

数据库操作

NSD python

基本配置



Mysql简介

- MySQL AB创建于1995年，从事制造及拥有关联式数据库管理系统MySQL以及相关的产品
- 2008年1月16日，Sun(太阳微系统)正式收购MySQL
- 2013年6月18日，甲骨文公司修改MySQL授权协议，移除GPL
- MariaDB数据库管理系统是MySQL的一个分支，主要由开源社区在维护，采用GPL授权许可



安装及启动

- RHEL7系统上默认的数据库是MariaDB

```
[root@localhost ~]# yum install -y mariadb-server
```

```
[root@localhost ~]# systemctl start mariadb
```

```
[root@localhost ~]# systemctl enable mariadb
```

```
[root@localhost packages]# mysqladmin password tedu.cn
```



查看数据库结构

- 列出当前MySQL服务器上有哪些库
 - SHOW DATABASES;

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| test |
+-----+
4 rows in set (0.00 sec)
```

——其中 mysql 库为授权库，test为测试库



查看数据库结构（续1）

- 使用/切换到指定的库
 - USE 数据库名;

```
mysql> USE mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> 
```



查看数据库结构（续2）

- 列出当前的库内有哪些表
 - SHOW TABLES;

```
mysql> SHOW TABLES;
```

```
+-----+
| Tables_in_mysql |
+-----+
| columns_priv    |
| db              |
| event           |
| func            |
| general_log     |
| help_category   |
| help_keyword    |
| help_relation   |
| help_topic      |
| innodb_index_stats |
| innodb_table_stats |
| ndb_binlog_index |
```



查看数据库结构（续3）

- 查看指定表的字段结构
 - DESCRIBE 表名;

```
mysql> DESCRIBE columns_priv\G
***** 1. row *****
Field: Host
Type: char(60)
Null: NO
Key: PRI
Default:
```

— \G 以列表形式展现，否则为表格形式

```
mysql> DESCRIBE columns_priv;
```

| Field | Type | Null |
|-------|----------|-------|
| Key | Default | Extra |
| Host | char(60) | NO |
| PRI | | |
| Db | char(64) | NO |



创建/删除库

- 数据库的命名规则
 - 可以使用数字/字母/下划线，但不能纯数字
 - 区分大小写，具有唯一性
 - 不可使用指令关键字、特殊字符



创建/删除库（续1）

- 新建指定名称的库
 - CREATE DATABASE 数据库名;

```
mysql> CREATE DATABASE newdb;  
Query OK, 1 row affected (0.07 sec)
```

```
mysql> SHOW DATABASES;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| newdb |  
| performance_schema |  
| test |  
+-----+  
5 rows in set (0.00 sec)
```



创建/删除库（续2）

- 查看当前所在的库
 - SELECT DATABASE();

```
mysql> USE newdb;
Database changed
mysql> SELECT DATABASE();
+-----+
| DATABASE() |
+-----+
| newdb      |
+-----+
1 row in set (0.00 sec)
```



创建/删除库（续3）

- 删除指定的库
 - DROP DATABASE 数据库名;

```
mysql> DROP DATABASE newdb;  
Query OK, 0 rows affected (0.11 sec)
```

```
mysql> SHOW DATABASES;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| performance_schema |  
| test |  
+-----+  
4 rows in set (0.00 sec)
```



创建/删除表

- 新建指定名称的表

- CREATE TABLE 表名(

- 字段名1 字段类型(宽度) 约束条件,

- 字段名2 字段类型(宽度) 约束条件,

-

- 字段名N 字段类型(宽度) 约束条件,

- PRIMARY KEY (主键名)

-);



创建/删除表（续1）

- 示例：在test库中新建pwlist表
 - 包括两列，其中name列为主键

```
mysql> USE test;
Database changed
mysql> CREATE TABLE pwlist (name CHAR(16) NOT NULL, password CHAR(48) DEFAULT '', PRIMARY KEY (name));
Query OK, 0 rows affected (0.06 sec)

mysql> DESCRIBE pwlist;
```

| Field | Type | Null | Key | Default | Extra |
|----------|----------|------|-----|---------|-------|
| name | char(16) | NO | PRI | NULL | |
| password | char(48) | YES | | | |

```
2 rows in set (0.00 sec)
```



创建/删除表（续2）

- 删除指定名称的表
 - DROP TABLE 表名;

```
mysql> DROP TABLE pwlist;
Query OK, 0 rows affected (0.02 sec)

mysql> DESCRIBE pwlist;
ERROR 1146 (42S02): Table 'test.pwlist' doesn't exist
mysql> 
```



MySQL数据类型



数值类型

| 类 型 | 大 小 | 范围（有符号） | 范围（无符号） | 用 途 |
|-----------|---|-------------------------|-------------------|--------|
| TINYINT | 1字节 | -128~127 | 0 ~ 255 | 微小整数 |
| SMALLINT | 2字节 | -32768~32767 | 0 ~ 65535 | 小整数 |
| MEDIUMINT | 3字节 | $-2^{23} \sim 2^{23}-1$ | $0 \sim 2^{24}-1$ | 中整数 |
| INT | 4字节 | $-2^{31} \sim 2^{31}-1$ | $0 \sim 2^{32}-1$ | 大整数 |
| BIGINT | 8字节 | $-2^{63} \sim 2^{63}-1$ | $0 \sim 2^{64}-1$ | 极大整数 |
| FLOAT | 4字节 | | | 单精度浮点数 |
| DOUBLE | 8字节 | | | 双精度浮点数 |
| DECIMAL | 对DECIMAL(M,D)，其中M为有效位数、D为小数位数，M应大于D，占用M+2字节 | | | |



整数型

- 关于整数型字段
 - 使用UNSIGNED修饰时，对应的字段只保存正数
 - 数值不够指定宽度时，在左边填充空格补位
 - 宽度仅是显示宽度，存数值的大小由类型决定
 - 使用关键字ZEROFILL 时，填0代替空格补位
 - 数值超出范围时，报错。



整数型（续1）

- 了解表的字段设置
 - DESC 或 DESCRIBE，查看表结构

```
mysql> USE test;
Database changed
mysql> CREATE TABLE t1(id tinyint unsigned zerofill);
Query OK, 0 rows affected (0.06 sec)
```

```
mysql> DESC t1;
```

| Field | Type | Null | Key | Default | Extra |
|-------|------------------------------|------|-----|---------|-------|
| id | tinyint(3) unsigned zerofill | YES | | NULL | |

```
1 row in set (0.00 sec)
```

字段名 | 字段类型 | 是否为空 | 是否为主键 | 默认值 | 描述信息



浮点型

- 关于浮点型字段
 - 定义格式：float(总宽度, 小数位数)
 - 当字段值与类型不匹配时，字段值作为0处理
 - 数值超出范围时，仅保存最大/最小值

```
mysql> CREATE TABLE t2(pay float(7,2));
Query OK, 0 rows affected (0.06 sec)
```

```
mysql> DESC t2;
```

| Field | Type | Null | Key | Default | Extra |
|-------|------------|------|-----|---------|-------|
| pay | float(7,2) | YES | | NULL | |

```
1 row in set (0.04 sec)
```



字符类型

字符类型

- 定长：char(字符数)
 - 最大长度255字符
 - 不够指定字符数时在右边用空格补齐
 - 字符数断超出时，无法写入数据。
- 变长：varchar(字符数)
 - 按数据实际大小分配存储空间
 - 字符数断超出时，无法写入数据。
- 大文本类型：text/blob
 - 字符数大于65535存储时使用



日期时间类型

日期时间类型

- 日期时间，DATETIME
 - 占用 8个字节
 - 范围：1000-01-01 00:00:00.000000
~ 9999-12-31 23:59:59.999999
- 日期时间，TIMESTAMP
 - 占用4个字节
 - 范围：1970-01-01 00:00:00.000000
~ 2038-01-19 03:14:07.999999



日期时间类型（续1）

- 日期，DATE
 - 占用4个字节
 - 范围：0001-01-01 ~ 9999-12-31
- 年份，YEAR
 - 占用1个字节
 - 范围：1901~2155
- 时间，TIME
 - 占用3个字节
 - 格式：HH:MM:SS



日期时间类型（续2）

- 关于日期时间字段
 - 当未给TIMESTAMP字段赋值时，自动以当前系统时间赋值，而DATETIME字段默认赋值为 NULL
- YEAR年份的处理
 - 默认用4位数字表示
 - 当只用2位数字赋值时，01~69视为2000~2069，而70~99视为1970~1999



日期时间类型（续3）

- 创建一个学员表
 - 包括姓名、入学年份、生日、培训时间段

```
mysql> CREATE TABLE stu_info(  
    -> name varchar(8),  
    -> starty year,  
    -> birth date,  
    -> ttime1 time,  
    -> ttime2 time  
    -> );  
Query OK, 0 rows affected (0.07 sec)
```



时间函数

| 类 型 | 用 途 |
|-----------|-----------------|
| now() | 获取调用此函数时的系统日期时间 |
| sysdate() | 执行时动态获得系统日期时间 |
| sleep(N) | 休眠N秒 |
| curdate() | 获取当前的系统日期 |
| curtime() | 获取当前的系统时刻 |
| month() | 获取指定时间中的月份 |
| date() | 获取指定时间中的日期 |
| time() | 获取指定时间中的时刻 |



时间函数（续1）

- 无需库、表，可直接调用
 - 使用SELECT指令输出函数结果

```
mysql> SELECT now(),sysdate(),curdate();
```

| now() | sysdate() | curdate() |
|---------------------|---------------------|------------|
| 2013-12-27 10:42:16 | 2013-12-27 10:42:16 | 2013-12-27 |

1 row in set (0.00 sec)

```
mysql> SELECT date(now()),time(now());
```

| date(now()) | time(now()) |
|-------------|-------------|
| 2013-12-27 | 10:42:58 |

1 row in set (0.00 sec)



枚举类型

枚举类型

- 从给定值集合中选择单个值，ENUM
 - 定义格式：enum(值1, 值2, 值N)
- 从给定值集合中选择一个或多个值，SET
 - 定义格式：set(值1, 值2, 值N)

```
mysql> CREATE TABLE tea(
    -> name varchar(4),
    -> gender enum("boy", "girl"),
    -> interest set("book", "film", "music", "football")
    -> );
Query OK, 0 rows affected (0.38 sec)
```



配置phpMyAdmin

安装

- 安装所需环境

```
# yum install -y httpd php php-mysql php-mbstring  
[root@localhost phpMyAdmin]# systemctl start httpd  
[root@localhost phpMyAdmin]# systemctl enable httpd
```

- 解压缩至/var/www/html目录下，并修改配置文件

```
[root@localhost phpMyAdmin]# cp config.sample.inc.php config.inc.php
```



运行phpMyAdmin

