

0410约束

真正约束**字段**的是数据类型，但是数据类型约束很单一，需要有一些额外的约束，更好的保证数据的合法性，从业务逻辑角度保证数据的正确性。

表的约束很多，这里主要介绍如下几个：`null/not null,default, comment, zerofill, primary key, auto_increment, unique key`。

5.1 空属性

- 两个值：`null`（默认的）和`not null`(不为空)
- **数据库默认字段基本都是字段为空**，但是实际开发时，尽可能保证字段不为空，因为数据为空没办法参与运算。

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| lesson2 |
| lesson3 |
| mysql |
| performance_schema |
| sys |
+-----+
6 rows in set (0.00 sec)

mysql> create database lesson4;
Query OK, 1 row affected (0.00 sec)

mysql> use lesson4
Database changed
mysql> show tables;
Empty set (0.00 sec)

mysql>
```

```
mysql> insert into myclass (class_name) values('105');
ERROR 1364 (HY000): Field 'class_room' doesn't have a default value
mysql>

mysql> create table if not exists `myclass` (
    -> class_name varchar(20) not null,
    -> class_room varchar(20) not null
    -> )engine=InnoDB default charset=utf8;
Query OK, 0 rows affected (0.05 sec)

mysql> desc myclass
-> ;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| class_name | varchar(20) | NO | | NULL | |
| class_room | varchar(20) | NO | | NULL | |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

如果我们只插入一个值，这是肯定不允许的

这个就是非空约束

5.2 默认值

默认值：某一种数据会经常性的出现某个具体的值，可以在一开始就指定好，在需要真实数据的时候，用户可以选择性的使用默认值。

```
mysql> create table tt10 (  
-> name varchar(20) not null,  
-> age tinyint unsigned default 0,  
-> sex char(2) default '男'  
-> );  
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> create table if not exists t1(  
-> name varchar(20) not null,  
-> age tinyint unsigned default 18,  
-> gender char(2) default '男'  
-> );  
Query OK, 0 rows affected (0.04 sec)
```

• 设置缺省值

```
mysql> desc t1;  
+-----+-----+-----+-----+-----+-----+  
| Field | Type                | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| name  | varchar(20)         | NO   |     | NULL    |      |  
| age   | tinyint(3) unsigned | YES  |     | 18      |      |  
| gender | char(2)             | YES  |     | 男      |      |  
+-----+-----+-----+-----+-----+-----+  
3 rows in set (0.01 sec)
```

```
mysql> █
```

```
MySQL@[lesson4] > desc t1;  
+-----+-----+-----+-----+-----+-----+  
| Field | Type                | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| name  | varchar(20)         | NO   |     | NULL    |      |  
| age   | tinyint(3) unsigned | YES  |     | 18      |      |  
| gender | char(2)             | YES  |     | 男      |      |  
+-----+-----+-----+-----+-----+-----+  
3 rows in set (0.00 sec)
```

```
MySQL@[lesson4] > insert into t1(name) values ('zhangsan');  
Query OK, 1 row affected (0.00 sec)
```

```
MySQL@[lesson4] > select * from t1;  
+-----+-----+-----+  
| name | age | gender |  
+-----+-----+-----+  
| zhangsan | 18 | 男 |  
+-----+-----+-----+  
1 row in set (0.00 sec)
```

虽然我们没有指明，但是我们有默认值
所以mysql自动帮我们插入了

```
MySQL@[lesson4] > █
```

```
MySQL[lesson4] > create table t2(
-> name varchar(20) not null,
-> age int not null default 18
-> );
```

Query OK, 0 rows affected (0.03 sec)

```
MySQL[lesson4] > desc t2;
```

Field	Type	Null	Key	Default	Extra
name	varchar(20)	NO		NULL	
age	int(11)	NO		18	

2 rows in set (0.00 sec)

```
MySQL[lesson4] > 
```

如果not null和default都设置了

当我们显式插入NULL的时候，是不行的

但是如果忽略那一列，此时就是用default，可以插入

```
MySQL[lesson4] > insert into t1 values ('zhangsanfeng',100);
ERROR 1136 (21S01): Column count doesn't match value count at row 1
MySQL[lesson4] > insert into t2 values ('zhangsanfeng',100);
Query OK, 1 row affected (0.00 sec)
```

```
MySQL[lesson4] > insert into t2 values ('Wangwu');
ERROR 1136 (21S01): Column count doesn't match value count at row 1
```

```
MySQL[lesson4] > insert into t2 values ('Wangwu',NULL);
```

```
ERROR 1048 (23000): Column 'age' cannot be null
```

```
MySQL[lesson4] > insert into t2 (name) values ('Wangwu');
```

```
Query OK, 1 row affected (0.00 sec)
```

```
MySQL[lesson4] > select * from t2
```

```
-> ;
```

name	age
zhangsanfeng	100
Wangwu	18

2 rows in set (0.00 sec)

```
MySQL[lesson4] > 
```

这样是不行的

这样是可以的!

```
MySQL@lesson4] > create table if not exists t3(
-> name varchar(20) not null,
-> age tinyint default 18 comment '这里是用户年纪',
-> gender char(1) not null default '男' comment '这里是用户性别'
-> );
```

Query OK, 0 rows affected (0.03 sec)

```
MySQL@lesson4] > desc t3;
```

Field	Type	Null	Key	Default	Extra
name	varchar(20)	NO		NULL	
age	tinyint(4)	YES		18	
gender	char(1)	NO		男	

3 rows in set (0.01 sec)

```
MySQL@lesson4] > █
```

```
MySQL@lesson4] > insert into t3 (name, age, gender) values ('张无忌',19,'男');
Query OK, 1 row affected (0.00 sec)
```

```
MySQL@lesson4] > insert into t3 (name, age) values ('张无忌',19);
Query OK, 1 row affected (0.00 sec)
```

```
MySQL@lesson4] > insert into t3 (name, age) values ('张无忌', NULL);
Query OK, 1 row affected (0.00 sec)
```

```
MySQL@lesson4] > select * from t3;
```

name	age	gender
张无忌	19	男
张无忌	19	男
张无忌	NULL	男

3 rows in set (0.00 sec)

```
MySQL@lesson4] > █
```

如果不写，就是 default

如果你都设置成 NULL 了
default 也没用，插入的就是 NULL

注意: not null和default一般不需要同时出现，因为default本身有默认值，不会为空

5.4 zerofill

刚开始学习数据库时，很多人对**数字类型**后面的长度很迷茫。通过show看看tt3表的建表语句：

int 后面那个()数字是什么呢？

```
MySQL@<lesson4> $ create table if not exists t4(
-> a int not null,
-> b int unsigned not null
-> );
Query OK, 0 rows affected (0.02 sec)

MySQL@<lesson4> $ desc t4;
+-----+-----+-----+-----+-----+-----+
| Field | Type                | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| a     | int(11)              | NO   |     | NULL    |       |
| b     | int(10) unsigned     | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

MySQL@<lesson4> $
```

可以看到int(10),这个代表什么意思呢？整型不是4字节码？这个10又代表什么呢？其实没有zerofill这个属性，括号内的数字是毫无意义的。a和b列就是前面插入的数据，如下：

这次可以看到a的值由原来的1变成00001，这就是zerofill属性的作用，如果宽度小于设定的宽度（这里设置的是5），自动填充0。要注意的是，这只是最后显示的结果，在MySQL中实际存储的还是1。为什么是这样呢？我们可以用hex函数来证明。

其实这个数字就是打印的时候的一个格式而已
就是打印的位数

5.5 主键

主键：primary key用来唯一的约束该字段里面的数据，不能重复，不能为空，**一张表中最多只能有一个主键**，主键所在的列通常是整数类型。

```
MySQL@<lesson4> $ create table t5(
-> id int unsigned primary key comment '学号不能为空',
-> name varchar(20) not null
-> );
Query OK, 0 rows affected (0.04 sec)
```

```
MySQL@<lesson4> $ desc t5;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id    | int(10) unsigned | NO   | PRI | NULL    |       |
| name  | varchar(20)      | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

不指定 not null
也会自动带有不能为空的属性

这就是主键约束

```
MySQL@<lesson4> $ alter table t5 drop primary key;
Query OK, 3 rows affected (0.07 sec)
Records: 3 Duplicates: 0 Warnings: 0
```

去掉主键约束

```
MySQL@<lesson4> $ desc t5;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id    | int(10) unsigned | NO   |     | NULL    |       |
| name  | varchar(20)      | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

```
MySQL@<lesson4> $
```

```
MySQL@<lesson4> $ insert into t5 (id,name) values (1, 'zhangsan');
Query OK, 1 row affected (0.00 sec)
```

```
MySQL@<lesson4> $ insert into t5 (id,name) values (2, 'lisi');
Query OK, 1 row affected (0.01 sec)
```

```
MySQL@<lesson4> $ insert into t5 (id,name) values (3, 'lisi');
Query OK, 1 row affected (0.01 sec)
```

```
MySQL@<lesson4> $ insert into t5 (id,name) values (2, 'wangwu');
ERROR 1062 (23000): Duplicate entry '2' for key 'PRIMARY'
MySQL@<lesson4> $ select * from t5l
```

-> ;

```
ERROR 1146 (42S02): Table 'lesson4.t5l' doesn't exist
```

```
MySQL@<lesson4> $ select * from t5;
```

```
+-----+-----+
| id | name |
+-----+-----+
| 1  | zhangsan |
| 2  | lisi    |
| 3  | lisi    |
+-----+-----+
3 rows in set (0.00 sec)
```

```
MySQL@<lesson4> $
```

```
MySQL@<lesson4> $ alter table t5 primary key (id);
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual
that corresponds to your MySQL server version for the right syntax to use
near 'primary key (id)' at line 1
MySQL@<lesson4> $
```

加上主键

- 复合主键

在创建表的时候，在所有字段之后，使用primary key(主键字段列表)来创建主键，如果有多个字段作为主键，可以使用复合主键。

```
mysql> create table tt14(
-> id int unsigned,
-> course char(10) comment '课程代码',
-> score tinyint unsigned default 60 comment '成绩',
-> primary key(id, course) -- id和course为复合主键
-> );
```

```
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual
that corresponds to your MySQL server version for the right syntax to use
near '(id, course)' at line 5
```

```
MySQL@<lesson4> $ create table t6( id varchar(20), course varchar(20) comm
ent 'course code', score tinyint unsigned default 60 ,primary key(id, cour
se) );
Query OK, 0 rows affected (0.03 sec)
```

```
MySQL@<lesson4> $ desc t6;
```

Field	Type	Null	Key	Default	Extra
id	varchar(20)	NO	PRI	NULL	
course	varchar(20)	NO	PRI	NULL	
score	tinyint(3) unsigned	YES		60	

3 rows in set (0.00 sec)

```
MySQL@<lesson4> $
```

```
MySQL@<lesson4> $ insert into t6(id,course,score) values ('0002','0004',84
);
Query OK, 1 row affected (0.02 sec)
```

```
MySQL@<lesson4> $ insert into t6(id,course,score) values ('0001','0009',56
);
Query OK, 1 row affected (0.00 sec)
```

```
MySQL@<lesson4> $ select * from t6;
```

id	course	score
0001	0004	78
0001	0009	56
0002	0004	84

3 rows in set (0.00 sec)

```
MySQL@<lesson4> $
```

如果设置了复合主键

不允许的是两个都重复
一个重复是可以的

5.6 自增长

auto_increment: 当对应的字段, 不给值, 会自动的被系统触发, 系统会从当前字段中已经有的最大值+1操作, 得到一个新的不同的值。通常和主键搭配使用, 作为逻辑主键。

自增长的特点:

- 任何一个字段要做自增长, 前提是本身是一个索引 (key一栏有值)
- 自增长字段必须是整数
- 一张表最多只能有一个自增长

如何设计主键呢?

1. 结合你的业务, 我就可以选择一个唯一的列作为主键
2. 选择与业务无关的唯一值 (特定设计的)

这样设计, 主键和唯一值无关, 业务调整, 不影响整体主键的表结构

自增长就是方案二

只要一个字段被设置成自增长, 那么它一定要被定义成主键

```
MySQL@<lesson4> $ create table t7( id int unsigned primary key auto_increment, name varchar(20) not null );
Query OK, 0 rows affected (0.03 sec)
```

```
MySQL@<lesson4> $ desc t7;
```

Field	Type	Null	Key	Default	Extra
id	int(10) unsigned	NO	PRI	NULL	auto_increment
name	varchar(20)	NO		NULL	

2 rows in set (0.00 sec)

```
MySQL@<lesson4> $
```

```

MySQL@<lesson4> $ insert into (name) values ('张飞');
ERROR 1064 (42000): You have an error in your SQL syntax; check the manu
that corresponds to your MySQL server version for the right syntax to u
near '(name) values ('张飞')' at line 1
MySQL@<lesson4> $ insert into t7 (name) values ('张飞');
Query OK, 1 row affected (0.01 sec)

MySQL@<lesson4> $
MySQL@<lesson4> $ insert into t7 (name) values ('张飞');
Query OK, 1 row affected (0.00 sec)

MySQL@<lesson4> $ insert into t7 (name) values ('张飞');
Query OK, 1 row affected (0.01 sec)

MySQL@<lesson4> $ insert into t7 (name) values ('张飞');
Query OK, 1 row affected (0.00 sec)

MySQL@<lesson4> $ select * from t7;
+----+-----+
| id | name |
+----+-----+
| 1  | 张飞 |
| 2  | 张飞 |
| 3  | 张飞 |
| 4  | 张飞 |
+----+-----+
4 rows in set (0.01 sec)

MySQL@<lesson4> $

```

我们没有指明id,
但是

如果我们非要自己插入auto_increment的数据也能插，不过不能重复

如果我们一开始让他自动插入1, 2, 3, 4, 5
然后手动插入1000

那么再自动插入的时候，插入的就是1001了

如果一开始我就想让他从10000开始呢？

```

MySQL@<lesson4> $ create table t8(
-> id bigint unsigned auto_increment,
-> name varchar(20) not null,
-> primary key(id)
-> )auto_increment=10000;
Query OK, 0 rows affected (0.03 sec)

MySQL@<lesson4> $ desc t8;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id    | bigint(20) unsigned | NO   | PRI | NULL    | auto_increment |
| name  | varchar(20)      | NO   |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

MySQL@<lesson4> $ insert into t8 (name) values('张飞');
Query OK, 1 row affected (0.01 sec)

MySQL@<lesson4> $ insert into t8 (name) values('张飞');
Query OK, 1 row affected (0.01 sec)

MySQL@<lesson4> $ select * from t8;
+----+-----+
| id  | name |
+----+-----+
| 10000 | 张飞 |
| 10001 | 张飞 |
+----+-----+
2 rows in set (0.00 sec)

MySQL@<lesson4> $

```

5.7 唯一键

一张表中有往往有很多字段需要唯一性，数据不能重复，但是一张表中只能有一个主键：唯一键就可以解决表中有多多个字段需要唯一性约束的问题。

唯一键的本质和主键差不多，唯一键允许为空，而且可以多个为空，空字段不做唯一性比较。

关于唯一键和主键的区别：

我们可以简单理解成，主键更多的是标识唯一性的。而唯一键更多的是保证在业务上，不要和别的信息出现重复。乍一听好像没啥区别，我们举一个例子

和主键的区别是什么？

```
-> ^C
MySQL<lesson4> $ create table if not exists stus(
-> id bigint unsigned primary key auto_increment,
-> name varchar(20) not null,
-> qq varchar(32) unique comment "qq号需要唯一",
-> telephone varchar(16) unique
-> )auto_increment=1000;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual
that corresponds to your MySQL server version for the right syntax to use
near 'bigint unsigned primary key auto_increment,
name varchar(20) not null,
qq varcha' at line 2
MySQL<lesson4> $ create table if not exists stus( id bigint unsigned prim
ary key auto_increment, name varchar(20) not null, qq varchar(32) unique c
omment "qq号需要唯一", telephone varchar(16) unique )auto_increment=1000;
Query OK, 0 rows affected (0.03 sec)

MySQL<lesson4> $ desc stus;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id | bigint(20) unsigned | NO | PRI | NULL | auto_increment |
| name | varchar(20) | NO | | NULL | |
| qq | varchar(32) | YES | UNI | NULL | |
| telephone | varchar(16) | YES | UNI | NULL | |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

MySQL<lesson4> $ inse
```