

## 1. 판매데이터

가장 중요한 entity로써 판매가 될 시에 모든 정보를 다 기입해놨고, 다른 entity에서 참조할 만한 외래키를 많이 가지고있는 개체이다. 이 개체를 참조해서, 지난 1년간 수익이라던가, 해당 기간에서 가장 많이 팔린 개체라던가, 특정 ID의 고객이 어떤 소비를 했는지 알 수 있다. 여기서 가장 많이 고민한 것은, 어떻게하면 한 판매번호에 여러개의 제품을 산 목록이 뜨게 할 수 있을까였는데, 그냥 한 판매번호의 나열식으로 늘어놓는 수밖에 생각이 나지않았다. 아마 익숙하지 않아서 그럴 것이다. 그래서 어쩔수없이 pk를 위해 index를 둘 수 밖에 없었고 가독성적인 부분에서 조금 부족함을 나중에 깨달았다.

## 2. 제품정보

해당 제품 id가 어떤 제품을 가리키는지, 어디것인지를 나타내주는 정보가 들어있다. 판매데이터 뿐 아니라 다른 제품 id의 속성을 가진 entity는 참조를 할 수 있다.

## 3. 고객정보

고객의 성명, 연락처, 기타정보(필요하다면 vip등의 표시)등을 할 수 있는 entity이다. 기본적으로 판매데이터와 연결시켜놓았지만 다른 고객ID를 속성으로 가진 entity도 참조할 수 있다. (고객id가 fk가 될 것)

## 4. 재고정보

재고 정보는 판매 데이터를 바탕으로 재고를 계산한후 최소 재고에서 부족하다면 재주문 필요성에서 표시를 해준뒤, 해당 부분의 주문정보를 표시한다.(그러므로 주문정보는 null값이 될 수 있다)

## 5. 주문서

주문서 청구를 하면 제품 id와 수량이 제공되며, 도착여부 조사한다. 언젠가 리턴해주기 위해서.

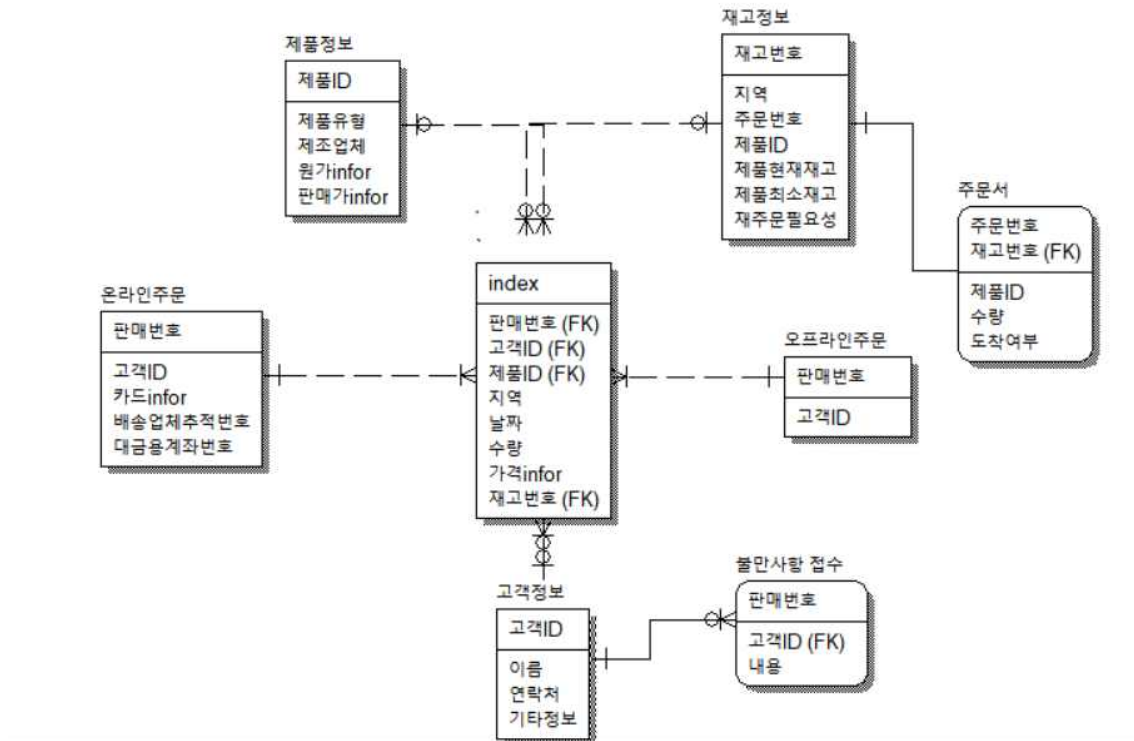
## 6. 온라인주문

온라인 주문은 해당 고객 id와 판매번호, 대금용계좌번호를 미리 넣어주고(null값이 될수있음

해당 값이 null이면 계좌사용하는 고객들), 그 외 고객들의 카드 번호를 저장한다.

## 7. 오프라인주문

오프라인주문은 카드번호도 기억하지 않기 때문에 고객id와 판매번호만 받아주었다.



하나가 추가되었음. 불만사항 접수라는 entity.

1. 온라인주문과 오프라인주문은 판매데이터 테이블에 대해서 하나의 판매번호에 많은 제품ID가 기록될것이기 때문에 1대다이며, optional하지 않아서 저렇게 표현을 해주었다. 만약 판매번호가 pk였다면 의존성있는 관계가 만들어졌을 것이다.
2. 고객정보 또한 판매데이터 테이블에 대해서 1대 다이며, 해당 고객이 산 제품이나 가격들을 판단해서 고객의 등급을 매길수도있을 것이다. 불만사항 접수는 판매번호에 귀속되고 하나의 고객에 많은 불만사항이 있을수있으며 고객정보가 없어지면 동시에 없어지는 개체라서 identify하다.
3. 제품정보는 판매데이터와 분리된다. 재고정보도 판매데이터와 분리된다. 그리고 재고정보가 판매데이터를 참고하기 때문에 1대 다로 처리했다.
4. 주문서는 재고정보에 귀속된다. 주문정보를 받을 때 재고정보도 같이 받아서 identify하게 생성하였다.

## 쿼리

- 추적 번호 123456과 함께 USPS에 의해 발송된 패키지가 사고로 파괴되었다고 보고된다고 가정합니다. 고객의 연락처 정보를 찾습니다. 또한 해당 발송의 내용을 찾아 교체 품목의 발송을 새로 만듭니다.
- >추적번호로 온라인주문 테이블에서 고객 ID를 찾은뒤, 판매데이터와 고객정보를 조회해

서 연락처 정보를 찾고, 판매데이터에 적혀있는 제품정보를 토대로 교체 품목을 만든다

- 지난 1년 동안 (가격별로) 가장 많이 산 고객을 찾습니다.  
-> 판매데이터에서 1년 기간을 정한뒤 1년동안 가격의 합이 가장 높은 고객 ID를 찾으려면 된다.
- 지난 1년간 판매된 금액별 상위 2개 제품을 찾아보십시오.  
-> 판매데이터에서 1년 기간을 정한뒤 1년동안 가격의 합이 가장 높은 제품 ID
- 지난 1년 단위 매출액 상위 2개 제품을 찾아보세요.  
-> 판매데이터에서 1년 기간 동안 팔린 제품갯수와 제품정보의 판매가-원가 한 값을 곱해준 것을 비교해야한다.
- 캘리포니아의 모든 상점에서 품절된 제품들을 찾아보세요.  
-> 재고 테이블에서 지역으로 캘리포니아 설정된 현재 재고 0를 찾는다.
- 약속된 시간 내에 배달되지 않은 패키지를 찾습니다.  
-> 고객정보에서 불만사항 접수에서 찾는다
- 지난 달 동안 각 고객에 대한 청구서를 생성합니다.  
-> 온라인 주문 테이블에서 계좌번호가 null이 아닌 모든 고객 ID 각각에 대해 판매데이터에서 한달 기간동안 사용한 가격들을 합한 값을 청구하면 된다.

프로젝트를 하면서..

판매데이터 테이블이 대체키가 많이 없어서 관계성이 identify하지못한 관계가 많이 나왔다. 만드는 사람의 입장에서는 조금더 정리해서 identify하게 설정하고 싶었지만 조금 부족했다. Fk와 pk 설정하는 것을 계산하는 과정에서 많이 배웠고, 많은 수정이 들어갔다. 해당 erwin에서 관계성에 속성을 부여할 수 있는 방법을 찾지못해서 하나의 entity 불만사항 접수라는 부분을 따로 만들었고, 이부분에 있어서 배달이 늦는다던가 하는 불만사항이 나오면, 그부분을 쿼리를 통해 찾으면 교재의 쿼리 부분을 만족시킬수있다고생각했다. 그러니까, 판매자입장에서는 현실적으로 불만사항 접수로 받지않는 그런 정보들은 꼭 필요한 정보가 아니라고 판단할것이라고 생각했다.