

# Мәліметтер қорымен ЖҰМЫС.

МҚ негізгі ұғымдары

МҚ жұмыс жасауға арналған компоненттер

- C++ Builder бағдарламалау ортасында бірнеше мәліметтер қорына қатынау механизмдері бар: InterBase, Informix, IB Server, BDE, ODBC және де ADO технологиясы. Солардың ішінен біразына тоқталып өтейік.
- C++ Builder көмегімен құрылған және файл-серверлік мәліметтер қорымен жұмыс істеуге арналған бағдарламалар арнайы BDE (Borland Database Engine – Borland корпорациясының мәліметтер қорының машинасы) бағдарламалар кітапханасына негізделеді. BDE әр түрлі форматтағы төмен деңгейлі DLL кітапханасының жиынтығын ұсынады. BDE C++ Builder -ді орнатып жатқан кезде автоматты түрде орнатылады және Windows-тің 32-разрядты нұсқамасында тіркеледі. BDE МҚБЖ-ның көптеген кестелерімен жұмыс атқара алады, мәселен, файл-серверлік (dBase, Paradox, FoxPro, Clipper), сондай-ақ клиент-серверлік (Inter Base, Microsoft SQL Server, Oracle және т.б). BDE механизмінде SQL тілінің өзінің интерпретаторы бар, яғни соның арқасында тек мәліметтер қорының серверіне ғана емес, сонымен бірге, файл-сервер кестелеріне де сұраныс құра алады.
- Компьютерде BDE орнатылмаса және тіркелмесе C++ Builder -де жасалған және BDE механизмін қолданатын бірде бір мәліметтер қоры жұмысын жалғастыра алмайды. Бұл жағдай бағдарламаны таратуды қиындата түседі, себебі бағдарламамен бірге BDE кітапханасының жиынтығын да бірге жіберу керек болады.

## БЕРІЛГЕНДЕР ҚОРЫНА ҚОЛ ЖЕТКІЗУГЕ МҮМКІНДІК БЕРЕТІН КОМПОНЕНТТЕР

Берілгендер қорымен байланыстырушы компоненттер Data Access вкладкисында орналасқан. Мұндай компоненттер өңдеушіге берілгендер қорымен ыңғайлы жұмыс істеуге мүмкіндік береді.



Яғни бағдарламаға 8 түрлі берілгендер қорына қол жеткізетін интерфейстік элементтерді (dataset) – кестелер, сұраныс, қойма процедуралары, сессия және тағы басқа қосуға болады, мұның негізі- 32 разрядтық механизм BDE болып табылады.



## **TDataSource**



Форма бетіндегі көрінетін басқару компоненттері мен берілгендер жиынына қол жеткізуге мүмкіндік беретін интерфейс. Берілгендер қорындағы мәліметтерді түзету, навигациялау және көрсету үшін берілгендер көзімен байланыстырды. Басқару компоненттерінің көмегімен мәліметтер ретін басқару мүмкіндігін берілгендер көзімен байланыстыру арқылы алуға болады.

TdataSource компоненттері master-detail принципі бойынша өзара жұп кестелерді ұйымдастыруға және мәліметтермен алмасу синхронизациясымен жұмыс істеуге болады. Мәліметтерді енгізуге мүмкіндік бермеу үшін AutoEdit қасиетіне false мәнін беру керек. DataSet қасиеті берілгендер көзімен байланысын анықтайды. Active қасиетін қосу арқылы кестенің мәліметтерін форма бетіне шығаруға болады. TableName - кесте атын анықтайды.

## **Table**



BDE механизмі TdataSource және TDBGrid компонентінің арасында байланыс орнатушы компонент. Жобалау кезінде оның көмегімен локальды берілгендер қорындағы кестенің барлық жазбаларына, яғни мәліметтеріне қол жеткізуге болады. DatabaseName қасиеті берілгендер қорының псевдонимін немесе оның каталогының толық адресін көрсетсе, ReadOnly қасиетіне **true** мәнімен өшіру қажет. Ал егер басқа қосымшаға сол кестемен байланыс орнатпау үшін Exclusive қасиетіне true мәнін беріңіз.



## Tquery

Бұл компонент Ttable компонентіне ұқсас. Оның көмегімен серверлік локальдық (немесе қашықтағы) берілгендер қоры мен TdataSource арасындағы өзара байланысты орнатып, жобалау кезіндегі бір немесе бірнеше кесте мәліметтеріне қол жеткізуге мүмкіндік береді. Құрылымдық сұраныстар тіліндегі (SQL) командалар көмегімен Tquery компоненті кестеге топтамалы түрде қол жеткізеді. Бұдан әрі SQL-ң барлық стандарттық версияларындағы негізгі 4 командаға негізделі жұмыс істейміз.

SELECT – мәліметтерді таңдау.

INSERT – жаңа жазбалар енгізу

UPDATE- кестені модификациялау

DELETE – мәліметтерді өшіру.

C++Builder тілі BDE серверіне (немесе SQL серверіне), сұраныс жасап, қосымшаға нәтижелер топтамасын (result set) - береді. Active қасиеті кесте мәліметтеріне жасалған **SQL** сұраныстан кейін нәтижелер **кесте** байланысын белсенді етеді. DatabaseName қасиетінде берілгендер қорының псевдонимі немесе сұранысты бағыттауға арналған каталогтың толық жолы көрсетіледі. Params қасиеті берілгендерінің орындалуы кезінде жасалынған динамикалық сұраныс параметрлерін анықтайды. Статикалық немесе динамикалық сұраныстың символикалық жазбасын енгізу үшін **SQL** қасиетінде мәндер графасындағы батырма басу керек.

## БЕРІЛГЕНДЕРДІ БАСҚАРУ КОМПОНЕНТТЕРІ

Қосымшаның мәліметтер көзімен пайдаланушының арасындағы байланысты **Data Controls** вкладкасының компоненттері қамтамасыз етеді.

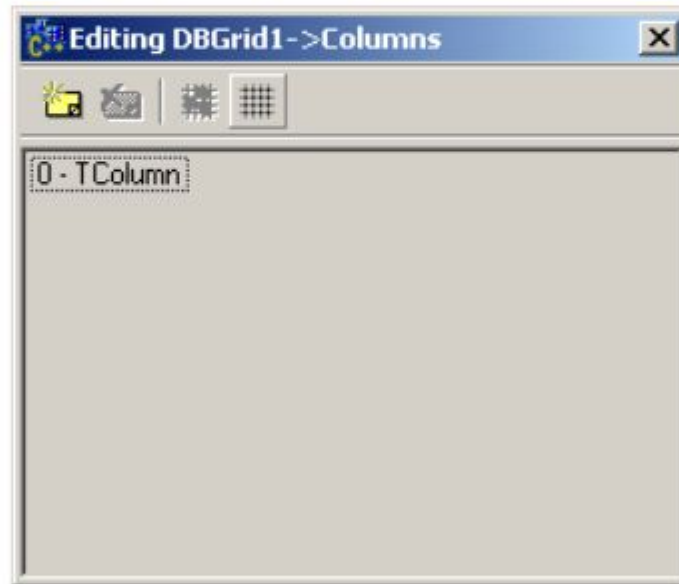


Мұндағы 12 элемент реляциондық берілгендер қорындағы сұраныстар мен кестелеріндегі жазбалар мен бағандардың редакциялануы мен визуализациясын басқарады.

### **TDBGrid**

Берілгендер қорындағы мәліметтерді жиеленген тор түрінде форма бетінде шығару компоненті ондағы мәліметтерді түзету және көріп шығу үшін TDBNavigator компонентін қолдану ыңғайлы.

Мәліметтер форма бетіне шығару үшін TDBGrid компонентін TdataSource компонентіндегі DataSource қасиетіне сәйкес мән бере отырып, байланыстыру қажет. Columns қасиетінде баған атауларының тізімін көрсетуге болады. **Columns Editor** редакциялау терезесін ашу үшін тышқанның оң батырмасын шерту керек.



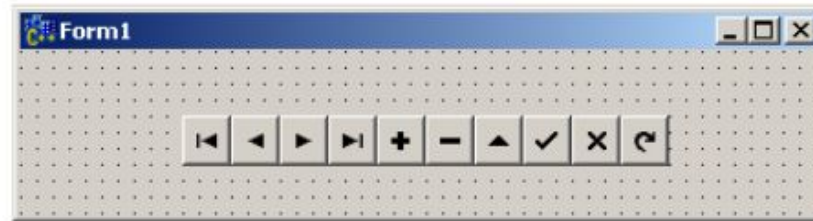
1.18 сурет. Тор компонентінің бағандар редакторы



Options қасиетіндегі мән топ түрін (мысалы бағандар арасындағы бөлгіш сызықтар) және күйін (мысалы 1 бағаннан 2-бағанға өту үшін Tab батырмасын пайдалану) белгілейді. Баған аттарының шрифтерін Tittle Font қасиетін пайдалану керек. Fields қасиетінде мәліметтер қорындағы өріс массивтері көрсетілген, мұндағы Fields қасиетінде – олардың санын, SelectedField - қасиеті – осы уақыттар таңдалған өрісті анықтайды.

### **TDBNavigator**

Берілгендер қорындағы мәліметтерді көру және түзету үшін қолданылады. Пайдаланушы навигатор панеліндегі кез келген батырма басқанда сәйкес іс- әрекет орындалады. Мысалы “+” батырмасын басқанда кестедегі алдыңғы жазу алдына жаңа жазу енгізіледі.



1.19 сурет. TDBGrid компоненті мен навигаторы орналасқан форма панелі

TDBNavigator компонентін мәліметтер қорымен байланыстыру үшін DataSource қасиетіне сәйкес мәнін беру қажет.



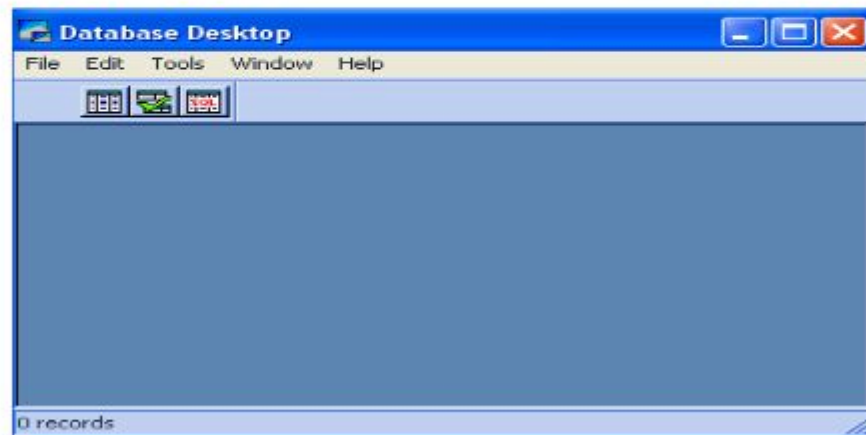
Төменде Делфидегі типтер мен оларға сәйкес келетін C++Builder типтері

<b>Delphi</b>	<b>Ұзындық және мән</b>	<b>C++Builder</b>	<b>Орындалуы</b>
Shortint	8-биттік бүтін	char	typedef
Smallint	16- биттік бүтін	short	typedef
Longint	32- биттік бүтін	long	typedef
Byte	таңбасыз 8- биттік бүтін	unsigned char	typedef
Word	таңбасыз 16- биттік бүтін	unsigned short	typedef
Integer	32- биттік бүтін	int	typedef
Cardinal	таңбасыз 32- биттік бүтін	unsigned long	typedef
Boolean	true/false	bool	typedef
ByteBool	true/false немесе таңбасыз 8- биттік бүтін	unsigned char	typedef
WordBool	true/false немесе таңбасыз 16- биттік бүтін	unsigned short	typedef
LongBool	true/false немесе таңбасыз 32- биттік бүтін	unsigned long	typedef
AnsiChar	таңбасыз 8-биттік символ	unsigned char	typedef
WideChar	Сөз - символ Unicode	wchar t	typedef
Char	8- биттік символ	char	typedef
String	Мәтіндік қатар Delphi	AnsiString	typedef
Single	32-биттік жылжымалы сан	float	typedef

Double	64-биттік жылжымалы сан	double	typedef
Extended	80-биттік жылжымалы сан	long double	typedef
Real	32-биттік жылжымалы сан	float	typedef
Comp	64-биттік жылжымалы сан	double	typedef
Pointer	32-битовый указатель	void *	typedef
PChar	таңбасыз 32-биттік символға сілтеме	unsigned char *	typedef
PansiChar	таңбасыз 32-биттік ANSI символға сілтеме	unsigned char *	typedef
Set	Жиын 1..32 байт	Set<type, minval, maxval>	template class
AnsiString	Мәтіндік қатар Delphi	AnsiString	class
Variant	Варианттық мән, 16 байт	Variant	class
TdateTime	даталық және уақыт мәні, 64-биттік жылжымалы сан	TDateTime	class
Currency	Валюта, 64-биттік жылжымалы сан, нүктеден кейін 4 сан	Currency	class

Бұл бағдарламаны құру үшін берілгендер қорын қолданамыз. Ол үшін бізге C++ Builder құрамына енгізілген Database Desktop қосымша бағдарлама утилитасы қажет. Database Desktop қосымша бағдарлама утилитасы былайша құрылады:

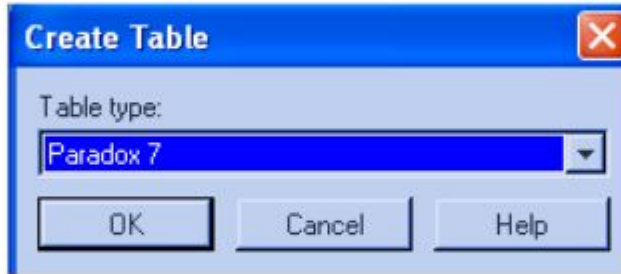
1. Ол үшін Tools - Database Desktop (Сервис – БҚ жұмыс столы / қабықшасы) командасын беру керек (немесе файл DBD32.exe табуға болады). Төмендегі суреттегідей бағдарлама көрінеді (Сурет 7).



Сурет 7- Database Desktop қосымша программасының негізгі терезесі

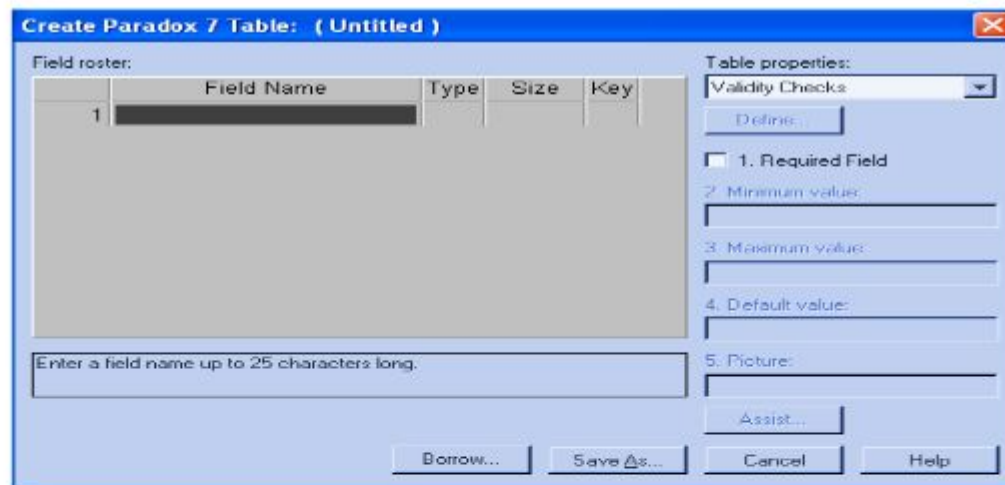


2. Терезеден File – New – Table (Файл – Жаңа - Кесте) командасын орнату керек. Create Table (Кесте құру) сұхбаттық терезесі шығады. Онда ашылатын тізімнен қажетті кесте форматын таңдау керек (Paradox не dBase). Алғашқы рет Paradox7 форматты іске қосылып, терезе өрісінде осы атау көрінеді (Сурет 2.8).



Сурет 8 – Paradox форматы

3. Paradox7 форматымен келісіп, ОК түймесін шерту керек (онда пайдалануға болатын өріс типтері жеткілікті). Бірінші бағанында автоматты түрде өрістің нөмірі жазылып қойылған кесте өрісі атауын сипаттау терезесі көрінеді (Сурет 9).



Сурет 9 – Суретті сипаттау терезесі

4. Екінші бағаннан бастап ретімен өріске берілетін атауды (FieldName), өріс типін (Type), өлшемін (Size) енгізу керек (өлшем өріс символдық типті болса ғана енгізіледі). Өрісті кілттік ету қажет болса, оның бесінші бағанда (Key) көрсетілуі тиіс. Ол ‘\*’ символы арқылы белгіленеді (жалпы, кілт C++ Builder-де және MS Access сияқты арнайы БҚБЖ-лерінде кеңінен пайдаланылады).

4. Екінші бағаннан бастап ретімен өріске берілетін атауды (FieldName), өріс типін (Type), өлшемін (Size) енгізу керек (өлшем өріс символдық типті болса ғана енгізіледі). Өрісті кілттік ету қажет болса, оның бесінші бағанда (Key) көрсетілуі тиіс. Ол ‘\*’ символы арқылы белгіленеді (жалпы, кілт C++ Builder-де және MS Access сияқты арнайы БҚБЖ-лерінде кеңінен пайдаланылады).

Өрістер сипатталып болған соң кестені құру қиын емес. Ол үшін Database Desktop терезесінің мәзірінен File- Open- Table (Файл-Ашу-Кесте) командасын беру керек. OpenTable терезесі көрінеді. Терезеден керекті кестені ашып, Table-Edit Data арқылы кестені толтыру керек (Сурет 2.5). Кестеде қазақ шрифті енгізу үшін Database Desktop мәзірінен Edit-Preferences командасын беріп, ашылған терезенің General қосымша бетінен Change түймесін шерту керек. Терезеден қажетті шрифті тандап, ОК түймесін шерту керек. Форма бетіне осы кестедегі мәліметті шығару үшін үш компонентті орналастыру керек. Олар BDE бетіндегі Table1, Data Access бетіндегі DataSource1, Data Controls бетіндегі DBGrid1 компоненттер. Инспектор объектісіндегі Table1 компонентінің Name- Table1, DatabaseName-Aigul, TableName-Sotka.db, Active-True; DataSource1 компонентінің Name-DataSource1, DataSet-Table1, DBGrid1 компонентінің Name-DBGrid1, DataSource-DataSource1, Font-Times kaz, 10 сәйкестендіріп жазамыз. Форманы компиляциялағаннан кейін кестеге мәліметтерді енгізе беруге болады.