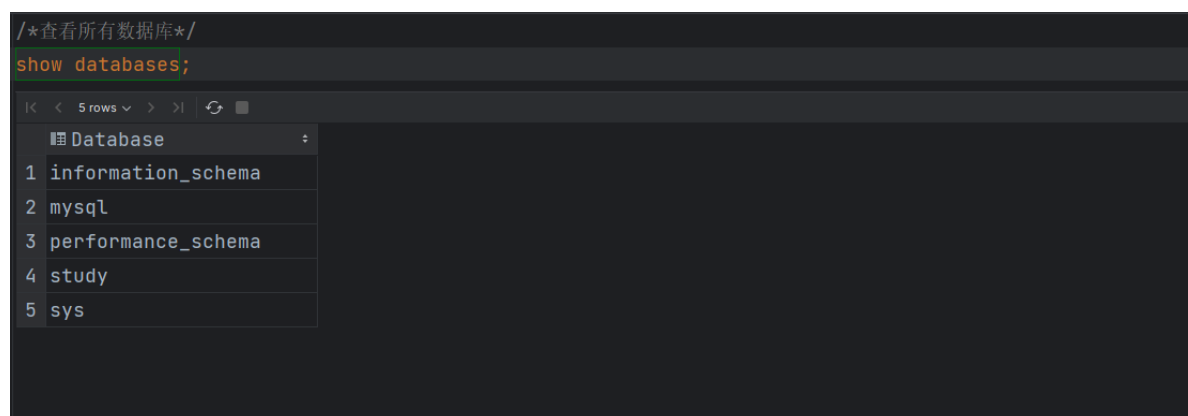


4. MySQL演示使用

4.1 MySQL的使用演示

1、查看所有数据库

```
show databases;
```



“**information_schema**”是 MySQL 系统自带的数据库，主要**保存 MySQL 数据库服务器的系统信息**，比如数据库的名称、数据表的名称、字段名称、存取权限、数据文件所在的文件夹和系统使用的文件夹，等等

“**performance_schema**”是 MySQL 系统自带的数据库，可以用来**监控 MySQL 的各类性能指标**。

“**sys**”数据库是 MySQL 系统自带的数据库，主要作用是以一种更容易被理解的方式**展示 MySQL 数据库服务器的各类性能指标**，帮助系统管理员和开发人员监控 MySQL 的技术性能。

“**mysql**”数据库**保存了 MySQL 数据库服务器运行时需要的系统信息**，比如数据文件夹、当前使用的字符集、约束检查信息，等等

为什么 Workbench 里面我们只能看到“demo”和“sys”这 2 个数据库呢？

这是因为，**Workbench 是图形化的管理工具**，**主要面向开发人员**，“demo”和“sys”这 2 个数据库已经够用了。如果有特殊需求，比如，需要监控 MySQL 数据库各项性能指标、直接操作 MySQL 数据库系统文件等，可以由 DBA 通过 SQL 语句，查看其它的系统数据库。

2、创建自己的数据库

```
create database <数据库名> [[charset] [option]];  
# 创建study数据库  
create database study;
```

注意：如果数据库名是一些关键字时，需要在数据库名称两边加`号。

charset：指定数据库使用的字符集

- option：有UTF-8, GBK, Latin1等

3、使用自己的数据库

```
use <数据库名>;  
# 使用数据库  
use study;
```

说明：如果没有使用use语句，后面针对数据库的操作也没有加“数据名”的限定，那么会报“ERROR 1046 (3D000): No database selected”（没有选择数据库）

使用完use语句之后，如果接下来的SQL都是针对一个数据库操作的，那就不用重复use了，如果要针对另一个数据库操作，那么要重新use。

4、查看某个库的所有表格

```
show tables;  
show tables from <数据库名>;
```

5、创建新的表格

```
create table <表名称> {  
    字段名 数据类型,  
    字段名 数据类型,  
};
```

说明：如果是最后一个字段，后面就用加逗号，因为逗号的作用是分割每个字段。

```
#创建学生表
create table student(
    id int,
    name varchar(20) #说名字最长不超过20个字符
);
```

6、查看一个表的数据

```
select * from <表名称>;
# 查看学生表数据
select * from student;
```

7、添加一条记录

```
insert into 表名称 values(值列表);
#添加两条记录到student表中
insert into student values(1,'张三');
insert into student values(2,'李四');
```

会出现如下错误：

```
study> insert into student values(1,'张三')
[2023-07-12 22:01:35] [HY000][1366] Incorrect string value: '\xE5\xBC\xA0\xE4\xB8\x89' for column 'name' at row 1
[2023-07-12 22:01:35] [HY000][1366] Incorrect string value: '\xE5\xBC\xA0\xE4\xB8\x89' for column 'name' at row 1
```

报错原因是字符集有问题

8、查看表的创建信息

```
show create table <表名称>\G
```

```
show create table student;
```

Table	Create Table
1 student	CREATE TABLE `student` (`id` int(11) DEFAULT NULL, `name` varchar(20) DEFAULT NULL) ENGINE=InnoDB DEFAULT CHARSET=latin1

上面的结果显示student的表格的默认字符集是“latin1”不支持中文。

9、查看数据库的创建信息

查看数据库创建信息

```
show create <数据库名>;
```

02-4.1. 查看数据库的创建信息

```
show create database study;
```

Database	Create Database
1 study	CREATE DATABASE `study` /*!40100 DEFAULT CHARACTER SET latin1 */

上面的结果显示atguigudb数据库也不支持中文，字符集默认是latin1。

10、删除表格

```
drop table <表名称>;
```

删除学生表

```
drop table student;
```

11、删除数据库

```
drop database <数据库名>;
```

#删除study数据库

```
drop database study;
```

4.2 MySQL的编码设置

MySQL5.7中

问题再现：命令行操作sql乱码问题

问题再现：命令行操作sql乱码问题

```
INSERT INTO t_stu VALUES(1,'张三','男');  
ERROR 1366 (HY000): Incorrect string value:  
'\xD5\xC5\xC8\xFD' for column 'sname' at  
row 1
```

问题解决

步骤1：查看编码命令

```
show variables like 'character_%';  
show variables like 'collation_%';
```

步骤2：修改mysql的数据目录下的my.ini配置文件

```
[mysql] #大概在63行左右，在其下添加  
...  
default-character-set=utf8 #默认字符集  
[mysqld] # 大概在76行左右，在其下添加  
...  
character-set-server=utf8  
collation-server=utf8_general_ci
```

注意：建议修改配置文件使用notepad++等高级文本编辑器，使用记事本等软件打开修改后可能会导致文件编码修改为“含BOM头”的编码，从而服务重启失败。

步骤3：重启服务

步骤4：查看编码命令

```
show variables like 'character_%';  
show variables like 'collation_%';
```

```
mysql> show variables like 'character_%';
```

variable_name	value
character_set_client	utf8
character_set_connection	utf8
character_set_database	utf8

```
| character_set_filesystem | binary
|
| character_set_results    | utf8
|
| character_set_server     | utf8
|
| character_set_system     | utf8
|
| character_sets_dir       | C:\Program Files\MySQL\MySQL
Server 5.7\share\charsets\
+-----+-----+
8 rows in set, 1 warning (0.00 sec)
```

```
mysql> show variables like 'collation_%';
+-----+-----+
| Variable_name | value          |
+-----+-----+
| collation_connection | utf8_general_ci |
| collation_database   | utf8_general_ci |
| collation_server     | utf8_general_ci |
+-----+-----+
3 rows in set, 1 warning (0.00 sec)
```

如果是以上配置就说明对了。接着我们就可以新创建数据库、新创建数据表，接着添加包含中文的数据了。

MySQL8.0中

在MySQL 8.0版本之前，默认字符集为latin1，utf8字符集指向的是utf8mb3。网站开发人员在数据库设计的时候往往会将编码修改为utf8字符集。如果遗忘修改默认的编码，就会出现乱码的问题。**从MySQL 8.0开始**，数据库的**默认编码改为 utf8mb4**，从而避免了上述的乱码问题。

6. MySQL目录结构与源码

6.1 主要目录结构

MySQL的目录结构	说明
bin目录	所有MySQL的可执行文件。如：mysql.exe
MySQLInstanceConfig.exe	数据库的配置向导，在安装时出现的内容
data目录	系统数据库所在的目录
my.ini文件	MySQL的主要配置文件
c:\ProgramData\MySQL\MySQL Server 8.0\data\	用户创建的数据库所在的目录

6.2 MySQL 源代码获取

首先，你要进入 MySQL 下载界面。这里你不要选择用默认的“Microsoft Windows”，而是要通过下拉栏，找到“Source Code”，在下面的操作系统版本里面，选择 Windows（Architecture Independent），然后点击下载。

接下来，把下载下来的压缩文件解压，我们就得到了 MySQL 的源代码。

MySQL 是用 C++ 开发而成的，我简单介绍一下源代码的组成。

mysql-8.0.22 目录下的各个子目录，包含了 MySQL 各部分组件的源代码：

名称	修改日期	类型	大小
client	2020-09-23 15:04	文件夹	
cmake	2020-09-23 15:04	文件夹	
components	2020-09-23 15:04	文件夹	
Docs	2020-09-23 15:04	文件夹	
doxygen_resources	2020-09-23 15:04	文件夹	
extra	2020-09-23 15:04	文件夹	
include	2020-09-23 15:04	文件夹	
libbinlogevents	2020-09-23 15:04	文件夹	
libbinlogstandalone	2020-09-23 15:04	文件夹	
libmysql	2020-09-23 15:04	文件夹	
libservers	2020-09-23 15:04	文件夹	
man	2020-09-23 15:04	文件夹	
mysql-test	2020-09-23 15:04	文件夹	
mysys	2020-09-23 15:04	文件夹	
packaging	2020-09-23 15:04	文件夹	
plugin	2020-09-23 15:04	文件夹	
router	2020-09-23 15:04	文件夹	
scripts	2020-09-23 15:04	文件夹	
share	2020-09-23 15:04	文件夹	
source_downloads	2020-09-23 15:04	文件夹	
sql	2020-09-23 15:05	文件夹	

- sql 子目录是 MySQL 核心代码;
- libmysql 子目录是客户端程序 API;
- mysql-test 子目录是测试工具;
- msys 子目录是操作系统相关函数和辅助函数;

7. 常见问题的解决(课外内容)

问题1: root用户密码忘记, 重置的操作

1: 通过任务管理器或者服务管理, 关掉mysqld(服务进程)

2: 通过命令行+特殊参数开启mysqld mysqld -- defaults-file="D:\ProgramFiles\mysql\MySQLServer5.7Data\my.ini" --skip-grant-tables

3: 此时, mysqld服务进程已经打开。并且不需要权限检查

4: mysql -uroot 无密码登陆服务器。另启动一个客户端进行

5: 修改权限表

(1) use mysql;

(2) update user set authentication_string=password('新密码')
where user='root' and Host='localhost';

(3) flush privileges;

6: 通过任务管理器, 关掉mysqld服务进程。

7: 再次通过服务管理, 打开mysql服务。

8: 即可用修改后的新密码登陆。

问题2: 命令行客户端的字符集问题

```
mysql> INSERT INTO t_stu VALUES(1, '张三', '男');  
ERROR 1366 (HY000): Incorrect string value:  
'\xD5\xC5\xC8\xFD' for column 'sname' at  
row 1
```


原因：服务器端认为你的客户端的字符集是utf-8，而实际上你的客户端的字符集是GBK。

查看所有字符集：SHOW VARIABLES LIKE 'character_set_%';

问题3：修改数据库和表的字符编码

修改编码：

(1)先停止服务，（2）修改my.ini文件（3）重新启动服务

说明：

如果是在修改my.ini之前建的库和表，那么库和表的编码还是原来的Latin1，要么删了重建，要么使用 alter语句修改编码。

```
mysql> create database 0728db charset Latin1; # 创建数据库指定字符集
Query OK, 1 row affected (0.00 sec)
```

```
mysql> use 0728db;
Database changed
mysql> create table student (
    id int,
    name varchar(20)
) charset Latin1; # 指定表的字符集
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> alter table student charset utf8; #修改表字符编码为UTF8

# alter table <表名称> modify <字段名> <数据类型> charset <字符集>
mysql> alter table student modify name varchar(20) charset utf8; #修改字段字符编码为UTF8

mysql> alter database 0728db charset utf8; #修改数据库的字符编码为utf8
```