

NSD DBA1 DAY01

1. [构建MySQL服务器](#)
2. [数据库基本管理](#)
3. [MySQL 数据类型](#)

1 构建MySQL服务器

1.1 问题

本案例要求熟悉MySQL官方安装包的使用，快速构建一台数据库服务器：

- 安装MySQL-server、MySQL-client软件包
- 修改数据库用户root的密码
- 确认MySQL服务程序运行、root可控

1.2 方案

本课程将使用64位的RHEL 7操作系统，MySQL数据库的版本是5.7.17。

访问<http://dev.mysql.com/downloads/mysql/>，找到MySQL Community Server下载页面，平台选择“Red Hat Enterprise Linux 7/ Oracle Linux”，然后选择64位的bundle整合包下载，如图-1所示。



图-1

注意：下载MySQL软件时需要以Oracle网站账户登录，如果没有请根据页面提示先注册一个（免费）

1.3 步骤

实现此案例需要按照如下步骤进行。

[Top](#)

步骤一：准备工作

1) 停止mariadb服务

```
01. [root@localhost ~]# systemctl stop mariadb
```

2) 删除/etc/my.cnf配置文件

此配置文件由RHEL自带的mariadb-libs库提供：

```
[root@localhost ~]# rm -rf /etc/my.cnf
```

3) 删除数据

```
01. [root@localhost ~]# rm -rf /var/lib/mysql/*
```

4) 卸载软件包 (没有会显示未安装软件包)

```
01. [root@localhost ~]# rpm -e --nodeps mariadb-server mariadb
```

```
02. 警告：/var/log/mariadb/mariadb.log 已另存为/var/log/mariadb/mariadb.log.rpm.save
```

步骤二：安装软件包

1) 安装mysql时可能会缺少某些依赖包，需提前单独安装

```
01. [root@localhost ~]# yum -y install perl-Data-Dumper perl-JSON perl-Time-HiR
```

2) 物理机传输解压包给虚拟机192.168.4.1

```
01. [root@room9pc01 ~]# cd 桌面
```

```
02. [root@room9pc01 桌面]# scp mysql-5.7.17.tar 192.168.4.1:/root/ //给虚拟机传包
```

```
03. root@192.168.4.1's password:
```

```
04. mysql-5.7.17.tar 100% 543MB 95.6MB/s 00:05
```

3) 虚拟机192.168.4.1解压mysql-5.7.17.tar 整合包

```
01. [root@localhost ~]# tar -xvf mysql-5.7.17.tar
```

//解压mysql整合包

```
02. ./mysql-community-client-5.7.17-1.el7.x86_64.rpm
```

```
03. ./mysql-community-common-5.7.17-1.el7.x86_64.rpm
```

```
04. ./mysql-community-devel-5.7.17-1.el7.x86_64.rpm
```

```
05. ./mysql-community-embedded-5.7.17-1.el7.x86_64.rpm
```

[Top](#)

06. ./mysql-community-embedded-compat-5.7.17-1.el7.x86_64.rpm
07. ./mysql-community-embedded-devel-5.7.17-1.el7.x86_64.rpm
08. ./mysql-community-libs-5.7.17-1.el7.x86_64.rpm
09. ./mysql-community-libs-compat-5.7.17-1.el7.x86_64.rpm
10. ./mysql-community-minimal-debuginfo-5.7.17-1.el7.x86_64.rpm
11. ./mysql-community-server-5.7.17-1.el7.x86_64.rpm
12. ./mysql-community-test-5.7.17-1.el7.x86_64.rpm

步骤三：启动MySQL数据库服务并设置开机自启

01. [root@localhost ~] # systemctl start mysqld //启动mysql服务
02. [root@localhost ~] # systemctl enable mysqld //设置开机自启
03. [root@localhost ~] # systemctl status mysqld //查看mysql服务状态
04. ● mysqld.service - MySQL Server
05. Loaded: loaded (/usr/lib/systemd/system/mysqld.service; enabled; vendor preset: disabled)
06. Active: active (running) since 二 2018-08-28 10:03:24 CST; 8min ago
07. Docs: man:mysqld(8)
08. http://dev.mysql.com/doc/refman/en/using-systemd.html
09. Main PID: 4284 (mysqld)
10. CGroup: /system.slice/mysqld.service
11. └─4284 /usr/sbin/mysqld --daemonize --pid-file=/var/r...
- 12.
13. 8月 28 10:02:56 localhost.localdomain systemd[1]: Starting My SQ...
14. 8月 28 10:03:24 localhost.localdomain systemd[1]: Started My SQL...
15. Hint: Some lines were ellipsized, use -l to show in full.

步骤四：连接MySQL服务器，修改密码

查看随机生成的root管理密码

01. [root@localhost ~] #grep 'temporary password' /var/log/mysqld.log
02. 2017-04-01T18:10:42.948679Z 1 [Note] A temporary password is generated for root@loc

2) 使用客户端命令mysql连接到MySQL服务器

提示验证时，填入前一步获得的随机密码，验证成功后即可进入“mysql>”环境：

[Top](#)

01. [root@localhost ~] # mysql -u root -p'mtoa>Av<p6Yk'

```

02.  mysql: [ Warning] Using a password on the command line interface can be insecure.
03.  Welcome to the MySQL monitor.  Commands end with ; or \g.
04.  Your MySQL connection id is 11
05.  Server version: 5.7.17
06.
07.  Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.
08.
09.  Oracle is a registered trademark of Oracle Corporation and/or its
10.  affiliates. Other names may be trademarks of their respective
11.  owners.
12.
13.  Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
14.  mysql>                                     //登录成功后，进入SQL操作环境

```

用该密码登录到服务端后，必须马上修改密码，否则会报如下错误：

```

01.  mysql> show databases;
02.  ERROR 1820 (HY000): You must reset your password using ALTER USER statement before

```

3) 执行SET PASSWORD命令修改密码

这个其实与validate_password_policy的值有关，默认为1，所以刚开始设置的密码必须符合长度，且必须含有数字，小写或大写字母，特殊字符。如果我们不希望密码设置的那么复杂，需要修改两个全局参数：validate_password_policy与validate_password_length。validate_password_length默认值为8，最小值为4，如果你显性指定validate_password_length的值小于4，尽管不会报错，但validate_password_length的值将设为4。

可参考下列指令：

```

01.  mysql>set global validate_password_policy=0;    //只验证长度
02.  Query OK, 0 rows affected (0.00 sec)
03.  mysql>set global validate_password_length=6;    //修改密码长度,默认值是8个字符
04.  Query OK, 0 rows affected (0.00 sec)
05.  mysql> alter user user() identified by "123456"; //修改登陆密码
06.  Query OK, 0 rows affected (0.00 sec)

```

上述操作的结果是——更改数据库用户root从本机访问时的密码，设为123456。

退出“mysql>”环境，重新登录验证，必须采用新的密码才能登入：

[Top](#)

```

01.  my sql> exit                //退出 my sql> 环境
02.  Bye
03.  [ root@localhost ~] # my sql - u root - p      //重新登录
04.  Enter password:                //输入新设置的密码
05.  Welcome to the My SQL monitor.  Commands end with ; or \g.
06.  Your My SQL connection id is 15
07.  Server version: 5.7.17 My SQL Community Server ( GPL)
08.
09.  Copyright ( c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.
10.
11.  Oracle is a registered trademark of Oracle Corporation and/or its
12.  affiliates. Other names may be trademarks of their respective
13.  owners.
14.
15.  Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

```

2 数据库基本管理

2.1 问题

本案例要求熟悉MySQL的连接及数据库表的增删改查等基本管理操作，主要完成以下几个方便的操作：

- 使用mysql命令连接数据库
- 练习查看/删除/创建库的相关操作
- 练习查看/删除/创建表的相关操作，表数据参考如表-1所示内容

表 - 1 测试用表数据

学号	姓名	性别	手机号	通信地址
NSD131201	张三	男	13012345678	朝阳区劲松南路..
NSD131202	韩梅梅	女	13722223333	海淀区北三环西路..
NSD131203	王五	男	18023445678	丰台区兴隆中街..

2.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：使用mysql命令连接数据库

连接MySQL服务器时，最基本的用法是通过 -u 选项指定用户名、-p指定密码。密码可以写在命令行（如果不写，则出现交互，要求用户输入），当然基于安全考虑一般不推荐这么做：

[Top](#)

```

01.  [ root@dbvr1 ~] # my sql - u root - p123456      //紧挨着选项，不要空格
02.  my sql: [ Warning] Using a password on the command line interface can be insecure.

```

```

03. Welcome to the MySQL monitor. Commands end with ; or \g.
04. Your MySQL connection id is 16
05. Server version: 5.7.17 MySQL Community Server (GPL)
06.
07. Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.
08.
09. Oracle is a registered trademark of Oracle Corporation and/or its
10. affiliates. Other names may be trademarks of their respective
11. owners.
12.
13. Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
14.
15. mysql> exit //退出已登录的mysql> 环境
16. Bye

```

默认情况下，mysql命令会连接本机的MySQL服务。但在需要的时候，可以通过 -h 选项指定远程主机；

```

01. [root@dbsvr1 ~]# mysql -h 127.0.0.1 -u root -p
02. Enter password:
03. Welcome to the MySQL monitor. Commands end with ; or \g.
04. Your MySQL connection id is 17
05. Server version: 5.7.17 MySQL Community Server (GPL)
06.
07. Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.
08.
09. Oracle is a registered trademark of Oracle Corporation and/or its
10. affiliates. Other names may be trademarks of their respective
11. owners.
12.
13. Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
14.
15. mysql> exit //退出已登录的mysql环境
16. Bye

```

步骤二：练习查看/删除/创建库的相关操作

以root用户登入“mysql>”环境后，可以执行各种MySQL指令、SQL指令。基本的用法事项如下：
[Top](#)

- 操作指令不区分大小写（库名/表名、密码、变量值等除外）。

- 每条SQL指令以 ; 结束或分隔。
- 不支持 Tab 键自动补齐。
- \c 可废弃当前编写错的操作指令。

1) 查看现有的库

```

01.  my sql> show databases;           //查看现有的库
02.  +-----+
03.  | Database |
04.  +-----+
05.  | information_schema |           //信息概要库
06.  | my sql   |           //授权库
07.  | performance_schema |         //性能结构库
08.  | sys      |           //系统元数据库
09.  +-----+
10.  4 rows in set ( 0.15 sec)

```

2) 切换/使用指定的库

```

01.  my sql> use sys;                 //切换到sys库
02.  Database changed
03.  my sql> select database();       //确认当前所在的库
04.  +-----+
05.  | DATABASE() |
06.  +-----+
07.  | sys      |
08.  +-----+
09.  1 row in set ( 0.00 sec)

```

切换到mysql库：

```

01.  my sql> use my sql;             //切换到my sql库
02.  Reading table information for completion of table and column names
03.  You can turn off this feature to get a quicker startup with - A
04.
05.  Database changed
06.  my sql> select database();       //确认当前所在的库
07.  +-----+
08.  | DATABASE() |

```

[Top](#)

```

09.  +-----+
10.  | my sql |
11.  +-----+
12.  1 row in set ( 0.00 sec)
13.  5 rows in set ( 0.00 sec)

```

3) 新建名为newdb的库，确认结果：

```

01.  my sql> create database newdb;           //新建名为newdb的库
02.  Query OK, 1 row affected ( 0.00 sec)
03.
04.  my sql> show databases;
05.  +-----+
06.  | Database |
07.  +-----+
08.  | information_schema |
09.  | my db      |           //新建的my db库
10.  | my sql     |
11.  | newdb      |           //新建的newdb库
12.  | performance_schema |
13.  | sy s      |
14.  +-----+
15.  6 rows in set ( 0.00 sec)

```

4) 删除指定的库

```

01.  my sql> drop database newdb;           //删除名为newdb的库
02.  Query OK, 0 rows affected ( 0.01 sec)
03.
04.  my sql> show databases;           //确认删除结果，已无newdb库
05.  +-----+
06.  | Database |
07.  +-----+
08.  | information_schema |
09.  | my db      |
10.  | my sql     |
11.  | performance_schema |
12.  | sy s      |
13.  +-----+

```

[Top](#)

14. 5 rows in set (0.00 sec)

步骤三：练习查看/删除/创建表的相关操作

1) 查看指定的库里有哪些表

查看mysql库里有哪些表：

```

01.  my sql> use my sql;
02.  Reading table information for completion of table and column names
03.  You can turn off this feature to get a quicker startup with -A
04.
05.  Database changed
06.  my sql> show tables;
07.  +-----+
08.  | Tables_in_my sql |
09.  +-----+
10.  | columns_priv      |
11.  | db                 |
12.  | engine_cost       |
13.  | event              |
14.  | func               |
15.  | general_log        |
16.  | gtid_executed      |
17.  | help_category      |
18.  | help_key word      |
19.  | help_relation      |
20.  | help_topic         |
21.  | innodb_index_stats  |
22.  | innodb_table_stats  |
23.  | ndb_binlog_index   |
24.  | plugin             |
25.  | proc               |
26.  | procs_priv         |
27.  | proxies_priv       |
28.  | server_cost        |
29.  | servers            |
30.  | slave_master_info   |
31.  | slave_relay_log_info |
32.  | slave_worker_info   |
33.  | slow_log           |
34.  | tables_priv        |

```

[Top](#)

```

35. | time_zone |
36. | time_zone_leap_second |
37. | time_zone_name |
38. | time_zone_transition |
39. | time_zone_transition_type |
40. | user | //存放数据库用户的表
41. +-----+
42. 31 rows in set (0.00 sec)

```

2) 查看指定表的字段结构

当前库为mysql，查看columns_priv表的结构，以列表形式展现：

```

01. mysql> desc columns_priv \G //查看表结构，以列表形式展现，末尾不用分号
02. ***** 1. row *****
03. Field: Host
04. Type: char(60)
05. Null: NO
06. Key: PRI
07. Default:
08. Extra:
09. ***** 2. row *****
10. Field: Db
11. Type: char(64)
12. Null: NO
13. Key: PRI
14. Default:
15. Extra:
16. ***** 3. row *****
17. Field: User
18. Type: char(32)
19. Null: NO
20. Key: PRI
21. Default:
22. Extra:
23. ***** 4. row *****
24. Field: Table_name
25. Type: char(64)
26. Null: NO
27. Key: PRI
28. Default:

```

[Top](#)

```

29.      Extra:
30.      ***** 5. row *****
31.      Field: Column_name
32.      Type: char( 64)
33.      Null: NO
34.      Key: PRI
35.      Default:
36.      Extra:
37.      ***** 6. row *****
38.      Field: Timestamp
39.      Type: timestamp
40.      Null: NO
41.      Key:
42.      Default: CURRENT_TIMESTAMP
43.      Extra: on update CURRENT_TIMESTAMP
44.      ***** 7. row *****
45.      Field: Column_priv
46.      Type: set( 'Select', 'Insert', 'Update', 'References')
47.      Null: NO
48.      Key:
49.      Default:
50.      Extra:
51.      7 rows in set ( 0.01 sec)

```

查看columns_priv表的结构，以表格形式展现：

```

01.  my sql> desc columns_priv;      //查看表结构，以表格形式展现末尾需要有分号
02.  +-----+-----+-----+-----+-----+
03.  | Field | Type | Null | Key | Default | Extra |
04.  +-----+-----+-----+-----+-----+
05.  | Host | char( 60) | NO | PRI | | |
06.  | Db | char( 64) | NO | PRI | | |
07.  | User | char( 32) | NO | PRI | | |
08.  | Table_name | char( 64) | NO | PRI | | |
09.  | Column_name | char( 64) | NO | PRI | | |
10.  | Timestamp | timestamp | NO | | CURRENT_TIMESTAMP | or
11.  | Column_priv | set( 'Select', 'Insert', 'Update', 'References') | NO | | |
12.  +-----+-----+-----+-----+-----+
13.  7 rows in set ( 0.00 sec)

```

[Top](#)

上述操作中，当引用非当前库中的表时，可以用“库名.表名”的形式。比如，切换为mysql库再执行“desc columns_priv;”，与以下操作的效果是相同的：

```

01.  my sql> desc my sql.columns_priv ;
02.  +-----+-----+-----+-----+
03.  | Field      | Type                               | Null | Key | Default      | Extra
04.  +-----+-----+-----+-----+
05.  | Host       | char( 60)                          | NO   | PRI |              |
06.  | Db         | char( 64)                          | NO   | PRI |              |
07.  | User       | char( 16)                          | NO   | PRI |              |
08.  | Table_name | char( 64)                          | NO   | PRI |              |
09.  | Column_name | char( 64)                          | NO   | PRI |              |
10.  | Timestamp  | timestamp                          | NO   |    | CURRENT_TIMESTAMP | or
11.  | Column_priv | set( 'Select','Insert','Update','References') | NO   |    |              |
12.  +-----+-----+-----+-----+
13.  7 rows in set ( 0.00 sec)

```

3) 在test库中创建一个名为pwlist的表

包括name、password两列，其中name列作为主键。两个字段值均不允许为空，其中密码列赋予默认空值，相关操作如下所述。

切换到mydb库：

```

01.  my sql> use my db;
02.  Database changed

```

新建pwlist表：

```

01.  my sql> create table pwlist(
02.      -> name char( 16) not null,
03.      -> password char( 48) default '',
04.      -> primary key( name)
05.      -> );
06.  Query OK, 0 rows affected ( 0.38 sec)

```

确认新创建的表：

[Top](#)

```

01.  my sql> show tables;
02.  +-----+
03.  | Tables_in_my_db |
04.  +-----+
05.  | pwlist          |           //新建的pwlist表
06.  +-----+
07.  1 rows in set (0.01 sec)

```

查看pwlist表的字段结构：

```

01.  my sql> desc pwlist;
02.  +-----+-----+-----+-----+-----+-----+
03.  | Field | Type | Null | Key | Default | Extra |
04.  +-----+-----+-----+-----+-----+-----+
05.  | name  | char(16) | NO | PRI | NULL | |
06.  | password | char(48) | YES | | | |
07.  +-----+-----+-----+-----+-----+-----+
08.  2 rows in set (0.01 sec)

```

4) 删除指定的表

删除当前库中的pwlist表：

```

01.  my sql> drop table pwlist;
02.  Query OK, 0 rows affected (0.01 sec)

```

确认删除结果：

```

01.  my sql> show tables;
02.  Empty set (0.00 sec)

```

5) 在mydb库中创建一个学员表

表格结构及数据内容如表-1所示。

在MySQL表内存储中文数据时，需要更改字符集（默认为latin1不支持中文），以便MySQL支持存储中文数据记录；比如，可以在创建库或表的时候，手动添加“DEFAULT [Top](#) CHARSET=utf8”来更改字符集。

根据上述表格结构，创建支持中文的student表：

```

01. my sql> CREATE TABLE my db.student(
02.     -> 学号 char( 9) NOT NULL,
03.     -> 姓名 varchar( 4) NOT NULL,
04.     -> 性别 enum( '男','女') NOT NULL,
05.     -> 手机号 char( 11) DEFAULT '',
06.     -> 通信地址 varchar( 64),
07.     -> PRIMARY KEY( 学号)
08.     -> ) DEFAULT CHARSET=utf8;           //手工指定字符集,采用utf8
09. Query OK, 0 rows affected ( 0.31sec)

```

查看student表的字段结构：

```

01. my sql> DESC my db.student;
02. +-----+-----+-----+-----+-----+
03. | Field      | Type          | Null | Key | Default | Extra |
04. +-----+-----+-----+-----+-----+
05. | 学号       | char( 9)      | NO   | PRI | NULL    |      |
06. | 姓名       | varchar( 4)   | NO   |     | NULL    |      |
07. | 性别       | enum( '男','女') | NO   |     | NULL    |      |
08. | 手机号     | char( 11)     | YES  |     |         |      |
09. | 通信地址   | varchar( 64)  | YES  |     | NULL    |      |
10. +-----+-----+-----+-----+-----+
11. 5 rows in set ( 0.00 sec)

```

查看student表的实际创建指令：

```

01. my sql> SHOW CREATE TABLE my db.student;
02. +-----+-----+-----+-----+-----+
03. | Table | Create Table
04. +-----+-----+-----+-----+-----+
05. | student | CREATE TABLE `student` (
06.   `学号` char( 9) NOT NULL,
07.   `姓名` varchar( 4) NOT NULL,
08.   `性别` enum( '男','女') NOT NULL,
09.   `手机号` char( 11) DEFAULT '',
10.   `通信地址` varchar( 64) DEFAULT NULL,
11.   PRIMARY KEY ( `学号` )
12. ) ENGINE=InnoDB DEFAULT CHARSET=utf8

```

[Top](#)

13. +-----+
14. 1 row in set (0.00 sec)



注意：若要修改MySQL服务的默认字符集，可以更改服务器的my.cnf配置文件，添加 character_set_server=utf8 配置，然后重启数据库服务。

```

01. [ root@dbsvr1 ~] # vim /etc/my.cnf                                //修改运行服务配置
02. [ my sql d]
03. ... ..
04. character_set_server=utf8
05.
06. [ root@dbsvr1 ~] # systemctl restart my sql d                    //重启服务
07. ... ..
08. [ root@dbsvr1 ~] # my sql - u root - p
09. Enter password:
10. ... ..
11. my sql> SHOW VARIABLES LIKE 'character%';                        //确认更改结果
12. +-----+
13. | Variable_name      | Value                |
14. +-----+
15. | character_set_client | utf8                 |
16. | character_set_connection | utf8                 |
17. | character_set_database | utf8                 |
18. | character_set_filesystem | binary               |
19. | character_set_results  | utf8                 |
20. | character_set_server   | utf8                 |
21. | character_set_system   | utf8                 |
22. | character_sets_dir     | /usr/share/mysql/charsets/ |
23. +-----+
24. 8 rows in set (0.03 sec)

```

3 MySQL 数据类型

3.1 问题

- 本案例要求熟悉MySQL的字段数据类型、表结构，各种时间函数的使用的使用，完成以下任务操作：
- 在stadb库里创建stuinfo表，表结构、字段类型自定义
- 表结构自定义，并合理使用数据类型
- 练习各种时间函数的使用
-

[Top](#)

3.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：创建studb库、stuinfo表

1) 新建studb库，并切换到studb库

```
01. my sql> CREATE DATABASE studb;
02. Query OK, 1 row affected ( 0.00 sec)
03. my sql> USE studb;
04. Database changed
```

2) 新建stuinfo表

假定stuinfo表用来记录每个家庭成员的姓名 (name)、性别 (gender)、出生日期 (birth)、职业 (job)、与户主关系 (relation)。

```
01. my sql> CREATE TABLE stuinfo (
02.     -> name varchar( 16) NOT NULL,
03.     -> gender enum( 'male','femal') DEFAULT 'male',
04.     -> birth date NOT NULL,
05.     -> job varchar( 16) DEFAULT '',
06.     -> relation varchar( 24) NOT NULL,
07.     -> PRIMARY KEY( name)
08.     -> );
09. Query OK, 0 rows affected ( 0.61sec)
```

查看stuinfo表的字段结构：

```
01. my sql> DESC stuinfo;
02. +-----+-----+-----+-----+-----+-----+
03. | Field | Type          | Null | Key | Default | Extra |
04. +-----+-----+-----+-----+-----+-----+
05. | name  | varchar( 16)  | NO   | PRI | NULL    |       |
06. | gender | enum( 'male','femal') | YES |     | male    |       |
07. | birth | date          | NO   |     | NULL    |       |
08. | job   | varchar( 16)  | YES  |     |         |       |
09. | relation | varchar( 24) | NO   |     | NULL    |       |
10. +-----+-----+-----+-----+-----+-----+
11. 5 rows in set ( 0.00 sec)
```

[Top](#)

步骤二：练习各种时间函数的使用

1) 使用now()查看当前的日期和时间

```

01.  my sql> SELECT now();
02.  +-----+
03.  | now()      |
04.  +-----+
05.  | 2017- 04- 02 04: 02: 42 |
06.  +-----+
07.  1 row in set ( 0.00 sec)

```

2) 使用sysdate()查看系统日期和时间

```

01.  my sql> SELECT sysdate();
02.  +-----+
03.  | sysdate()   |
04.  +-----+
05.  | 2017- 04- 02 04: 03: 21 |
06.  +-----+
07.  1 row in set ( 0.00 sec)

```

3) 使用curdate()获得当前的日期，不含时间

```

01.  my sql> SELECT curdate();
02.  +-----+
03.  | curdate()   |
04.  +-----+
05.  | 2017- 04- 02 |
06.  +-----+
07.  1 row in set ( 0.00 sec)

```

4) 使用curtime()获得当前的时间，不含日期

```

01.  my sql> SELECT curtime();
02.  +-----+
03.  | curtime()   |

```

[Top](#)

```

04.  +-----+
05.  | 04:04:55 |
06.  +-----+
07.  1 row in set ( 0.00 sec)

```

5) 分别获取当前日期时间中的年份、月份、日

```

01.  my sql> SELECT year( now() ),month( now() ),day( now() );
02.  +-----+-----+-----+
03.  | year( now() ) | month( now() ) | day( now() ) |
04.  +-----+-----+-----+
05.  |      2017    |      4    |      2    |
06.  +-----+-----+-----+
07.  1 row in set ( 0.00 sec)

```

6) 获取系统日期时间中的月份、日

```

01.  my sql> SELECT month( sy sdate() ),day( sy sdate() );
02.  +-----+-----+
03.  | month( sy sdate() ) | day( sy sdate() ) |
04.  +-----+-----+
05.  |      4    |      2    |
06.  +-----+-----+
07.  1 row in set ( 0.00 sec)

```

7) 获取系统日期时间中的时刻

```

01.  my sql> SELECT time( sy sdate() );
02.  +-----+
03.  | time( sy sdate() ) |
04.  +-----+
05.  | 04:06:08    |
06.  +-----+
07.  1 row in set ( 0.00 sec)

```

[Top](#)