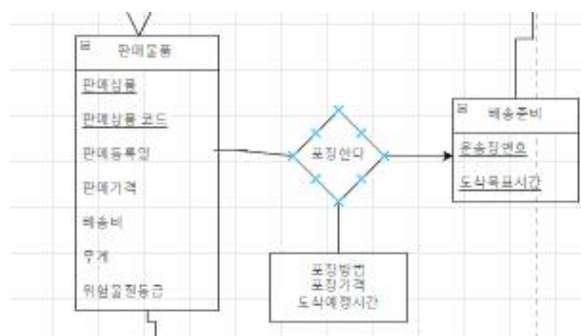


가장 기본적으로 생각한 모델은 많은 사람들이 쉽게 접하는 택배회사로, 예를 들어 cj대한통운 같은 회사이다. 물론 택배회사에서 다닌적은 없기에 정확한 정보일지는 모르겠지만, cj는 아무리 가까운 곳을 배송하더라도 먼저 배송기사가 선정되고 허브같은 보관지에 들른 이후, 분류 및 보관을 하고 그 이후 또 다른 배송기사가 선정되어 배달하는 순으로 진행되는 것으로 알고 있어 이를 바탕으로 er-모델링을 하였다. 또한 택배에는 반드시 운송장 번호가 등록되어 있고, 하나의 택배회사에서는 일단 절대로 동일한 운송장 번호를 쓰지않는다.(물론 동시에 시킨 물품에 대해선 같은 운송장 번호로 동시에 배송될 수는 있지만). 그럼 er-relationship의 엔티티와 관계들을 설명하도록 하겠다.



먼저 판매자 그리고 판매 상품, 구매자와의 관계이다. 이번 과제에서는 택배회사의 입장에서 생각하는 것이므로 구매자와 판매자에 대한 자세한 정보는 택배회사 관점에서 알필요가 없다. 즉 그냥 누가 어떤 물건을 어떻게 샀는지에 대해서만 알수 있게 relation을 만들었다. 판매자 이름, 주소, 회사명(개인판매자일시 null)과 판매자 ID로 unique하게 구분되는 판매자와 판매물품의 관계는 당연히 판매이다. 이때, 판매 물품은 무조건 판매자가 등록하는 것이고, 판매코드로 인해 unique해지기 때문에 그리고 한 판매자는 여러 물품을 판매할수 있기에 one to many관계이다. 이때 판매코드는 물건에 대해 가질수 있는 코드로 판매자에 따라 동일 물품에 대해선 unique한 코드를 가진다.(예를들어 컴퓨터_1, 컴퓨터_2 ..). 추가적인 속성으로 등록일, 가격, 배송비, 무게, 위험물질등급(숫자로 구분가능)을 속성으로 넣었다. 주소,이름, 번호, 통관부호(해외배송시, 국내일시 null),그리고 ID로 unidque하게 구분되는 구매자와 구매 관계는 한 판매자는 여러물품을, 혹은 하나의 물품은 여러 판매자에게 구입될수 있기 때문에 many to many관계이다. 이때 구매라는 relationship에 구매에 대한 정보 속성을 추가했는데, 수량, 결제 방법(신불, 카드, 후불 등등), 결제 확정일을 추가해 주었다.



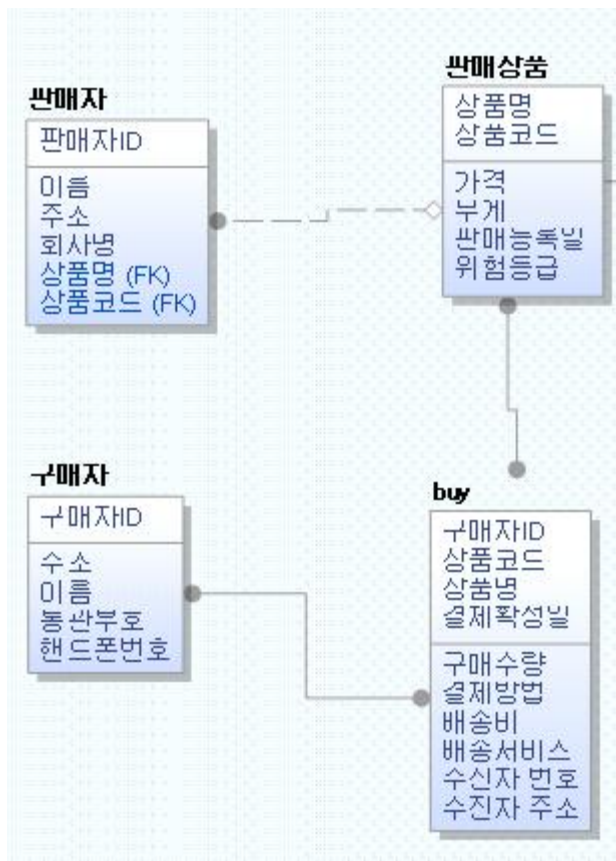
판매 상품은 판매 물품의 무게, 위험 등급에 따라 포장이 되는데, 이 포장이 되면 배송준비를 마치게 되는 것이다. 즉, 택배 회사에 등록을 한것이고 운송장번호가 등록이 되게된다. 이때부터 물품 추적이 가능하게 된다. 처음에 예시를 들었듯이, 한 운송장에 여러 물품이 들어있을수 있으므로 관계는 many to one이고 포장 relationship에는 약간의 포장방법에 대한 속성들을 추가해 주었다. 배송준비 entity는 운송장 번호를 가지고 있고, 이후 추가한 대부분 entity는 이 운송장 번호를 가져야만 unique해지는 weak entity들이다.

weak entity들을 설명하기 이전에 간단히 택배기사 entity를 추가해 주었다.

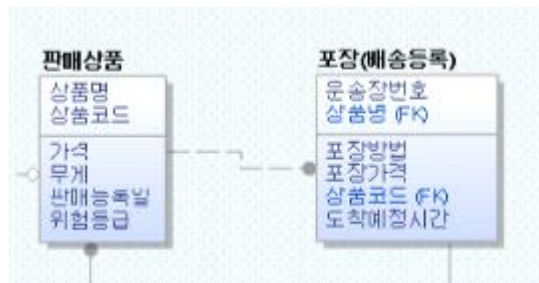
택배기사 entity는 이름, 핸드폰번호, 주소 경력 그리고 자신이 물고있는, 즉 배송하는 차(혹은 비행기)의 번호를 속성으로 가지고 있다. 이때, 동일한 차를 다른 기사가(예를들어 근무시간이 아니어서 그 차를 다른사람이 모는,...)물수도 있으므로 primary key는 차 번호, 핸드폰 번호로 두었다.

하게 구분이 되며, 분류를 하고, 다음 목적지를 계산하고, 그리고 다시 새로운 목적지에 따라 새로운 배송기사를 배정되게 된다. 그 관계를 표현한 것이다. 하나의 운송장번호도 여러번에 걸쳐 배송정보가 갱신되기에 many to many관계이고, 분류또한 하나의 운송장 정보를 가진 물품이 여러 허브에 도달할 수가 있다. 허브 또한 계속 새로운 배송기사와 함께 배송으로 이동한 many to many 관계이다. 다만 아래에 있는 주소, 도착날짜, 이름, 번호의 속성을 가진 수신자 entity는 하나의 운송장번호를 물품은 반드시 한명의 사람에게만 배송이 되므로, many to one관계이고, 수신자가 할수 있는 반품은 별도의 배송기사 배정없이 정확히 역순으로 흘러가기에 그냥 바로 배송정보와 1:1관계를 가지게 하였다. 한명이 여러개의 물품을 환불할수 있는 것은 맞지만, 수신자는 weak entity로 운송장번호로 구분이 된다. 즉, 운송장번호 하나당 한번의 환불이므로 1:1인 것이다.

이를 바탕으로 reduction한 결과는 다음과 같다.



먼저 판매자, 구매자, 상품, 구입의 관계를 나타내었다. 약간의 수정이 있기도 했지만, many to many의 관계였던 구매는 양쪽의 primary key를 가지게 설계했고, 판매자 entity는 FK를 가진 채로 판매 relationship에 대한 entity를 추가하지 않았다. 이유는 판매자 입장이 아니라 택배회사 입장이기 때문에, 물건을 등록하지도 않은 판매자는 알 필요가 없을 것 같기 때문이다.(즉, 여기에 있으면 그냥 무조건 물건을 판매자라고 생각을 했다.)



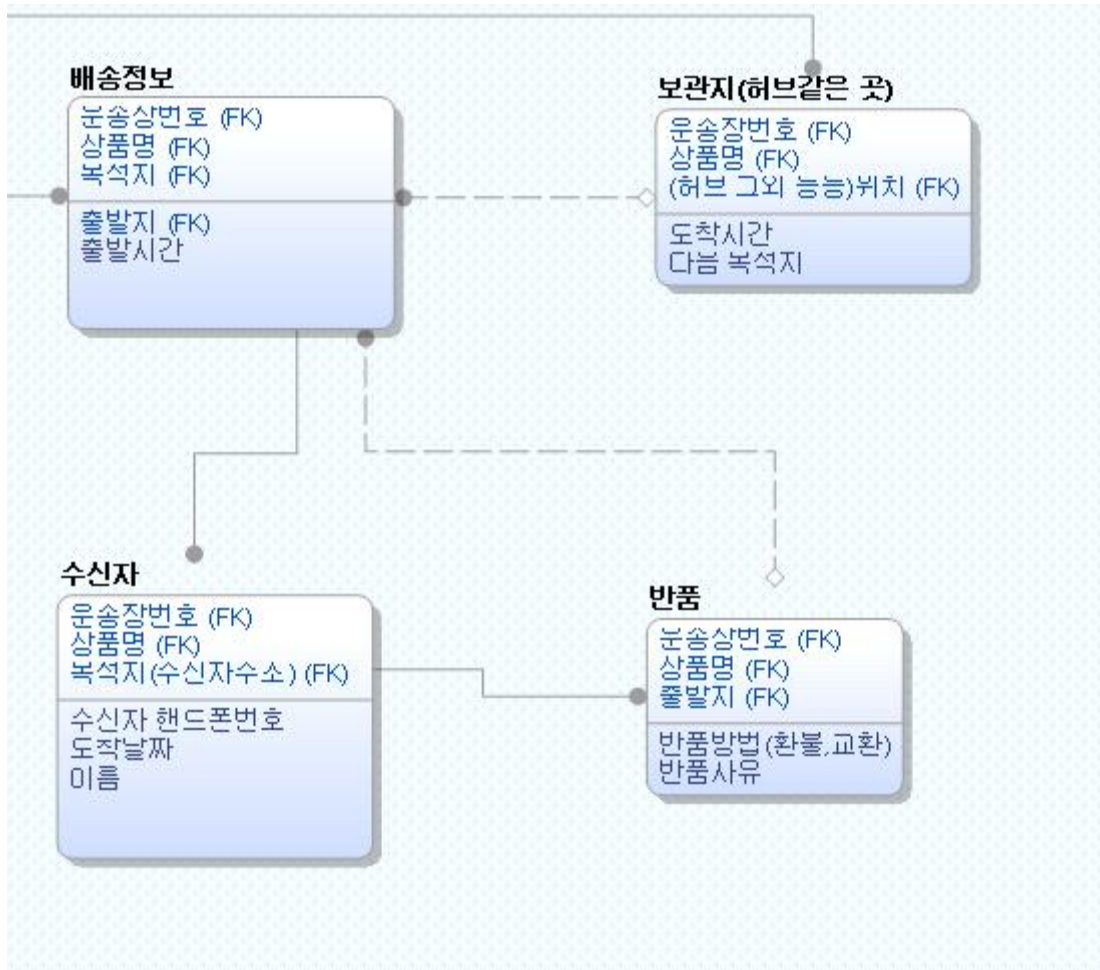
가장 달라진 부분은 이 부분이다. 포장과, 배송등록을 따로한 er-relationship과는 다르게 그냥 하나로 합쳤는데, 포장이 된다면 무조건 배송된다는 것이 아니고, 배송준비라 해도 배송된다는 말이 아니어서 두 entity가 똑같다고 판단이 들었기 때문이다. 이때 특정한 하나의 상품(상품명과 코드 때문에 unique 함)은 배송부터 끝까지 하나의 운송장정보만을 가지기 때문에, 한 운송장에는 여러개의 물품이 있을수도 있기 때문에, oneto many관계를 주었다. 이것을 제외하면 거의 비슷하다.



배송등록고, 배송 배정 그리고 배송기사간의 관계를 나타낸 db이다. 한명의 택배기사는 (수정). 먼저 포장에서 배정은 한 운송장당 단 한번 처음 겪는 배정이기 때문에(그 이후에선 보관지->배정) one to one관계이다. 택배 기사와 배정의 관계는, 한명의 배송기사가 여러물품을 배송하고, 그리고 배정도한 여러번 배송기사에게 배정되기 때문에 many to many관계이다. 이때 배정은 운송장번호, 상품명, 그리고 목적지를 가지고 unique하게 구분이 가능한 entity이다. (실제로 weak로 설계가 되었다.)



이후 목적지 운송장번호 상품명, 의 속성을 지닌 (예를 들어 서대문구가 목적지인 운송장번호 1234인 과자) 배송배정과 정확히 1:1로 배송정보는 매칭되며, 배송정보는 출발할 당시 갱신되기에 추가적인 속성으로 출발지와, 출발 시간이 있다. (만약 출발하지 않았다면 null인 것이다.). 또한 도착의 중간 개념이었던 보관지도 weak 였듯이, 운송장번호, 등의 속성을 primary key로 가지고 있고 추가적으로 도착 시간도 primary key로 가지고 있다. 이또한 한 물품(운송장과 목적지를 가진)에 대해 1:1로 매핑이 되므로 one to one이다. 주의할 점은 도착하지 않았을 경우에 상품tuple은 이 hub에 없다. 설계해야 될 db는 택배회사 입장이기 때문에, 굳이 hub에 도착 예정인것까지 고려하지 않은 것이다. 보관지에서 다시 배송배정또한 하나의 허브(옥천허브에서 다음 목적지가 부천인 운송장번호 1234인 과자는 unique)의 물품은 한명의 배송배정만 받을수 있으므로 이또한 1:1이다.



마지막으로 수신자와의 관계도 er-relationship과 유사하게 구성했는데, 당연히 택배회사의 db이므로 수신자도 weak entity이고, primary key로 운송장번호, 상품명, 주소를 primary key로 구분할수 있다. 보관지와 유사하게 수신자도 택배를 받지 못하면 그냥 db에 없다. 실제로 모든 운송장 번호를 따라가, buy entity에서 수신자를 받고, 그 수신자의 정보가 수신자 entity에 없다면, 택배가 없는 것으로 쉽게

판단도 가능하다. 받은 사람에 한해, 수신자는 반품을 요청할수 있고 이또한 relationship에서 설계했듯이 그대로 weak하게 설정하였다.

QUERY:

1. 만약 버스가 파손되었다면, 먼저 배송배정의 entity에서 어떤 물건들을 맡았는지 탐색을 먼저 해주면 그 버스가 맡았던 모든 물건들이 조회가 될 것이다. 이때, 파손된 시간을 알아야 하므로, 이 탐색된 운송장 번호를 바탕으로, 보관지, 수신자에게서 도착시간을 조회, 그리고 가장 최신일자 까지의 물품을 정상적으로 배송된 물건이라 판단하면 된다. 파손된 물건들은 배송정보 entity에서 그 시간 이후로 출발시간이 등록된 모든 물건들을 파손이라 판단할 수 있다.
2. 단순히 수신자이름과 주소를 기준으로 grouping해서 작년 시간을 조건으로 count하고 그중 max를 찾아주면 찾을수 있다.
3. 2번과 마찬가지로 구매자의 ID를 기준으로 , 작년시간을 조건으로 sum하고 그중 max
4. 수신자의 모든 운송장번호를 탐색후, 그것을 기준으로 포장 entity에서 도착예정시간과 도착날짜가 다른 모든 사람을 count하면 찾을수 있다.
5. 배송 서비스, 환불 등 대부분의 서비스를 찾을수 있다