

## 05 数据库约束&面试题

2024全新MySQL企业开发版





- 1 约束概述和分类
- 2 域 (列) 级约束
- **3** 实体 (行) 级约束
- 4 参照引用 (外键) 约束
- 5 约束常见面试题



## 1 约束概述和分类

### 1 约束概述和分类

### U) 尚硅谷 www.atguigu.com

### "约束" 概念



约束概念:表级别的规定,数据的限制语法

约束作用: 确保表数据的准确性、可靠性、正确性

#### 添加时机:

- 1.创建表时直接添加(CREATE TABLE)
- 2.创建表之后,通过ALTER TABLE 语句添加

### 1 约束概述和分类



### 约束分类

① 域(列)级约束

此类约束,只对当前列值有效果 例如:某列不能为null等

<mark>非空约束</mark>:NOT NULL, 列非空约

束!

默认值约束: DEFAULT, 某列默认

值!

<mark>检查约束</mark>:CHECK,检查约束!

2 实体(行)级约束

此类约束,需要对比同一表中其他行数据,才有效果!例如:某列值必须唯一等

主键约束:PRIMARY KEY,主键唯一

且不为空约束!

唯一约束: UNIQUE, 限制某一列值表

中唯一!

自增长约束: AUTO INCREMENT,数

字类型字段插入数据自增长约束!

③ 引用(多表)级约束

此类约束,需要对比其他表的列才有效

果!

例如:分数表中引用学生表学号等

参照 (外键) 约束: FOREIGN KEY,

限定表中某一列,正确引用其他表的

数据值!

约束学习主要三个方面:作用、添加、删除



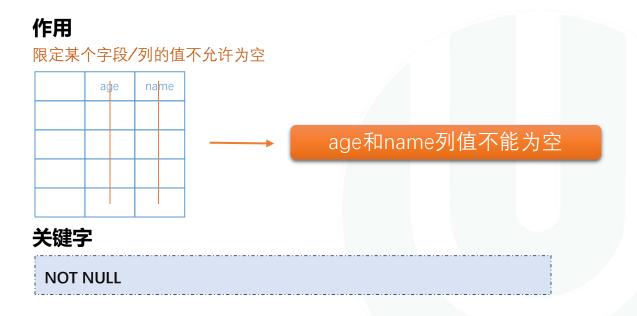


- 1 约束概述和分类
- 2 域 (列) 级约束
- **3** 实体 (行) 级约束
- 4 参照引用 (外键) 约束
- 5 约束常见面试题





### 2.1 非空约束



### 特点

<mark>¦ 默认</mark>:所有类型列默认都可以为null,包括数字类型。

,<mark>列上添加</mark>:非空约束只能添加到列上!

<mark>多次使用</mark>:一个表中可以有很多列进行非空限定!

<mark>空值判定</mark>:空字符串不是null,0也不是null!

### U) 尚硅谷 www.atguigu.com

### 2.1 非空约束

### 添加

### 建表时添加

```
CREATE TABLE 表名称(
字段名 数据类型,
字段名 数据类型 NOT NULL,
字段名 数据类型 NOT NULL
);
```

### 建表后修改

alter table 表名称 modify 字段名 数据类型 not null;

### 删除

```
alter table 表名称 modify 字段名 数据类型 NULL;
或
alter table 表名称 modify 字段名 数据类型;
#不加默认允许为null,
```

#### 示例:

```
CREATE TABLE emp1(
id INT(10) NOT NULL,

NAME VARCHAR(20) NOT NULL,

sex CHAR(1)
);
```

### 示例:

alter table emp1 modify sex char(1) not null;

#### 示例:

```
alter table emp1 modify sex varchar(30) null;
```

### U) 尚硅谷 www.atguigu.com

### 2.2 默认值约束

### 作用

限定某个字段/某列的添加默认值



### 关键字

DEFAULT 默认值

### 特点

位置约定: 默认值约束不能添加到唯一或者主键上,其他列都可以。

<mark>生效时机</mark>:当插入数据时,没有显示赋值,赋予默认值!

<mark>细节特点</mark>:添加约束时,default 默认值,默认值对应正确数据类型!



## 2.2 默认值约束添加

### 建表时添加

create table 表名称(

字段名 数据类型 default 默认值,

字段名 数据类型 not null default 默认值);

### 建表后修改

alter table 表名称 modify 字段名 数据类型 default 默认值; #如果这个字段原来有非空约束,你还保留非空约束,那么在加默认值约束时,还得保留非空约束,否则非空约束就被删除了 alter table 表名称 modify 字段名 数据类型 default 默认值 not null;

#### 删除

alter table 表名称 modify 字段名 数据类型;

#删除默认值约束, 也不保留非空约束

alter table 表名称 modify 字段名 数据类型 not null;

#删除默认值约束, 保留非空约束

#### 示例:

**CREATE TABLE** emp2(

id INT(10) default 0,

NAME VARCHAR(20) NOT NULL default '二狗子');

#### 示例:

alter table emp2 modify name varchar(20) default "not null;

#给gender字段增加默认值约束

alter table emp2 modify tel char(11) default ";

#给tel字段增加默认值约束

#### 示例:

alter table employee modify gender char;

#删除gender字段默认值约束,如果有非空约束,也一并删除

alter table employee modify tel char(11) not null;

#删除tel字段默认值约束, 保留非空约束

### U 尚硅谷 www.atguigu.com

### 2.3 检查约束

### 作用

限检查某个字段的值是否符号xx要求,一般指的是值的范围

age	sex			
		<b>─</b>	sex添加检查约束,必须男和女	

### 关键字

CHECK (限制表达式)

### 特点

<mark>新特性</mark>:5.7版本不支持check约束,8+版本才支持check约束。

<mark>万能约束</mark>:check(表达式),可以自定义表达式,变成任何约束!

<mark>不推荐</mark>:不推荐使用check约束,进行数据检查,建议程序级限制!

### 2.1 检查约束

#### 添加

### 建表时添加

create table 表名称(

字段名 数据类型,

check(表达式), #check约束属于表级别,不用添加到列后

字段名 数据类型 not null default 默认值);

### 建表后修改

alter table 表名 add constraint 约束名 CHECK (表达式); # 约束名不能重复

#### 删除

alter table 表名 drop constraint 约束名;



#### 示例:

```
CREATE TABLE emp3(
gender CHAR ,
CHECK (gender IN ('男','女')),
age INT ,
CHECK(age > 20) );
```

### 扩展: 查看表中的所有约束

```
SELECT *
FROM INFORMATION_SCHEMA.TABLE_CONSTRAINTS
WHERE TABLE_SCHEMA = 'your_database_name'
AND TABLE_NAME = 'your_table_name';
your_database_name 是你的数据库名称。
your_table_name 是你的表名。
```





- 1 约束概述和分类
- 2 域 (列) 级约束
- **3** 实体 (行) 级约束
- 4 参照引用 (外键) 约束
- 5 约束常见面试题





### 3.1 唯一约束



### 特点

**约束数量**:同一个表可以有多个唯一约束。

<mark>空值处理</mark>:唯一性约束允许列值为空。

<mark>约束名称</mark>:在创建唯一约束的时候,如果不给唯一约束命名,就默认和列名相同。

### 3 实体 (表) 级约束

#### U 尚 链 S www.atguigu.com

### 3.1 唯一约束 添加

### 建表时添加

create table 表名称(
字段名 数据类型 unique,
字段名 数据类型 unique key);
create table 表名称(
字段名 数据类型,
[constraint 约束名] unique key(字段名));

### 建表后修改

ALTER TABLE table\_name
ADD CONSTRAINT constraint\_name UNIQUE(列名,列名);

### 删除

# 查看约束
SELECT \*
FROM INFORMATION\_SCHEMA.TABLE\_CONSTRAINTS
WHERE TABLE\_SCHEMA = '库名'
AND TABLE\_NAME = '表名';
# 删除约束
ALTER TABLE table\_name

DROP CONSTRAINT constraint\_name;

### 实体(行)级约束

### 3.2 主键约束

#### 什么是主键?



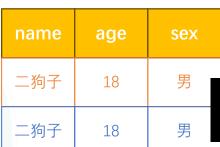


姓名: 二狗子

年龄: 18 性别: 男

姓名: 二狗子 年龄: 18 性别: 男

#### 学生表



在任何情况下,确保表中的行数据少有一列是不重复的! 避免整行数据的不重复! 那么永远不重复,且不会为null的列,我们就可以称作: 主键列(主键) 例如: 学生的学号、员工的编号、以及我们的身份证号等等

学号	name	age	sex
1	二狗子	18	男
2	二狗子	18	男



### 3.2 主键约束

#### 主键分类

自定义主键:人为的创建一列,专门用来做主键,它的使命就是保证行数据不重复!例如:学号,身份证号等等

**自然主键**: 不是人为创建的列,数据实体自带的属性列,当前环境下,此属性列唯一且不为空,可以做主键,但是它有两个使

命:第一,承担实体属性值 第二,作为表的主键列! 例如: DNA序列等等

学号	name	age	sex
1	二狗子	18	男
2	二狗子	18	男

→ 自定义主键

### 推荐使用自定义主键



### 3.2 主键约束

#### 主键细节说明

<mark>主键数量</mark>:每个表中只能有`<mark>一个`主键</mark>。

<mark>单一和复合</mark>: 主键可以由<mark>单个</mark>列或者<mark>多个</mark>列构成(<mark>复合主键</mark>)。

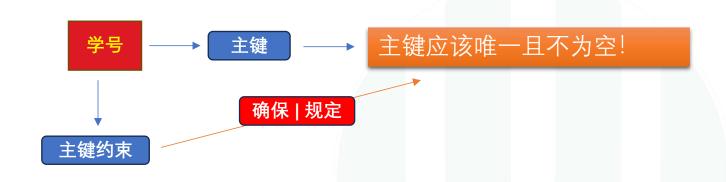
<mark>主键列类型</mark>:可以是<mark>任意类型</mark>,只要唯一且不重复即可。

<mark>主键命名</mark>:主键一般采用identify(标识)单词缩写 <mark>xxid</mark> | <mark>xx\_id </mark>等,但没有强制要求。

主键索引:创建主键约束时,系统默认会在所在的列或列组合上建立对应的**主键索引**(能够根据主键查询的,就根据主键查询、效率更高)。如果删除主键约束了,主键约束对应的索引就自动删除,主键索引固定命令:PRIMARY。



### 3.2 主键约束



主键: 确保行数据不重复的列, 主键列应该唯一且不为空

主键约束: 针对主键列的数据约束和限制, 确保主键列不会出现错误数据(保证唯一且不为空)



主键列一定添加主键约束么?

没有主键约束,那么还是不是主键列呢?

#### U 尚 链 S www.atguigu.com

### 3.2 主键约束

### 添加

### 建表时添加

```
create table 表名称(
字段名 数据类型 primary key, #列级模式
);
create table 表名称(
字段名 数据类型,
[constraint 约束名] primary key(字段名) #表级模式
);
```

### 建表后修改

ALTER TABLE 表名称 ADD PRIMARY KEY(字段列表); #字段列表可以是一个字段,也可以是多个字段

### 删除

删除主键约束,不需要指定主键名,因为一个表只有一个主键,删除主键约束后,<mark>非空还存在,唯一消失</mark>。

#删除主键约束和索引

alter table 表名称 drop primary key;



### 3.3 自增长约束

### 作用

限定某个整数类型字段,插入数据不显示维护,值自动增长!

id	age	phone
1		
2		
3		
4		

id是主键列,且值自动增长

### 关键字

**AUTO\_INCREMENT** 

### 特点

<mark>添加位置</mark>:只能添加到键列(主键,唯一),普通列不可以。

增加自增长约束的列必须是整数类型。

特殊情况:如果给自增长字段设置O或者null,列数据会自增长赋值,如果设置的是非零和非空数据,那么将真实设置值!



## 3.3 自增长约束添加

### 建表时添加

```
create table 表名称(
字段名 数据类型 primary key auto_increment,
);
create table 表名称(
字段名 数据类型 unique key auto_increment,
);
```

### 建表后修改

alter table 表名称 modify 字段名 数据类 auto\_increment;

### 删除

#alter table 表名称 modify 字段名 数据类型 auto\_increment;
#给这个字段增加自增约束
alter table 表名称 modify 字段名 数据类型;
#去掉auto\_increment相当于删除





- 1 约束概述和分类
- 2 域 (列) 级约束
- 3 实体 (表) 级约束
- 4 参照引用 (外键) 约束
- 5 约束常见面试题



### U)尚硅谷 www.atguigu.com

### 4.1 外键(参照|引用)约束

什么是外键和外键约束?

### 学生表

sid	姓名	年龄	性别
1	二狗子	18	男
2	驴蛋蛋	18	男

### 分数表

cid	分数	学科	sid
1	88	Java	1
2	90	Java	2
3	60	MySQL	4

## 4.1 外键(参照|引用)约束

什么是外键和外键约束?

### 学生表

sid	姓名	年龄	性别
1	二狗子	18	男
2	驴蛋蛋	18	男



### 分数表

cid	分数	学科	sid
1	88	Java	1
2	90	Java	2
3	60	MySQL	4

1. 二狗子Java考了多少分!



### 4.1 外键(参照|引用)约束

什么是外键和外键约束?

 学生表

 sid
 姓名
 年龄
 性別

 1
 二狗子
 18
 男

 2
 驴蛋蛋
 18
 男

 分数表

 cid
 分数
 学科
 sid

 1
 88
 Java
 1

 2
 90
 Java
 2

 3
 60
 MySQL
 4

1. 二狗子Java考了多少分

参照 | 引用



### 4.1 外键(参照|引用)约束

什么是外键和外键约束?



1. 二狗子Java考了多少

学科

Java

Java

MySQL

2

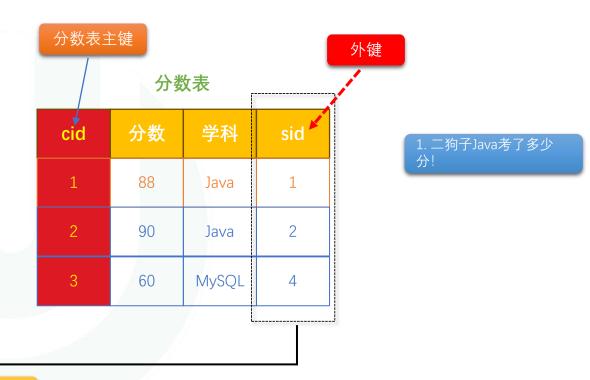
4



### 4.1 外键(参照|引用)约束

什么是外键和外键约束? 学生表主键 学生表主键

sid	姓名	年龄	性别
1	二狗子	18	男
2	驴蛋蛋	18	男
<u></u>			



参照 | 引用

<mark>外键:</mark>引用或参照其他表主键列值的列,我们称为外键,外键的值范围应当对应引用主键的值范围!!



### 4.1 外键(参照|引用)约束



外键: 引用或参照其他表主键列值的列,我们称为外键,外键的值范围应当对应引用主键的值范围!!

外键约束:外键应该引用主键的值,但是如果不添加约束,可能会出现错误数据,例如:MySQL数据,<mark>外键约束</mark>确保,外键必须且正确引用主键的值的限制!



### 4.1 外键约束

#### 外键细节说明

外键数量:每个表中可以包含多个外键。

<mark>外键跨表</mark>:外键是跨表引用其他表的主键,被引用为主表(学生),外键表为子表(分数)。

<mark>外键类型:外键类型不能是任意类型</mark>,应该和主键类型对应,尽量命名相同!

<mark>主外键关系</mark>:关系型数据库,关系指的就是主外键关系,有主外键的两张表能水平联查。

<mark>其他影响</mark>:存在主外键关系(外键约束),删除主表数据时,可能会因为子表引用而删除失败。可以先删除子表的所有引用数据再删除!

### 4.1 外键约束

#### 添加

### 建表时添加

```
create table 主表名称(
字段1 数据类型 primary key
);

# 子表中添加主外键约束
create table 子表名称(
字段1 数据类型 primary key,
[CONSTRAINT <外键约束名称>] FOREIGN KEY (外键)
references 主表名(主键) [on update xx][on delete xx]
);
```

### 建表后修改

alter table 从表名 add [CONSTRAINT 约束名] FOREIGN KEY (从表的字段) references 主表名(被引用字段) [on update xx][on delete xx];



#### 删除

(1)第一步先查看约束名和删除外键约束

SELECT \* FROM information\_schema.table\_constraints WHERE table\_name = '表名称';#查看某个表的约束名

ALTER TABLE 从表名 DROP FOREIGN KEY 外键约束名;

(2) 第二步查看索引名和删除索引。(注意,只能手动删除) SHOW INDEX FROM 表名称; #查看某个表的索引名 ALTER TABLE 从表名 DROP INDEX 索引名;



### 4.1 外键约束

### 约束等级设计

Cascade	在父表上update/delete记录时,同步update/delete子表的匹配记录!			
Set null	在父表上update/delete记录时,将子表上匹配记录的列设为null,但是要注意子表的外键列不能为not null!			
No action	如果子表中有匹配的记录,则不允许对父表对应候选键进行update/delete操作!			
Restrict【默认】	同no action,都是立即检查外键约束!			
Set default	父表有变更时,子表将外键列设置成一个默认的值,但Innodb不能识别!			

最好是采用: ON UPDATE CASCADE ON DELETE RESTRICT 的方式。

示例:当主表中的行被删除或更新时,从表中的相关行也会被相应地删除或更新。

ALTER TABLE 从表名 ADD CONSTRAINT 外键约束名

FOREIGN KEY (从表外键列) REFERENCES 主表名(主表主键列)

ON DELETE CASCADE

ON UPDATE CASCADE;





- 1 约束概述和分类
- 2 域 (列) 级约束
- 3 实体 (表) 级约束
- 4 参照引用 (外键) 约束
- 5 约束常见面试题



## 5 约束常见面试题

### 5 约束常见面试题



### ① 建和不建外键约束有什么区别?

答: 建外键约束, 你的操作(创建表、删除表、添加、修改、删除)会受到限制, 从语法层面受到限制。例如在员工表中不可能添加一个员工信息, 它的部门的值在部门表中找不到。

不建外键约束,你的操作(创建表、删除表、添加、修改、删除)不受限制,要保证数据的引用完整性,只能依 靠程序员的自觉,或者是在Java程序中进行限定。例如:在员工表中,可以添加一个员工的信息,它的部门指定为 一个完全不存在的部门。

### ② 建和不建外键约束对查询有什么影响?

答: 没有

添加约束可能影响查询速度和效率!

### ③ 表中字段为什么不想要 null 的值?

答: (1) 不好比较。null是一种特殊值,比较时只能用专门的is null 和 is not null来比较。碰到运算符,通常返回null。

(2) 效率不高。影响提高索引效果。因此,我们往往在建表时 not null default " 或 default O

### 5 约束常见面试题



### ④ 带AUTO\_INCREMENT约束的字段值是从1开始的吗?

答:在MySQL中,默认AUTO\_INCREMENT的初始值是1,每新增一条记录,字段值自动加1。设置自增属性(AUTO\_INCREMENT)的时候,还可以指定第一条插入记录的自增字段的值,这样新插入的记录的自增字段值从初始值开始递增,如在表中插入第一条记录,同时指定id值为5,则以后插入的记录的id值就会从6开始往上增加。添加主键约束时,往往需要设置字段自动增加属性。

除此之外,可以在创建表的时候,指定自增长起始值。

### ⑤ 是不是每个表都可以任意选择存储引擎?

答:外键约束 (FOREIGN KEY) 不能跨引擎使用。

MySQL支持多种存储引擎,每一个表都可以指定一个不同的存储引擎,需要注意的是:外键约束是用来保证数据的参照完整性的,如果表之间需要关联外键,却指定了不同的存储引擎,那么这些表之间是不能创建外键约束的。所以说,存储引擎的选择也不完全是随意的。

### ⑥ 请问你会不会创建数据库时给你的表添加完备的约束呢?

答: 不会, 一般情况下, 我们只会添加一些单表的约束(实体约束和域约束)!

不会添加外键和级联操作!

根据阿里开发规范,【强制】不得使用外键与级联,一切外键概念必须在应用层解决。

(概念解释) 学生表中的 student\_id 是主键,那么成绩表中的 student\_id 则为外键。如果更新学生表中的 student\_id, 同时触发成绩表中的 student\_id 更新,即为级联更新。外键与级联更新适用于单机低并发,不适合分布式、高并发集群;级联更新是强阻塞,存在数据库更新风暴的风险;外键影响数据库的插入速度。

# 感谢观看

