## 数据类型

### 时间和日期类型

Year 表示年，格式：YYYY 在数据库中填写方式：‘1901’~‘2155’, 1901~2055, 也可以用两位数字填写，可以是字符串格式，即加上单引号或者是双引号

Date 表示日期，格式：YYYY-MM-DD 在数据库中的填写格式YYYY-MM-DD或者YYYYMMDD 可以是字符串格式，即加上单引号或者是双引号。分隔符-还可以用点或者斜杠代替 “ . ” “/”

Time 表示时间，格式：HH:MM:SS 可以使用格式HH:MM:SS 或者HHMMSS ，单双引号可加可不加。 使用current\_time或者now() 表示系统当前时间。

DateTime 表示时间和日期： 格式：YYYY-MM-DD HH:MM:SS

Timestamp 表示时间和日期：格式：YYYY-MM-DD HH:MM:SS 取值范围比datetime小。使用current\_timestamp表示系统当前时间

使用timestamp是，默认不为空，默认值为curent\_timestamp, on update current\_timestamp

分别对应Java中的类型为data，time, timestamp (java.sql.包下的Java类型)

## 字符串类型：

Char（m） 固定长度字符串，占m个字节

Varchar（m）实际存储的长度为，实际字节数加一

如果字符串后面有空格，char会自动去掉后面的空格，但是varchar不会。

在使用where 条件判断的时候，使用“=” 运算符，会自动将空格忽略

同时由于使用的校验集合，char，varchar，text, set, enum对大小写不敏感。

但是对应数据类型：binary，varbinary，blob是区分大小写的，因为是使用二进制的方式存储数据。

如何实现char，varchar，text等区分大小写：方法一，更改校对集，方式二，使用关键字binary关键字 where ‘a’ = ‘A’, 和where binary ’a’=binary ’A’ 前者为true，后者为false。

**Text 文本类型，用于保存大文本数据。**

**Char，varchar，text对应的Java类型都为String类型**

### Enum类型

定义枚举类型的方式：enum（‘值1’，‘值2’，‘值3’，‘值n’）最多为6335个值

每个值都有一个编号，实际存储的是编号，但是使用的时候还是使用值。

Create table demo (

Sex enum(‘男’，‘女’)，

Name varchar（20）comment ‘姓名’

)

Insert into demo values (‘男’，‘张宽’)

### Set类型

用于保存字符串对象，定义格式为set(‘值1’, ‘值2’, ‘值3’)，但是和enum不同之处在于，一个被声明为set的对象，可以有多个值。

Enum相当于是单选框，set相当于是复选框。

Inster into demo value (“值1, 值2”)

Select \* from where x=”值1, 值2”

对应的Java对象类型为set集合

### Binary[m]和varbinary[m]

存储的是字节，一个可变一个不可变，对应的Java类型为byte[], 数组，而且是字节数组。

### Blob

和text一样，保存数据量很大的数据，但是不同的是，保存的是二进制数据。而text保存的是文本数据。

### Json类型

本质上就是一个字符串

insert into my\_json values ('{"k1":"v1", "k2":"v2"}', '["a", "b"]'); 里面的json格式要用双引号。

## 数字类型

### 整数类型：

Int , bigint, smallint, tinyint, mediumint等 都表示整形数，只是占的字节不同，所能表示的范围也不同。默认是有符号的，要是想实现无符号，要在int 后面加上unsigned，

Id int unsigned comment ‘表示学号’

Int（10）,其中的10表示显示的宽度，显示是个数字，包括符号。但是显示的宽度并不代表取值的范围，如果超过了宽度，就不会显示出来，小于宽度，就会填充空格。如果想要同0填充，就加上 int\_1 int(10) zerofill, 但是这样就不能表示负数。

Tips: 如果一个数被用来参加数学计算，就要使用整形，浮点型，decimal。

但是不参与到数学计算，只是用来展示，建议使用string类型

如果输入的类型和数据库中定义的类型不一致，数据库会尽可能的使用类型转换。

### 浮点数类型

Float和Double，取值范围大，但是精度并不高

### 定点类型

Decimal（d,m）d表示一共有多少个数字，不包括小数点和负号，m表示小数点后有几位。

如果小数位超过，将会对数据进行四舍五入，但是四舍五入之后，数据的范围可能会超过范围。如果数据超过范围，操作将会失败。

对应的Java类型为java.math.Bigdecimal

### Bit类型

Bit类型用来存储二进制数据。

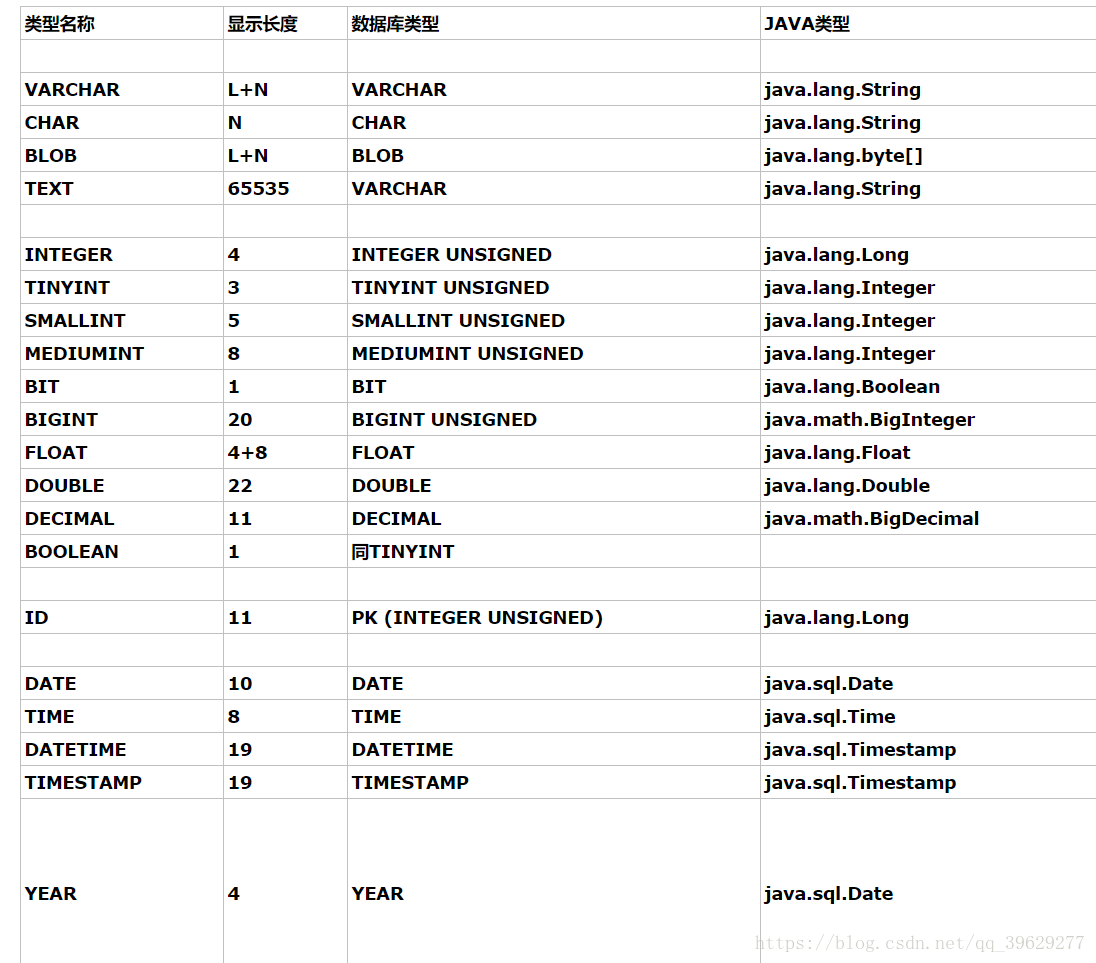
对应的Java类型位Boolean

Varchar,char,text 对应Java中的string

Blog对应的byte[],

Bit 对应的布尔Boolean

Decimal对应的位java.math.BigDecimal



## 数据约束

主要有：主键约束，外键约束，唯一约束，非空约束，默认约束（默认值）

默认约束：default 默认值：可以通过修改字段的属性来修改默认值。

Age int unsigned default 18;

Alter table my\_default modify age int unsigned;

非空约束：not null, 删除非空约束通过修改表的属性即可删除非空约束。添加非空约束的前提是表中该属性的记录中，不含有null。如果有，要先删除再添加。同时插入的数据相应字段不能为空，否则会报错。

一定要有值，可以是手动设置，也可以是用默认值设置的。但是这个值不能位null。

唯一约束：unique，一种是列级约束，一种是表级约束（符合唯一约束）。

列级约束只能是约束一个字段

语法 字段名 字段类型 unique；

表级约束可以约束多个字段。 Unique(字段1，字段2，字段3);

删除唯一约束的方法：像默认值，和非空值，可以通过修改属性的方式来进行删除。但是删除唯一约束，通过修改字段的属性是无法达到删除目的的。而是按照索引的方式来修改的。

Alter table 表名 drop index 约束名

创建索引的完整语法：unique key 索引名 （字段名）。省略索引名的话，就默认位字段名。

其中表级别的唯一约束中，有个**复合唯一约束**：特点是，把他们当作一个整体，只有都相同的时候才视为重复：unique (字段1， 字段2，字段3)只有当这3个字段都相同的时候，才是为重复，unique key 名字 （字段1，字段2， 字段3）；符合唯一索引默认为第一个字段的名字。

主键约束：列级主键约束，表级主键约束（单个属性，复合主键约束）

复合主键约束是多个字段当作一个整体：不为空就是有一个字段不为空即可。不重复是部分字段重复可以，但是不能全部重复。

删除主键约束：alter table 表名 drop primary key，这个时候还剩一个非空约束。

### 自动增长

设置主键之后，每次插入都要检查是否重复，很麻烦，所以就提供了自动增长功能。

不能为空，不能重复。Auto\_incremetn

一个表中只能使用一个属性为自动增长，且这个属性要为主键约束或者是唯一约束。这个属性的类型要为int类型

如果插入的值为0，default，null, 或者小于当前最大值，则自动忽略。生成当前最大值加一。

如果大于当前最大值，则有效。

ENGINE=InnoDB AUTO\_INCREMENT=6 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci

修改下一个增长的基数：alter table my\_auto auto\_increment=10;

删除自动增长：alter table my\_auto modify id int unsign

重新增加自动增长：alter table my\_auto modify id int unsigned auto\_increment

### 外键约束foreign key

[ constraint 约束名称 ] foreign key [外键索引名称] （从表属性）reference 主表名 （主表 中的主键）

[ on delete { restrict | cascade | set null | no action | set default } ]

[ on update { restrict | cascade | set null | no action | set default } ]

默认为拒绝更改相关值 restrict为默认

Set null 设置从表中的值位null

Set default 设置从表中的值为 默认值 目前innoDB中不支持

No action 和 restrict 拒绝主表中更改外键关联的字段。

Cascade 主表中修改值时候，同时从表自动修改或者更新

其中从表中的属性是主表中主键属性的子集。要先建完主表，才能在从表中建立外键。

因为要是主表的子集，所以主表改变会影响从表。

## 字符集和校对集

使用的字符集，可以设置，防止编码错误；charset=utf-8

这个utf8是使用3个字节的。和平常使用的utf-8还是有区别的。想一些emoj表情五福显示。所以引入了uft8mb4

校对集：指点字符集的比较和排序规则：Latil\_swedish\_ci (使用的字符集\_国家名\_比较方式)

Ci不区分大小写，cs区分大小写，bin二进制方式比较。

## 数据库的3大范式

第一范式:属性的原子性，不可以再分

第二范式：一范式的基础上，其他属性完全依赖于主键，而不是部分依赖于主键（这个是对于复合主键来说的。两个属性作为一个整体，共同决定一条记录）

第三范式：前两个的基础上，不存在依赖传递

不符合三大范式，通过拆分表格的方法，使得符合三大范式，但是这样有表格数量增加了，查询就要多张表的联合，导致查询效率变低。

反范式：不符合三大范式，但是效率高，通过一些存储过程，触发器，代码的逻辑上，保证数据不出错

## 集合中元素为空和空集合

Null相当于是一个啥都没有

‘’ 表示一个空字符串，只是没有指定内容。类使于一个集合，但是里面元素个数为零。