

데이터베이스시스템 project 1 보고서

20190415 이규현

1. 명세서 분석을 통해 entity와 attribute를 결정하였다.

필요한 entity와 attribute는 아래와 같다.

- customer : 전자제품의 온라인판매와 오프라인 판매를 동시에 하는 매장에 적용시키기 위해 구매자의 정보를 저장할 수 있어야 한다. primary key는 id에 해당하는 customer_id를 두었다. 속성으로는 누적 구매수량을 나타내는 cumulative_quantity와 누적소비액을 나타내는 cumulative_dollar를 두었다.

-product : 제품을 지칭하는 제품번호를 primary key 키로 두고 명세서의 요구대로 type manufacturer, quantity_store, quantity_warehouse (오프라인 매장의 수량, 온라인창고의 수량)을 속성으로 갖도록하였다.

-store: 오프라인 매장을 지칭하는 엔티티로 매장번호를 primary key로 갖고 매장 지역을 속성으로 갖는다.

-warehouse: 창고를 지칭하는 엔티티로 오프라인매장에 재고가 떨어지면 보충해주고 온라인 주문이 들어오면 구매자에게 배송한다. primary key로 공장번호를 갖고 공장지역을 속성으로 갖는다.

-manufacturer: 제조사를 나타내는 엔티티로, 공장에 해당 제조사 제품이 떨어지면 다시 주문해서 수량을 채워주는 역할을 한다. primary key로 제조사 번호인 manufacturer_id를 갖는다.

- on_bill : 온라인 구매와 오프라인 매장을 구별하기 위해 온라인 매장의 경우 on bill을 두어 경우를 나누었고, frequent 고객이면 account_num을 적고 on_card_num은 null로 두어 구별하도록 한다. infrequent 고객이면 card_num은 적고, account_num을 null로 기재한다. 때문에 primary key는 on_card_num과 on_account_num 으로 두었다

- off_bill : on_bill과 반대의 엔티티로 온라인 구매와 오프라인 매장을 구별하기 위해 오프라인 매장의 경우 off bill을 두어 경우를 나누었고, frequent 고객이면 account_num을 적고 on_card_num은 null로 두어 구별하도록 한다. infrequent 고객이면 card_num은 적고, account_num을 null로 작성해야한다. 때문에 primary key는 off_card_num과 off_account_num 으로 두었다.

-

- sale은 마케터가 보는 판매내역을 나타내는 엔티티로 온라인매장과 오프라인 매장의 구매기

록을 구분하기 위해 오프라인 판매정보를 지칭한다. sale_id 를 primary key로 나타내었고, 물품의 누적판매량, time_period, store_region, sale_season을 속성으로 갖는다.

- online sale: 은 마케터가 보는 판매내역을 나타내는 엔티티로 온라인매장과 오프라인 매장의 구매기록을 구분하기 위해 온라인 판매정보를 지칭한다. online_sale_id 를 primary key로 나타내었고, 물품의 on_cumulative_sale_quantity와 판매기간을 나타내는 time_period, 오프라인매장의 지역 store_region과 판매 계절을 나타내는 sale_season을 속성으로 갖는다. 온라인 배송의 경우 추적번호를 갖고 있어야 하므로 구매자를 식별할 수 있도록 online_sale_id와 tracking_num이 primary key가 된다.

- package: 명세서의 내용에 의해 product의 조합으로 패키지 엔티티를 설정하였다. primary key는 package_id이다.

- shipper : 창고에서 배송지로 보내는 운송업 회사를 지칭하며 primary key로 운송회사 번호를 갖고 있고 파손여부와 예상도착시간을 속성으로 갖도록 설계하였다.

- company : electronic vendor으로 공장을 소유하고 있는 이 데이터베이스의 사용자로 규정하는데 한 개의 회사로 가정한다. primary key 로 company_id를 갖는다.

2. 명세서 분석을 통해 relationship을 결정하였다.

- buy : customer와 product의 binary relationship으로 제품을 구매하는데 제품을 구매자가 여러개 구입가능하고, 한 제품을 고객 여러명이 주문 가능하므로 many to many 의 relationship을 갖는다.

- product_package : product와 package의 binary relationship 으로 명세서의 내용에 의해 product의 조합으로 패키지가 판매될 경우를 고려하여 관계를 설정하였다. 여러 제품이 모여 패키지를 이루므로 many to one의 relationship을 갖는다.

- customer_off_bill : 구매자가 오프라인으로 구매하는 경우를 고려하여 customer와 off_bill의 binary relationship 으로 relationship을 설정하였다. 한명의 사람이 여러번 구매가 가능하므로 one to many의 relationship을 갖는다.

- customer_on_bill : 구매자가 온라인으로 구매하는 경우를 고려하여 customer와 on_bill의 binary relationship 으로 relationship을 설정하였다. 마찬가지로 한명의 사람이 여러번 구매가 가능하므로 one to many의 relationship을 갖는다.

- product_sale : 오프라인으로 구매행위가 이루어지면 판매정보를 저장해야하는 행동이 일어나므로 여러 가지 제품을 한번에 판매할 수도 있고, 한 제품을 여러번 구매하는 행위도 일어날 수 있으므로 many to many 의 relationship을 갖는다.

-on_product_sale : 위와 반대로 온라인으로 구매행위가 이루어지면 판매정보를 저장해야하는 행동이 일어나므로 여러 가지 제품을 한번에 판매할 수도 있고, 한 제품을 여러번 구매하는 행위도 일어날 수 있으므로 many to many 의 relationship을 갖는다.

-product_sale : 오프라인으로 구매행위가 이루어지면 판매정보를 저장해야하는 행동이 일어나므로 여러 가지 제품을 한번에 판매할 수도 있고, 한 제품을 여러번 구매하는 행위도 일어날 수 있으므로 many to many 의 relationship을 갖는다.

-sale_on_bill : on_bill과 online_sale의 binary relationship으로 여러 고객이 동일한 판매자에 의해 판매될 수도 있고 여러 판매자가 한 개인에게 물건을 판매할수도 있으므로 many to many 의 relationship을 갖는다.

-sale_off_bill : off_bill과 sale의 binary relationship으로 여러 고객이 동일한 판매자에 의해 판매될 수도 있고 여러 판매자가 한 개인에게 물건을 판매할수도 있으므로 many to many 의 relationship을 갖는다.

-online shipper : online_sale과 shipper의 binary relationship으로 일대일 대응하는 관계이지만 여러 온라인판매가 이루어지면 하나의 배송이 시작되는 패키지의 경우 포함하면 many to one의 relationship을 갖는다.

-store_sale : sale과 store의 binary relationship으로 여러 판매자가 동일한 매장에 속해 판매될 수도 있고 여러 매장이 한 판매자에 의해 판매될 수도 있으므로 many to many 의 relationship을 갖는다.

-online_warehouse: online_sale과 warehouse의 binary relationship으로 여러 창고에서 같은 판매자에게 판매될 수도 있고, 여러판매자가 하나의 창고에 속한 제품을 판매할 수도 있으므로 many to many 의 relationship을 갖는다.

-send: shipper와 warehouse의 binary relationship으로 여러 창고에서 하나의 배송을 할 수도 있고(패키지의 경우), 하나의 창고에서 여러 배송이 요청될 수도 있으므로 many to many 의 relationship을 갖는다.

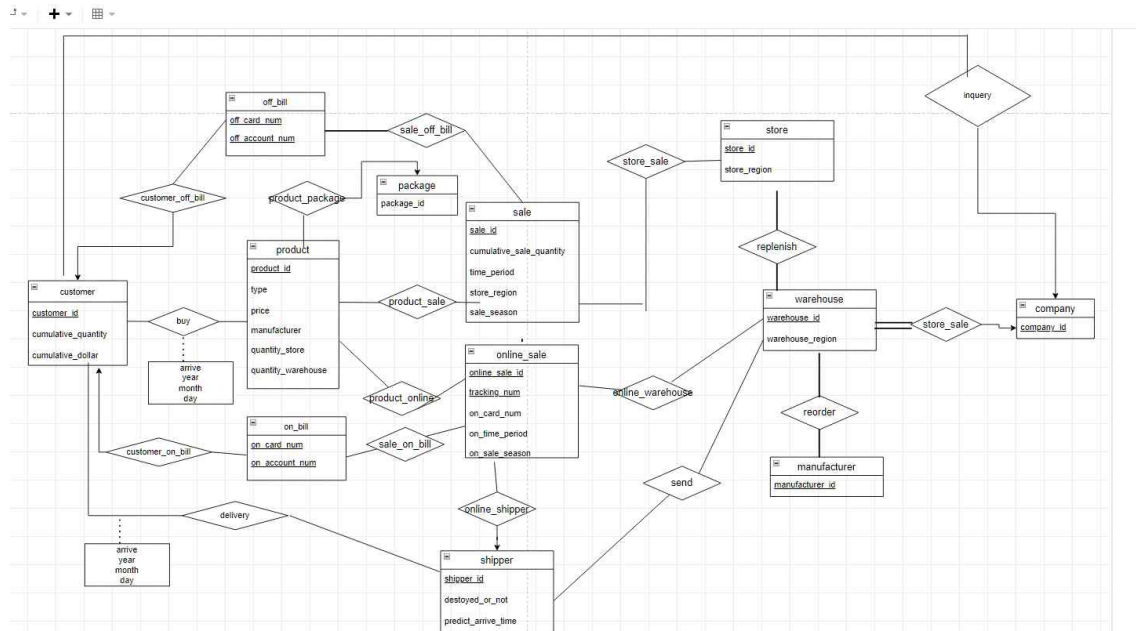
reoder : warehouse와 manufacturer의 binary relationship으로 여러 창고에 동일한 제조사에서 물량을 채워줄수도 있고, 여러 제조사가 하나의 창고에 재고 보충을 해 줄수도 있으므로 many to many 의 relationship을 갖는다.

own: warehouse 와 company 의 binary relationship으로 앞에서 회사인 company는 하나로 가정을 하고 여러 창고를 가질 수 있으므로 one to many 의 relationship을 갖는다.

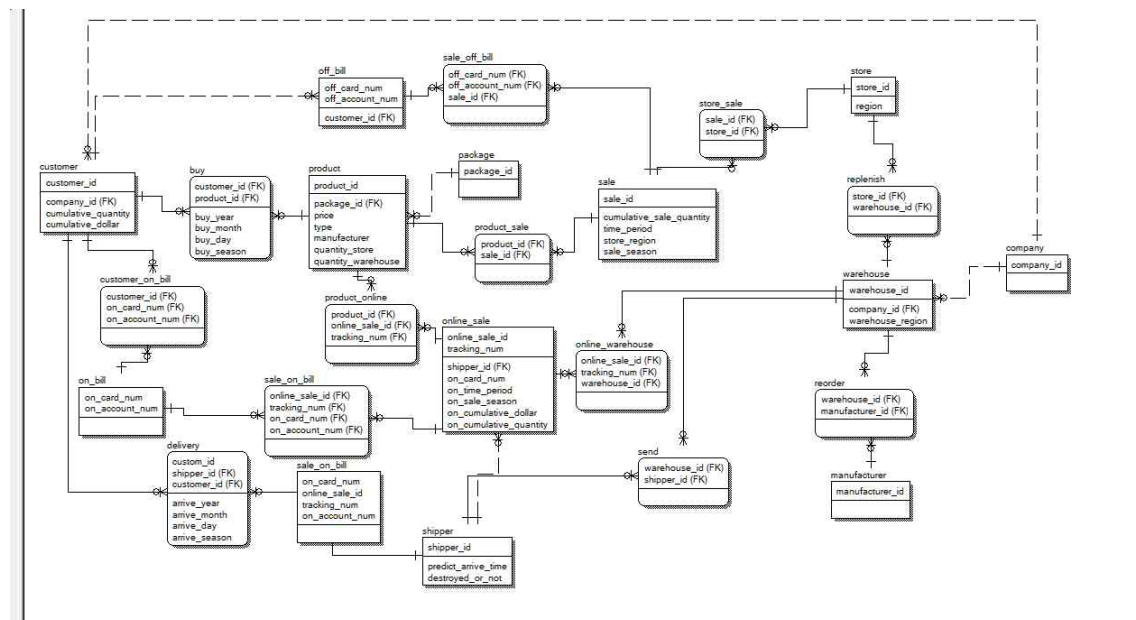
delivery : customer와 shipper의 binary relationship으로 동일한 배송업체가 한 구매자에게 배송을 올 수도 있고, 여러 개인이 한 배송업체로부터 물품을 받을 수도 있으므로 many to many 의 relationship을 갖는다.

y to many 의 relationship을 갖는다.

ER diagram



realtion schema



3.Query들을 요청이 적용되었는가..?

- Assume the package shipped by USPS with tracking number 123456 is reported to have been destroyed in an accident. Find the contact information for the customer. Also, find the contents of that shipment and create a new shipment of replacement items.
 - 위의 요청에 의해 shipper라는 엔티티와 many to one의 관계인 online_sale 엔티티 내부에 내용물이 무엇인지 알수있도록 content라는 속성을 갖게 하였고, 배송마다 tracking_number를 primary key 로 두었다. 상품이 파손된 경우 customer가 company에 재배송을 요청할 수 있도록 inquiry 라는 relationship을 설정하였다.
- Find the customer who has bought the most (by price) in the past year.
 - customer의 속성에 누적 소비금액을 (cumulative_quantity)을 두고 온라인 구매인 delivery와 오프라인 구매인 buy relation에 소비 날짜를 담아 작년 소비금액이 개인별로 확인가능하다.
- Find the top 2 products by dollar-amount sold in the past year.
 - 오프라인 구매 판매를 나타내는 sale의 속성에 누적 판매액과 온라인 구매를 나타내는 online_sale 엔티티에도 마찬가지로 누적판매액을 기록하여, time_period 속성을 이용하여 작년 가장 판매금액이 큰 제품 top2 를 구할 수 있다.
- Find the top 2 products by unit sales in the past year.
 - 오프라인 구매 판매를 나타내는 sale의 속성에 누적 판매량과 온라인 구매를 나타내는 online_sale 엔티티에도 마찬가지로 누적판매량을 기록하여, time_period 속성을 이용하여 작년 가장 많이 팔린 제품을 구할 수 있다.
- Find those products that are out-of-stock at every store in California.
 - 오프라인 매장을 나타내는 store의 속성의 region과 product 의 store 수량을 이용하여 캘리포니아에서 매진된 물품을 알 수 있다.
- Find those packages that were not delivered within the promised time.
 - shipper 엔티티에서 predict_arrive_time을 주고 relationship delivery 안의 실제 도착 날짜를 속성으로 두어 늦은 패키지 배송을 찾아낼 수 있다.
- Generate the bill for each customer for the past month.
 - relationship buy와 delivery에서 구매하는 시점이 다 기록되므로 지난달 소비내역이 확인가능하다.