

# SQL FOREIGN KEY 约束

## SQL FOREIGN KEY 约束

一个表中的 FOREIGN KEY 指向另一个表中的 UNIQUE KEY(唯一约束的键)。

让我们通过一个实例来解释外键。请看下面两个表：

"Persons" 表：

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger

"Orders" 表：

O_Id	OrderNo	P_Id
1	77895	3
2	44678	3
3	22456	2
4	24562	1

请注意，"Orders" 表中的 "P\_Id" 列指向 "Persons" 表中的 "P\_Id" 列。

"Persons" 表中的 "P\_Id" 列是 "Persons" 表中的 PRIMARY KEY。

"Orders" 表中的 "P\_Id" 列是 "Orders" 表中的 FOREIGN KEY。

FOREIGN KEY 约束用于预防破坏表之间连接的行为。

FOREIGN KEY 约束也能防止非法数据插入外键列，因为它必须是它指向的那个表中的值之一。

## CREATE TABLE 时的 SQL FOREIGN KEY 约束

下面的 SQL 在 "Orders" 表创建时在 "P\_Id" 列上创建 FOREIGN KEY 约束：

MySQL：

```
CREATE TABLE Orders
(
  O_Id int NOT NULL,
  OrderNo int NOT NULL,
  P_Id int,
  PRIMARY KEY (O_Id),
  FOREIGN KEY (P_Id) REFERENCES Persons (P_Id)
)
```

**SQL Server / Oracle / MS Access :**

```
CREATE TABLE Orders
(
O_Id int NOT NULL PRIMARY KEY,
OrderNo int NOT NULL,
P_Id int FOREIGN KEY REFERENCES Persons(P_Id)
)
```

如需命名 FOREIGN KEY 约束，并定义多个列的 FOREIGN KEY 约束，请使用下面的 SQL 语法：

**MySQL / SQL Server / Oracle / MS Access :**

```
CREATE TABLE Orders
(
O_Id int NOT NULL,
OrderNo int NOT NULL,
P_Id int,
PRIMARY KEY (O_Id),
CONSTRAINT fk_PerOrders FOREIGN KEY (P_Id)
REFERENCES Persons(P_Id)
)
```

## ALTER TABLE 时的 SQL FOREIGN KEY 约束

当 "Orders" 表已被创建时，如需在 "P\_Id" 列创建 FOREIGN KEY 约束，请使用下面的 SQL：

**MySQL / SQL Server / Oracle / MS Access :**

```
ALTER TABLE Orders
ADD FOREIGN KEY (P_Id)
REFERENCES Persons(P_Id)
```

如需命名 FOREIGN KEY 约束，并定义多个列的 FOREIGN KEY 约束，请使用下面的 SQL 语法：

**MySQL / SQL Server / Oracle / MS Access :**

```
ALTER TABLE Orders
ADD CONSTRAINT fk_PerOrders
FOREIGN KEY (P_Id)
REFERENCES Persons(P_Id)
```

## 撤销 FOREIGN KEY 约束

如需撤销 FOREIGN KEY 约束，请使用下面的 SQL：

**MySQL :**

```
ALTER TABLE Orders
DROP FOREIGN KEY fk_PerOrders
```

**SQL Server / Oracle / MS Access :**

```
ALTER TABLE Orders
DROP CONSTRAINT fk_PerOrders
```

[← SQL PRIMARY KEY 约束](#)[SQL CHECK 约束 →](#)



## 2 篇笔记

## 写笔记



### 创建外键约束的方式

在创建表的时候指定外键约束

```
CREATE TABLE 表名
(
    column1 datatype null/not null,
    column2 datatype null/not null,
    ...
    CONSTRAINT 外键约束名 FOREIGN KEY (column1,column2,... column_n)
    REFERENCES 外键依赖的表 (column1,column2,...column_n)
    ON DELETE CASCADE--级联删除
);
```

在创建表后增加外键约束

```
ALTER TABLE 表名
ADD CONSTRAINT 外键约束名
FOREIGN KEY (column1, column2,...column_n)
REFERENCES 外键所依赖的表 (column1,column2,...column_n)
ON DELETE CASCADE;--级联删除
```

使用工具plsql来新增外键约束

注意，在创建外键约束时，必须先创建外键约束所依赖的表，并且该列为该表的主键

小胖纸 7个月前 (09-01)



Oracle数据库中，对指定外键的表进行增删改的情况：

子表：谁创建外键谁就是子表

父表：这个外键所依赖的表

```
create table father(
f_no number(2));--插入数据1 2 3 4
create table son(
s_no number(2),
f_no number(2))--插入两列1 2 3 4
--删除父表数据
delete from father where f_no=1 --报错违反约束
--删除父表
drop table father --报错违反约束
--更新父表数据
--禁用约束可以更新ALTER TABLE son disable constraint f_f;
update father set f_no =7 where f_no=1--报错违反约束
--对父表插入数据
insert into father values(6)--可以插入
--删除子表数据
```

```
delete from son where s_on=1--可以删除
--删除子表
drop table son--可以删除
--更新子表数据
update son set s_no=6 where s_no=1 --可以更新
update son set f_no=6 where s_no=6 --可以更新
--对子表插入数据
insert into son values(7,7)--父表没有7，违反约束
insert into son values(1,1)--父表有1可以插入
insert into son values(8,1)--父表有1可以插入
```

总结：

### 一、删除时，未指定cascade时

#### 1) 删除父表/数据

- a. 因为子表与父表一一对应，删除父表数据时，需要先把子表对应数据删除否则无法删除
- b. 同理，删除表的时候，也需要先删除子表再删除父表

解决方案：

#### a. 指定cascade，删除父表、数据

CASCADE指当删除主表中被引用列的数据时，级联删除子表中相应的数据行。

#### b. 禁用约束（子表的外键约束）

ALTER TABLE 表名 disable constraint 约束名;

#### 2) 删除子表：可以删除子表或者数据不报错

### 二、更新时

- a. 更新父表会违反约束
- b. 可以更新子表
- c. 没有针对约束的级联更新

### 三、插入时

- a. 父表可以插入
- b. 子表插入会违反约束

小胖纸 7个月前 (09-01)