

购物网站的数据库设计

本小节以购物网站为例介绍关系数据库和文档数据库的设计。

概念结构设计

购物网站陈列了各种各样的商品，提供了商品的详细信息，包括商品名称、种类、价格、仓储地址等。用户登录网站之后可以浏览商品并选择想购买的商品，然后提交购物请求，购物请求中包括商品的种类、数量、购物客户的姓名、地址和电话等。系统审核用户的购物请求，完成购买行为并记录用户的购买历史。购买过程中的付款和送货通过调用外部系统进行实现。

通过分析发现购物网站中主要包含用户和商品两类实体对象。用户由用户号Uid、用户名Uname、地址Uadd、电话Tel和购物喜好Pref等特征属性刻画，其中用户号Uid是主码。商品由商品号Pid、商品名Pname、商品种类Category、商品价格Price、仓储地址Padd等特征属性刻画，其中商品号Pid是主码。用户可以进行多次购物，一次购物可以购买多件商品，同一个商品可以被多个用户购买。如何来刻画这种购物行为呢？一种方法是直接用用户和商品之间多对多的购买联系来刻画，购物网站的概念模型E-R图如图6.6(a)所示。另一种方法是新增一个订单实体对象，订单由订单号Did和订单时间Date等特征属性刻画，其中订单号Did是主码，然后用用户和订单之间一对多的下单联系与订单和商品之间多对多的包含联系来刻画用户的购物行为，购物网站的概念模型E-R图如图6.6(b)所示。

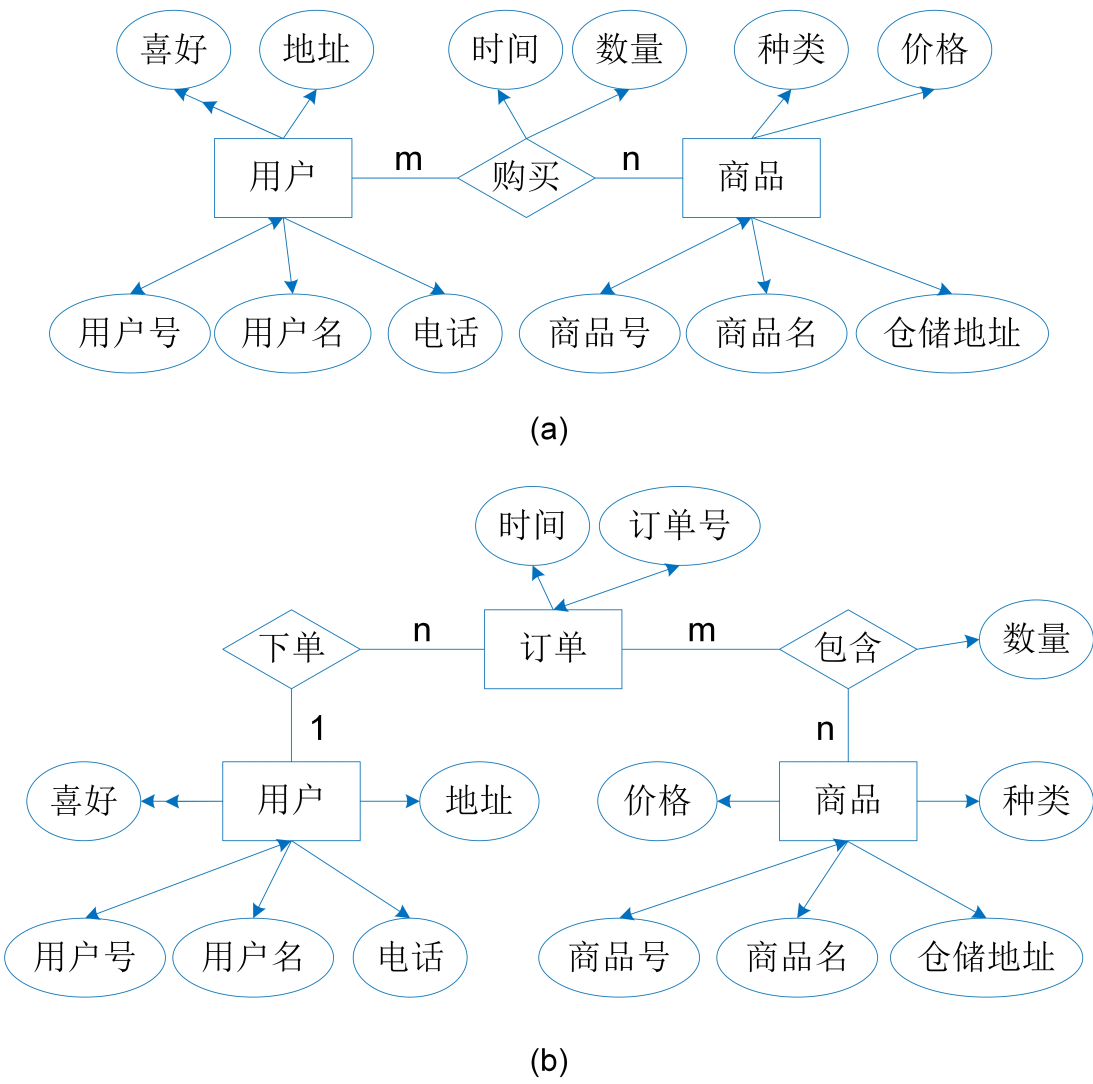


图 6.6 购物网站概念模型E-R图

哪一种概念模型更合理呢？在概念模型E-R图中，联系指的是两类实体之间的固定关系，也就是两个实体的主码能够唯一确定一条联系记录。在图6.6(a)的E-R图中，因为一个用户可以多次购买同一个商品，所以用户号和商品号不能唯一确定一条购买记录。在图6.6(b)的E-R图中，用户号和订单号能唯一确定一条下单记录，订单号和商品号能够唯一确定一条订单详情的记录，同一个商品的两次购买分别对应两个订单号。所以，图6.6(b)中的E-R图才是合适的购物网站概念模型。

关系模式设计

在概念结构设计的基础上设计购物网站的关系模式，即将6.6(b)的E-R转换为关系模式。

根据关系模式设计的规则，可以得到以下5个关系模式：

购物网站的关系模式设计：

用户表：User(Uid, Uname, Uadd, Tel), Uid是主码；

商品表：Product(Pid, Pname, Category, Price, Padd), Pid是主码；

订单表：Order(Oid, Uid, Date), OId是主码；

用户喜好表：Preference(Uid, Pref), Uid和Pref是主码；

订单详情表：OrderLine(Oid, Pid, Quantity), Oid和Pid是主码；

关系模式的设计步骤为：

- 首先，每类实体对象转换为一个关系模式，实体对象的唯一性属性和单值属性构成关系模式的属性，则生成用户表User(Uid, Uname, Uadd, Tel)、商品表Product(Pid, Pname, Category, Price, Padd)和订单表Order(Oid, Uid, Date)；
- 然后，用户实体对象的多值属性喜好转换为一个关系模式，用户的唯一性属性和多值属性构成关系模式的属性，则有用户喜好表Preference(Uid, Pref)；
- 其次，用户和订单之间的一对多联系合并到订单表中，将用户的唯一性属性添加到订单表中，则生成订单表Order(Oid, Uid, Date)；
- 最后，订单和商品之间的多对多联系转换为一个关系模式，订单的唯一属性、商品的唯一属性和联系的自身属性构成该关系模式的属性，则有订单详情表OrderLine(Oid, Pid, Quantity)。

文档模式设计

在概念结构设计的基础上设计购物网站的文档模式，即将6.6(b)的E-R转换为文档模式。

根据文档模式设计的规则，可以得到以下3个文档集：

购物网站的文档模式设计：

用户文档集：User{Uid, Uname, Uadd, Tel, Pref[]};

商品文档集：Product{Pid, Pname, Category, Price, Padd};

订单文档集：Order{Oid, Uid, Date, OrderLine[{Pid, Quantity}]};

文档模式设计的步骤为：

- 首先，每类实体对象转换为一个文档集，实体的所有属性转换为文档的属性，其中多值属性以数组的形式进行存储，则生成用户文档集User{Uid, Uname, Uadd, Tel, Pref[]}、商品文档集Product{Pid, Pname, Category, Price, Padd}和订单文档集Order{Oid, Date}；

- 然后，用户与订单之间的一对多联系合并到订单文档集中，将用户的唯一性属性加入订单文档集中，则生成订单文档集Order{Oid, Uid, Date};
- 最后，订单与商品之间的多对多联系以文档嵌套的形式存储到订单文档集中，商品的唯一性属性和联系的自身属性构成嵌套文档的属性，则有订单文档集Order{Oid, Uid, Date,OrderLine[{Pid, Quantity}]}。

可以看出，文档数据库的设计比关系数据库设计更加灵活，没有那么多的限制和约束。文档数据库允许实体的多值属性以数组形式存储，实体间的多对多联系以文档嵌套的形式存储，而关系数据库只能用独立的关系模式来刻画。