

数据库系统课程实验报告

|  |  |
| --- | --- |
| 实验名称： | 实验五 数据更新 |
| 实验日期： | 2023-5-3 |
| 实验地点： | 文宣楼B311 |
| 提交日期： | 2023-5-3 |
|  | |
| 学号： | 33920212204567 |
| 姓名： | 任宇 |
| 专业年级： | 软工2021级 |
| 学年学期： | 2022-2023学年第二学期 |

1. 实验环境

* 华为 ECS+openGauss 数据库服务器平台
* 前提：openGauss 数据库服务器正常运行
* 已建立带样例数据的 SALES 数据库

1. 实验目的

* 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）

1. 实验内容和步骤
2. 为地区表 regions 新增一条记录：(‘5’,’Oceania’)：

INSERT INTO regions VALUES('5','Oceania');



1. 将 countries 表中的国家名为 Austrialia 的 region\_id 改为 5：

UPDATE countries SET region\_id = 5 WHERE country\_name = 'Australia';

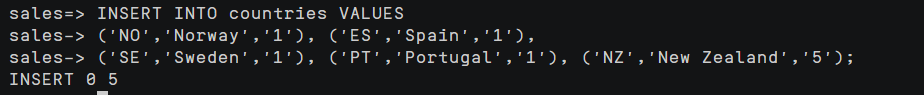


1. 使用一条批量插入数据语句为 countries 表新增 5 条记录：('NO','Norway','1'),('ES','Spain','1'),('SE','Sweden','1'), ('PT','Portugal','1'), ('NZ','New Zealand','5') :

INSERT INTO countries VALUES

('NO','Norway','1'), ('ES','Spain','1'),

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



1. 创建一张名为 Asia\_countries(country\_id,country\_name)的新表，其中字段为 countries 表中的同名字段:

CREATE TABLE Asia\_countries(

country\_id CHAR( 2 ) ,

country\_name VARCHAR2( 40 ),

CONSTRAINT asia\_countries\_country\_id\_key\_fk FOREIGN KEY(country\_id) REFERENCES countries(country\_id)

ON DELETE CASCADE

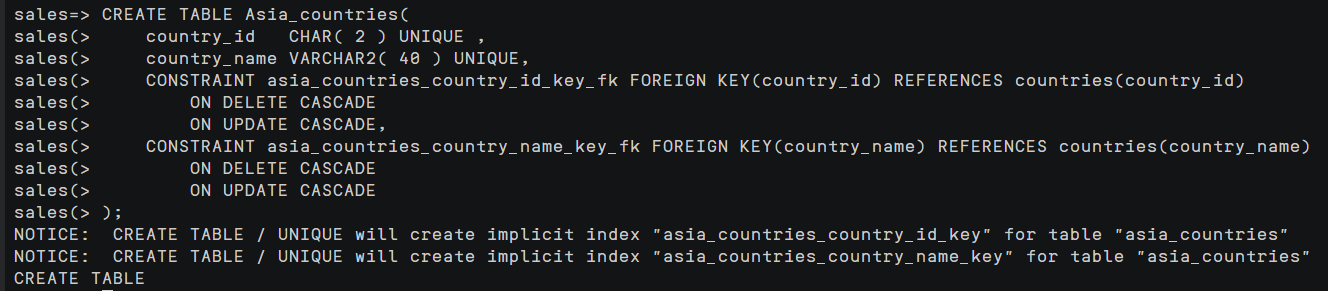
ON UPDATE CASCADE,

CONSTRAINT asia\_countries\_country\_name\_key\_fk FOREIGN KEY(country\_name) REFERENCES countries(country\_name)

ON DELETE CASCADE

ON UPDATE CASCADE

);

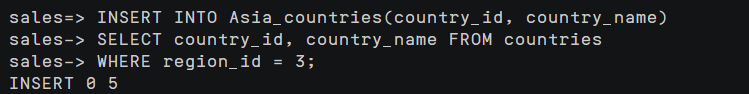


1. 将 countries 表中所有亚洲国家的数据插入到该表中。（要求使用插入子查询结果的方法实现）:

INSERT INTO Asia\_countries(country\_id, country\_name)

SELECT country\_id, country\_name FROM countries

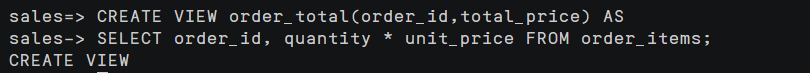
WHERE region\_id = 3;



1. 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:

CREATE VIEW order\_total(order\_id,total\_price) AS

SELECT order\_id, quantity \* unit\_price FROM order\_items;

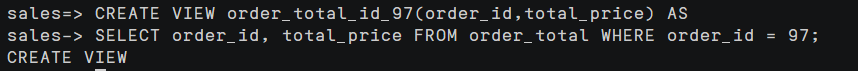


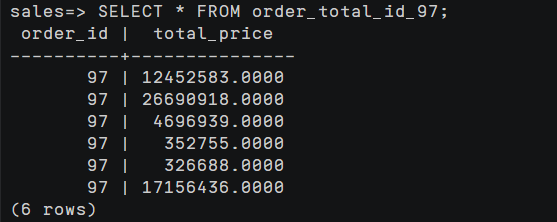
1. 查询 order\_total 视图中订单号 order\_id 为 97 的总价并记录该结果:

CREATE VIEW order\_total\_id\_97(order\_id,total\_price) AS

SELECT order\_id, total\_price FROM order\_total WHERE order\_id = 97;

SELECT \* FROM order\_total\_id\_97;





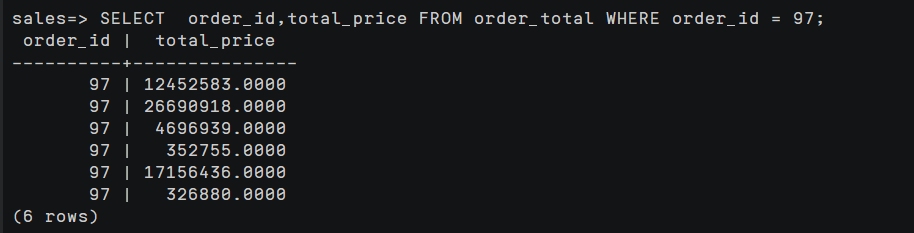
1. 将 order\_items 表中 product\_id 为 99 的单价 unit\_price 增加 4 元:

UPDATE order\_items SET unit\_price = unit\_price + 4 WHERE product\_id = 99;



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

SELECT order\_id,total\_price FROM order\_total WHERE order\_id = 97;



两者查询结果相同，但顺序略有不同

1. 使用 delete 命令删除 Asia\_countries 表中 country\_id 为 IN 的记录:

DELETE FROM Asia\_countries WHERE country\_id = 'IN';



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

TRUNCATE TABLE Asia\_countries;

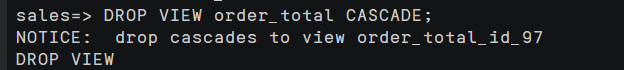


1. 删除 Asia\_countries 表和视图 order\_total:

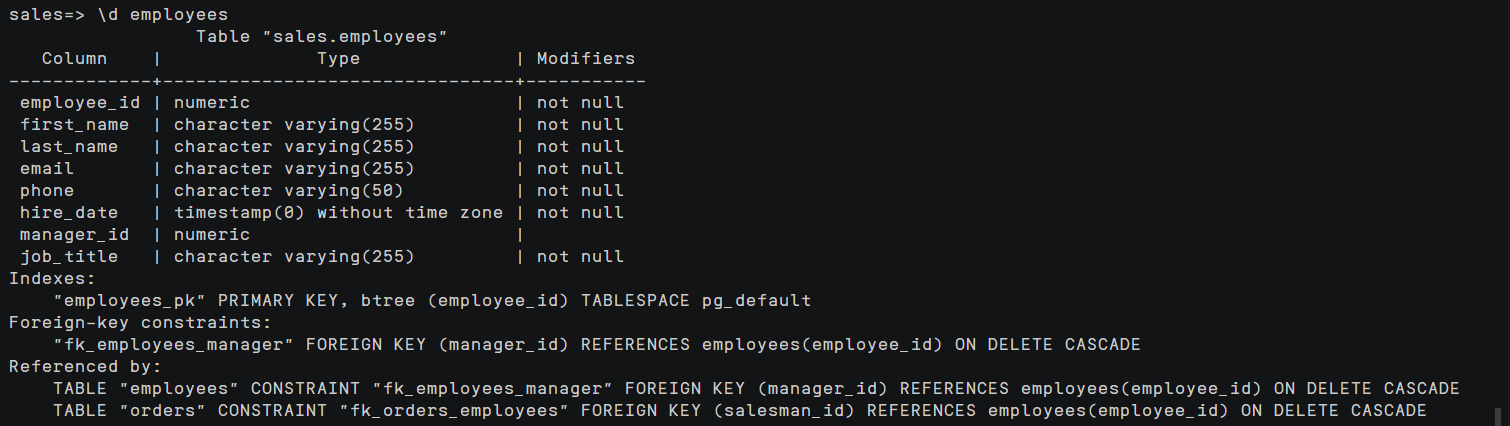
DROP TABLE Asia\_countries;

DROP VIEW order\_total CASCADE;



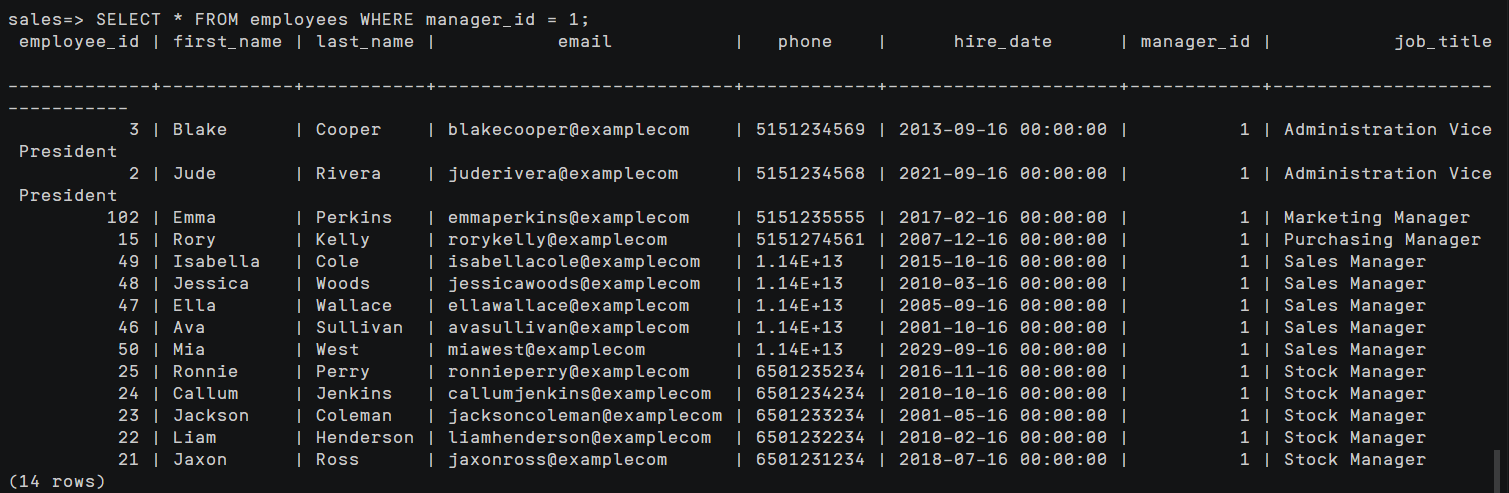


1. 使用命令\d employees 查看 employees 表的外码约束语句，包括 on delete cascade 选项:



1. 查询 employees 表中 manager\_id 为 1 的记录:

SELECT \* FROM employees WHERE manager\_id = 1;



1. 修改 employees 表的外码约束，去掉外码约束中的 on delete cascade 选项，但保留原有的外码引用，即 manager\_id 引用本表上的 employee\_id。（可通过先删后建实现）:

ALTER TABLE employees DROP CONSTRAINT fk\_employees\_manager;

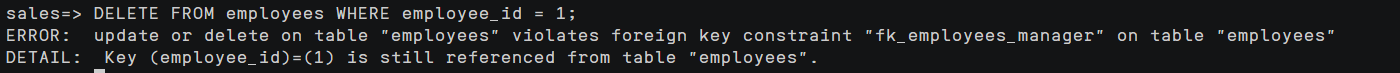
ALTER TABLE employees ADD CONSTRAINT fk\_employees\_manager FOREIGN KEY( manager\_id ) REFERENCES employees( employee\_id );





1. 删除 employees 表中 employee\_id 为 1 的记录，观察操作结果:

DELETE FROM employees WHERE employee\_id = 1;

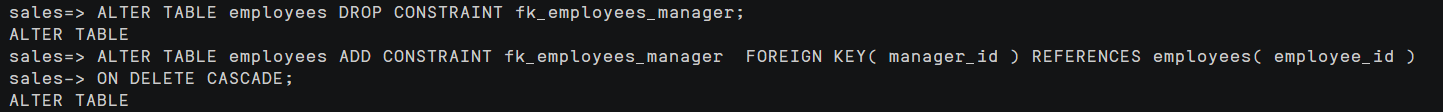


无法删除，执行语句报错

1. 修改 employees 表的外码约束，增加 on delete cascade 选项，即回到最初的外码约束状态:

ALTER TABLE employees DROP CONSTRAINT fk\_employees\_manager;

ALTER TABLE employees ADD CONSTRAINT fk\_employees\_manager FOREIGN KEY( manager\_id ) REFERENCES employees( employee\_id ) ON DELETE CASCADE ;



1. 再次执行第（16）步，观察操作结果:



此时删除成功

1. 实验总结

4.1 完成的工作

1. 登录到创建的Sales数据库中；

2. 使用SQL语句进行数据更新，视图的创建、用子查询语句创建视图、数据的完整性约束等操作。；

4.2 对实验的认识

通过本次实验，熟悉了SQL语句中数据更新的相关操作，感受到对于数据库的操作需要考虑属性的约束条件及之间的联系，不同表、视图时之间的依赖关系等。

4.3 遇到的困难及解决方法

无。

1. 实验思考

 当更新数据失败时，一个主要原因可能是因为违反了完整性约束，如主外码约束，唯一性约束等。

问题：请设计实例来验证外码约束中的 on update cascade 选项的作用。

答：假设我们有两个表，一个是 "users" 表，另一个是 "passwords" 表。"users" 表中有一个名为 "user\_id" 的主键，而 " passwords " 表中有一个外键 "user\_id"，用于关联到 "users" 表中的 "user\_id" 字段。

CREATE TABLE users (

user\_id INT PRIMARY KEY,

name VARCHAR(50)

);

CREATE TABLE passwords (

password VARCHAR(50) ,

user\_id INT,

FOREIGN KEY (user\_id)

REFERENCES users(user\_id)

ON UPDATE CASCADE

);

插入一条数据：



更新users表数据，可以观察到passwords表中数据也改变

