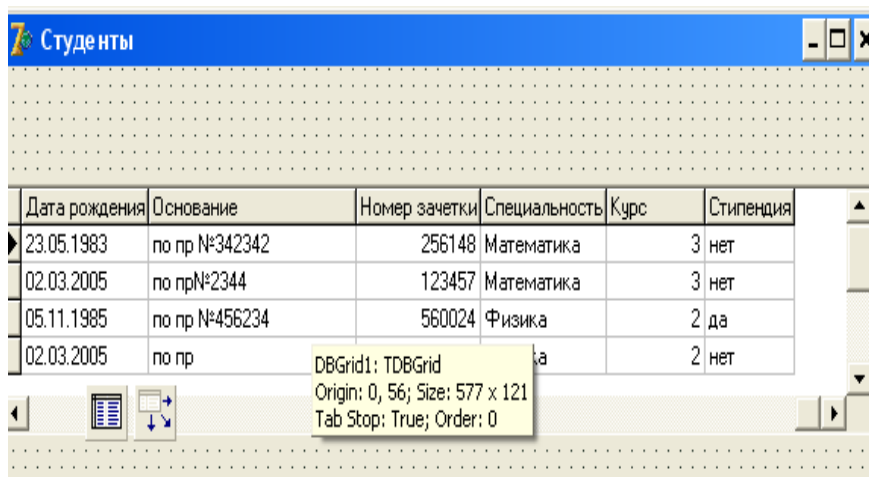


11 сурет – өрістер редакторының келесі қасиеттері

Student.db кестесі өрістері үшін қасиеттер анықтайық:

- 1) Кесте өрістері редакторы терезесінде SFio өрісін таңдау және объектілер инспекторының DisplayLabel қасиетінде SFio-ны ТАЖ-не өзгерту.
- 2) DisplayWidth қасиетін таңдау және өлшемін 35-ке өзгерту.
- 3) Сондай-ақ кестенің басқа өрістерінің қасиеттерін өзгерту.
- 4) Логикалық типті өрістер үшін DisplayValues қасиетінде True және False мәндері үшін нұсқалар жазуға болады.
- 5) SSpec өрісінде осы қасиетте «Математика; Физика» жазу (аралықтар қалдырмай, «;» белгісімен бөліп). Суретте көрсетілгендей болады.
- 6) Егер қажет болса кез келген өрісті жасыруға болады. Ол үшін оны таңдап және объектілер инспекторының Visible қасиетінде false мәнін орнату қажет [4].

Жасалған әрекеттерден кейін DBGrid1 торы келесі түрге келеді (12 сурет).



12 сурет – DBGrid1 торы

Мәліметтерді енгізу:

ДҚ кестелеріне рұқсатты ұйымдастыруға арналған компоненттер мәліметтер жиындарымен мүмкін болатын операциялар орындауға мүмкіндік

береді: жазуларды қосу немесе өшіру, олар бойынша қозғалу. Бірақ бұл жағдайда ескеретін жағдай – кез келген уақытта нақты әрекеттерді орындау үшін ағымдағы деп аталатын бір-ақ жазу қолжетімді болады. Бұл зертханалық жұмыста Table компонентінің жиі қолданылатын әдістері қарастырылған [5].

Table компонентінің рұқсат ұйымдастыру үшін қолданылатын негізгі әдістері:

Append – кесте соңына жаңа жазу қосу.

Delete – ағымдағы жолды өшіру.

Edit – редакторлау тәртіптемесіне (режим) өту. Осыдан кейін өрістер мәнін өзгертуге болады.

Insert – кестеге жаңа жол қою.

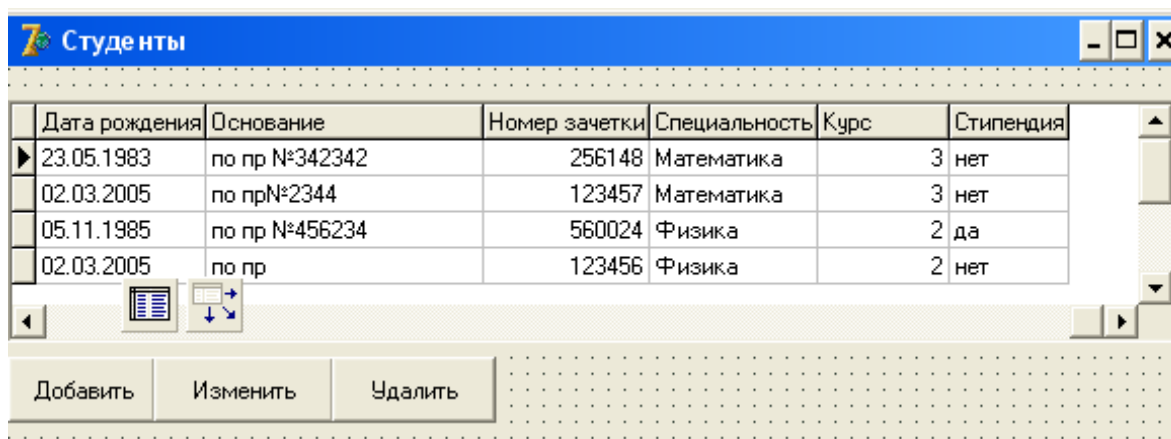
Post – барлық өзгерістерді қабылдау.

Refresh – мәліметтер туралы ақпаратты жаңарту.

UpdateRecord – ағымдағы жазуды жаңарту.

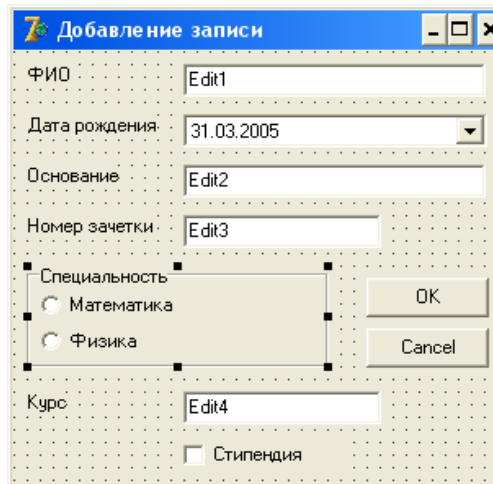
Құрылған қосымшаны ашу.

Формада Additional палитрасынан SpeedButton үш компонентін орнату. Батырмалардың біреуі жазу қосады, екіншісі жазудағы мәліметтерді өзгертеді, үшіншісі – өшіреді. Оларды сәйкесінше атаймыз (13 сурет).



13 сурет – SpeedButton үш компонентін орнату

«Қосу (Добавить)» батырмасын басу арқылы шақырылатын жаңа форма құрамыз. Формада Edit 4 компоненттері, Win32 бетбелгісі бар DateTimePicker компоненті, CheckBox компоненті және RadioGroup компоненті орналасқан (14 сурет) [5].



14 сурет – жазуларды қосу

Студенттер формасында «Қосу» батырмасының OnClick оқиғасы үшін процедура мәтіні:

```
procedure TForm1.SpeedButton1Click(Sender: TObject);
begin
    Form2.ShowModal; //«Жазу қосу» формасын ашады
end;
```

Жазу қосу формасында «OK» батырмасының OnClick оқиғасы үшін процедура мәтіні:

```
procedure TForm2.Button1Click(Sender: TObject);
begin
    Form1.Table1.Insert;
    Form1.Table1.FieldName('SFio').Text:=Edit1.Text;
    Form1.Table1.FieldName('SOsn').Text:=Edit2.Text;
    Form1.Table1.FieldName('SNom').Text:=Edit3.Text;
    Form1.Table1.FieldName('SKurs').Text:=Edit4.Text;
```

```
Form1.Table1.FieldName('SData').AsDateTime:=DateTimePicker1.Date;
if CheckBox1.Checked then
    Form1.Table1.FieldName('SStip').Text:='иә'
else
    Form1.Table1.FieldName('SStip').Text:='жоқ';
    //жалаушаға басқанда SStip өрісіне (Стипендия)
    //True мәні беріледі, кері жағдайда False мәні беріледі
case RadioGroup1.ItemIndex of
0: Form1.Table1.FieldName('SSpec').Text:='Математика';
1: Form1.Table1.FieldName('SSpec').Text:='Физика';
end;
if form1.Table1.Modified
then form1.Table1.Post;
close;
```

Түсіндірме:

Form1.Table1.Insert жолында «Студенттер» формасында орналасқан, кестеге жаңа жол қоюға мүмкіндік беретін әдіс шақырылады. Бұл әдіс шақырылмаса ары қарай кестеге жазуларды қою жұмысы мүмкін емес болады. Form1.Table1.FieldName('SFio').Text:=Edit1.Text жазуының мағынасы келесі: Edit1-де орналасқан мәтін батырманы басу кезінде «Студенттер» формасындағы кестенің ТАЖ мәтіндік өрісінің жаңа жазуына орын ауыстырылады. Процедурадағы қалған жазулар ұқсас түрде жұмыс жасайды. if form1.Table1.Modified then form1.Table1.Post жазуы кестедегі өзгерістерді сақтайды. Close – «Жазу қосу» формасын жабады.

Cancel батырмасын басу бойынша шығу іске асырылады. «Жазуды редакторлау» формасында дәл солай [5].

Студенттер формасында «Өшіру» батырмасының OnClick оқиғасы үшін процедура мәтіні:

```
procedure TForm1.SpeedButton3Click(Sender: TObject);
begin
    Table1.Delete           //кестедегі ағымдағы жазуды өшіреді
end;
```

Мәліметтерді өңдеу.

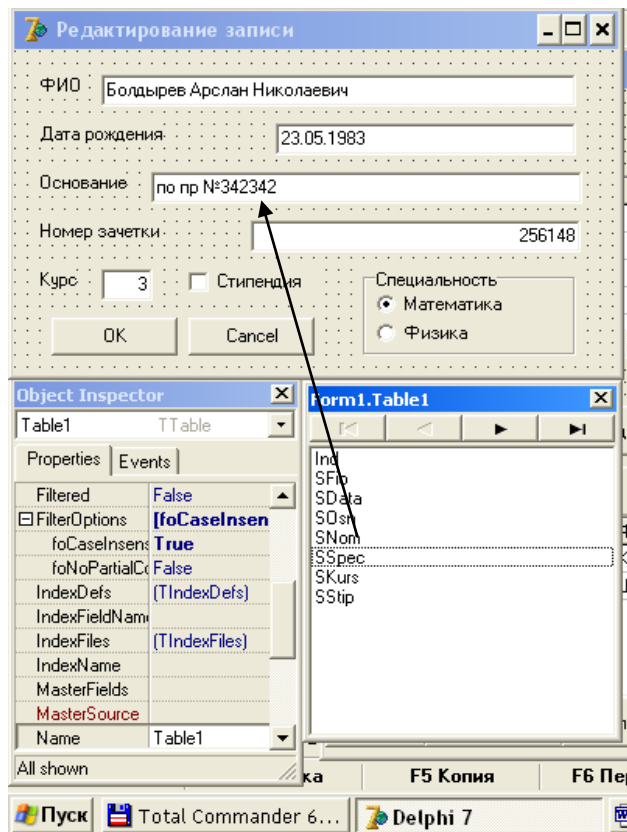
Ақпаратты бейнелейтін компоненттер екі категорияға бөлінеді – ДҚ кестелерімен байланыспаған және кестелермен байланысқан және олармен мәлімет алмасатын. Бірінші категорияға Delphi-дің қарапайым компоненттері кіреді. Екінші категория компоненттері Data Controls бетінде орналасқан. Олардың көпшілігінің қарапайым компоненттер арасында ұқсастары бар; негізгі ерекшелік, олар ДҚ сақталатын мәліметтермен жұмыс жасай алады. Бұл топқа, мәтіндік бір жолды ақпаратты енгізу үшін қолданылатын, DBEdit компоненті қатысты.

DBEdit компоненті кесте өрісінен ақпаратты көруі үшін қасиеттерде келесіні көрсету қажет:

DataSource – мәліметтер көзі; DataField – редакторлау үшін өріс.

Өрістің атын өріс редакторлау терезесінен тасу барысында, бұл компонент алдын-ала берілген қасиеттермен автоматты түрде пайда болады [6].

Жаңа форма құрамыз: жазуды редакторлау (15 сурет).



15 сурет – Жазуды редакторлау

Логикалық типті өрісті тасу жағдайында формада автоматты түрде DBCheckBox компоненті орнатылады. Ол әр кезде қолайлы емес.

Формаға Data Control бетбелгісінен DBRadioGroup компонентін орнатайық. DataSource қасиетінде Source1 көрсетеміз. DataField қасиетінде SSpec көрсетеміз.

Студенттер формасында «Өзгерту» батырмасының OnClick оқиғасы үшін процедура мәтіні:

```
begin
Form3.ShowModal //Form3-ті шақыру
end;
```

Редакциялау формасында «Сақтау» батырмасының OnClick оқиғасы үшін процедура мәтіні:

```
begin
if form1.Table1.Modified
then form1.Table1.Post; //кестедегі барлық өзгерістер
сақталады
close;
end;
```

Қолданушыда тура кестедегі жазуды редакциялауға мүмкіндігі бар. Мұны болдырмау үшін DBGrid dgEditing компонентінің қасиеті қолданылады. DBGrid1 ерекшелену қажет және объектілер инспекторының Options ► dgEditing қасиетінде false қою [6].

Бақылау сұрақтары.

1. ДҚ рұқсат беру компоненттерінің тағайындалуы.
2. ДҚ басқару компоненттерінің тағайындалуы.
3. «Өзгерту» батырмасын басқанда OnClick оқиғасы үшін процедура мәтінін түсіндіру.
4. «Қосу» батырмасының OnClick оқиғасы үшін процедура мәтінін түсіндіру.
5. «Сақтау» батырмасын басқанда OnClick оқиғасы үшін процедура мәтінін түсіндіру.
6. «Өшіру» батырмасын басқанда OnClick оқиғасы үшін процедура мәтінін түсіндіру
7. Table компонентінің рұқсатын ұйымдастыру үшін негізгі әдістер.
8. Өрістер редакторының қандай қасиеттері бар?

3 Зертханалық жұмыс №3. Деректер қорында жазбалар іріктемесі. Деректер қорында жазбаларды іріктеу және іздеу

Жұмыс мақсаты:

- 1) Деректер қорында жазуларды іріктеумен танысу.
- 2) Locate және Lookup әдістері көмегімен өрістерді іздеу.

Жалпы мағлұмат

Іріктеу.

ДҚ кестесіндегі жазулардың орналасу тәртібі белгісіз болуы мүмкін. Бірден қабылданған тәртіп бойынша жазбалар іріктелмеген болады немесе іріктеледі, мысалы, Paradox кестесі үшін кілттік өрістер бойынша, ал dBase кестесі үшін олардың кесте файлына түсу тәртібі бойынша.

Іріктелген жазулармен мәліметтер жиыны әлдеқайда ыңғайлырақ жұмыс жасайды. Іріктеу – жазуларды белгілі өріс бойынша ондағы жазулардың өсу немесе кему ретімен реттеу.

TTable мәліметтер жиынының іріктелуі автоматты түрде ағымдағы индекс бойынша орындалады. Индексті ауыстарғаннан кейін жазуларды қайта реттеу іске асырылады. Осылайша, индекс құрылған өрістер бойынша да мүмкін. Бірнеше өрістер бойынша іріктеу үшін осы өрістерді қамтитын индекс құру қажет.

Жазулардың іріктелуі орындалатын индексті келесі қасиеттер арқылы беруге болады:

- 1) IndexName – құрылу кезінде белгіленген индекс аты көрсетіледі;
- 2) IndexFieldName – лайықты индексті құратын өріс аттары көрсетіледі.

Іздеу.

Locate әдісі іздеу критерийлеріне сәйкес келетін бірінші жазуды іздейді, және егер бұндай жазу табылса, оны ағымдағы жазу қылдырады. Бұл

жағдайда нәтиже ретінде True мәні қайтады. Егер жазу табылмаса False мәні қайтады және курсор өзінің орнын ауыстырмайды.

function Locate (const KeyFields: String; const KeyValues: Variant;
Options: TLocateOptions): Boolean;

Іздеу жүргізілетін жолдар тізімі KeyFields параметрлерінде беріледі, өрістер үтірлі нүктемен бөлінеді. Variant типті KeyValues параметрі іздеу үшін өрістердің мәнін көрсетеді. Егер іздеу бір ғана өріс бойынша жүргізілсе, параметр, іздеуге берілген өріс типіне сәйкес, бір мәннен тұрады.

Options параметрі, әдетте жолдарды іздеуде қолданылатын мәнді беруге мүмкіндік береді. Бұл параметр TLocateOptions көптік типке жатады және келесі мәндердің қиыстырылуын қабылдайды:

LoCaseInsensitive – әріптер регистрі ескерілмейді;

LoPartialKey – ішінара сәйкестік рұқсат етіледі [6].

А, Б тапсырмалары

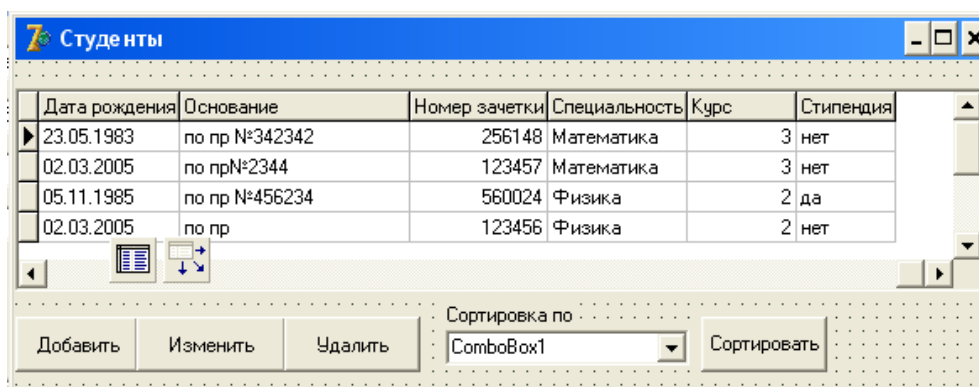
Керекті нұсқаны таңдап, Delphi бағдарламалық ортасында Locate және Lookup әдістерін қолданып жазуларды іріктеу формаларын құру (А қосымшасы). Студенттер барлық тапсырмаларды екі нұсқа бойынша орындаулары қажет (А, Б).

Әдістемелік нұсқаулықтар

Іріктеу.

1) «Студенттер» қосымшасын ашу.

2) Формаға ComboBox және Button компоненттерін қосу (16 сурет).



16 сурет – Формаға ComboBox және Button компоненттерін қосу

ComboBox компонентінің Items қасиетінде іріктеу параметрлерін жазамыз: тегі, мамандығы, курс, дүниеге келу күні, сынақ кітапшасының нөмірі.

Іріктеу шарттары екінші индекспен орнатылады. Дәлірек айтқанда, тегі бойынша іріктеу IDFio екінші кілт бойынша өтеді, өйткені оған бірінші

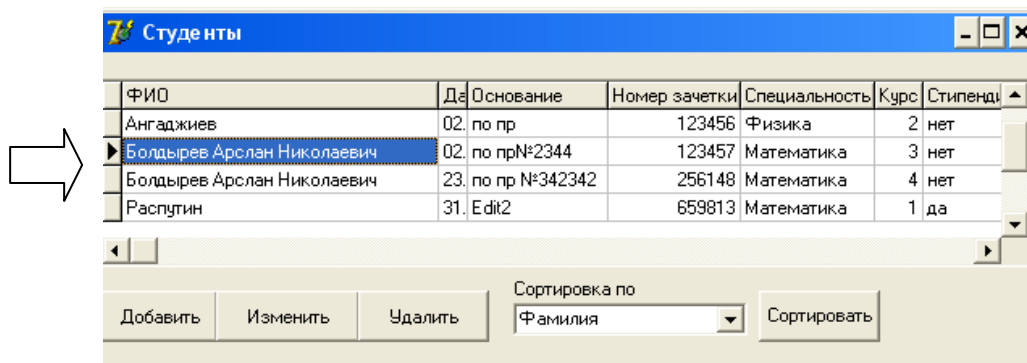
болып SFio өрісі кіреді. Іріктеу таңдалған параметрлер бойынша өтуі үшін тиісті өрістердің әр түрлі екінші кілттерге кіруі қажет.

Студенттер формасында «Іріктеу» батырмасының OnClick оқиғасы үшін процедура мәтіні:

```
begin
  Case ComboBox1.ItemIndex of
    0: Table1.IndexFieldNames:=’Sfio’; // «Тегі» жолын таңдау кезінде
      //іріктеу екінші индекс IDFio бойынша өтеді
    1: Table1.IndexFieldNames:=’Sspec’;
    2: Table1.IndexFieldNames:=’Skurs’;
    3: Table1.IndexFieldNames:=’Sdata’;
    4: Table1.IndexFieldNames:=’Snom’;
  end;
end;
```

Ескерту:

IDFio екінші индексіне келесі өрістер кіреді: Sfio, Skurs, Sspec. Тегі сәйкес келген жағдайда іріктеу курс бойынша жүреді және т.с.с. (17 сурет) [6].



ФИО	Да	Основание	Номер зачетки	Специальность	Курс	Стипендия
Ангаджиев	02.	по пр	123456	Физика	2	нет
Болдырев Арслан Николаевич	02.	по пр№2344	123457	Математика	3	нет
Болдырев Арслан Николаевич	23.	по пр №342342	256148	Математика	4	нет
Распутин	31.	Edit2	659813	Математика	1	да

17 сурет – Іріктеу

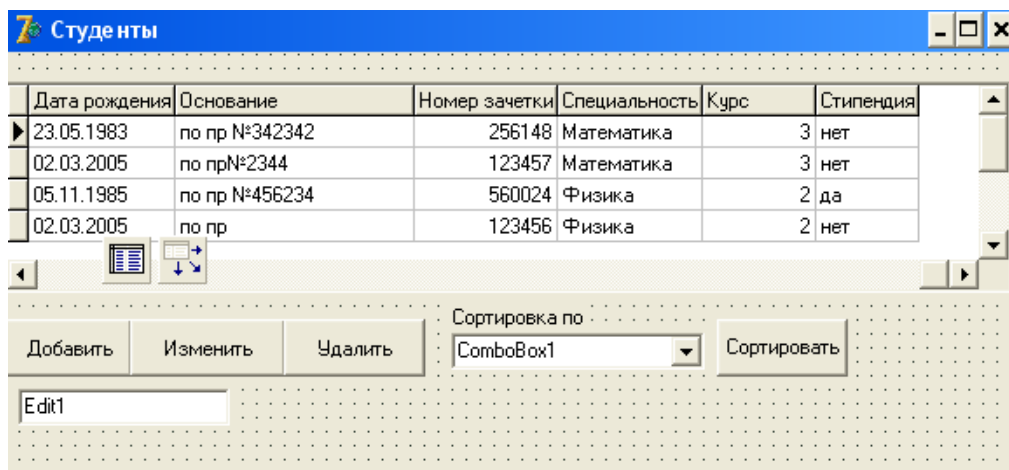
Ақпаратты іздеу.

Формаға Edit компонентін қосу.

Студенттер формасында «Edit» компонентінің OnChange оқиғасы үшін процедура мәтіні:

```
begin
  table1.Locate(’Sfio’,Edit1.Text,[loPartialKey]);
end;
```

Жазуды тегі бойынша іздеу ұйымдастырылды. Әріп үлкендіктері ескерілмейді (18 сурет).



18 сурет – Ақпаратты іздеу

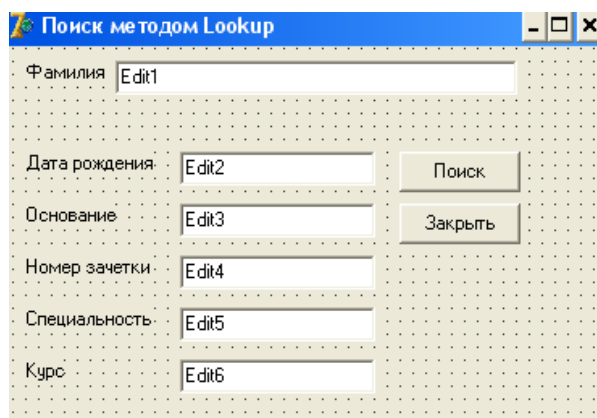
Lookup әдісі іздеу шартына сәйкес келетін жазуды табады, бірақ оны ағымдағы қылдырмайды, бірақ оның кейбір өрістерінің мәндерін қайтарады. Жазуды іздеу нәтижелеріне қарамастан ДҚ ағымдағы жазу көрсеткіші өзгермейді. Locate әдісіне қарағанда, Lookup әдісі тек қана жазуды іздеу өрісінің мәндерді іздеу критерийлеріне дәл сәйкестігі бойынша іздеуді іске асырылады.

Function Lookup (const KeyFields: String; const KeyValues: Variant;
const ResultFields: String): Variant;

Сәтті іздеу нәтижесінде мәндерін алу қажет өрістер ResultFields параметрінде келтіріледі. Нәтиже типі – Variant немесе нұсқаулық массив.

Бас формаға жаңа «Іздеу» батырмасын қосу.

19-суретте көрсетілгендей жаңа форма ашу және компоненттерді енгізу. Бұл форма бас формадағы іздеу батырмасын басу арқылы шақырылады [6].



19 сурет – Lookup әдісімен іздеу

Формадағы «Lookup әдісімен іздеу» батырмасын басқаннан кейін, іздеу Edit1 компонентіне енгізілген тегі бойынша орындалады.

```
procedure TForm5.Button1Click(Sender: TObject);
var LookupResult: Variant;
begin
```

```

LookupResult:=Form1.Table1.Lookup('SFio', Edit1.Text, 'SData; SOsn;
SNom; SSpec; SKurs'); // дүниеге келу күні '
//Негіз', 'Сынақ кітапшасы нөмірі', 'Мамандық', 'Курс' өрістері
ізделеді
        if VarIsArray (LookupResult) then
            begin
                Edit2.Text:=LookupResult[0]; //ізделетін өрістерде
                Edit3.Text:=LookupResult[1]; // тиісті компоненттерге
                Edit4.Text:=LookupResult[2]; //мәндер
                Edit5.Text:=LookupResult[3]; //жазады
                if Edit5.Text='False' then
                    Edit5.Text:='Физика' //логикалық типтегі өрістерді
іздеу
                    else Edit5.Text:='Математика';
                Edit6.Text:=LookupResult[4];
            end;
        end;

```

Бақылау сұрақтары.

1. «Lookup әдісімен іздеу» батырмасының бағдарлама мәтіні.
2. «Edit» компонентінің OnChange оқиғасы үшін процедура мәтіні.
3. Ақпаратты іздеу қалай іске асырылады?
4. Ақпаратты іріктеу деген не?
5. Іздеу әдістері.
6. Индекстер деген не?
7. Locate әдісі.
8. Lookup әдісі.

4 Зертханалық жұмыс № 4. Жазбаларды фильтрлеу

Жұмыс мақсаты:

- 1) Жазуларды фильтрлеумен танысу.
- 2) Ақпаратты іздеу мүмкіндігі бар форманы құру.

Жалпы мағлұмат

Фильтрация – мәліметтер жиынының арасынан таңдау (тек қана нақты шарттарға сәйкес келетін жазулар арасынан ғана) [7].

Мысалы, «Тегі» өрісі «Ахметов» мәнінен ғана тұратын жазуларды ғана шығаруды көрсетуге болады. Фильтрлерді мәліметтер жиынына қолдану логикалық типтің Filtered қасиеттерімен анықталады. True мәні фильтр ретінде Filter қасиетінде көрсетілген жазуды қолдануды анықтайды:

Өріс [Салыстыру операторы] 'Мән'.

Мысалы, егер «Тегі» өрісі «Алмасов» мәніне тең болатын барлық жазуларды көрсетсе, келесіні көрсету қажет:

```
Table1.Filter:='Тегі=' Алмасов'';
```

А, Б тапсырмалары

Керекті нұсқаны таңдап, Delphi ортасында ақпаратты іздеу және жазуларды фильтрлеу мүмкіндігі бар формаларын құру (А қосымшасы). Студенттер барлық тапсырмаларды екі нұсқа бойынша орындаулары қажет (А, Б).

Әдістемелік нұсқаулықтар

Жазуларды фильтрлеу

«Студенттер» қосымшасын ашу.

Формаға TEdit компонентін қосу.

OnChange оқиғасы үшін процедура мәтіні:

```
begin
```

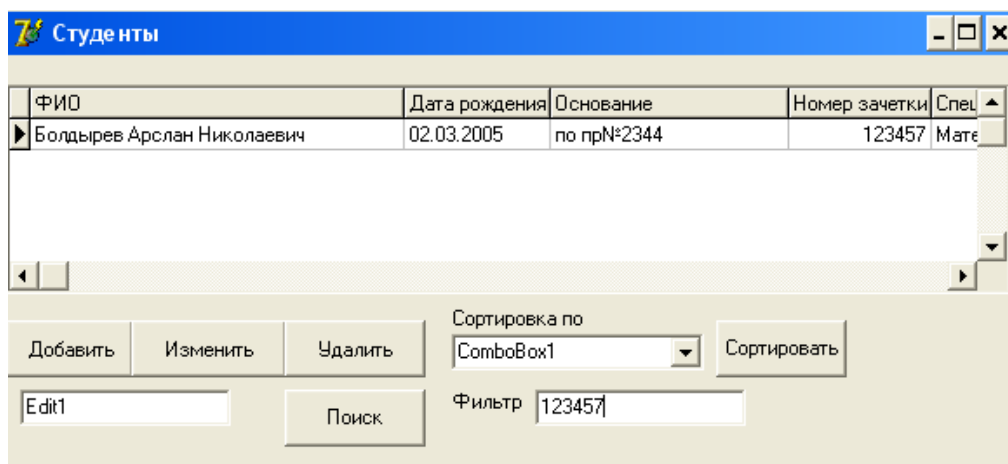
```
    Table1.Filtered:=true;           //фильтрлеуді қосу
```

```
    Table1.Filter:='SNom = '+Edit2.Text;
```

```
//фильтрлеу критерийлерін береді
```

```
end;
```

Фильтрлеудің бұл әдісі тек қана сандық өрістерге жарамды (20 сурет).



20 сурет – Сандық өрістерге жарамды фильтрлеу

Фильтрлерді қолданғанда келесі қасиеттерді көрсетуге болады:

- 1) foCaseInsensitive – Әріп регистріне сезімтал еместік;
- 2) foNoPartialCompare – нақты сәйкестікті іздеу [7].

Мәтіндік өрістерді фильтрлеу үшін, мысалы «Тегі» өрісі бойынша, процедура мәтінін өзгерту қажет.

```
procedure TForm1.Edit2Change(Sender: TObject);
```

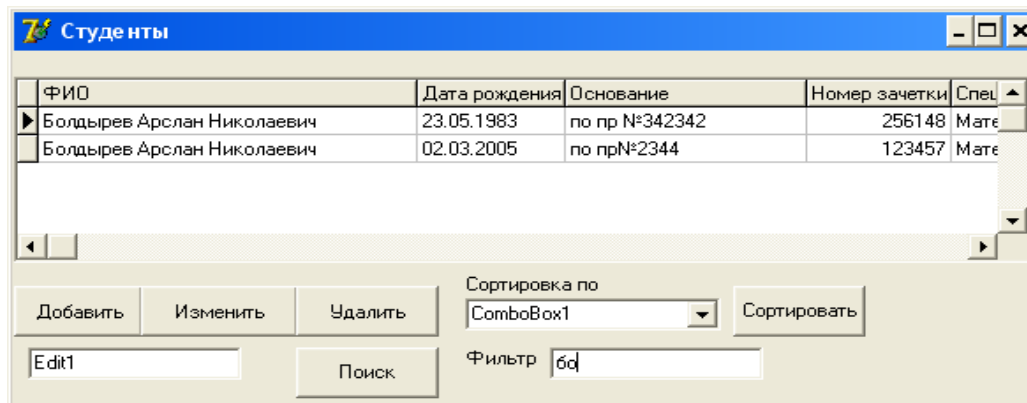
begin

```
Table1.Filtered:=true;
```

```
Table1.Filter:='SFio='+#39+Edit2.Text+'*'+#39;
```

end;

Бұл жағдайда фильтрлеу мәтіндік өріс бойынша іске асырылады. «#39» белгісі апострофа белгісін білдіреді (өйткені фильтрді қолдануда тегін енгізу апострофтармен ісе асады). «*» белгісі кез келген белгілерді білдіреді, дәлірек айтқанда тек қана бір әріпті ғана енгізген кезде экранда берілген әріпке басталатын барлық тектер шығады (21 сурет) [7].



21 сурет – Бір әріпті енгізіп фильтрлеу

Бақылау сұрақтары.

1. «#39» белгісі нені білдіреді?
2. Мәтіндік өрістерді фильтрлеу үшін процедураның қандай мәтіні қажет?
3. OnChange оқиғасы үшін процедура мәтіні.
4. foCaseInsensitive қасиетінің міндеті.
5. foNoPartialCompare қасиетінің міндеті.
6. Мәліметтер жиынына фильтр қолдану қандай қасиеттермен анықталады?
7. Мәтіндік өрістерді фильтрлеу үшін қандай командалар қажет?
8. Фильтрлеу критерийлері қалай беріледі?

5 Зертханалық жұмыс № 5. Кестелер арасында байланыс орнату

Жұмыс мақсаты:

- 1) Кестелер арасындағы байланысты оқып білу.
- 2) Кестелер арасында байланыс орнату.

Жалпы мағлұмат

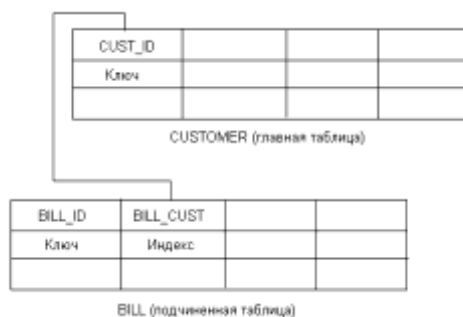
Мәліметтерді ұйымдастыру және кестелер арасындағы байланыс.

Кілттер мен индекстерді қарастырғанда талай рет ДҚ аясындағы әр түрлі кестелер арасындағы байланысты ұйымдастыру туралы айтылған. ДҚ тек қана бір ғана, қызметкерлердің аттары және туған күнінен (күні, айы, жылы) құралған кестеден тұра алады, бірақ тәжірибеде ДҚБЖ мұндай мақсатта қолданылмайды: мұндай ақпарат қарапайым электронды кестеде немесе мәтіндік файлда сақтала алады.

Кестелер арасында қарым-қатынасты ұйымдастыру байланыстыру немесе кестелерді қосу деп аталады және ДҚ құрудағы негіз болып табылады. Байланыстар кестелерді құрғанда (кілттер мен индекстерді тағайындау), сол сияқты құру процессі барысында және қосымша жұмысы кезінде беріледі. Осы мақсатта ДҚБЖ беретін әр түрлі құралдар қолданылады.

Кестелерді байланыстыру үшін байланыс өрістері деп аталатын өрістер қолданылады. Мұндай өрістер индекстелген болулары қажет. Кестелерді байланыстырғанда олардың бірі бас кесте (Master), ал басқасы – бағынышты (Slave, Detail) деп аталады. Бірақ, бағынышты кестедегі байланысқа жауапты индекс «сыртқы кілт» деп аталады. Бас кестедегі байланысқа қатысатын өріс «бас индекс» болуы қажет (егер қолданыстағы ДҚБЖ-де ондай түсінік болса). Байланысқа қатысатын өрістердің типтері сәйкес келулері қажет, мысалы, екеуі де бүтін санды болулары қажет [8].

Мысалы, 2 кестеден тұратын ДҚ қарастырайық. Олардың бірінде клиенттер тізімі сақталған (оны CUSTOMER деп атайық), екіншісінде – осы клиенттерге берілетін жөнелтпе құжат (BILL). Бұл жағдайда бірінші кестенің кілті ретінде клиент идентификаторы болуы қажет (CUST_ID типті автоинкрементті өріс деп алайық), ал екіншісінде, мүмкіндігінше, жөнелтпе құжат қай клиентке берілгені жөнінде ақпараты бар, идентифицирленген өріс болуы қажет (BILL_CUST өрісі). Мұнда екінші кестенің кілттік өрісі ретінде жөнелтпе құжаттың нөмірі қабылданады (BILL_ID өрісі). Пайда болған байланыста CUST_ID және BILL_CUST байланысқан өрістер болып келеді, бұл жағдайда соңғысы BILL кестесі үшін сыртқы кілт болып табылады (22 сурет).



22 сурет – Кестелер арасындағы байланыс

Өзара 2 немесе бірден бірнеше кестелерді байланыстыруға болады. Бұл жағдайда кестелердің бірі бас кесте, ал басқасы – бағынышты болады. Мұндай байланысты әдетте «бас-бағынышты» деп атайды. Берілген шарттарға қарай, бұл байланыс келесі 4 түрдің біріндей бола алады:

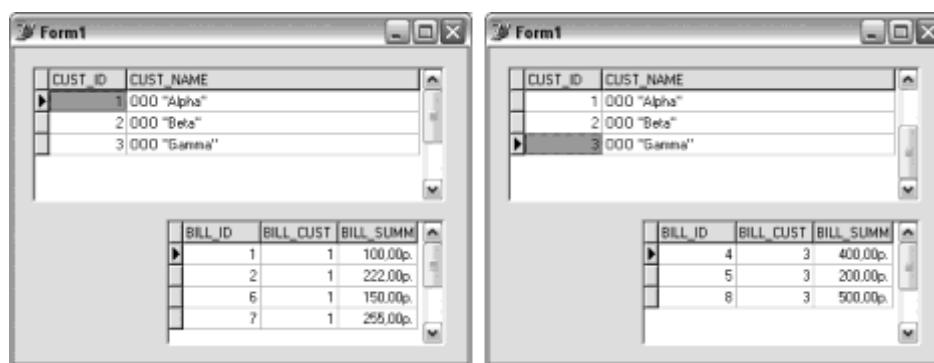
1) «Бір бірге», мұнда бас кестедегі әр бір жазуға бағынышты кестедегі 1 жазу сәйкес келеді;

2) «Бір көпке», мұнда бас кестедегі әр бір жазуға бағынышты кестедегі 0 немесе көп жазулар сәйкес келеді;

3) «Көп бірге», мұнда бас кестедегі бірнеше жазуға бағынышты кестедегі 1 жазу сәйкес келеді;

4) Мұнда бас кестедегі жазулардың кез келген санына бағынышты кестедегі сондай белгісіз жазу саны сәйкес келеді.

2 – нұсқа жиірек кездеседі. Оған жоғарыда қарастырылған мысал да сәйкес келеді: клиент туралы әрбір жазуға жөнелтпе құжат кестесіндегі кез келген жазу саны сәйкес келе алады, сонымен қатар егер клиент әлі еш нәрсе сатып алмаса олар болмайды. Кестелер арасындағы байланыс орнатылғаннан кейін, бас кестеде белгілі бір жазудың таңдалуы автоматты түрде бағынышты кестедегі таңдауға алып келеді (23 сурет).



23 сурет – «Бір көпке байланысы»

Кестелерді IDE құралдарымен байланыстыру технологиялары кейінірек қарастырылады. Бірақ алдын-ала айта кетсек, бұл жағдайда BILL кестесі үшін Master-Кесте CUSTOMER таңдалған және BILL_CUST ' CUST_ID типті өрістерді біріктіру арқылы байланыс орнатылған [8].

А, Б тапсырмалары

Керекті нұсқаны таңдап, Delphi ортасында кестелер арасында байланыс орнату (А қосымшасы). Студенттер барлық тапсырмаларды екі нұсқа бойынша орындаулары қажет (А, Б).

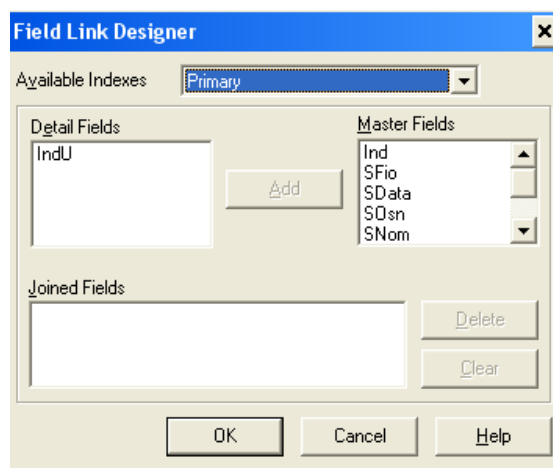
Әдістемелік нұсқаулықтар

Кеселер арасындағы байланысты көрсету үшін тағы бір кесте құру қажет. Студенттердің үлгерімі туралы кесте құру. Оған келесі өрістер кіреді: оқу жылы, сессия (қыс немесе жаз), пән, оқытушының ТАЖ, пән бойынша аттестаттау күні, тапсыру күні, бағасы.

Бұл кесте Студенттер кестесі үшін еншілес болады. Үлгерім кестесіне қосымша сандық өріс енгізу қажет және оны екінші кілтпен анықтау қажет.

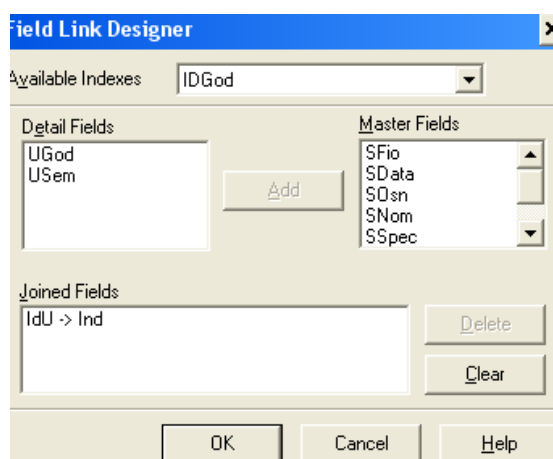
Үлгерім кестесін Студенттер формасында орналастыру.

Table2 Master Source компонентінің қасиеттерінде Data Source1 жазу. Бұл екінші кесте біріншісі үшін еншілес болатынын білдіреді. Master Fields қасиетіне екі рет басумен Field Link Designer терезесін шақырамыз (Байланыс өрісінің дизайнері) (24 сурет).



24 сурет – Байланыс өрісінің дизайнері

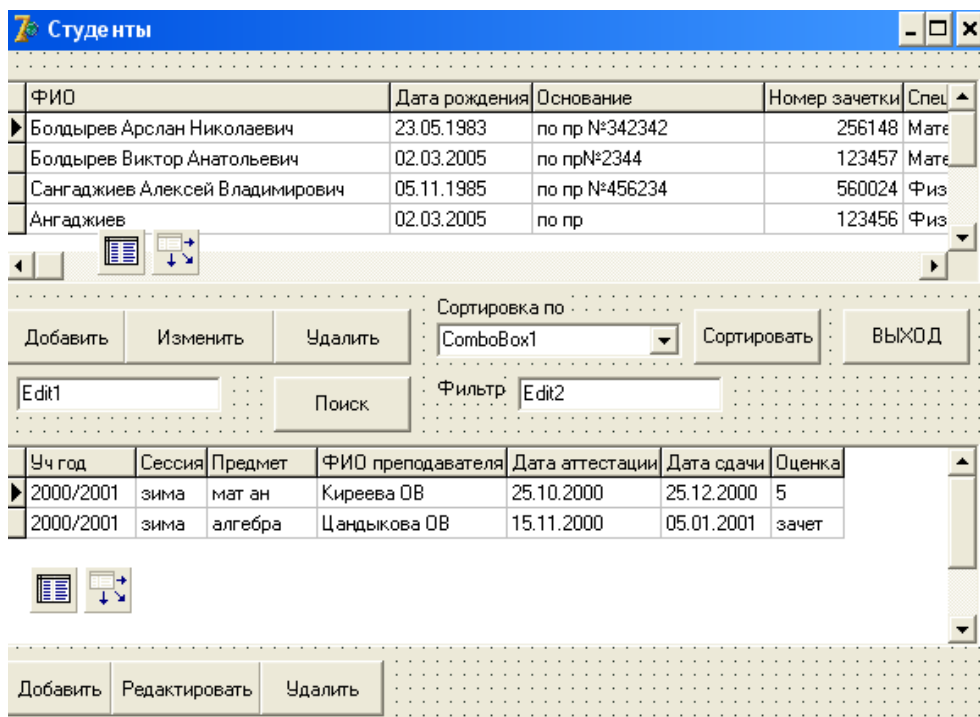
Available Index тізімінде (Қолжетімді индекстер) Field Link Designer терезесінен IDGod индексін таңдау (бұл екінші кестенің екінші индексі). Detail Field сол жақ тізімінде IdU ерекшелеу, ал оң жақтағы Master Field (Негіз) тізімінен Ind ерекшелеу. Add (Қосу) батырмасына басу және терезені жабу (25 сурет) [8].



25 сурет – Field Link Designer терезесі

Осылайша, кестелер арасында байланыс орнатылды. Ол бір көпке байланысы деп аталады.

Еншілес кестеге, Студенттер кестесіндегі сияқты, мәліметтерді енгізу және редакциялау батырмаларын қосу (26 сурет).



26 сурет – Еншілес кестеге батырмаларды қосу

Бақылау сұрақтары.

1. Master Source компоненті.
2. Кестелер арасындағы байланыс түрлері туралы айтып беру.
3. «Бір бірге» байланысы туралы айтып беру.
4. «Бір көпке» байланысының қызметі.
5. «Көп бірге» байланысы.
6. «Көп көпке» байланысы.
7. Available Index тізім параметрлері.
8. Байланыс өрісі дизайнерінің қызметі.

6 Зертханалық жұмыс № 6. Есептемелер құру.

Жұмыс мақсаты: есептемелерді құру мүмкіншіліктерімен танысу.

Жалпы мағлұмат

Есептемелерді құру.

Есептеме – ол ДҚ жазбаларынан тұратын басылған құжат. Delphi-де есептемелер құру үшін, компоненттердің жалпы жиынын қамтитын, QuickReport есептемелер генераторы қызмет етеді.

Компоненттер есептемелерді құруға арналған, компоненттер палитрасының Qreport бетбелгісінде орналасқан [9].

Есептеменің басты элементі QuickRep есептеме-компонент болып табылады. Ол басқа компоненттер орнатылатын негіз болып келеді. QuickRep компоненті әдетте, есептеме құруға арналған, бөлек формада орнатылады.

QuickRep компонентінің қасиеттері:

1) Bands – мұнда QuickRep-те орналастырылатын компоненттер көрсетіледі.

2) DataSet – мұнда есептеме мәліметтер алатын мәліметтер жйыны көрсетіледі.

3) Frame – мұнда жиектеме параметрлері көрсетіледі.

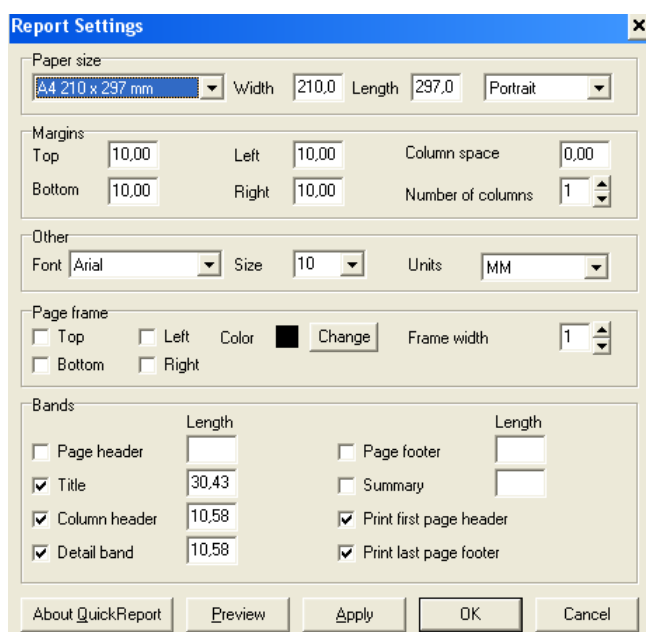
4) Options – мұнда үш параметр қол жетімді. Егер FirstPageHeader мәні true-ға тең болса, онда тақырып есептеменің тек қана бірінші бетінде басылады. Егер LastPageFooter мәні true-ға тең болса, онда төменгі колонтитул есептеменің тек қана соңғы бетінде ғана басылады. Егер Compression қасиетін true-ға орнатса, онда есептеме сығылған түрде сақталады.

5) ReportTitle – мұнда басылатын құжаттың тақырыбы орналасқан.

6) SnapToGrid – белгіленген тор бойынша компоненттерді реттеп түзету қажет пе?

7) Zoom – мәліметтерді көрсету масштабы.

Есептеме параметрлерінің икемдеуді Report Settings терезесінің көмегімен орындауға болады. Ол QuickRep компонентін тышқанмен екі рет басу арқылы шақырылады. Осы терезені қолдану ұнамдырақ, өйткені мұнда әрдайым болашақ нәтижені көруге болады (27 сурет) [9].

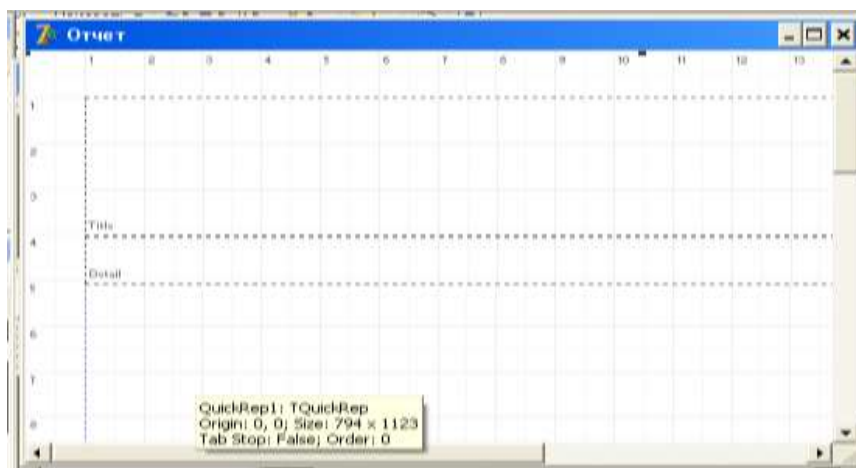


27 сурет – Report Settings терезесі

Әдістемелік нұсқаулықтар

1. «Студенттер» қосымшасын ашу.

2. Бас формаға «Есептеме құру» батырмасын қосу.
3. «Есептеме құру» батырмасына басу арқылы шақырылатын «Есептеме» жаңа формасын құру.
4. Формаға QReport бетбелгісінен QuickRep компонентін орнату. Бұл компонентті ерекшелеу және объектілі инспекторда Bands қасиетінің HasTitle және HasDetail параметрлерін қосу (28 сурет).



28 сурет – Формаға QReport бетбелгісінен QuickRep компонентін орнату

5. Есептеменің керекті ақпаратын көрсететін компоненттерді QuickRep1 секцияларында орналастырайық.

Компоненттер палитрасының *QReport бетбелгісінде* төмендегі компоненттер қолжетімді. Оларды осы бөлімдерде орналастыруға болады:

- 1) QRLabel – жазба. Бұл компонент стандартты TLabel компонентіне ұқсас және жай ғана керекті мәліметтерді көрсетеді.

- 2) QRDBText – мәліметтер. Бұл компонент те TLabel компонентіне ұқсас, бірақ ол деректер қорынан кез келген өрістің тағайындалуын көрсету үшін арналған.

- 3) QRSysData – жүйелік ақпарат. Бұл компонент те TLabel компонентіне ұқсас, бірақ жүйелік ақпаратты көрсету қабілеті бар (күні, уақыт, бет нөмірі, кестедегі жол нөмірі, беттердің жалпы саны және т.б.).

- 4) QRImage – сурет. TImage-ға ұқсас компонент [9].

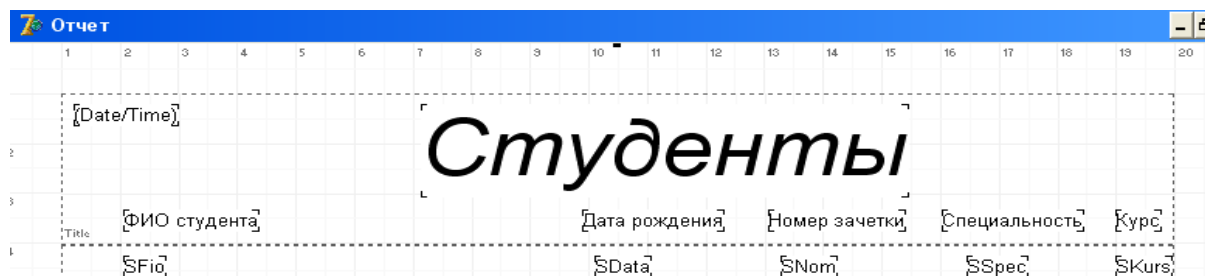
6. Title тақырып облысын үлкейту. Жоғарғы бұрышқа QRSysData-ның бір компонентін орнату. Оны ерекшелеу және Data қасиетінде qrsDateTime мәнін таңдау. Енді бұл компонент оң жақ жоғарғы бұрышта құжатты басып шығару уақытын көрсетеді.

7. Tittle облысының ортасында QRLabel компонентін орнату, Font қасиетінде әріп көлемін үлкейту және Caption қасиетінде «Студенттер» мәтінін жазу.

8. Tittle облысында QRLabel компоненттерін орнату және оларға келесі тақырып беру: ТАЖ, Дүниеге келу күні, Сынақ кітапшасының нөмірі, Мамандығы, Курсы.

9. Detail облысына өту. Тақырыпша астында бес QRDBText компоненттерін қою. DataSet компоненттерінің қасиеттерінде QRDBText мәліметтер жиынын орнату – Form1.Table1, ал DataField қасиетінде QRDBText1 үшін SFio көрсету.

10. QRDBText-тің барлық басқа компоненттерінде сәйкес жолдар атауларын орнату (29 сурет).



29 сурет – QReport компоненттерін орнату

11. «Баспаға шығару» батырмасын басып бас модульге ауысу және келесі кодты жазу.

```
procedure TForm1.SpeedButton5Click(Sender: TObject);
```

```
begin
```

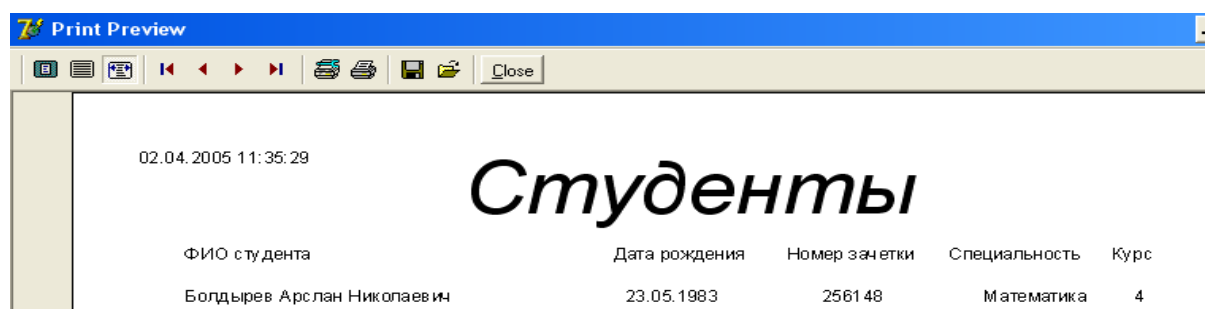
```
Form4.QuickRep1.Preview; // Preview әдісі шақырылады
```

```
// QuickRep компоненті. Бұл әдіс құрылған құжаттың алдын-ала көру // терезесін көрсетеді.
```

```
end;
```

12. Бағдарламаны қосу, кез-келген жолды ерекшелеу және баспаға жіберу батырмасын басу.

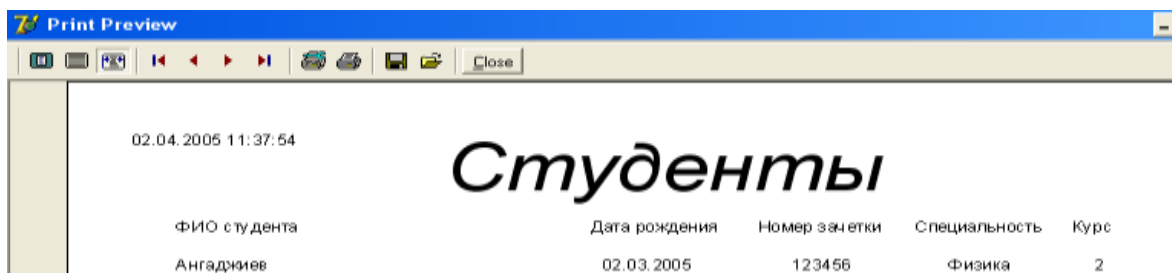
Алдын-ала көру терезесі ашылады (30 сурет).



30 сурет – Алдын-ала көру терезесі ашылады

13. QuickRep1 компонентін ерекшелеу және DataSet қасиетінде Form1.Table1 кестесін көрсету. Егер бұл әрекетті орындаса QuickRep1 компоненті автоматты түрде бұл кестеден барлық жазуларды сұрыптайды және оларды DetailBand1 блогында тұрған компоненттерде қолданады [9].

14. Бұдан кейін есептемеде кестенің барлық жазулары пайда болады (31 сурет).



31 сурет – Есептемеде кестенің барлық жазуларын шығару

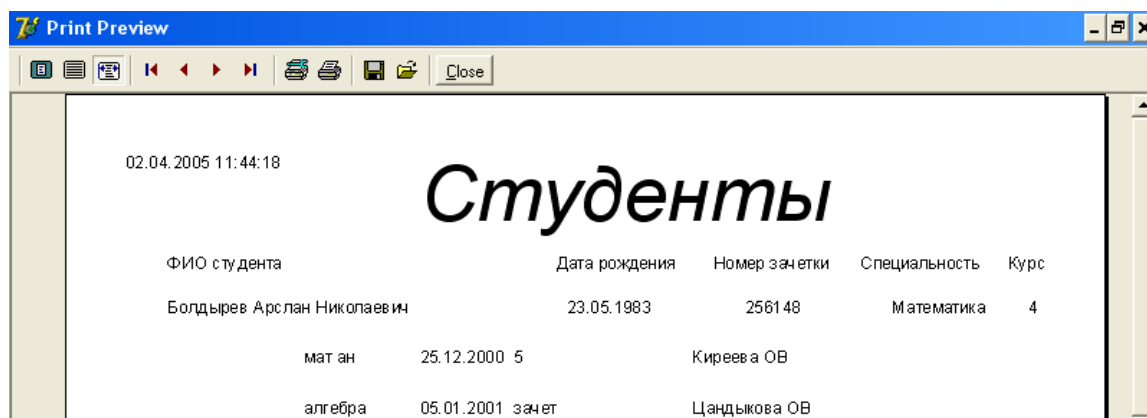
15. Есептеме формасына QReport бетбелгісінен QRSubDetail компонентін орнату. Бұл компонент бағыңқы кестелерге қатысты мәліметтерді іріктеуге арналған.

16. Келесі қасиеттерді орнату: DataSet – Form1.Table2, Studenti.db негізгі кестесіне бағыңқы болып келетін Uspevaemost.db кестесімен блокты байланыстыру үшін.

17. Master қасиетінде негізгі мәліметтермен бас компонентті көрсету қажет. Бұл қасиетте QuickRep1 таңдау.

18. QRSubDetail компонентінде QRDBText компонентын орналастыру. Қасиеттерінде олар бағыңқы кестенің қай өрістеріне жүгінетіндігін көрсету [9].

19. Есептеменің келесі түрі шығады (32 сурет).



32 сурет – Есептеме

Бақылау сұрақтары.

1. Есептеме деген не?
2. Есептемелер қалай құрылады?
3. «Басып шығару» батырмасының коды.
4. QReport бетбелгі компоненттері.
5. QuickRep компоненті туралы айтып беру.
6. QRSubDetail компонентінің функциялары.
7. QRSysData компонентінің тағайындалуы.
8. QRLabel компонентінің тағайындалуы.

А қосымшасы

Деректер қорлары тақырыптарының тізімі (ДҚ)

А нұсқасы

1. «Тауар өткізу» ДҚ құру.
2. «Қойма» ДҚ және қосымшасын құру.
3. «Жылдық өнім есебі» ДҚ және қосымшасын құру.
4. «Тұрмыстық қызмет есебі» ДҚ және қосымшасын құру.
5. «Өнім өткізу» ДҚ және қосымшасын құру.
6. «Жылдық тасымал есебі» ДҚ және қосымшасын құру.
7. «Салық төлемдері» ДҚ құру.
8. «Тауарлар сапасы сараптамасы туралы акт құру» ДҚ құру.
9. «Дүкендегі тауар қозғалысының есебі» ДҚ құру.
10. «Қоймадағы құрал-саймандарды есептеу және бақылау» ДҚ құру.
11. «Айлық төлемді есептеу» ДҚ құру.
12. «Автобуска (поезд, самолет) билеттер сату» ДҚ құру.
13. «Кітапханадағы оқырмандарға қызмет көрсету» ДҚ құру.
14. «Кадр бөлімі» ДҚ және қосымшасын құру.
15. «Сауда үйі» ДҚ құру.
16. «Халықтың тұрмыс қажетін өтеу қызметі» ДҚ құру.
17. «Кеден қызметі» ДҚ құру.
18. «Өнеркәсіп мекемесі» ДҚ құру.
19. «Қаржылық орган» ДҚ құру.
20. «Көлік қызметі» ДҚ құру.

Б нұсқасы

1. «Мемлекеттік қызмет» ДҚ құру.
2. «Сақтандыру компаниясы» ДҚ құру.
3. «Электр жабдықтарды сертификаттау» ДҚ құру.
4. «Материалдық-техникалық жабдықтау» ДҚ құру.
5. «Қонақ үй кешенін сертификаттау» ДҚ құру.
6. «Аудиторлық фирма» ДҚ құру.
7. «Құрылыс компаниясы» ДҚ құру.
8. «Бөлшек сауда кәсіпорны» ДҚ құру.
9. «Қоғамдық тамақтандыру кәсіпорны» ДҚ құру.
10. Үлестірілген деректер қорлары ДҚ құру.
11. Нәтижелерді статистикалық өңдеу үшін ДҚ қолдану.
12. «Өнеркәсіптік класс өнімін өндіру» ДҚ құру.
13. Қызмет көрсету аясы ДҚБЖ қолданып сараптау жүйесін құру.
14. ЖТК (КСК) үшін ДҚ құру.
15. «Мұражай» ДҚ құру.
16. «Сабақтар кестесі» ДҚ құру.
17. «Көмір шығару және өткізу есебі» ДҚ құру.

18. Медициналық мекеме үшін ДҚ құру.
19. Қонақ үй кешені үшін ДҚ құру.
20. Бағдарламалық жабдық таңдауды анықтау үшін ДҚ құру.

Әдебиеттер тізімі

1. Дейт К. Дж. Введение в системы баз данных. — М.: Издательский дом «Вильямс», 2008.
2. Коннолли Т., Бегг К. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. — М.: Издательский дом «Вильямс», 2003.
3. Кузнецов С. Д. Основы баз данных. — 1-е изд. — М.: «Интернет-университет информационных технологий - ИНТУИТ.ру», 2005.
4. Харрингтон Дж. Разработка баз данных. — М.: ДМК Пресс, 2005.
5. Хомоненко А. Д., Цыганков В. М., Мальцев М. Г. - Базы данных. Учебник для вузов. — М.: Корона-Принт, 2004.
6. Кандзюба С. П., Громов В. Н. Delphi 6/7. Базы данных и приложения. — СПб: ООО «ДиаСофтЮП», 2002.
7. Григорьев Ю. А., Ревунов Г. И. Банки данных. — М.: Изд. МГТУ им. Н. Э. Баумана, 2002.
8. Харрингтон Д. Проектирование реляционных баз данных просто и доступно. — М.: Изд. «Лори», 2004.

Балнур Каирбайкызы Каирбаева
Наталья Александровна Водолазкина

МӘЛІМЕТТЕР ҚОРЛАРЫНЫҢ ЖҮЙЕСІ

5B070300 – Ақпараттық жүйелер
мамандығының студенттері үшін
зертханалық жұмыстарды орындауға арналған әдістемелік нұсқаулықтар

Редактор Б.С. Қасымжанова
Стандарттау бойынша маман Н.К. Молдабекова

____.____.____. басуға қол қойылды
Таралымы 30 дана.
Көлемі 2,37 оқу.-бас.ә.

Пішімі 60x84 1/16
№1 типографиялық қағаз
Тапсырыс __ Бағасы 1188.

«Алматы энергетика және байланыс университеті»
коммерциялық емес акционерлік қоғамының
көшірмелі-көбейткіш бюросы
050013, Алматы, Байтұрсынұлы көшесі, 126