1. E-R diagram

1.1. Entity

1) customer

: 고객에 대한 정보를 나타낸다. 각 고객 별로 고유한 ID(c.ID)(양의 정수)를 부여하여 이를 primary key로 설정하였다. 고객 이름에 대한 name 속성(문자열), 고객 연락처에 대한 phone 속성(양의 정수), 고객 주소에 대한 address 속성(문자열), 금액 지불 방법을 나타내는 payment 속성(문자열)이 있다. payment의 값으로 가능한 것은 'account'와 'card', null 등이 있다. default는 null 값이고, 회사와 따로 계약한 고객은 'account'를, online을 통한 card 결제 기록이 있는 고객은 'card'를 저장한다.

2) product

: 제품에 대한 정보를 나타낸다. 각 제품 별로 고유한 ID(p.ID)(양의 정수)를 부여하여 이를 primary key로 설정하였다. 그리고 속성으로는 어떤 종류의 어떤 제조사 제품인지에 대한 정보를 저장하는 type(문자열), manufacturer(문자열), 그리고 상품 가격을 저장하는 p.price(양의 정수) 등이 있다.

3) package

: 몇 개의 제품을 묶어 함께 판매하는 패키지에 대한 정보를 나타낸다. 한 패키지는 두 개 이상의 제품이 속해있어야 하며, 각 패키지 별로 고유한 ID(pk.ID)(양의 정수)를 부여하여 이를 primary key로 설정하였다. 해당 패키지에 어떤 제품이 속하는지를 나타내는 pd 속성(multi-valued, p.ID), 패키지 가격을 저장하는 pk.price(양의 정수) 속성이 있다.

4) store

: 상점에 대한 정보를 나타낸다. in-store 상점과, online 상점이 있다. 각 in-store 상점과 online 상점에 고유한 ID(s_ID)(양의 정수)를 부여하여 이를 primary key로 설정하였다. 상점의 지역을 나타내는 region 속성(문자열)이 있다. online 상점의 경우, region 값은 online이다.

5) card

: 고객의 카드 정보를 저장한다. primary key는 customer의 c.ID이고, 카드 회사를 저장하는 문자열 속성 c.com, 카드 번호를 저장하는 정수형 속성 c.num이 있다.

6) sale

: 고객이 상품을 구매했을 때 판매 정보를 나타내는 관계이다. primary key는 o.ID(주문번호)이고, 속성은 p.ID(multi-valued), pk.ID(multi-valued), o.price(구매 가격, 양의 정수), pay(결제 수단, 문자열), ord.date(구매 날짜, 양의 정수, ex.20220428)

7) shipping

: 온라인 주문 시 배송에 관한 정보로 primary key는 tr.num(tracking number, 양의 정수)이고, 속성은 ship.com(shipping company, 문자열), pro.date(약속 날짜, 양의 정수), arr.date(배송된 날짜, 양의 정수) 등이 있다.

1-2. Relationship

1) sale_store

: sale과 store에 관한 관계이다. sale당 하나의 상점만 가능하고, 하나의 상점에 여러 sale이 가능하므로 many-to-one relation이다. 또한, 모든 sale은 상점에 연결되어 있어야 하므로 sale 쪽은 total이다.

2) sale_cust

: sale과 customer에 관한 관계이다. sale당 한 명의 고객만 가능하고, 한 명의 고객에 여러 sale이 가능하므로 many-to-one relation이다. 또한, 모든 sale은 고객에 연결되어 있어야 하므로 sale 쪽은 total이다.

3) inventory

: 재고에 대한 정보를 나타낸다. in-store 상점의 재고와, warehouse(online)의 재고를 나타낸다. warehouse에 대한 재고 정보는 online 상점으로 표시한다. 제품의 재고를 나타내는 정수형 속성 stock이 있다. 각 상점마다 각 제품의 재고를 표시해야 하므로 many-to-many relation이며, 양쪽 모두 total이다.

4) cust_card

: card와 customer에 대한 관계로, 모든 카드 정보는 고객과 연결되어야 하고, 고객은 여러 개의 카드를 가질 수 있으므로 many-to-one relation이며, card 쪽은 total이다. 또한, card의 primary key인 c.ID가 customer의 primary key인 c.ID에 dependent하므로, card는 weak entity set이고, customer은 identifying set, cust_card는 identifying relationship이다.

5) pro_pack

: product와 package에 대한 관계로, 한 패키지에 다수의 제품이 연결되어 있고, 각 제품은 여러 패키지에 연결될 수 있으므로 many-to-many relationship이다. 또한, 모든 패키지는 제품에 연결되어 있어야 하므로 package 쪽은 total이다.

6) delivery

: sale과 shipping에 대한 관계로, 한 sale에 한 shipping만 가능하므로 one-to-one relationship이다.

2. Schema diagram

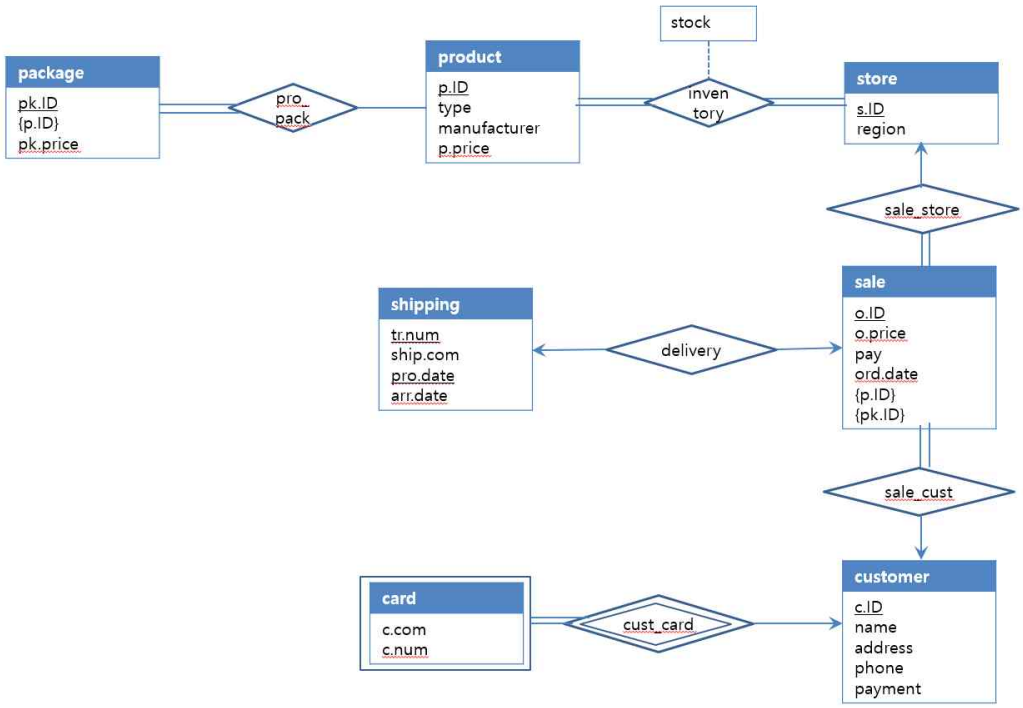
1) pro_pack은 package와 product의 many-to-many relationship이었으므로 schema diagram에서 하나의 table이 되고, primary key는 package와 product의 primary key가 된다.

2) inventory는 product와 store의 many-to-many relationship이었으므로 schema diagram에서 하나의 table이 되고, primary key는 store와 product의 primary key가 되며, stock은 속성이 된다.

3) card는 weak entity set이므로 identifying strong entitiy set의 primary key를 포함하는 하나의 table이 되고, identifying relationship인 cust_card는 many-to-one이고, total이므로 별도의 schema를 작성하지 않는다. 또한, sale의 foreign key로 customer의 primary key인 c.ID가 온다.

4) sale_store과 sale_cust는 many-to-one이고, total이므로 별도의 schema를 작성하지 않는다. 또한, sale의 foreign key로 store의 primary key인 s.ID가 온다.

5) delivery는 one-to-one이므로 한 쪽의 primary key를 다른 쪽의 속성으로 집어 넣고 delivery는 별도의 schema를 작성하지 않는다. 여기서는 shipping의 primaey key인 tr.num을 sale의 속성으로 집어 넣었다.


package
pk.ID
{p.ID}
pk.price
pro_
pack
product
p.ID
type
manufacturer
p.price
stock
inven
tory
store
s.ID
region
sale_store
sale
o.ID
o.price
pay
ord.date
{p.ID}
{pk.ID}
shipping
tr.num
ship.com
pro.date
arr.date
delivery
sale_cust
card
c.com
c.num
cust_card
customer
c.ID
name
address
phone
payment