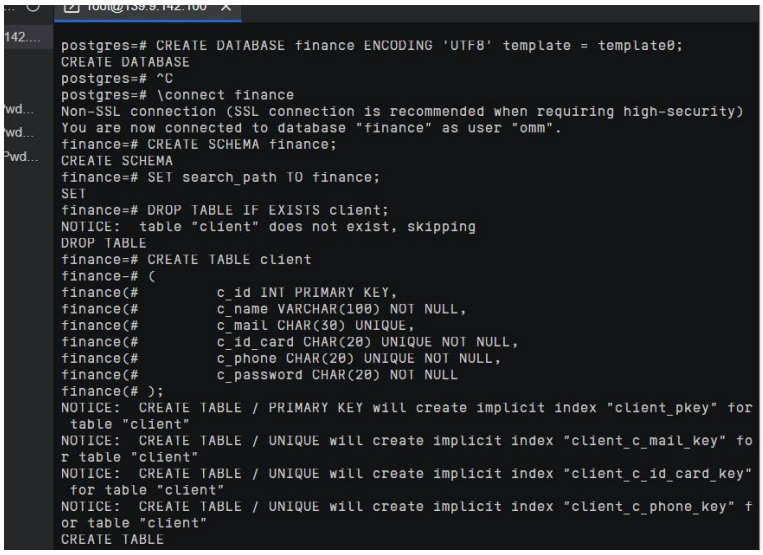
数据库开发查询实验--金融场景

(1)实验环境说明，并说明你选择此实验环境进行实验的原因。

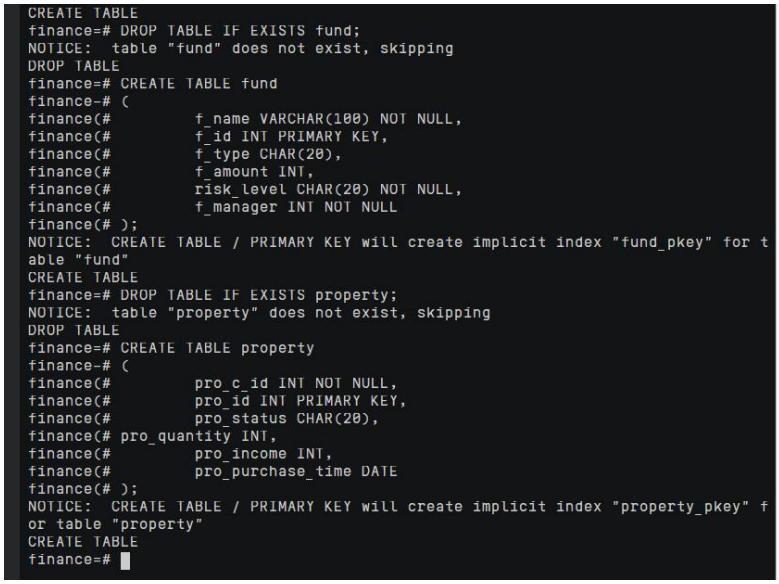
OpenGauss 的优点:高效的存储引擎和查询优化器,数据安全性、可靠性和稳定性，支持主备复制、灾备恢复,支持分布式部署，可以横向扩展，实现可扩展性和高可用性，支持 ANSI SQL 标准和 PostgreSQL 协议，且兼容 PostgreSQL 生态系统，高性能、高可靠、分布式、兼容性、安全性和开源性.

(2)1.1.3-1.1.13 中完成主要步骤后的执行结果截图(每小节截图不少于 2 张)。

1.1.3



```
postgres=# CREATE DATABASE finance ENCODING 'UTF8' template = template0;
CREATE DATABASE
postgres=# ^C
postgres=# \connect finance
Non-SSL connection (SSL connection is recommended when requiring high-security)
You are now connected to database "finance" as user "omm".
finance=# CREATE SCHEMA finance;
CREATE SCHEMA
finance=# SET search_path TO finance;
SET
finance=# DROP TABLE IF EXISTS client;
NOTICE:  table "client" does not exist, skipping
DROP TABLE
finance=# CREATE TABLE client
finance-# (
finance(#         c_id INT PRIMARY KEY,
finance(#         c_name VARCHAR(100) NOT NULL,
finance(#         c_mail CHAR(30) UNIQUE,
finance(#         c_id_card CHAR(20) UNIQUE NOT NULL,
finance(#         c_phone CHAR(20) UNIQUE NOT NULL,
finance(#         c_password CHAR(20) NOT NULL
finance(# );
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "client_pkey" for
 table "client"
NOTICE:  CREATE TABLE / UNIQUE will create implicit index "client_c_mail_key" fo
r table "client"
NOTICE:  CREATE TABLE / UNIQUE will create implicit index "client_c_id_card_key"
 for table "client"
NOTICE:  CREATE TABLE / UNIQUE will create implicit index "client_c_phone_key" f
or table "client"
CREATE TABLE
```



```
CREATE TABLE
finance=# DROP TABLE IF EXISTS fund;
NOTICE:  table "fund" does not exist, skipping
DROP TABLE
finance=# CREATE TABLE fund
finance-# (
finance(#         f_name VARCHAR(100) NOT NULL,
finance(#         f_id INT PRIMARY KEY,
finance(#         f_type CHAR(20),
finance(#         f_amount INT,
finance(#         risk_level CHAR(20) NOT NULL,
finance(#         f_manager INT NOT NULL
finance(# );
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "fund_pkey" for t
able "fund"
CREATE TABLE
finance=# DROP TABLE IF EXISTS property;
NOTICE:  table "property" does not exist, skipping
DROP TABLE
finance=# CREATE TABLE property
finance-# (
finance(#         pro_c_id INT NOT NULL,
finance(#         pro_id INT PRIMARY KEY,
finance(#         pro_status CHAR(20),
finance(# pro_quantity INT,
finance(#         pro_income INT,
finance(#         pro_purchase_time DATE
finance(# );
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "property_pkey" f
or table "property"
CREATE TABLE
finance=#
```

1.1.4

```
finance=# INSERT INTO client(c_id,c_name,c_mail,c_id_card,c_phone,c_password) VA
LUES (1,'张一','zhangyi@huawei.com','340211199301010001','18815650001','gaussdb_
001');
INSERT 0 1
finance=# INSERT INTO client(c_id,c_name,c_mail,c_id_card,c_phone,c_password) VA
LUES (2,'张二','zhanger@huawei.com','340211199301010002','18815650002','gaussdb_
002');
INSERT 0 1
finance=# INSERT INTO client(c_id,c_name,c_mail,c_id_card,c_phone,c_password) VA
LUES (3,'张三','zhangsan@huawei.com','340211199301010003','18815650003','gaussdb
_003');
INSERT 0 1
finance=# INSERT INTO client(c_id,c_name,c_mail,c_id_card,c_phone,c_password) VA
LUES (4,'张四','zhangsi@huawei.com','340211199301010004','18815650004','gaussdb_
004');
INSERT 0 1
finance=# INSERT INTO client(c_id,c_name,c_mail,c_id_card,c_phone,c_password) VA
LUES (5,'张五','zhangwu@huawei.com','340211199301010005','18815650005','gaussdb_
005');
INSERT 0 1
finance=# INSERT INTO client(c_id,c_name,c_mail,c_id_card,c_phone,c_password) VA
LUES (6,'张六','zhangliu@huawei.com','340211199301010006','18815650006','gaussdb
_006');
INSERT 0 1
finance=# INSERT INTO client(c_id,c_name,c_mail,c_id_card,c_phone,c_password) VA
LUES (7,'张七','zhangqi@huawei.com','340211199301010007','18815650007','gaussdb_
007');
INSERT 0 1
finance=# INSERT INTO client(c_id,c_name,c_mail,c_id_card,c_phone,c_password) VA
LUES (8,'张八','zhangba@huawei.com','340211199301010008','18815650008','gaussdb_
```

```
 count
-------
     4
(1 row)

finance=# INSERT INTO property(pro_c_id,pro_id,pro_status,pro_quantity,pro_incom
e,pro_purchase_time) VALUES (5,1,'可用',4,8000,'2018-07-01');
INSERT 0 1
finance=# INSERT INTO property(pro_c_id,pro_id,pro_status,pro_quantity,pro_incom
e,pro_purchase_time) VALUES (10,2,'可用',4,8000,'2018-07-01');
INSERT 0 1
finance=# INSERT INTO property(pro_c_id,pro_id,pro_status,pro_quantity,pro_incom
e,pro_purchase_time) VALUES (15,3,'可用',4,8000,'2018-07-01');
INSERT 0 1
finance=# INSERT INTO property(pro_c_id,pro_id,pro_status,pro_quantity,pro_incom
e,pro_purchase_time) VALUES (20,4,'冻结',4,8000,'2018-07-01');
INSERT 0 1
finance=# select count(*) from property;
 count
-------
     4
(1 row)

finance=#
```

1.1.5

```
finance=# INSERT INTO client(c_id,c_name,c_mail,c_id_card,c_phone,c_password) VA
LUES (31,'李丽','lili@huawei.com','340211199301010005','18815650005','gaussdb_00
5');
ERROR:  duplicate key value violates unique constraint "client_c_id_card_key"
DETAIL:  Key (c_id_card)=(340211199301010005  ) already exists.
finance=# INSERT INTO client(c_id,c_name,c_mail,c_id_card,c_phone,c_password) VA
LUES (31,'李丽','lili@huawei.com','340211199301010031','18815650031','gaussdb_03
1');
INSERT 0 1
finance=#
```

1.1.6

```
INSERT 0 1
finance=# ALTER table finances_product ADD CONSTRAINT c_p_mount CHECK (p_amount
>=0);
ALTER TABLE
finance=# INSERT INTO finances_product(p_name,p_id,p_description,p_amount,p_year
) VALUES ('信贷资产',10,'一般指银行作为委托人将通过发行理财产品募集资金委托给信
托公司，信托公司作为受托人成立信托计划，将信托资产购买理财产品发售银行或第三方信
贷资产。',-10,6);
ERROR:  new row for relation "finances_product" violates check constraint "c_p_m
ount"
DETAIL:  Failing row contains (信贷资产, 10, 一般指银行作为委托人将通过发行理财
产品募集..., -10, 6).
finance=# ALTER table fund ADD CONSTRAINT c_f_mount CHECK (f_amount >=0);
ALTER TABLE
finance=# ALTER table insurance ADD CONSTRAINT c_i_mount CHECK (i_amount >=0);
ALTER TABLE
finance=#
```

1.1.7 查询数据

```
finance=# select b_number,b_type from bank_card;
          b_number           |      b_type
-----------------------------+------------------
 6222021302020000001         | 信用卡
 6222021302020000002         | 信用卡
 6222021302020000003         | 信用卡
 6222021302020000004         | 信用卡
 6222021302020000005         | 信用卡
 6222021302020000006         | 信用卡
 6222021302020000007         | 信用卡
 6222021302020000008         | 信用卡
 6222021302020000009         | 信用卡
 6222021302020000010         | 信用卡
 6222021302020000011         | 储蓄卡
 6222021302020000012         | 储蓄卡
 6222021302020000013         | 储蓄卡
 6222021302020000014         | 储蓄卡
 6222021302020000015         | 储蓄卡
 6222021302020000016         | 储蓄卡
 6222021302020000017         | 储蓄卡
 6222021302020000018         | 储蓄卡
 6222021302020000019         | 储蓄卡
 6222021302020000020         | 储蓄卡
(20 rows)
```

```
finance=# select * from property where pro_status = '可用'
finance-# ;
 pro_c_id | pro_id |     pro_status     | pro_quantity | pro_income |  pro
_purchase_time
----------+--------+--------------------+--------------+------------+-----
----------------
        5 |      1 | 可用               |            4 |       8000 | 2018
-07-01 00:00:00
       10 |      2 | 可用               |            4 |       8000 | 2018
-07-01 00:00:00
       15 |      3 | 可用               |            4 |       8000 | 2018
-07-01 00:00:00
(3 rows)
```

```
finance=# select count(*) from client;
 count
-------
    31
(1 row)
```

```
finance=# select b_type,count(*) from bank_card group by b_type;
      b_type       | count
-------------------+-------
 储蓄卡            |    10
 信用卡            |    10
(2 rows)
```
(3)

```
finance=# select avg(i_amount) from insurance;
          avg
------------------------
 2700.0000000000000000
(1 row)
```

```
finance=# select i_name, i_amount from insurance where i_amount in (select
 max(i_amount) from insurance)
finance-# union
finance-# select i_name, i_amount from insurance where i_amount in (select
 min(i_amount) from insurance);
    i_name     | i_amount
---------------+----------
 意外保险      |     5000
 财产损失保险  |     1500
(2 rows)
```

```
finance=# select c_id,c_name, c_id_card from client where c_id not in(select b_c_id from bank_card where b_number like '622202150202000001_');
 c_id | c_name |      c_id_card
------+--------+---------------------
    1 | 张一   | 340211199301010001
    2 | 张二   | 340211199301010002
    3 | 张三   | 340211199301010003
```

```
finance=# select c_id,c_name,c_id_card from client where exists(select * from bank_card where client.c_id = bank_card.b_c_id);
 c_id | c_name |      c_id_card
------+--------+---------------------
    1 | 张一   | 340211199301010001
    3 | 张三   | 340211199301010003
    5 | 张五   | 340211199301010005
    7 | 张七   | 340211199301010007
    9 | 张九   | 340211199301010009
   10 | 李一   | 340211199301010010
   12 | 李三   | 340211199301010012
   14 | 李五   | 340211199301010014
   16 | 李七   | 340211199301010016
   18 | 李九   | 340211199301010018
```

```
finance=# select i1.i_name,i1.i_amount,i1.i_person from insurance i1
finance-# where i_amount > (select avg(i_amount) from insurance i2);
   i_name   | i_amount |      i_person
------------+----------+---------------------
 人寿保险   |     3000 | 老人
 意外保险   |     5000 | 所有人
(2 rows)
```

```
finance=# select i_name,i_amount,i_person from insurance where i_id > 2
finance-# order by i_amount desc;
    i_name     | i_amount |     i_person
---------------+----------+------------------
 意外保险      |     5000 | 所有人
 医疗保险      |     2000 | 所有人
 财产损失保险  |     1500 | 中年人
(3 rows)
```

```
finance=# select p_year,count(p_id) from finances_product group by p_year
 p_year | count
--------+-------
      6 |     4
(1 row)
```

```
finance=# select i_person,count(i_amount) from insurance group by i_pe
 having count(i_amount) = 2;
      i_person       | count
---------------------+-------
 老人                |     2
 所有人              |     2
(2 rows)
```

```
finance=# with temp as (select f_name,ln(f_amount) from fund order by f_ma
nager desc) select * from temp;
   f_name     |        ln
--------------+------------------
 沪深300指数  | 9.21034037197618
 国债         | 9.21034037197618
 投资         | 9.21034037197618
 股票         | 9.21034037197618
(4 rows)
```

对于 1.1.7 中的每个查询需求，请分别提供对应的 SQL 查询语句和能够满足查询需求的关系代数表达式。

SQL 查询语句如截图：

关系代数表达式为：

1.1.8

```
finance=# create view v_client as select c_id,c_name,c_id_card from client where exists (select * from bank_card where client.c_id = bank_card.b_c_id);
CREATE VIEW
finance=# select * from v_client;
 c_id | c_name |      c_id_card
------+--------+---------------------
    1 | 张一   | 340211199301010001
    3 | 张三   | 340211199301010003
    5 | 张五   | 340211199301010005
```

```
finance=# select * from v_client;
 c_id | c_name |      c_id_card
------+--------+---------------------
    1 | 张一   | 340211199301010001
    3 | 张三   | 340211199301010003
    5 | 张五   | 340211199301010005
    7 | 张七   | 340211199301010007
```

```
finance=# alter view v_client rename to v_client_new;
ALTER VIEW
finance=# drop view v_client_new;
DROP VIEW
finance=#
```

1.1.9

```
finance=# create index idx_property on property(pro_c_id desc,pro_income,p
ro_purchase_time);
CREATE INDEX
finance=# alter index idx_property rename to idx_property_temp;
ALTER INDEX
finance=# drop index idx_properety_temp;
ERROR:  index "idx_properety_temp" does not exist
finance=# drop index idx_property_temp;
DROP INDEX
finance=#
```

1.1.10

```
 6222021302020000017            | 储蓄卡             |     26
 6222021302020000018            | 储蓄卡             |     27
 6222021302020000020            | 储蓄卡             |     29
(20 rows)

finance=# select * from fund;
   f_name    | f_id |      f_type       | f_amount |     risk_level      |
 f_manager
-------------+------+-------------------+----------+---------------------+
-----------
 股票        |    1 | 股票型            |    10000 | 高                  |
       1
 投资        |    2 | 债券型            |    10000 | 中                  |
       2
 国债        |    3 | 货币型            |    10000 | 低                  |
       3
 沪深300指数 |    4 | 指数型            |    10000 | 中                  |
       4
(4 rows)

...skipping 1 line
finance=# delete from fund where f_id < 3;
DELETE 2
finance=# select* from fund;
   f_name    | f_id |      f_type       | f_amount |     risk_level      |
 f_manager
-------------+------+-------------------+----------+---------------------+
-----------
 国债        |    3 | 货币型            |    10000 | 低                  |
       3
 沪深300指数 |    4 | 指数型            |    10000 | 中                  |
       4
(2 rows)
```

1.1.11

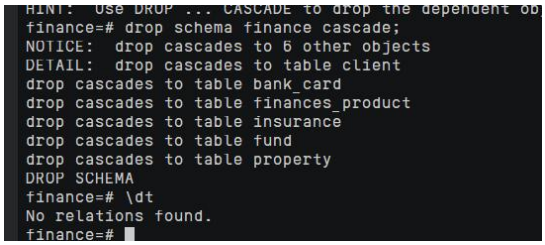
```
finance=# create user dbuser identified by 'Gauss#3demo';
CREATE ROLE
finance=# grant select,insert on finance.bank_card to dbuser;
GRANT
finance=# grant all on schema finance to dbuser;
GRANT
finance=#
```

1.1.12

```
[omm@ecs-c55f ~]$ gs_om -t start;
Starting cluster.
=========================================
[SUCCESS] ecs-c55f:
[2023-05-08 11:04:14.152][6297][][gs_ctl]: gs_ctl started,datadir is /gaussdb/data/db1
[2023-05-08 11:04:14.157][6297][][gs_ctl]:  another server might be running; Please use the restart command
=========================================
Successfully started.
[omm@ecs-c55f ~]$ gsql -d finance -U dbuser -p 26000;
Password for user dbuser:
gsql ((openGauss 2.0.0 build 78689da9) compiled at 2021-03-31 21:03:52 commit 0 last mr  )
Non-SSL connection (SSL connection is recommended when requiring high-security)
Type "help" for help.

finance=> select * from finance.bank_card where b_c_id < 10;
             b_number             |       b_type       | b_c
_id
----------------------------------+--------------------+----
----
 6222021302020000001              | 借记卡             |
  1
 6222021302020000002              | 借记卡             |
  3
 6222021302020000003              | 借记卡             |
  5
 6222021302020000004              | 借记卡             |
```

1.1.13

```
finance=# \dn
   List of schemas
    Name     | Owner
-------------+--------
 cstore      | omm
 dbe_perf    | omm
 dbuser      | dbuser
 finance     | omm
 pkg_service | omm
 public      | omm
 snapshot    | omm
(7 rows)

finance=# set search_path to finance;
SET
finance=# \dt
                        List of r
 Schema  |       Name       | Type  | Ow

---------+------------------+-------+---
-----
 finance | bank_card        | table | omm
=no}
```



关系代数：

（4）如果你的初始 SQL 执行结果和要求的执行结果不符，其原因是什么？请就和要求结果不符的 SQL 执行内容分别进行说明。

不那么熟悉 sql 语句，要多练习

（5）实验总时长分析及遇到的问题、以及实验中学习到的知识点分析。

数据记录的插入、查询、更新和删除

- INSERT INTO table_name (column1, column2, column3, ...) VALUES (value1, value2, value3, ...);（插入数据记录）
- SELECT column1, column2, ... FROM table_name WHERE condition;（查询数据记录）
- UPDATE table_name SET column1 = value1, column2 = value2, ... WHERE condition;（更新数据记录）
- DELETE FROM table_name WHERE condition;（删除数据记录）

数据库的聚合函数

- COUNT：返回某个列中非空值的数量。
- SUM：返回某个列中所有值的总和。
- AVG：返回某个列中所有值的平均值。
- MAX：返回某个列中所有值的最大值。
- MIN：返回某个列中所有值的最小值。