

admin

mysql

```
1  mysql --help
2  mysql -?
3  mysql --version
4  mysql -V
5  mysql -h[host] -uroot -p -P[port]
6
```

mysqld

```
1  mysqld --verbose --help
2  mysqld --version
3  mysqld -V
4
```

mysqladmin

```
1  mysqladmin --help
```

mysqldump

```
1  mysqldump -uroot -p database > database.sql
2  mysql -uroot -p database < database.sql
3
```

mysql

```
1  help
2  status;
3  use databases
4  show databases;
5  show tables;
6  desc table;
7  /*
8  describe table;
9  show columns from table;
10 */
11 show status;
```

```
12 show create database;
13 show create table;
14 show grants;
15 show errors;
16 show warnings;
17 help show;
18 show variables like pattern
```

```
1 3中注释方法
2 # select
3 -- select
4 /*
5 select
6 */
7
```

```
1 # 函数
2 1 select version();
3 2 select database();
4
5
```

information_schema

```
1 # 事务表
2 innodb_trx
3 # 进程表
4 processlist
5
```

Chapter 3 Tutorial

3.5在批处理模式下使用mysql

在前面的部分中，您以交互方式使用[mysql](#)输入语句并查看结果。您也可以以批处理模式运行[mysql](#)。为此，将要运行的语句放在文件中，然后告诉 [mysql](#)从文件中读取其输入：

```
1 shell> mysql < batch-file
```

如果您在Windows下运行[mysql](#)并在文件中包含一些导致问题的特殊字符，则可以执行以下操作：

```
1 C:\> mysql -e "source batch-file"
```

如果需要在命令行上指定连接参数，则命令可能如下所示：

```
1 shell> mysql -h host -u user -p < batch-file
2 Enter password: *****
```

当您以这种方式使用[mysql](#)时，您将创建一个脚本文件，然后执行该脚本。

如果您希望脚本继续运行，即使其中的某些语句产生错误，也应使用 [--force](#) 命令行选项。

为什么要使用脚本？原因如下：

- 如果您反复（例如每天或每周）运行查询，将其设置为脚本可以避免每次执行时都重新键入查询。
- 您可以通过复制和编辑脚本文件从相似的现有查询中生成新查询。
- 在开发查询时，批处理模式也很有用，特别是对于多行语句或多语句序列。如果输入有误，则无需重新输入所有内容。只需编辑脚本以更正错误，然后告诉[mysql](#)重新执行即可。
- 如果您的查询产生大量输出，则可以通过寻呼机运行输出，而不用看着它滚动到屏幕顶部之外：

```
1 shell> mysql < batch-file | more
```

- 您可以将输出捕获到文件中以进行进一步处理：

```
1 shell> mysql < batch-file > mysql.out
```

- 您可以将脚本分发给其他人，以便他们也可以运行这些语句。
- 在某些情况下，例如，当您从[cron](#)作业运行查询时，不允许进行交互使用。在这种情况下，您必须使用批处理模式。

在批处理模式下运行[mysql](#)时，与以交互方式使用时，默认输出格式不同（更简洁）。例如，以交互方式运行[mysql](#) `SELECT DISTINCT species FROM pet` 时，输出如下所示：

```
1 +-----+
2 | species |
3 +-----+
4 | bird   |
5 | cat    |
6 | dog    |
7 | hamster|
8 | snake  |
9 +-----+
```

在批处理模式下，输出看起来像这样：

```
1 species
2 bird
3 cat
4 dog
5 hamster
6 snake
```

如果要以批处理方式获取交互式输出格式，请使用[mysql-t](#)。要将执行的语句回显到输出，请使用[mysql-v](#)。

您还可以通过使用以下命令，从[mysql](#)提示符中使用脚本：`source ``\.`

```
1 mysql> source filename;
2 mysql> \. filename
```

有关更多信息，请参见[第4.5.1.5节“从文本文件执行SQL语句”](#)。

经验

```
1 select 'CA20181110683' % 4;
2 select null % 4;
3
4 select substr('FA2019071001-2022-02', 4, 9);
5
6 select * from cooperation_projects_clean
7 where
8 project_code = case substr(project_code, 4, 9) % 4
9 when 0 then project_code
10 end;
11
12 select
13 project_no, @n:=@n+1
14 -- *
15 from sync_project_info, (select @n:=0) a;
16
17
18 SHOW VARIABLES LIKE 'event_scheduler';
19
20 SET GLOBAL event_scheduler = ON;
```