

第十讲 Delphi数据库基础

吕松茂

lsmtech@163.com

内容提要

✿ 数据库基础知识

✿ SQL语言基础

✿ Delphi的数据库访问机制

内容提要

数据库基础知识

- ✿ SQL语言基础

- ✿ Delphi的数据库访问机制

一、数据库基础知识

❁数据库（Database，DB）：为了满足一定范围内众多用户的需要，在计算机中建立的一个可以长期存储的、有组织的、可共享的数据集合。一个数据库可以包含许多数据表文件、索引文件以及其他文件。

❁数据库系统（Database System，DBS）：具有数据管理功能的计算机系统，一般由数据库、数据库管理系统和应用系统构成。

❁数据库管理系统（Database Management System，DBMS）：为用户或应用程序提供访问数据库的方法，包括数据库的创建、查询、更新以及各种数据控制等。

一、数据库基础知识

数据库系统根据数据的组织方式，主要分为层次数据库系统、网状数据库系统、关系型数据库系统和面向对象数据库系统等几类。

一、数据库基础知识

Delphi支持关系型数据库，关系型数据库由表组成，一个关系用一个二维表来定义：表的行定义了事物的一组属性数据，称为记录；表的列定义了事物的某种属性，称为字段。

名称	省会	面积	人口
四川省	成都市	510000 平方千米	8673万
广东省	广州市	440000 平方千米	7859万

一、数据库基础知识

一个数据库应用程序在逻辑上通常由两部分组成：数据库访问模块和用户界面，这就是数据库应用程序的层次结构。在单层的数据库应用程序中，应用程序和数据库共享一个文件系统，它们使用本地数据库或文件来存取数据；在双层的数据库应用程序中，客户程序提供用户界面，通过各种数据访问引擎（如BDE、ADO等）从远程数据库服务器获取数据；在多层的数据库应用程序中，客户程序、应用服务器和远程服务器分布在不同的机器上。

内容提要

✿ 数据库基础知识

➡ SQL语言基础

✿ Delphi的数据库访问机制

二、SQL语言基础

- ✿ SQL (Structured query Language, 结构化查询语言), 功能包括查询、操纵、定义、控制。
- ✿ 功能丰富、语言简洁、使用灵活, 倍受欢迎。
- ✿ 1974年由Boyce, Chamberlin提出, 1975年在IBM公司的System R上首次实现。原型称为SEQUEL (Structured English QUery Language)。
- ✿ 1986年ANSI SQL-86标准, 是第一个SQL 标准。1987年成为国际标准。
- ✿ 1992年SQL-92标准 (简称SQL2) 。

二、SQL语言基础

✿ 1999年SQL-99标准（简称SQL3）。增加了对对象关系模型的支持。目前仅部分实现。

✿ 最流行的国际标准数据库语言。

主要特点：

✿ SQL是非结构化的语言，使用SQL 查询数据库时，只需要告诉它做什么而不用告诉它如何去做。

✿ SQL 本身不提供任何程序流程控制结构，而是通过PL/SQL提供SQL语言的过程化功能

✿ SQL提供相对固定的数据类型，一般不需要扩展

✿ SQL本身十分灵活，方便易学。

二、SQL语言基础

标准SQL只包含8种语句：

✿数据查询： `select`

✿数据定义： `create, drop, alter`

✿数据操纵： `insert, update, delete`

✿数据控制： `grant, revoke`

二、SQL语言基础

- ✿ 数据定义语言（DDL），用于定义数据结构
- ✿ 数据操纵语言（DML），用于检索和修改数据结构
- ✿ 数据控制语言（DCL），用于规定数据库用户的各种权限
- ✿ 数据库事务处理，用来保证数据库的完整性

二、SQL语言基础

SQL语句的基本语法如下：

- ✿ 每条SQL语句必须以分号结束
- ✿ 每条SQL语句可以单独写成一行，也可以分成若干行
- ✿ SQL语句对大小写不敏感，对于SQL语句的关键字（如Insert等），表名，列名等，可以大小写混合；但对列的内容则是大小写敏感的。

二、SQL语言基础

查询语句Select简介:

- ✱查询语句Select是使用最多的SQL语句，它完成的是数据库的查询功能。
- ✱在 SQL 中，使用 SELECT 语句可以在需要的表单中检索数据
- ✱在进行检索之前，必须知道需要的数据存储在哪里
- ✱ SELECT 语句可以由多个查询子句组成

二、SQL语言基础

Select语句可用于检索：

- ✱ 全部的行和列
- ✱ 全部的行和特定的列
- ✱ 限定范围的行
- ✱ 与一组值匹配的行
- ✱ 根据未知值检索的行
- ✱ 隐藏有重复值的行
- ✱ 根据多个搜索条件检索的行

二、SQL语言基础

Select语句的基本结构如下：

```
SELECT [ALL | DISTINCT] select_list  
[INTO [new_table_name]]  
FROM {table_name | view_name}      [[,{table_name2 |  
view_name2} [...,{table_name16 | view_name16} ] ]  
[WHERE search_conditions]  
[GROUP BY group_by_list]  
    [HAVING search_conditions]  
    [ORDER BY order_list [ASC|DESC]]
```


二、SQL语言基础

Select例子:

✿Select Emp. 员工编号, emp. 员工姓名, sp. 部门名称
From 员工数据表 as emp,
(select 部门数据表. 部门编号, 部门数据表. 部门名称
From 部门数据表
Where 部门数据表. 部门编号>2) as sp
Where emp. 部门编号=sp. 部门编号

首先使用select语句检索高级部门, 并用别名sp表示该派生表, 然后从员工数据表和sp派生表中检索数据。

二、SQL语言基础

数据插入Insert

✱格式1: INSERT INTO <表名> [(<列名1> [, <列名2>...])] VALUES (<值1> [, <值2 > ...]) ;

- insert语句把values后给出的各值按排列顺序依次赋到表名列出的各列。每次只能插入一条记录。

✱例: INSERT INTO student VALUES (95006, ‘张三’, ‘男’, 21, ‘cs’);

✱例: insert into student(sname, sno, sdept)
values(‘章小山’, 96008, ‘IS’);

二、SQL语言基础

数据插入Insert

★格式2: INSERT INTO <表名> [(<列名1> [, <列名2>...])] Select 语句;

- 把select语句查询的结果按顺序插入到对应的列中。允许一次插入多条记录。
- Select查得的各列值必须与<表名>后的各列名在个数、类型及顺序上一致。
- 若<表名>后没有(列名), 则对该表的所有字段插入。

二、SQL语言基础

数据修改Update

- ★格式：UPDATE <表名> SET <列名1>=<表达式1> [, <列名2>=<表达式2> ...] [WHERE <条件>];
- ★Where后的条件与查询语句中的where条件语法和作用相同。
- ★该语句将指定表中符合WHERE <条件>的行中的指定列赋新值。若没有where，则对整个表修改。
- ★该语句一次可以更新一列或多列；更新一行或多行(由where的选择条件决定)。

二、SQL语言基础

例：将95001学生转入MA系

```
Update Student
```

```
Set sdept = 'MA'
```

```
Where sno = '95001'
```

例：所有学生年龄加1

```
Update Student
```

```
Set sage = sage + 1
```

二、SQL语言基础

删除数据Delete

✿DELETE [FROM] {table_name | view_name}

[WHERE 子句]

- ✿删除表中符合选择条件的行（一行或多行）。
- ✿若无where部分，则删除整个表中全部数据，但表结构仍存在，即成为空表。
- ✿delete只能整行删除，不能只删一行的部分。

二、SQL语言基础

❁ 删除单个元组

例：删除学号为95001的学生的选课信息

```
Delete From SC
```

```
Where sno = '95001'
```

❁ 删除多个元组

例：删除选课而未参加考试的学生的选课信息

```
Delete From SC
```

```
Where grade is null
```

❁ 删除整个关系中的所有数据

例：删除所有学生的选课信息

```
Delete From SC
```

二、SQL语言基础

注意：

- ✱数据更新语句只能对单表操作，不能同时对多表更新。
(从而引起阶段性的数据不一致性)
- ✱数据更新语句仅当事务(transaction)提交(commit)后才正式生效。也可通过事务回滚(rollback)来作废。

内容提要

✿ 数据库基础知识

✿ SQL语言基础

 **Delphi**的数据库访问机制

三、Delphi的数据库访问机制

✿ Delphi与数据库

✿ Delphi的数据库体系

✿ Borland数据库引擎（BDE）

✿ Database Desktop的使用

✿ 数据库浏览器（Database Explorer）

三、Delphi的数据库访问机制

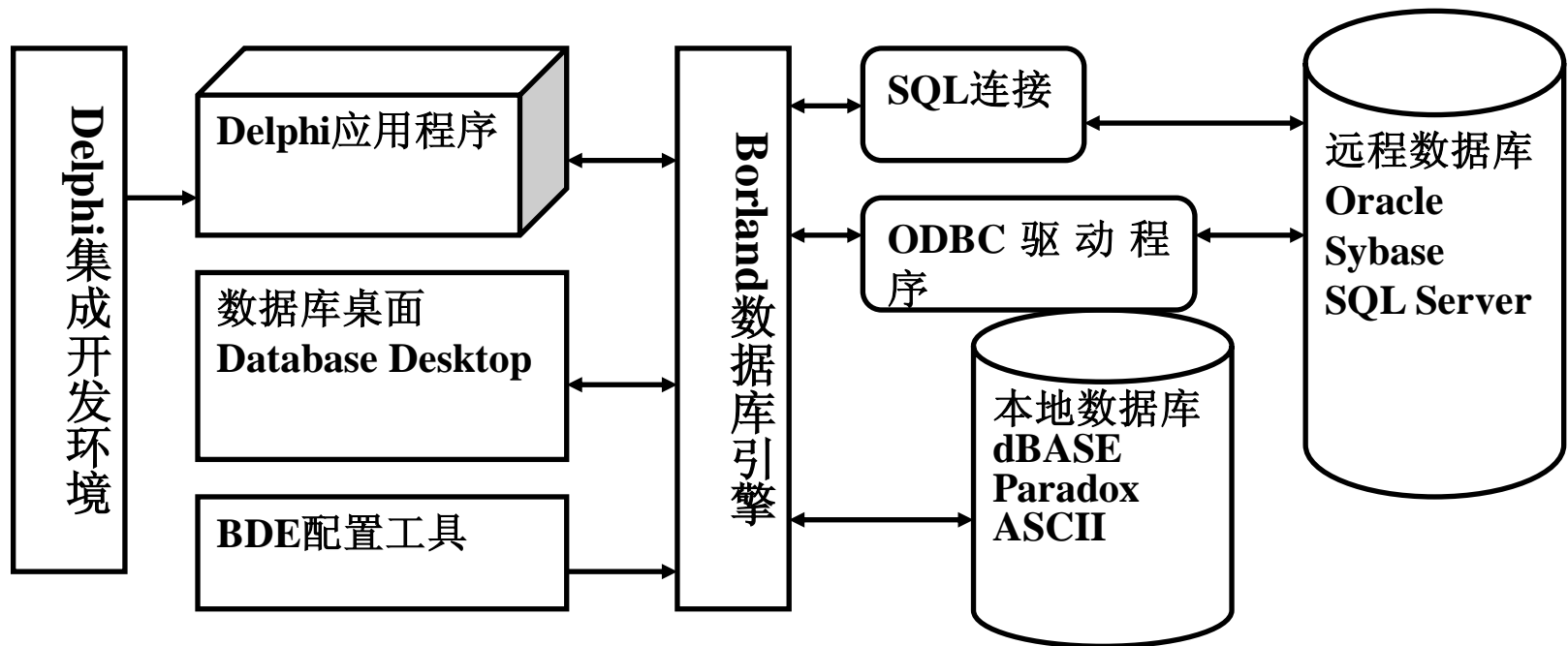
3.1 Delphi与数据库

✿ Delphi提供丰富的组件来访问各种类型的数据库，例如，组件面板的**BDE**页和**ADO**页，以及其他页上的一些通用组件都用于数据库的访问。

✿ Delphi利用**BDE**或**ADO**提供的驱动程序，可以访问不同的数据库。**BDE**提供的驱动程序多用于访问本地数据库，例如**Paradox**和**dBASE**。

三、Delphi的数据库访问机制

3.2 Delphi的数据库体系



三、Delphi的数据库访问机制

3.3 Borland数据库引擎（BDE）

Borland提供的数据库引擎（Borland Database Engine, BDE），通过它可以访问本地数据库和远程数据库中的数据，开发出具有可伸缩体系结构的数据库应用程序。有了BDE，就可以将一组相关的表放进一个目录中，给这组相关表起个别名，从而得到一个逻辑表集合，即虚拟数据库。通过BDE中的这些虚拟数据库，就可以连接到这些表。

三、Delphi的数据库访问机制

通过BDE创建虚拟数据库

BDE 管理器（BDE Administrator）是设置和管理 BDE 的工具，它可以用来管理 BDE 中的数据库别名和驱动器。通过 BDE 管理器，建立一个虚拟数据库，其操作步骤如下：

◆ 步骤一：执行【开始】→【程序】→【Borland Delphi7】→【BDE Administrator】菜单命令，打开 BDE 管理器窗口。

◆ 步骤二：右击 Database 页标签，选择【Object】→【New】菜单项，弹出选择数据库驱动器的对话框，根据需要选择适当的驱动器。单击 OK 按钮，关闭该对话框，回到 BDE 管理器窗口。

三、Delphi的数据库访问机制

通过BDE创建虚拟数据库

◆步骤三：在**BDE**管理器左窗口的新建项目中，输入一个新的数据库别名**MyData**。

◆步骤四：在**Path**框中输入**D:\ Data**（该文件夹由用户事先建立），为数据库别名**MyData**指定路径**D:\ Data**。

注：要删除别名，可在**Database**页标签中选择要删除的别名，右击后选择**Delete**即可。也可以使用后面介绍的数据库桌面和数据库浏览器定义和删除数据库别名。

三、Delphi的数据库访问机制

3.4 Database Desktop的使用

Database Desktop是Delphi中提供的数据库管理工具，使用它可以建立和修改数据表结构，完成简单数据的输入及其它操作。

选择执行菜单【Tools】→【DataBase Desktop】项，或【开始】→【程序】→【Borland Delphi7】→【Database Desktop】菜单项，可以打开Database Desktop的操作窗口。

三、Delphi的数据库访问机制

1. 定义数据库别名

✿选择执行菜单【Tools】→【Alias Manager】项，打开别名管理操作窗口。

✿单击【New】按钮，在Database Alias 和Path 框中分别输入新的别名和路径，可以定义新的别名。

✿单击【Remove】按钮可以删除别名。

三、Delphi的数据库访问机制

2. 设置工作目录及私有目录

✿ 工作目录是指数据库工作平台首先找到文件的地方，私有目录是用户自己使用的目录，其他网络用户无法看到这个目录。

✿ 工作目录菜单项“Work Directory”及私有目录菜单项“Private Directory”都在“File”菜单下。

三、Delphi的数据库访问机制

3. 建立数据表结构

在数据库桌面环境下执行菜单 **【File】** → **【New】** → **【Table】** 项，选择数据库类型Paradox，单击Ok，进入表结构定义窗口。

	Field Name	Type	Size	Key
1	NO	A	5	*
2	NAME	A	10	
3	SEX	A	2	
4	BIRTHDAY	D		
5	MEMBER	L		
6	MAJOR_KEY	A	2	
7	REMARK	M	200	

三、Delphi的数据库访问机制

4. 保存数据表文件

可以将数据表保存在当前目录（当前目录就是工作目录），也可以指定数据库别名，即为数据表确定了相应的文件夹位置，还可以选择其他的文件夹。

三、Delphi的数据库访问机制

5. 建立索引

✿在建立数据表的结构后，还要根据功能的要求对数据表定义关键字。定义关键字的目的是为了对数据表建立一种索引，以便实现多表间的同步操作以及实现对数据表分类排序和快速查询。

✿要建立一个索引，就要指定一个索引关键字（它由一个或几个连续的字段组成）。同时系统要为每个索引建立一个对应的索引文件。

✿索引可分为主索引与次索引两种。主索引关键字所对应的数据必须唯一，且必须是最前面的几个字段，而次索引没有这个限制。每个文件只能建立一个主索引，而次索引可以建立多个。

三、Delphi的数据库访问机制

✿ 建立主索引

在数据库桌面的结构定义窗口，将光标移到关键字字段的**Key**列的下面，双击或按空格键，系统在该位置显示一个“*”号，此字段即成了关键字字段。

✿ 建立次索引

在数据库桌面的结构定义窗口，单击**Table Properties**下拉按钮，选择**Secondary Indexes**菜单项，单击**Define**按钮，打开“**Define Secondary Indexes**”对话框，

选择次关键字字段（可选择多个），完成后按**OK**按钮。这时会出现**Save Index As**对话框，在该对话框中指定次索引文件名。

三、Delphi的数据库访问机制

6. 设置有效性检查

点击Table Properties下拉按钮，选择Validity Checks功能项，然后用鼠标选择一个字段，便可以对该字段设置有效性检查。

7. 设置口令

点击Table Properties下拉按钮，选择Password Security功能项，在弹出的对话框中输入指定的口令，即可对该数据表实现数据保护。

三、Delphi的数据库访问机制

8. 设置参照完整性

点击Table Properties下拉按钮，选择Table Lookup功能项，可以进行参照完整性设置。

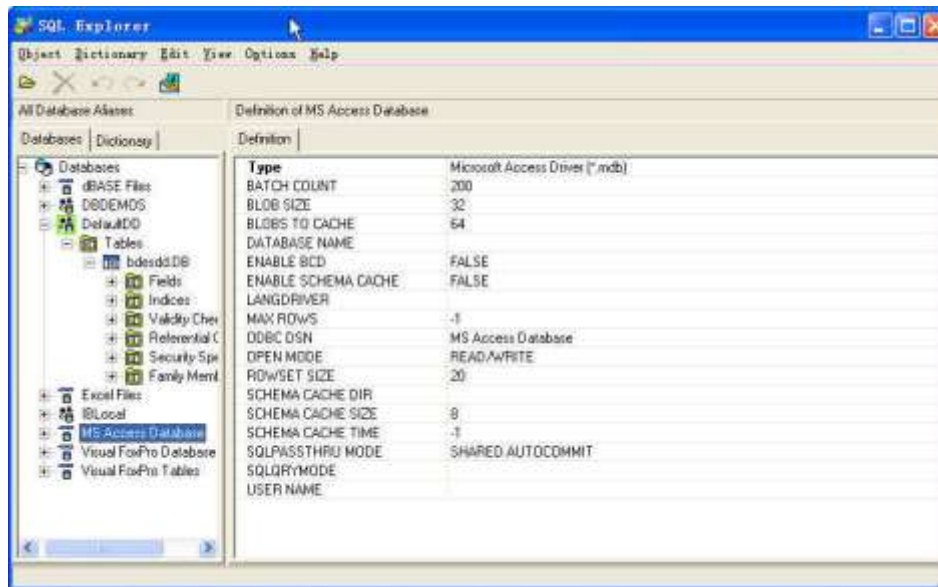
9. 输入数据

数据表中数据的输入，一般来说，既可以在数据库桌面环境下进行，也可以在应用程序的运行环境进行。但有些类型的字段（例如，备注型M、位图型G及对象型O）必须由应用程序提供相应的功能，在程序运行状态下，通过执行某种操作才能完成信息的输入。

三、Delphi的数据库访问机制

3.5 数据库浏览器SQL Explorer

✿ SQL Explorer 也叫做数据库浏览器，单击“开始 | 程序”选项，单击其中的“Borland Delphi7 | SQL Explorer”菜单项，就可以打开SQL Explorer。



三、Delphi的数据库访问机制

✿SQL Explorer主要用于对数据库进行集中管理。使用这个工具我们可以完成下面这些工作：

- ✿浏览和编辑数据库的对象
- ✿创建视图和编辑数据表中的数据
- ✿创建和维护数据库的别名
- ✿输入SQL语句查询特定的信息
- ✿引导数据库管理员配置BDE

总结

✿ 数据库基础知识

✿ SQL语言基础

✿ Delphi的数据库访问机制

谢谢大家

