

Mysql Data

[零 参考资料 2](#_Toc45729633)

[一 Character Sets 2](#_Toc45729634)

[1 Character Sets & Collations 2](#_Toc45729635)

[2 Unicode 2](#_Toc45729636)

[二 Data Types 2](#_Toc45729637)

[1 数字 2](#_Toc45729638)

[1.1 概述 2](#_Toc45729639)

[1.2 整数 2](#_Toc45729640)

[1.3 Fixed-Point Types 3](#_Toc45729641)

[1.4 浮点 3](#_Toc45729642)

[1.5 其他 4](#_Toc45729643)

[2 时间 4](#_Toc45729644)

[2.1 概述 4](#_Toc45729645)

[3 字符 5](#_Toc45729646)

[4 bit 5](#_Toc45729647)

[5 JSON 5](#_Toc45729648)

# 零 参考资料

《高性能Mysql 第三版》

<https://dev.mysql.com/doc/refman/5.7/en/charset.html>

<https://dev.mysql.com/doc/refman/5.7/en/data-types.html>

<https://dev.mysql.com/doc/refman/5.7/en/literals.html>

<https://dev.mysql.com/doc/refman/5.7/en/precision-math.html>

# Character Sets

## Character Sets & Collations

## Unicode

# Data Types

## 数字

<https://dev.mysql.com/doc/refman/5.7/en/numeric-types.html>

<https://dev.mysql.com/doc/refman/5.7/en/number-literals.html>

### 概述

### 整数

|  |  |
| --- | --- |
| 类型 | 说明 |
| TINYINT | TINYINT[(M)] [UNSIGNED] [ZEROFILL]  存储大小：1byte  有符号取值：-128～127  无符号取值：0-25 |
| SMALLINT | SMALLINT[(M)] [UNSIGNED] [ZEROFILL]  存储大小：2byte  有符号取值：-32768～32767  无符号取：0-65535 |
| MEDUIMINT | MEDIUMINT[(M)] [UNSIGNED] [ZEROFILL]  存储大小：3byte  有符号取值：-8388608 ～ 8388607  无符号取：0 to 16777215 |
| INT | INT[(M)] [UNSIGNED] [ZEROFILL]  存储大小：4byte  有符号取值：-2147483648 ～ 2147483647  无符号取：0- 4294967295 |
| INTEGER | 同INT |
| BIGINT | BIGINT[(M)] [UNSIGNED] [ZEROFILL]  存储大小：8byte  有符号取值：-9223372036854775808～9223372036854775807  无符号取：0- 18446744073709551615 |

### Fixed-Point Types

<https://dev.mysql.com/doc/refman/5.7/en/precision-math.html>

|  |  |
| --- | --- |
| 类型 | 说明 |
| DECIMAL | DECIMAL[(M[,D])] [UNSIGNED] [ZEROFILL]  1 精确的有小数的数值(fixed-point number)，可以存储指定M，D的所有值，比如DECIMAL(5,2)，可以存储-999.99 ～ 999.99  2 M：整数部分位数，不包含小数点和负号，最大值为65，默认值是10  3 D：小数点后面部分位数，取值0-30，默认值为0，当值为0时，值为整数  4 如果指定UNSIGNED，值不能为负数  5 Mysql采用二进制格式存储数值  6 如果小数点后面的数值长度大于D，会被转为D位，一般情况下是直接截断 |
| NUMERIC | NUMERIC[(M[,D])] [UNSIGNED] [ZEROFILL]  同DECIMAL，实际也是以DECIMAL实现 |
| DEC | DEC[(M[,D])] [UNSIGNED] [ZEROFILL]  同DECIMAL |
| FIXED | FIXED[(M[,D])] [UNSIGNED] [ZEROFILL]  同DECIMAIL，为了兼容其他数据库 |
|  |  |
|  |  |
|  |  |

### 浮点

<https://dev.mysql.com/doc/refman/5.7/en/problems-with-float.html>

|  |  |
| --- | --- |
| 类型 | 说明 |
| FLOAT | 所有FLOAT计算，Mysql都使用DOUBLE处理  大小：4byte  FLOAT[(M,D)] [UNSIGNED] [ZEROFILL]  1 理论上取值范围，IEEE标准：  -3.402823466E+38 ～ -1.175494351E-38  0  1.175494351E-38 ～3.402823466E+38  2 实际取值范围依赖OS/硬件  3 M：不包含符号和小数点的总长度  4 D：小数点后面位数  5 如果M和D省略，默认为硬件支持的值范围  6 如果指定UNSIGNED，则不能存储负数  7 非标准Mysql扩展  8 精确到小数点后7位  FLOAT(p) [UNSIGNED] [ZEROFILL]  1 p为精度位数，只用于确定具体值处理方式，不影响实际取值范围   * 1. p值为0-23时，实际处理为无D/M的FLOAT   2. p值为24-53时，实际为无D/M的DOUBLE   2 实际取值范围，根据实际值类型决定  3 此类型是为了兼容ODBC |
| DOUBLE | DOUBLE[(M,D)] [UNSIGNED] [ZEROFILL]  大小：8byte  IEEE标准取值范围：  -1.7976931348623157E+308 ～ -2.2250738585072014E-308  0  2.2250738585072014E-308 to 1.7976931348623157E+308  实际取值范围依赖OS/硬件  如果M和D省略，默认为硬件支持的值范围  精确到小数点后15位  非标准Mysql扩展  如果指定UNSIGNED，则不能存储负数 |
| DOUBLE PRECISION | DOUBLE PRECISION[(M,D)] [UNSIGNED] [ZEROFILL]  等同于DOUBLE |
| REAL | REAL[(M,D)] [UNSIGNED] [ZEROFILL]  当REAL\_AS\_FLOAT配置启用时，等同于FLOAT，否则等同于DOUBLE |

### 其他

<https://dev.mysql.com/doc/refman/5.7/en/bit-value-literals.html>

|  |  |
| --- | --- |
| 类型 | 说明 |
| BIT | BIT[(M)]  M取值1-64，默认是1  如果实际值长度小于M，会左边0补齐  在NDB cluster 中，表中的所有bit字段长度总和不能超过4096 |
| BOOLEAN | 1实际存储为TINYINT(1)；0:false，非0:true  2在Mysql中，FALSE=0，TRUE=1  2.1 SELECT IF(0 = FALSE, 'true', 'false') 》 true;  2.3 SELECT IF(1 = TRUE, 'true', 'false') 》true ;  2.2 SELECT IF(2 = TRUE, 'true', 'false') 》 false; |
| BOOL | 同BOOLEAN |
|  |  |
|  |  |
|  |  |
|  |  |

## 时间

<https://dev.mysql.com/doc/refman/5.7/en/date-and-time-types.html>

<https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html>

### 概述

* + - 1. 日期部分值默认是year-month-day的顺序，其他顺序需要通过STR\_TO\_DATE函数转换的
      2. 年用两位数字表达会自动转成四位：
         1. 年在70-99的会转为1970-1999
         2. 年在00-69的会转为2000-2069
      3. 日期时间跟数字可以根据上下文自动互相转换
      4. 如果赋给时间或日期字段一个越界值或不符合规范的值：
         1. 默认情况下会自动转为对应数据类型的ZERO Value
         2. 如果是TIME类型，则会被裁减到合适的TIME范围内的值

|  |  |
| --- | --- |
| 类型 | 说明 |
| DATA | Zero Value: '0000-00-00' |
| TIME | Zero Value: '00:00:00' |
| DATETIME | Zero Value: '0000-00-00 00:00:00' |
| TIMESTAMP | Zero Value: '0000-00-00 00:00:00' |
| YEAR | Zero Value:0000 |

## 字符

## bit

## JSON