

实验五：miniOB 入门实验

一、实验目的

1. 在了解 OceanBase 数据库的基础上，尝试在本地成功运行 miniOB；
2. 在 miniOB 数据库上编写基础的 SQL 命令：建表、插入、更新、删除；
3. 了解和探索 miniOB 数据库的特点，为后续编写内核代码做准备。

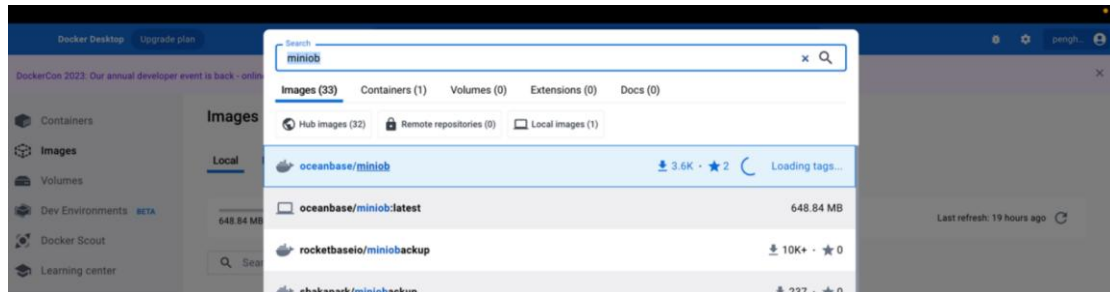
二、实验要求说明

1. 本次实验使用 miniOB 数据库完成；
2. 在个人电脑完成实验环境搭建；
3. 提供 Docker 运行截图；
4. 按要求完成第四节中的实验，每步操作后要求截图，实验截图要求包含完整的 SQL 执行语句和输出的实验结果。

三、miniob 安装说明

1. 在 Docker 中下载官方镜像

在 Docker 顶部搜索框搜索 miniob，oceanbase/miniob 即为官方镜像。



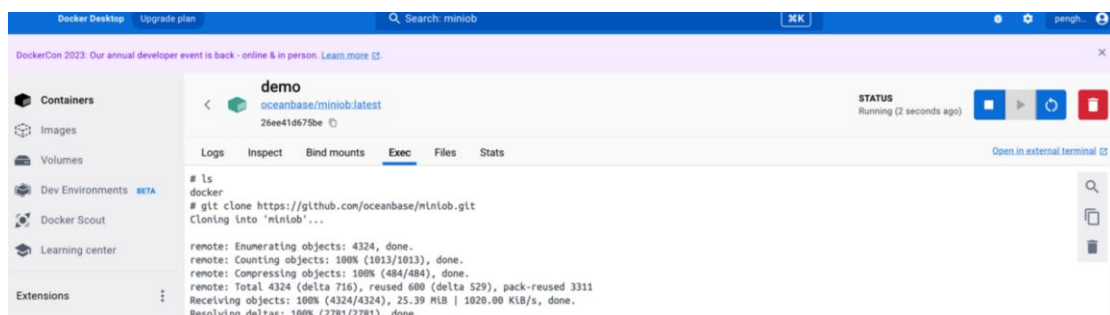
点击打开后，选择 pull 并 run 即可运行。

2. 进入容器，下载并编译 miniOB

在 Exec 下运行

```
git clone https://github.com/oceanbase/miniob.git
```

从 github 上下载 miniOB。



进入 miniOB 目录，使用

```
bash build.sh --make -j4
```

进行编译。

```
# cd miniob
# ls
benchmark  cmake          CODE_OF_CONDUCT.md  deps  docs  etc  NOTICE  src  tools
build.sh   CMakeLists.txt CONTRIBUTING.md     docker Doxyfile License README.md test unittest
# bash build.sh --make -j4
build.sh --make -j4
create soft link for build_debug, linked by directory named build
cmake -DCMAKE_EXPORT_COMPILE_COMMANDS=1 --log-level=STATUS /root/miniob -DCMAKE_BUILD_TYPE=Debug -DDEBUG=ON
-- The C compiler identification is GNU 11.4.0
-- The CXX compiler identification is GNU 11.4.0
-- Detecting C compiler ABI info
[100%] Linking CXX executable ../bin/pidfile_test
[100%] Built target pidfile_test
[100%] Linking CXX executable ../bin/ring_buffer_test
[100%] Linking CXX executable ../bin/record_manager_test
[100%] Built target persist_test
[100%] Built target ring_buffer_test
[100%] Built target record_manager_test
#
```

3. 运行 miniOB

编译完成后，进入 build 目录，使用

```
./bin/observer -s miniob.sock -f ../etc/observer.ini &
```

在后台启动服务端。

```
# cd build
# ./bin/observer -s miniob.sock -f ../etc/observer.ini &
# Successfully load ../etc/observer.ini
```

启动服务端后，使用

```
./bin/obclient -s miniob.sock
```

启动 miniOB 客户端。

```
# ./bin/obclient -s miniob.sock
miniob > help
Commands
show tables;
desc `table name`;
create table `table name` (`column name` `column type`, ...);
create index `index name` on `table` (`column`);
insert into `table` values(`value1`,`value2`);
update `table` set column=value [where `column`=`value`];
delete from `table` [where `column`=`value`];
select [ * | `columns` ] from `table`;
miniob >
```

四、实验内容

1. 在 Docker 中建立 miniob 环境（参考第 0 章的教程 PPT 和第三节的内容）。
2. 创建一张表，包括学号，姓名，绩点，学分。

表名：Student		
字段	说明	类型
No	学号	int
Name	姓名	char(10)
Grade	绩点	float
Credit	学分	float

3. 向该表插入几行数据，其中需要包含一条包含个人学号、姓名的数据。

No	Name	Grade	Credit
2350001	张三	4.8	120
2350002	李四	4.7	124.5
2350003	王五	4.2	99.5
你的学号	你的姓名	自定	自定

4. 使用 select 语句展示学号，绩点，学分。
5. 尝试修改指定行的绩点或学分如下表所示，能否成功？为什么？

No	Name	Grade	Credit
2350001	张三	4.82	120
2350002	李四	4.65	124.5
2350003	王五	4.2	103.5
你的学号	你的姓名	自定	自定

6. 删除张三的记录。
7. 使用 select 语句展示你的学号。
8. 对 miniob 源码进行阅读，主要选取一个功能（如 create table、insert、delete 等）进行分析理解，做简要报告（不超过两页）。