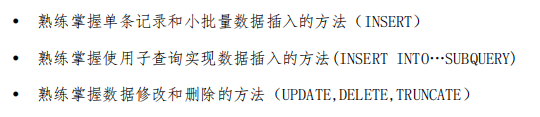


数据库系统课程实验报告

|  |  |
| --- | --- |
| 实验名称： | 数据更新 |
| 实验日期： | 2022/5/26 |
| 实验地点： | 四号楼 |
| 提交日期： | 2022/5/26 |
|  | |
| 学号： | 22920202202879 |
| 姓名： | 陈奕培 |
| 专业年级： | 软工2020级 |
| 学年学期： | 2021-2022学年第二学期 |

1. 实验目的



1. 实验内容和步骤

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

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



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

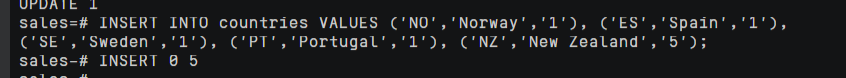
UPDATE countries SET region\_id='5' WHERE country\_name='Australia';



（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')。

INSERT INTO countries VALUES ('NO','Norway','1'), ('ES','Spain','1'),

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

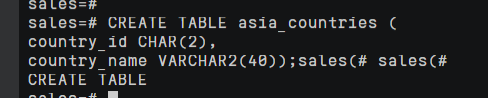


（4）创建一张名为 Asia\_countries(country\_id,country\_name)的新表，其中字段为 countries 表 中的同名字段。

CREATE TABLE asia\_countries (

country\_id CHAR(2),

country\_name VARCHAR2(40));



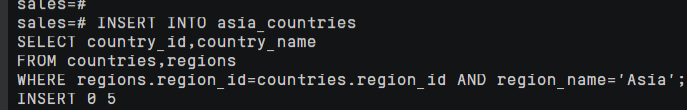
（5）将 countries 表中所有亚洲国家的数据插入到该表中。（要求使用插入子查询结果的方法实现）

INSERT INTO asia\_countries

SELECT country\_id,country\_name

FROM countries,regions

WHERE regions.region\_id=countries.region\_id AND region\_name='Asia';



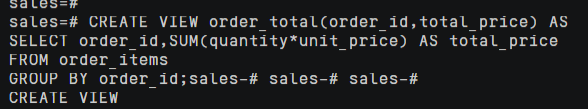
（6）创建一张名为 order\_total(order\_id,total\_price)的视图，该视图存放每个订单号及其总价，其中 total\_price 为总价，其值为数量 quantity 与单价 unit\_price 乘积之和，order\_id，quantity和 unit\_price 为 order\_items 表中的同名字段。

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

SELECT order\_id,SUM(quantity\*unit\_price) AS total\_price

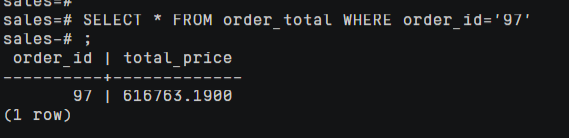
FROM order\_items

GROUP BY order\_id;



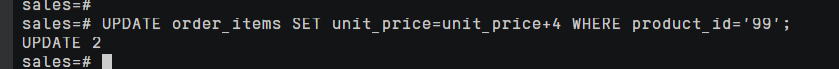
（7）查询 order\_total 视图中订单号 order\_id 为 97 的总价并记录该结果。

SELECT \* FROM order\_total WHERE order\_id='97';



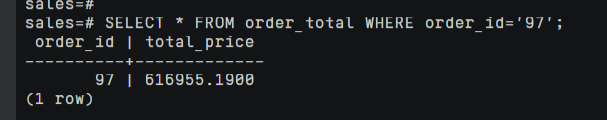
（8）将 order\_items 表中 product\_id 为 99 的单价 unit\_price 增加 4 元。

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



（9）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 \* FROM order\_total WHERE order\_id='97';



总价提升，视图会跟着表一起修改

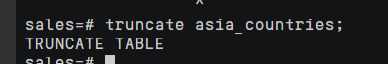
（10）使用 delete 命令删除 Asia\_countries 表中 country\_id 为 IN 的记录。

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



（11）使用 truncate 命令清空 Asia\_countries 表的所有记录。

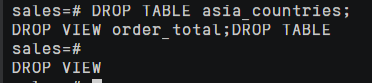
truncate asia\_countries;



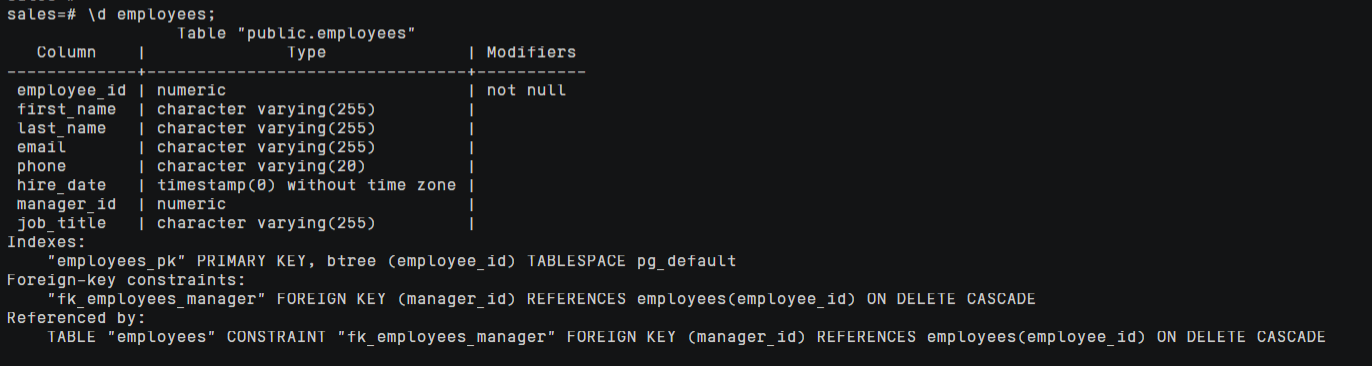
（12）删除 Asia\_countries 表和视图 order\_total。

DROP TABLE asia\_countries;

DROP VIEW order\_total;



（13）使用命令\d employees 查看 employees 表的外码约束语句，包括 on delete cascade 选项。



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

SELECT \* FROM employees WHERE employee\_id='1';

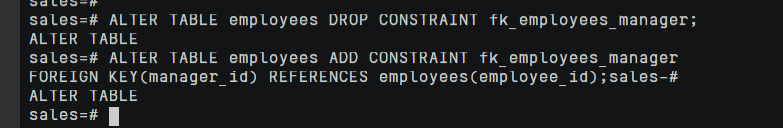


（15）修改 employees 表的外码约束，去掉外码约束中的 on delete cascade 选项，但保留原有的外 码引用，即 manager\_id 引用本表上的 employee\_id。（可通过先删后建实现）

ALTER TABLE employees DROP CONSTRAINT fk\_employees\_manager;

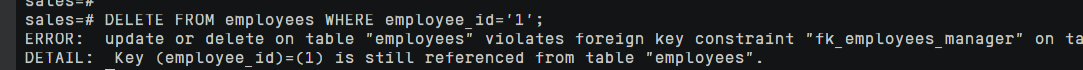
ALTER TABLE employees ADD CONSTRAINT fk\_employees\_manager

FOREIGN KEY(manager\_id) REFERENCES employees(employee\_id);



（16）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';



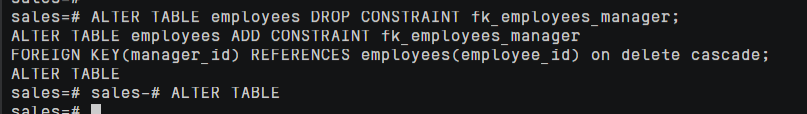
删除失败，因为表内包含了对该条记录的外键约束

（17）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;



（18）再次执行第（16）步，观察操作结果。

DELETE FROM employees WHERE employee\_id='1';



删除成功，将对该条的引用级联删除

思考：

建立测试表（无ON UPDATE CASCADE）

CREATE TABLE test(

    id CHAR(4) PRIMARY KEY,

    father\_id CHAR(4),

    CONSTRAINT test\_fk FOREIGN KEY(father\_id)

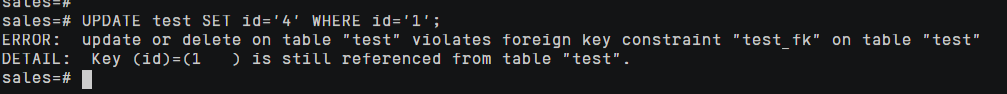
    REFERENCES test(id) ON UPDATE CASCADE

);

插入数据

INSERT INTO test VALUES('1',NULL),('2','1'),('3','1');

对id=1进行修改



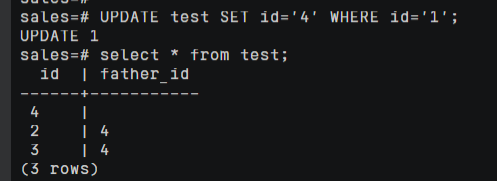
无法修改

加上ON UPDATE CASCADE后，再对id=1修改

ALTER TABLE test DROP CONSTRAINT test\_fk;

ALTER TABLE test ADD CONSTRAINT test\_fk FOREIGN KEY(father\_id)

REFERENCES test(id) ON UPDATE CASCADE;



此时其他记录级联修改，加快效率

1. 实验总结

3.1 完成的工作

建表；

建视图；

插入数据；

修改外码；

级联删除；

3.2 对实验的认识

通过学习，掌握了：

单条记录的插入、

批量记录的插入、

通过select语句插入、

删除命令truncate和delete、

级联删除和修改；

3.3 遇到的困难及解决方法

无