实验一结果截图

180110704-计算机7班-段裕

一、实验截图

*（注意截图清晰，包含完整的sql语句和运行结果）*

**① 查询emp\_no是“10029”的员工信息，显示其 emp\_no, birth\_date, first\_name, last\_name, gender, hire\_date, title；**

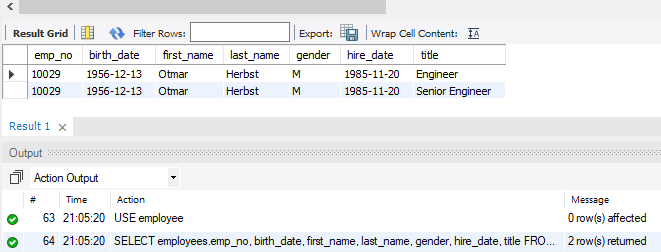
***USE employee;***

***SELECT employees.emp\_no, birth\_date, first\_name, last\_name, gender, hire\_date, title***

***FROM employees, titles***

***WHERE employees.emp\_no = titles.emp\_no AND***

***employees.emp\_no = '10029';***

******

**② 查询入职时间在1990年后且在“Finance”部门工作过的男员工姓名；**

***USE employee;***

***SELECT CONCAT(first\_name, ' ', last\_name) PNAME***

***FROM employees, departments, dept\_emp***

***WHERE***

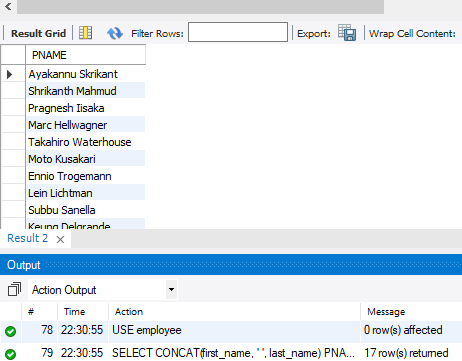
***employees.emp\_no = dept\_emp.emp\_no AND***

***departments.dept\_no = dept\_emp.dept\_no AND***

***departments.dept\_name = 'Finance' AND***

***employees.gender = 'M' AND***

***YEAR(employees.hire\_date) > '1990';***

****

**③ 查询没有在“Production”部门工作过且first\_name 是“Ge”开头的的员工信息，显示其 emp\_no, birth\_date, first\_name, last\_name, gender, hire\_date；**

***USE employee;***

***SELECT DISTINCT employees.emp\_no, birth\_date, first\_name, last\_name, gender, hire\_date***

***FROM employees, departments, dept\_emp***

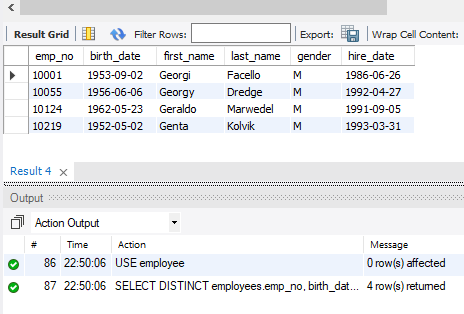
***WHERE***

***employees.emp\_no = dept\_emp.emp\_no AND***

***dept\_emp.dept\_no = departments.dept\_no AND***

***departments.dept\_name <> 'Production' AND***

***first\_name LIKE 'Ge%';***

****

**④ 查询 first\_name 相同且人数超过3人的员工信息，显示其 emp\_no, birth\_date, first\_name, last\_name, gender, hire\_date，要求按first\_name升序显示；**

仅对一个表操作，但需要自连接，同时需要查找与自己first\_name相同的人，这里注意到没有使用员工编号作为连接条件，而是使用first\_name作为条件，同时编号不等（保证不是一个人，这样得到一个表，使用group by语句加count语句得到相同人数超过3人的信息，保留其一，最后使用order by进行排序，最后select出相应的字段。

***USE employee;***

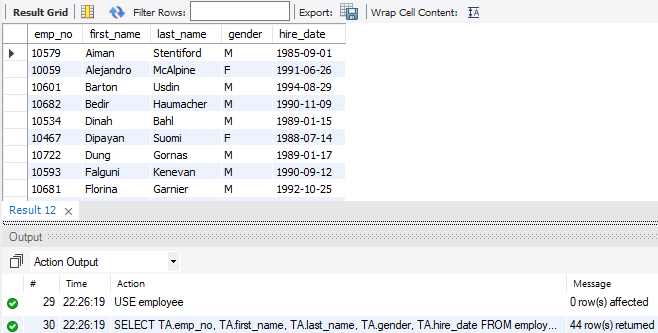
***SELECT TA.emp\_no, TA.first\_name, TA.last\_name, TA.gender, TA.hire\_date***

***FROM employees TA, employees TB***

***WHERE TA.emp\_no <> TB.emp\_no AND TA.first\_name = TB.first\_name***

***GROUP BY TA.first\_name HAVING count(\*)>=3***

***ORDER BY TA.first\_name;***



**⑤ 查询至少在“Production”和“Quality Management”两个部门都工作过的员工编号；**

***USE employee;***

***SELECT employees.emp\_no***

***FROM employees, dept\_emp, departments***

***WHERE employees.emp\_no = dept\_emp.emp\_no AND***

***dept\_emp.dept\_no = departments.dept\_no AND***

***departments.dept\_name = 'Production'***

***AND employees.emp\_no IN(***

***SELECT employees.emp\_no***

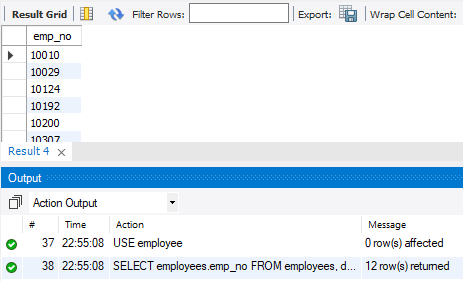
***FROM employees, dept\_emp, departments***

***WHERE employees.emp\_no = dept\_emp.emp\_no AND***

***dept\_emp.dept\_no = departments.dept\_no AND***

***departments.dept\_name = 'Quality Management'***

***)***

****

**⑥ 查询至少在2个部门工作过的员工人数；**

通过group by 语句根据编号分组，保留分组后组中条目数大于等于2的组，将选出的记录取一个别名，再select count(\*)，注意一定要取一个别名，否则报错。

***USE employee;***

***SELECT COUNT(\*) cnt***

***FROM***

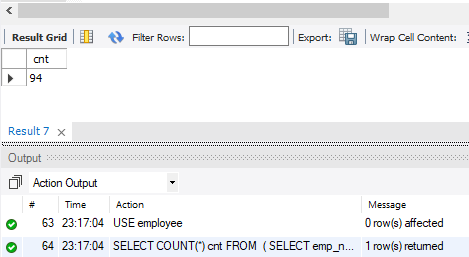
***(***

***SELECT emp\_no***

***FROM dept\_emp***

***GROUP BY dept\_emp.emp\_no HAVING COUNT(\*) >= 2***

***) res;***

****

**⑦ 查询在“d003”部门工作过的且工资最高的员工编号及其最高工资；**

这里给出的是我想到的一个较为简洁的方式，这里需要用到salaries表和dept\_emp两个表，salaries表相对较大，原因在于，没有说明是“当前”，每一个员工的工资按照年限划分，每个人不同的年份各自不同，对应于多条记录，因而可以认为salaries没有主键。我的想法是根据员工编号分组，选出每组工资的最大值（也即这个人历史最高工资），得到一个表，再对表排序并输出第一条（如果使用ALL关键字会出现一些比较麻烦的事）

***USE employee;***

***SELECT dept\_emp.emp\_no, MAX(salaries.salary) as PerMax***

***FROM dept\_emp, salaries***

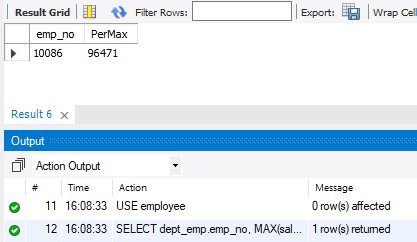
***WHERE dept\_emp.emp\_no = salaries.emp\_no AND***

***dept\_emp.dept\_no = 'd003'***

***GROUP BY dept\_emp.emp\_no***

***ORDER BY PerMax DESC***

***LIMIT 1***

******

**⑧ 查询“d002”部门的当前领导姓名；**

***USE employee;***

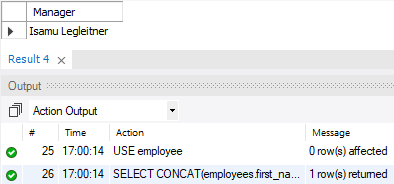
***SELECT CONCAT(employees.first\_name, ' ', employees.last\_name) Manager***

***FROM employees, dept\_manager***

***WHERE employees.emp\_no = dept\_manager.emp\_no***

***AND dept\_manager.dept\_no = 'd002'***

***AND YEAR(dept\_manager.to\_date) = '9999'***

****

**⑨ 查询当前每个部门的部门编号和员工总工资；**

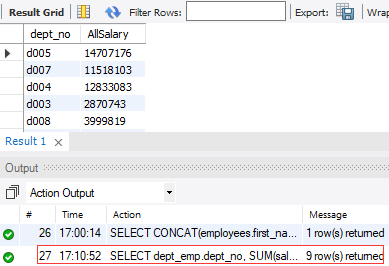
***SELECT dept\_emp.dept\_no, SUM(salaries.salary) AllSalary***

***FROM dept\_emp, salaries***

***WHERE dept\_emp.emp\_no = salaries.emp\_no***

***AND YEAR(salaries.to\_date) = '9999'***

***GROUP BY dept\_emp.dept\_no***

****

**⑩ 查询当前部门员工平均工资在70000元到80000元（包含70000，低于80000）的部门编号，部门名称和员工平均工资；**

根据题意，是要求部门员工当前工资平均值

***USE employee;***

***SELECT dept\_emp.dept\_no, departments.dept\_name, AVG(salaries.salary) as aver***

***FROM dept\_emp, departments, salaries***

***WHERE dept\_emp.emp\_no = salaries.emp\_no***

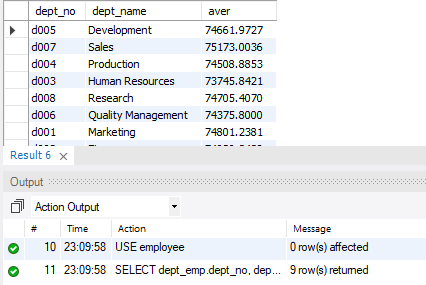
***AND departments.dept\_no = dept\_emp.dept\_no***

***AND YEAR(dept\_emp.to\_date) = '9999'***

***AND salaries.salary >= 70000***

***AND salaries.salary < 80000***

***GROUP BY dept\_emp.dept\_no***

****

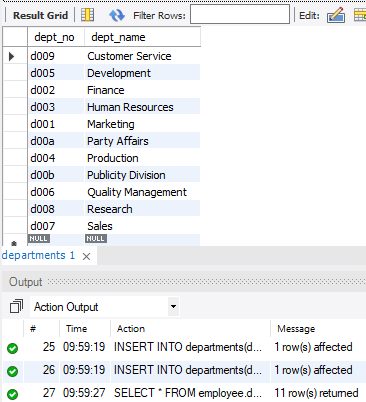
**⑪ 在departments表新增2条记录（内容自定）；**

***INSERT INTO departments(dept\_no, dept\_name)***

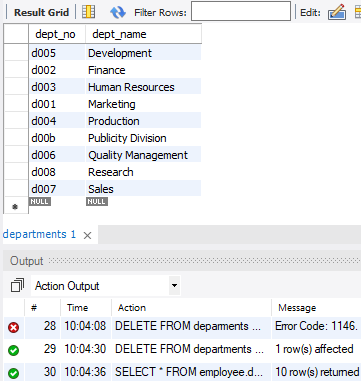
***VALUES('d00a', 'Party Affairs');***

***INSERT INTO departments(dept\_no, dept\_name)***

***VALUES('d00b', 'Publicity Division');***

****

**⑫ 在departments表中删除刚才新增的2条记录中的1条；**

****

***DELETE FROM departments***

***WHERE dept\_no = 'd00a'***

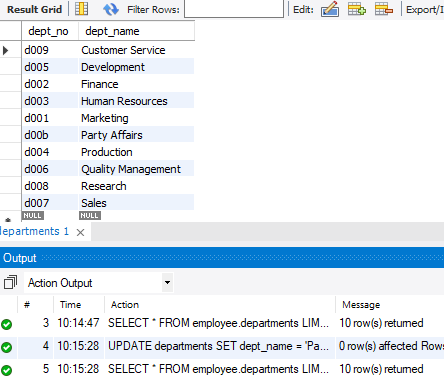
**⑬ 在departments表中修改步骤11新增的记录；**

***UPDATE departments***

***SET dept\_name = 'Party Affairs'***

***WHERE dept\_no = 'd00b';***

***SELECT \* FROM employee.departments;***

****

**⑭ 新建视图，查询所有在1990年后入职过“Finance”部门的男员工信息，包括：emp\_no, birth\_date, first\_name, last\_name, hire\_date, from\_date, to\_date。**

注意：在进行查询时已经将上面做出的修改恢复为原来的数据

***CREATE VIEW MFEMP AS(***

***SELECT \****

***FROM employees***

***WHERE employees.gender = 'M'***

***);***

***SELECT MFEMP.emp\_no, birth\_date, first\_name, last\_name, hire\_date, from\_date, to\_date***

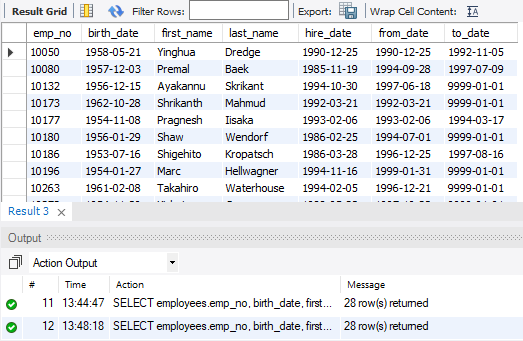
***FROM MFEMP, departments, dept\_emp***

***WHERE MFEMP.emp\_no = dept\_emp.emp\_no***

***AND dept\_emp.dept\_no = departments.dept\_no***

***AND departments.dept\_name = 'Finance'***

***AND YEAR(dept\_emp.from\_date) >= '1990';***

****

二、思考题

*（PPT中的思考题）*

1. 如果 insert 一条数据到Departments，但dept\_no 或 dept\_name 和

已有数据重复，会发生什么？同学们请自己尝试一下。

如果对Dept\_emp表新增数据，数据需满足哪些条件？有什么机制可以

保证数据的正确性？答：由于设定了dept\_no为主键，同时dept\_name是unique属性，故而插入失败。由于一个公司不会有太多的部门，并且功能各异，故要求部门编号为主键，同时名称互不相同。可以在建表时通过属性约束保证数据准确性。

2. 使用Query Profiler、Explain进行SQL语句性能分析

导入Employees数据库：Employees Sample Database 是MySQL官方提供的测试数据库。该测试库含有6个表，总计4百万数据记录。

一个SQL语句

举例：查询年薪大于10万的员工姓名

***SELECT DISTINCT CONCAT(e.first\_name, ‘ ’, e.last\_name) Name***

***FROM employees e, salaries s***

***WHERE e.emp\_no = s.emp\_no AND s.salary > 100000***

***SELECT DISTINCT CONCAT(e.first\_name, ‘ ’, e.last\_name) Name***

***FROM employees e***

***WHERE e.emp\_no IN (***

***SELECT DISTINCT s.emp\_no***

***FROM salaries s***

***WHERE s.salary > 100000)***

哪一个SQL语句性能更好呢？

答：导入employees数据库后再进行测试

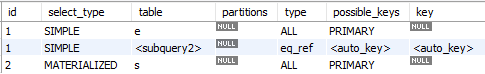
（1）explain语句

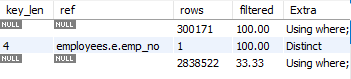
第一条语句





第二条语句





主要关注的指标：

type:

system > const > eq\_ref > ref > fulltext > ref\_or\_null > index\_merge > unique\_subquery > index\_subquery > range > index > ALL

•eq\_ref: 最多只返回一条符合条件的记录，通过使用在两个表有关联字段的时候

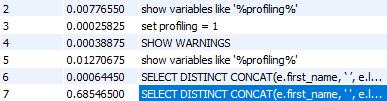
•ref: 通过普通索引查询匹配的很多行时的类型

•eq\_ref: 最多只返回一条符合条件的记录，通过使用在两个表有关联字段的时候

•ref: 通过普通索引查询匹配的很多行时的类型

（2）使用Query Profiler

第一条语句



***# show profile cpu, block io for query 7***

第二条语句



综述：第二条语句性能更差（中间表、耗时长）

（实验提交邮箱：hit\_database\_cs3@163.com）