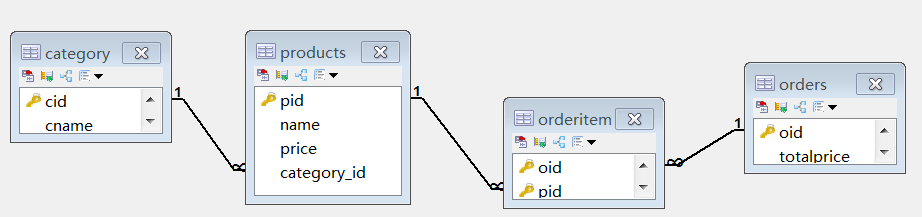
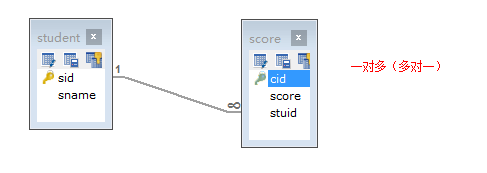
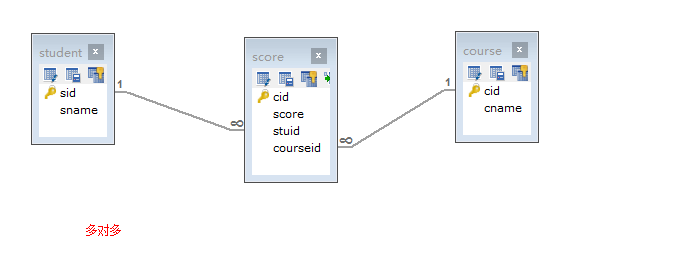
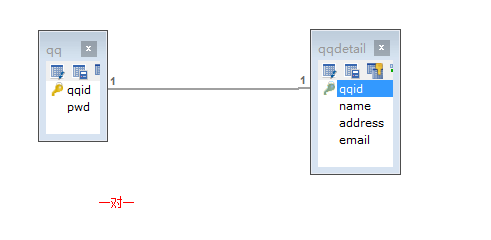
# 多表操作

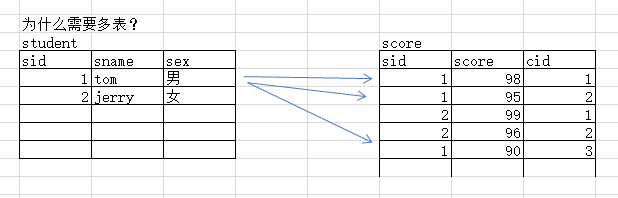
实际开发中，一个项目通常需要很多张表才能完成。例如：一个商城项目就需要分类表(category)、商品表(products)、订单表(orders)等多张表。且这些表的数据之间存在一定的关系，接下来我们将在单表的基础上，一起学习多表方面的知识。







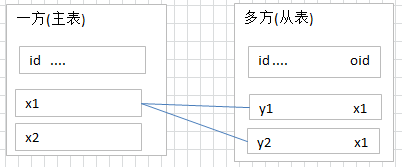




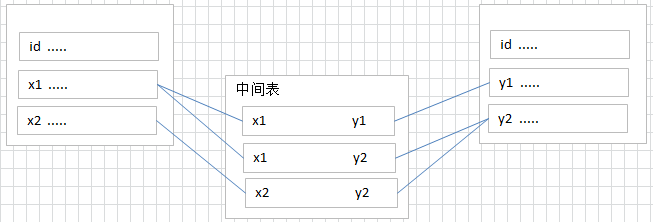
## 表与表之间的关系

有3类表关系：一对多（多对一）、多对多、一对一（了解）

* 一对多关系：
  + 常见实例：学生和考试成绩(画图),客户和订单，分类和商品，部门和员工.
  + 一对多建表原则：在从表(多方)创建一个字段，字段作为外键指向主表(一方)的主键.



* 多对多关系：
  + 常见实例：学生和教师,商品和订单(画图),学生和课程、用户和角色
  + 多对多关系建表原则：需要创建第三张表,中间表中至少两个字段，这两个字段分别作为外键指向各自一方的主键.

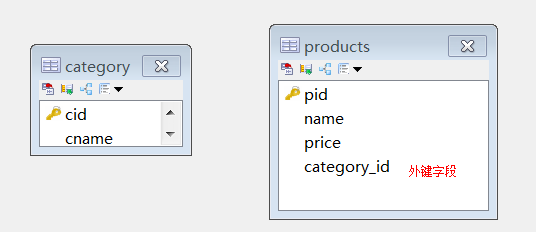


* 一对一关系：(了解)
  + 在实际的开发中应用不多,比如QQ号码,和QQ用户信息
    - 因为一对一可以创建成一张表.
  + 两种建表原则：
    - 外键唯一：主表的主键和从表的外键（唯一），形成主外键关系，外键唯一unique。
    - 外键是主键：主表的主键和从表的主键，形成主外键关系。

## 外键约束

现在我们有两张表“分类表”和“商品表”，为了表明商品属于哪个分类，通常情况下，我们将在商品表上添加一列，用于存放分类cid的信息，此列称为：外键





此时“分类表category”称为：主表，“cid”我们称为主键。“商品表products”称为：从表，category\_id称为外键。我们通过主表的主键和从表的外键来描述主外键关系，呈现就是一对多关系。

外键特点：

* + - 从表外键的值是对主表主键的引用。
    - 从表外键类型，必须与主表主键类型一致。
* 声明外键约束

语法：alter table 从表 add [constraint] [外键名称] foreign key (从表外键字段名) references 主表 (主表的主键);

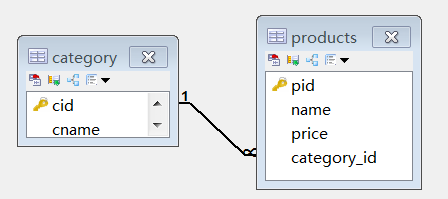
[外键名称] 用于删除外键约束的，一般建议“\_fk”结尾

alter table 从表 drop foreign key 外键名称

* 使用外键目的：
  + 保证数据完整性

## 一对多操作

### 分析



* category分类表，为一方，也就是主表，必须提供主键cid
* products商品表，为多方，也就是从表，必须提供外键category\_id

### 实现：分类和商品

###创建分类表

create table category(

cid int(32) PRIMARY KEY ,

cname varchar(100) #分类名称

);

# 商品表

CREATE TABLE products (

pid int PRIMARY KEY ,

pname VARCHAR(40) ,

price DOUBLE ,

category\_id int

);

#添加约束

alter table products add constraint product\_fk foreign key (category\_id) references category (cid);

### 操作

#1 向分类表中添加数据

INSERT INTO category (cid ,cname) VALUES(1,'服装');

#2 向商品表添加普通数据,没有外键数据，默认为null

INSERT INTO products (pid,pname) VALUES(1,'商品名称');

#3 向商品表添加普通数据，含有外键信息(数据存放在)

INSERT INTO products (pid ,pname ,category\_id) VALUES(2,'商品名称2', 1);

#4 向商品表添加普通数据，含有外键信息(数据不存在) -- 不能异常

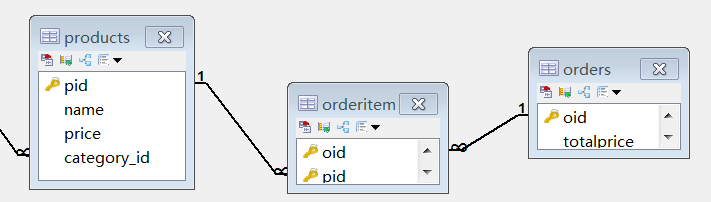
INSERT INTO products (pid ,pname ,category\_id) VALUES(3,'商品名称2',9);

#5 删除指定分类(分类被商品使用) -- 执行异常

DELETE FROM category WHERE cid = 1;

## 多对多

### 分析



* 商品和订单多对多关系，将拆分成两个一对多。
* products商品表，为其中一个一对多的主表，需要提供主键pid
* orders 订单表，为另一个一对多的主表，需要提供主键oid
* orderitem中间表，为另外添加的第三张表，需要提供两个外键oid和pid

### 实现：订单和商品

### 商品表[已存在]

### 订单表

create table orders(

oid int PRIMARY KEY ,

totalprice double #总计

);

### 订单项表

create table orderitem(

oid int,-- 订单id

pid int(50)-- 商品id

);

###---- 订单表和订单项表的主外键关系

alter table orderitem add constraint orderitem\_orders\_fk foreign key (oid) references orders(oid);

###---- 商品表和订单项表的主外键关系

alter table orderitem add constraint orderitem\_product\_fk foreign key (pid) references products(pid);

### 操作

#1 向商品表中添加数据

INSERT INTO products (pid,pname) VALUES(3,'商品名称');

#2 向订单表中添加数据

INSERT INTO orders (oid ,totalprice) VALUES(1,998);

INSERT INTO orders (oid ,totalprice) VALUES(2,100);

#3向中间表添加数据(数据存在)

INSERT INTO orderitem(pid,oid) VALUES(1, 1);

INSERT INTO orderitem(pid,oid) VALUES(1, 2);

INSERT INTO orderitem(pid,oid) VALUES(2,2);

#4删除中间表的数据

DELETE FROM orderitem WHERE pid=2 AND oid = 2;

#5向中间表添加数据(数据不存在) -- 执行异常

INSERT INTO orderitem(pid,oid) VALUES(2, 3);

#6删除商品表的数据 -- 执行异常

DELETE FROM products WHERE pid = 1;