

## sql 基础语法七-表操作

### 建表

创建MySQL数据表需要以下信息：

- 表名
- 表字段名
- 定义每个表字段

SQL

```
1 CREATE TABLE table_name (column_name column_type);
```

以前面创建的国家信息为例

SQL

```
1 CREATE TABLE country_info_test
2     (`country_id` int NOT NULL
3     , `country_name` varchar(45)
4     , PRIMARY KEY ( `country_id` )
5     )
6 ;
```

- 如果你不想字段为 **NULL** 可以设置字段的属性为 **NOT NULL**，在操作数据库时如果输入该字段的数据为**NULL**，就会报错。
- PRIMARY KEY关键字用于定义列为主键。您可以使用多列来定义主键，列间以逗号分隔。
- 注意MySQL命令终止符为分号；

### 插入数据

SQL

```
1 INSERT INTO table_name ( field1, field2,...fieldN )
2     VALUES( value1, value2,...valueN );
```

比如前面国家数据的插入代码

SQL

```
1 INSERT INTO data.country_info_test (country_id,country_name)
2     VALUES (1,'阿尔巴尼亚');
3 INSERT INTO data.country_info_test (country_id,country_name)
4     VALUES (2,'阿尔及利亚');
5 INSERT INTO data.country_info_test (country_id,country_name)
6     VALUES (3,'阿富汗');
```

如果数据是字符型，必须使用单引号或者双引号，如：'阿尔巴尼亚'。

## 删除字段

使用了 ALTER 命令及 DROP 子句来删除以上创建表的 i 字段：

SQL
1 ALTER TABLE testalter_tbl DROP i;

比如，删除国家信息表中的country\_id字段

SQL
1 ALTER TABLE country_info_test DROP country_id;

## 添加字段

使用了 ALTER 命令及 ADD 子句来向数据表中添加列，如下实例在表 country\_info\_test 中添加 country\_id字段，并定义数据类型：

SQL
1 ALTER TABLE country_info_test ADD country_id INT;

执行以上命令后，country\_id字段会自动添加到数据表字段的末尾

## 修改字段

果需要修改字段类型及名称，你可以在ALTER命令中使用 MODIFY 或 CHANGE 子句。

例如，把字段country\_id 的类型从INT改为 varchar(10)，可以执行以下命令：

SQL
1 ALTER TABLE country_info_test MODIFY country_id varchar(10);

使用 CHANGE 子句, 语法有很大的不同。在 CHANGE 关键字之后，紧跟着的是你要修改的字段名，然后指定新字段名及类型。比如：

SQL
1 ALTER TABLE country_info_test CHANGE country_id country_id2 INT;

# sql 基础语法八-数据类型

在第七章中，讲到表的操作，其中涉及到数据类型。比如coutry\_name就是字符串类型。下面枚举以下数据都有哪些类型。

## 数据类型

类型	大小	范围（有符号）	范围（无符号）	用途
TINYINT	1 Bytes	(-128, 127)	(0, 255)	小整数值
SMALLINT	2 Bytes	(-32 768, 32 767)	(0, 65 535)	大整数值
MEDIUMINT	3 Bytes	(-8 388 608, 8 388 607)	(0, 16 777 215)	大整数值
INT或 INTEGER	4 Bytes	(-2 147 483 648, 2 147 483 647)	(0, 4 294 967 295)	大整数值
BIGINT	8 Bytes	(-9,223,372,036,854,775,808, 9 223 372 036 854 775 807)	(0, 18 446 744 073 709 551 615)	极大整数值
FLOAT	4 Bytes	(-3.402 823 466 E+38, -1.175 494 351 E-38), 0, (1.175 494 351 E-38, 3.402 823 466 823 466 351 E+38)	0, (1.175 494 351 E-38, 3.402 823 466 E+38)	单精度 浮点数值
DOUBLE	8 Bytes	(-1.797 693 134 862 315 7 E+308, -2.225 073 858 507 201 4 E-308), 0, (2.225 073 858 507 201 4 E-308, 1.797 693 134 862 315 7 E+308)	0, (2.225 073 858 507 201 4 E-308, 1.797 693 134 862 315 7 E+308)	双精度 浮点数值
DECIMAL	对DECIMAL(M,D) ，如果M>D，为 M+2否则为D+2	依赖于M和D的值	依赖于M和D的值	小数值

## 日期和时间类型

表示时间值的日期和时间类型为DATETIME、DATE、TIMESTAMP、TIME和YEAR。

类型	大小 ( bytes)	范围	格式	用途
DATE	3	1000-01-01/9999-12-31	YYYY-MM-DD	日期值
TIME	3	'-838:59:59'/'838:59:59'	HH:MM:SS	时间值或持续时间
YEAR	1	1901/2155	YYYY	年份值
DATETIME	8	1000-01-01 00:00:00/9999-12-31 23:59:59	YYYY-MM-DD HH:MM:SS	混合日期和时间值
TIMESTAMP	4	1970-01-01 00:00:00/2038 结束时间是第 <b>2147483647</b> 秒，北京时间 <b>2038-1-19 11:14:07</b> ，格林尼治时间 2038年1月19日 凌晨 03:14:07	YYYYMMDD HHMMSS	混合日期和时间值，时间戳

## 字符串类型

字符串类型指CHAR、VARCHAR、BINARY、VARBINARY、BLOB、TEXT、ENUM和SET。该节描述了这些类型如何工作以及如何在查询中使用这些类型。

类型	大小	用途
CHAR	0-255 bytes	定长字符串
VARCHAR	0-65535 bytes	变长字符串
TINYBLOB	0-255 bytes	不超过 255 个字符的二进制字符串
TINYTEXT	0-255 bytes	短文本字符串
BLOB	0-65 535 bytes	二进制形式的长文本数据
TEXT	0-65 535 bytes	长文本数据
MEDIUMBLOB	0-16 777 215 bytes	二进制形式的中等长度文本数据
MEDIUMTEXT	0-16 777 215 bytes	中等长度文本数据
LONGBLOB	0-4 294 967 295 bytes	二进制形式的极大文本数据
LONGTEXT	0-4 294 967 295 bytes	极大文本数据

- char(n) 和 varchar(n) 中括号中 n 代表字符的个数，并不代表字节个数，比如 CHAR(30) 就可以存储 30 个字符。
- CHAR 和 VARCHAR 类型类似，但它们保存和检索的方式不同。它们的最大长度和是否尾部空格被保留等方面也不同。在存储或检索过程中不进行大小写转换。
- BINARY 和 VARBINARY 类似于 CHAR 和 VARCHAR，不同的是它们包含二进制字符串而不要非二进制字符串。也就是说，它们包含字节字符串而不是字符串。这说明它们没有字符集，并且排序和比较基于列值字节的数值值。
- BLOB 是一个二进制大对象，可以容纳可变数量的数据。有 4 种 BLOB 类型：TINYBLOB、BLOB、MEDIUMBLOB 和 LONGBLOB。它们区别在于可容纳存储范围不同。