데이터베이스시스템
PROJECT1

심리학과

20190345 김동현

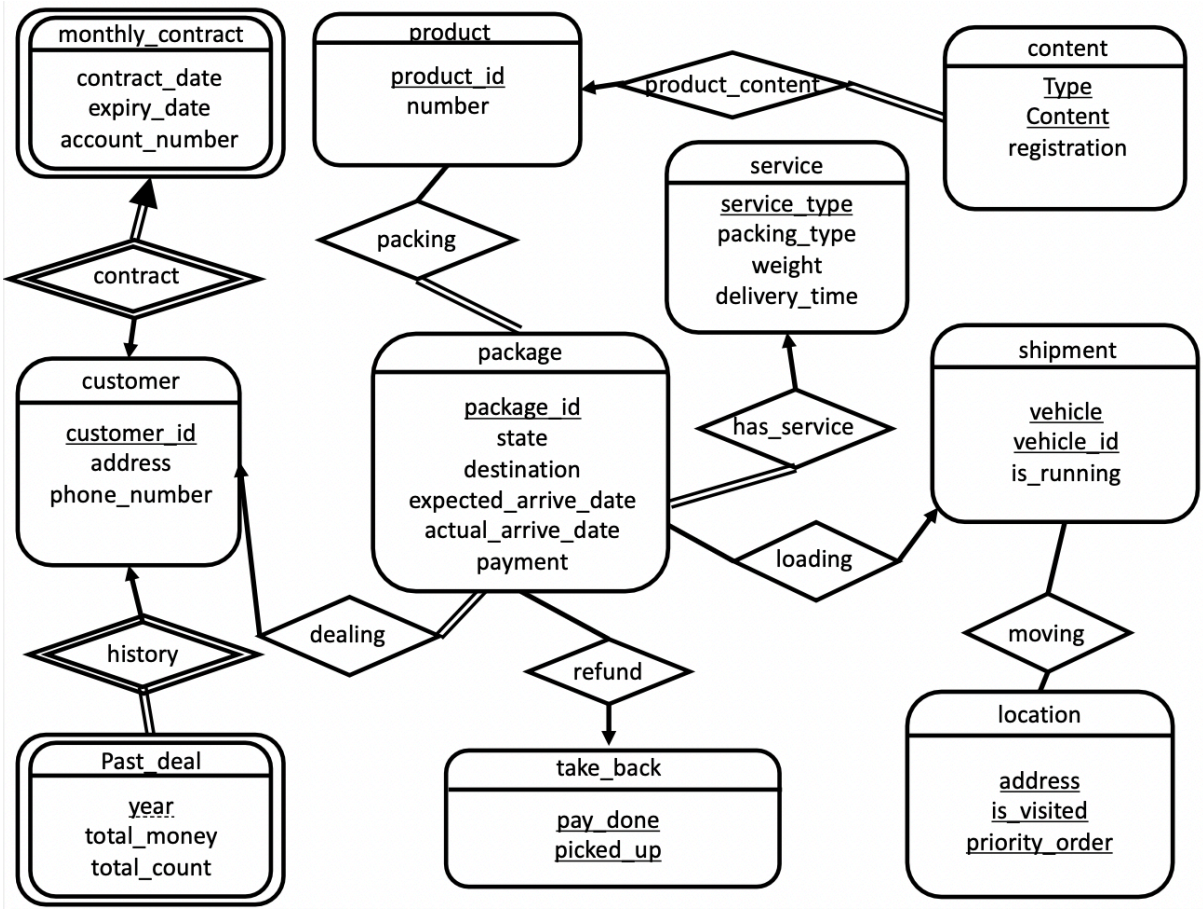
목차

1. 프로젝트 소개

해당 프로젝트는 가상의 Package Delivery 회사의 데이터 베이스 시스템을 구축하는 것을 목표로 한다. 특히 package의 처리와 package에 대한 청구 측면을 중심으로 데이터 베이스 시스템을 구축해야 한다. 이를 위해, package delivery 회사의 특성에 맞추어 entity set과 relationship set을 구성한 뒤, E-R diagram으로 표현한다. 또한 Erwin을 활용하여 E-R diagram을 relational schema diagram으로 reduction한다.

2. E-R diagram

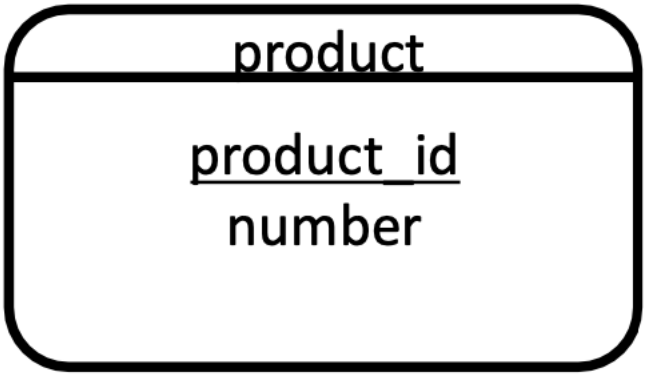
전체적인 E-R diagram의 모습이다. 각각의 entity set과 relationship set의 구성 및 관계에 대해 소개한다면 다음과 같다.



2.1 Entity set

Entity set의 구성에 대한 설명이다. 총 10개의 entity set이 있다.

2.1.1 Product



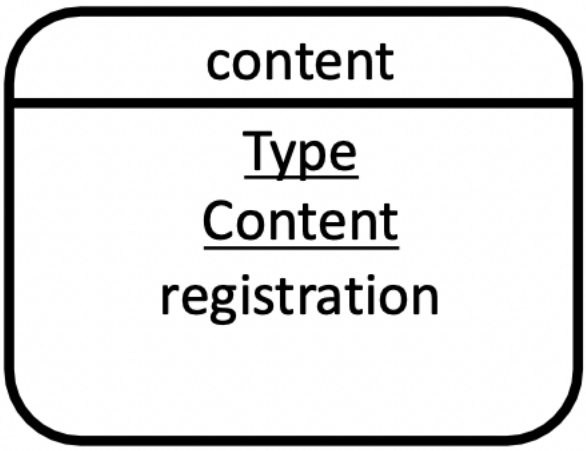
제품을 나타내는 entity set으로, 제품에 대한 정보를 담고 있다. 하나의 package에 여러 제품이 담겨 배송될 수 있다는 점을 고려하여 제품을 하나의 entity set으로 표현하였다. 이로 인해 package entity에 들어갈 제품을 나열할 필요없이, product entity를 통해 package에 어떤 제품이 들어갔는지, 총 몇개의 제품이 포함되어 있는지 나타낼 수 있다.

Product entity set의 attribute는 총 2가지이다. Product_id는 제품의 고유 아이디를 의미한다. 특정한 경우를 제외하고는 제품이 무엇인지 알 필요가 없기에 제품에 구체적인 정보를 모두 포함하지 않고 제품의 id로만 제품을 구별하고자 한다. Number는 해당 제품의 개수를 의미한다. Package를 구성할 때, 동일한 id의 제품이 여러 개 포장될 수 있으므로, 해당 제품의 개수를 나타낸다.

Product entity set의 primary key는 {product_id, number}로 구성된다. 제품 간 구별은 product_id로 가능하지만, package에 포함되는 개수에 따라 구별될 수 있도록, number도 primary key에 포함하였다

.

2.1.2 content



제품이 무엇인지를 나타내는 Entity set으로, 제품의 세부 정보를 담고 있다. 비록 package delivery 회사에서 어떤 물품을 배송하는지를 알 필요는 없지만, 위험물이나 해외로 배송되는 물품에 대해서는 택배 회사가 이에 대한 정보를 인지하고 있어야 한다.

content entity set의 attribute는 총 3가지이다. Type은 해당 product가 위험물 혹은 해외배송 물품인지를 나타낸다. Content는 위험물 혹은 해외배송 물품의 구체적인 정보를 나타낸다.

만약 위험물이라면, 해당 물품이 폭탄인지, 화학물품인지 등을 나타내게 된다. Registration은 해당 물품에 대한 적절한 신고 절차가 이루어졌는지를 확인하는 속성이다. 위험무에 대한 적법한 위험물 신고절차 혹은 해외배송 물품에 대한 관세/세금 신고 절차가 이루어졌는지 여부를 나타내 배송을 할 수 있는 물건인지를 확인한다.
content entity set의 primary key는 {type, content}로 구성된다. 제품의 정확한 구별이 필요하기 보다 택배회사가 알아야만 하는 항목인지(위험물 혹은 해외배송 상품인지), 알아야하는 항목이라면 어떠한 종류인지만(어느 이유로 해당 항목을 지정되었는지) 알면 되므로 두 attribute를 content entity set의 primary key로 설정하였다.

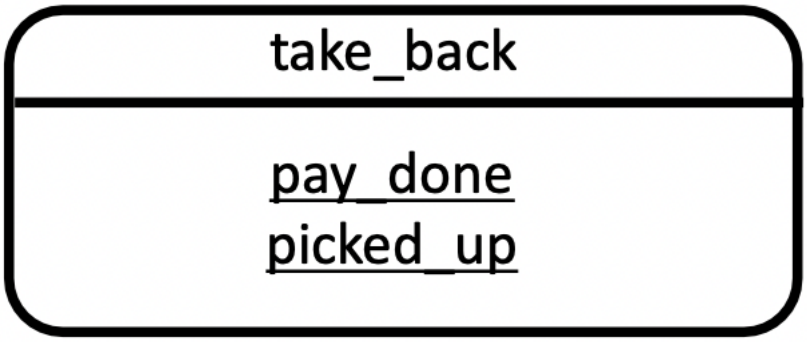
2.1.3 package

배송되는 택배 패키지에 대한 entity set으로, 실제로 배송되는 패키지 단위를 나타내고 있다. Product entity set과 product entity set을 구별한 이유는 하나의 패키지에 여러 개의 제품이 담길 수 있기 때문이다. 혹여나 패키지를 배송하는 과정에서 물품이 일부 분실되거나(사고로 인해), 일부 물품만 반품하는 경우에 대비하기 위해 두 entity set으로 분리하여 구성하였다. 또한 package entity set을 중심으로, 택배의 구성, 배송 추적, 고객에 대한 정보가 구성되기에 본 E-R model의 핵심적인 entity set이라 할 수 있다.
Package entity set은 총 6개의 attribute로 구성된다. Package_id는 현재 패키지의 고유 번호를 나타내며, 실제 택배 서비스를 이용하면 나오는 송장번호와 유사한 개념이다. State는 현재 패키지의 상태를 나타내기 위한 attribute로 창고에 있는 경우, 운송수단에 실려 배송중인 경우, 배송이 완료된 경우, 반품 절차를 진행 중인 경우를 나타낸다. 이는 다른 entity set과의 관계를 알아채기 쉽도록 설정한 속성이다. Destination은 해당 패키지가 이동해야할 장소를 나타낸다. Expected_arrive_date는 예상 도착 일자로 패키지의 서비스 타입에 따른 배송 기간으로부터 일자를 구할 수 있다. Actual_arrive_date는 실제 도착 일자로 패키지가 배송이 완료되기 전까지는 Null값을 가지다가 배송완료 후 해당 값이 설정된다.

이후 예상 도착 일자와 실제 도착 일자와의 값을 비교하여 패키지가 제시간에 도착했는지 여부를 확인할 수 있다. Payment는 현재 패키지를 배송하는데 필요한 배송비를 의미한다. 이는 패키지를 배송할 때에 드는 비용 뿐만 아니라 패키지를 반품하는 데 드는 비용 역시 포함한다.

Package entity set의 primary key는 {package_id}이다. Package_id가 패키지마다 고유한 값으로 초기화 되어 있어 해당 속성만 가지고 모든 패키지를 구분할 수 있다.

2.1.4 take_back

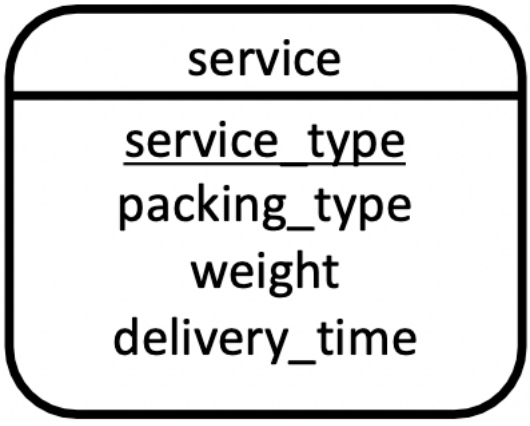


반품되는 패키지에 대한 entity set으로, 반품되는 패키지에 대한 정보를 담고 있다. Package 중에서도 반품되는 패키지를 따로 구별하기 위해 entity set을 따로 구성하였다. 반품되는 상품은 본래 배송된 패키지와 별개의 package entity로 표현된다. 그 이유는 패키지 배송 사실을 남기기 위해, 패키지의 id와 다른 새로운 package_id를 부여해야 하고, 패키지 안에 구성되어 있는 모든 제품이 반품의 대상이 아닐 수도 있기 때문이다.

Take_ back entity set은 총 2개의 attribute로 구성이 된다. Pay_done는 반품하는 패키지에 대한 반품비 결재 여부를 나타낸다. Picked_up은 정상적인 반품여부를 나타낸다. 패키지의 반품을 신청하였지만, 반품되는 물건이 없다거나, 잘못된 상품이 반품되는 경우를 대비하여 반품이 정상적으로 이루어졌는지를 나타낸다.

Take_back entity set의 primary key는 {pay_done, picked_up}이다. 이는 take_back entity set이 반품비 결재 여부와 반품 여부에 따라 모든 상태를 구분할 수 있기 때문이다.

2.1.5 service

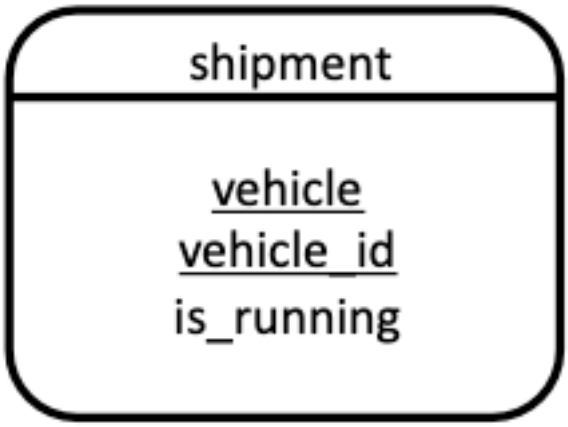


패키지에 적용되는 서비스의 종류를 나타내는 entitiy set으로, 패키지의 타입(포장 종류), 패키지의 무게(무게 범위), 배송 시간에 따라 서비스 타입이 결정된다. 3가지 요소에 의해 서비스 타입이 결정되기에 패키지의 타입이 N가지, 패키지의 무게 범위를 M가지, 배송 시간을 K가지로 나누게 된다면, 총 N*M*K개의 서비스 타입으로 나뉜다. 이는 패키지에 적용되어 패키지의 요금을 결정하는 하나의 요소가 될 수 있다.

Service entity set은 총 4개의 attribute로 구성된다. Service_type은 앞서 말한 대로 3가지 요소에 따라 구분된 N*M*K개의 서비스 타입이다. Packing_type은 패키지의 포장 방법을 의미한다. Initialization에 언급되었듯이 플랫 봉투, 상자의 크기에 따라 구분할 수 있다. Weight는 패키지의 무게를 의미한다. 패키지의 정확한 무게를 일일이 저장한다면, 이에 따른 서비스 타입이 무한하기 때문에 일정 범위로 나누어서 구분하도록 한다. Delivery_time은 패키지에 대한 배송 시간을 의미한다. 당일 배송, 1박 2일, 혹은 그 이상 등 배송에 걸리는 시간을 나타낸다.

Service entity set의 primary key는 {service_type}이다. 앞서 서비스 타입을 결정하는 세가지 요소를 범위와 같이 유한한 개수로 나누었다. 이로 인해 서비스 타입으로만 service entity set을 구분할 수 있게 되었다.

2.1.6 shipment



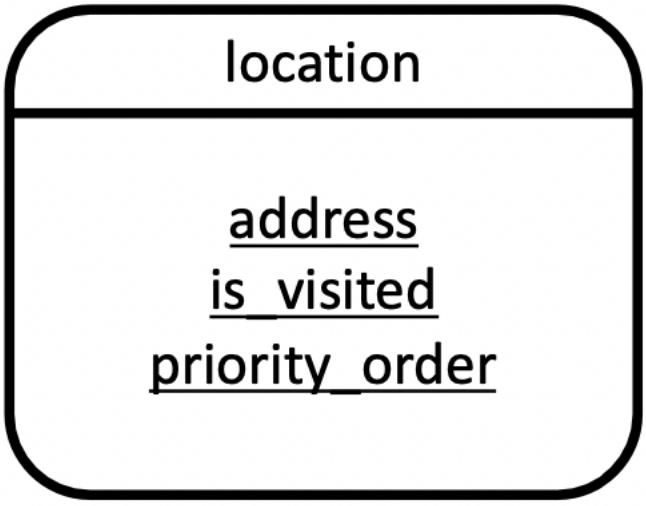
Shipment entity set은 패키지를 운반하는 운행수단을 나타내는 entity set이다. 패키지를 운반하는 운행 수단의 종류(ex. 트럭, 선박, 항공기 등)와 그 운행수단의 고유 id와 현재 운행 여부를 나타낸다. Shipment entity는 운반하게 될 패키지와 관계를 이루게 되며 패키지를 목적지 혹은 허브와 같은 중간 집하 시설로 이동시키게 된다.

Shipment entity set은 총 3개의 attribute로 구성된다. Vehicle은 운행 수단의 종류로 트럭, 선박, 항공기와 같이 분류되게 된다. 주로 패키지가 이동하는 시간과 거리에 따라 운반 수단이 결정될 것이다. Vehicle_id는 이러한 운행 수단의 고유한 아이디로 같은 운반 수단 간을 구분하기 위해 사용하는 속성이다. 운행 수단과 운행 수단의 아이디를 통해 현재 운반 수단이 무슨 패키지를 운반하고 있으며, 이동 경로가 어떻게 되는지를 확인할 수 있다. Is_running은 현재 운반 수단의 운행 상태를 나타내는 속성이다. 현재 운행을 하며

패키지를 배송하고 있는지, 아직 패키지 배송을 시작하지 않았는지, 모든 패키지를 배송하고 운반을 마무리 했는지를 확인할 수 있다.

Shipment entity set의 primary key는 {vehicle, vehicle_id}이다. 운반 수단의 종류와 해당 운반 수단의 아이디로 모든 shipment entity를 구분할 수 있다. 아이디만으로는 운반수단이 다르지만 아이디가 같은 경우로 구별이 불가능할 경우를 대비해 두 속성을 모두 primary key로 설정하였다.

2.1.7 location



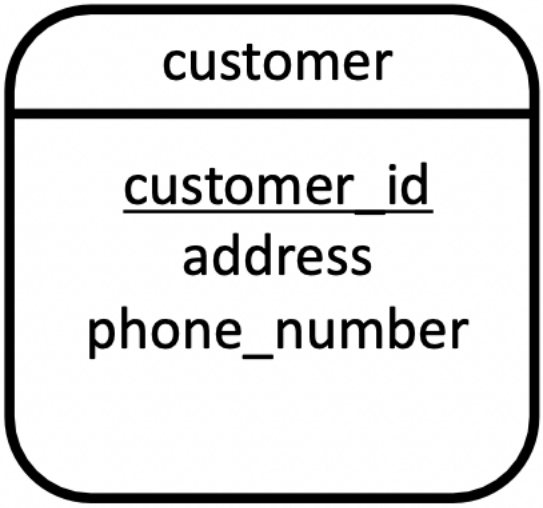
Location entity set은 위치에 대한 정보를 나타내고 있다. 위치는 shipment entity가 이동해야 할 위치로 한정되어 있다. 이 location entity set과의 관계에 따라 운송수단이 어느 곳으로 이동해야 하는지, 어느 순서로 이동해야하는지, 어느 곳까지 이동했는지를 표현할 수 있다.

Location entity set의 attribute는 총 3개이다. Address는 운송수단이 이동해야할 주소를 나타내는 속성이다. 이 주소는 패키지를 집하하는 집하장이 될 수도, 패키지를 배송하는 주소가 될 수도 있다. 이처럼 주소는 운송수단이 패키지에 대한 특정 행위를 하는 곳에 대한 정보를 나타낸다. Is_visited는 저장되어 있는 주소에 도착했는지 여부를 나타내는 속성이다. 이로 인해 현재 운송수단의 위치를 추적할 수 있고, 현재 운송수단에 위치한 패키지를 알 수 있기 때문에 패키지의 위치 역시 추적할 수 있게 된다. 마지막으로 priority_order는 운송수단이 이동하는 위치의 우선순위를 나타내며, 우선순위에 따라 이동하게 된다. 이를 통해 운송수단이 현재 어느 곳으로 향하는지를 알 수 있고, 지난 경로 및 앞으로의 경로를 알 수 있다. 이러한 운송 수단 및 패키지의 위치를 추적하기 위한 entity set으로 운반 중 사고가 난 경우 어느 패키지가 사고에 영향을 받았는지를 확인할 수 있다.

Location entity set의 Primary key는 {address, is_visited, priority_order}가 된다. 모든 속성이 primary key가 되는 이유는 다음과 같다. 우선 같은 주소에 여러 운송 수단이 도착할 수 있고, 이러한 운송 수단마다 해당 주소에 방문하는 우선순위, 방문 여부가 다 다를 것이

다. 이로 인해 모든 속성을 통해서만 entity set을 구분할 수 있다.

2.1.8 customer

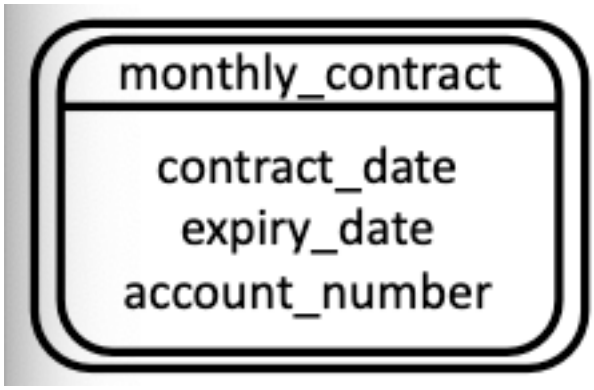


패키지 배송을 이용하는 고객의 정보를 나타내는 entity set이다. 고객은 패키지 배송 서비스를 이용하는 사람으로 패키지를 보내는 사람을 가리킨다. Customer entity set을 중심으로 고객이 보낸 패키지의 내역, 월간 계약 여부, 과거 거래 정보를 확인할 수 있다.

Customer entity set의 attribute는 총 3가지 이다. Customer_id는 고객을 구분하기 위한 고유한 아이디이다. 고객을 구분하기 위해 아이디를 선택한 이유는 동명이인으로 인한 중복 문제를 배제시킴과 동시에 주민번호로 구분할 경우 발생할 수 있는 개인 정보 문제를 피하기 위함이다. Address는 고객의 주소로 만약 물품이 반품될 경우, 반품된 물건은 패키지를 배송한 고객의 주소로 배송하기 위함이다. 이로 인해 반품된 물건이 돌아갈 곳이 없는 이슈를 배제시킬 수 있다. 마지막으로 phone_number는 고객에게 패키지의 상태를 알려주기 위한 정보이다. 패키지의 배송 시작, 배송 완료, 반품 접수 등 기본적인 정보와 더불어 패키지가 사고로 인해 문제가 생긴 경우에도 고객에게 공지할 수 있도록 하기 위해 해당 속성을 포함하였다.

Customer entity set의 primary key는 {customer_id}이다. 이는 앞서 언급했듯이, 고객의 개인 정보 및 동명이인 문제를 배제하기 위해 아이디를 사용하게 되었고, 이는 고객을 구별할 수 있는 값이 된다. 이로 인해 패키지를 보낸 고객이 누구인지 겹치지 않게 확인할 수 있다.

2.1.9 monthly_contract

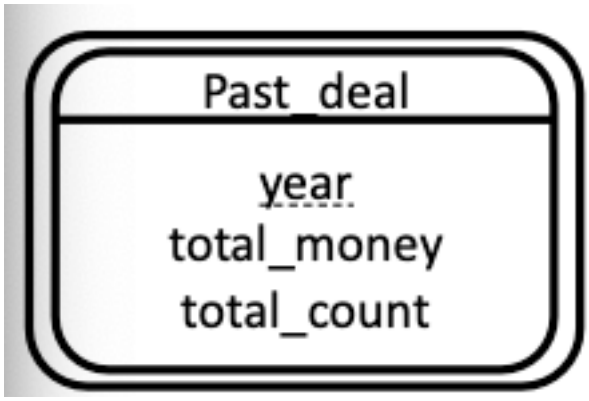


달마다 계약한 고객을 나타내는 entity set이다. 고객이 패키지 서비스 회사와 계약을 했는지에 대한 여부와 계약을 했다면, 계약 시작일은 언제이며, 계약 만료일은 언제인지를 알 수 있다. Monthly_contract entity set은 weak entity set으로 customer와 관계를 가진다.

Monthly contract entity set의 attribute는 총 4가지이다. 우선 customer_id는 strong entity set인 customer entity set으로부터 도출되며, 해당 속성은 어떤 고객이 매달 계약을 했는지를 알 수 있도록 해준다. Contract_date는 해당 고객의 계약 시작 일자이며, expiry_date는 해당 고객의 계약 만료 일자이다. 해당 일자 사이에 배송되는 패키지에 대해서는 패키지 개별로 요금을 청구하는 것이 아닌, 매달 모아서 패키지를 청구하게 된다. 즉, 계약 기간과 패키지의 배송시작 기간을 통해 해당 패키지가 매달 한번에 청구될 수 있도록 해준다.

Monthly contract entity set의 primary key는 {Customer_id}이다. 이는 계약의 대상이 고객이며, customer entity set과 strong-weak entity set관계를 맺고 있기에, 고객만이 계약을 하게 된다. 이로 인해 고객의 아이디로 해당 entity set을 구별할 수 있다.

2.1.10past_deal



1년간 해당 고객이 지출한 배송비와 배송한 패키지 개수를 나타내는 entity set이다. 어느 고객이 1년간 가장 많은 돈을 지불하였는지, 가장 많은 패키지를 발송하였는지를 알 수 있도록 한다. Past_deal entity set은 weak entity set으로 customer와 관계를 가진다.

Past_deal entity set의 attribute는 총 4가지이다. 우선 customer_id는 strong entity set인 customer entity set으로부터 도출되며, 해당 속성은 고객이 누구인지를 알 수 있다. Year은 해당 집계를 기록한 년도를 의미한다. 한명의 고객이 다년간 패키지 배송 서비스를 이용할 경우, 년마다 소비 금액과 배송 개수가 다르므로 이를 여러해 동안의 정보를 저장할

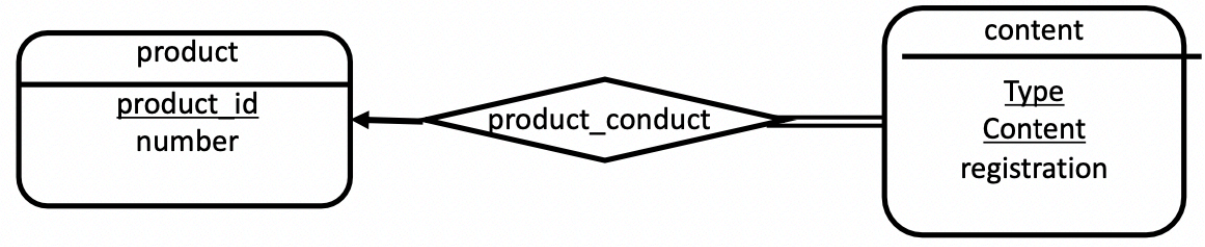
수 있다. Total_money는 해당 년도에 해당 고객이 패키지 배송에 소비한 금액을 의미한다. Total_count는 해당 년도에 해당 고객이 배송한 패키지의 개수를 의미한다. 이를 통해 해당 년도의 Entity들을 비교하여 total_money와 total count가 가장 많은 고객을 도출해낼 수 있다.

Past_deal entity set의 primary key는 {Customer_id,year}이다. 이는 금액과 배송한 패키지의 개수를 집계 대상이 고객이며, 집계 및 통계가 년 단위로 이루어지고 있기 때문이다. 이때, year이라는 속성은 discriminator가 된다.

2.2 Relationship set

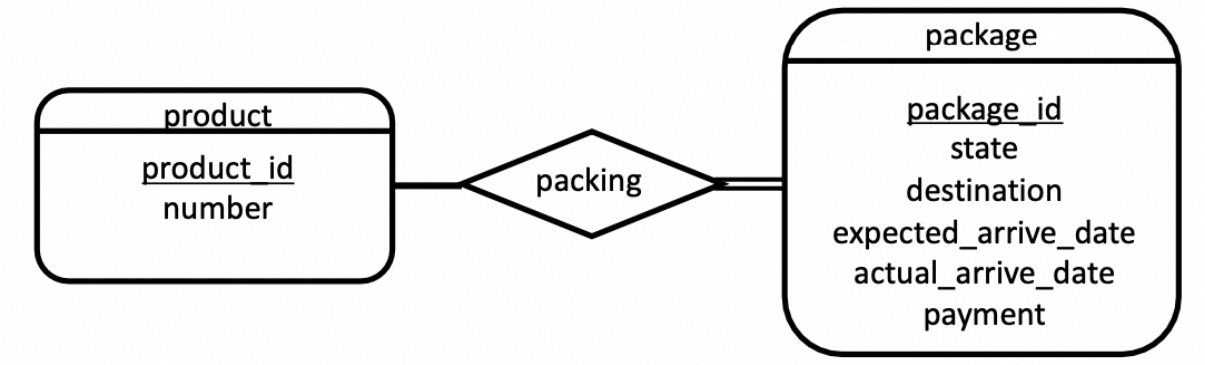
Relationship set의 구성에 대한 설명이다. 총 9개의 relationship set이 있다.

2.2.1 Product_content



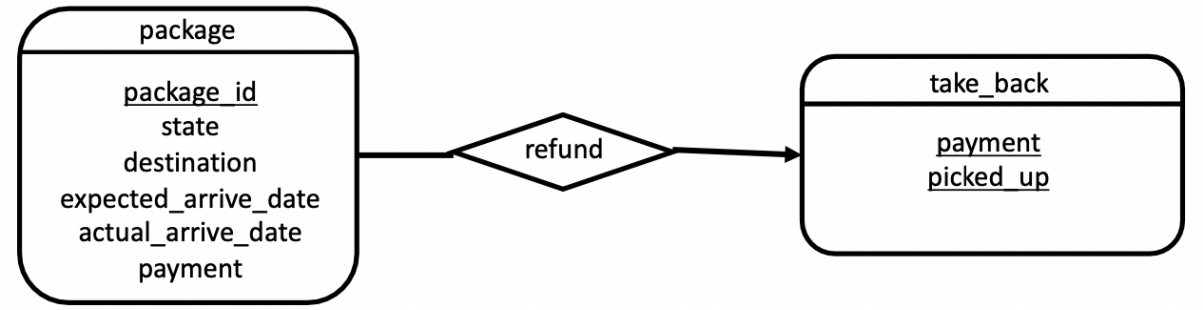
Product to content는 product_content relation을 통해 one to many 관계로 나타난다. 이는 하나의 제품이 위험물인 동시에 해외 배송 품목이라면, product_content와 두 개의 관계를 가지게 될 것이다. 하지만 하나의 제품 상태를 나타내는 entity는 하나의 제품과의 관계만을 맺을 것이다. 이는 content가 weak entity set으로 제품의 존재에 종속적이기 때문이다. 모든 content는 제품과 관계를 가지기 때문에 total 관계로 표현 하였다. 이는 제품의 내용을 패키지 배송 회사에서 인지하여야 하는 경우에만 관계가 성립하기 때문에 모든 내용은 제품과 관계를 가진다. 하지만 product는 product_content relation과 partial 관계를 이루는데, 이는 제품이 패키지 회사에서 알아야하는 제품이 아닌 경우 굳이 content와 관계를 가질 필요가 없기 때문이다. 이와 같은 관계를 통해, 현재 제품이 위험물 혹은 해외 배송과 같이 제품에 대한 정보를 알아야 하는지 여부를 확인할 수 있다.

2.2.2 Packing



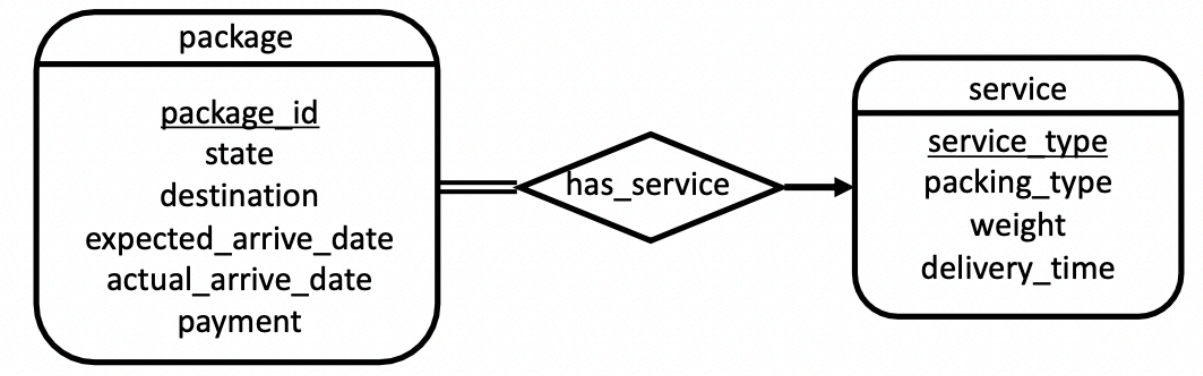
Product to package는 packing relation을 통해 many to many 관계로 나타난다. 하나의 제품 구성이 여러개의 패키지에 들어갈 수 있으며, 하나의 패키지가 여러가지 제품으로 구성될 수 있음을 의미한다. 또한 하나의 패키지에는 반드시 하나 이상의 제품이 포함되어야 하기 때문에 package와 packing은 total 관계로 표현되었다. 반면, 모든 제품이 패키지의 구성품이 되지 않을 수도 있을 경우가 존재하므로 proudct와 packing은 parital 관계를 이룬다. 이로 인해 패키지가 어떤 제품으로 구성되어 있는지를 알 수 있으며, 더 나아가 해당 패키지에 구성된 제품이 위험물질 혹은 해외배송 제품인지 여부 역시 확인할 수 있다.

2.2.3 Refund



Package to take_back은 refund relation을 통해 many to one 관계로 나타난다. 하나의 패키지가 반품되고자 한다면, take_back과 관계를 가지게 된다. 이때 반품 정보는 해당 패키지에 대한 반품비 청구가 완료되거나 완료되지 않거나, 반품 패키지를 가지고 가거나 가지고 가지 않은 상태로만 특정되기 때문에 하나의 상태와의 관계만을 유지할 수 있다. 반면 하나의 반품 상태는 여러 반품되는 패키지와 관계를 이룰 수 있기 때문에 many side가 된다. 예를 들어 2개 이상의 패키지가 반품비 결재가 완료된 아직 반품되지 않았다면 같은 take_back entity와 관계를 이루게 된다. 반품 상태는 반품지 결재가 이루어지지 않은 반품되지 않은 패키지가 없는 경우가 발생할 수 있으므로, partial 관계를 이룬다. 패키지 또한 모든 패키지가 반품되는 것은 아니기 때문에 partial 관계로 표현된다.
이와 같은 관계를 통해 해당 패키지가 반품되는 패키지인지 여부를 알 수 있다.

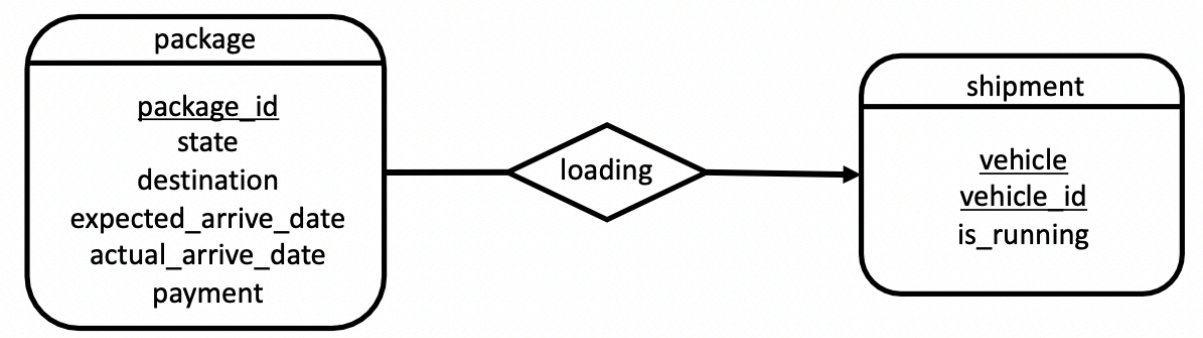
2.2.4 Has_service



Package to service는 has_service를 통해 many to one 관계로 나타난다. 하나의 패키지는 하나의 서비스 타입만을 선택할 수 있는 반면, 하나의 서비스 종류는 여러개의 패키지에

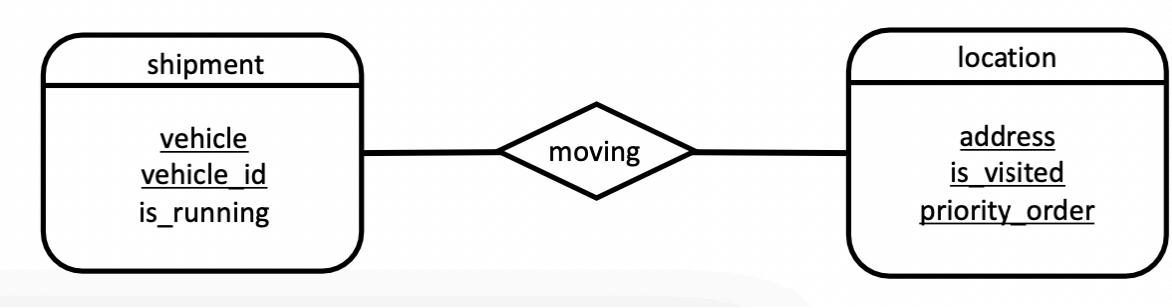
적용되는 서비스 타입이 될 수 있기 때문이다. 또한 모든 패키지는 하나의 서비스 타입에 속하기 때문에 total관계를 가지는 반면, 특정한 서비스 타입이 모든 패키지에 적용되지 않는 경우가 발생할 수 있으므로 partial 관계를 가진다. 이와 같은 관계를 통해 해당 패키지가 어떤 서비스 타입을 적용받는지 확인할 수 있다.

2.2.5 Loading



Package to shipment는 loading을 통해 many to one의 관계로 나타난다. 하나의 패키지는 특정 시간에 하나의 운송수단에 적재되어있는 반면, 하나의 운송 수단에는 여러개의 패키지를 적재할 수 있기 때문이다. 또한 패키지가 운송수단에 적재되어 있지 않고, 창고에 있거나 배송이 완료된 상태라면 운송수단과 관계를 맺지 않기 때문에 모든 패키지가 운송수단과 관계를 맺지 않을 수 있는 partial 관계로 표현될 수 있다. 운송수단 역시 모든 운송수단이 패키지를 싣고 있지 않을 수 있기 때문에, 배송을 모두 완료하여 패키지가 없거나, 아직 패키지를 적재하기 전의 상태와 같은 상태가 존재할 수 있기 때문에 partial 관계로 표현된다. 이와 같은 관계를 통해 패키지가 운송수단에 적재되어 있는지, 패키지가 적재되어 있다면 어떤 운송수단에 적재되어 있는지를 알 수 있다.

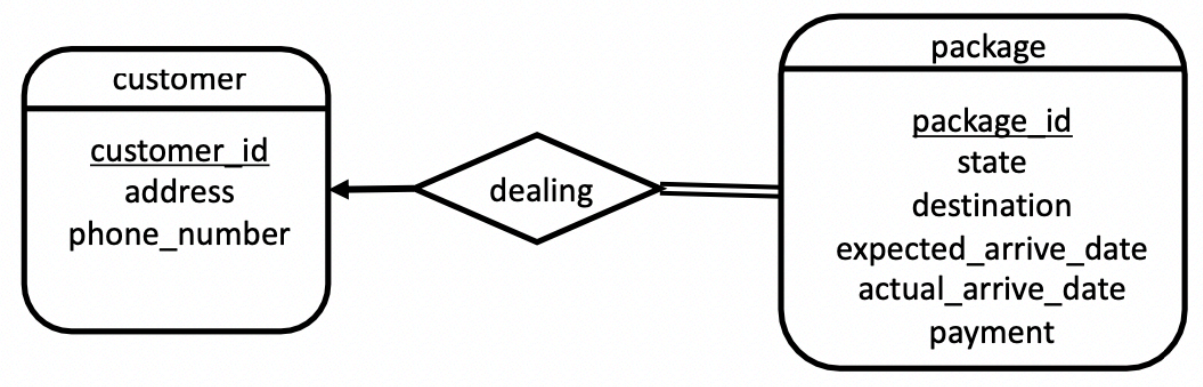
2.2.6 Moving



Shipment to location은 moving을 통해 many to many의 관계로 나타난다. 하나의 운송수단은 여러 장소를 이동하기에 여러 장소와 관계를 맺을 수 있다. 또한 하나의 장소 역시 여러 대의 운송수단이 방문할 수 있다. 가령 물류창고와 같은 곳은 많은 운송수단이 모이는 곳이 될 것이다. 하지만 모든 운송수단이 하나의 위치에 가지 않을 수 있다. 운행을 하지 않는 운송수단이라면 목적지가 없으므로 장소와 관계를 맺지 않을 것이다. 또한 장소 역시 아무런 운송수단이 목적지로 하고 있지 않을 수도 있다. 물류창고와 같은 장소는 패키지 이동에 반드시 필요한 장소이므로 entity set에 포함되어 있겠지만 운송수단이 오지

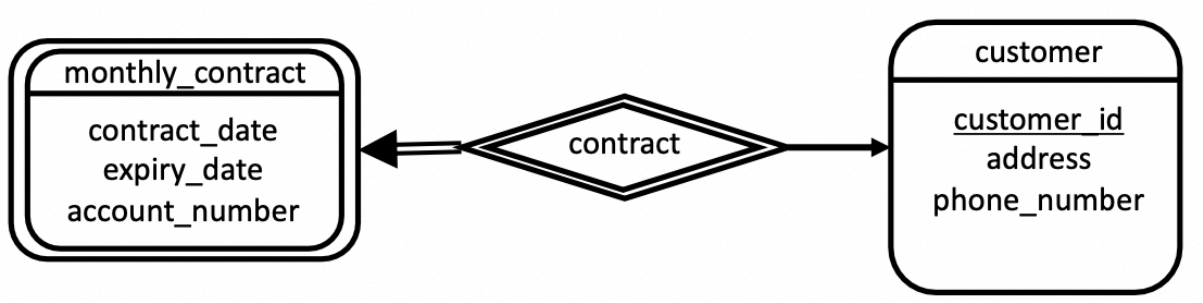
않는 경우도 존재할 수 있기 때문이다. 이와 같은 관계를 통해 현재 운송수단이 루트를 알 수 있다.

2.2.7 Dealing



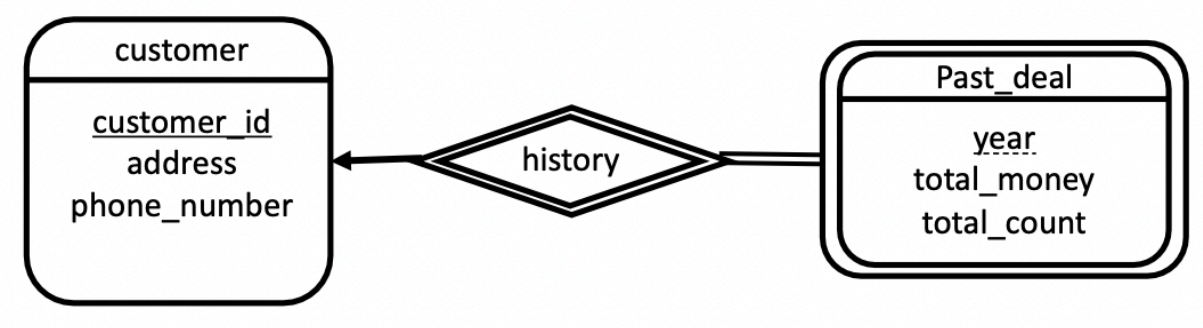
Customer to package는 dealing을 통해 one to many의 관계로 나타난다. 하나의 패키지는 한 명의 고객에 의해서만 접수가 되지만, 한 명의 고객이 여러 개의 패키지를 접수할 수 있기 때문이다. 하지만 등록된 모든 고객이 최소 한번씩은 패키지를 보냈다는 확신이 없으므로, 고객으로 등록만 하고 패키지를 아직 보내보지 않은 경우가 있을 수 있으므로 partial 관계를 가진다. 반면 모든 패키지는 고객들의 접수에 의해 배송되는 것이므로 total 관계를 가진다. 이를 통해 어느 고객이 어느 패키지를 배송하고자 하는지를 알 수 있으며, 추후 패키지가 반품될 때, 어디로 패키지가 반품되어야할지를 알 수 있다.

2.2.8 Contract



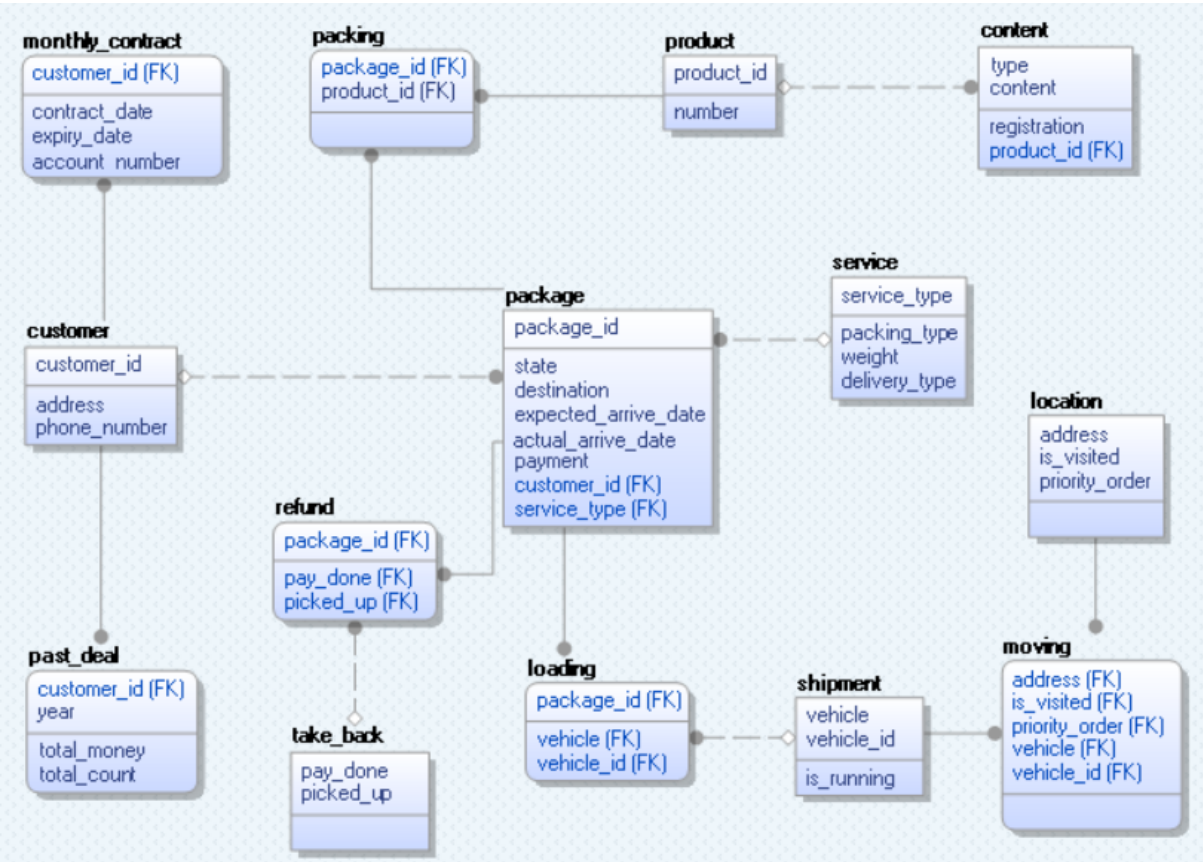
Monthly_contract to customer는 contract를 통해 one to one의 관계로 나타난다. 하나의 고객이 최대 하나의 계약만 할 수 있으며, 계약에 관한 정보는 해당 고객의 정보를 포함하고 있기 때문에 한명의 고객과만 관계를 가질 수 있다. 모든 고객이 월간 계약을 맺지 않기 때문에 customer는 Partial 관계를 맺는 반면, 계약과 관련된 정보는 고객의 정보(customer_id)를 포함하고 있는 종속적인 weak entity set이기 때문에 total 관계를 맺는다. 이를 통해 어떤 고객이 월간 계약을 맺었는지를 알 수 있다.

2.2.9 History



Customer to past_deal은 history를 통해 one to many 관계로 나타난다. 하나의 과거 정보는 한명의 고객에 대한 정보만을 나타내는 반면, 한 명의 고객이 여러 해 동안 거래를 했다면 여러 개의 과거 기록을 가질 수 있기 때문이다. 이는 과거 기록이 년 단위로 기록되기 때문이다. 또한 아직 패키지 배송을 해보지 않은 고객은 과거 기록이 없기 때문에 past_deal과 관계를 맺지 않고 있을 수 있다. 이는 customer가 partial 관계를 맺고 있음을 의미한다. 반면 모든 과거 기록은 고객에게 종속적이기 때문에 total 관계를 이루게 된다. 이와 같은 관계를 통해 해당 고객이 한 해 동안 얼마의 금액을 사용했고, 얼마나 많은 패키지를 보냈는지를 확인할 수 있고, 이로 인해 한해 동안 가장 많은 돈을 소비한 고객과 가장 많은 패키지를 보낸 고객을 찾을 수 있다.

3. Relational Schema diagram



구성한 E-R Model을 바탕으로 Reduction을 거쳐 다음과 같은 Relational Schema diagram을

도출하였다. 각각의 relation schema의 도출 과정과 attribute의 특징을 살펴보고자 한다.

3.1monthly_contract – customer

monthly_contract와 customer를 이어주는 contract relationship set은 one to one 관계로 이어져 있다. Contract 관계는 weak entity set과 strong entity set을 이어주고 있기 때문에 다음과 같이 Reduction할 수 있다. 우선 strong entity set인 customer set은 기존의 속성을 그대로 유지하는 schema가 되며, weak entity set인 monthly_contract는 기존의 속성에 strong entity set인 customer의 primary key를 추가로 가지게 되며, 해당 키인 customer_id는 monthly_contract의 primary key이자 customer schema를 referencing하는 foreign key가 된다. 이때 relationship set인 contract는 schema로 표현하지 않는다.

3.2customer – past_deal

past_deal와 customer를 이어주는 history relationship set은 many to one 관계로 이어져 있다. history 관계는 weak entity set과 strong entity set을 이어주고 있기 때문에 다음과 같이 Reduction할 수 있다. 우선 strong entity set인 customer은 기존의 속성을 그대로 유지하는 schema가 되며, weak entity set인 past_deal는 기존의 속성에 strong entity set인 customer의 primary key를 추가로 가지게 되며, 해당 키인 customer_id는 monthly_contract의 primary key이자 customer schema를 referencing하는 foreign key가 된다. 또한 기존 past_deal entity set의 discriminator인 year도 primary key가 된다. 이때 relationship set인 history는 schema로 표현하지 않는다.

3.3customer – package

customer와 package를 이어주는 dealing relationship set은 one to many 관계로 이어져 있다. 또한 many side는 total로 이어져 있기 때문에 다음과 같이 Reduction할 수 있다. One side entity set인 customer set은 기존의 속성을 그대로 유지하는 schema가 되며, many side entity set인 package는 기존의 속성에 one side entity set인 customer의 primary key를 추가로 가지게 되며, 해당 키인 customer_id는 customer schema를 referencing하는 package의 foreign key가 된다. 이때 relationship set인 dealing는 schema로 표현하지 않는다.

3.4take_back – package

take_back과 package를 이어주는 refund relationship set은 one to many 관계로 이어져 있다. 하지만 many side가 partial로 이어져 있기 때문에 Reduction하지 않고 별도의 스키마를 생성하여 relationship set을 나타낸다. One side entity set인 take_back set의 primary key와 many side entity set인 package의 primary key를 속성으로 하는 refund schema를 생성한다. 이때 refund schema의 primary key는 many side의 primary key인 package_id가 해당 스키마의 primary key가 된다. 또한 모든 속성은 foreign key가 되며, pay_done, picked_up은 take_back schema를 referencing하고, package_id는 package schema를 referencing한다.

3.5service – package

service와 package를 이어주는 has_service relationship set은 one to many 관계로 이어져 있다. 또한 many side는 total로 이어져 있기 때문에 다음과 같이 Reduction할 수 있다. One side entity set인 service set은 기존의 속성을 그대로 유지하는 schema가 되며, many side entity set인 package는 기존의 속성에 one side entity set인 service의 primary key를 추가로 가지게 되며, 해당 키인 service_type는 service schma를 referencing하는 package의 foreign key가 된다. 이때 relationship set인 dealing는 schema로 표현하지 않는다.

3.6product – package

product와 package를 이어주는 packing relationship set은 many to many 관계로 이어져 있기 때문에 Reduction할 수 없다. many side entity set인 product의 primary key와 또 다른 many side entity set인 package의 primary key를 속성으로 하는 packing schema를 생성한다. 이때 packing schema의 primary key는 양쪽 모두의 primary key인 product_id, package_id로 이루어진다. 또한 모든 속성은 foreign key가 되며, product_id은 product schema를 referencing하고, package_id는 package schema를 referencing한다.

3.7product – content

product와 content를 이어주는 product_content relationship set은 one to many 관계로 이어져 있다. 또한 many side는 total로 이어져 있기 때문에 다음과 같이 Reduction할 수 있다. One side entity set인 product set은 기존의 속성을 그대로 유지하는 schema가 되며, many side entity set인 content는 기존의 속성에 one side entity set인 product의 primary key를 추가로 가지게 되며, 해당 키인 product_id는 product schema를 referencing하는 content의 foreign key가 된다. 이때 relationship set인 product_content는 schema로 표현하지 않는다.

3.8shipment – package

shipment과 package를 이어주는 loading relationship set은 one to many 관계로 이어져 있다. 하지만 many side가 partial로 이어져 있기 때문에 Reduction하지 않고 별도의 스키마를 생성하여 relationship set을 나타낸다. One side entity set인 shipment의 primary key와 many side entity set인 package의 primary key를 속성으로 하는 loading schema를 생성한다. 이때 loding schema의 primary key는 many side의 primary key인 package_id가 해당 스키마의 primary key가 된다. 또한 모든 속성은 foreign key가 되며, vehicle과 vehicle_id는 shipment schema를 referencing하고, package_id는 package schema를 referencing한다.

3.9shipment – location

shipment와 location를 이어주는 moving relationship set은 many to many 관계로 이어져 있기 때문에 Reduction할 수 없다. many side entity set인 shipment의 primary key와 또 다른 many side entity set인 location의 primary key를 속성으로 하는 moving schema를 생성한다. 이때 moving schema의 primary key는 양쪽 모두의 primary key인 vehicle, vehicle_id, address,

is_visited, priority_order로 이루어진다. 또한 모든 속성은 foreign key가 되며, vehicle, vehicle_id는 shipment schema를 referencing하고, address, is_visited, priority_order는 location schema를 referencing한다.