

第7章 数据管理——常用数据方案对象

本章通过实例的方法详细介绍 Oracle 9i 数据库中常用的数据方案对象的使用和管理方法。

7.1 Oracle 9i 的方案对象

7.1.1 数据方案对象

表 7.1 Oracle 9i 的数据方案对象

名称	含义
数据表	表是用于存放用户数据的数据库对象。数据库中的表按照功能的不同可以分为系统表和用户表两类。系统表用于存储管理用户数据和数据库本身的数据，又称数据字典表；用户表用于存放用户的数据信息，默认建立的数据表就是用户表
索引	索引就犹如一本书的目录，利用它可以快速找到所需要的内容。索引总是和数据表紧密相关联的
视图	视图是查看数据表中数据的一种方法，使用视图的主要目的就是确保数据表的安全性和隐蔽数据的复杂性。视图不是数据表，仅仅是一些 SQL 查询语句的定义

7.1.2 管理方案对象

表 7.2 Oracle 9i 的管理方案对象

名称	含义
数据库链接	管理分布式网络数据库服务器环境的登录用户名、口令和数据库位置
同义词	同义词用于隐藏表的用户名和数据表名，提供安全性
序列	直接产生唯一的顺序序号的一种方案对象
实体化视图	包含了对一个或多个数据表的查询结果的数据表，又称快照
实体化视图日志	记录上一次快照刷新后对数据表所做的所有修改信息的数据表，又称快照日志
刷新组	设置对数据库用户下的所有对象什么时间、按照什么条件进行刷新
簇	将一些互相关联，具有相同字段的数据表集中存储的一种管理结构
表类型	表类型也可以称为嵌套表类型，在表的定义里可以包含嵌套表

7.1.3 PL/SQL 编程方案对象

表 7.3 Oracle 9i 的 PL/SQL 编程方案对象

名称	含义
过程	过程也叫做存储过程，是由 SQL 语句和 PL/SQL 语句组合在一起为执行某一个任务的一个可执行单位，类似于高级程序设计语言中的模块
函数	函数和过程的结构类似。过程和函数差别在于，函数总返回单个值给调用者，而过程没有值返回给调用者
程序包	程序包也称为包，是被集中到一个单独的单元的一组过程、函数、变量和 SQL 语句的定义
程序包体	程序包体也称为包体，是与对应的程序包同名的，关于程序包内声明的函数、过程的详细执行代码
触发器	触发器是一种特殊类型的存储过程，由一些 SQL 语句组成，主要用于执行强制性的业务规则或要求，但不返回结果
对象类型	对象是面向对象分析与设计的基本概念，具有相同的属性和服务的对象被称为类，在 Oracle 9i 中把类称为对象类型，在一些数据库书籍里也称为抽象数据类型
数组类型	在 Oracle 8 中被称为可变数组。提供了自定义数组功能，包括数组元素的个数、元素的类型、长度和精度等。
高级队列	一种数据的存储结构，特点是先进先出
Java 源	一些 Java 源代码，这些源代码可以作为 Java 共享过程相互调用
维	对数据仓库中的数据进行分类的逻辑结构

7.1.4 如何查看方案对象

如图 7.1 所示界面。列举了 Oracle 9i 中的 21 种方案对象。



图 7.1 Oracle 9i 的方案对象

7.2 Oracle 9i 的数据类型

Oracle 9i 共提供了 16 种标量数据类型，如表 7.4 所示。

表 7.4 Oracle 9i 的标量数据类型

名称	含义
Char	用于描述定长的字符型数据，长度≤2000 字节
varchar2	用于描述变长的字符型数据，长度≤4000 字节
nchar	用来存储 Unicode 字符集的定长字符型数据，长度≤1000 字节
nvarchar2	用来存储 Unicode 字符集的变长字符型数据，长度≤1000 字节
number	用来存储整型或者浮点型数值
Date	用来存储日期数据
Long	用来存储最大长度为 2GB 的变长字符数据
Raw	用来存储非结构化数据的变长字符数据，长度≤2000 字节
Long raw	用来存储非结构化数据的变长字符数据，长度≤2GB
rowid	用来存储表中列的物理地址的二进制数据，占用固定的 10 个字节
Blob	用来存储多达 4GB 的非结构化的二进制数据
Clob	用来存储多达 4GB 的字符数据
nclob	用来存储多达 4GB 的 Unicode 字符数据
Bfile	用来把非结构化的二进制数据存储在数据库以外的操作系统文件中
urowid	用来存储表示任何类型列地址的二进制数据
float	用来存储浮点数

7.3 Oracle 9i 的数据表类型

表 7.5 Oracle 9i 的数据表类型

方式	特点
关系表	默认的表类型，存储永久性的数据，可以被分区，这样可以改善表的性能并易于管理
临时表	存储私有数据或一个会话中特定的数据，数据库中的其他用户不能使用这些数据
索引表	按照结构化主关键字进行排序的方式存储数据，和关系表不同的是不能把表和主关键字分开存储
外部表	数据存储在 Oracle 数据库外部的文件中，只能读，因此任何索引都不能存储在外部表中
对象表	支持面向对象的数据表

7.4 创建数据表

7.4.1 要创建的两个数据表

1. 研究生信息表

表名：scott.student。

表结构如表 7.6 所示。

表 7.6 scott.student 数据表结构

名称	数据类型	大小	小数位数	说明
STUDENT_ID	NUMBER	8	0	学生编号（主码）
NAME	VARCHAR2	10		姓名
PROFESSIONAL	VARCHAR2	20		专业
BIRTHDAY	DATE			生日
DIRECTOR_ID	NUMBER	6	0	导师编号（外码）

2. 导师信息表

表名：scott.director。

表结构如表 7.7 所示。

表 7.7 scott.director 数据表结构

名称	数据类型	大小	小数位数	说明
DIRECTOR_ID	NUMBER	6	0	导师编号（主码）
NAME	VARCHAR2	10		姓名
ZHICHENG	VARCHAR2	20		职称
ZHIWU	VARCHAR2	20		职务

3. 两个数据表的关系

导师信息表（scott.director）的主码“DIRECTOR_ID”是研究生信息表（scott.student）的外码。也就是说，当导师信息表的“DIRECTOR_ID”字段发生变化时，研究生信息表的字段“DIRECTOR_ID”也会随之自动发生变化。

7.4.2 创建的步骤

1. 创建导师信息表的步骤

（1）如图 7.2 所示界面。

（2）出现如图 7.3 所示的创建表的【一般信息】选项卡。

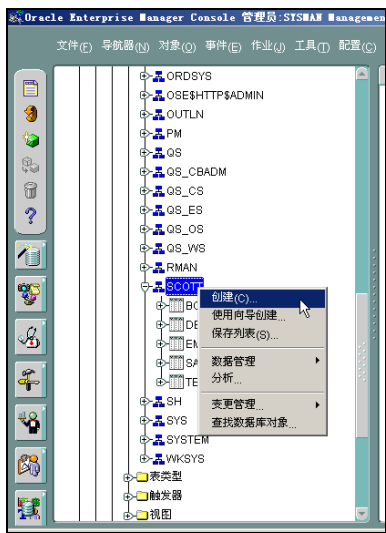


图 7.2 选择创建表

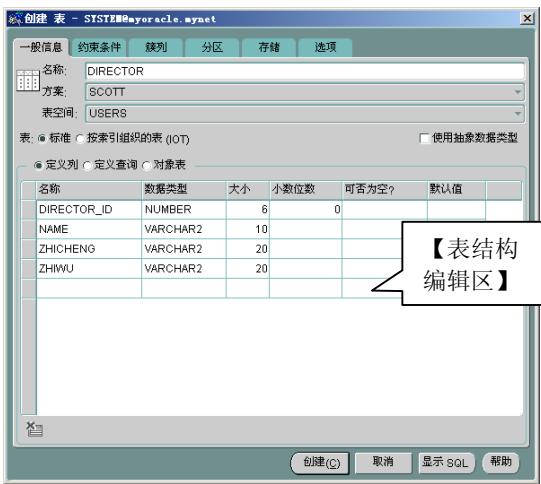


图 7.3 创建表的【一般信息】选项卡

(3) 切换到图 7.4 所示的创建表的【约束条件】选项卡。

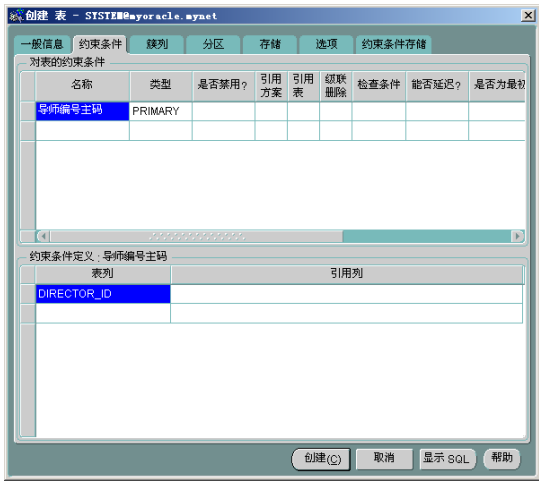


图 7.4 创建表的【约束条件】选项卡

在【类型】单元格下拉列表框中有 5 个选项，如表 7.8 所示。

表 7.8 约束条件类型

名称	含义
PRIMARY	主关键字，即主码
FOREIGN	外关键字，即外码
NOT NULL	非空，字段值不能为空值
UNIQUE	惟一，字段取值不能重复
CHECK	检查，检查是否满足指定的条件

(4) 切换到图 7.5 所示的创建表的【簇列】选项卡。

(5) 切换到图 7.6 所示的创建表的【分区】选项卡。



图 7.5 创建表的【簇列】选项卡

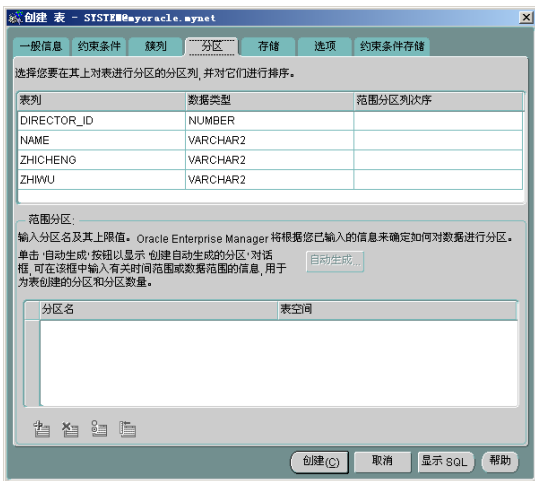


图 7.6 创建表的【分区】选项卡

(6) 切换到图 7.7 所示的创建表的【存储】选项卡。

(7) 切换到图 7.8 所示的创建表的【选项】选项卡。



图 7.7 创建表的【存储】选项卡

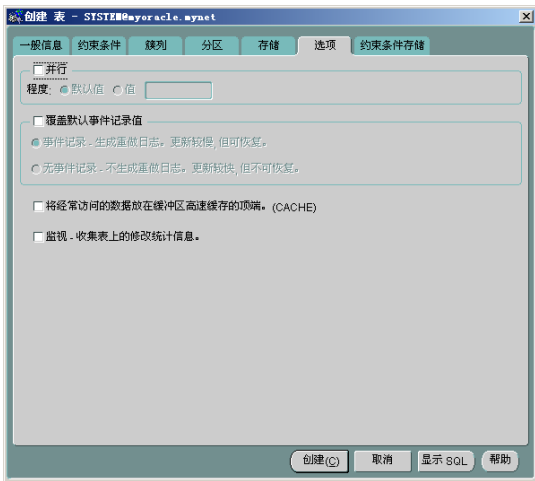


图 7.8 创建表的【选项】选项卡

(8) 切换到图 7.9 所示的创建表的【约束条件存储】选项卡。

(9) 出现如图 7.10 所示界面。



图 7.9 创建表的【约束条件存储】选项卡



图 7.10 成功创建表 scott.director

(10) 在【企业管理器】中可以查看创建的数据表，如图 7.11 所示。

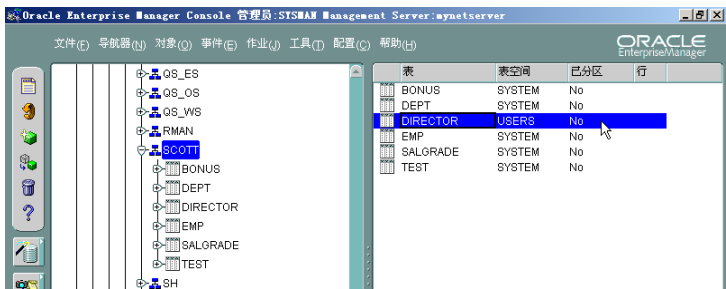


图 7.11 查询表 scott.director

(11) 上述步骤创建 scott.director 数据表的 SQL 代码如下。

```
CREATE TABLE "SCOTT"."DIRECTOR" (  
    "DIRECTOR_ID"    NUMBER(6)        NOT NULL,  
    "NAME"           VARCHAR2(10)     NOT NULL,  
    "ZHICHENG"       VARCHAR2(20)     NOT NULL,  
    "ZHIWU"          VARCHAR2(20)     NOT NULL,  
    CONSTRAINT "导师编号主码" PRIMARY KEY("DIRECTOR_ID")  
    USING INDEX  
    TABLESPACE "USERS")  
TABLESPACE "USERS"
```

【参见光盘文件】：\第 7 章\createdirector.sql。



在创建主关键字约束条件时将自动建立该字段的索引。

(12) 读者也可以在【SQL Plus Worksheet】中直接执行 createdirector.sql 文件创建数据

表 scott.director，如图 7.12 所示。

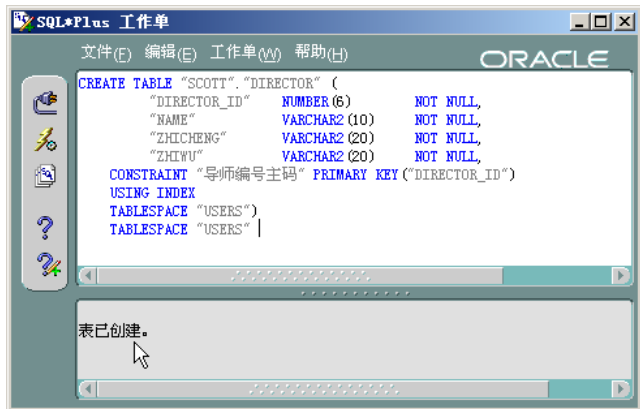


图 7.12 在【SQLPlus Worksheet】中创建数据表 scott.director

2. 创建研究生信息表的步骤

- (1) 与创建导师信息表一样进行操作。
- (2) 在图 7.13 所示的创建表的【一般信息】选项卡中按照如下步骤进行配置。
- (3) 图 7.14 所示为创建表的主码的【约束条件】选项卡。



图 7.13 创建表的【一般信息】选项卡

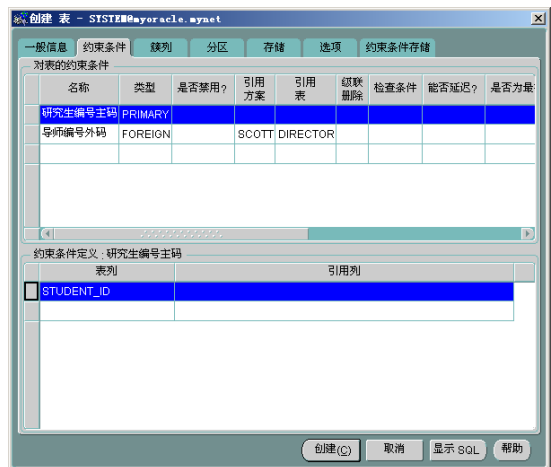


图 7.14 创建表的主码的【约束条件】选项卡

- (4) 图 7.15 所示为创建表的外码的【约束条件】选项卡。
- (5) 【簇列】、【分区】、【存储】和【选项】选项卡按照默认设置即可。
- (6) 图 7.16 所示的【约束条件存储】选项卡按照如下配置。

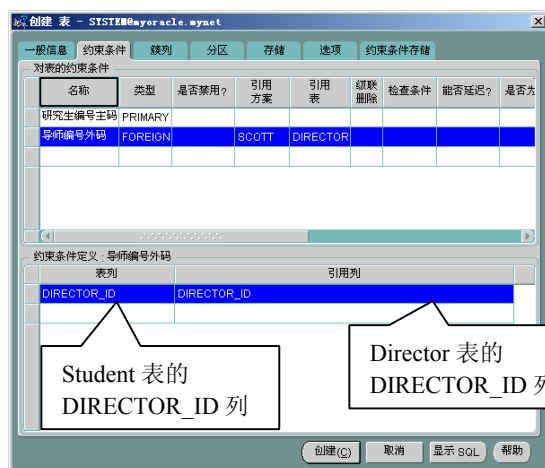


图 7.15 创建表的外码的【约束条件】选项卡



图 7.16 创建表的【约束条件存储】选项卡

(7) 系统将成功创建数据表 scott.student。对应的 SQL 代码如下。

```
CREATE TABLE "SCOTT"."STUDENT" (
    "STUDENT_ID" NUMBER(8) NOT NULL,
    "NAME" VARCHAR2(10) NOT NULL,
    "PROFESSIONAL" VARCHAR2(10) NOT NULL,
    "BIRTHDAY" DATE NOT NULL,
    "DIRECTOR_ID" NUMBER(6) NOT NULL,
    CONSTRAINT "研究生编号主码" PRIMARY KEY("STUDENT_ID")
    USING INDEX
    TABLESPACE "USERS",
    CONSTRAINT "导师编号外码" FOREIGN KEY("DIRECTOR_ID")
    REFERENCES "SCOTT"."DIRECTOR"("DIRECTOR_ID"))
    TABLESPACE "USERS"
```

【参见光盘文件】: \第 7 章
\createstudent.sql。

(8) 读者也可以在【SQL Plus Worksheet】中直接执行 createstudent.sql 文件创建数据表 scott.student, 如图 7.17 所示。

(9) 在【企业管理器】中可以查看建立的两个范例数据表, 如图 7.18 所示。

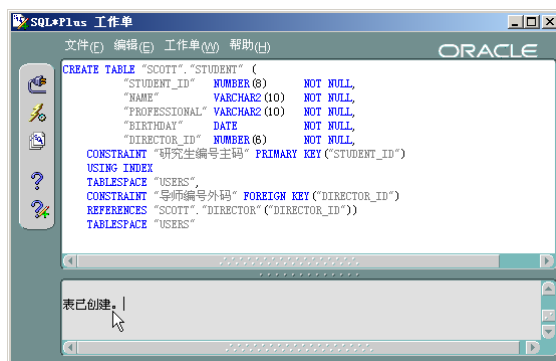


图 7.17 在【SQLPlus Worksheet】中创建数据表 scott.student

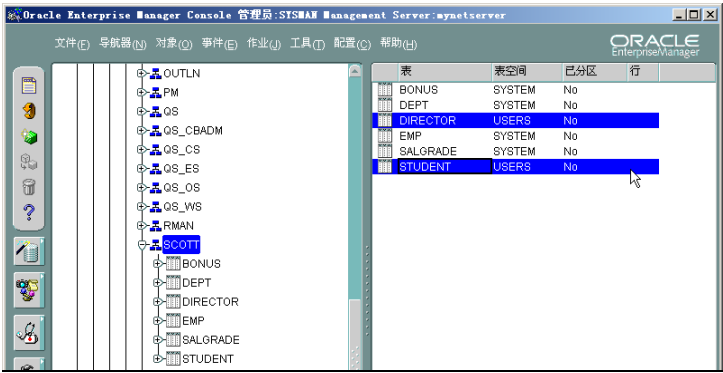


图 7.18 成功建立的两个范例数据表

7.5 修改数据表结构

(1) 如图 7.19 所示。

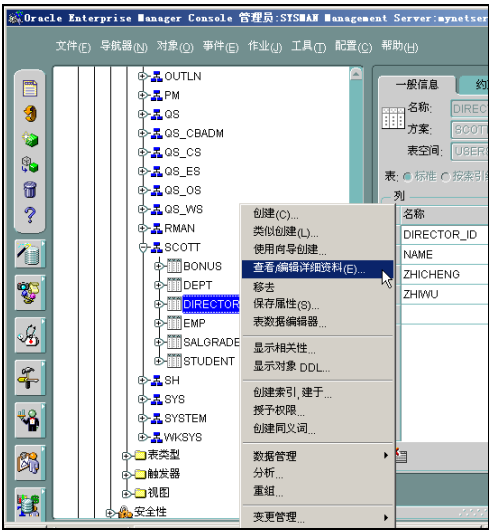


图 7.19 选择修改数据表结构



图 7.20 编辑表的【一般信息】选项卡

(2) 出现如图 7.20 所示的编辑表的【一般信息】选项卡。

(3) 其他选项卡的操作比较简单, 这里不再赘述。

7.6 数据表中数据的管理

下面介绍如何操作数据表中的数据。

7.6.1 向数据表中插入数据

1. 向导师信息表中插入数据

(1) 在图 7.19 所示的快捷菜单中选择【表数据编辑器】选项，出现如图 7.21 所示的界面。可以按照二维表格输入数据。

(2) 单击 **显示 SQL** 按钮可以查看输入的数据对应的 SQL 代码。

(3) 数据输入完毕后，单击 **应用(P)** 按钮可以完成数据的插入。

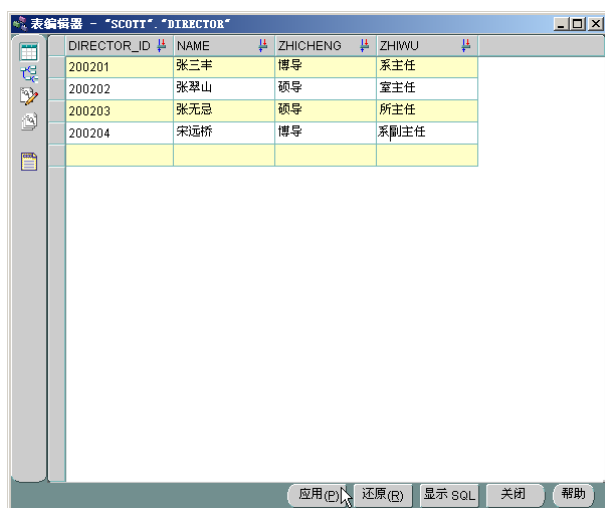


图 7.21 表数据编辑器

(4) 上述过程对应的 SQL 代码如下。

```
INSERT INTO "SCOTT"."DIRECTOR" ("DIRECTOR_ID", "NAME", "ZHICHENG", "ZHIWU" )
VALUES (200201, '张三丰', '博导', '系主任');
INSERT INTO "SCOTT"."DIRECTOR" ("DIRECTOR_ID", "NAME", "ZHICHENG", "ZHIWU" )
VALUES (200202, '张翠山', '硕导', '室主任');
INSERT INTO "SCOTT"."DIRECTOR" ("DIRECTOR_ID", "NAME", "ZHICHENG", "ZHIWU" )
VALUES (200203, '张无忌', '硕导', '所主任');
INSERT INTO "SCOTT"."DIRECTOR" ("DIRECTOR_ID", "NAME", "ZHICHENG", "ZHIWU" )
VALUES (200204, '宋远桥', '博导', '系副主任');
```

【参见光盘文件】：\第7章\insertdirector.sql。

(5) 读者也可以在【SQL Plus Worksheet】中直接执行 insertdirector.sql 文件向数据表 scott.director 中插入数据。

2. 向研究生信息表中插入数据

(1) 对研究生信息表进行同样的数据插入操作。如图 7.22 所示。

STUDENT_ID	NAME	PROFESSIONAL	BIRTHDAY	DIRECTOR_ID
20020101	周正若	软件工程		200201
20020102	赵敏	计算机安全		200202
20020103	小昭	图形图像		200203
20020104	蛛儿	电子商务		200204
20030101	金花婆婆	数据库		200201
20030102	胡青牛	网络安全		200203
20030103	丁敏君	Web技术		200201
20030104	殷素素	Web安全		200202

图 7.22 向 scott.student 表中插入的数据

(2) 字段“BIRTHDAY”是日期型字段，读者可能不知道该如何插入日期型的数据。单击 **显示 SQL** 按钮，出现如图 7.23 所示的界面。

可以观察到这样的日期型字段数据插入格式。

TO_DATE(' ', 'dd-Mon-yyyy HH:MI:SS AM')

这表明日期型字段是按照“日期-月份-一年 小时 分钟 秒”来插入的。

STUDENT_ID	NAME	PROFESSIONAL	BIRTHDAY	DIRECTOR_ID
20020101	周正若	软件工程		200201

图 7.23 分析日期型数据的插入格式

(3) 分析出格式还不一定能够正确录入日期型数据。这里告诉读者一个笔者摸索出来的简便的办法。在 scott 用户下有一个数据表 emp，其中有一个 hiredate 字段是日期型的。



图 7.24 日期型数据的范例

(4) 最后完成完整的 scott.student 数据表数据的插入，如图 7.25 所示。



图 7.25 完整的日期型数据插入实例

【参见光盘文件】：\第7章\insertstudent.sql。

(5) 上述过程对应的 SQL 代码如下。读者也可以在【SQLPlus Worksheet】中直接执行 insertstudent.sql 文件完成数据的插入。

```
INSERT INTO "SCOTT"."STUDENT"
("STUDENT_ID", "NAME", "PROFESSIONAL", "BIRTHDAY", "DIRECTOR_ID")
VALUES (20020101, '周芷若', '软件工程', TO_DATE('20-11月-1976', 'dd-Mon-yyyy HH:MI:SS AM'), 200201);

INSERT INTO "SCOTT"."STUDENT"
("STUDENT_ID", "NAME", "PROFESSIONAL", "BIRTHDAY", "DIRECTOR_ID")
VALUES (20020102, '赵敏', '计算机安全', TO_DATE('7-10月-1980', 'dd-Mon-yyyy HH:MI:SS
```

```
AM'),200202 );
INSERT INTO "SCOTT"."STUDENT"
("STUDENT_ID","NAME","PROFESSIONAL","BIRTHDAY","DIRECTOR_ID")
VALUES (20020103,'小昭','图形图像',TO_DATE('22-10月-1973','dd-Mon-yyyy HH:MI:SS
AM'),200203 );
INSERT INTO "SCOTT"."STUDENT"
("STUDENT_ID","NAME","PROFESSIONAL","BIRTHDAY","DIRECTOR_ID")
VALUES (20020104,'蛛儿','电子商务',TO_DATE('11-1月-1971','dd-Mon-yyyy HH:MI:SS
AM'),200204 );
INSERT INTO "SCOTT"."STUDENT"
("STUDENT_ID","NAME","PROFESSIONAL","BIRTHDAY","DIRECTOR_ID")
VALUES (20030101,'金花婆婆','数据库',TO_DATE('6-8月-1945','dd-Mon-yyyy HH:MI:SS
AM'),200201 );
INSERT INTO "SCOTT"."STUDENT"
("STUDENT_ID","NAME","PROFESSIONAL","BIRTHDAY","DIRECTOR_ID")
VALUES (20030102,'胡青牛','网络安全',TO_DATE('2-5月-1923','dd-Mon-yyyy HH:MI:SS
AM'),200203 );
INSERT INTO "SCOTT"."STUDENT"
("STUDENT_ID","NAME","PROFESSIONAL","BIRTHDAY","DIRECTOR_ID")
VALUES (20030103,'丁敏君','Web技术',TO_DATE('12-4月-1967','dd-Mon-yyyy HH:MI:SS
AM'),200201 );
INSERT INTO "SCOTT"."STUDENT"
("STUDENT_ID","NAME","PROFESSIONAL","BIRTHDAY","DIRECTOR_ID")
VALUES (20030104,'殷素素','Web安全',TO_DATE('14-7月-1971','dd-Mon-yyyy HH:MI:SS
AM'),200202 );
Commit;
```

7.6.2 查询数据表的数据

1. 查询导师信息表数据

(1) 利用【SQLPlus Worksheet】执行下列语句。

```
select * from scott.director;
```

【参见光盘文件】: \第 7 章\selectdirector.sql。

(2) 查询结果如图 7.26 所示。

2. 查询研究生信息表数据

(1) 利用【SQLPlus Worksheet】执行下列语句。

```
select * from scott.student;
```

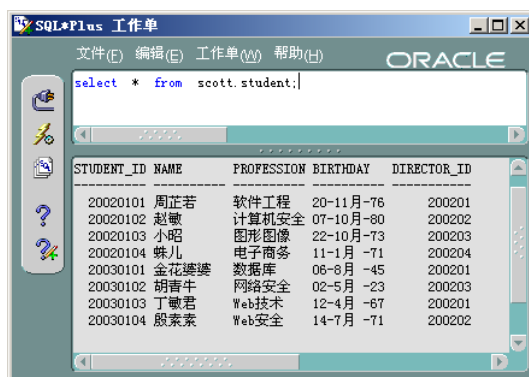
【参见光盘文件】：\第7章\selectstudent.sql。

(2) 查询结果如图 7.27 所示。



DIRECTOR_ID	NAME	ZHICHENG	ZHIWU
200201	张三丰	博导	系主任
200202	张翠山	硕导	室主任
200203	张无忌	硕导	所主任
200204	宋远桥	博导	系副主任

图 7.26 查询 scott.director 数据表的数据



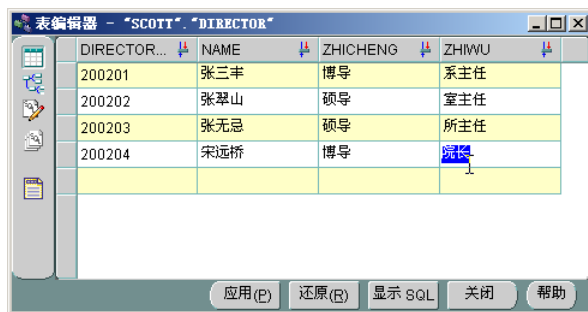
STUDENT_ID	NAME	PROFESSION	BIRTHDAY	DIRECTOR_ID
20020101	周正若	软件工程	20-11月-76	200201
20020102	赵敏	计算机安全	07-10月-80	200202
20020103	小昭	图形图像	22-10月-73	200203
20020104	蛛儿	电子商务	11-1月-71	200204
20030101	金花婆婆	数据库	06-8月-45	200201
20030102	胡青牛	网络安全	02-5月-23	200203
20030103	丁敏君	Web技术	12-4月-67	200201
20030104	殷素素	Web安全	14-7月-71	200202

图 7.27 查询 scott.student 数据表的数据

7.6.3 更新数据表的数据

1. 更新导师信息表数据

(1) 图 7.28 所示为数据表编辑器界面。在表格中直接将“DIRECTOR_ID”为“200204”的导师的“ZHIWU”由“系副主任”更改为“院长”。



DIRECTOR_ID	NAME	ZHICHENG	ZHIWU
200201	张三丰	博导	系主任
200202	张翠山	硕导	室主任
200203	张无忌	硕导	所主任
200204	宋远桥	博导	院长

图 7.28 在表数据编辑器界面中更新 scott.director 数据表的数据

(2) 单击 **显示 SQL** 按钮，出现的更新数据的 SQL 语句如下。

```
UPDATE "SCOTT"."DIRECTOR"
```

```
SET ZHIWU = '院长'
```

```
WHERE rowid = 'AAAH2kAAIAAAAYAAD'
```

这是什么东西？

(3) 读者可能会疑惑，rowid = 'AAAH2kAAIAAAAYAAD' 代表什么意思？

(4) Oracle 9i 在创建数据表时，默认会为每个数据表建立一个隐含的字段，叫 ROWID。在向数据表中插入记录时，系统将自动为每条记录分配惟一的一个 ROWID 号，利用这个

ROWID 号，可以快速定位到记录。

(5) 但是，利用 ROWID 更新数据的 SQL 语句只适用在同一台计算机上。不同的计算机环境可能为数据会分配不同的 ROWID 号，因此，我们需要给出一种普遍适用的更新数据表数据的语法。

(6) 在【SQLPlus Worksheet】中执行下列语句，作用是一样的。这是通过主码 DIRECTOR_ID 来定位记录的。这样的更新语句适合在不同的计算机环境上使用。

执行结果如图 7.29 所示。

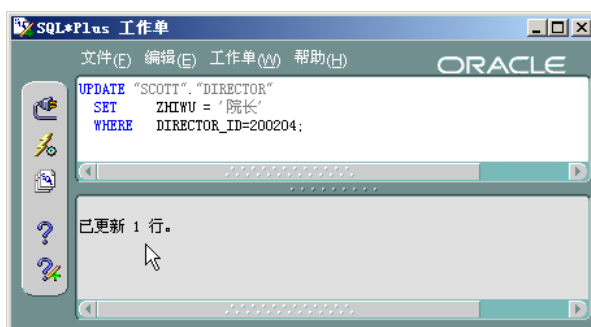


图 7.29 在【SQLPlus Worksheet】中更新 scott.director 的数据

【参见光盘文件】：\第 7 章\updatedirector.sql。

```
UPDATE "SCOTT"."DIRECTOR"
SET     ZHIWU = '院长'
WHERE   DIRECTOR_ID=200204;
```

2. 更新研究生信息表数据

按照同样的方法可以更新研究生数据表的数据。

(1) 在表数据编辑器中将研究生编号 STUDENT_ID 为“20030103”的研究生的导师编号字段 DIRECTOR 更改为 200204，如图 7.30 所示。

图 7.30 在表数据编辑器界面中更新 scott.student 数据表的数据

(2) 对应的 SQL 代码如下。


```
UPDATE "SCOTT"."STUDENT"
SET    DIRECTOR_ID = 200204
WHERE  STUDENT_ID = 20030103;
```

【参见光盘文件】: \第7章\updatestudent.sql。

7.6.4 删除数据表的数据

(1) 在图 7.31 所示的表数据编辑器界面中执行下列操作。



图 7.31 在表数据编辑器界面中更新 scott.student 数据表的数据

(2) 删除数据的 SQL 语句如下。

```
DELETE FROM "SCOTT"."STUDENT"
WHERE  STUDENT_ID = 20030101;
```

【参见光盘文件】: \第7章\deletestudent.sql。

7.7 截断数据表

截断操作的 SQL 语法如下。

```
TRUNCATE TABLE 用户名.表名 [DROP|REUSE STORAGE]
```

其中,若使用“DROP STORAGE”子句,显式指明释放数据表和索引的空间。若使用“REUSE STORAGE”子句,显式指明不释放数据表和索引的空间。下面以截断数据表 scott.director 为例进行介绍。

(1) 在【SQLPlus Worksheet】中执行 SQL 代码,结果如图 7.32 所示。

```
truncate table scott.director drop storage;
```

【参见光盘文件】: \第 7 章\truncatedirector.sql。

(2) 结果表明无法完成截断操作。这是因为导师信息表是父表, 研究生信息表引用导师信息表字段作为外码。

(3) 因此, 必须首先将 scott.student 的外码关系失效。在【SQLPlus Worksheet】中执行下列代码。执行结果如图 7.33 所示。

```
-----
alter table scott.student
disable constraint "导师编号外码";
-----
```

【参见光盘文件】: \第 7 章\disableconstraint.sql。

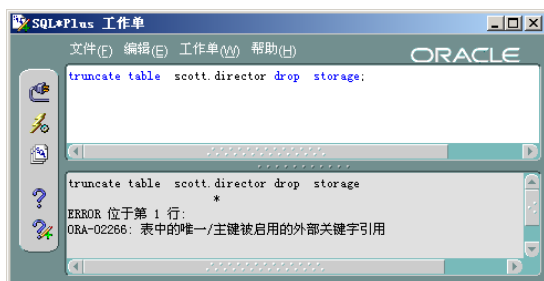


图 7.32 无法截断数据表的提示信息

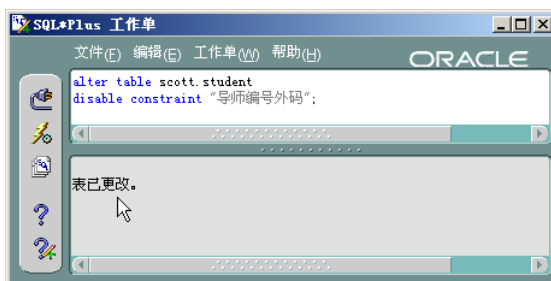


图 7.33 断开子表 scott.student 的外码关系

(4) 重新在【SQLPlus Worksheet】中执行 truncatedirector.sql, 结果如图 7.34 所示。表明成功完成表的截断。

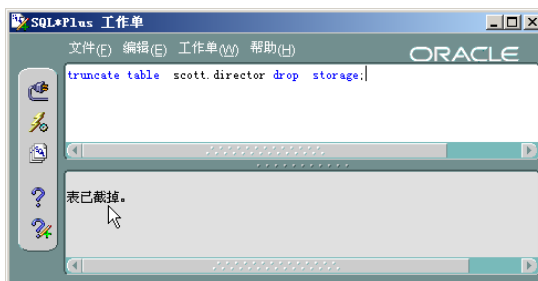


图 7.34 成功完成数据表的截断

7.8 删除数据表

数据表的删除比较简单, 表删除后其占用的空间就被系统释放和回收, 表的删除是无法回滚的操作。可删除的内容包括。

- ☐ 表的定义 表中的数据
- ☐ 表中的索引 表中的约束条件
- ☐ 表上的触发器 表中的权限

7.8.1 在【企业管理器】里删除表

- (1) 如图 7.35 所示。
- (2) 出现如图 7.36 所示界面。

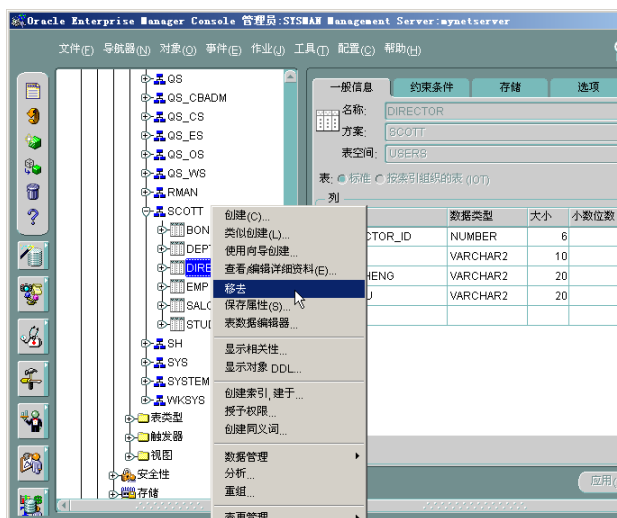


图 7.35 选择删除表



图 7.36 确定删除表

- (3) 对应上述删除数据表 director 的 SQL 代码为如下。

```
DROP TABLE SCOTT.DIRECTOR CASCADE CONSTRAINTS;
```

【参见光盘文件】: \第 7 章\dropdirector.sql。

- (4) 在【SQLPlus Worksheet】中执行 dropdirector.sql 的结果如图 7.37 所示。

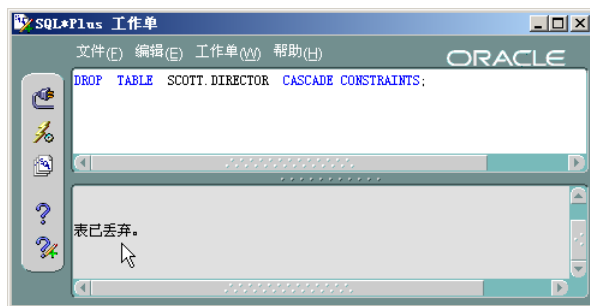


图 7.37 在【SQLPlus Worksheet】删除表

7.8.2 截断和删除的比较

TRUNCATE 对所有的表的操作速度都很快，DELETE 操作由于要产生大量的回滚信息，所以如果表很大，则删除速度较慢。

TRUNCATE 是 DDL 语言（数据定义语言），隐含了提交操作，因此不能回滚。

TRUNCATE 在表上或者在所有的索引中重新设置阈值,由于全部扫描操作和索引全表的快速操作将读所有未超过阈值的数据块,因此 DELETE 操作以后全表扫描的性能不会改进,但 TRUNCATE 操作以后全表扫描速度将加快。

截断表时,表和所有索引的存储参数可以设置为初始值,但 DELETE 操作不能缩小一个表及索引的大小。

截断操作不能删除完整性约束条件,而 DELETE 操作可以删除。

7.9 索引

索引是若干数据行的关键字的列表,查询数据时,通过索引中的关键字可以快速定位到要访问的记录所在的数据块,从而大大减少读取数据块的 I/O 次数,因此可以显著提高性能。

7.9.1 索引的原理

下面通过查询数据表“scott.student”的 ROWID 列为例,在【SQLPlus Worksheet】中执行下面的语句,查询结果如图 7.38 所示。

```
select rowid,student_id,name,professional,birthday,director_id
from scott.student;
```

【参见光盘文件】: \第 7 章\selectrowid.sql。

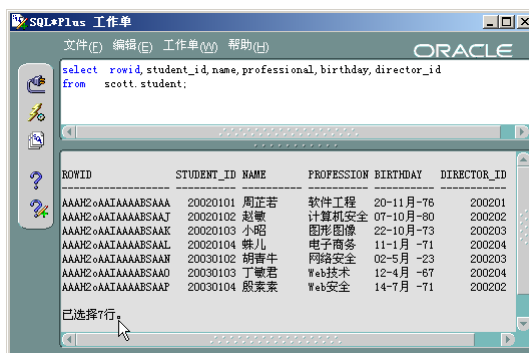


图 7.38 查询 ROWID 数据列

以数据表“scott.student”的索引为例,下面的数据就是以“student_id”数据列为例建立索引后的部分数据。

ROWID	STUDENT_ID
AAAH2oAAIAAAABSAAA	20020101
AAAH2oAAIAAAABSAAJ	20020102
AAAH2oAAIAAAABSAAK	20020103

AAAH2oAAIAAAAABSAAL	20020104
AAAH2oAAIAAAAABSAAN	20030102
AAAH2oAAIAAAAABSAAO	20030103
AAAH2oAAIAAAAABSAAP	20030104

7.9.2 Oracle 9i 支持的索引

Oracle 9i 中的索引可以分为两大类：B-树索引和位图索引。

1. B-树索引

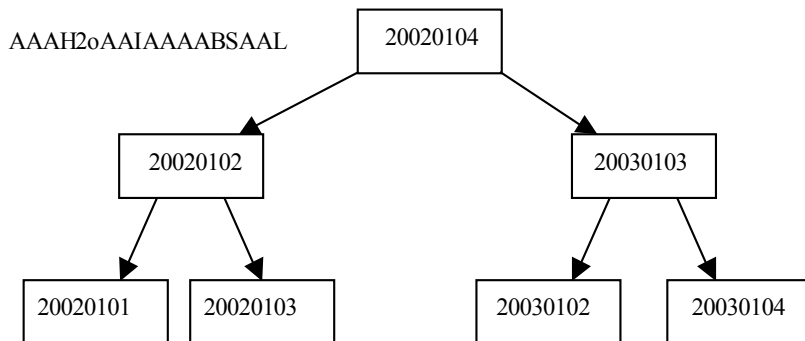


图 7.39 B-树索引的原理

表 7.9 B-树索引的分类

名称	含义
Non-Unique	非惟一索引，默认的 B-树索引，索引列值可以不是惟一的
Unique	惟一索引，在创建索引时指定“UNIQUE”关键字可以创建惟一索引。当建立“主码约束条件”时也会建立惟一索引，索引列值是惟一的
Reverse Key	反向关键字索引，通过在创建索引时指定“REVERSE”关键字，可以创建反向关键字索引，被索引的每个数据列中的数据都是反向存储的，但仍然保持原来数据列的次序
Function-based	基于函数的索引，对数据列使用表达式，按照表达式结果来创建 B-树索引的各节点，适合特定的，经常使用该表达式进行类似查询的数据表的索引的组织

2. 位图索引

对“scott.student”数据表的数据列“professional”建立位图索引可能的实例如下。

记录号	professional取值	位图索引值
1	软件工程	1
2	计算机安全	2
3	图形图像	3
.....		

7.9.3 主码自动建立的索引

(1) 如图 7.40 所示。

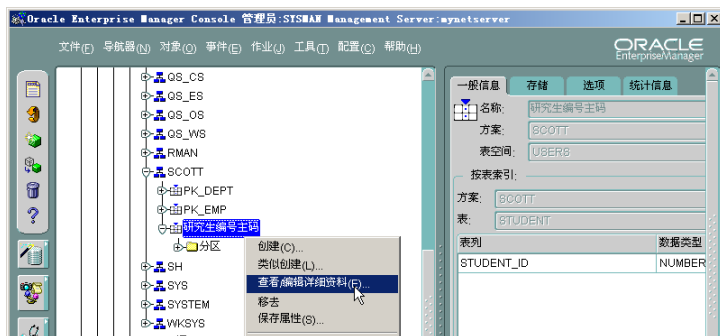


图 7.40 选择查看主码的索引

(2) 出现如图 7.41 所示的编辑索引的【一般信息】选项卡。

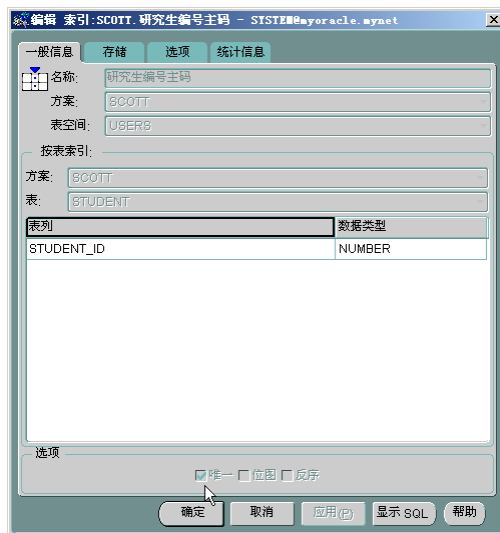


图 7.41 编辑索引的【一般信息】选项卡

7.9.4 如何创建索引

(1) 如图 7.42 所示。

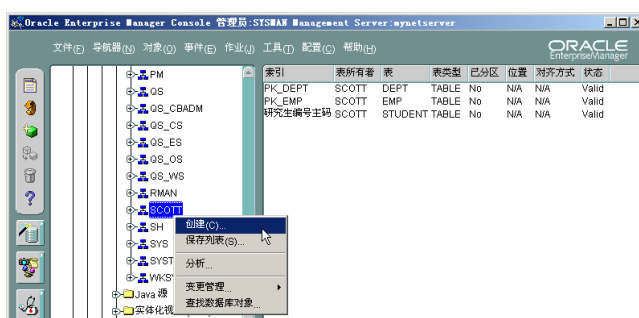


图 7.42 选择创建索引

(2) 出现如图 7.43 所示的创建索引的【一般信息】选项卡。

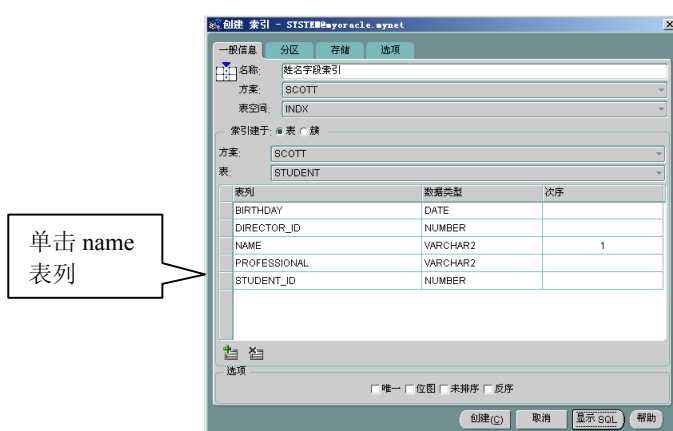


图 7.43 创建索引的【一般信息】选项卡

(3) 切换到图 7.44 所示的创建索引的【分区】选项卡。

(4) 切换到图 7.45 所示的创建索引的【存储】选项卡。

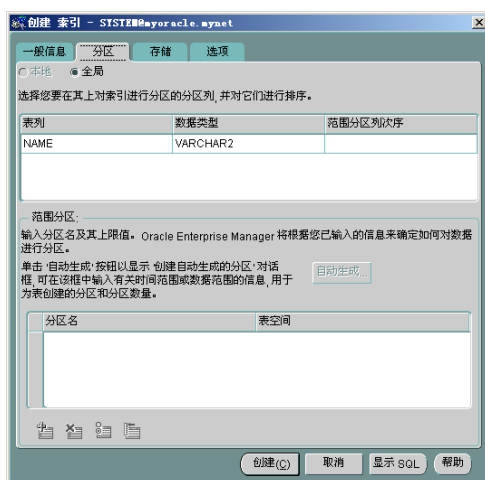


图 7.44 创建索引的【分区】选项卡

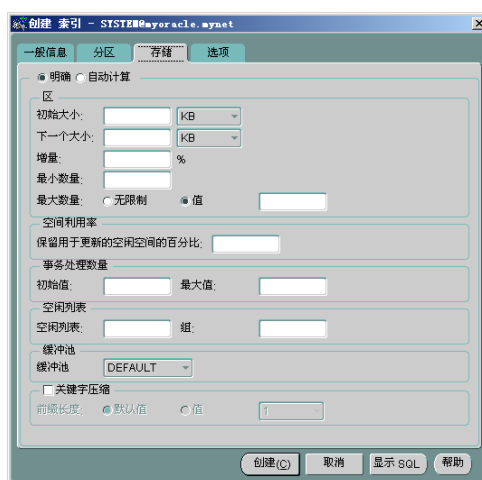


图 7.45 创建索引的【分区】选项卡

(5) 切换到图 7.46 所示的创建索引的【选项】选项卡。

(6) 成功创建索引后出现如图 7.47 所示界面。

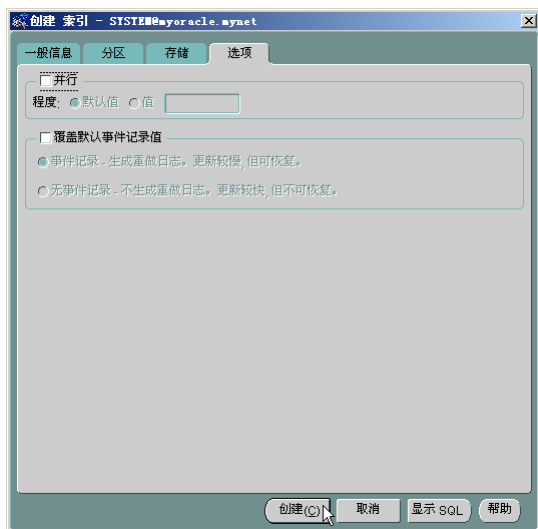


图 7.46 创建索引的【选项】选项卡



图 7.47 【成功创建索引】界面

(7) 在【企业管理器】中可以查看创建的索引，如图 7.48 所示。

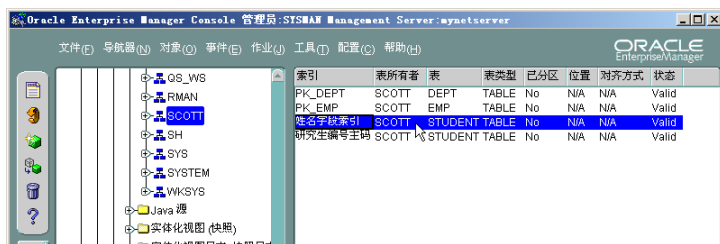


图 7.48 创建的索引

(8) 上述创建索引对应的 SQL 代码如下。

```
CREATE INDEX "SCOTT"."姓名字段索引"
ON "SCOTT"."STUDENT" ("NAME")
TABLESPACE "INDX";
```

【参见光盘文件】：\第 7 章\createindex.sql。

(9) 读者也可以直接在【SQLPlus Worksheet】中执行 createindex.sql 文件完成索引的创建，如图 7.49 所示。

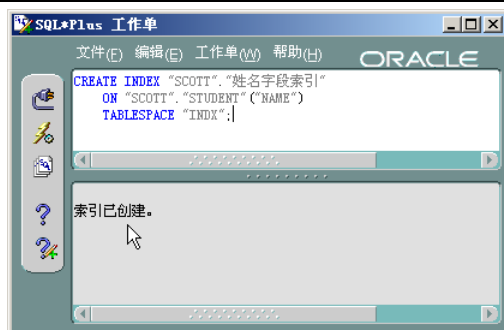


图 7.49 在【SQLPlus Worksheet】中创建索引

7.9.5 如何删除索引

- (1) 如图 7.50 所示。
- (2) 出现如图 7.51 所示的【索引删除确认】界面。

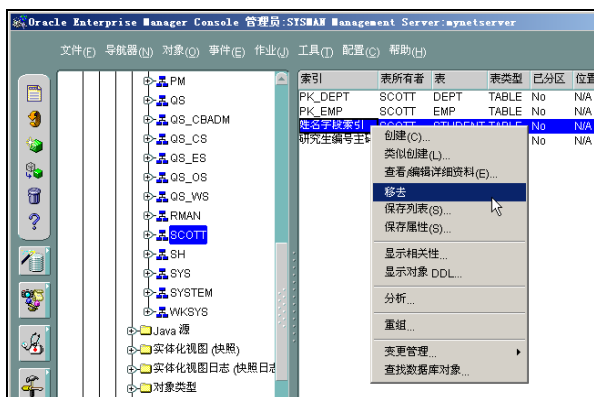


图 7.50 选择删除索引



图 7.51 【索引删除确认】界面

7.10 视图

7.10.1 什么是视图

视图犹如数据表的窗户，管理员定义这些“窗户”的位置后，用户就只能查看他可以看到的的数据。视图不是数据表，它仅是一些 SQL 查询语句的集合，作用是按照不同的要求从数据表中提取不同的数据。

7.10.2 如何创建视图

- (1) 如图 7.52 所示。
- (2) 出现如图 7.53 所示的创建视图的【一般信息】选项卡。

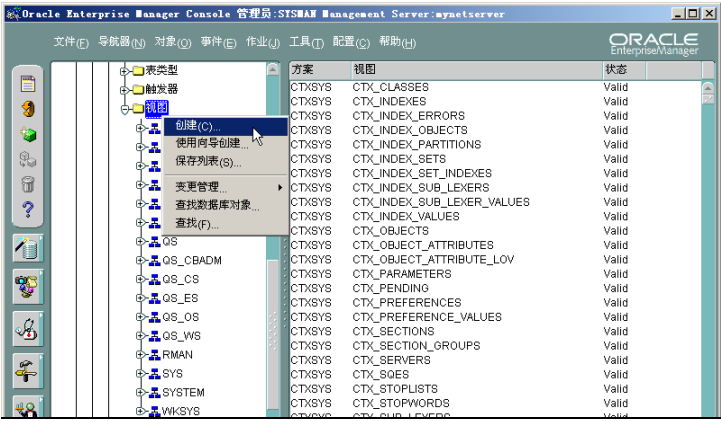


图 7.52 选择创建视图

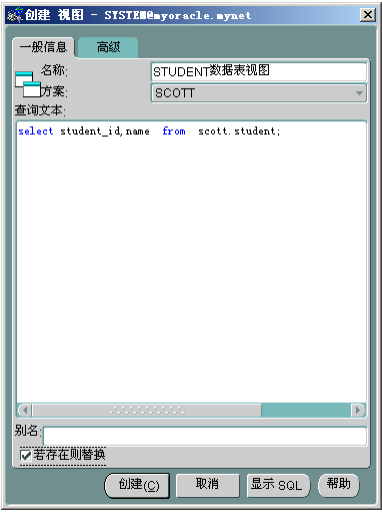



图 7.53 创建视图的【一般信息】选项卡

- (3) 切换到如图 7.54 所示的创建视图的【高级】选项卡。
- (4) 成功创建视图后出现如图 7.55 所示界面。单击  按钮。

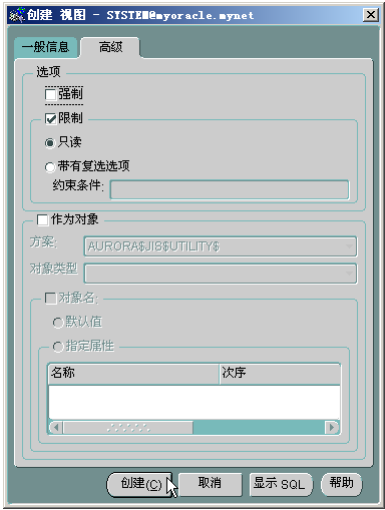


图 7.54 创建视图的【高级】选项卡

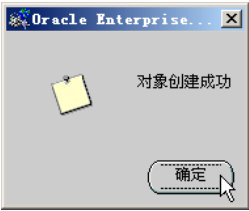


图 7.55 【成功创建视图】界面

- (5) 在【企业管理器】中可以查看创建的视图，如图 7.56 所示。

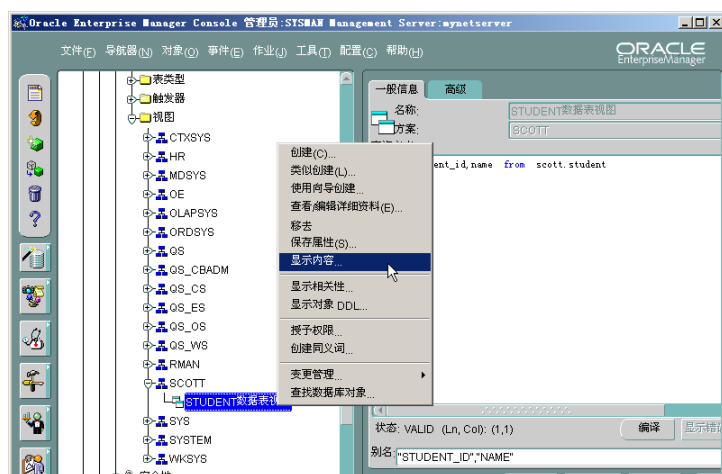


图 7.56 成功创建的视图

(6) 上述创建视图的 SQL 代码如下。

```
CREATE OR REPLACE VIEW "SCOTT"."STUDENT数据表视图" AS
  select student_id,name
  from scott.student WITH READ ONLY
```

【参见光盘文件】: \第7章\createview.sql。

(7) 读者也可以直接在【SQLPlus Worksheet】中执行 createview.sql 文件完成视图的创建，如图 7.57 所示。

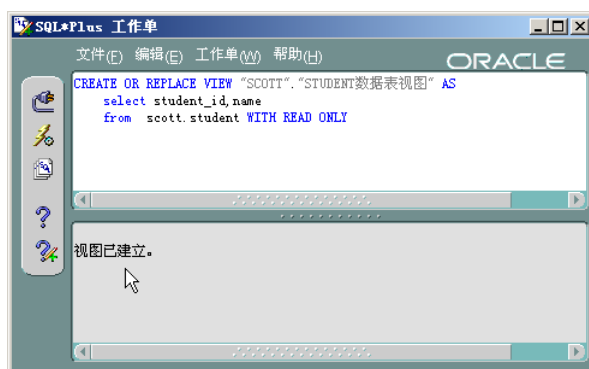


图 7.57 在【SQLPlus Worksheet】中创建视图

7.10.3 如何查询视图的数据

(1) 如图 7.56 所示。

(2) 出现如图 7.58 所示的【内容查看器】界面。显示了视图对应的数据。



图 7.58 【内容查看器】界面

(3) 上述过程对应的 SQL 代码如下。

```
select "SCOTT"."STUDENT数据表视图"."STUDENT_ID",
       "SCOTT"."STUDENT数据表视图"."NAME"
from "SCOTT"."STUDENT数据表视图"
```

【参见光盘文件】: \第 7 章\selectview.sql。

(4) 读者也可以直接在【SQLPlus Worksheet】中执行 selectview.sql 文件完成视图的查询，如图 7.59 所示。

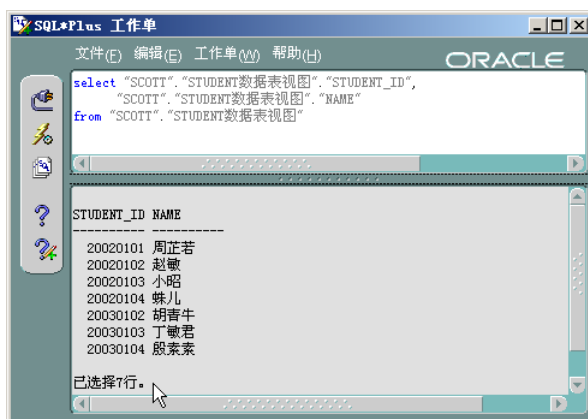


图 7.59 【内容查看器】界面

7.10.4 如何删除视图

(1) 如图 7.56 所示。

(2) 出现如图 7.60 所示的【视图删除确认】界面。

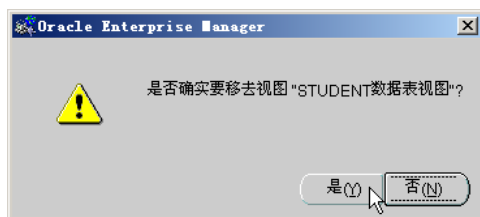


图 7.60 【视图删除确认】界面

(3) 这样，视图就被成功删除。

7.11 约束条件

约束条件就是 Oracle 数据库系统提供的对数据的完整性进行制约的机制。Oracle 9i 允许创建 5 种约束条件。参见表 7.8。

7.10.1 创建检查约束条件

- (1) 在【管理目标导航器】中按照 7.6 节修改数据表结构的步骤进行操作。
- (2) 切换到图 7.61 所示的编辑表的【约束条件】选项卡。
- (3) 上述创建检查约束条件的 SQL 代码如下。

```
ALTER TABLE "SCOTT"."STUDENT"
  ADD (CONSTRAINT "研究生编号检查约束条件"
  CHECK(student_id>=20020101 and student_id<=20030909))
```

【参见光盘文件】：\第 7 章\createcheck.sql。

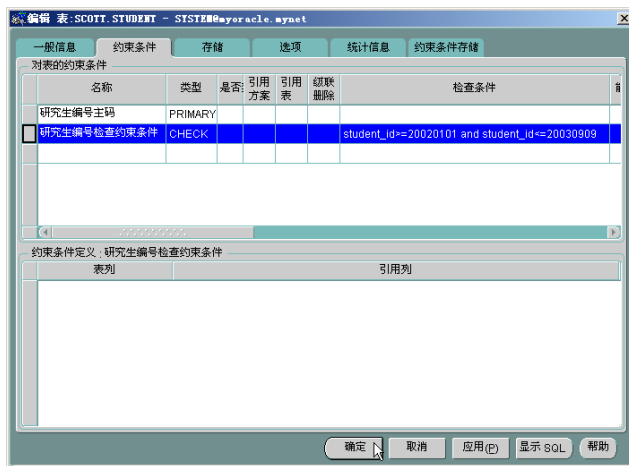


图 7.61 编辑表的【约束条件】选项卡

(4) 读者也可以直接在【SQLPlus Worksheet】中执行 createcheck.sql 文件完成检查约束条件的创建, 如图 7.62 所示。

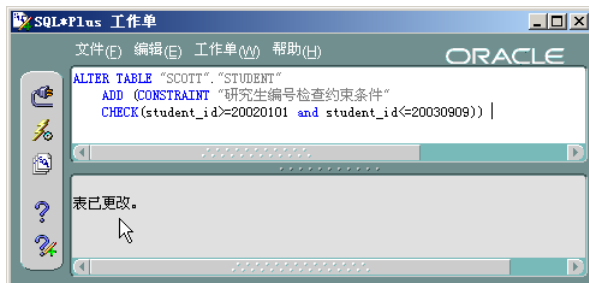


图 7.62 在【SQLPlus Worksheet】中创建检查约束条件

7.10.2 测试检查约束条件

(1) 在 7.63 所示的【表数据编辑器】界面中按照图示内容输入, 单击 **应用(P)** 按钮。

(2) 上述输入数据的 SQL 代码如下。

```
INSERT INTO "SCOTT"."STUDENT"
("STUDENT_ID", "NAME", "PROFESSIONAL", "BIRTHDAY", "DIRECTOR_ID")
VALUES (20010101, '纪晓芙', '软件工程', TO_DATE('15-7月 -1971', 'dd-Mon-yyyy HH:MI:SS
AM'), 200201)
```

【参见光盘文件】: \第 7 章\testcheck.sql。



图 7.63 在【表数据编辑器】中输入数据

(3) 出现如图 7.64 所示界面。

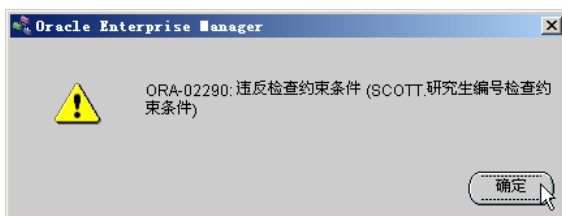


图 7.64 检查约束条件的执行结果

(4) 读者也可以直接在【SQLPlus Worksheet】中执行 testcheck.sql 文件完成检查约束条件的测试，结果如图 7.65 所示。

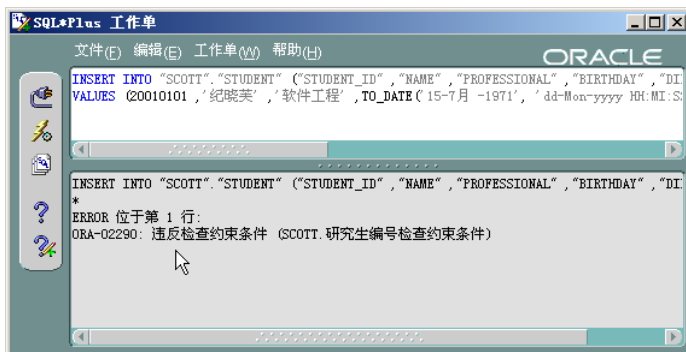


图 7.65 在【SQLPlus Worksheet】中测试检查约束条件

7.12 习题

- (1) Oracle 9i 中的 21 种方案对象，各有什么用途？
- (2) 通过实验熟悉 Oracle 9i 的 16 种基本数据类型的使用。
- (3) Oracle 9i 的数据表有哪些类型？各有什么用途？
- (4) 什么是主码？什么是外码？
- (5) 设计数据表的结构，通过实验完成数据表的创建。
- (6) 通过实验，熟悉数据表的数据的日常管理操作。
- (7) 在录入日期型字段的数据时，如何处理？
- (8) 截断和删除数据表有什么区别？
- (9) 索引的原理是什么？
- (10) B-树索引为什么能够加快数据查询速度？试通过实例说明 Oracle 9i 中的索引数据存储机制。
- (11) 视图有什么作用？
- (12) 检查约束条件的作用是什么？如何创建检查约束条件？
- (13) 自行设计有数据关联关系的至少两个数据表，独立完成数据表的创建、删除、数据的关联、索引的建立、视图的建立等环节，写出 SQL 代码报告，并阐述索引和视图建立的理由。
- (14) 自行设计数据表，设计主关键字、外关键字、惟一和检查约束条件，写出实验的 SQL 代码报告，并阐明设计约束条件的理由。
- (15) 通过实验，理解约束条件、视图、索引和数据表的关系。