

实验题目	关系数据库管理系统与 SQL			实验日期	2022 年 4 月 3 日
班级	1903104	学号	1190201421	姓名	张瑞

CS33503 数据库系统实验

实验检查记录

实验结果的正确性 (60%)		表达能力 (10%)	
实验过程的规范性 (10%)		实验报告 (20%)	
加分 (5%)		总成绩 (100%)	

实验报告

一、实验目的（介绍实验目的）

1. 掌握一种关系数据库管理系统(RDBMS)的使用方法。
2. 学会使用 SQL 创建、修改、查询和控制关系数据库。

二、实验环境（介绍实验使用的硬件设备、软件系统、开发工具等）

1. 硬件设备：Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz 2.71 GHz; 8GB RAM
2. 软件系统：Windows 10
3. 开发工具：8.0.28 MySQL Community Server - GPL

三、实验过程（介绍实验过程、设计方案、实现方法、实验结果等）

1. 连接 MySQL 数据库：

在命令行输入：mysql -u root -p

然后输入密码即可登录：

```
C:\Users\ZR>mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 16
Server version: 8.0.28 MySQL Community Server - GPL

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

2. MySQL 实用命令：

查看数据库列表：SHOW DATABASES;

连接数据库（以 College 数据库为例）：USE College;

查看数据库中关系名：SHOW TABLES;

查看关系模式（以 Student 关系为例）：DESCRIBE Student;

创建新数据库：CREATE DATABASE mydb;

创建新关系（以 Student 关系为例）：CREATE TABLE Student(属性名及属性类型...);

实验题目	关系数据库管理系统与 SQL			实验日期	2022 年 4 月 3 日
班级	1903104	学号	1190201421	姓名	张瑞

删除关系：DROP TABLE Student;
删除数据库：DROP DATABASE mydb;
寻求帮助：\h;
退出：\q;

3. 使用 MySQL 创建 College 数据库：

首先创建 College 数据库：

```
mysql> CREATE DATABASE College;
Query OK, 1 row affected (0.01 sec)
```

然后创建 Student 表：

```
mysql> CREATE TABLE Student(
-> Sno CHAR(6) PRIMARY KEY,
-> Sname VARCHAR(10),
-> Ssex CHAR,
-> Sage INT,
-> Sdept VARCHAR(20)
-> );
Query OK, 0 rows affected (0.06 sec)
```

接下来插入第一条关系元组：

```
mysql> INSERT INTO Student VALUES ('PH-001', 'Nick', 'M', 20, 'Physics');
Query OK, 1 row affected (0.02 sec)
```

查看插入后 Student 表中结果：

```
mysql> SELECT * FROM Student;
```

Sno	Sname	Ssex	Sage	Sdept
PH-001	Nick	M	20	Physics

```
1 row in set (0.00 sec)
```

可见已成功插入数据。

接下来继续插入 Student 表中的其他关系元组，当 Student 表创建结束后，同理创建 Course 表和 SC 表，最终创建完整个 College 数据库。

4. 验证课程示例

由于课程示例较多，此处仅选取部分进行展示。

查询计算机系学生的学号和姓名：

```
mysql> SELECT Sno, Sname FROM Student WHERE Sdept = 'CS';
```

Sno	Sname
CS-001	Elsa
CS-002	Ed

```
2 rows in set (0.01 sec)
```

查询选了课但还未取得成绩的学生：

实验题目	关系数据库管理系统与 SQL			实验日期	2022 年 4 月 3 日
班级	1903104	学号	1190201421	姓名	张瑞

```
mysql> SELECT Sno FROM SC WHERE Grade IS NULL;
+-----+
| Sno   |
+-----+
| MA-001 |
+-----+
1 row in set (0.00 sec)
```

查询计算机系学生的最大年龄:

```
mysql> SELECT MAX(Sage) FROM Student WHERE Sdept = 'CS';
+-----+
| MAX(Sage) |
+-----+
| 19         |
+-----+
1 row in set (0.01 sec)
```

查询选修了 2 门以上课程的学生的学号和选课数:

```
mysql> SELECT Sno, COUNT(*) FROM SC GROUP BY Sno HAVING COUNT(*) >= 2;
+-----+-----+
| Sno   | COUNT(*) |
+-----+-----+
| CS-001 | 2         |
| PH-001 | 3         |
+-----+-----+
2 rows in set (0.00 sec)
```

查询学生及其选课情况:

```
mysql> SELECT Student.Sno, Sname, Cno, Grade FROM Student JOIN SC USING (Sno);
+-----+-----+-----+-----+
| Sno   | Sname | Cno   | Grade |
+-----+-----+-----+-----+
| CS-001 | Elsa  | 1002  | 95    |
| CS-001 | Elsa  | 3006  | 90    |
| CS-002 | Ed    | 3006  | 80    |
| MA-001 | Abby  | 1002  | NULL  |
| PH-001 | Nick  | 1002  | 92    |
| PH-001 | Nick  | 2003  | 85    |
| PH-001 | Nick  | 3006  | 88    |
+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

查询年龄最大的学生的学号:

```
mysql> SELECT Sno FROM Student WHERE Sage = (SELECT MAX(Sage) FROM Student);
+-----+
| Sno   |
+-----+
| PH-001 |
+-----+
1 row in set (0.00 sec)
```

查询和 Elsa 在同一个系学习的学生的学号和姓名:

实验题目	关系数据库管理系统与 SQL			实验日期	2022 年 4 月 3 日
班级	1903104	学号	1190201421	姓名	张瑞

```
mysql> SELECT Sno, Sname FROM Student WHERE Sdept = (SELECT Sdept FROM Student
WHERE Sname = 'Elsa');
+-----+-----+
| Sno   | Sname |
+-----+-----+
| CS-001 | Elsa  |
| CS-002 | Ed    |
+-----+-----+
2 rows in set (0.00 sec)
```

5. Product 数据集上的查询

(a) Find the manufacturers that sell laptops but not PC's. (使用集合差运算)

由于 MySQL 不支持集合差 (MINUS/EXCEPT) 的查询, 此处用外连接实现。

```
mysql> SELECT DISTINCT T1.Maker FROM (SELECT Maker FROM Product WHERE Type = 'laptop') AS
T1 NATURAL LEFT JOIN (SELECT Maker FROM Product WHERE Type = 'pc') AS T2 WHERE T2.Maker
IS NULL;
+-----+
| Maker |
+-----+
| F     |
| G     |
+-----+
2 rows in set (0.00 sec)
```

(b) Find the manufacturers that sell laptops but not PC's. (使用含有 IN 的嵌套查询)

```
mysql> SELECT DISTINCT Maker FROM Product WHERE Maker IN (SELECT Maker FROM Pro
duct WHERE Type = 'laptop') AND Maker NOT IN (SELECT Maker FROM Product WHERE T
ype = 'pc');
+-----+
| Maker |
+-----+
| F     |
| G     |
+-----+
2 rows in set (0.01 sec)
```

(c) Find the manufacturers that sell laptops but not PC's. (使用含有 EXISTS 的嵌套查询)

```
mysql> SELECT DISTINCT Maker FROM Product AS P1 WHERE P1.Type = 'laptop' AND NO
T EXISTS (SELECT * FROM Product AS P2 WHERE P2.Maker = P1.Maker AND P2.Type = '
pc');
+-----+
| Maker |
+-----+
| F     |
| G     |
+-----+
2 rows in set (0.00 sec)
```

(d) Find the model numbers of all printers that are cheaper than the printer model 3002. (使用内连接查询)

实验题目	关系数据库管理系统与 SQL			实验日期	2022 年 4 月 3 日
班级	1903104	学号	1190201421	姓名	张瑞

```
mysql> SELECT P1.Model FROM Printer AS P1 JOIN Printer AS P2 ON P1.Model != P2.Model AND P1.Price < P2.Price WHERE P2.Model = '3002';
```

Model
3001
3004
3005
3006
3007

```
5 rows in set (0.00 sec)
```

(e) Find the model numbers of all printers that are cheaper than the printer model 3002. (使用含有比较运算符的嵌套查询)

```
mysql> SELECT Model FROM Printer WHERE Price < (SELECT Price FROM Printer WHERE Model = '3002');
```

Model
3001
3004
3005
3006
3007

```
5 rows in set (0.00 sec)
```

(f) Find the model numbers of all printers that are cheaper than the printer model 3002. (使用含有 EXISTS 的嵌套查询)

```
mysql> SELECT Model FROM Printer AS P1 WHERE EXISTS (SELECT * FROM Printer AS P2 WHERE P2.Model = '3002' AND P1.Price < P2.Price);
```

Model
3001
3004
3005
3006
3007

```
5 rows in set (0.00 sec)
```

(g) Find the PC model with the highest available speed. (使用外连接查询)

```
mysql> SELECT PC1.Model FROM PC AS PC1 LEFT JOIN PC AS PC2 ON (PC1.Speed < PC2.Speed) WHERE PC2.Model IS NULL;
```

Model
1005
1006

```
2 rows in set (0.00 sec)
```

(h) Find the PC model with the highest available speed. (使用含有 IN 的嵌套查询)

实验题目	关系数据库管理系统与 SQL			实验日期	2022 年 4 月 3 日
班级	1903104	学号	1190201421	姓名	张瑞

```
mysql> SELECT Model FROM PC WHERE Speed IN (SELECT MAX(Speed) FROM PC);
+-----+
| Model |
+-----+
| 1005  |
| 1006  |
+-----+
2 rows in set (0.00 sec)
```

(i) Find the PC model with the highest available speed. (使用含有=的嵌套查询)

```
mysql> SELECT Model FROM PC WHERE Speed = (SELECT MAX(Speed) FROM PC);
+-----+
| Model |
+-----+
| 1005  |
| 1006  |
+-----+
2 rows in set (0.00 sec)
```

(j) Find the PC model with the highest available speed. (使用含有>=的嵌套查询)

```
mysql> SELECT Model FROM PC WHERE Speed >= ALL (SELECT Speed FROM PC);
+-----+
| Model |
+-----+
| 1005  |
| 1006  |
+-----+
2 rows in set (0.00 sec)
```

(k) Find the PC model with the highest available speed. (使用含有 EXISTS 的嵌套查询)

```
mysql> SELECT Model FROM PC AS PC1 WHERE NOT EXISTS (SELECT * FROM PC AS PC2 WHERE PC2.Speed > PC1.Speed);
+-----+
| Model |
+-----+
| 1005  |
| 1006  |
+-----+
2 rows in set (0.00 sec)
```

(l) Find the manufacturers of PC's with at least three different speeds. (使用内连接查询)

```
mysql> WITH T AS (SELECT Maker, Speed FROM Product NATURAL JOIN PC) SELECT DISTINCT T1.Maker FROM T AS T1 JOIN T AS T2 ON T1.Maker = T2.Maker AND T1.Speed != T2.Speed JOIN T AS T3 ON T1.Maker = T3.Maker AND T1.Speed != T3.Speed AND T2.Speed != T3.Speed;
+-----+
| Maker |
+-----+
| A     |
| D     |
| E     |
+-----+
3 rows in set (0.00 sec)
```

(m) Find the manufacturers of PC's with at least three different speeds. (使用分组查询)

实验题目	关系数据库管理系统与 SQL			实验日期	2022 年 4 月 3 日
班级	1903104	学号	1190201421	姓名	张瑞

```
mysql> SELECT Maker FROM PC NATURAL JOIN Product GROUP BY Maker HAVING COUNT(DISTINCT Speed) >= 3;
+-----+
| Maker |
+-----+
| A     |
| D     |
| E     |
+-----+
3 rows in set (0.00 sec)
```

(n) Find the manufacturers of PC's with at least three different speeds. (使用派生关系)

```
mysql> SELECT Maker FROM (SELECT Maker, COUNT(DISTINCT Speed) AS Amt FROM PC NATURAL JOIN Product GROUP BY Maker) AS T WHERE T.Amt >= 3;
+-----+
| Maker |
+-----+
| A     |
| D     |
| E     |
+-----+
3 rows in set (0.00 sec)
```

(o) Decrease the price of all PC's made by maker A by 10%. (使用含有=的更新条件)

```
mysql> UPDATE PC SET Price = Price * 0.9 WHERE Model = ANY (SELECT Model FROM Product WHERE Maker = 'A');
Query OK, 3 rows affected (0.01 sec)
Rows matched: 3  Changed: 3  Warnings: 0
```

(p) Decrease the price of all PC's made by maker A by 10%. (使用含有 IN 的更新条件)

```
mysql> UPDATE PC SET Price = Price * 0.9 WHERE Model IN (SELECT Model FROM Product WHERE Maker = 'A');
Query OK, 3 rows affected (0.01 sec)
Rows matched: 3  Changed: 3  Warnings: 0
```

(q) Decrease the price of all PC's made by maker A by 10%. (使用含有 EXISTS 的更新条件)

```
mysql> UPDATE PC SET Price = Price * 0.9 WHERE EXISTS (SELECT * FROM Product WHERE PC.Model = Product.Model AND Product.Maker = 'A');
Query OK, 3 rows affected (0.01 sec)
Rows matched: 3  Changed: 3  Warnings: 0
```

(r) 题目(g)–(k)用不同方法编写相同的查询。请你从 SQL 语句的易读性和执行效率两方面对题目(g)–(k)的 SQL 语句进行分析和比较。在做效率分析时，我们假定每个关系上只有主索引，而没有其他索引(请自学第 6 章中索引的概念和功能)。

易读性：(i)的易读性最好，符合直接查找最大值的一般思路，(h)和(j)的查询语句较短，查询思路也和(i)类似，因此易读性也较高，(g)和(k)的查询语句较长，而且是通过两两比较的方式选出最大值，与一般思路不同，易读性稍低一些。

执行效率：假定只有主索引，则 PC 关系上的主索引只能是 Model，在 Speed 属性上没有索引，各个查询都较慢。假设 PC 关系中有 n 个元组，(g)在内连接时复杂度为 $O(n^2)$ ，之后还要在 PC 上进行一次遍历；(k)是包含相关子查询的嵌套查询，复杂度也为 $O(n^2)$ ；(j)虽然是包含不相关子查询的嵌套查询，但其子查询结果包含多个值（可近似看成 n 个），则整个查询的复杂度大概也为 $O(n^2)$ 。(h)和(i)是包含不相关子查询的嵌套查询，且子查询结果只

实验题目	关系数据库管理系统与 SQL			实验日期	2022 年 4 月 3 日
班级	1903104	学号	1190201421	姓名	张瑞

有一个值，整个查询在 PC 上遍历两次即可完成，复杂度可近似认为是 $O(n)$ 。故可认为(g)、(j)和(k)的查询效率低，(h)和(i)的查询效率稍高些。

四、实验结论（总结实验发现及结论）

数据库管理系统是一种有效的数据管理方式，具有便于操作、安全、访问效率高等优点。SQL 是创建、修改、查询和控制数据库的语言工具，有数据定义、数据更新和数据查询等三大功能。

虽然在不同的数据库管理系统上，SQL 语言会有一定的差异，但总体来说差异并不大，它们的语法高度相似并能实现几乎一致的功能。

在 SQL 中，经常使用到数据查询相关的语句，合理选择查询方式（单关系查询、连接查询或嵌套查询）可以实现不同的查询需求。如果要提高查询效率，可以结合索引相关知识，利用索引完成更高效的查询。当然，编写易读性更好的查询语句也很重要，可以更清晰地展示出查询目的。