实验0：MySQL关系数据库管理系统

及SQL语言的使用

**1. 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。

**2. 实验环境**

Windows 或macOS操作系统、MySQL5.6版本及以上，Code Blocks编程环境。(可以用其他操作系统和编程环境，数据库尽量选择MySQL)

**3. 实验内容**

**3.1 实验任务**

创建关系数据库COMPANY，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；

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

3：没有参加项目编号为P1的项目的员工姓名；

4：由张红领导的工作人员的姓名和所在部门的名字；

5：至少参加了项目编号为P1和P2的项目的员工号；

6：参加了全部项目的员工号码和姓名；

7：员工平均工资低于3000元的部门名称；

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；

9：每个部门的员工小时平均工资；

**注意：**

**（1）本次实验只需要提交实验报告；**

**3.2 关系数据库COMPANY介绍**

创建关系数据库COMPANY，其模式如下（下划线表示关系的主键）：

关系EMPLOYEE (ENAME, ESSN, 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

关系DEPARTMENT (DNAME, DNO, 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

关系PROJECT (PNAME, PNO, 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

关系WORKS\_ON (ESSN, PNO, HOURS)

ESSN: 工作人员身份证号，

PNO: 工程项目号，

HOURS: 工作小时数

**3.3 数据准备**

向创建的数据库COMPANY中添加数据，以备后续查询使用。

要求数据库中至少包含50个员工，5个部门，10项工程，并且必须包含“研发部”、编号为P1和P2的项目、名叫张红的员工。

**4. MySQL手册**

**4.1 MySQL基本命令**

1. 连接MySQL服务器

在命令行下输入mysql –h localhost –u root –p

当显示“Enter password:”时，输入密码mysql

1. 查看MySQL中有哪些数据库

mysql> show databases; （注意SQL语句结尾的分号！）

1. 使用数据库mysql

mysql> use mysql;

1. 查看当前使用的数据库

mysql> select database();

1. 查看数据库mysql中有哪些关系

mysql> show tables;

1. 查看数据库mysql中关系的模式（以关系user为例）

mysql> describe user;

1. 使用help命令来了解其他命令和变量类型等的含义

mysql> help;

mysql> help use;

1. 使用SQL语言在数据库mysql上进行简单查询

mysql> select \* from user;

mysql> select user, host, password from user;

mysql> select count(\*) from user;

mysql> select count(\*) as ucount from user;

1. 取消命令

若要取消一条正在编辑命令，键入\c并回车

1. 断开MySQL服务器连接

mysql> quit 或 mysql> exit

**4.2使用SQL语言管理MySQL数据库**

1. 创建数据库menagerie

mysql> CREATE DATABASE menagerie;

1. 查看数据库menagerie是否创建成功

mysql> show databases;

1. 使用menagerie数据库

mysql> use menagerie;

1. 创建宠物信息关系pet，包含宠物名字、主人、种类、性别、出生和死亡日期

mysql> CREATE TABLE pet (name VARCHAR(20), owner VARCHAR(20),

-> species VARCHAR(20), sex CHAR(1), birth DATE, death DATE);

1. 查看关系pet是否创建成功

mysql> show tables;

1. 查看关系pet的模式

mysql> describe pet;

1. 向关系pet中插入元组

mysql> INSERT INTO pet

-> VALUES ('Puffball','Diane','hamster','f','1999-03-30',NULL);

1. 从文件中批量导入数据

设D:\pet.txt是一个由字符Tab分隔的文本文件

Fluffy Harold cat f 1993-02-04 \N

Claws Gwen cat m 1994-03-17 \N

Buffy Harold dog f 1989-05-13 \N

Fang Benny dog m 1990-08-27 \N

Bowser Diane dog m 1998-08-31 1995-07-29

Chirpy Gwen bird f 1998-09-11 \N

Whistler Gwen bird \N 1997-12-09 \N

Slim Benny snake m 1996-04-29 \N

在MySQL提示符下执行

mysql> LOAD DATA LOCAL INFILE "D:\pet.txt" INTO TABLE pet;

1. 查询关系pet中所有元组

mysql> SELECT \* FROM pet;

1. 更改关系pet中的数据

mysql> UPDATE pet SET birth = "1989-08-31" WHERE name = "Bowser";

1. 查询名叫Bowser的宠物信息

mysql> SELECT \* FROM pet WHERE name = "Bowser";

1. 查询所有90后小猫和小狗的信息

mysql> SELECT \* FROM pet WHERE birth >= "1990-1-1" AND

-> (species = "dog" OR species = "cat");

1. 查询所有宠物的主人

mysql> SELECT owner FROM pet;

mysql> SELECT DISTINCT owner FROM pet;

1. 查询所有宠物的名字及生日，并按其年龄递增排序

mysql> SELECT name, birth FROM pet ORDER BY birth;

1. 查询所有宠物的名字及生日，并按其年龄递减排序

mysql> SELECT name, birth FROM pet ORDER BY birth DESC;

1. 查询所有宠物的名字、种类及生日，并先按种类名称递增排序，同种宠物按年龄递减排序

mysql> SELECT name, species, birth FROM pet ORDER BY species, birth DESC;

1. 查询所有活着的宠物的信息

mysql> SELECT \* FROM pet WHERE death IS NULL;

1. 查询关系pet中有多少只宠物

mysql> SELECT COUNT(\*) FROM pet;

1. 查询每个主人有多少只宠物

mysql> SELECT owner, COUNT(\*) FROM pet GROUP BY owner;

1. 查询每种宠物的数量

mysql> SELECT species, COUNT(\*) FROM pet GROUP BY species;

1. 查询雄狗和雌狗的数量

mysql> SELECT species, sex, COUNT(\*) FROM pet

-> WHERE species = "dog" GROUP BY species, sex;

1. 查询每种宠物的最大年龄

mysql> SELECT species, MAX(birth) FROM pet GROUP BY species;

**5. 参考资料**

Abraham Silberschatz, Henry F.Korth. 《数据库系统概念（第六版）》

《MySQL中文参考手册》（MySQLBook.chm）