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:	2022/5/29
实验地点:	厦门大学德旺图书馆
提交日期:	2022/5/29

学号:	20420192201952
姓名:	庾晓萍
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

- 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)

2. 实验内容和步骤

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

```
sale=> INSERT INTO regions VALUES('5','Oceania');
INSERT 0 1
```

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

```
sale=> UPDATE countries SET region_id = 5 WHERE country_name = 'Australia';
UPDATE 1
```

（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')。

```
sale=> INSERT INTO countries VALUES
sale->       ('NO','Norway','1'),
sale->       ('ES','Spain','1'),
sale->       ('SE','Sweden','1'),
sale->       ('PT','Portugal','1'),
sale->       ('NZ','New Zealand','5');
INSERT 0 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。

```
sale=> ALTER TABLE countries ADD CONSTRAINT uni_country_id UNIQUE (country_id);
NOTICE:  ALTER TABLE / ADD UNIQUE will create implicit index "uni_country_id" for table "countries"
ALTER TABLE
sale=> ALTER TABLE countries ADD CONSTRAINT uni_country_name UNIQUE (country_name);
NOTICE:  ALTER TABLE / ADD UNIQUE will create implicit index "uni_country_name" for table "countries
"
ALTER TABLE
sale=> Create Table  Asia_countries(
sale(>      country_name VARCHAR2( 40 ) UNIQUE,
sale(>      country_id CHAR( 2 ) UNIQUE ,
sale(>      CONSTRAINT asia_countries_country_id_key_fk FOREIGN KEY(country_id) REFERENCES countries(
country_id)
sale(>          ON DELETE CASCADE
sale(>          ON UPDATE CASCADE,
sale(>      CONSTRAINT asia_countries_country_name_key_fk FOREIGN KEY(country_name) REFERENCES countr
ies(country_name)
sale(>          ON DELETE CASCADE
sale(>          ON UPDATE CASCADE
sale(> );
NOTICE:  CREATE TABLE / UNIQUE will create implicit index "asia_countries_country_name_key" for tabl
e "asia_countries"
NOTICE:  CREATE TABLE / UNIQUE will create implicit index "asia_countries_country_id_key" for table
"asia_countries"
CREATE TABLE
sale=>
```

（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）

```
sale=> INSERT INTO Asia_countries(country_id,country_name)
sale-> SELECT country_id,country_name FROM countries c,regions r
sale-> WHERE r.region_id=c.region_id AND r.region_name='Asia';
INSERT 0 5
sale=>
```

（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。

```
24  -- 创建视图
25  CREATE VIEW order_total(order_id,total_price) AS
26  SELECT order_id,quantity*unit_price FROM order_items;
27  SELECT * FROM order_total;
```

```
 order_id  |  total_price
-----------+---------------
        70 |   62038.6800
        73 |   64478.7600
        74 |   73668.0800
        75 |  108798.7200
        76 |   11658.9400
        77 |   81339.3200
        81 |   83818.7300
        82 |    7588.6200
        83 |   93396.4200
        84 |    9519.6600
        87 |   31876.2200
        90 |  107800.0000
        91 |   23858.4500
        93 |  101800.5900
        94 |   27224.3400
        99 |  100657.4400
       102 |   50658.5100
       104 |  289984.0500
       105 |   71099.2100
         2 |   46571.2500
         4 |   92684.0000
         6 |   81037.8100
         8 |    7086.2400
         2 |   33186.7800
         3 |    7424.7900
         4 |    7044.3600
         5 |   19808.6000
         6 |   43415.3300
         7 |   22570.0000
         8 |   69919.0800
--More--
```

（7）查询 order_total 视图中订单号 order_id 为 97 的总价并记录该结果。

```
sale=> CREATE VIEW total_price_97(order_id,total_price) AS
sale-> SELECT order_id,total_price FROM order_total WHERE order_id=97;
CREATE VIEW
sale=> SELECT * FROM total_price_97;
 order_id | total_price
----------+-------------
       97 | 124525.8300
       97 | 266909.1800
       97 |  46969.3900
       97 |   3527.5500
       97 |   3266.8800
       97 | 171564.3600
(6 rows)
```

（8）将 order_items 表中 product_id 为 99 的单价 unit_price 增加

4 元。

```
sale=> UPDATE order_items SET unit_price=unit_price+4 WHERE PRODUCT_ID=99;
UPDATE 2
```

（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，发现得到的结果是相同的。

```
sale=> SELECT order_id,total_price FROM order_total WHERE ORDER_ID=97;
 order_id | total_price
----------+--------------
       97 | 124525.8300
       97 | 266909.1800
       97 |  46969.3900
       97 |   3527.5500
       97 | 171564.3600
       97 |   3458.8800
(6 rows)
```

（10）使用 delete 命令删除 Asia_countries 表中 country_id 为 IN 的记录。

```
sale=> DELETE FROM Asia_countries WHERE country_id='IN';
DELETE 1
```

（11）使用 truncate 命令清空 Asia_countries 表的所有记录。

```
sale=> TRUNCATE TABLE Asia_countries;
TRUNCATE TABLE
```

（12）删除 Asia_countries 表和视图 order_total。

```
sale=> DROP TABLE Asia_countries;
DROP TABLE
```

```
sale=> DROP VIEW total_price_97;
DROP VIEW
sale=> DROP VIEW order_total;
DROP VIEW
```

（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。

```
sale=> \d employees;
                Table "icebear.employees"
    Column     |                Type                | Modifiers
---------------+------------------------------------+-----------
 employee_id   | numeric                            | not null
 first_name    | character varying(255)             | not null
 last_name     | character varying(255)             | not null
 email         | character varying(255)             | not null
 phone         | character varying(50)              | not null
 hire_date     | timestamp(0) without time zone     | not null
 manager_id    | numeric(12,0)                      |
 job_title     | character varying(255)             | not null
Indexes:
    "employeepk" PRIMARY KEY, btree (employee_id) TABLESPACE pg_default
Foreign-key constraints:
    "fk_employees_manager" FOREIGN KEY (manager_id) REFERENCES employees(employee_id) ON DELETE CASC
ADE
Referenced by:
    TABLE "employees" CONSTRAINT "fk_employees_manager" FOREIGN KEY (manager_id) REFERENCES employee
s(employee_id) ON DELETE CASCADE
    TABLE "orders" CONSTRAINT "fk_orders_employees" FOREIGN KEY (salesman_id) REFERENCES employees(e
mployee_id) ON DELETE SET NULL
```

（14）查询 employees 表中 manager_id 为 1 的记录。

```
 employee_id | first_name | last_name |            email            |        phone        |        hire
_date        | manager_id |                job_title
-------------+------------+-----------+-----------------------------+---------------------+-----------
------------+------------+-----------------------------------
           3 | Blake      | Cooper    | blake.cooper@example.com    | 515.123.4569        | 2016-01-1
3 00:00:00 |          1 | Administration Vice President
           2 | Jude       | Rivera    | jude.rivera@example.com     | 515.123.4568        | 2016-09-2
1 00:00:00 |          1 | Administration Vice President
         102 | Emma       | Perkins   | emma.perkins@example.com    | 515.123.5555        | 2016-02-1
7 00:00:00 |          1 | Marketing Manager
          15 | Rory       | Kelly     | rory.kelly@example.com      | 515.127.4561        | 2016-12-0
7 00:00:00 |          1 | Purchasing Manager
          49 | Isabella   | Cole      | isabella.cole@example.com   | 011.44.1344.619268  | 2016-10-1
5 00:00:00 |          1 | Sales Manager
          48 | Jessica    | Woods     | jessica.woods@example.com   | 011.44.1344.429278  | 2016-03-1
0 00:00:00 |          1 | Sales Manager
          47 | Ella       | Wallace   | ella.wallace@example.com    | 011.44.1344.467268  | 2016-01-0
5 00:00:00 |          1 | Sales Manager
          46 | Ava        | Sullivan  | ava.sullivan@example.com    | 011.44.1344.429268  | 2016-10-0
1 00:00:00 |          1 | Sales Manager
          50 | Mia        | West      | mia.west@example.com        | 011.44.1344.429018  | 2016-01-2
9 00:00:00 |          1 | Sales Manager
          25 | Ronnie     | Perry     | ronnie.perry@example.com    | 650.123.5234        | 2016-11-1
6 00:00:00 |          1 | Stock Manager
          24 | Callum     | Jenkins   | callum.jenkins@example.com  | 650.123.4234        | 2016-10-1
0 00:00:00 |          1 | Stock Manager
          23 | Jackson    | Coleman   | jackson.coleman@example.com | 650.123.3234        | 2016-05-0
1 00:00:00 |          1 | Stock Manager
          22 | Liam       | Henderson | liam.henderson@example.com  | 650.123.2234        | 2016-04-1
0 00:00:00 |          1 | Stock Manager
          21 | Jaxon      | Ross      | jaxon.ross@example.com      | 650.123.1234        | 2016-07-1
8 00:00:00 |          1 | Stock Manager
--More--
```

（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）

```
sale=> ALTER TABLE employees DROP CONSTRAINT fk_employees_manager;
ALTER TABLE
sale=> ALTER TABLE employees ADD CONSTRAINT fk_employees_manager
sale-> FOREIGN KEY(manager_id) REFERENCES employees(employee_id);
ALTER TABLE
sale=>
```

（16）删除 employees 表中 employee_id 为 1 的记录，观察操作结果。

答：出现报错，原因是表“employees”上的更新或删除违反了表“employees”上的外键约束“fk_employees_manager”键

(employee_id)=(1) 仍然从表“employees”中引用。

```
sale=> DELETE FROM employees WHERE employee_id=1;
ERROR:  update or delete on table "employees" violates foreign key constraint "fk_employees_manager"
 on table "employees"
DETAIL:  Key (employee_id)=(1) is still referenced from table "employees".
sale=>
```

（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。

```
sale=> ALTER TABLE employees ADD CONSTRAINT fk_employees_manager
sale-> FOREIGN KEY(manager_id) REFERENCES employees(employee_id)
sale-> ON DELETE CASCADE;
ALTER TABLE
```

（18）再次执行第（16）步，观察操作结果。

答：成功删除。

```
sale=> DELETE FROM employees WHERE employee_id = 1;
DELETE 1
```

3. 实验总结

3.1 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：比如下面在外码约束中增加 on update cascade 级联增加，当主键表 update 时，外键表同时更新。

```
sale=> ALTER TABLE employees DROP CONSTRAINT fk_employees_manager;
ALTER TABLE
sale=> ALTER TABLE employees ADD CONSTRAINT fk_employees_manager
sale-> FOREIGN KEY(manager_id) REFERENCES employees(employee_id)
sale-> ON DELETE CASCADE ON UPDATE CASCADE;
ALTER TABLE
sale=>
```

3.2 对实验的认识

通过实验我对 openGauss 中的一些语句更熟悉了。如

如 SELECT * FROM customer_t1;可以用来查询表 customer_t1 的所有数据。gsql -d sale -p 26000 -U yuxiaoping -W yuxiaoping@123 -r 或者 gsql -d sale -p 26000 -U user1 -W user1@123 -r 可以用来将新用户连接到数据库。可以使用 gsql -d postgres -p 26000 -r 连接到 postgres。gs_om -t start 可以开启数据库。

3.3 遇到的困难及解决方法

要更改当前会话的默认 Schema，请使用 SET 命令。执行如下命令 SET SEARCH_PATH To icebear,public;将搜索路径设置为 myschema、public，首先搜索 myschema。

```
sale=> SET SEARCH_PATH TO icebear, public;
SET
```

高斯默认有 session 超时时间，若想要 session 一直保持，需要修改配置项：ALTER DATABASE sale SET session_timeout TO 0;

```
postgres=# ALTER DATABASE postgres SET session_timeout TO 0;
ALTER DATABASE
```