openGauss 数据库开发查询实验

姓名：王娇妹　　学号：2012679

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

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
```

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 | ca_state |   c
a_zip   | ca_country | ca_gmt_offset |   ca_location_type
---------------+------------------+------------------+----------------+-----------------+-----------------+---------+-----------+----------+----
--------+------------+---------------+---------------------
          1 | a                |                1 | a              | a               | a               | a       | a         | a        | a
       | a          |         1.00 | a
          2 | b                |                2 | b              | b               | b               | b       | b         | b        | b
       | b          |         1.10 | b
       5050 | c                |              300 | c              | c               | c               | c       | c         | c        | c
       | c          |         1.20 | c
      14888 | d                |              400 | d              | d               | d               | d       | d         | d        | d
       | d          |         1.50 | d
(4 rows)
```

截图 4：指导手册第 19 页，创建分区索引截图。

```
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 |   c
a_zip   | ca_country | ca_gmt_offset |   ca_location_type
---------------+------------------+------------------+----------------+-----------------+-----------------+---------+-----------+----------+----
--------+------------+---------------+---------------------
       5050 | c                |              300 | c              | c               | c               | c       | c         | c        | c
       | c          |         1.20 | c
       7050 | c                |              300 | c              | c               | c               | c       | c         | c        | c
       | c          |         1.20 | c
       8888 | d                |              400 | d              | d               | d               | d       | d         | d        | d
       | d          |         1.50 | d
      14888 | d                |              400 | d              | d               | d               | d       | d         | d        | d
       | d          |         1.50 | 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;
/
```

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）

   删除数据库中名为 kim 的用户以及该用户下所有对象。

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？

向表中插入数据时，可以只对部分属性插入数值。

允许的情况： INSERT INTO mytable(列 1, 列 2) VALUES (值 1, 值 2);

INSERT 语句后面的列名称顺序可以不是 mytable 表定义时的顺序，即插入数据时，不需要按照表定义的顺序插入，只要保证值的顺序与列字段的顺序相同就可以。

不允许的情况：

虽然使用 INSERT 插入数据时可以忽略插入数据的列名称，若不包含列名称，则 VALUES 关键字后面的值不仅要求完整，而且顺序必须和表定义时列的顺序

相同。此时不允许只对部分属性插入数值。

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？

可以向表中一次性插入多条数据。

如果要向表中插入多条数据，可以使用 INSERT INTO table_name VALUES 命令。可以同时插入多条数据，只需将每条数据用圆括号括起来即可；也可以多次执行插入一条数据命令实现。

在 MySQL 中，用单条 INSERT 语句处理多个插入要比使用多条 INSERT 语句更快。

5. openGauss 中将表中所有元组删除的两种命令是？

（i）DELETE FROM table_name;　（ii）TRUNCATE TABLE table_name;

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）

创建表达式索引。假如经常需要查询 number 小于 1000 的信息，执行如下命令进行查询：

```
SELECT* FROM mytable WHERE trunc(number) < 1000;
```

可以为上面的查询创建表达式索引：

```
CREATE INDEX para_index ON mytable (trunc(number));
```

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？

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。

物化视图和普通视图的区别：

普通视图就是一个虚拟表，不占内存，并不存储任何数据。应用局限性大，

任何对视图的查询，都实际上转换为视图 SQL 语句的查询，普通视图并不能提升查询性能，只是看起来直观，简便。

物化视图是包括一个查询结果的数据库对象，是远程数据的本地副本。物化视图存储基于远程表的数据，占用磁盘存储空间，通过 oracle 的内部机制可以定期更新，将一些大的耗时的表连接用物化视图实现，会提高查询的效率。

8. 学校模型 ER 图绘制