

# 과목 명: 데이터베이스시스템

담당 교수 명: 정성원

<<Project 1>>

20170795 문채원

## <목차>

### 1. Entity, Relationship 도출

1.1 Customer

1.2 Package

1.3 Recipient

1.4 Vehicle

1.5 Tracking

1.6 Bill

1.7 Billed By

1.8 Buy

1.9 Delivery

1.10 Sends

1.11 Delivered By

### 2. Relationship&Cardinality

2.1 Customer to Bill

2.2 Customer to Package

2.3 Package to Tracking

2.4 Package to Recipient

2.5 Tracking to Vehicle

### 3. Query 처리

### 4. Schema Diagram 으로의 Reduction

### 5. Schema Diagram

### 6. ER Diagram

## 1. Entity, Relationship 도출

### 1.1 Customer

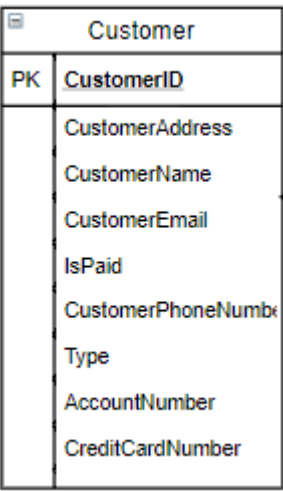
#### - Entity 생성 이유

Delivery Company가 Customer에 대한 정보를 관리한다고 했다. 따라서 Customer의 정보를 관리하기 위해 Customer라는 Entity를 생성하였다.

#### - Key 설정

각각의 Customer을 관리하기 위해 CustomerID라는 Primary Key를 설정했다. Customer마다 각자의 고유한 CustomerID를 가져야하며, 중복을 피하기 위해 등록일자와 등록 순서 등을 조합하여 생성한다.

#### - Attribute 도출

 A UML class diagram for the Customer entity. The class is named 'Customer'. It has a primary key attribute 'CustomerID' marked with 'PK'. Other attributes listed are 'CustomerAddress', 'CustomerName', 'CustomerEmail', 'IsPaid', 'CustomerPhoneNumber', 'Type', 'AccountNumber', and 'CreditCardNumber'. Customer PK CustomerID CustomerAddress CustomerName CustomerEmail IsPaid CustomerPhoneNumber Type AccountNumber CreditCardNumber	CustomerID	Primary Key로서 Customer를 구분할 수 있도록 한다
	CustomerAddress	Customer의 주소정보를 저장한다
	CustomerName	Customer의 이름정보를 저장한다
	CustomerEmail	Customer의 이메일 정보를 저장한다
	IsPaid	결제여부를 저장한다. Prepaid인 경우 결제하지 않는다
	CustomerPhoneNumber	Customer의 연락처정보를 저장한다
	Type	Frequent한 customer인지 infrequent한 customer인지 여부를 저장한다
	AccountNumber	Customer의 계좌정보를 저장한다
	CreditCardNumber	신용카드 정보를 저장한다

### 1.2 Package

#### - Entity 생성 이유

Delivery Company가 package에 대한 정보를 관리한다고 했다. 따라서 package의 정보를 관리하기 위해 package라는 Entity를 생성하였다.

#### - Key 설정

각각의 Customer을 관리하기 위해 packageID라는 Primary Key를 설정했다. package마다 각자의 고유한 packageID를 가져야하며, 중복을 피하기 위해 상품품번, 항목 등을 조합하여 생성한다.

### - Description

Package 정보에는 package와 관련된 물품 정보 뿐만 아니라 물품이 생기는 순간 배달해야하기때문에 배달관련 정보도 일부 포함하고 있다.

### - Attribute 도출

Package		PackageID	Primary Key로서 Package를 구분할 수 있도록한다
PK	PackageID	Weight	Package의 무게정보를 저장한다
	Weight	Package Type	패키지 포장 정보 (ex. flat envelope, small box, larger boxex)를 저장한다
	PackageType	Timeliness	패키지 배달 정보(overnight, second day, or longer)를 저장한다.
	Timeliness	ProductType	
	ProductType	Departure Time	Package의 배달 출발시간 정보를 저장한다
	DepartureTime	Arrival Time	Package의 배달 도착시간 정보를 저장한다
	Hazardous	Hazardous	Package의 위험한 물질을 포함하는지를 저장한다
	ArrivalTime	International	Package가 international shipments에 해당하는지 저장한다

## 1.3 Recipient

### - Entity 생성 이유

Delivery Company가 배송을 할 주소 정보를 관리해야하기 때문에 만들었다.

### - Key 설정

각각의 Recipient는 RecipientName과 RecipientAddress를 함께 Primary key로 저장한다. 한 주소에 같은 이름의 가진 사람이 하나 이상 있지 않기 때문이다.

### - Attribute 도출

	RecipientName	Primary Key로서 수신자의 이름정보를 저장한다
	RecipientAddress	Primary Key로서 수신자의 주소정보를 저장한다

<b>Recipient</b> <div> RecipientName  RecipientAddress  RecipientPhoneNumber  PackageID </div>	RecipientPhoneNumber	수신자의 연락처 정보를 저장한다
	PackageID	수신자가 받게될 상품 정보를 저장한다

#### 1.4 Vehicle

##### - Entity 생성 이유

Delivery Company가 제품 위치 트래킹에 대해 vehicle 관점에서의 정보를 관리한다고 했다. 따라서 Vehicle entity를 생성해 vehicle의 정보를 관리한다.

##### - Key 설정

각각의 vehicle을 관리하기 위해 vehicleID를 primary key로 생성했다. Vehicle ID는 고유하게 Vehicle을 나타낼 수 있게 plate number, vehicle type 등을 조합해서 생성한다.

##### - Attribute 도출

<b>Vehicle</b> <div> VehicleID  Name  Type  PlateNum </div>	VehicleID	Primary Key로서 vehicle을 구분할 수 있도록한다
	Name	Vehicle의 이름정보를 저장한다
	Type	Vehicle Type(motorcycle, airplane) 정보를 저장한다
	PlateNum	Vehicle의 차량 번호판 정보를 저장한다

#### 1.5 Tracking

##### - Entity 생성 이유

Delivery Company가 package에 대한 정보를 관리한다고 했다. 따라서 package의 정보를 관리하기 위해 package라는 Entity를 생성하였다.

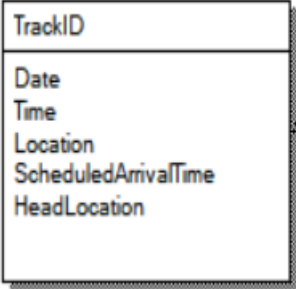
Delivery Company가 제품의 현위치,

##### - Key 설정

각각의 Customer을 관리하기 위해 packageID라는 Primary Key를 설정했다. package 마

다 각자의 고유한 packageID를 가져야하며, 중복을 피하기 위해 상품품번, 항목 등을 조합하여 생성한다.

- Attribute 도출

	TrackID	Primary Key로서 Tracking을 구분할 수 있도록 한다
	Date	Tracking의 날짜정보를 저장한다
	Time	Tracking의 시간정보를 저장한다
	Location	Tracking의 현재 장소정보를 저장한다
	ScheduledArrivalTime	Tracking의 예정된 도착정보를 저장한다
	HeadLocation	Tracking의 이동할 장소정보를 저장한다 (head)

## 1.6 Bill

- Entity 생성 이유

Delivery Company가 Bill에 대한 정보를 관리한다고 했으며, Bill을 여러 타입별로 보여줄 수 있어야한다고 했다. 따라서 bill의 정보를 관리하기 위해 bill라는 Entity를 생성하였다.

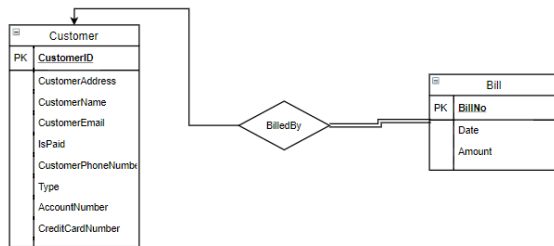
- Key 설정

각각의 Customer을 관리하기 위해 billNo라는 Primary Key를 설정했다. BillNo는 영수증 번호로서, 영수증이 매겨진 순서를 나타내며 고유하게 bill relationship을 나타낸다.

- Attribute 도출

	BillNo	PrimaryKey로서 영수증을 구분할 수 있도록한다
	Date	영수증의 날짜정보를 저장한다
	Amount	영수증의 총 구매액을 저장한다

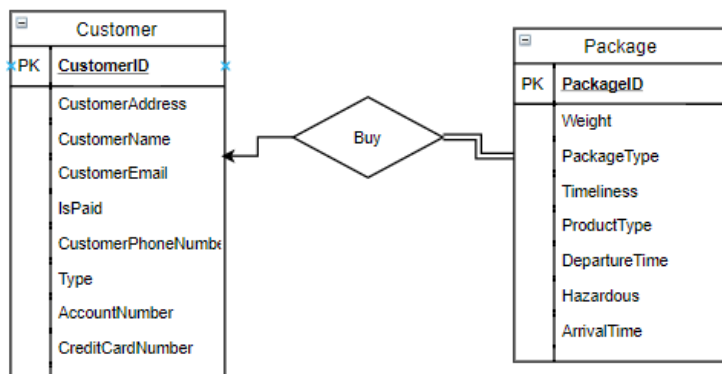
## 1.7 Billed By



#### - Relationship 생성 이유

고객이 구매한 영수증 정보를 관리하기 위해 생성하였다. 명세서에서 영수증 정보를 관리하라고 명시되어있는데, 첫번째와 세번째 영수증 정보를 billed by relationship을 통해 얻을 수 있다. itemize billing list를 고객정보에서 type과 IsPaid 속성을 활용하여 추출할 수 있으며, simple bill은 customer의 name, address 등의 속성을 활용하여 추출할 수 있다.

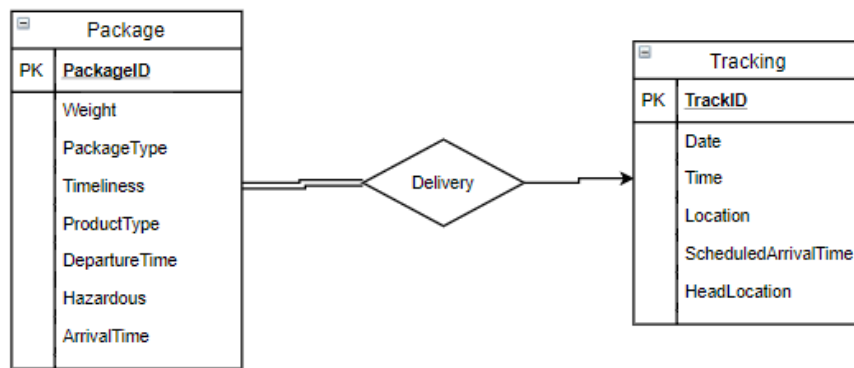
### 1.8 Buy



#### - Relationship 생성 이유

고객이 구매한 물품 정보를 나타내기 위해 생성했다. 고객이 구매한 물품 리스트를 뽑을 수 있게해, 영수증을 발행할 때나 tracking을 할 때 사용될 수 있다.

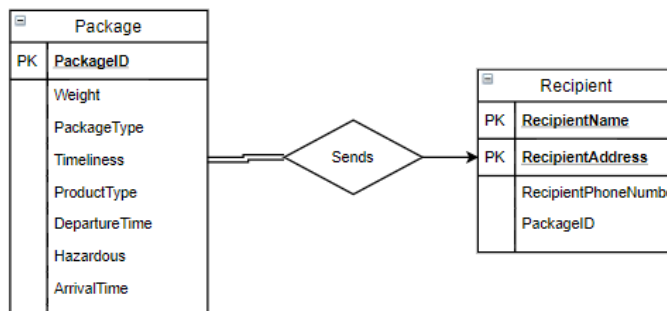
### 1.9 Delivery



#### -Relationship 생성 이유

명세서에서 트래킹과 관련된 정보를 생성하라고 했으며, 물품의 배송정보를 나타내기 위해 생성했다. 물품마다 다른 트래킹 정보를 가지고 있으며, 이를 통해 물품의 현재 위치, 도착예정시각, 향하는 장소 등의 정보를 얻을 수 있다.

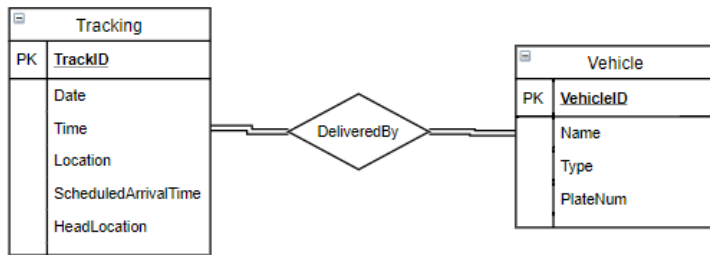
#### 1.10 Sends



물품이 보내지는 대상에 대한 정보를 나타낸다. 패키지와 조인을 통해 패키지가 현재 있는 위치가 수신자의 주소가 같다면 상품이 도착한 것으로 확인할 수 있다.

#### 1.11 Delivered By



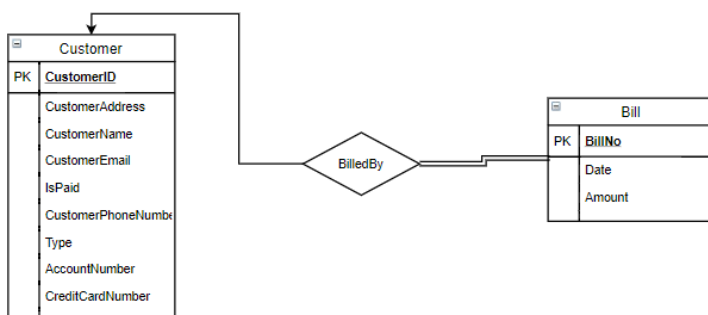


relationship 생성 이유

트래킹할 때 어떤 운송수단을 썼는지 확인할 수 있다. 또, 명세서에서 vehicle의 입장에서 시점에서 tracking을 바라보고자 한다고 했다. delivered by relationship을 통해 vehicle과 트래킹간의 정보를 vehicle관점에서 바라볼 수 있다.

## 2. Relationship&Cardinality

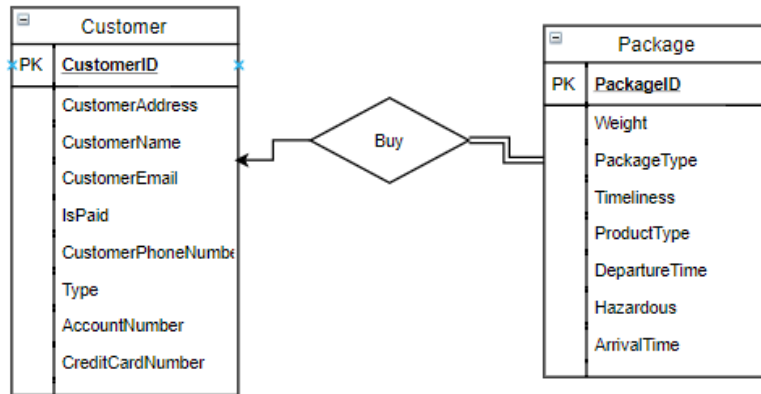
### 2.1 Customer to Bill



- Relation name : Billed by
- Cardinality : 1: N relationship, Zero, One or More
- Description :

회원이 구매할 경우 영수증 정보를 관리할 수 있도록 회원정보와 영수증 정보 사이에 binary relationship을 설정하였다. 한 회원이 여러개의 영수증을 가질 수 있으므로 1:N Cardinality를 갖는다. 또, 회원이 회원가입 만하고 구매를 하지 않은 경우 영수증이 발행되지 않으므로 Zero, one or More relationship으로 설정했다.

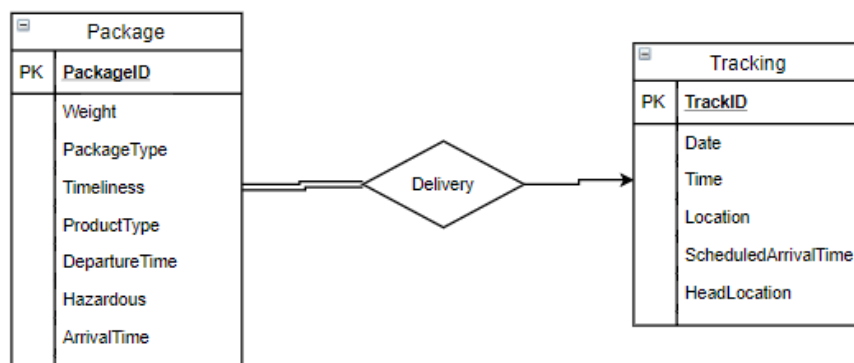
## 2.2 Customer to Package



- Relation Name : Buy
- Cardinality : 1:N relationship, Zero, One or More
- Description :

회원이 구매할 때 구매 물품 정보를 관리할 수 있도록 회원정보와 구매물품 정보 사이에 binary relationship을 생성하였다. 한 회원이 여러개의 물품을 구매할 수 있기 때문에 1:N cardinality로 설정했다. 또, 회원이 회원가입만 하고 물품을 구매하지 않은 경우가 있을 수 있으므로 Zero, One or More로 설정했다.

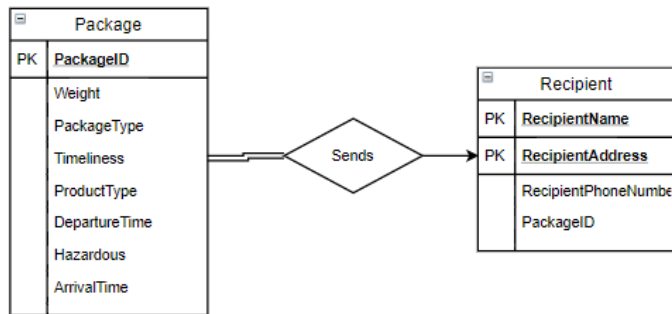
## 2.3 Package to Tracking



- Relation name : Delivery
- Cardinality : 1: N, One or More
- Description : 제품이 발생했을 때 제품을 tracking하기 위해 트래킹 정보를 나타내는 entity와의

relation을 형성하였다. package는 하나의 트래킹 정보를 갖지만 트래킹의 경우 여러개의 상품을 운반할 수 있으므로 1:N cardinality로 설정했다. 패키지가 없는 경우 트래킹 정보가 생길 수 없으며 트래킹 정보 또한 패키지가 있어야하므로 One or More로 설정했다.

## 2.4 Package to Recipient



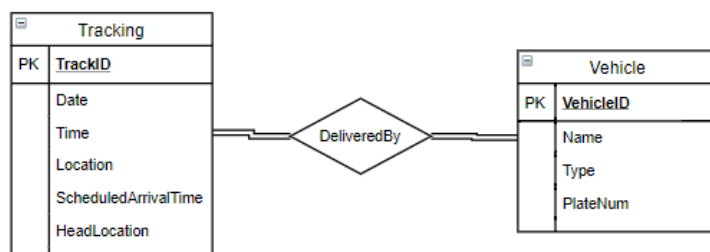
Relation Name : Sends

Cardinality : N:1, One or More

Description:

상품과 수령인 사이의 정보를 나타내기 위해 Sends relation을 만들었다. 상품은 하나의 Recipient를 가질 수 있는 반면에 Recipient는 여러개의 상품을 받을 수 있기 때문에 N:1 Cardinality로 설정했다. 또, 패키지와 수령인 역시 패키지가 0인 경우 수령인이 존재하지 않으므로 One or More로 설정했다.

## 2.5 Tracking to Vehicle



Relation Name : DeliveredBy

Cardinality : N:N

Description:

트래킹 정보와 vehicle 사이의 정보를 나타내기 위해 binary relationship을 설정하였다. 트래킹마다 여러개의 vehicle을 사용할 수 있고 vehicle도 여러개의 tracking을 가질 수 있으므로 N:N Cardinality를 갖는다.

### 3. Query 처리

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

위의 쿼리는 작년에 가장 많은 상품을 배송시킨 고객을 찾는다. 이는 Bill, Customer간의 relationship을 통해 해결할 수 있다. Bill relationship을 date를 작년 속성으로 준 후, customerID 별로 grouping 한 후, count로 개수를 센 후에 max(count)값의 CustomerID를 출력시키면 된다.

쿼리문은 다음과 같다

```
select name, max(num) from(
```

```
select Customer.ID name, count(amount) num
```

```
from Customer, bill
```

```
where Customer.CustomerID = bill.CustomerID and bill.date>=20190101 and bill.date<20200101
```

```
group by Customer.CustomerID
```

```
)
```

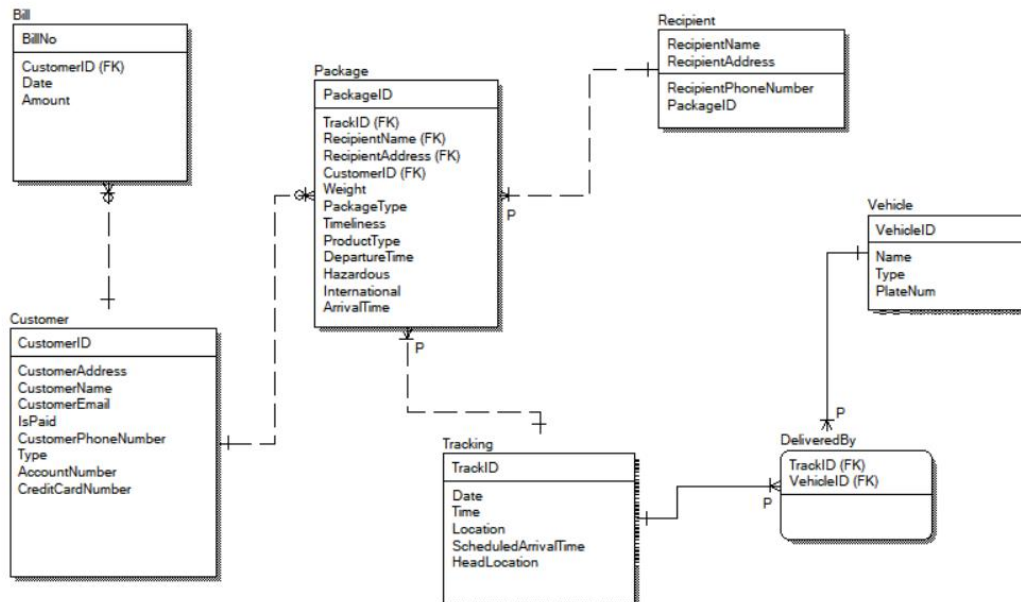
### 4. Schema Diagram으로의 Reduction

ER diagram에서 schema diagram으로 위의 cardinality를 바탕으로 reduction했다. 달라진 부분은 identification과 many-to-many relationship reduction이다. 모든 entity들이 그 자체로서 존재하며 fk가 pk가 될 필요가 없기 때문에 non-identifying relationship으로 설정했다. cardinality 부분은 위의 2절에서 설명한 부분과 동일하다.

many to many관계인 tracking과 vehicle을 reduction할 때 relation인 delivered by를 기준으로 각 vehicle과 tracking 사이의 관계를 따져보았다. delivered by는 하나의 vehicle을 가져야하는 반면에 vehicle은 여러개의 delivered by를 가질 수 있으므로 vehicle 사이에서는 many to one relationship을 설정했다. tracking과의 관계에 있어서도 역시 delivered by는 하나의 tracking을 가질 수 있는 반면에 tracking은 여러 delivered by를 가질 수 있기 때문에 이 역시 many to one relationship으로 설정했다. 그리고 delivered by는 tracking과 vehicle을 primary key로 갖기 때문에

identifying relationship으로 설정했다.

## 5. Schema Diagram



## 6. ER Diagram

