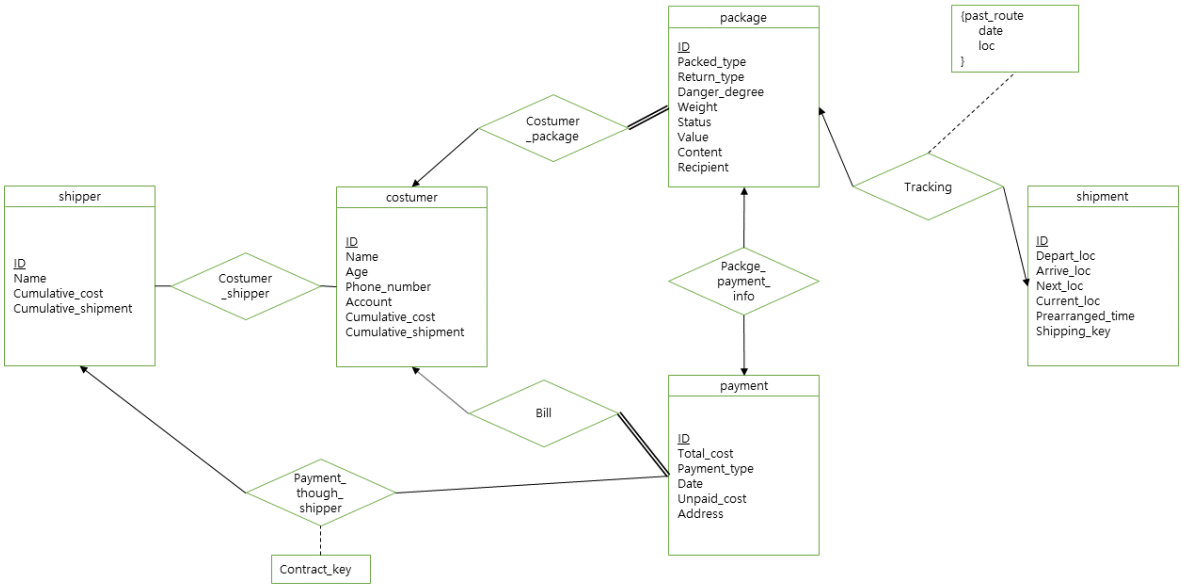
DBMS Project 1 Report

20180492 김범준

1. E-R Diagram



1) entity

- costumer

Package delivery system의 costumer entity는 택배회사 또는 배송업체를 통해 배송을 주문한 고객을 나타낸다. PK로 고객의 ID를 가지며, 고객 중 동명이인이 있을 수 있기 때문에 ID를 PK로 설정하였다. 전화번호와 계좌번호 또한 다른 사람들이 같은 번호를 사용할 수 있으므로(ex. 가족관계) 식별 가능한 ID가 있어야 한다고 판단하였다. attribute로는 이름, 나이, 전화번호, 계좌번호, 누적 주문 금액, 누적 주문 건수를 설정하였다. 전화번호 및 계좌번호는 택배회사가 가지고 있어야 할 필요가 있는 고객의 정보이고, 누적 금액과 누적 주문 건수는 프로젝트 명세의 요구사항에 따라 추가하였다.

- package

package entity는 ID(PK), packaged_type, return_type, danger_degree, weight, status, value, content, recipient의 attribute를 가지고 있다. ID는 각 배송물을 식별 가능한 PK로 작용한다.

packaged_type은 배송물이 어떠한 방식으로 포장되었는지 나타낸다. 봉투, 박스와 같은 포장 종류와 크기를 표현할 수 있다. return_type은 해당 배송물이 일반 배송인지 혹은 반품 배송인지 나타낸다. 일반 배송의 경우 normal의 값을 가지며 반품 배송의 경우 return의 값을 가질 수 있다. 이외의 상황에 해당되는 배송의 경우에도 다양한 값을 가질 수 있다. danger_degree는 해당 배송물의 파손 위험 정도를 등급으로 나누어 표현하는 attribute이다. weight는 배송물의 무게를 나타낸다. status의 경우에는 배송물이 잘 배송되고 있는지, 혹은 파손되었는지를 나타낸다. 이외에도 포장물에 약간의 파손이 일어났으나 배송에는 문제가 없는 경우 등을 표현할 수 있다. value는 배송물의 가치를, content는 배송물의 내용물이 무엇인지를 나타낸다.

- payment

payment는 배송물에 대한 지불 정보를 나타낸다. payment entity의 attribute를 package entity 안에 구현하는 방식으로도 구성할 수 있었지만, 고객이 금액 청구서를 받을 때 배송물의 ID와 다른 청구서의 ID가 필요하다고 판단하여 새로운 entity로 만들어 구성하였다. 또한, 각 지불이 배송업체를 통한 지불일 경우를 식별하기 위해서도 entity를 분리하는 것이 좋다고 판단하였다. 따라서 ID는 각 지불을 식별하기 위한 PK로 사용되는 동시에 고객이 받는 청구서의 식별 가능한 ID로도 사용된다. Total_cost는 지불의 총 금액, payment_type은 지불 방식을 나타낸다. 여기서 지불 방식은 계좌이체, 신용카드, 배송업체의 구독 서비스 등을 표현한다. date는 지불 날짜, unpaid cost는 미지불 금액을 나타낸다. 여기서 미지불 금액은 청구된 금액 중 미지불된 금액을 나타내며 만약 후불결제나 부분적인 후불결제를 했을 경우 양수의 값을 가지게 된다. address는 지불이 요청되는 주소를 나타낸다. 주소의 경우 costumer의 정보에 넣을 수도 있지만, 어떠한 고객이 각기 다른 장소로 배송을 요청할 수 있어 payment의 attribute로 삽입하였다.

- shipper

shipper는 배송업체에 대한 정보를 나타낸다. shipper는 PK로 ID를 가지며, 다른 attribute로 이름, 누적 금액, 누적 배송 건수를 가진다. 고객뿐만 아니라 배송업체 또한 누적 금액과 누적 배송 건수에 대한 데이터를 저장하는 것이 낫다고 판단하였다. 택배회사에서 누적 금액이나 누적 배송 건수의 데이터를 통해 배송업체의 규모나 고객들의 사용 빈도를 식별할 수 있기 때문이다.

- shipment

shipment는 배송 자체에 대한 정보를 나타낸다. ID를 PK로 가지며 출발지, 도착지, 현재 위치, 다음 위치를 attribute로 가진다. 배송 예정 시간 정보 또한 가지고 있으며, 해당 attribute를 통해 배송 기한이 지난 물품인지를 식별 가능하다. shipping_key는 교통 수단과 차량 번호를 저장한다. 프로젝트 명세의 query 요구사항에 대입시키면, 트럭1721의 경우 truck_1721과 같이 저장된다.

2) relationship

- costumer_shipper

costumer_shipper는 고객과 고객이 가입한 배송업체의 관계를 나타낸다. 여러명의 고객이 여러 개의 배송업체에 가입할 수 있으므로, many-to-many의 관계를 가진다. 배송업체에 가입하지 않은 고객이 있거나 고객이 한 명도 존재하지 않는 배송업체가 존재할 수 있으므로 모두 partial participation이다.

- payment_through_shipper

payment_through_shipper는 배송업체를 통한 지불에 대한 정보를 나타내는 관계이다. 여러 건의 지불이 관계에 참여할 수 있으며 택배회사를 통한 직접적인 지불도 있을 수 있으므로 partial participation이다. 또한 어떠한 지불이 배송업체를 통해서 이루어졌다면 이용된 배송업체는 반드시 하나이기 때문에 many-to-one 관계를 가진다. relation attribute로는 contract_key를 가지는데, 해당 결제가 일시적인 일반결제인지 혹은 매달 일정 금액의 구독료를 지불한 구독결제인지를 나타낸다.

- bill

bill은 고객과 지불 간의 relation으로, 고객이 받아볼 수 있는 청구서의 관계를 가진다. 여러 건의 지불이 한 명의 고객에게 해당될 수 있으므로 many to one의 관계를 가지며 어떠한 지불이 청구되는 고객은 무조건 존재해야 하기 때문에 total participation이다.

- package_payment_info

package_payment_info는 어떠한 배송물에 대응되는 결제 정보를 담고 있는 관계이다. 하나의 배송물마다 하나의 결제 정보가 존재하므로 one-to-one 관계이다.

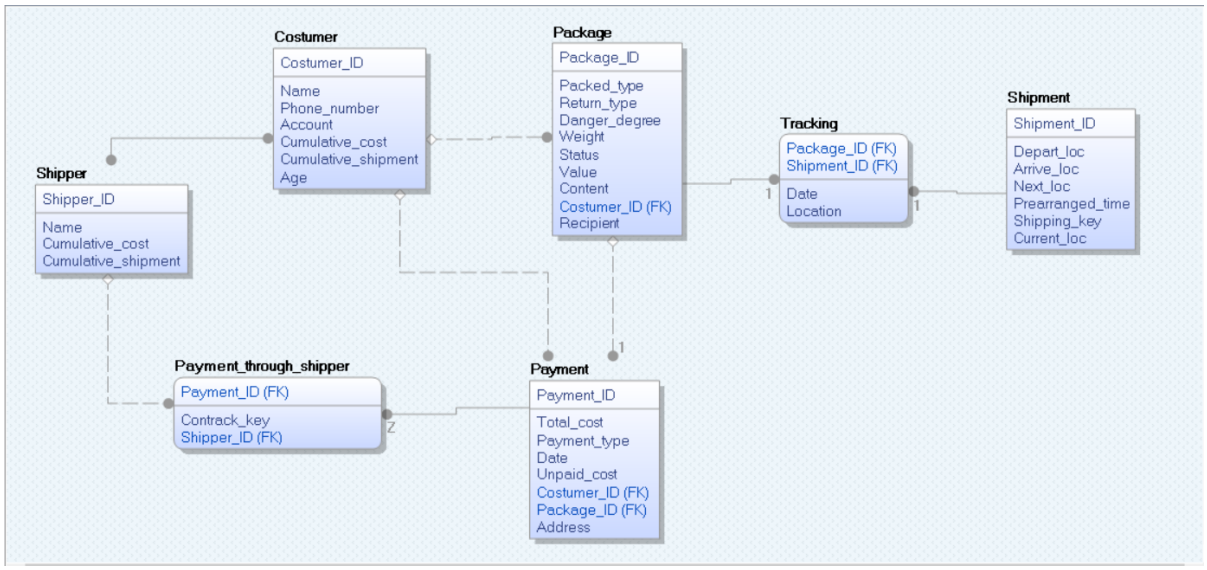
- costumer_package

costumer_package는 고객과 배송물의 관계를 나타내는 relationship이다. 어떠한 배송물은 한 명의 고객에 대응되며 한 명의 고객이 여러 개의 배송물을 주문할 수 있으므로 many-to-one의 관계를 가진다.

- tracking

tracking은 배송물이 배송되는 것에 대한 추적을 나타내는 relationship으로, 각 배송물에 대한 배송 정보는 하나의 배송에만 해당되므로 one-to-one 관계이다. relation attribute로 multi-value이자 composite attribute인 past_route를 가지는데, 해당 attribute는 과거 배송 경로를 추적하는 역할을 한다. past route는 component로 date와 loc를 가지며 각각 경로의 날짜와 장소를 나타낸다.

2. Schema Diagram(Erwin)



다음은 erwin을 통해 작성한 package delivery system의 schema diagram이다. 사용한 버전이 Erwin 실습에서 사용한 버전과 달라 relation을 나타내는 화살표의 모양이 상이하다. many-to-many와 many-to-one의 경우 many side는 원으로, one side의 경우 마름모 모양으로 표현된다. one-to-one 관계의 경우 한 side가 원으로 표현되고, 그 옆에 1이 표기된다. schema diagram에서 relation attribute를 가지는 relationship을 entity로 추가하였고 나머지 relationship은 one side의 PK가 many side의 FK로 추가된 것 이외에는 ERD와 같다. schema diagram 파트에서는 추가된 entity에 대해서만 서술한다.

- payment_through_shipper

contract_key를 relation attribute로 가지는 relationship인 payment_through_shipper를 entity로 추가하였다. 둘 모두 many-to-one 으로 연결하였다. 배송업체를 통한 결제는 FK인 결제ID를 통해 식별하도록 하였다.

- tracking

multi-value이자 composite인 past_route attribute를 가지는 relationship인 tracking을 entity로 추가하였다. erwin에서 composite attribute인 past_route의 component를 모두 attribute로 추가하였다. 해당 attribute인 date와 loc는 모두 multi-value인데, array type으로 추가하였다. 각 array의 index가 같은 element들이 하나의 past_route로 인식된다. tracking은 package와 shipment 모두 one-to-one 관계로 연결하였다. package_ID와 shipment_ID 모두를 PK로 사용하도록 하였다.

3. Query

i. Assume truck 1721 is destroyed in a crash. Find all customers who had a package on the truck at the time of the crash. Find all recipients who had a package on that truck at the time of the crash. Find the last successful delivery by that truck prior to the crash.

ii. Find the customer who has shipped the most packages in the past year.

iii. Find the customer who has spent the most money on shipping in the past year.

iv. Find those packages that were not delivered within the promised time.

v. Generate the bill for each customer for the past month. Consider creating several types of bills.
- A simple bill: customer, address, and amount owed.
- A bill listing charges by type of service.
- An itemize billing listing each individual shipment and the charges for it.

i.

```
select distinct costumer_ID, recipient
from costumer natural join package, shipment
where shipment_ID = "truck_1721";
```

ii.

```
select costumer_ID
from costumer order by cumulative_shipmet desc
limit 1;
```

iii.

```
select costumer_ID
from costumer order by cumulative_cost desc
limit 1;
```

iv.

```
select distinct package_ID
from package natural join shipment
where sysdate > prearranged_time;
```

v.

```
select *
from payment
where costumer_ID = 12345;
```

(해당 쿼리에서 청구서를 받고자 하는 고객의 ID가 12345라고 가정하였다.)