

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

目录

1	实验目的	3
2	实验环境	3
3	实验内容	3
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 EMPLOYEE 数据库	7
4.1.3	创建 6 个表	9
4.1.4	导入数据	12
4.2	数据的增删改查	16
4.2.1	查询	17
4.2.2	新增	18
4.2.3	修改	19
4.2.4	删除	20
4.3	视图	20
5	附录	22
5.1	MySQL 手册（命令行方式）	22
5.1.1	启动 MySQL 服务	22
5.1.2	连接 MySQL 服务器	23
5.1.3	查看 MySQL 中有哪些数据库	23
5.1.4	选定数据库	24
5.1.5	查看当前在的数据库	24
5.1.6	查看当前数据库中的表	24
5.1.7	查看某个表的表结构	25
5.1.8	使用 help 命令来了解其他命令和变量类型等的含义	25
5.1.9	取消命令	26
5.1.10	断开 MySQL 服务器连接	26
5.1.11	导出数据库	26
5.1.12	创建数据库	27
5.1.13	导入数据库	27
5.1.14	使用数据文件导入数据	28
5.2	附加内容*	29
5.3	常见问题	34
6	参考资料	44

*为选做内容

1 实验目的

掌握 MySQL 关系数据库管理系统的基本操作，熟练使用 SQL 语言管理 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。*使用 Query Profiler、Explain 进行查询诊断分析（选做）。

2 实验环境

Windows 7 操作系统、MySQL8.0。

3 实验内容

参考第 4 节实验步骤，完成以下内容，注意边做边截图填写截图报告（**截图时需要体现 SQL 语句、执行结果、Output 窗口**）：

- 1、创建关系数据库 EMPLOYEE;
- 2、建 6 个表，并导入数据;
- 3、使用 SQL 语言完成如下 14 个内容:

- (1) 查询 emp_no 是 “10029” 的员工信息，显示其 emp_no, birth_date, first_name, last_name, gender, hire_date, title;
- (2) 查询入职时间在 1990 年后且在 “Finance” 部门工作过的男员工姓名;
- (3) 查询没有在 “Production” 部门工作过且 first_name 是 “Ge” 开头的的员工信息，显示其 emp_no, birth_date, first_name, last_name, gender, hire_date;
- (4) 查询 first_name 相同且人数超过 3 人的员工信息，显示其 emp_no, birth_date, first_name, last_name, gender, hire_date, 要求按 first_name 升序显示;
- (5) 查询至少在 “Production” 和 “Quality Management” 两个部门都工作过的员工编号;
- (6) 查询至少在 2 个部门工作过的员工人数;
- (7) 查询在 “d003” 部门工作过的且工资最高的员工编号及其最高工资;
- (8) 查询 “d002” 部门的当前领导姓名;
- (9) 查询当前每个部门的部门编号和员工总工资;
- (10) 查询当前部门员工平均工资在 70000 元到 80000 元（包含 70000，低于 80000）的部门编号，部门名称和员工平均工资;
- (11) 在 departments 表新增 2 条记录（内容自定）;
- (12) 在 departments 表中删除刚才新增的 2 条记录中的 1 条;
- (13) 在 departments 表中修改步骤 11 新增的记录;

(14)新建视图，查询所有在 1990 年后入职过“Finance”部门的男员工信息，包括：emp_no, birth_date, first_name, last_name, hire_date, from_date, to_date。

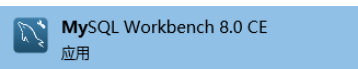
4 实验步骤

本节是以图形界面做示范，同学们也可参考 5.1 节用命令行方式完成本次实验。

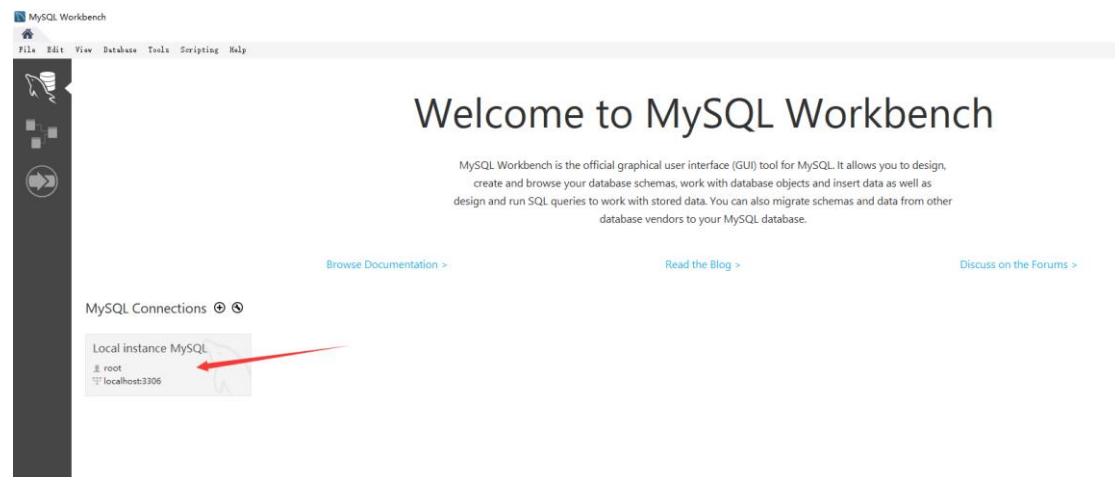
4.1 数据准备

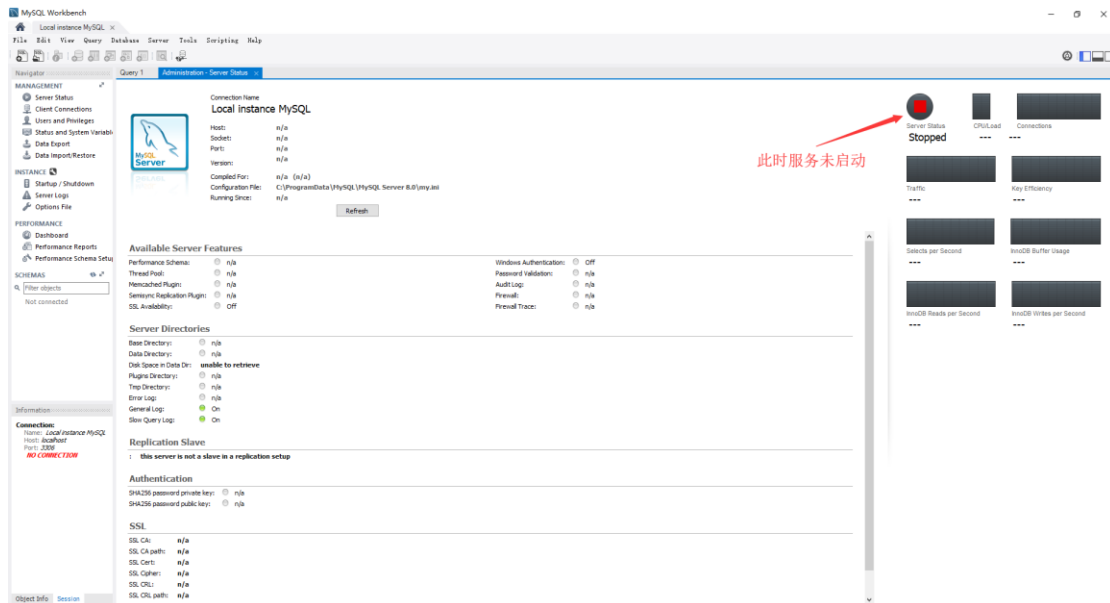
4.1.1 启动 MySQL 服务

1、打开 MySQL Workbench

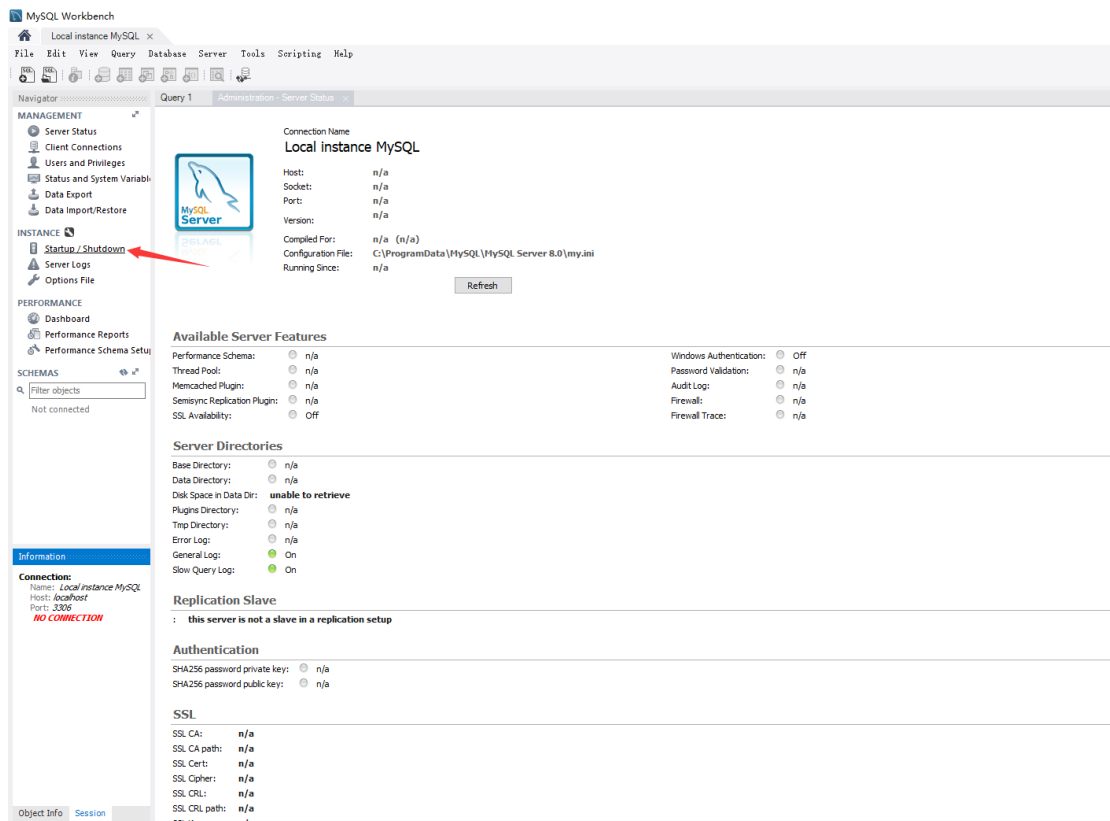


2、点击“Local instance MySQL”

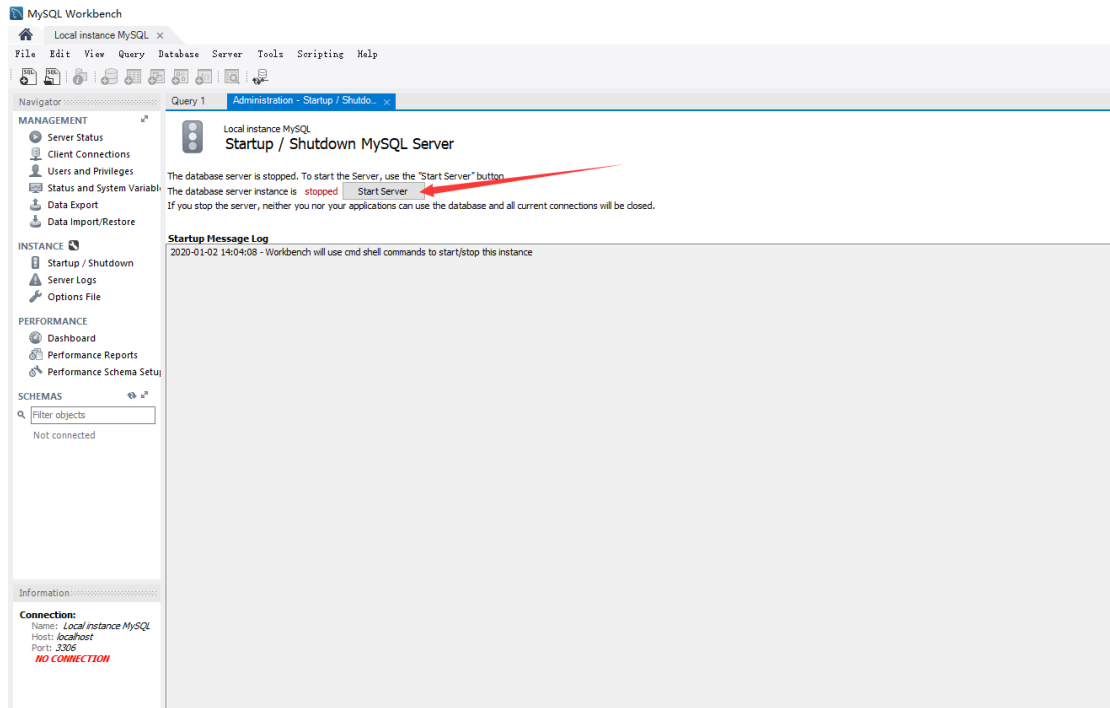




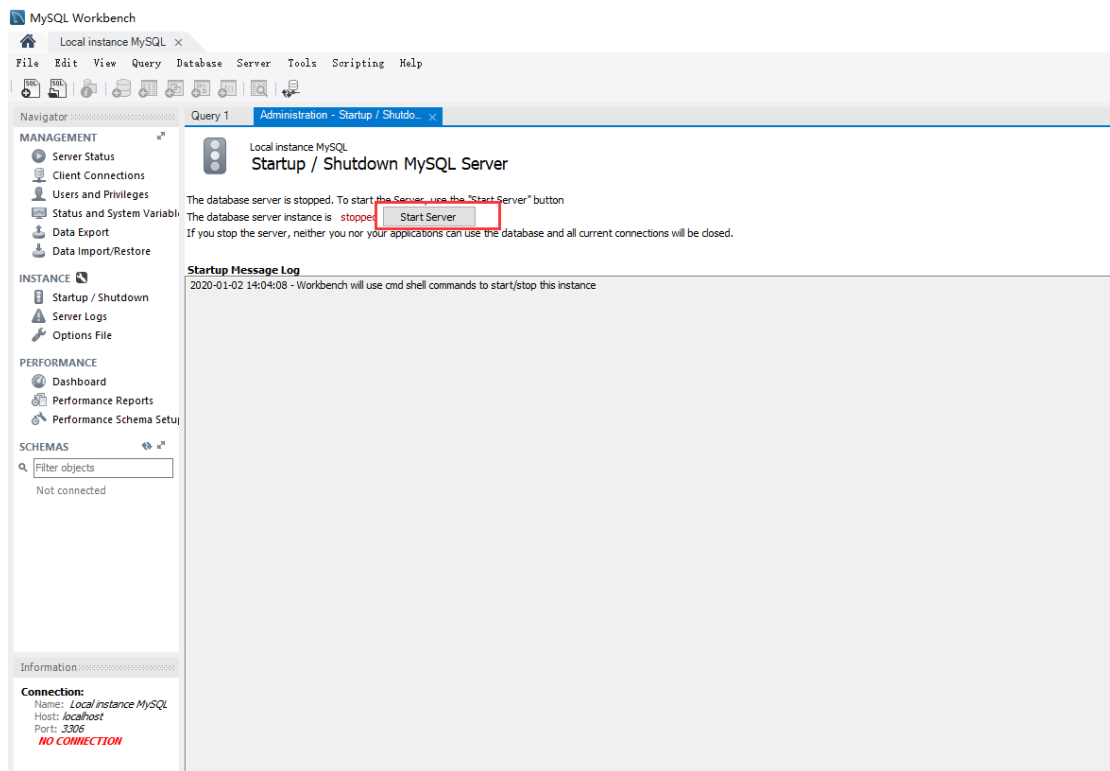
3、点击左边菜单里“Startup/Shutdown”



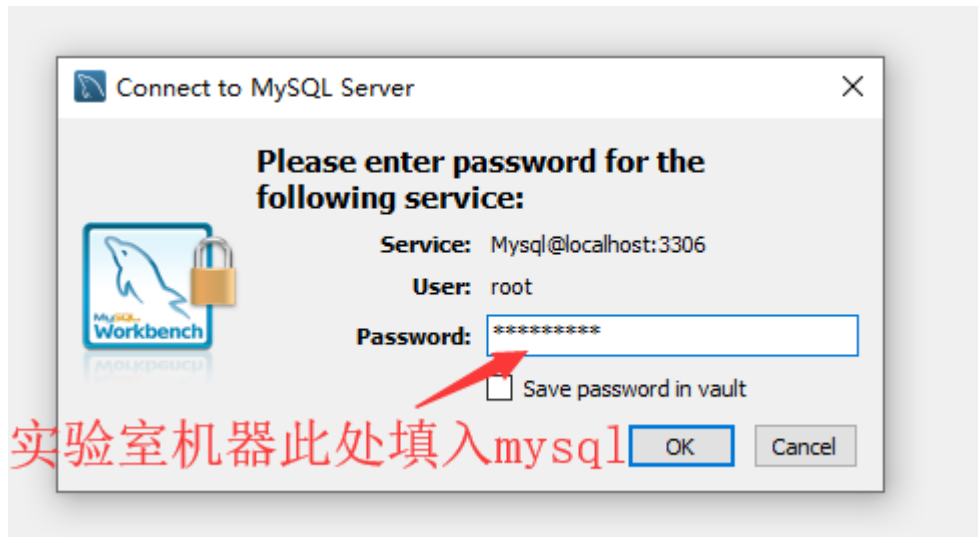
进去服务启停界面：



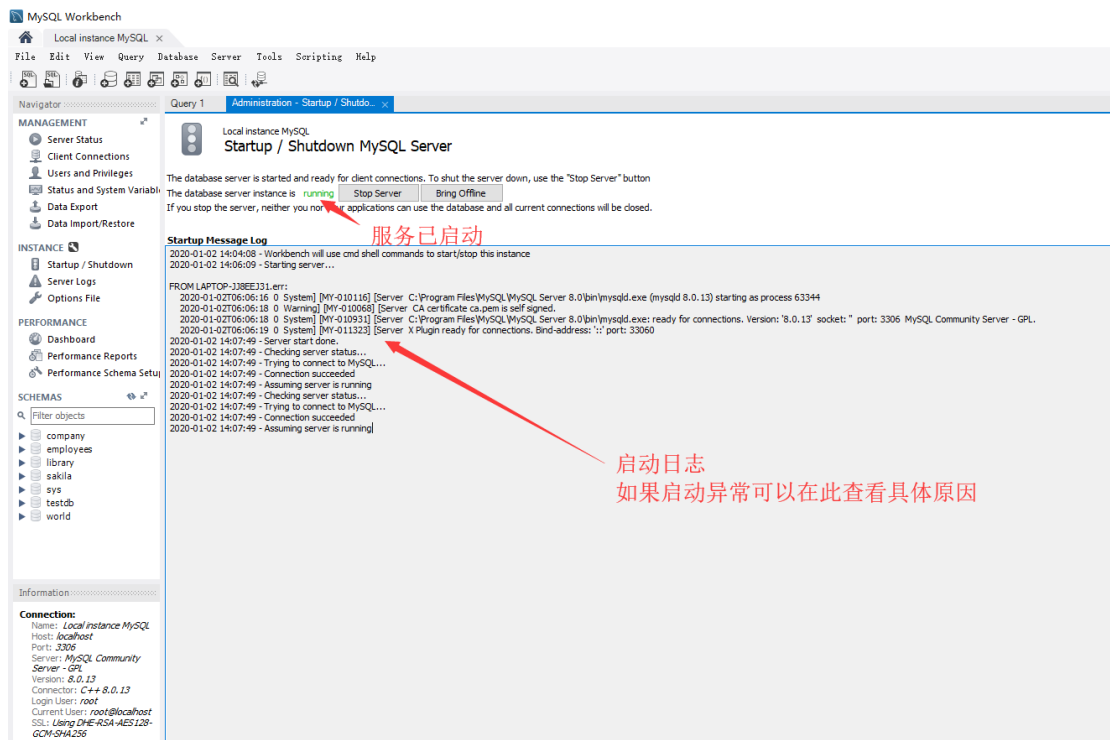
4、点击“Start Server”按钮：



输入 root 密码：




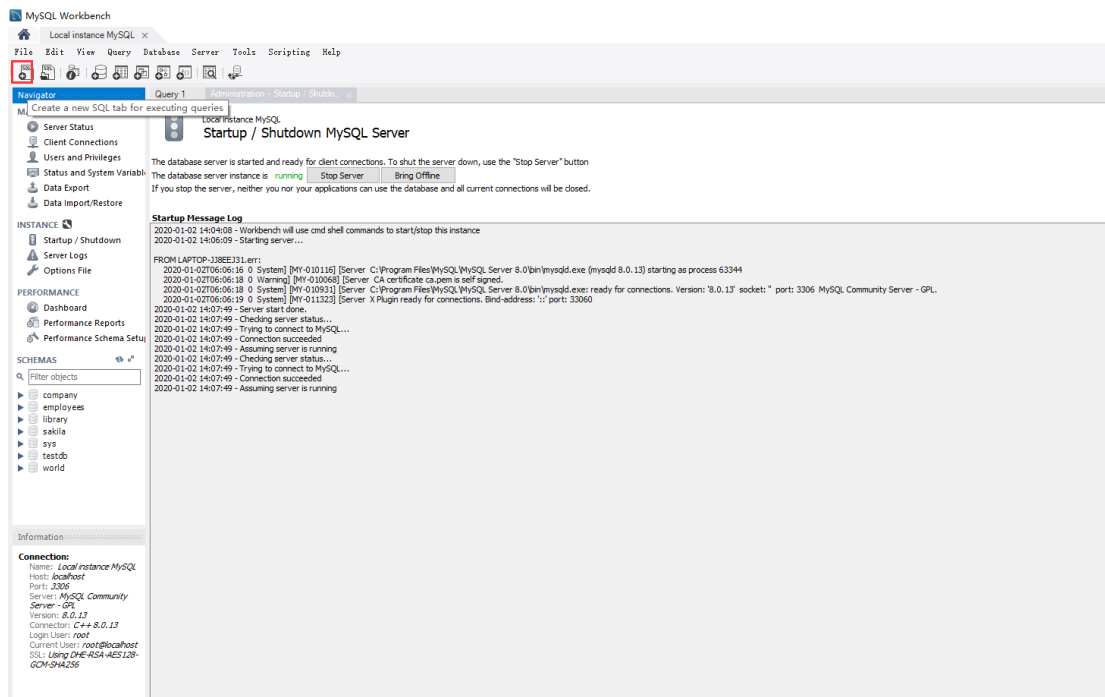
等待片刻，显示服务已启动：



4.1.2 创建 EMPLOYEE 数据库

为了练习 DDL 语句，请同学们使用输入 DDL 语句然后执行的方式建库建表。


1、 点击左上角  按钮

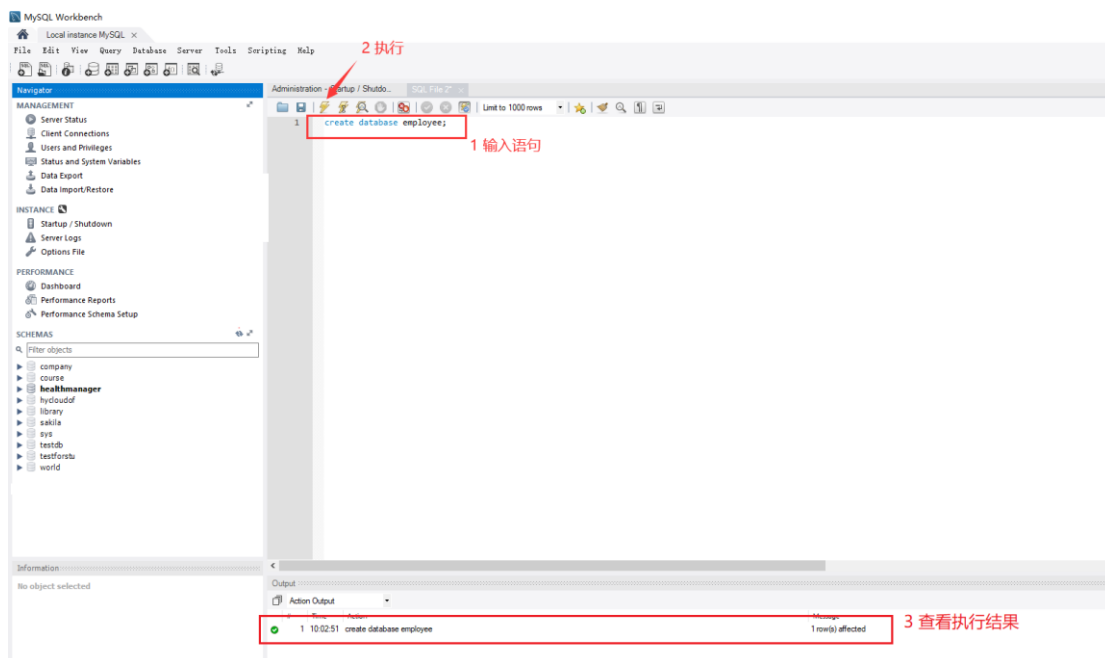


打开一个 SQL File。

2、输入建库语句

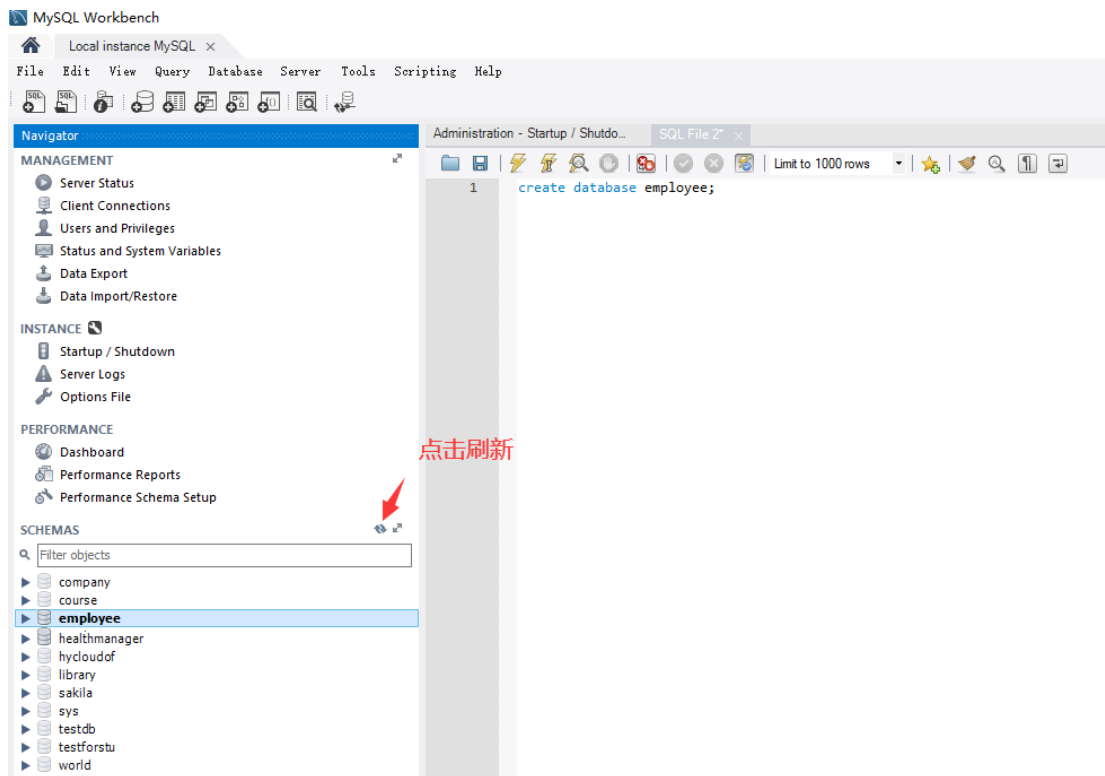
create database employee;

然后点击  按钮执行。下方 output 窗口可以看到执行情况：

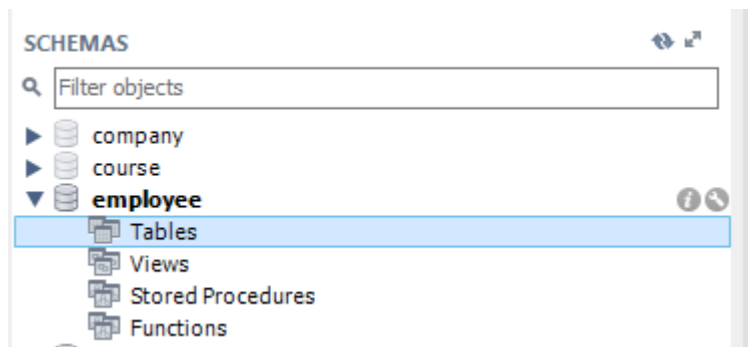


3、新建的数据库 employee 在哪里？

点击左边 SCHEMAS 旁边的刷新符号：



此时新建的 employee 数据库还没有表：



4.1.3 创建 6 个表

我们本次实验用到 6 张表：

1、Departments 表

Table Name: departments Schema: **employees**

Charset/Collation: utf8mb4 utf8mb4_0900_ai_ci Engine: **InnoDB**

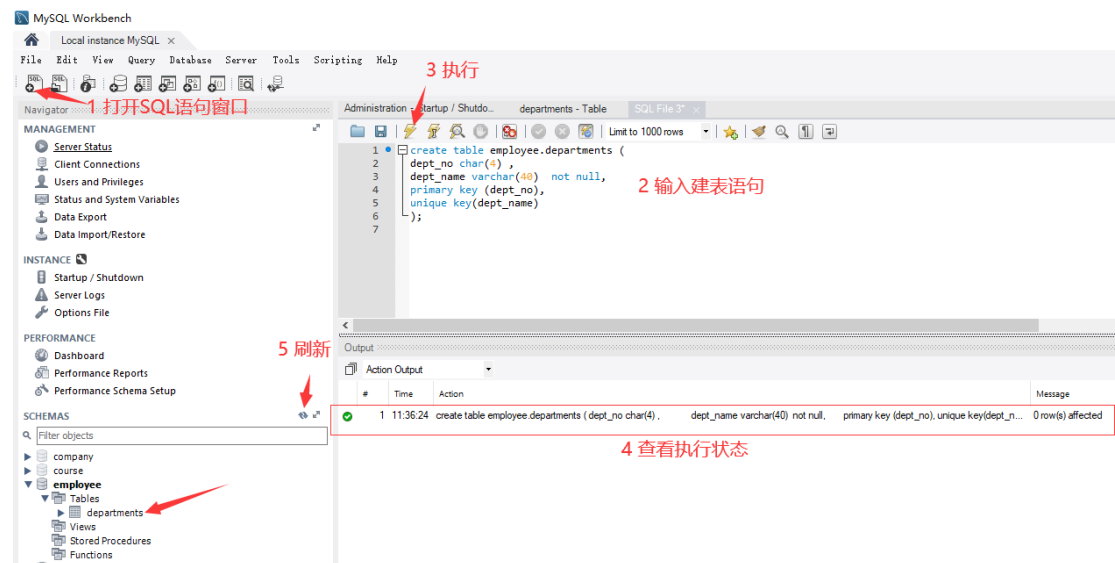
Comments:


Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
dept_no	CHAR(4)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
dept_name	VARCHAR(40)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

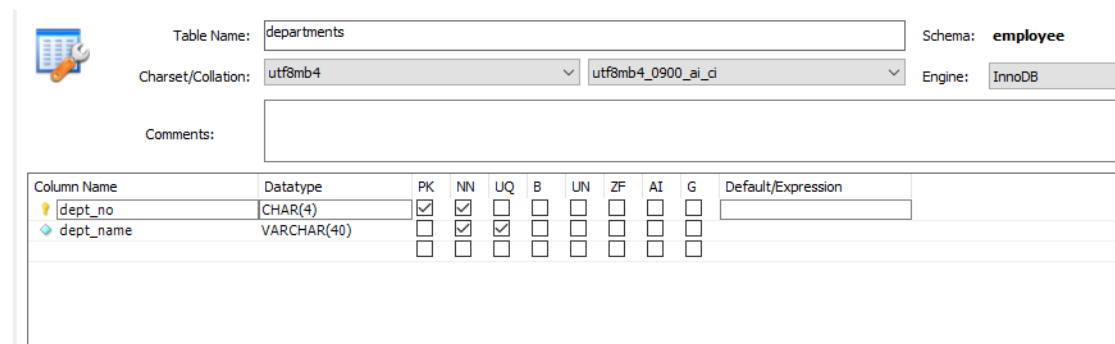
1) 我们示例用 DDL 建表：

```
create table employee.departments (
dept_no char(4) ,          #部门编号
```

```
dept_name varchar(40) not null,           #部门名称
primary key (dept_no),
unique key (dept_name)
);
```

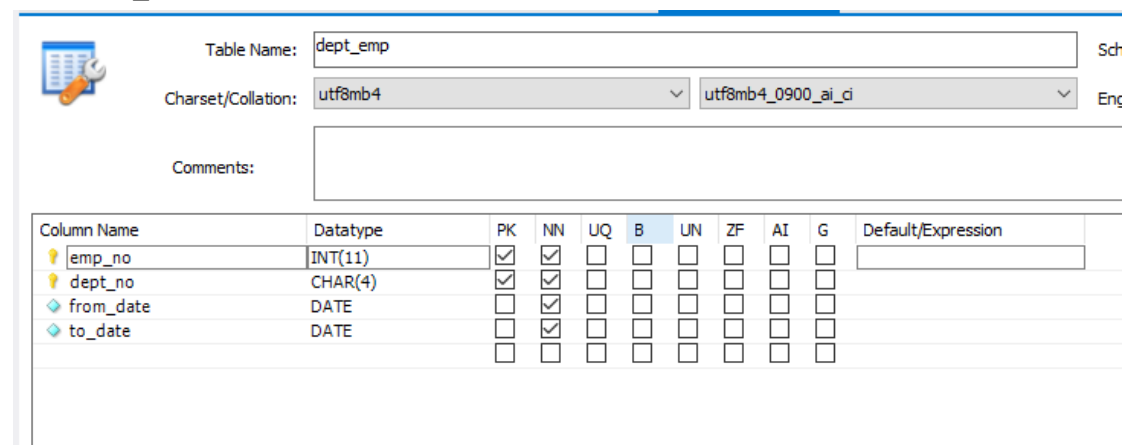


2) 点击 departments 表右边的 ，打开表结构，核对是否与预期一致：








下面请同学们根据下面给出的表结构自行建好另外 5 个表：

2、dept_emp 表










3、dept_manager 表


 Table Name: Sc
 Charset/Collation: Er
 Comments:


Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
 emp_no	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>
 dept_no	CHAR(4)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 from_date	DATE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 to_date	DATE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	





4、employees 表


 Table Name: Sc
 Charset/Collation: Er
 Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
 emp_no	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>
 birth_date	DATE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 first_name	VARCHAR(14)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 last_name	VARCHAR(16)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 gender	ENUM('M', 'F')	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 hire_date	DATE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

5、salaries 表


 Table Name: Sc
 Charset/Collation: Er
 Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
 emp_no	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>
 salary	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 from_date	DATE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 to_date	DATE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

6、titles 表

Table Name:

Charset/Collation:

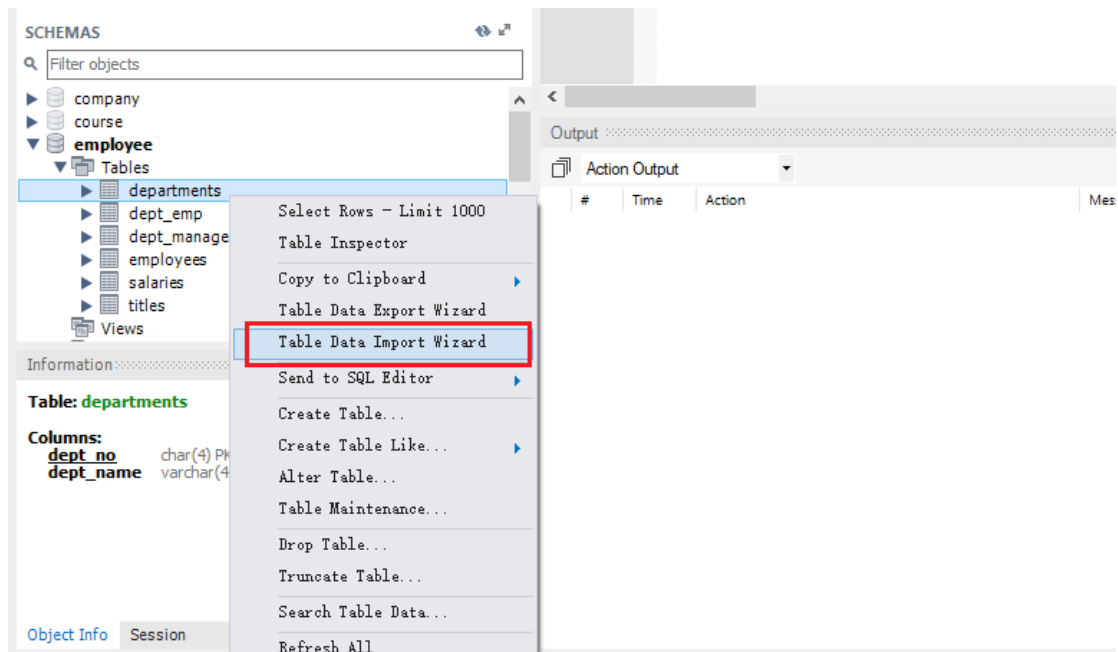
Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
emp_no	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
title	VARCHAR(50)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
from_date	DATE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
to_date	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

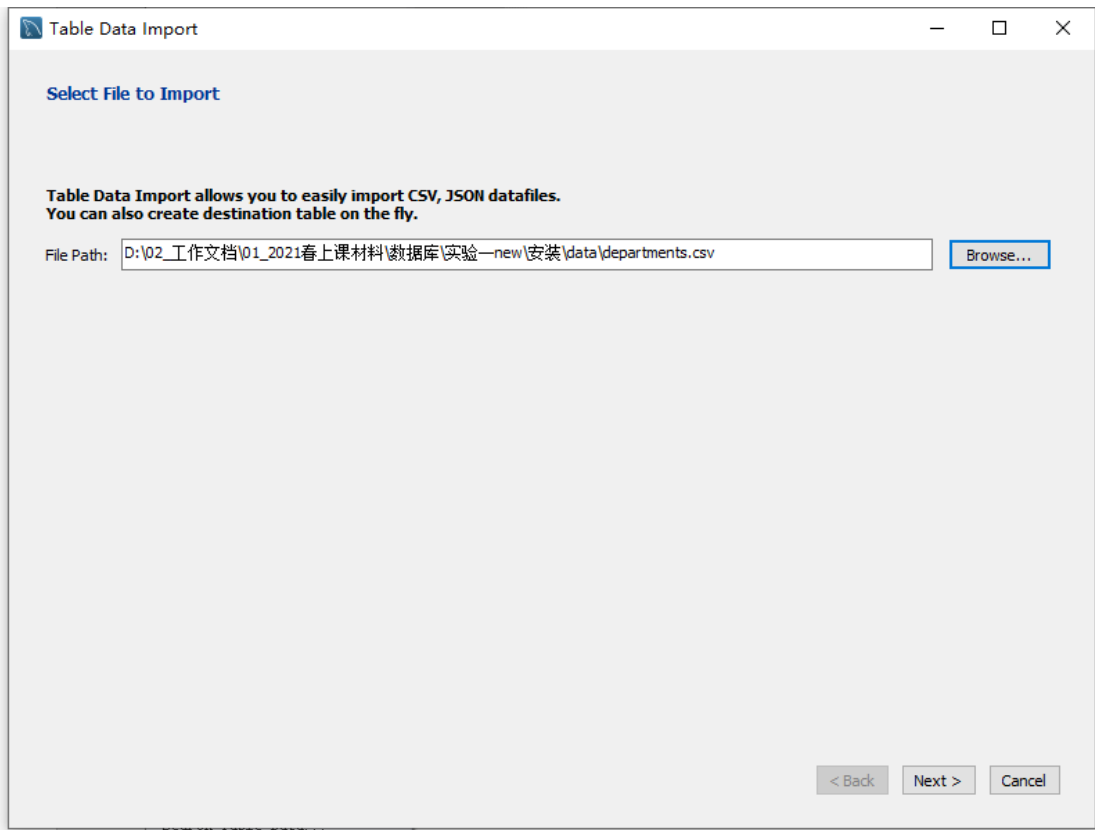
4. 1. 4 导入数据

下面用 MySQL Workbench 提供的数据库导入工具 Table Data Import Wizard 导入数据，步骤：

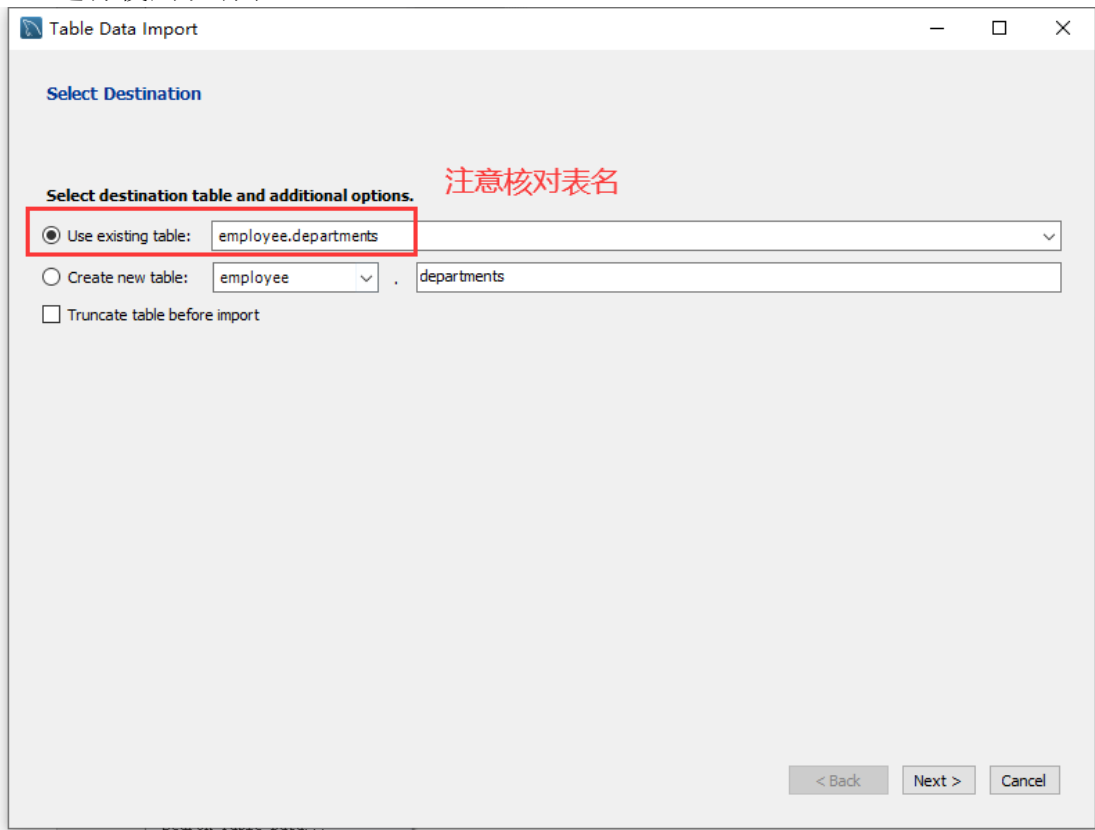
1、建好表后，选中要导入数据的表，右键 Table Data Import Wizard：



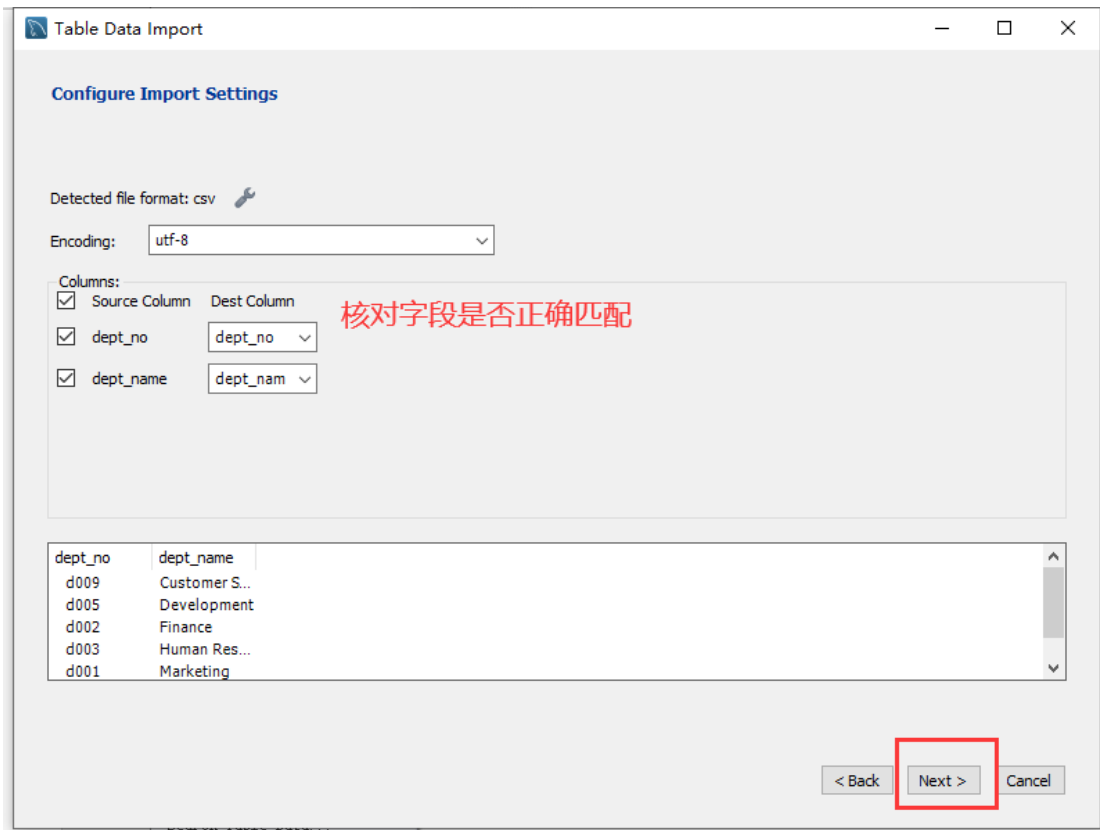
2、选中数据文件：注意是 csv 或者 json 格式，然后 Next：



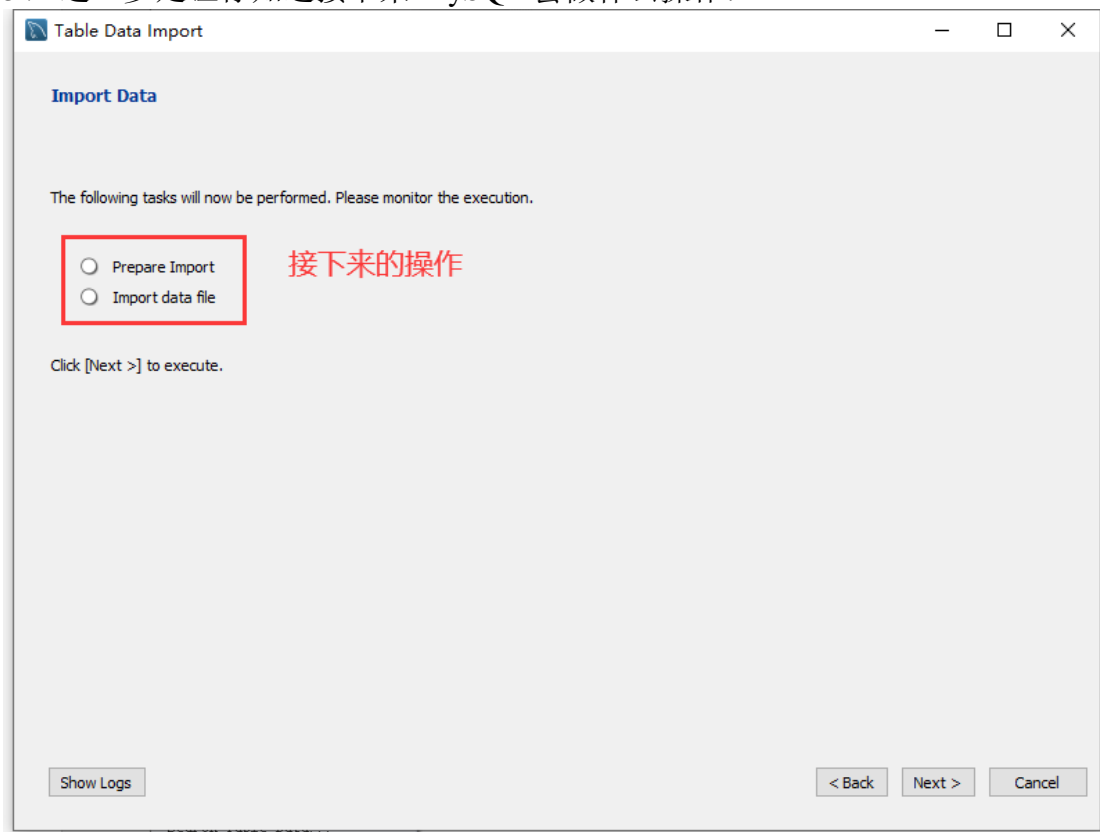
3、选择使用现有表:



4、选择编码格式, 检查数据与字段的对应关系, 可以修改。如果没问题就 Next:

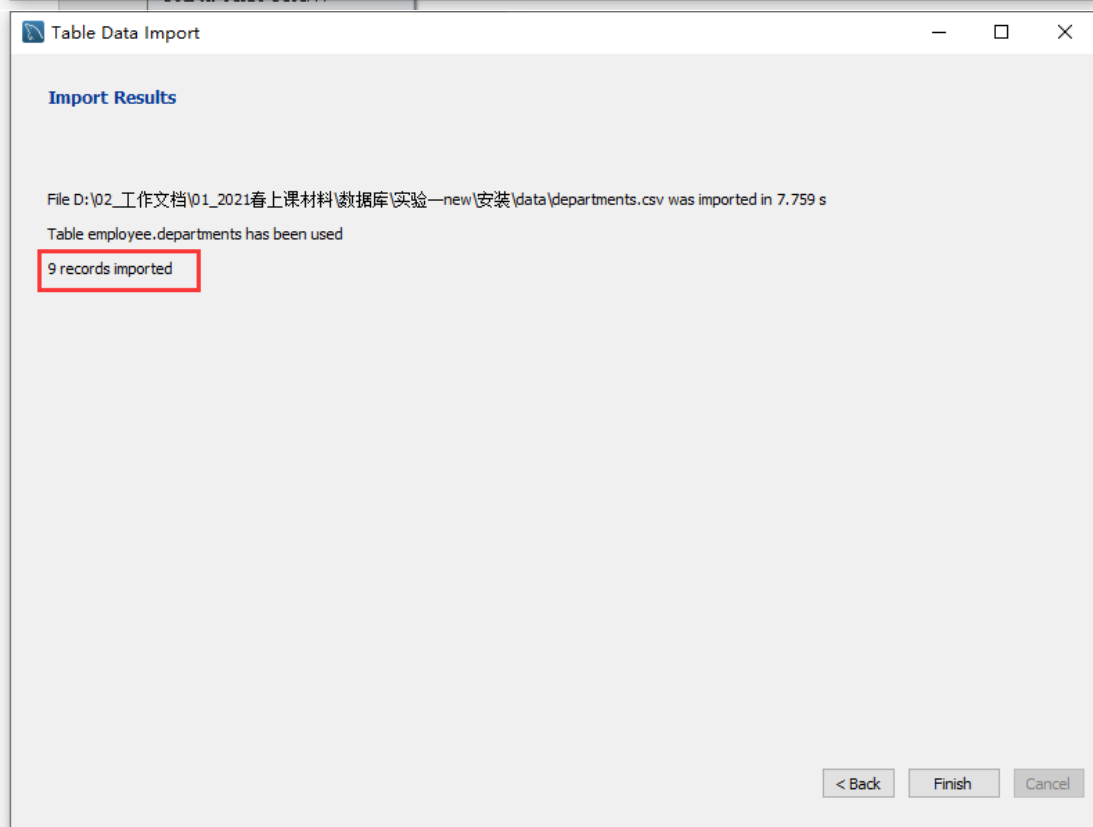
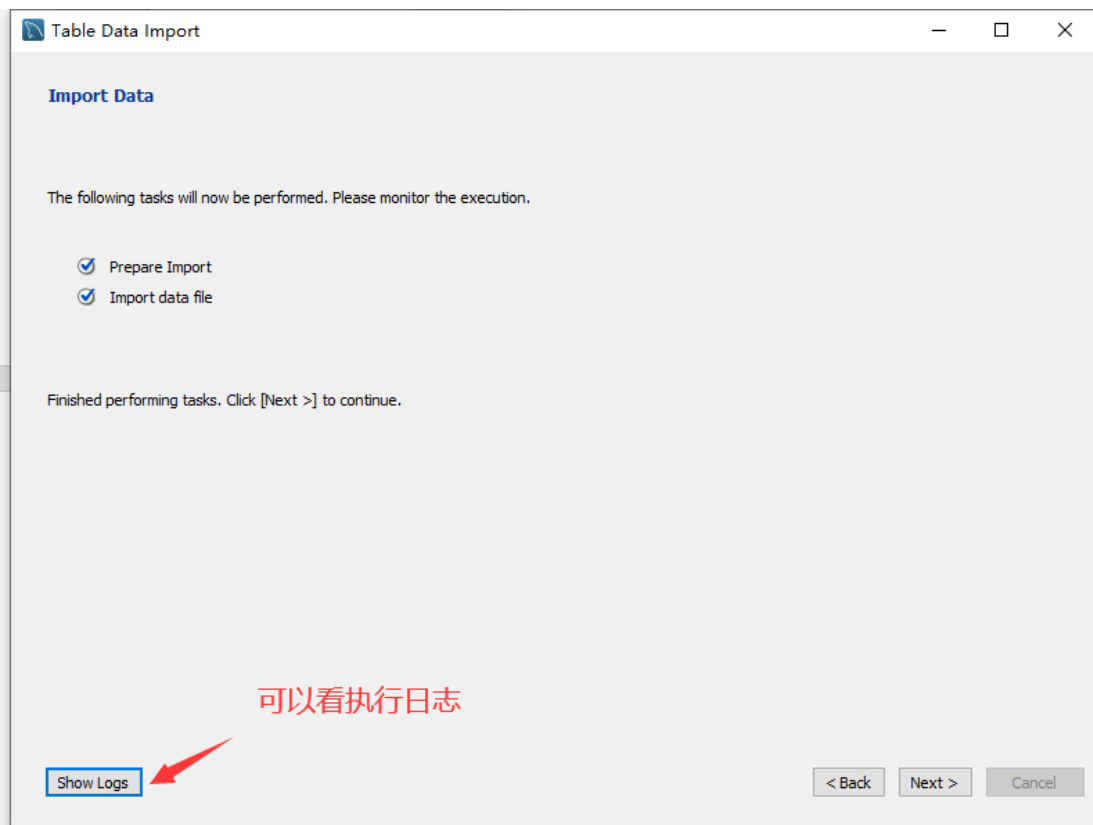


5、这一步是让你知道接下来 MySQL 会做什么操作：



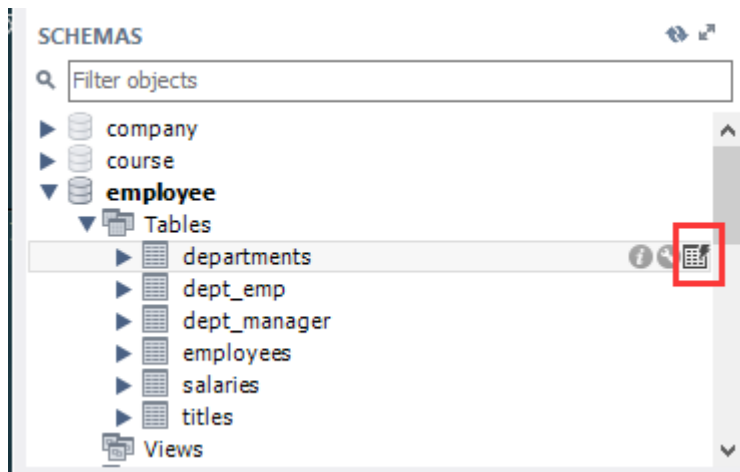
点 Next 开始执行。

6、完成后显示结果：

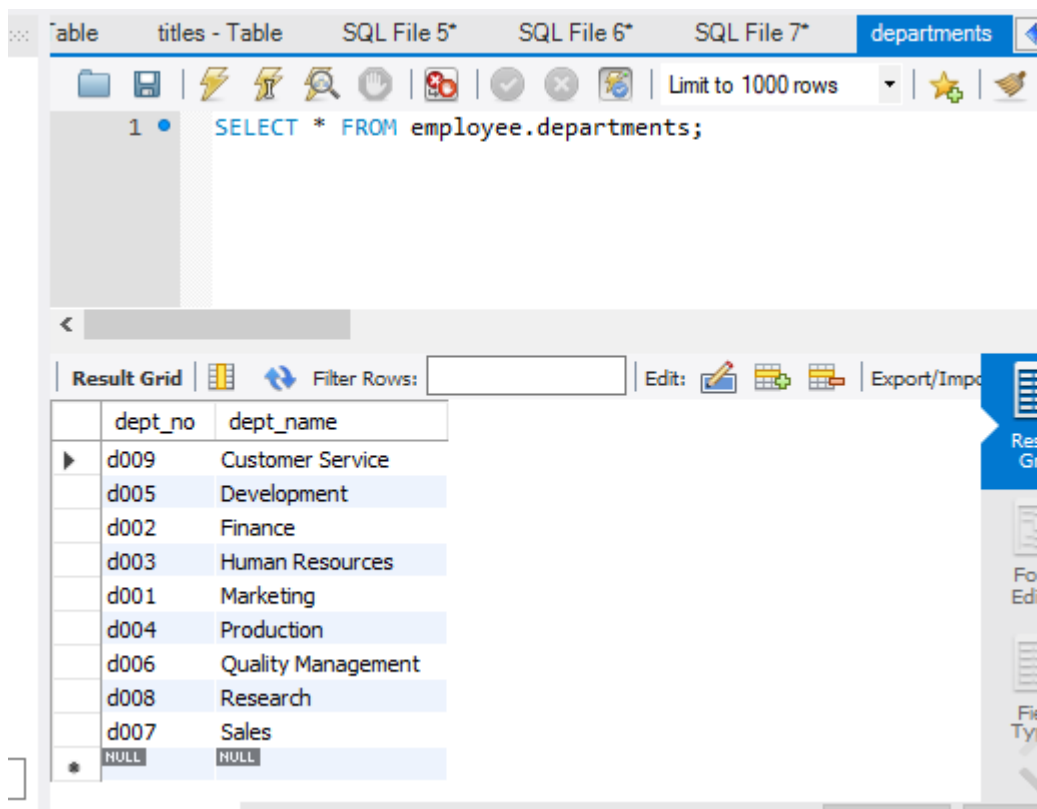


注意：导完数据后确认记录条数是否与源文件一致。

7、检查数据，点击表右边的按钮



可以查看表里的数据：



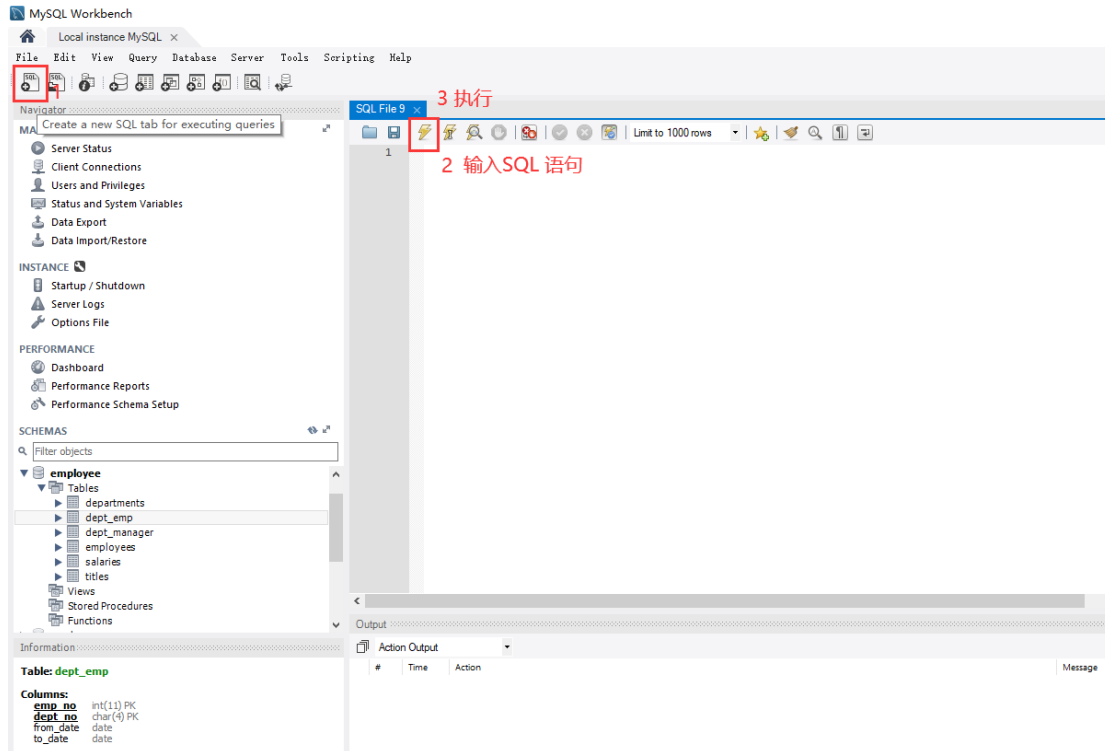
请同学们用下发的数据文件自行导入另外 5 张表的数据。

注：

- 下发的数据中，有两种格式的文件，其中 csv 文件用于 table data import 方式，txt 文件用于 load data 方式（这种导入方式参考第 5.1.14 节）。
- 建议对 dept_emp, employees, titles, salaries 表用 load data 方式导入，速度会快很多。

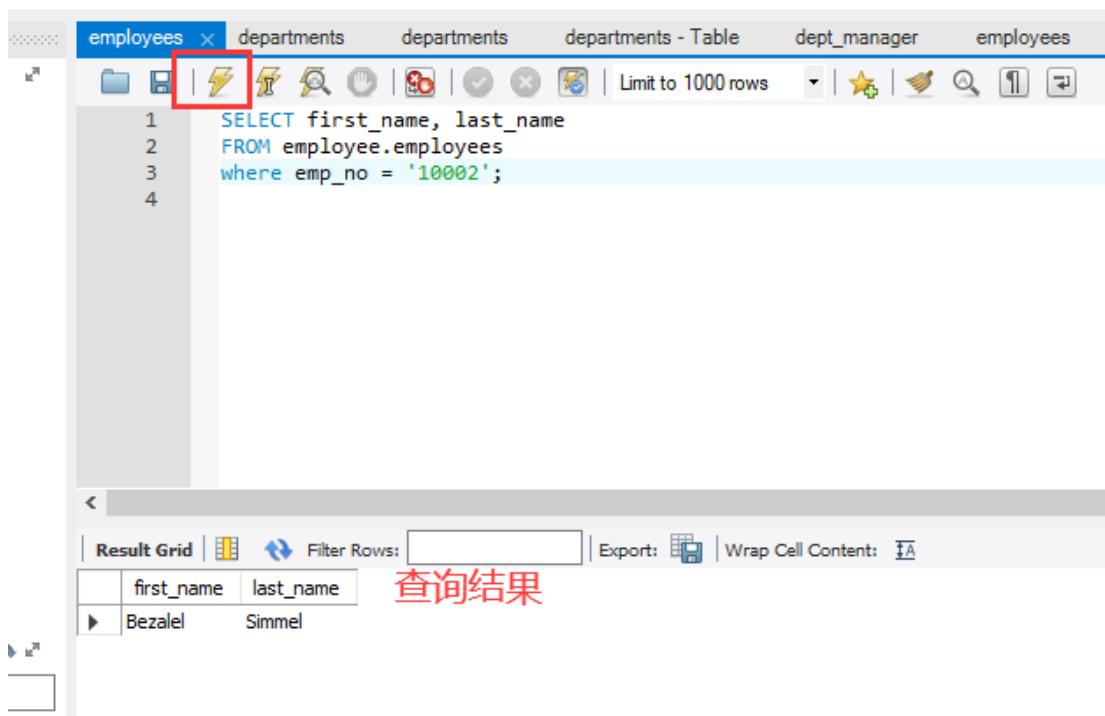
4.2 数据的增删改查

增、删、改、查操作都需要先打开一个 SQL File 页面：



4.2.1 查询

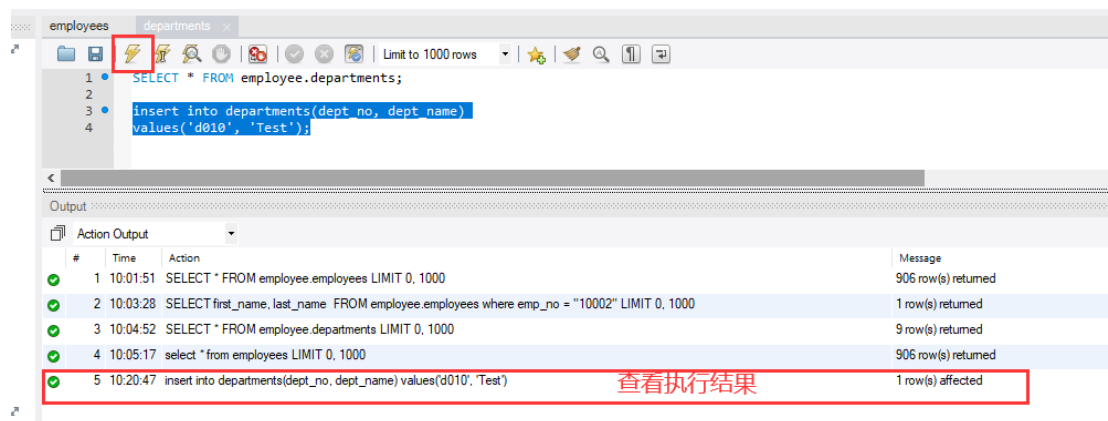
比如：想要查询编号为“10002”的员工姓、名，可以在上图空白处输入
 SELECT first_name, last_name
 FROM employee.employees
 where emp_no = '10002';



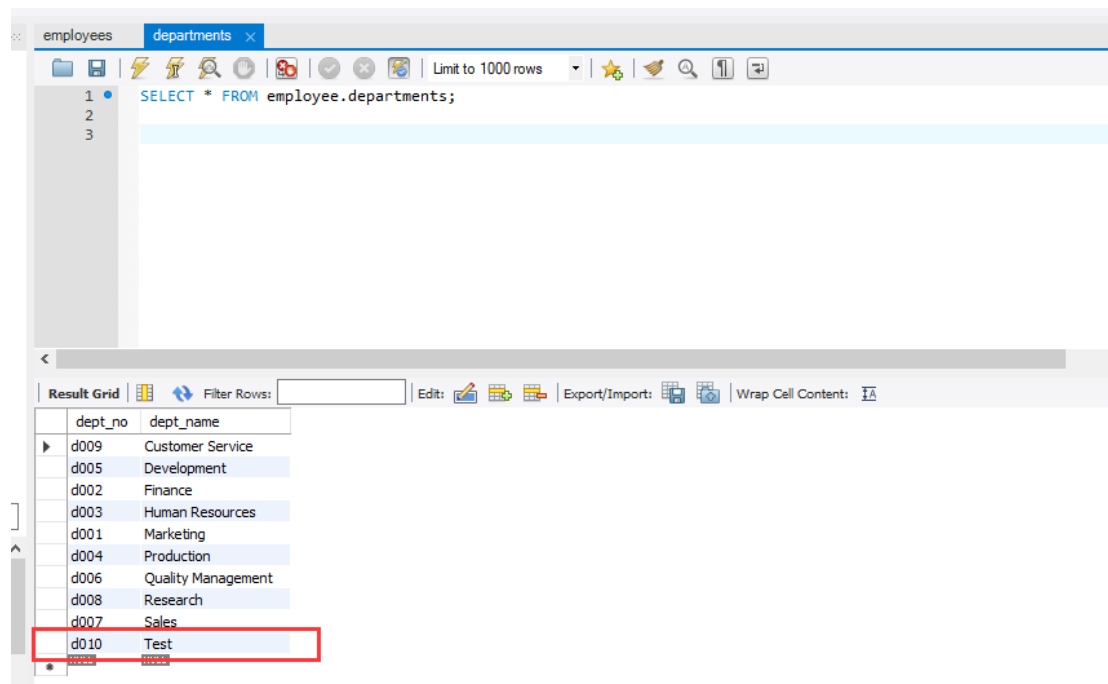
4.2.2 新增

比如：想要在 departments 表新增一条记录，可以在上图空白处输入
`insert into departments(dept_no, dept_name)`
`values('d010', 'Test');`

然后点击执行按钮：



可以检查一下新增数据：



细心的同学估计已经发现，这里新增的数据，并不是随便设置的，insert 语句要满足条件：dept_no 不能与已有 dept_no 重复，因为 dept_no 是主键；dept_name 不能与已有 dept_name 重复，因为 dept_name 是 Unique key。

思考题*:

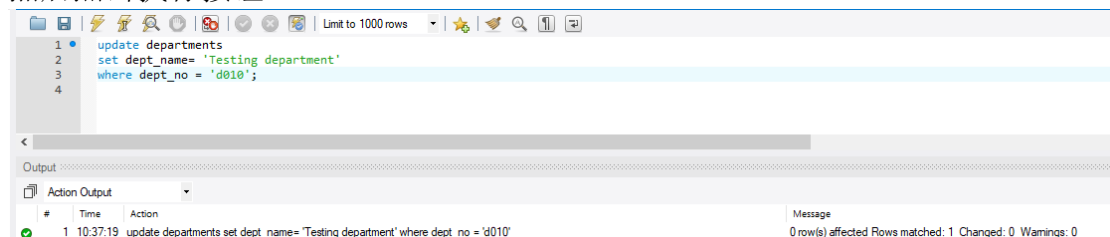
- a) 如果 insert 一条数据到 departments, 但 dept_no 或 dept_name 和已有数据重复, 会发生什么? 同学们请自己尝试一下。
- b) 对 dept_emp 表新增数据, 数据需满足哪些条件? 有什么机制可以保证数据正确性?

同学们可以思考这附加题并在实验截图文档里详细回答。

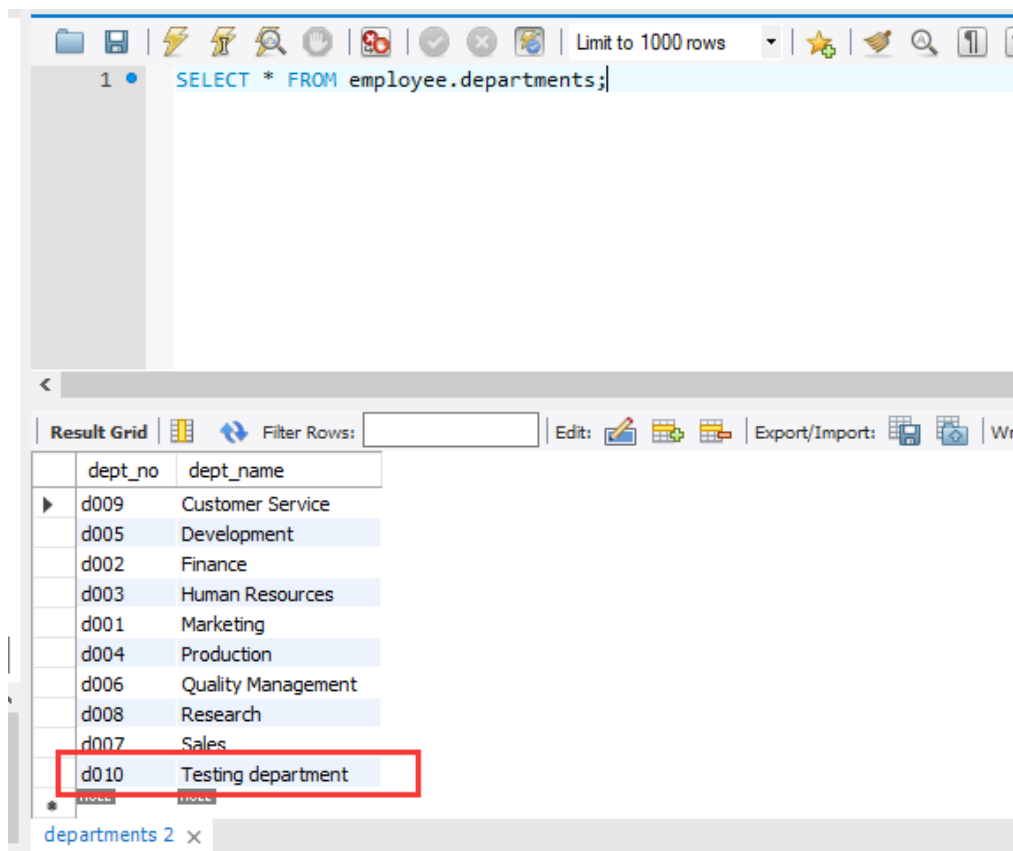
4.2.3 修改

比如: 想要在 departments 表修改一条记录, 可以在上图空白处输入
update departments
set dept_name= 'Testing department'
where dept_no = 'd010';

然后点击执行按钮:



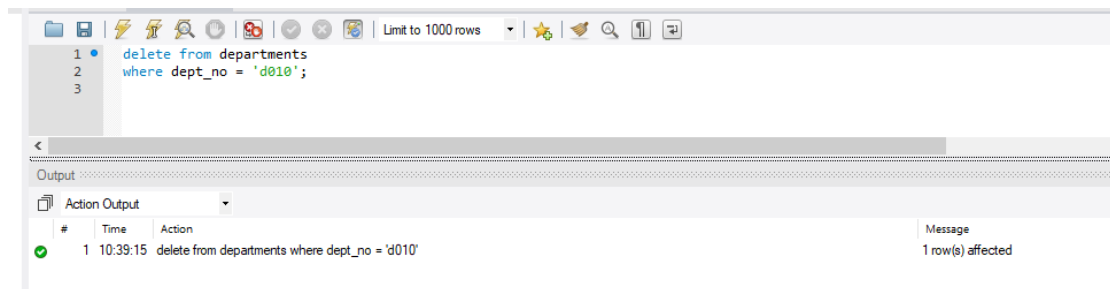
可以检查一下修改数据:



4.2.4 删除

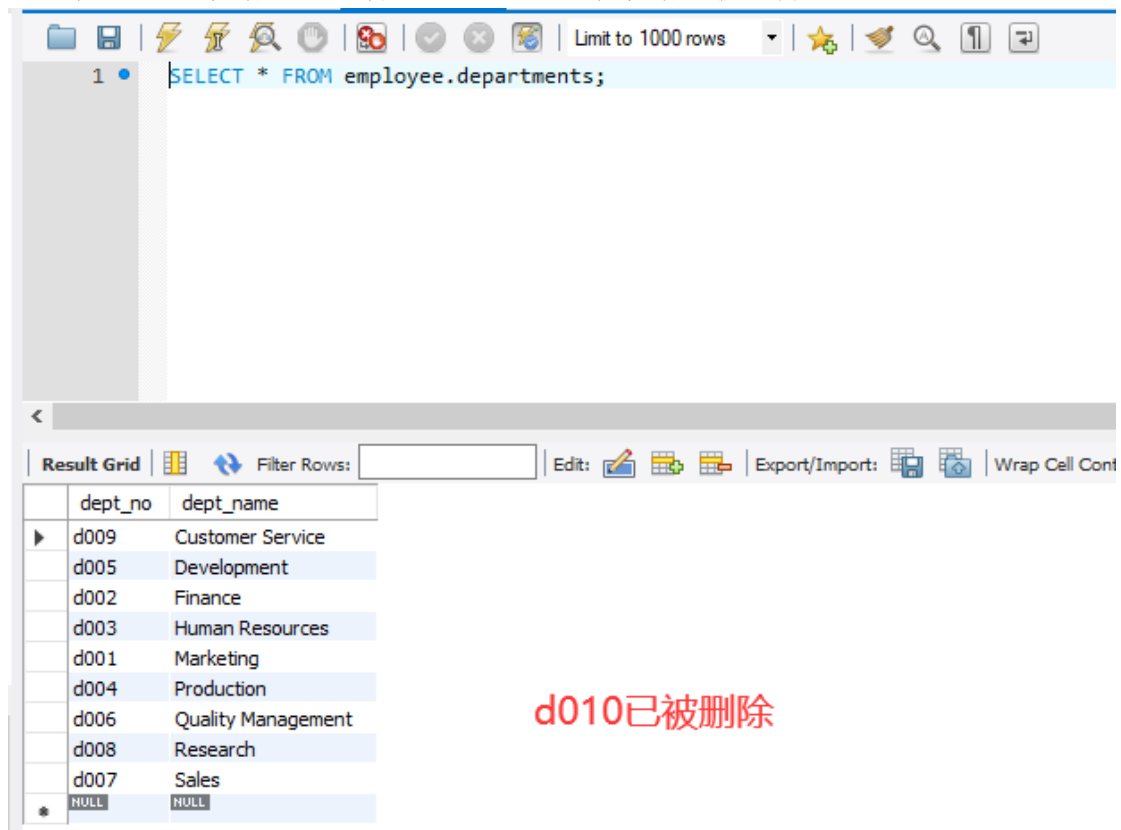
比如：想要在 departments 表删除一条记录，可以在上图空白处输入
delete from departments
where dept_no = 'd010';

然后点击执行按钮：



下面 output 窗口可以看到执行成功。

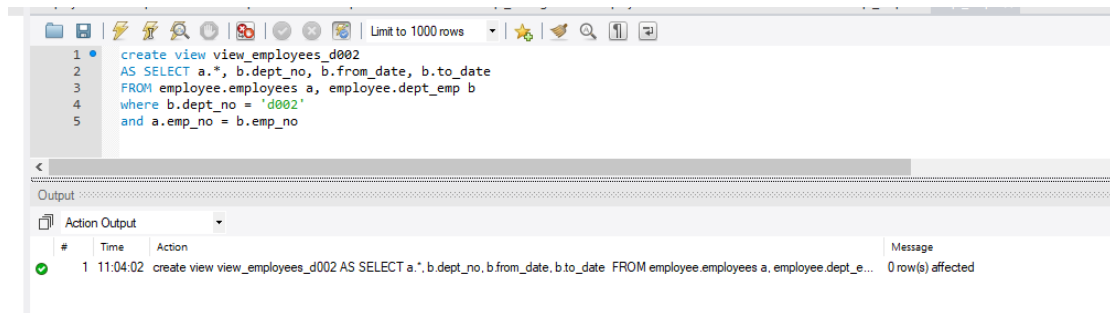
可以检查一下数据，可以看到“d010”这条数据已被删除：



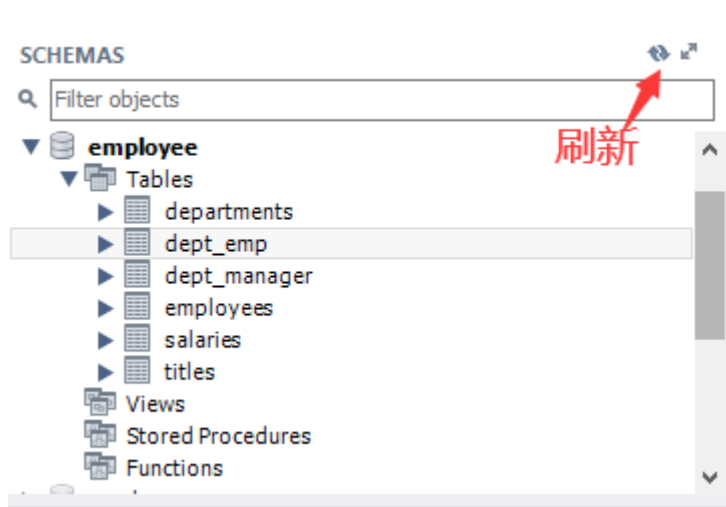
4.3 视图

1. 打开 SQL File，在空白处输入语句，然后执行：
create view view_employees_d002
AS SELECT a.*, b.dept_no, b.from_date, b.to_date
FROM employee.employees a, employee.dept_emp b

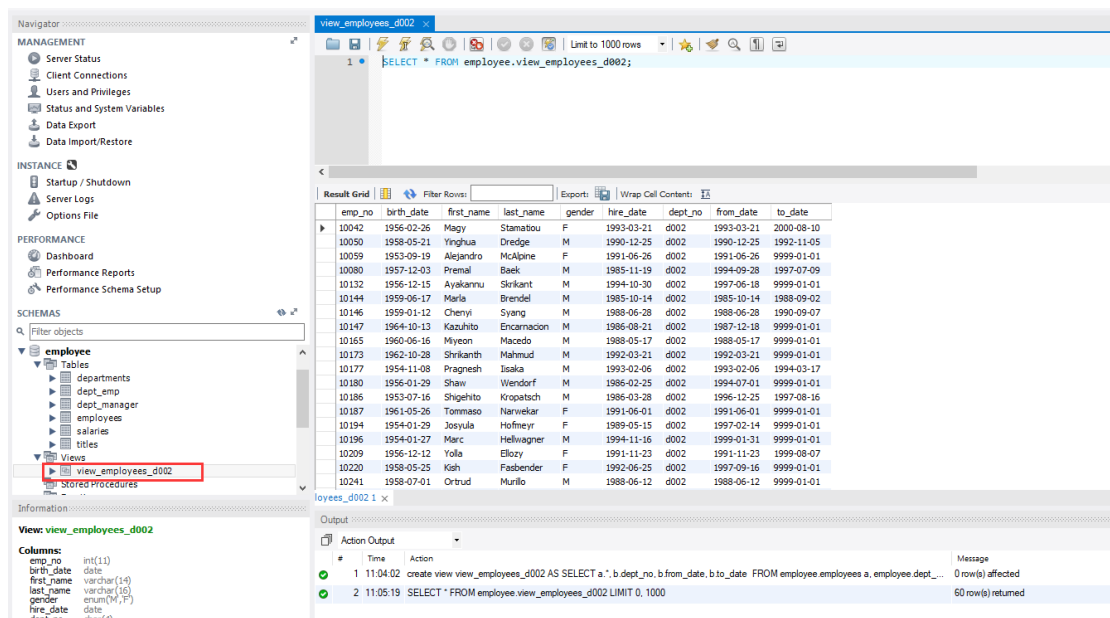
where b.dept_no = 'd002'
and a.emp_no = b.emp_no;



2. 在左边点刷新:



3. 查看视图里的记录:

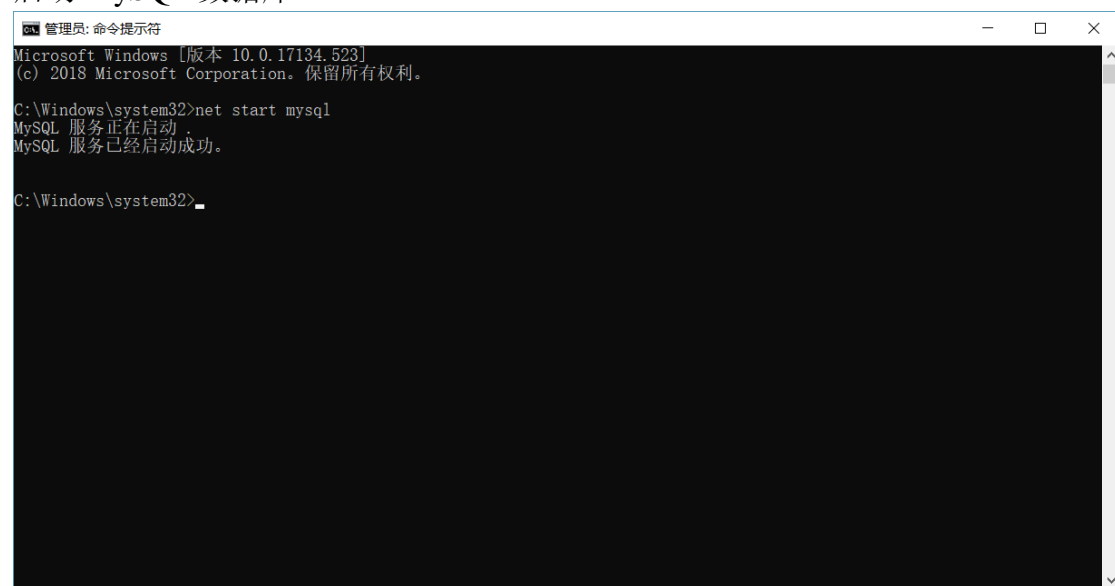


5 附录

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。



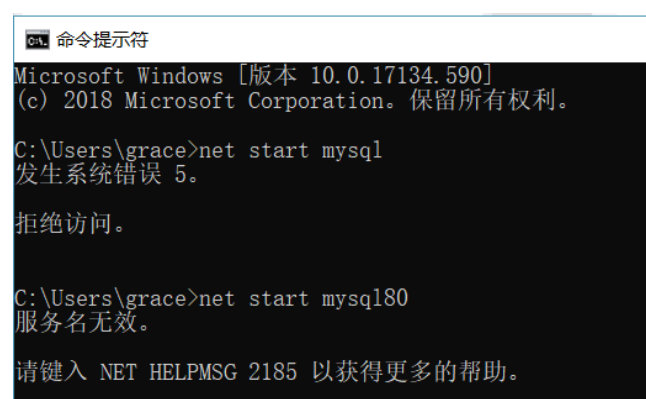
```
管理员: 命令提示符
Microsoft Windows [版本 10.0.17134.523]
(c) 2018 Microsoft Corporation。保留所有权利。

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。

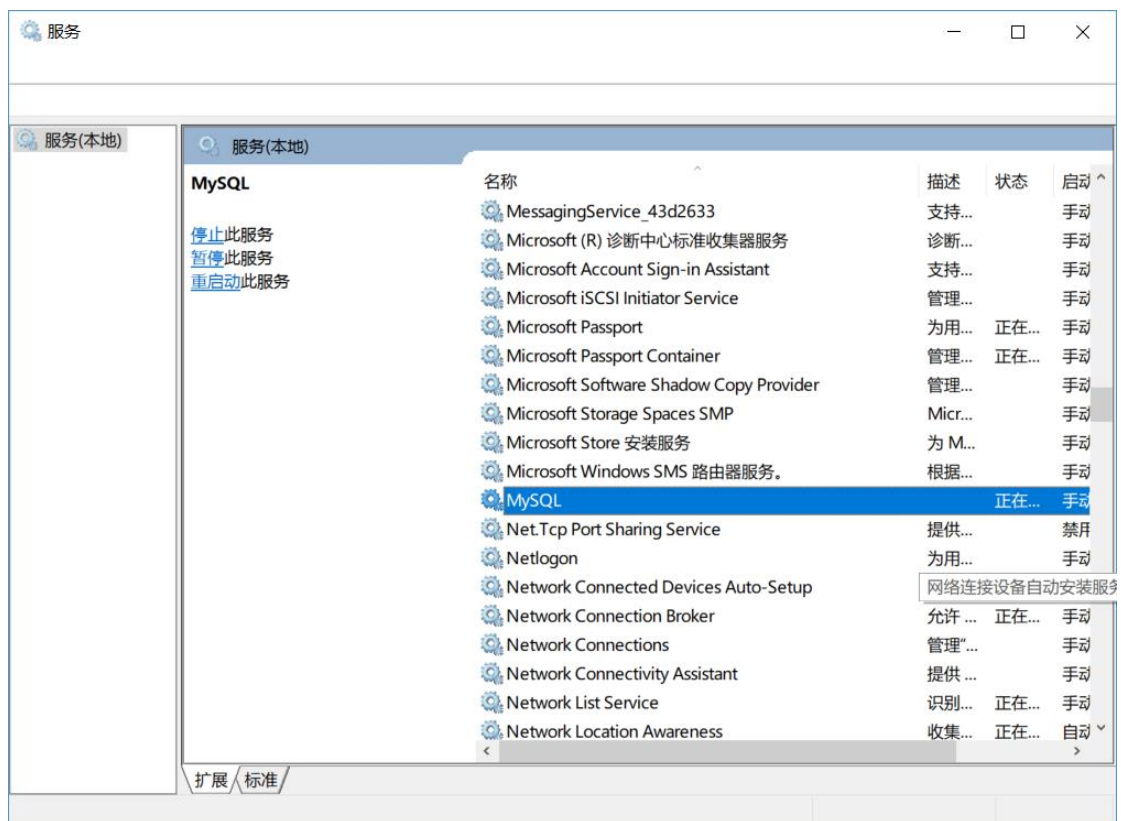
C:\Users\grace>net start mysql80
服务名无效。

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> _
```

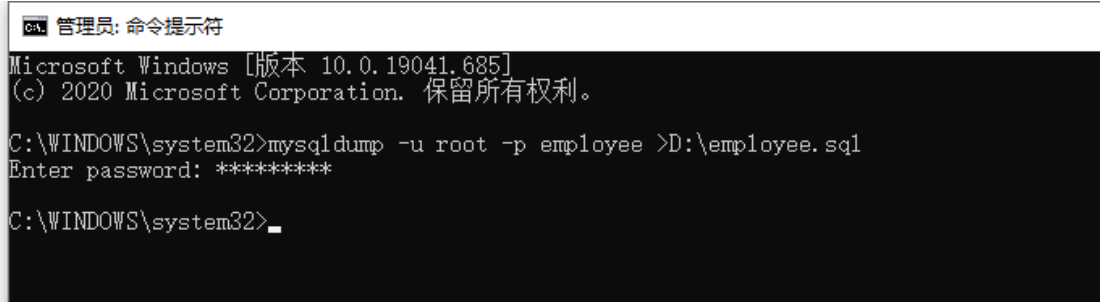
5.1.10 断开 MySQL 服务器连接

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 5.0.13, for Win64 (x64)
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 ('d000','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 附加内容*

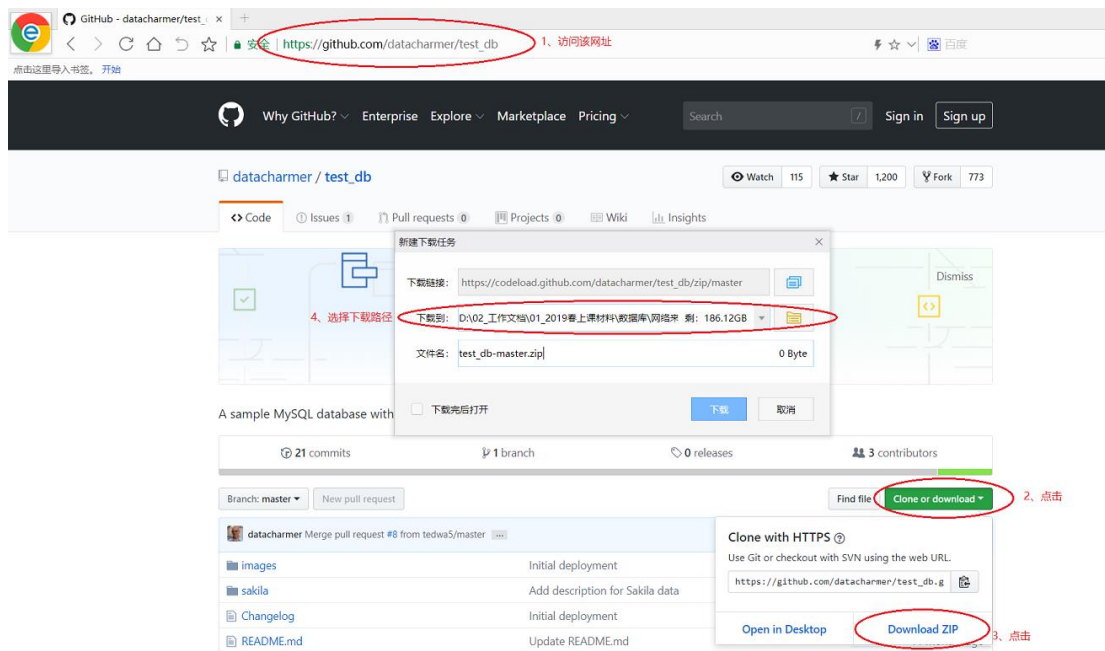
注：本节内容作为思考题，有兴趣的同学可选做。

一、导入 MySQL 官方提供的 Employees 数据库

Employees Sample Database 是官方自带的数据库，里面有几百万条数据。我们可以用该数据库进行 SQL 语句的练习以及对数据库的优化。

安装步骤：

- 1、下载



- 2、解压到本地某个目录，cmd 进入这个目录，执行
mysql -u root -p < employees.sql

```
命令提示符
D:\02_工作文档\01_2019春上课材料\数据库\网络来源材料\test_db-master\test_db-master>mysql -u root -p < employees.sql
Enter password: *****
INFO
CREATING DATABASE STRUCTURE
INFO
storage engine: InnoDB
INFO
LOADING departments
INFO
LOADING employees
INFO
LOADING dept_emp
INFO
LOADING dept_manager
INFO
LOADING titles
INFO
LOADING salaries
data_load_time_diff
00:01:16
```

- 3、验证是否装载成功，执行 mysql -u root -p < test_employees_md5.sql

```
D:\02_工作文档\01_2019春上课材料\数据库\网络来源材料\test_db-master\test_db-master>mysql -u root -p < test_employees_md5.sql
Enter password: *****
INFO
TESTING INSTALLATION
table_name      expected_records  expected_crc
employees       300024            4ec56ab5ba37218d187cf6ab09ce1aa1
departments     9                 d1af5e170d2d1591d776d5638d71fc5f
dept_manager    24                8720e2f0853ac9096b689c14664f847e
dept_emp        331603            ccf6fe516f990bdaa49713fc478701b7
titles 443308      bfa016c472df68e70a03facafabc0a8
salaries        2844047           fd220654e95aealb169624ffe3fca934
table_name      found_records     found_crc
employees       300024            4ec56ab5ba37218d187cf6ab09ce1aa1
departments     9                 d1af5e170d2d1591d776d5638d71fc5f
dept_manager    24                8720e2f0853ac9096b689c14664f847e
dept_emp        331603            ccf6fe516f990bdaa49713fc478701b7
titles 443308      bfa016c472df68e70a03facafabc0a8
salaries        2844047           fd220654e95aealb169624ffe3fca934
table_name      records_match     crc_match
employees       OK                ok
departments     OK                ok
dept_manager    OK                ok
dept_emp        OK                ok
titles OK        ok                ok
salaries        OK                ok
computation_time
00:00:12
summary result
CRC            OK
count          OK
```

看到这么多测试数据，是不是欣喜若狂？

二、查询诊断分析工具 Query Profiler

Query Profiler 是 MYSQL 自带的一种查询诊断分析工具，通过它可以分析出一条 SQL 语句的性能瓶颈在什么地方。Query Profiler 可以定位出一条 SQL 语句执行的各种资源消耗情况，比如 CPU、IO 等，以及该 SQL 执行所耗费的时间等。

步骤：

- 1、查看 profile 是否开启，数据库默认是不开启的。变量 profiling 是用户变量，每次都得重新启用。

```
mysql> show variables like "prof%";
+-----+-----+
| Variable_name | Value |
+-----+-----+
| profiling      | OFF   |
| profiling_history_size | 15    |
+-----+-----+
2 rows in set, 1 warning (0.00 sec)

mysql> set profiling = ON;
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql>
```

- 2、关闭查询缓存。

```
mysql> SHOW VARIABLES LIKE '%query_cache%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| have_query_cache | NO    |
+-----+-----+
1 row in set, 1 warning (0.01 sec)
```

如果没有关查询缓存的请修改配置文件，然后重启数据库。

- 3、执行你想执行的查询语句。
来几个复杂查询呗！
- 4、查看查询执行时间：show profiles;

```
mysql> show profiles;
+-----+-----+-----+
| Query_ID | Duration | Query |
+-----+-----+-----+
| 1 | 0.21064025 | select employees.* ,departments.dept_name, salaries.salary from employees, dept_emp, departments, salaries where dept_emp.dept_no = departments.dept_no and employee... |
| 2 | 0.39495700 | select employees.* ,departments.dept_name, salaries.salary from employees, dept_emp, departments, salaries where dept_emp.dept_no = departments.dept_no and employee... |
| 3 | 0.36456775 | select employees.* ,departments.dept_name, salaries.salary from employees, dept_emp, departments, salaries where dept_emp.dept_no = departments.dept_no and employee... |
| 4 | 0.38505900 | select employees.* ,departments.dept_name, salaries.salary from employees, dept_emp, departments, salaries where dept_emp.dept_no = departments.dept_no and employee... |
| 5 | 0.00064575 | explain select employees.* ,departments.dept_name, salaries.salary from employees, dept_emp, departments, salaries where dept_emp.dept_no = departments.dept_no and employee... |
| 6 | 0.01991400 | show global variables like 'query_cache%' |
| 7 | 0.00322625 | show variables like 'cache%' |
| 8 | 0.01661000 | explain select * from employees |
| 9 | 0.00691950 | SELECT DATABASE() |
| 10 | 0.05881625 | select * from employee |
| 11 | 0.01220825 | select * from department |
| 12 | 0.00483200 | select * from emp_dept |
| 13 | 0.00718150 | select a.*, b.dname from employee a, department b where a.dno = b.dno |
| 14 | 0.00046150 | select * from department |
```

5、查看 SQL 执行耗时详细信息：show profile for query [Query_ID];

```
mysql> show profile for query 13;
```

Status	Duration
starting	0.004549
checking permissions	0.000022
checking permissions	0.000006
Opening tables	0.000152
init	0.000013
System lock	0.000014
optimizing	0.000021
statistics	0.001379
preparing	0.000041
executing	0.000005
Sending data	0.000794
end	0.000010
query end	0.000005
waiting for handler commit	0.000013
query end	0.000008
closing tables	0.000011
freeing items	0.000111
cleaning up	0.000030

18 rows in set, 1 warning (0.00 sec)

6、查看运行时消耗的资源：CPU、I/O 等。

```
mysql> show profile cpu,block io for query 13;
```

Status	Duration	CPU_user	CPU_system	Block_ops_in	Block_ops_out
starting	0.004549	0.000000	0.000000	NULL	NULL
checking permissions	0.000022	0.000000	0.000000	NULL	NULL
checking permissions	0.000006	0.000000	0.000000	NULL	NULL
Opening tables	0.000152	0.000000	0.000000	NULL	NULL
init	0.000013	0.000000	0.000000	NULL	NULL
System lock	0.000014	0.000000	0.000000	NULL	NULL
optimizing	0.000021	0.000000	0.000000	NULL	NULL
statistics	0.001379	0.000000	0.000000	NULL	NULL
preparing	0.000041	0.000000	0.000000	NULL	NULL
executing	0.000005	0.000000	0.000000	NULL	NULL
Sending data	0.000794	0.015625	0.000000	NULL	NULL
end	0.000010	0.000000	0.000000	NULL	NULL
query end	0.000005	0.000000	0.000000	NULL	NULL
waiting for handler commit	0.000013	0.000000	0.000000	NULL	NULL
query end	0.000008	0.000000	0.000000	NULL	NULL
closing tables	0.000011	0.000000	0.000000	NULL	NULL
freeing items	0.000111	0.000000	0.000000	NULL	NULL
cleaning up	0.000030	0.000000	0.000000	NULL	NULL

18 rows in set, 1 warning (0.00 sec)

三、SQL 执行分析工具 EXPLAIN

使用 EXPLAIN 关键字可以模拟优化器执行 SQL 查询语句，从而知道 SQL 语句的执行计划，分析你的查询语句或是表结构的性能瓶颈。

使用方式：explain + SQL 语句

```
mysql> explain select * from employees;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	employees	NULL	ALL	NULL	NULL	NULL	NULL	299556	100.00	NULL

1 row in set, 1 warning (0.02 sec)

通过 EXPLAIN，我们可以得到以下结果：

- id: SELECT 查询的标识符. 每个 SELECT 都会自动分配一个唯一的标识符.
- select_type: SELECT 查询的类型.
- table: 查询的是哪个表
- partitions: 匹配的分区
- type: join 类型
- possible_keys: 此次查询中可能选用的索引
- key: 此次查询中确切使用到的索引.
- ref: 哪个字段或常数与 key 一起被使用
- rows: 显示此查询一共扫描了多少行. 这个是一个估计值.
- filtered: 表示此查询条件所过滤的数据的百分比
- extra: 额外的信息

在 MySQL Workbench 里面也能找到执行计划:

The screenshot displays the MySQL Workbench interface. At the top, a SQL query is entered in the editor:

```
select *
from employee a, works_on b, department c
where a.essn = b.essn
and a.dno = c.dno;
```

Below the editor, the 'Result Grid' tab is active, showing a table of employee data. To the right of the grid, a sidebar contains icons for 'Result Grid', 'Form Editor', 'Field Types', 'Query Stats', and 'Execution Plan'. The 'Execution Plan' icon is highlighted with a red box.

At the bottom of the interface, the 'Visual Explain' tab is active, showing the execution plan for the query. The plan is a nested loop join. It starts with a 'Full Table Scan' on table 'c' (cost 0.75, 5 rows). This is followed by a 'block nested loop' join with table 'a' (cost 31.5, 61 rows). Finally, it performs a 'Non-Unique Key Lookup' on table 'b' (cost 13.18, 1 row). The total query cost is 44.68.

有兴趣的同学们可以自己找资料学习。

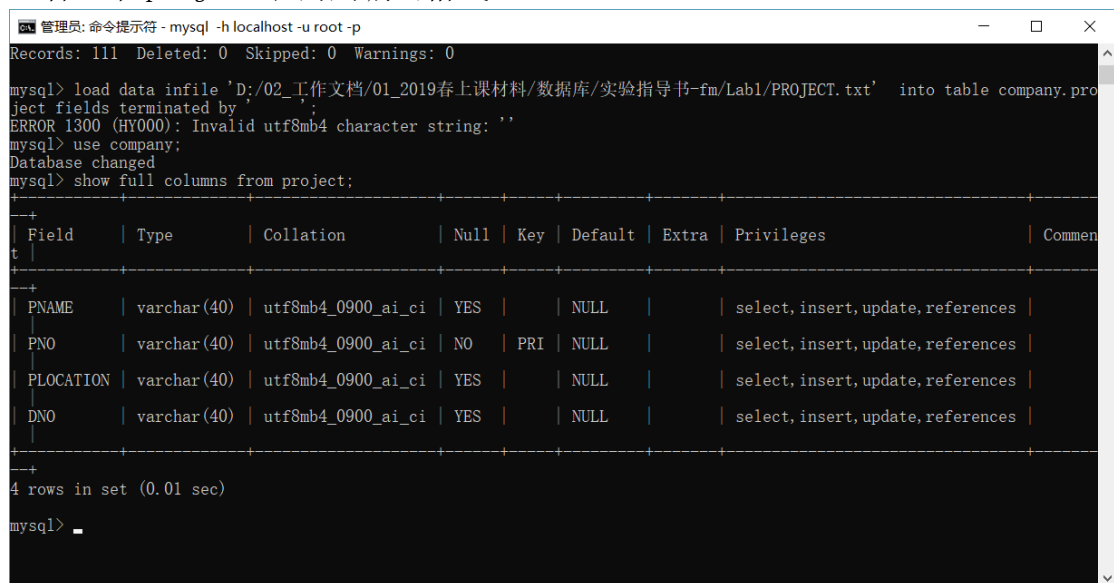
5.3 常见问题

1、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: ''
```

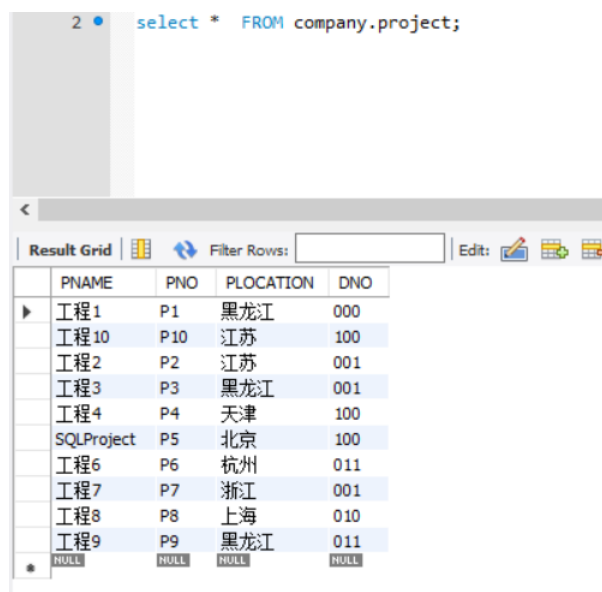
查看这个 project 表的列编码格式:



```
mysql> show full columns from project;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Field | Type | Collation | Null | Key | Default | Extra | Privileges | Comment |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| PNAME | varchar(40) | utf8mb4_0900_ai_ci | YES | | NULL | | select, insert, update, references | 
| PNO | varchar(40) | utf8mb4_0900_ai_ci | NO | PRI | NULL | | select, insert, update, references | 
| PLOCATION | varchar(40) | utf8mb4_0900_ai_ci | YES | | NULL | | select, insert, update, references | 
| DNO | varchar(40) | utf8mb4_0900_ai_ci | YES | | NULL | | select, insert, update, references | 
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)
```

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;
```

	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)
```

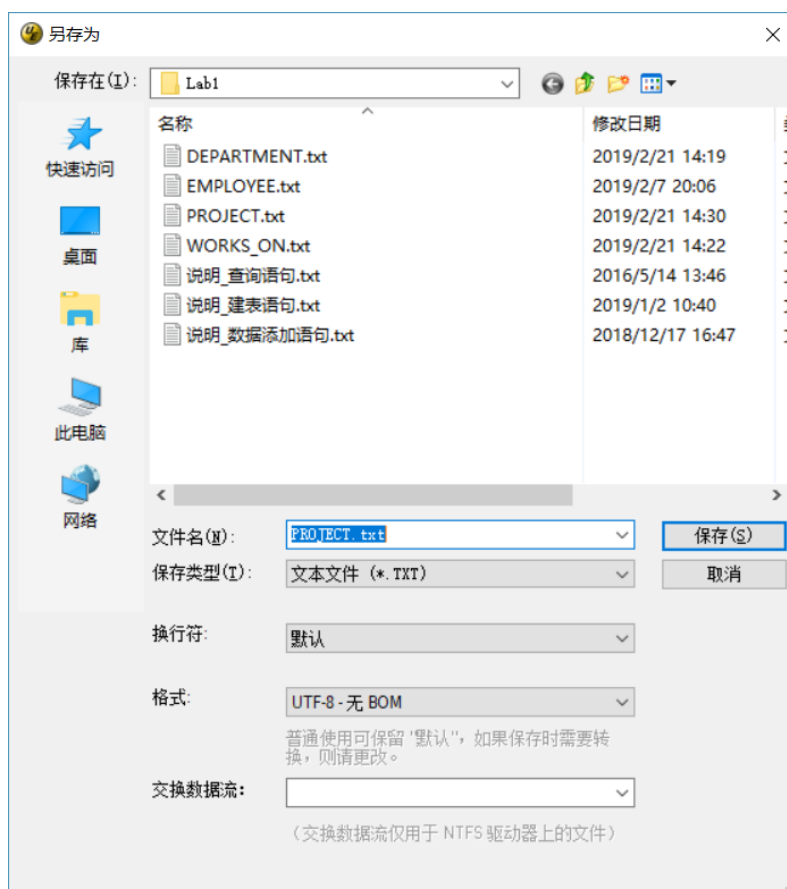
看出问题了吗？第一条记录的第一个字段前有个问号。

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。

2、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.project 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!

3、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。

4、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、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
陈晓冬	08	清远市北江三路海关大楼6楼	3000.00	24	100
陈秀芬	11	广州市花都区建设北路79号	5500.00	01	100
成秀山	13	韶关市环园东路1号	2600.00	24	100
仇腊梅	16	从化市街口街关前路1栋	1900.00	01	100
戴金辉	17	广州市站南路4号流花邮政大院	2800.00	01	100
翟雷	18	清远市北江三路	3300.00	01	100
丁素琴	20	广州市南沙区南沙街进港大道1	4000.00	NULL	100
张红	22	广州市萝岗区开创大道北1168号	3500.00	24	100
伏严成	24	肇庆市端州七路13号	2000.00	20	100
皋歌	26	佛山市顺德区容桂街容奇大道东5号	4500.00	01	100
管小云	28	广州市天河区东站1号	7000.00	01	100
侯志玲	31	广州市天河区东站2号	2900.00	01	100
李铜然	33	广州市天河区东站4号	3500.00	01	100
蒋雯	38	广州市天河区东站9号	2700.00	24	100
李金凤	40	广州市天河区东站11号	4800.00	01	100
潘杰	43	广州市天河区东站14号	4400.00	01	100
邱文干	52	广州市天河区东站23号	6300.00	01	100
	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：

```
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
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

正常的应该是这样的，看出区别了吗？

命令提示符 - mysql -h localhost -u root -p

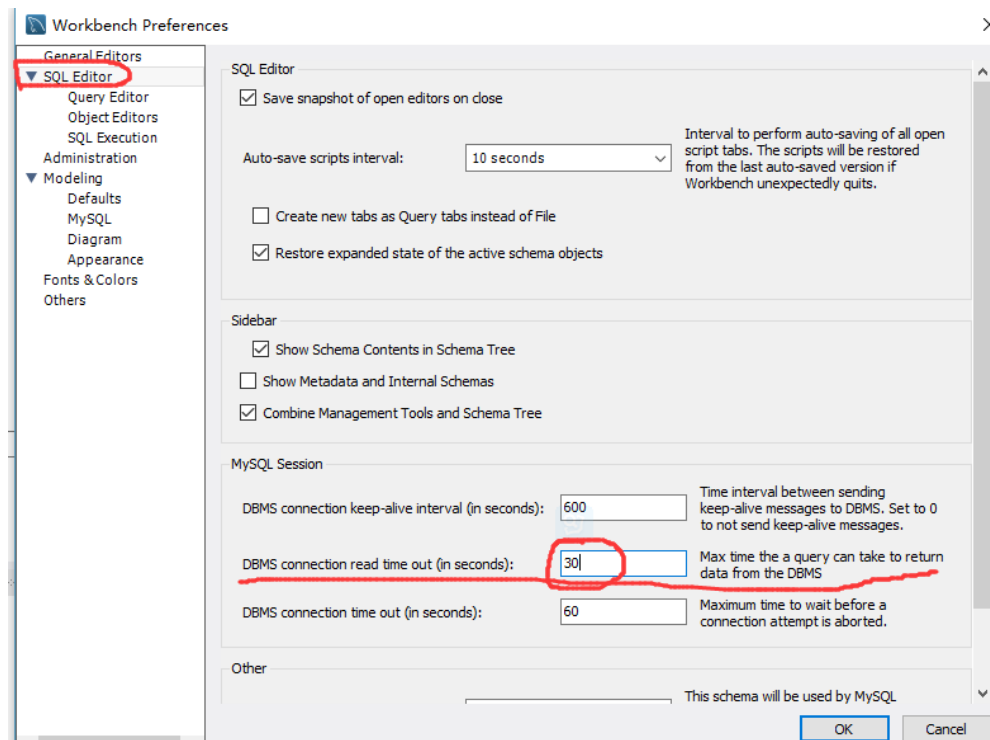
```
mysql> select * from employee;
```

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
金存连	01	广东省广州市石牌西路68号	2000.00	20	100
金江	02	广州市番禺区市桥清河东路41号	1500.00	14	011
陈超云	03	佛山市江西南路13号	3000.00	02	011
陈浩	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
陈宏梅	06	肇庆市端州七路13号	2800.00	NULL	001
陈金源	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
陈小光	12	广州市机场路561号	3600.00	NULL	010
陈秀芳	13	韶关市环园东路1号	2600.00	24	100
陈桂华	14	佛山市禅城区汾江西路13号	4600.00	NULL	011
陈英因	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
曾浩	20	广州市南沙区南沙街道港大道1	4000.00	NULL	100
丁碧明	21	广州市白云区	2900.00	07	011
丁嘉号	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	3000.00	20	100
冯海	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道容奇大道东5号	4500.00	01	100
卓德华	27	佛山市三水西南镇广海大道中17号	5600.00	15	010
卓城	28	广州市天河区东山1号	7000.00	01	100
高伟	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景区内	7500.00	07	011
官小云	31	广州市天河区东站2号	2900.00	01	100
何伟刚	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
黄志刚	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
纪海燕	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
康海霞	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	5500.00	02	011
金金凤	43	广州市天河区东站14号	4400.00	01	100
李宏	44	广州市天河区东站15号	7200.00	02	011

6、Error Code: 2013. Lost connection to MySQL server during query 30.000 sec

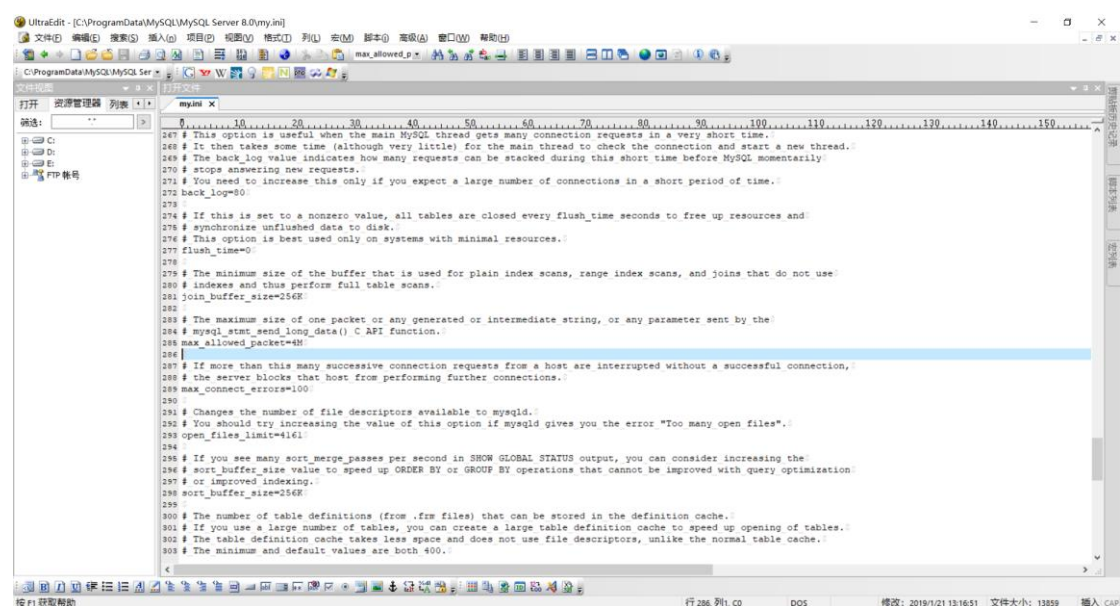
解决办法:

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



7、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：

```
C:\> 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。

8、'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


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

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。

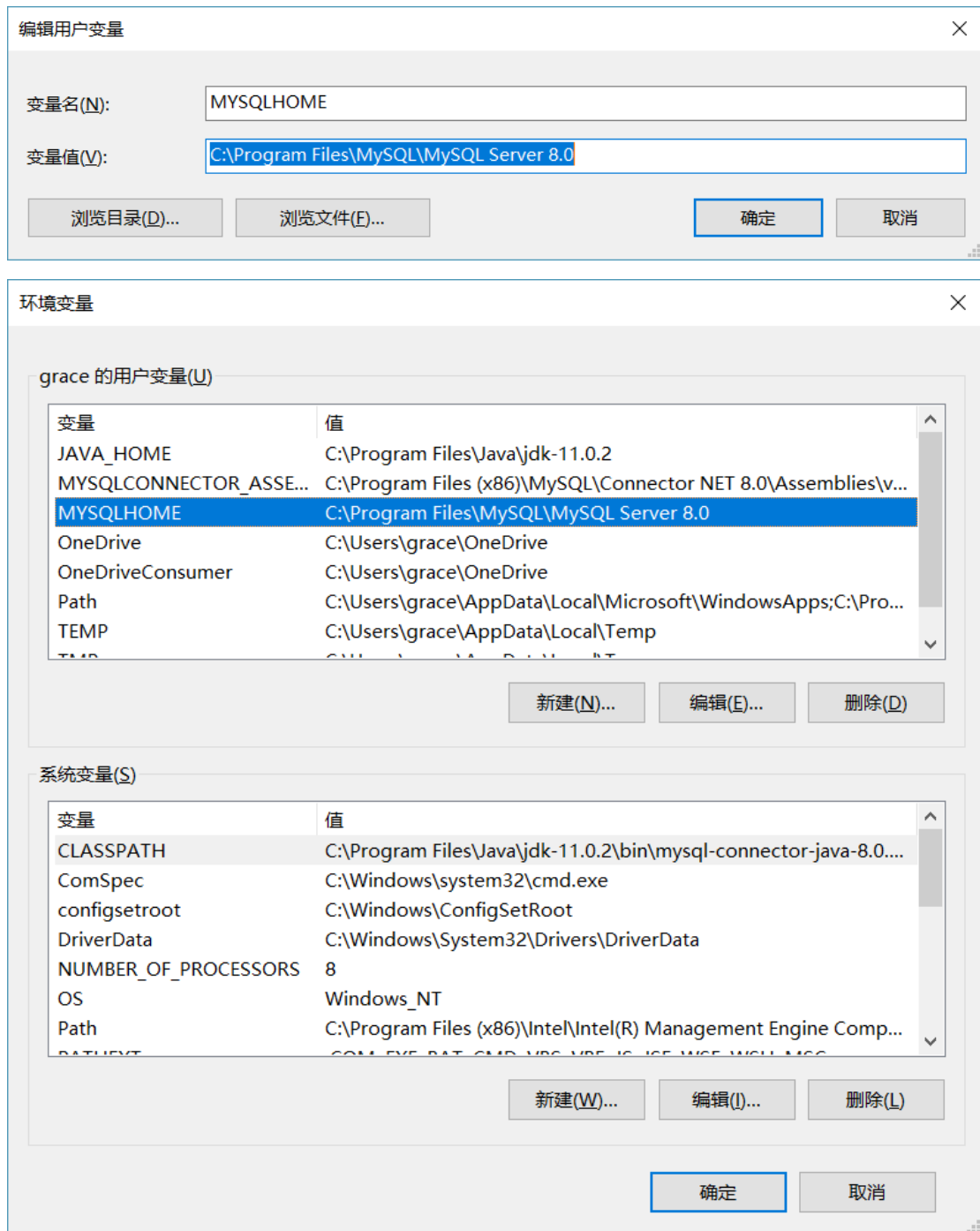
C:\Windows\system32>
```

这个问题说明没有设置环境变量：

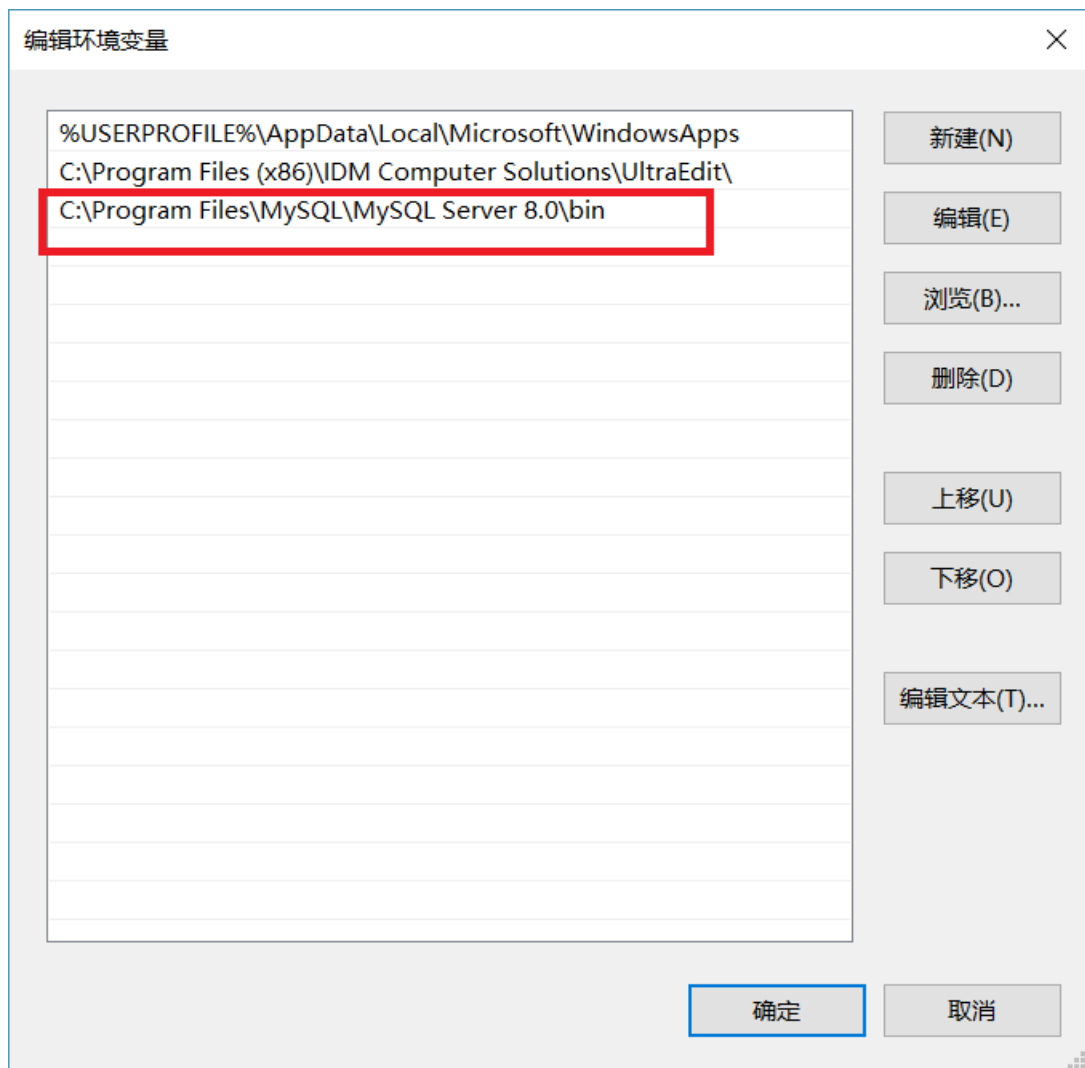
- 1、打开电脑的属性，找到高级系统设置



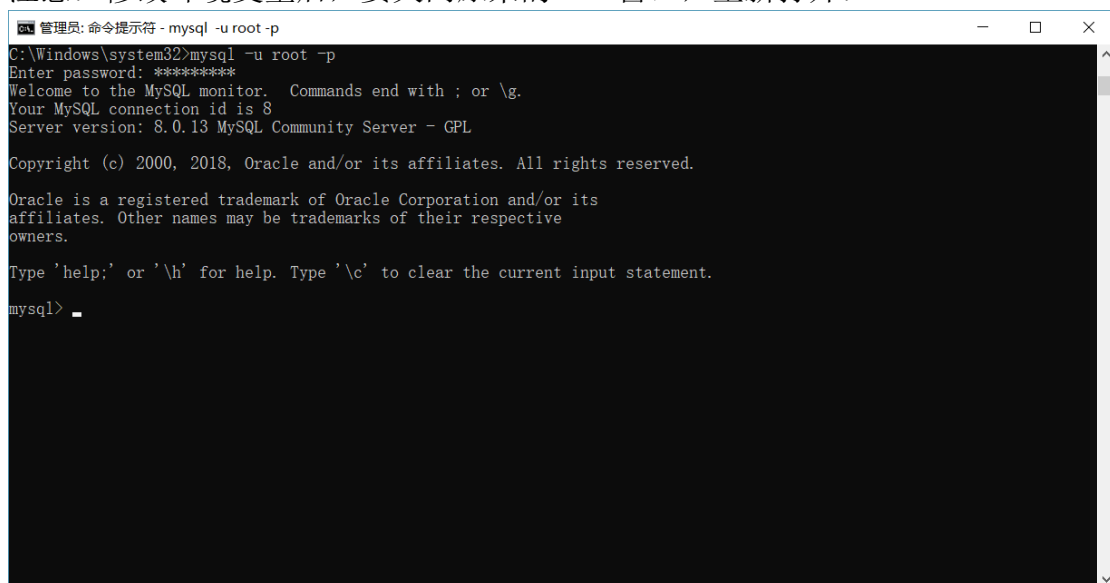
2、新建用户变量：变量名 MYSQLHOME，变量值 MySQL 安装路径



3、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：



9、ERROR 1045 (28000): Access denied for user

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

检查密码是不是输错了！！

```
管理员: 命令提示符 - mysql -u root -p
Microsoft Windows [版本 10.0.17134.590]
(c) 2018 Microsoft Corporation。保留所有权利。

C:\Windows\system32>mysql -u root -p
Enter password: ****
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)

C:\Windows\system32>mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
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>
```

10、中文显示乱码

```
mysql> select *from employee;
```

ename	essn	address	salary	superssn	dno
铭跨椅整斐姘	00	寡垮容甯傲 ㄆ 婁冲脰鰓狗睨鋸板輝鐸卞輝潼 ㄣ 83 鑒?			1000.00 24
浠撒楓鑾?	01	寡夸篋鏽 估箍宸殆競鏽崇境媛胯硯68鑒?	2000.00	20	100
浠摔孩	02	寡垮容甯 偲陽級哄脰甯俗 ㄣ 妮呖泐涓涓硯41鑒?	1500.00	14	011
閨堦枯浜?	03	浣浹北甯 俗本姪燭夕鑒?3鑒?	3000.00	02	011
閨墾瓊	04	浣浹北甯 倣蝓姻脰杳倣輝潼 ㄣ 綉涓爕硯1鑒峰 ㄣ 閨?	4000.00	06	001
閨塢涑纒?	05	浣浹北甯 倣 ㄣ 衰峰脰淪現 ㄣ ㄣ 涓?3鑒?	3500.00	09	000
閨塢峻鑼?	06	鑼固簡甯 偲 ㄣ 底息錢鑒?3鑒?	2800.00		001
閨墾噲涓?	07	寡垮容甯 偲熒鏽男夕鑒?8鑒?	4500.00	14	011
閨墾附涓?	08	妮些繙甯 倣資姪燭竿鑒 ㄣ 揭鐸冲 ㄣ 好?好?	3000.00	24	100
閨塢鈎	09	閨蹼另甯 偲暢銀 ㄣ 篋鑒?鏽?好?	3000.00		000
閨塢涓涓?	10	浜曼誕甯 倣緩璫惧蝓鑒?0鑒?	2500.00	09	000
閨塢樁鏼?	11	寡垮容甯 偲姪閨藉脰甯鴻 ㄣ 錯樞硯79鑒?	5500.00	01	100
閨塢確鑼?	12	寡垮容甯 俗清鐸鴻硯561鑒?	3600.00		010
閨塢 鐸?	13	閨蹼另甯 偲暢銀 ㄣ 篋鑒?鑒?	2600.00	24	100
閨塢墾鐸?	14	浣浹北甯 偲 錯宸脰姪炬睨媛胯硯13鑒?	4600.00		011
閨塢厝鐸?	15	寡垮容甯 倣 ㄆ 婁冲脰鰓狗睨鋸板輝鐸辰埠鑒?1鑒?	2700.00	12	010
鋸樞 灑?	16	浠宸疑甯 偲 ㄣ 鐸冲埤鑒?鏽?	1900.00	01	100
浠凶匪姊?	17	寡垮容甯 偲瑤鐸樞硯4鑒鋒襍璫過偲鎮垮 ㄣ 閨?	2800.00	01	100
鋸撒噲杳?	18	妮些繙甯 倣資姪燭竿鑒?	3300.00	01	100
閨撒揭鐸?	19	寡垮容甯 偲陽級哄脰妮呖泐涓涓硯43鑒?	2000.00	12	010
紗燭睜	20	寡垮容甯 倣蝓婁脰鐸格璫蹼樞纒 ㄣ 閨?	4000.00		100

修改编码格式为 gbk

6 参考资料

Abraham Silberschatz, Henry F.Korth. 《数据库系统概念（第六版）》
《MySQL 中文参考手册》（MySQLBook.chm）