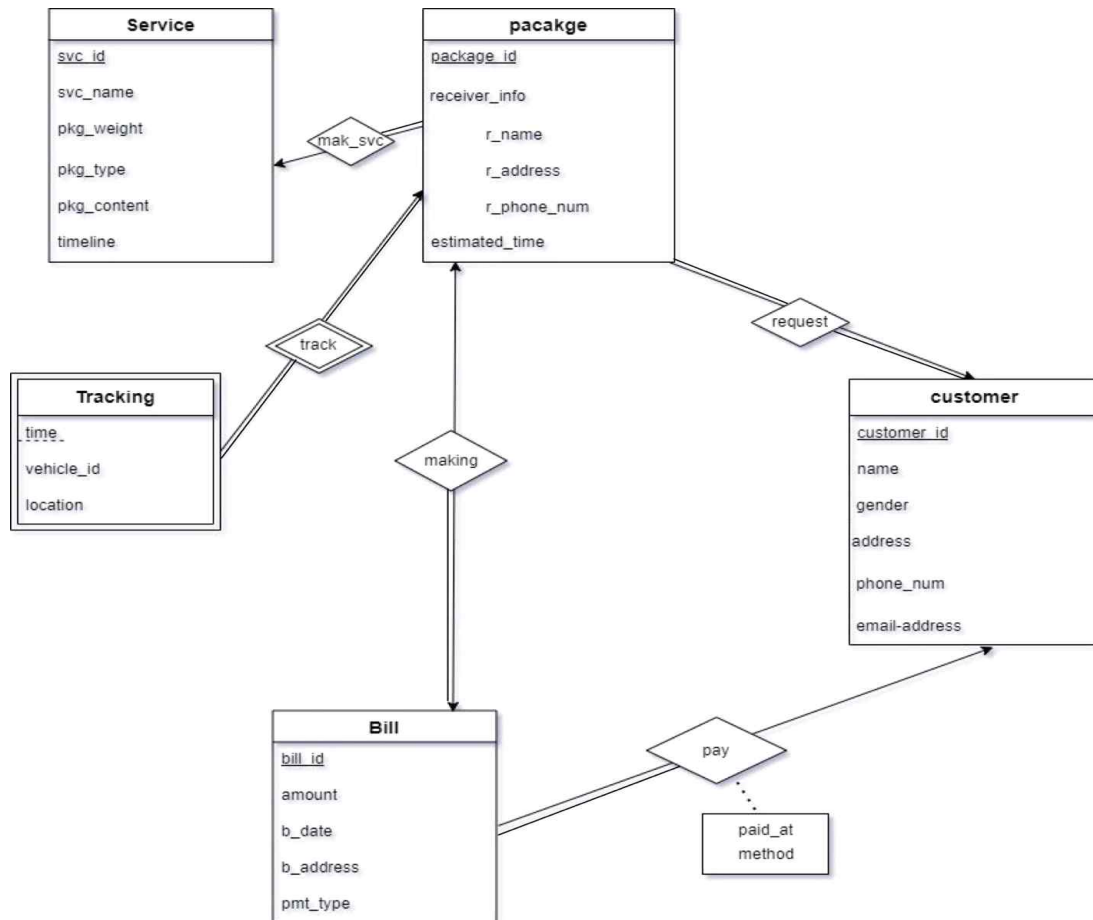


1. ER diagram



1. entity set을 설정한 logic

이번 프로젝트에서 고려해야 할 것은 고객(customer)이 요청한 택배를 수령인(receiver)에게 보내는 것, 이 과정에서 택배 추적 그리고 지난 달의 배송 서비스에 대한 bill을 작성하여 수령인에게 보내는 것입니다.

이를 고려한 ERD를 제작하기 위해서 가장 처음 설정한 entity는 배달 중인 택배의 정보를 담고 있는 package entity, 택배를 보낸(요청한) customer entity, 그리고 청구서에 대한 정보를 담고 있는 bill entity입니다.

package의 특성에 따라 서비스가 결정되고 bill에도 해당 서비스 type이 listing되기 때문에 service 개념이 중요하다고 생각되어 따로 service entity를 만들어 주었습니다. service entity는 배달 중인 물품에 대해 결정되는 것이므로 package entity와 관계를 맺고 있습니다 그리고 주요한 프로젝트 요구사항 중 하나는 추적이므로 이를 위한 tracking entity를 만들어 주었습니다. tracking도 service와 마찬가지로 배달 중인 물품에 대한 추적이므로 package entity와 관계를 맺고 있습니다.

2. detail of entity, attribute and relationship

1. package entity

배송 중인 택배에 대한 정보를 담고 있는 entity입니다. 이번 프로젝트에서 모든 entity들과 관계를 맺고 있는 가장 중요한 entity입니다.

① relationship

모든 entity와 관계를 맺고 있으므로 다음 entity들을 설명하면서 relationship을 설명하겠습니다.

② attribute

-package_id : package entity의 PK로 소위 운송장 번호와 같은 개념입니다. 이는 보내는 물품마다 고유합니다. customer로부터 shipping company에 물품이 도착하면 바로 package_id가 발급된다고 가정했습니다.

-receiver_info : component attributes r_name, r_address, r_phone_num을 가지는 composite attribute입니다. 이는 수령인에 대한 정보를 담고 있습니다. r_name, r_address, r_phone_num은 각각 수령인의 이름, 수령할 주소, 수령인의 전화번호를 의미합니다. 택배는 주문한 이 즉, customer가 같더라도 택배의 배송지는 다를 수 있습니다. 그러므로 수령인의 정보는 꼭 포함되어야 할 attribute입니다. 택배를 주문한(보내는)이는 customer entity로 설정하여 relationship을 만들어주었으므로 attribute로 추가하지 않아도 됩니다.

-estimated_time : 해당 택배가 배송될 것으로 추정되는 시간입니다. 수행해야 할 queries 중 예상된 시간에 도착하였는지 판단하는 문제가 있었습니다. 이를 위해 추가된 attribute입니다. 예상된 시간에 도착하였는지는 estimated_time 이외에도 실제 도착 시간이 필요한데 이는 후에 설명할 tracking을 이용하면 구할 수 있습니다.

2. Tracking entity

이번 프로젝트 주요한 이슈, query였던 tracking을 수행하기 위한 entity입니다. tracking은 우리가 흔히 택배를 시키면 시간에 따라 어디에 도착했는지를 알려주는 것입니다. 고객은 이를 통해 택배의 위치를 알 수 있고 더 나아가, 운송 중 운송수단에 사고가 발생하더라도 마지막으로 안전하게 배송된 시간과 장소 그리고 해당 운송수단엔 어떤 물품들이 배송 중이였는지를 알 수 있습니다.

해당 entity를 통해 배송 중 사고가 나더라도 해당 운송수단에 들어있는 물품을 알아내고 마지막으로 안전하게 배송된 장소가 어딘지 알아내는 query를 수행할 수 있습니다.

① relationship

Tracking은 package에 대해서 수행 되어야 하는 것이므로 package entity와 track관계를 맺습니다. tracking은 특정 package에 대해 원하는 정보를 쉽게 검색할 수 있어야 합니다. 그렇기 때문에 tracking entity는 package_id를 PK로 가져야 합니다. 즉, tracking entity는 identifying entity인 package entity에 종속된 weak entity가 됩니다. tracking entity는 한 package_id에 대해 여러 시간대 별로 존재하기 때문에 package_id만으로는 unique하지 않습니다. 때문에 time이라는 attribute를 식별자로 가집니다. 또한 customer로부터 shipping

company에 도착한 시점부터 추적이 시작된다고 가정하였습니다.

한 package가 shipping company에 도착하고 receiver에게 도착할 때까지 시간별로 추적이 되므로 package entity와 tracking entity는 one to many 관계를 맺고 있습니다. 또한, shipping company에 도착하면 tracking이 시작된다고 가정했으므로 package entity, tracking entity 모두 total로 관계에 참여하고 있습니다.

② attribute

-**time** : tracking이 되고있는 시간입니다. 날짜부터 초단위까지 담고있습니다.

-**vehicle id** : 운송수단의 id입니다. 운송수단의 종류(트럭, 비행기등)와 관계없이 id는 unique 하다고 설정했습니다. 예를들어 트럭 1754가 존재하면 비행기 1754는 존재할 수 없습니다. 때문에 id만 알아도 운송수단의 종류와 해당 운송수단이 무엇인지 알 수 있습니다.

-**location** : 해당 package와 time에 운송되고 있는 장소입니다. shipping company에 도착한 후부터 추적이 시작되고 package는 company에 도착하자마자 package_id가 생성된다고 가정하였기 때문에 해당 시간에 택배가 창고에 있는지, 도착을 하였는지 등을 알 수 있습니다.

3. service

service를 결정 짓는 entity입니다.

① relationship

서비스는 배송되는 물품에대해서 생성되는 것이므로 package entity와 mak_svc 관계를 맺습니다. 여러 package가 한가지 service에 귀결될 수 있으므로 package entity와 service entity의 관계는 many to one의 관계입니다. 또한 모든 package는 service가 있어야 배송이 가능하므로 total로 해당 관계에 참여합니다. 하지만 service는 존재하지만 해당되는 package가 있을 수도 있으므로 partial로 참여합니다.

② attribute

-**svc_id** : 서비스의 고유번호로 PK입니다.

-**svc_name** : 서비스 이름입니다. 각 회사마다 창의력 있게 작명한 서비스 이름이 있기 때문에 attribute로 넣어주었습니다.

-**pkg_weight** : package의 무게로 서비스가 결정되는 요인 중 한가지입니다.

-**pkg_content** : package의 내용물로 서비스가 결정되는 요인 중 한가지입니다. 위험물품인지 국제배송인지 등의 여부를 담고있습니다.

-**pkg_type** : package의 포장 type으로 서비스가 결정되는 요인 중 한가지입니다.

-**timeline** : 해당 택배가 당일 발송되었는지 이틀 이내 발송되었는지 등에 관한 정보를 담고 있는 attribute입니다. 서비스가 결정되는 요인 중 한가지입니다.

4. Bill

청구서에 대한 entity입니다. 프로젝트 명세서에 따라 지난달에 package에 대한 bill을 회사가 고객에게 청구합니다. package 하나당 bill 하나가 생성된다고 가정하였고, 한 package 안에 여러 개의 물품에 포장되어있어도 이는 하나로 세아립니다.

① relationship

청구서는 package에 대해 생성되는 것이므로 package entity와 making 관계를 맺습니다. 앞서 언급한 대로 한 package에 대해 하나의 bill이 생성되므로 one to one의 관계를 맺습니다. 또한 지난달에 대한 bill을 생성하기 때문에 청구서가 발급되지 않은 package가 있을 수 있습니다. 때문에 package는 관계에 partial로 참여하고 bill은 total로 참여합니다.

bill은 customer에게 전달되어 customer가 이에 대한 돈을 지불하는 것이므로 pay의 관계로 연결되어 있습니다. 한 고객이 지난 달 여러개의 package를 주문해 여러 bill을 받을 수 있으므로 bill entity와 customer entity의 관계는 many to one의 관계입니다. 또한 모든 bill은 customer에게 전달이 되므로 bill은 total로 관계에 참여하지만 customer 중에는 이번달에 처음 주문을 하여 아직 bill이 생성되지 않은 경우가 있을 수 있습니다. 때문에 customer은 관계에 partial로 참여합니다. 청구서에 대한 지불시기나 지불 방법은 그때그때 다릅니다. 다시 말해 같은 고객이라도 bill에 따라 결제 방식이나, 날짜가 다를 수 있습니다. 그래서 pay의 attribute로 지불방식은 method와 지불시기인 paid_at을 담고 있습니다.

지난달 가장 많은 package를 order한 고객, 가장 많은 돈을 지불한 고객들을 알아내는 query를 수행할 수 있게 하는 entity입니다. 이때 시간 시간기준은 고객이 돈을 지불한 날짜 또는 청구서가 만들어진 날짜 중 자유롭게 수행할 수 있습니다.

② attribute

-**bill_id** : 각 청구서마다 있는 고유한 번호이며 PK입니다. package와 one to one의 관계를 맺고 있어 package_id만으로 unique하여 weak entity로 만들까 생각했지만, bill_id로 검색할 일이 잦을 거 같아 추가한 attribute입니다.

-**amount** : 지불해야 할 혹은 이미 지불된 돈을 담고 있습니다. 이미 지불된 돈은 영수증 개념으로 청구되는 것으로 이전에 customer와 company사이에 contract를 맺은 경우입니다.

-**b_date** : 청구서가 생성된 날짜입니다.

-**b_address** : 청구서가 보내질 주소입니다. 전자 청구서가 아닌 종이 청구서로 가정했습니다. 저희 어머니처럼 거주지의 주소로 bill을 보내지 않는 경우가 있으므로 청구서에 청구서가 보내질 주소 address attribute를 추가해 주었습니다.

-**pmt_type** : 선불 후불에 대한 attribute입니다. 앞서 말했던 선불의 경우 영수증의 개념으로 bill이 청구 됩니다.

5. customer entity

고객 entity입니다. 여기서 고객이란, package를 택배 회사에 요청한 사람을 의미합니다.

① relationship

customer는 shipper company에 package delivery를 요청한 사람이므로 package entity와 request 관계를 맺습니다. 이때 한 사람이 여러 package를 요청할 수 있으므로 delivery package와 customer는 many to one의 관계를 맺게 됩니다. 모든 package는 요청한 customer가 있으므로 total로 관계에 참여하게 되고 customer도 적어도 한 번 이상 package delivery를 요청했기 때문에 존재할 것이므로 total로 관계에 참여하게 됩니다.

customer는 bill entity와도 관계를 맺지만 앞서 설명하였기 때문에 생략하겠습니다.

② attribute

-**customer_id** : 주민등록번호나 회사고유번호와 같이 customer를 구별하는 pk입니다.

-**name** : customer의 이름입니다.

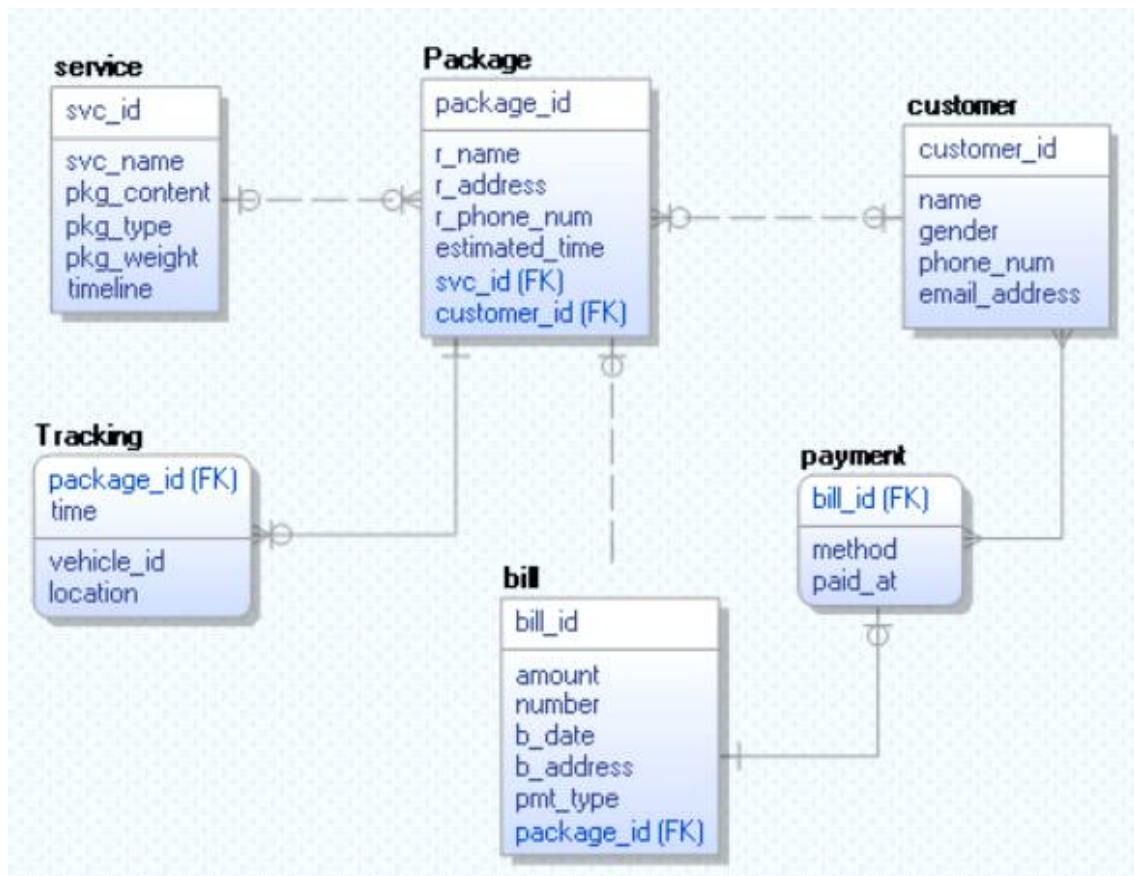
-**gender** : customer의 성별입니다.

-**address** : customer의 주소로 package가 반품될 때 사용될 주소입니다.

-**phone_num** : customer의 휴대폰 번호입니다.

-**email-address** : customer의 email 주소입니다.

2. Schema diagram



ERD에서 나오지 않았던 개념인 FK와 ERD에선 attribute였지만 새롭게 table로 구성된 payment을 위주로 설명하겠습니다.

1. Tracking은 Package entity에 종속된 weak entity입니다. 그렇기 때문에 package entity를 참조하게 되고 primary key인 package_id를 foreign key로 가집니다. FK인 package_id만으로는 tracking을 정의할 수 없으므로 package_id와 time을 PK로 가집니다.
2. payment는 ERD에서는 relationship의 attribute로 존재했던 entity입니다. bill을 참조하므로 bill의 PK(bill_id)를 FK로 가집니다. bill과 payment는 one to one이기 때문에 bill_id만으로 payment는 정의됩니다. 그렇기 때문에 bill_id를 PK로 가집니다.

3. 가장 많은 entity와 관계를 맺는 package는 service, customer과 many to one 관계를 가집니다. 그렇기 때문에 service의 PK(svc_id), customer의 PK(customer_id)를 FK로 가집니다.
4. bill은 package와 one to one와 관계를 맺고 이를 참조하므로 package의 PK(package_id)를 FK로 가집니다.