

# 哈尔滨工业大学

## <<数据库系统>>

### 实验报告一

(2024 年度秋季学期)

姓名:	许坤彬
学号:	2022113586
学院:	计算学部
教师:	张浩

## 实验一

### 一、实验目的

### 二、实验环境

### 三、实验过程及结果

创建数据库 company，创建各张数据表，使用下面的 SQL 语句。

```
-- 创建数据库
CREATE DATABASE company;
use company;

-- 创建表
employee CREATE TABLE employee(
    ename VARCHAR(10),
    essn NCHAR(18) PRIMARY KEY ,
    address VARCHAR(30),
    salary INT,
    superssn NCHAR(18),
    dno VARCHAR(3)
);

-- 创建表
department CREATE TABLE department(
    dname VARCHAR(20),
    dno VARCHAR(3) PRIMARY KEY,
    mgrssn NCHAR(18),
    mrgstartdate DATE
);

-- 创建表
project CREATE TABLE project(
    pname VARCHAR(20),
    pno VARCHAR(3) PRIMARY KEY,
    plocation VARCHAR(20),
```

```
    dno VARCHAR(3)
);
-- 创建表
works_on CREATE TABLE works_on (
    essn VARCHAR(18),
    pno VARCHAR(3),
    hours INT PRIMARY KEY (essn, pno)
);
```

接下来使用 python，利用 pymysql 包完成与数据库的连接，随后通过文件读写，将 txt 中的数据读取到内存中，再写入到数据库中，下面给出一段示例的 python 代码。

```
import pymysql

# 连接到 MySQL 数据库
conn = pymysql.connect(
    host="localhost", # 数据库主机
    user="root", # 数据库用户名
    password="751016xkb", # 数据库密码
    database="company", # 数据库名称
    charset="utf8mb4" # 使用 utf8mb4 字符集
)

# 创建游标对象
cursor = conn.cursor()

# 读取 department.txt 文件
with open('./dbms_lab/department.txt', 'r', encoding='utf-8') as file:
    for line in file:
        # 按制表符 (Tab) 分割数据
        data = line.strip().split('\t')
        dname, dno, mgrssn, mrgstartdate = data

        # 插入数据到 department 表
        sql = "INSERT INTO department (dname, dno, mgrssn, mrgstartdate)
VALUES (%s, %s, %s, %s)"
        values = (dname, dno, mgrssn, mrgstartdate)

        try:
            cursor.execute(sql, values)
        except pymysql.MySQLError as err:
            print(f"Error: {err}")
```

```
# 提交更改
conn.commit()

# 关闭游标和连接
cursor.close()
conn.close()

print("数据插入完成!")
```

完成两步操作之后，数据库就已经被成功建立并且将全部数据写入了数据库，接下来可以开始查询实验。为了方便数据库的使用和结果的即时查询，在本实验中使用 pycharm 内置的图形化数据库工具来查看数据库中各个表及相关内容。接下来列举每一个实验任务所使用的 SQL 语言和查询结果。

#### 1. 参加了项目名为“SQL”的员工名字

```
SELECT ename
FROM employee ,works_on, project
WHERE employee.essn = works_on.essn AND works_on.pno = project.pno AND
project.pname = 'SQL';
```

查询结果如下

	ename
1	王大一
2	王小二
3	王三
4	王四
5	王五
6	王六
7	王七
8	王八
9	王九
10	王十
11	孙小二

#### 2. 在“Research Department”工作且工资低于 3000 元的员工名字和地址

```
SELECT ename, address
FROM employee
JOIN department on employee.dno = department.dno
WHERE department.dname = '研发部' AND employee.salary < 3000
```

查询结果如下：

	ename	address
1	孙四	孙村
2	孙五	孙村
3	孙六	孙村
4	孙张红十	孙村

### 3. 没有参加项目编号为 P1 的项目的员工姓名

```
SELECT ename
FROM employee
WHERE essn NOT IN (
    SELECT employee.essn
    FROM employee
    JOIN works_on ON employee.essn = works_on.essn
    WHERE works_on.pno = 'P1'
)
GROUP BY essn;
```

查询结果如下

ename	
1 张大一	20 李十
2 张小二	21 刘大一
3 张三	22 刘小二
4 张四	23 刘三
5 张五	24 刘四
6 张六	25 刘五
7 张七	26 刘六
8 张八	27 刘七
9 张九	28 刘八
10 张红	29 刘九
11 李大一	30 刘十
12 李小二	31 孙大一
13 李三	32 孙三
14 李四	33 孙四
15 李五	34 孙五
16 李六	35 孙六
17 李七	36 孙七
18 李八	37 孙八
19 李九	38 孙九
	39 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

```
SELECT e1.ename, dname
FROM employee AS e1, employee as e2, department
WHERE e1.superssn = e2.essn AND e2.ename = '张红' AND department.dno =
e1.dno
```

查询结果如下：

	ename	dname
1	张大一	甲类二车间
2	张小二	甲类二车间
3	张三	甲类二车间
4	张四	甲类二车间
5	张五	甲类二车间
6	张六	甲类二车间
7	张七	甲类二车间
8	张八	甲类二车间
9	张九	甲类二车间
10	张红	甲类二车间

5. 至少参加了项目编号为 P1 和 P2 的项目的员工号

```
select w1.essn
FROM works_on AS w1, works_on AS w2
WHERE w1.essn = w2.essn AND w1.pno = 'P1' AND w2.pno = 'P2'
```

	essn
1	131181199901012113
2	131181199901022113
3	131181199901032113
4	131181199901042113
5	131181199901052113
6	131181199901062113
7	131181199901072113
8	131181199901082113
9	131181199901092113
10	131181199901102113
11	131181199905022153

6. 参加了全部项目的员工号码和姓名

```
SELECT essn, ename
FROM employee
WHERE essn IN(
    SELECT essn
    FROM works_on
    GROUP BY essn)
```

```
HAVING COUNT(works_on.pno) = (  
    SELECT COUNT(*)  
    FROM project  
    )  
);
```

对应查询结果为:

	essn	ename
1	131181199905022153	孙小二

7. 员工平均工资低于 3000 元的部门名称

```
SELECT dname  
FROM department, employee  
WHERE department.dno = employee.dno  
GROUP BY employee.dno  
HAVING AVG(employee.salary) < 3000;
```

	dname
1	乙类二车间

8. 至少参与了 3 个项目并且工作总时间不超过 8 小时的员工名字

```
SELECT employee.ename  
FROM employee  
JOIN works_on ON employee.essn = works_on.essn  
GROUP BY employee.essn  
HAVING SUM(works_on.hours) <= 8;
```

对应的查询结果为:

	ename		
1	王大一	20	刘大一
2	王小二	21	刘小二
3	王四	22	刘三
4	王六	23	刘四
5	王十	24	刘五
6	张大一	25	刘六
7	张小二	26	刘七
8	张三	27	刘八
9	张四	28	刘九
10	张五	29	刘十
11	张六	30	孙三
12	张七	31	孙四
13	张八	32	孙五
14	张九	33	孙六
15	张红	34	孙七
16	李大一	35	孙八
17	李七	36	孙九
18	李八	37	孙张红十
19	李十		

### 9. 每个部门的员工小时平均工资

```
SELECT department.dname, SUM(employee.salary) / SUM(works_on.hours)
FROM employee, works_on, department
WHERE employee.essn = works_on.essn AND employee.dno = department.dno
GROUP BY department.dno
```

	dname	SUM(employee.salary) / SUM(works_on.hours)
1	甲类一车间	1317.0732
2	甲类二车间	652.1739
3	乙类一车间	2741.5730
4	乙类二车间	638.7097
5	研发部	2079.2542

## 四、实验心得

1. SQL 语言中的字符串用 ‘ ’
2. 在需要查询一个表内的某一属性存在有两个或多个特殊值的一个属性时，可以使用换名的方法，形如第 5 个任务
3. 使用连接的方法再筛选列可能会出现重复值的情况，这时可以用 GROUP BY 来去除重复值，形如第 3 和第 6 个任务。
4. 聚合函数能出现在两个地方，一个是 select 后，另一个是 having 后



5. 对于 3 这种类型，由于一个员工可能参与不止 P1 一个项目，所以不能单纯用

```
# SELECT ename
# FROM employee
# JOIN works_on ON works_on.essn = employee.essn
# WHERE works_on.pno != 'P1'
# GROUP BY employee.essn;
```

进行实现，因为会把同时做过 P1 和其他项目的人再加回来，所以需要嵌套查询，用 NOT IN (<做过 P1 项目>) 的形式来完成