# 一 表关系

## 0 前言

假如有一个包含产品目录的表，其中每类物品占一行。每种物品都具有 产品描述，价格，供应商信息。

现在，假如有由同一个供应商生产的多种物品，那么在何处存储供应商的信息（供应商名，地址，电话）？

可以创建两个表：一个表只存储供应商信息，一个表存储商品信息。

因为同一供应商生产的每个产品的供应商信息都是相同的，如果在产品表里重复描述，浪费时间和存储空间，且如果供应商信息改变，那么对应要修改很多处。

解决关联关系的方案：在产品表中，每个产品额外存储对应供应商的ID，这个存储我们称之为外键。

select vendor\_name,product\_name,product\_price

from vendors,products

where vendors.vendor\_id = products.product\_id

order by vendor\_name,product\_name;

如果上述案例没有书写 where，那么就会将所有的能够对应的数据全部展示出来，会造成大量的数据重复，这种没有联结关系的查询结果我们称之为： 笛卡尔积。检索出的数目将是第一个表中的行数乘以第二个表中的行数。

新的写法：推荐使用

select vendor\_name,product\_name,product\_price

from vendors inner join products

on vendors.vendor\_id = products.product\_id;

联结语句也支持多张表的联结，但是联结越多，性能越低，假如还有一张订单表：

select vendor\_name,product\_name,product\_price ，quantity

from orders,vendors,products

where vendors.vendor\_id = products.product\_id

and orders.product\_id = products.product\_id

and order\_num = 2000;

假如发现某物品(id)存在问题，那么想直达生产该物品的供应商生产的其他物品是否也存在问题：

select product\_id,product\_name

from products

where vendor\_id = (

select vendor\_id from products where product\_id = ‘3’

);

上述使用了子查询，我们还可使用联结查询来实现：

select p1.product\_id,p1.product\_name

form products as p1,products as p2

where p1.vendor\_id = p2.vendor\_id

and p2.product\_id = ‘3’;

这种查询中需要的两个表实际上是相同的表，使用表别名才能让mysql正确识别。

小提示：通配符也可以在上述案例中使用，比如 p1.\*

## 1 关系

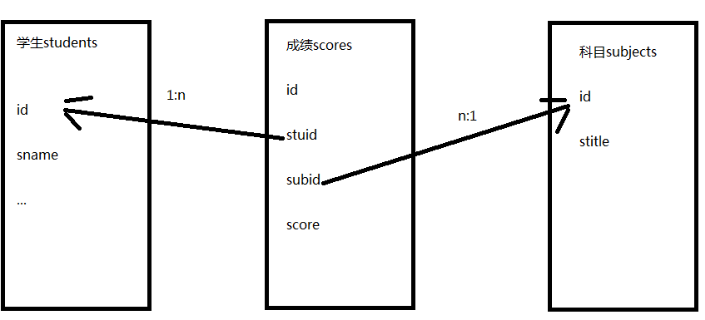
创建成绩表scores，字段有：id 学生 科目 成绩

思考：学生列应该存什么信息呢？

回答：学生列的数据不是在这里新建的，而应该从学生表引用过来。

关系也是一条数据，根据范式要求应该存储学生的编号，而不是学生的姓名等其它信息

同理，科目表也是关系列，引用科目表中的数据。



所以，关系应该是一个字段，创建表的语句如下

create table scores(

id int primary key auto\_increment,

stuid int,

subid int,

score decimal(5,2)

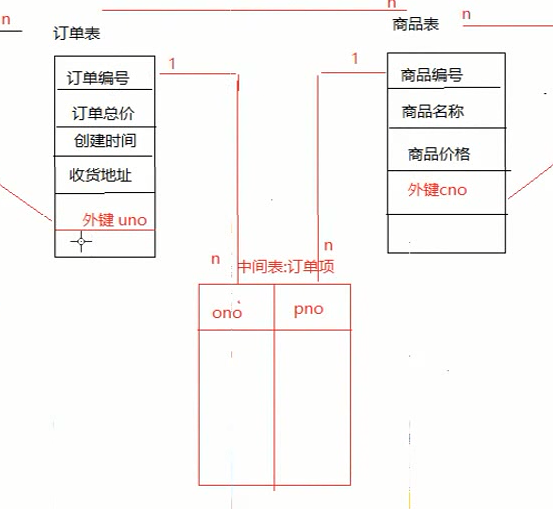
);

decimal是数据的精准度控制，表示：最多存储6位，小数位2位

一对多：例如商品和商品分类，关系字段存在‘多’的那张表中；

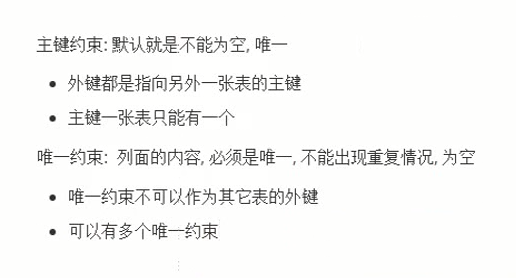
一对一：例如人和身份证，关系字段存在两张表中哪一张都可以；

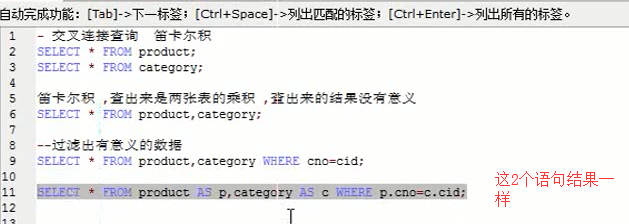
多对多：例如学生与选课，新建一张表存储关系，如下案例：

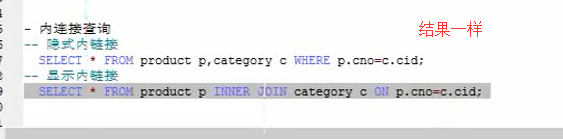




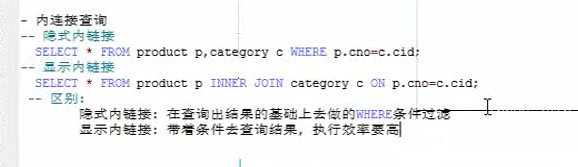






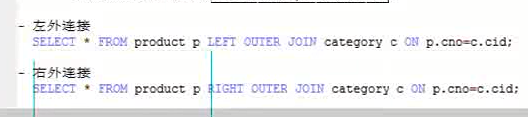


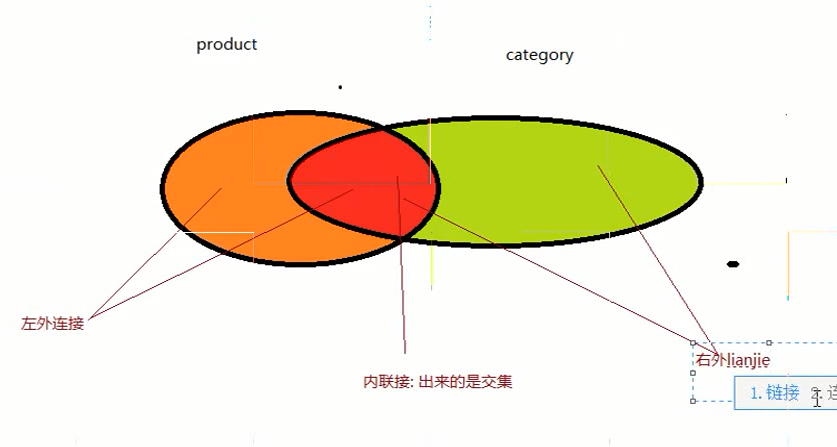
区别：



左外连接：会将左表中数据都查询出来，如果右表中没有对应数据，全是NULL







## 2 外键

虽然我们创建了 stuid subid这样的字段，但是在录入数据的时候，如果录入了 students表没有的id，仍然不会报错，会直接录入。如何保证关系列数据的有效性呢？

录入的stuid必须是students表中的id列存在的数据，可以通过外键约束进行数据的有效性验证。

为stuid添加外键约束

alter table scores add foreign key(stuid) references students(id);

此时插入或者修改数据时，如果stuid的值在students表中不存在则会报错。

在创建表时可以直接创建约束

create table scores(

id int primary key auto\_increment,

stuid int,

subid int,

score decimal(5,2),

foreign key(stuid) references students(id),

foreign key(subid) references subjects(id)

);

注意：主键 外键属性的一致性！！！！！

如果要删除students表中的tables中存在的外键，必须先在tables表中的所有该外键下的列删除，然后才能删除tables表中的外检。

## 3 外键的级联操作

在删除students表的数据时，如果这个id值在scores中已经存在，则会抛异常。

可以创建表时指定级联操作，也可以在创建表后再修改外键的级联操作

alter table scores add constraint stu\_sco foreign key(stuid) references students(id) on delete cascade;

级联操作的类型包括（即删除的4种方案）：

restrict（内置）：默认值，抛异常

cascade（级联）：如果主表的记录删掉，则从表中相关联的记录都将被删除

set null：将外键设置为空

no action：什么都不做

推荐：当表的数据会被别的表引用时，可以使用逻辑删除。

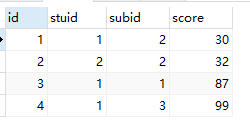
## 4 连接查询

怎么将3个表放到一起查询，并将结果显示在同一个结果集中呢？

问题：查询每个学生每个科目的分数

分析：学生姓名来源于students表，科目名称来源于subjects，分数来源于scores表，

直接使用 select \* from scores 得到的结果集如下：



根本无法查看是哪个学生哪个科目的成绩是多少，这时候需要使用连接查询。

则上面问题的答案是：（下列两种写法均可）

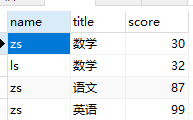
SELECT students.name,subjects.title,scores.score

FROM scores

INNER JOIN students ON scores.stuid=students.id

INNER JOIN subjects ON scores.subid=subjects.id

查询结果如下：



结论：当需要对有关系的多张表进行查询时，需要使用连接join，inner join 前后表谁先写都可以。

如下所示：

SELECT students.name,subjects.title,scores.score

FROM students

INNER JOIN scores ON scores.stuid=students.id

INNER JOIN subjects ON scores.subid=subjects.id

连接查询分类如下：

内连接（表A inner join 表B）：表A与表B匹配的行会出现在结果中。

比如有一个学生没有参加考试，那么scores表中就没有他的成绩记录，查询结果是无法查到的。

左连接（表A left join 表B）：表A与表B匹配的行会出现在结果中，外加表A中独有的数据，未对应的数据使用null填充

右连接（表A right join 表B）：表A与表B匹配的行会出现在结果中，外加表B中独有的数据，未对应的数据使用null填充

在查询或条件中推荐使用“表名.列名”的语法

如果多个表中列名不重复可以省略“表名.”部分

如果表名太长，可以在表名后使用' as 简写名'或' 简写名'，为表起个临时的简写名称

示例:

查询学生的姓名、平均分

select students.sname,avg(scores.score)

from scores

inner join students on scores.stuid=students.id

group by students.sname;

查询男生的姓名、总分

select students.sname,avg(scores.score)

from scores

inner join students on scores.stuid=students.id

where students.gender=1

group by students.sname;

查询科目的名称、平均分

select subjects.stitle,avg(scores.score)

from scores

inner join subjects on scores.subid=subjects.id

group by subjects.stitle;

查询未删除科目的名称、最高分、平均分

select subjects.stitle,avg(scores.score),max(scores.score)

from scores

inner join subjects on scores.subid=subjects.id

where subjects.isdelete=0

group by subjects.stitle;

## 5 自关联

### 5.0 问题

设计省信息的表结构provinces：id ptitle

设计市信息的表结构citys：id ctitle proid

citys表的proid表示城市所属的省，对应着provinces表的id值

问题：能不能将两个表合成一张表呢？

思考：观察两张表发现，citys表比provinces表多一个列proid，其它列的类型都是一样的

意义：存储的都是地区信息，而且每种信息的数据量有限，没必要增加一个新表，或者将来还要存储区、乡镇信息，都增加新表的开销太大

答案：定义表areas，结构如下

id

atitle

pid

因为省没有所属的省份，所以可以填写为null

城市所属的省份pid，填写省所对应的编号id

这就是自关联，表中的某一列，关联了这个表中的另外一列，但是它们的业务逻辑含义是不一样的，城市信息的pid引用的是省信息的id

在这个表中，结构不变，可以添加区县、乡镇街道、村社区等信息

创建areas表的语句如下：

create table areas(

id int primary key,

atitle varchar(20),

pid int,

foreign key(pid) references areas(id)

);

### 5.1从sql文件中导入数据

source areas.sql;

### 5.2 案例

查询一共有多少个省

查询省的名称为“山西省”的所有城市

select city.\* from areas as city

inner join areas as province on city.pid=province.id

where province.atitle='山西省';

查询市的名称为“广州市”的所有区县

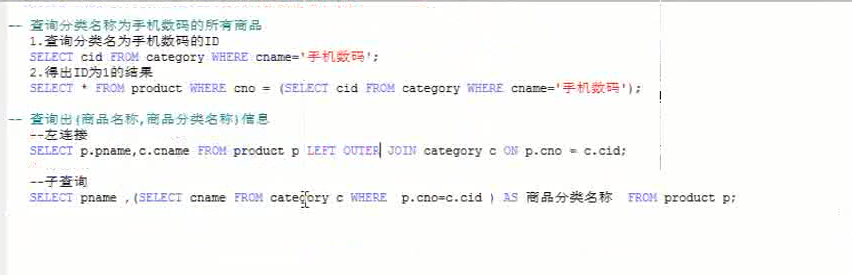
select dis.\*,dis2.\* from areas as dis

inner join areas as city on city.id=dis.pid

left join areas as dis2 on dis.id=dis2.pid

where city.atitle='广州市';

## 6 子查询



### 6.1 使用子查询的原因

连接查询虽然实现了夺标记录的查询，但是性能极差，一般使用子查询来替代。

一般的查询语句：

select \* from students s,teachers t where s.teacherno = t.number;

上述的查询语句，首先会对两个表进行笛卡尔积操作，然后再选区符合匹配条件的数据进行记录。笛卡尔积的数据是相当大的，如果原始数据量很大，很容易死机。

所谓子查询是指在查询中嵌套了其他的若干查询，又称嵌套查询。

子查询经常出现在where和from子句中：

where子句：一般返回单行单列，多行单列，单行多列数据

from子句： 一般返回多行多列数据记录，可以当做一张临时表。

### 6.2 返回单行多列的子查询

select ename,sal,job from t\_employee where(sal,job)=(

select sal,job from t\_employee where ename='Smith

);

### 6.3 带in any all的子查询

当主查询的条件在子查询的结果中时，使用in；

any关键字用来表示主查询的条件为满足子查询返回查询结果中任意一条数据记录，有三种匹配方式：

=any：与 in 作用一致；

>any、>=any：

<any、<=any:

select ename,sal from t\_employee where sale >=any(

select sal,job from t\_employee where ename='Smith

);

all表示朱查询的条件为满足子查询返回结果中所有数据记录：

>all >=all

<all <=all

如下：查询各学生的语文、数学、英语的成绩

select sname,

(select sco.score from scores sco inner join subjects sub on sco.subid=sub.id where sub.stitle='语文' and stuid=stu.id) as 语文,

(select sco.score from scores sco inner join subjects sub on sco.subid=sub.id where sub.stitle='数学' and stuid=stu.id) as 数学,

(select sco.score from scores sco inner join subjects sub on sco.subid=sub.id where sub.stitle='英语' and stuid=stu.id) as 英语

from students stu;

## 7 分页查询

