GaussDB数据库实验一：SQL基础实验

姓名：**陶拥鼎**
学号：**2021110801011**

0. 基本操作

```
\l  列出所有数据库
\d tablename 列出指定表的所有字段
\d+ tablename 查看指定表的基本情况
\d 列出当前数据库下的表
\c database_name 切换数据库
\dn 展示当前数据库下所有schema信息
\du 列出角色
\dv 列表视图
\di 列表索引
\q 退出登录
gsql -d 数据库名字 -p 端口 -U 用户名字 -W '密码' -r
cm_ctl query -Cvipd 查询集群的状态
```

```
SHOW search_path; 显示当前使用的schema
SET search_path TO myschema; 切换当前schema
```

1. 实验目标

- 掌握数据初始化的方法;
- 掌握数据查询语句`SELECT`，包括基本查询，统计查询，连接查询，子查询，查询集合等;
- 掌握数据更新语句，包括`INSERT`，`DELETE`，`UPDATE`.

2. 数据初始化

1.雇佣历史表

```
create table employment_history(
Staff_id number,
Start_date date,
End_date date,
Employment_id varchar2(10),
Section_id number(4)
);
```

2.部门表

```
create table sections(
section_id number,
section_name varchar2(30),
manager_id number,
place_id number
);
```

3.工作地点表 places

```
create table places(
place_id number,
street_address varchar2(40),
postal_code varchar2(12),
city varchar2(30),
state_province varchar2(25),
state_id char(10)
);
```

4.区域表 areas

```
create table areas(
area_id number,
area_name varchar2(25)
);
```

5.大学表 college

```
create table college(
college_id number,
college_name varchar2(40)
);
```

6.雇佣表 employments

```
create table employments(
employment_id varchar2(10),
employment_title varchar2(35),
min_salary number,
max_salary number
);
```

7.国家及地区表 states

```
create table states(
state_id char(2),
state_name varchar2(40),
area_id number
);
```

8.员工表 staffs

```
create table staffs(
staff_id number,
first_name varchar2(40),
last_name varchar2(12),
email varchar2(30),
phone_number varchar2(25),
hire_date date,
employment_id varchar2(10),
salary number,
commission_pct number,
manager_id number,
section_id number
);
```

3. 实验任务

0. 实验准备

- ECS开机
- 使用putty远程连接服务器
- 以omm的用户登录服务器
- 开启gauss数据库

```
[omm@ecs-8342 ~]$ gs_om -t start
Starting cluster.
=========================================
[SUCCESS] ecs-8342
2023-05-07 15:50:26.555 64575842.1 [unknown] 281469543776272 [unknown] 0 dn_6001
01000  0 [BACKEND] WARNING:  could not create any HA TCP/IP sockets
=========================================
Successfully started.
```

- 忘了之前创建的人力资源数据库的名字和用户，连接`postgres`数据库查看

```
[omm@ecs-8342 ~]$ gsql -d postgres -p 26000
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.

postgres=# \l
                               List of databases
      Name      |    Owner    | Encoding | Collate | Ctype | Access privileges
----------------+-------------+----------+---------+-------+-------------------
 askerdb        | asker       | UTF8     | C       | C     |
 db_tpcc        | joe         | UTF8     | C       | C     |
 human_resource | taoyongding | UTF8     | C       | C     |
 postgres       | omm         | UTF8     | C       | C     |
 template0      | omm         | UTF8     | C       | C     | =c/omm            +
                |             |          |          |        | omm=CTc/omm
 template1      | omm         | UTF8     | C        | C      | =c/omm          +
                |             |          |          |        | omm=CTc/omm
(6 rows)

postgres=# \q
```

- 以 `taoyongding` 用户登录 `human_resource` 数据库，端口号为 `26000`

```
[omm@ecs-8342 ~]$ gsql -d human_resource -p 26000 -U taoyongding -W
taoyongding@123 -r
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.

human_resource=>
```

- 根据给定信息初始化数据表

1. 查看雇佣表的所有信息

- SQL code

```
SELECT * FROM employments;
```

- results

```

 employment_id |        employment_title          | min_salary | max_salary
---------------+----------------------------------+------------+------------
 AD_PRES       | President                        |      20000 |      40000
 AD_VP         | Administration Vice President    |      15000 |      30000
 AD_ASST       | Administration Assistant         |       3000 |       6000
 FI_MGR        | Finance Manager                  |       8200 |      16000
 FI_ACCOUNT    | Accountant                       |       4200 |       9000
 AC_MGR        | Accounting Manager               |       8200 |      16000
 AC_ACCOUNT    | Public Accountant                |       4200 |       9000
 SA_MAN        | Sales Manager                    |      10000 |      20000
 SA_REP        | Sales Representative             |       6000 |      12000
 PU_MAN        | Purchasing Manager               |       8000 |      15000
 PU_CLERK      | Purchasing Clerk                 |       2500 |       5500
 ST_MAN        | Stock Manager                    |       5500 |       8500
 ST_CLERK      | Stock Clerk                      |       2000 |       5000
 SH_CLERK      | Shipping Clerk                   |       2500 |       5500
 IT_PROG       | Programmer                       |       4000 |      10000
 MK_MAN        | Marketing Manager                |       9000 |      15000
 MK_REP        | Marketing Representative         |       4000 |       9000
 HR_REP        | Human Resources Representative   |       4000 |       9000
 PR_REP        | Public Relations Representative  |       4500 |      10500
(19 rows)
```

2. 查询编号为60的部门名称

- SQL code

```sql
select section_name from sections where section_id = 60;
```

- results

```
 section_name
--------------
 IT
(1 row)
```

3. 查询工资最高的五名员工，返回员工编号及员工姓名

- SQL code

```sql
SELECT staff_id, first_name || ' ' || last_name as full_name
FROM staffs
ORDER BY salary DESC
LIMIT 5;
```

- results

```
 staff_id |     full_name
----------+-------------------
      100 | Steven King
      101 | Neena Kochhar
      102 | Lex De Haan
      201 | Michael Hartstein
      205 | Shelley Higgins
(5 rows)
```

4. 查询编号为201员工的部门经理编号及其姓名

- SQL code

```sql
SELECT s.manager_id, m.first_name || ' ' || m.last_name as manager_name
FROM staffs s
INNER JOIN staffs m ON s.manager_id = m.staff_id
WHERE s.staff_id = 201;
```

- results

```
 manager_id | manager_name
------------+--------------
        100 | Steven King
(1 row)
```

5. 查询工资差距最大的职位，返回职位编号，和职位名称

- 法1
- SQL code

```sql
SELECT employment_id, employment_title
FROM employments
ORDER BY (max_salary - min_salary) DESC
LIMIT 1;
```

- results

```
employment_id | employment_title
---------------+------------------
 AD_PRES       | President
(1 row)
```

- 法2
- SQL code

```sql
SELECT employment_id, employment_title
FROM employments
WHERE max_salary-min_salary in (SELECT MAX(max_salary-min_salary) FROM
employments);
```

- results

```
 employment_id | employment_title
---------------+------------------
 AD_PRES       | President
(1 row)
```

6. 查询各部门工资最高的员工姓名，及其对应的部门名称

- SQL code

```sql
SELECT d.section_name, s.first_name || ' ' || s.last_name AS full_name
FROM (
  SELECT section_id, MAX(salary) AS max_salary
  FROM staffs
  GROUP BY section_id
) m
INNER JOIN staffs s ON m.section_id = s.section_id AND m.max_salary = s.salary
INNER JOIN sections d ON s.section_id = d.section_id;
```

- results

```
   section_name    |     full_name
-------------------+-------------------
 Administration    | Jennifer Whalen
 Marketing         | Michael Hartstein
 Human Resources   | Susan Mavris
 Shipping          | Douglas Grant
 Shipping          | Donald OConnell
 IT                | Alexander Hunold
 Public Relations  | Hermann Baer
 Executive         | Steven King
 Finance           | Nancy Greenberg
 Accounting        | Shelley Higgins
(10 rows)
```

Reference

7. 查询雇佣历史表中担任过AC_ACCOUNT和AC_MGR 职位的员工编号

- SQL code

```
SELECT staff_id
FROM employment_history
WHERE employment_id IN ('AC_ACCOUNT', 'AC_MGR')
GROUP BY staff_id
HAVING COUNT(DISTINCT employment_id) = 2;
```

- results

```
 staff_id
----------
      101
(1 row)
```

8. 查询雇佣历史表中员工的雇佣时长，返回并显示如下信息：员工编号，职位编号，部门编号，雇佣时长

- SQL code

```
SELECT staff_id, employment_id, section_id, (end_date-start_date) AS
employment_time
FROM employment_history;
```

- results

```
 staff_id | employment_id | section_id | employment_time
----------+---------------+------------+-----------------
      102 | IT_PROG       |         60 | 2018 days
      101 | AC_ACCOUNT    |        110 | 1497 days
      101 | AC_MGR        |        110 | 1234 days
      201 | MK_REP        |         20 | 1401 days
      114 | ST_CLERK      |         50 | 647 days
      122 | ST_CLERK      |         50 | 364 days
      200 | AD_ASST       |         90 | 2100 days
      176 | SA_REP        |         80 | 647 days
      176 | SA_MAN        |         80 | 364 days
      200 | AC_ACCOUNT    |         90 | 1644 days
(10 rows)
```

9. 查询在城市（city）South San Francisco工作的员工编号和员工姓名，按工资降序排列

- SQL code

```
SELECT staff_id, first_name || ' ' || last_name AS full_name, salary
FROM staffs
WHERE section_id IN
(SELECT section_id
FROM sections
WHERE place_id IN
(SELECT place_id FROM places WHERE city = 'South San Francisco'))
ORDER BY salary DESC;
```

- results

```
 staff_id |     full_name     | salary
----------+-------------------+---------
      199 | Douglas Grant     | 2600.00
      198 | Donald OConnell   | 2600.00
```

10. 查询员工平均工资在5000以上的部门，返回部门编号及部门名称

- SQL code

```
SELECT s.section_id, s.section_name
FROM
(SELECT section_id, AVG(salary) AS avg_salary
FROM staffs
GROUP BY section_id
HAVING AVG(salary) > 5000) m
INNER JOIN sections s ON m.section_id = s.section_id;
```

- results

```
 section_id |   section_name
------------+------------------
         20 | Marketing
         40 | Human Resources
         60 | IT
         70 | Public Relations
         90 | Executive
        100 | Finance
        110 | Accounting
(7 rows)
```

11. 查询last_name 以 字母F开头的员工，返回员工编号和姓名

- SQL code

```
SELECT staff_id, first_name || ' ' || last_name AS full_name
FROM staffs
WHERE last_name LIKE 'F%';
```

- results

```
 staff_id |   full_name
----------+---------------
      202 | Pat Fay
      109 | Daniel Faviet
(2 rows)
```

12. 查询雇佣历史表中在两个及以上不同职位工作过的员工，返回员工编号

- SQL code

```
SELECT staff_id
FROM employment_history
GROUP BY staff_id
HAVING COUNT(DISTINCT employment_id) >= 2;
```

- results

```
 staff_id
----------
      101
      176
      200
(3 rows)
```

13. 查询各个国家办事处的数量

- SQL code

```
SELECT s.state_name, m.count
FROM
(SELECT state_id, COUNT(place_id) AS count
FROM places
GROUP BY state_id) m
INNER JOIN states s ON m.state_id = s.state_id
ORDER BY m.count DESC;
```

- results

```
        state_name        | count
--------------------------+-------
 United States of America |     4
 United Kingdom           |     3
 Italy                    |     2
 Japan                    |     2
 Canada                   |     2
 Switzerland              |     2
 Australia                |     1
 Netherlands              |     1
 Brazil                   |     1
 Germany                  |     1
 Mexico                   |     1
 Singapore                |     1
 India                    |     1
 China                    |     1
(14 rows)
```

14. 在员工表中新增一位员工信息（内容自拟，但符合其他表约束）

- SQL code

```
INSERT INTO staffs (staff_id, first_name, last_name, email, phone_number,
hire_date,employment_id, salary, commission_pct, manager_id, section_id)
VALUES (111, 'Yongding', 'Tao', 'MYEMAIL', '123.456.7890', to_date('19-10-2002',
'dd-mm-yyyy'), 'IT_PROG', 88888.88, null, 103, 60);

SELECT * FROM staffs WHERE last_name = 'Tao';
```

- results

```
 staff_id | first_name | last_name |  email   | phone_number |       hire_date
   | employment_id |  salary  | commission_pct | manager_id | section_id
----------+------------+-----------+---------+--------------+-------------------
--+---------------+----------+----------------+------------+------------
      111 | Yongding   | Tao        | MYEMAIL | 123.456.7890 | 2002-10-19 00:00:0
0 | IT_PROG       | 88888.88 |                |        103 |          60
(1 row)
```

15. 为平均工资在5000及以下的部门 每位员工加薪1000元

- 修改前，查询工资

```
SELECT section_id, staff_id, salary
FROM staffs
WHERE section_id IN
(SELECT section_id
FROM staffs
GROUP BY section_id
HAVING AVG(salary) <= 5000);
```

- results

```
 section_id | staff_id | salary
------------+----------+---------
         50 |      198 | 2600.00
         50 |      199 | 2600.00
         10 |      200 | 4400.00
(3 rows)
```

- SQL code

```
UPDATE staffs
SET salary = salary + 1000
WHERE section_id in
(SELECT section_id
FROM staffs
GROUP BY section_id
HAVING AVG(salary) <= 5000);
```

- 修改后，查询工资

```
SELECT section_id, staff_id, salary
FROM staffs
WHERE section_id IN (10, 50);
```

- results

```
 section_id | staff_id | salary
------------+----------+---------
         50 |      198 | 3600.00
         50 |      199 | 3600.00
         10 |      200 | 5400.00
(3 rows)
```