

# 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		实验二 SQL 数据查询				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，进行完成 SQL 查询：

### 单表查询

1. 查询全体员工的姓名和出生年份；
2. 查询工作在 416 房间的员工的所有信息；
3. 查询年龄在 30 到 50 岁之间的员工姓名、年龄；
4. 查询借了书的员工的编号，排除相同的元素；
5. 查询名字中包含字“小”的员工姓名、办公地点；
6. 查询姓名中第二个字为“小”的员工姓名、办公地点；
7. 查询所有不姓“李”的员工姓名、性别；
8. 查询 Book 表中价格不为空值的书名、出版社；
9. 查询清华大学出版社和高等教育出版社出版的所有书籍，按照出版社降序、书名升序排列；
10. 查询员工的总人数；
11. 查询借了书的员工的人数；
12. 查询“张三”所借图书的数量；（注意，员工里有两个人叫张三）；
13. 查询最贵的书籍的作者姓名；
14. 查询 Book 表中书籍的平均价格，查询结果说明了什么；
15. 查询 book 中包含的各个出版社及其出版书籍的数量。

### 复合查询

1. 查询每个员工及其借书情况，列出员工编号、姓名和借书日期；
2. 查询每个员工及其借书所有字段（没有借书的员工也列出来）；
3. 查询与“李小龙”工作在同一个办公室的员工信息（用两种方法：自身连接、子查询）；
4. 查询借阅了“数据库原理”的员工所有信息（两种方法：连接查询、子查询）；
5. 查询小于或等于同一办公室中成员的平均年龄的员工编号、姓名、年龄；
6. 查询比所有数据库原理价格都低、并且不是清华大学出版社出版的书籍的信息；
7. 查询借阅了 B001 的员工的编号、姓名、办公室；
8. 查询没有借阅 B001 的员工的编号、姓名、办公室；
9. 查询借阅了所有书籍的员工的姓名；
10. 查询至少借阅了编号为 R004 的员工借阅的全部书籍的员工姓名、性别、办公室；
11. 查询年龄大于 30 岁或者工作在 416 的员工信息（用集合查询完成）；
12. 查询年龄大于 30 岁并且工作不在 416 的员工信息（用集合查询完成）。

## 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，与实验指导书不同不区分大小写的模式不同。）

### 2. 单表查询

-- 1. 查询全体员工的姓名和出生年份

```
SELECT
  "Rname" AS "员工姓名",
  EXTRACT(YEAR FROM SYSDATE)-"Rage" AS "出生年份"
FROM "Reader";
```

-- 2. 查询工作在 416 房间的员工的所有信息

```
SELECT * FROM "Reader" WHERE "Raddress"='416';
```

-- 3. 查询年龄在 30 到 50 岁之间的员工姓名、年龄

-- 注意 SQL BETWEEN 操纵的区间为闭区间[30, 50]

```
SELECT "Rname", "Rage" FROM "Reader" WHERE "Rage" BETWEEN 30 AND 50;
```

-- 4. 查询借了书的员工的编号，排除相同的元素

```
SELECT DISTINCT "RNO" FROM "RB";
```

-- 5. 查询名字中包含字“小”的员工姓名、办公地点

```
SELECT "Rname", "Raddress" FROM "Reader" WHERE "Rname" LIKE '%小%';
```

-- 6. 查询姓名中第二个字为“小”的员工姓名、办公地点；（原题为名字中第二个字不为小）

```
SELECT "Rname", "Raddress" FROM "Reader" WHERE "Rname" LIKE '_小%';
```

-- 7. 查询所有不姓“李”的员工姓名、性别；

```
SELECT "Rname", "Rsex" FROM "Reader" WHERE "Rname" NOT LIKE '李%';
```

-- 8. 查询 Book 表中价格不为空值的书名、出版社；

```
SELECT "Bname", "Bpress", "Bprice" FROM "Book" WHERE "Bprice" IS NOT NULL;
```

-- 9. 查询清华大学出版社和高等教育出版社出版的所有书籍，按照出版社降序、书名升序排列；

-- 【验证语句】

```
SELECT * FROM "Book";
```

-- 【查询语句】

```
SELECT * FROM "Book" WHERE "Bpress" in ('清华大学出版社', '高等教育出版社')
ORDER BY "Bpress" DESC, "Bname" ASC;
```

-- 10. 查询员工的总人数

```
SELECT COUNT("RNO") FROM "Reader";
```

-- 11. 查询借了书的员工的人数

```
SELECT COUNT(DISTINCT "RNO") FROM "Reader"
WHERE "Reader"."RNO" IN (SELECT "RNO" FROM "RB");
```

-- 或

```
SELECT COUNT(DISTINCT "RNO") FROM "RB";
```

-- 12. 查询“张三”所借图书的数量(注意，员工里有两个人叫张三)

```
SELECT "RNO", COUNT(*) FROM "RB"
WHERE "RNO" IN (SELECT "RNO" FROM "Reader" WHERE "Rname"='张三')
GROUP BY "RNO";
```

-- 13. 查询最贵的书籍的作者姓名；

```
SELECT "Bname", "Bauthor", "Bprice" FROM "Book"
WHERE "Bprice"=(
  SELECT MAX("Bprice") FROM "Book"
);
```

-- 14. 查询 Book 表中书籍的平均价格

-- 注意分组函数自动过滤掉了空值 null

```
SELECT AVG("Bprice") FROM "Book";
```

-- 15. 查询 Book 中包含的各个出版社及其出版书籍的数量

```
SELECT "Bpress", COUNT("BNO") FROM "Book"
GROUP BY "Bpress";
```

### 3. 复合查询

```

-- 1. 查询每个员工及其借书情况，列出员工编号、姓名和借书日期
SELECT "Reader"."RNO", "Rname", "RBdate" From "Reader", "RB"
WHERE "Reader"."RNO"="RB"."RNO";

-- 2. 使用左外连接查询每个员工及其借书所有字段（没有借书的员工也列出来）
SELECT "Reader"."RNO", "Rname", "Rsex", "BNO", "RBdate" From "Reader"
LEFT JOIN "RB" ON "Reader"."RNO"="RB"."RNO";

-- 3. 查询与“李小龙”工作在同一个办公室的员工信息；
-- 自身连接
SELECT b."Rname" FROM "Reader" a, "Reader" b
WHERE b."Raddress"=a."Raddress" AND a."Rname"='李小龙'
AND a."Rname"!=b."Rname";
-- 子查询
SELECT "Rname" FROM "Reader"
WHERE "Raddress"=(SELECT "Raddress" FROM "Reader" WHERE "Rname"='李小龙')
AND "Rname"!='李小龙';

-- 4. 查询借阅了“数据库原理”的员工所有信息（两种方法：连接查询、子查询）
-- 【连接查询】
SELECT DISTINCT "Reader"."RNO", "Rname", "Rsex", "Raddress" FROM "Reader", "Book", "RB"
WHERE "Reader"."RNO"="RB"."RNO" AND "Book"."BNO"="RB"."BNO"
AND "Book"."Bname"='数据库原理';

-- 【子查询】
SELECT DISTINCT "RNO", "Rname", "Rsex", "Raddress" FROM "Reader"
WHERE "RNO" IN (
    SELECT "RNO" FROM "RB"
    WHERE "BNO" IN ( -- 此处注意不能使用相等判断“=”，因为有多本书名为“数据库原理”，而“=”
只能用于单值
        SELECT "BNO" FROM "Book" WHERE "Bname"='数据库原理'
    )
);

```

```

-- 5. 查询小于或等于同一办公室中成员的平均年龄的员工编号、姓名、年龄
-- 【验证语句】
SELECT "RNO", "Rname", "Rage", "Raddress" FROM "Reader"
ORDER BY "Raddress";

SELECT "Raddress", AVG("Rage") FROM "Reader"
GROUP BY "Raddress";

-- 【查询语句】
SELECT a."RNO", a."Rname", a."Rage", a."Raddress" FROM "Reader" a
WHERE a."Rage" <= (
    SELECT AVG(b."Rage") FROM "Reader" b WHERE a."Raddress"=b."Raddress"
);

-- 6. 查询比所有数据库原理价格都低、并且不是清华大学出版社出版的书籍的信息
-- 【验证语句】
SELECT * FROM "Book" WHERE "Bname"='数据库原理';
SELECT MIN("Bprice") FROM "Book" WHERE "Bname"='数据库原理';
SELECT * FROM "Book";

-- 【查询语句】
SELECT * FROM "Book"
WHERE "Bprice" < (
    SELECT MIN("Bprice") FROM "Book" WHERE "Bname"='数据库原理'
)
AND "Bpress"!='清华大学出版社';

-- 7. 查询借阅了 B001 的员工的编号、姓名、办公室
-- 【验证语句】
SELECT "Reader"."RNO", "Rname", "Raddress" FROM "Reader";
SELECT * FROM "RB" WHERE "BNO"='B001';

-- 【查询语句】
SELECT "Reader"."RNO", "Rname", "Raddress" FROM "Reader", "RB"
WHERE "Reader"."RNO"="RB"."RNO" AND "RB"."BNO"='B001';

-- 8. 查询没有借阅 B001 的员工的编号、姓名、办公室
SELECT "Reader"."RNO", "Rname", "Raddress" FROM "Reader"
WHERE "Reader"."RNO" NOT IN (
    SELECT "RNO" FROM "RB" WHERE "BNO"='B001'
);

```

```
-- 9. 查询借阅了所有书籍的员工的姓名
-- 【验证语句】
SELECT "BNO" FROM "Book";
SELECT "Reader"."RNO", "Rname", "RB"."BNO" FROM "RB"
LEFT OUTER JOIN "Reader" ON ("Reader"."RNO"="RB"."RNO");

-- 【查询语句】
SELECT "Rname" FROM "Reader"
WHERE NOT EXISTS (
    SELECT "BNO" FROM "Book"
    WHERE NOT EXISTS (
        SELECT * FROM "RB"
        WHERE "RB"."RNO"="Reader"."RNO" AND "RB"."BNO"="Book"."BNO"
    )
);
-- 或借助集合查询
SELECT "Rname" FROM "Reader"
WHERE NOT EXISTS (
    SELECT "BNO" FROM "Book"
    MINUS
    SELECT "BNO" FROM "RB" WHERE "RB"."RNO"="Reader"."RNO"
);
-- 或借助 COUNT()
SELECT "Rname" FROM "Reader"
WHERE
(SELECT COUNT(*) FROM "Book")=
(SELECT COUNT(*) FROM "RB" WHERE "RB"."RNO"="Reader"."RNO");
```

-- 10. 查询至少借阅了编号为 R004 的员工借阅的全部书籍的员工姓名、性别、办公室

```
SELECT * FROM "RB" WHERE "RNO"='R004';
SELECT "Reader"."RNO", "Rname", "RB"."BNO" FROM "RB"
LEFT OUTER JOIN "Reader" ON ("Reader"."RNO"="RB"."RNO");
```

```
-- 【查询语句】
SELECT "RNO", "Rname" FROM "Reader"
WHERE NOT EXISTS (
    SELECT "BNO" FROM "RB" a WHERE a."RNO"='R004'
    AND NOT EXISTS (
        SELECT * FROM "RB" b
        WHERE b."RNO"="Reader"."RNO" AND b."BNO"=a."BNO"
    )
);
-- 或借助集合查询
SELECT "Rname", "Rsex", "Raddress" FROM "Reader"
WHERE NOT EXISTS (
    SELECT "BNO" FROM "RB" WHERE "RNO"='R004'
    MINUS
    SELECT "BNO" FROM "RB" WHERE "RB"."RNO"="Reader"."RNO"
);
```

-- 11. 查询年龄大于 30 岁或者工作在 416 的员工信息（用集合查询完成）

```
SELECT * FROM "Reader" WHERE "Rage">30
UNION
SELECT * FROM "Reader" WHERE "Raddress"='416';
```

-- 12. 查询年龄大于 30 岁并且工作不在 416 的员工信息（用集合查询完成）

```
SELECT * FROM "Reader" WHERE "Rage">30
MINUS
SELECT * FROM "Reader" WHERE "Raddress"='416';
```

## 五、分析总结

完成了全部的实验内容。

在使用 Oracle PL/SQL 时，EXECUTE IMMEDIATE 子句接受的 SQL 语句中不要使用分号作为结束符（不需要结束符），否则执行时会报出极具误导性的“标识符非法”错误。

```
EXECUTE IMMEDIATE 'DROP TABLE "Reader"'; -- 传递的 SQL 语句不应该附加“;”作为结束符
```

Oracle 集合查询的集合差关键词是 MINUS，而 MariaDB 是 EXCEPT。

Oracle 与 MariaDB 在创建表的别名时语法不同：Oracle 在为表创建别名时不能使用 AS 关键词，但在为字段创建别名时可以使用 AS 关键词。MariaDB 则是创建表和字段的别名时都可以使用 AS 关键词。

-- 使用 AS 关键词创建别名

```
SELECT COLUMN_NAME AS col FROM TABLE_NAME tb; -- Oracle
```

```
SELECT COLUMN_NAME AS col FROM TABLE_NAME AS tb; -- MariaDB
```