

2022-01

CSE4110-02

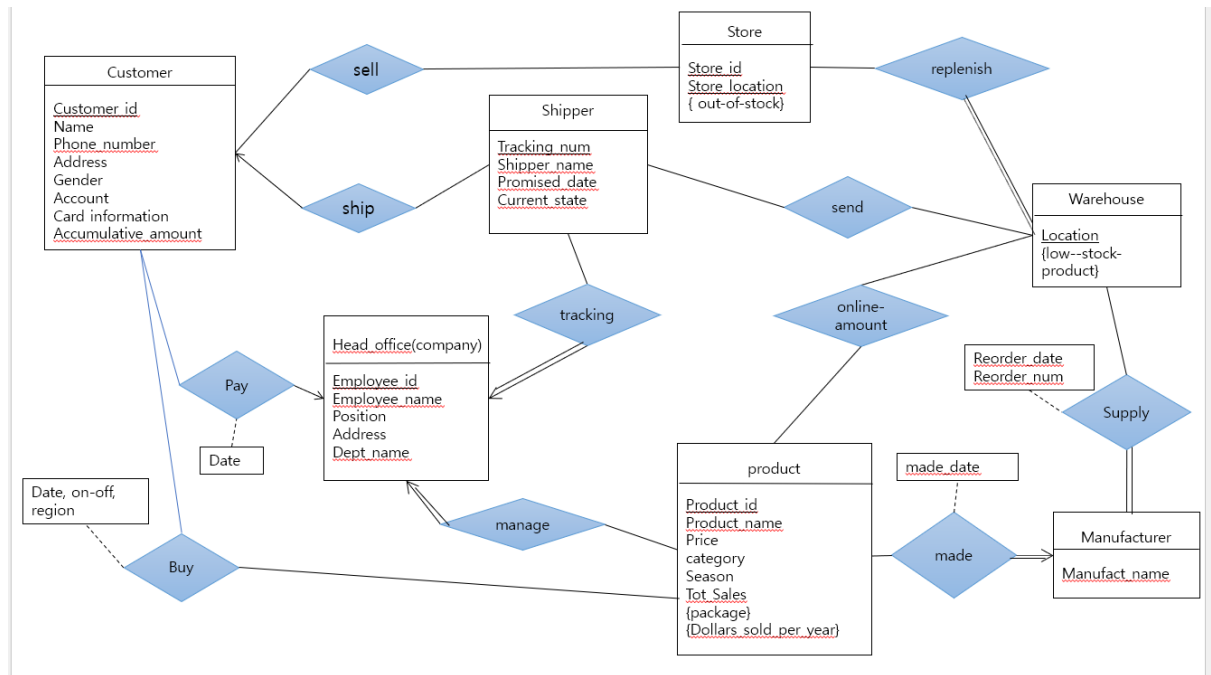
Database system

Project1 E-R Diagram and Relational Schema design

국어국문학, 컴퓨터공학

20190018 김유이

1. E-R Diagram



<그림1: E-R diagram>

1) Entity

- ① Customer : 고객을 나타낸다. 고객 id, 이름, 핸드폰 번호, 주소, 성별, 계좌번호, 카드 정보, 누적 구매액의 정보를 저장한다. 이름은 중복될 수 있으므로 고객 id가 primary key가 되며, account와 card information은 온라인 고객이 아닌 경우에는 null일 수 있다. Account의 경우 정기 결제 고객만 입력되며, 카드 정보는 비정기 결제 고객의 경우만 가지고 있다.
- ② Product: 제품을 나타낸다. 제품 id, 제품명, 가격, 카테고리(생활 가전인지 주방 가전인지 컴퓨터인지 등등), season(특정 제품의 수요가 증가하는 계절을 의미), Tot_sales(해당 제품의 총 판매량) package(컴퓨터의 경우 모니터와 스피커 등), dollars_sold_per_year(년 매출액)의 정보를 저장한다. 제품id를 primary key로 가진다.
- ③ head_office(company): 오프라인 매장 및 온라인 판매를 관리하는 본사(의 직원들)를 의미한다. 한 직원이 여러 고객의 결제 관리, 제품들 관리, 각 배송의 tracking_num 저장 및 관리를 담당할 수 있다. 직원id, 이름, 직급, (메일)주소, 부서명의 정보를 가진다. 직원 id를 primary key로 가진다.
- ④ Manufacturer : 제품 제조사이다. 제조사 이름을 저장하고 primary_key로 가진다. Warehouse에 제품을 공급한다.
- ⑤ Warehouse: 제품 재고를 저장하는 창고이다. Location(상하이 공장, 파주공장 등),

low_stock_product (재고 부족 상품들)의 정보를 저장한다. 따라서, primary_key는 location이다. 제조사로부터 제품을 공급받고, 제품의 온라인 판매량을 받아서 shipper에게 배송해야 할 제품을 넘긴다. 오프라인 매장에 재고를 보충한다.

- ⑥ Store: 오프라인 매장이다. 어느 지역에 있는 어떤 매장인지 구별하기 위해 store_location과 store_id를 primary_key로 가진다. Out-of-stock(재고가 떨어진 상품들)을 저장해서 warehouse로부터 재고를 보충받는다.
- ⑦ Shipper: 운송 정보이다. 정확하게는 shipment이지만 각 배송 정보에 적힌 회사가 제품들을 고객에게 배송하므로 shipper 라고 이름 붙였다. Shipper_name(운송 회사 이름), tracking_num(각 배송 추적 넘버), promised date(배송 예정일), current_state(배송 현황)의 정보를 가지고 있다. 배송 정보는 본사에서 관리 및 추적가능해야 하며, shipper는 고객에게 제품을 배송한다. 각 배송 정보를 구분할 수 있는 tracking_num을 primary_key로 가진다.

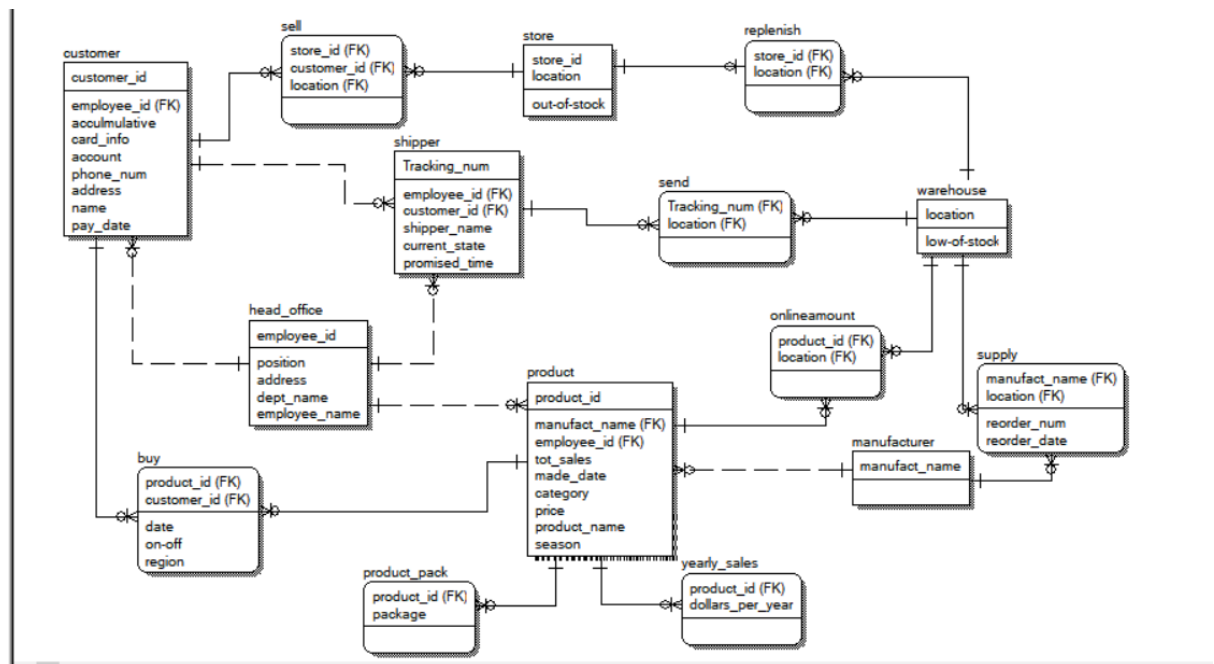
2) relationship

- ① Buy: 고객은 물건을 구매한다. 고객과 제품 사이의 relationship이다. 이때 relationship은 구매 날짜(date)와 온라인 구매인지 오프라인 구매인지 여부(on-off)와 오프라인 구매라면 어느 지역에서 샀는지(region, 온라인이라면 null) 등 자체적인 속성을 가지고 있다.
- ② Pay: 온라인 고객의 경우 회사와 contact하여 결제를 하므로 본사와 고객 사이의 관계이다. 이때 relationship은 pay라는 자체적인 속성을 추가로 가진다.
- ③ Manage: head-office와 product 사이의 relationship이다. 본사에서는 제품을 관리한다. 본사의 직원 중 marketer는 제품의 sales, season, category, 기간별 매출액 등을 관리한다.
- ④ Made: product와 manufacturer 사이의 relationship이다. 제품은 제조사에서 만들어진 다. 이 relationship은 제조일이라는 자체적인 속성을 가진다.
- ⑤ Supply: manufacturer과 warehouse 사이의 relationship이다. 제조사에서는 warehouse에 제품을 공급한다. 이 relationship은 제품이 얼마나 여러 번 공급되었는지 알 수 있도록 reorder하는 횟수와 날짜(reorder_date, reorder_num)라는 자체적인 속성을 가진다.
- ⑥ Online-amount: product와 warehouse 사이의 relationship이다. 온라인으로 판매된 제품의 수량이 warehouse에 전달된다.
- ⑦ Send: warehouse와 shipper 사이의 관계이다. warehouse는 online-amount relationship

을 통해 온라인 판매 제품의 수량을 알고 이를 shipper에게 배송할 제품을 넘길 수 있다.

- ⑧ Replenish: warehouse와 store 사이의 relationship이다. store에서 품절된 제품은 warehouse에서 보충해주어야 한다.
- ⑨ Sell: store와 customer 사이의 relationship이다. store에서는 customer에게 제품을 판매한다.
- ⑩ Ship: shipper와 customer 사이의 relationship이다. Customer는 shipment에 적힌 shipper로부터 제품을 배송 받는다. (배송 정보 역시 제공받을 수 있다.)
- ⑪ Tracking: 회사와 shipper 사이의 relationship이다. 회사는 각 배송마다 tracking_num 을 저장해서 current_state를 확인하여 해당 배송 정보를 추적할 수 있다. 이를 추적하여 배송에 문제가 있을 때 대체품을 발송하도록 하는 등 배송을 관리한다.

2. Schema diagram

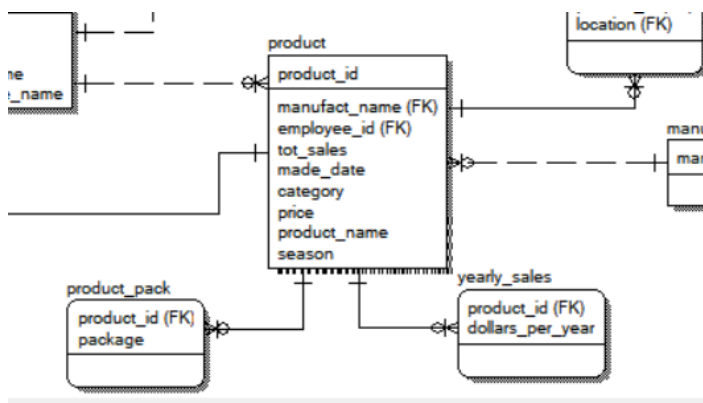


<그림2: Schema diagram>

Many-to-many relationship인 경우에는 별도로 schema를 생성하고, 그렇지 않은 경우에는 many 쪽에 one의 primary_key와 relationship 자체의 속성을 추가한다. 또한, multivalued 속성인 경우에는 원래 entity set의 primary_key와 해당 속성으로 구성된 새로운 entity set을 만들어준다. 만약, 새로 만든 entity set이 원래의 entity set을 포함하게 된다면, 새로 만들지 않는다.

- ① Buy: customer와 product의 primary_key인 customer_id, product_id를 primary_key로 가지고, 해당 relationship의 자체 속성인 date, on-off, region을 추가적인 속성으로 가진 schema가 만들어진다.
- ② Online-amount: product와 warehouse의 primary_key로 가진 schema가 만들어진다.
- ③ Send: warehouse와 shipper의 primary_key로 가진 schema가 만들어진다.
- ④ Replenish: warehouse와 store의 primary_key로 가진 schema가 만들어진다.
- ⑤ Sell: customer와 store의 primary_key로 가진 schema가 만들어진다.
- ⑥ Supply: manufacturer과 warehouse의 primary_key로 가지고, Relationship의 자체적인 속성인 reorder_date과 reorder_num을 추가적인 속성으로 가진 schema가 만들어진다.

이 외의 mange, made, tracking, ship, pay의 경우에는 각각 many쪽의 one의 속성과 relationship 자체의 속성을 추가한다.



<그림3: product_pack과 yearly_sales>

Product의 속성 중 Package와 dollars_sold_per_year은 multivalued 속성이므로 product_id와 각 relationship의 속성을 primary_key로 가지는 새로운 schema를 각각 생성한다.

Store의 out-of-stock과 warehouse의 low-of-stock은 multivalued 속성이지만 따로 schema를 생성하지 않는다.