# 外键约束（FOREIGN KEY）

## 认识外键约束

在实际开发项目中，一个健壮数据库中的数据一定有很好的参照完整性。例如有学生档案和成绩单两张表，如果成绩单中有张三的成绩，学生档案中张三的档案却被删除了，这样就会产生垃圾数据或者错误数据。为了保证数据的完整性，将两表之间的数据建立关系，因此就需要在成绩表中添加外键约束。

1. 外键约束指的是表中的某字段的值是引用了其他表中某字段的值。被引用的字段应 该具有主键约束或唯一约束，引用其他其他字段的字段应该添加外键约束；
2. 被引用的表称为主表，引用了其他表的字段的表称为从表。
3. 引入外键后，插入数据时外键约束字段的值只能是主表中被引用字段的值，主 表中被引用字段的值不能被删除和修改，除非其值没有被从表所引用。

## 添加外键约束

为字段添加外键约束后该字段的值只能是被引用字段的值，语法为：  
 **ALTER TABLE 表名 ADD CONSTRAINT 外键名 FOREIGN KEY(字段名) REFERENCES 主表表名(字段名) ;**

1）在主表中被引用的字段必须拥有主键约束或者唯一约束。

2）在从表中引用了主表的字段和主表中被引用的字段两者的类型应相同。

3）只有InnoDB型存储引擎的表才支持创建外键约束。

4）添加外键后，会在表中自动创建外键约束字段的普通索引，索引名与外键名相同。

成功添加外键后，可以通过查看表的创建结构的方式来验证是否存在外键约束。

注意：主表被引用的字段必须拥有唯一约束，或者是单字段主键约束（多字段主键无效），

## 删除外键约束

当需要解除两个表之间的关联关系时，就需要删除外键约束。其语法格式为：

**ALTER TABLE 表名 DROP FOREIGN KEY 外键名 ;**

## 表的关联关系

表之表之间的关联关系是通过外键建立的。MySQL中数据表的关联关系有三种。

1. 多对一：比如员工与部门之间的关系，一个员工只能属于一个部门，而一个部门可 以有多个员工。在多对一的表关系中，应该将外键约束创建在多的一方。
2. 多对多：比如学生与课程之间的关系，一个学生可以选择多门课程，一门课程也可 供多个学生选择。实现多对多关系需要定义一张中间表，称为连接表，该 表中需存在两个外键，分别引用课程表和学生表。
3. 一对一：比如身份证和人的关系，一人只能有一个身份证，一个身份证只能匹配一 个人。实现一对一关系，需要分清主从关系，从表应因主表存在而有意义，

分清主从关系后，在从表中建立外键约束。一对一关系在数据库中并不常 见，因为以这种方式存储的信息通常会放在一个表中。

## 在从表中添加数据

在从表中添加数据时，外键约束字段的值必须是其引用字段的所有值之一。

## 主表的删除与修改

1）只有当从表未引用主表中被引用字段的任何值时，主表才可以被删除。

2）只有当主表中被引用字段的某值未被从表引用时，才可以修改和删除该值。

# 连接查询

## 认识连接查询

1）在关系型数据库中，建立表时各个数据之间的关系可以不必确定，通常将每个实体 的所有信息存放在一个表中。当查询数据时，通过连接操作查询多个表中的实体信 息。连接查询包括交叉连接查询、内连接查询、以及外连接查询。

2）在连接查询中，当使用连接关键字连接多个表，多个表中存在相同的字段名时。为 了区分，可以在字段名前加上表名以便区分。如表名.字段名或表名.\*。

3）连接查询又称多表查询；

## 交叉连接查询

交叉连接查询是使用CROSS JOIN关键字连接要查询的表中进行查询。其语法格式为：

**SELECT 指定要显示的字段 FROM 表名1 CROSS JOIN 表名2 ;**

交叉连接查询的结果将以笛卡尔积的形式进行返回。也就是返回第一个表中符合查询条 件的记录数乘以第二个表中符合查询条件的记录数。

## 内连接查询

内连接查询是使用INNER JOIN关键字连接要查询的表，并使用ON指定连接条件进行 查询。其语法格式为：  
  **SELECT \*|字段名1,字段名2,... FROM 表名1 INNER JOIN 表名2**

**ON 关系表达式 ;**

inner join关键字可忽略。在内连接查询中，on与where子句功能相同。

## 外连接查询

在内连接查询中，返回的结果只包含符合查询条件和连接条件的记录。但有时还需要包 含没有关系关联的数据，即返回查询结果中不仅包含符合条件的数据，而且还包含不符 合条件查询的数据。

外连接分为左外连接和右外连接。其语法格式为：

**SELECT 所查字段 FROM 表1 LEFT|RIGHT JOIN 表2**

**ON 表1.关系字段=表2.关系字段**

**WHERE 条件 ;**

1. 左连接LEFT JOIN：左连接查询的结果为LEFT JOIN关键字左边表中所有的记 录，以及LEFT JOIN关键字右边表中满足连接条件的记录。

如果左表中的某条记录存在，而右表中不存在，则在右表中显示为空。

1. 右连接RIGTH JOIN：右连接查询的结果为RIGHT JOIN关键字右边表中所有的记录， 以及左表表中满足条件的记录。

如果右表中某条记录存在，而左表中不存在，则在左表显示为空。

## 复合条件查询

复合条件查询是指在连接查询中，通过添加过滤条件来限制查询结果，使查询结果更加 准确。例如添加排序、分组、限制或者WHERE子句来对结果进行过滤。

# 子查询

子查询是指在查询语句中嵌套一条查询语句。在执行带有子查询语句的查询语句时，首先会先执行子查询语句，将子查询语句返回的结果作为外层查询语句的过滤条件。

在子查询中通常使用IN、EXISTS、ANY、ALL、关系运算符连接子查询语句。

## 带关系运算符的子查询

将子查询语句返回的结果作为外层查询语句中WHERE子句关系表达式中运算符的比较 值进行比较。其语法格式为：

**SELECT \* FROM 表1 WHERE 表1字段 关系运算符(子查询语句) ;**

其中的，带关系运算符的子查询语句只能返回单个值。

## 带IN关键字的子查询

带有IN关键字的子查询语句仅返回一个数据列，这个数据列中的值将供外层查询语 句进行操作,也就是将子查询的结果作为外层查询语句WHERE子句中IN条件表达式的 值集合。其语法格式为：

**SELECT \* FROM department WHERE did IN(SELECT did FROM employee WHERE age=20);**

## 带EXISTS关键字的子查询

EXISTS关键字后面的子查询语句将返回一个布尔值，如果子查询语句查询到了数据， 则返回TRUE，外层查询语句才会执行。如果没有查询到数据则返回FALSE，外层查询语 句不会执行。

其语法格式为：

**SELECT \* FROM department WHERE EXISTS(SELECT did FROM employee WHERE age>21);**

## 带ANY关键字的子查询

ANY关键字表示满足其中任意一个条件。外层查询语句创建一个表达式对子查询语句返 回的数据列进行比较。其语法格式如：

**SELECT \* FROM department WHERE did>ANY(SELECT did FROM employee) ;**

执行子查询语句后，返回employee表did这一列的数据。在外层查询语句指定的 WHERE子句中，只要did值大于employee表返回的did这一列的任意一值时，则 did 值所在记录将作为查询结果返回。

## 带ALL关键字的子查询

ALL关键字表示满足所有条件。外层查询语句创建一个表达式对子查询语句返回的数据 列值进行比较。其语法格式如：

**SELECT \* FROM department WHERE did>ALL(SELECT did FROM employee);**

执行子查询语句后，返回employee表did这一列的数据。在外层查询语句指定的WHERE 子句中，只要did值大于employee表返回的did这一个列的所有值时，该did值所在的 记录将作为查询结果返回。