

实验题目	关系数据库管理系统与 SQL			实验日期	2022. 4. 3
班级	1903103	学号	1190200208	姓名	李旻翀

# 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）

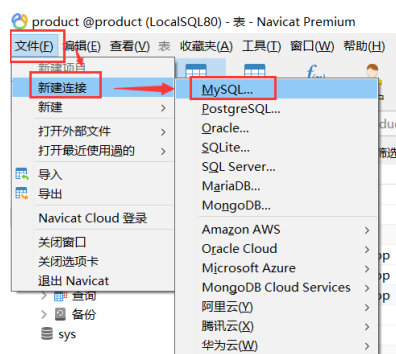
系统: Windows 10  
数据库: MySQL 8.0.28  
软件: Navicat

### 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. 连接数据库

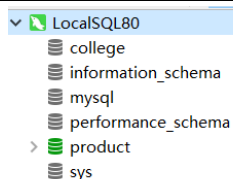
因为我在此次实验中采用 MySQL 数据库，故此处讲解如何将 navicat 连接至 MySQL 数据库。

如图所示，点击文件→连接→新建连接，输入数据库端口和密码即可。



连接完成后，效果如下：

实验题目	关系数据库管理系统与 SQL			实验日期	2022. 4. 3
班级	1903103	学号	1190200208	姓名	李旻翀



2. 练习使用 2.2 节介绍的 `gsqI` 命令  
因为 MySQL 数据库就在本地，使用我没有用到类似的功能，该部分暂且省略。

3. 练习使用 2.3 节介绍的实用 SQL 命令  
MySQL 数据库中的实用命令如下：  
`create database XXX`: 创建数据库  
`drop database XXX`: 删除数据库  
`alter database XXX ...`: 对数据库进行修改  
`show databases`: 列出现有所有数据库的名字

4. 创建 College 数据库  
按如图所示步骤创建 `college` 数据库：

```
mysql> use college
Database changed
mysql> create table Student(
  -> sno char(6) primary key,
  -> `C`
mysql> create table Student(
  -> Sno char(6) primary key,
  -> Sname varchar(10) not null,
  -> Ssex char check (Ssex in ('M','F')),
  -> Sage int check (Sage>0),
  -> Sdept varchar(20)
  -> );
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> insert into Student
  -> values
  -> ('PH-001', 'Nick', 'M', 20, 'Physics'),
  -> ('CS-001', 'Elsa', 'F', 19, 'CS'),
  -> ('CS-002', 'Ed', 'M', 19, 'CS'),
  -> ('MA-001', 'Abby', 'F', 18, 'Math'),
  -> ('MA-002', 'Cindy', 'F', 19, 'Math');
Query OK, 5 rows affected (0.02 sec)
Records: 5 Duplicates: 0 Warnings: 0
```

```
mysql> create table Course(
  -> Cno char(4) primary key
  -> );
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> insert into Course
  -> values
  -> ('1002'), ('2003'), ('3006');
Query OK, 3 rows affected (0.01 sec)
Records: 3 Duplicates: 0 Warnings: 0
```

实验题目	关系数据库管理系统与 SQL			实验日期	2022. 4. 3
班级	1903103	学号	1190200208	姓名	李旻翀

```
mysql> create table SC(
  -> Sno char(6),
  -> Cno char(4),
  -> Grade int,
  -> primary key (Sno, Cno),
  -> foreign key (Sno) references Student(Sno),
  -> foreign key (Cno) references Course(Cno)
  -> );
Query OK, 0 rows affected (0.04 sec)

mysql> insert into SC
  -> values
  -> ('PH-001', '1002', 92),
  -> ('PH-001', '2003', 85),
  -> ('PH-001', '3006', 88),
  -> ('CS-001', '1002', 95),
  -> ('CS-001', '3006', 90),
  -> ('CS-002', '3006', 80),
  -> ('MA-001', '1002', NULL);
Query OK, 7 rows affected (0.01 sec)
Records: 7  Duplicates: 0  Warnings: 0
```

5. 验证第 3 章例子中给出的 College 数据库上的 SQL 语句的正确性  
经过验证，例子中的 SQL 语句没有问题，此处仅简单举几个例子作为示意：

```
mysql> SELECT Sno, Sname FROM Student WHERE Sname LIKE 'E%';
+-----+-----+
| Sno   | Sname |
+-----+-----+
| CS-001 | Elsa  |
| CS-002 | Ed    |
+-----+-----+
2 rows in set (0.03 sec)
```

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

```
mysql> WITH T AS
  -> (SELECT Sno, COUNT(*) AS Amt
  -> FROM SC GROUP BY Sno)
  -> SELECT Sno, Amt FROM T
  -> WHERE Amt >= 2;
+-----+-----+
| Sno   | Amt |
+-----+-----+
| CS-001 | 2   |
| PH-001 | 3   |
+-----+-----+
2 rows in set (0.01 sec)
```

实验题目	关系数据库管理系统与 SQL			实验日期	2022. 4. 3
班级	1903103	学号	1190200208	姓名	李旻翀

6. 编写本课程第 3 章习题 11 中的全部数据库查询和更新语句  
首先在 MySQL 数据库中导入 SQL 脚本，操作如下：

```
mysql> create database product;
Query OK, 1 row affected (0.02 sec)

mysql> use product;
Database changed
mysql> source C:/UserData/Desktop/product.sql
Query OK, 0 rows affected (0.04 sec)

Query OK, 30 rows affected (0.00 sec)
Records: 30 Duplicates: 0 Warnings: 0

Query OK, 0 rows affected (0.01 sec)

Query OK, 13 rows affected (0.00 sec)
Records: 13 Duplicates: 0 Warnings: 0

Query OK, 0 rows affected (0.01 sec)

Query OK, 10 rows affected (0.00 sec)
Records: 10 Duplicates: 0 Warnings: 0

Query OK, 0 rows affected (0.02 sec)

Query OK, 7 rows affected (0.00 sec)
Records: 7 Duplicates: 0 Warnings: 0
```

Find the manufacturers that sell laptops but not PC' s.

(a)使用集合差运算

```
mysql> select distinct maker from product where type='laptop'
-> minus
-> select distinct maker from product where type='PC';
```

由于 MySQL 中没有差操作，所以该语句实际上无法执行。

(b)使用含有 IN 的嵌套查询

```
mysql> select distinct maker from product where maker in (select maker from product where type='laptop'
-> and maker not in (select maker from product where type='PC');

+-----+
| maker |
+-----+
| F     |
| G     |
+-----+
2 rows in set (0.00 sec)
```

(c) 使用含有 EXISTS 的嵌套查询

```
mysql> select distinct maker from product as p1
-> where p1.type='laptop' and not exists
-> (select * from product as p2 where p1.maker=p2.maker and type='PC');

+-----+
| maker |
+-----+
| F     |
| G     |
+-----+
2 rows in set (0.00 sec)
```

实验题目	关系数据库管理系统与 SQL			实验日期	2022. 4. 3
班级	1903103	学号	1190200208	姓名	李旻翀

Find the model numbers of all printers that are cheaper than the printer model 3002.

(d) 使用内连接查询

```
mysql> select p2.model from printer as p1 join printer as p2 on
-> (p1.model=3002 and p2.price<p1.price);
```

model
3001
3004
3005
3006
3007

5 rows in set (0.01 sec)

(e) 使用含有比较运算符的嵌套查询

```
mysql> select model from printer where price < all
-> (select price from printer where model=3002);
```

model
3001
3004
3005
3006
3007

5 rows in set (0.00 sec)

(f) 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 (0.01 sec)

Find the PC model with the highest available speed.

(g) 使用外连接查询

实验题目	关系数据库管理系统与 SQL			实验日期	2022. 4. 3
班级	1903103	学号	1190200208	姓名	李旻翀

```
mysql> select p1.model from pc as p1 left join pc as p2 on (p1.speed<p2.speed)
-> where p2.speed is null;
+-----+
| model |
+-----+
| 1005  |
| 1006  |
+-----+
2 rows in set (0.02 sec)
```

(h)使用含有 IN 的嵌套查询

```
mysql> select model from pc where speed in (select max(speed) from pc);
+-----+
| model |
+-----+
| 1005  |
| 1006  |
+-----+
2 rows in set (0.01 sec)
```

(i)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 (0.01 sec)
```

(j)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)使用含有 EXISTS 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. 4. 3
班级	1903103	学号	1190200208	姓名	李旻翀

```
mysql> select model from pc as p1 where not exists
-> (select * from pc as p2 where p2.speed>p1.speed);
+-----+
| model |
+-----+
| 1005  |
| 1006  |
+-----+
2 rows in set (0.00 sec)
```

Find the manufacturers of PC' s with at least three different speeds.

注：此处的(l)，(m)，(n)均为同一条查询语句，该查询语句同时用到了内连接，分组与派生。

(l)使用内连接查询

```
mysql> with temp as
-> (select maker,count(*) as amt from (select distinct maker,speed from product join pc on (product.type='pc' and product.model=pc.model)) as T group by maker)
-> select maker from temp where amt>=3;
+-----+
| maker |
+-----+
| A      |
| D      |
| E      |
+-----+
3 rows in set (0.00 sec)
```

(m)使用分组查询

```
mysql> with temp as
-> (select maker,count(*) as amt from (select distinct maker,speed from product join pc on (product.type='pc' and product.model=pc.model)) as T group by maker)
-> select maker from temp where amt>=3;
+-----+
| maker |
+-----+
| A      |
| D      |
| E      |
+-----+
3 rows in set (0.00 sec)
```

(n)使用派生关系

```
mysql> with temp as
-> (select maker,count(*) as amt from (select distinct maker,speed from product join pc on (product.type='pc' and product.model=pc.model)) as T group by maker)
-> select maker from temp where amt>=3;
+-----+
| maker |
+-----+
| A      |
| D      |
| E      |
+-----+
3 rows in set (0.00 sec)
```

Decrease the price of all PC' s made by maker A by 10%.

注：(o)，(p)的语句相同，因为该语句同时使用了=与 IN。

(o)使用含有=的更新条件

```
mysql> update pc set price = 0.9 * price where model in
-> (select model from product where maker='A' and type='PC');
Query OK, 3 rows affected (0.08 sec)
Rows matched: 3  Changed: 3  Warnings: 0
```

(p)使用含有 IN 的更新条件

实验题目	关系数据库管理系统与 SQL			实验日期	2022. 4. 3
班级	1903103	学号	1190200208	姓名	李旻翀

```
mysql> update pc set price = 0.9 * price where model in
-> (select model from product where maker='A' and type='PC');
Query OK, 3 rows affected (0.08 sec)
Rows matched: 3  Changed: 3  Warnings: 0
```

(q) 使用含有 EXISTS 的更新条件

```
mysql> update pc set price=0.9*price where exists
-> (select * from product where
-> product.maker='A' and
-> product.type='pc' and
-> product.model=pc.model);
Query OK, 3 rows affected (0.01 sec)
Rows matched: 3  Changed: 3  Warnings: 0
```

(r) 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  
易读性:

就易读性而言, 使用 IN, >=, = 的语句易读性最高, 使用外连接和 EXISTS 语句的查询易读性稍差, 因为涉及重命名, 以及多个表之间的连接和比较操作。

执行效率:

从执行效率来看, 使用 EXISTS 语句的执行效率最高, 耗时最短, 而使用外连接的执行效率最低。这可能是因为外连接操作时涉及到每个元组的条件判断, 需要进行的判断次数至少为  $m \times n$  次 ( $m, n$  为进行外连接的两张表的大小)。另外, 查询资料得知, IN 语句在查询时会将表的数据缓存, 因此不适用于大型表的查询。而 EXISTS 则无缓存操作, 因此更适合大型表的查询。

#### 四、实验结论 (总结实验发现及结论)

通过此次实验, 熟悉了 MySQL 数据库的基本操作, 能够熟练地进行数据库的创建, 数据的增删改查等操作。