Chapter 6
Delphi's Database Application Development

Professor Zhaoyun Sun

INTRODUCTION

- This chapter introduces Delphi's database tools, including the Data Access and Data Controls component pages, the Fields Editor, the Database Desktop, and the Database Forms Expert.

6.1 What you should know first

- Building a database application is similar to building any other Delphi application

- This book assumes you understand the basic application development techniques, including:

6.1 What you should know first

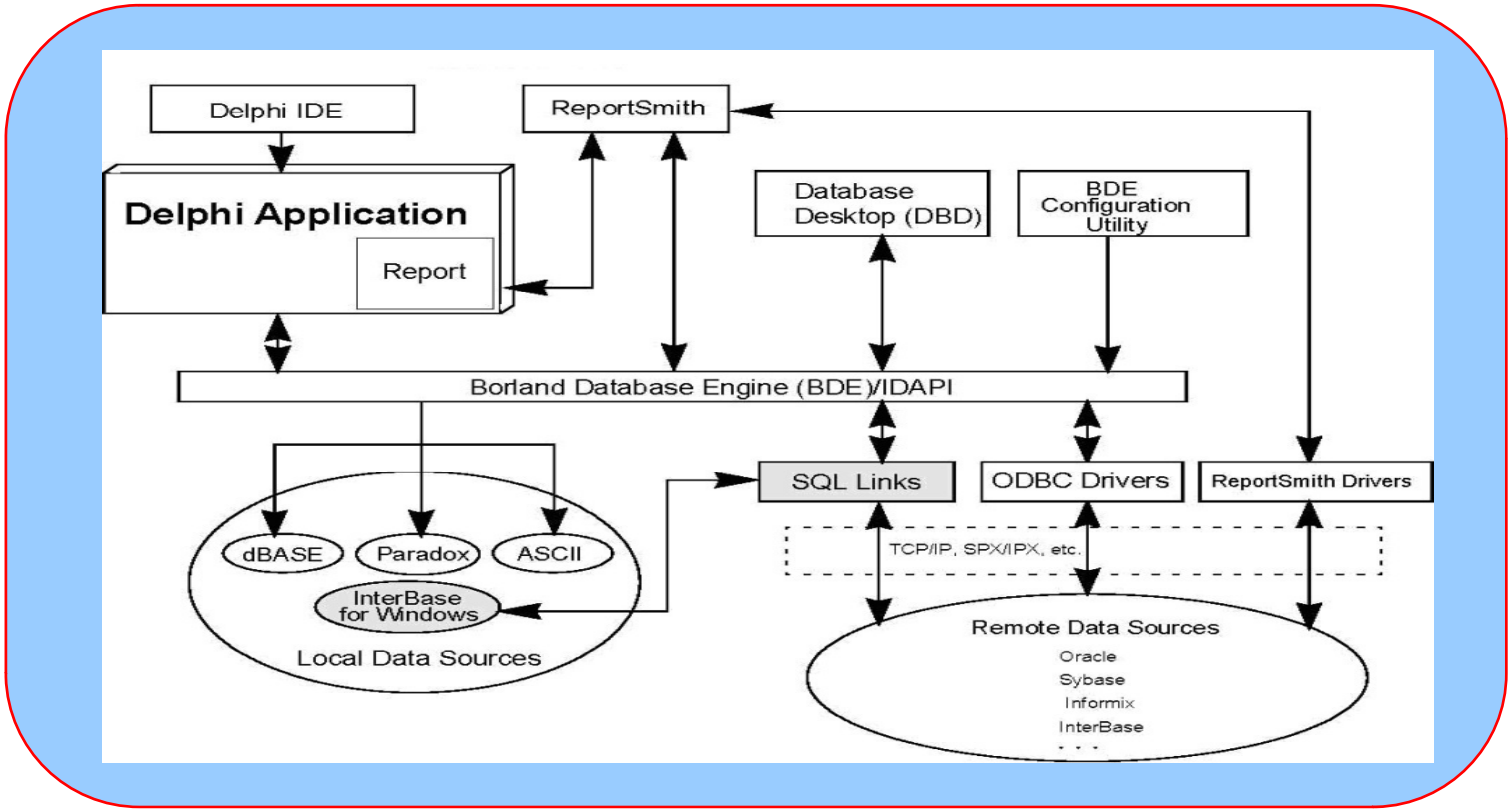
- Creating and managing projects
- Creating forms and managing units
- Working with components, properties,
- and events
- Writing simple Object Pascal source code

6.1 What you should know first

- You also need to have a working knowledge of the Database Management System (DBMS) your Delphi database applications access, whether it is a desktop database such as dBASE or Paradox, or an SQL server.

6.1 What you should know first

Delphi Database Architecture



6.2 Overview of Delphi's database features and capabilities

- A Delphi database application is built using Delphi database development tools, Delphidata-access components, and data-aware GUI components.

- The following table summarizes Delphi's database features

6.2 Overview of Delphi's database featuresand capabilities

Database features summary

Tool	Purpose
Data Access components	Access databases, tables, stored procedures, and custom component editors
Data Control components	Provide user interface to database tables.
Database Desktop (DBD)	Create, index, and query Paradox and dBASE tables, and SQL databases. Access and edit data from all sources.
ReportSmith	Create, view, and print reports.
Borland Database Engine	Access data from file-based Paradox and dBASE tables, and from local InterBase server databases.
BDE Configuration Utility	Create and manage database connection Aliases used by the BDE.
Local InterBase Server	Provides a single-user, multi-instance desktop SQL server for building and testing Delphi applications, before scaling them up to a production database, such as Oracle, Sybase, Informix, or InterBase on a remote server.
InterBase SQL Link	Native driver that connect Delphi applications to the Local InterBase Server.

6.2 Overview of Delphi's database featuresand capabilities

Additional Delphi Client/Server database features

Tool	Purpose
SQL Drivers	Both SQL Links and ReportSmith provide native drivers that connect Delphi database applications to remote SQL database servers, such as Oracle, Sybase, Microsoft SQL Server, Informix, and InterBase.
Visual Query Builder	Creates SQL statements by visually manipulating tables and columns.

What is a database?

- A database consists of one or more tables, where each table contains a series of columns into which records (also called “rows”) are stored.
- Each record is identical in structure.
- For example, a database of addresses consists of a table with name, street address, city, state, and zipcode columns.

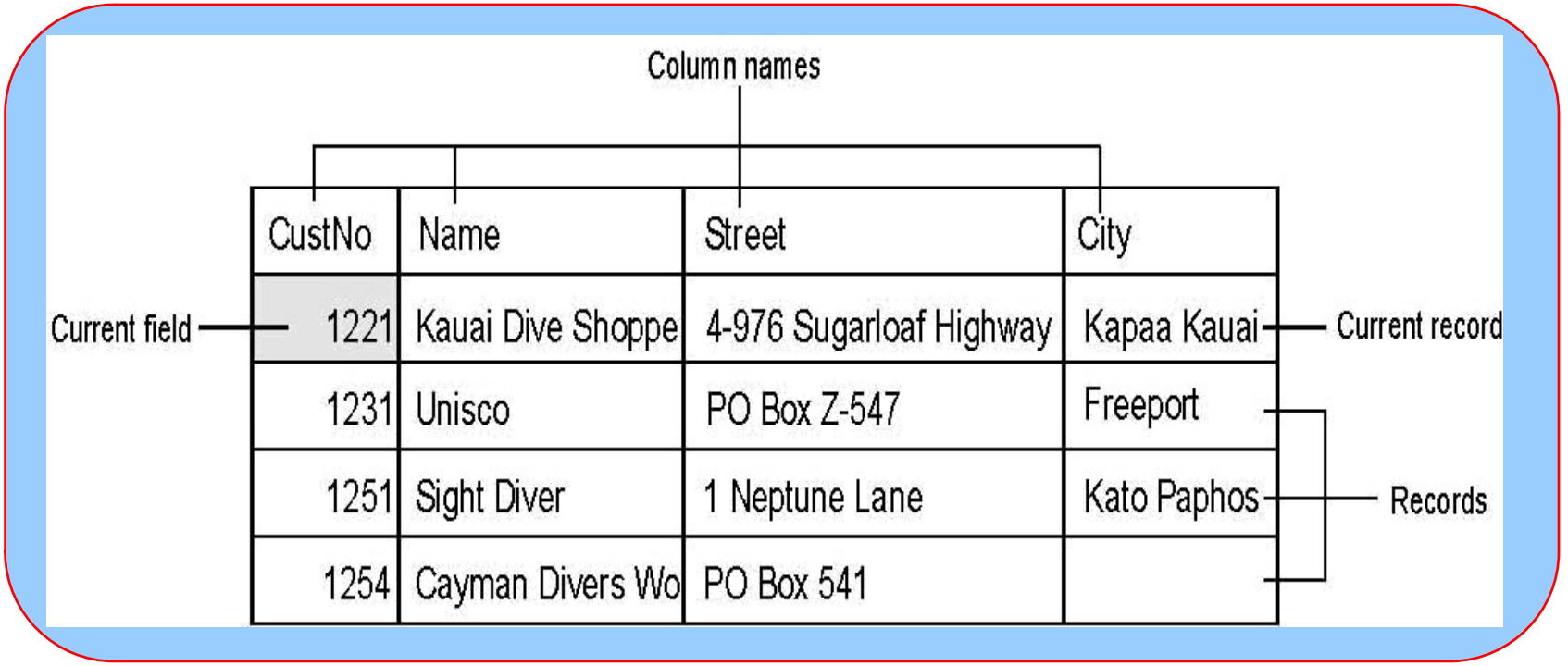
What is a database?

- The intersection of a single column and row is referred to as a field.

- Fields contain values. The following figure illustrates these concepts

What is a database?

Structure of a table



What is data access?

- Delphi applications can access data from desktop database tables on a file server or local disk drive and from remote database servers.

- To access a data source, a Delphi application uses Data Access components to establish a connection through the BDE.

What is data access?

- The installation program for Delphi installs drivers and sets up configurations for Paradox, dBASE, and the Local InterBase Server so you can begin working with tables native to these systems immediately.

What is data access?

- The BDE uses aliases as convenient shorthand names for often-used data sources, whether local or remote.

- The BDE Configuration Utility enables you to define and modify aliases that Delphi applications can use immediately.

What is data access?

- Once drivers are installed and network connections established, Delphi applications can access data from any authorized server

What is data access? Delphi data sources

Data source	Description	File extension
Paradox	Tables created with Paradox or Database Desktop. Each table is in a separate file.	.DB
dBASE	Tables created with dBASE or Database Desktop. Each table is in a separate file.	.DBF
ASCII files	Tables created with Database Desktop. Each table is in a separate file.	.TXT
Local InterBase Server	Database created with InterBase Windows ISQL. Multiple tables in a single database file.	.GDB
SQL Database Server: Oracle, Sybase, Microsoft SQL Server, Informix, InterBase	Database created with server-specific tools, or the DBD, accessed across network with SQL Links. Delphi Client/Server Edition only.	Depends on server
ODBC data sources	Databases such as Microsoft Access, Btrieve, FoxPro, etc.	Depends on data source

6.3 Delphi database architecture

- Delphi uses object-oriented components to create database applications, just as it does with non-database applications.

- Like standard components, database components have attributes, or properties, that are set by the programmer at design time. These properties can also be set programmatically at run time.

6.3 Delphi database architecture

- The Delphi Component palette provides two database component pages:

  – The *Data Access page* contains Delphi objects that simplify database access by encapsulating database source information, such as the database to connect to, the tables in that database to access, and specific field references within those tables.

6.3 Delphi database architecture

- **The Delphi Component palette provides two database component pages:**

–Examples of the most frequently used data access objects include *TTable*, *TQuery*, *TDataSource*, and *TReport*.

6.3 Delphi database architecture

–The *Data Controls page* contains data-aware user interface components for displaying database information in forms.

–Data Control components are like standard user interface components, except that their contents can be derived from or passed to database tables.

6.3 Delphi database architecture

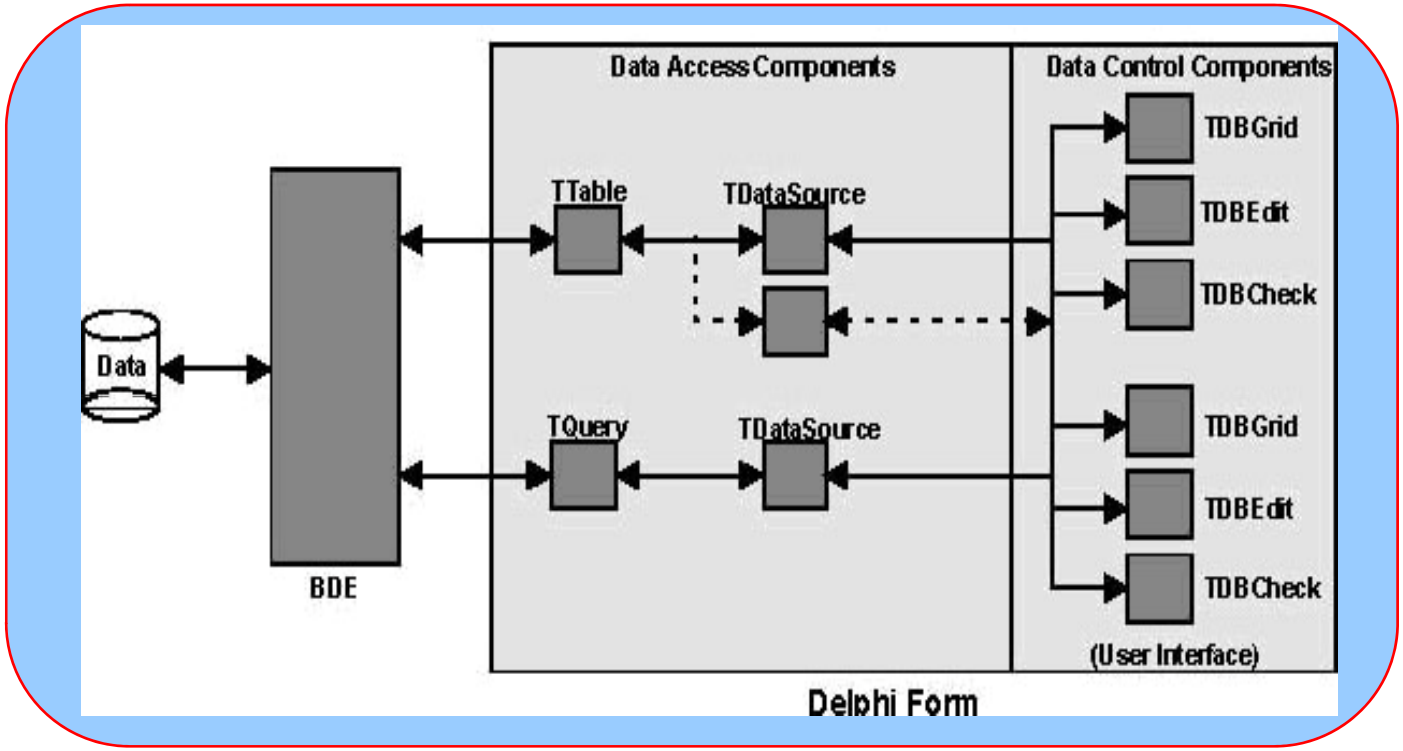
–Examples of the most frequently used data control components include *TDBEdit*, *TDBNavigator*, and *TDBGrid*.

–Datasets, such as *TTable*, *TQuery*, and *TStoredProc* components, are not visible at run time, but provide applications their connection to data through the BDE.

– Data Control components are attached to dataset components by a *TDataSource* component, to provide a visual interface to data.

6.3 Delphi database architecture

Database Components Architecture



6.3 Delphi database architecture

- As this figure illustrates, a form usually contains at least three database components: a dataset component (*TTable* and *TQuery* in the figure) that communicates with the BDE;

6.3 Delphi database architecture

- a *TDataSource* component that acts as a conduit between a dataset component and the user interface;

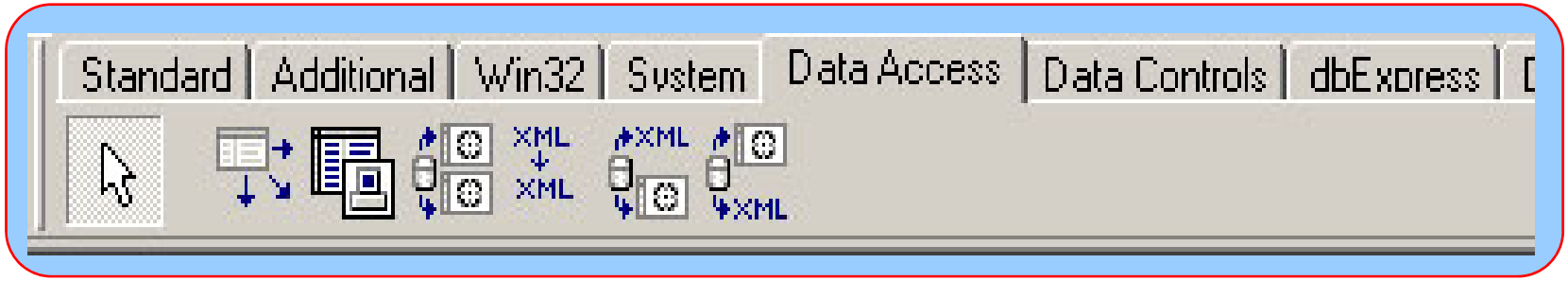
- and one or more data control components, such as *TDBEdit* or *TDBGrid*, that enable a user to browse, edit, or enter data.

6.4 Overview of the Data Access page

- The Data Access page of the Delphi Component palette provides a set of database encapsulation objects that simplify database access.

6.4 Overview of the Data Access page

Data Access page of the component Paltette



6.4 Overview of the Data Access page

- The following table lists the data access objects on the Data Access page, and briefly describes how they are used:

6.4 Overview of the Data Access page

Data Access components

Component	Purpose
TdataSource	Acts as a conduit between a TTable, TQuery, TStoredProc component and data-aware components, such as TDBGrid.
Ttable	Retrieves data from a database table via the BDE and supplies it to one or more data-aware components through a *TDataSource* component. Sends data received from a component to a database via the BDE.
TQuery	Uses SQL statements to retrieve data from a database table via the BDE and supplies it to one or more data-aware components through a TDataSource component, or uses SQL statements to send data from a component to a database via the BDE.
TstoredProc	Enables an application to access server stored procedures. Sends data received from a component to a database via the BDE.
TDatabase	Sets up a persistent connection to a database, especially a remote database requiring a user login and password.
TbatchMove	Copies a table structure or its data. Can be used to move entire tables from one database format to another.
Treport	Enables printing and viewing of database reports through ReportSmith.

6.5 Understanding TTable

- The *TTable* component is the easiest way for a programmer to specify a database table for access. To put a *TTable* component on a form:

—Select the Data Access page from the Component palette.

— Click the Table icon.

— Click on the form to drop the TTable component.

6.5 Understanding TTable

- Enter the directory where the database resides in the *DatabaseName* property of the Object Inspector window. For SQL databases, enter an alias name.

- Enter the name of the table to use in the *TableName* property of the Object Inspector window, or you can also choose a table from the drop-down list instead of entering the name.

6.5 Understanding TTable

- **Note** An alias can also be used for local Paradox and dBASE tables. You can choose an alias from a drop-down list in the Object Inspector.

6.6 Understanding TQuery

- The *TQuery* component provides a tool for data access using SQL statements, such as a SELECT statement, to specify a set of records and a subset of columns from a table.

6.6 Understanding TQuery

- ***TQuery*** **is useful for building local SQL queries against Paradox and dBASE data, and for building client/server applications that run against SQL servers.**

6.6 Understanding TQuery

- To put a *TQuery* component on a form:

  – Select the Data Access page from the Component palette.
  – Choose the Query icon.
  – Click on the form to drop the *TQuery* component.

6.6 Understanding TQuery

- Enter the directory where the database resides (or select an alias for SQL databases) in the *DatabaseName* property of the Object Inspector window.

- Enter the SQL statement to use for data access in the SQL property of the Object Inspector window by clicking the list button to open the String Editor.

6.7 Understanding TDataSource

- Every dataset that supplies a data control component must have at least one *TDataSource* component.

- *TDataSource* acts as a bridge between one *TTable*, *TQuery*, or *TStoredProc* component and one or more data control components that provide a visible user interface to data.

6.7 Understanding TDataSource

- ***TTable*** **and** ***TQuery*** **can establish connections to a database through the BDE, but they cannot display database information on a form.**

6.7 Understanding TDataSource

- Data Control components provide the visible user interface to data, but are unaware of the structure of the table from which they receive (and to which they send) data.

- A *TDataSource* component bridges the gap.

6.7 Understanding TDataSource

- To put a *TDataSource* component on a form:
  - Select the Data Access page from the Component palette.
  - Choose the DataSource icon.
  - Click on the form to create the *TDataSource* component.

  - Enter the name of the *TTable* or *TQuery* component to use as a database connection source in the DataSet property of the Object Inspector. If the form contains any *Ttable* or *TQuery* components, you can choose a component from the drop-down list instead.

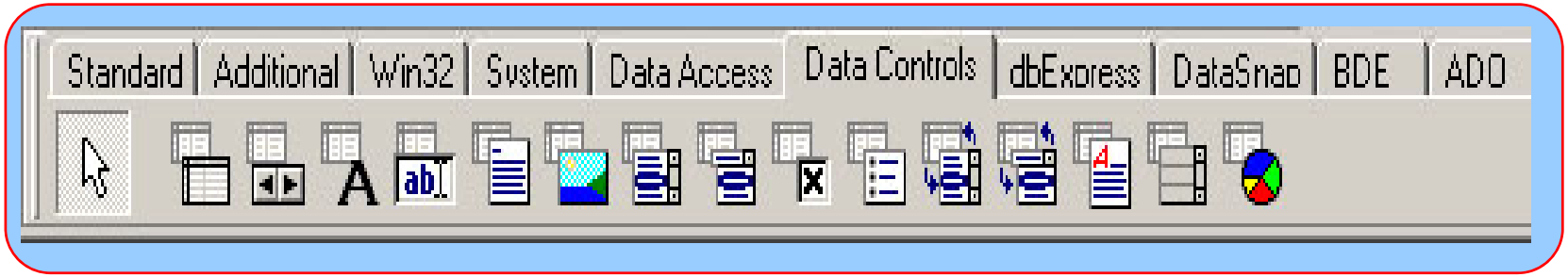
6.8 Overview of the Data Controls page

- **Note** TDataSource is also used to link tables or queries in a master/detail form.

- The Data Controls page provides a set of data-aware user-interface components that you can use to create forms-based database applications.

6.8 Overview of the Data Controls page

The Data Controls Page of the Component Palette



- **data controls can display data from a field in a database table, or send new or modified data from a form to a database table.**

6.8 Overview of the Data Controls page

Data Controls components

Component	Purpose
TDBNavigator	Data-aware navigation buttons that move a table's current record pointer forward or backward; start Insert or Edit mode; post new or modified records; cancel Edit mode; and refresh display to retrieve updated data.
TDBText	Data-aware label that can display a field from a currently active record.
TDBEdit	Data-aware edit box that can display or edit a field from a currently active record.
TDBCheckBox	Data-aware check box that can display or edit a Boolean data field from a currently active record.
TDBListBox	Data-aware list box that can display values from a column in a table.
TDBComboBox	Data-aware combo box that can display or edit values from a column in a table.
TDBRadioGroup	Data-aware radio group populated with radio buttons that can display or set column values.
TDBGrid	Data-aware custom grid that enables viewing and editing data in a tabular form similar to a spreadsheet; makes extensive use of *TField* properties (set in the Fields Editor) to determine a column's visibility, display format, ordering, etc.
TDBMemo	Data-aware memo box that can display or edit text BLOB data from a currently active record.
TDBImage	Data-aware image box that can display, cut, or paste bitmapped BLOB images to and from a currently active record.
TDBLookupList	Data-aware list box that displays values mapped through another table at run time.
TDBLookupCombo	Data-aware combo box that displays values mapped through another table at run time.

6.8 Overview of the Data Controls page

- Data control components make up a consistent visual user interface for Delphi database applications, whether the application accesses a local database file, or a remote database server.

6.9 Overview of the Database Forms Expert

- The Database Forms Expert automates many of the tasks necessary for creating dataentry or tabular forms from an existing database table.

- It can generate simple or master/detail forms using *TTable* or *TQuery* components. The Database Forms Expert automates such form building tasks as:

6.9 Overview of the Database Forms Expert

- Placing database components on a form. Connecting *TDataSet* components (e.g., *TTable* and *TQuery*) to a database.

- Connecting *TDataSource* components to interactive data control components and *TTable* or *TQuery* data access objects.

- Writing SQL statements for *TQuery* objects. Defining a tab order for components.

6.9 Overview of the Database Forms Expert

- Inexperienced database applications programmers can use the Database Forms Expert to learn how to build database forms.

- Experienced database applications programmers can use it to speed application development.

6.10 Overview of the Database Desktop

- The Database Desktop (DBD) is a database maintenance and data definition tool.
- It enables programmers to query, create, restructure, index, modify, and copy database tables, including Paradox and dBASE files, and SQL tables.
- You do not have to own Paradox or dBASE to use the DBD with desktop files in these formats.

6.10 Overview of the Database Desktop

- The DBD can copy data and data dictionary information from one format to another.

- For example, you can copy a Paradox table to an existing database on a remote SQL server.