

哈尔滨工业大学

<<数据库系统>>

实验报告一

(2023 年度秋季学期)

姓名:	张智雄
学号:	2021112845
学院:	计算学部
教师:	程思瑶

实验一 MySQL 关系数据库管理系统及 SQL 语言的使用

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 关系数据库管理系统。

三、实验过程及结果

本实验创建关系数据库 COMPANY，使用 SQL 语言完成对指定内容的查询。

3.1 创建关系数据库 COMPANY

3.1.1 建立关系

创建关系数据库 COMPANY，其模式如下（下划线表示关系的主键）：

表格 1 关系的建立及 SQL 代码

关系的模式	对应的创建 SQL 代码
关系 EMPLOYEE (ENAME, <u>ESSN</u> , ADDRESS, SALARY, SUPERSSN, DNO) ENAME: 工作人员名字, ESSN: 工作人员身份证号, ADDRESS: 工作人员住址, SALARY: 工作人员工资, SUPERSSN: 工作人员直接领导的身份证号, DNO: 所属部门号	<pre>CREATE TABLE EMPLOYEE(ENAME VARCHAR(15), ESSN CHAR(20) NOT NULL, ADDRESS VARCHAR(15), SALARY INTEGER, SUPERSSN CHAR(20), DNO char(3), Primary key (ESSN), foreign key (DNO) references DEPARTMENT(DNO))charset=utf8;</pre>
关系 PROJECT (PNAME, <u>PNO</u> , PLOCATION, DNO) PNAME: 工程项目名, PNO: 工程项目号, PLOCATION: 工程项目所在地, DNO: 工程项目所属部门号	<pre>CREATE TABLE PROJECT (PNAME VARCHAR(10), PNO VARCHAR(3) NOT NULL, PLOCATION CHAR(10), DNO char(3), Primary key (PNO), foreign key (DNO) references DEPARTMENT(DNO))charset=utf8;</pre>
关系 DEPARTMENT (DNAME, <u>DNO</u> , MGRSSN, MGRSTARTDATE) DNAME: 部门名, DNO: 部门号, MGRSSN: 部门领导身份证号, MGRSTARTDATE: 部门领导开始领导工作的日期	<pre>CREATE TABLE DEPARTMENT (DNAME VARCHAR(25), DNO CHAR(3) NOT NULL, MGRSSN CHAR(20), MGRSTARTDATE DATE, Primary key (DNO))charset=utf8;</pre>
关系 WORKS_ON (<u>ESSN</u> , <u>PNO</u> , HOURS) ESSN: 工作人员身份证号, PNO: 工程项目号, HOURS: 工作小时数	<pre>CREATE TABLE WORKS_ON (ESSN CHAR(20) NOT NULL, PNO VARCHAR(3) NOT NULL, HOURS INT, Primary key (ESSN,PNO), foreign key (ESSN) references EMPLOYEE(ESSN), foreign key (PNO) references PROJECT(PNO))charset=utf8;</pre>

在创建时，上述根据实际和建立完整性约束，设置了主码与外码如下：

✧ 关系 EMPLOYEE：主码为工作人员身份证号 ESSN，所属部门号为参照 DEPARTMENT 的外码；

✧ 关系 PROJECT：主码为工程项目号 PNO，工程项目所属部门号为参照 DEPARTMENT 的外码；

✧ 关系 DEPARTMENT：主码为部门号 DNO；

✧ 关系 WORKS_ON：主码为工作人员身份证号和工程项目号 (ESSN, PNO)，同时工作人员身份证号为参照 EMPLOYEE 的外码，工程项目号为参照 PROJECT 的外码

同时，这里为了后续的文件导入数据中文信息不会出现乱码，在建立时进行了” charset=utf8 ”的字符设置。

3.1.2 导入数据

使用” LOAD DATA LOCAL INFILE “PATH” INTO TABLE <关系名> ”将文本文件中的数据导入到上述建立的关系之中。

为了避免换行符的影响，在上述 SQL 代码后加入” LINES TERMINATED BY ‘\r\n’ ”解决。导入后查看如下：

		DNAME varchar(25)	DNO char(3)	MGRSSN char(20)	MGRSTARTDAT date
	1	甲类一车间	A01	131181199901012113	2020-10-10
	2	甲类二车间	A02	131181199902102123	2023-01-25
	3	乙类一车间	B01	131181199903012133	2022-05-06
	4	乙类二车间	B02	131181199904022143	2021-09-11
	5	研发部	C10	131181199905032153	2022-06-09

图 1 关系 DEPARTMENT

		ENAME varchar(15)	ESSN char(20)	ADDRESS varchar(15)	SALARY int(11)	SUPERSSN char(20)	DNO char(3)
	1	王大一	131181199901012113	王家村	7000	131181199901012113	A01
	2	王小二	131181199901022113	王家村	5000	131181199901012113	A01
	3	王三	131181199901032113	王家村	3000	131181199901012113	A01
	4	王四	131181199901042113	王家村	3000	131181199901012113	A01
	5	王五	131181199901052113	王家村	3000	131181199901012113	A01
	6	王六	131181199901062113	王家村	3000	131181199901012113	A01
	7	王七	131181199901072113	王家村	3000	131181199901012113	A01

图 2 关系 EMPLOYEE（部分）

		PNAME varchar(10)	PNO varchar(3)	PLOCATIO char(10)	DNO char(3)			ESSN char(20)	PNO varchar(3)	HOURS int(11)
	1	SQL	P1	S市	A01		1	131181199901012113	P1	1
	2	PPA	P2	S市	A01		2	131181199901012113	P2	3
	3	D3T	P3	D市	B01		3	131181199901012113	P4	2
	4	ABZ	P4	S市	A01		4	131181199901022113	P1	2
	5	QC	P5	D市	B01		5	131181199901022113	P2	2

图 3 关系 PROJECT（右）和关系 WORKS_ON（左）

3.2 SQL 语言的使用

3.2.1 参加了项目名为”SQL Project”的员工名字

此查询通过连接三个关系（EMPLOYEE、WORKS_ON、PROJECT）后从中选择参与名为”SQL”项目的员工姓名。查询 SQL 语句如下：

```
SELECT ENAME
FROM EMPLOYEE, WORKS_ON, PROJECT
WHERE EMPLOYEE.ESSN=WORKS_ON.ESSN AND WORKS_ON.PNO=project.PNO AND PNAME='SQL';
```

查询结果截图如下：

	ENAME varchar
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

图 4 查询结果截图 1

3.2.2 在”Research Department”工作且工资低于 3000 元的员工名字和地址

根据实际，将”Research Department”改为“研发部”。此查询从 EMPLOYEE 和 DEPARTMENT 的两个关系中选择满足条件（在“研发部”工作且工资低于 3000 元）的工作人员姓名 ENAME 和地址 ADDRESS。

查询 SQL 语句如下：

```
SELECT ENAME, ADDRESS
FROM EMPLOYEE, DEPARTMENT
WHERE EMPLOYEE.DNO=DEPARTMENT.DNO AND DNAME='研发部' AND SALARY<3000;
```

查询结果截图如下：

		ENAME varchar	ADDRESS varchar
	1	孙四	孙村
	2	孙五	孙村
	3	孙六	孙村
	4	孙张红十	孙村

图 5 查询结果截图 2

3.2.3 没有参加项目编号为 P1 的项目的员工姓名

此查询使用嵌套子查询，首先选择参与了项目编号为 P1 的工作人员身份证号，然后再通过主查询选择没有在此列表中的工作人员姓名。

具体查询 SQL 语句如下：

```
SELECT ENAME
FROM EMPLOYEE
WHERE ESSN NOT IN
(SELECT ESSN FROM WORKS_ON WHERE PNO='P1');
```

查询结果截图如下：

		ENAME varchar
	1	张大一
	2	张小二
	3	张三
	4	张四
	5	张五
	6	张六
	7	张七
	8	张八
	9	张九
	10	张红
	11	李大一
	12	李小二
	25	刘五
	26	刘六
	27	刘七
	28	刘八
	29	刘九
	30	刘十
	31	孙大一
	32	孙三
	33	孙四
	34	孙五
	35	孙六
	36	孙七
	13	李三
	14	李四
	15	李五
	16	李六
	17	李七
	18	李八
	19	李九
	20	李十
	21	刘大一
	22	刘小二
	23	刘三
	24	刘四
	37	孙八
	38	孙九
	39	孙张红十

图 6 查询结果截图 3

3.2.4 由张红领导的工作人员姓名和所在部门的名称

此查询使用嵌套子查询，首先选择领导为“张红”的工作人员身份证号。而后主查询从EMPLOYEE和DEPARTMENT两个关系在上述子查询的结果列表中选择工作人员姓名ENAME和部门名称DNAME。

查询SQL语句如下：

```
SELECT ENAME, DNAME
FROM EMPLOYEE, DEPARTMENT
WHERE EMPLOYEE.DNO=DEPARTMENT.DNO AND SUPERSSN IN
(SELECT ESSN FROM EMPLOYEE WHERE ENAME='张红');
```

查询结果截图如下：



	ENAME varchar	DNAME varchar
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

图 7 查询结果截图 4

3.2.5 至少参加了项目编号为 P1 和 P2 的项目的员工号

此查询使用嵌套子查询，首先选择参加了项目编号为 P1 的工作人员身份证号。而后主查询从WORKS_ON关系在上述子查询的结果列表中再选择参加了项目编号为 P2 的工作人员身份证号ESSN。

查询SQL语句如下：

```
SELECT ESSN
FROM WORKS_ON
WHERE PNO='P1' AND ESSN IN
(SELECT ESSN FROM WORKS_ON WHERE PNO='P2');
```

查询结果截图如下：

	✓	Q	ESSN string	↕
		1	131181199901012113	
		2	131181199901022113	
		3	131181199901032113	
		4	131181199901042113	
		5	131181199901052113	
		6	131181199901062113	
		7	131181199901072113	
		8	131181199901082113	
		9	131181199901092113	
		10	131181199901102113	
		11	131181199905022153	

图 8 查询结果截图 5

3.2.6 参加了全部项目的员工号码和姓名

这个查询使用嵌套子查询和 NOT EXISTS 条件，首先在 EMPLOYEE 关系中选择所有的工作人员，然后在 PROJECT 关系中检查是否存在一个项目没有被该工作人员，如果存在这样的项目，那么这个员工就不会被包括在结果中。

查询 SQL 语句如下：

```
SELECT ESSN, ENAME
FROM EMPLOYEE
WHERE NOT EXISTS
  (SELECT PNO FROM PROJECT WHERE NOT EXISTS
    (SELECT PNO FROM WORKS_ON WHERE WORKS_ON.ESSN=EMPLOYEE.ESSN AND
    WORKS_ON.PNO=PROJECT.PNO));
```

查询结果截图如下：

	✓	Q	ESSN string	↕	ENAME varchar	↕
		1	131181199905022153		孙小二	

图 9 查询结果截图 6

3.2.7 员工平均工资低于 3000 元的部门名称

这个查询使用嵌套子查询，首先在 EMPLOYEE 关系中通过 GROUP BY 将结果按部门分组，而后使用 HAVING 子句筛选平均工资低于 3000 元的部门。而后主查询选择在上述子查询的结果列表中的部门名称 DNAME。

查询 SQL 语句如下：

```
SELECT DNAME
FROM DEPARTMENT
WHERE DNO IN
    (SELECT DNO FROM EMPLOYEE
     GROUP BY DNO
     HAVING AVG(SALARY)<3000);
```

查询结果截图如下：

<input checked="" type="checkbox"/>	Q	DNAME varchar
	1	乙类二车间

图 10 查询结果截图 7

3.2.8 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

这个查询使用嵌套子查询，首先在 **WORKS_ON** 关系中通过 **GROUP BY** 将结果按工作人员身份证号分组，而后使用 **HAVING** 子句筛选至少参与了 3 个项目且工作总时间不超过 8 小时的员工。而后主查询选择在上述子查询的结果列表中的员工名字 **ENAME**。

查询 SQL 语句如下：

```
SELECT ENAME
FROM EMPLOYEE
WHERE ESSN IN
    (SELECT ESSN FROM WORKS_ON
     GROUP BY ESSN
     HAVING COUNT(PNO)>=3 AND SUM(HOURS)<=8);
```

查询结果截图如下：

<input checked="" type="checkbox"/>	Q	ENAME varchar
	1	王大一
	2	王小二
	3	王四
	4	王六
	5	王十
	6	李大一
	7	李七
	8	李八
	9	李十

图 11 查询结果截图 8

3.2.9 每个部门的员工小时平均工资

这个查询通过将 EMPLOYEE, WORKS_ON 关系自然连接后，按部门进行分组，并计算每个部门员工时薪的均值。

查询 SQL 语句如下：

```
SELECT DNO, AVG(SALARY/HOURS)
FROM EMPLOYEE, WORKS_ON
WHERE EMPLOYEE.ESSN=WORKS_ON.ESSN
GROUP BY DNO;
```

查询结果截图如下：

<input checked="" type="checkbox"/>	Q	DNO string	AVG(SALARY/HOURS) newdecimal
	1	A01	1777.08332917
	2	A02	659.52381000
	3	B01	2893.70748571
	4	B02	645.83333750
	5	C10	4117.19674211

图 12 查询结果截图 9

四、实验心得

4.1 问题解决

4.1.1 中文乱码问题

最开始直接导入 txt 文件时，中文会以乱码形式显示。这是因为操作系统是中文操作系统，默认使用的字符集是 GB2312，可以通过以下命令成功设置为 UTF-8 编码格式：

```
set character_set_client = 'utf8';
set character_set_connection = 'utf8';
set character_set_results = 'utf8';
set character_set_results = 'utf8';
set character_set_server = 'utf8';
set character_set_database = 'utf8';
```

同时在创建关系时也需要进行了”charset=utf8”的字符设置。

```
CREATE TABLE EMPLOYEE(
.....
)charset=utf8;
```

这样就能够解决中文乱码的问题了。

4.1.2 文件导入换行符识别问题

导入 txt 文件时，数据的导入会出现不对齐的错误。原因是换行符的识别出现问题。为了避免换行符的影响，在上述 SQL 代码后加入” LINES TERMINATED BY '\r\n' ”解决。

LOAD DATA LOCAL INFILE "....." INTO TABLE DEPARTMENT LINES TERMINATED BY '\r\n';
--

4.2 实验收获

在本次实验中，学习了 MYSQL 语句的各种用法，包括数据定义、批量导入、查询、删除等操作。结合实际加深了课上所学的内容的理解。