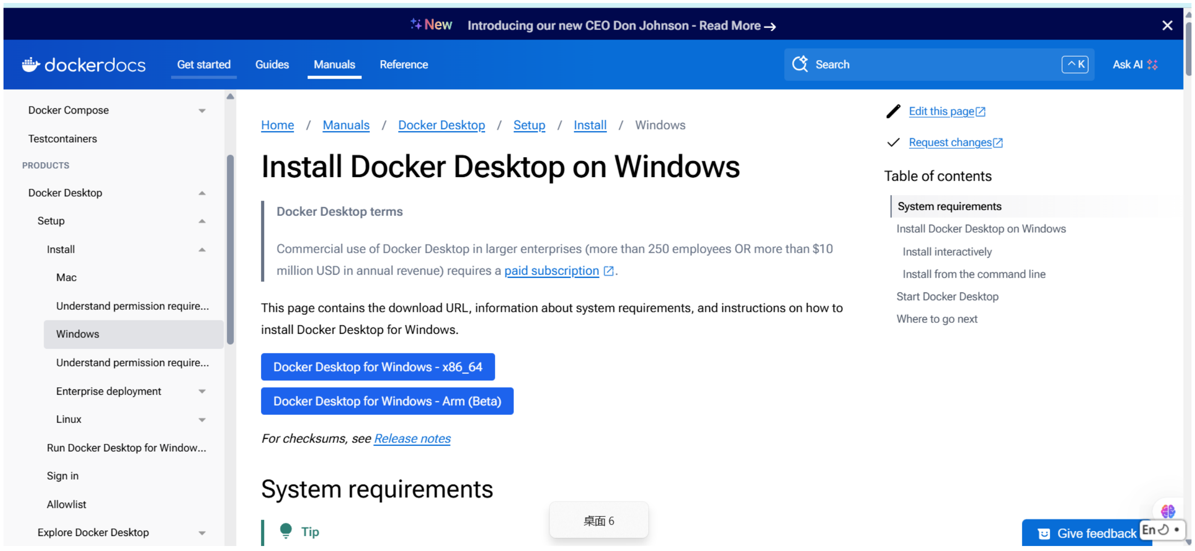
配置OceanBase数据库环境的方法

下载docker

在这个网址：https://docs.docker.com/desktop/setup/install/windows-install/里面有windows版本的docker下载链接：



Home / Manuals / Docker Desktop / Setup / Install / Windows

Install Docker Desktop on Windows

Docker Desktop terms

Commercial use of Docker Desktop in larger enterprises (more than 250 employees OR more than $10 million USD in annual revenue) requires a paid subscription.

This page contains the download URL, information about system requirements, and instructions on how to install Docker Desktop for Windows.

Docker Desktop for Windows - x86_64

Docker Desktop for Windows - Arm (Beta)

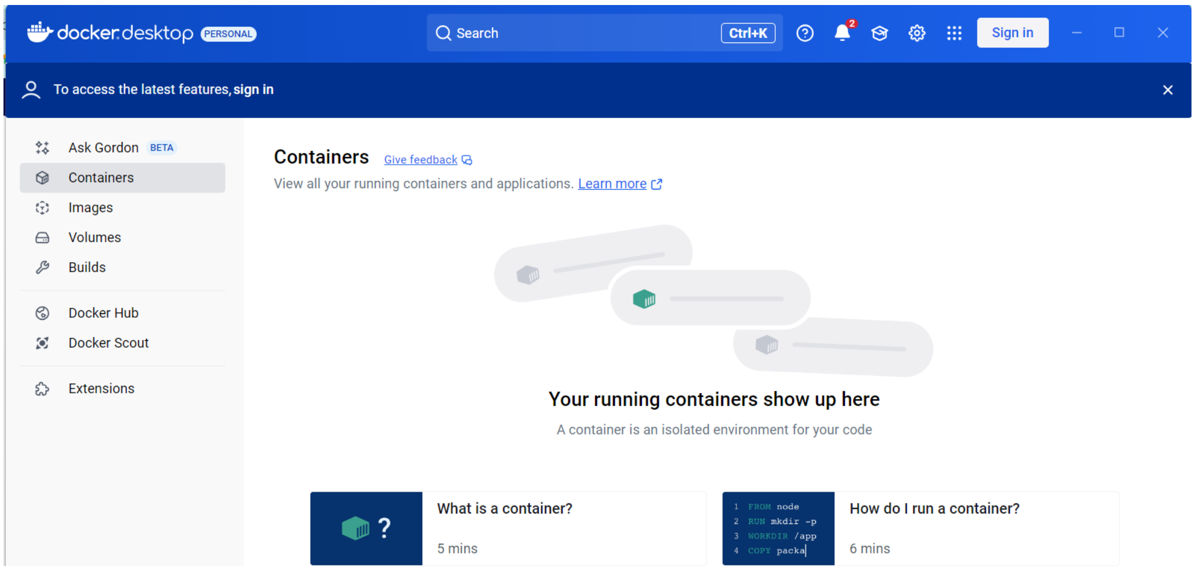
For checksums, see Release notes

System requirements

同级目录下也有mac和linux下载的链接。

docker配置

网页登录



To access the latest features, sign in

Containers

View all your running containers and applications. Learn more

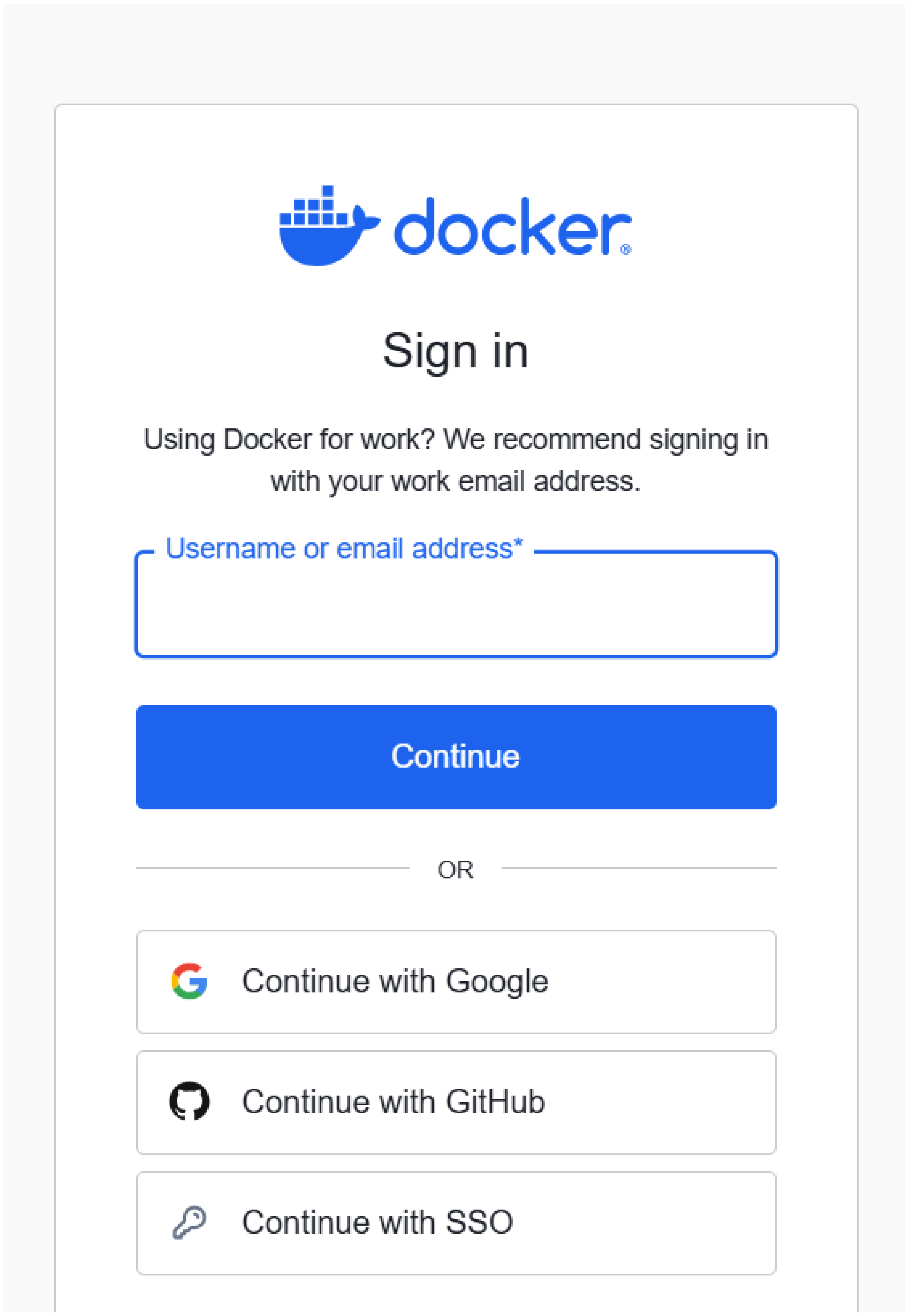
Your running containers show up here

A container is an isolated environment for your code

What is a container? 5 mins

How do I run a container? 6 mins

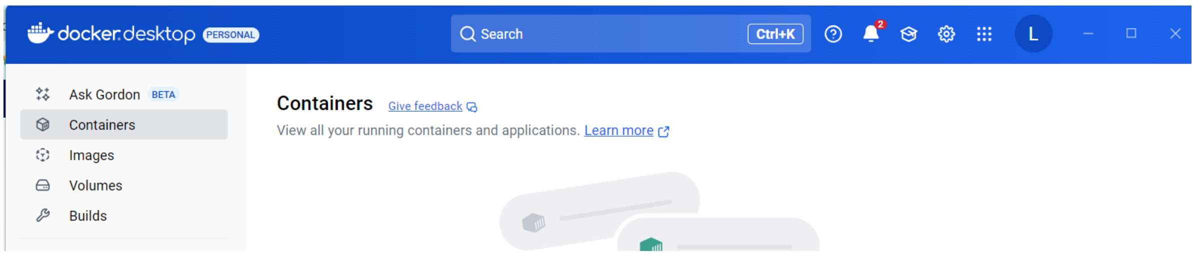
点击docker desktop右上角的sign in登录。


Sign in

Using Docker for work? We recommend signing in with your work email address.

Username or email address*

这里推荐大家选github来登录。登录后网页会弹出提示页，点击后回到docker desktop，此时应该已经完成登录，如下图：

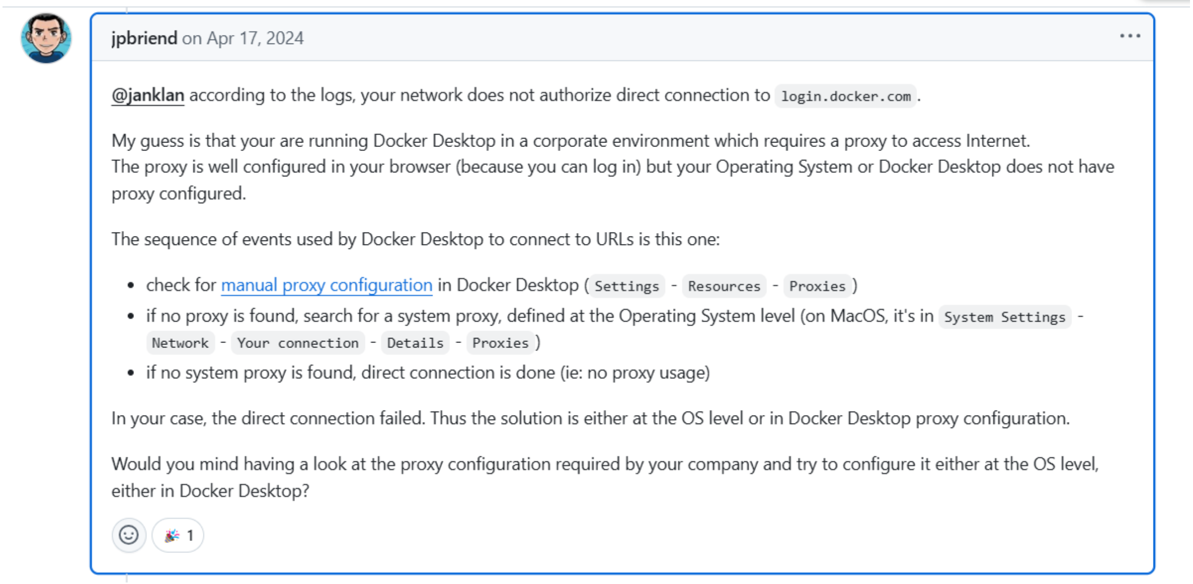


Containers

View all your running containers and applications.

（选用）如果网页登录不上去

如果网页登录不上去，先看看是不是代理设置问题，可以参考如下链接：https://blog.csdn.net/Vikanill/article/details/139420782，或者参考下面这张图的解决办法。



jpbriend on Apr 17, 2024

@janklan according to the logs, your network does not authorize direct connection to `login.docker.com`.

My guess is that your are running Docker Desktop in a corporate environment which requires a proxy to access Internet.
The proxy is well configured in your browser (because you can log in) but your Operating System or Docker Desktop does not have proxy configured.

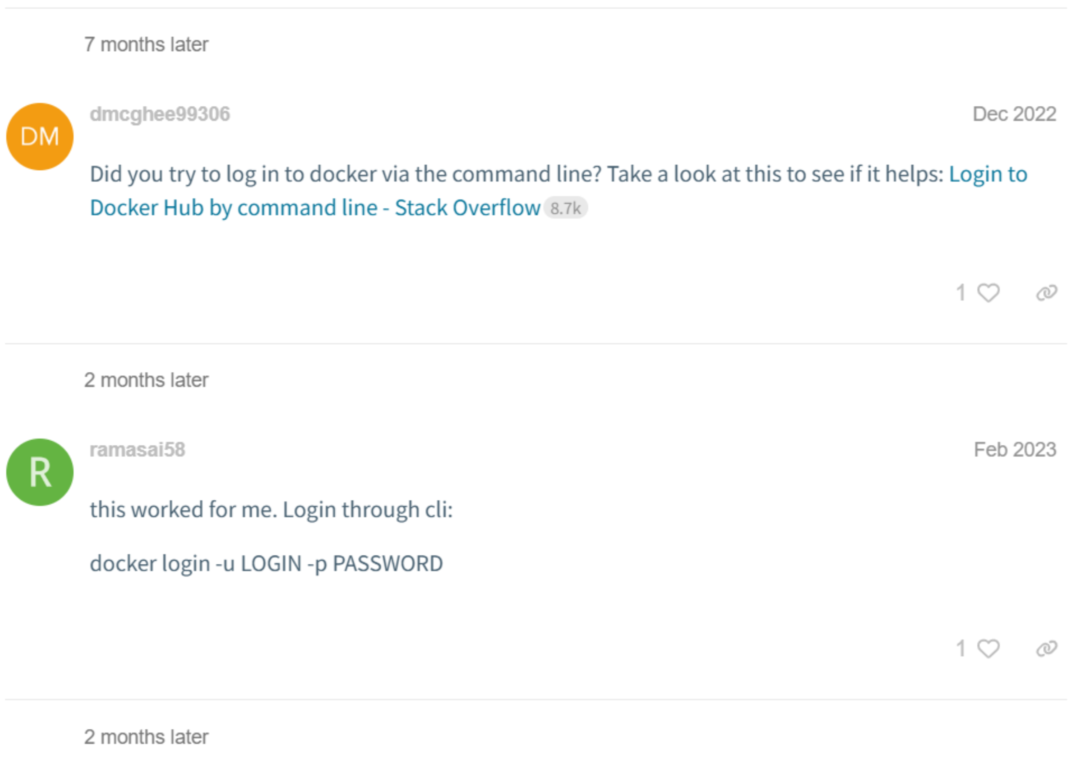
The sequence of events used by Docker Desktop to connect to URLs is this one:

- check for manual proxy configuration in Docker Desktop (`Settings` - `Resources` - `Proxies`)
- if no proxy is found, search for a system proxy, defined at the Operating System level (on MacOS, it's in `System Settings` - `Network` - `Your connection` - `Details` - `Proxies`)
- if no system proxy is found, direct connection is done (ie: no proxy usage)

In your case, the direct connection failed. Thus the solution is either at the OS level or in Docker Desktop proxy configuration.

Would you mind having a look at the proxy configuration required by your company and try to configure it either at the OS level, either in Docker Desktop?

如果还是不行，可以考虑从命令行登录，参考：https://stackoverflow.com/questions/57108005/how-to-login-to-docker-hub-on-the-command-line，登录方法如下图：



7 months later

dmcghee99306 Dec 2022

Did you try to log in to docker via the command line? Take a look at this to see if it helps: Login to Docker Hub by command line - Stack Overflow 8.7k

2 months later

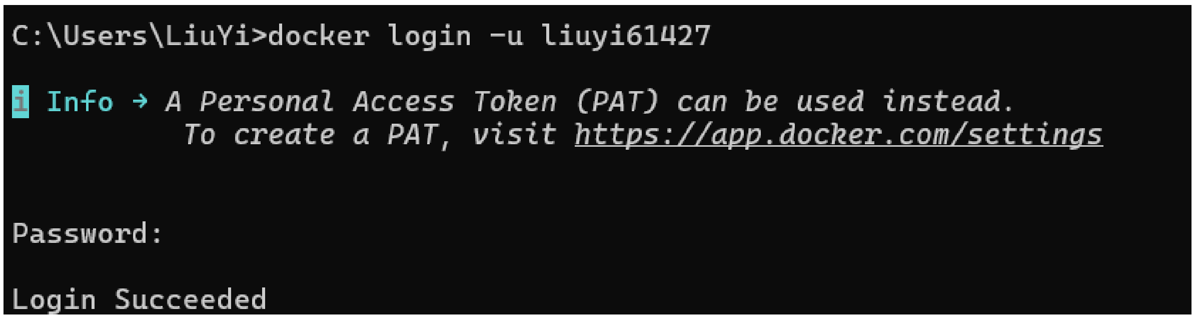
ramasai58 Feb 2023

this worked for me. Login through cli:

docker login -u LOGIN -p PASSWORD

2 months later

如下是命令行登录成功的结果：

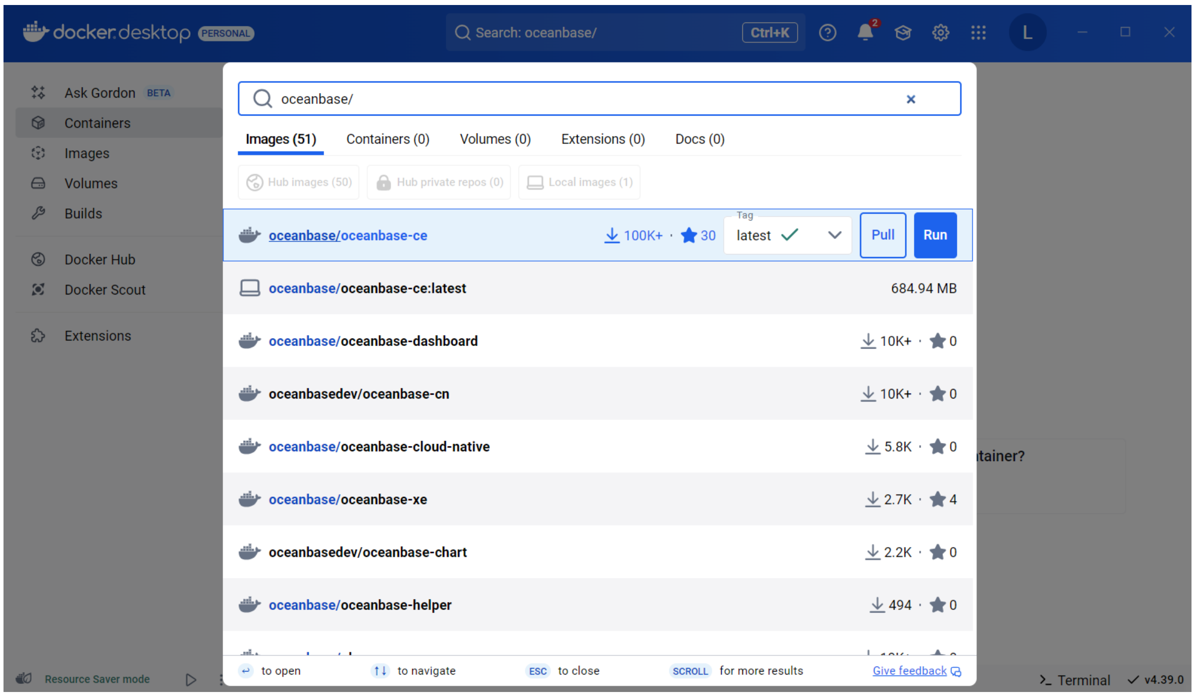


```
C:\Users\LiuYi>docker login -u liuyi61427

i Info → A Personal Access Token (PAT) can be used instead.
         To create a PAT, visit https://app.docker.com/settings


Password:

Login Succeeded
```

如果还有什么登录问题，请把你的问题报告给我们，我们再想办法。

拉取镜像

docker desktop拉取



oceanbase/oceanbase-ce 100K+ · 30
oceanbase/oceanbase-ce:latest 684.94 MB
oceanbase/oceanbase-dashboard 10K+ · 0
oceanbasedev/oceanbase-cn 10K+ · 0
oceanbase/oceanbase-cloud-native 5.8K · 0
oceanbase/oceanbase-xe 2.7K · 4
oceanbasedev/oceanbase-chart 2.2K · 0
oceanbase/oceanbase-helper 494 · 0

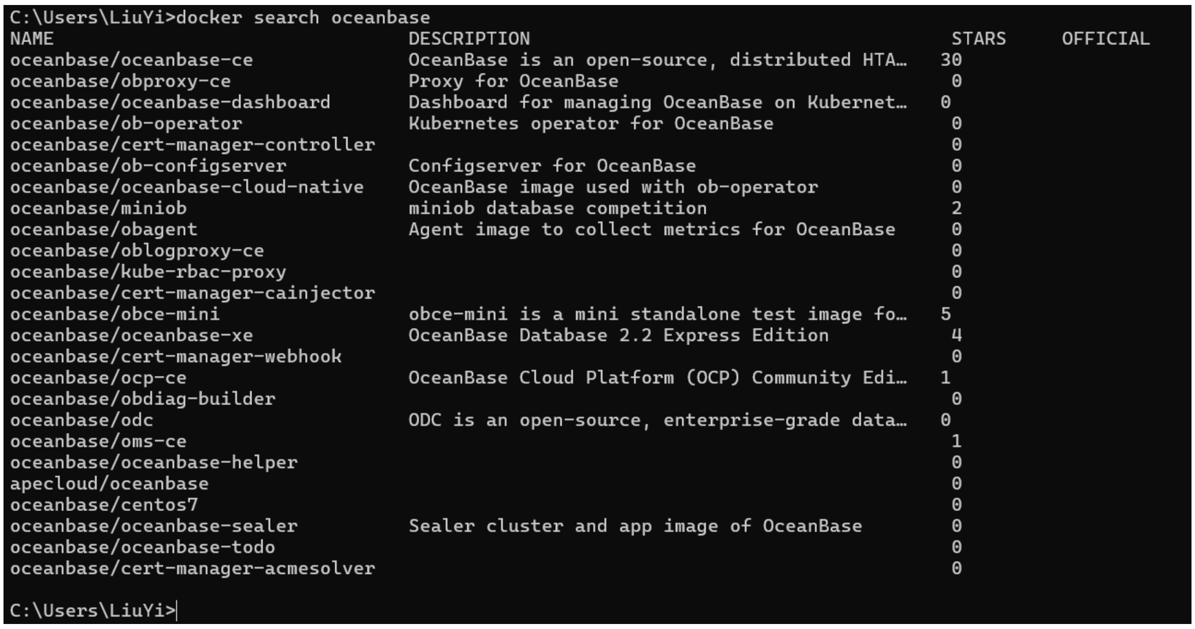
搜索oceanbase-ce，然后右侧点击Pull即可拉取。如果你拉取不下来，有可能是你第一步使用了邮箱来登录docker。

如果出现其他问题，还请你将问题报告给我们，我们会想办法解决。

命令行拉取

先搜索一下oceanbase的镜像

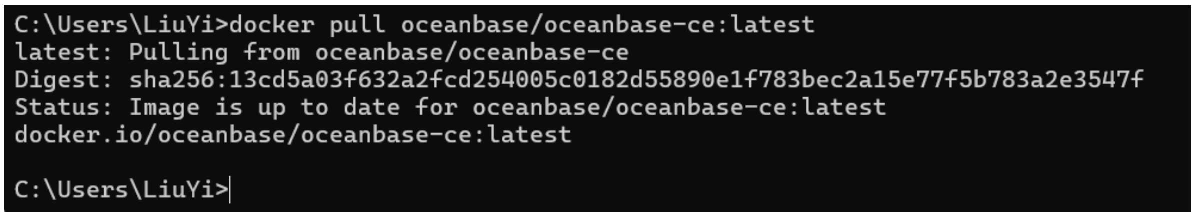
```
docker search oceanbase
```



```
C:\Users\LiuYi>docker search oceanbase
NAME                                   DESCRIPTION                                     STARS     OFFICIAL
oceanbase/oceanbase-ce                 OceanBase is an open-source, distributed HTA…   30
oceanbase/obproxy-ce                   Proxy for OceanBase                              0
oceanbase/oceanbase-dashboard          Dashboard for managing OceanBase on Kubernet…   0
oceanbase/ob-operator                  Kubernetes operator for OceanBase                0
oceanbase/cert-manager-controller                                                       0
oceanbase/ob-configserver              Configserver for OceanBase                       0
oceanbase/oceanbase-cloud-native       OceanBase image used with ob-operator            0
oceanbase/miniob                       miniob database competition                      2
oceanbase/obagent                      Agent image to collect metrics for OceanBase     0
oceanbase/oblogproxy-ce                                                                 0
oceanbase/kube-rbac-proxy                                                               0
oceanbase/cert-manager-cainjector                                                       0
oceanbase/obce-mini                    obce-mini is a mini standalone test image fo…   5
oceanbase/oceanbase-xe                 OceanBase Database 2.2 Express Edition           4
oceanbase/cert-manager-webhook                                                          0
oceanbase/ocp-ce                       OceanBase Cloud Platform (OCP) Community Edi…   1
oceanbase/obdiag-builder                                                                0
oceanbase/odc                          ODC is an open-source, enterprise-grade data…   0
oceanbase/oms-ce                                                                        1
oceanbase/oceanbase-helper                                                              0
apecloud/oceanbase                                                                      0
oceanbase/centos7                                                                       0
oceanbase/oceanbase-sealer             Sealer cluster and app image of OceanBase        0
oceanbase/oceanbase-todo                                                                0
oceanbase/cert-manager-acmesolver                                                       0

C:\Users\LiuYi>
```

使用如下方法拉取镜像：

```
docker pull oceanbase/oceanbase-ce:latest
```

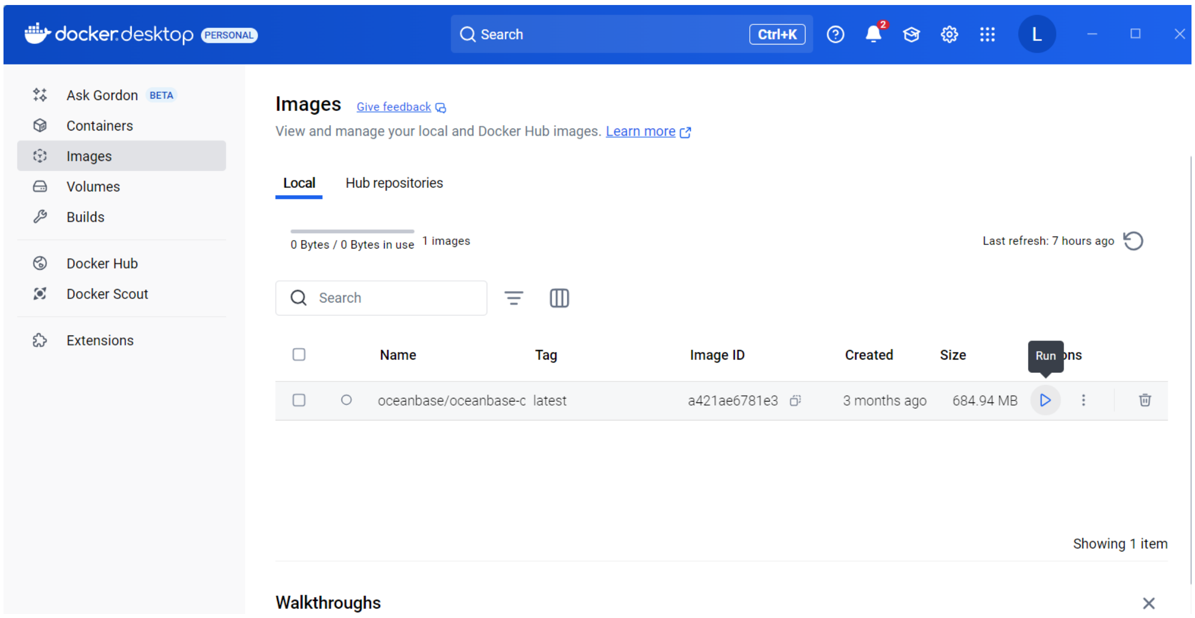


```
C:\Users\LiuYi>docker pull oceanbase/oceanbase-ce:latest
latest: Pulling from oceanbase/oceanbase-ce
Digest: sha256:13cd5a03f632a2fcd254005c0182d55890e1f783bec2a15e77f5b783a2e3547f
Status: Image is up to date for oceanbase/oceanbase-ce:latest
docker.io/oceanbase/oceanbase-ce:latest

C:\Users\LiuYi>
```

运行环境

docker desktop运行

点击Images也里面镜像的run按钮：



View and manage your local and Docker Hub images. Learn more

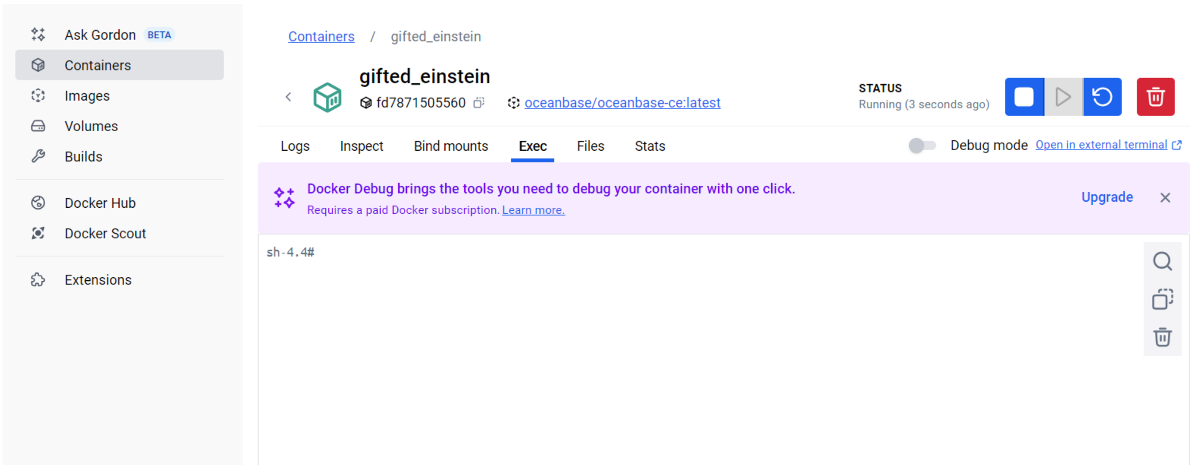
0 Bytes / 0 Bytes in use 1 images

Last refresh: 7 hours ago

Name	Tag	Image ID	Created	Size	Actions
oceanbase/oceanbase-c	latest	a421ae6781e3	3 months ago	684.94 MB	

Showing 1 item

然后再Exec栏就可以看到运行中环境（后续操作在这里面进行）。



Docker Debug brings the tools you need to debug your container with one click.
Requires a paid Docker subscription. Learn more.

```
sh-4.4#
```

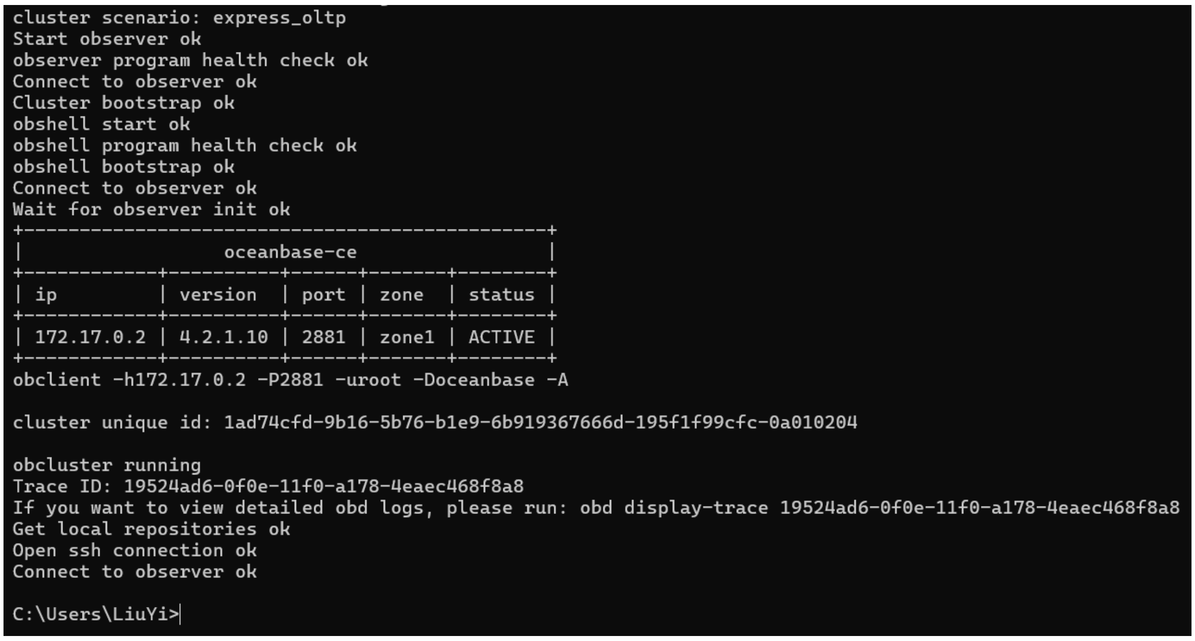
命令行运行

使用如下命令运行镜像：

```
docker run -p 2881:2881 --name obstandalone -e MODE=MINI -e
OB_TENANT_PASSWORD=123456 -d oceanbase/oceanbase-ce
```

使用如下命令查看镜像状态：

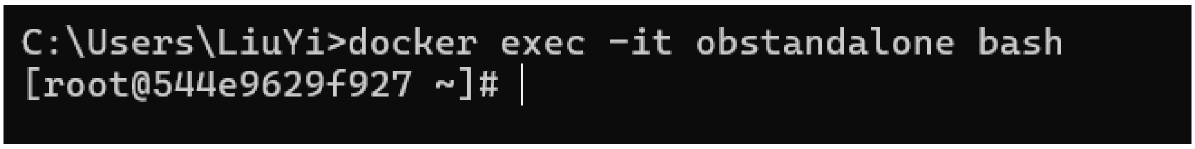
```
docker logs obstandalone
```



```
cluster scenario: express_oltp
Start observer ok
observer program health check ok
Connect to observer ok
Cluster bootstrap ok
obshell start ok
obshell program health check ok
obshell bootstrap ok
Connect to observer ok
Wait for observer init ok
+---------------------------------------------+
|                 oceanbase-ce                |
+------------+---------+------+-------+--------+
| ip         | version | port | zone  | status |
+------------+---------+------+-------+--------+
| 172.17.0.2 | 4.2.1.10 | 2881 | zone1 | ACTIVE |
+------------+---------+------+-------+--------+
obclient -h172.17.0.2 -P2881 -uroot -Doceanbase -A

cluster unique id: 1ad74cfd-9b16-5b76-b1e9-6b919367666d-195f1f99cfc-0a010204

obcluster running
Trace ID: 19524ad6-0f0e-11f0-a178-4eaec468f8a8
If you want to view detailed obd logs, please run: obd display-trace 19524ad6-0f0e-11f0-a178-4eaec468f8a8
Get local repositories ok
Open ssh connection ok
Connect to observer ok

C:\Users\LiuYi>
```

进行容器环境：

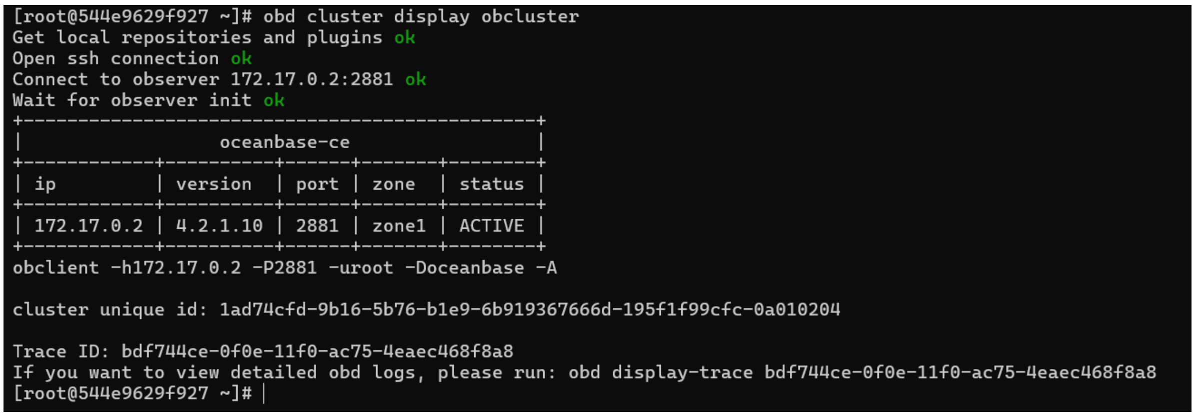
```
docker exec -it obstandalone bash
```



```
C:\Users\LiuYi>docker exec -it obstandalone bash
[root@544e9629f927 ~]#
```

查看oceanbase配置：

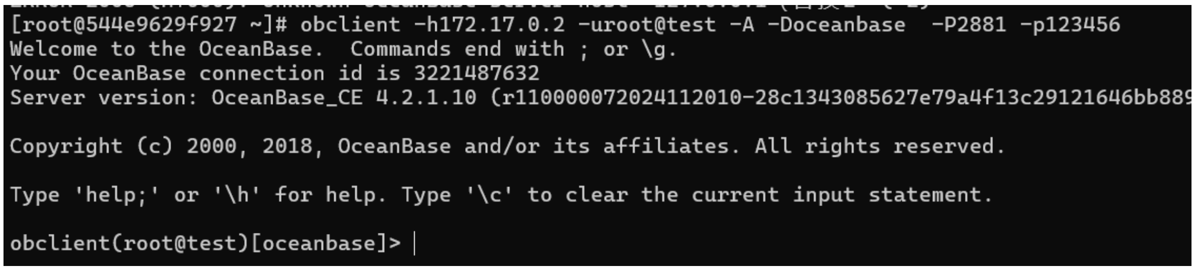
```
obd cluster display obcluster
```



```
[root@544e9629f927 ~]# obd cluster display obcluster
Get local repositories and plugins ok
Open ssh connection ok
Connect to observer 172.17.0.2:2881 ok
Wait for observer init ok
+---------------------------------------------+
|                 oceanbase-ce                |
+------------+---------+------+-------+--------+
| ip         | version | port | zone  | status |
+------------+---------+------+-------+--------+
| 172.17.0.2 | 4.2.1.10 | 2881 | zone1 | ACTIVE |
+------------+---------+------+-------+--------+
obclient -h172.17.0.2 -P2881 -uroot -Doceanbase -A

cluster unique id: 1ad74cfd-9b16-5b76-b1e9-6b919367666d-195f1f99cfc-0a010204

Trace ID: bdf744ce-0f0e-11f0-ac75-4eaec468f8a8
If you want to view detailed obd logs, please run: obd display-trace bdf744ce-0f0e-11f0-ac75-4eaec468f8a8
[root@544e9629f927 ~]#
```

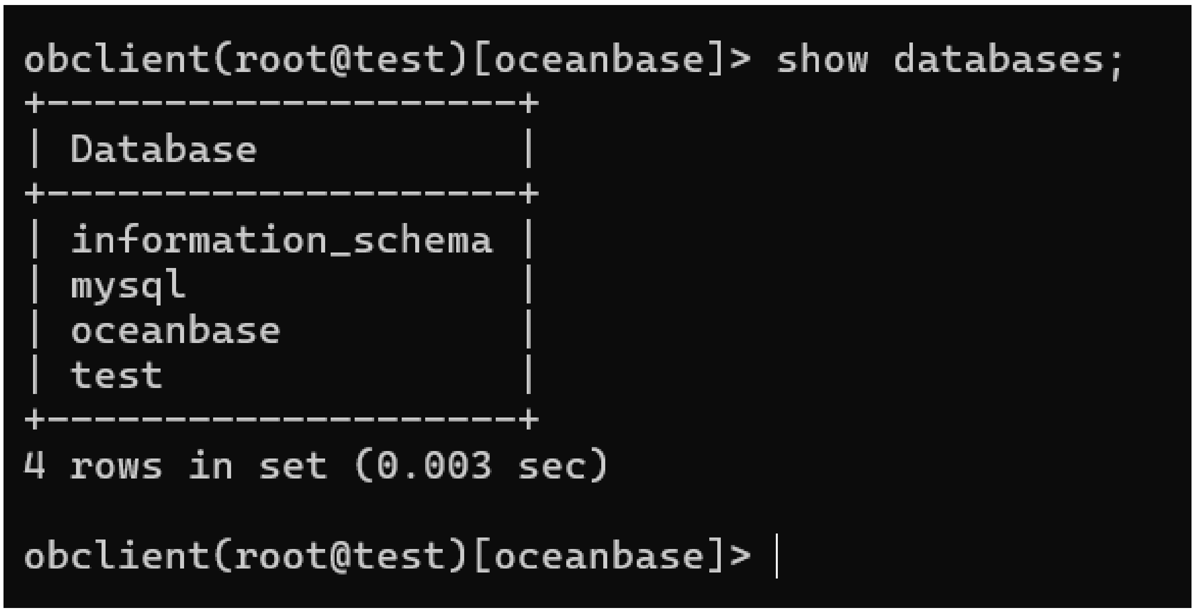
obclient连接：

```
obclient -h127.0.0.1（替换为如上的地址） -uroot@test -A -Doceanbase -P2881 -p123456
```


```
[root@544e9629f927 ~]# obclient -h172.17.0.2 -uroot@test -A -Doceanbase  -P2881 -p123456
Welcome to the OceanBase.  Commands end with ; or \g.
Your OceanBase connection id is 3221487632
Server version: OceanBase_CE 4.2.1.10 (r110000072024112010-28c1343085627e79a4f13c29121646bb889

Copyright (c) 2000, 2018, OceanBase and/or its affiliates. All rights reserved.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

obclient(root@test)[oceanbase]>
```

用如下命令查看数据库：

```
show databases;
```

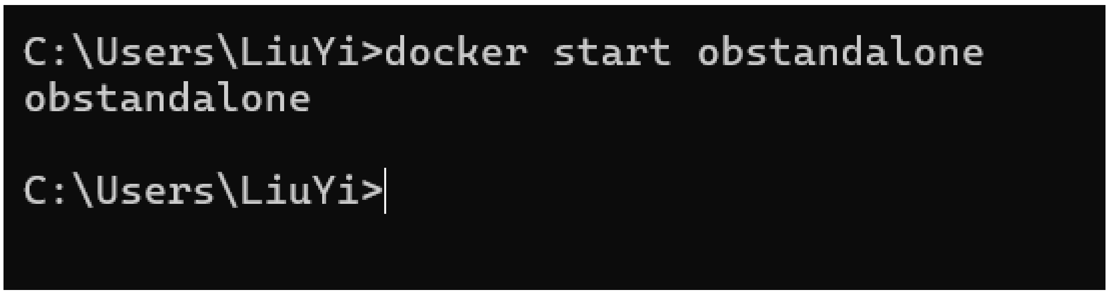

```
obclient(root@test)[oceanbase]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| oceanbase          |
| test               |
+--------------------+
4 rows in set (0.003 sec)

obclient(root@test)[oceanbase]>
```

DDL语言实验

进入容器

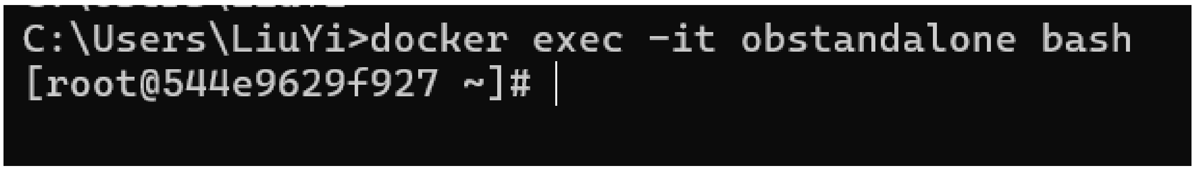
如果做完上面的配置实验把容器关了的话，使用如下命令用docker启动容器：

```
docker start obstandalone
```


```
C:\Users\LiuYi>docker start obstandalone
obstandalone

C:\Users\LiuYi>
```

然后进入容器，使用如下命令：

```
docker exec -it obstandalone bash
```


```
C:\Users\LiuYi>docker exec -it obstandalone bash
[root@544e9629f927 ~]#
```

表结构设计

按照实验要求完成数据库中表的创建。

```sql
-- 创建Experiment1数据库
CREATE DATABASE IF NOT EXISTS Experiment1;
USE Experiment1;
-- 创建Customers表
CREATE TABLE Customers (
    CustomerID INT AUTO_INCREMENT PRIMARY KEY NOT NULL,
    Name VARCHAR(50) NOT NULL,
    Email VARCHAR(100) NOT NULL UNIQUE CHECK (Email REGEXP '^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,4}$'),
    Phone VARCHAR(20) CHECK (Phone REGEXP '^((13[0-9]|14[01456879]|15[0-35-9]|16[2567]|17[0-8]|18[0-9]|19[0-35-9])[0-9]{8})|((0[0-9]{2,3})-?([0-9]{7,8}))$'),
    RegistrationDate DATE
);

-- 创建Categories表（需先创建，因Products依赖它）
CREATE TABLE Categories (
    CategoryID INT AUTO_INCREMENT PRIMARY KEY NOT NULL,
    CategoryName VARCHAR(50) NOT NULL UNIQUE
);

-- 创建Products表
CREATE TABLE Products (
    ProductID INT AUTO_INCREMENT PRIMARY KEY NOT NULL,
    ProductName VARCHAR(100) NOT NULL,
    Price DECIMAL(10,2) NOT NULL CHECK (Price > 0),
    StockQuantity INT NOT NULL CHECK (StockQuantity >= 0),
    CategoryID INT NOT NULL,
    FOREIGN KEY (CategoryID) REFERENCES Categories(CategoryID)
);

-- 创建Orders表
CREATE TABLE Orders (
    OrderID INT AUTO_INCREMENT PRIMARY KEY NOT NULL,
    CustomerID INT NOT NULL,
    OrderDate DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
    TotalAmount DECIMAL(10,2) NOT NULL CHECK (TotalAmount > 0),
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
);

-- 创建OrderItems表
CREATE TABLE OrderItems (
    OrderItemID INT AUTO_INCREMENT PRIMARY KEY NOT NULL,
    OrderID INT NOT NULL,
    ProductID INT NOT NULL,
    Quantity INT NOT NULL CHECK (Quantity > 0),
    Subtotal DECIMAL(10,2) NOT NULL CHECK (Subtotal > 0),
    FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
);

-- 添加Customers表中默认的RegistrationDate的触发器
DELIMITER $$
```

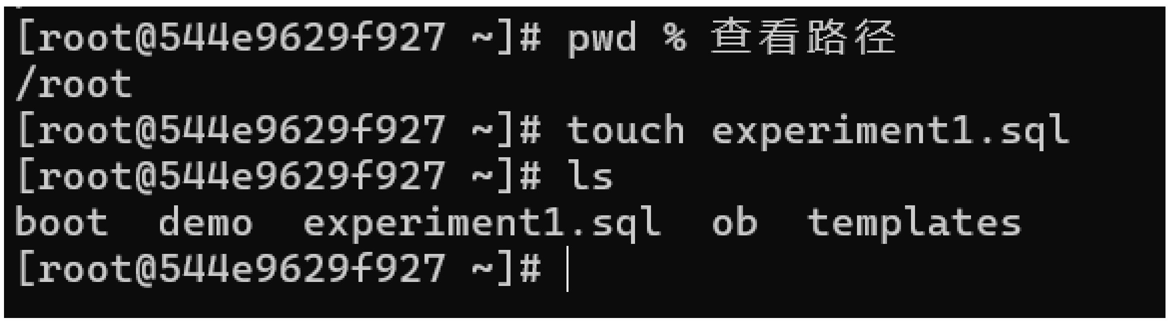
```
CREATE TRIGGER set_registration_date
BEFORE INSERT ON Customers
FOR EACH ROW
BEGIN
    IF NEW.RegistrationDate IS NULL THEN
        SET NEW.RegistrationDate = CURRENT_DATE();
    END IF;
END$$
DELIMITER ;

-- 添加计算Subtotal的触发器（确保与Price*Quantity一致）
DELIMITER $$
CREATE TRIGGER CalculateSubtotal
BEFORE INSERT ON OrderItems
FOR EACH ROW
BEGIN
    DECLARE product_price DECIMAL(10,2);
    SELECT Price INTO product_price FROM Products WHERE ProductID = 
NEW.ProductID;
    SET NEW.Subtotal = product_price * NEW.Quantity;
END$$
DELIMITER ;
```

关于上面的代码，有如下一些解释：

- 关于邮箱的匹配，这里写了一个案例，用的是正则表达式：`^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,4}$`，能够匹配绝大多数的邮箱。但其实只要写的合理即可，不要求强行覆盖所有的情况。
- 关于电话的匹配，这里也写了个案例，正则表达式是：`^((13[0-9]|14[01456879]|15[0-35-9]|16[2567]|17[0-8]|18[0-9]|19[0-35-9])[0-9]{8})|((0[0-9]{2,3})-?([0-9]{7,8}))$`，这里可以分开匹配手机号码和座机号码，第一部分匹配手机号码还考虑了电话区段的问题，第二部分座机号码还考虑了区号的问题。注意，这个正则表达式只是一个案例，也没办法覆盖所有情况，例如如果座机号码区号加上括号就没办法匹配。同样这里只要写的合理即可，不要求强行覆盖所有的情况。
- （不确定的问题）按现在的测试结果看，Oceanbase的正则表达式约束里面写`\d`好像会碰到一些问题，尽量改写为`[0-9]`。
- 最后实现了一段触发器，他在向`OrderItems`表中插入数据的时候被触发，将新插入数据的`Subtotal`栏设置为`product_price`和`Quantity`的乘积，`product_price`是通过`ProductID`在`Products`表中查询得到的。

使用linux命令行命令创建一个sql文件，将上面的指令写入文件中：

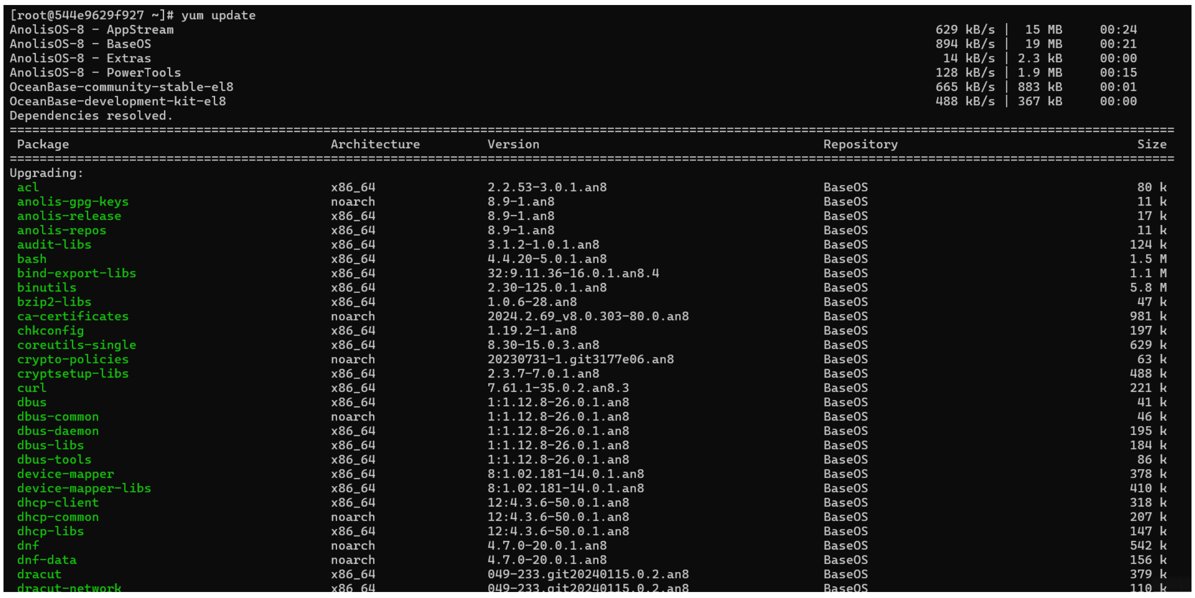
```
pwd % 使用该命令看所在的路径
touch experiment1.sql
ls % 查看结果
```



```
[root@544e9629f927 ~]# pwd % 查看路径
/root
[root@544e9629f927 ~]# touch experiment1.sql
[root@544e9629f927 ~]# ls
boot   demo   experiment1.sql   ob   templates
[root@544e9629f927 ~]# 
```

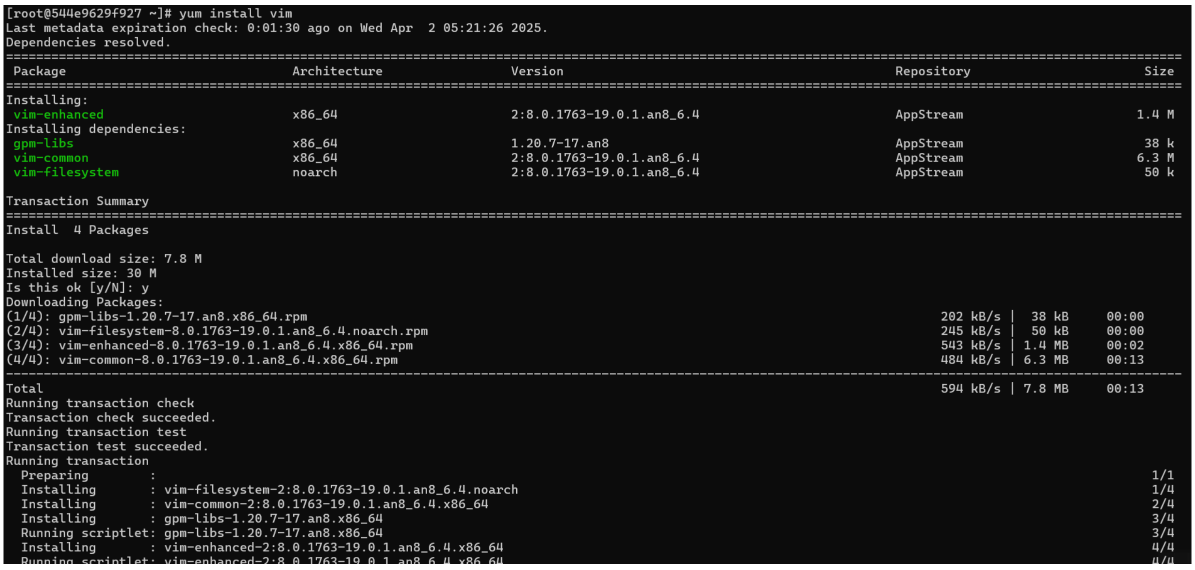
然后可以向创建的sql文件中用vim写入命令，先下载vim：

```
yum update
```



```
[root@544e9629f927 ~]# yum update
AnolisOS-8 - AppStream                                                          629 kB/s |  15 MB     00:24
AnolisOS-8 - BaseOS                                                             894 kB/s |  19 MB     00:21
AnolisOS-8 - Extras                                                              14 kB/s | 2.3 kB     00:00
AnolisOS-8 - PowerTools                                                         128 kB/s | 1.9 MB     00:15
OceanBase-community-stable-el8                                                  665 kB/s | 883 kB     00:01
OceanBase-development-kit-el8                                                   488 kB/s | 367 kB     00:00
Dependencies resolved.
==============================================================================================================================
 Package                                  Architecture        Version                                      Repository           Size
==============================================================================================================================
Upgrading:
 acl                                      x86_64              2.2.53-3.0.1.an8                             BaseOS               80 k
 anolis-gpg-keys                          noarch              8.9-1.an8                                    BaseOS               11 k
 anolis-release                           x86_64              8.9-1.an8                                    BaseOS               17 k
 anolis-repos                             x86_64              8.9-1.an8                                    BaseOS               11 k
 audit-libs                               x86_64              3.1.2-1.0.1.an8                              BaseOS              124 k
 bash                                     x86_64              4.4.20-5.0.1.an8                             BaseOS              1.5 M
 bind-export-libs                         x86_64              32:9.11.36-16.0.1.an8.4                      BaseOS              1.1 M
 binutils                                 x86_64              2.30-125.0.1.an8                             BaseOS              5.8 M
 bzip2-libs                               x86_64              1.0.6-28.an8                                 BaseOS               47 k
 ca-certificates                          noarch              2024.2.69_v8.0.303-80.0.an8                  BaseOS              981 k
 chkconfig                                x86_64              1.19.2-1.an8                                 BaseOS              197 k
 coreutils-single                         x86_64              8.30-15.0.3.an8                              BaseOS              629 k
 crypto-policies                          noarch              20230731-1.git3177e06.an8                    BaseOS               63 k
 cryptsetup-libs                          x86_64              2.3.7-7.0.1.an8                              BaseOS              488 k
 curl                                     x86_64              7.61.1-35.0.2.an8.3                          BaseOS              221 k
 dbus                                     x86_64              1:1.12.8-26.0.1.an8                          BaseOS               41 k
 dbus-common                              noarch              1:1.12.8-26.0.1.an8                          BaseOS               46 k
 dbus-daemon                              x86_64              1:1.12.8-26.0.1.an8                          BaseOS              195 k
 dbus-libs                                x86_64              1:1.12.8-26.0.1.an8                          BaseOS              184 k
 dbus-tools                               x86_64              1:1.12.8-26.0.1.an8                          BaseOS               86 k
 device-mapper                            x86_64              8:1.02.181-14.0.1.an8                        BaseOS              378 k
 device-mapper-libs                       x86_64              8:1.02.181-14.0.1.an8                        BaseOS              410 k
 dhcp-client                              x86_64              12:4.3.6-50.0.1.an8                          BaseOS              318 k
 dhcp-common                              noarch              12:4.3.6-50.0.1.an8                          BaseOS              207 k
 dhcp-libs                                x86_64              12:4.3.6-50.0.1.an8                          BaseOS              147 k
 dnf                                      noarch              4.7.0-20.0.1.an8                             BaseOS              542 k
 dnf-data                                 noarch              4.7.0-20.0.1.an8                             BaseOS              156 k
 dracut                                   x86_64              049-233.git20240115.0.2.an8                  BaseOS              379 k
 dracut-network                           x86_64              049-233.git20240115.0.2.an8                  BaseOS              110 k
```

```
yum install vim
```



```
[root@544e9629f927 ~]# yum install vim
Last metadata expiration check: 0:01:30 ago on Wed Apr  2 05:21:26 2025.
Dependencies resolved.
==============================================================================================================================
 Package                        Architecture          Version                                     Repository               Size
==============================================================================================================================
Installing:
 vim-enhanced                   x86_64                2:8.0.1763-19.0.1.an8_6.4                   AppStream               1.4 M
Installing dependencies:
 gpm-libs                       x86_64                1.20.7-17.an8                               AppStream                38 k
 vim-common                     x86_64                2:8.0.1763-19.0.1.an8_6.4                   AppStream               6.3 M
 vim-filesystem                 noarch                2:8.0.1763-19.0.1.an8_6.4                   AppStream                50 k

Transaction Summary
==============================================================================================================================
Install  4 Packages

Total download size: 7.8 M
Installed size: 30 M
Is this ok [y/N]: y
Downloading Packages:
(1/4): gpm-libs-1.20.7-17.an8.x86_64.rpm                                        202 kB/s |  38 kB     00:00
(2/4): vim-filesystem-8.0.1763-19.0.1.an8_6.4.noarch.rpm                        245 kB/s |  50 kB     00:00
(3/4): vim-enhanced-8.0.1763-19.0.1.an8_6.4.x86_64.rpm                          543 kB/s | 1.4 MB     00:02
(4/4): vim-common-8.0.1763-19.0.1.an8_6.4.x86_64.rpm                            484 kB/s | 6.3 MB     00:13
------------------------------------------------------------------------------------------------------------------------------
Total                                                                           594 kB/s | 7.8 MB     00:13
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                                                                       1/1
  Installing       : vim-filesystem-2:8.0.1763-19.0.1.an8_6.4.noarch                                                       1/4
  Installing       : vim-common-2:8.0.1763-19.0.1.an8_6.4.x86_64                                                           2/4
  Installing       : gpm-libs-1.20.7-17.an8.x86_64                                                                         3/4
  Running scriptlet: gpm-libs-1.20.7-17.an8.x86_64                                                                         3/4
  Installing       : vim-enhanced-2:8.0.1763-19.0.1.an8_6.4.x86_64                                                         4/4
  Running scriptlet: vim-enhanced-2:8.0.1763-19.0.1.an8_6.4.x86_64                                                         4/4
```

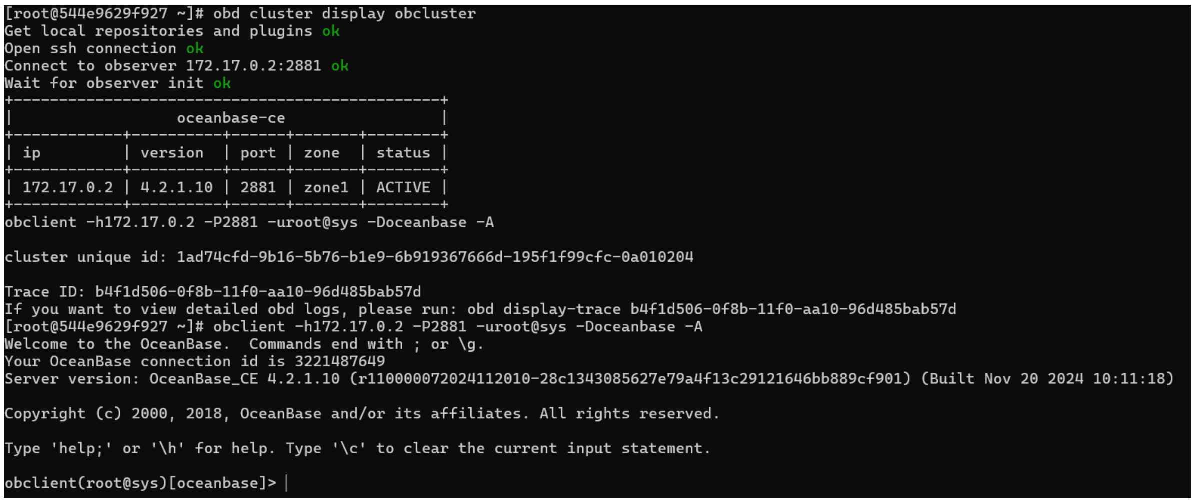
使用vim修改sql文件：

```
vim experiment1.sql
```

按`i`输入文本，将文本输入进去后，按`Esc`后再按`:wq!`写入并退出（注意冒号不要丢失）。

查看oceanbase配置信息，并按照配置信息启动oceanbase数据库：

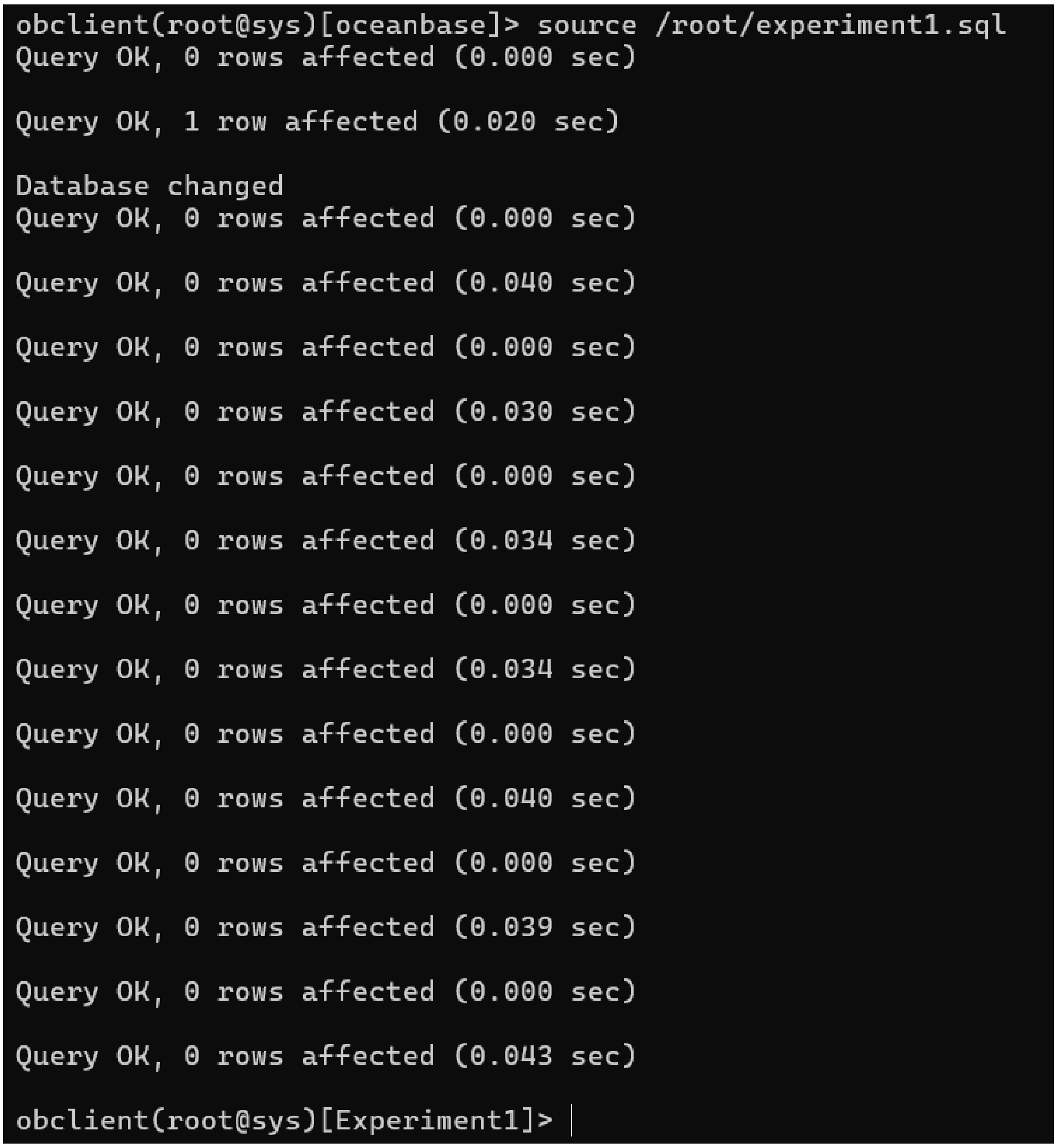
```
obd cluster display obcluster
obclient -h172.17.0.2 -P2881 -uroot@sys -Doceanbase -A % 根据你自己的配置信息修改这条命令
```



```
[root@544e9629f927 ~]# obd cluster display obcluster
Get local repositories and plugins ok
Open ssh connection ok
Connect to observer 172.17.0.2:2881 ok
Wait for observer init ok
+---------------------------------------------+
|                 oceanbase-ce                |
+------------+----------+------+-------+--------+
| ip         | version  | port | zone  | status |
+------------+----------+------+-------+--------+
| 172.17.0.2 | 4.2.1.10 | 2881 | zone1 | ACTIVE |
+------------+----------+------+-------+--------+
obclient -h172.17.0.2 -P2881 -uroot@sys -Doceanbase -A

cluster unique id: 1ad74cfd-9b16-5b76-b1e9-6b919367666d-195f1f99cfc-0a010204

Trace ID: b4f1d506-0f8b-11f0-aa10-96d485bab57d
If you want to view detailed obd logs, please run: obd display-trace b4f1d506-0f8b-11f0-aa10-96d485bab57d
[root@544e9629f927 ~]# obclient -h172.17.0.2 -P2881 -uroot@sys -Doceanbase -A
Welcome to the OceanBase.  Commands end with ; or \g.
Your OceanBase connection id is 3221487649
Server version: OceanBase_CE 4.2.1.10 (r110000072024112010-28c1343085627e79a4f13c29121646bb889cf901) (Built Nov 20 2024 10:11:18)

Copyright (c) 2000, 2018, OceanBase and/or its affiliates. All rights reserved.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

obclient(root@sys)[oceanbase]> 
```

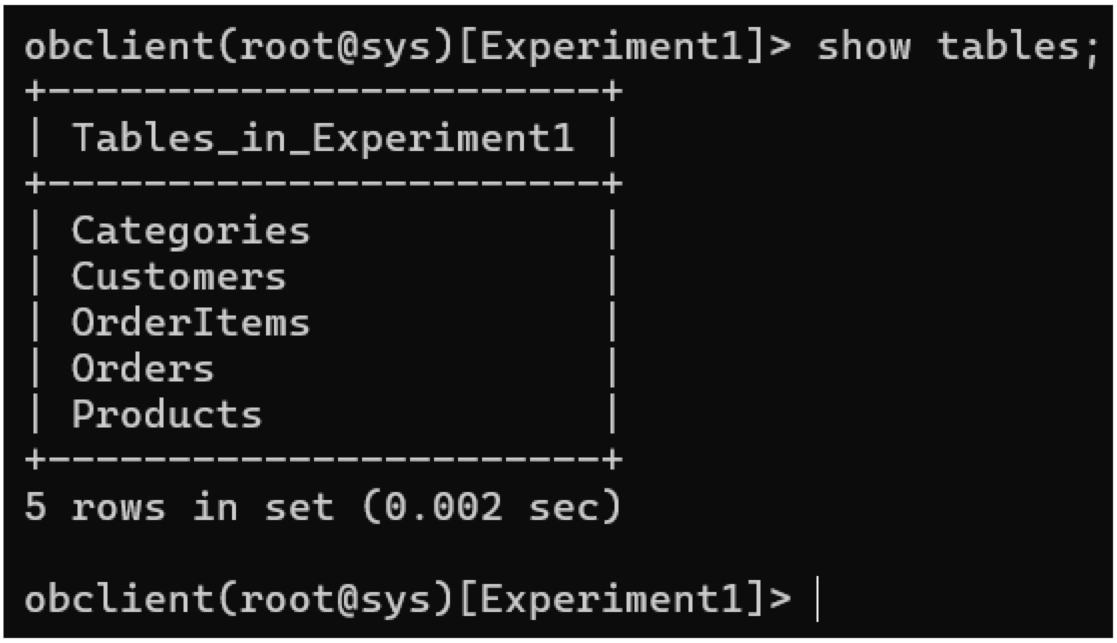
使用如下命令执行sql文件：

```
source /root/experiment1.sql % 注意你的sql文件的位置
```



```
obclient(root@sys)[oceanbase]> source /root/experiment1.sql
Query OK, 0 rows affected (0.000 sec)

Query OK, 1 row affected (0.020 sec)

Database changed
Query OK, 0 rows affected (0.000 sec)

Query OK, 0 rows affected (0.040 sec)

Query OK, 0 rows affected (0.000 sec)

Query OK, 0 rows affected (0.030 sec)

Query OK, 0 rows affected (0.000 sec)

Query OK, 0 rows affected (0.034 sec)

Query OK, 0 rows affected (0.000 sec)

Query OK, 0 rows affected (0.034 sec)

Query OK, 0 rows affected (0.000 sec)

Query OK, 0 rows affected (0.040 sec)

Query OK, 0 rows affected (0.000 sec)

Query OK, 0 rows affected (0.039 sec)

Query OK, 0 rows affected (0.000 sec)

Query OK, 0 rows affected (0.043 sec)

obclient(root@sys)[Experiment1]> 
```

最后结果如下：

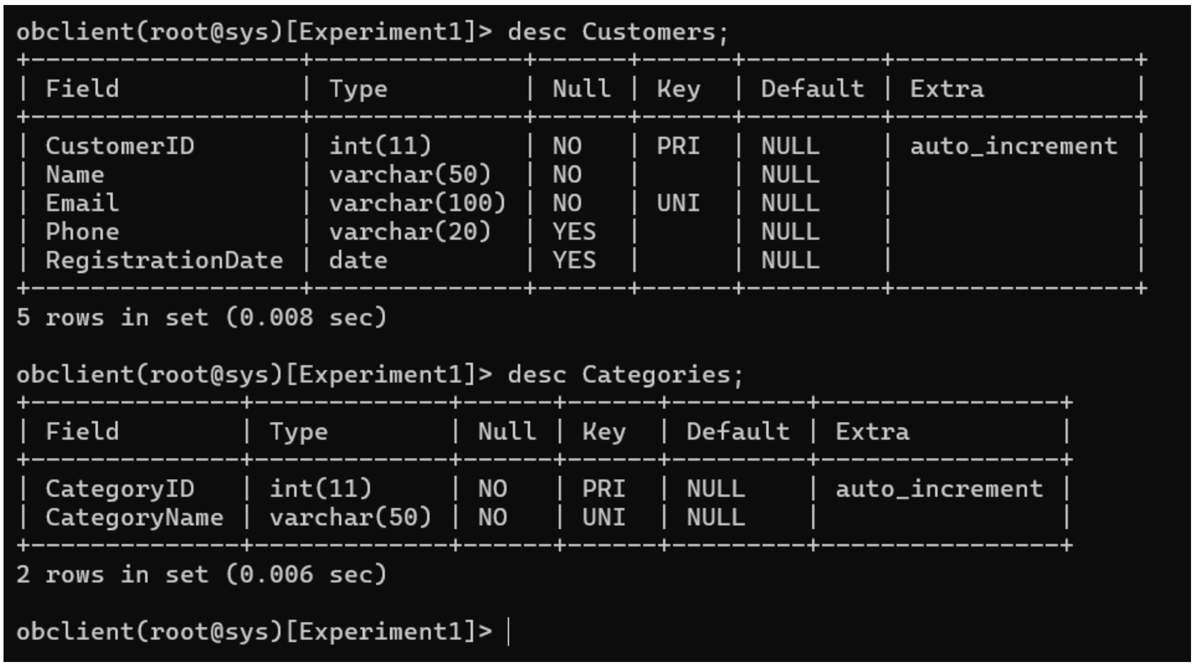


```
obclient(root@sys)[Experiment1]> show tables;
+------------------------+
| Tables_in_Experiment1  |
+------------------------+
| Categories             |
| Customers              |
| OrderItems             |
| Orders                 |
| Products               |
+------------------------+
5 rows in set (0.002 sec)

obclient(root@sys)[Experiment1]>
```

表结构查询

使用如下命令查看表的结构:

```
desc Customers; % 查询其他表的信息用一样的命令
```

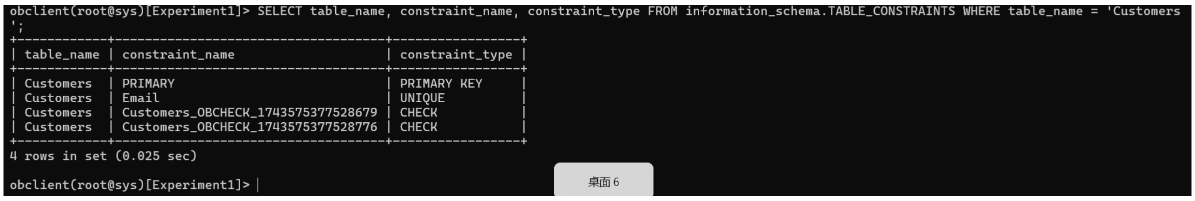


```
obclient(root@sys)[Experiment1]> desc Customers;
+------------------+--------------+------+-----+---------+----------------+
| Field            | Type         | Null | Key | Default | Extra          |
+------------------+--------------+------+-----+---------+----------------+
| CustomerID       | int(11)      | NO   | PRI | NULL    | auto_increment |
| Name             | varchar(50)  | NO   |     | NULL    |                |
| Email            | varchar(100) | NO   | UNI | NULL    |                |
| Phone            | varchar(20)  | YES  |     | NULL    |                |
| RegistrationDate | date         | YES  |     | NULL    |                |
+------------------+--------------+------+-----+---------+----------------+
5 rows in set (0.008 sec)

obclient(root@sys)[Experiment1]> desc Categories;
+--------------+-------------+------+-----+---------+----------------+
| Field        | Type        | Null | Key | Default | Extra          |
+--------------+-------------+------+-----+---------+----------------+
| CategoryID   | int(11)     | NO   | PRI | NULL    | auto_increment |
| CategoryName | varchar(50) | NO   | UNI | NULL    |                |
+--------------+-------------+------+-----+---------+----------------+
2 rows in set (0.006 sec)

obclient(root@sys)[Experiment1]>
```

表约束查询

使用如下命令查询建立的约束:

```
-- table_name可以改成其他的表名称
SELECT table_name, constraint_name, constraint_type FROM
information_schema.TABLE_CONSTRAINTS WHERE table_name='Customers';
```



```
obclient(root@sys)[Experiment1]> SELECT table_name, constraint_name, constraint_type FROM information_schema.TABLE_CONSTRAINTS WHERE table_name = 'Customers
';
+------------+------------------------------------+-----------------+
| table_name | constraint_name                    | constraint_type |
+------------+------------------------------------+-----------------+
| Customers  | PRIMARY                            | PRIMARY KEY     |
| Customers  | Email                              | UNIQUE          |
| Customers  | Customers_OBCHECK_1743575377528679 | CHECK           |
| Customers  | Customers_OBCHECK_1743575377528776 | CHECK           |
+------------+------------------------------------+-----------------+
4 rows in set (0.025 sec)

obclient(root@sys)[Experiment1]> |
```

表结构修改与操作

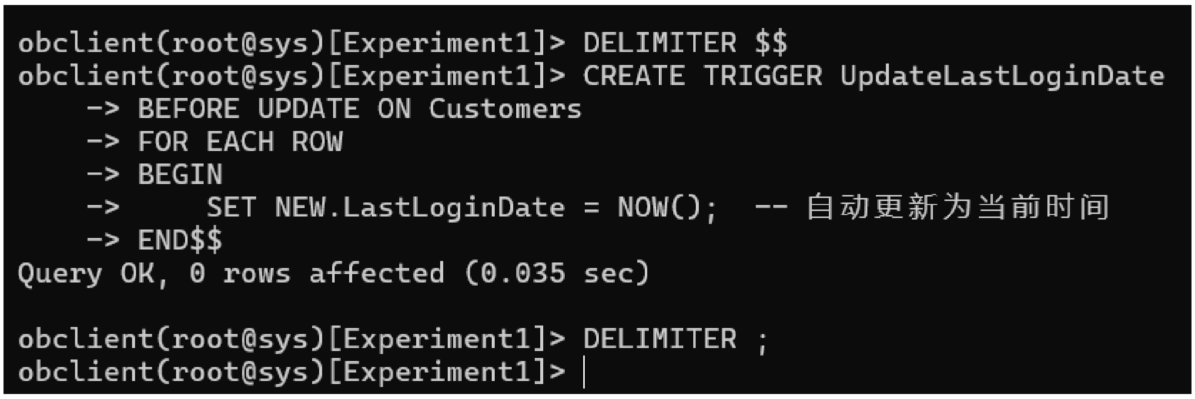
添加一个新字段 LastLoginDate

- 在 Customers 表中添加一个新字段 LastLoginDate，类型为 DATETIME，当客户每次登录时，该字段自动更新为当前时间。需创建触发器实现此功能。

使用的代码是：

```sql
ALTER TABLE Customers ADD COLUMN LastLoginDate DATETIME;
DELIMITER $$
CREATE TRIGGER UpdateLastLoginDate
BEFORE UPDATE ON Customers
FOR EACH ROW
BEGIN
    SET NEW.LastLoginDate = NOW();  -- 自动更新为当前时间
END$$
DELIMITER ;
```

运行效果如下：



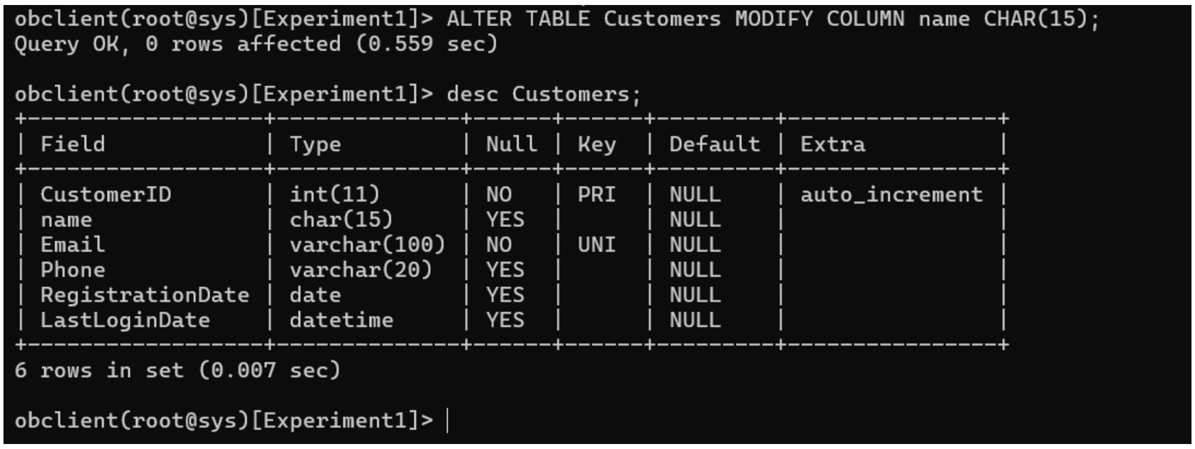
```
obclient(root@sys)[Experiment1]> DELIMITER $$
obclient(root@sys)[Experiment1]> CREATE TRIGGER UpdateLastLoginDate
    -> BEFORE UPDATE ON Customers
    -> FOR EACH ROW
    -> BEGIN
    ->     SET NEW.LastLoginDate = NOW();  -- 自动更新为当前时间
    -> END$$
Query OK, 0 rows affected (0.035 sec)

obclient(root@sys)[Experiment1]> DELIMITER ;
obclient(root@sys)[Experiment1]> |
```

改变name类型

- 将 Customers 中的 name 由 varchar 改为 char(15)。

使用如下命令：

```sql
ALTER TABLE Customers MODIFY COLUMN name CHAR(15);
```



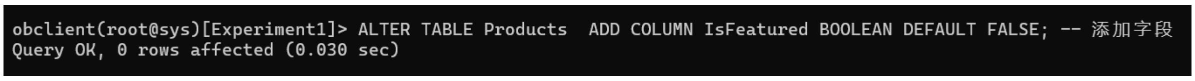
```
obclient(root@sys)[Experiment1]> ALTER TABLE Customers MODIFY COLUMN name CHAR(15);
Query OK, 0 rows affected (0.559 sec)

obclient(root@sys)[Experiment1]> desc Customers;
+------------------+--------------+------+-----+---------+----------------+
| Field            | Type         | Null | Key | Default | Extra          |
+------------------+--------------+------+-----+---------+----------------+
| CustomerID       | int(11)      | NO   | PRI | NULL    | auto_increment |
| name             | char(15)     | YES  |     | NULL    |                |
| Email            | varchar(100) | NO   | UNI | NULL    |                |
| Phone            | varchar(20)  | YES  |     | NULL    |                |
| RegistrationDate | date         | YES  |     | NULL    |                |
| LastLoginDate    | datetime     | YES  |     | NULL    |                |
+------------------+--------------+------+-----+---------+----------------+
6 rows in set (0.007 sec)

obclient(root@sys)[Experiment1]> |
```

添加字段和创建索引

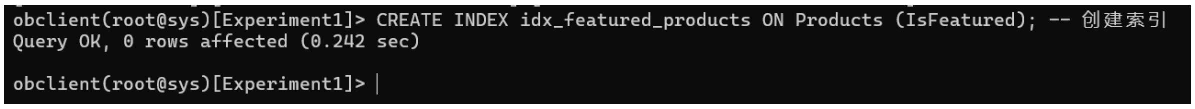
- 修改 Products 表，添加一个 IsFeatured 字段，类型为 BOOLEAN，默认值为 FALSE。同时，创建一个索引，用于快速查询特色商品（IsFeatured 为 TRUE 的商品）。

使用如下命令：

```sql
ALTER TABLE Products ADD COLUMN IsFeatured BOOLEAN DEFAULT FALSE; -- 添加字段
CREATE INDEX idx_featured_products ON Products (IsFeatured); -- 创建索引
```



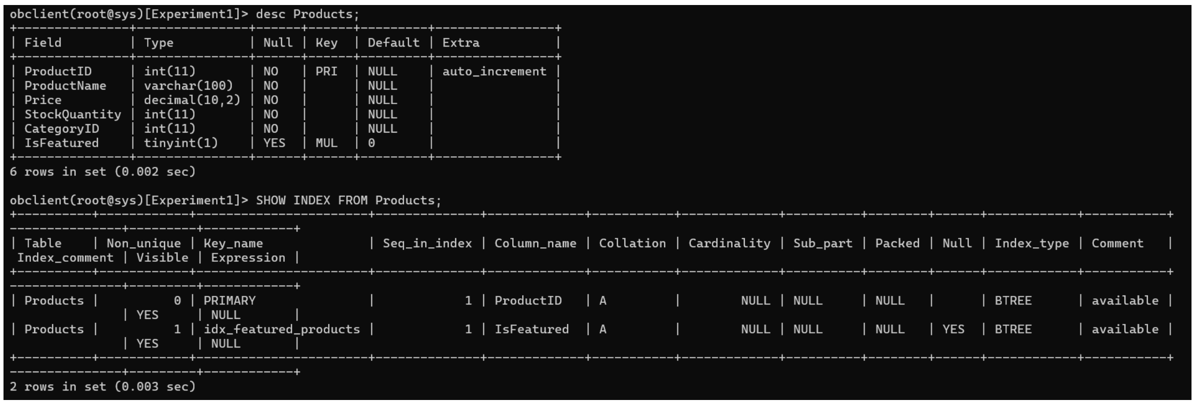
```
obclient(root@sys)[Experiment1]> ALTER TABLE Products  ADD COLUMN IsFeatured BOOLEAN DEFAULT FALSE; -- 添加字段
Query OK, 0 rows affected (0.030 sec)
```



```
obclient(root@sys)[Experiment1]> CREATE INDEX idx_featured_products ON Products (IsFeatured); -- 创建索引
Query OK, 0 rows affected (0.242 sec)

obclient(root@sys)[Experiment1]> |
```

查看数据表的情况：

```sql
desc Products;
SHOW INDEX FROM Products;
```



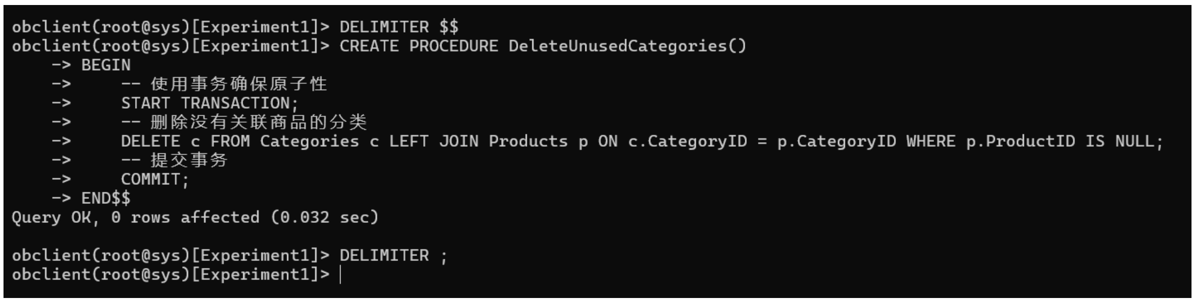
```
obclient(root@sys)[Experiment1]> desc Products;
+---------------+---------------+------+-----+---------+----------------+
| Field         | Type          | Null | Key | Default | Extra          |
+---------------+---------------+------+-----+---------+----------------+
| ProductID     | int(11)       | NO   | PRI | NULL    | auto_increment |
| ProductName   | varchar(100)  | NO   |     | NULL    |                |
| Price         | decimal(10,2) | NO   |     | NULL    |                |
| StockQuantity | int(11)       | NO   |     | NULL    |                |
| CategoryID    | int(11)       | NO   |     | NULL    |                |
| IsFeatured    | tinyint(1)    | YES  | MUL | 0       |                |
+---------------+---------------+------+-----+---------+----------------+
6 rows in set (0.002 sec)

obclient(root@sys)[Experiment1]> SHOW INDEX FROM Products;
+----------+------------+-----------------------+--------------+-------------+-----------+-------------+----------+--------+------+------------+-----------+
---------------+---------+------------+
| Table    | Non_unique | Key_name              | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment   |
 Index_comment | Visible | Expression |
+----------+------------+-----------------------+--------------+-------------+-----------+-------------+----------+--------+------+------------+-----------+
---------------+---------+------------+
| Products |          0 | PRIMARY               |            1 | ProductID   | A         |        NULL | NULL     | NULL   |      | BTREE      | available |
               | YES     | NULL       |
| Products |          1 | idx_featured_products |            1 | IsFeatured  | A         |        NULL | NULL     | NULL   | YES  | BTREE      | available |
               | YES     | NULL       |
+----------+------------+-----------------------+--------------+-------------+-----------+-------------+----------+--------+------+------------+-----------+
---------------+---------+------------+
2 rows in set (0.003 sec)
```

删除分类记录

- 删除 Categories 表中所有没有关联商品的分类记录，同时要保证删除操作符合数据库的参照完整性。需创建一个存储过程实现此功能。

使用的命令为：

```sql
DELIMITER $$
CREATE PROCEDURE DeleteUnusedCategories()
BEGIN
    -- 使用事务确保原子性
    START TRANSACTION;
    -- 删除没有关联商品的分类
    DELETE c FROM Categories c LEFT JOIN Products p ON c.CategoryID = p.CategoryID WHERE p.ProductID IS NULL;
    -- 提交事务
    COMMIT;
END$$
DELIMITER ;
```



```
obclient(root@sys)[Experiment1]> DELIMITER $$
obclient(root@sys)[Experiment1]> CREATE PROCEDURE DeleteUnusedCategories()
    -> BEGIN
    ->     -- 使用事务确保原子性
    ->     START TRANSACTION;
    ->     -- 删除没有关联商品的分类
    ->     DELETE c FROM Categories c LEFT JOIN Products p ON c.CategoryID = p.CategoryID WHERE p.ProductID IS NULL;
    ->     -- 提交事务
    ->     COMMIT;
    -> END$$
Query OK, 0 rows affected (0.032 sec)

obclient(root@sys)[Experiment1]> DELIMITER ;
obclient(root@sys)[Experiment1]> |
```

可以测试一下这个存储过程：

1. 插入测试数据

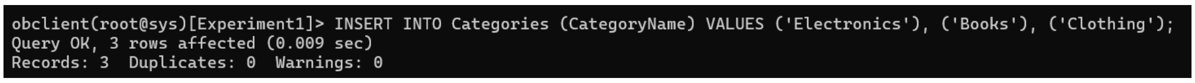
```sql
-- 插入测试分类
INSERT INTO Categories (CategoryName) VALUES ('Electronics'), ('Books'), ('Clothing');

-- 插入测试商品（仅关联 Electronics 和 Books）
INSERT INTO Products (ProductName, CategoryID, Price, StockQuantity) VALUES ('Laptop', 1, 100, 10), ('Smartphone', 1, 50, 20), ('Novel', 2, 20, 50);
```

2. 执行存储过程

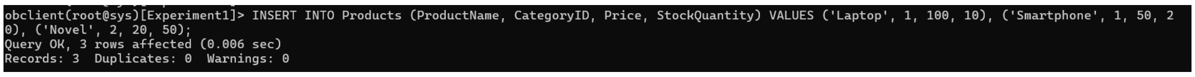
```sql
CALL DeleteUnusedCategories();
```

3. 检查结果

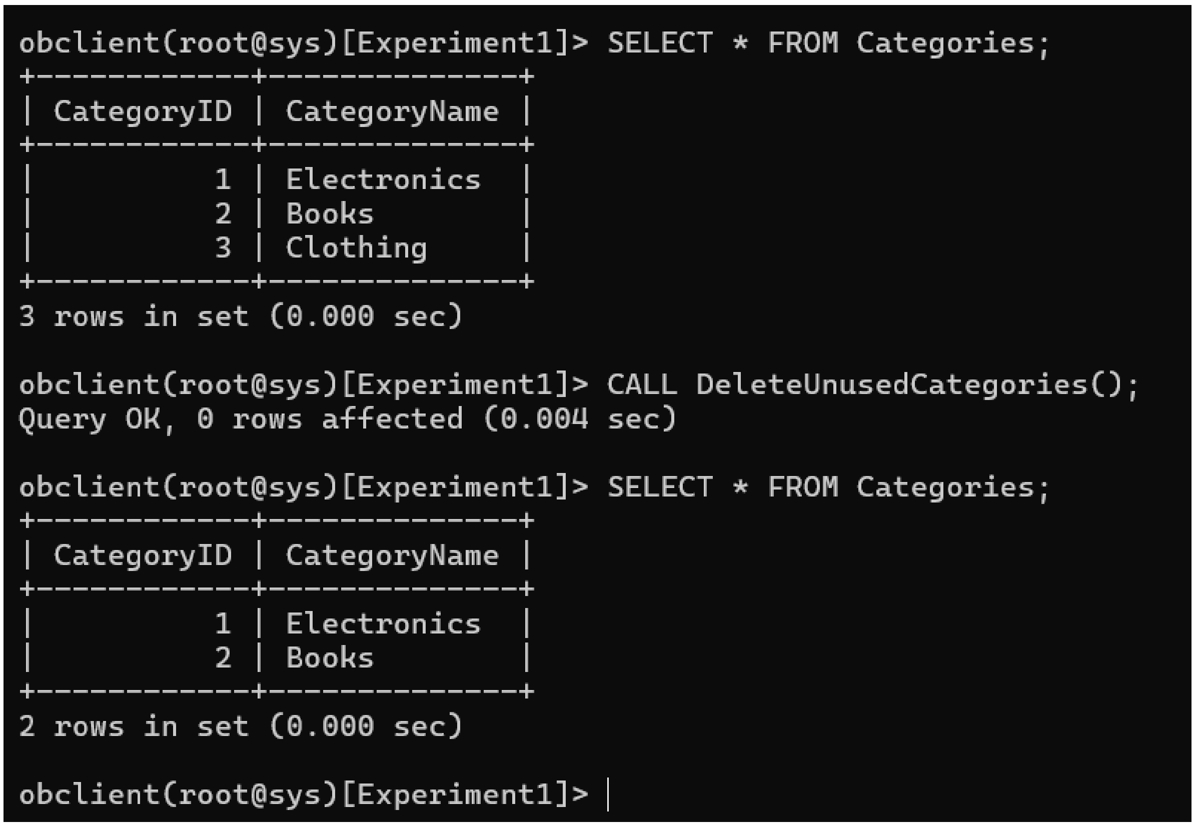
```sql
SELECT * FROM Categories;
```



```
obclient(root@sys)[Experiment1]> INSERT INTO Categories (CategoryName) VALUES ('Electronics'), ('Books'), ('Clothing');
Query OK, 3 rows affected (0.009 sec)
Records: 3  Duplicates: 0  Warnings: 0
```



```
obclient(root@sys)[Experiment1]> INSERT INTO Products (ProductName, CategoryID, Price, StockQuantity) VALUES ('Laptop', 1, 100, 10), ('Smartphone', 1, 50, 2
0), ('Novel', 2, 20, 50);
Query OK, 3 rows affected (0.006 sec)
Records: 3  Duplicates: 0  Warnings: 0
```



```
obclient(root@sys)[Experiment1]> SELECT * FROM Categories;
+------------+--------------+
| CategoryID | CategoryName |
+------------+--------------+
|          1 | Electronics  |
|          2 | Books        |
|          3 | Clothing     |
+------------+--------------+
3 rows in set (0.000 sec)

obclient(root@sys)[Experiment1]> CALL DeleteUnusedCategories();
Query OK, 0 rows affected (0.004 sec)

obclient(root@sys)[Experiment1]> SELECT * FROM Categories;
+------------+--------------+
| CategoryID | CategoryName |
+------------+--------------+
|          1 | Electronics  |
|          2 | Books        |
+------------+--------------+
2 rows in set (0.000 sec)

obclient(root@sys)[Experiment1]>
```

可见执行存储过程后Clouthing被移除了。