

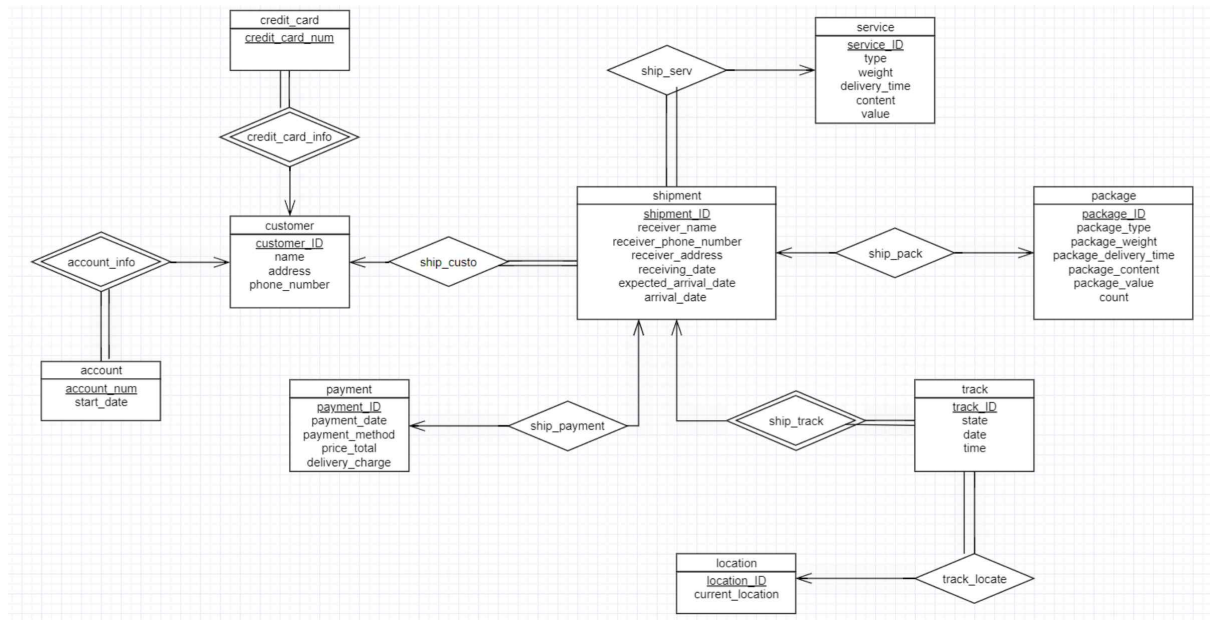
CSE4110 - Database System

Spring2023

전공 : 컴퓨터공학과

학번 : 20211569

이름 : 이승연



<E-R diagram>

1. Entity 설명

- shipment

택배 상품 각각을 저장한다. shipment_ID를 primary key로 가지고 있다. 이외에는 택배 수령자의 정보인 receiver_name, receiver_phone_number, receiver_address, 택배 접수일인 receiving_date, 예상 도착일인 expected_arrival_date, 실제 택배 도착일인 arrival_date를 attribute로 가진다.

- customer

판매자 정보를 저장한다. customer_ID를 primary key로 가지고 있다. 이외에는 name, address, phone_number를 attribute로 가진다.

- package

택배에는 합배송이 있을 수 있다. 즉, 한 택배박스 안에 여러 물건이 있을 수 있다. 따라서 이후 많은 package를 발송한 customer를 구하는 쿼리를 위해 package라는 entity set을 만들었다. package_ID를 primary key로 가지고 있다. 발송물 종류를 나타내는 package_type, package 무게인 package_weight, 배송 시기가 얼마나 걸리는지에 관한 package_delivery_time, 또한 내용물이 위험물질 또는 국제 발송인 경우를 구분해주기 위한 package_content와 package_value, 마지막으로 한 package에 몇 개의 택배가 존재하는지에 관한 정보인 count까지 attribute로 가진다. count는 이후 가장 많은 패키지를 발송한 고객을 찾는 쿼리를 위해 필요하다.

- track

택배를 보낸 package를 추적하기 위한 entity이다. 어느 시점에서든지 package에 관한 상세정보를 알 수 있어야 한다. 나는 CJ 대한통운 배송추적 서비스를 참고하여 총 5가지의 단계로 분류하기로 했다. track_ID에 따라 state가 바뀐다. 1은 상품준비, 2는 상품인수, 3은 상품이동, 4는 배달준비, 5는 배달완료 상태라는 뜻이다. 또한 이 track_ID 값이 primary key가 된다. track_ID에 따라 state가 설정되고, 날짜와 시간인 date, time까지 attribute로 가진다.

- location

현재 package가 현재 어디에 있는지(트럭 or 비행기 or 창고 등)를 알려주기 위한 entity이다. 회사가 package를 트럭이나 창고의 관점에서 조사할 수 있는 경우를 대비해 설정해주었다. location_ID를 가지고 있다. 또 장소를 나타내는 location_type까지 primary key이자 attribute로 가진다. 예를 들어 '트럭 1721'이라면 location_ID는 1721, location_type은 truck이 된다.

- service

package의 여러 종류에 따라 다양한 서비스가 있을 수 있다. service_ID를 primary key로 가진다. 이외에는 flat envelope, small box, larger boxes, etc..를 구분하는 type, package 무게 weight, 배달 시간이 얼마나 걸리는지에 관한 delivery_time, 내용물이 위험물질 또는 국제 발송인 경우를 나타내는 content, value까지 attribute로 가진다.

- payment

package에 대한 청구서 entity이다. payment_ID를 primary key로 가진다. attribute로는 package를 주문한 날인 payment_date, 결제 수단인 payment_method, 청구가격인 price_total, 배달비인 delivery_charge를 가진다.

payment_date는 고객의 지불 방법에 따라 즉시 결제될 수도, 월에 한 번 정해진 날짜에 결제될 수도 있다.

payment_method에는 두 가지가 있다. 첫 째는 발송인과 계약을 맺고 매달 계좌번호로 가격을 청구하는 방법, 두 번째는 자주 이용하지 않아 신용카드로 결제하는 방법이다. 그리고 이 이외에도 customer이 반품을 하는 경우 prepaid된 것인지 판단하는 경우도 있는데 이는 고객이 맞지 않는 옷이나 손상된 상품을 반품할 때 지불할 배송비를 미리 지불하는 것을 뜻한다.

delivery_charge는 service entity를 참고해 책정하는데, 이후 배송비를 가장 많이 지출한 고객을 찾는 쿼리를 위해 필요하다

- account

발송인과 계약을 맺고 매달 계좌번호로 가격을 청구하는 고객에 대한 정보를 나타내기 위한 entity이다. 계좌번호인 account_num을 primary key로 가진다. 또한 실제 결제일인 start_date를

attribute로 가진다.

- credit_card

자주 이용하지 않아 신용카드로 결제하는 고객에 대한 정보를 나타내기 위한 entity이다. 카드 정보인 credit_card_num을 primary key로 가진다.

2. relation 설명

- ship_custo

customer이 주문한 shipment에 대한 정보를 연결시킨다. 이후 대부분의 쿼리에서 customer, shipment에 관한 정보가 필요하기 때문에 이 relation은 굉장히 중요하다. customer 한 명이 여러 shipment를 가질 수 있지만 shipment 하나는 customer 한 명만을 가지기 때문에 one to many 관계이며 모든 shipment는 주문해준 customer을 가져야 하기 때문에 total 조건이 있다.

- ship_pack

한 택배에 여러 개의 상품을 주문할 수 있다. 이 때 한 택배가 shipment에 해당되고 여러 개의 상품이 package에 해당된다. 하지만 말 그대로 한 shipment에 여러 package가 있는 것이 아니라 한 shipment에 한 package가 연결되고 그 package set 안에 count attribute를 통해 상품 개수를 지정하는 형태이다. 따라서 one to one 관계이다.

- ship_serv

shipment와 service의 관계이다. shipment의 package 종류(ex 유형, 무게, 배달 시간 등)에 따라 하나의 service가 정해진다. 반대로 하나의 service에는 여러 shipment가 해당될 수 있다. 따라서 many to one 관계이다. shipment에 대한 service는 무조건 하나 존재해야 하므로 total 조건이 있다.

- ship_track

shipment와 track의 관계이다. shipment가 존재해야 track entity가 존재할 수 있으므로 track은 shipment의 weak entity이다. 하나의 shipment에는 배송 진행 상태가 최대 5개 있을 수 있지만, track entity에 대한 shipment entity는 하나 뿐이다. 따라서 one to many 관계이다. track entity는 shipment entity에 dependent 하므로 total 조건이 있다.

- track_locate

track과 location의 관계이다. 현재 추적 상태에 대해 상품이 어디에 있는지 알려주는 관계이다. (예를 들자면, ○○이가 □□에게 보낸 택배는 현재 '상품 인수' 상태이며 '창고 1167'에 있다.) 현재 상태에 대해서 상품은 한 곳에 있을 것이고, 그 장소에 있는 택배는 여러 개가 있을 수 있다.

따라서 many to one의 관계이다. 이후 특정 트럭의 특정 순간에 짐을 싣고 있던 모든 고객의 정보를 구할 때 이 relation이 사용된다. 추적 상태에 따라 상품이 존재하는 장소는 한 곳이 무조건 존재하므로 total 조건이 있다.

- ship_payment

shipment과 payment의 관계이다. 하나의 shipment는 하나의 청구서를 가진다. 마찬가지로 하나의 청구서는 하나의 shipment에 관한 것이므로 one to one 관계이다. 이후 각 발송물에 대한 청구서 목록을 추출할 때 이 relation을 활용한다.

- payment_custo

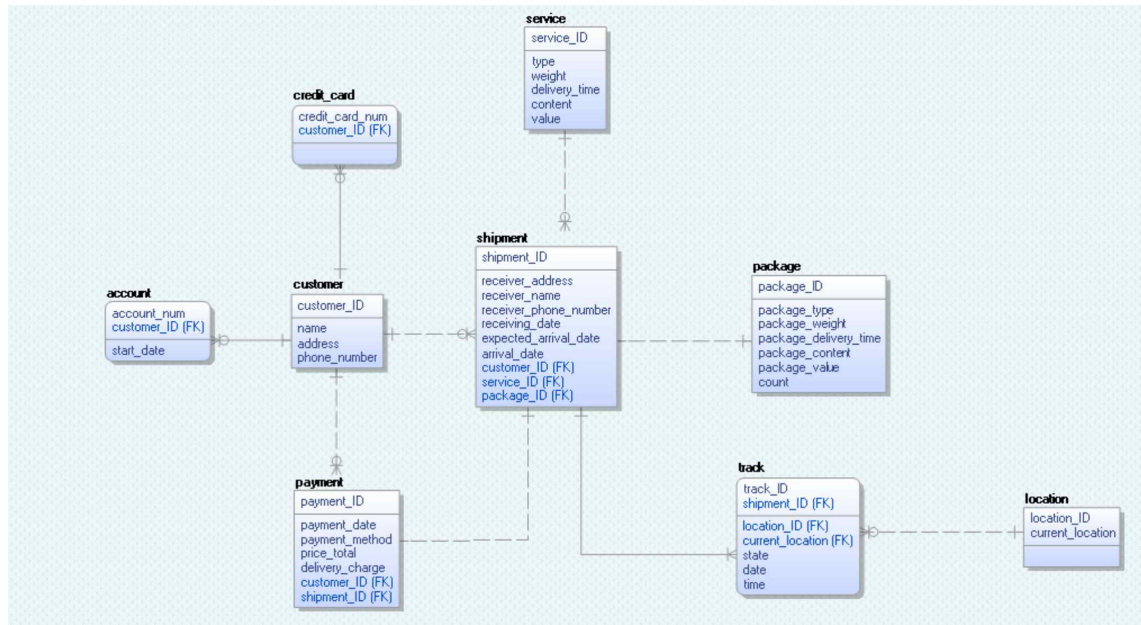
customer과 payment의 관계이다. 고객 한 명은 여러 청구서를 가질 수 있으며 한 청구서는 한 고객에 의해 생성되므로 one to many 관계이다. payment에는 어떤 방식으로 결제할지에 관한 payment_method 속성이 존재한다. 따라서 이 정보들에 따라 계좌이체인지 신용카드인지 등을 나타낼 수 있다. 이 속성에 따라 customer의 weak entity인 account, credit_card entity 정보를 이용하여 결제를 진행한다. 모든 청구서는 구매한 고객이 존재하므로 total이다. 이후 각 고객에 대한 청구서를 추출할 때 이 relation을 사용한다.

- account_info

customer과 account 사이의 관계로 account가 customer의 weak entity이다. 매달 계약을 맺어 결제하는 고객에 대한 계좌번호가 저장된다. 한 고객은 여러 계좌를 가질 수 있으므로 one to many 관계이다. customer에 dependent 하기 때문에 total 조건이 있다.

- credit_card_info

customer과 credit_card 사이의 관계로 credit_card가 customer의 weak entity이다. 고객 정보에 따라 신용카드 정보가 결정된다. 한 고객은 여러 신용카드를 소지할 수 있으므로 one to many 관계이다. customer에 dependent 하기 때문에 total 조건이 있다.



<schema diagram>