

哈尔滨工业大学（深圳）

# 数据库实验指导书

实验一 MySQL 关系数据库管理系统  
及 SQL 语言的使用

2022 秋

## 目录

1	实验目的 .....	4
2	实验环境 .....	4
3	实验内容 .....	4
4	实验步骤 .....	4
4.1	数据准备 .....	4
4.1.1	启动 MySQL 服务 .....	4
4.1.2	创建 sakila 数据库 .....	8
4.1.3	检查 sakila 数据库中数据 .....	14
4.2	熟悉 sakila 数据库并回答问题 .....	16
4.3	数据的增删改查 .....	18
3.1.1	跟着完成以下内容 .....	18
3.1.2	自己写 SQL 语句完成以下内容 .....	22
5	附录 .....	23
5.1	MySQL 手册（命令行方式） .....	23
5.1.1	启动 MySQL 服务 .....	23
5.1.2	连接 MySQL 服务器 .....	24
5.1.3	查看 MySQL 中有哪些数据库 .....	25
5.1.4	选定数据库 .....	25
5.1.5	查看当前在的数据库 .....	25
5.1.6	查看当前数据库中的表 .....	26
5.1.7	查看某个表的表结构 .....	26
5.1.8	使用 help 命令来了解其他命令和变量类型等的含义 .....	26
5.1.9	取消命令 .....	27
5.1.10	断开 MySQL 服务器连接 .....	27
5.1.11	导出数据库 .....	27
5.1.12	创建数据库 .....	28
5.1.13	导入数据库 .....	28
5.1.14	使用数据文件导入数据 .....	29
5.2	常见问题 .....	30
5.2.1	'mysql' 不是内部或外部命令,也不是可运行的程序或批处理文件 .....	30
5.2.2	通过 MySQL Workbench 启动服务报错 .....	34
5.2.3	Run SQL Script: Error executing SQL script.....	34
5.2.4	打开 sakila.mwb 的时候报 Error unserializing GRT data string too long.....	37
5.2.5	ERROR 1300 (HY000): Invalid utf8mb4 character string: '' .....	37
5.2.6	BOM(of UTF-8) is not correctly handled by LOAD DATA..	38
5.2.7	load data 报错 ERROR 1062 (23000): Duplicate entry '' for key 'PRIMARY' .....	40

5.2.8 load data 报错 ERROR 29 (HY000): File '*****' not found (OS errno 2 - No such file or directory).....	40
5.2.9 load data 没有报错, 但数据不正确 .....	40
5.2.10 Error Code: 2013. Lost connection to MySQL server during query 30.000 sec.....	43
5.2.11 Errcode: 28 - No space left on device.....	44
5.2.12 ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES).....	44
5.2.13 中文显示乱码 .....	45
6、参考资料.....	45

# 1 实验目的

熟悉 MySQL 关系数据库管理系统的基本操作，学习并掌握使用 SQL 语言操作 MySQL 数据库。掌握 SQL 语言的使用方法，学会使用 SQL 语言进行增、删、改、查，特别是聚集查询、连接查询和嵌套查询。

# 2 实验环境

Windows 10 操作系统、MySQL8.0。

# 3 实验内容

1. 导入 MySQL 样例数据库 Sakila;
2. 观察和理解 Sakila 数据库;
3. 使用 SQL 语言完成要求的增、删、改、查操作。

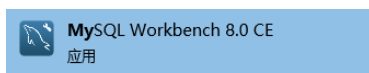
# 4 实验步骤

本节是以图形界面做示范，同学们也可参考 4.1 节用命令行方式完成本次实验。

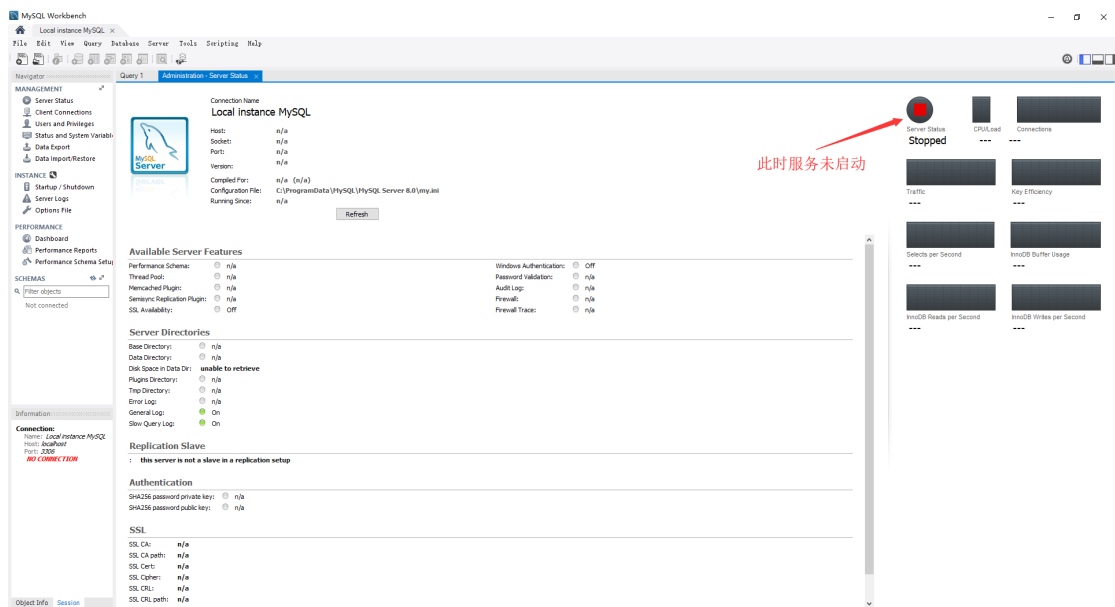
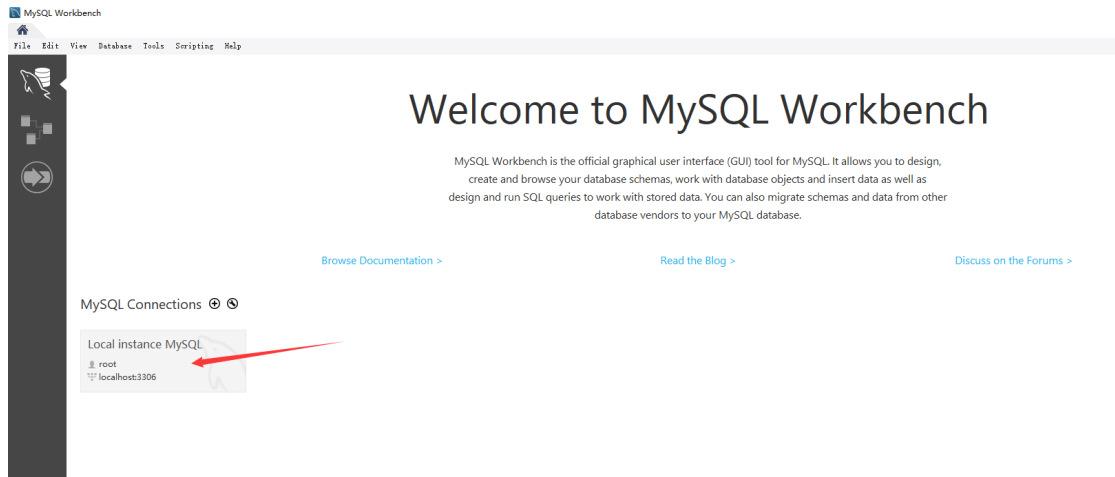
## 4.1 数据准备

### 4.1.1 启动 MySQL 服务

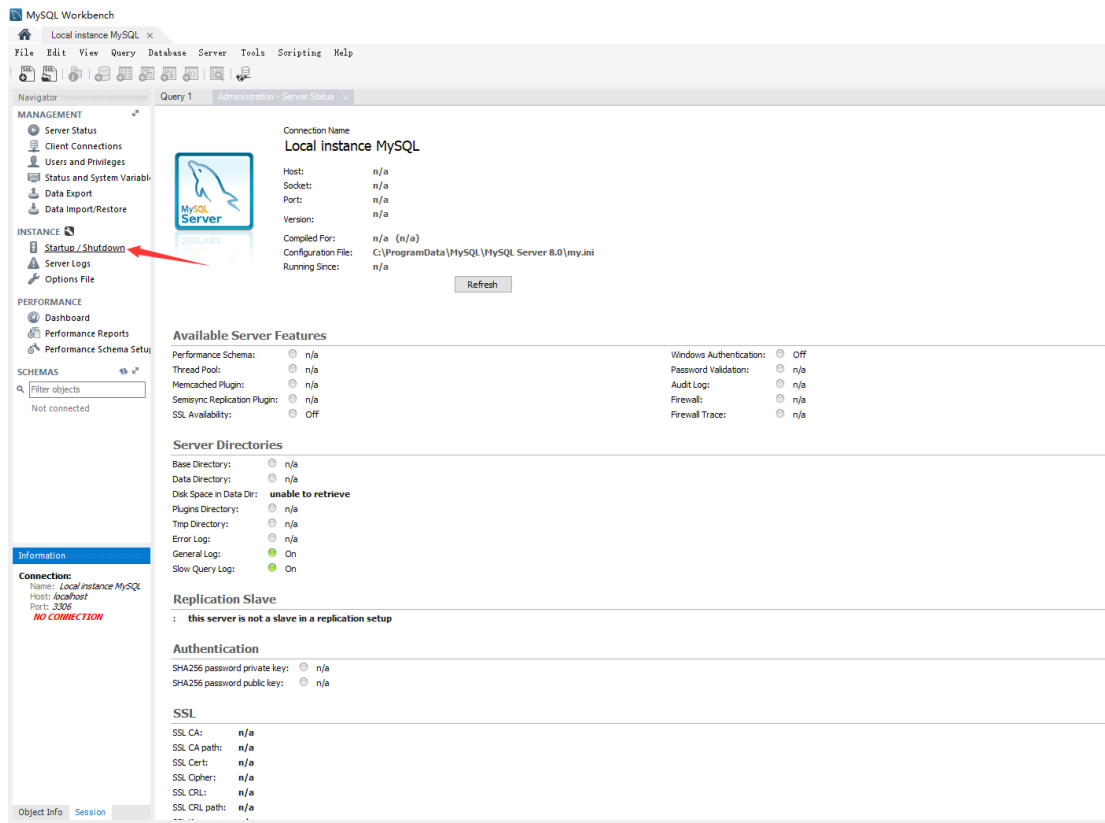
- 1、打开 MySQL Workbench



- 2、点击 “Local instance MySQL”

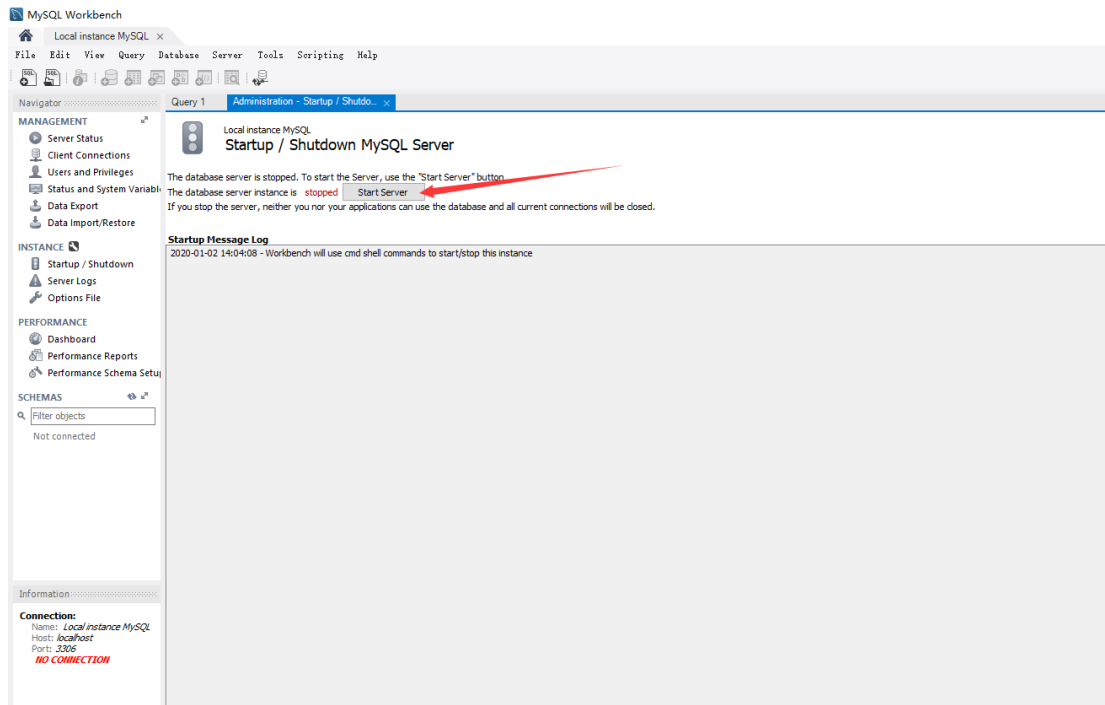


3、点击左边菜单里“Startup/Shutdown”

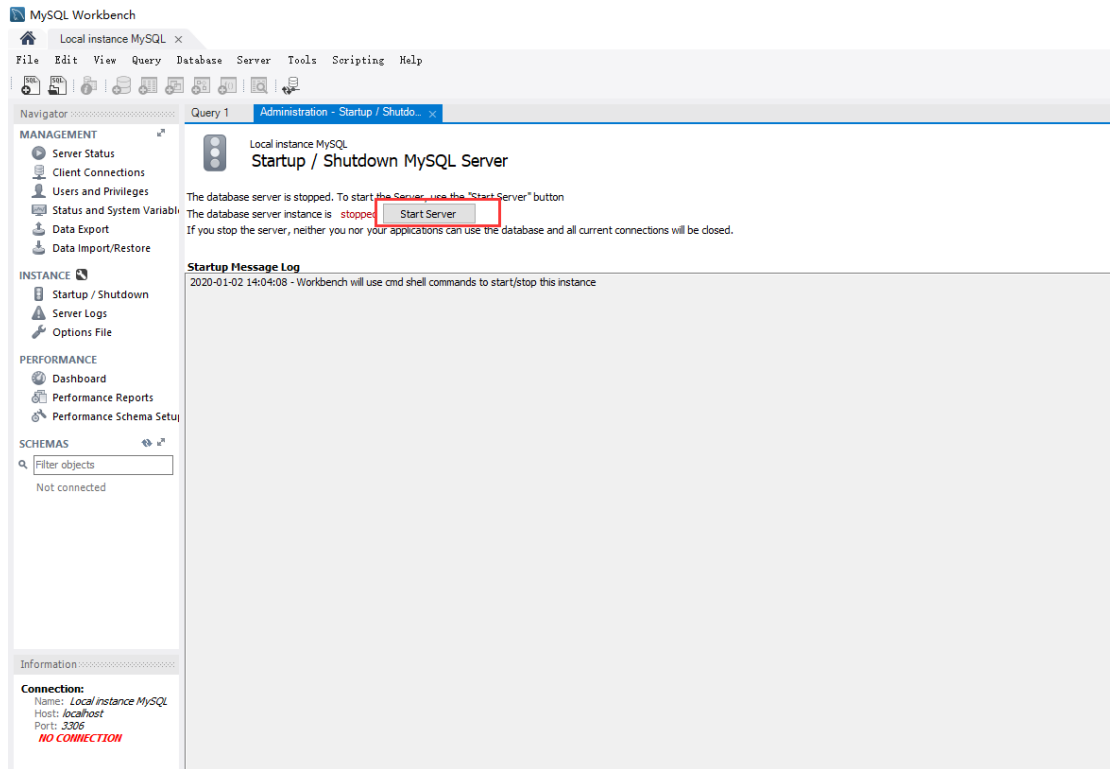


如果上面这一步报错，请看 4.2.2。

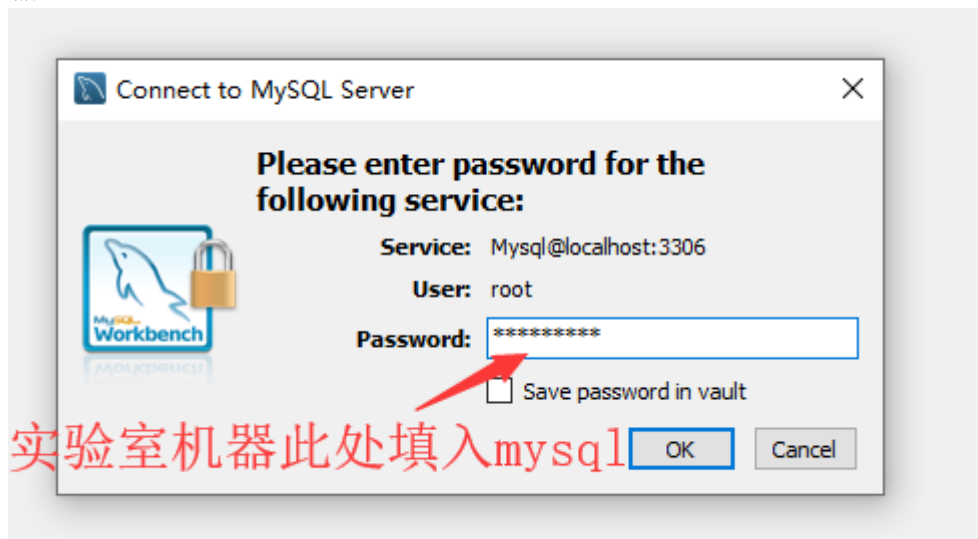
进去服务启停界面：



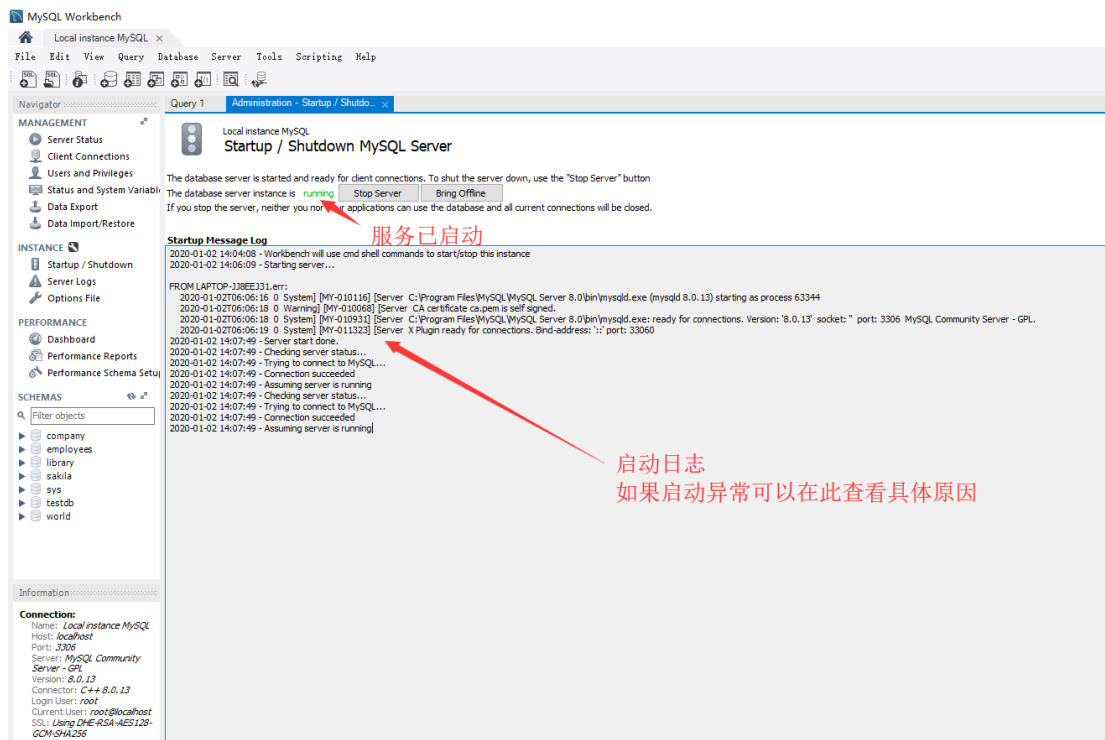
4、点击“Start Server”按钮：



输入 root 密码:

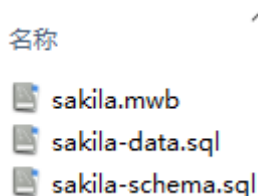


等待片刻，显示服务已启动:



## 4.1.2 创建 sakila 数据库

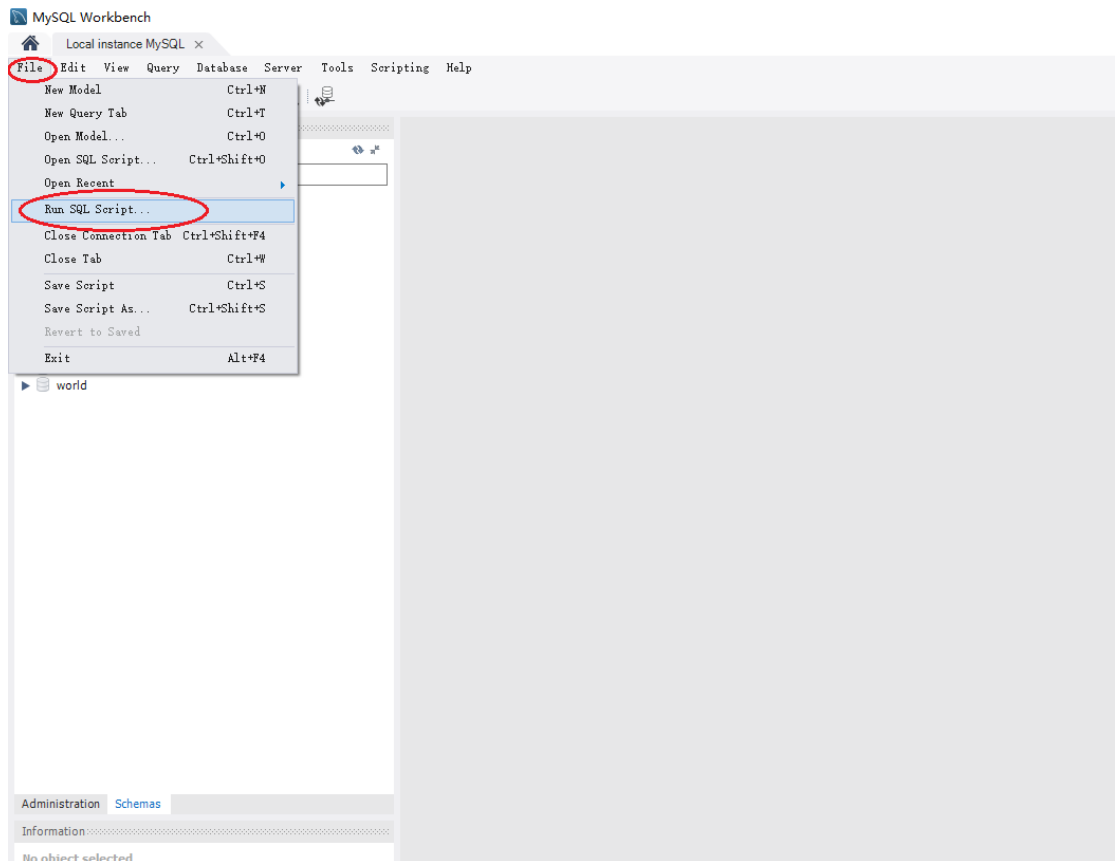
- 1、将下发的 sakila-db.zip 解压，打开解压后的目录可见 3 个文件：



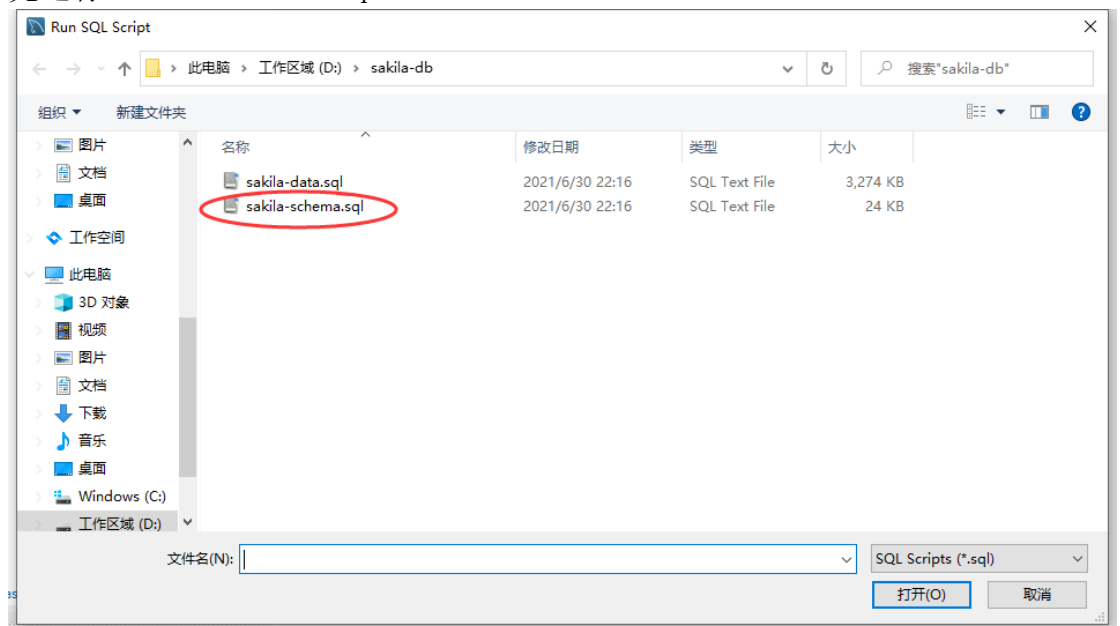
注意：不要放到有中文的目录，否则执行第 3 步会报错。

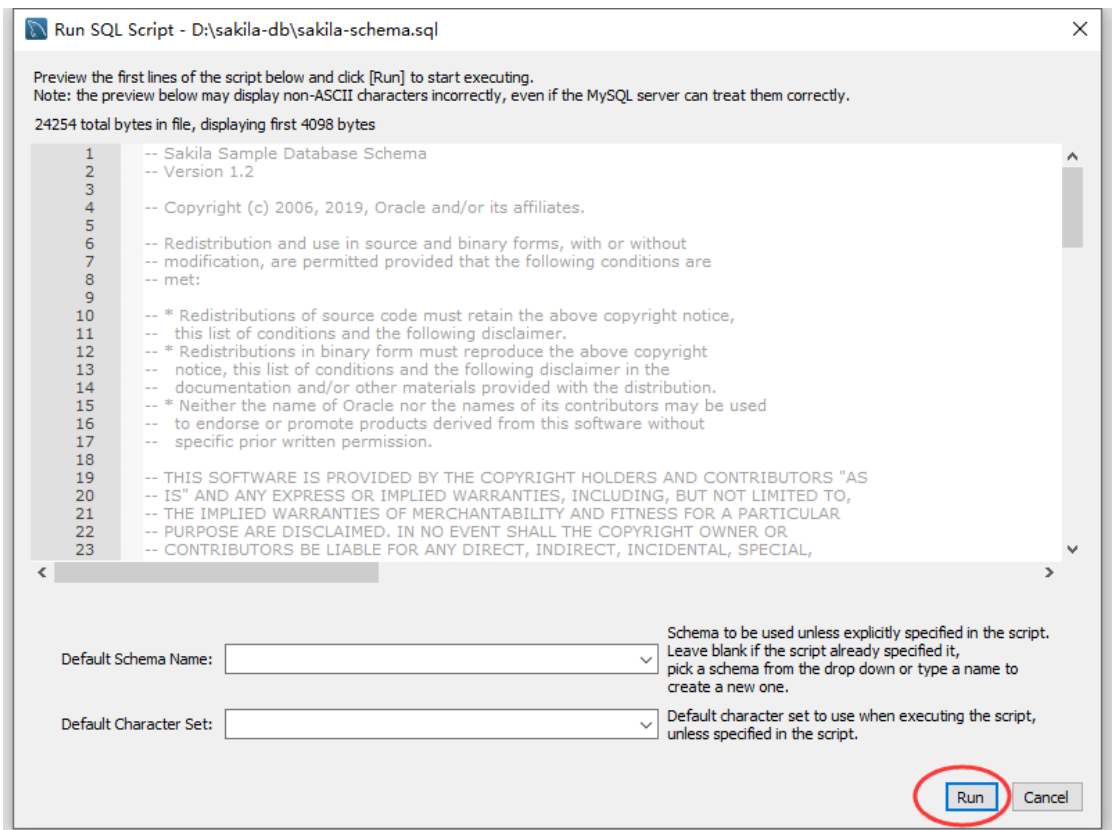
- 2、参考 sakila 数据库的官方文档  
<https://dev.mysql.com/doc/sakila/en/sakila-installation.html>，了解每个文件的用法。
- 3、可按照官方文档提供的命令行方式建库导入数据，也可以按下面步骤操作：
  - 1) 打开 MySQL Workbench, File→Run SQL Script...



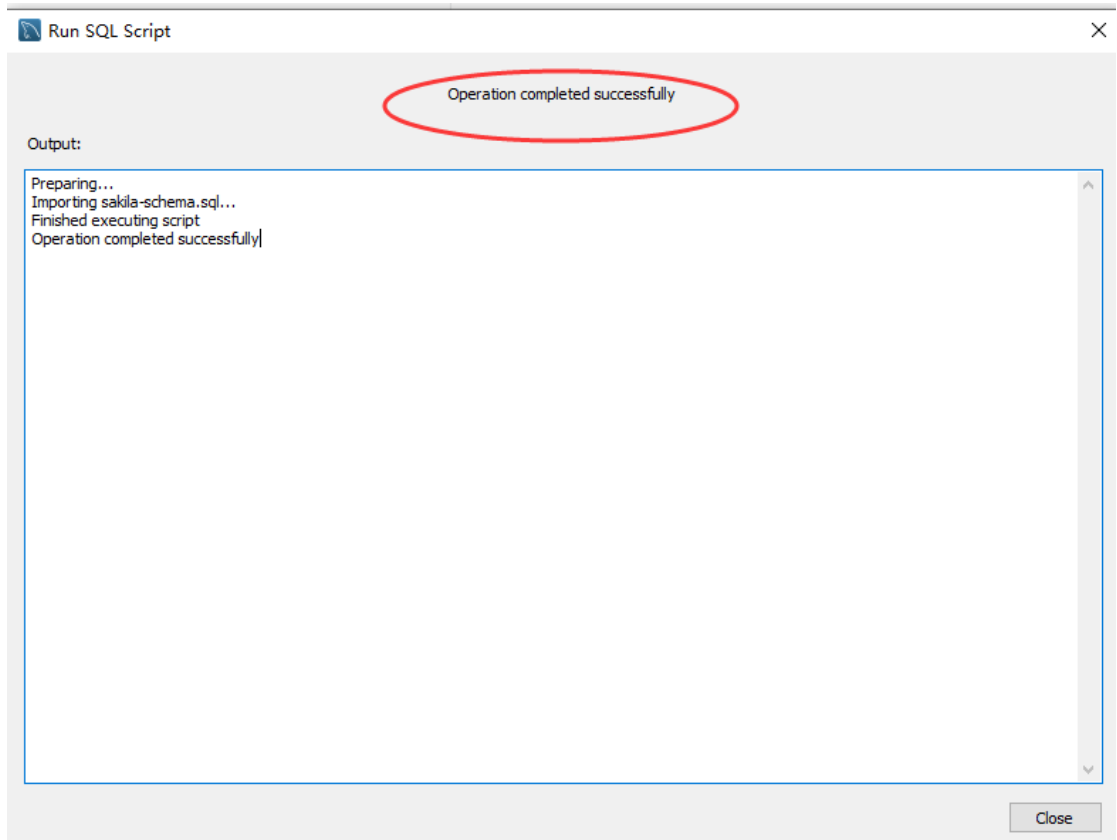


2) 先运行 sakila-schema.sql

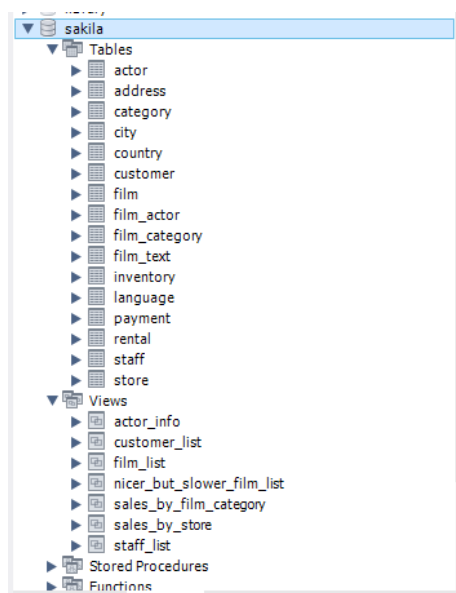
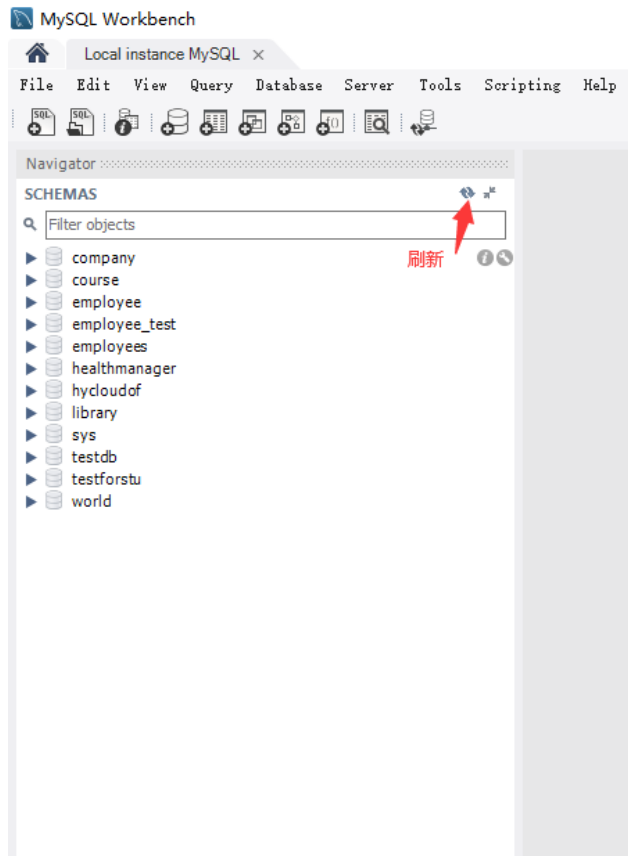




直接 Run。  
运行完毕如下：

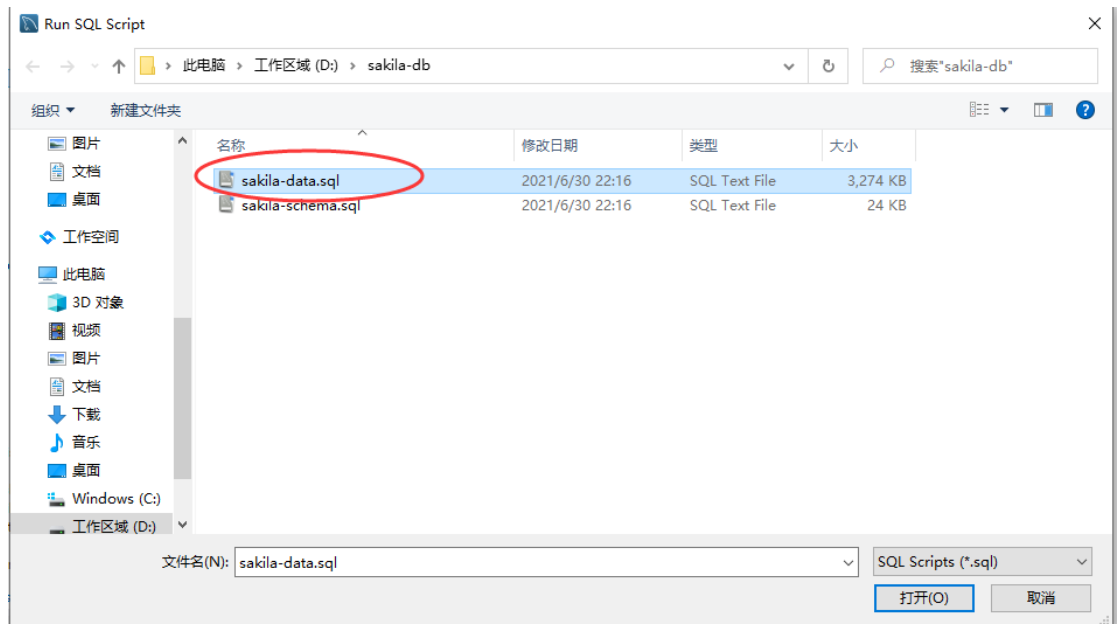


点刷新，可以查看 sakila 数据库：

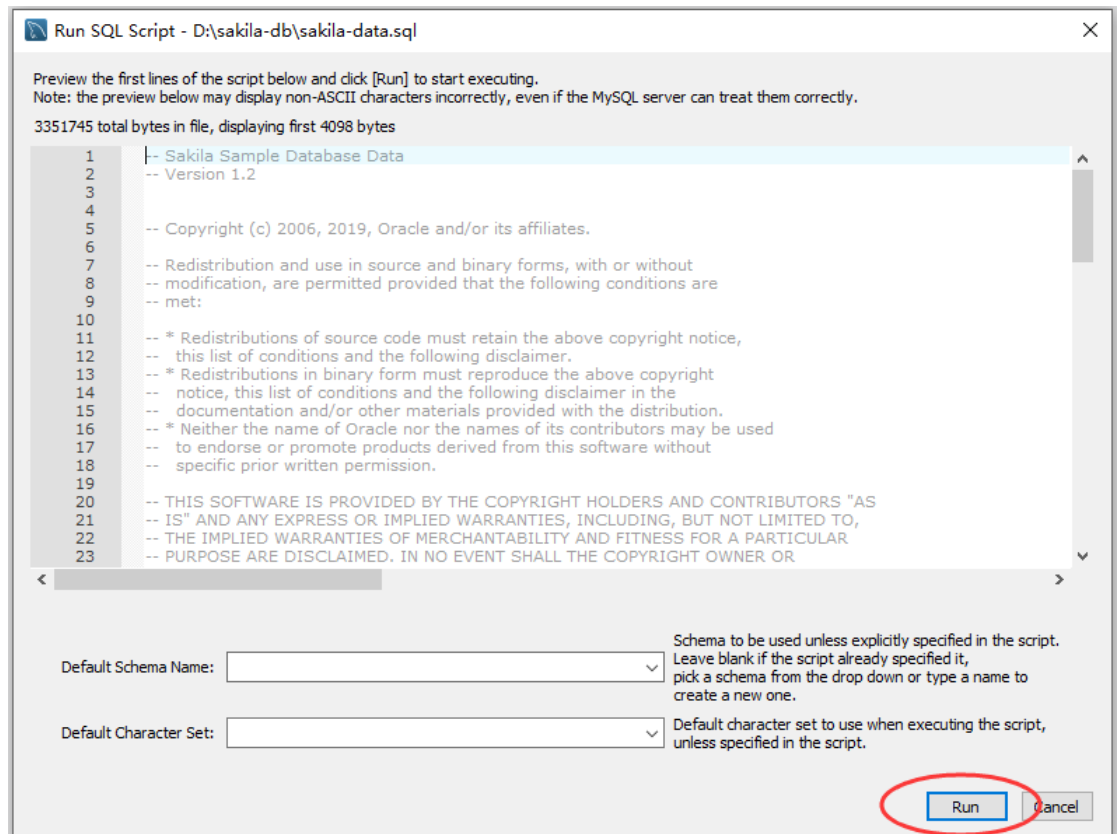


此时表里面并没有数据。

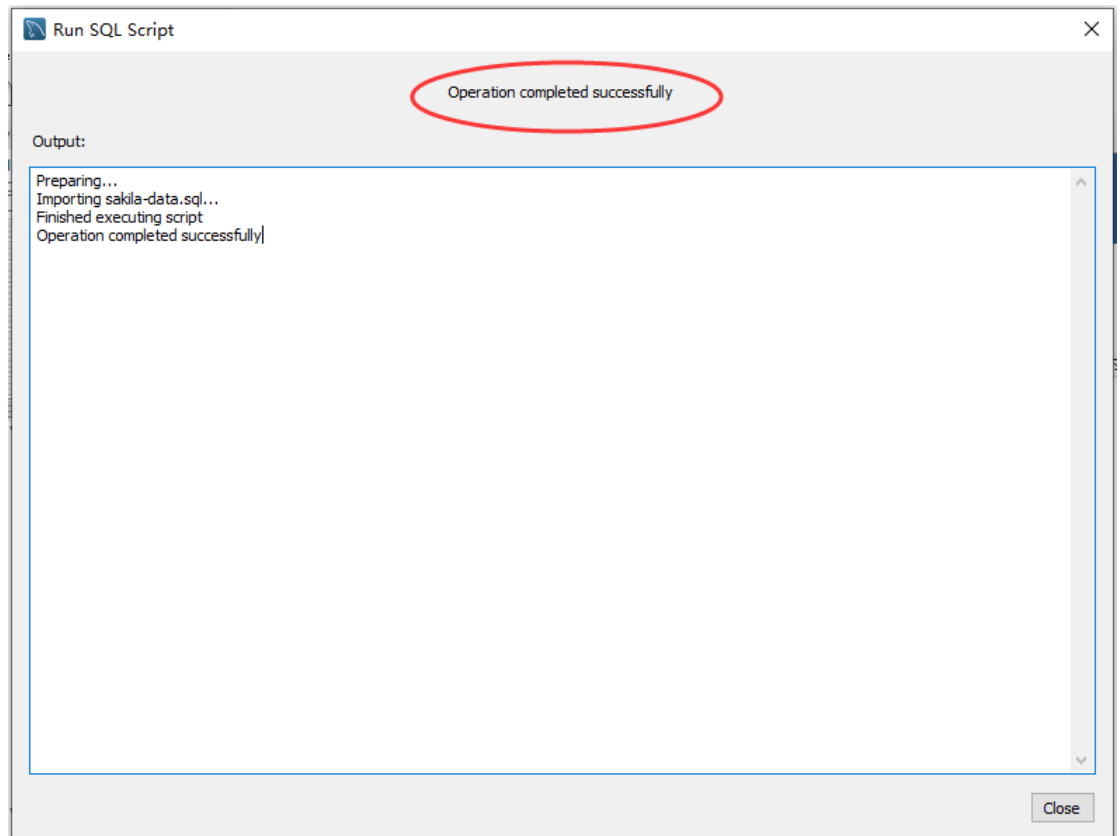
- 3) 按步骤 2 的方法运行 sakila-data.sql:



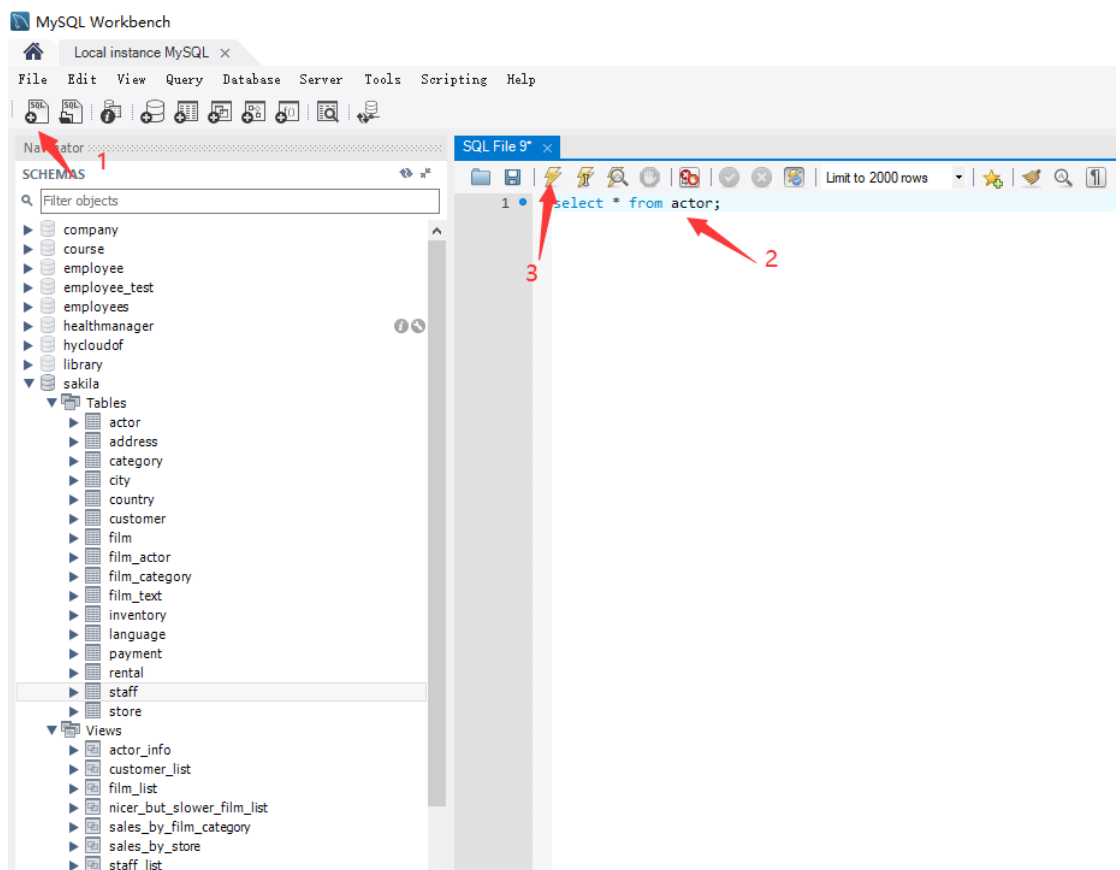
如下图，直接点 Run：



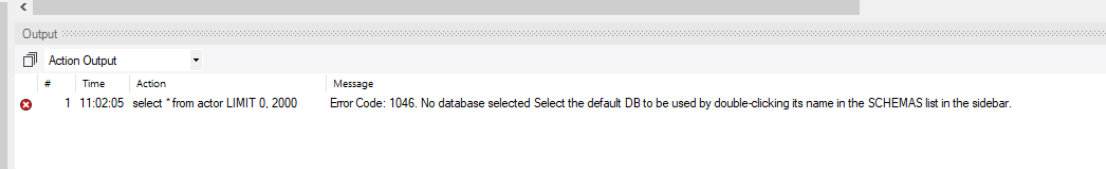
执行完毕如下图：



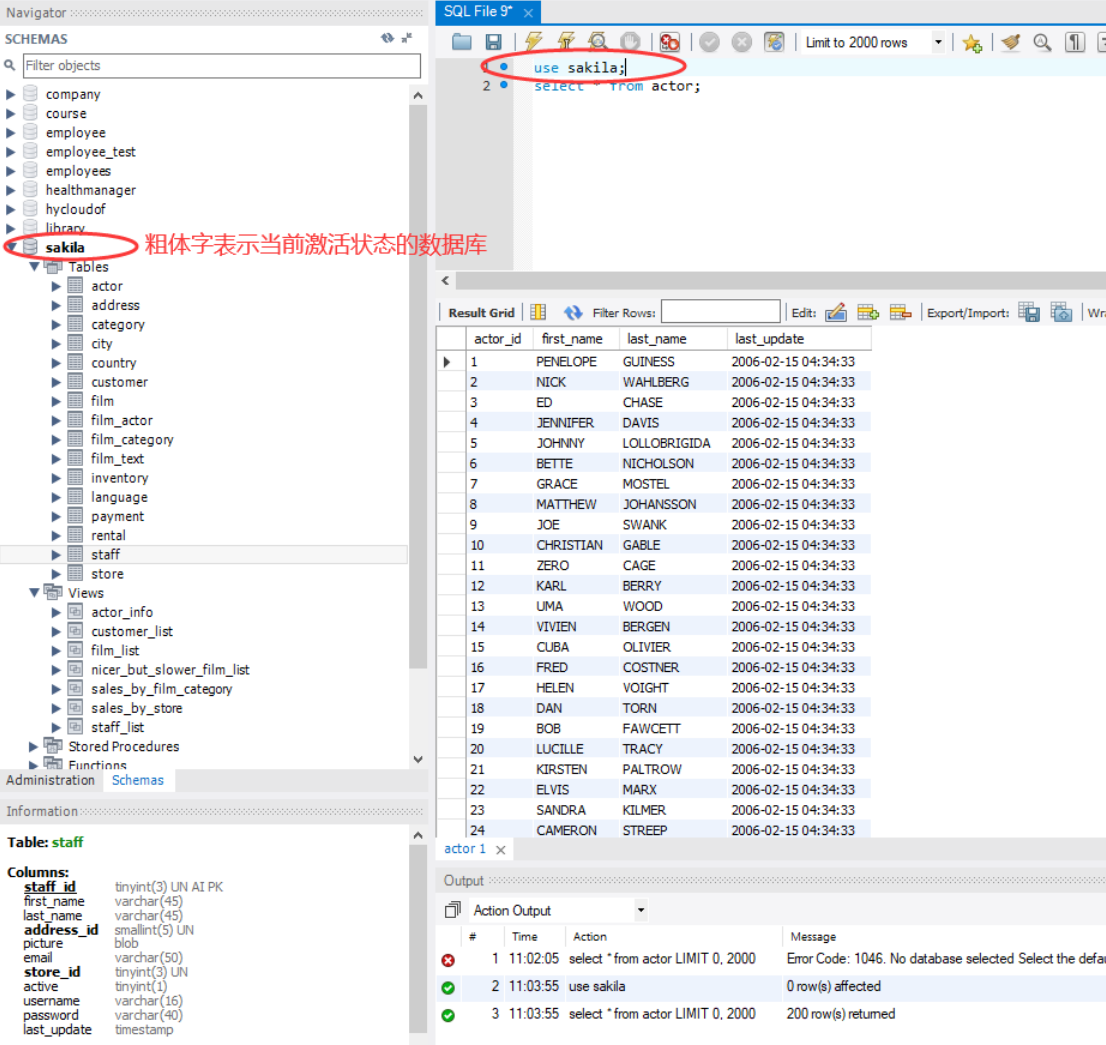
如果本步骤报错可参考 5.2.3 节。  
正常导入数据后可以在数据库中查到数据：



立马给个报错：



这是因为我们没有选定数据库，可以双击左边 SCHEMAS 栏的 sakila，让它变成粗体字，也可以在 select 之前加多一句 “use sakila;”：



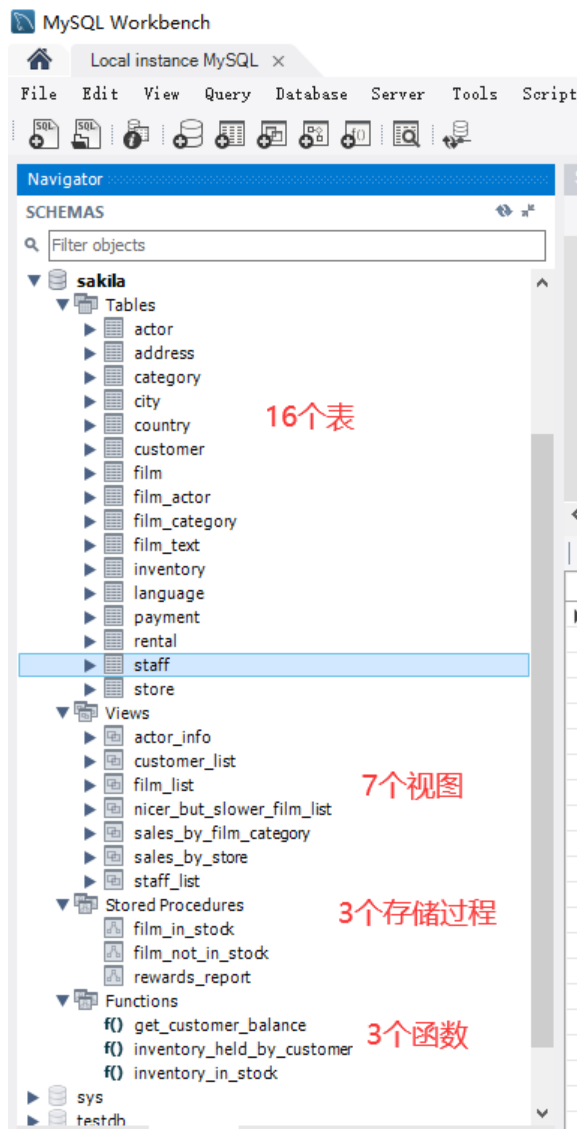
### 4.1.3检查 sakila 数据库中数据

导入以后请自查 16 个表的记录条数是否正确：

表名	记录数
actor	200
address	603
category	16

city	600
country	109
customer	599
film	1000
film_actor	5462
film_category	1000
film_text	1000
inventory	4581
language	6
payment	16049
rental	16044
staff	2
store	2

## 4.2 熟悉 sakila 数据库并回答问题



要搞懂 sakila 数据库，**有效方法**是参考 sakila 数据库的官方文档，并结合 sakila 数据库的模型文件。

1) 打开官方文档：

<https://dev.mysql.com/doc/sakila/en/sakila-installation.html>



MySQL Server MySQL Enterprise Workbench InnoDB Cluster MySQL NDB Cluster Connectors More

Sakila Sample Database / Structure

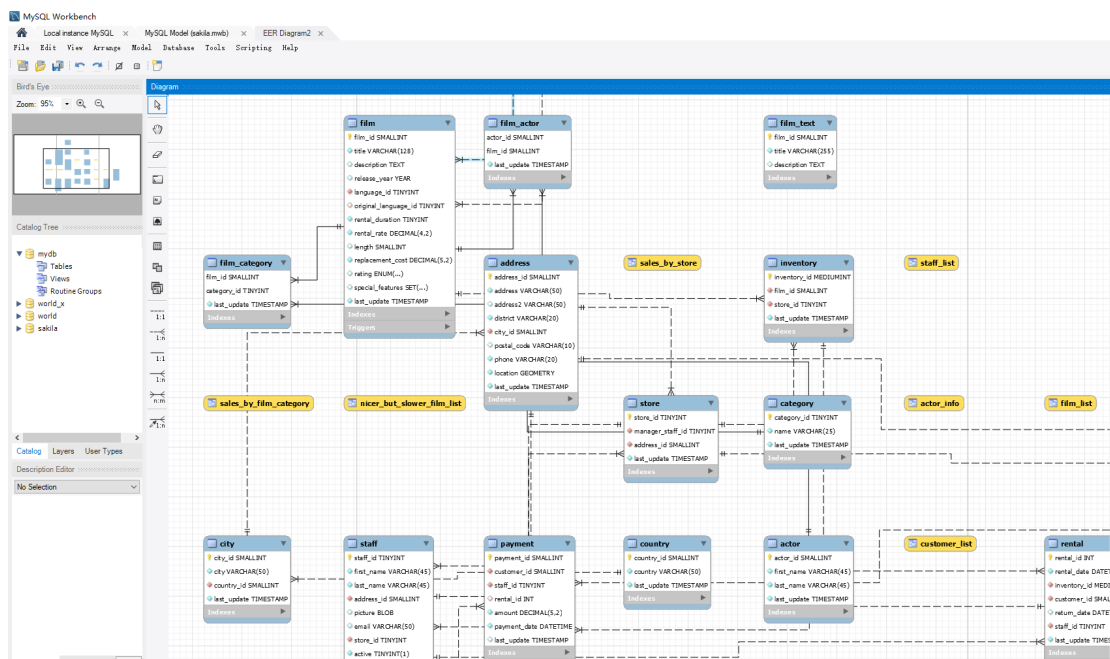
## 5 Structure

5.1 Tables  
5.2 Views  
5.3 Stored Procedures  
5.4 Stored Functions  
5.5 Triggers

The following diagram provides an overview of Sakila sample database structure. The diagram source file (for use with MySQL Workbench) is in distribution and is named `sakila.mwb`.

**Figure 1 The Sakila Schema**

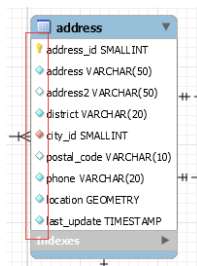
## 2) 用 MySQL Workbench 打开 sakila.mwb 文件:



注意：如果打开时报错 `mysql Error unserializing GRT data string too long`, 重启 MySQL Workbench 再打开就可以了。

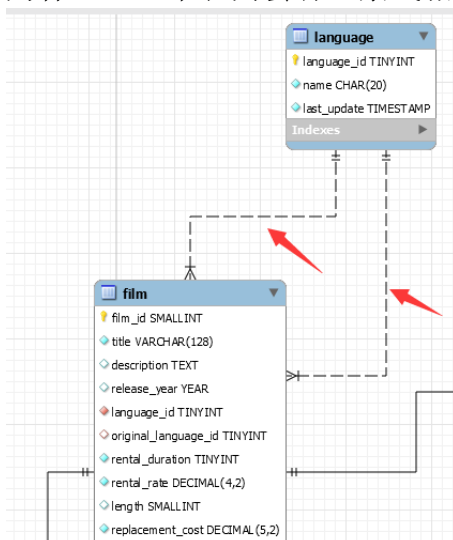
接下来我们边了解这个数据库边回答以下问题（答案请写在实验报告）：

1. sakila.mwb 模型中，表结构里每个字段前面的小标记分别表示什么意思？



标记	意义

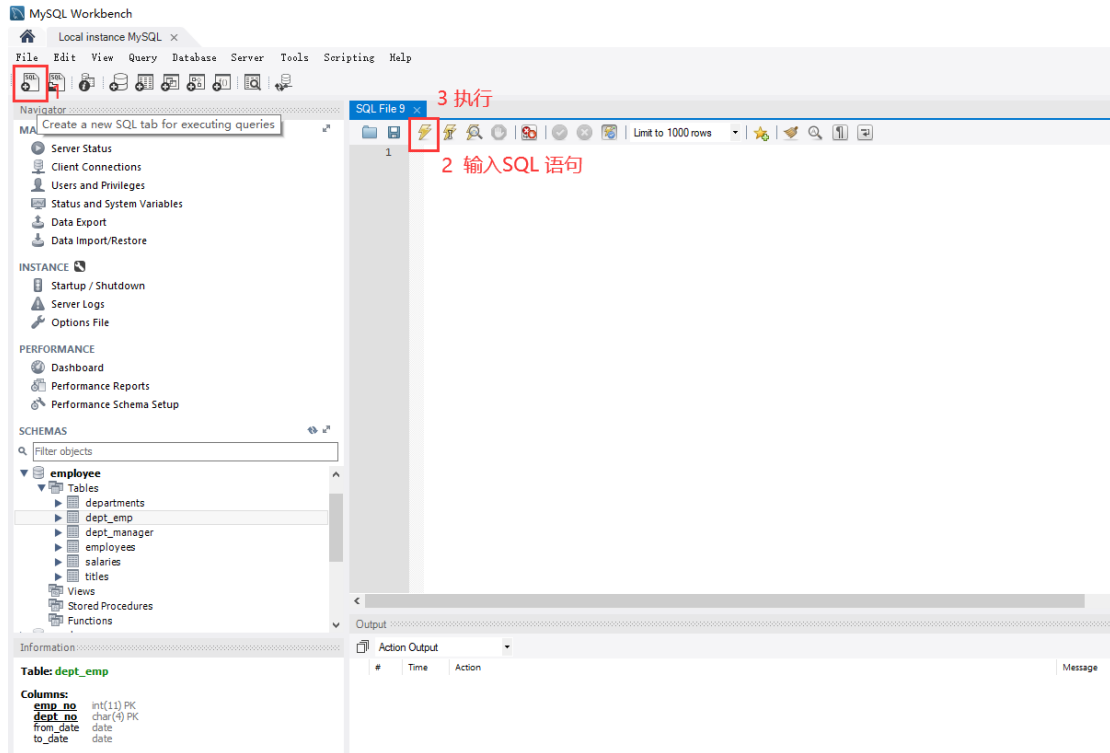
2. 图中哪部分体现影片-演员关系？换句话说，如果要找出演某个影片的演员名字，访问哪几张表可以获得信息？
3. 如果已知某个顾客姓名，要找到他租借的所有影片名，需要访问哪几张表？
4. 为什么这 2 个表间会有 2 条线相连？



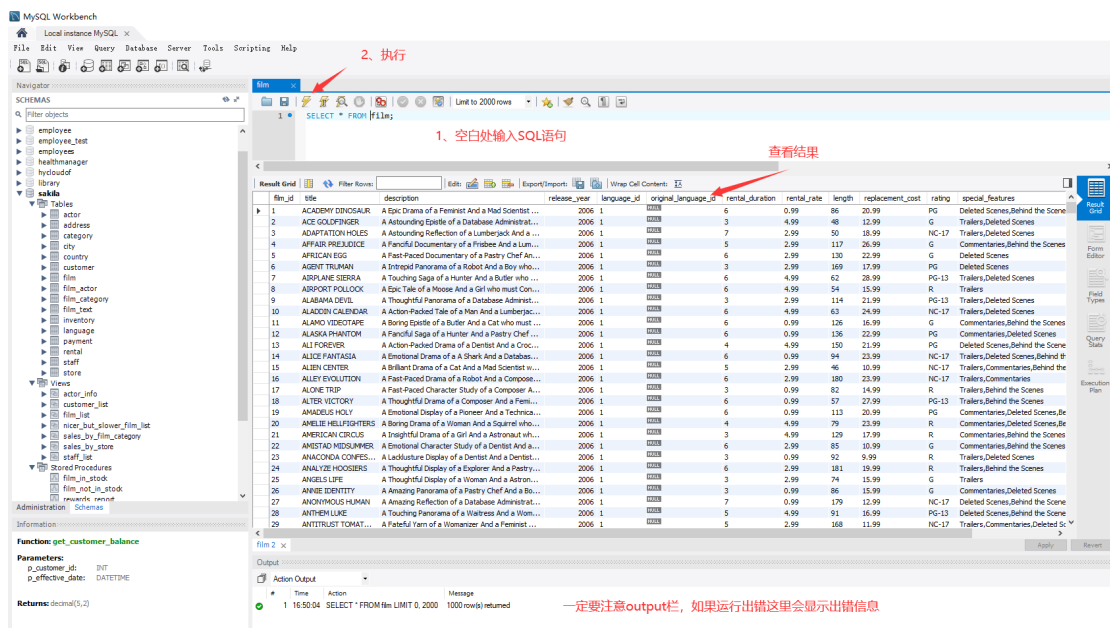
## 4.3 数据的增删改查

### 3. 1. 1 跟着完成以下内容

- 1、增、删、改、查操作都需要先打开一个 SQL File 页面：

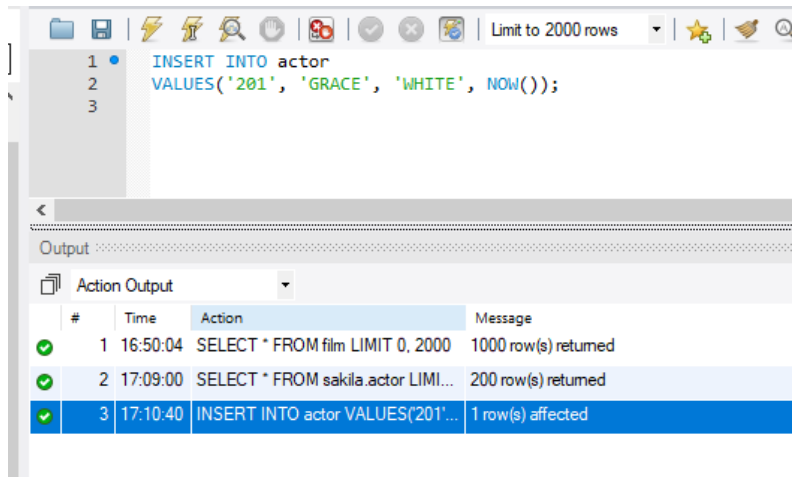


2、我们查询 film 表的数据，可以在上图空白处输入：  
SELECT \* FROM film;

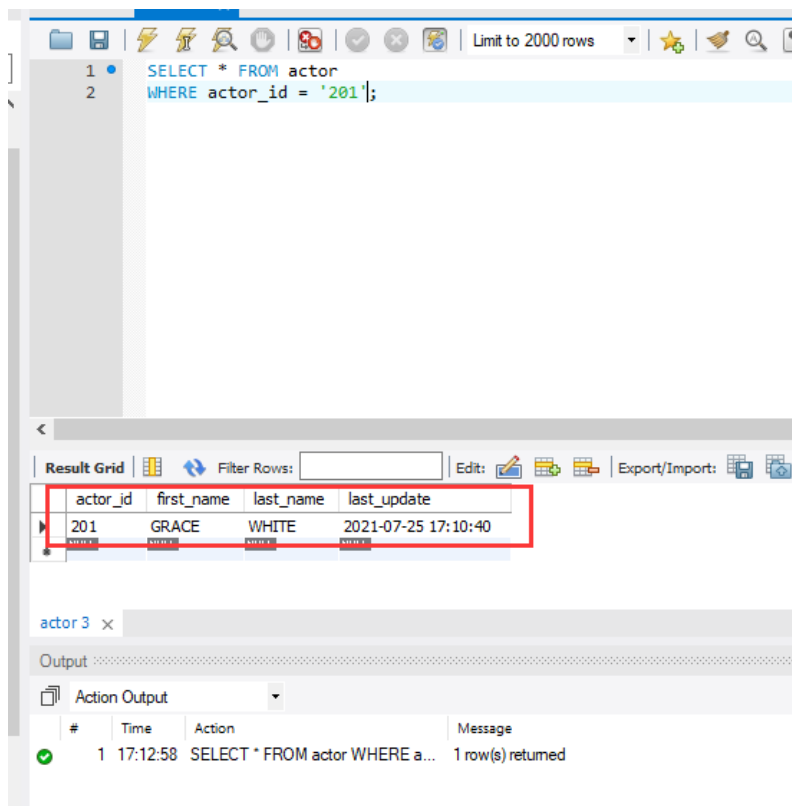


3、我们尝试在 actor 表增加一条数据，可以在上图空白处输入：  
INSERT INTO actor  
VALUES ('201', 'GRACE', 'WHITE', NOW());

然后点击执行按钮：



可以检查一下新增数据：



请思考：

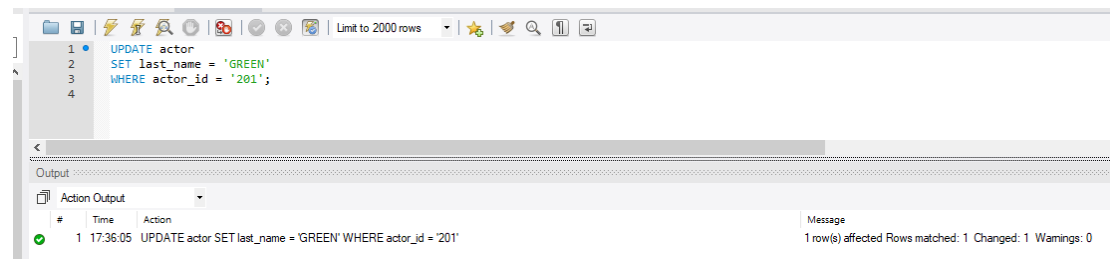
- 如果 insert 一条数据到 actor 表，但 actor\_id 和已有数据重复，会发生什么？同学们请自己尝试一下。
- insert 语句还用了一个函数 NOW()，是做什么的呢？请同学们 **在实验报告里** 详细回答。

4、我们尝试在 actor 表**修改**刚才新增的数据，可以在上图空白处输入：

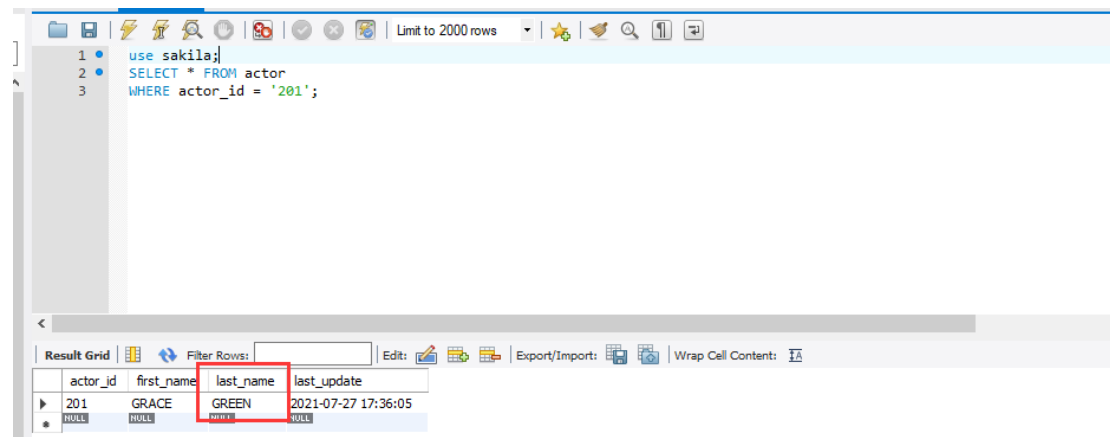
```
UPDATE actor
SET lastname = 'GREEN'
```

Where actor\_id = '201';

然后点击执行按钮：



可以检查一下修改数据：

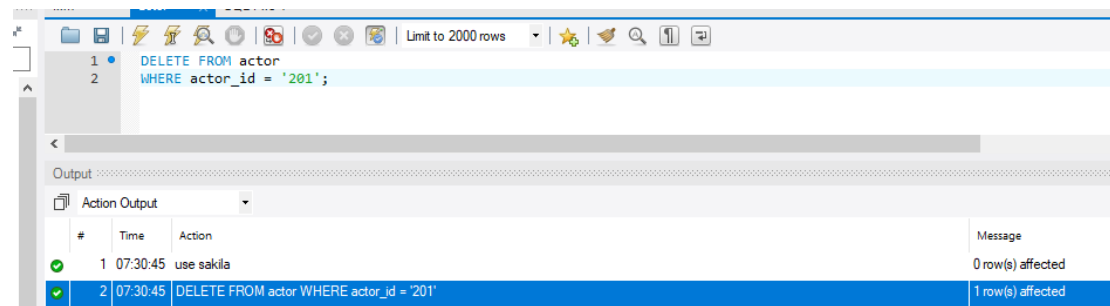


同学们注意一下 last\_update 发生了什么？请思考一下为什么。

5、我们尝试在 actor 表删除刚才新增的数据，可以在上图空白处输入：

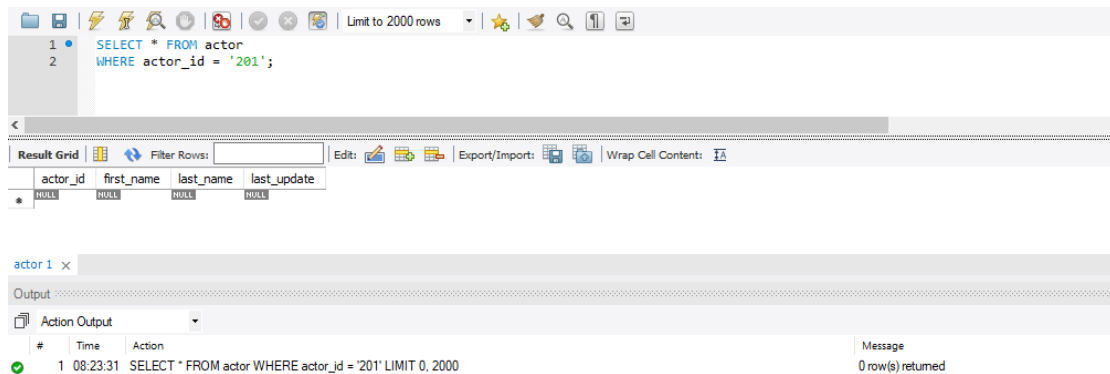
```
DELETE FROM actor
WHERE actor_id = '201';
```

然后点击执行按钮：



下面 output 窗口可以看到执行成功。

可以检查一下数据，可以看到“201”这条数据已被删除：



### 3.1.2 自己写 SQL 语句完成以下内容

注意边做边填写实验报告（截图时需要体现 SQL 语句、执行结果、Output 窗口）：

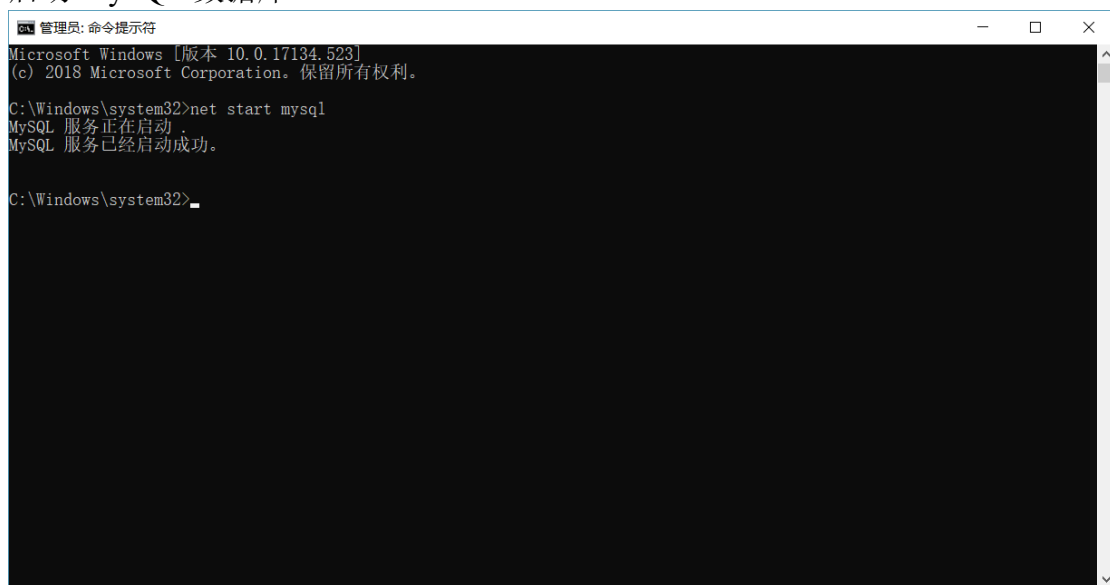
- 1、请列出所有商店的详细地址，显示商店 id，商店地址，所在区域，所在城市，所在国家；
- 2、哪些演员出演过影片《ROCKY WAR》？请列出他的姓和名；
- 3、找出租 DVD 花费最高的前 5 名，请列出他们的 first\_name, last\_name 和每个人花费的金额；
- 4、哪个影片获得了总体最高的租金？请列出影片 id、影片名、总租金；
- 5、哪个演员出演的电影超过 35 部？请列出演员 id、演员名、出演的电影数；
- 6、请找出没有租借过电影《TELEGRAPH VOYAGE》的顾客姓名；
- 7、查询演过《ELEPHANT TROJAN》和《SPLASH GUMP》这两部电影的演员，列出其姓名；
- 8、统计每种类型的影片数，显示类型编号、类型名称、该类型影片数；
- 9、有哪些影片是 2 个商店都有库存的？
- 10、查询单次租借影片时间最长的 6 位客户，列出其 first\_name、last\_name 和当次租借时长；
- 11、在 customer 表中新增一条数据，注意 customer 表与其他表的关系；
- 12、修改刚才在 customer 表中新增的那条数据；
- 13、删除第 11 步新增的那条数据。

## 5 附录

### 5.1 MySQL 手册（命令行方式）

#### 5.1.1 启动 MySQL 服务

管理员身份运行 cmd，执行：net start mysql  
启动 MySQL 数据库。



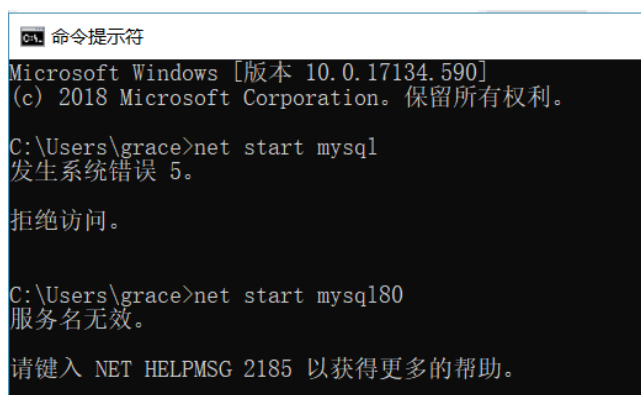
```
管理员: 命令提示符
Microsoft Windows [版本 10.0.17134.523]
(c) 2018 Microsoft Corporation. 保留所有权利。

C:\Windows\system32>net start mysql
MySQL 服务正在启动。
MySQL 服务已经启动成功。

C:\Windows\system32>_
```

注意两点：

- a) 用管理员用户执行 cmd；
  - b) net start mysql，这个 mysql 是要启动的数据库服务，对应的是服务名。
- 如果遇到如下报错：



```
命令提示符
Microsoft Windows [版本 10.0.17134.590]
(c) 2018 Microsoft Corporation. 保留所有权利。

C:\Users\grace>net start mysql
发生系统错误 5。
拒绝访问。

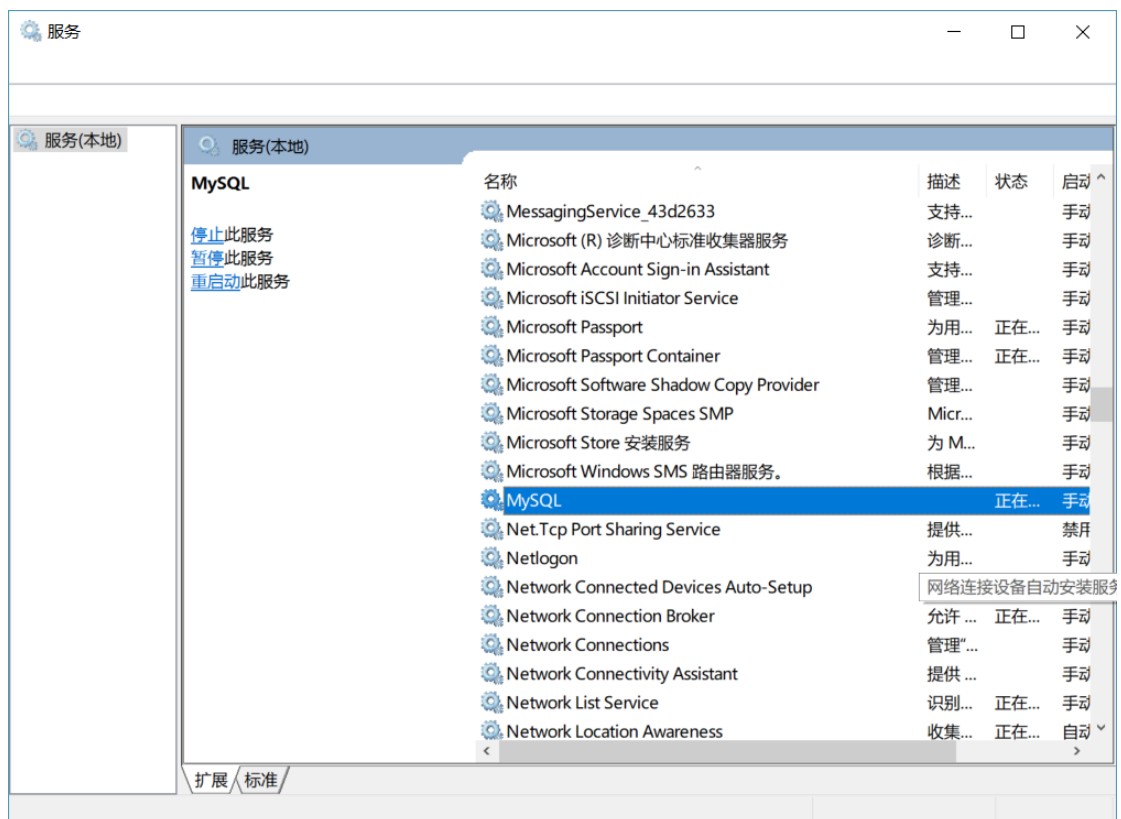
C:\Users\grace>net start mysql80
服务名无效。

请键入 NET HELPMSG 2185 以获得更多的帮助。
```

“拒绝访问”是因为用的不是管理员账号执行。

“服务名无效”请按下面方法查看服务名（服务名是在安装 mysql 的时候输入的，所以每个机器可能会有不同）。

打开控制面板，搜索服务，打开服务列表如下图：



### 5.1.2 连接 MySQL 服务器

在命令行下输入 `mysql -h localhost -u root -p`

当显示“Enter password:”时，输入 root 密码 **mysql**

```
C:\windows\system32>mysql -h localhost -u root -p
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.13 MySQL Community Server - GPL

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

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql>
```



### 5.1.3 查看 MySQL 中有哪些数据库

mysql> show databases; (注意 SQL 语句结尾的分号!)

```
mysql> show databases;
+-----+
| Database |
+-----+
| company  |
| course   |
| employee |
| healthmanager |
| hycloudof |
| information_schema |
| library  |
| mysql    |
| performance_schema |
| sakila   |
| sys      |
| testdb   |
| testforstu |
| world    |
+-----+
15 rows in set (0.01 sec)
```

### 5.1.4 选定数据库

mysql> use employee;

### 5.1.5 查看当前在的数据库

mysql> select database();

```
mysql> use employee;
Database changed
mysql> select database();
+-----+
| database() |
+-----+
| employee   |
+-----+
1 row in set (0.00 sec)
```

### 5.1.6 查看当前数据库中的表

mysql> show tables;

```
mysql> show tables;
+-----+
| Tables_in_employee |
+-----+
| departments        |
| dept_emp            |
| dept_manager        |
| employees            |
| salaries            |
| titles              |
| view_employees_d002 |
+-----+
7 rows in set (0.00 sec)
```

### 5.1.7 查看某个表的表结构

比如想看 salaries 表的表结构，输入

mysql> describe salaries;

```
mysql> describe salaries;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| emp_no     | int(11)   | NO   | PRI | NULL    |       |
| salary     | int(11)   | YES  |     | NULL    |       |
| from_date  | date      | NO   | PRI | NULL    |       |
| to_date    | date      | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)
```

### 5.1.8 使用 help 命令来了解其他命令和变量类型等的含义

mysql> help;

mysql> help use;

```
mysql> help use;
Name: 'USE'
Description:
Syntax:
USE db_name

The USE db_name statement tells MySQL to use the db_name database as
the default (current) database for subsequent statements. The database
remains the default until the end of the session or another USE
statement is issued:

USE db1;
SELECT COUNT(*) FROM mytable;    # selects from db1.mytable
USE db2;
SELECT COUNT(*) FROM mytable;    # selects from db2.mytable

URL: http://dev.mysql.com/doc/refman/8.0/en/use.html
```

### 5.1.9 取消命令

若要取消一条正在编辑命令，键入\c 并回车

```
mysql> select count(*) from \c
mysql> _
```

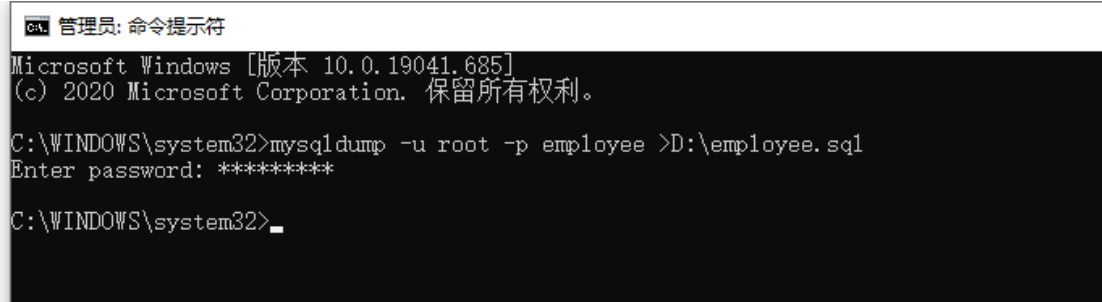
### 5.1.10 断开 MySQL 服务器连接

mysql> quit 或 mysql> exit

### 5.1.11 导出数据库

用 mysqldump 是将指定数据库的表结构、视图、数据等所有信息都一并导出。在命令行方式下执行：

mysqldump -u root -p employee > D:\employee.sql



```
管理员: 命令提示符
Microsoft Windows [版本 10.0.19041.685]
(c) 2020 Microsoft Corporation. 保留所有权利。

C:\WINDOWS\system32>mysqldump -u root -p employee >D:\employee.sql
Enter password: *****

C:\WINDOWS\system32>_
```

注意：该命令在 cmd 中执行，不是在 mysql 下执行。

打开导出的 sql 文件

```
employee.sql x
1  -- MySQL dump 10.13 Distrib 5.0.13, for Win64 (x86_64)
2  --
3  -- Host: localhost    Database: employee
4  --
5  -- Server version 5.0.13
6  --
7  /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
8  /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
9  /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
10 SET NAMES utf8mb4 ;
11 /*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
12 /*!40103 SET TIME_ZONE='+00:00' */;
13 /*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
14 /*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
15 /*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
16 /*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
17
18 --
19 -- Table structure for table 'departments'
20 --
21
22 DROP TABLE IF EXISTS `departments`;
23 /*!40101 SET @saved_cs_client = @@character_set_client */;
24 SET character_set_client = utf8mb4 ;
25 CREATE TABLE `departments` (
26   `dept_no` char(4) NOT NULL,
27   `dept_name` varchar(40) NOT NULL,
28   PRIMARY KEY (`dept_no`),
29   UNIQUE KEY `dept_name` (`dept_name`)
30 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
31 /*!40101 SET character_set_client = @saved_cs_client */;
32
33 --
34 -- Dumping data for table 'departments'
35 --
36
37 LOCK TABLES `departments` WRITE;
38 /*!40000 ALTER TABLE `departments` DISABLE KEYS */;
39 INSERT INTO `departments` VALUES ('d005','Customer Service'),('d005','Development'),('d002','Finance'),('d003','Human Resources'),('d001','Marketing'),('d004','Production'),('d006','Quality Management');
40 /*!40000 ALTER TABLE `departments` ENABLE KEYS */;
41 UNLOCK TABLES;
42
43 --
44 -- Table structure for table 'dept_emp'
45 --
46
47 DROP TABLE IF EXISTS `dept_emp`;
48 /*!40101 SET @saved_cs_client = @@character_set_client */;
49 SET character_set_client = utf8mb4 ;
50 CREATE TABLE `dept_emp` (

```

### 5.1.12 创建数据库

mysql> CREATE DATABASE employee\_test;

### 5.1.13 导入数据库

例如导入上一步准备好的 sql 文件，首先新建一个 employee\_test 数据库，然后在 cmd 窗口执行：

mysql -u root -p employee\_test< [你的存放路径]employee.sql

```
C:\WINDOWS\system32>mysql -u root -p employee_test< D:\employee.sql
Enter password: *****
```

注意：该命令在 cmd 中执行，不是在 mysql 下执行。

检查是否导入成功：

```
mysql> use employee_test;
Database changed
mysql> show tables;
+-----+
| Tables_in_employee_test |
+-----+
| departments              |
| dept_emp                 |
| dept_manager             |
| employees                |
| salaries                 |
| titles                   |
| view_employees_d002      |
+-----+
7 rows in set (0.01 sec)
```

```
mysql> select count(*) from employees;
+-----+
| count(*) |
+-----+
|      906 |
+-----+
1 row in set (0.01 sec)
```

我们可以看到表结构和数据均已成功导入。

#### 5.1.14 使用数据文件导入数据

如果我们拿到的是数据文件，要导入事先建好的 employee 数据库的几个表里，我们需要以下步骤：

1) 先用命令看一下 MySQL 的 secure-file-priv 特性：

```
mysql> SHOW VARIABLES LIKE "secure_file_priv";
+-----+-----+
| Variable_name | Value |
+-----+-----+
| secure_file_priv | C:\ProgramData\MySQL\MySQL Server 8.0\Uploads\ |
+-----+-----+
1 row in set, 1 warning (0.01 sec)
```

注意：MYSQL 的新特性 secure\_file\_priv 用来限制 LOAD DATA, SELECT ... OUTFILE, and LOAD\_FILE()的数据目录，就是数据导入和导出到哪个路径。

secure_file_priv	意义
null	不允许导入 导出
路径	限制导入 导出只能发生在指定路径下
没有具体值	不对导入 导出做限制

如果想修改 secure\_file\_priv 的值，

windows 下：修改 my.ini 在[mysqld]内加入 secure\_file\_priv = [你想设置的路径]

linux 下：修改 my.cnf 在[mysqld]内加入 secure\_file\_priv = [你想设置的路径]

然后重启 mysql，再查询 secure\_file\_priv。

**注意：**

C:\ProgramData 目录是隐藏目录，你可能需要设置显示隐藏目录才能看到它。如果你想修改 my.ini 文件，每个版本的 my.ini 可能位置会有不同，8.0 版本在 C:\ProgramData\MySQL\MySQL Server 8.0 下。

- 2) 将要导入的数据文件放在 C:\ProgramData\MySQL\MySQL Server 8.0\Uploads\下：

电脑 > Windows (C:) > ProgramData > MySQL > MySQL Server 8.0 > Uploads			
名称	修改日期	类型	大小
departments.txt	2021/1/13 10:17	文本文档	1 KB
dept_emp.txt	2021/1/13 11:49	文本文档	34 KB
dept_manager.txt	2021/1/13 10:17	文本文档	1 KB
employees.txt	2021/1/13 15:31	文本文档	42 KB
salaries.txt	2021/1/13 15:33	文本文档	294 KB
titles.txt	2021/1/13 15:33	文本文档	53 KB

- 3) 执行

```
mysql> load data infile "C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\departments.txt" into table employee.departments;
```

```
mysql> load data infile "C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\departments.txt" into table employee.departments;
Query OK, 9 rows affected (0.11 sec)
Records: 9 Deleted: 0 Skipped: 0 Warnings: 0
```

如果导入的时候遇到问题可以看一下第 5.3 节常见问题里面的整理的。

## 5.2 常见问题

### 5.2.1 'mysql' 不是内部或外部命令，也不是可运行的程序或批处理文件

```
C:\Windows\system32>mysql -u root -p
'mysql' 不是内部或外部命令，也不是可运行的程序或批处理文件。
C:\Windows\system32>
```

解决办法有两个，**方法一：**

找到 mysql 的安装目录下的 bin 目录，比如实验室机器上是 C:\Program Files\MySQL\MySQL Server 8.0\bin，用管理员身份打开 cmd 控制台，进入 bin 目录，然后执行 mysql：

```
管理员: 命令提示符 - mysql -u root -p
Microsoft Windows [版本 10.0.19043.1165]
(c) Microsoft Corporation。保留所有权利。

C:\WINDOWS\system32>cd C:\Program Files\MySQL\MySQL Server 8.0\bin

C:\Program Files\MySQL\MySQL Server 8.0\bin>mysql
ERROR 1045 (28000): Access denied for user 'ODBC'@'localhost' (using password: NO)

C:\Program Files\MySQL\MySQL Server 8.0\bin>mysql -u root -p
Enter password:
```

## 方法二：

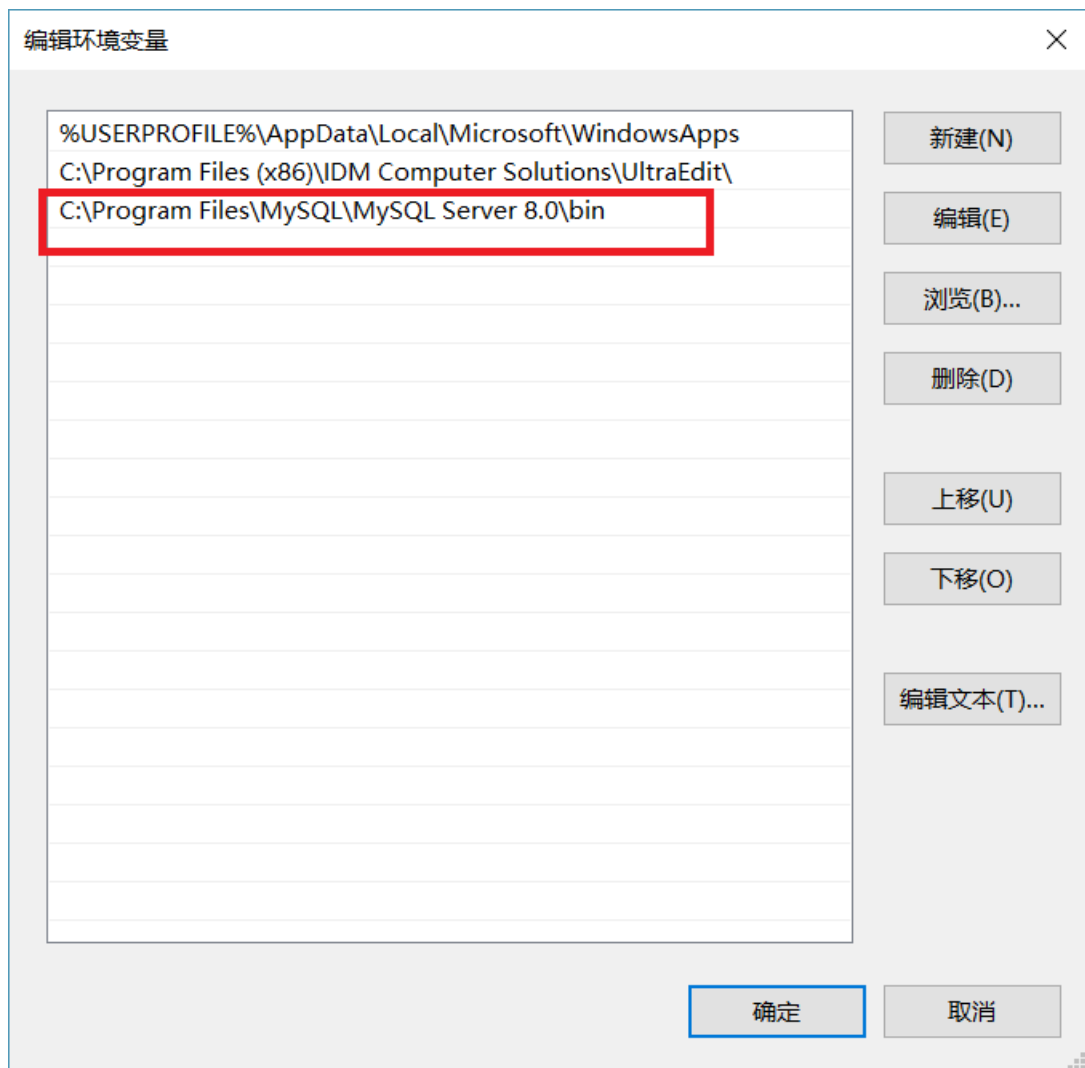
- 1、打开电脑的属性，找到高级系统设置



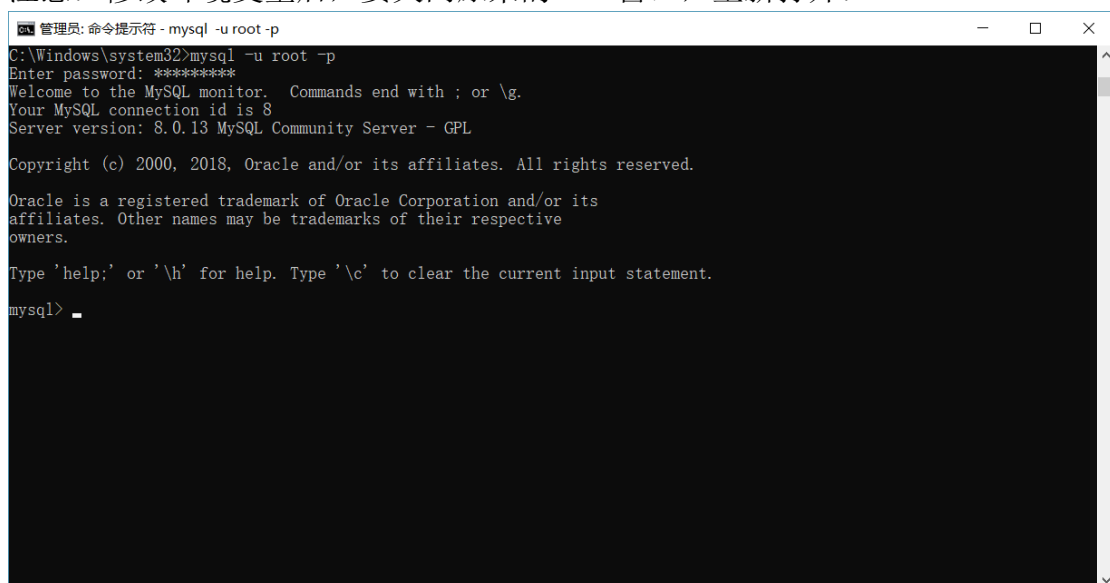


2、修改 Path，增加 MySQL 安装路径下的 bin

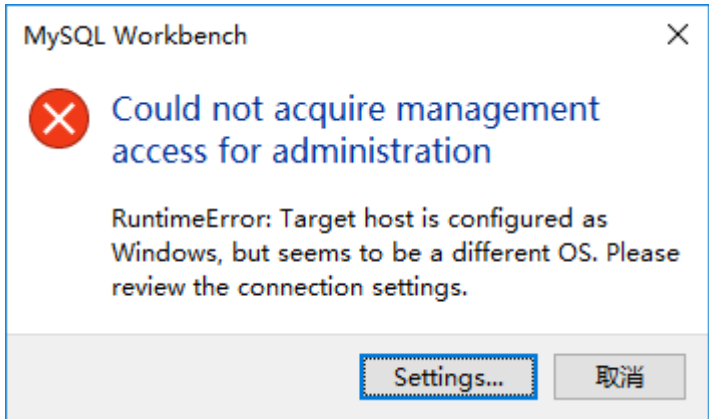




注意：修改环境变量后，要关闭原来的 cmd 窗口，重新打开：

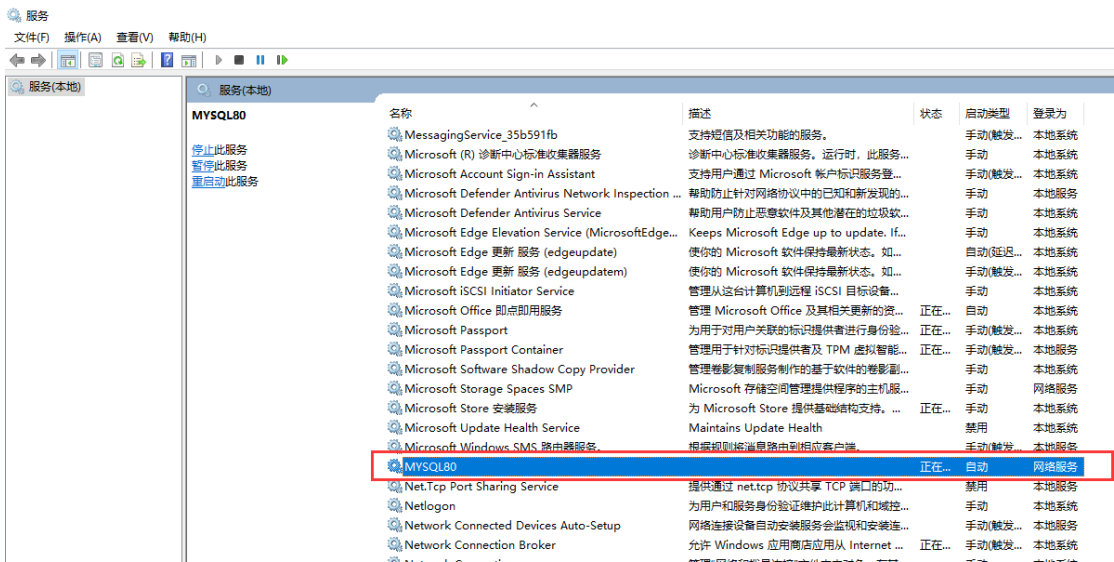


5.2.2 通过 MySQL Workbench 启动服务报错



解决方法 2 选 1:

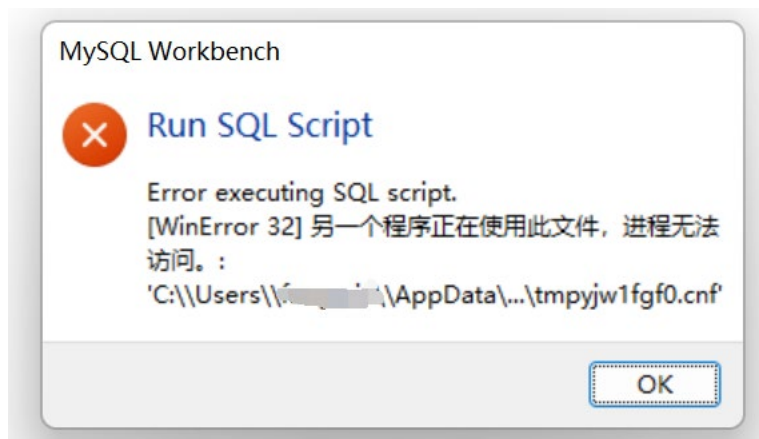
方法一： 电脑里打开“服务”，找到 MYSQL80，右键→启动。



方法二： 用命令行方式启动 MySQL 服务，详见本手册 4.1.1 节。

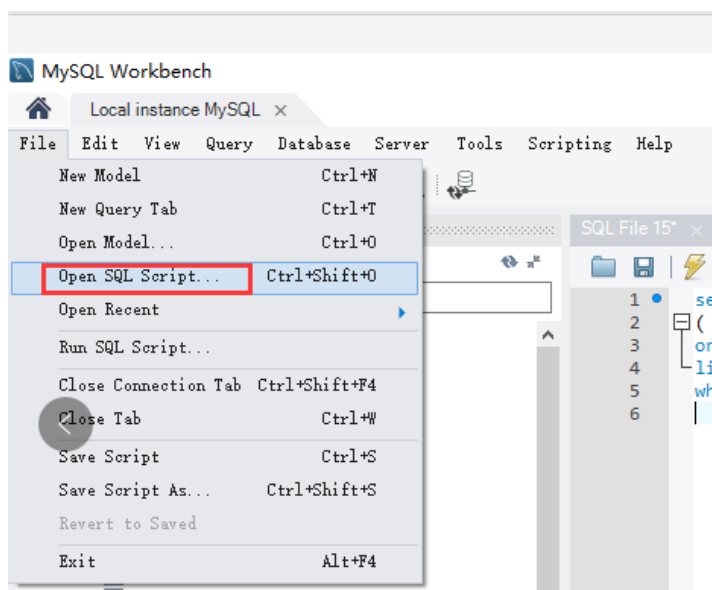
5.2.3 Run SQL Script: Error executing SQL script.

在 Workbench 里用 Run SQL Script 时报错:

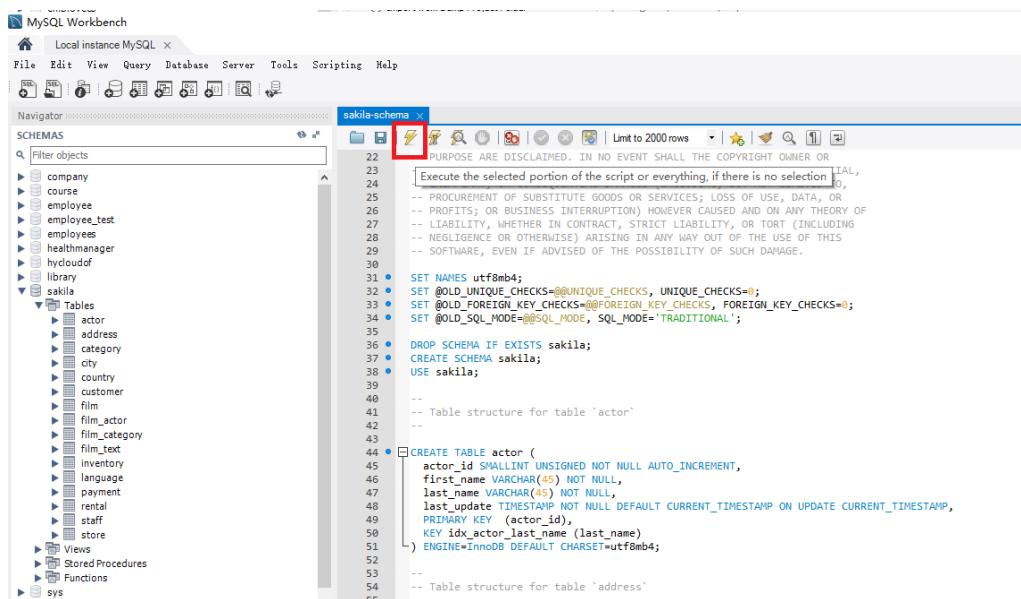


解决办法有 2 个。方案一：

1、点击 File→Open SQL Script...

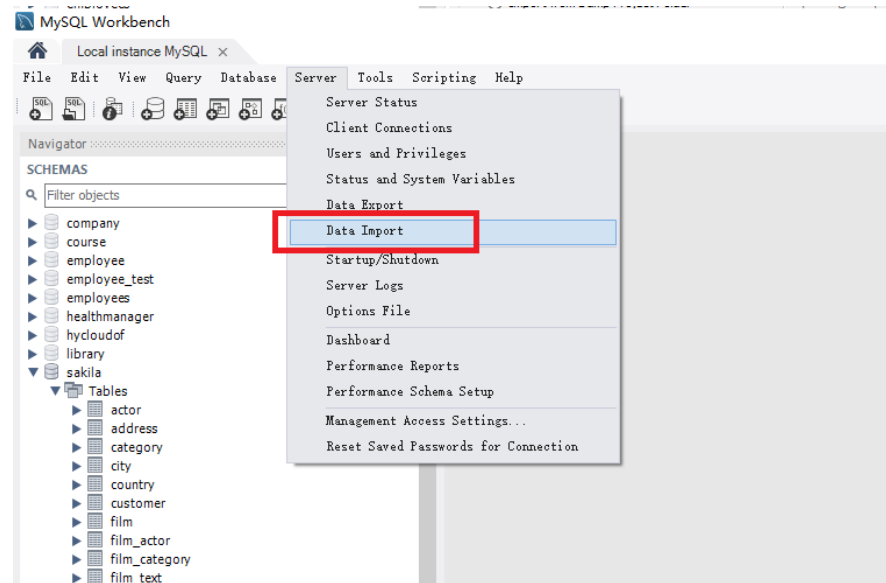


2、选择你要执行的 sql 文件，打开后点击执行按钮：

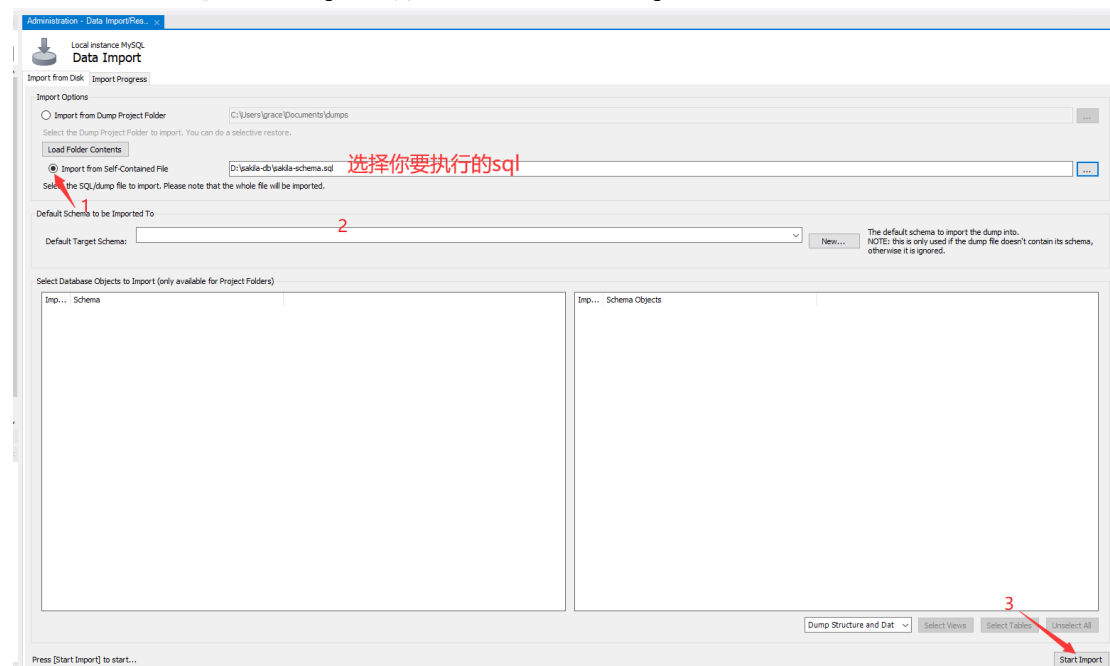


## 方案二:

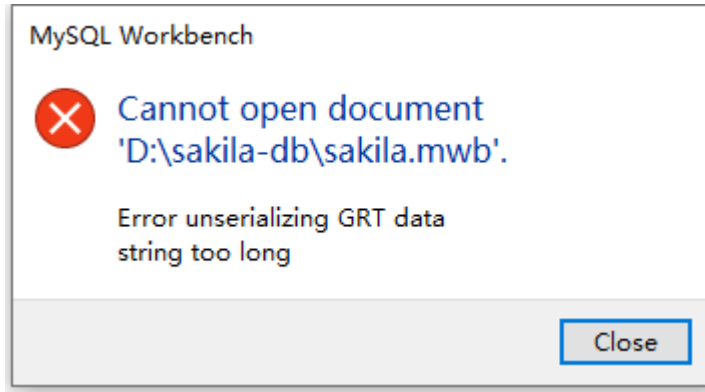
### 1、点击 Sever→Data Import:



### 2、选择你要执行的 sql 文件，点击 Start Import:



#### 5.2.4 打开 sakila.mwb 的时候报 Error unserializing GRT data string too long



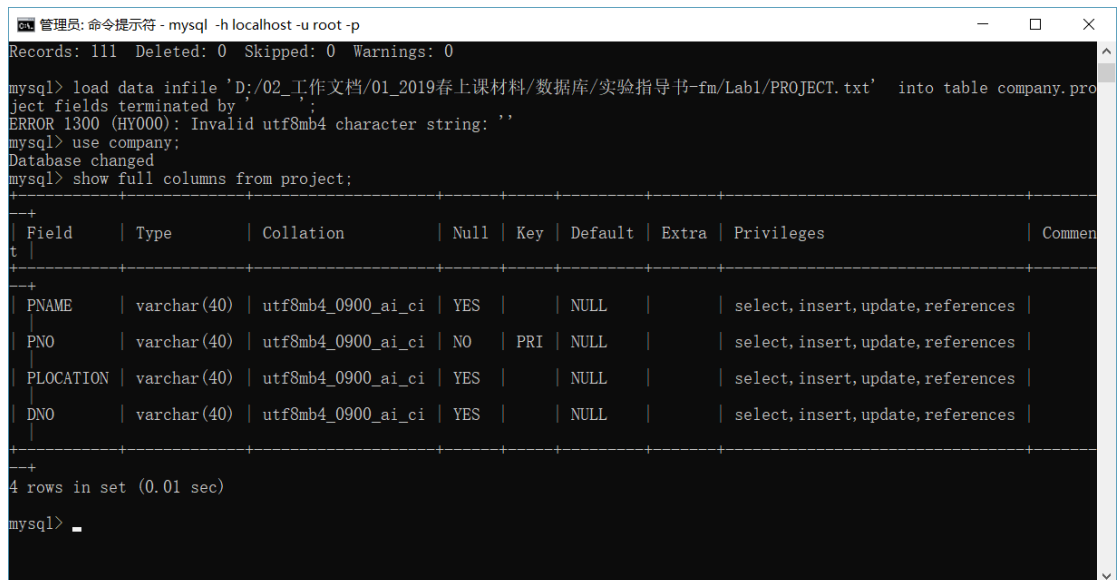
解决办法：重启 MySQL Workbench。

#### 5.2.5 ERROR 1300 (HY000): Invalid utf8mb4 character string: ''

Load data 报错

```
mysql> load data infile 'D:/02_工作文档/01_2019春上课材料/数据库/实验指导书-fm/Lab1/PROJECT.txt' into table company.pro
ject fields terminated by ',';
ERROR 1300 (HY000): Invalid utf8mb4 character string: ''
```

查看这个 project 表的列编码格式：



检查数据文件，另存为 UTF-8。

如果是中文和英文夹杂的数据文件，导入后可能出现另外一个问题：正常导入，并在 MySQL Workbench 中看似正常，如下图：

2 • `select * FROM company.project;`

Result Grid Filter Rows: Edit:

PNAME	PNO	PLOCATION	DNO
工程1	P1	黑龙江	000
工程10	P10	江苏	100
工程2	P2	江苏	001
工程3	P3	黑龙江	001
工程4	P4	天津	100
SQLProject	P5	北京	100
工程6	P6	杭州	011
工程7	P7	浙江	001
工程8	P8	上海	010
工程9	P9	黑龙江	011
NULL	NULL	NULL	NULL

但是用 mysql 命令行查询结果如下图：

```
mysql> select * from project;
```

PNAME	PNO	PLOCATION	DNO
?工程1	P1	黑龙江	000
工程10	P10	江苏	100
工程2	P2	江苏	001
工程3	P3	黑龙江	001
工程4	P4	天津	100
SQLProject	P5	北京	100
工程6	P6	杭州	011
工程7	P7	浙江	001
工程8	P8	上海	010
工程9	P9	黑龙江	011

10 rows in set (0.00 sec)

看出问题了吗？第一条记录的第一个字段前有个问号。

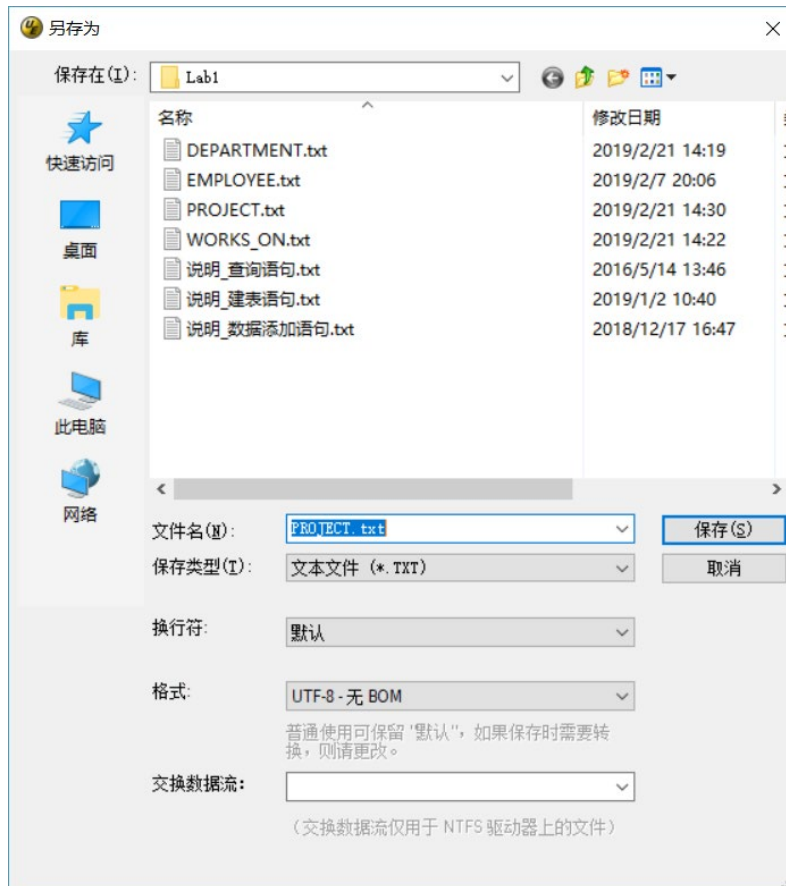
这是怎么回事？

UTF-8 编码的文件开头有三个 bytes，存储的是 0xef, 0xbb, 0xbf，用来表明这个文件是 UTF-8 文件，这就是 UTF-8 的 BOM 头。正是因为这个 mysql 在做 load data 的时候不能正确分辨文件的数据部分（参考

<https://bugs.mysql.com/bug.php?id=10573>）。怎么解决？请看下一节。

## 5.2.6 BOM(of UTF-8) is not correctly handled by LOAD DATA

用 UltraEdit 打开数据文件，另存为 UTF-8 无 BOM：



重新 load data，然后检查数据：

```

管理员: 命令提示符 - mysql -h localhost -u root -p

工程7      P7      浙江      001
工程8      P8      上海      010
工程9      P9      黑龙江    011
-----
10 rows in set (0.00 sec)

mysql>
mysql> load data infile 'D:/02_工作文档/01_2019春上课材料/数据库/实验指导书-fm/Lab1/PROJECT.txt' into table company.pro
ject fields terminated by
;
Query OK, 10 rows affected (0.05 sec)
Records: 10 Deleted: 0 Skipped: 0 Warnings: 0

mysql> select * from project;
+-----+-----+-----+-----+
| PNAME | PNO | PLOCATION | DNO |
+-----+-----+-----+-----+
| 工程1 | P1  | 黑龙江 | 000 |
| 工程10 | P10 | 江苏 | 100 |
| 工程2 | P2  | 江苏 | 001 |
| 工程3 | P3  | 黑龙江 | 001 |
| 工程4 | P4  | 天津 | 100 |
| SQLProject | P5 | 北京 | 100 |
| 工程6 | P6  | 杭州 | 011 |
| 工程7 | P7  | 浙江 | 001 |
| 工程8 | P8  | 上海 | 010 |
| 工程9 | P9  | 黑龙江 | 011 |
+-----+-----+-----+-----+
10 rows in set (0.00 sec)

mysql>

```

干净、漂亮！

### 5.2.7 load data 报错 ERROR 1062 (23000): Duplicate entry '' for key 'PRIMARY'

```
mysql> LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/DEPARTMENT.txt' INTO TABLE department fields terminated by ' ' ;
ERROR 1062 (23000): Duplicate entry '' for key 'PRIMARY'
mysql> LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/DEPARTMENT.txt' INTO TABLE department fields terminated by ' ' ;
Query OK, 5 rows affected (0.07 sec)
Records: 5 Deleted: 0 Skipped: 0 Warnings: 0
```

肉眼看不出上下 2 个命令的区别，但为什么上面那个命令报错，下面的执行成功呢？实际上，上面命令是 copy 的，所以 fields terminated by ' '，两个引号之间是空格；下面命令的引号之间是键盘敲入的 Tab 键。

### 5.2.8 load data 报错 ERROR 29 (HY000): File '\*\*\*\*\*' not found (OS errno 2 - No such file or directory)

```
mysql> LOAD DATA INFILE "D:\02_工作文档\01_2019春上课材料\数据库\实验指导书-fm\Lab1\DEPARTMENT2.txt" INTO TABLE department;
ERROR 29 (HY000): File 'D:\' not found (OS errno 2 - No such file or directory)
```

这种是路径问题，请注意的：

- a) 注意斜线的方向！
- b) 注意''之间不要有多余的空格。

### 5.2.9 load data 没有报错，但数据不正确

Load data 看似导入正确执行，查询查询貌似没有问题：



```

mysql> LOAD DATA INFILE "D:/02_工作文档/01_2019春上课材料/数据库/实验指导书-fm/Lab1/EMPLOYEE.txt" INTO TABLE employee;
Query OK, 61 rows affected (0.07 sec)
Records: 61 Deleted: 0 Skipped: 0 Warnings: 0

mysql> select * from employee;

```

ENAME	ESSN	ADDRESS	SALARY	SUPERSSN	DNO
白玉芬	00	广州市天河区珠江新城花城大道83号	1000.00	24	100
仓春莲	01	广东省广州市石牌西路68号	2000.00	20	100
仓红	02	广州市番禺区市桥清河东路41号	1500.00	14	011
陈超云	03	佛山市汾江西路13号	3000.00	02	011
陈高	04	佛山市南海区桂城天佑三路1号大院	4000.00	06	001
陈国祥	05	佛山市顺德区容奇大道东23号	3500.00	09	000
陈宏柳	06	肇庆市端州七路13号	2800.00	NULL	001
陈金娣	07	广州市石牌西路68号	4500.00	14	011
陈丽丽	08	清远市北江三路海关大楼6楼	3000.00	24	100
陈平	09	韶关市环园东路1栋8楼	3000.00	NULL	000
陈向东	10	云浮市建设南路60号	2500.00	09	000
陈晓冬	11	广州市花都区建设北路79号	5500.00	01	100
陈小荣	12	广州市机场路561号	3600.00	NULL	010
陈秀芬	13	韶关市环园东路1号	2600.00	24	100
陈艳华	14	佛山市禅城区汾江西路13号	4600.00	NULL	011
陈兆国	15	广州市天河区珠江新城华利路61号	2700.00	12	010
成秀山	16	从化市街口街关前路1栋	1900.00	01	100
仇腊梅	17	广州市站南路4号流花邮政大院	2800.00	01	100
戴金辉	18	清远市北江三路	3300.00	01	100
邓海燕	19	广州市番禺区清河东路43号	2000.00	12	010
崔雷	20	广州市南沙区南沙街进港大道1	4000.00	NULL	100
丁德明	21	广州市白云区	2900.00	07	011
丁素琴	22	广州市萝岗区开创大道北1168号	3500.00	24	100
董荣柱	23	河源市河源大道北229号	1500.00	19	010
张红	24	肇庆市端州七路13号	2000.00	20	100
冯霞	25	佛山市禅城区汾江西路13号	3400.00	02	011
伏严成	26	佛山市顺德区容桂街容奇大道东5号	4500.00	01	100
岑德华	27	佛山市三水区西南镇广海大道中17号	5600.00	15	010
岑歌	28	广州市天河区东站1号	7000.00	01	100
高艳	29	广州市革新路122号	8000.00	02	011
顾其兰	30	广州市番禺区莲花山旅游区内	7500.00	07	011
管小云	31	广州市天河区东站2号	2900.00	01	100
何佳丽	32	广州市天河区东站3号	2800.00	07	011
侯志玲	33	广州市天河区东站4号	3500.00	01	100
黄慧	34	广州市天河区东站5号	4800.00	05	000
黄伟	35	广州市天河区东站6号	5600.00	02	011
惠志刚	36	广州市天河区东站7号	5500.00	15	010
李赛花	37	广州市天河区东站8号	4900.00	19	010
李铜然	38	广州市天河区东站9号	2700.00	24	100
纪海燕	39	广州市天河区东站10号	3400.00	15	010
蒋雯	40	广州市天河区东站11号	4800.00	01	100

但是换一句查询语句：

```

mysql> select * from employee where dno = "100";
Empty set (0.01 sec)

mysql> _

```

可是明明有 DNO 等于 100 的数据，为什么查不出来？？？！！！！

这时候要考虑是否数据文件的换行符问题：

如果是 Windows 系统的文本文件，可能必须使用 LINES TERMINATED BY '/r/n' 来读取文件，这是因为 Windows 系统的特点是使用两个字符作为行终止符。在某些程序中，在书写文件时，可能使用/r 作为行终止符，如写字板。读取这类文件时，可以用 LINES TERMINATED BY '/r' 。

解决办法：

```

LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server
8.0/Uploads/EMPLOYEE.txt' INTO TABLE employee LINES TERMINATED BY
'\r\n' ;

```

```
mysql> LOAD DATA INFILE "D:/02_工作文档/01_2019春上课材料/数据库/实验指导书-fm/Lab1/EMPLOYEE.txt" INTO TABLE employee LINES TERMINATED BY '\r\n' ;
Query OK, 61 rows affected (0.10 sec)
Records: 61 Deleted: 0 Skipped: 0 Warnings: 0

mysql> SELECT * FROM company.employee where DNO = "100";
```

ENAME	ESSN	ADDRESS	SALARY	SUPERSSN	DNO
白玉芬	00	广州市天河区珠江新城花城大道83号	1000.00	24	100
仓春莲	01	广东省广州市石牌西路68号	2000.00	20	100
陈超云	03	清远市北江三路海关大楼6楼	3000.00	24	100
陈晓冬	11	广州市花都区建设北路79号	5500.00	01	100
陈秀芬	13	韶关市环园东路1号	2600.00	24	100
成秀山	16	从化市街口街关前路1栋	1900.00	01	100
仇腊梅	17	广州市站南路4号流花邮政大院	2800.00	01	100
戴金辉	18	清远市北江三路	3300.00	01	100
翟蕾	20	广州市南沙区南沙街进港大道1	4000.00	NULL	100
丁素琴	22	广州市萝岗区开创大道北1168号	3500.00	24	100
张红	24	肇庆市端州七路13号	2000.00	20	100
伏严成	26	佛山市顺德区容桂街容奇大道东5号	4500.00	01	100
皋歌	28	广州市天河区东站1号	7000.00	01	100
管小云	31	广州市天河区东站2号	2900.00	01	100
侯志玲	33	广州市天河区东站4号	3500.00	01	100
李铜然	38	广州市天河区东站9号	2700.00	24	100
蒋雯	40	广州市天河区东站11号	4800.00	01	100
李金凤	43	广州市天河区东站14号	4400.00	01	100
潘杰	52	广州市天河区东站23号	6300.00	01	100
邱文干	54	广州市天河区东站25号	4000.00	01	100

20 rows in set (0.00 sec)

这样就可以查出来了。

其实再看认真一点，有问题的是这样的：

```
选择命令提示符 - mysql -h localhost -u root -p

mysql> LOAD DATA INFILE "D:/02_工作文档/01_2019春上课材料/数据库/实验指导书-fm/Lab1/EMPLOYEE.txt" INTO TABLE employee;
Query OK, 61 rows affected (0.07 sec)
Records: 61 Deleted: 0 Skipped: 0 Warnings: 0

mysql> select * from employee;
```

ENAME	ESSN	ADDRESS	SALARY	SUPERSSN	DNO
白玉芬	00	广州市天河区珠江新城花城大道83号	1000.00	24	100
仓春莲	01	广东省广州市石牌西路68号	2000.00	20	100
仓红	02	广州市番禺区市桥清河东路41号	1500.00	14	011
陈超云	03	佛山市汾江西路13号	3000.00	02	011
陈高	04	佛山市南海区桂城天佑三路1号大院	4000.00	06	001
陈国祥	05	佛山市顺德区容奇大道东23号	3500.00	09	000
陈宏柳	06	肇庆市端州七路13号	2800.00	NULL	001
陈金焯	07	广州市石牌西路68号	4500.00	14	011
陈丽丽	08	清远市北江三路海关大楼6楼	3000.00	24	100
陈平	09	韶关市环园东路1栋8楼	3000.00	NULL	000
陈向东	10	云浮市建设南路60号	2500.00	09	000
陈晓冬	11	广州市花都区建设北路79号	5500.00	01	100
陈小荣	12	广州市机场路561号	3600.00	NULL	010
陈秀芬	13	韶关市环园东路1号	2600.00	24	100
陈艳华	14	佛山市禅城区汾江西路13号	4600.00	NULL	011
陈兆国	15	广州市天河区珠江新城华利路61号	2700.00	12	010
成秀山	16	从化市街口街关前路1栋	1900.00	01	100
仇腊梅	17	广州市站南路4号流花邮政大院	2800.00	01	100
戴金辉	18	清远市北江三路	3300.00	01	100
邓海燕	19	广州市番禺区清河东路43号	2000.00	12	010
翟蕾	20	广州市南沙区南沙街进港大道1	4000.00	NULL	100
丁德明	21	广州市白云区	2900.00	07	011
丁素琴	22	广州市萝岗区开创大道北1168号	3500.00	24	100
董荣柱	23	河源市河源大道北229号	1500.00	19	010
张红	24	肇庆市端州七路13号	2000.00	20	100
冯霞	25	佛山市禅城区汾江西路13号	3400.00	02	011
伏严成	26	佛山市顺德区容桂街容奇大道东5号	4500.00	01	100
皋德华	27	佛山市三水区西南镇广海大道中17号	5600.00	15	010
皋歌	28	广州市天河区东站1号	7000.00	01	100
高艳	29	广州市革新路122号	8000.00	02	011
顾其兰	30	广州市番禺区莲花山旅游区内	7500.00	07	011
管小云	31	广州市天河区东站2号	2900.00	01	100
何佳丽	32	广州市天河区东站3号	2800.00	07	011
侯志玲	33	广州市天河区东站4号	3500.00	01	100
黄慧	34	广州市天河区东站5号	4800.00	05	000
黄伟	35	广州市天河区东站6号	5600.00	02	011
惠志刚	36	广州市天河区东站7号	5500.00	15	010
季寒花	37	广州市天河区东站8号	4900.00	19	010
李铜然	38	广州市天河区东站9号	2700.00	24	100
纪海燕	39	广州市天河区东站10号	3400.00	15	010
蒋雯	40	广州市天河区东站11号	4800.00	01	100

正常的应该是这样的，看出区别了吗？

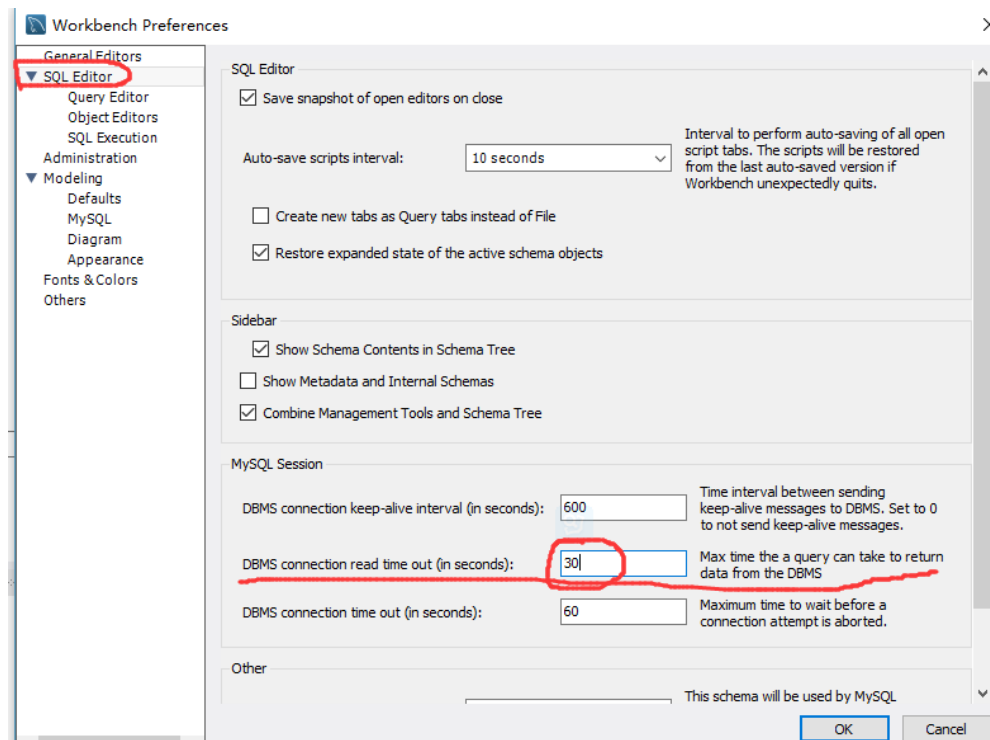
mysql> select \* from employee;

ENAME	ESSN	ADDRESS	SALARY	SUPERSSN	DNO
白玉芬	00	广州市天河区珠江新城花城大道83号	1000.00	24	100
仓春莲	01	广东省广州市石牌西路68号	2000.00	20	100
仓红	02	广州市番禺区市桥滘河东路41号	1500.00	14	011
陈道云	03	佛山市汾江西路13号	3000.00	02	011
陈浩	04	佛山市南海区桂城大佑三路1号大院	4000.00	06	001
陈国祥	05	佛山市顺德区容奇大道东23号	3500.00	09	000
陈宏梅	06	肇庆市端州七路13号	2800.00	NULL	001
陈金强	07	广州市石牌西路68号	4500.00	14	011
陈丽丽	08	清远市北江三路海关大楼6楼	3000.00	24	100
陈平	09	韶关市环园东路1栋8楼	3000.00	NULL	000
陈海东	10	云浮市建设南路60号	2500.00	09	000
陈晓冬	11	广州市花都区建设北路79号	5500.00	01	100
陈小东	12	广州市机场路561号	3600.00	NULL	010
陈秀芬	13	韶关市环园东路1号	2600.00	24	100
陈德华	14	佛山市禅城区汾江西路13号	4600.00	NULL	011
陈英国	15	广州市天河区珠江新城华利路61号	2700.00	12	010
成青山	16	从化市街口街关前路1栋	1900.00	01	100
仇碧梅	17	广州市站南路4号流花邮政大院	2800.00	01	100
黄金碧	18	清远市北江三路	3300.00	01	100
邓海燕	19	广州市番禺区清河东路43号	2000.00	12	010
霍雷	20	广州市南沙区南沙街进港大道1	4000.00	NULL	100
丁慧明	21	广州市白云区	2900.00	07	011
丁家号	22	广州市萝岗区开创大道北1168号	3500.00	24	100
董家柱	23	河源市河源大道北229号	1500.00	19	010
张红	24	肇庆市端州七路13号	2000.00	20	100
冯雷	25	佛山市禅城区汾江西路13号	3400.00	02	011
伏广成	26	佛山市顺德区容桂街容奇大道东5号	4500.00	01	100
梁德华	27	佛山市三水区西南镇广海大道中17号	5600.00	15	010
卓建	28	广州市天河区东站1号	7000.00	01	100
高锦	29	广州市东新路122号	8000.00	02	011
顾其兰	30	广州市番禺区莲花山旅游区	7500.00	07	011
官小云	31	广州市天河区东站2号	2900.00	01	100
梅佳丽	32	广州市天河区东站3号	2800.00	07	011
侯志玲	33	广州市天河区东站4号	3500.00	01	100
黄慧	34	广州市天河区东站5号	4800.00	05	000
周市	35	广州市天河区东站6号	5600.00	02	011
曲生刚	36	广州市天河区东站7号	5500.00	15	010
李慕花	37	广州市天河区东站8号	4900.00	19	010
李钢然	38	广州市天河区东站9号	2700.00	24	100
纪国森	39	广州市天河区东站10号	3400.00	15	010
蒋志	40	广州市天河区东站11号	4800.00	01	100
廖海霞	41	广州市天河区东站12号	2600.00	10	000
宋梅	42	广州市天河区东站13号	3500.00	02	011
金凤	43	广州市天河区东站14号	4400.00	01	100
李军	44	广州市天河区东站15号	7200.00	02	011

## 5.2.10 Error Code: 2013. Lost connection to MySQL server during query 30.000 sec

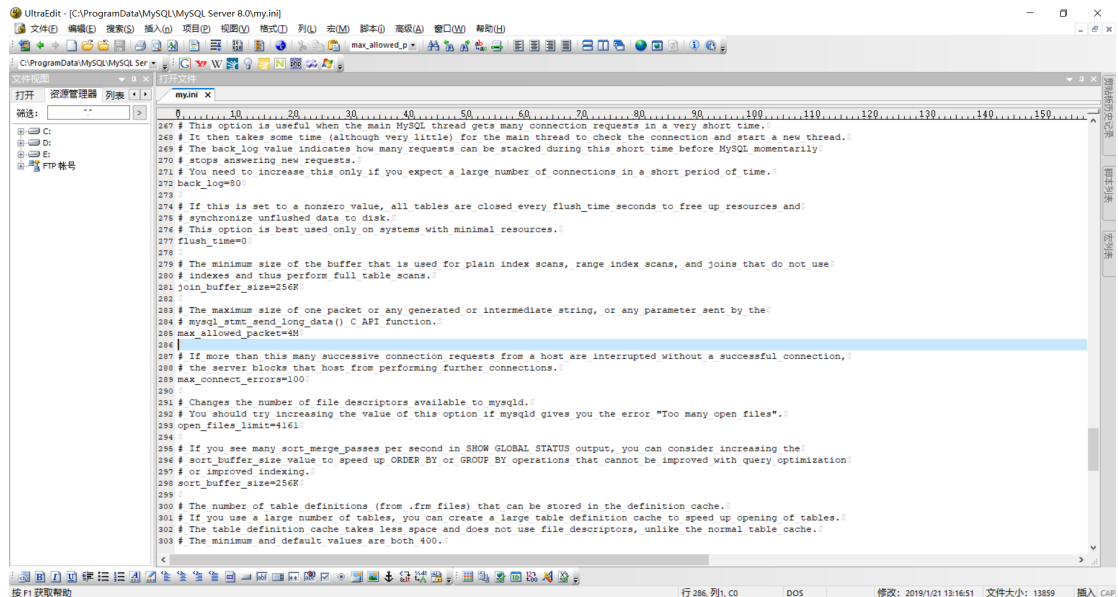
解决办法:

Edit->Preferences->SQL Editor 将下图中的 30 改大一点, 重启 workbench 即可。



## 5.2.11 Errcode: 28 - No space left on device

临时表空间不足，找到 my.ini 配置文件，查找 `max_allowed_packet`



默认只有 4M，修改大些，保存后，重启 MySQL 服务。

然后可以用 `show VARIABLES like 'max_allowed_packet%'` 命令查看是否修改成功：

```
CA 命令提示符 - mysql -u root -p

mysql> show VARIABLES like 'max_allowed_packet%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| max_allowed_packet | 1073741824 |
| mysqlx_max_allowed_packet | 67108864 |
| slave_max_allowed_packet | 1073741824 |
+-----+-----+
3 rows in set, 1 warning (0.01 sec)
```

当然，这个值并不是随便设置越大越好，大部分时候先想想自己的 SQL 语句是否可以优化。

## 5.2.12 ERROR 1045 (28000): Access denied for user

`'root'@'localhost' (using password: YES)`

检查密码是不是输错了！！

