
哈尔滨工业大学

<<数据库系统>>

实验报告一

(2023 年度春季学期)

姓名:	刘天瑞
学号:	7203610121
学院:	计算学部
教师:	李东博

实验一

一、实验目的

掌握 MySQL 关系数据库管理系统的基本命令，并熟练使用 SQL 语言管理 MySQL 数据库。掌握 SQL 语言的使用方法，学会使用 SQL 语言进行关系数据库查询，特别是聚集查询、连接查询和嵌套查询。

二、实验环境

Windows 11 操作系统、mysql-8.0.32-winx64

三、实验过程及结果

（一）准备数据

使用命令行方式：

1、首先登录账号，再创建数据库 COMPANY，然后查看数据库是否创建成功：

```
C:\Users\刘天瑞>mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 13
Server version: 8.0.32 MySQL Community Server - GPL

Copyright (c) 2000, 2023, 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.

mysql> create database company;
Query OK, 1 row affected (0.01 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| company  |
| information_schema |
| mysql    |
| performance_schema |
| sys      |
+-----+
5 rows in set (0.01 sec)

mysql> |
```

2、接着使用数据库 COMPANY，然后在该数据库中分别创建表：EMPLOYEE、DEPARTMENT、PROJECT 以及 WORKS_ON:

```
mysql> use company;
Database changed
mysql> create table employee(
    ->  ename varchar(12),
    ->  essn char(18),
    ->  address text,
    ->  salary int,
    ->  superssn char(18),
    ->  dno char(3),
    ->  primary key(essn));
Query OK, 0 rows affected (0.02 sec)

mysql> show tables;
+-----+
| Tables_in_company |
+-----+
| employee           |
+-----+
1 row in set (0.01 sec)

mysql> describe employee;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| ename  | varchar(12) | YES  |     | NULL    |       |
| essn   | char(18)    | NO   | PRI | NULL    |       |
| address | text        | YES  |     | NULL    |       |
| salary | int         | YES  |     | NULL    |       |
| superssn | char(18)   | YES  |     | NULL    |       |
| dno    | char(3)     | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

```
mysql> create table department(
  -> dname varchar(15),
  -> dno char(3),
  -> mgrssn char(18),
  -> mgrstartdate date,
  -> primary key(dno));
Query OK, 0 rows affected (0.02 sec)

mysql> create table project(
  -> pname varchar(3),
  -> pno char(2),
  -> plocation text,
  -> dno char(3),
  -> primary key(pno));
Query OK, 0 rows affected (0.02 sec)

mysql> create table works_on(
  -> essn char(18),
  -> pno char(12),
  -> hours int,
  -> primary key(essn,pno));
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'hours int,primary key(essn,pno))' at line 3
mysql> create table works_on(
  -> essn char(18),
  -> pno char(12),
  -> hours int,
  -> primary key(essn,pno));
Query OK, 0 rows affected (0.02 sec)

mysql> show tables;
+-----+
| Tables_in_company |
+-----+
| department         |
| employee            |
| project             |
| works_on            |
+-----+
4 rows in set (0.00 sec)
```

3、导入 txt 文件到数据库中（注意：首先需要通过本地上传文件模式登录 mysql 账号，然后需要手动将 local_infile 选项设置为开启状态，原因是 MySQL 默认不允许用户从本地载入文件，之后别忘了再查询是否设置成功。并且导入文件前需要使用该数据库将事先准备的表格清空）：

```
C:\Users\刘天瑞>mysql --local-infile=1 -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 14
Server version: 8.0.32 MySQL Community Server - GPL

Copyright (c) 2000, 2023, 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.

mysql> show global variables like "local_infile";
+-----+-----+
| Variable_name | Value |
+-----+-----+
| local_infile  | OFF   |
+-----+-----+
1 row in set, 1 warning (0.01 sec)

mysql> set global local_infile=true;
Query OK, 0 rows affected (0.00 sec)

mysql> show global variables like "local_infile";
+-----+-----+
| Variable_name | Value |
+-----+-----+
| local_infile  | ON    |
+-----+-----+
1 row in set, 1 warning (0.00 sec)

mysql> |
```

```
mysql> use company;
Database changed
mysql> delete from employee;
Query OK, 0 rows affected (0.00 sec)

mysql> delete from department;
Query OK, 0 rows affected (0.00 sec)

mysql> delete from project;
Query OK, 0 rows affected (0.00 sec)

mysql> delete from works_on;
Query OK, 0 rows affected (0.00 sec)

mysql> load data local infile "D:/dbms_lab/employee.txt" into table employee;
Query OK, 50 rows affected (0.01 sec)
Records: 50 Deleted: 0 Skipped: 0 Warnings: 0

mysql> load data local infile "D:/dbms_lab/department.txt" into table department;
Query OK, 5 rows affected, 4 warnings (0.01 sec)
Records: 5 Deleted: 0 Skipped: 0 Warnings: 4

mysql> load data local infile "D:/dbms_lab/project.txt" into table project;
Query OK, 10 rows affected (0.01 sec)
Records: 10 Deleted: 0 Skipped: 0 Warnings: 0

mysql> load data local infile "D:/dbms_lab/works_on.txt" into table works_on;
Query OK, 109 rows affected (0.01 sec)
Records: 109 Deleted: 0 Skipped: 0 Warnings: 0
```

4、设置外键，employee 关系中 dno 为外键，project 关系中 dno 为外键，都对应 department 关系中的主键 dno；works_on 关系中外键 essn 对应 employee 关系中的主键 essn，外键 pno 对应 project 关系中的主键 pno。（这一步骤可以不用写，后面的查询任务有包括）

（二）使用 SQL 语言完成如下查询：

1、参加了项目名为“SQL Project”的员工名字：

（1）ENAME 在 employee 中；PNAME 在 project 中；

（2）每个员工只属于一个部门；员工可以参加其他部门的项目；

（3）参加项目的员工的工作时间>0；工作时间=0 的没参加项目；

```
mysql> select ename from employee
-> where essn in(
-> select essn from project,works_on
-> where project.pno=works_on.pno and pname="SQL" and hours>0);
```

ename
王大一
王小二
王三
王四
王五
王六
王七
王十
孙小二

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

2、在“Research Department”（研发部）工作且工资低于 3000 元的员工名字和地址：

DNAME 在 department 中；SALARY/ENAME/ADDRESS 在 employee 中；

```
mysql> select ename,address from employee,department
-> where employee.dno=department.dno and dname="研发部" and salary<3000;
```

ename	address
孙四	孙村
孙五	孙村
孙六	孙村
孙张红十	孙村

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

3、没有参加项目编号为 P1 的项目的员工姓名：

- (1) PNO 在 works_on 中; ENAME 在 employee 中;
- (2) 没有参加项目编号 P1 的员工 NOT IN 参加了项目 P1 的员工;
- (3) 参加项目的员工的工作时间>0; 工作时间=0 的没参加项目;
- (4) 用主键索引数据记录;

```
mysql> select ename from employee
-> where essn not in(
-> select essn from works_on
-> where pno="P1" and hours>0);
```

ename
王八
王九
张大一
张小二
张三
张四
张五
张六
张七
张八
张九
张红
李大一
李小二
李三
李四
李五
李六
李七
李八
李九
李十
刘大一
刘小二
刘三
刘四
刘五
刘六
刘七
刘八
刘九
刘十
孙大一
孙三
孙四
孙五
孙六
孙七
孙八
孙九
孙张红十

4、由张红领导的工作人员的姓名和所在部门的名字:

ENAME 在 employee 中；DNAME 在 department 中用主键索引数据记录；

```
mysql> select ename,dname from employee,department
-> where employee.dno=department.dno and superssn in(
-> select essn from employee
-> where ename="张红");
```

ename	dname
张大一	甲类二车间
张小二	甲类二车间
张三	甲类二车间
张四	甲类二车间
张五	甲类二车间
张六	甲类二车间
张七	甲类二车间
张八	甲类二车间
张九	甲类二车间
张红	甲类二车间

10 rows in set (0.00 sec)

5、至少参加了项目编号为 P1 和 P2 的项目的员工号：

- (1) PNO/ESSN 在 works_on 中；
- (2) 参加项目的员工的工作时间>0；工作时间=0 的没参加项目 DISTINCT；
- (3) 去掉重复记录“至少参加……”等价于“既参加……又参加……”；
- (4) 第二种方法要注意列名模糊不清；


```
mysql> select essn from works_on
-> where pno="P1" and hours>0 and essn in(
-> select essn from works_on
-> where pno="P2" and hours>0);
+-----+
| essn |
+-----+
| 131181199901012113 |
| 131181199901022113 |
| 131181199901032113 |
| 131181199901042113 |
| 131181199901052113 |
| 131181199901062113 |
| 131181199901072113 |
| 131181199905022153 |
+-----+
8 rows in set (0.00 sec)

mysql> select w1.essn from works_on as w1,works_on as w2
-> where w1.essn=w2.essn and w1.pno="P1" and w1.hours>0 and w2.pno="P2" and w2.hours>0;
+-----+
| essn |
+-----+
| 131181199901012113 |
| 131181199901022113 |
| 131181199901032113 |
| 131181199901042113 |
| 131181199901052113 |
| 131181199901062113 |
| 131181199901072113 |
| 131181199905022153 |
+-----+
8 rows in set (0.00 sec)
```

6、参加了全部项目的员工号码和姓名：

- (1) ESSN 在 works_on 中；ENAME 在 employee 中；
- (2) “参加全部项目”等价于“不存在有一个项目该员工没参加”；
- (3) “参加全部项目”等价于“该员工参加的项目为项目总数”；

```
mysql> select essn,ename from employee
-> where not exists(
-> select pno from project
-> where not exists(
-> select * from works_on
-> where works_on.pno=project.pno and works_on.essn=employee.essn));
+-----+-----+
| essn | ename |
+-----+-----+
| 131181199905022153 | 孙小二 |
+-----+-----+
1 row in set (0.00 sec)

mysql> select essn,ename from employee
-> where essn in(
-> select essn from works_on
-> group by essn having count(*) in(
-> select count(*) from project));
+-----+-----+
| essn | ename |
+-----+-----+
| 131181199905022153 | 孙小二 |
+-----+-----+
1 row in set (0.00 sec)
```

7、员工平均工资低于 3000 元的部门名称：

SALARY 在 employee 中；DNAME 在 department 中；

```
mysql> select dname from department
-> where dno in(
-> select dno from employee
-> group by dno having avg(salary)<3000);
+-----+
| dname  |
+-----+
| 乙类二车间 |
+-----+
1 row in set (0.00 sec)
```

8、至少参与了 3 个项目且工作总时间不超过 8 小时的员工名字：

PNO/HOURS 在 works_on 中；ENAME 在 employee 中；

```
mysql> select ename from employee
-> where essn in(
-> select essn from works_on
-> group by essn having count(pno)>=3 and sum(hours)<=8);
+-----+
| ename  |
+-----+
| 王大一 |
| 王小二 |
| 王四   |
| 王六   |
| 王十   |
| 李大一 |
| 李七   |
| 李八   |
| 李十   |
+-----+
9 rows in set (0.00 sec)
```

9、每个部门的员工小时平均工资：

- (1) HOURS 在 works_on 中；DNO/SALARY 在 employee 中；
- (2) 每个部门的员工小时平均工资=部门总工资÷部门总工作时长；
- (3) 注意临时表的使用；
- (4) 注意列名模糊不清；

```
mysql> select sums.dno,sumsalary/sumhours as hoursavgsalary
-> from (select dno,sum(salary) as sumsalary from employee group by dno) as sums,(s
elect dno,sum(hours) as sumhours from works_on join employee on works_on.essn = employee
e.essn group by dno) as sumh
-> where sums.dno=sumh.dno;
+-----+-----+
| dno | hoursavgsalary |
+-----+-----+
| A01 | 439.0244 |
| A02 | 652.1739 |
| B01 | 685.3933 |
| B02 | 638.7097 |
| C10 | 858.9153 |
+-----+-----+
5 rows in set (0.00 sec)
```

10、查询所有干过非自己部门任务的员工的姓名和小时平均工资

```
mysql> select ename, salary/sumhours as hoursavgsalary
-> from employee, (select essn, sum(hours) as sumhours from works_on group by essn) as sumh
-> where employee.essn in (select essn from works_on, project
-> where employee.essn = sumh.essn
-> and employee.dno != project.dno
-> and employee.essn = works_on.essn
-> and works_on.pno = project.pno);
+-----+-----+
| ename | hoursavgsalary |
+-----+-----+
| 孙小二 | 444.4444 |
+-----+-----+
1 row in set (0.01 sec)
```

11、查询小时平均工资超过 1000 的员工姓名和小时平均工资

```
mysql> select ename, salary/sumhours as hoursavgsalary
-> from employee, (select essn, sum(hours) as sumhours from works_on group by essn) as sumh
-> where employee.essn = sumh.essn and salary/sumhours > 1000;
+-----+-----+
| ename | hoursavgsalary |
+-----+-----+
| 王大一 | 1166.6667 |
| 李大一 | 1250.0000 |
| 孙三 | 1428.5714 |
| 孙七 | 2000.0000 |
| 孙八 | 1500.0000 |
| 孙九 | 1150.0000 |
+-----+-----+
6 rows in set, 2 warnings (0.00 sec)
```

四、实验心得

列举遇到并解决的问题等。

答：通过本次数据库的第一次实验，我开始逐渐使用并且熟悉了 SQL 语言格式规范，其次我还意外地了解到数据库可视化工具 MySQL workbench 的一些应用方法，这也使得我对关系数据库有了进一步的认识与理解，同时在本次实验中，我也遇到了如下的一些困难及其解决办法：

问题一：在我利用命令行方式将本地 txt 文件导入到数据库时，报错如下：
ERROR 1148 (42000): The used command is not allowed with this MySQL version,
解决方法是，登录 local_infile 模式 MySQL 的账号，然后输入 set global

local_infile=on, 将 local_infile 属性设置为 on, 接着再进行本地 txt 文件的导入, 最后成功导入。

问题二: 在进行文件导入的过程中, 我因为疏忽过于天真地直接将 windows 资源管理器上的文件路径直接复制粘贴到命令行上, 报错。解决方法是将路径里的左斜线更改为右斜线, 问题解决。

问题一: 在我额外尝试利用 MySQL workbench 来导入数据时, 刚开始导入失败了, 原因是 MySQL workbench 并不支持 txt 文件的导入, 需要先将 txt 格式文件转换为 csv 格式文件, 可继续; 但是在导入具有中文内容关系时, 又出现报错, 这是由于编码格式不同造成的, 通过 VScode 将编码转化为 GB2312 格式即可成功导入。