

### Тема 13. Работа с реляционными базами данных в Delphi

Delphi - система быстрой разработки приложений. Одновременно она обладает практически всеми возможностями современных СУБД, таких как Microsoft Access, Visual Fox Pro и др. Она позволяет удобно создавать приложения, визуально подготавливать запросы к БД, задавать SQL-запросы.

#### *Общие принципы организации БД в Delphi*

Приложение осуществляет доступ к БД через BDE (Borland Database Engine – процессор БД Borland). Процессор BDE является совокупностью динамических библиотек и драйверов, обеспечивающих доступ к данным. Приложение через BDE передает запрос к БД, а обратно получает требуемые данные.

Различают локальные БД и удаленные БД. В первом случае используются локальные приложения, во втором – клиент-серверные приложения (рис. 13.1, 13.2).

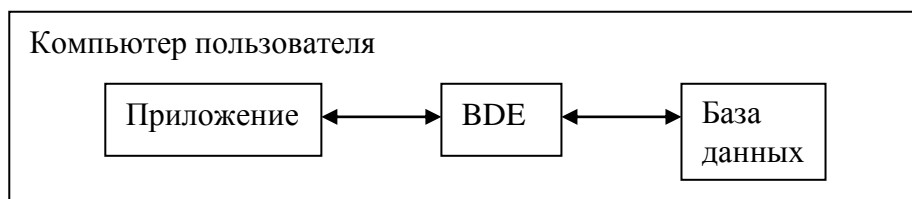


Рис. 13.1. Локальная архитектура ИС.

Для работы с базами данных в Delphi имеется следующий набор инструментальных средств:

- BDE.
- Database Desktop – программа для создания и редактирования таблиц, SQL-запросов, запросов QBE.
- SQL Explorer – проводник БД, позволяющий просматривать и редактировать БД и словари данных (описания).
- SQL Builder – программа визуального конструирования SQL-запросов.

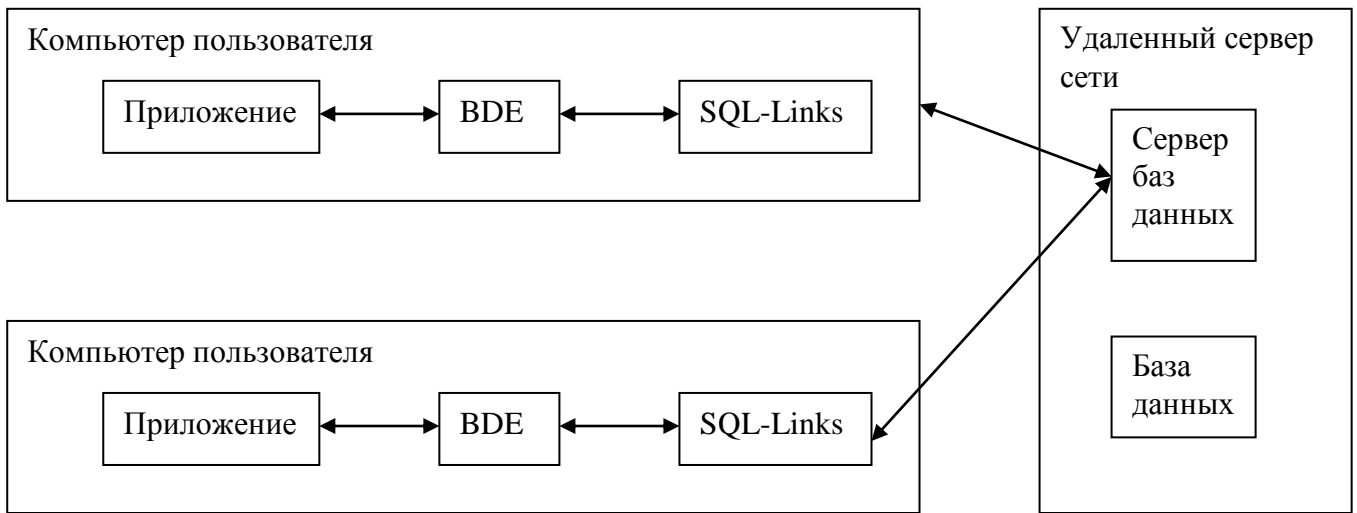


Рис. 13.2. Архитектура «клиент-сервер».

Разработчику БД доступны разнообразные, связанные с БД компоненты. Невизуальные компоненты предназначены для организации доступа к данным таблиц. Они являются промежуточным звеном между таблицами и визуальными компонентами. Визуальные компоненты используются для создания пользовательских интерфейсов приложений. С их помощью выполняется просмотр, редактирование данных и др.

### Построение приложений

Рассмотрим создание приложения, способного перемещаться по строкам таблицы БД, просматривать и редактировать поля и записи. Взаимосвязь компонентов, формы и таблиц БД для этого случая представлены на рис. 13.3.

Компонент Table1 обеспечивает взаимодействие с таблицами БД. Для этого устанавливаются свойства DataBaseName (путь), TableName и Active.

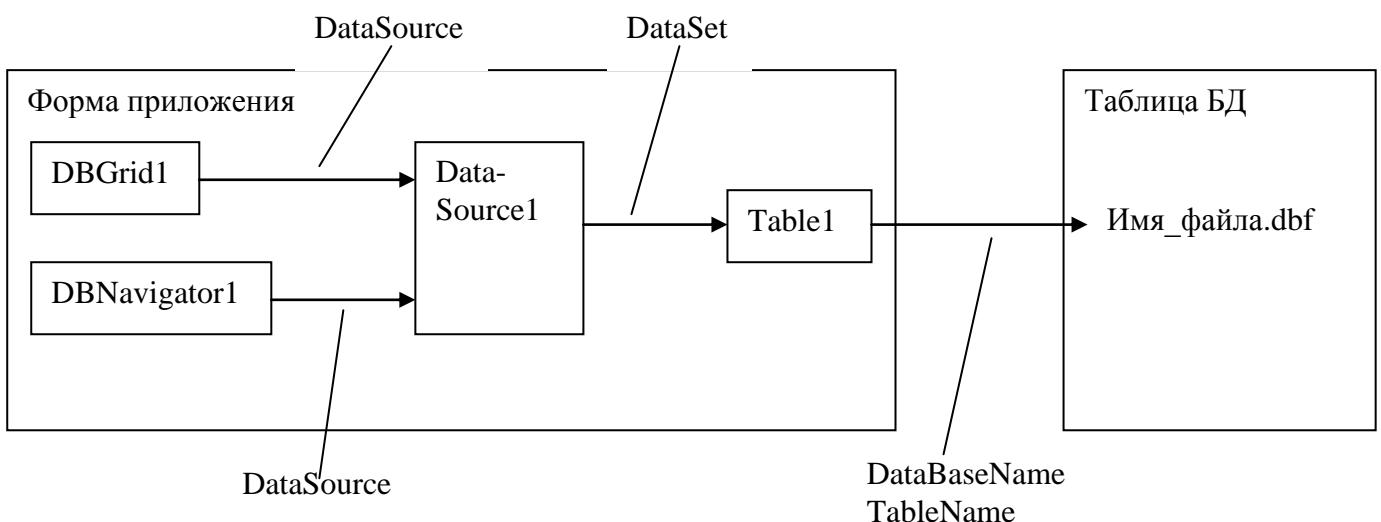


Рис. 13.3. Связь компонентов и таблиц БД.

Компонент DataSource1 – промежуточное звено между Table1, с одной стороны, и DBGrid1 и DBNavigator1, с другой стороны.

Компонент DBGrid1 отображает содержимое таблицы в виде сетки, где столбцы – поля, а строки – записи. Компонент позволяет просматривать и редактировать данные.

Компонент DBNavigator1 позволяет перемещаться по таблице, редактировать, вставлять и удалять записи.

Для настройки свойств компонентов можно использовать инспектор объектов.

Delphi позволяет разгрузить форму от невизуальных компонентов. Для этого используется модуль данных (DataModule). Компоненты DataSource1 и Table1 можно разместить в модуле данных, отделяя тем самым управление БД от обработки данных.

### ***Компоненты доступа к данным***

Компоненты доступа – невизуальные компоненты, служащие для связи с таблицами баз данных. Основным компонентом является набор данных (DataSet), представляющий совокупность записей, взятых из одной или нескольких таблиц, и рассматриваемый как логическая таблица.

Для работы с наборами данных используются компоненты Table, Query, UpdateSQL, DecisionQuery, StoredProc, являющиеся потомками класса TDataSet. Особенно часто используются компоненты Table и Query.

Набор данных связывается с таблицами БД с помощью свойства DatabaseName. При установке DatabaseName свойство Active должно иметь значение False. Размер набора данных задается свойством RecordCount.

### **Пример.**

Просмотр записей набора данных Table1.

...

```
Table1.First;
```

```
for i:=1 to Table1.RecordCount do
```

```
begin
```

```
Table1.Next;
```

...

```
end;
```

Здесь First – чтение первой записи, Next – чтение последней записи набора данных.

Имеются два способа доступа к данным:

- навигационный,
- реляционный.

Навигационный способ предполагает позаписную обработку набора данных Table или Query; используется в локальных базах данных.

Реляционный способ доступа основан на обработке группы записей Query, связанной с SQL-запросом; ориентирован на работу с удаленной БД.