

# 《数据库系统及其应用实践》课程实验报告

## 实验 X: XXXXXXXXX

学号: \*\*\*\*\*

姓名: \*\*\*

完成日期: 2023 年 3 月 7 日

### 实验目标

学习 Docker 环境的原理和基本操作, 能够通过 Docker 容器启动 openGauss 数据库实例, 掌握连接和操作数据库的基本命令。

### 实验要求

- 1、按照实验内容, 依次完成每个实验步骤;
- 2、操作实验步骤时, 需要理解该操作步骤的目的, 预判操作的结果; 当操作结果与预判不符时, 及时向任课教师和助教咨询;
- 3、在实验报告中依次记录**主要操作步骤**的内容和结果(返回的消息或截图);
- 4、对实验中遇到的问题、解决方案及收获进行总结;
- 5、确保实验报告整洁、美观(注意字体、字号、对齐、截图大小和分页等;)

### 实验过程记录

#### 步骤 1

操作内容:

- 1, 打开一个 PowerShell 窗口;
- 2, 执行 docker, 确认已启动 Docker 环境;

操作结果:

```

PS C:\WINDOWS\system32> docker

Usage:  docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Options:
  --config string      Location of client config files (default
                        "C:\Users\像一条鱼沉沦在星辰大海\.docker")
  -c, --context string  Name of the context to use to connect to the
                        daemon (overrides DOCKER_HOST env var and
                        default context set with "docker context use")
  -D, --debug           Enable debug mode
  -H, --host list       Daemon socket(s) to connect to
  -l, --log-level string Set the logging level
                        ("debug"|"info"|"warn"|"error"|"fatal")
                        (default "info")
  --tls                Use TLS; implied by --tlsverify
  --tlscacert string    Trust certs signed only by this CA (default
                        "C:\Users\像一条鱼沉沦在星辰大海\.docker\ca.pem")
  --tlscert string      Path to TLS certificate file (default
                        "C:\Users\像一条鱼沉沦在星辰大海\.docker\cert.pem")
  --tlskey string       Path to TLS key file (default
                        "C:\Users\像一条鱼沉沦在星辰大海\.docker\key.pem")
  --tlsverify           Use TLS and verify the remote
  -v, --version         Print version information and quit

```

## 步骤 2

操作内容：

3, 执行 `docker pull enmotech/opengauss: 2.1.0`, 从 dockerhub 拉取 2.1.0 版本的 openGauss 镜像；

4, 执行 `docker run --name opengauss --privileged=true -d -e GS_PASSWORD=Password@123 -p 15432:5432 enmotech/opengauss:latest`, 在名为 opengauss 的容器中启动一个 openGauss 数据库的实例；

操作结果：

```

PS C:\WINDOWS\system32> docker pull enmotech/opengauss:2.1.0
2.1.0: Pulling from enmotech/opengauss
Digest: sha256:d5a3e38fa2553a44e7falc5cad0b4f0845a679858764067d7b0052a228578a0
Status: Image is up to date for enmotech/opengauss:2.1.0
docker.io/enmotech/opengauss:2.1.0
PS C:\WINDOWS\system32> docker run --name opengauss --privileged=true -d -e GS_PASSWORD=Password@123 -p 15432:5432 en
ech/opengauss:2.1.0
docker: Error response from daemon: Conflict. The container name "/opengauss" is already in use by container "63a736b
aafdc6124060145b91110bf2f0de52366e1b8d7f3071bee295d62b". You have to remove (or rename) that container to be able to
se that name.
See 'docker run --help'.
PS C:\WINDOWS\system32>

```

(之前已经拉取过了)

## 步骤 3

操作内容：

5, 执行 `docker ps`, 查看当前系统中的容器；

6, 执行 `docker exec -it opengauss bash`, 在容器内部启动一个 Bash Shell 进程, 并连接到当前 PowerShell 窗口；

7, 执行 `su omm`, 切换到 omm 用户；

8, 执行 `gsql -d postgres -U gaussdb -W 'Password@123'`, 连接 openGauss 数据库；

9, 执行内部命令 `\?` 查看 openGauss 内部命令的帮助信息；

10, 执行内部命令 `\help` 查看 SQL 语句的帮助信息；

11, 执行 SQL 语句 create database db2022;, 创建一个名为 db2022 的数据库对象;

12, 查看系统中的所有数据库对象;

(1) 方法一: 执行内部命令 \l

(2) 方法二: 执行 SQL 语句 SELECT datname FROM pg\_database; 查询系统表 pg\_database

操作结果:

```
PS C:\WINDOWS\system32> docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                    NAMES
33a736b458aa  errotech/opengauss:2.1.0           "entrypoint.sh gauss..." 46 hours ago  Up 13 seconds  0.0.0.0:15432->5432/tcp  opengauss

PS C:\WINDOWS\system32> docker exec -it opengauss bash
root@63a736b458aa:/# su omm
omm@63a736b458aa:/$ gsql -d postgres -U gaussdb -W 'Password@123'
gsql (openGauss 2.1.0 build 590b0f8e) compiled at 2021-09-30 14:29:04 commit 0 last mr )
Non-SSL connection (SSL connection is recommended when requiring high-security)
Type "help" for help.

openGauss=> \l
      List of databases
+----+-----+-----+-----+-----+-----+
| Name | Owner | Encoding | Collate | Ctype | Access privileges |
+----+-----+-----+-----+-----+-----+
| db2022 | gaussdb | UTF8 | en_US.UTF-8 | en_US.UTF-8 |
| omm | omm | UTF8 | en_US.UTF-8 | en_US.UTF-8 |
| postgres | omm | UTF8 | en_US.UTF-8 | en_US.UTF-8 |
| template0 | omm | UTF8 | en_US.UTF-8 | en_US.UTF-8 | =c/omm +
| template1 | omm | UTF8 | en_US.UTF-8 | en_US.UTF-8 | =c/omm =CTC/omm +
(5 rows)

openGauss=>
openGauss=> SELECT datname FROM pg_database;
 datname
+-----+
| template1 |
| omm |
| db2022 |
| template0 |
| postgres |
(5 rows)
```

#### 步骤 4

操作内容:

13, 执行内部命令 \c db2022, 将当前连接的数据库切换为 db2022;

14, 执行下列创建数据表的 SQL 语句;

15, 执行下列 SQL 语句, 向数据表中插入数据;

16, 执行下列 SQL 语句, 向数据表中插入数据;

(14, 15, 16 布的文件在附件中)

17, 执行内部命令 \d 或 \dt, 查看当前数据库中的所有数据表对象;

操作结果:

```
db2022=> \d
```

List of relations				
Schema	Name	Type	Owner	Storage
public	advisor	table	gaussdb	{orientation=row,compression=no}
public	classroom	table	gaussdb	{orientation=row,compression=no}
public	course	table	gaussdb	{orientation=row,compression=no}
public	department	table	gaussdb	{orientation=row,compression=no}
public	instructor	table	gaussdb	{orientation=row,compression=no}
public	prereq	table	gaussdb	{orientation=row,compression=no}
public	section	table	gaussdb	{orientation=row,compression=no}
public	student	table	gaussdb	{orientation=row,compression=no}
public	takes	table	gaussdb	{orientation=row,compression=no}
public	teaches	table	gaussdb	{orientation=row,compression=no}
public	time_slot	table	gaussdb	{orientation=row,compression=no}

(11 rows)

## 步骤 5

操作内容：

18, 执行内部命令 \d tablename 或 \d+ tablename, 查看数据表的结构和属性；

19, 执行内部命令 \di, 查看当前数据库中的所有索引对象；

操作结果：

```
db2022=> \d course
```

Column	Table "public.course" Type	Modifiers
course id	character varying(8)	not null
title	character varying(50)	
dept name	character varying(20)	
credits	numeric(2, 0)	

Indexes:  
 "course\_pkey" PRIMARY KEY, btree (course\_id) TABLESPACE pg\_default

Check constraints:  
 "course\_credits\_check" CHECK (credits > 0::numeric)

Foreign-key constraints:  
 "course\_dept\_name\_fkey" FOREIGN KEY (dept\_name) REFERENCES department(dept\_name) ON DELETE SET NULL

Referenced by:  
 TABLE "prereq" CONSTRAINT "prereq\_course\_id\_fkey" FOREIGN KEY (course\_id) REFERENCES course(course\_id) ON DELETE CASCADE  
 TABLE "prereq" CONSTRAINT "prereq\_prereq\_id\_fkey" FOREIGN KEY (prereq\_id) REFERENCES course(course\_id)  
 TABLE "section" CONSTRAINT "section\_course\_id\_fkey" FOREIGN KEY (course\_id) REFERENCES course(course\_id) ON DELETE CASCADE

```
db2022=> \di
```

List of relations					
Schema	Name	Type	Owner	Table	Storage
public	advisor_pkey	index	gaussdb	advisor	
public	classroom_pkey	index	gaussdb	classroom	
public	course_pkey	index	gaussdb	course	
public	department_pkey	index	gaussdb	department	
public	instructor_pkey	index	gaussdb	instructor	
public	prereq_pkey	index	gaussdb	prereq	
public	section_pkey	index	gaussdb	section	
public	student_pkey	index	gaussdb	student	
public	takes_pkey	index	gaussdb	takes	
public	teaches_pkey	index	gaussdb	teaches	
public	time_slot_pkey	index	gaussdb	time_slot	

(11 rows)

## 步骤 6

操作内容：

20, 执行内部命令 \c postgres, 将当前连接的数据库切换为 postgres；

21, 执行内部命令 \d 或 \dt, 查看当前数据库中的所有数据表对象；

22 执行内部命令 \q, 退出客户端工具 gsql, 断开与数据库的连接；

操作结果：（已经通过步骤七中的图形数据端 DataStudio 创建好表）

```
db2022=> \c postgres
Password for user gaussdb:
Non-SSL connection (SSL connection is recommended when requiring high-security)
You are now connected to database "postgres" as user "gaussdb".
openGauss=> \d
```

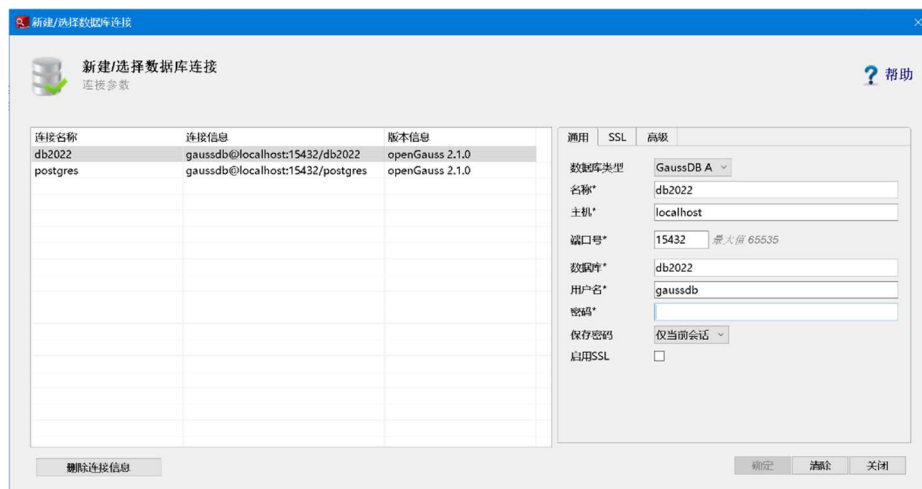
Schema	Name	Type	Owner	Storage
gaussdb	advisor	table	gaussdb	{orientation=row,compression=no}
gaussdb	classroom	table	gaussdb	{orientation=row,compression=no}
gaussdb	course	table	gaussdb	{orientation=row,compression=no}
gaussdb	department	table	gaussdb	{orientation=row,compression=no}
gaussdb	instructor	table	gaussdb	{orientation=row,compression=no}
gaussdb	prereq	table	gaussdb	{orientation=row,compression=no}
gaussdb	section	table	gaussdb	{orientation=row,compression=no}
gaussdb	student	table	gaussdb	{orientation=row,compression=no}
gaussdb	takes	table	gaussdb	{orientation=row,compression=no}
gaussdb	teaches	table	gaussdb	{orientation=row,compression=no}
gaussdb	time_slot	table	gaussdb	{orientation=row,compression=no}

(11 rows)

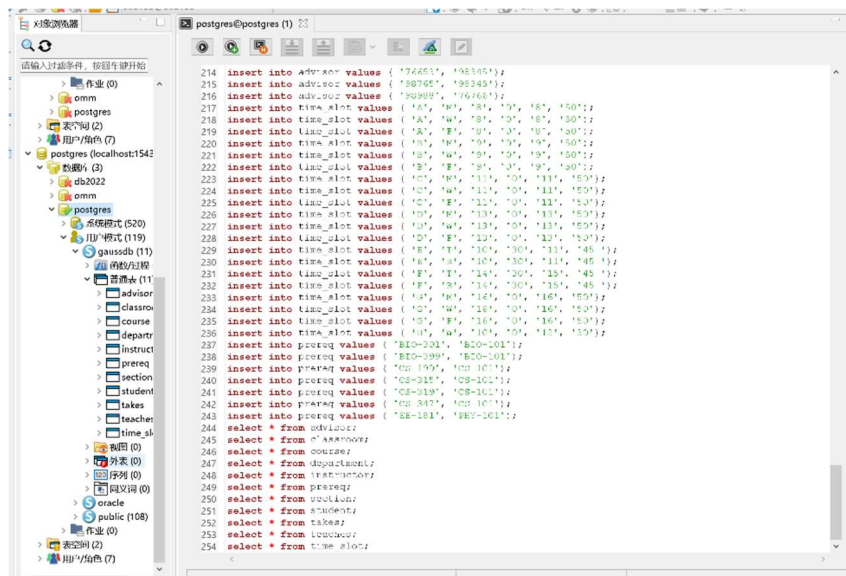
## 步骤 7：使用 openGauss 官方图形客户端工具 Data Studio

操作内容：

- 1，下载并运行客户端工具 Data Studio；
- 2，点击左上角“新建连接”按钮，在弹出的对话框中填入连接所需参数；
- 3，针对 postgres 数据库连接，通过打开新的终端按钮新建一个终端窗口，并在此终端窗口中执行步骤 13~15；







## 步骤 8: 讲 Docker 保存为本地镜像文件

操作内容:

- 1, 打开一个新的 PowerShell 窗口, 执行 `docker ps` 查看运行中的容器;
- 2, 执行 `docker commit opengauss mygauss:2022-01-14`, 将容器 opengauss 打包成名为 mygauss 的镜像;
- 3, 执行 `docker images` 查看本地仓库中的镜像;

操作结果:

```
PS C:\WINDOWS\system32> docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
63a736b458aa   enmotech/opengauss:2.1.0           "entrypoint.sh gauss..." 47 hours ago   Up 37 minutes   0.0.0.0:15432->5432/tcp
cp            opengauss
PS C:\WINDOWS\system32> docker commit opengauss mygauss:2022-01-14
sha256:75f0db610f330337bf715aef0072b7df03383ad3da58b858ca4eccf5a82d4d2a
PS C:\WINDOWS\system32> docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
mygauss        2022-01-14 75f0db610f33   12 seconds ago 1.22GB
<none>        <none>    809effdd01a3   46 hours ago   1.22GB
<none>        <none>    7ca89493a645   46 hours ago   1.22GB
enmotech/opengauss latest    c34d1c0662c9   3 months ago   522MB
enmotech/opengauss 2.1.0     b4dd24d09223   17 months ago   383MB
PS C:\WINDOWS\system32>
```

## 步骤 9

操作内容:

- 4, 执行 `docker save -o mygauss.tar mygauss`, 将镜像保存为文件;

操作结果:

```
2023/3/7 10:39:15
-a----- 2023/3/7 10:39 1226393600 mygauss.tar
```

## 步骤 10

操作内容:

5, 执行 `docker image rm mygauss:2022-01-14`, 将本地仓库中的 mygauss 镜像删除;

6, 执行 `docker load --input mygauss.tar`, 将镜像文件 mygauss.tar 加载到本地仓库;

7, 执行 `docker stop opengauss`, 停止容器 opengauss 的运行;

8, 执行 `docker rm opengauss`, 删除容器 opengauss;

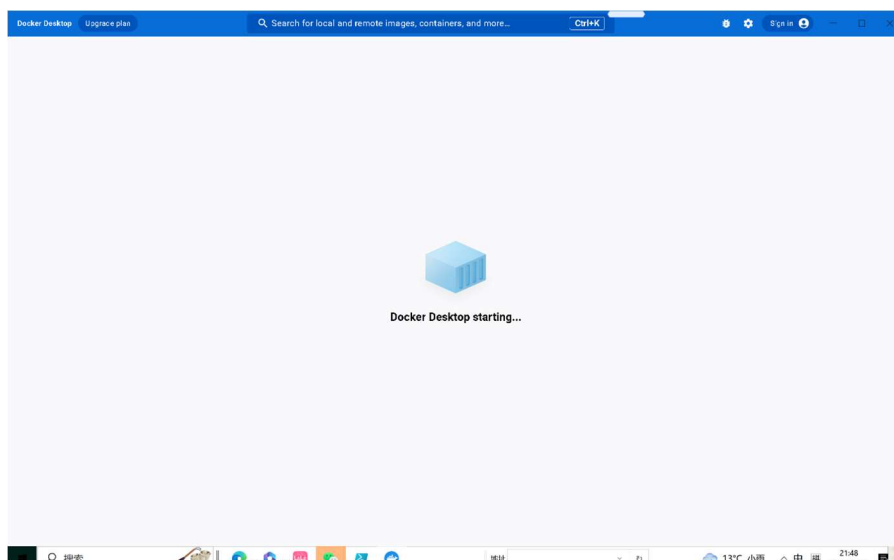
9, 执行 `docker run --name opengauss --privileged=true -d -e GS_PASSWORD=Password@123 -p 15432:5432 mygauss:2022-01-14`, 基于 mygauss 镜像启动一个名为 opengauss 的容器;

操作结果:

```
PS C:\WINDOWS\system32> where mygauss.tar
PS C:\WINDOWS\system32> docker image rm mygauss:2022-01-14
Untagged: mygauss:2022-01-14
Deleted: sha256:75f0db610f330337bf715aef0072b7df03383ad3da58b858ca4eccf5a82d4d2a
Deleted: sha256:a501eaba4b1ff6c6aa19772bb90e7a9bd978898821034e1dc314bebbba02f10d9
PS C:\WINDOWS\system32> docker load --input mygauss.tar
f22e4b1087d5: Loading layer [=====] 839.3MB/839.3MB
Loaded image: mygauss:2022-01-14
PS C:\WINDOWS\system32> docker stop opengauss
opengauss
PS C:\WINDOWS\system32> docker rm opengauss
opengauss
PS C:\WINDOWS\system32> docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS   NAMES
PS C:\WINDOWS\system32> docker run --name opengauss --privileged=true -d -e GS_PASSWORD=Password@123 -p 15432:5432 mygauss:2022-01-14
fbf53875fc11e9124a961f4e0263373cbc5461a4f56049c1284df3d563dad15
PS C:\WINDOWS\system32> docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS   NAMES
fbf53875fc11e9124a961f4e0263373cbc5461a4f56049c1284df3d563dad15   mygauss:2022-01-14   "entrypoint.sh gauss..."   About a minute ago   Up About a minute   0.0.0.0:15432->5432/tcp   opengauss
PS C:\WINDOWS\system32>
```

## 存在的问题及解决方案

在官网下载最新版版本的 docker 应根据自身电脑选择 Hyper-v 还是 WSL2,windows 专业版选择 WSL 后可能会出现没有反应的情况



具体参考官方文档：<https://docs.docker.com/docker-for-windows/install/>选择合适的（我

是使用的 Hyper-v 正常最终正常启动

2，在执行到这一步时

```
PS C:\WINDOWS\system32> docker exec -it opengauss bash
```

计算机没有反应根据 ChatGPT 的指引仍然不能够解决

如果您在运行“docker exec -it bash”命令时没有看到任何反应，则可能是以下原因之一：1. 没有正确指定容器名称或ID。请确保您已正确指定要运行bash命令的容器的名称或ID。2. 容器没有运行。请确保您的容器正在运行，否则您将无法使用“docker exec”命令进入容器。3. 没有安装bash。请确保您的容器中已安装bash，否则您将无法使用“docker exec”命令进入容器。您可以尝试使用其他shell，如sh或zsh。4. 您的Docker版本过低。请确保您的Docker版本至少为1.3，否则您将无法使用“docker exec”命令进入容器。如果您已经排除了以上问题，但仍然无法使用“docker exec -it bash”命令进入容器，请尝试使用“docker exec -it /bin/bash”命令进入容器。

检查发现：window powershell ISE 会卡住没有反应，使用 Powershell 可以正常运行。

## 实验小结

- 1、对于数据库的操作我们可以通过命令行界面与图形界面进行操作
- 2、在用 datastudio 时应当注意 jdk 版本以及，不要安装多个 jdk 版本，否则会报错。
- 3、跟着步骤慢慢来，出问题就换一些方法。