

姓名： 汤清云 学号： 2013536

实验步骤：

- 创建和管理用户、表空间和数据库
- 创建和管理表
- 创建和管理其他数据库对象
- 学校数据模型创建及表操作

实验报告

实验步骤截图：

截图 1：指导手册第 8 页，查询表空间当前使用情况截图

```
postgres=# \db
          List of tablespaces
   Name   | Owner | Location
-----+-----+-----
 fastspace | omm   | tablespace/tablespace_1
 pg_default | omm   |
 pg_global | omm   |
(3 rows)

postgres=# SELECT PG_TABLESPACE_SIZE('fastspace');
 pg_tablespace_size
-----
                4096
(1 row)

postgres=#
```

截图 2：指导手册第 10 页，创建表截图

```

postgres=# CREATE TABLE customer_t1
postgres=# (
postgres=#      c_customer_sk          integer,
postgres=#      c_customer_id         char(5),
postgres=#      c_first_name          char(6),
postgres=#      c_last_name           char(8)
postgres=# );
CREATE TABLE
postgres=# INSERT INTO customer_t1(c_customer_sk, c_customer_id, c_first_name) VALUES

```

截图 3：指导手册第 16 页，向分区表中插入数据后查看分区表中所有数据并截图（该命令需自行撰写）

```

postgres=# select * from tpcds.web_returns_p2;

```

ca_address_sk	ca_address_id	ca_street_number	ca_street_name	ca_street_type	ca_suite_number	ca_city	ca_county
a	1	a	1	1.00	a	a	a
b	2	b	2	1.10	b	b	b
c	5050	c	300	1.20	c	c	c
d	14888	d	400	1.50	d	d	d

```

(4 rows)
postgres=#

```

截图 4：指导手册第 19 页，创建分区索引截图。

```

CREATE INDEX
postgres=# CREATE INDEX tpcds_web_returns_p2_index2 ON tpcds.web_returns_p2 (ca_address_sk) LOCAL
postgres=# (
postgres=#      PARTITION web_returns_p2_P1_index,
postgres=#      PARTITION web_returns_p2_P2_index TABLESPACE example3,
postgres=#      PARTITION web_returns_p2_P3_index TABLESPACE example4,
postgres=#      PARTITION web_returns_p2_P4_index,
postgres=#      PARTITION web_returns_p2_P5_index,
postgres=#      PARTITION web_returns_p2_P6_index,
postgres=#      PARTITION web_returns_p2_P7_index,
postgres=#      PARTITION web_returns_p2_P8_index
postgres=# ) TABLESPACE example2;
CREATE INDEX

```

截图 5：指导手册第 23 页，更新物化视图。

```
postgres=# SELECT * FROM MV_MyView;
ca_address_sk | ca_address_id | ca_street_number | ca_street_name | ca_street_type | ca_suite_number | ca_city | ca_county
| ca_state | ca_zip | ca_country | ca_gmt_offset | ca_location_type
-----+-----+-----+-----+-----+-----+-----+-----
| c | 5050 | c | | 300 | | c | | c |
| c | 7050 | c | | 300 | | c | | c |
| c | c | c | | 1.20 | c | | c |
| d | 8888 | d | | 400 | d | | d |
| d | d | d | | 1.50 | d | | d |
| d | 14888 | d | | 400 | d | | d |
| d | d | d | | 1.50 | d | | d |
(4 rows)
```

截图 6：指导手册第 26 页，管理存储过程

```
postgres=# \sf insert_data
CREATE OR REPLACE PROCEDURE public.insert_data()
AS DECLARE
a int;
b int;
begin
a=1;
b=2;
insert into t_test values(a,b);
insert into t_test values(b,a);
end;
/
postgres=# drop procedure insert_data;
DROP PROCEDURE
```

截图 7：指导手册第 39 页，删除数据后表中内容截图

```
postgres=# SELECT * FROM school_department;
depart_id | depart_name | depart_teacher
-----+-----+-----
1 | 计算机学院 | 2
2 | 自动化学院 | 4
3 | 航空宇航学院 | 6
5 | 理学院 | 11
6 | 人工智能学院 | 13
8 | 管理学院 | 17
9 | 农学院 | 22
10 | 医学院 | 28
(8 rows)
```

实验思考题：

1. 在 openGauss 中，创建具有“创建数据库”权限的用户 Alice，并设置其初始密码为“openGauss@0331”，应使用的语句是：

```
CREATE USER Alice CREATEDB PASSWORD 'openGauss@0331';
```

2. 命令 “DROP USER kim CASCADE” 的效果是？（可以预习参考第八周主讲课内容，权限和授权）

删除用户 kim 及 kim 名下所有的表和视图，如果 kim 名下有 schema，则在删除表的时候删除所有与该表相关的主键与外键。

3. 向表中插入数据时，是否允许只对部分属性插入数值？在何种情况下允许，应如何书写语句？何种情况下不允许？

1.允许只对部分属性插入数值。

2.在该表中此属性为非键属性时允许不插入数值，系统将默认为 NULL，创建语句有：

```
INSERT INTO 表名(插入属性, 以逗号分隔) VALUES (具体数值, 以逗号分隔, 字符需带 " );  
INSERT INTO 表名(按顺序输入属性数值);  
INSERT INTO 表名 (属性 1, 属性 2) VALUES (数值, DEFAULT);  
INSERT INTO 表名 DEFAULT VALUES; 此处则均设为空值
```

3.当表中该属性为键属性时不允许为空值。

4. 是否可以向表中一次性插入多条数据？何种插入效率较高？

可以，一次性插入多条数据的效率较高。因为每次调用 insert 都需要与服务器建立一次链接，若多次插入，则服务器负荷增加，效率降低。

5. openGauss 中将表中所有元组删除的两种命令是？

```
DELETE FROM 表名;  
或者 TRUNCATE TABLE 表名;
```

6. 如果经常需要查询某字段值小于某一指定值的信息,可以如何操作?(提示,从索引角度思考)

可以在符合条件的该字段上建立一条索引,之后每次查询时只需要提供索引即可返回所需值。

```
如: CREATE INDEX 索引名称 ON 表名称(该字段名称,如有多个以逗号分隔) WHERE 该字段 < 指定值;
```

7. 在什么场景下可以使用物化视图?物化视图和普通视图的区别是?

1.物化视图使用场景:报表统计、大表统计等,定期固化数据快照,避免对多表重复跑相同的查询。

2. 普通视图是不存储任何数据的,它只有定义,在查询中必须转换为对应的定义 SQL 去查询;而物化视图则将数据转换为实际存储着数据的表,这样查询数据时不用关联许多表,而如果表很大的话,会在临时表空间内做大量的操作。普通视图在多个表关联时只允许查看不允许修改,但物化视图可以修改值,

8. 学校模型 ER 图绘制

