数据库系统课程实验报告

实验名称：	实验二：数据库与基本表的创建、修改与删除
实验日期：	2022/3/24
实验地点：	厦门大学德旺图书馆
提交日期：	2022/3/26

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

- 理解 openGauss 中用户、数据库、模式和表等数据库对象之间的关系。掌握用户、数据库、模式和基本表的创建、修改和删除方法
- 掌握数据类型的选择和使用
- 掌握数据的导入导出方法

2. 实验内容和步骤

（1）理解 Sales 数据库中各表及各表之间的联系

（2）创建新用户、数据库和模式；新用户名用名字拼音 xiaoping

步骤如下：

1. 以 root 用户登录到 ECS 服务器

```
>_ root@114.116.235.207 ×

Welcome to 4.19.90-2003.4.0.0036.oe1.aarch64

System information as of time:  Fri Mar 25 10:03:42 CST 2022

System load:    0.72
Processes:      109
Memory used:    24.7%
Swap used:      0.0%
Usage On:       15%
IP address:     192.168.0.183
Users online:   1
```

2. 以 omm 操作系统管理员身份登录数据库

```
[root@ecs-ad18 ~]# su - omm
Last login: Thu Mar 24 21:15:35 CST 2022 on pts/0


Welcome to 4.19.90-2003.4.0.0036.oe1.aarch64

System information as of time:  Fri Mar 25 10:04:08 CST 2022

System load:    0.61
Processes:      111
Memory used:    24.8%
Swap used:      0.0%
Usage On:       15%
IP address:     192.168.0.183
Users online:   1
```

3. 查看服务是否启动，启动服务器。

```
[omm@ecs-ad18 ~]$ gs_om -t status
-----------------------------------------------------------------------

cluster_name    : dbCluster
cluster_state   : Normal
redistributing  : No

-----------------------------------------------------------------------
[omm@ecs-ad18 ~]$ gs_om -t start
Starting cluster.
=========================================
[SUCCESS] ecs-ad18:
[2022-03-25 10:04:31.724][42514][][gs_ctl]: gs_ctl started,datadir is /gaussdb/data/db1
[2022-03-25 10:04:31.728][42514][][gs_ctl]:  another server might be running; Please use the restart command
=========================================
Successfully started.
```

4. 使用 gsql 连接到数据库。

```
[omm@ecs-ad18 ~]$ gsql -d postgres -p 26000 -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.
```

5. 创建数据库用户，用户名为 yuxiaoping，密码为 yuxiaoping@123。

创建数据库 sale。

```
postgres=# CREATE USER yuxiaoping WITH PASSWORD "yuxiaoping@123";
CREATE ROLE
postgres=# CREATE DATABASE sale OWNER yuxiaoping;
CREATE DATABASE
```

6. 进入数据库。

```
postgres=# \q
[omm@ecs-ad18 ~]$ gsql -d sale -p 26000 -U xiaoping -W yuxiaoping@123  -r
gsql: FATAL:  Invalid username/password,login denied.
[omm@ecs-ad18 ~]$ gsql -d sale -p 26000 -U yuxiaoping -W yuxiaoping@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.

sale=>
```

7. 创建名为 icebear（我的英文名）的 SCHEMA，并设置 yuxiaoping 为当前的 role,将默认搜索路径设为 icebear。

```
sale=> CREATE SCHEMA icebear AUTHORIZATION yuxiaoping;
CREATE SCHEMA
sale=> SET search_path TO icebear;
SET
```

（3）分别创建 12 张表，创建时不添加约束。

```
sale=> CREATE TABLE regions(region_id NUMBER ,region_name VARCHAR2( 50 ) );
CREATE TABLE
sale=>  CREATE TABLE countries
sale->   (
sale(>     country_id   CHAR( 2 ) ,
sale(>     country_name VARCHAR2( 40 ) ,
sale(>     region_id    NUMBER
sale(>   );
CREATE TABLE
sale=> CREATE TABLE locations
sale->   (
sale(>     location_id NUMBER ,
sale(>     address     VARCHAR2( 255 ),
sale(>     postal_code VARCHAR2( 20 )          ,
sale(>     city        VARCHAR2( 50 )          ,
sale(>     state       VARCHAR2( 50 )          ,
sale(>     country_id  CHAR( 2 )
sale(>   );
CREATE TABLE
sale=>   CREATE TABLE warehouses
sale->   (
sale(>     warehouse_id NUMBER ,
sale(>     warehouse_name VARCHAR( 255 ) ,
sale(>     location_id    NUMBER( 12, 0 )
sale(>   );
CREATE TABLE
sale=>  CREATE TABLE inventories
sale->   (
sale(>     product_id   NUMBER( 12, 0 )        ,
sale(>     warehouse_id NUMBER( 12, 0 )        ,
sale(>     quantity     NUMBER( 8, 0 ) NOT NULL
sale(>   );
CREATE TABLE
```

```
sale=>   CREATE TABLE employees
sale->   (
sale(>     employee_id NUMBER ,
sale(>     first_name VARCHAR( 255 ) ,
sale(>     last_name  VARCHAR( 255 ) ,
sale(>     email      VARCHAR( 255 ) ,
sale(>     phone      VARCHAR( 50 ) ,
sale(>     hire_date  DATE  ,
sale(>     manager_id NUMBER( 12, 0 )  ,
sale(>     job_title  VARCHAR( 255 )
sale(>   );
CREATE TABLE
sale=>   CREATE TABLE product_categories
sale->   (
sale(>     category_id NUMBER ,
sale(>     category_name VARCHAR2( 255 )
sale(>   );
CREATE TABLE
```

```
sale=>   CREATE TABLE customers
sale->   (
sale(>     customer_id NUMBER ,
sale(>     name        VARCHAR2( 255 ) ,
sale(>     address     VARCHAR2( 255 )        ,
sale(>     website     VARCHAR2( 255 )        ,
sale(>     credit_limit NUMBER( 8, 2 )
sale(>   );
CREATE TABLE
sale=>   CREATE TABLE contacts
sale->   (
sale(>     contact_id NUMBER ,
sale(>     first_name  VARCHAR2( 255 ) ,
sale(>     last_name   VARCHAR2( 255 ) ,
sale(>     email       VARCHAR2( 255 ) ,
sale(>     phone       VARCHAR2( 20 )         ,
sale(>     customer_id NUMBER
sale(>   );
CREATE TABLE
sale=> CREATE TABLE orders
sale->   (
sale(>     order_id NUMBER ,
sale(>     customer_id NUMBER( 6, 0 ) , -- fk
sale(>     status      VARCHAR( 20 )  ,
sale(>     salesman_id NUMBER( 6, 0 )        , -- fk
sale(>     order_date  DATE
sale(>   );
CREATE TABLE
```

```
sale=>   CREATE TABLE order_items
sale->   (
sale(>     order_id   NUMBER( 12, 0 )                                  , -- fk
sale(>     item_id    NUMBER( 12, 0 )                                  ,
sale(>     product_id NUMBER( 12, 0 )                        , -- fk
sale(>     quantity   NUMBER( 8, 2 )                        ,
sale(>     unit_price NUMBER( 8, 2 )
sale(>   );
CREATE TABLE
sale=>   CREATE TABLE products
sale->   (
sale(>     product_id NUMBER ,
sale(>     product_name  VARCHAR2( 255 ) ,
sale(>     description   VARCHAR2( 2000 )        ,
sale(>     standard_cost NUMBER( 9, 2 )          ,
sale(>     list_price    NUMBER( 9, 2 )          ,
sale(>     category_id   NUMBER
sale(>   );
CREATE TABLE
sale=>
```

（4）查看建立的表。

```
 icebear           | regions                   | yuxiaoping |           | f         | f       | f          | yuxiaoping   | 2022-03-25 10:50:14.0
42882+08 | 2022-03-25 10:50:14.042882+08
 icebear           | countries                 | yuxiaoping |           | f         | f       | f          | yuxiaoping   | 2022-03-25 10:51:29.0
50964+08 | 2022-03-25 10:51:29.050964+08
 icebear           | locations                 | yuxiaoping |           | f         | f       | f          | yuxiaoping   | 2022-03-25 10:51:51.2
33486+08 | 2022-03-25 10:51:51.233486+08
 icebear           | warehouses                | yuxiaoping |           | f         | f       | f          | yuxiaoping   | 2022-03-25 10:51:51.2
41585+08 | 2022-03-25 10:51:51.241585+08
 icebear           | inventories               | yuxiaoping |           | f         | f       | f          | yuxiaoping   | 2022-03-25 10:51:51.2
48903+08 | 2022-03-25 10:51:51.248903+08
 icebear           | employees                 | yuxiaoping |           | f         | f       | f          | yuxiaoping   | 2022-03-25 10:51:51.2
51+08    | 2022-03-25 10:51:51.251+08
 pg_catalog        | pg_job_proc               |            | pg_global | t         | f       | f          |              |
          |
 icebear           | product_categories        | yuxiaoping |           | f         | f       | f          | yuxiaoping   | 2022-03-25 10:51:51.2
57515+08 | 2022-03-25 10:51:51.257515+08
 icebear           | customers                 | yuxiaoping |           | f         | f       | f          | yuxiaoping   | 2022-03-25 10:51:51.2
66327+08 | 2022-03-25 10:51:51.266327+08
 icebear           | contacts                  | yuxiaoping |           | f         | f       | f          | yuxiaoping   | 2022-03-25 10:51:51.2
74556+08 | 2022-03-25 10:51:51.274556+08
 icebear           | orders                    | yuxiaoping |           | f         | f       | f          | yuxiaoping   | 2022-03-25 10:51:51.2
81454+08 | 2022-03-25 10:51:51.281454+08
 icebear           | order_items               | yuxiaoping |           | f         | f       | f          | yuxiaoping   | 2022-03-25 10:51:53.3
8687+08  | 2022-03-25 10:51:53.38687+08
 icebear           | products                  | yuxiaoping |           | f         | f       | f          | yuxiaoping   | 2022-03-25 10:59:14.3
```

（5）修改 12 张表的结构，根据上图添加相应的约束，如主码、外码。

1. 添加约束

① regions 表

```
sale=> ALTER TABLE regions ALTER region_id SET NOT NULL;
ALTER TABLE
```

```
sale=> ALTER TABLE regions ALTER region_name SET NOT NULL;
ALTER TABLE
```

```
sale=> ALTER TABLE regions ADD CONSTRAINT pk PRIMARY KEY(region_id);
NOTICE:  ALTER TABLE / ADD PRIMARY KEY will create implicit index "pk" for table "regions"
ALTER TABLE
```

```
sale=> \d+ regiona
                                Table "icebear.regiona"
   Column    |          Type          | Modifiers | Storage  | Stats target | Description
-------------+------------------------+-----------+----------+--------------+-------------
 region_id   | numeric                | not null  | main     |              |
 region_name | character varying(50)  | not null  | extended |              |
Indexes:
    "regiona_pkey" PRIMARY KEY, btree (region_id) TABLESPACE pg_default
Has OIDs: no
Options: orientation=row, compression=no
```

② countries 表

```
sale=> ALTER TABLE countries ALTER country_name SET NOT NULL;
ALTER TABLE
sale=> ALTER TABLE countries ADD CONSTRAINT pk PRIMARY KEY(country_id);
```

```
sale=> ALTER TABLE countries ADD CONSTRAINT ctpk PRIMARY KEY(country_id);
NOTICE:  ALTER TABLE / ADD PRIMARY KEY will create implicit index "ctpk" for table "countries"
ALTER TABLE
sale=> ALTER TABLE countries ADD CONSTRAINT fk_countries_regions FOREIGN KEY( region_id )
sale->  REFERENCES regions( region_id )
sale->  ON DELETE CASCADE;
ALTER TABLE
```

```
sale=> \d+ countries
                                Table "icebear.countries"
   Column    |          Type          | Modifiers | Storage  | Stats target | Description
-------------+------------------------+-----------+----------+--------------+-------------
 country_id   | character(2)           | not null  | extended |              |
 country_name | character varying(40)  | not null  | extended |              |
 region_id    | numeric                |           | main     |              |
Indexes:
    "ctpk" PRIMARY KEY, btree (country_id) TABLESPACE pg_default
Foreign-key constraints:
    "fk_countries_regions" FOREIGN KEY (region_id) REFERENCES regions(region_id) ON DELETE CASCADE
Has OIDs: no
Options: orientation=row, compression=no
```

③ location 表

```
sale=> ALTER TABLE locations ADD CONSTRAINT loctpk PRIMARY KEY(location_id);
NOTICE:  ALTER TABLE / ADD PRIMARY KEY will create implicit index "loctpk" for table "locations"
ALTER TABLE
sale=> ALTER TABLE locations ALTER address SET NOT NULL;
ALTER TABLE
sale=> ALTER TABLE locations ADD CONSTRAINT fk_locations_countries
sale->       FOREIGN KEY( country_id )
sale->       REFERENCES countries( country_id )
sale->       ON DELETE CASCADE;
ALTER TABLE
sale=> \d+ locations
                                Table "icebear.locations"
   Column    |          Type          | Modifiers | Storage  | Stats target | Description
-------------+------------------------+-----------+----------+--------------+-------------
 location_id | numeric                | not null  | main     |              |
 address     | character varying(255) | not null  | extended |              |
 postal_code | character varying(20)  |           | extended |              |
 city        | character varying(50)  |           | extended |              |
 state       | character varying(50)  |           | extended |              |
 country_id  | character(2)           |           | extended |              |
Indexes:
    "loctpk" PRIMARY KEY, btree (location_id) TABLESPACE pg_default
Foreign-key constraints:
    "fk_locations_countries" FOREIGN KEY (country_id) REFERENCES countries(country_id) ON DELETE CASCADE
Has OIDs: no
Options: orientation=row, compression=no
```

④ employees 表

```
sale=> ALTER TABLE employees ALTER email SET NOT NULL;
ALTER TABLE
sale=> ALTER TABLE employees ALTER phone SET NOT NULL;
ALTER TABLE
sale=> ALTER TABLE employees ALTER hire_date SET NOT NULL;
ALTER TABLE
sale=> ALTER TABLE employees ALTER job_title SET NOT NULL;
ALTER TABLE
sale=> ALTER TABLE employees ADD CONSTRAINT fk_employees_manager
sale->         FOREIGN KEY( manager_id )
sale->         REFERENCES employees( employee_id )
sale->         ON DELETE CASCADE;
ALTER TABLE
sale=> \d+ employees
                                    Table "icebear.employees"
   Column    |              Type              | Modifiers | Storage  | Stats target | Description
-------------+--------------------------------+-----------+----------+--------------+-------------
 employee_id | numeric                        | not null  | main     |              |
 first_name  | character varying(255)         | not null  | extended |              |
 last_name   | character varying(255)         | not null  | extended |              |
 email       | character varying(255)         | not null  | extended |              |
 phone       | character varying(50)          | not null  | extended |              |
 hire_date   | timestamp(0) without time zone | not null  | plain    |              |
 manager_id  | numeric(12,0)                  |           | main     |              |
 job_title   | character varying(255)         | not null  | extended |              |
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 CASCADE
Referenced by:
    TABLE "employees" CONSTRAINT "fk_employees_manager" FOREIGN KEY (manager_id) REFERENCES employees(employee_id) ON DELETE CASCADE
Has OIDs: no
Options: orientation=row, compression=no
```

⑤ warehouse 表

```
sale=> ALTER TABLE warehouses ADD CONSTRAINT warepk PRIMARY KEY(warehouse_id);
NOTICE:  ALTER TABLE / ADD PRIMARY KEY will create implicit index "warepk" for table "warehouses"
ALTER TABLE
sale=> ALTER TABLE warehouses ADD  CONSTRAINT fk_warehouses_locations
sale->       FOREIGN KEY( location_id )
sale->       REFERENCES locations( location_id )
sale->       ON DELETE CASCADE
sale-> ;
ALTER TABLE
sale=> \d+ warehouses
                                  Table "icebear.warehouses"
     Column     |          Type          | Modifiers | Storage  | Stats target | Description
----------------+------------------------+-----------+----------+--------------+-------------
 warehouse_id   | numeric                | not null  | main     |              |
 warehouse_name | character varying(255) |           | extended |              |
 location_id    | numeric(12,0)          |           | main     |              |
Indexes:
    "warepk" PRIMARY KEY, btree (warehouse_id) TABLESPACE pg_default
Foreign-key constraints:
    "fk_warehouses_locations" FOREIGN KEY (location_id) REFERENCES locations(location_id) ON DELETE CASCADE
Has OIDs: no
Options: orientation=row, compression=no
```

⑥ products 表

```
sale=> ALTER TABLE products ADD CONSTRAINT product_idpk PRIMARY KEY(product_id);
NOTICE:  ALTER TABLE / ADD PRIMARY KEY will create implicit index "product_idpk" for table "products"
ALTER TABLE
sale=> ALTER TABLE products ALTER product_name SET NOT NULL;
ALTER TABLE
sale=> ALTER TABLE products ALTER category_id SET NOT NULL;
ALTER TABLE
sale=> ALTER TABLE products ADD  CONSTRAINT fk_products_categories
sale->       FOREIGN KEY( category_id )
sale->       REFERENCES product_categories( category_id )
sale->       ON DELETE CASCADE;
ALTER TABLE
sale=> \d+ products
                                   Table "icebear.products"
    Column     |          Type          | Modifiers | Storage  | Stats target | Description
---------------+------------------------+-----------+----------+--------------+-------------
 product_id    | numeric                | not null  | main     |              |
 product_name  | character varying(255) | not null  | extended |              |
 description   | character varying(2000) |          | extended |              |
 standard_cost | numeric(9,2)           |           | main     |              |
 list_price    | numeric(9,2)           |           | main     |              |
 category_id   | numeric                | not null  | main     |              |
Indexes:
    "product_idpk" PRIMARY KEY, btree (product_id) TABLESPACE pg_default
Foreign-key constraints:
    "fk_products_categories" FOREIGN KEY (category_id) REFERENCES product_categories(category_id) ON DELETE CASCADE
Has OIDs: no
Options: orientation=row, compression=no
```

⑦ customers 表

```
sale=> ALTER TABLE customers ADD CONSTRAINT customers_idpk PRIMARY KEY(customer_id);
NOTICE:  ALTER TABLE / ADD PRIMARY KEY will create implicit index "customers_idpk" for table "customers"
ALTER TABLE
sale=> ALTER TABLE customers ALTER name SET NOT NULL;
ALTER TABLE
sale=> \d+ customers
                                Table "icebear.customers"
    Column    |          Type          | Modifiers | Storage  | Stats target | Description
--------------+------------------------+-----------+----------+--------------+-------------
 customer_id  | numeric                | not null  | main     |              |
 name         | character varying(255) | not null  | extended |              |
 address      | character varying(255) |           | extended |              |
 website      | character varying(255) |           | extended |              |
 credit_limit | numeric(8,2)           |           | main     |              |
Indexes:
    "customers_idpk" PRIMARY KEY, btree (customer_id) TABLESPACE pg_default
Has OIDs: no
Options: orientation=row, compression=no
```

⑧ contacts 表

```
sale=> ALTER TABLE contacts ADD CONSTRAINT contactspk PRIMARY KEY(contact_id);
NOTICE:  ALTER TABLE / ADD PRIMARY KEY will create implicit index "contactspk" for table "contacts"
ALTER TABLE
sale=> ALTER TABLE contacts ALTER first_name SET NOT NULL;
ALTER TABLE
sale=> ALTER TABLE contacts ALTER last_name SET NOT NULL;
ALTER TABLE
sale=> ALTER TABLE contacts ALTER email SET NOT NULL;
ALTER TABLE
sale=> ALTER TABLE contacts ADD CONSTRAINT fk_contacts_customers
sale->       FOREIGN KEY( customer_id )
sale->       REFERENCES customers( customer_id )
sale->       ON DELETE CASCADE
sale-> ;
ALTER TABLE
sale=> \d+ contacts
                                Table "icebear.contacts"
   Column    |          Type          | Modifiers | Storage  | Stats target | Description
-------------+------------------------+-----------+----------+--------------+-------------
 contact_id  | numeric                | not null  | main     |              |
 first_name  | character varying(255) | not null  | extended |              |
 last_name   | character varying(255) | not null  | extended |              |
 email       | character varying(255) | not null  | extended |              |
 phone       | character varying(20)  |           | extended |              |
 customer_id | numeric                |           | main     |              |
Indexes:
    "contactspk" PRIMARY KEY, btree (contact_id) TABLESPACE pg_default
Foreign-key constraints:
    "fk_contacts_customers" FOREIGN KEY (customer_id) REFERENCES customers(customer_id) ON DELETE CASCADE
Has OIDs: no
Options: orientation=row, compression=no
```

⑨ orders 表

```
sale=> ALTER TABLE orders ADD CONSTRAINT orderspk PRIMARY KEY(order_id);
NOTICE:  ALTER TABLE / ADD PRIMARY KEY will create implicit index "orderspk" for table "orders"
ALTER TABLE
sale=> ALTER TABLE orders ALTER customer_id SET NOT NULL;
ALTER TABLE
sale=> ALTER TABLE orders ALTER status SET NOT NULL;
ALTER TABLE
sale=> ALTER TABLE orders ALTER order_date SET NOT NULL;
ALTER TABLE
sale=> ALTER TABLE orders ADD CONSTRAINT fk_orders_customers
sale->       FOREIGN KEY( customer_id )
sale->       REFERENCES customers( customer_id )
sale->       ON DELETE CASCADE;
ALTER TABLE
sale=> ALTER TABLE orders ADD CONSTRAINT fk_orders_employees
sale->       FOREIGN KEY( salesman_id )
sale->       REFERENCES employees( employee_id )
sale->       ON DELETE SET NULL;
ALTER TABLE
sale=> \d+ orders
                                        Table "icebear.orders"
   Column    |              Type              | Modifiers | Storage  | Stats target | Description
-------------+--------------------------------+-----------+----------+--------------+-------------
 order_id    | numeric                        | not null  | main     |              |
 customer_id | numeric(6,0)                   | not null  | main     |              |
 status      | character varying(20)          | not null  | extended |              |
 salesman_id | numeric(6,0)                   |           | main     |              |
 order_date  | timestamp(0) without time zone | not null  | plain    |              |
Indexes:
    "orderspk" PRIMARY KEY, btree (order_id) TABLESPACE pg_default
Foreign-key constraints:
    "fk_orders_customers" FOREIGN KEY (customer_id) REFERENCES customers(customer_id) ON DELETE CASCADE
    "fk_orders_employees" FOREIGN KEY (salesman_id) REFERENCES employees(employee_id) ON DELETE SET NULL
Has OIDs: no
Options: orientation=row, compression=no
```

⑩ order_items 表

```
sale=> ALTER TABLE order_items ALTER product_id SET NOT NULL;
ALTER TABLE
sale=> ALTER TABLE order_items ALTER quantity SET NOT NULL;
ALTER TABLE
sale=> ALTER TABLE order_items ALTER unit_price SET NOT NULL;
ALTER TABLE
sale=> ALTER TABLE order_items ADD  CONSTRAINT pk_order_items
sale->       PRIMARY KEY( order_id, item_id );
NOTICE:  ALTER TABLE / ADD PRIMARY KEY will create implicit index "pk_order_items" for table "order_items"
ALTER TABLE
sale=> ALTER TABLE order_items ADD CONSTRAINT fk_order_items_products
sale->       FOREIGN KEY( product_id )
sale->       REFERENCES products( product_id )
sale->       ON DELETE CASCADE;
ALTER TABLE
sale=> ALTER TABLE order_items ADD CONSTRAINT fk_order_items_orders
sale->       FOREIGN KEY( order_id )
sale->       REFERENCES orders( order_id )
sale->       ON DELETE CASCADE;
ALTER TABLE
sale=> \d+ order_items
                          Table "icebear.order_items"
   Column   |     Type      | Modifiers | Storage | Stats target | Description
------------+---------------+-----------+---------+--------------+-------------
 order_id   | numeric(12,0) | not null  | main    |              |
 item_id    | numeric(12,0) | not null  | main    |              |
 product_id | numeric(12,0) | not null  | main    |              |
 quantity   | numeric(8,2)  | not null  | main    |              |
 unit_price | numeric(8,2)  | not null  | main    |              |
Indexes:
    "pk_order_items" PRIMARY KEY, btree (order_id, item_id) TABLESPACE pg_default
Foreign-key constraints:
    "fk_order_items_orders" FOREIGN KEY (order_id) REFERENCES orders(order_id) ON DELETE CASCADE
    "fk_order_items_products" FOREIGN KEY (product_id) REFERENCES products(product_id) ON DELETE CASCADE
Has OIDs: no
Options: orientation=row, compression=no
```

⑩① product_categories 表

```
sale=> ALTER TABLE product_categories ADD CONSTRAINT category_idpk PRIMARY KEY(category_id);
NOTICE:  ALTER TABLE / ADD PRIMARY KEY will create implicit index "category_idpk" for table "product_categories"
ALTER TABLE
sale=>  ALTER TABLE product_categories ALTER category_name SET NOT NULL;
ALTER TABLE
sale=> \d+ product_categories
                                Table "icebear.product_categories"
    Column     |          Type          | Modifiers | Storage  | Stats target | Description
---------------+------------------------+-----------+----------+--------------+-------------
 category_id   | numeric                | not null  | main     |              |
 category_name | character varying(255) | not null  | extended |              |
Indexes:
    "category_idpk" PRIMARY KEY, btree (category_id) TABLESPACE pg_default
Has OIDs: no
Options: orientation=row, compression=no
```

⑩② inventories 表

```
sale=> ALTER TABLE inventories ALTER quantity SET NOT NULL;
ALTER TABLE
sale=> ALTER TABLE inventories ADD  CONSTRAINT pk_inventories
sale->       PRIMARY KEY( product_id, warehouse_id );
NOTICE:  ALTER TABLE / ADD PRIMARY KEY will create implicit index "pk_inventories" for table "inventories"
ALTER TABLE
sale=> ALTER TABLE inventories ADD  CONSTRAINT fk_inventories_products
sale->       FOREIGN KEY( product_id )
sale->       REFERENCES products( product_id )
sale->       ON DELETE CASCADE;
ALTER TABLE
sale=> ALTER TABLE inventories ADD   CONSTRAINT fk_inventories_warehouses
sale->       FOREIGN KEY( warehouse_id )
sale->       REFERENCES warehouses( warehouse_id )
sale->       ON DELETE CASCADE;
ALTER TABLE
sale=> \d+ inventories
                              Table "icebear.inventories"
    Column     |      Type      | Modifiers | Storage | Stats target | Description
--------------+---------------+-----------+---------+--------------+-------------
 product_id   | numeric(12,0) | not null  | main    |              |
 warehouse_id | numeric(12,0) | not null  | main    |              |
 quantity     | numeric(8,0)  | not null  | main    |              |
Indexes:
    "pk_inventories" PRIMARY KEY, btree (product_id, warehouse_id) TABLESPACE pg_default
Foreign-key constraints:
    "fk_inventories_products" FOREIGN KEY (product_id) REFERENCES products(product_id) ON DELETE CASCADE
    "fk_inventories_warehouses" FOREIGN KEY (warehouse_id) REFERENCES warehouses(warehouse_id) ON DELETE CASCADE
Has OIDs: no
Options: orientation=row, compression=no
```

（6）为 12 张表插入示例数据。

方法一：使用 gsql 元命令导入数据

方法二：通过 excel 的公式对 csv 文件转化为 sql 的 insert 语句，直接使用 insert 语句。代码见 3-插入数据.sql。由于用 gsql 导入数据时，分隔符不好确定，有的表有数据元素就有逗号。如果采用，作为分隔符会出错，所以最后采用了方法二。

```sql
REM INSERTING into  REGIONS
SET DEFINE OFF;
Insert into  REGIONS (REGION_ID,REGION_NAME) values (1,'Europe');
Insert into  REGIONS (REGION_ID,REGION_NAME) values (2,'Americas');
Insert into  REGIONS (REGION_ID,REGION_NAME) values (3,'Asia');
Insert into  REGIONS (REGION_ID,REGION_NAME) values (4,'Middle East and Africa');

REM INSERTING into  COUNTRIES
SET DEFINE OFF;
Insert into  COUNTRIES (COUNTRY_ID,COUNTRY_NAME,REGION_ID) values ('AR','Argentina',2);
Insert into  COUNTRIES (COUNTRY_ID,COUNTRY_NAME,REGION_ID) values ('AU','Australia',3);
Insert into  COUNTRIES (COUNTRY_ID,COUNTRY_NAME,REGION_ID) values ('BE','Belgium',1);
Insert into  COUNTRIES (COUNTRY_ID,COUNTRY_NAME,REGION_ID) values ('BR','Brazil',2);
Insert into  COUNTRIES (COUNTRY_ID,COUNTRY_NAME,REGION_ID) values ('CA','Canada',2);
Insert into  COUNTRIES (COUNTRY_ID,COUNTRY_NAME,REGION_ID) values ('CH','Switzerland',1);
Insert into  COUNTRIES (COUNTRY_ID,COUNTRY_NAME,REGION_ID) values ('CN','China',3);
Insert into  COUNTRIES (COUNTRY_ID,COUNTRY_NAME,REGION_ID) values ('DE','Germany',1);
Insert into  COUNTRIES (COUNTRY_ID,COUNTRY_NAME,REGION_ID) values ('DK','Denmark',1);
Insert into  COUNTRIES (COUNTRY_ID,COUNTRY_NAME,REGION_ID) values ('EG','Egypt',4);
Insert into  COUNTRIES (COUNTRY_ID,COUNTRY_NAME,REGION_ID) values ('FR','France',1);
Insert into  COUNTRIES (COUNTRY_ID,COUNTRY_NAME,REGION_ID) values ('IL','Israel',4);
Insert into  COUNTRIES (COUNTRY_ID,COUNTRY_NAME,REGION_ID) values ('IN','India',3);
Insert into  COUNTRIES (COUNTRY_ID,COUNTRY_NAME,REGION_ID) values ('IT','Italy',1);
Insert into  COUNTRIES (COUNTRY_ID,COUNTRY_NAME,REGION_ID) values ('JP','Japan',3);
Insert into  COUNTRIES (COUNTRY_ID,COUNTRY_NAME,REGION_ID) values ('KW','Kuwait',4);
Insert into  COUNTRIES (COUNTRY_ID,COUNTRY_NAME,REGION_ID) values ('ML','Malaysia',3);
Insert into  COUNTRIES (COUNTRY_ID,COUNTRY_NAME,REGION_ID) values ('MX','Mexico',2);
Insert into  COUNTRIES (COUNTRY_ID,COUNTRY_NAME,REGION_ID) values ('NG','Nigeria',4);
Insert into  COUNTRIES (COUNTRY_ID,COUNTRY_NAME,REGION_ID) values ('NL','Netherlands',1);
Insert into  COUNTRIES (COUNTRY_ID,COUNTRY_NAME,REGION_ID) values ('SG','Singapore',3);
Insert into  COUNTRIES (COUNTRY_ID,COUNTRY_NAME,REGION_ID) values ('UK','United Kingdom',1);
```

（7）分别查询 12 张表的数据。

步骤如下：

使用 SELECT * FROM tablename;查询 12 张表的数据，其中 tablename 指 12 张表的表名。删除用户及其包含的数据库、模式和所有的表。

① regions 表

```
sale=> SELECT * FROM regions;
 region_id |       region_name
-----------+-------------------------
         1 | Europe
         2 | Americas
         3 | Asia
         4 | Middle East and Africa
(4 rows)
```

② countries 表

```
sale=> SELECT * FROM countries;
 country_id |        country_name        | region_id
------------+----------------------------+-----------
 AR         | Argentina                  |         2
 AU         | Australia                  |         3
 BE         | Belgium                    |         1
 BR         | Brazil                     |         2
 CA         | Canada                     |         2
 CH         | Switzerland                |         1
 CN         | China                      |         3
 DE         | Germany                    |         1
 DK         | Denmark                    |         1
 EG         | Egypt                      |         4
 FR         | France                     |         1
 IL         | Israel                     |         4
 IN         | India                      |         3
 IT         | Italy                      |         1
 JP         | Japan                      |         3
 KW         | Kuwait                     |         4
 ML         | Malaysia                   |         3
 MX         | Mexico                     |         2
 NG         | Nigeria                    |         4
 NL         | Netherlands                |         1
 SG         | Singapore                  |         3
 UK         | United Kingdom             |         1
 US         | United States of America   |         2
 ZM         | Zambia                     |         4
 ZW         | Zimbabwe                   |         4
(25 rows)
```

③ locations 表

```
sale=> SELECT * FROM locations;
 location_id |                address                | postal_code |        city         |       state       | country_id
-------------+---------------------------------------+-------------+---------------------+-------------------+------------
           1 | 1297 Via Cola di Rie                  | 00989       | Roma                |                   | IT
           2 | 93091 Calle della Testa               | 10934       | Venice              |                   | IT
           3 | 2017 Shinjuku-ku                      | 1689        | Tokyo               | Tokyo Prefecture  | JP
           4 | 9450 Kamiya-cho                       | 6823        | Hiroshima           |                   | JP
           5 | 2014 Jabberwocky Rd                   | 26192       | Southlake           | Texas             | US
           6 | 2011 Interiors Blvd                   | 99236       | South San Francisco | California        | US
           7 | 2007 Zagora St                        | 50090       | South Brunswick     | New Jersey        | US
           8 | 2004 Charade Rd                       | 98199       | Seattle             | Washington        | US
           9 | 147 Spadina Ave                       | M5V 2L7     | Toronto             | Ontario           | CA
          10 | 6092 Boxwood St                       | YSW 9T2     | Whitehorse          | Yukon             | CA
          11 | 40-5-12 Laogianggen                   | 190518      | Beijing             |                   | CN
          12 | 1298 Vileparle (E)                    | 490231      | Bombay              | Maharashtra       | IN
          13 | 12-98 Victoria Street                 | 2901        | Sydney              | New South Wales   | AU
          14 | 198 Clementi North                    | 540198      | Singapore           |                   | SG
          15 | 8204 Arthur St                        |             | London              |                   | UK
          16 | Magdalen Centre, The Oxford Science Park | OX9 9ZB     | Oxford              | Oxford            | UK
          17 | 9702 Chester Road                     | 09629850293 | Stretford           | Manchester        | UK
          18 | Schwanthalerstr. 7031                 | 80925       | Munich              | Bavaria           | DE
          19 | Rua Frei Caneca 1360                  | 01307-002   | Sao Paulo           | Sao Paulo         | BR
          20 | 20 Rue des Corps-Saints               | 1730        | Geneva              | Geneve            | CH
          21 | Murtenstrasse 921                     | 3095        | Bern                | BE                | CH
          22 | Pieter Breughelstraat 837             | 3029SK      | Utrecht             | Utrecht           | NL
          23 | Mariano Escobedo 9991                 | 11932       | Mexico City         | Distrito Federal, | MX
(23 rows)
```

④ employees 表

```
>_ root@114.116.235.207 ×
 employee_id | first_name | last_name  |          email           |     phone     |     hire_date     | manager_id |          job_title
-------------+------------+------------+--------------------------+---------------+-------------------+------------+-----------------------------
--
         107 | Summer     | Payne      | summerpayne@examplecom   | 5151238181    | 2016-06-07 00:00:00 |        106 | Public Accountant
           1 | Tommy      | Bailey     | tommybailey@examplecom   | 5151234567    | 2016-06-17 00:00:00 |            | President
           3 | Blake      | Cooper     | blakecooper@examplecom   | 5151234569    | 2016-09-13 00:00:00 |          1 | Administration Vice President
           2 | Jude       | Rivera     | juderivera@examplecom    | 5151234568    | 2016-09-21 00:00:00 |          1 | Administration Vice President
         101 | Annabelle  | Dunn       | annabelledunn@examplecom | 5151234444    | 2016-09-17 00:00:00 |          2 | Administration Assistant
           9 | Mohammad   | Peterson   | mohammadpeterson@examplecom | 5151244569 | 2016-09-17 00:00:00 |          2 | Finance Manager
         104 | Harper     | Spencer    | harperspencer@examplecom | 5151237777    | 2016-06-07 00:00:00 |          2 | Human Resources Representative
           4 | Louie      | Richardson | louierichardson@examplecom | 5904234567 | 2016-09-03 00:00:00 |          3 | Programmer
           5 | Nathan     | Cox        | nathancox@examplecom     | 5904234568    | 2016-05-21 00:00:00 |          4 | Programmer
           8 | Bobby      | Torres     | bobbytorres@examplecom   | 5904235567    | 2016-02-07 00:00:00 |          4 | Programmer
           7 | Charles    | Ward       | charlesward@examplecom   | 5904234560    | 2016-02-05 00:00:00 |          4 | Programmer
           6 | Gabriel    | Howard     | gabrielhoward@examplecom | 5904234569    | 2016-06-25 00:00:00 |          4 | Programmer
         102 | Emma       | Perkins    | emmaperkins@examplecom   | 5151235555    | 2016-02-17 00:00:00 |          1 | Marketing Manager
         103 | Amelie     | Hudson     | ameliehudson@examplecom  | 6031236666    | 2016-09-17 00:00:00 |        102 | Marketing Representative
         105 | Gracie     | Gardner    | graciegardner@examplecom | 5151238888    | 2016-06-07 00:00:00 |          2 | Public Relations Representativ
e
          11 | Tyler      | Ramirez    | tylerramirez@examplecom  | 5151244269    | 2016-09-28 00:00:00 |          9 | Accountant
          10 | Ryan       | Gray       | ryangray@examplecom      | 5151244169    | 2016-09-16 00:00:00 |          9 | Accountant
          14 | Elliot     | Brooks     | elliotbrooks@examplecom  | 5151244567    | 2016-12-07 00:00:00 |          9 | Accountant
          12 | Elliott    | James      | elliottjames@examplecom  | 5151244369    | 2016-09-30 00:00:00 |          9 | Accountant
          15 | Rory       | Kelly      | rorykelly@examplecom     | 5151274561    | 2016-12-07 00:00:00 |          1 | Purchasing Manager
          49 | Isabella   | Cole       | isabellacole@examplecom  | 01144134461926 8 | 2016-10-15 00:00:00 |          1 | Sales Manager
          48 | Jessica    | Woods      | jessicawoods@examplecom  | 01144134442927 8 | 2016-03-10 00:00:00 |          1 | Sales Manager
          47 | Ella       | Wallace    | ellawallace@examplecom   | 01144134446726 8 | 2016-09-05 00:00:00 |          1 | Sales Manager
          46 | Ava        | Sullivan   | avasullivan@examplecom   | 01144134442926 8 | 2016-10-01 00:00:00 |          1 | Sales Manager
          50 | Mia        | West       | miawest@examplecom       | 01144134442901 8 | 2016-09-29 00:00:00 |          1 | Sales Manager
          56 | Evie       | Harrison   | evieharrison@examplecom  | 01144134448650 8 | 2016-11-23 00:00:00 |         46 | Sales Representative
          57 | Scarlett   | Gibson     | scarlettgibson@examplecom | 01144134542926 8 | 2016-09-30 00:00:00 |         47 | Sales Representative
          58 | Ruby       | Mcdonald   | rubymcdonald@examplecom  | 01144134592926 8 | 2016-03-04 00:00:00 |         47 | Sales Representative
          59 | Chloe      | Cruz       | chloecruz@examplecom     | 01144134582926 8 | 2016-09-01 00:00:00 |         47 | Sales Representative
          60 | Isabelle   | Marshall   | isabellemarshall@examplecom | 01144134572926 8 | 2016-03-10 00:00:00 |         47 | Sales Representative
          61 | Daisy      | Ortiz      | daisyortiz@examplecom    | 01144134562926 8 | 2016-12-15 00:00:00 |         47 | Sales Representative
          62 | Freya      | Gomez      | freyagomez@examplecom    | 01144134552926 8 | 2016-11-03 00:00:00 |         47 | Sales Representati
--More--
```

⑤ warehouses 表

```
sale=> SELECT * FROM warehouses;
 warehouse_id |    warehouse_name     | location_id
--------------+-----------------------+-------------
            1 | Southlake, Texas      |           5
            2 | San Francisco         |           6
            3 | New Jersey            |           7
            4 | Seattle, Washington   |           8
            5 | Toronto               |           9
            6 | Sydney                |          13
            7 | Mexico City           |          23
            8 | Beijing               |          11
            9 | Bombay                |          12
(9 rows)
```

⑥ products 表

```
 product_id |                product_name                |                          description                          | standard_cost | list_p
rice | category_id
-----------+--------------------------------------------+-------------------------------------------------------------+---------------+-------
-----+------------
        228 | Intel Xeon E5-2699 V3 (OEM/Tray)           | Speed:2.3GHz,Cores:18,TDP:145W                              |       2867.51 |    341
0.46 |           1
        248 | Intel Xeon E5-2697 V3                      | Speed:2.6GHz,Cores:14,TDP:145W                              |       2326.27 |    277
4.98 |           1
        249 | Intel Xeon E5-2698 V3 (OEM/Tray)           | Speed:2.3GHz,Cores:16,TDP:135W                              |       2035.18 |    266
0.72 |           1
          2 | Intel Xeon E5-2697 V4                      | Speed:2.3GHz,Cores:18,TDP:145W                              |       2144.40 |    255
4.99 |           1
         45 | Intel Xeon E5-2685 V3 (OEM/Tray)           | Speed:2.6GHz,Cores:12,TDP:120W                              |       2012.11 |    250
1.69 |           1
         46 | Intel Xeon E5-2695 V3 (OEM/Tray)           | Speed:2.3GHz,Cores:14,TDP:120W                              |       1925.13 |    243
1.95 |           1
         47 | Intel Xeon E5-2697 V2                      | Speed:2.7GHz,Cores:12,TDP:130W                              |       2101.59 |    237
7.09 |           1
         51 | Intel Xeon E5-2695 V4                      | Speed:2.1GHz,Cores:18,TDP:120W                              |       1780.35 |    226
9.99 |           1
         91 | Intel Xeon E5-2695 V2                      | Speed:2.4GHz,Cores:12,TDP:115W                              |       1793.53 |    225
9.99 |           1
         92 | Intel Xeon E5-2643 V2 (OEM/Tray)           | Speed:3.5GHz,Cores:6,TDP:130W                               |       1940.18 |    220
0.00 |           1
         93 | Intel Xeon E5-2690 (OEM/Tray)              | Speed:2.9GHz,Cores:8,TDP:135W                               |       1888.33 |    211
6.72 |           1
         98 | Intel Xeon E5-2687W V3                     | Speed:3.1GHz,Cores:10,TDP:160W                              |       1781.47 |    206
4.99 |           1
        102 | Intel Xeon E5-2687W V4                     | Speed:3.0GHz,Cores:12,TDP:160W                              |       1723.83 |    204
2.69 |           1
        158 | Intel Xeon E5-2667 V3 (OEM/Tray)           | Speed:3.2GHz,Cores:8,TDP:135W                               |       1504.08 |    200
9.46 |           1
        159 | Intel Xeon E5-2690 V4                      | Speed:2.6GHz,Cores:14,TDP:135W                              |       1499.26 |    199
4.49 |           1
        160 | Intel Xeon E5-2690 V3                      | Speed:2.6GHz,Cores:12,TDP:135W                              |       1540.35 |    190
8.73 |           1
        162 | Intel Xeon E5-2470V2                       | Speed:2.4GHz,Cores:10,TDP:95W                               |       1671.95 |    190
--More--
```

⑦ customers 表

```
>_ root@114.116.235.207 ×
 customer_id |                name                |                address                |                website                | credit_lim
it
------------+------------------------------------+--------------------------------------------+--------------------------------------+-----------
---
         177 | United Continental Holdings        | 2904 S Salina St, Syracuse, NY             | http://www.unitedcontinentalholdings.com |      5000.
00
         180 | INTL FCStone                       | 5344 Haverford Ave, Philadelphia, PA       | http://www.intlfcstone.com               |      5000.
00
         184 | Publix Super Markets               | 1795 Wu Meng, Muang Chonburi,              | http://www.publix.com                    |      1200.
00
         187 | ConocoPhillips                     | Walpurgisstr 69, Munich,                   | http://www.conocophillips.com            |      2400.
00
         190 | 3M                                 | Via Frenzy 6903, Roma,                     | http://www.3m.com                        |      1200.
00
         192 | Exelon                             | Via Luminosa 162, Firenze,                 | http://www.exeloncorp.com                |       500.
00
         208 | Tesoro                             | Via Notoriosa 1942, Firenze,               | http://www.tsocorp.com                   |       500.
00
         207 | Northwestern Mutual                | 1831 No Wong, Peking,                      | http://www.northwesternmutual.com        |      3600.
00
         200 | Enterprise Products Partners       | Via Notoriosa 1949, Firenze,               | http://www.enterpriseproducts.com        |      2400.
00
         204 | Rite Aid                           | Piazza Cacchiatore 23, San Giminiano,      | http://www.riteaid.com                   |      3600.
00
         212 | Qualcomm                           | Piazza Svizzera, Milano,                   | http://www.qualcomm.com                  |       500.
00
         216 | EMC                                | Via Delle Grazie 11, San Giminiano,        | http://www.emc.com                       |       700.
00
         220 | Time Warner Cable                  | 1597 Legend St, Mysore, Kar                | http://www.twc.com                       |      3700.
00
         223 | Northrop Grumman                   | 1606 Sangam Blvd, New Delhi,               | http://www.northropgrumman.com           |      5000.
00
          39 | Lear                               | 2115 N Towne Ln Ne, Cedar Rapids, IA       | http://www.lear.com                      |       500.
00
          43 | Facebook                           | 5112 Sw 9Th St, Des Moines, IA             | http://www.facebook.com                  |       500.
00
          46 | Supervalu                          | 8989 N Port Washington Rd, Milwaukee, WI   | http://www.supervalu.com                 |       500.
--More--
```

⑧ contacts 表

```
 contact_id | first_name | last_name  |                email                |       phone        | customer_id
------------+------------+------------+-------------------------------------+--------------------+-------------
          1 | Flor       | Stone      | flor.stone@raytheon.com             | +1 317 123 4104    |           1
          2 | Lavera     | Emerson    | lavera.emerson@plainsallamerican.com | +1 317 123 4111    |           2
          3 | Fern       | Head       | fern.head@usfoods.com               | +1 812 123 4115    |           3
          4 | Shyla      | Ortiz      | shyla.ortiz@abbvie.com              | +1 317 123 4126    |           4
          5 | Jeni       | Levy       | jeni.levy@centene.com               | +1 812 123 4129    |           5
          6 | Matthias   | Hannah     | matthias.hannah@chs.net             | +1 219 123 4136    |           6
          7 | Matthias   | Cruise     | matthias.cruise@alcoa.com           | +1 219 123 4138    |           7
          8 | Meenakshi  | Mason      | meenakshi.mason@internationalpaper.com | +1 317 123 4141    |           8
          9 | Christian  | Cage       | christian.cage@emerson.com          | +1 219 123 4142    |           9
         10 | Charlie    | Sutherland | charlie.sutherland@up.com           | +1 317 123 4146    |          10
         11 | Charlie    | Pacino     | charlie.pacino@amgen.com            | +1 812 123 4150    |          11
         12 | Guillaume  | Jackson    | guillaume.jackson@usbank.com        | +1 812 123 4151    |          12
         13 | Daniel     | Costner    | daniel.costner@staples.com          | +1 812 123 4153    |          13
         14 | Dianne     | Derek      | dianne.derek@danaher.com            | +1 812 123 4157    |          14
         15 | Geraldine  | Schneider  | geraldine.schneider@whirlpoolcorp.com | +1 313 123 4159    |          15
         16 | Geraldine  | Martin     | geraldine.martin@aflac.com          | +1 313 123 4160    |          16
         17 | Guillaume  | Edwards    | guillaume.edwards@autonation.com    | +1 616 123 4162    |          17
         18 | Maurice    | Mahoney    | maurice.mahoney@progressive.com     | +1 616 123 4181    |          18
         19 | Maurice    | Hasan      | maurice.hasan@abbott.com            | +1 517 123 4191    |          19
         20 | Diane      | Higgins    | diane.higgins@dollargeneral.com     | +1 517 123 4199    |          20
         21 | Dianne     | Sen        | dianne.sen@tenethealth.com          | +1 517 123 4201    |          21
         22 | Maurice    | Daltrey    | maurice.daltrey@lilly.com           | +1 517 123 4206    |          22
         23 | Tess       | Roth       | tess.roth@dteenergy.com             | +1 313 123 4219    |          23
         24 | Ka         | Kaufman    | ka.kaufman@southwest.com            | +1 313 123 4222    |          24
         25 | Sharyl     | Montoya    | sharyl.montoya@penskeautomotive.com | +1 517 123 4225    |          25
         26 | Daniel     | Glass      | daniel.glass@manpowergroup.com      | +1 313 123 4226    |          26
         27 | Rena       | Arnold     | rena.arnold@assurant.com            | +1 517 123 4228    |          27
         28 | Arlyne     | Ingram     | arlyne.ingram@kohlscorporation.com  | +1 313 123 4230    |          28
         29 | Willie     | Barrera    | willie.barrera@starbucks.com        | +1 616 123 4234    |          29
         30 | Mireya     | Cochran    | mireya.cochran@paccar.com           | +1 313 123 4242    |          30
         31 | Marlene    | Odom       | marlene.odom@cummins.com            | +1 616 123 4245    |          31
         32 | Jaclyn     | Atkinson   | jaclyn.atkinson@globalp.com         | +1 313 123 4248    |          32
         33 | Al         | Schultz    | al.schultz@altria.com               | +1 517 123 4253    |          33
         34 | Felicitas  | Riley      | felicitas.riley@xerox.com           | +1 313 123 4255    |          34
         35 | Cora       | Calhoun    | cora.calhoun@kimberly-clark.com     | +1 313 123 4263    |          35
--More--
```

⑨ orders 表

order_id	customer_id	status	salesman_id	order_date
105	1	Pending	54	2016-11-17 00:00:00
44	2	Pending	55	2017-02-20 00:00:00
101	3	Pending	55	2017-01-03 00:00:00
1	4	Pending	56	2017-10-15 00:00:00
5	5	Canceled	56	2017-04-09 00:00:00
28	6	Canceled	57	2017-08-15 00:00:00
87	7	Canceled	57	2016-12-01 00:00:00
4	8	Shipped	59	2015-04-09 00:00:00
41	9	Shipped	59	2017-05-11 00:00:00
82	44	Shipped	60	2016-12-03 00:00:00
102	45	Shipped	61	2016-12-20 00:00:00
26	46	Shipped	62	2016-08-16 00:00:00
43	47	Shipped	62	2015-05-02 00:00:00
53	48	Shipped	62	2016-09-29 00:00:00
81	49	Shipped	62	2016-12-13 00:00:00
83	16	Shipped	62	2016-12-02 00:00:00
93	17	Shipped	62	2016-10-27 00:00:00
94	1	Shipped	62	2017-10-27 00:00:00
79	2	Shipped	64	2016-12-14 00:00:00
80	3	Shipped	64	2016-12-13 00:00:00
2	4	Shipped		2015-04-26 00:00:00
3	5	Shipped		2017-04-26 00:00:00
6	6	Shipped		2015-04-09 00:00:00
7	7	Shipped		2017-02-15 00:00:00
8	8	Shipped		2017-02-14 00:00:00
9	9	Shipped		2017-02-14 00:00:00
10	44	Pending		2017-01-24 00:00:00
11	45	Shipped		2016-11-29 00:00:00
12	46	Shipped		2016-11-29 00:00:00
13	47	Shipped		2016-11-29 00:00:00
14	48	Shipped		2017-09-28 00:00:00
15	49	Shipped		2017-09-27 00:00:00
16	16	Pending		2016-09-27 00:00:00
17	17	Shipped		2017-09-27 00:00:00
18	18	Shipped		2016-08-16 00:00:00

--More--

⑩ orders_items 表

```
 order_id | item_id | product_id | quantity | unit_price
----------+---------+------------+----------+------------
       70 |       7 |         32 |   132.00 |     469.99
       73 |       5 |        192 |   124.00 |     519.99
       74 |       7 |         27 |    92.00 |     800.74
       75 |      11 |          6 |   128.00 |     849.99
       76 |      10 |         95 |   106.00 |     109.99
       77 |       5 |        271 |   148.00 |     549.59
       81 |       7 |         79 |   127.00 |     659.99
       82 |       9 |        284 |   138.00 |      54.99
       83 |       8 |        174 |   117.00 |     798.26
       84 |       6 |        131 |    34.00 |     279.99
       87 |      11 |        271 |    58.00 |     549.59
       90 |       8 |         92 |    49.00 |    2200.00
       91 |      11 |        226 |    77.00 |     309.85
       93 |       5 |        121 |   141.00 |     721.99
       94 |       9 |         12 |    33.00 |     824.98
       99 |       9 |         17 |   144.00 |     699.01
      102 |       3 |        247 |   149.00 |     339.99
      104 |       7 |        178 |   145.00 |    1999.89
      105 |       6 |        183 |    79.00 |     899.99
        2 |       9 |        200 |    75.00 |     620.95
        4 |       8 |         18 |   116.00 |     799.00
        6 |       7 |          7 |   119.00 |     680.99
        8 |       9 |        268 |   148.00 |      47.88
        2 |       1 |         43 |   111.00 |     298.98
        3 |       1 |         24 |   111.00 |      66.89
        4 |       1 |        280 |    47.00 |     149.88
        5 |       1 |        149 |    70.00 |     282.98
        6 |       1 |        199 |    67.00 |     647.99
        7 |       1 |        227 |    74.00 |     305.00
        8 |       1 |         87 |    92.00 |     759.99
        9 |       1 |        108 |   139.00 |     849.99
       10 |       1 |        145 |   118.00 |     287.00
       11 |       1 |         96 |   113.00 |     141.56
       12 |       1 |        191 |    41.00 |     573.99
       13 |       1 |         84 |    46.00 |     440.30
--More--
```

⑩① product_categories 表

```
sale=> SELECT * FROM product_categories;
 category_id | category_name
-------------+---------------
           1 | CPU
           2 | Video Card
           3 | RAM
           4 | Mother Board
           5 | Storage
(5 rows)
```

⑩② inventories 表

```
product_id | warehouse_id | quantity
-----------+--------------+----------
       210 |            8 |      122
       211 |            8 |      123
       212 |            8 |      123
       214 |            8 |      124
       216 |            8 |      125
       217 |            8 |      125
       218 |            8 |      126
       220 |            8 |      149
       221 |            8 |      150
       222 |            8 |      150
       223 |            8 |      151
       229 |            8 |      123
       230 |            8 |      124
       231 |            8 |      124
       232 |            8 |      124
       233 |            8 |      124
       234 |            8 |      124
       235 |            8 |      125
       241 |            8 |      121
       242 |            8 |      121
       243 |            8 |      121
       245 |            8 |      137
       254 |            8 |      136
       255 |            8 |      136
       256 |            8 |      136
       257 |            8 |      136
       258 |            8 |      136
       259 |            8 |      137
       260 |            8 |      137
       263 |            8 |      137
       264 |            8 |      137
       269 |            8 |      178
       270 |            8 |      178
       271 |            8 |      183
       272 |            8 |      184
--More--
```

（7）完成上述任务后删除用户、数据库、模式和所有的表。

```
[omm@ecs-ad18 ~]$ gsql -d postgres -p 26000 -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.

postgres=# DROP DATABASE sale;
DROP DATABASE
```

```
postgres=# DROP USER yuxiaoping CASCADE;
DROP ROLE
```

```
postgres=# SELECT * FROM pg_user;
 usename | usesysid | usecreatedb | usesuper | usecatupd | userepl |  passwd  | valbegin | valuntil |
egroup | tempspacelimit | spillspacelimit | usemonitoradmin | useoperatoradmin | usepolicyadmin
---------+----------+-------------+----------+-----------+---------+----------+----------+----------+-
-------+----------------+-----------------+-----------------+------------------+----------------
 omm     |       10 | t           | t        | t         | t       | ******** |          |          |
       |                |                 | t               | t                | t
(1 row)
```

```
postgres=# \l
                              List of databases
   Name    | Owner | Encoding | Collate | Ctype | Access privileges
-----------+-------+----------+---------+-------+-------------------
 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
(3 rows)

postgres=#
```

3. 实验总结

3.1 完成的工作

在这次实验中，我理解了 Sales 数据库中各表及各表之间的联系，创建新用户、数据库和模式，分别创建了 12 张表，创建时不添加约束。修改 12 张表的结构，根据上图添加相应的约束，如主码、外码。为 12 张表插入示例数据。分别查询 12 张表的数据。完成上述任务后删除用户、数据库、模式和所有的表。

3.2 对实验的认识

通过实验我理解了 openGauss 中用户、数据库、模式和表等数据库对象之间的关系。掌握用户、数据库、模式和基本表的创建、修改和删除方法，掌握数据类型的选择和使用，掌握数据的导入导出方法。

对 openGauss 中的一些语句更熟悉了。如 SET SEARCH_PATH TO icebear, public;可以将搜索路径设置为 icebear、public，首先搜索 icebear。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 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

3.3 遇到的困难及解决方法

创建的数据库名能否与用户名相同？请上机验证。

```
postgres=# CREATE DATABASE yuxiaoping OWNER yuxiaoping;
CREATE DATABASE
```

创建的数据库名可以与用户名相同。