

Project1 Report

20190963 한다현

1. E-R diagram

E-R diagram에서 사용한 entity set은 customer, recipient, package, vehicle, 그리고 warehouse로 총 5개이며, relationship set은 order, receive 그리고 shipment로 총 3개이다.

먼저 customer entity의 attribute는 customer_ID, name, address인데, customer_ID가 customer의 primary key이고 name과 address는 각각 customer의 이름과 주소를 의미한다. recipient의 attribute는 recipient_ID, name, address이고, primary key는 recipient_ID이며, name과 address는 각각 recipient의 이름과 주소를 의미한다. package의 attribute는 package_ID, package_type, weight, timeliness, content, value이다. 이 중 package_ID가 primary key이며, package_type은 말그대로 패키지의 유형(envelope, small box, large box)를 의미하며, weight은 패키지의 무게, timeliness는 배송 기한, content와 value는 각각 패키지의 내용물과 그 가치를 의미한다. content는 후에 항목별 bill을 제공할 때 사용되기 때문에 hazardous materials와 international shipments가 아닌 경우에도 패키지의 내용을 데이터베이스에 저장한다. Vehicle의 attribute는 vehicle_ID와 vehicle_type이며 vehicle_ID가 primary key이고, vehicle_type은 운송수단의 종류, 즉 운송수단이 트럭인지 비행기인지 등을 저장한다. 마지막으로 warehouse의 attribute는 warehouse_ID와 location이며, primary key는 warehouse_ID이고, location은 각 warehouse의 위치를 저장한다.

order은 customer entity set과 package entity set이 one to many의 관계로 이루어진 것이다. 한 명의 customer는 여러 개의 package를 주문할 수 있지만, 각각의 package는 한 명의 customer의 소유이기 때문에 one to many 관계이다. 또한 모든 package는 customer와 관계를 가져야 하므로 total이다. One to many 관계의 경우, relationship set의 primary key는 many side의 primary key가 되므로, order의 primary key는 package_ID이다. 이 외에도 order에는 order_date, price, 그리고 payment_type 속성이 추가되어야 한다. order_date는 고객이 패키지를 주문한 날짜

를, price는 패키지 운송의 가격을, payment_type은 고객이 비용을 지불하는 방식을 나타낸다.

payment_type을 order relationship set에 추가한 이유는 한 명의 사용자가 패키지를 주문할 때마다 다른 지불 방식을 선택할 수도 있기 때문에 order relationship set에 추가하였다. recipient와 package의 관계 역시 one to many의 관계이며, 이 relationship set의 이름은 receive이다. 한 명의 recipient는 여러 개의 패키지를 수신 받을 수 있지만 각각의 패키지는 한 명의 recipient에게 전달되기 때문에 one to many 관계이며 모든 package는 recipient가 필요하기 때문에 total이다. 또한 receive에는 is_in_time이라는 속성이 있는데, 이는 package가 기한 내에 도착했는지 여부를 저장하는 속성이다. 마지막 relationship set은 shipment인데, 이 관계는 package, warehouse, vehicle 3개의 entity sets가 연관된 ternary relationship이다. 여기서 shipment란, 같은 목적지를 갖는 package의 집합이 아니라, 한 package를 기준으로 그 package가 어떤 운송 수단을 통해 어떤 warehouse에서 다음 목적지로 이동하는 지를 나타내는 관계이다. 예를 들어, 하나의 shipment는 pack1이라는 package가 truck1을 타고 ware1에서 ware2로 이동하는 과정인 것이다. 이 세 개의 entity set은 모두 many to many 관계이다. package는 여러 vehicle을 통해 여러 warehouse를 거칠 수 있다. warehouse에는 여러 개의 package가 여러 개의 vehicle을 통해 도착하고 출발한다. vehicle 역시 여러 개의 package를 가지고 여러 개의 warehouse를 방문한다. 따라서 shipment relationship set의 primary key는 세 entity set의 primary key의 조합인 (package_ID, vehicle_ID, warehouse_ID)이다. 또한 shipment에는 출발 시간인 depart_time, 출발 날짜인 depart_date, 도착 시간인 arrive_time, 도착 날짜인 arrive_date, 그리고 다음 목적지인 destination이라는 속성들이 있다. shipment relationship에서 warehouse_ID는 shipment가 출발하는 장소이고, depart_time과 depart_date은 warehouse에서 출발할 때의 날짜와 시간이다. 이 두 속성의 값이 null이면 아직 shipment가 warehouse에서 출발하지 않았다는 것을 의미한다. destination은 다음 목적지를 의미하는데, 이는 다음에 경유할 warehouse가 될 수도 있고, recipient의 주소가 될 수도 있다. arrive_time과 arrive_date는 shipment가 목적지에 도착한 시간과 날짜를 저장하는데, shipment가 목적지에 도착하기 전에는 null 값을 가질 것이다.

2. Relational Schema

customer entity는 customer_ID, name, address라는 속성을 갖고 있다. customer_ID는 char 데이터 타입이며 primary key이다. name과 address 모두 데이터 타입은 char이고, null 값을 허용하지 않는다. 따라서 모든 customer는 그들의 이름과 주소가 필요하다. package entity는 package_ID, package_type, weight, timeliness, content, value를 속성으로 갖고 있다. package_ID는 primary key이고, 데이터 타입은 char이다. package_type, timeliness, content는 모두 데이터 타입이 char이며, null 값을 허용하지 않도록 한다. weight는 데이터 타입이 number이고 null 값을 허용하지 않으며, value의 데이터 타입은 money이고, value는 international shipments일 때 필요한 속성이기 때문에 null 값을 허용한다. order는 customer와 one to many(order가 many)의 관계이다. 한 명의 customer가 여러 개의 주문을 할 수 있기 때문이다. 따라서 customer의 primary key인 customer_ID가 order에 foreign key로 추가된다. 모든 order는 customer와 관계를 가져야 하므로 customer_ID는 null 값을 허용하지 않는다. 또한 package와 order는 one to one 관계인데, 이는 하나의 패키지가 하나의 주문과 관계를 갖기 때문이다. 따라서 package의 PK인 package_ID를 order의 FK이자 PK로 추가한다. package의 PK가 order의 PK이므로 둘은 identifying 관계이고, customer와 order는 non-identifying 관계이다. 이 때 모든 order는 그에 해당하는 customer와 관계를 맺어야 하기 때문에 null을 허용하지 않는다. 하지만 customer가 아직 어떠한 주문도 하지 않은 customer가 존재할 수 있기 때문에 order의 cardinality는 zero, one or more이다. 또한 모든 package와 모든 order는 관계를 맺어야 한다. 이 외에도 order에는 order_date, price, payment_type이 있으며, E-R diagram에서 확인했던 바와 같이 order의 primary key는 package_ID가 된다. order_date의 데이터 타입은 date, price의 데이터 타입은 money, payment_type의 데이터 타입은 char이며, FK를 포함한 모든 속성은 null 값을 허용하지 않는다. order table에서 customer_ID와 order_date, price 속성을 활용하여 한 해에 가장 많은 주문을 한 고객과, 가장 많은 돈을 소비한 고객을 찾을 수 있을 것이다. 또한 customer와 order table을 통해 simple bill을 생성하고, package와 order table의 속성들을 활용해 service type별 bill과 항목별 bill을 생성할 수 있을 것이다.

recipient entity는 recipient_ID, name, address 속성들을 포함하며, recipient_ID가 PK이다. 세 개의 속성의 데이터 타입은 모두 char이며, null 값을 허용하지 않는다. recipient와 package의 관계인 receive는 recipient와 one to many(receive가 many), package와 one to one 관계를 갖는데, 한 명의 recipient가 여러 배송을 받을 수 있고, 하나의 package가 하나의 receive와 관계를 갖기 때문이다. 따라서 receive는 package_ID와 recipient_ID를 FK로 추가하고, 이 중 package_ID가 PK가 되며, is_in_time이라는 속성도 포함하고 있다. package의 PK가 receive의 PK이므로 둘은 identifying 관계이고, recipient와 receive는 non-identifying 관계이다. 이 때 모든 receive는 그에 해당하는 recipient와 관계를 맺어야 하기 때문에 null을 허용하지 않는다. 또한 모든 package와 모든 receive는 관계를 맺어야 한다. is_in_time은 배송 기한을 지켰는지 여부를 저장하기 때문에 Boolean 데이터 타입이고, FK를 포함한 모든 속성은 null 값을 허용하지 않는다. is_in_time 속성을 통해 제시간에 배송이 완료되지 않은 package들을 찾을 수 있을 것이다.

vehicle entity는 vehicle_ID와 vehicle_type을 속성으로 갖고, 두 속성 모두 데이터 타입은 char이며 null 값을 허용하지 않고, PK는 vehicle_ID이다. warehouse entity의 속성은 warehouse_ID(PK)와 location이고, 두 속성 모두 데이터 타입은 char이며 null 값을 허용하지 않는다. shipment는 E-R diagram에서 설명했던 것처럼, 하나의 package가 특정 warehouse에서 특정 vehicle을 통해 이동하는 것을 의미한다. 따라서 shipment table은 package, vehicle, warehouse와 각각 many to one(shipment가 many)의 관계를 갖는다. 하나의 package, vehicle, warehouse가 여러 개의 shipment를 가질 수 있는 반면, 하나의 shipment는 하나의 package, 하나의 vehicle, 하나의 warehouse를 갖기 때문이다. 그러므로 shipment는 세 entity set의 primary key를 foreign key로 추가하며, 이들의 조합이 shipment의 primary key가 되고, package, vehicle, warehouse와 shipment의 관계는 모두 identifying이다. 또한 배송을 시작하지 않은 package, 운송수단 구매 후 첫 배송을 출발하지 않은 vehicle, 건설 후 아직 운영을 시작하지 않은 warehouse의 경우 어떠한 shipment와의 관계를 갖지 않기 때문에 각각의 entity와의 관계에서 shipment의 cardinality는 zero, one or many이다. 이 외에 shipment는 depart_time, arrive_time, depart_date, arrive_date, destination이라는 속성들을 포함하는데, depart_time과 arrive_time의 데이터 타입은 DATETIME HOUR TO SECOND이며, depart_date와 arrive_date의 데이터 타입은 date이고, destination의 데이

터 타입은 char이다. 이 때, depart_time, arrive_time, depart_date, arrive_date은 모두 null 값을 허용한다. arrive_time과 arrive_date가 null인 경우는 아직 shipment가 목적지에 도착하지 않은 것을 의미하며, depart_time과 depart_date도 null인 경우는 아직 shipment가 warehouse에서 출발하지 않은 것을 의미한다. 하지만 destination은 null 값을 허용하지 않는다. shipment table과 다른 table들을 활용하여 한 package가 거쳐왔던 장소들을 도출할 수도 있고, 현재 특정 vehicle에 있는 package들의 customer와 recipient를 찾을 수도 있고, vehicle이 마지막으로 배송한 package들을 찾을 수도 있을 것이다.