

数据库系统课程实验报告

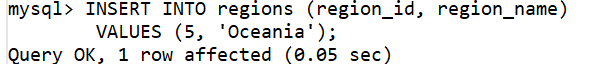
|  |  |
| --- | --- |
| 实验名称： | 数据更新 |
| 实验日期： | 2025.4.26 |
| 实验地点： | 西部片区4号楼105 |
| 提交日期： | 5.05前 |
|  | |
| 学号： | 37220232203786 |
| 姓名： | 潘腾凯 |
| 专业年级： | 软工2023级 |
| 学年学期： | 2024-2025学年第二学期 |

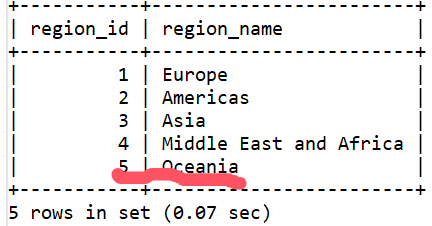
1. 实验目的

* 掌熟练掌握单条记录和小批量数据插入的方法（INSERT）
* 熟练掌握使用子查询实现数据插入的方法(INSERT INTO…SUBQUERY)
* 熟练掌握数据修改和删除的方法（UPDATE, DELETE, TRUNCATE）
* 理解约束对数据更新的影响

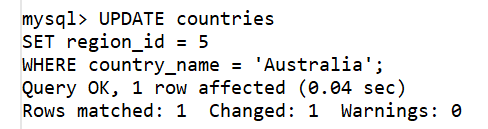
1. 实验内容和步骤

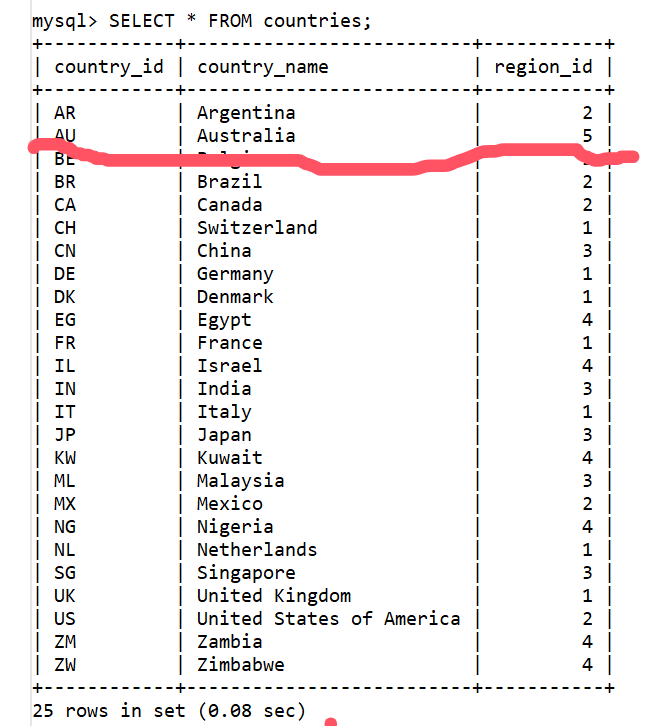
（1） 为地区表 regions 新增一条记录： (‘5’, ‘Oceania’)





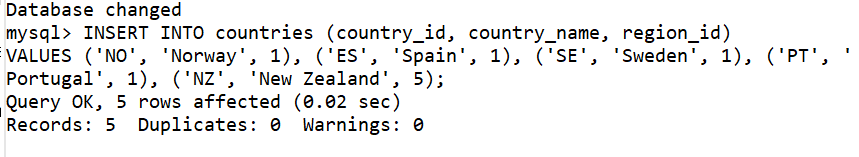
（2） 将 countries 表中的国家名为 Australia 的 region\_id 改为 5

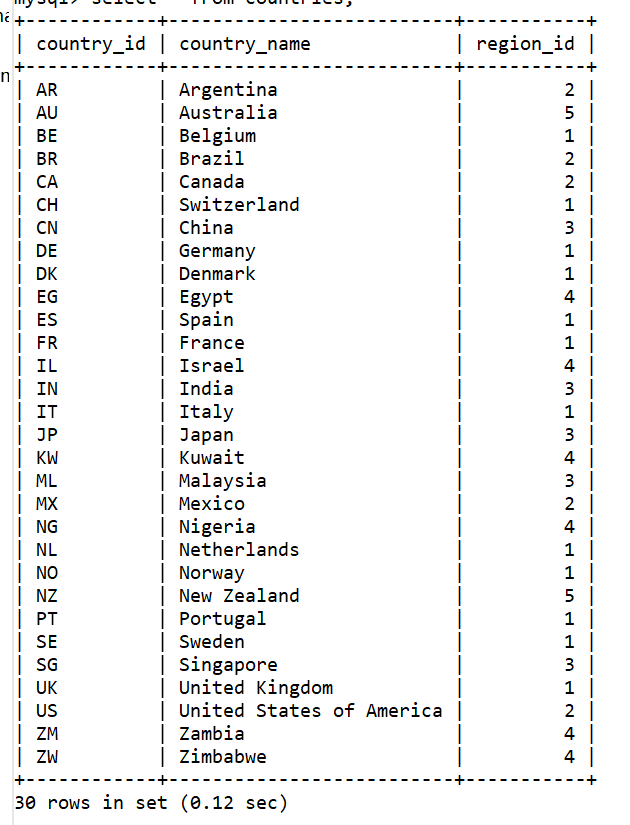




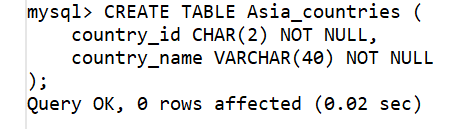
（3） 使用一条批量插入数据语句为 countries 表新增 5 条记录： ('NO','Norway','1'), ('ES','Spain','1'),

('SE','Sweden','1'), ('PT','Portugal','1'), ('NZ','New Zealand','5')

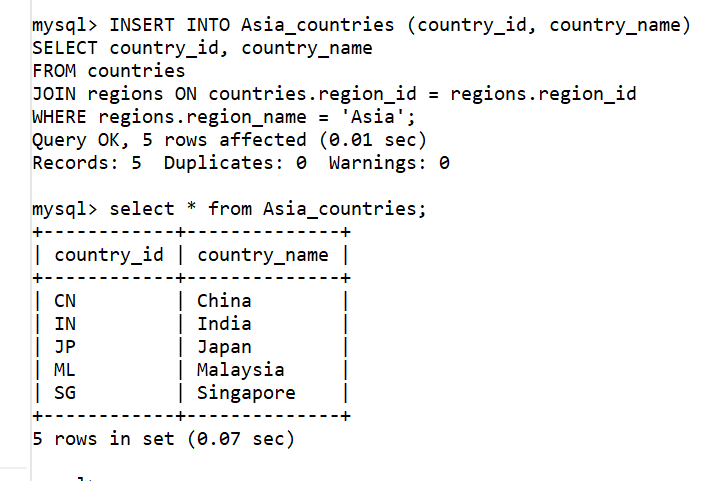




（4） 创建一张名为 Asia\_countries(country\_id, country\_name)的新表，其中字段就是 countries 表中的同名字段

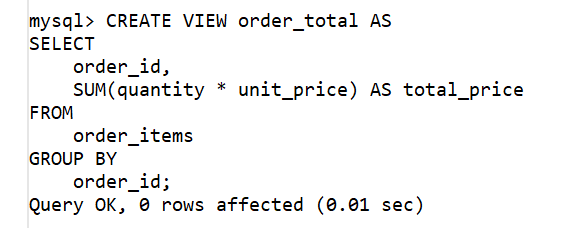


（5） 将 countries 表中所有亚洲国家的数据插入到该表中（要求使用插入子查询结果的方法实现）

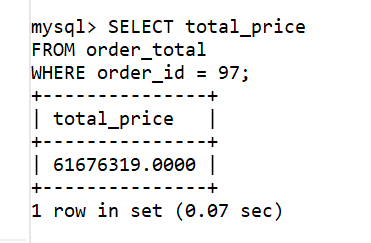


（6） 创建一张名为 order\_total(order\_id, total\_price)的视图，该视图存放每个订单号及其总价，其中total\_price 为总价，其值为数量 quantity 与单价 unit\_price 乘积之和， order\_id， quantity 和

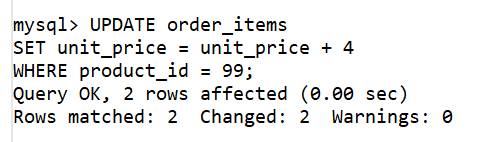
unit\_price 为 order\_items 表中的同名字段



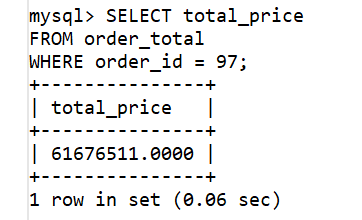
（7） 查询 order\_total 视图中订单号 order\_id 为 97 的总价并记录该结果



（8） 将 order\_items 表中 product\_id 为 99 的单价 unit\_price 增加 4 元

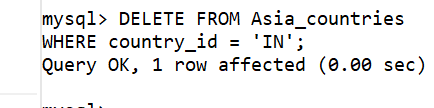


（9） 查询视图 order\_total 中订单号 order\_id 为 97 的总价，将其与第（7）步结果进行比较，观察其异同

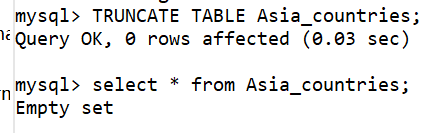


比第七步的查询结果多了192

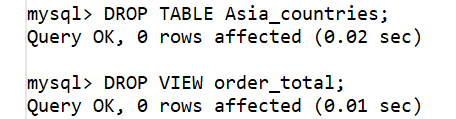
（10） 使用 delete 命令删除 Asia\_countries 表中 country\_id 为 IN 的记录



1. 使用 truncate 命令清空 Asia\_countries 表的所有记录



1. 删除 Asia\_countries 表和视图 order\_total



（13）使用 show create table employees; 命令查看 employees 表的外键约束。

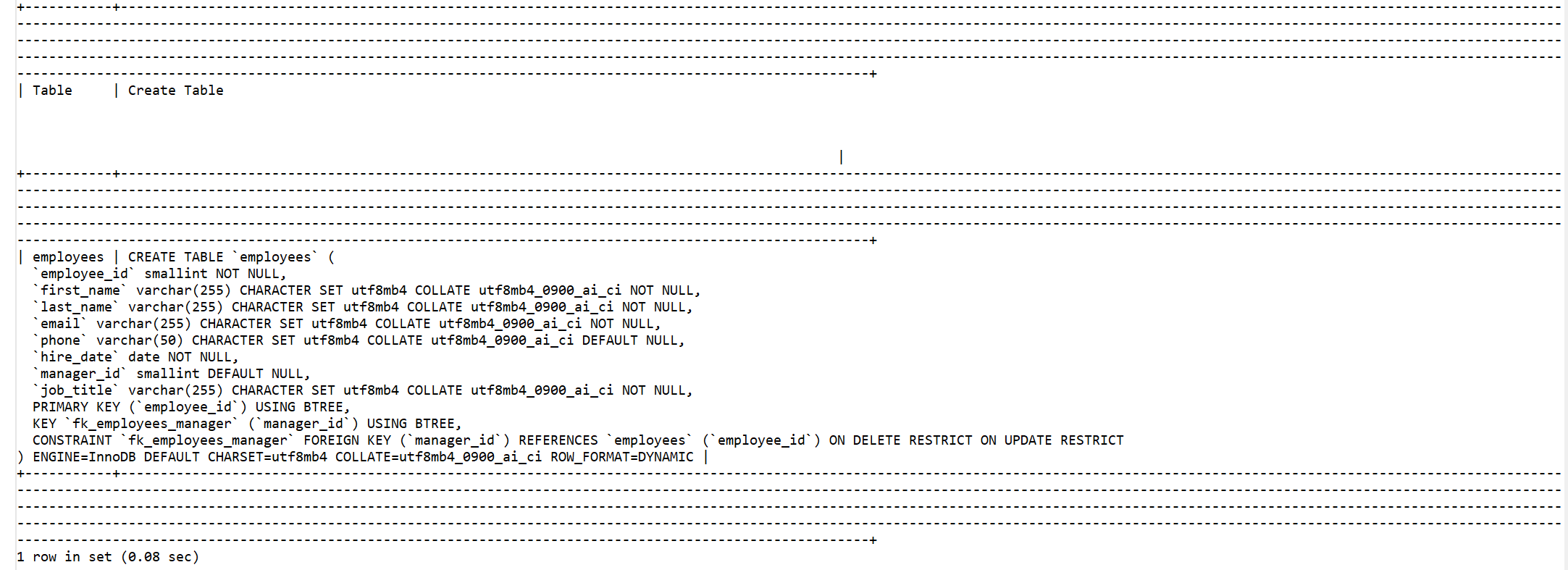
若结果显示没有外键约束，则使用以下命令建立外键约束：

alter table employees add constraint fk\_employees\_manager foreign key (manager\_id) references employees(employee\_id);

若有外键约束且出现 ON DELETE CASCADE 选项（下图），则先删后建外键约束，保证不出现 ON DELETE CASCADE 选项



若有外键约束但没有 ON DELETE CASCADE 选项，则进行第（14）步。



结果如上，有外键约束但没有ON DELETE CASCADE选项

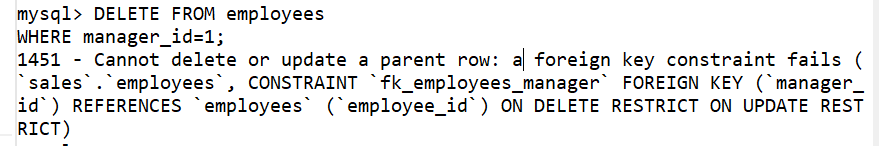
（14） 查询 employees 表中 manager\_id 为 1 的记录



（15） 执行以下删除命令：

DELETE FROM employees where manager\_id=1;

观察执行结果



结果如上：因为有外键约束不能删除

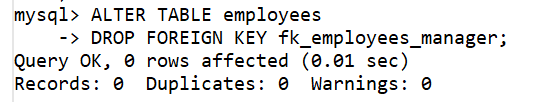
（16）删除 employees 表上的外键约束 fk\_employees\_manager，然后使用以下命令重建该外键约束：

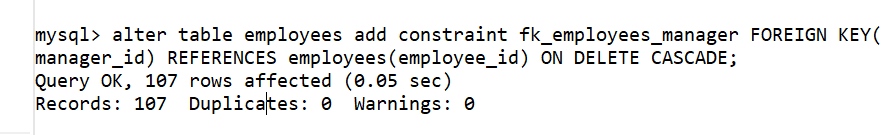
alter table employees add constraint fk\_employees\_manager FOREIGN KEY(manager\_id) REFERENCES employees(employee\_id) ON DELETE CASCADE;

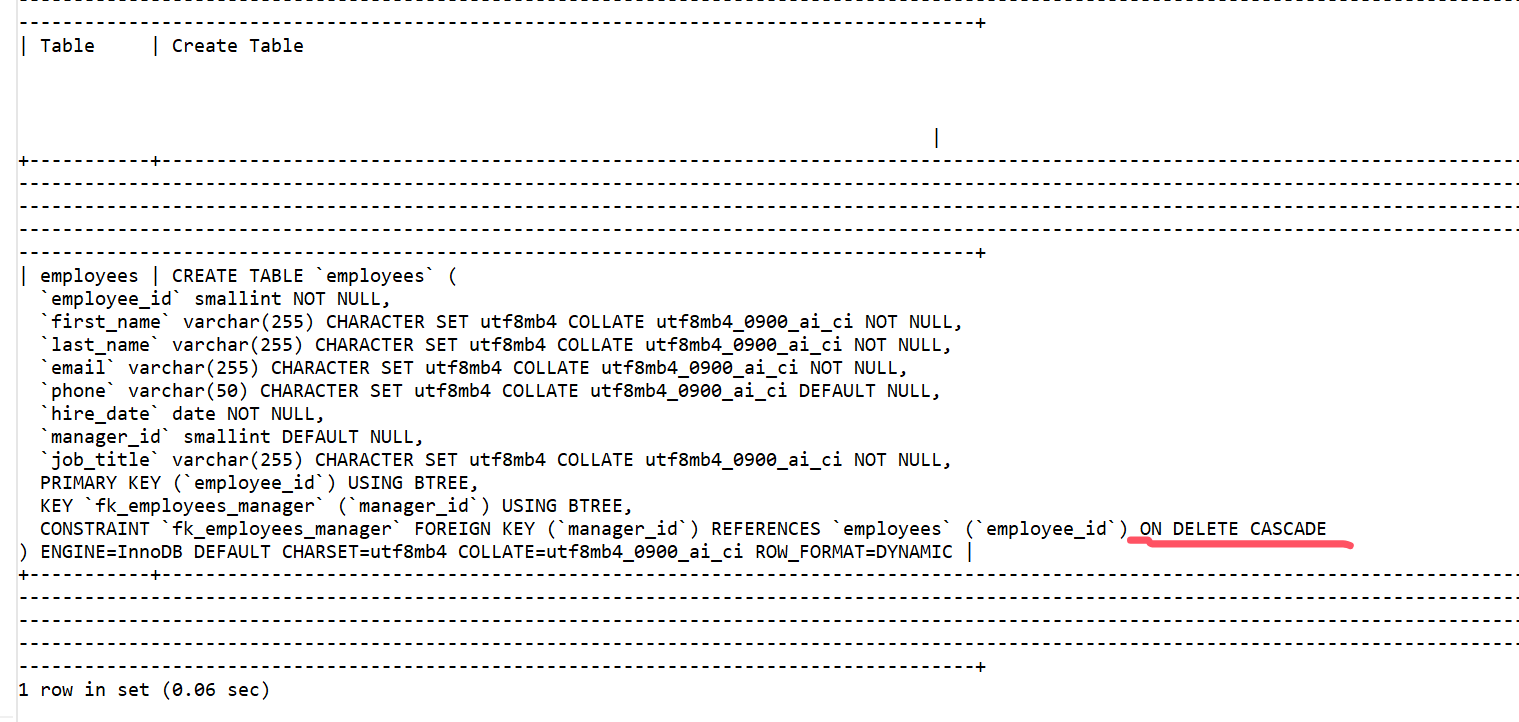
执行后使用命令 show create table employees;查看结果



这一步的目的是测试 ON DELETE CASCADE 的作用和效果

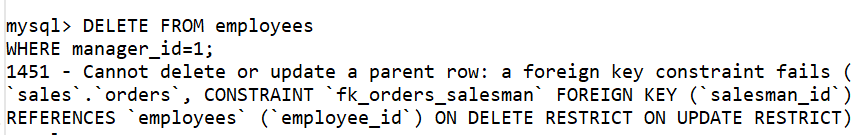




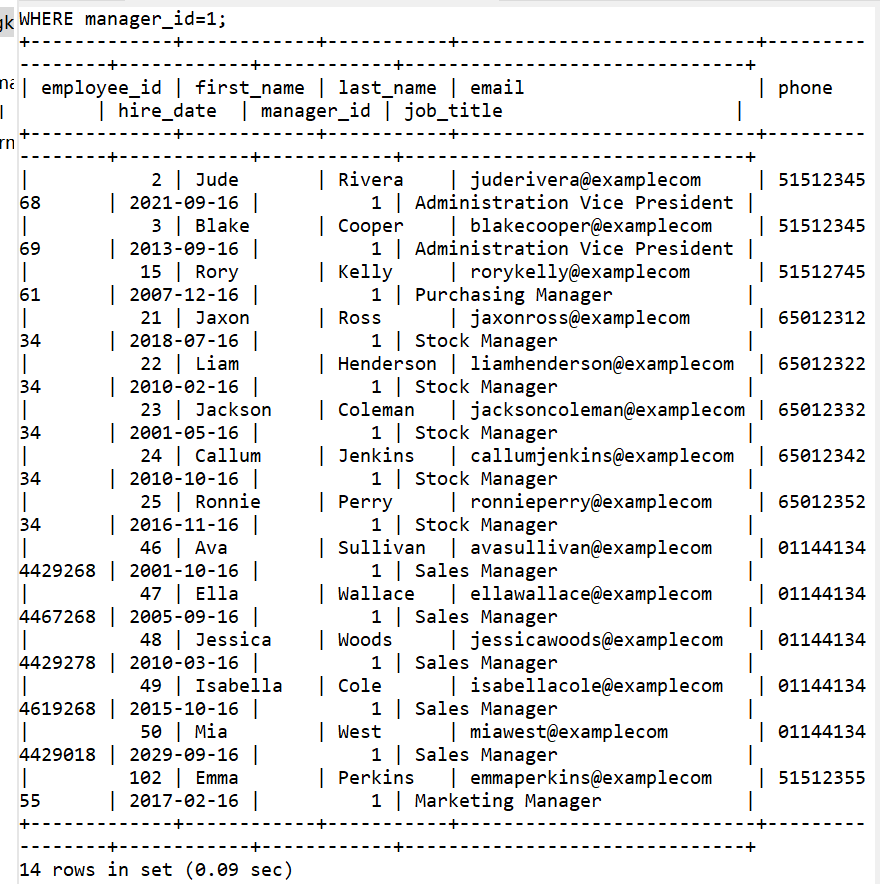


可以看到原来划线处的ON DELETE RESTRICT 和 ON UPDATE RESTRICT变为ON DELETE CASCADE

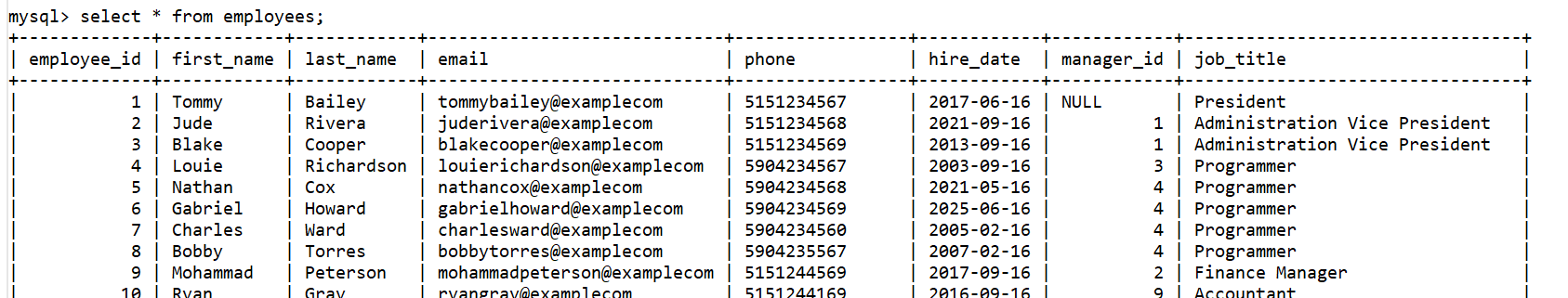
（17） 执行删除命令 DELETE FROM employees where manager\_id=1;

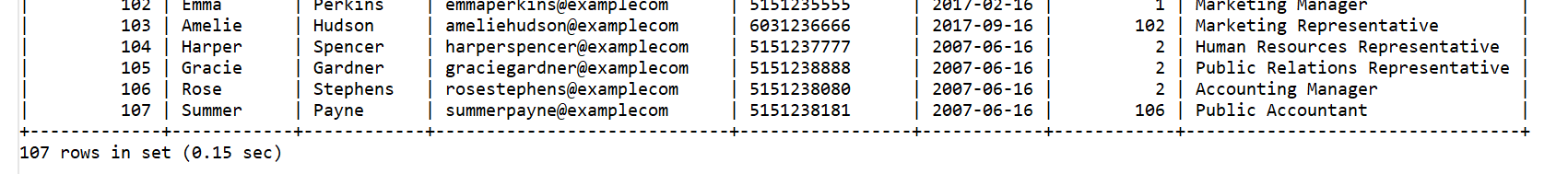


（18） 查询 employees 表中 manager\_id 为 1 的记录，观察执行结果



（19） 查询 employees 表中的所有记录，观察执行结果





（20） 分析步骤（14）、（18）和（19）的执行结果并给出理由

从前后show create table employees的结果来看，外键约束已经将删除行为设置为级联删除。但是删除manager\_id=1的元组时仍出现同样的报错，没办法成功删除。原因推测如下：

1.其他相关外键约束影响：employees 表可能存在多个外键约束，虽然 fk\_employees\_manager 已设为 CASCADE ，但可能有其他外键（比如与 orders 表关联的 fk\_orders\_salesman ）依旧是 RESTRICT ，且删除操作触发了这个未修改的约束，导致报错 。

2.复杂数据依赖：

多层关联关系：假设存在表 A 关联 employees 表，employees 表又关联 orders 表这种多层关系，即使 employees 表自身外键约束设为 CASCADE ，但更深层级表间外键约束为 RESTRICT ，也会阻止删除 。

循环依赖：若数据库设计有循环外键引用关系，比如 employees 表和其他表相互引用，设置 CASCADE 后执行删除，可能因逻辑冲突导致失败 。

3.引擎问题：MySQL 里只有 InnoDB 存储引擎支持外键约束。如果相关表不是 InnoDB 引擎，修改外键约束为 CASCADE 不会生效，删除时仍按默认规则（类似 RESTRICT ）检查 。可通过 SHOW CREATE TABLE 表名; 语句查看表的存储引擎 。InnoDB用的是B+Tree结构，而我这里显示B-Tree结构，说明不是InnoDB引擎。

综合考虑应该是MySQL引擎问题。

（21） 使用所提供的 employees 表的 SQL 脚本将 employees 表复原到原始状态，包括约束和数据

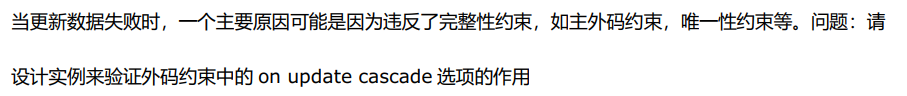
1. 实验总结

3.1 完成的工作

完成了插入、修改、删除数据等操作并理解约束对数据的影响。

3.2 对实验的认识

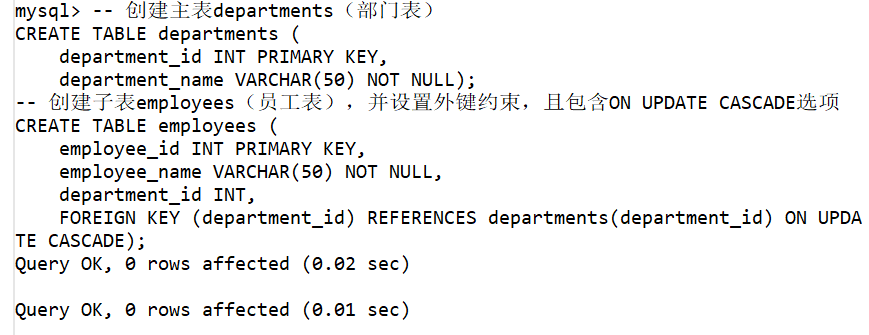
思考题：



答：

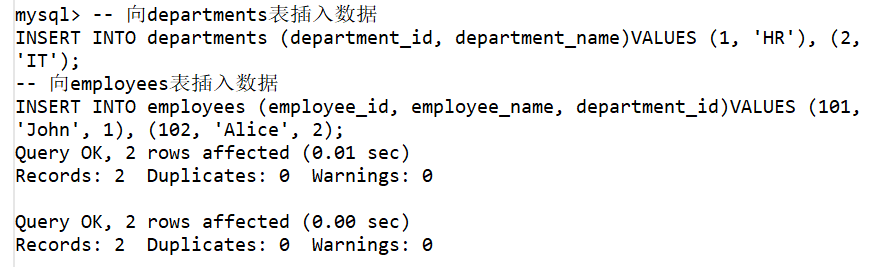
步骤一：创建主表和子表并设置外键约束

-- 创建主表departments（部门表）



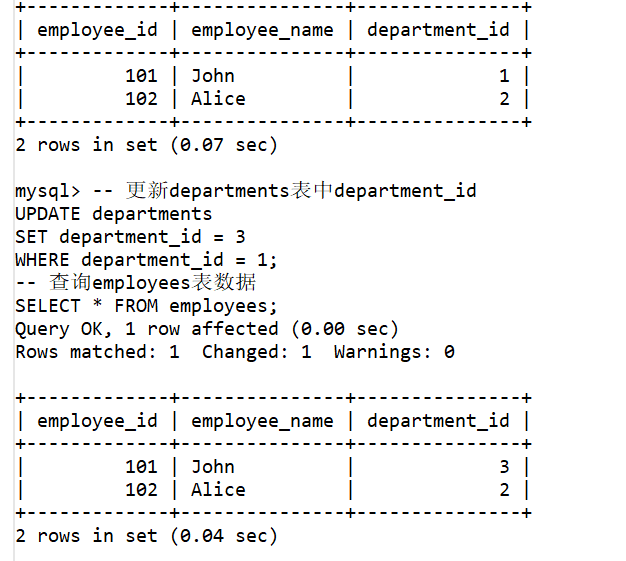
先创建了 departments 主表，包含部门编号（主键）和部门名称字段 。接着创建 employees 子表，包含员工编号（主键）、员工姓名和部门编号字段，通过 FOREIGN KEY 子句将 department\_id 设为外键，引用 departments 表的 department\_id ，并指定 ON UPDATE CASCADE 选项，即当主表 departments 中 department\_id 更新时，子表 employees 中对应的 department\_id 也会自动更新。

步骤二：插入测试数据



在 departments 表插入两条部门记录，在 employees 表插入两条员工记录，分别关联到对应的部门。

步骤三：验证 ON UPDATE CASCADE 作用



更新 departments 表中 department\_id 为 1 的记录，将其改为 3 。由于设置了 ON UPDATE CASCADE ，再查询 employees 表时，会发现原本 department\_id 为 1 的员工记录，其 department\_id 已自动更新为 3 。 这样就验证了 ON UPDATE CASCADE 选项，在主表主键更新时，能自动级联更新子表中外键值的作用。

3.3 遇到的困难及解决方法

将employees的外键约束删除行为设置为级联删除。但是删除manager\_id=1的元组时仍出现同样的报错，没办法成功删除。没有完全确定是什么原因导致的。