

表设计

- 主键：添加在表中某个字段上的一种设置，该字段就可以唯一标识当前的这一行数据，默认情况下每张表都应该有主键，一张表只能有一个主键，所谓的多个主键是指联合主键。

```
create table user(  
    id int primary key auto_increment,  
    ...  
)
```

```
create table user(  
    id int not null unique,  
    id2 int not null unique,  
    primary key(id,id2),  
    ...  
)
```

- 外键：添加在表中某个字段上的一种设置，一旦添加外键（被其他表的主键所约束），该表就会被其他表所约束（数据不能随意添加），叫做从表，约束它的表叫做主表。表的主从关系是通过主外键的约束关系来维护的，主表中的主键会约束从表中的外键，主表是class，从表是student，student中有外键被class的主键约束。
 - 从表中添加的数据（外键的值）必须是主表中主键列中存在的值。
 - 删除主表的数据时，必须先解除主外键约束关系。

```
create table student(  
    id int primary key auto_increment,  
    ...  
    cid int,  
    foreign key(cid) references class(id)  
)
```

```
create table student(  
    id int primary key auto_increment,  
    ...  
);  
alter table student add cid int;  
alter table student add foreign key(cid) references class(id);
```

```
alter table student drop foreign key 外键名称;
```

表与表的关系

- 一对多
- 一旦两张表建立了主外键约束关系，那么这两张表就有了一对多的关系，有外键的表是从表，约束此外键的主键所在的表是主表。

```
use demo;

create table customer(
    id int primary key auto_increment,
    name varchar(11)
);

create table orders(
    id int primary key auto_increment,
    name varchar(11),
    cid int,
    foreign key (cid) references customer(id)
);

insert into customer(name) values ("张三");
insert into customer(name) values ("李四");

insert into orders(name,cid) values ("订单1",1);
insert into orders(name,cid) values ("订单2",1);
insert into orders(name,cid) values ("订单3",2);
```

```
package com.southwind.entity;

import java.util.List;

public class Customer {
    private int id;
    private String name;
    private List<Orders> orders;

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
```

```

        this.name = name;
    }

    public List<Orders> getOrders() {
        return orders;
    }

    public void setOrders(List<Orders> orders) {
        this.orders = orders;
    }

    public Customer(int id, String name) {
        this.id = id;
        this.name = name;
    }
}

```

```

package com.southwind.entity;

public class Orders {
    private int id;
    private String name;
    private Customer customer;

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public Customer getCustomer() {
        return customer;
    }

    public void setCustomer(Customer customer) {
        this.customer = customer;
    }
}

```

```
public Orders(int id, String name) {  
    this.id = id;  
    this.name = name;  
}  
}
```

嵌套查询：

```
select * from orders where cid = (select id from customer where name = "张三");
```

关联查询：

```
select customer.id, customer.name, orders.id, orders.name from customer, orders  
where orders.cid = customer.id and customer.name = "张三";
```

连接查询：

- 内连接

```
select * from tab1 inner join tab2;
```

```
select * from customer inner join orders;
```

```
select customer.id, customer.name, orders.id, orders.name from customer  
inner join orders where orders.cid = customer.id and customer.name =  
"张三";
```

- 外连接：必须加条件

- 左连接：左表所有的数据和右表满足条件的数据。

```
select * from orders left join customer on customer.id =  
orders.cid and customer.name = "张三";
```

- 右连接：右表所有的数据和左表满足条件的数据。

```
select * from orders right join customer on customer.id =  
orders.cid and customer.name = "张三";
```

```
select * from customer left join orders on customer.id =  
orders.cid and customer.name = "张三";  
等于  
select * from orders right join customer on customer.id =  
orders.cid and customer.name = "张三";
```

- 多对多
- 多对多是由两个一对多关系组合起来的，需要通过一张中间表间接建立两张目标表的多对多关系，

两张目标表与中间表都是一对多关系，两张目标表都是主表，中间表是从表。

```
use demo;  
create table student(  
    id int primary key auto_increment,  
    name varchar(11)  
);  
create table course(  
    id int primary key auto_increment,  
    name varchar(11),  
    sid int,  
    foreign key (sid) references student(id)  
);  
create table student_course(  
    id int primary key auto_increment,  
    sid int,  
    cid int,  
    foreign key (sid) references student(id),  
    foreign key (cid) references course(id)  
);
```