数据库系统课程实验报告

批注 [m1]: 文件名：学号-姓名-实验#，如 00000-张三-实验 2

实验名称：	实验二 – 数据定义
实验日期：	2023/3/24
实验地点：	文宣楼 A402
提交日期：	2023.3.27

学号：	22920212204392
姓名：	黄勖
专业年级：	软工 2021 级
学年学期：	2022-2023 学年第二学期

1.实验目的

- 理解用户、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、基本表和索引的创建、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；

现有一个销售数据库 Sales，该数据库包含 12 张表。各表的表名和信息描述见下表。

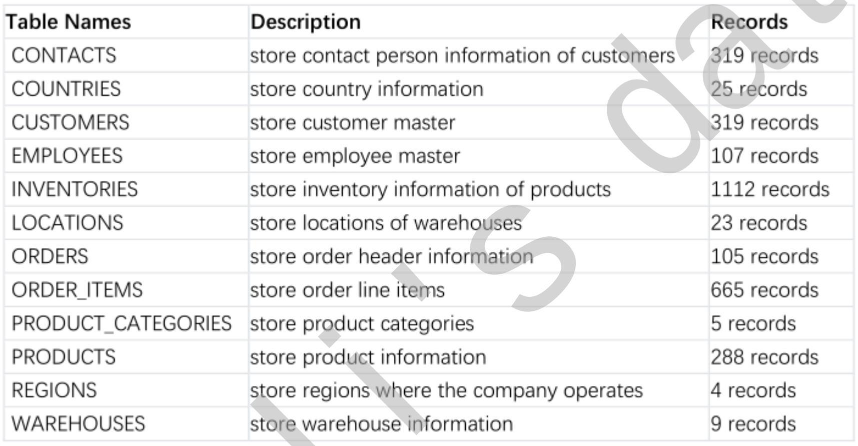


Table Names	Description	Records
CONTACTS	store contact person information of customers	319 records
COUNTRIES	store country information	25 records
CUSTOMERS	store customer master	319 records
EMPLOYEES	store employee master	107 records
INVENTORIES	store inventory information of products	1112 records
LOCATIONS	store locations of warehouses	23 records
ORDERS	store order header information	105 records
ORDER_ITEMS	store order line items	665 records
PRODUCT_CATEGORIES	store product categories	5 records
PRODUCTS	store product information	288 records
REGIONS	store regions where the company operates	4 records
WAREHOUSES	store warehouse information	9 records

关系如图所示（见实验要求）

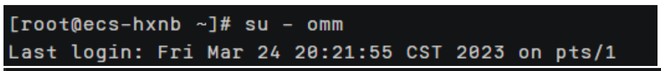
带星号“*”的字段为非空；employees 表字段 manager_id 值引用的是 employees 表字段 employee_id 的值。

（2）创建一个以自己中文姓名拼音(可缩写)的新用户；

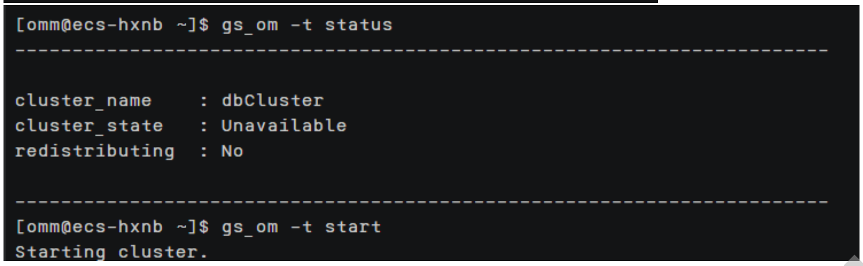
步骤如下：

1. 以 root 用户登录到 ECS 服务器>以 omm 操作系统管理员身份

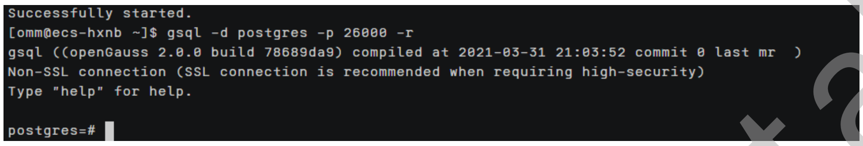
登录数据库>使用 gsql 连接到数据库。（每次断线都需要这一步）



```
[root@ecs-hxnb ~]# su - omm
Last login: Fri Mar 24 20:21:55 CST 2023 on pts/1
```

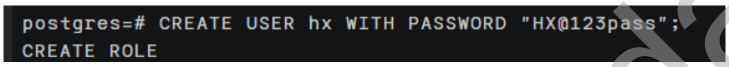


```
[omm@ecs-hxnb ~]$ gs_om -t status
-----------------------------------------------------------------------

cluster_name     : dbCluster
cluster_state    : Unavailable
redistributing   : No

-----------------------------------------------------------------------
[omm@ecs-hxnb ~]$ gs_om -t start
Starting cluster.
```



```
Successfully started.
[omm@ecs-hxnb ~]$ 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=#
```

2. 创建数据库用户：

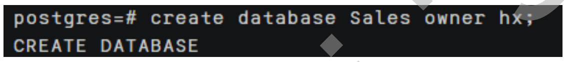


```
postgres=# CREATE USER hx WITH PASSWORD "HX@123pass";
CREATE ROLE
```

（3）创建一个名为 Sales 的数据库，要求该数据库属于新用户；

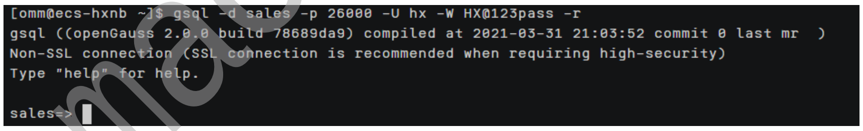
步骤：

1.先创建



```
postgres=# create database Sales owner hx;
CREATE DATABASE
```

2. 使用新用户连接到此数据库执行接下来的创建表等操作(此时在 postgres 数据库下)。退出 postgres 数据库，使用新用户连接到此数据库。（每次断线都需要这一步开始重新连接用户）

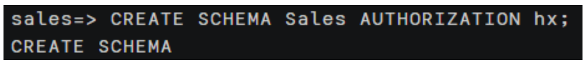


```
[omm@ecs-hxnb ~]$ gsql -d sales -p 26000 -U hx -W HX@123pass -r
gsql ((openGauss 2.0.0 build 78689da9) compiled at 2021-03-31 21:03:52 commit 0 last mr  )
Non-SSL connection (SSL connection is recommended when requiring high-security)
Type "help" for help.

sales=>
```

（4）创建一个名为 Sales 的模式，要求该模式属于 Sales 数据库；

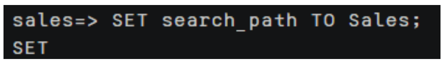
步骤：

1. 创建名为 Sales 的 SCHEMA



```
sales=> CREATE SCHEMA Sales AUTHORIZATION hx;
CREATE SCHEMA
```

2. 设置 Sales 为当前的 schema.将默认搜索路径设为 Sales（每次断线都需要从这一步）



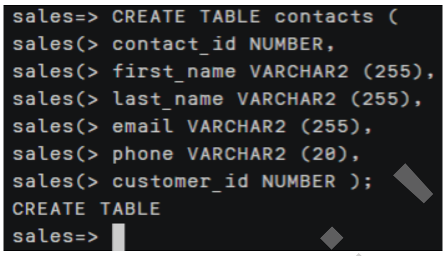
```
sales=> SET search_path TO Sales;
SET
```

（5）根据图例，在 Sales 模式下创建 12 张表，创建表时不添加约束；

步骤：

1. CONTACTS

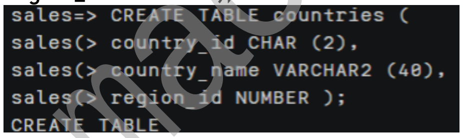
```
CREATE TABLE contacts (
contact_id NUMBER,
first_name VARCHAR2 (255),
last_name VARCHAR2 (255),
email VARCHAR2 (255),
phone VARCHAR2 (20),
customer_id NUMBER );
```



```
sales=> CREATE TABLE contacts (
sales(> contact_id NUMBER,
sales(> first_name VARCHAR2 (255),
sales(> last_name VARCHAR2 (255),
sales(> email VARCHAR2 (255),
sales(> phone VARCHAR2 (20),
sales(> customer_id NUMBER );
CREATE TABLE
sales=>
```

2. COUNTRIES

```
CREATE TABLE countries (
country_id CHAR (2),
country_name VARCHAR2 (40),
region_id NUMBER );
```

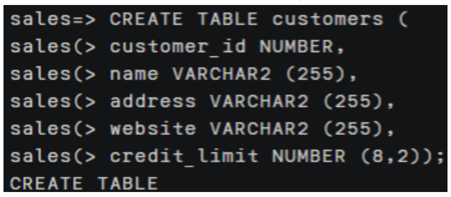


```
sales=> CREATE TABLE countries (
sales(> country_id CHAR (2),
sales(> country_name VARCHAR2 (40),
sales(> region_id NUMBER );
CREATE TABLE
```

3. CUSTOMERS

```
CREATE TABLE customers (
customer_id NUMBER,
name VARCHAR2 (255),
```

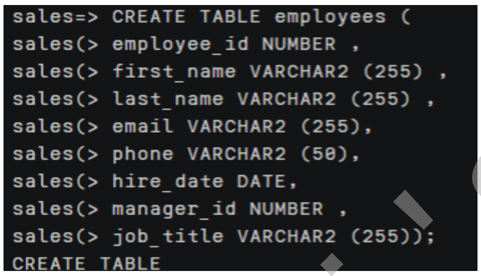
address VARCHAR2 (255),
website VARCHAR2 (255),
credit_limit NUMBER (8,2));



```
sales=> CREATE TABLE customers (
sales(> customer_id NUMBER,
sales(> name VARCHAR2 (255),
sales(> address VARCHAR2 (255),
sales(> website VARCHAR2 (255),
sales(> credit_limit NUMBER (8,2));
CREATE TABLE
```

4. EMPLOYEES

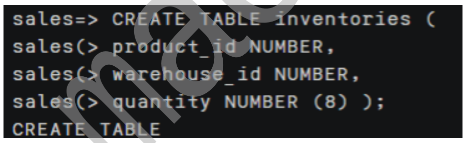
CREATE TABLE employees (
employee_id NUMBER ,
first_name VARCHAR2 (255) ,
last_name VARCHAR2 (255) ,
email VARCHAR2 (255),
phone VARCHAR2 (50),
hire_date DATE,
manager_id NUMBER ,
job_title VARCHAR2 (255));



```
sales=> CREATE TABLE employees (
sales(> employee_id NUMBER ,
sales(> first_name VARCHAR2 (255) ,
sales(> last_name VARCHAR2 (255) ,
sales(> email VARCHAR2 (255),
sales(> phone VARCHAR2 (50),
sales(> hire_date DATE,
sales(> manager_id NUMBER ,
sales(> job_title VARCHAR2 (255));
CREATE TABLE
```

5. INVENTORIES

CREATE TABLE inventories (
product_id NUMBER,
warehouse_id NUMBER,
quantity NUMBER (8));

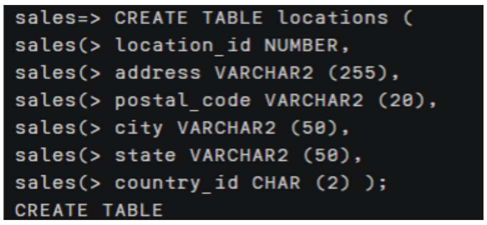


```
sales=> CREATE TABLE inventories (
sales(> product_id NUMBER,
sales(> warehouse_id NUMBER,
sales(> quantity NUMBER (8) );
CREATE TABLE
```

6. LOCATIONS

CREATE TABLE locations (
location_id NUMBER,

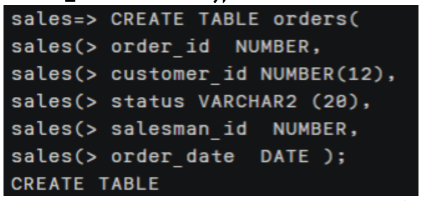
address VARCHAR2 (255),
postal_code VARCHAR2 (20),
city VARCHAR2 (50),
state VARCHAR2 (50),
country_id CHAR (2));



```
sales=> CREATE TABLE locations (
sales(> location_id NUMBER,
sales(> address VARCHAR2 (255),
sales(> postal_code VARCHAR2 (20),
sales(> city VARCHAR2 (50),
sales(> state VARCHAR2 (50),
sales(> country_id CHAR (2) );
CREATE TABLE
```

7. ORDERS

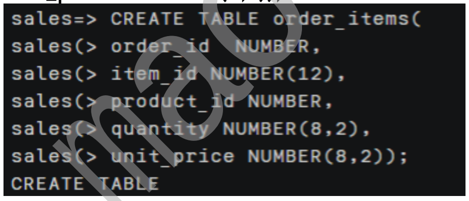
CREATE TABLE orders(
order_id   NUMBER,
customer_id NUMBER(12),
status VARCHAR2 (20),
salesman_id   NUMBER,
order_date   DATE);



```
sales=> CREATE TABLE orders(
sales(> order_id  NUMBER,
sales(> customer_id NUMBER(12),
sales(> status VARCHAR2 (20),
sales(> salesman_id  NUMBER,
sales(> order_date  DATE );
CREATE TABLE
```

8. ORDER_ITEMS

CREATE TABLE order_items(
order_id   NUMBER,
item_id NUMBER(12),
product_id NUMBER,
quantity NUMBER(8,2),
unit_price NUMBER(8,2));

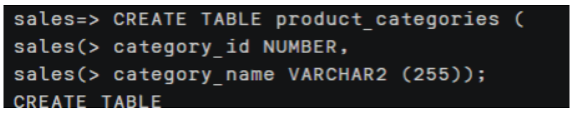


```
sales=> CREATE TABLE order_items(
sales(> order_id  NUMBER,
sales(> item_id NUMBER(12),
sales(> product_id NUMBER,
sales(> quantity NUMBER(8,2),
sales(> unit_price NUMBER(8,2));
CREATE TABLE
```

9. PRODUCT_CATEGORIES

CREATE TABLE product_categories (
category_id NUMBER,

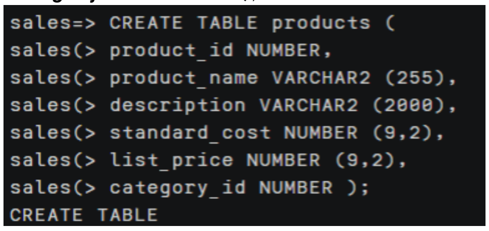
category_name VARCHAR2 (255));



```
sales=> CREATE TABLE product_categories (
sales(> category_id NUMBER,
sales(> category_name VARCHAR2 (255));
CREATE TABLE
```

10. PRODUCTS

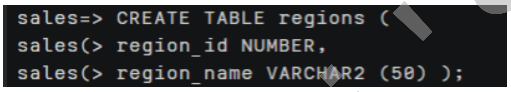
CREATE TABLE products (
product_id NUMBER,
product_name VARCHAR2 (255),
description VARCHAR2 (2000),
standard_cost NUMBER (9,2),
list_price NUMBER (9,2),
category_id NUMBER);



```
sales=> CREATE TABLE products (
sales(> product_id NUMBER,
sales(> product_name VARCHAR2 (255),
sales(> description VARCHAR2 (2000),
sales(> standard_cost NUMBER (9,2),
sales(> list_price NUMBER (9,2),
sales(> category_id NUMBER );
CREATE TABLE
```

11. REGIONS

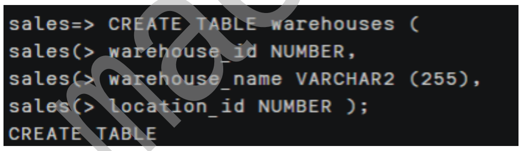
CREATE TABLE regions (
region_id NUMBER,
region_name VARCHAR2 (50));



```
sales=> CREATE TABLE regions (
sales(> region_id NUMBER,
sales(> region_name VARCHAR2 (50) );
```

12. WAREHOUSES

CREATE TABLE warehouses (
warehouse_id NUMBER,
warehouse_name VARCHAR2 (255),
location_id NUMBER);



```
sales=> CREATE TABLE warehouses (
sales(> warehouse_id NUMBER,
sales(> warehouse_name VARCHAR2 (255),
sales(> location_id NUMBER );
CREATE TABLE
```

以上，全部表创建完毕。

（6）分别在 customers 表的 name 字段上建立名为 idx_name 的索

引、website 字段上名为 unique_idx_website 的唯一索引、name 字段和 address 字段上名为 combined_idx_name-address 的组合索引；

步骤：

1.在 customers 表的 name 字段上建立名为 idx_name 的索引：

```
sales=> create index idx_name on customers(name);
CREATE INDEX
```

2.在 website 字段上名为 unique_idx_website 的唯一索引

```
sales=> create unique index unique_idx_website on customers(website);
CREATE INDEX
```

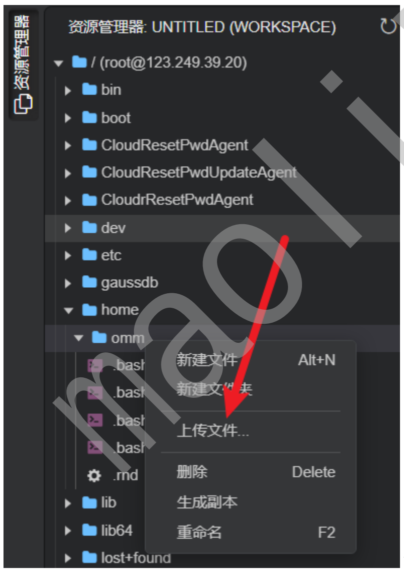
3.在 name 字段和 address 字段上名为 combined_idx_name_address 的组合索引

```
sales=> create index combined_idx_name_address on customers(name,address);
CREATE INDEX
```

（7）为 12 张表插入示例数据；

步骤：

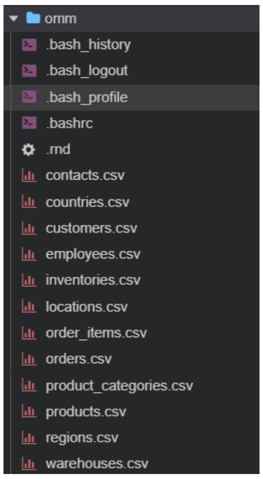
1.先把数据传入云服务器



注：order_itmes.csv 该文件命名有

误，需要先把名字修正为
order_items.csv

上传完毕：


omm
.bash_history
.bash_logout
.bash_profile
.bashrc
.rnd
contacts.csv
countries.csv
customers.csv
employees.csv
inventories.csv
locations.csv
order_items.csv
orders.csv
product_categories.csv
products.csv
regions.csv
warehouses.csv

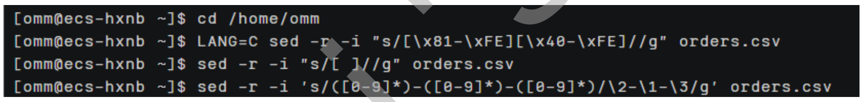
2. 先按照群里的通知修改出错文件的编码格式和日期格式(指令有修改)

cd /home/omm

LANG=C sed -r -i "s/[\x81-\xFE][\x40-\xFE]//g" orders.csv

sed -r -i "s/[ ]//g" orders.csv

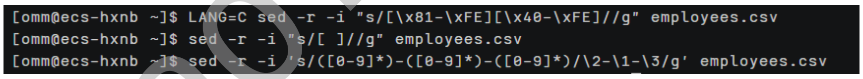
sed -r -i 's/([0-9]*)-([0-9]*)-([0-9]*)/\2-\1-\3/g' orders.csv


```
[omm@ecs-hxnb ~]$ cd /home/omm
[omm@ecs-hxnb ~]$ LANG=C sed -r -i "s/[\x81-\xFE][\x40-\xFE]//g" orders.csv
[omm@ecs-hxnb ~]$ sed -r -i "s/[ ]//g" orders.csv
[omm@ecs-hxnb ~]$ sed -r -i 's/([0-9]*)-([0-9]*)-([0-9]*)/\2-\1-\3/g' orders.csv
```

LANG=C sed -r -i "s/[\x81-\xFE][\x40-\xFE]//g" employees.csv

sed -r -i "s/[ ]//g" employees.csv

sed -r -i 's/([0-9]*)-([0-9]*)-([0-9]*)/\2-\1-\3/g' employees.csv


```
[omm@ecs-hxnb ~]$ LANG=C sed -r -i "s/[\x81-\xFE][\x40-\xFE]//g" employees.csv
[omm@ecs-hxnb ~]$ sed -r -i "s/[ ]//g" employees.csv
[omm@ecs-hxnb ~]$ sed -r -i 's/([0-9]*)-([0-9]*)-([0-9]*)/\2-\1-\3/g' employees.csv
```

3. 通过上传的本地文件，为表中导入数据：

~~\copy contacts FROM '/home/omm/contacts.csv' WITH (delimiter',',IGNORE_EXTRA_DATA 'on');~~

~~\copy contacts FROM '/home/omm/countries.csv' WITH (delimiter',',IGNORE_EXTRA_DATA 'on');~~

~~\copy contacts FROM '/home/omm/customers.csv' WITH (delimiter',',IGNORE_EXTRA_DATA 'on');~~

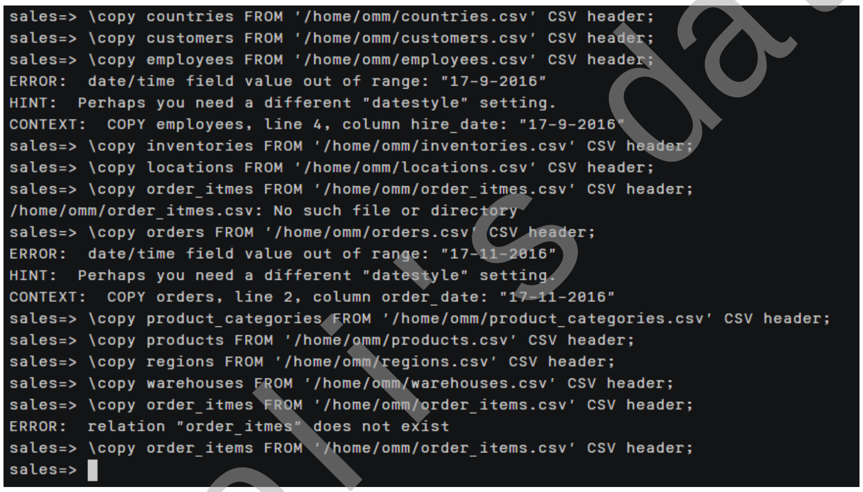
~~\copy contacts FROM '/home/omm/employees.csv' WITH (delimiter',',IGNORE_EXTRA_DATA 'on');~~

~~\copy contacts FROM '/home/omm/inventories.csv' WITH (delimiter',',IGNORE_EXTRA_DATA 'on');~~

~~\copy contacts FROM '/home/omm/locations.csv' WITH (delimiter',',IGNORE_EXTRA_DATA 'on');~~

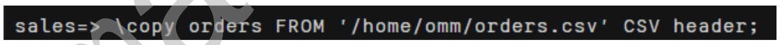
~~\copy contacts FROM '/home/omm/order_items.csv' WITH (delimiter',',IGNORE_EXTRA_DATA 'on');~~
~~\copy contacts FROM '/home/omm/orders.csv' WITH (delimiter',',IGNORE_EXTRA_DATA 'on');~~
~~\copy contacts FROM '/home/omm/product_categories.csv' WITH (delimiter',',IGNORE_EXTRA_DATA 'on');~~
~~\copy contacts FROM '/home/omm/products.csv' WITH (delimiter',',IGNORE_EXTRA_DATA 'on');~~
~~\copy contacts FROM '/home/omm/regions.csv' WITH (delimiter',',IGNORE_EXTRA_DATA 'on');~~
~~\copy contacts FROM '/home/omm/warehouses.csv' WITH (delimiter',',IGNORE_EXTRA_DATA 'on');~~

```
\copy contacts FROM '/home/omm/contacts.csv' CSV header;
\copy countries FROM '/home/omm/countries.csv' CSV header;
\copy customers FROM '/home/omm/customers.csv' CSV header;
\copy employees FROM '/home/omm/employees.csv' CSV header;
\copy inventories FROM '/home/omm/inventories.csv' CSV header;
\copy locations FROM '/home/omm/locations.csv' CSV header;
\copy order_items FROM '/home/omm/order_items.csv' CSV header;
\copy orders FROM '/home/omm/orders.csv' CSV header;
\copy product_categories FROM '/home/omm/product_categories.csv' CSV header;
\copy products FROM '/home/omm/products.csv' CSV header;
\copy regions FROM '/home/omm/regions.csv' CSV header;
\copy warehouses FROM '/home/omm/warehouses.csv' CSV header;
```

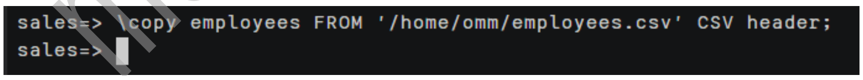


```
sales=> \copy countries FROM '/home/omm/countries.csv' CSV header;
sales=> \copy customers FROM '/home/omm/customers.csv' CSV header;
sales=> \copy employees FROM '/home/omm/employees.csv' CSV header;
ERROR:  date/time field value out of range: "17-9-2016"
HINT:  Perhaps you need a different "datestyle" setting.
CONTEXT:  COPY employees, line 4, column hire_date: "17-9-2016"
sales=> \copy inventories FROM '/home/omm/inventories.csv' CSV header;
sales=> \copy locations FROM '/home/omm/locations.csv' CSV header;
sales=> \copy order_itmes FROM '/home/omm/order_itmes.csv' CSV header;
/home/omm/order_itmes.csv: No such file or directory
sales=> \copy orders FROM '/home/omm/orders.csv' CSV header;
ERROR:  date/time field value out of range: "17-11-2016"
HINT:  Perhaps you need a different "datestyle" setting.
CONTEXT:  COPY orders, line 2, column order_date: "17-11-2016"
sales=> \copy product_categories FROM '/home/omm/product_categories.csv' CSV header;
sales=> \copy products FROM '/home/omm/products.csv' CSV header;
sales=> \copy regions FROM '/home/omm/regions.csv' CSV header;
sales=> \copy warehouses FROM '/home/omm/warehouses.csv' CSV header;
sales=> \copy order_itmes FROM '/home/omm/order_items.csv' CSV header;
ERROR:  relation "order_itmes" does not exist
sales=> \copy order_items FROM '/home/omm/order_items.csv' CSV header;
sales=> 
```

由于部分 csv 日期有误，修改数据后：



```
sales=> \copy orders FROM '/home/omm/orders.csv' CSV header;
```

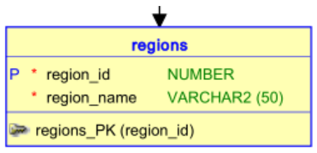


```
sales=> \copy employees FROM '/home/omm/employees.csv' CSV header;
sales=> 
```

（8）修改 12 张表的结构，根据 4.1 节图例含义添加相应的约束，如主码、外码、非空；（部分添加依旧有报错，待后续学习）

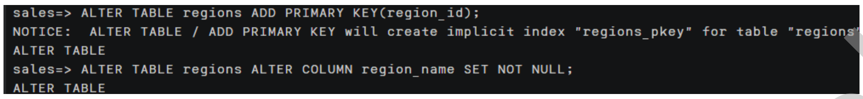
步骤:

1.regions



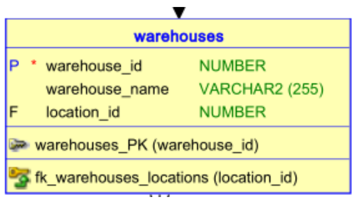
ALTER TABLE regions ADD PRIMARY KEY(region_id);

ALTER TABLE regions ALTER COLUMN region_name SET NOT NULL;



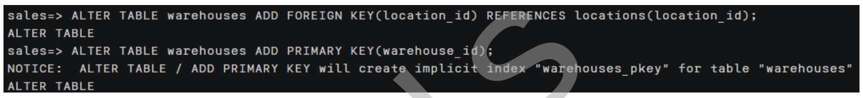
```
sales=> ALTER TABLE regions ADD PRIMARY KEY(region_id);
NOTICE:  ALTER TABLE / ADD PRIMARY KEY will create implicit index "regions_pkey" for table "regions"
ALTER TABLE
sales=> ALTER TABLE regions ALTER COLUMN region_name SET NOT NULL;
ALTER TABLE
```

2. warehouses


warehouses
P * warehouse_id NUMBER
warehouse_name VARCHAR2 (255)
F location_id NUMBER
warehouses_PK (warehouse_id)
fk_warehouses_locations (location_id)

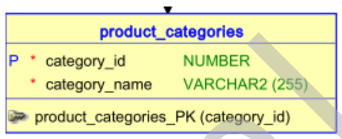
ALTER TABLE warehouses ADD FOREIGN KEY(location_id) REFERENCES locations(location_id);

ALTER TABLE warehouses ADD PRIMARY KEY(warehouse_id);



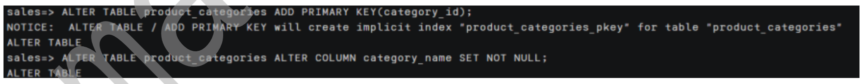
```
sales=> ALTER TABLE warehouses ADD FOREIGN KEY(location_id) REFERENCES locations(location_id);
ALTER TABLE
sales=> ALTER TABLE warehouses ADD PRIMARY KEY(warehouse_id);
NOTICE:  ALTER TABLE / ADD PRIMARY KEY will create implicit index "warehouses_pkey" for table "warehouses"
ALTER TABLE
```

3. product_categories


product_categories
P * category_id NUMBER
* category_name VARCHAR2 (255)
product_categories_PK (category_id)

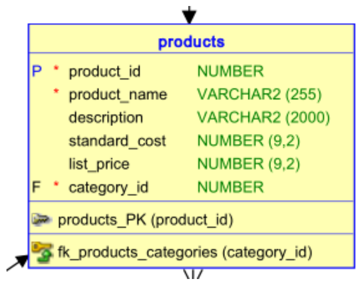
ALTER TABLE product_categories ADD PRIMARY KEY(category_id);

ALTER TABLE product_categories ALTER COLUMN category_name SET NOT NULL;



```
sales=> ALTER TABLE product_categories ADD PRIMARY KEY(category_id);
NOTICE:  ALTER TABLE / ADD PRIMARY KEY will create implicit index "product_categories_pkey" for table "product_categories"
ALTER TABLE
sales=> ALTER TABLE product_categories ALTER COLUMN category_name SET NOT NULL;
ALTER TABLE
```

4.product


products
P * product_id NUMBER
* product_name VARCHAR2 (255)
description VARCHAR2 (2000)
standard_cost NUMBER (9,2)
list_price NUMBER (9,2)
F * category_id NUMBER
products_PK (product_id)
fk_products_categories (category_id)

ALTER TABLE products ADD PRIMARY KEY(product_id);

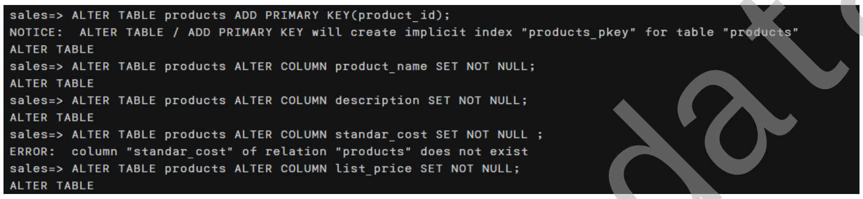
ALTER TABLE products ALTER COLUMN product_name SET NOT NULL;

ALTER TABLE products ALTER COLUMN description SET NOT NULL;

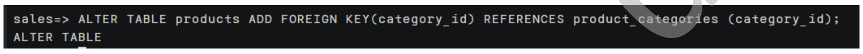
ALTER TABLE products ALTER COLUMN standar_cost SET NOT NULL ;

ALTER TABLE products ALTER COLUMN list_price SET NOT NULL;

ALTER TABLE products ADD FOREIGN KEY(category_id) REFERENCES product_categories (category_id);

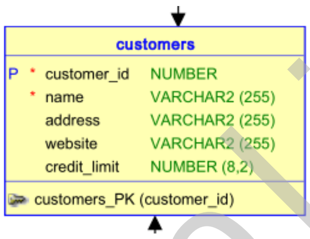


```
sales=> ALTER TABLE products ADD PRIMARY KEY(product_id);
NOTICE:  ALTER TABLE / ADD PRIMARY KEY will create implicit index "products_pkey" for table "products"
ALTER TABLE
sales=> ALTER TABLE products ALTER COLUMN product_name SET NOT NULL;
ALTER TABLE
sales=> ALTER TABLE products ALTER COLUMN description SET NOT NULL;
ALTER TABLE
sales=> ALTER TABLE products ALTER COLUMN standar_cost SET NOT NULL ;
ERROR:  column "standar_cost" of relation "products" does not exist
sales=> ALTER TABLE products ALTER COLUMN list_price SET NOT NULL;
ALTER TABLE
```



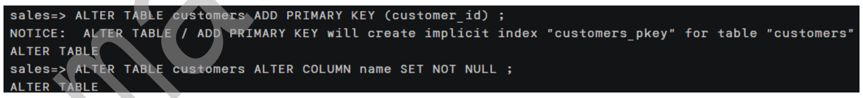
```
sales=> ALTER TABLE products ADD FOREIGN KEY(category_id) REFERENCES product_categories (category_id);
ALTER TABLE
```

5.customers



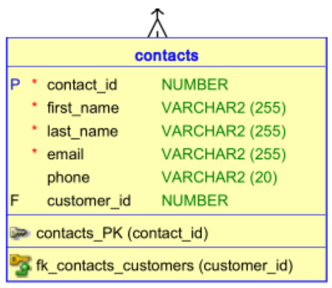
ALTER TABLE customers ADD PRIMARY KEY (customer_id) ;

ALTER TABLE customers ALTER COLUMN name SET NOT NULL ;



```
sales=> ALTER TABLE customers ADD PRIMARY KEY (customer_id) ;
NOTICE:  ALTER TABLE / ADD PRIMARY KEY will create implicit index "customers_pkey" for table "customers"
ALTER TABLE
sales=> ALTER TABLE customers ALTER COLUMN name SET NOT NULL ;
ALTER TABLE
```

6.contacts



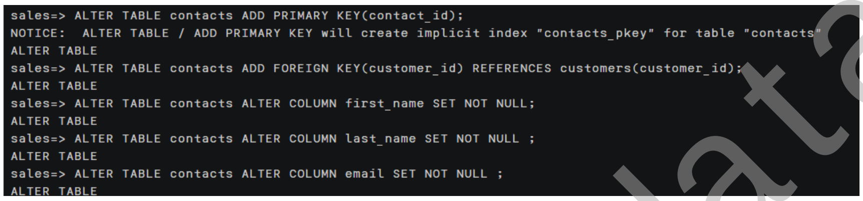
ALTER TABLE contacts ADD PRIMARY KEY(contact_id);

ALTER TABLE contacts ADD FOREIGN KEY(customer_id) REFERENCES customers(customer_id);

ALTER TABLE contacts ALTER COLUMN first_name SET NOT NULL;

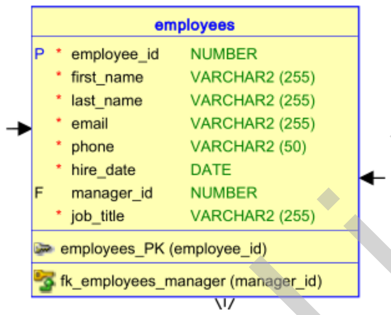
ALTER TABLE contacts ALTER COLUMN last_name SET NOT NULL ;

ALTER TABLE contacts ALTER COLUMN email SET NOT NULL ;



```
sales=> ALTER TABLE contacts ADD PRIMARY KEY(contact_id);
NOTICE:  ALTER TABLE / ADD PRIMARY KEY will create implicit index "contacts_pkey" for table "contacts"
ALTER TABLE
sales=> ALTER TABLE contacts ADD FOREIGN KEY(customer_id) REFERENCES customers(customer_id);
ALTER TABLE
sales=> ALTER TABLE contacts ALTER COLUMN first_name SET NOT NULL;
ALTER TABLE
sales=> ALTER TABLE contacts ALTER COLUMN last_name SET NOT NULL ;
ALTER TABLE
sales=> ALTER TABLE contacts ALTER COLUMN email SET NOT NULL ;
ALTER TABLE
```

7.employees


employees
P * employee_id NUMBER
* first_name VARCHAR2 (255)
* last_name VARCHAR2 (255)
* email VARCHAR2 (255)
* phone VARCHAR2 (50)
* hire_date DATE
F manager_id NUMBER
* job_title VARCHAR2 (255)
employees_PK (employee_id)
fk_employees_manager (manager_id)

ALTER TABLE employees ADD PRIMARY KEY (employee_id);

ALTER TABLE employees ADD FOREIGN KEY(manager_id) REFERENCES employees(manager_id);

ALTER TABLE employees ALTER COLUMN first_name SET NOT NULL ;

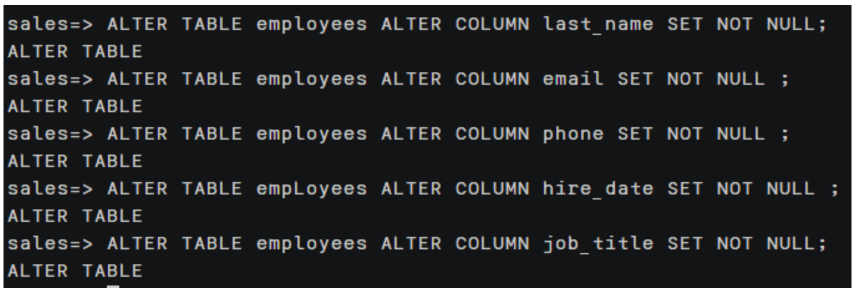
ALTER TABLE employees ALTER COLUMN last_name SET NOT NULL;

ALTER TABLE employees ALTER COLUMN email SET NOT NULL ;

ALTER TABLE employees ALTER COLUMN phone SET NOT NULL ;

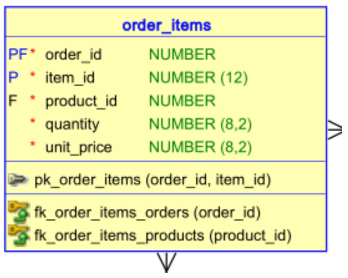
ALTER TABLE empLoyees ALTER COLUMN hire_date SET NOT NULL ;

ALTER TABLE employees ALTER COLUMN job_title SET NOT NULL;



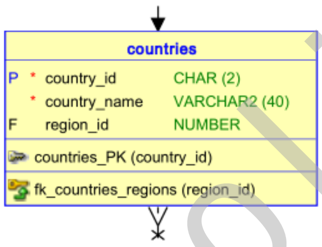
```
sales=> ALTER TABLE employees ALTER COLUMN last_name SET NOT NULL;
ALTER TABLE
sales=> ALTER TABLE employees ALTER COLUMN email SET NOT NULL ;
ALTER TABLE
sales=> ALTER TABLE employees ALTER COLUMN phone SET NOT NULL ;
ALTER TABLE
sales=> ALTER TABLE employees ALTER COLUMN hire_date SET NOT NULL ;
ALTER TABLE
sales=> ALTER TABLE employees ALTER COLUMN job_title SET NOT NULL;
ALTER TABLE
```

8.order_items（后期截图略）


order_items
PF * order_id NUMBER
P * item_id NUMBER (12)
F * product_id NUMBER
* quantity NUMBER (8,2)
* unit_price NUMBER (8,2)
pk_order_items (order_id, item_id)
fk_order_items_orders (order_id)
fk_order_items_products (product_id)

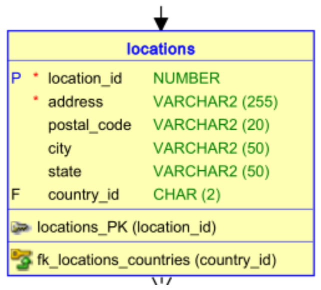
ALTER TABLE order_items ADD FOREIGN KEY(order_id) REFERENCES orders(order_id);
ALTER TABLE order_items ADD PRIMARY KEY(item_id);
ALTER TABLE order_items ADD FOREIGN KEY(products_id) REFERENCES products(product_id);
ALTER TABLE order_items ALTER COLUMN quantity SET NOT NULL;
ALTER TABLE order_items ALTER COLUMN unit_price SET NOT NULL;

9.countries


countries
P * country_id CHAR (2)
* country_name VARCHAR2 (40)
F region_id NUMBER
countries_PK (country_id)
fk_countries_regions (region_id)

ALTER TABLE countries ADD PRIMARY KEY(country_id);
ALTER TABLE countries ADD FOREIGN KEY(region_id) REFERENCES regions(region_id);
ALTER TABLE countries ALTER COLUMN country_name SET NOT NULL;

10.locations

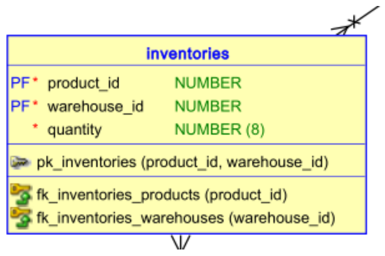


ALTER TABLE locations ALTER COLUMN city SET NOT NULL；

ALTER TABLE locations ADD PRIMARY KEY(location_id);

ALTER TABLE locations ADD FOREIGN KEY(country_id) REFERENCES countries(country_id);

11.inventories


inventories
PF * product_id NUMBER
PF * warehouse_id NUMBER
* quantity NUMBER (8)
pk_inventories (product_id, warehouse_id)
fk_inventories_products (product_id)
fk_inventories_warehouses (warehouse_id)

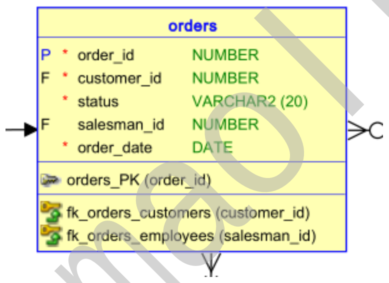
ALTER TABLE inventories ADD PRIMARY KEY(product_ id);

ALTER TABLE inventories ADD FOREIGN KEY(product_id) REFERENCES products(product_id);

ALTER TABLE inventories ADD FOREIGN KEY(warehouse_id) REFERENCES warehouses(warehouse_id);

ALTER TABLE inventories ALTER COLUMN quantity SET NOT NULL;

12.orders



ALTER TABLE orders ADD PRIMARY KEY(order_id);

ALTER TABLE orders ADD FOREIGN KEY(customer_id) REFERENCES customers(customer_id);

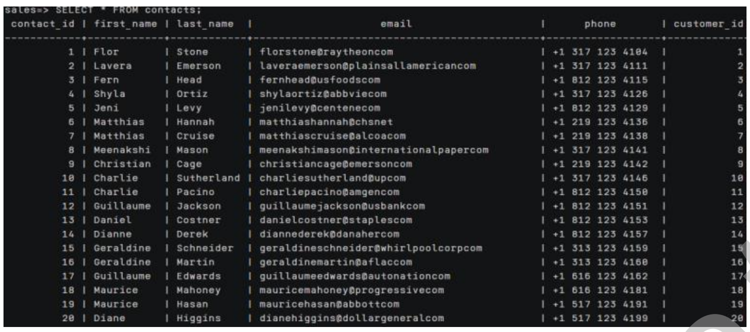
ALTER TABLE orders ADD FOREIGN KEY(salesman_id) REFERENCES employees(employee_id);

ALTER TABLE orders ALTER COLUMN status SET NOT NULL;

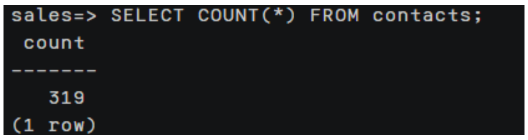
ALTER TABLE orders ALTER COLUMN order_date SET NOT NULL;

（9）依次使用查询语句“SELECT * FROM 表名;” 查询所有 12 张表的数据，依次使用查询语句“SELECT COUNT(*) FROM 表名”查询所有 12 张表的每张表所含数据的数目;

1. CONTACTS

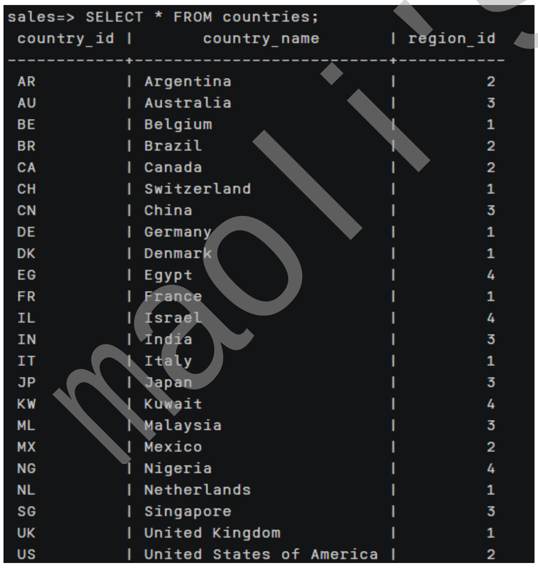


```
sales=> SELECT * FROM contacts;
 contact_id | first_name | last_name  |                 email                 |      phone      | customer_id
------------+------------+------------+---------------------------------------+-----------------+-------------
          1 | Flor       | Stone      | florstone@raytheoncom                 | +1 317 123 4104 |           1
          2 | Lavera     | Emerson    | laveraemerson@plainsallamericancom    | +1 317 123 4111 |           2
          3 | Fern       | Head       | fernhead@usfoodscom                   | +1 812 123 4115 |           3
          4 | Shyla      | Ortiz      | shylaortiz@abbviecom                  | +1 317 123 4126 |           4
          5 | Jeni       | Levy       | jenilevy@centenecom                   | +1 812 123 4129 |           5
          6 | Matthias   | Hannah     | matthiashannah@chsnet                 | +1 219 123 4136 |           6
          7 | Matthias   | Cruise     | matthiascruise@alcoacom               | +1 219 123 4138 |           7
          8 | Meenakshi  | Mason      | meenakshimason@internationalpapercom  | +1 317 123 4141 |           8
          9 | Christian  | Cage       | christiancage@emersoncom              | +1 219 123 4142 |           9
         10 | Charlie    | Sutherland | charliesutherland@upcom               | +1 317 123 4146 |          10
         11 | Charlie    | Pacino     | charliepacino@amgencom                | +1 812 123 4150 |          11
         12 | Guillaume  | Jackson    | guillaumejackson@usbankcom            | +1 812 123 4151 |          12
         13 | Daniel     | Costner    | danielcostner@staplescom              | +1 812 123 4153 |          13
         14 | Dianne     | Derek      | diannederek@danahercom                | +1 812 123 4157 |          14
         15 | Geraldine  | Schneider  | geraldineschneider@whirlpoolcorpcom   | +1 313 123 4159 |          15
         16 | Geraldine  | Martin     | geraldinemartin@aflaccom              | +1 313 123 4160 |          16
         17 | Guillaume  | Edwards    | guillaumeedwards@autonationcom        | +1 616 123 4162 |          17
         18 | Maurice    | Mahoney    | mauricemahoney@progressivecom         | +1 616 123 4181 |          18
         19 | Maurice    | Hasan      | mauricehasan@abbottcom                | +1 517 123 4191 |          19
         20 | Diane      | Higgins    | dianehiggins@dollargeneralcom         | +1 517 123 4199 |          20
```

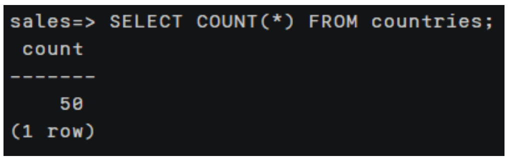


```
sales=> SELECT COUNT(*) FROM contacts;
 count
-------
   319
(1 row)
```

2. COUNTRIES

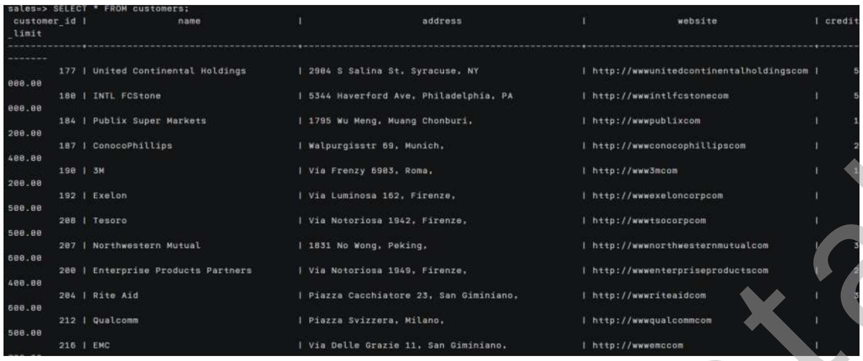


```
sales=> SELECT * FROM countries;
 country_id |       country_name       | region_id
------------+--------------------------+-----------
 AR         | Argentina                |         2
 AU         | Australia                |         3
 BE         | Belgium                  |         1
 BR         | 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
```

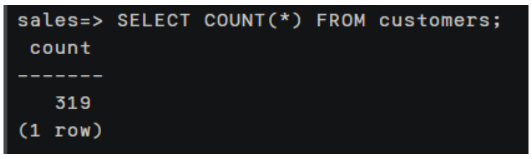


```
sales=> SELECT COUNT(*) FROM countries;
 count
-------
    50
(1 row)
```

3. CUSTOMERS

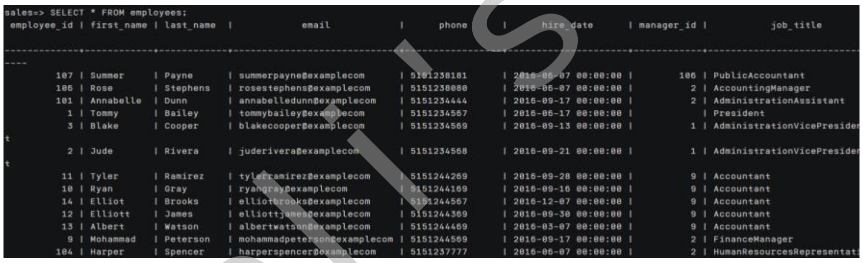


```
sales=> SELECT * FROM customers;
 customer_id |              name              |                address                |                  website                  | credit
_limit
-------------+--------------------------------+---------------------------------------+-------------------------------------------+-------
-------
         177 | United Continental Holdings    | 2904 S Salina St, Syracuse, NY        | http://wwwunitedcontinentalholdingscom |      5
000.00
         180 | INTL FCStone                   | 5344 Haverford Ave, Philadelphia, PA  | http://wwwintlfcstonecom                |      5
000.00
         184 | Publix Super Markets           | 1795 Wu Meng, Muang Chonburi,         | http://wwwpublixcom                     |      1
200.00
         187 | ConocoPhillips                 | Walpurgisstr 69, Munich,              | http://wwwconocophillipscom             |      2
400.00
         190 | 3M                             | Via Frenzy 6903, Roma,                | http://www3mcom                         |      1
200.00
         192 | Exelon                         | Via Luminosa 162, Firenze,            | http://wwwexeloncorpcom                 |
500.00
         208 | Tesoro                         | Via Notoriosa 1942, Firenze,          | http://wwwtsocorpcom                    |
500.00
         207 | Northwestern Mutual            | 1831 No Wong, Peking,                 | http://wwwnorthwesternmutualcom         |      3
600.00
         200 | Enterprise Products Partners   | Via Notoriosa 1949, Firenze,          | http://wwwenterpriseproductscom         |      2
400.00
         204 | Rite Aid                       | Piazza Cacchiatore 23, San Giminiano, | http://wwwriteaidcom                    |      3
600.00
         212 | Qualcomm                       | Piazza Svizzera, Milano,              | http://wwwqualcommcom                   |
500.00
         216 | EMC                            | Via Delle Grazie 11, San Giminiano,   | http://wwwemccom                        |
```

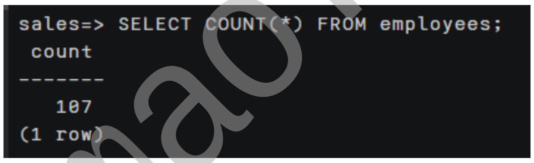


```
sales=> SELECT COUNT(*) FROM customers;
 count
-------
   319
(1 row)
```

4. EMPLOYEES

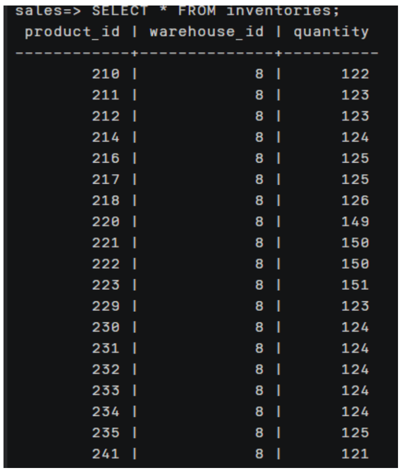


```
sales=> SELECT * FROM employees;
 employee_id | first_name | last_name |             email              |    phone     |      hire_date      | manager_id |        job_title

-------------+------------+-----------+--------------------------------+--------------+---------------------+------------+----------------------
----
         107 | Summer     | Payne     | summerpayne@examplecom         | 5151238181   | 2016-06-07 00:00:00 |        106 | PublicAccountant
         106 | Rose       | Stephens  | rosestephens@examplecom        | 5151238080   | 2016-06-07 00:00:00 |          2 | AccountingManager
         101 | Annabelle  | Dunn      | annabelledunn@examplecom       | 5151234444   | 2016-09-17 00:00:00 |          2 | AdministrationAssistant
           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 | AdministrationVicePresiden
t
           2 | Jude       | Rivera    | juderivera@examplecom          | 5151234568   | 2016-09-21 00:00:00 |          1 | AdministrationVicePresiden
t
          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
          13 | Albert     | Watson    | albertwatson@examplecom        | 5151244469   | 2016-03-07 00:00:00 |          9 | Accountant
           9 | Mohammad   | Peterson  | mohammadpeterson@examplecom    | 5151244569   | 2016-09-17 00:00:00 |          2 | FinanceManager
         104 | Harper     | Spencer   | harperspencer@examplecom       | 5151237777   | 2016-06-07 00:00:00 |          2 | HumanResourcesRepresentat
```

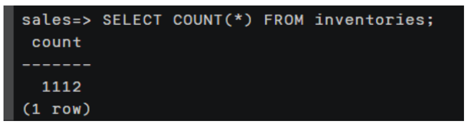


```
sales=> SELECT COUNT(*) FROM employees;
 count
-------
   107
(1 row)
```

5. INVENTORIES

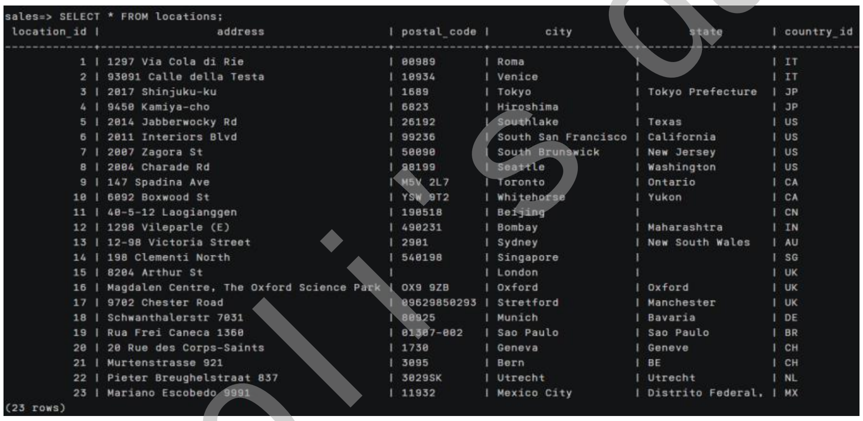


```
sales=> SELECT * FROM inventories;
 product_id | warehouse_id | quantity
------------+--------------+----------
        210 |            8 |      122
        211 |            8 |      123
        212 |            8 |      123
        214 |            8 |      124
        216 |            8 |      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
```

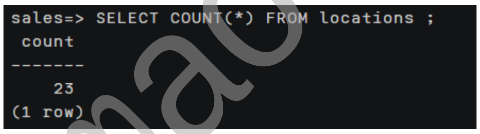


```
sales=> SELECT COUNT(*) FROM inventories;
 count
-------
  1112
(1 row)
```

6. LOCATIONS

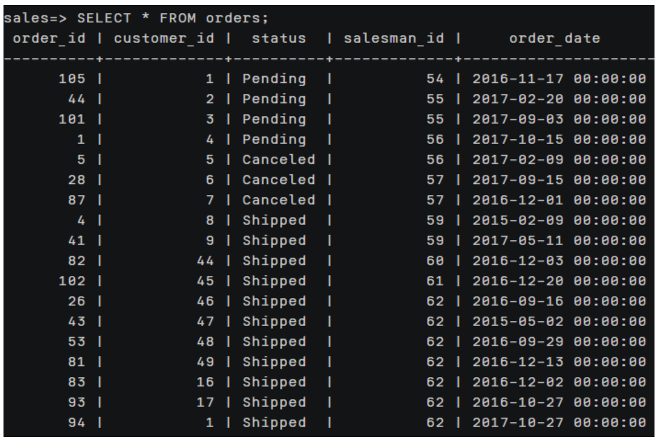


```
sales=> SELECT * FROM locations;
 location_id |                  address                   | postal_code |         city        |       state       | country_id
-------------+--------------------------------------------+-------------+---------------------+-------------------+------------
           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)
```

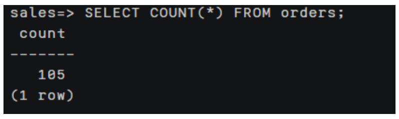


```
sales=> SELECT COUNT(*) FROM locations ;
 count
-------
    23
(1 row)
```

7. ORDERS

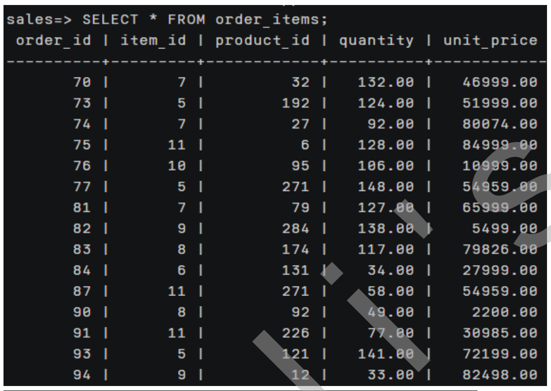


```
sales=> SELECT * FROM orders;
 order_id | customer_id |  status   | salesman_id |      order_date
----------+-------------+-----------+-------------+---------------------
      105 |           1 | Pending   |          54 | 2016-11-17 00:00:00
       44 |           2 | Pending   |          55 | 2017-02-20 00:00:00
      101 |           3 | Pending   |          55 | 2017-09-03 00:00:00
        1 |           4 | Pending   |          56 | 2017-10-15 00:00:00
        5 |           5 | Canceled  |          56 | 2017-02-09 00:00:00
       28 |           6 | Canceled  |          57 | 2017-09-15 00:00:00
       87 |           7 | Canceled  |          57 | 2016-12-01 00:00:00
        4 |           8 | Shipped   |          59 | 2015-02-09 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 | 2016-09-16 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
```

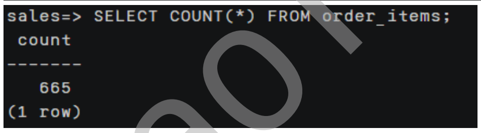


```
sales=> SELECT COUNT(*) FROM orders;
 count
-------
   105
(1 row)
```

8. ORDER_ITEMS

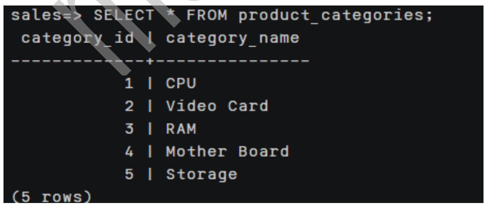


```
sales=> SELECT * FROM order_items;
 order_id | item_id | product_id | quantity | unit_price
----------+---------+------------+----------+------------
       70 |       7 |         32 |   132.00 |   46999.00
       73 |       5 |        192 |   124.00 |   51999.00
       74 |       7 |         27 |    92.00 |   80074.00
       75 |      11 |          6 |   128.00 |   84999.00
       76 |      10 |         95 |   106.00 |   10999.00
       77 |       5 |        271 |   148.00 |   54959.00
       81 |       7 |         79 |   127.00 |   65999.00
       82 |       9 |        284 |   138.00 |    5499.00
       83 |       8 |        174 |   117.00 |   79826.00
       84 |       6 |        131 |    34.00 |   27999.00
       87 |      11 |        271 |    58.00 |   54959.00
       90 |       8 |         92 |    49.00 |    2200.00
       91 |      11 |        226 |    77.00 |   30985.00
       93 |       5 |        121 |   141.00 |   72199.00
       94 |       9 |         12 |    33.00 |   82498.00
```

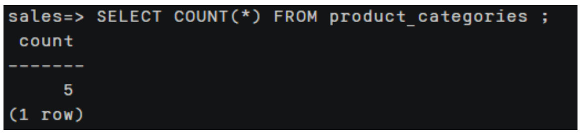


```
sales=> SELECT COUNT(*) FROM order_items;
 count
-------
   665
(1 row)
```

9. PRODUCT_CATEGORIES

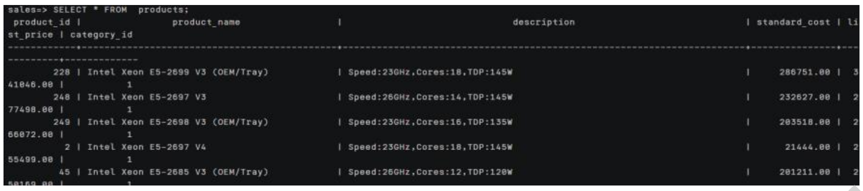


```
sales=> SELECT * FROM product_categories;
 category_id | category_name
-------------+---------------
           1 | CPU
           2 | Video Card
           3 | RAM
           4 | Mother Board
           5 | Storage
(5 rows)
```

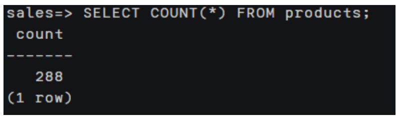


```
sales=> SELECT COUNT(*) FROM product_categories ;
 count
-------
     5
(1 row)
```

10. PRODUCTS

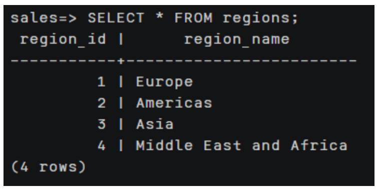


```
sales=> SELECT * FROM  products;
 product_id |              product_name              |                          description                          | standard_cost | li
st_price | category_id
------------+----------------------------------------+---------------------------------------------------------------+---------------+---
---------+-------------
        228 | Intel Xeon E5-2699 V3 (OEM/Tray)       | Speed:23GHz,Cores:18,TDP:145W                                 |     286751.00 |  3
41046.00 |           1
        248 | Intel Xeon E5-2697 V3                  | Speed:26GHz,Cores:14,TDP:145W                                 |     232627.00 |  2
77498.00 |           1
        249 | Intel Xeon E5-2698 V3 (OEM/Tray)       | Speed:23GHz,Cores:16,TDP:135W                                 |     203518.00 |  2
66072.00 |           1
          2 | Intel Xeon E5-2697 V4                  | Speed:23GHz,Cores:18,TDP:145W                                 |      21444.00 |  2
55499.00 |           1
         45 | Intel Xeon E5-2685 V3 (OEM/Tray)       | Speed:26GHz,Cores:12,TDP:120W                                 |     201211.00 |  2
50169.00 |           1
```

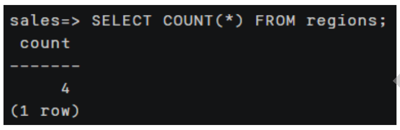


```
sales=> SELECT COUNT(*) FROM products;
 count
-------
   288
(1 row)
```

11. REGIONS

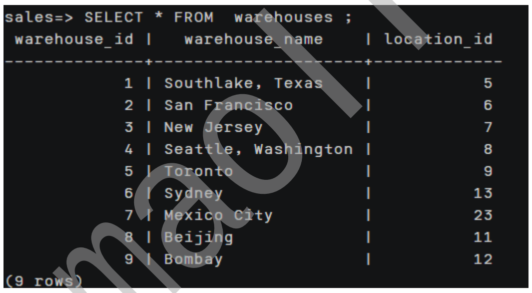


```
sales=> SELECT * FROM regions;
 region_id |       region_name
-----------+------------------------
         1 | Europe
         2 | Americas
         3 | Asia
         4 | Middle East and Africa
(4 rows)
```

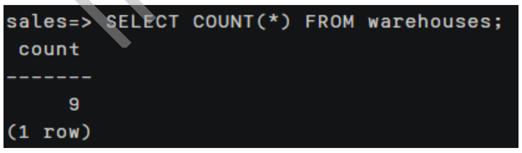


```
sales=> SELECT COUNT(*) FROM regions;
 count
-------
     4
(1 row)
```

12. WAREHOUSES



```
sales=> 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)
```



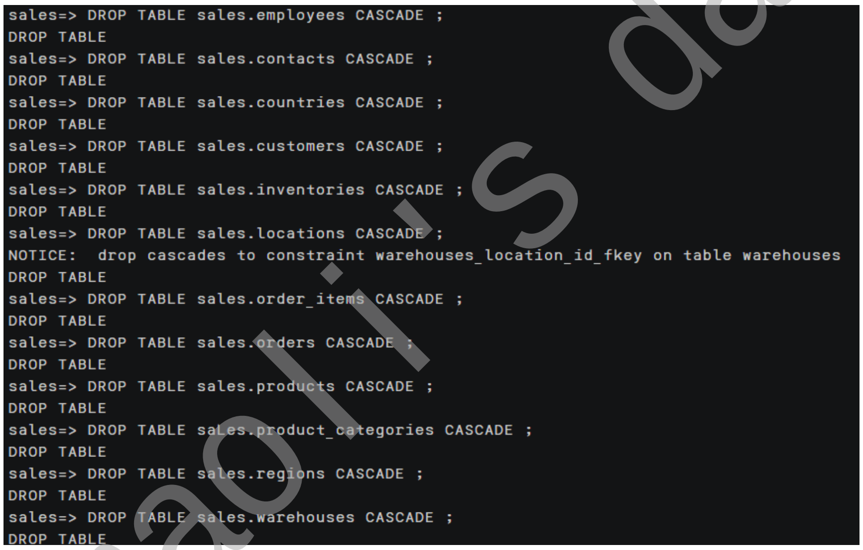
```
sales=> SELECT COUNT(*) FROM warehouses;
 count
-------
     9
(1 row)
```

（10）完成上述任务后依次删除用户、数据库、模式、索引以及所有的表。

步骤：

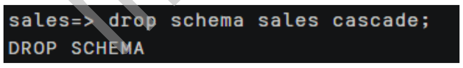
1.删除所有的表

```
DROP TABLE sales.employees CASCADE ;
DROP TABLE sales.contacts CASCADE ;
DROP TABLE sales.countries CASCADE ;
DROP TABLE sales.customers CASCADE ;
DROP TABLE sales.inventories CASCADE ;
DROP TABLE sales.locations CASCADE ;
DROP TABLE sales.order_items CASCADE ;
DROP TABLE sales.orders CASCADE ;
DROP TABLE sales.products CASCADE ;
DROP TABLE saLes.product_categories CASCADE ;
DROP TABLE sales.regions CASCADE ;
DROP TABLE sales.warehouses CASCADE ;
```



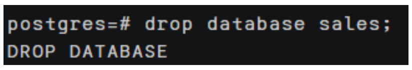
```
sales=> DROP TABLE sales.employees CASCADE ;
DROP TABLE
sales=> DROP TABLE sales.contacts CASCADE ;
DROP TABLE
sales=> DROP TABLE sales.countries CASCADE ;
DROP TABLE
sales=> DROP TABLE sales.customers CASCADE ;
DROP TABLE
sales=> DROP TABLE sales.inventories CASCADE ;
DROP TABLE
sales=> DROP TABLE sales.locations CASCADE ;
NOTICE:  drop cascades to constraint warehouses_location_id_fkey on table warehouses
DROP TABLE
sales=> DROP TABLE sales.order_items CASCADE ;
DROP TABLE
sales=> DROP TABLE sales.orders CASCADE ;
DROP TABLE
sales=> DROP TABLE sales.products CASCADE ;
DROP TABLE
sales=> DROP TABLE saLes.product_categories CASCADE ;
DROP TABLE
sales=> DROP TABLE sales.regions CASCADE ;
DROP TABLE
sales=> DROP TABLE sales.warehouses CASCADE ;
DROP TABLE
```

2.删除模式



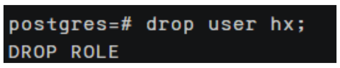
```
sales=> drop schema sales cascade;
DROP SCHEMA
```

3.删除数据库



```
postgres=# drop database sales;
DROP DATABASE
```

4.删除用户



```
postgres=# drop user hx;
DROP ROLE
```

3.实验总结

3.1 完成的工作

设计正确的 SQL 语句实现用户、数据库、模式、基本表和索引的创建、修改和删除。

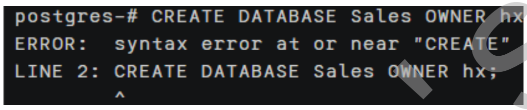
3.2 对实验的认识

通过实验掌握了什么，理解了什么，对实验的思考，等等。

（1）创建的数据库名能否与用户名相同？请上机验证。

3.3 遇到的困难及解决方法

1. 出现以下错误：



```
postgres-# CREATE DATABASE Sales OWNER hx
ERROR:  syntax error at or near "CREATE"
LINE 2: CREATE DATABASE Sales OWNER hx;
        ^
```

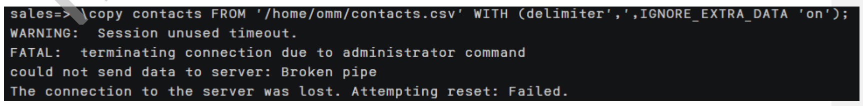
原因：上一步操作没有分号，导致错误

2. postgres ERROR: syntax error at or near "create"

首次接触 postgres 时，创建数据库报错

原因：这数据库区分大小写，而且和 linux 的没有结果就是最好的结果的理论相反，如果没有输出结果，说明命令写错了

3.出现以下错误：

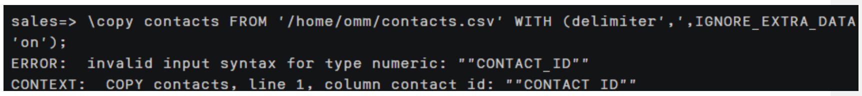


```
sales=> \copy contacts FROM '/home/omm/contacts.csv' WITH (delimiter',',IGNORE_EXTRA_DATA 'on');
WARNING:  Session unused timeout.
FATAL:  terminating connection due to administrator command
could not send data to server: Broken pipe
The connection to the server was lost. Attempting reset: Failed.
```

原因：这个错误是由于长时间未操作掉线导致的，需要按照上文

提到的实验步骤重新操作部分指令。

4.出现以下错误：



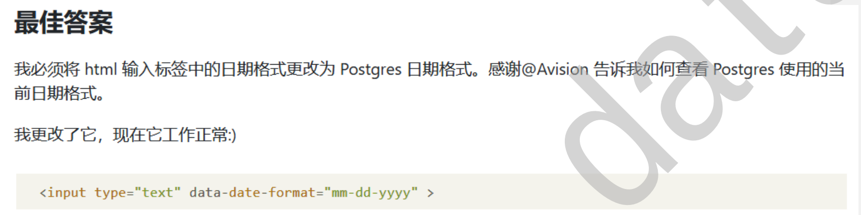
```
sales=> \copy contacts FROM '/home/omm/contacts.csv' WITH (delimiter',',IGNORE_EXTRA_DATA
'on');
ERROR:  invalid input syntax for type numeric: ""CONTACT_ID""
CONTEXT:  COPY contacts, line 1, column contact_id: ""CONTACT_ID""
```

原因：csv 文件中包含表头，不能按照文档中的方法导入，需要更改为\copy countries FROM '/home/omm/countries.csv' CSV header;

5.导入数据有误

解决：

https://www.coder.work/article/502432



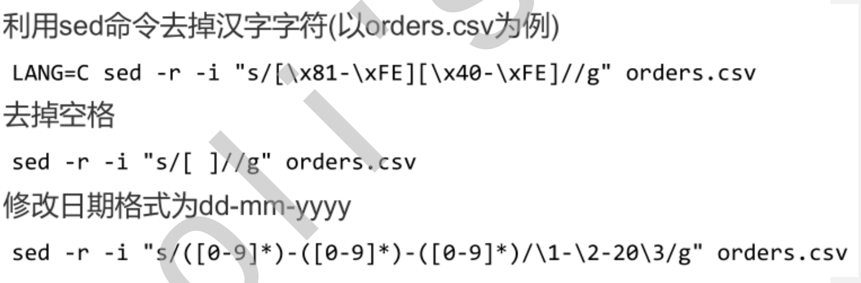
最佳答案

我必须将 html 输入标签中的日期格式更改为 Postgres 日期格式。感谢@Avision 告诉我如何查看 Postgres 使用的当前日期格式。

我更改了它，现在它工作正常:)

```
<input type="text" data-date-format="mm-dd-yyyy" >
```

Sed 详解：https://blog.csdn.net/h4241778/article/details/125263518



利用sed命令去掉汉字字符(以orders.csv为例)

```
LANG=C sed -r -i "s/[\x81-\xFE][\x40-\xFE]//g" orders.csv
```

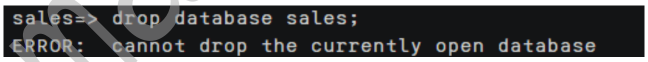
去掉空格

```
sed -r -i "s/[ ]//g" orders.csv
```

修改日期格式为dd-mm-yyyy

```
sed -r -i "s/([0-9]*)-([0-9]*)-([0-9]*)/\1-\2-20\3/g" orders.csv
```

6.出现错误：



```
sales=> drop database sales;
ERROR:  cannot drop the currently open database
```

原因：未退出，用 postgres 即可