

MySQL多表

MySQL多表

电商案例分析

分析商城需要的表：

分析表之间的关系

一对多：

多对多：

一对一：

实现

一对多（分类和商品）

一对多（用户和订单）

多对多（订单和商品）

多表查询

插入测试数据

交叉查询

内联查询

外联查询

子查询

学习目标：

1. 描述表与表之间的关系
2. 一对多和多对多表关系的建立
3. 多表查询数据
4. 加深对JDBC使用的理解

电商案例分析

分析商城需要的表：

1. 用户表

字段名	类型	主键	说明
uid	INT	Y	用户ID
username	CHAR		
password	CHAR		
gender	CHAR		
tel	CHAR		
email	CHAR		
intro	CHAR		
activeCode	CHAR		
state	CHAR		

2. 商品表

字段名	类型	主键	说明
pid	INT	Y	
name	CHAR		
price	Double		现价
pnum	INT		
des	CHAR		
origin_price	Double		原价
category	INT		类型

3. 分类表

字段名	类型	主键	说明
cid	INT	Y	
cname	CHAR		类型名

4. 订单表

字段名	类型	主键	说明
oid	INT	Y	
total_price	Double		
receiver_info	INT		收货信息
order_user	INT		下单用户
state	CHAR		订单状态

思考：

订单中如何保存购买的商品和数量

分析表之间的关系

主从表概念：主从表，从表数据依赖于主表，一般最后查询数据时把主表与从表进行关联查询。主表可用于存储主要信息，如客户资料（客户编号，客户名称，客户公司，客户单位等），从表用来存储客户扩展信息（客户订单信息，客户地址信息，客户联系方式信息等）。

注意：主从表是相对概念，一个表既可以是主表，同时也可以是从表

一对多：

- 示例：客户-订单，班级-学生，分类-商品
- 建表原则：在从表创建一个字段，字段作为外键指向主表的主键

多对多：

- 示例：学生-选修课，商品-订单，用户和群组
- 建表原则：需要创建第三张表（从表），与已存在的两张表都是1对多关系，至少应该有两个字段，分别是其他两张主表中的主键

一对一：

- 一对一可以构造一张表，所以不多见
- 建表原则：外检唯一、外键是主键

实现

一对多（分类和商品）

```
CREATE TABLE IF NOT EXISTS category (  
  cid varchar(40) PRIMARY KEY,  
  cname varchar(100)  
);  
  
CREATE TABLE IF NOT EXISTS product (  
  pid varchar(40) PRIMARY KEY,  
  pname varchar(100),  
  price DOUBLE  
);  
  
ALTER TABLE product ADD COLUMN category_id varchar(40);  
  
ALTER TABLE product add CONSTRAINT FOREIGN KEY (category_id) REFERENCES category  
(cid);
```

一对多（用户和订单）

```
CREATE TABLE IF NOT EXISTS `order` (  
  oid varchar(40) primary key,  
  total_price double,  
  receive_info varchar(40),  
  order_user varchar(40),  
  state varchar(40)  
);  
  
CREATE TABLE IF NOT EXISTS `user` (  
  uid varchar(40) PRIMARY KEY,  
  password varchar(100),  
  price DOUBLE,  
  gender varchar(20),  
  tel varchar(40),  
  state varchar(40)  
);  
  
ALTER TABLE `order` add CONSTRAINT FOREIGN KEY (order_user) REFERENCES `user`  
(uid);
```

多对多（订单和商品）

```
create table if not exists `orderitem`(  
oid varchar(40),  
pid varchar(40)  
);  
  
ALTER table `orderitem` add PRIMARY key (oid,pid);  
  
ALTER TABLE `orderitem` add CONSTRAINT foreign key (oid) REFERENCES `order`(oid);  
ALTER TABLE `orderitem` add CONSTRAINT foreign key (pid) REFERENCES product(pid);
```

多表查询

插入测试数据

对于主从表的插入和删除需要注意：

- 在从表插入时，主表中应该已有相应的数据 (先插入主表，再插入从表)
- 删除必须先删除从表中的数据（依赖主表中的数据的关联数据），才能删除主表

```
INSERT INTO category(cid,cname) values ('c001','家电');  
INSERT INTO category(cid,cname) values ('c002','服装');  
INSERT INTO category(cid,cname) values ('c003','外设');  
  
INSERT INTO product(pid,pname,price,category_id) values('p001','小米电视  
4S',2499,'c001');  
INSERT INTO product(pid,pname,price,category_id) values('p002','SONY电视  
A77',5499,'c001');  
INSERT INTO product(pid,pname,price,category_id) values('p003','海信电视  
K7000',3499,'c001');  
  
INSERT INTO product(pid,pname,price,category_id) values('p004','Prada经典风  
衣',2499,'c002');  
INSERT INTO product(pid,pname,price,category_id) values('p005','LEVIS男  
裤',499,'c002');  
INSERT INTO product(pid,pname,price,category_id) values('p006','ADIDAS球  
鞋',499,'c002');  
  
INSERT INTO product(pid,pname,price,category_id) values('p007','FLICO NAJIA2键  
盘',1499,'c003');  
INSERT INTO product(pid,pname,price,category_id) values('p008','HHKB  
P2BT',1999,'c003');  
INSERT INTO product(pid,pname,price,category_id) values('p009','MAGIC  
mouse',499,'c003');
```

交叉查询

```
select * from category,product;
```

得到两个表的乘积，可以用来生成交叉数据

内联查询

```
select * from product inner join category on category_id=cid;
```

以product为基础，根据on后条件，把category表的数据匹配到每一条数据上

在最后可以追加条件，条件可以是两个表的任意字段：

```
select * from product inner join category on category_id=cid where price<=1000;  
select * from product inner join category on category_id=cid where cname LIKE '外设';
```

为了防止字段重名，可以使用表明加字段方式，为了简化可以使用别名的方式：

```
select * from product inner join category on category_id=cid where category.cname  
LIKE '外设';  
select * from product inner join category as c on category_id=cid where c.cname  
LIKE '外设';
```

外联查询

```
INSERT INTO category VALUES('c004','日化');  
INSERT INTO product VALUES ('p010','海飞丝',20,NULL);
```

对比

```
select * from product left join category on category_id=cid where price<=1000;  
select * from product INNER join category on category_id=cid where price<=1000;
```

对比

```
select * from product right join category on category_id=cid;  
select * from product left join category on category_id=cid;
```

子查询

把查询结果作为新的表进行查询

```
select * from product where category_id = (SELECT cid from category where  
cname='外设');
```