



重构IT职业教育新生态

MySql

● 数据完整性与DML语句

目录

C O N T E N T S

1

数据完整性及约束

2

数据库设计及范式

3

DML语句

- 一．理解数据完整性的含义
- 二．掌握实现各种数据完整性类型的方法
- 三．掌握创建各种约束的语法
- 四．数据库中的三大范式
- 五．熟练掌握DML语句

可靠性



准确性



数据完整性

- 数据存放在表中
- 数据完整性的问题大多是由于设计引起的
- 创建表的时候，就应当保证以后数据输入是正确的
- 错误的数据、不符合要求的数据不允许输入

完整性包括...

- 输入的类型是否正确？
 - ◆ ——年龄必须是数字
- 输入的格式是否正确？
 - ◆ ——身份证号码必须是18位
- 是否在允许的范围内？
 - ◆ ——性别只能是“男”或者“女”
- 是否存在重复输入？
 - ◆ ——学员信息输入了两次
- 是否符合其他特定要求？
 - ◆ ——信誉值大于5的用户才能够加入会员列表
-

列值要求（约束）

整行要求（约束）

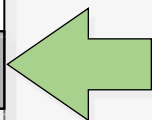
完整性包括...

域完整性



Blue	Green	Blue
White	Gray	White
Gray	Gray	Gray
White	Gray	White

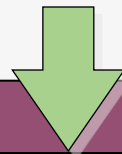
实体完整性



参照完整性

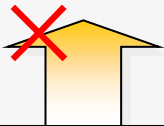


自定义完整性



Blue	Maroon	Blue
White	Blue	White
White	White	White

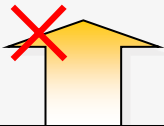
学号	姓名	地址
0010012	李山	山东定陶	
0010013	吴兰	湖南新田	
0010014	雷铜	江西南昌	
0010015	张丽鹃	河南新乡	
0010016	赵可以	河南新乡	



0010014	雷铜	江西南昌
---------	----	------

约束方法：唯一约束、主键约束、自增列

学号	姓名	地址
0010012	李山	山东定陶	
0010013	吴兰	湖南新田	
0010014	雷铜	江西南昌	
0010015	张丽鹃	河南新乡	
0010016	赵可以	河南新乡	



6666666666	李四	江西南昌	
------------	----	------	--

6

约束方法：限制数据类型、外键约束、默认值、非空约束

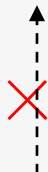
参照完整性

学号	姓名	地址	...
0010012	李山	山东定陶	
0010013	吴兰	湖南新田	
0010014	雷铜	江西南昌	
0010015	张丽鹃	河南新乡	
0010016	赵可以	河南新乡	

科目	学号	分数	...
数学	0010012	88	
数学	0010013	74	
语文	0010012	67	
语文	0010013	81	
数学	0010016	98	

✗	🏠	数学	0010021	98	
---	---	----	---------	----	--

用户编号	用户姓名	会员证
AV0012	孙悟空	AV378289	
AV0013	猪悟能	AV378290	
AV0014	沙悟净	AV378291	
CV0015	玄痛	AV372133	
CV0016	乔峰	AV121322	



触发器：检查信用值

帐号	姓名	信用
00192	孙悟空	7	
00288	猪悟能	6	
12333	段誉	8	
90111	虚竹	40	
93000	岳不群	-10	

约束方法：规则、存储过程、触发器

- 表中有一列或几列组合的值能用来唯一地标识表中的每一行，这样的一列或者多列的组合叫做表的主键。
 - ◆ 如：学号可以作为学生表的主键，课程号可以作为课程表的主键，（学号，课程号）作为成绩表的主键（组合键）
- 说明：
 - ◆ （1）一个表只能有一个主键，主键约束确保了表中的行是唯一的。
 - ◆ （2）表中可以没有主键，但是通常情况下应当为表设置一个主键。
- 选择主键的原则：
 - ◆ 最少性：尽量选择单个键作为主键
 - ◆ 稳定性：尽量选择数值更新少的列作为主键

ID	用户姓名	会员证
AV0012	孙悟空	AV378289	
AV0013	猪悟能	AV378290	
AV0014	沙悟净	AV378291	
CV0015	唐僧	AV372133	
CV0016	白龙马	AV121322	



如何给ID字段创建主键？

1. 直接在字段定义后面声明主键

```
CREATE TABLE demo(  
    ID INT PRIMARY KEY,  
    .....  
)
```

2. 用constraint声明主键

```
CREATE TABLE demo(  
    ID INT,  
    .....,  
    -- 外键命名规范 : pk_表名,  
    CONSTRAINT pk_demo PRIMARY KEY (ID))
```

也可以简写成 :

```
CREATE TABLE demo(  
    ID INT,  
    NAME VARCHAR(10),  
    .....,  
    PRIMARY KEY (ID, NAME)) -- 复合主键
```

3. 用ALTER语句补充声明主键

```
ALTER TABLE demo ADD CONSTRAINT pk_demo PRIMARY KEY (sid);
```

4. 删除主键

```
ALTER TABLE demo DROP PRIMARY KEY;
```

创建自增长

ID	用户姓名	会员证
12	孙悟空	AV378289	
13	猪悟能	AV378290	
14	沙悟净	AV378291	
15	唐僧	AV372133	
16	白龙马	AV121322	



示例

给ID字段创建主键

字段和主键同时定义并设定自增长

```
CREATE TABLE demo(  
    -- 设定为自增的列必须设置为主键，并且一张表里只能有一个自增列  
    ID INT AUTO_INCREMENT PRIMARY KEY ,  
    name VARCHAR(10)  
)  
INSERT INTO demo VALUES (3, 'tester1');  
INSERT INTO demo (name) VALUES ('tester2');
```

创建唯一约束

ID	name	vip
AV0012	孙悟空	AV378289	
AV0013	猪悟能	AV378290	
AV0014	沙悟净	AV378291	
CV0015	玄痛	AV372133	
CV0016	乔峰	AV121322	

试图为该字段插入重复的数据时会提示错误信息



给VIP字段创建唯一约束

定义vip字段唯一

```
CREATE TABLE demo(  
    ....  
    vip VARCHAR(50) UNIQUE,  
    .....  
)
```

创建非空约束

ID	name	会员证
AV0012	孙悟空	AV278289	
AV0013	猪悟能		
AV0014	沙悟净		
CV0015	玄痛	AV372133	
CV0016	乔峰	AV121322	

试图为该字段插入空值时
数据库会提示错误信息



给name字段创建非空约束

定义name字段非空

```
CREATE TABLE demo(  
    ....
```

```
    ....
```

```
    name VARCHAR(50) NOT NULL,
```

```
    .....
```

```
)
```


ID	name	性别
AV0012	孙悟空	男	
AV0013	白骨精	女	
AV0014	沙悟净	男	
CV0015	西施	女	
CV0016	乔峰	男	

试图为该字段插入除“男”、“女”外的值会报错



给性别字段创建检查约束

```
CREATE TABLE demo(  
    ....  
    sex SET('男', '女'),  
    payment ENUM('货到付款', '在线支付')  
    .....  
)
```

Mysql原生并不支持检查约束。对于离散值可使用set或enum方式解决。对于连续值只能通过触发器解决。

创建默认值

ID	name	vip	tel
AV0012	孙悟空	AV378289	default
AV0013	猪悟能	AV378290	
AV0014	沙悟净	AV378291	
CV0015	玄痛	AV372133	default
CV0016	乔峰	AV121322	default

试图为该字段插入重复的数据时会自动填充



给name字段创建非空约束

定义tel字段默认值

```
CREATE TABLE demo(  
  ....
```

```
.....
```

```
tel VARCHAR(50) DEFAULT 'default',
```

```
.....
```

```
)
```

- 简单地说，就是“子表”中对应于“主表”的列，在子表中称为外键或者引用键。
- 它的值要求与主表的主键或者唯一键相对应，外键用来强制引用完整性。例如在成绩表中，学号为外键。一个表可以有多个外键。

创建外键约束



snum	姓名	地址	主表
0010012	李山	山东定陶	
0010013	吴兰	湖南新田	
0010014	雷倩	江西南昌	
0010015	张丽鹃	河南新乡	
0010016	赵可以	河南新乡	

snum INT PRIMARY KEY

科目	snum	分数	从表
数学	0010012	88	
数学	0010013	74	
语文	0010012	67	
语文	0010013	81	
数学	0010016	98	

num INT ,
FOREIGN KEY (snum) REFERENCES 主表(snum)

数学	0010021	98	
----	---------	----	--

```
CREATE TABLE IF NOT EXISTS demo_new(  
    cid INT PRIMARY KEY,  
    sid INT,  
    FOREIGN KEY (sid) REFERENCES demo(sid) )
```

```
ALTER TABLE demo_new ADD CONSTRAINT fk_1 FOREIGN KEY ( sid )  
REFERENCES demo ( sid )  
ON DELETE CASCADE ON UPDATE CASCADE
```



如何指定外键在update或删除主表时的行为？

- 如果子表试图创建一个在父表中不存在的外键值，InnoDB会拒绝任何INSERT或UPDATE操作。
- 如果父表试图UPDATE或者DELETE任何子表中存在或匹配的外键值，最终动作取决于外键约束定义中的ON UPDATE和ON DELETE选项。InnoDB支持5种不同的动作，如果没有指定ON DELETE或者ON UPDATE，默认的动作作为RESTRICT。
- **CASCADE:** 从父表中删除或更新对应的行，同时自动的删除或更新自表中匹配的行。ON DELETE CASCADE和ON UPDATE CASCADE都被InnoDB所支持。
- **SET NULL:** 从父表中删除或更新对应的行，同时将子表中的外键列设为空。注意，这些在外键列没有被设为NOT NULL时才有效。ON DELETE SET NULL和ON UPDATE SET SET NULL都被InnoDB所支持。
- **NO ACTION:** InnoDB拒绝删除或者更新父表。
- **RESTRICT:** 拒绝删除或者更新父表。指定RESTRICT (或者NO ACTION) 和忽略ON DELETE或者ON UPDATE选项的效果是一样的。

指定外键更新或删除的行为

```
CREATE TABLE IF NOT EXISTS demo_new(  
    cid INT PRIMARY KEY,  
    sid INT,  
    FOREIGN KEY (sid) REFERENCES demo(sid) ON UPDATE CASCADE  
)
```

```
CREATE TABLE IF NOT EXISTS demo_new(  
    cid INT PRIMARY KEY,  
    sid INT,  
    FOREIGN KEY (sid) REFERENCES demo(sid) ON DELETE CASCADE ON UPDATE CASCADE  
)
```

```
CREATE TABLE IF NOT EXISTS demo_new(  
    cid INT PRIMARY KEY,  
    sid INT,  
    FOREIGN KEY (sid) REFERENCES demo(sid) ON UPDATE CASCADE ON DELETE RESTRICT  
)
```

代码改错

默认的不可能是唯一的

```
Name VARCHAR(50) UNIQUE DEFAULT '123456'
```

外键不能像主键这样定义

```
Uid INT FOREIGN KEY
```

少了连接关键字

```
Uid INT ,  
FOREIGN KEY(uid) 主表(uid)
```


- 为了建立冗余较小、结构合理的数据库，设计数据库时必须遵循一定的规则。在关系型数据库中这种规则就称为范式。范式是符合某一种设计要求的总结。要想设计一个结构合理的关系型数据库，必须满足一定的范式。
- 第一范式:确保每列保持原子性（字段不可再分）
 - ◆ 第一范式是最基本的范式。如果数据库表中的所有字段值都是不可分解的原子值，就说明该数据库表满足了第一范式。

用户信息表

编号	姓名	性别	年龄	联系电话	省份	城市	详细地址
1	张红欣	男	26	0378-23459876	河南	开封	朝阳区新华路23号
2	李四平	女	32	0751-65432584	广州	广东	白云区天明路148号
3	刘志国	男	21	0371-87659852	河南	郑州	二七区大学路198号
4	郭小明	女	27	0371-62556789	河南	郑州	新郑市薛店北街218号

- 第二范式:每个实例或行必须可以被惟一地区分 (表不可再分)
 - ◆ 第二范式 (2NF) 是在第一范式 (1NF) 的基础上建立起来的,即满足第二范式 (2NF) 必须先满足第一范式 (1NF)。第二范式 (2NF) 要求数据库表中的每个实例或行必须可以被惟一地区分。**即要求每个实体都必须具有主键进行区分。**
 - ◆ 要求实体的属性完全依赖于主关键字,不是依赖部分关键字

订单编号	商品编号	商品名称	数量	单位	价格	客户	所属单位	联系方式
001	1	挖掘机	1	台	1200000¥	张三	上海玖智	020-1234567
001	2	冲击钻	8	把	230¥	张三	上海玖智	020-1234567
002	3	铲车	2	辆	980000¥	李四	北京公司	010-1234567

- 这样就产生一个问题:这个表中是以订单编号和商品编号作为联合主键。这样在该表中商品名称、单位、商品价格等信息不与该表的主键相关,而仅仅是与商品编号相关。所以在这里违反了第二范式的设计原则。

第二范式实例

订单信息表

订单编号	客户	所属单位	联系方式
001	张三	上海玖智	020-1234567
002	李四	北京公司	010-1234567

订单项目表

订单编号	商品编号	数量
001	1	1
001	2	8
002	3	2

商品信息表

商品编号	商品名称	单位	商品价格
1	挖掘机	台	1200000¥
2	冲击钻	个	230¥
3	铲车	辆	980000¥

- 第三范式：确保每列都和主键列直接相关,而不是间接相关
 - ◆ 第三范式需要确保数据表除外键外，其他非主键属性不得重复出现在第二张表上。
 - ◆ 比如在设计一个订单数据表的时候，可以将客户编号作为一个外键和订单表建立相应的关系。而不可在订单表中添加关于客户其它信息（比如姓名、所属公司等）的字段。如下面这两个表所示的设计就是一个满足第三范式的数据库表。

订单信息表

订单编号	订单项目	负责人	业务员	订单数量	客户编号
001	挖掘机	刘明	李东明	1台	1
002	冲击钻	李刚	霍新峰	8个	2
003	铲车	郭新一	艾美丽	2辆	1

客户信息表

客户编号	客户名称	所属公司	联系方式
1	李聪	五一建设	13253661015
2	刘新明	个体经营	13285746958

```
CREATE TABLE Students(  
    SCode INT auto_increment PRIMARY KEY,  
    SName VARCHAR(10) NOT NULL,  
    SAddress VARCHAR(10) NOT NULL,  
    SGrade INT,  
    SEmail VARCHAR(20),  
    SSex TINYINT  
)
```

```
INSERT INTO <表名> ([列名]) VALUES (<值列表>)
```

```
INSERT INTO Students (SName, SAddress ,SGrade,SEmail,SSEX)  
VALUES ('张三','上海松江',6,'ZS@Sohu.com',0)
```

插入数据行

注意事项1:

每次插入一行数据，不能只插入半行或者几列数据
插入的数据是否有效将按照整行的完整性的要求来检验

代码错误

```
INSERT INTO Students (SName,SAddress,SGrade,SEmail,SSEX)
VALUES ('张三')
```

注意事项2:

每个数据值的数据类型、精度和小数位数必须与相应的列匹配

代码错误

```
INSERT INTO Students (SName,SAddress,SGrade,SEmail,SSEX)
VALUES ('张三','上海松江','ZS','ZQC@Sohu.com', 'NAN')
```

注意事项3:

如果列是自增列，一般不会手动为该列指定值

不推荐这样写

```
INSERT INTO Students  
(SCode,SName,SAddress,SGrade,SEmail,SSEX)  
VALUES (32,'张青裁','上海松江',6,'ZQC@Sohu.com',0)
```

注意事项4:

如果在设计表的时候就指定了某列不允许为空，则必须插入数据

代码错误

```
INSERT INTO Students (SAddress,SGrade,SEmail,SSEX)  
VALUES ('上海松江',6,'ZQC@Sohu.com',0)
```

注意事项5:

具有缺省值的列，可以使用DEFAULT（缺省）关键字来代替插入的数值

代码错误

```
INSERT INTO Students (SName,SAddress,SGrade,SEmail,SSEX)  
VALUES ('张青裁',DEFAULT,6,'ZQC@Sohu.com',0)
```


插入多行数据

```
INSERT INTO <表名>(列名)
SELECT <列名>
FROM <源表名>
```

```
INSERT INTO      TongXunLu (姓名,地址,电子邮件)
SELECT      SName,SAddress,SEmail
FROM        Students
```

```
INSERT INTO <表名> ([列名]) VALUES (<值列表>),( <值列表>),( <值列表>).....
```

```
INSERT INTO STUDENTS (SName,SGrade,SSex) VALUES
('张可',7,1),
('李扬',4,0),
('杨晓',2,0),
('王娟娟',7,0);
```

```
INSERT INTO <表名> ([列名]) VALUES (<值列表>),( <值列表>),( <值列表>).....
```

```
INSERT INTO STUDENTS (SName,SGrade,SSex) VALUES
```

```
('张可',7,1),
```

```
('李扬',4,0),
```

```
('杨晓',2,0),
```

```
('汤美',3,0),
```

```
('苏三东',7,1),
```

```
('王立岩',3,1),
```

```
('张伟',7,1),
```

```
('陈刚',4,1),
```

```
('王娟娟',7,0);
```

```
UPDATE <表名> SET <列名 = 更新值>  
[WHERE <更新条件>]
```

```
UPDATE Students SET SSEX = 0
```

```
UPDATE Students  
SET SAddress = '北京女子职业技术学校家政班'  
WHERE SAddress = '北京女子职业技术学校刺绣班'
```

```
UPDATE Scores  
SET Scores = Scores + 5  
WHERE Scores <= 95
```

删除数据

```
DELETE FROM <表名> [WHERE <删除条件>]
```

学号	姓名	地址
0010012	李山	山东定陶	
0010013	吴兰	湖南新田	
0010014	雷铜	江西南昌	
0010015	张丽鹃	河南新乡	
0010016	赵可以	河南新乡	

科目	学号	分数
数学	0010012	88	
数学	0010013	74	
语文	0010012	67	
语文	0010013	81	
数学	0010016	98	

```
TRUNCATE TABLE <表名>
```

```
TRUNCATE TABLE Students
```

=

功能类似

```
DELETE FROM Students
```

注意：

- 1.表结构、列、约束等不被改动
- 2.不能用于有外键约束的表
- 3.标识列重新开始编号



谢谢聆听

蜗牛学院，只为成就更好的你！

