

# 广州大学学生实验报告

开课实验室： 电子楼 418

2019 年 4 月 26 日

学院	计算机科学与网络 工程学院	年级、专 业、班	软件 171	姓名	谢金宏	学号	1706300001
实验课程名称		数据库原理				成绩	
实验项目名称		实验三 SQL 语言				指导老师	张少宏

教师评语：

## 一、实验目的

1. 熟悉并掌握使用 SQL 语句创建表、插入记录、查询记录、删除记录和修改记录的方法。
2. 尝试创建索引和删除索引。
3. 尝试创建视图、修改视图和删除视图。

## 二、实验环境

安装有 Oracle 11g 数据库软件的远程计算机和安装有 SQL Developer 软件的本地计算机。

## 三、实验内容

现有一个单位内部的小型图书借阅系统，假设每本图书的数量无限制，并且可以借给任何单位成员，每个单位成员可以借多本书，单位成员与图书的关系是多对多的关系。假设系统中仅有三个关系模式。

Reader 表

属性名	类型	长度	是否空	含义
RNO	varchar2	4		员工编号（主码）
Rname	varchar2	10	否	员工姓名
Rsex	varchar2	2		性别
Rage	integer	integer		年龄
Rboss	varchar2	10		直接上司
Raddress	varchar2	30		办公地点

Book 表

属性名	类型	长度	是否空	含义
BNO	varchar2	4		书本编号（主码）
Bname	varchar2	50	否	书名

Bauthor	varchar2	50		作者
Bpress	varchar2	50		出版社
Bprice	varchar2	numeric(6,2)		价格

RB 表

属性名	类型	长度	是否空	含义
RNO	varchar2	4		员工编号
BNO	varchar2	4		书本编号
RBdate	date			借阅日期

按照实验指导书建立对应表格，进行 SQL 查询：

## 基本操作

1. 以系统管理员身份连接成功后创建新的用户 exp3 并授权；
2. 新用户 exp3 的身份建立连接，并在此连接下执行后面的操作；
3. 建立表格 Reader；
4. 拷贝代码运行，向 Reader 表格中插入十条数据；
5. 查询记录：在 Reader 表中查询直接上司是“李四”的员工的名字；
6. 修改记录：在 Reader 表中把直接上司是“李四”的员工的办公地点统一改为“420”；
7. 删除记录：在 Reader 表中把直接上司未赋值（NULL）是记录删去；
8. 删去数据表：把整个 Reader 表删去；
9. 重新执行第一、二步，即建立数据表、插入数据。
10. 创建表格 Reader2，比较 Reader2 和 Reader 中的记录和结构是否相同；
11. 分别执行下面的每行语句，查看语句是否执行成功，分析为什么？

```
UPDATE "Reader" SET "RNO"='R001' WHERE "Rname"='张三';
UPDATE "Reader2" SET "RNO"='R001' WHERE "Rname"='张三';
INSERT INTO "Reader2"("RNO", "Rname", "Rsex", "Rage", "Rboss", "Raddress")
VALUES(null, 'Lisi', null, null, null, null);
```

12. 删除 Reader2 表格；
13. 拷贝代码运行，建立表格 Book；
14. 拷贝代码运行，向 Book 表格中插入 5 条数据；
15. 拷贝代码运行，建立表格 RB；
16. 拷贝代码运行，向 RB 表格中插入 13 条数据。

## 查询 READER 表中创建的约束类型

1. 执行 insert into RB(RNO,BNO) values('R010','B005')；
2. 写出删除 Reader 表格中编号为' R010' 的员工，如果执行错误，分析错误原因。
3. 想办法删除 Reader 中的' R010' 员工；  
方法一：先把 RB 中所有' R010' 的借书记录都删掉  
方法二：修改数据表 READER 的结构，允许级联删除（注：ORACLE 不支持级联更新）。
4. 为 Reader 表添加一个属性列“出生年份”，名为 Rbirthday，整数；

- 5. 对于 Reader 表格，员工编号可以确定年龄，年龄又可以确定出生年份，因此存在传递函数依赖关系，删除 Rbirthday 列，使关系模式符合第三范式要求；
- 6. 修改 Reader 表格的 Raddress 属性，使其长度为 50，数据类型不变；
- 7. 修改 Book 表的 Bprice 属性，使其值得范围在 10 到 100 之间；
- 8. 修改 Reader 表的 Rage 属性，使其值得范围为 16 到 60 之间。
- 9. 试试是否可以删除 Reader 表，使用 CASCADE 是否可以删除？
- 10. 删除 Reader、Book 和 RB 表。

索引的建立与删除

- 1. 重新执行前面的代码创建三个数据表并插入数据；
- 2. 为 Reader 表格的 Rname 建立 UNIQUE 索引；
- 3. 删除索引。

视图

- 1. 使用系统管理员身份为 exp3 用户授予 DBA 权限；
- 2. 建立在 416 办公室工作的视图 V416，视图包括员工的编号、姓名、年龄等信息；
- 3. 从 V416 中查询年龄大于 30 的员工信息；
- 4. 向视图 V416 中插入一条新的员工记录，然后从 V416 中查找该条记录，测试是否可以找到；
- 5. 插入成功了吗？为什么通过视图插入的记录在视图中看不见？
- 6. 建立在 417 办公室工作的视图 V417，视图包括员工的编号、姓名、性别、年龄等信息，视图定义带 With Check Option 选项；
- 7. 向视图 V417 中插入一条新的员工记录，然后从 V417 中查找该条记录，测试是否可以找到；
- 8. 通过视图删除刚插入的员工记录，在基本表 Reader 中查看是否已经删除成功？
- 9. 在视图 V417 上建立所有女员工信息的视图 FV417，查询视图结果；
- 10. 删除视图 V417 中没有借阅图书的员工信息；

```
SELECT * FROM V417;
Delete from v417 where v417.rno not in (SELECT rno from rb);
SELECT * FROM V417;
```

- 11. 建立视图 GV，数据包括每本图书的编号及其借阅数量。思考是否可以向 GV 中插入数据，为什么？

```
SELECT BNO,COUNT(*) FROM RB GROUP BY BNO;
CREATE VIEW GV(BNO,COUNTS) AS SELECT BNO,COUNT(*) FROM RB GROUP BY BNO;
SELECT * FROM GV;
```

- 12. 删除视图 V417

触发器

参考 PL/SQL 帮助文档，试试建立一个触发器。在用户执行删除某个员工的时候触发，该触发器

的功能是先删除 RB 中该员工的借书记录，然后删除 Reader 中该员工记录。

四、实验步骤

- 1. 创建新用户并授权；以新用户身份连接，创建表和插入初始数据。省略创建表和插入初始数据的语句。（此步骤创建的表和字段区分大小写，与实验指导书不区分大小写的模式不同。）

```
CREATE USER "exp3" IDENTIFIED BY "exp3pwd";
GRANT RESOURCE, CONNECT TO "exp3";
```

- 2. 以 Exp3 用户身份连接到数据库。
- 3. 若有重名的表 Reader、Book 或 RB，则将重名的表删去。
- 4. 建立表 Reader。

```
CREATE TABLE "Reader"
(
    "RNO"    VARCHAR2(4 CHAR) PRIMARY KEY,
    "Rname"  VARCHAR2(5 CHAR) NOT NULL,
    "Rsex"   VARCHAR2(1 CHAR),
    "Rage"   INTEGER,
    "Rboss"  VARCHAR2(10),
    "Raddress" VARCHAR2(30)
);
```

- 5. 使用拷贝的代码向表中插入 10 条数据。
- 6. 查询记录：在 Reader 表中查询直接上司是“李四”的员工的名字。

```
SELECT "Rname" FROM "Reader" WHERE "Rboss"='李四';
```

- 7. 修改记录：在 Reader 表中把直接上司是“李四”的员工的办公地点统一改为“420”。

```
UPDATE "Reader" SET "Raddress"='420'
WHERE "Rboss"='李四';
```

- 8. 删除记录：在 Reader 表中把直接上司未赋值（NULL）的记录删去。

```
DELETE FROM "Reader" WHERE "Rboss" IS NULL;
```

- 9. 删去数据表：把整个 Reader 表删去。

```
DROP TABLE "Reader";
```

- 10. 重新执行第一、二步，即建立数据表、插入数据。
- 11. 创建表格 Reader2，比较 Reader2 和 Reader 中的记录和结构是否相同。

```
CREATE TABLE "Reader2" AS SELECT * FROM "Reader";
DESC "Reader";
DESC "Reader2";
```

分析 Reader 表与 Reader2 表的 DDL 可知 Reader2 表的结构与 Reader 表基本相同，只是丢失了 Reader 表中原有的表级约束条件，如 Reader 表中原有的 RNO 主键，复制到 Reader2 表中后，并没有自然地称为 Reader2 表的主键。Reader2 表中仍然保有列级别的约束条件，如 Rname 非空。

我猜测表级约束条件没有自然地复制到新表的原因是用户可能从多个数据源复制数据到新表，如果自然地复制表级约束条件，那么从多个表复制时就可能出现表级约束条件重定义的错误。

```
CREATE TABLE "Reader" AS
SELECT * FROM "Reader1"
UNION ALL
SELECT * FROM "Reader2";
```

12. 分别执行下面的每行语句，查看语句是否执行成功，分析为什么？

```
UPDATE "Reader" SET "RNO"='R001' WHERE "Rname"='张三';
```

不成功，违反了 Reader 表中主键的唯一约束。

```
UPDATE "Reader2" SET "RNO"='R001' WHERE "Rname"='张三';
```

成功，Reader2 表中没有主键约束。

```
INSERT INTO "Reader2" ("RNO", "Rname", "Rsex", "Rage", "Rboss", "Raddress")
VALUES(NULL, 'Lisi', NULL, NULL, NULL, NULL);
```

成功，上述语句满足了 Reader2 表中的行级约束条件。

13. 删除 Reader2 表格。

```
DROP TABLE "Reader2";
```

14. 拷贝代码运行，建立表格 Book。

15. 拷贝代码运行，向 Book 表格中插入 5 条数据。

16. 拷贝代码运行，建立表格 RB。

17. 拷贝代码运行，向 RB 表格中插入 13 条数据。

#### 查询 READER 表中创建的约束类型

1. 执行 INSERT INTO RB(RNO,BNO) VALUES('R010','B005');

2. 写出删除 Reader 表格中编号为 R010 的员工，如果执行错误，分析错误原因。

```
DELETE FROM "Reader" WHERE "RNO"='R010';
```

无法删除，因为 RB 表中存在 RNO 为 R010 的记录，而为了保证完整性约束，DBMS 禁止删除这条记录。

3. 设法删除 Reader 中的 R010 员工

-- 3.a) 先把 RB 中所有'R010'的借书记录都删掉

```
DELETE FROM "RB" WHERE "RNO"='R010';
DELETE FROM "Reader" WHERE "RNO"='R010';
```

-- 3.b) 修改 RB 表的结构中外键的定义，允许级联删除

-- 确认旧外键约束的名称

```
SELECT constraint_name FROM user_constraints
WHERE table_name='RB' AND constraint_type='R'
AND r_constraint_name=(
    SELECT constraint_name FROM user_constraints
    WHERE table_name='Reader' AND constraint_type='P'
);
```

-- 删除旧的外键约束

```
ALTER TABLE "RB"
DROP CONSTRAINT SYS_C007345;
```

-- 建立新的外键约束

```
ALTER TABLE "RB"
ADD FOREIGN KEY ("RNO") REFERENCES "Reader"("RNO")
ON DELETE CASCADE;
```

-- 删除记录

```
DELETE FROM "Reader" WHERE "RNO"='R010';
```

-- 注：不能修改约束，只能删除约束后重建

4. 为 Reader 表添加一个属性列“出生年份”，名为 Rbirthday，整数。

```
ALTER TABLE "Reader"
ADD "Rbirthday" INT;
```

5. 对于 Reader 表格，员工编号可以确定年龄，年龄又可以确定出生年份，因此存在传递函数

依赖关系，删除 Rbirthday 列，使关系模式符合第三范式要求。

```
ALTER TABLE "Reader"
DROP COLUMN "Rbirthday";

-- 注意添加列时不需要指明 COLUMN 但删除列时需要指明 COLMN
```

向表中添加对象（列或约束条件）时不需要使用 COLUMN 或 CONSTRAINT 关键词指明对象类型，因为对象类型蕴含在添加对象的语句中；但从表中删除对象时需要使用 COLUMN 或 CONSTRAINT 关键词指明对象类型，因为同一个表中不同类型的对象可能拥有相同的标识符。

6. 修改 Reader 表格的 Raddress 属性，使其长度为 50，数据类型不变。

```
ALTER TABLE "Reader"
MODIFY "Raddress" VARCHAR2(50 CHAR);
```

因为约束类型不可被修改，因此 MODIFY 语句也不需要使用 COLUMN 来指明类型。

7. 修改 book 表的 Bprice 属性，使其值得范围在 10 到 100 之间。

```
ALTER TABLE "Book"
ADD CHECK ("Bprice" BETWEEN 10 AND 100);
```

8. 修改 Reader 表的 Rage 属性，使其值得范围为 16 到 60 之间。

```
ALTER TABLE "Reader"
ADD CHECK ("Rage" BETWEEN 16 AND 60);
```

9. 尝试不带 CASCADE 的和带 CASCADE 选项的语句删除 READER 表。

```
DROP TABLE "Reader"; -- 无法删除，表中的属性被其他表的外键所引用。

DROP TABLE "Reader" CASCADE CONSTRAINTS;
```

10. 删除 Book 表和 RB 表。

### 索引的建立与删除

1. 重新执行前面的语句创建表和插入数据。
2. 为 Reader 表的 Rname 建立 Unique 索引。

```
CREATE UNIQUE INDEX "Rname" ON "Reader"("Rname");
```

由于数据表中存在重名的 Rname，DBMS 会拒绝创建表。这时候可以选择建立普通索引，如果坚

持要建立 UNIQUE 索引，则必须删去表中一定的记录使得 Rname 唯一。

-- 2+. 可以建立普通索引

```
CREATE INDEX "Rname" ON "Reader"("Rname");
```

-- 2+. 删除重复的 Rname 键后建立唯一索引

-- 在删除 Reader 表中 Rname 重复的记录前，先删除与之以外键约束关联的 RB 表中的记录

```
DELETE FROM "RB"
WHERE "RNO" IN (
    SELECT "RNO" FROM "Reader" a
    WHERE ROWID > (
        SELECT MIN(ROWID) FROM "Reader" b
        WHERE b."Rname"=a."Rname"
    )
);
```

-- 删除 Reader 表中 Rname 重复的记录

```
DELETE FROM "Reader" a
WHERE ROWID > (
    SELECT MIN(ROWID) FROM "Reader" b
    WHERE b."Rname"=a."Rname"
);
```

-- 建立唯一索引

```
CREATE UNIQUE INDEX "Rname" ON "Reader"("Rname");
```

3. 删除索引。

```
DROP INDEX "Rname";
```

### 视图

1. 为当前用户授予必要的 DBA 权限以便创建视图的操作。
2. 建立在 416 办公室工作的视图 V416。



```
CREATE VIEW "V416" AS SELECT "RNO", "Rname", "Rage" FROM "Reader"
WHERE "Raddress"='416';
```

3. 从视图中查询年龄大于 30 岁的员工的信息

```
SELECT * FROM "V416" WHERE "Rage">30;
```

4. 向视图 V416 中插入一条新的员工记录，然后从 V416 中查找该条记录，测试是否可以找到。

```
INSERT INTO "V416" ("RNO", "Rname", "Rage") VALUES('R999', '新人', 99);

SELECT * FROM "Reader";
SELECT * FROM "V416";
```

向 Reader 表中插入成功，但由于新人的 Raddress 不是 416，因此不会在 V416 中出现。

5. 建立在 417 办公室工作的视图 V417，视图包括员工的编号、姓名、性别、年龄等信息，视图定义带 with check option 选项。

```
CREATE VIEW "V417" AS
SELECT "RNO", "Rname", "Rsex", "Rage", "Raddress" FROM "Reader" WHERE
"Raddress"='417'
WITH CHECK OPTION;
```

6. 向视图 V417 中插入一条新的员工记录，然后从 V417 中查找该条记录，测试是否可以找到。

```
INSERT INTO "V417" ("RNO", "Rname", "Rage")
VALUES('R777', '417 新人', 66); -- 违反了With Check Option 选项, 插入失败
```

```
INSERT INTO "V417" ("RNO", "Rname", "Rage", "Raddress")
VALUES('R777', '417 新人', 66, '417'); -- 插入成功
```

7. 通过视图删除刚插入的员工记录。

```
DELETE FROM "V417" WHERE "RNO"='R777';
```

基本表中也对应删除成功。

8. 在视图 V417 上建立所有女员工信息的视图 FV417，查询视图结果。

```
CREATE VIEW "FV417" AS
SELECT * FROM "V417"

WHERE "Rsex"='女';
```

```
SELECT * FROM "FV417";
```

9. 删除视图 V417 中没有借阅图书的员工信息。

```
DELETE FROM "V417" where "V417"."RNO" NOT IN (SELECT "RNO" FROM "RB");
```

10. 建立视图 GV，数据包括每本图书的编号及其借阅数量。思考是否可以向 GV 中插入数据，为什么？

```
CREATE VIEW "GV" AS
SELECT "BNO", COUNT(*) "Bcount" FROM "RB"

GROUP BY "BNO"; -- 注意必须为视图中的列创建别名
```

不能向 GV 中插入数据，因为 GV 表中的记录是经过计算得到的，没有基本表与之直接关联。

11. 删除前几步中创建的视图。

```
DROP VIEW "GV";
DROP VIEW "FV417";
DROP VIEW "V417";
DROP VIEW "V416";
```

## 触发器

参考 PLSQL 帮助文档，建立一个触发器。在用户执行删除某个员工的时候触发，该触发器的功能是先删除 RB 中该员工的借书记录，然后删除 Reader 中该员工记录。

```
CREATE TRIGGER "REMOVE_RELATED_RECORDS"
BEFORE DELETE ON "Reader"
FOR EACH ROW
BEGIN
    DELETE FROM "RB"
    WHERE "RNO"=:old."RNO";
END;
```

## 五、分析总结

完成了包括触发器在内的全部的实验内容，掌握了与表、索引、视图和触发器相关的一些重要的 SQL 语法，特别是了解了 ADD、DROP 和 MODIFY 几个动词之间微妙的语法差异。

触发器的调试是相对不容易的，写起来要有耐心。