

# MySQL 数据库中数据参照完整性和一致性的设计与实现

李艳杰\*

LI Yan-jie

## 摘要

MySQL 作为后台数据库,应用非常广泛,在使用时会设计多个表,并且多个表之间存在关联关系,对于有联系的表而言,解决多表之间数据的一致性和参照完整性就显得尤为重要,本文详细阐述了如何利用主键和外键使多个表建立联系,并且保证在操作的时候使多个表中的数据始终保持一致。

## 关键词

MySQL 数据库;主键;外键;一致性

doi: 10.3969/j.issn.1672-9528.2020.11.030

## 0 引言

在利用数据库设计数据表的时候,往往会设计多个表,并且多个表之间是有联系的,这个联系指的是一个表中的某个字段的值,和另外一个表中的字段的值存在某种意义上的联系,即一个表(子表)中字段的值依赖与另外一个表(主表)中的主键的值,这两个表通过字段建立关联关系,这种关联关系就是参照完整性约束,这种参照完整性约束可以实现多表之间数据的一致性。要实现数据的一致性,本文通过建立两个表,并在表中创建主键和外键的形式来实现。

### 1 主键的含义和分类

主键是指在表中可以唯一的标识表中的一条记录,并且主键的值不能重复,主键的值也不能为空,一个表中只能有一个主键存在;主键可以分为单字段主键和多字段主键两种,单字段主键由表中的一个字段组成,多字段主键由表中的两个或两个以上的字段组成。

### 2 主键的创建与删除

主键的创建有两种方式,一种是创建表的时候同时指定主键,例如:在数据库 k 下创建一个 person 表,包括编号 id 字段,姓名 name 字段,年龄 age 字段,信息 info 字段,并指定 id 字段为主键,代码为: `mysql>create table person(id int unsigned,name char(40),age int,info char(50),primary key(id));` 其中,primary key 关键字即为主键的关键字,后面加上括号,括号里声明设置为主键的字段。在软件中的结果如下图 1 所示。

```
mysql> create database k1 default charac
ter set gbk;
Query OK, 1 row affected (0.00 sec)

mysql> use k1;
Database changed
mysql> create table person(id int unsign
ed,name char(40),age int,info char(50),p
rimary key(id));
Query OK, 0 rows affected (0.07 sec)
```

图 1 建表同时创建主键图

另一种是表创建完成之后,再设置主键,例如:仍然是创建上面的 person 表,但是创建时并未指定主键,而是表完成之后创建主键,此时,必须删除刚刚创建完成的主键,然后重新创建主键,代码为: `mysql> alter table person drop primary key;` 删除主键的过程也是修改表结构的过程,句子中的 alter 为修改表的关键字,drop 为删除的关键字,删除时注意,因为一个表中只能有一个主键,所以删除时不需要指定主键字段,只需要说明删除哪个表中的主键即可。如下图 2 所示。

```
mysql> alter table person drop primary key;
Query OK, 0 rows affected (0.23 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

图 2 删除主键图

主键删除之后,便可以用第二种方式进行主键的创建,代码为: `mysql>alter table person add primary key(id);` 创建与删除均为修改表的结构,所以仍然使用 alter 关键字,table 是表关键字,add 是增加主键的关键字。创建完成之后,MySQL 软件会提示创建成功。如下图 3 所示。

```
mysql> alter table person add primary key(id);
Query OK, 0 rows affected (0.17 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

图 3 单独创建主键图

\* 山东华宇工学院 山东德州 253034

[基金项目] 本文为 2020 年校级教育教学改革研究项目“自建线上课程的探索与实践——以《MySQL 数据库开发与实现》为例(项目编号:2020ZD10)”阶段性成果

### 3 外键的含义

外键是参照完整性的体现,表与表之间往往存在一种“父子”关系,例如:字段 s 是一个表 A 的字段,但是 s 的值却依赖于 B 表中的主键的值,则称 B 表为主表, A 表为子表,通常将 s 设置为 A 表的外键,参照 B 表的主键字段而建立,并且 s 的数据类型必须和 B 表中的主键的数据类型一致, s 这个字段名称可以和主键相同,也可以不同,在大多数情况下,外键和主键的字段名称是相同的,因为很容易联想到主键和外键,建立外键之后,主表和子表就有了一定的关联关系,这种关联关系即参照完整性约束,这种约束便可以实现数据的参照完整性和一致性。注意:必须有了主表和主键,才会有子表和外键。当外键存在时,主表里的主键是不能被删除的。可以先删除外键,然后再删除主键。一个表中可以有多个外键。

### 4 创建外键的方式

创建外键的格式为: constraint 外键名 foreign key(外键字段名) references 主表名(主键名)。

constriaint 是创建约束的关键字,参照完整性约束是一中特殊的约束,也需要用此关键字去创建,外键名是约束的名称,由于一个表中可以存在多个外键,所以必须给外键声明一个名称以区别其他的外键名, foreign key 是外键的关键字,后跟外键的字段名称, references 是参照的关键字,因为外键参照主键而建立,所以此关键字不能缺少,后跟外键所参照的主表名称和主键名称。

由于创建外键必须依赖于主表,建立外键,首先需要创建主表,建立班级表作为主表,包括整型班级编号 bid 字段,字符长度为 20 的字符型班级名称 bname 字段,并指定 bid 为主键。代码为: mysql>create table banji(bid int,bname char(20), primary key(bid)); 建立的主表如下图 4 所示。

```
mysql> create table banji(bid int,bname
char(20), primary key(bid));
Query OK, 0 rows affected (0.06 sec)
```

图 4 带有主键的主表图

主表建立完成,然后创建子表,创建学生表,包括学号 xh 字段,姓名 xm 字段,以及学生所在班级 bid 字段,注意,主表和子表中的 bid 字段的数据类型必须相同,否则外键无法成功建立,创建子表代码为: mysql>create table xuesheng(xh int,xm char(6),bid int,constraint waijian1 foreign key(bid) references banji(bid)); 建立的子表如下图 5 所示。

```
mysql> create table xuesheng(xh int,xm c
har(6),bid int,constraint waijian1 forei
gn key(bid) references banji(bid));
Query OK, 0 rows affected (0.07 sec)
```

图 5 带有外键的子表图

### 5 参照完整性的规则及实现过程

通过上面创建的主表和子表,并且为主表创建了主键,为子表创建了外键,此时,两个表便具有了参照完整性约束,约束主要有三个规则,即如果子表中存在匹配记录,则不能从主表中删除记录;如果子表中存在匹配记录,则不能从主表中修改主键的值;主表中没有的值,子表中也不能有。那针对这三个规则,在主键和外键的约束下是如何实现的呢,根据上面建立的主表和子表进行详细说明。

下面为主表和子表输入记录,对约束的三个规则进行说明和验证;班级表中输入记录代码为: insert into banji values(201,'soft1'), (202,'soft2'), (203,'soft3'); 通过 select 语句进行查看,得到的结果如下图 6 所示:

```
mysql> select * from banji;
+----+-----+
| bid | bname |
+----+-----+
| 201 | soft1 |
| 202 | soft2 |
| 203 | soft3 |
+----+-----+
3 rows in set (0.00 sec)
```

图 6 主表记录图

在学生表中输入记录代码为: insert into xuesheng values(201801,'zhaoyu',201), (201802,'lili',201), (201810,'wanghong',202), (201811,'wanglin',202); 通过 select 语句进行查看,得到的结果如下图 7 所示:

```
mysql> select * from xuesheng;
+----+-----+-----+
| xh  | xm    | bid  |
+----+-----+-----+
| 201801 | zhaoyu | 201 |
| 201802 | lili   | 201 |
| 201810 | wangho | 202 |
| 201811 | wangli | 202 |
+----+-----+-----+
4 rows in set (0.00 sec)
```

图 7 子表记录图

验证规则 1: 如果子表中存在匹配记录,则不能从主表中删除记录;此时如果删除班级表中的 bid 为 201 的记录,此项操作不能完成,因为子表中存在两条 bid 为 201 的相关的记录,当执行 delete from banji where bid=201 操作语句时,会提示错误信息: ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key constraint fails (`a`.`xuesheng`, CONSTRAINT

`waijian1` FOREIGN KEY (`bid`) REFERENCES `banji` (`bid`)), 此错误信息含义为, 不能删除或更新主表中的一行数据, 一个外键约束存在, 正是因为此约束的存在, 才保持了数据的一致性, 因为这条记录如果删除成功, 即主表中的 bid 为 201 的班级不存在了, 但是子表此班级还有学生存在, 这就违反了现实生活中的规律, 两个表中的数据不再保持一致。如下图 8 所示。

```
mysql> delete from banji where bid=201;
ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key constraint fails (`k1`.`xuesheng`, CONSTRAINT `waijian1` FOREIGN KEY (`bid`) REFERENCES `banji` (`bid`))
```

图 8 验证规则 1 图

如果想从主表中删除记录, 则必须先把子表中与之匹配的记录删除掉, 才能删除主表中的记录, 这样仍然保证数据的一致性, 即匹配的记录或者同时存在, 或者同时消失, 从而保持数据的一致性原则。

验证规则 2: 如果子表中存在匹配记录, 则不能从主表中修改主键的值; 此时如果修改班级表中的 bid 为 201 的记录, 改成 bid 为 205, 此项操作不能完成, 因为子表中存在两条 bid 为 201 的相关的记录, 当执行更新语句 update banji set bid=205 where bid=201 时, 也会提示错误信息: ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key constraint fails (`a`.`xuesheng`, CONSTRAINT `waijian1` FOREIGN KEY (`bid`) REFERENCES `banji` (`bid`)), 此错误信息提示和验证规则 1 中的错误信息一致, 因为这条记录如果更新成功, 即主表中的 bid 为 201 的班级不存在了, 但是子表此班级还有学生存在, 同样违反了现实生活中的规律, 两个表中的数据不再保持一致。如下图 9 所示。

```
mysql> update banji set bid=205 where bid=201;
ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key constraint fails (`k1`.`xuesheng`, CONSTRAINT `waijian1` FOREIGN KEY (`bid`) REFERENCES `banji` (`bid`))
```

图 9 验证规则 2 图

验证规则 3: 主表中没有的值, 子表中也不能有, 此时如果在子表中插入一条 bid 为 205 的记录, 代码为: insert into xuesheng values(201816,'wanglin',205); 词条语句

不能执行成功, 错误提示信息如下: ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails (`a`.`xuesheng`, CONSTRAINT `waijian1` FOREIGN KEY (`bid`) REFERENCES `banji` (`bid`)), 其含义为不能添加或者更新子表中的行, 因为外键约束存在, 假如更新成功, 主表中没有 205 这个班级, 但是班级中却有学生存在, 这也是不符合规律的, 所以外键不允许这样, 从而保证数据的参照完整性和一致性。如下图 10 所示。

```
mysql> insert into xuesheng values(201816,'wanglin',205);
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails (`k1`.`xuesheng`, CONSTRAINT `waijian1` FOREIGN KEY (`bid`) REFERENCES `banji` (`bid`))
```

图 10 验证规则 3 图

建立主键和外键的注意事项: 1. 一个表中主键只能有一个, 外键却可以有多个。2. 主键的值必须唯一并且不能为空, 外键的值则可以重复。

## 6 结语

从以上的验证可以得出, 外键对于保持数据的参照完整性和一致性起到关键作用, 在 MySQL 数据库中, 除了外键可以起到此作用, 在表结构之外建立触发器也可以保证多表中数据的一致性。

## 参考文献:

- [1] 叶文.NoSQL 数据库与缓存一致性研究[J].信息与电脑(理论版), 2018(21):143-144.
- [2] 朱涛,郭进伟,周欢,等.分布式数据库中一致性与可用性的关系[J].软件学报, 2018,29(1):131-149.
- [3] 张亮,周晶,孙喜民,等.分布式数据库跨数据中心同步数据研究与设计,2019 电力行业信息化年会论文集[C].人民邮电出版社电信科学编辑部,2019(3):111-113.
- [4] 罗世刚,冯丽丽,耿兰,等.基于一致性算法的数据库双活架构应用[J].集成电路应用,2020,37(2):60-61.

## 【作者简介】

李艳杰(1978—),女,汉族,山东德州人,副教授,硕士,主要研究领域:数据挖掘技术。

(收稿日期: 2020-09-21 修回日期: 2020-10-14)