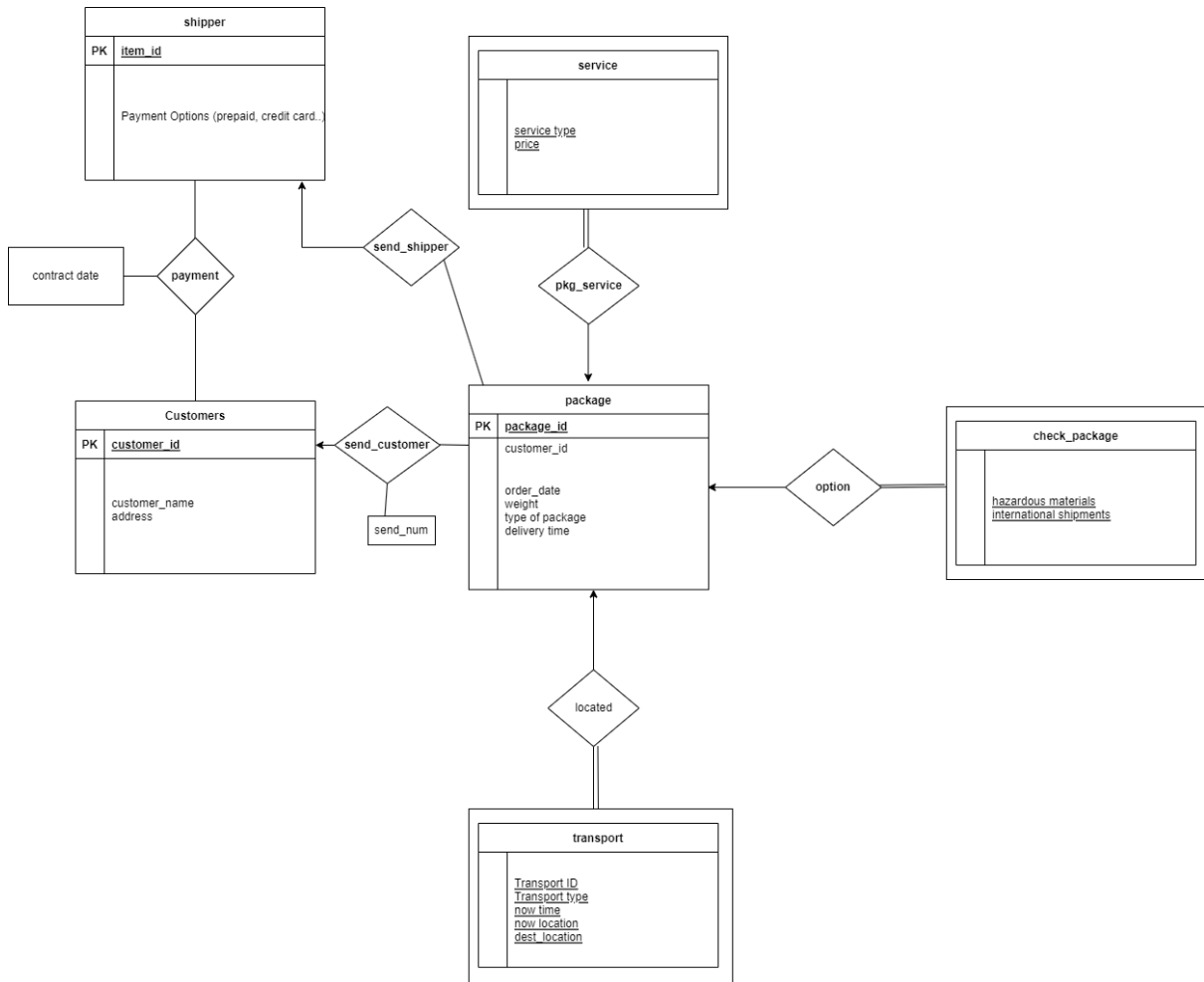


# [CSE4110-02] 데이터베이스 Project 1

20200118 조창희

## 1. E-R model



### Strong entity

#### (1) Customers

- A. Primary key: customer\_id
- B. Attribute: customer\_naem, address

각 고객의 정보를 담고 있는 entity이며 id를 primary key로 두었다.

#### (2) Package

- A. Primary key: package\_id

- B. Attribute: customer\_id (외래 키), order\_date(주문 날짜), weight, type of package, delivery time

운송물에 대한 정보를 갖고 있는 entity이다. 해당 다이어그램에서 운송물을 보내는 경우는 1. 고객이 직접 보내는 경우 2. 쇼핑몰(shipper)이 보내는 경우 두 가지이기 때문에, 각 entity에서 하지 않고 package entity에서 수행하여 속성 수를 줄이고자 하였다.

### (3) Shipper

- A. Primary key: item id

- B. Attribute: Payment Option

쇼핑몰에서 판매하는 item들을 식별하기 위해 따로 item id를 두었다. 또한 모든 item 배송비는 선불이거나, 착불이거나, 혹은 계약으로 맺은 형태이므로 세가지 중 하나를 적도로 payment option을 두었다.

## Weak entity

해당 다이어그램에서 약한 개체들은 모두 package에 의존하는 형태를 띄고 있다.

### (1) Service

Service는 package의 요소에 따라 결정되고, package가 없으면 service도 있을 수 없으므로 weak entity이다. Service의 종류와 service 가격이 식별자로 정의되어 있다.

### (2) Check\_package

Shipping company는 package의 전부를 볼 필요가 없고, package에서 위험하거나 관세 상품인 경우만 주시할 필요가 있다. 그런데 이 속성들을 package의 속성들로 두면 특수한 경우에 속하지 않는 운송물들이 모두 null이 될 것이다. 따라서 package 중에서도 특수한 경우에 대한 것을 따로 entity를 두었으며, 이것도 package에 의존적인 경우이므로 weak entity이다.

### (3) Transport

Shipping company는 운송물이 어디로 가는지를 주시할 필요가 있다. 하지만 배송 과정을 package의 속성으로 놓아버리면, 회사가 굳이 몰라도 되는 정보까지 항상 track할 수 있다. 따라서 이 경우를 따로 뺐으며, package가 없으면 존재할 수 없으므로 weak entity이다. 이곳은 운송 번호, 운송 수단, 현재 시간, 현재 위치, 목적 위치를 식별자로 두었다. 처음에는 이전 시간 또한 속성으로 넣으려고 했는데, 현재 시간을 알 수 있으며 주기적으로 갱신을 한다면 이전 시간을 넣었을 때 정보의 중복이 발생하므로 (이전 시간의 현재 시간 = 현재 시간의 이전 시간) 넣지 않았다. 이전 속성으로

올라가다 보면 과거의 시간과 이후 시간을 알 수 있을 것이다.

## Relationship set & Cardinality

### (1) Payment

Customer이 shipper에게 가격을 지불한다. 그리고 계약관계일 경우 계약 날짜를 따로 저장해두어 해당 날짜에 맞게 청구가 이루어지도록 하였다. 한 고객이 여러 shipper와 거래할 수도 있고, 한 shipper이 여러 고객과 거래할 수 있으므로 many to many 관계이다.

### (2) Send\_shipper

Shipper이 직접 배송하는 경우이다. 한 shipper이 여러 package를 보낼 수 있지만, 한 package가 여러 shipper에 의해 배송될 수는 없으므로 (모든 package가 식별가능하다는 전제 하에) one to many 관계이다.

### (3) Send\_customer

고객이 직접 배송하는 경우이다. 이것도 마찬가지로 한 고객이 여러 package를 보낼 수는 있지만, 한 package가 여러 고객에 의해 보내질 수는 없으므로 one to many 관계이다. 그리고 고객이 배송을 보내는 경우 배송 횟수를 기억하기 위해 send\_num을 추가적인 속성으로 넣어주었다.

### (4) Pkg\_service

모든 서비스는 package에 참여하므로 total participation 관계이다. 그리고 한 package는 하나의 서비스만 사용하므로 one 관계이다.

### (5) Located

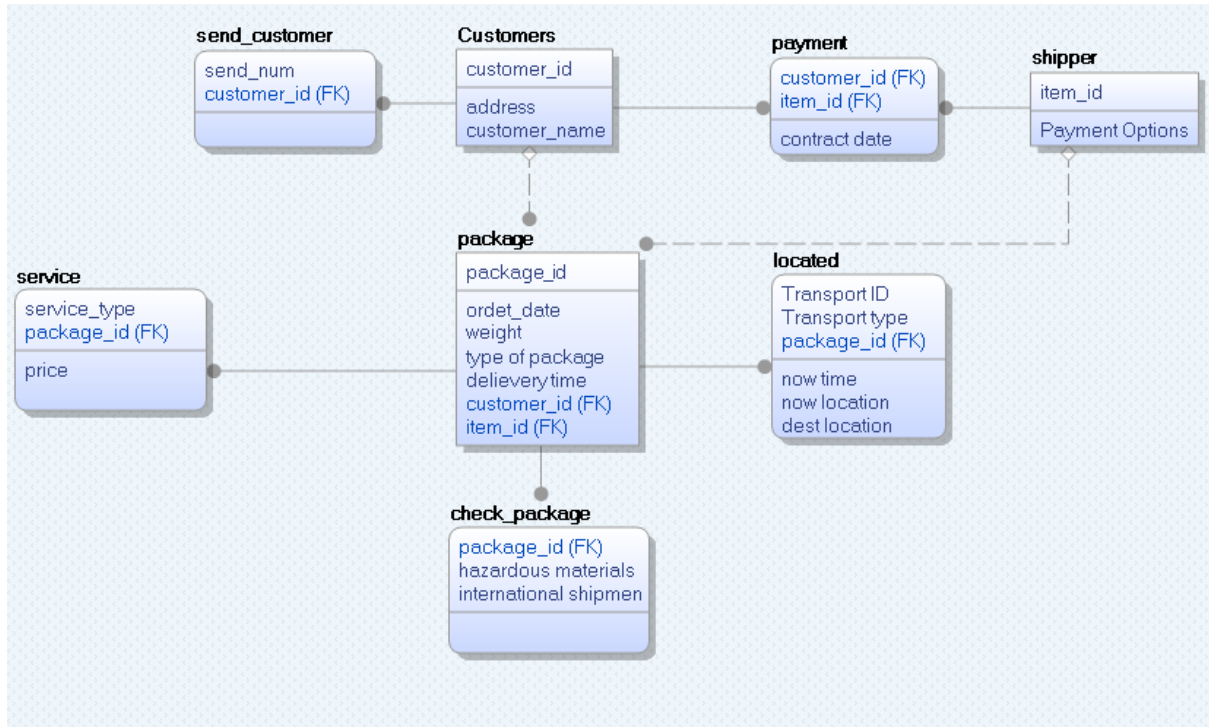
(4)과 같은 경우이다.

### (6) Option

(4)과 같은 경우이다.

## 2. Relational Schema diagram model

ER다이어그램에서 schema로 변환되는 과정을 중심으로 설명할 것이다.



- (1) Customer과 send\_customer은 one to one 관계이다. (버전 문제인지, one to one 관계로 설정하여도 one to many와 똑같이 보인다.) 또한, send\_customer은 customer이 있어야 존재하는 weak entity이다. Send\_customer의 primary key는 외래키인 customer\_id와 발송 횟수인 send\_num의 조합이다.
- (2) Customer과 shipper은 many to many관계이며, 양쪽 수를 가지므로 불가피하게 추가적인 릴레이션인 payment를 만들어주었다. 양쪽의 primary key의 조합이 payment의 primary key이다.
- (3) Shipper과 package는 강한 개체 간 관계이며, 비식별관계이므로 점선으로 표현된다. Customer과 package 또한 비식별관계이므로 점선으로 표기된다.
- (4) Package와 located, service, check\_package는 1 : N의 식별 관계이며 3 entity는 package가 있어야 존재하는 약한 개체이다. 따라서 package의 primary key와 각 개체의 식별자의 조합을 primary key로 사용하고 있다.