

本章要点

- 删除和创建数据库
- 表空间

在 Oracle 数据库系统安装完成后,可以创建自己需要的数据库,在数据库应用和开发过程中,需要对数据库使用的表空间进行管理。

3.1 删除和创建数据库

本书在安装 Oracle 数据库时,已创建了数据库 orcl,学生信息数据 stsystem 是本书的案例数据库,在以后的例题中会多次用到,现在使用图形界面方式的数据库配置助手(Database Configuration Assistant, DBCA)删除数据库 orcl,再重新创建数据库 stsystem。

3.1.1 删除数据库

删除数据库的举例如下。

【例 3.1】 使用 DBCA 删除数据库 orcl。

使用 DBCA 删除数据库 orcl 的操作步骤如下。

(1) 选择“开始”→Oracle-OraDB19Home1→Database Configuration Assistant 命令,如图 3.1 所示。

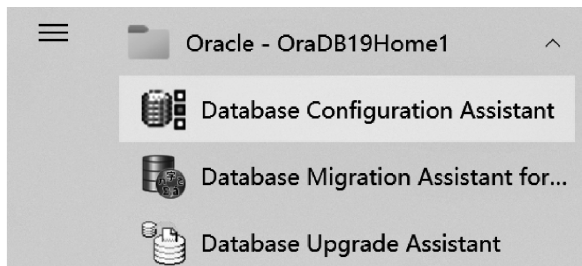


图 3.1 选择 Database Configuration Assistant 命令

(2) 在弹出的“选择数据库操作”窗口中,选中“删除数据库”单选按钮,如图 3.2 所示,然后单击“下一步”按钮。

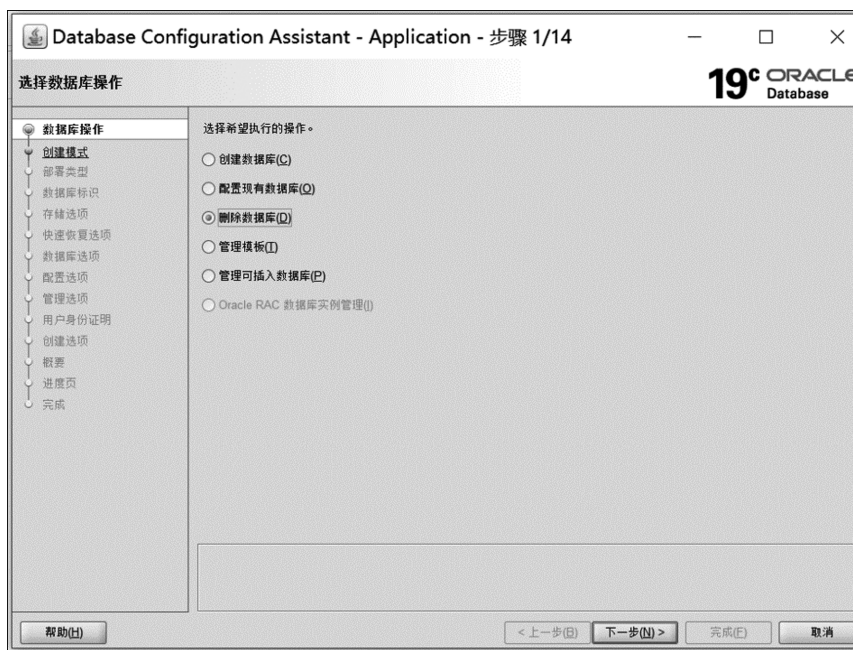


图 3.2 “选择数据库操作”窗口

(3) 进入“选择源数据库”窗口,这里选择 ORCL 数据库,如图 3.3 所示,单击“下一步”按钮。



图 3.3 “选择源数据库”窗口

(4) 进入“选择注销管理选项”窗口,单击“下一步”按钮,进入“概要”窗口,如图 3.4 所示,单击“完成”按钮,在弹出的确认删除提示框中单击“是”按钮,将会显示删除进度,直至完成删除数据库操作。

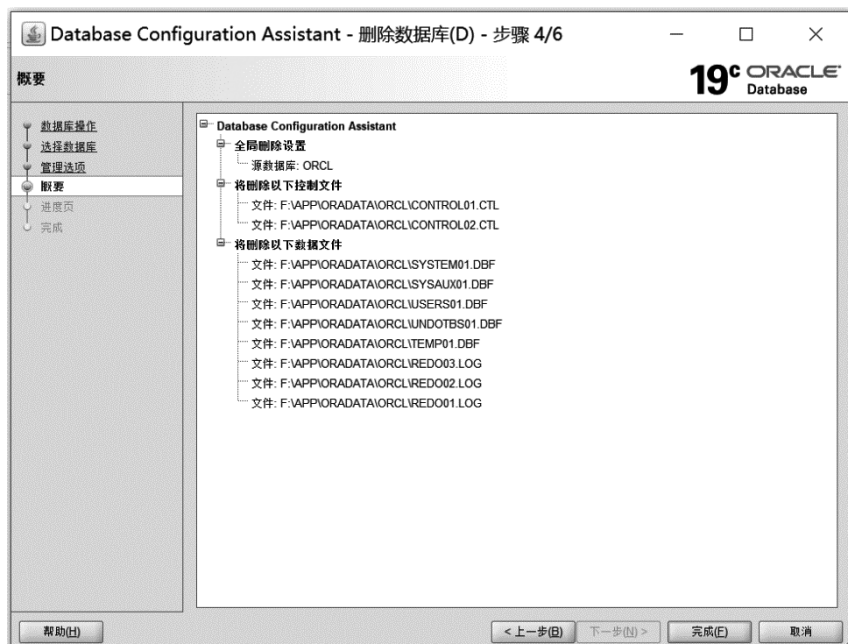


图 3.4 “概要”窗口

3.1.2 创建数据库

使用图形界面方式的数据库配置助手 DBCA 创建数据库的举例如下。

【例 3.2】 使用 DBCA 创建数据库 stsystem。

使用 DBCA 创建数据库 stsystem 的操作步骤如下。

(1) 选择“开始”→Oracle-OraDB19Home1→Database Configuration Assistant 命令。

(2) 在弹出的“选择数据库操作”窗口中,选中“创建数据库”单选按钮,如图 3.5 所示。

(3) 单击“下一步”按钮,将弹出“选择数据库创建模式”对话框,如图 3.6 所示。

- 在“全局数据库名”文本框中输入 stsystem。
- 在“数据库字符集”下拉框中选择“ZHS16GBK-BGK16 位简体中文”。
- 在“管理口令”“确认口令”和“Oracle 主目录用户口令”文本框中,分别输入 Ora123456。
- 取消选中“创建为容器数据库”复选框。

(4) 单击“下一步”按钮,进入“概要”窗口,如图 3.7 所示,单击“完成”按钮,出现“进度页”提示框,直至数据库 stsystem 创建完成。

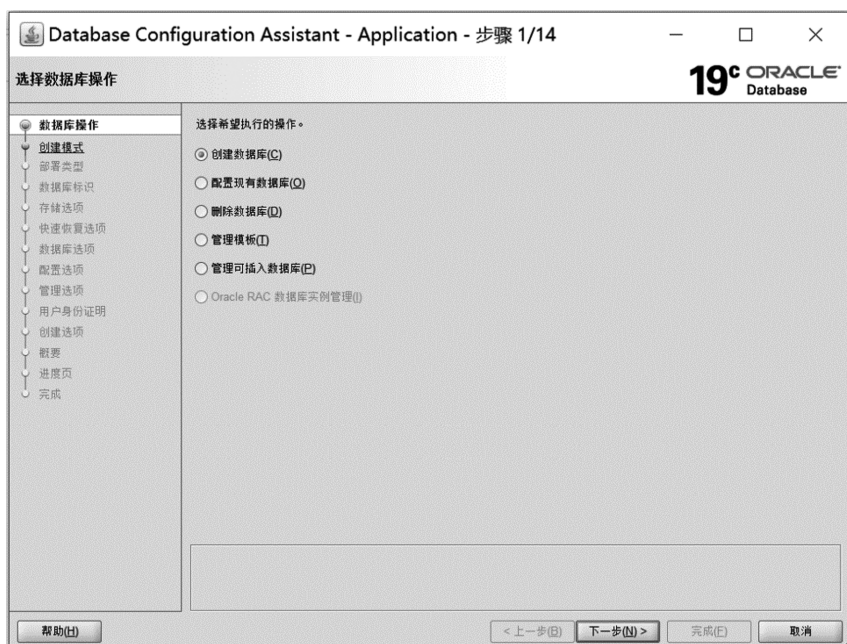


图 3.5 “选择数据库操作”窗口

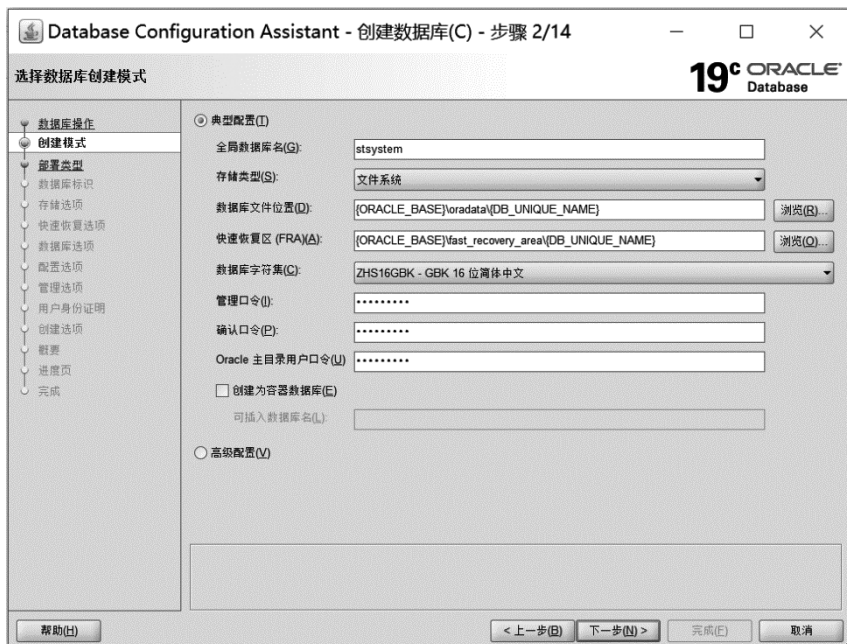


图 3.6 “选择数据库创建模式”窗口



图 3.7 “概要”窗口

3.2 表 空 间

下面介绍在 SQL * Plus 中使用 PL/SQL 语句对表空间进行创建、管理和删除。

3.2.1 创建表空间

创建表空间在 SQL * Plus 中使用 CREATE TABLESPACE 语句,创建的用户必须拥有 CREATE TABLESPACE 系统权限,在创建之前必须创建包含表空间的数据库。

语法格式：

```
CREATE TABLESPACE tablespace_name
DATAFILE filename SIZE size
[AUTOEXTEND[ON/OFF]]NEXT size
[MAXSIZE size]
[PERMANENT|TEMPORARY]
[EXTENT MANAGEMENT
[DICTIONARY|LOCAL
[AUTOALLOCATE|UNIFORM.[SIZE integer[K|M]]]]]
```

说明：

- tablespace_name: 创建的表空间的名称。
- DATAFILE filename SIZE size: 在表空间中存放数据文件的名称和大小。
- [AUTOEXTEND[ON/OFF]]NEXT size: 指定数据文件的扩展方式,ON 为自动

扩展,OFF 为非自动扩展,NEXT size 指定自动扩展的大小。

- [MAXSIZE size]: 指定数据文件为自动扩展方式的最大值。
- [PERMANENT|TEMPORARY]: 指定表空间的类型,PERMANENT 为永久性表空间,TEMPORARY 为临时性表空间。如果不指定表空间的类型,则默认为永久性表空间。
- EXTENT MANAGEMENT DICTIONARY|LOCAL: 指定表空间的管理方式,DICTIONARY 为字典管理方式,LOCAL 为本地管理方式。默认管理方式为本地管理方式。

启动 SQL * Plus,在提示符 SQL>后,即可输入以下例题代码,按 Enter 键执行后,在 SQL * Plus 窗口中将会出现运行结果。

【例 3.3】 创建表空间 testspace1、testspace2、testspace3,各个表空间大小为 50MB,禁止自动扩展数据文件。

代码和运行结果如下:

```
SQL>CREATE TABLESPACE testspace1
  2  DATAFILE 'D:\app\testspace1_1.DBF' SIZE 50M
  3  AUTOEXTEND OFF;
```

表空间已创建。

```
SQL>CREATE TABLESPACE testspace2
  2  DATAFILE 'D:\app\testspace2_1.DBF' SIZE 50M
  3  AUTOEXTEND OFF;
```

表空间已创建。

```
SQL>CREATE TABLESPACE testspace3
  2  DATAFILE 'D:\app\testspace3_1.DBF' SIZE 50M
  3  AUTOEXTEND OFF;
```

表空间已创建。

【例 3.4】 创建表空间 testspace4、testspace5、testspace6,各个表空间大小为 50MB,允许自动扩展数据文件。

代码和运行结果如下:

```
SQL>CREATE TABLESPACE testspace4
  2  DATAFILE 'D:\app\testspace4_1.DBF' SIZE 50M
  3  AUTOEXTEND ON NEXT 5M MAXSIZE 100M
  4  EXTENT MANAGEMENT LOCAL
  5  UNIFORM SIZE 2M;
```

表空间已创建。

```
SQL>CREATE TABLESPACE testspace5
  2  DATAFILE 'D:\app\testspace5_1.DBF' SIZE 50M
  3  AUTOEXTEND ON NEXT 5M MAXSIZE 100M
  4  EXTENT MANAGEMENT LOCAL
```

```
5 UNIFORM SIZE 2M;
```

表空间已创建。

```
SQL>CREATE TABLESPACE testspace6
```

```
2 DATAFILE 'D:\app\testspace6_1.DBF' SIZE 50M
```

```
3 AUTOEXTEND ON NEXT 5M MAXSIZE 100M
```

```
4 EXTENT MANAGEMENT LOCAL
```

```
5 UNIFORM SIZE 2M;
```

表空间已创建。

3.2.2 表空间的基本管理

表空间的基本管理包括更改表空间的名称和删除表空间。

1. 更改表空间的名称

在 SQL * Plus 中使用 ALTER TABLESPACE 语句更改表空间的名称。

语法格式：

```
ALTER TABLESPACE oldname RENAME TO newname
```

【例 3.5】 将表空间 testspace6 更名为 newspace6。

代码和运行结果如下：

```
SQL>ALTER TABLESPACE testspace6 RENAME TO newspace6;
```

表空间已更改。

2. 删除表空间

在 SQL * Plus 中使用 DROP TABLESPACE 语句删除已经创建的表空间。

语法格式：

```
DROP TABLESPACE <表空间名>
```

```
[ INCLUDING CONTENTS [ {AND | KEEP} DATAFILES ]
```

```
[ CASCADE CONSTRAINTS ]
```

```
];
```

【例 3.6】 删除表空间 newspace6 及其对应的数据文件。

代码和运行结果如下：

```
SQL>DROP TABLESPACE newspace6
```

```
2 INCLUDING CONTENTS AND DATAFILES;
```

表空间已删除。

3.3 小 结

本章主要介绍了以下内容。

(1) 使用图形界面方式创建和删除数据库。

(2) 在 SQL * Plus 中,使用 CREATE TABLESPACE 语句创建表空间,使用 ALTER TABLESPACE 语句更改表空间的名称,使用 DROP TABLESPACE 语句删除表空间。

习 题 3

一、选择题

1. 使用图形界面方式创建数据库的工具是_____。
A. Oracle Net Manager B. Database Configuration Assistant
C. SQL * Plus D. Oracle SQL Developer
2. 当数据库创建时,哪个表空间会自动生成_____。
A. SYSTEM 表空间 B. TEMP 表空间
C. USERS 表空间 D. TOOLS 表空间

二、填空题

1. 创建表空间使用_____语句。
2. 删除表空间使用_____语句。

三、问答题

1. 简述使用图形界面创建数据库的步骤。
2. 简述使用图形界面删除数据库的步骤。

四、应用题

1. 创建表空间 tbspace,大小为 35MB,禁止自动扩展数据文件。
2. 创建表空间 tbspace1,允许自动扩展数据文件。
3. 将表空间 tbspace1 更名为 tbspace2。
4. 删除表空间 tbspace2 及其对应的数据文件。

本章要点

- 表的基本概念
- 数据类型
- 表结构设计
- 使用图形界面方式定义表
- 使用 PL/SQL 语句定义表
- 使用 PL/SQL 语句操作表数据
- 使用图形界面方式操作表数据
- 分区表

数据库对象是构成数据库的组成元素,在 Oracle 数据库中,有多种类型的数据库对象,如表、视图、索引、存储过程、触发器等。表是数据库中最基本的数据库对象,用来存储数据库中的数据。

4.1 表的基本概念

在创建数据库的过程中,最重要的一步就是创建表,在工作和生活中,表是经常使用的一种表示数据及其关系的形式,在学生信息系统中,学生表如表 4.1 所示。

表 4.1 学生表(student)

学 号	姓 名	性 别	出 生 日 期	专 业	总 学 分
221001	何德明	男	2001-07-16	计算机	52
221002	王丽	女	2002-09-21	计算机	50
221004	田桂芳	女	2021-12-05	计算机	52
224001	周思远	男	2001-03-18	通信	52
224002	许月琴	女	2002-06-23	通信	48
224003	孙俊松	男	2001-10-07	通信	50

Oracle 数据库中的表包含以下几个基本概念。

(1) 表。表是数据库中存储数据的数据库对象,每个数据库包含了若干表,表由行和列组成。例如,表 4.1 的内容由 6 行 6 列组成。

(2) 表结构。每个表具有一定的结构,表结构包含一组固定的列,列由数据类型、长度、允许 Null 值等组成。

(3) 记录。每个表包含若干行数据,表中一行称为一个记录(Record)。表 4.1 中有 6 个记录。

(4) 字段。表中每列称为字段(Field),每个记录由若干数据项(列)构成,构成记录的每个数据项就称为字段。表 4.1 中有 6 个字段。

(5) 空值。空值(Null)通常表示未知、不可用或将在以后添加的数据。

(6) 关键字。关键字用于唯一标识记录,如果表中记录的某一字段或字段组合能唯一标识记录,则该字段或字段组合称为候选关键字(Candidate Key)。如果一个表有多个候选关键字,则选定其中的一个为主关键字(Primary Key),又称为主键。表 4.1 中的主键为“学号”。

4.2 数据类型

Oracle 常用的数据类型有数值型、字符型、日期型、其他数据类型等,下面分别进行介绍。

1. 数值型

常用的数值型有 number、float 两种,其格式和取值范围如表 4.2 所示。

表 4.2 数值型

数据类型	格 式	说 明
number	NUMBER[($\langle$ 总位数 $\rangle$ [, $\langle$ 小数点右边的位数 $\rangle$)]	可变长度数值列,允许值为 0、正数和负数,总位数默认为 38,小数点右边的位数默认为 0
float	FLOAT[($\langle$ 数值位数 $\rangle$)]	浮点型数值列

2. 字符型

常用的字符型有 char、nchar、varchar2、nvarchar2、long 这 5 种,它们在数据库中以 ASCII 码的格式存储,其取值范围和作用如表 4.3 所示。

表 4.3 字符型

数据类型	格 式	说 明
char	CHAR[($\langle$ 长度 $\rangle$ [BYTE CHAR])]]	固定长度字符域,最大长度为 2000B
nchar	NCHAR[($\langle$ 长度 $\rangle$)]	多字节字符集的固定长度字符域,最多为 2000 个字符或 2000B
varchar2	VARCHAR2[($\langle$ 长度 $\rangle$ [BYTE CHAR])]]	可变长度字符域,最大长度为 4000B
nvarchar2	NVARCHAR2[($\langle$ 长度 $\rangle$)]	多字节字符集的可变长度字符域,最多为 4000 个字符或 4000B
long	LONG	可变长度字符域,最大长度为 2GB

3. 日期型

日期型常用的有 date 和 timestamp 两种,用来存放日期和时间,取值范围和作用如表 4.4 所示。

表 4.4 日期型

数据类型	格 式	说 明
date	DATE	存储全部日期和时间的固定长度字符域,长度为 7B,查询时日期默认格式为 DD-MON-RR,除非通过设置 NLS_DATE_FORMAT 参数取代默认格式
timestamp	TIMESTAMP [(<位数>)]	用亚秒的粒度存储一个日期和时间,参数是亚秒粒度的位数,默认为 6,范围为 0~9

4. 其他数据类型

除上述类型外,Oracle 11g 还提供存放大数据的数据类型和二进制文件的数据类型 blob、clob、bfile,如表 4.5 所示。

表 4.5 其他数据类型

数据类型	格 式	说 明
blob	BLOB	二进制大对象,最大为 4GB
clob	CLOB	字符大对象,最大为 4GB
bfile	BFILE	外部二进制文件,大小由操作系统决定

4.3 表结构设计

在数据库设计过程中,最重要的是表结构设计,好的表结构设计,对应着较高的效率和安全性,而差的表结构设计,对应着较低的效率和安全性。

创建表的核心是定义表结构及设置表和列的属性,创建表以前,首先要确定表名和表的属性,表所包含的列名及列的数据类型、长度、非空、是否主键等,这些属性构成表结构。

例如,学生表 student 包含 sid、sname、ssex、sbirthday、speciality、tc 等列。

- sid 列是学生的学号,例如,221001 中的 22 表示 2022 年入学,01 表示学生的序号,所以 sid 列的数据类型选字符型 char[(n)],n 的值为 6,不允许空。
- sname 列是学生的姓名,姓名一般不超过 4 个中文字符,所以选字符型 char[(n)],n 的值为 12,不允许空。
- ssex 列是学生的性别,选字符型 char[(n)],n 的值为 3,不允许空。
- sbirthday 列是学生的出生日期,选 date 数据类型,不允许空。
- speciality 列是学生的专业,选字符型 char[(n)],n 的值为 12,允许空。
- tc 列是学生的总学分,选 number 数据类型,允许空。

在 student 表中,只有 sid 列能唯一标识一个学生,所以将 sid 列设为主键。student 的

表结构设计如表 4.6 所示。

表 4.6 student 的表结构

列 名	数据类型	非 空	是 否 主 键	说 明
sid	char(6)	√	主键	学号
sname	char(12)	√		姓名
ssex	char(3)	√		性别
sbirthday	date	√		出生日期
speciality	char(12)			专业
tc	number			总学分

4.4 使用图形界面方式定义表

定义表包括创建表、修改表、删除表等内容,本节介绍使用图形界面方式创建表、修改表、删除表,4.5 节介绍使用 PL/SQL 语句创建表、修改表、删除表。

4.4.1 使用图形界面方式创建表

使用图形界面方式创建表的举例如下。

【例 4.1】 在 stsystem 数据库中创建 student 表。

创建 student 表的操作步骤如下。

(1) 启动 SQL Developer,在“连接”节点下打开数据库连接 st_test,右击“表”节点,在弹出的快捷菜单中选择“新建表”命令。

(2) 屏幕出现“创建表”对话框,在“名称”文本框中输入表名 STUDENT,根据已经设计好的 student 的表结构分别输入或选择 SID、SNAME、SSEX、SBIRTHDAY、SPECIALITY、TC 等列的 PK(是否主键)、列名、数据类型、长度大小、非空性等栏信息,输入完一列后单击“+”按钮添加下一列,输入完成后的结果如图 4.1 所示。

(3) 输完最后一列的信息后,选中右上角的“高级”复选框,此时会显示出更多的表的选项,如表的类型、列的默认值、约束条件和存储选项等,如图 4.2 所示。

(4) 单击“确定”按钮,创建 student 表完成。

4.4.2 使用图形界面方式修改表

使用图形界面方式修改表结构(如增加列、删除列、修改已有列的属性等)的举例如下。

【例 4.2】 在 student 表中增加一列 telephone,然后删除该列。

操作步骤如下。

(1) 启动 SQL Developer,在“连接”节点下打开数据库连接 st_test,展开“表”节点,选中表 student 后右击,在弹出的快捷菜单中选择“编辑”命令。

(2) 进入“编辑表”对话框,单击“+”按钮,在“列属性”栏的“名称”框中输入 TELEPHONE,在“类型”框中选择 VARCHAR2,在“大小”框中输入 11,如图 4.3 所示,单击“确定”按钮,完成插入新列 TELEPHONE 的操作。

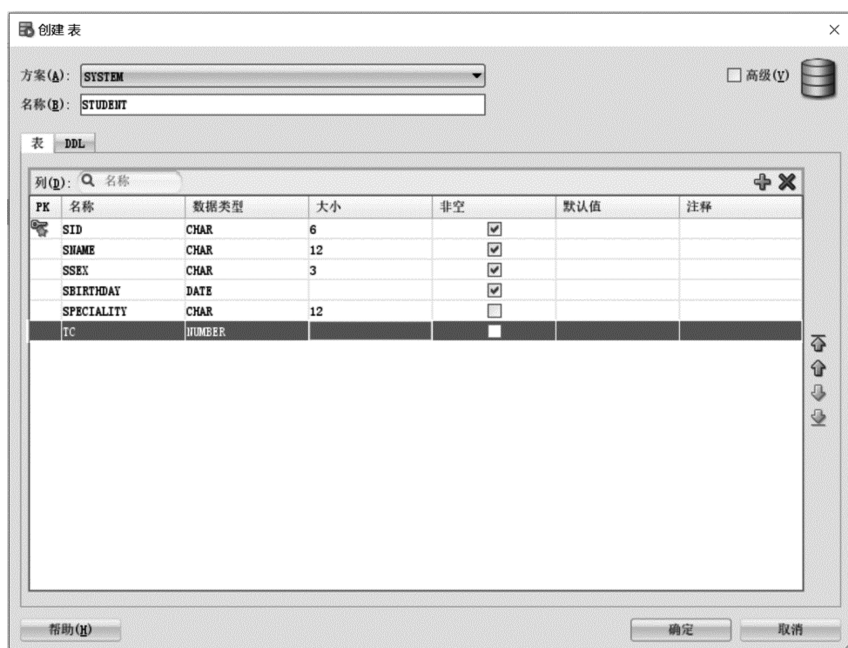


图 4.1 “创建表”对话框

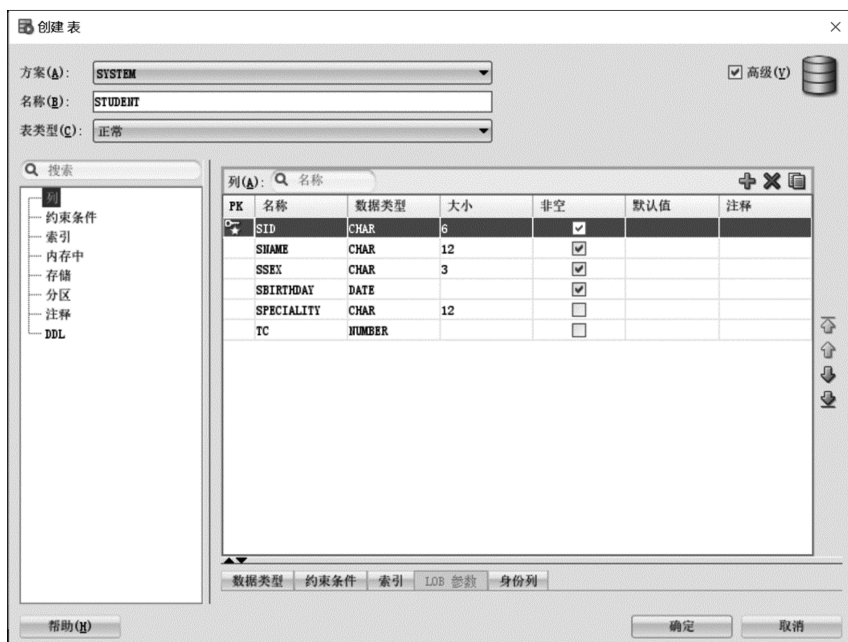


图 4.2 选中“高级”复选框

(3) 选中表 student 后右击,在弹出的快捷菜单中选择“编辑”命令,进入“编辑表”对话框,在“列”栏中选中 TELEPHONE,如图 4.4 所示,单击  按钮,TELEPHONE 列即被删

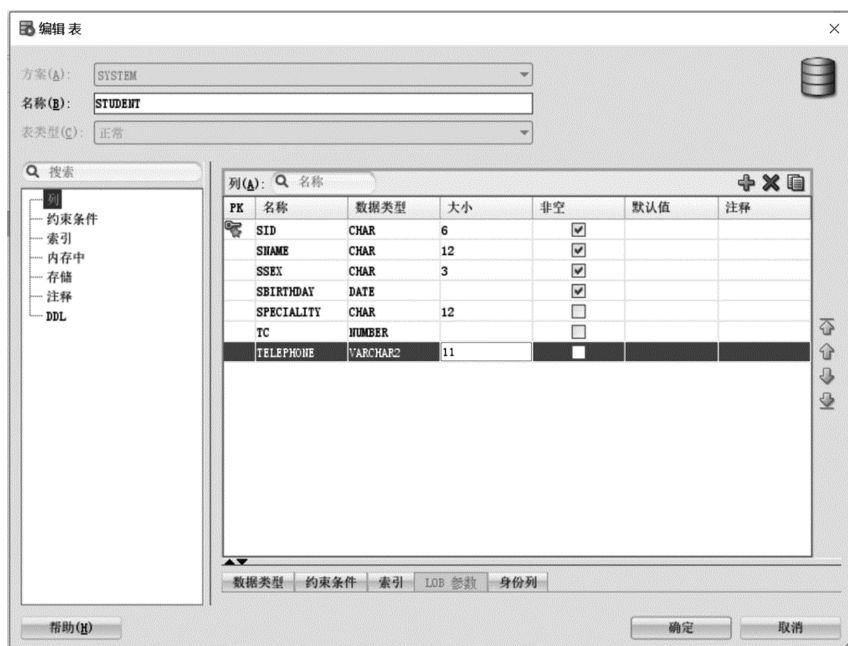


图 4.3 插入列操作

除,单击“确定”按钮,完成删除 TELEPHONE 列的操作。

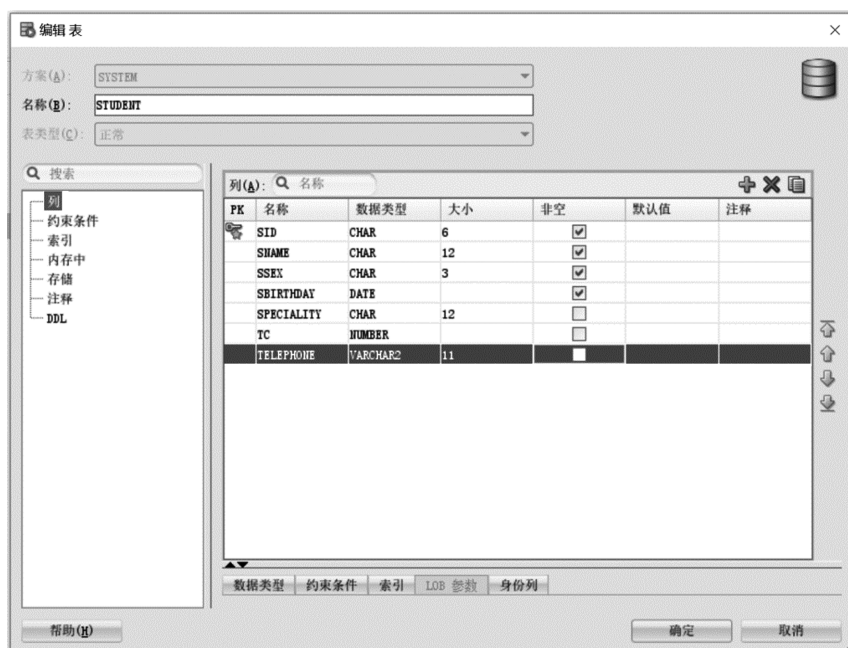


图 4.4 删除列操作

4.4.3 使用图形界面方式删除表

当表不需要的时候,可将其删除。删除表时,表的结构定义、表中的所有数据及表的索引、触发器、约束等都被删除掉。下面介绍使用图形界面方式删除表。

【例 4.3】 删除 student 表。

(1) 启动 SQL Developer,在“连接”节点下打开数据库连接 st_test,展开“表”节点,选中表 student 并右击,在弹出的快捷菜单中选择“表”→“删除”命令。

(2) 进入“删除”对话框,单击“应用”按钮,将弹出“确认”对话框,单击“确定”按钮,即可删除 student 表。

4.5 使用 PL/SQL 语句定义表

本节介绍使用 PL/SQL 语句创建表、修改表、删除表。

4.5.1 使用 PL/SQL 语句创建表

使用 CREATE TABLE 语句创建表。

语法格式:

```
CREATE TABLE [<用户方案名>.] <表名>
(
    <列名 1> <数据类型> [DEFAULT <默认值>] [<列约束>]
    <列名 2> <数据类型> [DEFAULT <默认值>] [<列约束>]
    [, ...n]
    <表约束> [, ...n]
)
[PCTFREE <数字值>]
[PCTUSED <数字值>]
[INITRANS <数字值>]
[MAXTRANS <最大并发事务数>]
[TABLESPACE <表空间名>]
[STORGE <参数>]
[AS <子查询>]
```

【例 4.4】 使用 PL/SQL 语句,在 stsystem 数据库中创建 student 表。

```
CREATE TABLE student
(
    sid char(6) NOT NULL PRIMARY KEY,
    sname char(12) NOT NULL,
    ssex char(3) NOT NULL,
    sbirthday date NOT NULL,
    speciality char(12) NULL,
    tc number NULL
```



);

启动 SQL Developer,在主界面中展开 st_test 连接,单击工具栏中的  按钮,主界面弹出“工作表”窗口,在窗口中输入上述语句,单击  按钮,在“脚本输出”窗口显示“Table STUDENT 已创建。”,如图 4.5 所示。

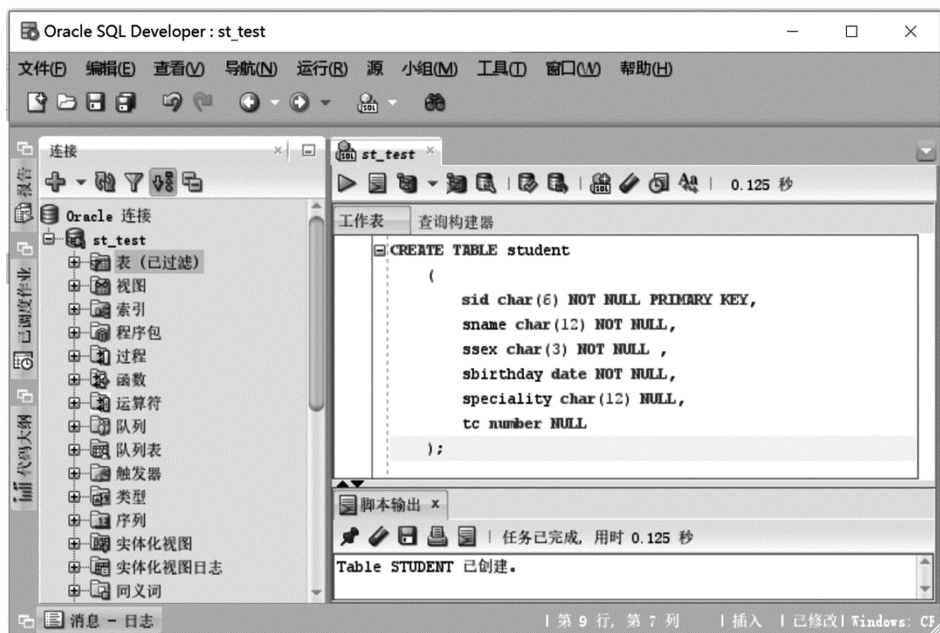


图 4.5 创建 student 表的“工作表”窗口和“脚本输出”窗口

提示:

由一条或多条 PL/SQL 语句组成一个程序,通常以 .sql 为扩展名存储,称为 sql 脚本。SQL 工作表窗口内的 PL/SQL 语句,可通过“文件”→“保存”命令命名并存入指定目录。

4.5.2 使用 PL/SQL 语句修改表

使用 ALTER TABLE 语句来修改表的结构。

语法格式:

ALTER TABLE [<用户方案名>.] <表名>

[ADD(<新列名><数据类型>[DEFAULT <默认值>][列约束],...n)] /* 增加新列 */

[MODIFY([<列名>[<数据类型>][DEFAULT <默认值>][列约束],...n)] /* 修改已有列的属性 */

[STORAGE <存储参数>] /* 修改存储特征 */

[<DROP 子句>] /* 删除列或约束条件 */

其中,<DROP 子句>用于从表中删除列或约束。

语法格式:

<DROP 子句>::=

```
DROP
{
    COLUMN <列名>
    | PRIMARY [KEY]
    | UNIQUE (<列名>, ...n)
    | CONSTRAINT <约束名>
    | [ CASCADE ]
}
```

【例 4.5】 使用 ALTER TABLE 语句修改 stsystem 数据库中的 student 表。

(1) 在 student 表中增加一列 native。

```
ALTER TABLE student
    ADD native varchar(12);
```

(2) 在 student 表中删除 native 列。

```
ALTER TABLE student
    DROP COLUMN native;
```

4.5.3 使用 PL/SQL 语句删除表

使用 DROP TABLE 语句删除表。

语法格式：

```
DROP TABLE table_name
```

其中, table_name 是要删除的表的名称。

【例 4.6】 删除 stsystem 数据库中的 student 表。

```
DROP TABLE student;
```

4.6 使用 PL/SQL 语句操作表数据



对表中数据进行的操作,包括数据的插入、删除和修改。本节介绍使用 PL/SQL 语句操作表数据,4.7 节介绍采用图形界面方式操作表数据。

4.6.1 使用 PL/SQL 语句插入记录

在数据库中可用 INSERT 语句向数据库的表插入一行数据,由 VALUES 给定该行各列的值。

语法格式：

```
INSERT INTO <表名> [(<列名 1>, <列名 2>, ...n)]
    VALUES (<列值 1>, <列值 2>, ...n)
```

其中,列值表必须与列名表一一对应,且数据类型相同。向表的所有列添加数据时,可以省

略列名表,但列值表必须与列名表顺序和数据类型一致。

在输入 student 表的 sbirthday 列的数据时,数据库默认的日期格式是: DD-MON-RR, 为了将日期格式改为习惯的格式: YYYY-MM-DD,需要在 SQL Developer 命令窗口中执行以下语句:

```
ALTER SESSION
SET NLS_DATE_FORMAT='YYYY-MM-DD';
```

说明: 该语句只在当前会话中起作用,下次打开 SQL Developer 窗口,还需重新执行该语句。

注意: 使用 PL/SQL 语句进行数据的插入、修改和删除后,为将数据的改变保存到数据库中,应使用 COMMIT 命令进行提交,使用方法如下:

```
COMMIT;
```

本书后面的 SQL 语句都省略 COMMIT 命令,运行时请读者添加。

1. 对表的所有列插入数据

1) 省略列名表

必须为每个列都插入数据,值的顺序必须与表定义的列的顺序一一对应,且数据类型相同。

设 student 表、student1 表已创建,其表结构参见附录 B。

【例 4.7】 使用省略列名表的插入语句,对 student1 表插入一条记录('221001','何德明','男','20010716','YYYYMMDD'),'计算机',52)。

代码如下:

```
INSERT INTO student1
VALUES('221001','何德明','男',TO_DATE('20010716','YYYYMMDD'),'计算机',52);
```

由于插入的数据包含各列的值并按表中各列的顺序列出这些值,所以省略列名表。

2) 不省略列名表

如果插入值的顺序和表定义的列的顺序不同,在插入全部列时,则不能省略列名表。

【例 4.8】 使用不省略列名表的插入语句,对 student1 表插入一条记录,姓名为“王丽”,专业为“计算机”,学号为“221002”,性别为“女”,出生日期为“20020921”,总学分为 50。

代码如下:

```
INSERT INTO student1(sname, speciality, sid, ssex, sbirthday, tc)
VALUES('王丽','计算机','221002','女',TO_DATE('20020921','YYYYMMDD'),50);
```

2. 对表的指定列插入数据

在插入语句中,只给出了部分列的值,其他列的值为表定义时的默认值,或允许该列取空值。此时,不能省略列名表。

【例 4.9】 只给出部分列的值,对 student1 表插入一条记录,学号为“224004”,专业为“通信”,学分空值,姓名为“颜群”,性别为“女”,出生日期为“20020112”。

代码如下:

```
INSERT INTO student1(sid, speciality, sname, ssex, sbirthday)
VALUES('224004','通信','颜群','女', TO_DATE('20020112','YYYYMMDD'));
```

3. 插入多条记录

【例 4.10】 向 student 表插入表 4.1 各行的数据。

向 student 表插入表 4.1 各行数据的代码如下：

```
INSERT INTO student VALUES('221001','何德明','男',TO_DATE('20010716','YYYYMMDD'),
'计算机',52);
INSERT INTO student VALUES('221002','王丽','女',TO_DATE('20020921','YYYYMMDD'),
'计算机',50);
INSERT INTO student VALUES('221004','田桂芳','女',TO_DATE('20021205','YYYYMMDD'),
'计算机',52);
INSERT INTO student VALUES('224001','周思远','男',TO_DATE('20010318','YYYYMMDD'),
'通信',52);
INSERT INTO student VALUES('224002','许月琴','女',TO_DATE('20020623','YYYYMMDD'),
'通信',48);
INSERT INTO student VALUES('224003','孙俊松','男',TO_DATE('20011007','YYYYMMDD'),
'通信',50);
```

使用 SELECT 语句查询插入的数据：

```
SELECT *
FROM student;
```

查询结果：

SID	SNAME	SSEX	SBIRTHDAY	SPECIALITY	TC
221001	何德明	男	2001-07-16	计算机	52
221002	王丽	女	2002-09-21	计算机	50
221004	田桂芳	女	2002-12-05	计算机	52
224001	周思远	男	2001-03-18	通信	52
224002	许月琴	女	2002-06-23	通信	48
224003	孙俊松	男	2001-10-07	通信	50

4.6.2 使用 PL/SQL 语句修改记录

在数据库中可用 UPDATE 语句来修改表中指定记录的列值。

语法格式：

```
UPDATE <表名>
SET <列名>=<新值>|<表达式> [,...n]
[WHERE <条件表达式>]
```

其中,在满足 WHERE 子句条件的行中,将 SET 子句指定的各列的列值设置为 SET 指定的新值,如果省略 WHERE 子句,则更新所有行的指定列值。

注意：UPDATE 语句修改的是一行或多行中的列。

1. 修改指定记录

修改指定记录需要通过 WHERE 子句指定要修改的记录满足的条件。

【例 4.11】 在 course 表中,将数据结构的学分修改为 4 分。

代码如下：

```
UPDATE course
  SET credit=4
  WHERE cname='数据结构';
```

2. 修改全部记录

修改全部记录可不指定 WHERE 子句。

【例 4.12】 在 student 表中,将所有学生的学分增加 2 分。

代码如下：

```
UPDATE student
  SET tc=tc+2;
```

4.6.3 使用 PL/SQL 语句删除记录

在数据库中删除记录,可以使用 DELETE 语句或 TRUNCATE 语句。

1. 使用 DELETE 语句删除记录

在进行删除操作时,DELETE 语句用于删除表中的一行或多行记录。

语法格式：

```
DELETE FROM <表名>
[WHERE <条件表达式>]
```

该语句的功能为从指定的表中删除满足 WHERE 子句的条件行,若省略 WHERE 子句,则删除所有行。

注意：DELETE 语句删除的是一行或多行。如果删除所有行,表结构仍然存在,即存在一个空表。

【例 4.13】 在 student1 表中,删除学号为 224004 的行。

代码如下：

```
DELETE FROM student1
  WHERE sid='224004';
```

【例 4.14】 在 student1 表中,删除全部记录。

代码如下：

```
DELETE FROM student1;
```

2. 使用 TRUNCATE 语句删除全部记录

当需要删除一个表里的全部记录时,使用 TRUNCATE TABLE 语句,它可以释放表的