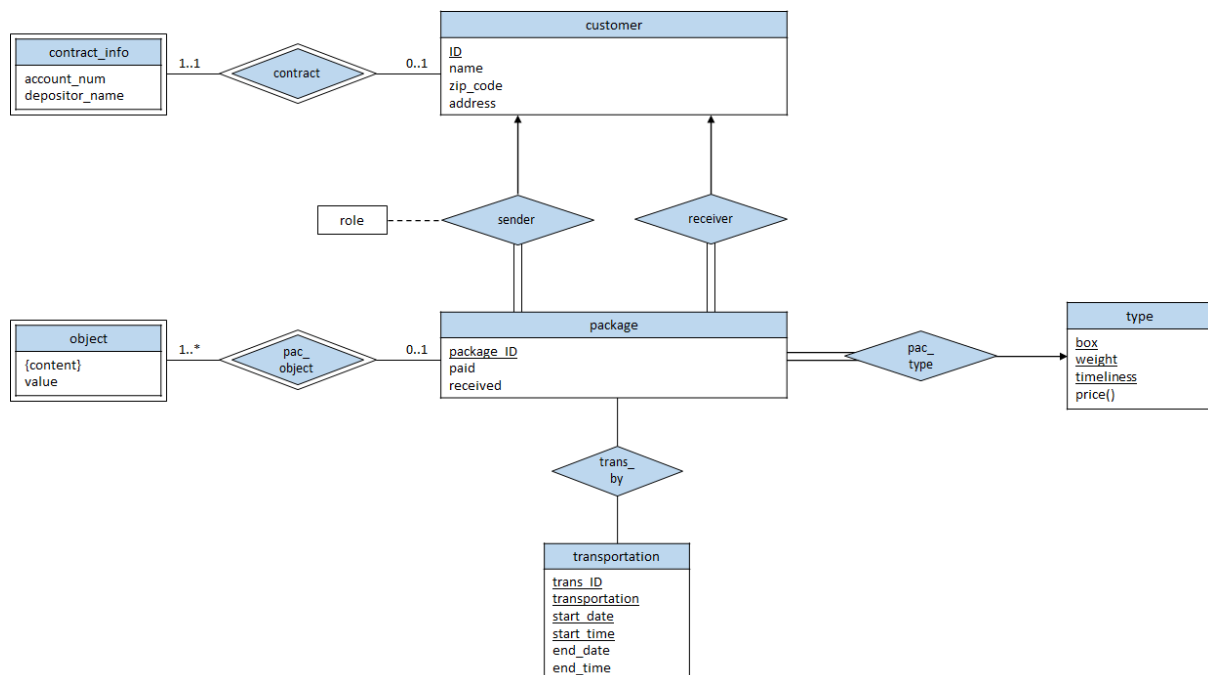


PROJECT 1 REPORT

20211584 장준영

1. E-R Model



1) Entities

- customer

택배 회사를 이용하는 고객들의 정보를 저장하는 entity이다. 고객마다 고유한 식별자(ID)를 가지므로 ID를 통해 모든 고객을 구별할 수 있다. 따라서 ID가 primary key이다. 고객이 택배를 받거나 보낼 때의 필수적 정보인 이름(name), 우편번호(zip_code), 주소(address)를 저장한다.

- contract_info

택배 회사와 계약을 맺어 달에 한 번 택배비를 정산하는 고객들의 정보를 저장하는 entity이다. customer에 완전히 종속되므로 customer를 identifying entity로 갖는 weak entity이다. 택배 회사 측으로 입금을 하는 계좌번호(account_num)와 입금주명(depositor_name)을 attribute로 갖는다. 계약 고객 역시 고객이므로 primary key로는 customer의 ID를 사용한다.

- package

보내고자 하는 포장된 물건의 정보를 저장하는 entity이다. 배송물마다 고유한 운송장 번호(package_ID)를 가지므로 package_ID를 primary key로 이용한다. 택배 서비스 제공으로 지불되어야 하는 돈을 모두 받았다면 paid에 True를, 아니면 False를 저장한다. 이는 선불과 착불을 구분하거나, 계약 고객이 제대로 송금을 하였는지 검증하기 위해 사용된다. 마찬가지로, 택배가 수신자에게 제대로 도착했다면 received에 True를, 아니면 False를 저장한다.

- object

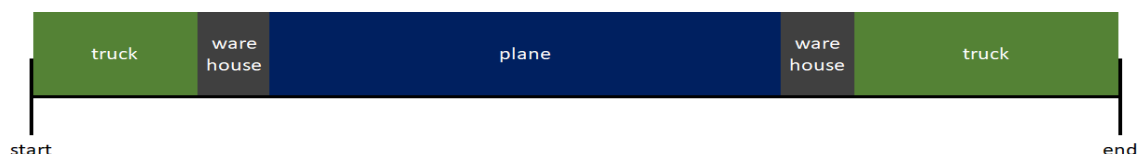
택배로 보내는 물건이 위험하거나, 세관 신고가 필요한 국제 택배인 경우 택배의 내용물과 가격을 알아야 한다. 이런 경우 object는 택배에 실린 물건과 그것의 가격을 저장하는 entity이다. package에 완전히 종속되므로 package를 identifying entity로 갖는 weak entity이다. 필요한 정보인 내용물(content)와 가격(value)을 저장하고, 국제 택배의 세관 신고의 경우 택배의 내용물을 모두 신고해야 하므로 content는 multivalued attribute로 정의하였다.

- type

택배 회사에서 지원하는 택배의 종류를 구분하여 저장하는 entity이다. 택배의 구분 요소인 (1)패키지 박스의 종류(box), (2)패키지의 무게(weight), (3)예정된 배송 완료 시간(timeliness)를 attribute로 가지고 있다. 또, 이 세 가지 요소에 의해 서비스의 가격(price())이 결정된다고 가정하였기 때문에 price()는 세 요소에 의한 derived attribute로 정의하였다. box는 envelope, smallb, largeb를 값으로 갖는다. weight는 kg 단위로 floor 연산을 한 값을 저장한다(0.69kg >> 0, 4.59kg >> 4). timeliness는 걸리는 날의 수에 24를 곱한 값을 저장한다(second day >> 48, fourth day >> 96). overnight의 경우 12를 저장한다.

- transportation

택배가 배송되는 과정을 저장하는 entity이다. 포함하는 정보는 이동수단 ID(trans_ID), 이동수단의 종류(transportation), 출발 날짜(start_date), 출발 시간(start_time), 도착 날짜(end_date), 도착 시간(end_time)이 있다. 수송 과정은 다음과 같은 과정으로 이루어진다.



(1) 배송의 시작과 끝은 항상 트럭이다. 배송 시작 시각은 첫 번째 트럭의 출발 시간, 도착 시각은 마지막 트럭의 도착 시간으로 간주한다. (따라서, 배송 소요 시간은 마지막 트럭의 도착 시간에서 첫 번째 트럭의 출발 시간을 뺀 것으로 계산할 수 있다.)

(2) 특정 운송 수단이 도착점까지 이동하면서 중간 경유지에 물건을 내리는 경우는 없다. 물건을 내리는 경우 그곳을 도착점으로 하고, 다시 시작점을 설정하여 출발해야 한다. 따라서 start_date,

start_time을 primary key로 설정할 수 있다.

(3) 택배가 운송 수단에 포함되어 있지 않는 시간은 없다고 간주한다. 출발부터 도착까지 항상 특정 truck, plane, warehouse에 포함되어 있다.

2) Relationships

- contract

customer과 contract_info를 연결하는 relationship이다. identifying relationship이므로 double diamond로 표기한다. customer 중 계약 고객 각각의 계좌 번호와 입금주명을 연결해준다. contract_info는 customer에 종속적이기 때문에 관계에 전부 참여하고, 하나의 계약 정보를 서로 다른 고객이 공유할 수 없으므로 하나를 넘는 관계를 가질 수는 없다. 또, 고객은 계약을 하지 않은 경우 계약 정보와 관계가 없을 수 있지만, 계약을 했다면 단 한 개의 계약 정보와 관계를 갖는다.

- sender

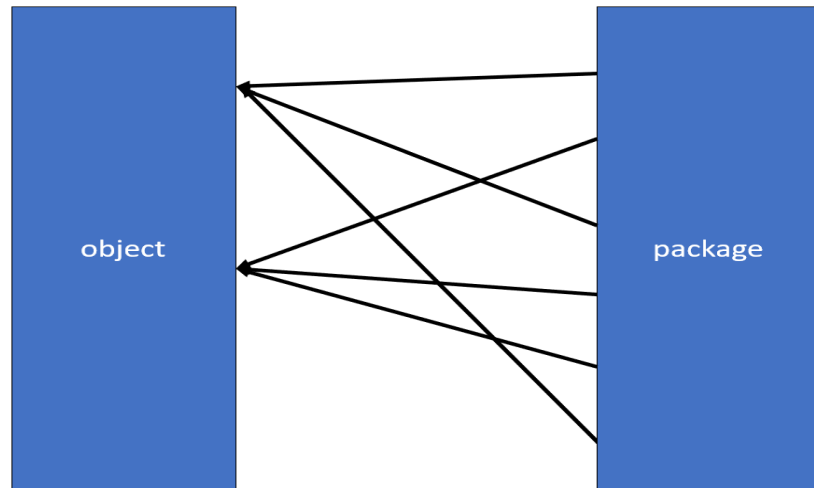
customer과 package를 송신자의 관계로 연결하는 relationship이다. 제시된 조건에 따르면 sender는 크게 customer, shipper, shipping company 세 가지로 분류할 수 있다. 이 특성을 customer에 넣기에는 customer가 receiver가 될 수도 있고, package에 넣기에는 package와 관련된 정보가 아니므로, relationship인 sender의 attribute로 설정하였다. package가 존재한다는 것은 정해진 한 명의 sender가 반드시 존재한다는 뜻이고, 하나의 package가 여러명의 sender에 의해 보내질 수 없으므로 customer(sender)-package 관계는 package가 모두 참여하는 One-to-Many 관계이다.

- receiver

customer과 package를 수신자의 관계로 연결하는 relationship이다. package가 존재한다는 것은 정해진 한 명의 receiver가 반드시 존재한다는 뜻이고, 하나의 package가 여러명의 receiver에게 보내질 수 없으므로 customer(receiver)-package 관계는 package가 모두 참여하는 One-to-Many 관계이다.

- pac_object

package와 object를 연결하는 relationship이다. identifying relationship이므로 double diamond로 표기한다. package 중 object에 해당하는 속성을 명시해야 하는 경우 연결해준다. object는 package에 종속적이기 때문에 관계에 전부 참여한다. 서로 다른 package가 완전히 같은 내용을 배송할 수 있고, 하나의 package가 두 개 이상의 object와 관계될 수 없으므로, object가 모두 참여하는 One-to-Many 관계이다.

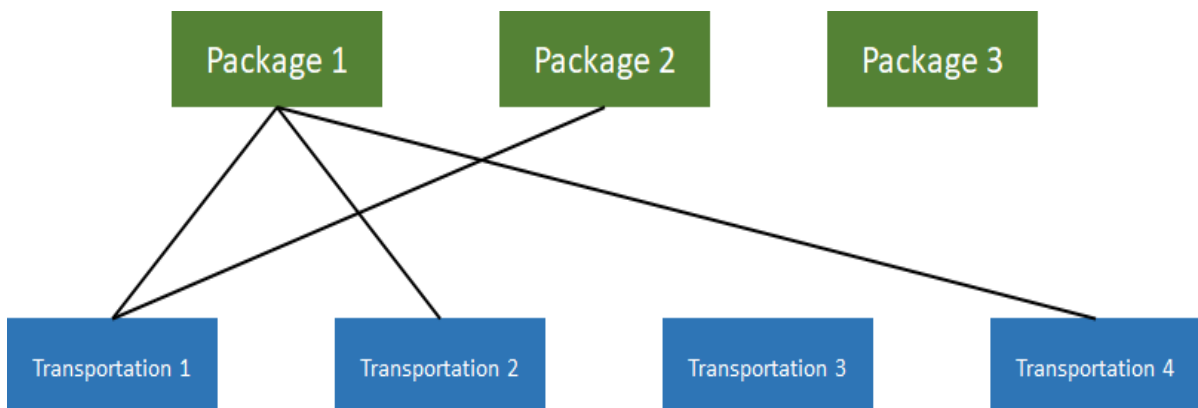


- pac_type

package와 type을 연결하는 relationship이다. type에는 package가 반드시 지니고 있는 기본 속성들이 있으므로 모든 package는 적어도 하나의 type을 가지고, package마다 정해진 하나의 값을 기준으로 type을 정하므로 하나를 초과하는 type을 가질 수 없다. 하지만 속성이 완벽하게 같은 패키지의 경우 같은 타입을 가질 수 있으므로 package-type 관계는 package가 모두 참여하는 Many-to-One 관계이다.

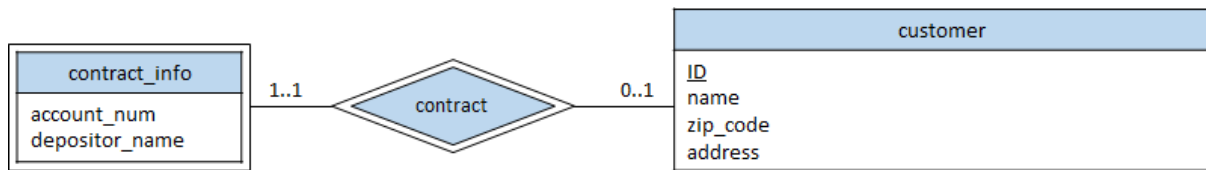
- trans_by

package와 transportation을 연결하는 relationship이다. trans_by는 package마다 배송 과정에서 거처간 모든 transportation을 연결해준다. 둘의 관계는 다음과 같다.



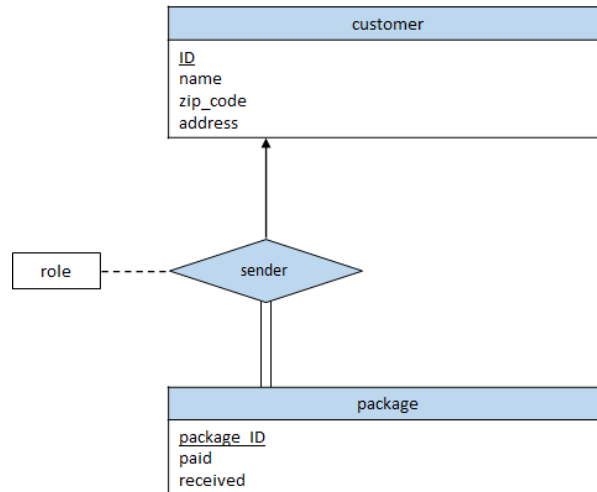
Package 1은 Transportation 1, 2, 4로 수송되었다. Package 2는 Transportation 1로만 수송되었다. Package 3은 아직 아무런 Transportation로도 수송되지 않았다. Transportation 3은 어떤 지점으로 이동하는 동안 아무런 Package도 실지 않았다. 위 diagram을 통해, 둘이 Many-to-Many 관계임을 알 수 있다.

3) Reducing to Schema



contract_info(ID, account_num, depositor_name)

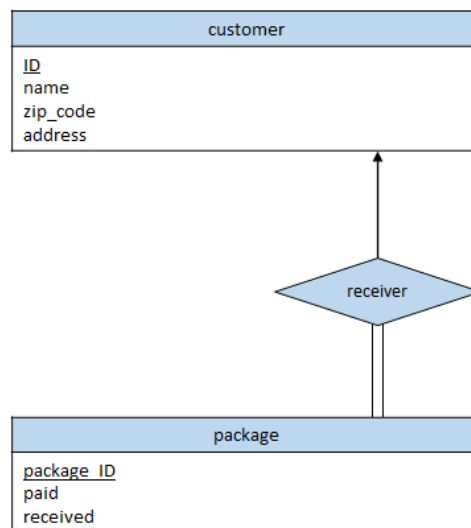
customer(ID, name, zip_code, address)



customer(ID, name, zip_code, address)

package(package_ID, paid, received)

sender(package_ID, ID, role)

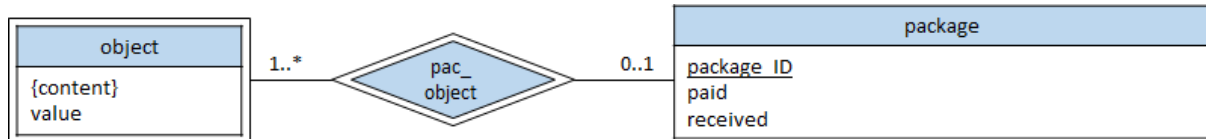


customer(ID, name, zip_code, address)

package(package_ID, paid, received)

receiver(package_ID, ID)

※sender와 receiver가 customer와 package를 완전히 동일하게 관계시키므로, “One” side의 primary key를 “Many” side (total)의 attribute로 포함시키면 두 관계를 혼동하게 된다. 따라서, 관계의 이름을 가진 schema를 각각 만들어 사용한다.

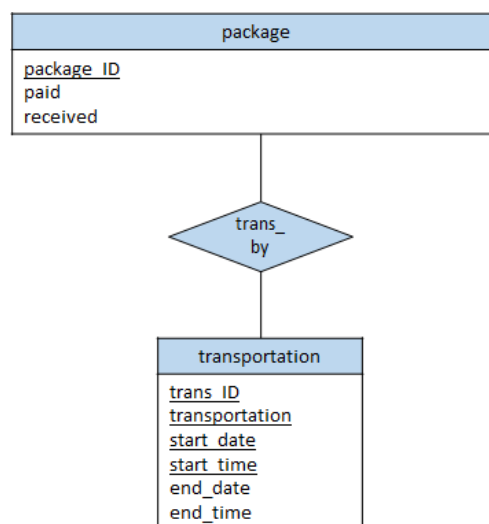


package(package_ID, paid, received)

object(package_ID, value)

obj_content(package_ID, content)

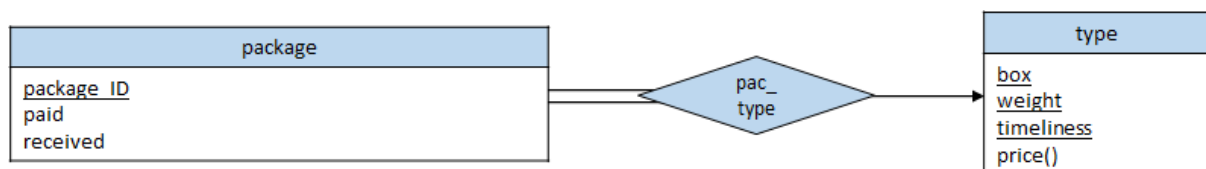
※pac_object(package_ID)는 이미 만들어진 schema에 의해 중복되기 때문에 굳이 만들지 않는다. multivalued attribute인 {content}는 obj_content schema를 통해 따로 구분해준다.



package(package_ID, paid, received)

trans_by(package_ID, trans_ID, transportation, start_date, start_time)

transportation(trans_ID, transportation, start_date, start_time, end_date, end_time)



type(box, weight, timeliness)

package(package_ID, paid, received, box, weight, timeliness)

► contract_info(ID, account_num, depositor_name)

customer(ID, name, zip_code, address)

package(package_ID, paid, received, box, weight, timeliness)

sender(package_ID, ID, role)

receiver(package_ID, ID)

object(package_ID, value)

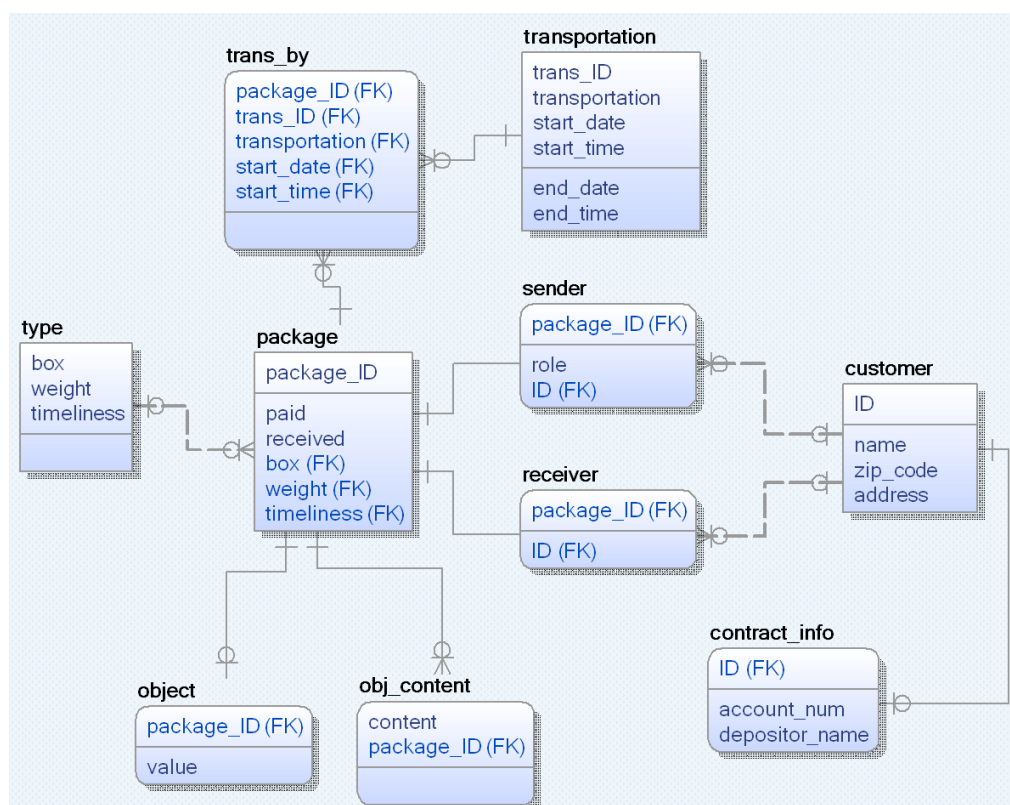
obj_content(package_ID, content)

trans_by(package_ID, trans_ID, transportation, start_date, start_time)

transportation(trans_ID, transportation, start_date, start_time, end_date, end_time)

type(box, weight, timeliness)

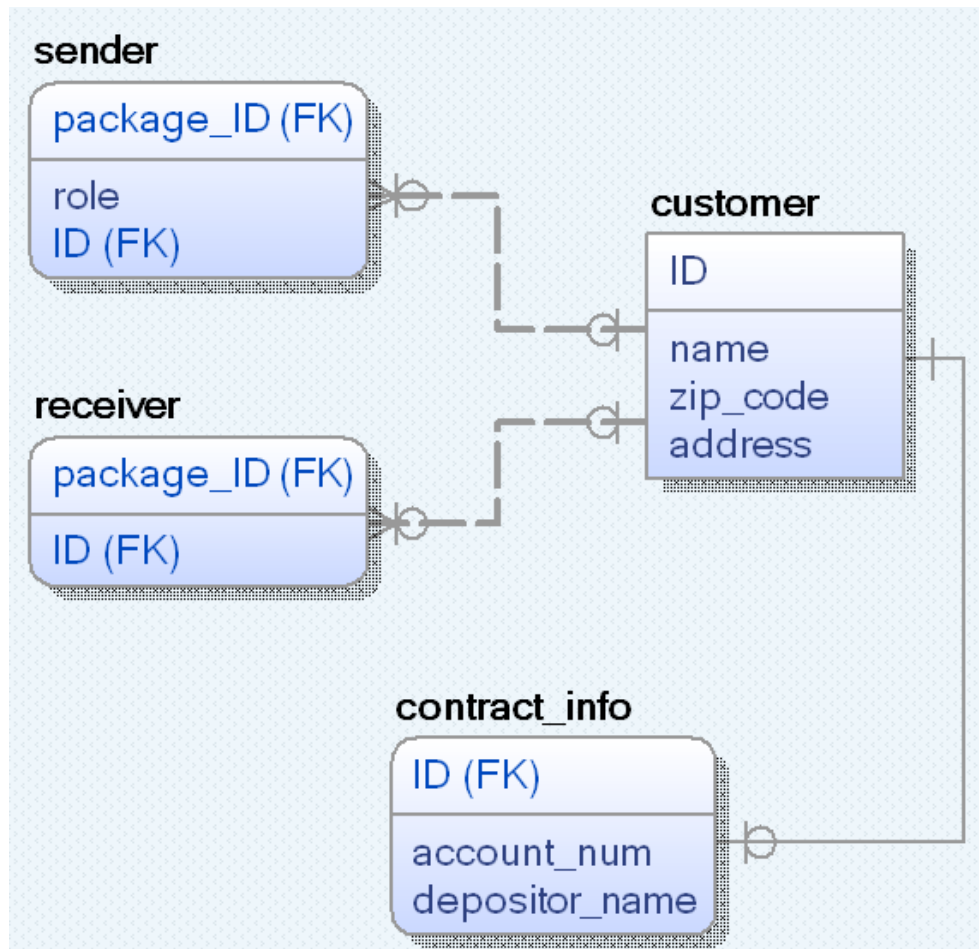
2. Relational Scheme Diagram



(※편의성을 위해 Logical Notation을 IDEF1x에서 Information Engineering(IE)로 변경하였음.)

각각의 entity와 그의 attribute는 앞서 설계한 ER Model을 관계형 schema로 만든 것과 동일한 모습을 보인다. transportation의 end_date, end_time은 null을 허용하고, 둘을 제외한 모든 attribute는 not null로 null을 허용하지 않는다.

1) customer→



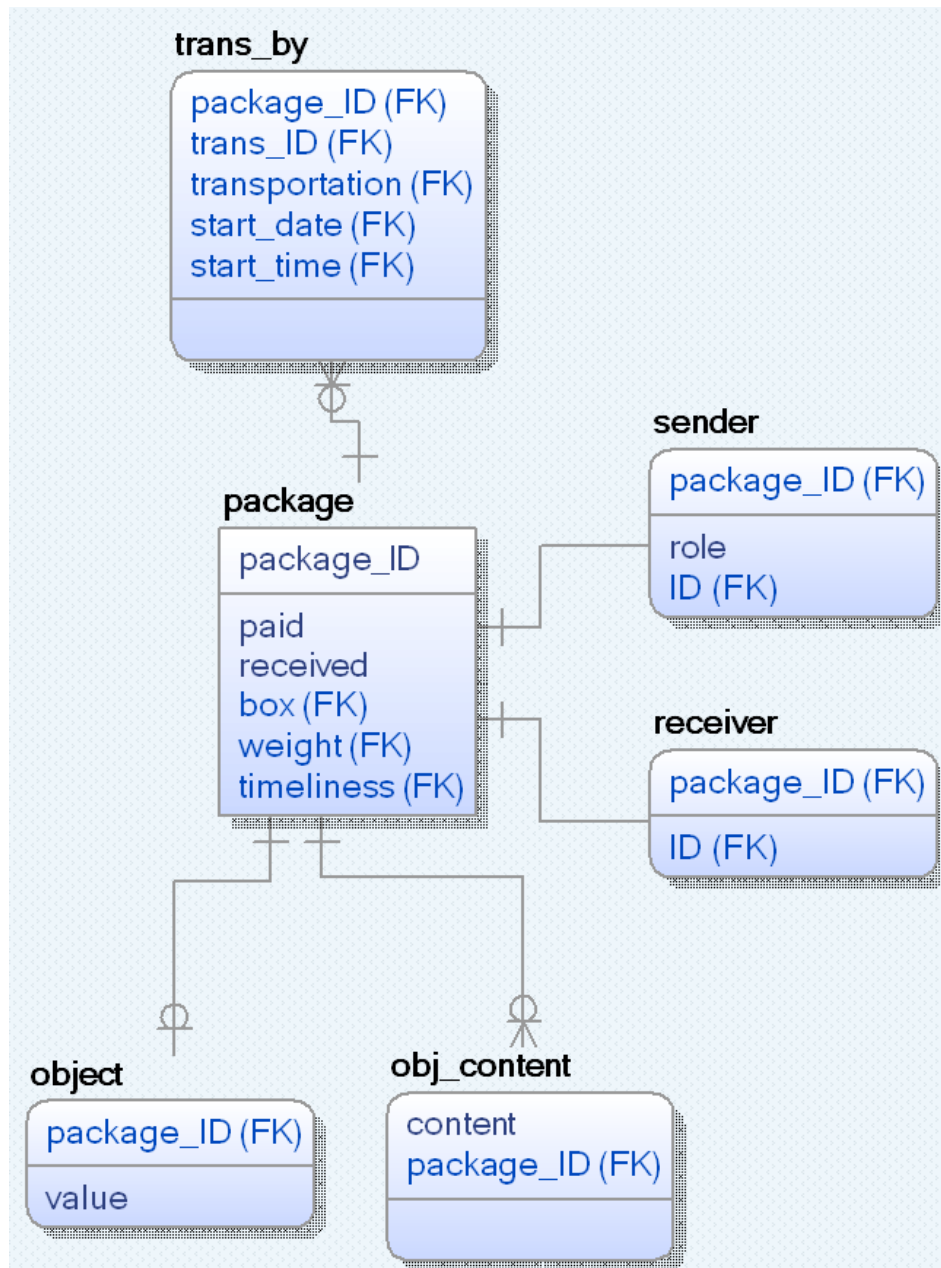
customer의 primary key인 ID는 contract_info, receiver, sender의 attribute가 되어 foreign key 역할을 한다.

→contract_info : foreign key인 ID가 contract_info의 primary key 역할을 하고 있으므로 contract_info는 weak entity가 되고 둘의 관계는 identifying하다. customer는 계약을 했을 수도 하지 않았을 수도 있고, 계약을 한 customer만 contract_info를 하나 가지므로 cardinality는 Zero or One이 된다.

→sender : foreign key인 ID가 sender의 primary key가 아니므로 둘의 관계는 non-identifying하다. customer는 아무 택배도 보내지 않았을 수도 있고, 한 개 이상의 택배를 보냈을 수도 있기 때문에 cardinality는 Zero, One or More이다.

→receiver : foreign key인 ID가 receiver의 primary key가 아니므로 둘의 관계는 non-identifying하다. customer는 아무 택배도 받지 않았을 수도 있고, 한 개 이상의 택배를 받았을 수도 있기 때문에 cardinality는 Zero, One or More이다.

2) package→



package의 primary key인 package_ID는 object, obj_content, sender, receiver, trans_by의 attribute가 되어 foreign key 역할을 한다.

→object : foreign key인 package_ID가 object의 primary key 역할을 하고 있으므로 object는 weak entity가 되고 둘의 관계는 identifying하다. package는 가치를 알아야 하는 경우와 알지 않아도 되는 경우가 있고, 알아야 하는 경우에만 object를 하나 가지므로 cardinality는 Zero or One이 된다.

→obj_content : foreign key인 package_ID가 obj_content의 primary key 역할을 하고 있으므로 obj_content는 weak entity가 되고 둘의 관계는 identifying하다. package는 내용물을 알아야 하는 경우와 알지 않아도 되는 경우가 있고, 알아야 하는 경우에만 obj_content를 하나 이상 가지므로

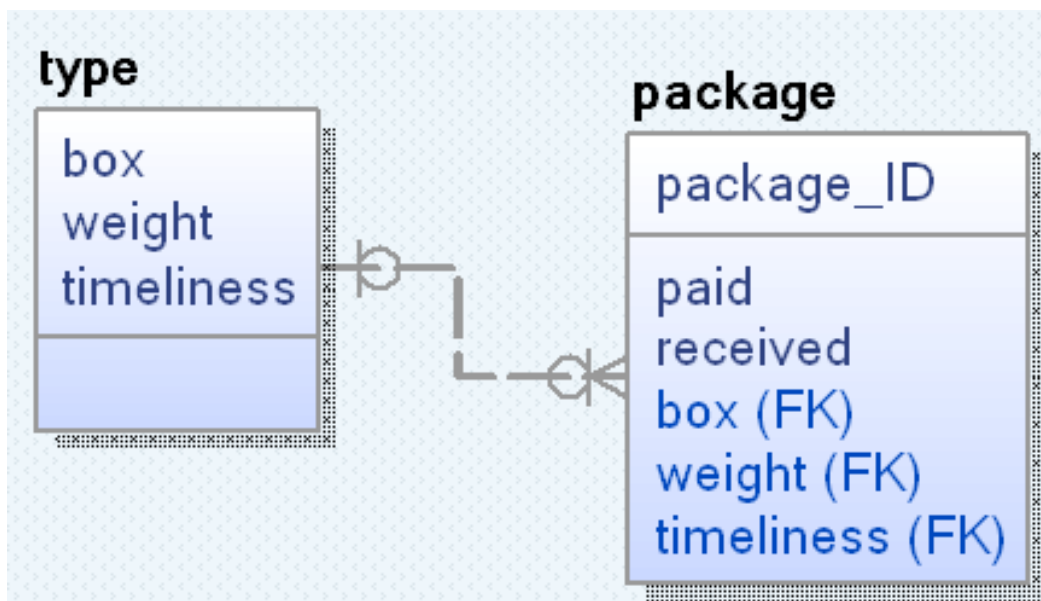
cardinality는 Zero, One or More이 된다.

→sender : foreign key인 package_ID가 sender의 primary key 역할을 하고 있으므로 sender는 weak entity가 되고 둘의 관계는 identifying하다. package는 반드시 sender를 하나만 가지고 있으므로, cardinality는 1..1이 된다.

→receiver : foreign key인 package_ID가 receiver의 primary key 역할을 하고 있으므로 receiver는 weak entity가 되고 둘의 관계는 identifying하다. package는 반드시 receiver를 하나만 가지고 있으므로, cardinality는 1..1이 된다.

→trans_by : foreign key인 package_ID가 trans_by의 primary key 역할을 하고 있으므로 trans_by는 weak entity가 되고 둘의 관계는 identifying하다. package는 배달 전인 상태와 배달이 진행되는 상태로 나뉘며, 배달 전이라면 trans_by를 갖지 않고 진행 중이라면 trans_by를 하나 이상 가지므로 cardinality는 Zero, One or More이 된다.

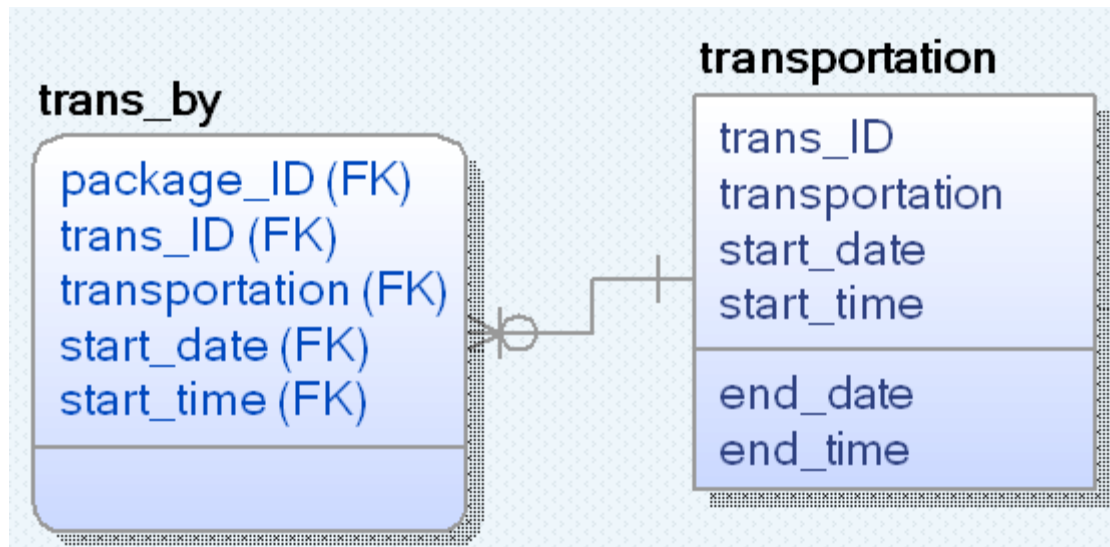
3) type→



type의 primary key인 box, weight, timeliness는 package의 attribute가 되어 foreign key 역할을 한다.

→package : foreign key인 box, weight, timeliness가 package의 primary key가 아니므로 둘의 관계는 non-identifying하다. 모든 package에 대해 택배 회사가 지원하는 type 중, 어떤 package도 포함하지 않는 type도 있을 수 있고, 하나 혹은 그 이상의 package를 포함하는 type도 있을 수 있다. 따라서 cardinality는 Zero, One or More이다.

4) transportation→



transportation의 primary key인 trans_ID, transportation, start_date, start_time은 trans_by의 attribute가 되어 foreign key 역할을 한다.

→trans_by : foreign key인 trans_ID, transportation, start_date, start_time가 trans_by의 primary key 역할을 하고 있으므로 trans_by는 weak entity가 되고 둘의 관계는 identifying하다. transportation은 package를 신지 않고 이동할 수도 있고, 하나 이상을 신고 이동할 수도 있다. 따라서 cardinality는 Zero, One or More이 된다.

