

《数据库系统原理》实验报告（八）

题目：上机实验课（八）openGauss

学号	1853790	姓名	庄镇华	日期	2020. 11. 24
----	---------	----	-----	----	--------------

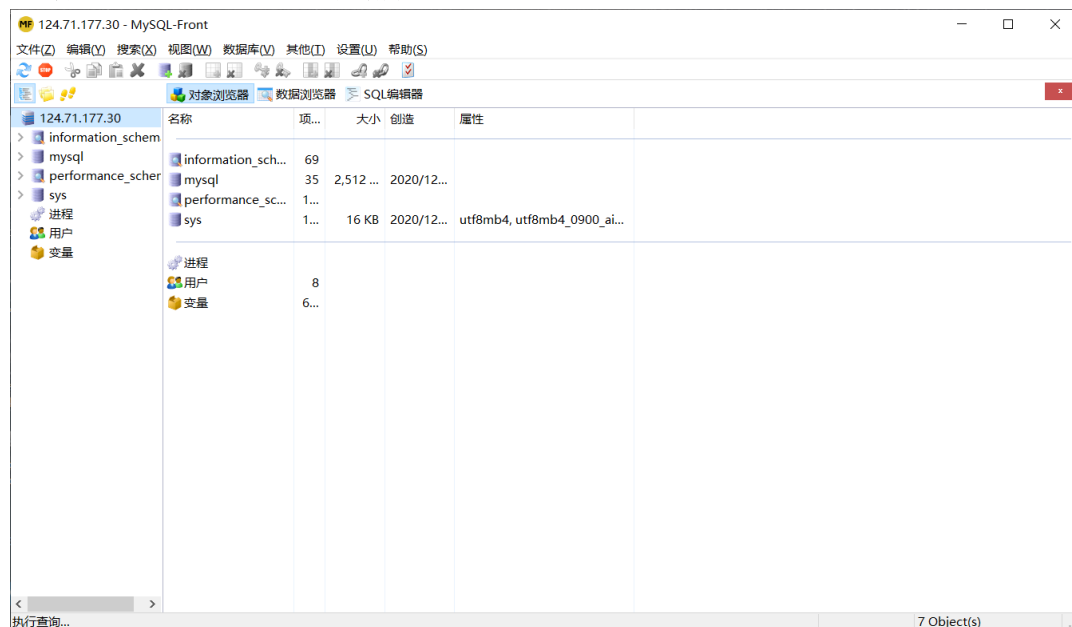
实验环境：华为云数据库 GaussDB (for MySQL)

实验步骤及结果截图：

1. 根据连接 openGauss 操作指南.pdf 购买 openGauss 云数据库



2. 绑定公网 IP，成功连接数据库

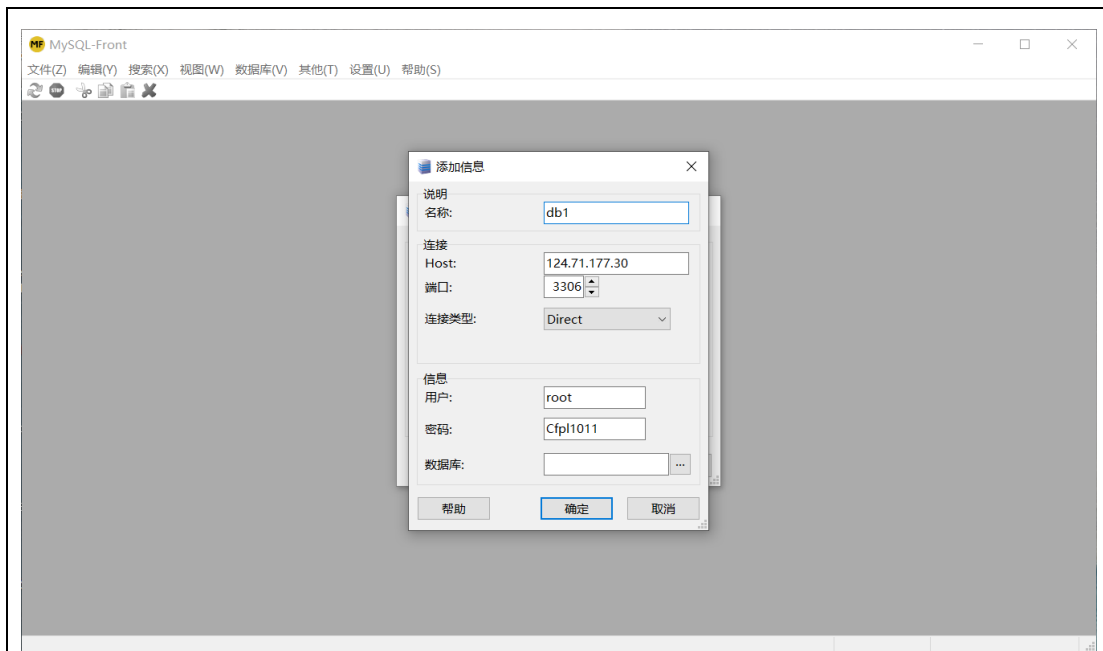


3. 本地连接数据库（使用 MySQL-Front 或 SSL 方式）

连接数据库参考链接：

https://support.huaweicloud.com/qs-gaussdb/gaussdb_02_0015.html

使用 MySQL-Front 连接，注意 Host 是读写公网地址：



4. 创建学生表

```
CREATE TABLE IF NOT EXISTS `SNO` (
  `S_id` INT UNSIGNED AUTO_INCREMENT,
  `S_NAME` VARCHAR(40),
  PRIMARY KEY ( `S_id` )
);
```

```
CREATE TABLE IF NOT EXISTS SNO(
  S_id INT UNSIGNED AUTO_INCREMENT,
  S_NAME VARCHAR(40),
  PRIMARY KEY (S_id)
);
```

-----开始执行-----

【拆分SQL完成】：将执行SQL语句数量：（1条）

【执行SQL：（1）】

```
CREATE TABLE IF NOT EXISTS SNO(
  S_id INT UNSIGNED AUTO_INCREMENT,
  S_NAME VARCHAR(40),
  PRIMARY KEY (S_id)
)
执行成功，耗时：[15ms.]
```

5. 插入 10000 条数据并记录运行时间（方式不限）

```
DROP PROCEDURE IF EXISTS INSERT_DATA;
DELIMITER //
CREATE PROCEDURE INSERT_DATA()
BEGIN
    DECLARE i INT DEFAULT 1;
    WHILE i <= 10000 DO
        INSERT INTO SNO(S_id, S_NAME) VALUES(i, 'zzh');
        SET i = i + 1;
    END WHILE;
    COMMIT;
END //
DELIMITER;
```

【执行SQL: (2)】

```
create procedure insert_while()
begin
    declare i int default 1;
    while i<= 10000
    do
        insert into SNO (S_id,S_NAME) VALUES (i,'zzh');
        set i=i+1;
    end while;
    commit;
end
执行成功，耗时：[4ms.]
```

```
CALL INSERT_DATA();
```

-----开始执行-----

【拆分SQL完成】：将执行SQL语句数量：（1条）

【执行SQL: (1)】

```
CALL INSERT_DATA()
```

执行成功，当前返回：[0]行，耗时：[17229ms.]

耗时 17229ms

6. 在本地 Oracle 上重复执行上述 4.5 操作，并记录运行时间

```
CREATE TABLE SNO(
    S_id INT,
    S_NAME VARCHAR(40),
    PRIMARY KEY (S_id)
);
```

```
SET serveroutput ON;
CREATE OR REPLACE PROCEDURE INSERT_DATA
IS i number;
BEGIN
    i := 1;
    dbms_output.PUT_LINE('t0' || dbms_utility.GET_TIME());
    FOR i IN 1..10000 LOOP
        INSERT INTO SNO VALUES(i, 'zzh');
    END LOOP;
    dbms_output.PUT_LINE('t1' || dbms_utility.GET_TIME());
END INSERT_DATA;
```

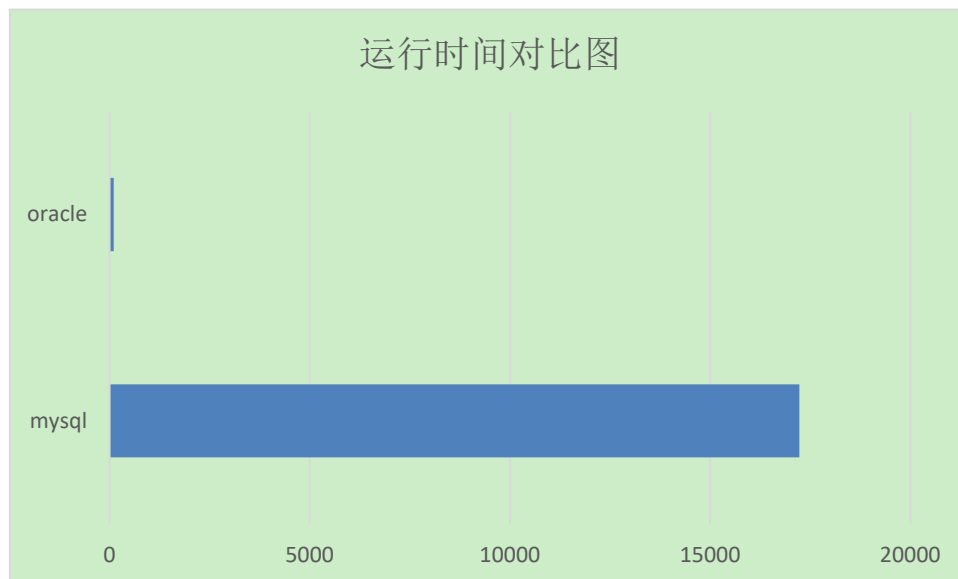
Procedure created.

```
EXEC INSERT_DATA;
```

t0434913588
t1434913692
PL/SQL procedure successfully completed.

耗时 434913692 - 434913588 = 104ms

7. 画出两个数据库运行时间对比条形图



出现的问题:

连接 GaussDB(for MySQL) 实例时出现了很多问题, 通过认真阅读说明指南解决。
oracle 和 mysql 的语法不同, 在建表、查询、插入等方面都遇到了一些问题。

解决方案:

弄清 oracle、mysql 语法的不同之处, 下面以建表情况举例:

mysql

```
CREATE TABLE IF NOT EXISTS SNO(  
    S_id INT UNSIGNED AUTO_INCREMENT,  
    S_NAME VARCHAR(40),  
    PRIMARY KEY (S_id)  
);
```

oracle

```
CREATE TABLE SNO(  
    S_id INT,  
    S_NAME VARCHAR(40),  
    PRIMARY KEY (S_id)  
);
```