

# 实验报告

肖易佳 计 83

## 实验目的：

通过练习对数据库、数据表各类操作，熟悉 MySQL 的基本语法，以及 Navicat 在 MySQL 数据导入导出方面的应用

## 实验环境：

Win10 系统，MySQL 与 Navicat 客户端

## 实验过程：

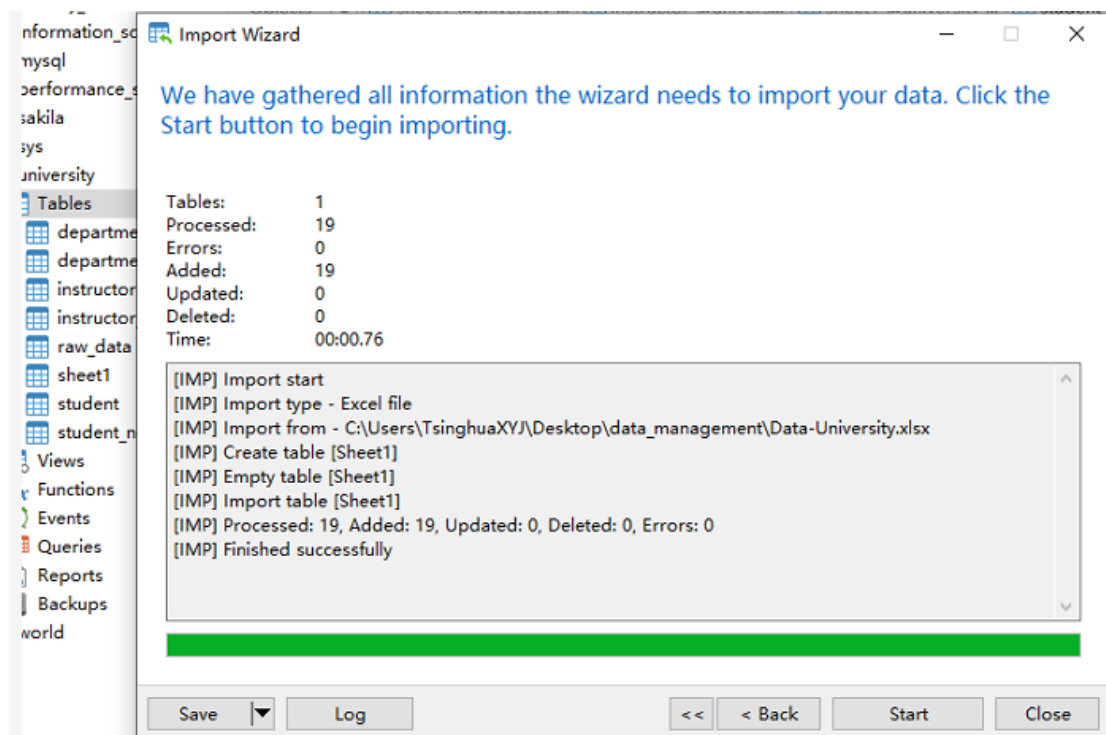
### 1. 数据表结构

根据 ER 示意图，不难确定应采用“3 entities + 3 relations”的数据表。

- 其中 entities 各自存储其 attributes，并设置主键 primary key (默认状态根据主键升序排列)
- 其中 entities 各自存储其对应两端的 entity 实例 instance，1：N、M：1 情况下使用 M、N 的一端为主键，N：M 情况下则使用复合主键

### 2. 处理数据集

- 首先建立 database – create database university
- 进入 university，并且建立 table – use university; create table table\_name(……)
- 导入数据



#### d) 单表查询

此部分较为简单，主要应用

- **select** \* **from** table\_name **where** condition 语句

```
• drop table student_new;
• create table if not exists student_new (select s_ID, s_name, tot_cred from sheet1 where tot_cred >= 100);
• select * from student_new;
```

- select **avg**(attri\_name) where ...语句

```
select avg(salary)
from instructor_new
where 50000.00 <= salary and salary <= 100000.00;
```

- select \* from table\_name where condition **ORDER BY** 语句

```
• drop table department_new;
• create table if not exists department_new (select dept_name, building, budget from sheet1 where budget >= 80000.00) order by budget desc;
• select * from department_new;
```

- **group by** 语句（实现学生数目统计，也与第 7 问 update 相关 – 需要 join 两表）
- Ps. 设置主键 **alter**

```
• drop table department;
• create table if not exists department(select distinct dept_name, building, budget from sheet1 where dept_name != 'null');
• alter table department drop primary key;
• select * from department;
• alter table department add primary key(dept_name);
• select * from department;
```

#### e) 多表查询

- Join 语句

- **Natural** join 执行条件语句，对表中具有相同名称的列自动进行匹配
- **Inner** join: 相当于参与运算的两个表格做笛卡尔积，取出满足 ON condition 中 condition 语句的行，形成一个新的表格
- **Left** outer join: 相当于 inner join 后，将左表未能配对的数据行添加到 inner join 的表格，缺失的数据用 null 补齐
- **Right** outer join: 类似 left join，区别是以右表补齐
- **Full** outer join: 在 inner join 的基础上，补齐左、右表的缺失数据，本实验使用的 MySQL 不支持直接 outer join，可以使用 left join + **union** + right join 实现

- 实现 instructor\_to\_studentNumber 表格

- Instructor 表格已存在，观察原始表格，发现存在 instructor 没有学生，即：有 instructor 对应 0 个学生（可用 null 代表），因此分两步完成：先采用 full join 保证数据不遗漏，接着使用 **group by** 语句（按照教师 ID 分组），结合 **count** (student\_ID) 即可完成统计

#### f) Update

关键语句：update table\_name set action where condition;

作用：对名为 table\_name 的表格中的所有满足 condition 条件的条目进行 action 操作

#### g) Delete

较为简单，delete from **table\_name** where **condition** 即可删除 **table\_name** 中满足 **condition** 条件的条目

## 实验收获：

1. 若变量名与**关键字**重复（在统计导师学生数部分，出现'count(s\_ID)'作为 column name），则在使用时， 需要加上**反引号**
2. Union 求并集时，若出现**相同的两列**，则 MySQL 不支持将其建立为 table，在这个情况下，需提前**修改表头**-其中一表的 column name（使用 **alter table table\_name CHANGE old new type**）
3. 若是表头出现**特殊字符**（如#、[、]、space 等），需用**反引号**

```
--
41 • create table if not exists student_new (select s_ID, `s_name`, `tot_cred` from sheet1 where `tot_cred` >= 100);
42
43 • create table if not exists instructor_new (select `i_ID`, `i_name`, `salary` from sheet1 where `tot_cred` >= 100);
44
45 • select * from student_new;
46
47 • select * from instructor_new;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	i_ID	i_name	salary
▶	76766	Crick	72000
	45565	Katz	75000
	76543	Singh	80000

(题目材料 Excel 表格表头出现空格，有两种处理方案：1)添加反引号，2)导入数据时规范化处理)上图展示反引号处理，最终版本采用 Navicat 导入时**规范化 target field**策略