

mysql数据表设计

课程内容

- 1. varchar与char类型，text类型
- 2. 数字类型
- 3. 时间类型
- 4. 不推荐存储的数据类型

1. varchar与char类型以及text类型

char和varchar都是用来存储字符串类型的数据,但是他们保存和检索的方式不一样.char属于固定长度的字符类型, varchar属于可变长的字符类型

gbk,1个字符，两个字节

值	char(4)	字节	varchar(4)	字节
"	"	8个字节	"	1个字节
'ab'	'ab '	8个字节	'ab'	5个字节
'abcd'	'abcd'	8个字节	'abcd'	9个字节
'abcdefg'	'abcd'	8个字节	'abcd'	9个字节

utf8(1个字符，两个字节)

值	char(4)	字节	varchar(4)	字节
"	"	12个字节	"	1个字节
'ab'	'ab '	12个字节	'ab'	7个字节
'abcd'	'abcd'	12个字节	'abcd'	13个字节
'abcdefg'	'abcd'	12个字节	'abcd'	13个字节

由于char是固定长度的,所以它的处理速度比varchar快得多,但是其缺点是浪费存储空间,程序需要对尾行空格进行处理,所以对那些变化不打并且查询速度有较高的要求的数据可以考虑使用char类型来存储

在mysql中,不同的存储引擎对char和varchar的使用原则有所不同

MyISAM存储引擎

- 建议使用固定长度的数列代替可变长度的数据列

InnoDB存储引擎

- 建议使用varchar类型,对于InnoDB数据表,内部的行存储格式没有区分固定长度和可变长度,因此使用char列不一定比 可变长度的varchar性能好
- 由于char平均占用空间多余varchar,因此varchar来UI消化需要处理的数据航的存储总量和磁盘I/O是比较好的.

2. 数字类型

数字类型解释 <https://blog.csdn.net/zgmu/article/details/52118165>

数据类型	范围	字节

bigint	-2 ⁶³ (-9,223,372,036,854,775,808) 到 2 ⁶³ -1 (9,223,372,036,854,775,807)	8 字节
	0到18446744073709551615	较大整数
int	-2 ³¹ (-2,147,483,648) 到 2 ³¹ -1 (2,147,483,647)	4 个字节
	0到4294967295	标准整数
smallint	-2 ¹⁵ (-32,768) 到 2 ¹⁵ -1 (32,767)	2 字节
	0到65535	较小整数
tinyint	-2 ⁷ (-128) 到 2 ⁷ - 1 (123)	1 字节
	0到255	非常小的整数
float(m,d)	±1.175494351e - 38	4字节
double(m, d)	±2.2250738585072014e - 308	8字节
decimal (m,d)	可变；其值的范围依赖于m 和d	m+2字节

id 必为主键，类型为int bigint unsigned、单表时自增、步长为 1；注意一下因为一些表可能因为数据量的关系导致主键会很大可能会超出int的范围这个时候就比较建议使用bigint通常int即可

注：不过当一个表数据量超过了500万的时候或者单表容量超过2GB的时候推介分库分表；这一步操作是需要实先对于数据量在项目上线之后的思考点

UNSIGNED属性就是将数字类型无符号化, unsigned的使用

注意 unsigned tinyint 的范围就是 0-255

```
CREATE TABLE `yyy` (
  `id` tinyint(3) unsigned NOT NULL,
  PRIMARY KEY (`id`) )
ENGINE=InnoDB DEFAULT CHARSET=utf8
mysql> desc yyy;
+-----+-----+-----+-----+-----+-----+
| Field | Type                | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id    | tinyint(3) unsigned | NO   | PRI | NULL    |       |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> insert into yyy values(255);
Query OK, 1 row affected (0.06 sec)

mysql> select * from yyy;
+-----+
| id |
+-----+
| 255 |
+-----+
1 row in set (0.00 sec)

mysql> insert into yyy values(256);
ERROR 1264 (22003): Out of range value for column 'id' at row 1
mysql> insert into yyy values(-1);
ERROR 1264 (22003): Out of range value for column 'id' at row 1
mysql> insert into yyy values(0);
Query OK, 1 row affected (0.09 sec)
```

人的年龄用 unsigned tinyint（表示范围 0-255，人的寿命不会超过 255 岁）

status,sexuality 相对来说在系统中我们更愿意使用数字作为表示，代替实际该字段所传单的意思，因为char所占用的3个字节，相对来说比较适合于unsigned tinyint

3. 时间类型

注意：

默认情况下，当MySQL遇到超出范围的日期或时间类型的值或该类型的其他无效值时，它会将该值转换为“零”值的值。唯一的例外是超出了范围。TIME值被裁剪到TIME范围。

MySQL允许在DATE或DATETIME列。这对于需要存储您可能不知道确切日期的生日的应用程序非常有用。在这种情况下，您只需将日期存储为'2009-00-00'或'2009-01-00'。如果存储这样的日期，就不应该期望得到正确的结果，例如DATE_SUB()或DATE_ADD()需要完整的日期。若要在日期中不允许零个月或日部分，请启用NO_ZERO_IN_DATE模式。

MySQL允许您存储“零”价值'0000-00-00'作为“假约会。”在某些情况下，这比使用NULL值，并使用较少的数据和索引空间。不允许'0000-00-00'，启用NO_ZERO_DATE模式。

DATETIME

DATETIME 用于表示 年月日 时分秒，是 DATE 和 TIME 的组合，并且记录的年份（见上表）比较长久。如果实际应用中有这样的需求，就可以使用 DATETIME 类型。

- 1、DATETIME列可以设置为多个，默认可为null，可以手动设置其值。
- 2、DATETIME列不可设定默认值。
- 3、DATETIME列可以变相的设定默认值，比如通过触发器、或者在插入数据时候，将DATETIME字段值设置为now()，这样可以做到了，尤其是后者，在程序开发中常常用到。

TIMESTAMP

TIMESTAMP 用于表示 年月日 时分秒，但是记录的年份（见上表）比较短暂，TIMESTAMP列必须有默认值，默认值可以为“0000-00-00 00:00:00”，但不能为null。TIMESTAMP 和时区相关，更能反映当前时间。当插入日期时，会先转换为本地时区后再存放；当查询日期时，会将日期转换为本地时区后再显示。所以不同时区的人看到的同一时间是不同的。表中的第一个 TIMESTAMP 列自动设置为系统时间（CURRENT_TIMESTAMP）。当插入或更新一行，但没有明确给TIMESTAMP列赋值，也会自动设置为当前系统时间。如果表中有第二个TIMESTAMP列，则默认值设置为0000-00-00 00:00:00。TIMESTAMP的属性受Mysql版本和服务器的SQLMode的影响较大。如果记录的日期需要让不同时区的人使用，最好使用TIMESTAMP。

注：一般建表时候，创建时间用datetime，更新时间用timestamp。

DATE

DATE 用于表示 年月日，如果实际应用值需要保存 年月日 就可以使用 DATE。

TIME

TIME 用于表示 时分秒，如果实际应用值需要保存 时分秒 就可以使用 TIME。

YEAR

YEAR 用于表示 年份，YEAR 有 2 位（最好使用4位）和 4 位格式的年。默认是4位。如果实际应用只保存年份，那么用 1 bytes 保存 YEAR 类型完全可以。不但能够节约存储空间，还能提高表的操作效率。

4. 不适合存数据库中的数据

1. 二进制多媒体数据 将二进制多媒体数据存放在数据库中，一个问题是数据库空间资源耗用非常严重，另一个问题是这些数据的存储很消耗数据库主机的CPU资源。这种数据主要包括图片、音频、视频和其他一些相关的二进制文件。这些数据的处理本不是数据的优势，如果我们硬要将他们塞入数据库，肯定会造成数据库的处理资源消耗严重。
2. 流水队列数据 我们都知道，数据库为了保证事务的安全性（支持事务的存储引擎）以及可恢复性，都是需要记录所有变更的日志信息的。而流水队列数据的用途就决定了存放这种数据的表中的数据会不断的被INSERT，UPDATE和DELETE，而每一个操作都会生成与之对应的日志信息。在MySQL中，如果是支持事务的存储引擎，这个日志的产生量更是要翻倍。而如果我们通过一些成熟的第三方队列软件来实现这个Queue数据的处理功能，性能将会成倍的提升。
3. 超大文本数据 对于5.0.3之前的MySQL版本，VARCHAR类型的数据最长只能存放255个字节，如果需要存储更长的文本数据到一个字段，我们就必须使用TEXT类型（最大可存放64KB）的字段，甚至是更大的LONGTEXT类型（最大4GB）。而TEXT类型数据的处理性能要远比VARCHAR类型数据的处理性能低下很多。从5.0.3版本开始，VARCHAR类型的最大长度被调整到64KB了，但是当实际数据小于255Bytes的时候，实际存储空间和实际的数据长度一样，可一旦长度超过255Bytes之后，所占用的存储空间就是实际数据长度的两倍。对于图片的存储，如果说是特殊情况可以使用BLOB，但是通常来说跟推介使用varchar存图片路径，而图片会放在一个文件夹中