

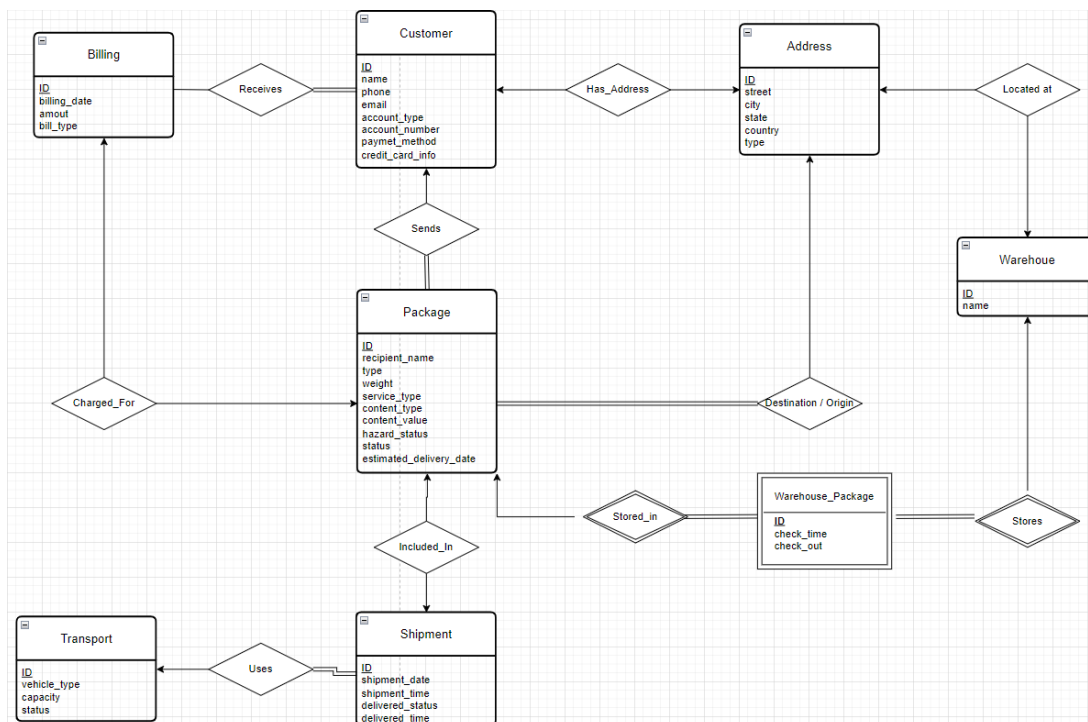
Package Delivery System Design

컴퓨터공학과 20191619 이동석

1. 개요

본 레포트에서는 택배 회사의 요구사항을 충족하는 데이터 베이스 설계에 대해 소개한다. 주요한 엔티티로는 고객(Customer), 주소(Address), 패키지(Package), 창고(Warehouse), Warehouse_Package, 운송(Transport), 발송(Shipment), 청구(Billing)이다. 이러한 엔티티들간의 관계와 속성을 설명하며, 작성된 E-R 다이어그램과 릴레이션 스키마를 소개한다. 또한, 소개에 앞서 다음 PK는 Primary Key, FK는 Foreign Key를 의미한다.

2. E-R Diagram



3. 엔티티 및 속성

3.1 Customer

고객 엔티티는 택배 회사의 서비스를 이용하는 모든 고객을 나타낸다. 고유 식별자인 ID를 PK로 설정했다. 고객의 이름, 전화번호, 이메일 등 기본정보와 계약 고객과 비정기 고객을 구분하는 `account_type`, 그리고 계좌번호, 결제 방법을 지정할 수 있는 `payment_method`, 마지막으로 신용카드의 정보를 포함한다.

3.2 Address

주소 엔티티는 고객 및 수신자, 창고의 주소 정보를 나타낸다. 마찬가지로 고유 식별자인 ID를 PK로 설정했다. 주소의 기본 구성요소를 포함한다. 또한, 발송지, 수신지, 창고 주소 등을 구분하기 위해 `type` 속성을 추가했다. 주소 엔티티를 독립적으로 만들어 고객, 수신자, 창고와 관련된 주소정보를 저장하고 관리한다.

3.3 Package

패키지 엔티티는 택배 회사에서 배송하는 모든 소포를 나타낸다. 마찬가지로 고유 식별자인 ID를 PK로 설정했다. 이 외에도 배송에 필요한 수신자 이름, 패키지의 유형, 무게, 서비스 유형, 내용물 유형, 내용물 가치, 위험 상태, 배송 상태등의 정보를 포함한다. 이때 패키지는 고객, 발송지 주소, 수신지 주소의 관계를 나타내기 위해 Address와 Customer와 릴레이션을 가진다.

3.4 Warehouse

창고 엔티티는 택배 회사가 관리하는 모든 창고를 나타낸다. 고유 식별자인 ID를 PK로 설정했다. 창고의 이름을 속성으로 가지며 주소 정보를 포함하기 위해서 Address와 릴레이션을 가진다.

3.5 Warehouse_Package

이 엔티티는 창고와 패키지 사이의 관계를 나타낸다. 배송 중간에 패키지가 창고로

갈 수 있다. 이때, 이 패키지의 입출고를 기록하기 위해 사용된다. 따라서 E-R 다이어그램 상에서 PK인 ID는 복합키로서 (Package_id, Warehouse_id) 가 사용된다. 이를 통해 패키지의 추적이 가능하다. 또한, weak entity 이다.

3.6 Shipment

발송 엔티티는 택배 회사에서 발송되는 모든 소포의 정보를 나타낸다. 고유 식별자인 ID를 PK로 설정하였다. 발송 날짜, 시간과 함께 패키지와 운송수단의 관계를 나타내기 위해 Package와 Transport와 릴레이션을 가진다. 이를 통해 각 발송 정보와 관련된 패키지와 운송수단을 조회할 수 있다.

3.7 Billing

청구 엔티티는 택배 회사의 고객에게 청구되는 모든 요금을 나타낸다. 고유 식별자인 ID를 PK로 설정했다. 청구 날짜, 금액, 유형의 정보가 포함된다. 고객과 패키지와의 릴레이션을 가진다. 이를 통해 각 청구 정보와 관련된 고객과 패키지를 조회할 수 있다.

3.8 Transport

운송 엔티티는 택배 회사의 모든 운송수단을 나타낸다. 고유 식별자인 ID를 PK로 설정하였다. 운송 수단의 유형, 용량, 현재 상태의 정보가 포함된다.

4. 릴레이션

4.1 Customer – Address : one – to – one

한명의 고객의 발송지 주소는 하나이므로 one-to-one으로 하였다. 또한, 이때 고객은 적어도 하나의 주소를 가져야 하므로 Total이며, 주소는 창고주소도 포함하므로 Partial 이다.

4.2 Package – Customer : many – to – one

여러 패키지는 동일한 고객에게 속할 수 있다. 또한 각 패키지는 단 한명의 고객에게 속한다. 더불어 고객은 partial, 패키지는 total이다. 등록된 모든 고객이 패키지를 보내는 것은 아니지만, 모든 패키지는 반드시 어떤 고객에 의해 발송된다.

4.3 Package – Address : many – to – one

패키지는 동일한 발송지와 수신지 주소를 가질 수 있으며, 각 패키지는 단 한 개의 발송지와 수신지 주소를 가진다. 또한, 패키지는 반드시 하나의 수신지 주소를 가지므로 Total이 되며 주소는 Partial이 된다.

4.4 Warehouse – Address : one – to – one

창고는 단 하나의 주소를 가지며, 주소 역시 단 하나의 창고에만 속한다.

4.5 Warehouse_Package – Warehouse : many – to – one

여러 창고-패키지 관계는 동일한 창고에 속할 수 있으며, 각 창고-패키지 관계는 단 하나의 창고에 속한다. 패키지는 창고를 거치지 않을 수 있지만, 창고-패키지는 반드시 패키지에 속해야하므로 Total이다.

4.6 Warehouse_Package – Package : many – to – one

여러 창고-패키지 관계는 동일한 창고에 속할 수 있으며, 각 창고-패키지 관계는 단 하나의 패키지에 속한다. 또한, 창고는 패키지를 처리하지 않을 수 있지만, 창고-패키지 엔티티는 반드시 창고에 속해야하므로 Total 이다.

4.7 Shipment –Package : one – to – one

각 발송 정보는 단 하나의 패키지에 속하며, 각 패키지 역시 단 하나의 발송 정보를 가진다. 또한, 패키지는 반드시 하나의 배송정보를 가져야하므로 Total이며, 배송 정보가 반드시 패키지와 연결될 필요는 없으므로(아직 배송하지 않은 패키지 존재) partial이다.

4.8 Shipment – Transport : many – to – one

여러 발송 정보는 동일한 운송 수단에 속할 수 있으며, 각 발송 정보는 단 하나의 운송 수단에 속한다. 또한, 배송은 반드시 하나의 운송 수단을 가지므로 Total이며, 현재 사용하지 않는 운송수단이 존재할 수 있으므로 Partial이다.

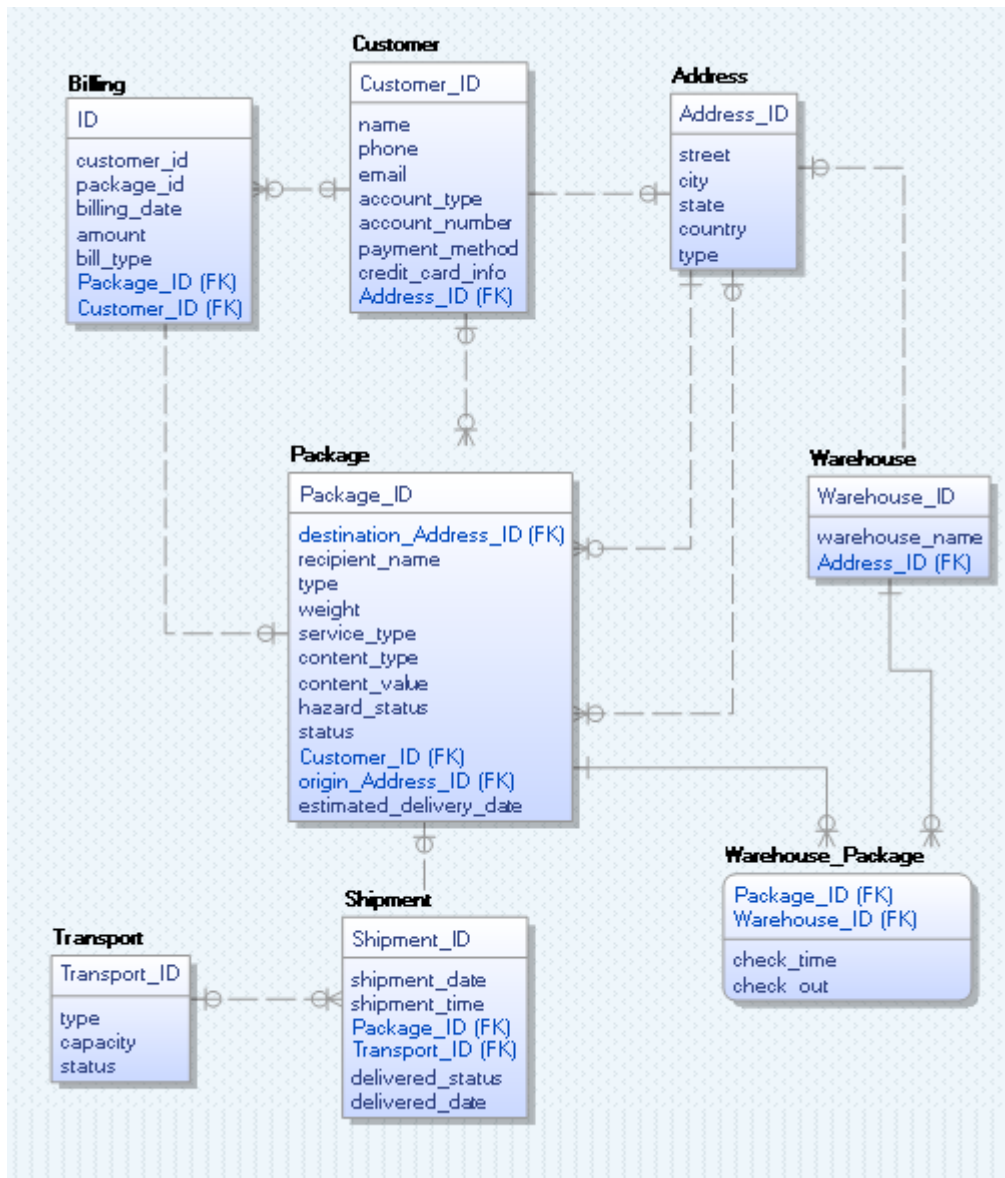
4.9 Billing – Customer : many – to – one

여러 청구 정보는 동일한 고객에게 속할 수 있으며, 각 청구 정보는 단 하나의 고객에게 속한다. 또한, 고객은 적어도 하나 이상의 청구 정보를 가질 수 있으므로 Total이며, 반대로 아직 청구 되지 않은 패키지가 할 수 있으므로 청구정보는 Partial이다.

4.10 Billing – Package : one – to – one

마찬가지로 각 청구 정보는 단 하나의 패키지에 속하며, 각 패키지는 단 하나의 청구 정보를 가진다. 또한, 패키지는 반드시 하나의 청구 정보를 가져야하므로 Total이다.

5. 릴레이션 스키마



위에서 설명한 것을 바탕으로 릴레이션 스키마를 위 그림과 같이 작성할 수 있다.

6. 쿼리

위 릴레이션을 사용해 프로젝트에서 요구하는 쿼리를 다음과 같이 처리할 수 있다.

1. 사고 당시 트럭 1721에 소포가 있던 모든 고객 및 수령인

Shipment와 Transport를 조인하여 트럭 1721에 운송 중인 모든 패키지를 찾을 수 있다. 다음으로, 찾은 Package를 기준으로 Package와 Customer 그리고 Package와 Address 를 조인하여 고객 및 수령인 정보를 얻을 수 있다.

2. 사고 이전에 마지막으로 배달한 소포

마찬가지로 Shipment와 Transport를 조인하여 트럭 1721에 운송 중인 모든 패키지를 찾고 배송 상태가 Delivered 된 패키지 중 가장 최신 날짜를 찾는다.

3. 지난 1년동안 가장 많은 소포를 발송한 고객

1년 동안 기록에서 Package와 Customer를 조인하여 각 고객별로 발송한 패키지 수를 계산한다.

4. 지난 1년동안 가장 많은 돈을 지불한 고객

Billing과 Customer를 조인하여 찾는다.

5. 약속된 시간 내에 배송되지 않은 소포

Package에서 예상 배송일이 현재 날짜 보다 이전이며 Shipment에서 Delivered가 아닌 패키지

이러한 방법을 사용하여 주어진 릴레이션 스키마를 기반으로 각 쿼리를 처리할 수 있다.