如何创建数据表——数据类型

数据库系统上机

计算机学院&网络空间安全学院 乜鹏

https://dbis.nankai.edu.cn/2019/0417/c12139a128118/page.htm

×

声明：上机课程的内容偏向举例，通俗化，一些术语的准确定义请查看理论课程。

目录 Contents

南開大學 Nankai University

NANKAI UNIVERSITY

投票 最多可选2项

南开大学 Nankai University

1. 设计一个南开大学校友信息的数据库时，需要构建一张国内民族表，以标记每个人员的民族（**mz_id, mz_name**），请问**mz_id**这个字段你会设置为什么类型?

A int

B smallint

C tinyint

D mediumint

E int unsigned

F smallint unsigned

G tinyint unsigned

H mediumint unsigned

投票 最多可选2项

南开大学 Nankai University

2.设计一个南开大学校友信息的数据库时，需要构建一张国家名称表，以标记每个人员的国籍（gj_id, gj_name），请问gj_id这个字段你会设置为什么类型?

A int

B smallint

C tinyint

D mediumint

E int unsigned

F smallint unsigned

G tinyint unsigned

H mediumint unsigned

允公允能 日新月异 NANKAI UNIVERSITY

投票 最多可选2项

南开大学 Nankai University

3.设计一个南开大学校友信息的数据库时，现在要构建主表，也就是用户表（u_id, u_name），请问u_id这个字段你会设置为什么类型?

A int

B smallint

C bigint

D mediumint

E int unsigned

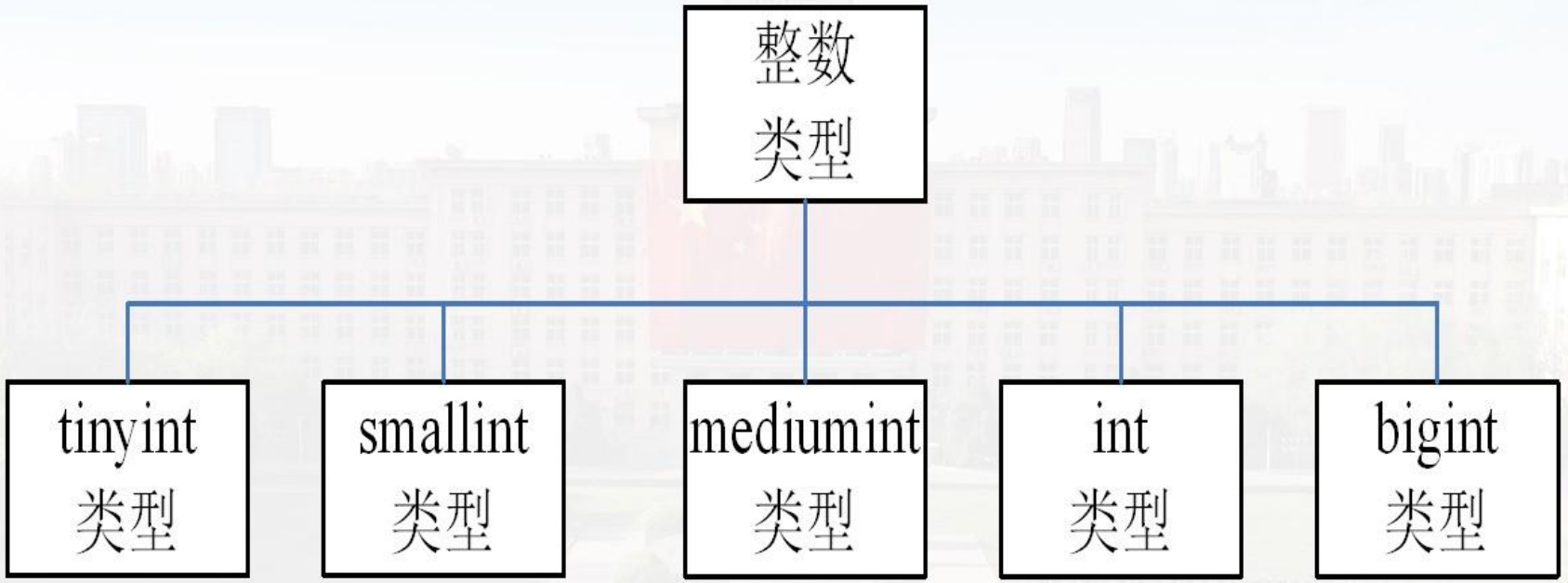
F smallint unsigned

G bigyint unsigned

H mediumint unsigned

允公允能 日新月异 NANKAI UNIVERSITY

MySQL数值类型——整数类型



MySQL数值类型——整数类型

南開大學 Nankai University

整数类型的数，默认情况下既可以表示正整数又可以表示负整数（此时称为有符号数）。如果只希望表示零和正整数，可以使用无符号关键字“**unsigned**”对整数类型进行修饰（此时称为无符号整数）。

例如：score tinyint unsigned

MySQL数值类型——整数类型

南開大學 Nankai University

表1：各类型字节数和取值范围

类型	字节数	范围（有符号）	范围（无符号）
tinyint	1字节	(-128，127)	(0，255)
smallint	2字节	(-32768，32767)	(0，65535)
mediumint	3字节	(-8388608，8388607)	(0，16777215)
int	4字节	(-2147483648，2147483647)	(0，4294967295)
bigint	8字节	(-9233372036854775808，9223372036854775807)	(0，18446744073709551615)

投票 最多可选2项

南开大学 Nankai University

我们经常看见一些数据类型这样定义，**tinyint(1)**，**int(4)**，请问**tinyint(2)**可以存储的最大整数是?

A 2

B 3

C 4

D 127

MySQL数值类型——整数类型

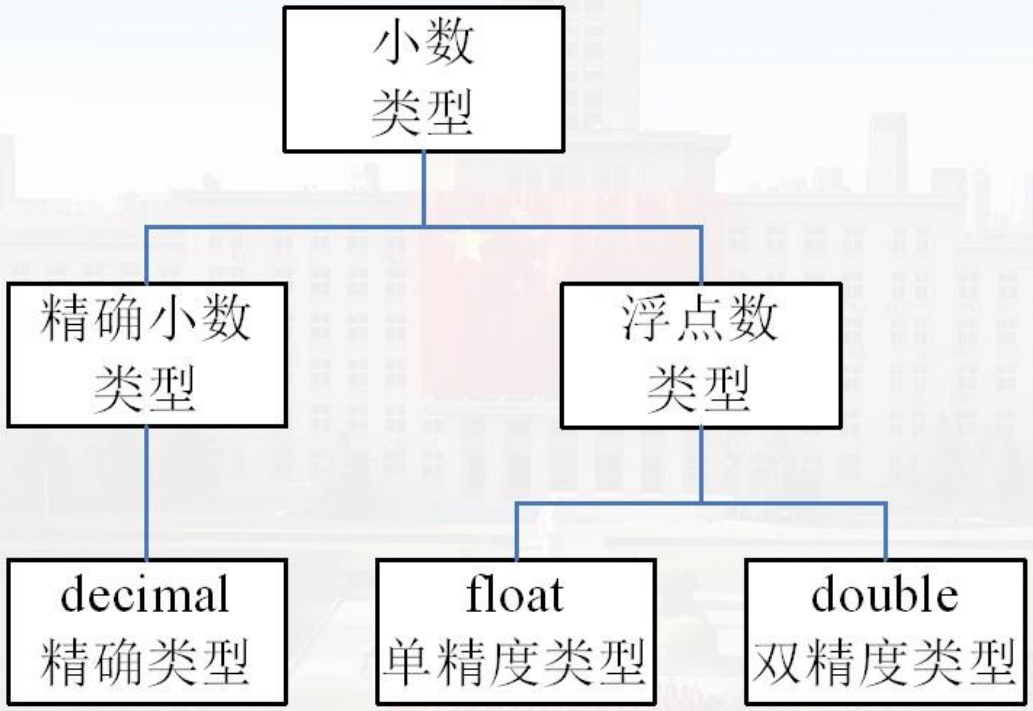
南開大學 Nankai University

int(1)、tinyint(4) 哪个大?

答：**int** 大。

注意数字类型后面括号中的数字，不表示长度，表示的是显示宽度。

MySQL数值类型——小数类型



MySQL数值类型——小数类型

decimal(length, precision)用于表示精度确定（小数点后数字的位数确定）的小数类型，**length**决定了该小数的最大位数，**precision**用于设置精度（小数点后数字的位数）。

例如：

decimal (5,2)表示小数取值范围：**-999.99 ~ 999.99**

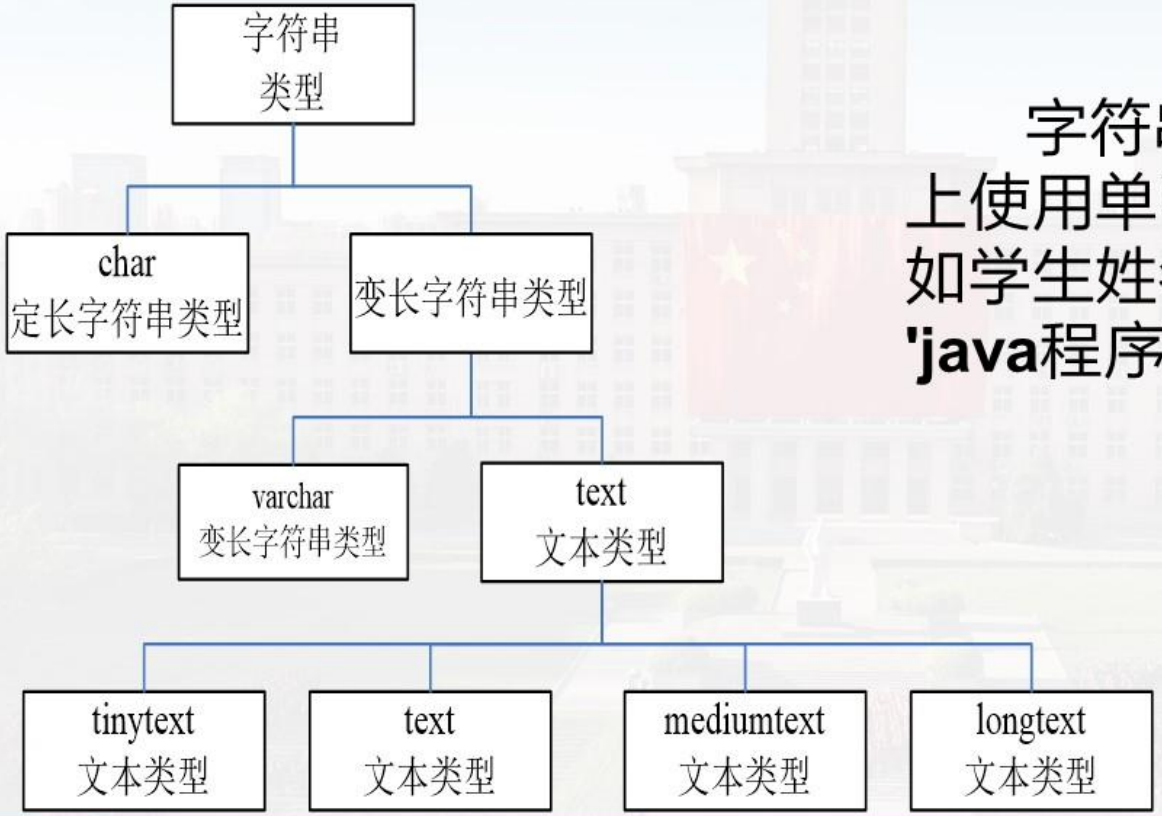
decimal (5,0)表示：**-99999 ~ 99999**的整数。

MySQL数值类型——小数类型

表2：各类型字节数和取值范围

类型	字节数	负数的取值范围	非负数的取值范围
float	4	-3.402823466E+38到-1.175494351E-38	0和1.175494351E-38到3.402823466E+38
double	8	-1.7976931348623157E+308到-2.2250738585072014E-308	0和2.2250738585072014E-308到1.7976931348623157E+308

MySQL字符串类型



字符串类型的数据外观上使用单引号括起来，例如学生姓名'张三'、课程名'java程序设计'等。

MySQL字符串类型——char 与 varchar

南開大學 Nankai University

- char()与varchar():存储或检索过程中不进行大小写转换。对于简体中文gbk编码的字符串而言，varchar(255)表示可以存储255个汉字，而每个汉字占用两个字节的存储空间。假如这个字符串没有那么多汉字，例如仅仅包含一个‘中’字，那么varchar(255)仅仅占用1个字符（两个字节）的储存空间，另加一个字节记录长度.

投票 最多可选1项

南开大学 Nankai University

对于简体中文utf-8编码的字符串而言，包含'中国'两个汉字，那么如果存入varchar(1000)类型的字段中，请问占用几个字节?

A 2

B 3

C 4

D 6

E 8

F 10

G 999

H 1000

MySQL字符串类型——char与 varchar

- 而char(255)则必须占用255个字符长度的存储空间，哪怕里面只存储一个汉字。
- char、varchar，当存储或检索尾部带空格的字符串时有什么区别？

存储或检索带空格的字符串?

南開大學 Nankai University

```
mysql> create table chartest (char_col CHAR(10), varchar_col varchar(10));
Query OK, 0 rows affected (0.03 sec)

mysql> insert into chartest values ('string1', 'string1'), ('  string2', '  string2'), ('string3 ', 'string
3 ')
    -> ;
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

```
mysql> select concat("'", char_col, "'") from chartest;
+---------------------------+
| concat("'", char_col, "'") |
+---------------------------+
| 'string1'                 |
| '  string2'               |
| 'string3'                 |
+---------------------------+
3 rows in set (0.00 sec)
```

```
mysql> select concat("'", varchar_col, "'") from chartest;
+------------------------------+
| concat("'", varchar_col, "'") |
+------------------------------+
| 'string1'                    |
| '  string2'                  |
| 'string3 '                   |
+------------------------------+
3 rows in set (0.00 sec)
```

MySQL字符串类型

- 可以看到，char值在存储尾部有空格的字符串的时候，会将尾部的空格去除，而varchar则不会。对于char和varchar类型，在字符串比较的时候MySQL使用PADSPACE校对规则，会忽略字段末尾的空格字符。如：select * from chartest where char_col='string1 '，可以匹配到结果。如果我们希望进行区分，则需要使用length函数进行辅助。

MySQL字符串类型

```
mysql> select * from chartest where varchar_col='string3' and length(varchar_col)=length('string3');
Empty set (0.00 sec)

mysql> select * from chartest where char_col='string3' and length(char_col)=length('string3');
+----------+-------------+
| char_col | varchar_col |
+----------+-------------+
| string3  | string3     |
+----------+-------------+
1 row in set (0.00 sec)
```

如图，char类型，‘string3 ’可以匹配到'string3' 而 varchar类型则不能

MySQL字符串类型——varchar和char使用

南開大學 Nankai University

- 以下情况使用varchar是合适的：
  - 字符串列的最大长度比平均长度大很多
  - 列的更新次数很少，碎片不成问题
  - UTF-8这样的复杂编码，每个字符都是用不同的字节数存储
- 以下情况使用char是合适的：
  - 字符串非常短或者所有值都接近一个长度，如MD5。
  - 经常变更的值，定长的char不容易产生碎片。
  - 列非常短，char存储效率更高。如char(1)存储Y, N. varchar需要两个字节，其中一个额外字节记录长度。

MySQL字符串类型——varchar和nvarchar使用

南開大學 Nankai University

- varchar和nvarchar的区别
- 1. varchar(n)： 长度为n个字节的可变长度且非Unicode的字符数据。
- 2. nvarchar(n)： 包含n个字符的可变长度Unicode字符数据。

场景：

nvarchar适用中文和其他字符，其中N表示Unicode编码，可以解决多语言之间的转换问题。

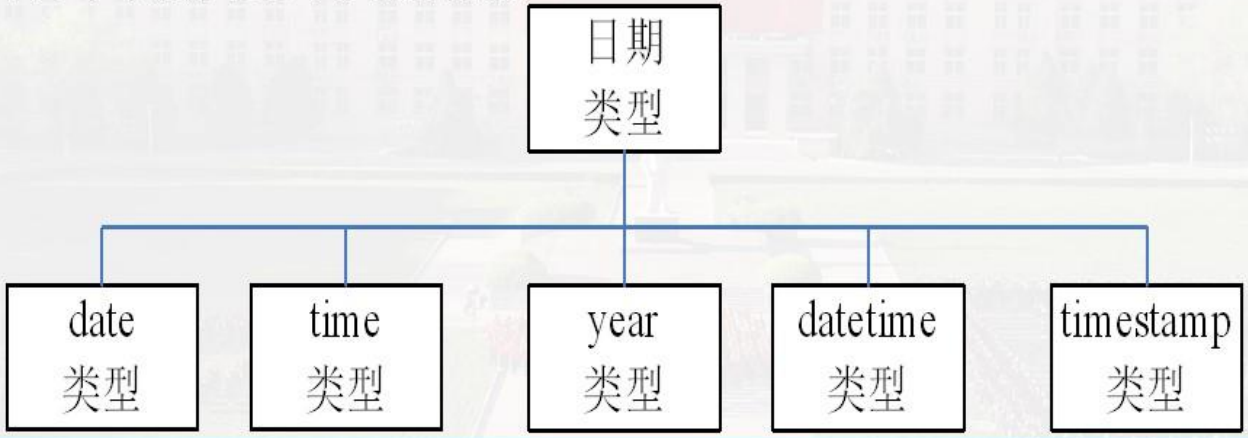
MySQL日期类型

date表示日期，默认格式为'YYYY-MM-DD'；

time表示时间，格式为'HH:mm:ss'；

year表示年份；

datetime与timestamp是日期和时间的混合类型，格式为'YYYY-MM-DD HH:mm:ss'



MySQL日期类型——datetime和timestamp区别

- **datetime**（**8**字节）与**timestamp**（**4**字节）都是日期和时间的混合类型，区别在于：

➢ 表示的取值范围不同，**datetime**的取值范围远远大于**timestamp**的取值范围。

- datetime 保存的时间范围较大，年的数字在：1000---9999之间
- timestamp 保存的时间较小，年的数字在：1970 --2037 之间

MySQL日期类型——datetime和timestamp区别

南開大學 Nankai University

- 将NULL插入timestamp字段后，该字段的值实际上是MySQL服务器当前的日期和时间。

➢ 同一个timestamp类型的日期或时间，不同的时区，显示结果不同。

datetime和timestamp区别

南開大學 Nankai University

```
mysql> create table timetest (datetime_col datetime, timestamp_col timestamp);
Query OK, 0 rows affected (0.02 sec)

mysql> insert into timetest values (null, null);
Query OK, 1 row affected (0.00 sec)

mysql> select * from timetest;
+--------------+---------------------+
| datetime_col | timestamp_col       |
+--------------+---------------------+
| NULL         | 2020-03-16 11:51:35 |
+--------------+---------------------+
1 row in set (0.00 sec)

mysql>
```

投票 最多可选2项

现在设计一个项目管理系统，涉及合同、财务等众多信息，里面有很多需要人工录入时间的数据，并做大量的查询，请问应该用哪种类型？

A date

B datetime

C timestamp

D Int unsigned

datetime和timestamp区别

南開大學 Nankai University

查询时间(秒)		int	timestamp		datetime	
			UNIX_TIMESTAMP	直接和时间比较	UNIX_TIMESTAMP	直接和时间比较
MyISAM	**无索引**	0.0780	0.0780	0.4368	0.7498	0.1370
	有索引	0.3824	0.0780	0.5696	0.7614	0.4508
InnoDB	无索引	0.3092	0.3160	0.7092	0.9794	0.3834
	有索引	0.0522	0.2944	0.1776	0.9994	0.0820

MySQL复合类型

MySQL 支持两种复合数据类型：enum枚举类型和set集合类型。

enum类型的字段类似于单选按钮的功能，一个enum类型的数据最多可以包含65535个元素。

set 类型的字段类似于复选框的功能，一个set类型的数据最多可以包含64个元素。

字符-数字映射表

```
mysql> create table enum_test(
    ->  e enum('fish', 'apple', 'dog') NOT NULL
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql> insert into enum_test(e) values
    -> ('fish'), ('dog'), ('apple');
Query OK, 3 rows affected (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

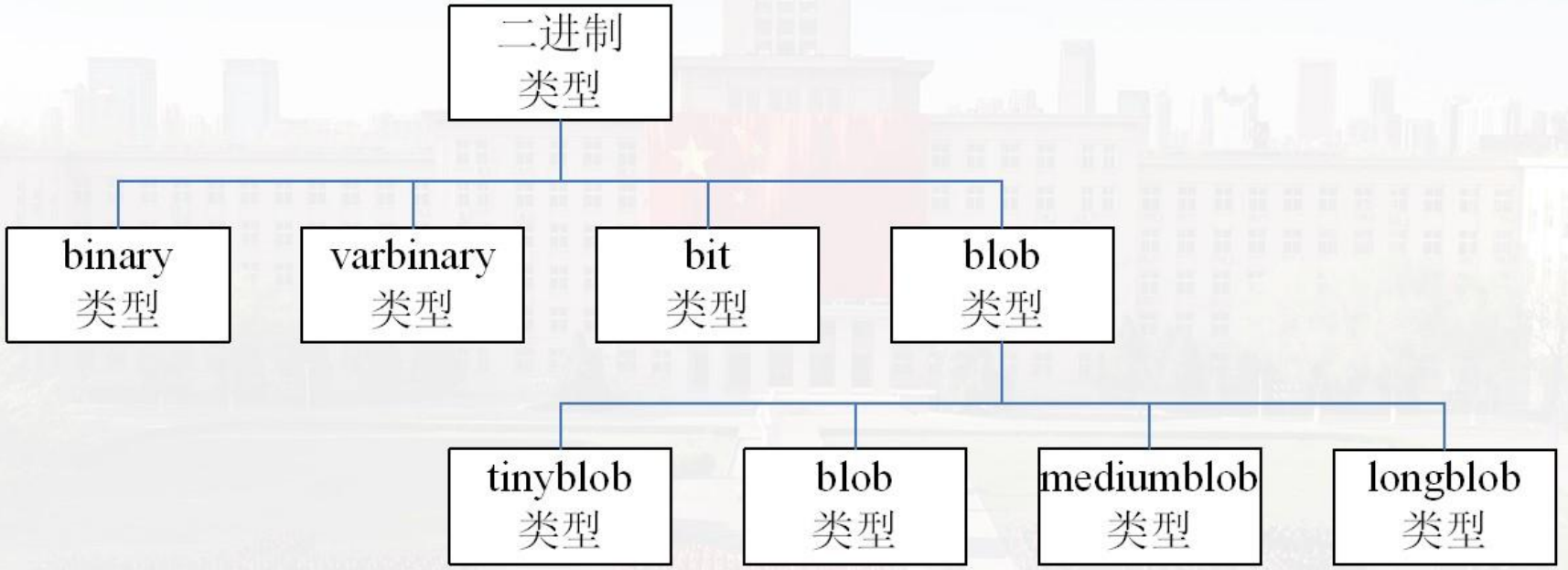
```
mysql> select * from enum_test;
+-------+
| e     |
+-------+
| fish  |
| dog   |
| apple |
+-------+
3 rows in set (0.00 sec)

mysql> select e+1 from enum_test;
+-----+
| e+1 |
+-----+
|   2 |
|   4 |
|   3 |
+-----+
3 rows in set (0.00 sec)
```

三行数据实际存储为整数，而不是字符串，所以e+1可以执行。

MySQL二进制类型

- 二进制类型的字段主要用于存储由‘0’和‘1’组成的字符串，因此从某种意义上将，二进制类型的数据是一种特殊格式的字符串。
- 二进制类型与字符串类型的区别在于：字符串类型的数据按字符为单位进行存储，因此存在多种字符集、多种字符序；而二进制类型的数据按字节为单位进行存储，仅存在二进制字符集binary。

MySQL二进制类型


二进制
类型
binary
类型
varbinary
类型
bit
类型
blob
类型
tinyblob
类型
blob
类型
mediumblob
类型
longblob
类型

Blob和text类型

- blob二进制大数据存储：声音、图像、图片、视频
- Text字符主要是文字方面的数据存储:简介、说明等
- 存储或检索过程中不进行大小写转换，排序和索引使用前几个字符：max_short_length
- 最大值由数据类型确定，客户端和服务器之间传递的最大值由可用内存和通信缓存区大小确定：max_allowd_packet修改

选择最合适的数据类型

选择合适的数据类型，不仅可以节省储存空间，还可以有效地提升数据的计算性能。 通常遵循以下原则：

- 越小越好：在符合应用要求（取值范围、精度）的前提下，尽量使用"短"数据类型
- 简单就好：数据类型越简单越好
- 在**MySQL**中，应该用内置的日期和时间数据类型，而不是用字符串来存储日期和时间。

选择最合适的数据类型

- 尽量采用精确小数类型（例如**decimal**），而不采用浮点数类型。使用精确小数类型不仅能够保证数据计算更为精确，还可以节省储存空间，例如百分比使用**decimal(4,2)**即可。计算时转化为**double**
- 尽量避免**NULL**字段，建议将字段指定为**NOT NULL**约束。**Null**字段很难优化，使得索引、索引统计和值比较都更复杂。
- 整型是最好的选择，位数据类型可以用整数代替。

选择最合适的数据类型

南開大學 Nankai University

IP地址最好选择哪种类型进行存储( )?

- A CHAR
- B INT
- C INT UNSIGNED ←
- D BIGINT

整型可以提高存储效率，存储空间小，查询速度快。

IP的格式是A.B.C.D，其中A,B,C,D均为0~255内的整数，例如127.0.0.1，192.168.53.65。

0~255就是一个8位的2进制的数，00000000（0） - 11111111（255）

整个ip就是一个32位的2进制数，范围是

00000000 00000000 00000000 00000000 0

- 11111111 11111111 11111111 11111111 4294967295

可以转化为一个整数存到数据库里面。

无符号整型满足IP的取值范围。