

데이터베이스시스템

(CSE4110-02)

Project 1. E-R design and Relational Schema design

서강대학교 컴퓨터공학과
20181614
김주연

목차

1. 프로그램 개요
 - 1.1 프로젝트 문제
 - 1.2 프로젝트 목표
2. 본 프로젝트에서 Entity, Relationship 정의와 그 이유
 - 2.1 Entity
 - 2.2 Relationship
3. ERD에서 각각의 Relationship의 cardinality와 그 이유
4. Query 처리 예시
5. E-R diagram
6. Relational Schema diagram

1. 프로그램 개요

1.1 프로젝트 문제

The application is a package delivery company (e.g. FedEx, UPS, DHL, CJ Logistics). The company needs to keep track of packages shipped and their customers. To find out more about this application, think about any experiences you may have had shipping packages and receiving packages, and browse shippers' web sites.

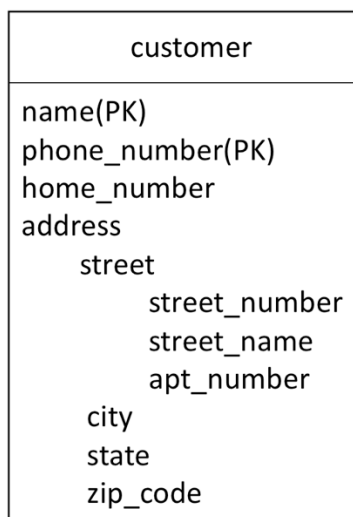
1.2 프로젝트 목표

The goal of this project is to provide a realistic experience in the conceptual design, logical design and maintenance of a small relational database.

2. 본 프로젝트에서 Entity, Relationship 정의와 그 이유

2.1 Entity

- customer



customer entity는 Delivery Service를 필요로 하는 사람, 즉 택배를 보내는 사람의 데이터를 저장하기 위한 entity이다. customer의 attribute로는 택배를 접수할 때 필요한 발송자의 정보들을 설정하였다. customer entity의 primary key는 customer의 이름과 휴대폰 번호이다. 이름은 겹치기 쉽고, 번호는 쉽게 바꿀 수 있기 때문에 두 정보를 같이 PK로 사용하도록 하였다. 이외의 attribute로는 customer의 집 전화번호인 home_number와 주소인 address가 있다. 주소는 composite attribute 형식으로 작성하였다.

- receipt

receipt
receipt_number(PK) date month year amount payment_type name(FK) phone_number(FK) package_id(FK)

receipt entity는 customer가 Delivery Service를 접수할 때 생성되는 데이터들을 담기 위한 entity이다. 택배를 접수(예약)하게 되면 접수번호, 접수일, 금액, 지불 정보 등이 생성되는데 해당 항목들이 receipt entity의 attribute가 된다. receipt entity의 primary key는 receipt_number이다. 택배 하나에 대한 접수 번호는 하나만 존재하므로 receipt_number를 PK로 사용하도록 하였다. date, month, year는 각각 해당하는 접수일의 날짜 정보를 저장하는 attribute이다. amount는 해당 택배의 택배비 정보를 담고 있다. payment_type은 지불 방법을 문자열로 저장하는 attribute이다. 택배사와 계약을 맺어 월 단위로 한번에 후불 결제하는 경우는 'Monthly', 신용카드를 사용하여 개별적으로 후불 결제하는 경우는 'Credit', 현금이나 체크카드로 개별적인 선불 결제를 하는 경우는 'Prepaid', 착불 방식은 'Collectible'의 형태로 저장한다. Foreign key로는 customer entity의 name과 phone_number, 그리고 package entity의 package_id(송장번호)가 있다. 접수자인 customer entity의 PK를 attribute로 가짐으로써 해당 택배의 발송자(접수자)와 택배 접수 간 관계가 있음을 나타내고자 하였고, 택배 자체의 정보를 저장하는 package entity의 PK를 attribute로 가짐으로써 택배 접수와 그에 해당하는 택배 정보 간 관계가 있음을 나타내고자 하였다.

또한 package entity에서 recipient 정보를 가지고 있기 때문에, customer와 package entity의 PK를 모두 가지고 있는 receipt entity는 customer와 recipient, package entity 정보에 모두 접근이 가능하다. 따라서 후에 택배비 지불에 대한 Bill을 작성한다고 하면 receipt entity가 가지고 있는 접수일, 지불 금액 정보와 더불어 payment_type에 따라 발송자와 수취자 중 Bill을 받게 될 사람의 정보를 찾을 수 있다. 또한 package type이나 individual shipment charge에 따른 bill도 쉽게 작성할 수 있다.

- package

package
package_id(PK) type weight timeliness special_case name(FK) phone_number(FK)

package entity는 택배 자체의 데이터를 저장하기 위한 entity이다. 택배의 데이터로는 type, weight, timeliness, special_case 등이 있으며 이를 package entity의 attribute로 하였다. package entity의 primary key는 package_id로 하였으며 이는 송장번호를 말한다. 택배 하나마다 송장번호는 하나가 지정되므로 package_id를 entity의 PK로 하였다. type은 package의 type을 문자열로 저장하고 있으며 'Flat envelope', 'Small box', 'Large box' 등의 종류가 있다. weight는 package의 무게를 숫자로 저장한다. timeliness는 택배의 배송 타입을 문자열로 저장하고 있으며, 'Overnight', 'Second day' 등이 있다. special_case는 택배의 특이사항을 문자열로 저장하고 있으며 'Hazardous', 'International' 등의 정보를 저장한다. Foreign key로는 recipient entity의 name과 phone_number가 있다. 수취자인 recipient entity의 PK를 attribute로 가짐으로써 해당 택배와 택배 수취자 간 관계가 있음을 나타내 고자 하였다.

- shipment

shipment
package_id(PK, FK) time(PK) date(PK) month(PK) year(PK) place_type place_number available

shipment entity는 택배의 배송 과정 데이터를 저장하기 위한 entity이다. 배송 과정 데이터로는 time, date, month, year, place 등이 있으며 이를 shipment entity의 attribute로 하였다. package entity는 weak entity로 package entity의 primary key인 package_id를 primary key로 하였으며, time, date, month, year 정보가 discriminator로서 package_id와 함께 shipment entity의 primary key 역할을 수행

한다. time, date, month, year는 각각 해당하는 시점을 정수 또는 문자열 형태로 저장하며, place_type은 그 시점에서의 택배의 위치를 문자열로 저장한다. place_type에 해당하는 항목에는 'Truck', 'Airplane', 'Warehouse', 'Recipient' 등이 있다. place type이 'Recipient'인 경우 배송 완료임을 나타낸다. place_number는 현재 택배가 위치한 place에 대한 세부정보를 담고 있다. 이는 항공편명, 트럭 번호, 창고 번호 등을 말하며 숫자(또는 문자열)로 저장하게 된다. available은 해당 place의 상태를 문자열로 나타낸다. place가 정상적으로 운행(또는 가동) 중이어서 택배가 정상적인 배송 과정에 있다면 'Yes', place가 수리 중이거나 고장 등으로 택배가 정상적인 배송 과정에 있지 않다면 'No' 등의 정보를 저장하게 된다.

- recipient

recipient
name(PK) phone_number(PK) home_number address street street_number street_name apt_number city state zip_code

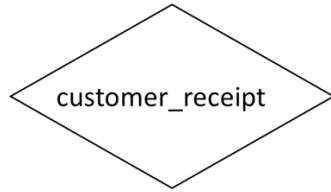
recipient entity는 Delivery Service를 받는 사람, 즉 택배를 받는 사람의 데이터를 저장하기 위한 entity이다. recipient의 attribute로는 택배 수취자의 정보들을 설정하였다. recipient entity의 primary key는 recipient의 이름과 휴대폰 번호이다. 이름은 겹치기 쉽고, 번호는 쉽게 바꿀 수 있기 때문에 두 정보를 같이 PK로 사용하도록 하였다. 이외의 attribute로는 recipient의 집 전화번호인 home_number와 주소인 address가 있다. 주소는 composite attribute 형식으로 작성하였다.

해당 Entity의 데이터 형태는 customer entity의 데이터 형태와 동일하다. 따라서 두 entity를 그냥 하나의 entity로 저장하고 관계를 다르게 지정하여 관계에 따라 사용을 다르게(발송자/수취자) 할 수도 있었지만, 발송자와 수취자라는 개념의 차이와, 두 entity와 직접적으로 관련 있는 항목(발송자-택배 접수, 수취자-택배 정보)을 보다 explicit하게 보여주기 위해 2개의 entity로 따로 나타내었다.

2.2 Relationship

(Cardinality 설명은 3에서 수행)

- customer_receipt



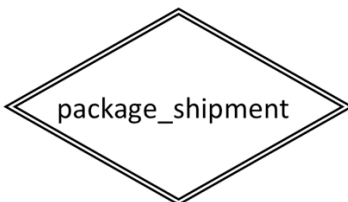
customer_receipt relationship은 택배 발송자와 택배 접수 간 관계를 나타내는 relationship이다. 발송자가 택배를 접수하는 action 관계를 나타낸다.

- receipt_package



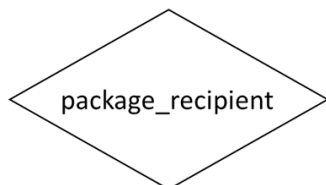
receipt_package relationship은 택배 접수와 택배 정보 간 관계를 나타내는 relationship이다. 택배를 접수하면 해당 택배에 대한 송장 번호가 생성되고 택배 정보가 package entity에 저장되는 관계를 capture하고 있다.

- package_shipment



package_shipment relationship은 택배 자체의 정보와 해당 택배의 배송 정보 간 관계를 나타내는 relationship이다. 택배가 배송되는 과정을 identifying relation으로 저장하고 있다.

- package_recipient



package_recipient relationship은 택배 정보와 택배 수취자 간 관계를 나타내는 relationship이다.

3. ERD에서 각각의 Relationship의 cardinality와 그 이유

- customer_receipt

The screenshot shows the 'Relationships' dialog box for the relationship 'R/2 (customer to receipt)'. The 'Name' field is 'R/2'. The 'General' tab is selected. Under 'Verb Phrase', 'Parent-to-Child' and 'Child-to-Parent' are empty. Under 'Relationship Cardinality', the 'Summary' is 'One-to-Zero-One-or-More'. The 'Cardinality' section has 'Zero, One or More' selected. The 'Relationship Type' section has 'Non-Identifying' selected. The 'Nulls' section has 'No Nulls' selected. There is a 'Logical Only' checkbox and a 'Reset Cardinality' button. At the bottom are 'OK' and 'Cancel' buttons.

customer_receipt relationship은 택배 발송자와 택배 접수 간 관계를 나타내는 relationship이다. 한 명의 발송자(접수자)는 하나 이상의 택배를 접수할 수 있고 (발송자의 정보는 등록했으나 아직 보낼 택배의 정보를 기입하지 않은 경우 접수 가짓수가 0이 될 수도 있다), 하나의 택배는 반드시 한 명의 접수자를 가지므로 customer와 receipt는 one to many의 cardinality를 가진다. 또한 접수자 없이 택배의 접수는 불가능하므로 receipt의 참여는 total이다.

- receipt_package

The screenshot shows the 'Relationships' dialog box for the relationship 'R/3 (package to receipt)'. The 'Name' field is 'R/3'. The 'General' tab is selected. Under 'Verb Phrase', 'Parent-to-Child' and 'Child-to-Parent' are empty. Under 'Relationship Cardinality', the 'Summary' is 'Zero-or-One-to-Zero-or-One (Z)'. The 'Cardinality' section has 'Zero or One (Z)' selected. The 'Relationship Type' section has 'Non-Identifying' selected. The 'Nulls' section has 'Nulls Allowed' selected. There is a 'Logical Only' checkbox and a 'Reset Cardinality' button. At the bottom are 'OK' and 'Cancel' buttons.

receipt_package relationship은 택배 접수와 택배 정보 간 관계를 나타내는 relationship이다. 하나의 택배를 접수하면 그에 해당하는 하나의 택배 정보가 생성되고, 접수하지 않은 택배 정보는 존재할 수 없으며, 택배 없이 접수할 수 없기 때문에 receipt와 package는 one to one의 cardinality를 가진다.

- package_shipment

The screenshot shows a 'Relationships' dialog box with the following details:

- Relationship:** R/5 (package to shipment)
- Name:** R/5
- General Tab:**
 - Verb Phrase:** Parent-to-Child and Child-to-Parent fields are empty.
 - Relationship Cardinality:**
 - Summary:** One-to-Zero-One-or-More
 - Cardinality:** ☒ Zero, One or More; ☐ One or More (P); ☐ Zero or One (Z); ☐ Exactly: []
 - Relationship Type:** ☒ Identifying; ☐ Non-Identifying
 - Nulls:** ☐ Nulls Allowed; ☐ No Nulls
 - ☐ Logical Only
 - Buttons:** OK, Cancel, Reset Cardinality

package_shipment relationship은 택배 자체의 정보와 해당 택배의 배송 정보 간 관계를 나타내는 relationship이다. 하나의 택배에 대해 다수의 배송 정보가 생성되게 되고(아직 배송 출발하지 않은 경우 배송 과정이 기록되지 않아 배송 정보의 개수가 0일 수 있음), 하나의 배송 정보는 그에 해당하는 하나의 택배를 나타내므로 package와 shipment는 one to many의 cardinality를 가진다. 또한 택배 없이 배송 이력이 존재할 수 없으므로 shipment의 참여는 total이다.

- package_recipient

package_recipient relationship은 택배 정보와 택배 수취자 간 관계를 나타내는 relationship이다. 하나의 택배는 그것이 배송될 한 명의 수취자 정보를 가지게 되고, 한 명의 수취자는 여러 택배를 받을 수 있기 때문에 package와 recipient는 many to one의 cardinality를 가진다. 또한 수취자 없이 택배의 배송은 불가능하므로 package의 참여는 total이다.

4. Query 처리 예시

- Assume truck 1721 is destroyed in a crash. Find all customers who had a package on the truck at the time of the crash. Find all recipients who had a package on that truck at the time of the crash. Find the last successful delivery by that truck prior to the crash.

: shipment entity는 배송 현황에 관한 정보를 저장하고 있다. shipment entity의 tuple 중 place_type='Truck', place_number='1721'인 항목에 대해 주어진 사고 시점 이후의 time, date(, month, year) 정보를 가진 package_id만 select하면 해당 트럭에 들어 있었던 package_id들을 찾을 수 있다.

package entity는 package_id와 recipient의 정보를 저장하고 있으므로, 위에서 구한 package_id들에 해당하는 recipient 정보를 select하면 트럭에 들어 있던 package들의 수취자들을 찾을 수 있다.

receipt entity는 package_id와 customer의 정보를 저장하고 있으므로, 위에서 구한 package_id들에 해당하는 customer 정보를 select하면 트럭에 들어 있던 package들의 발송자들을 찾을 수 있다.

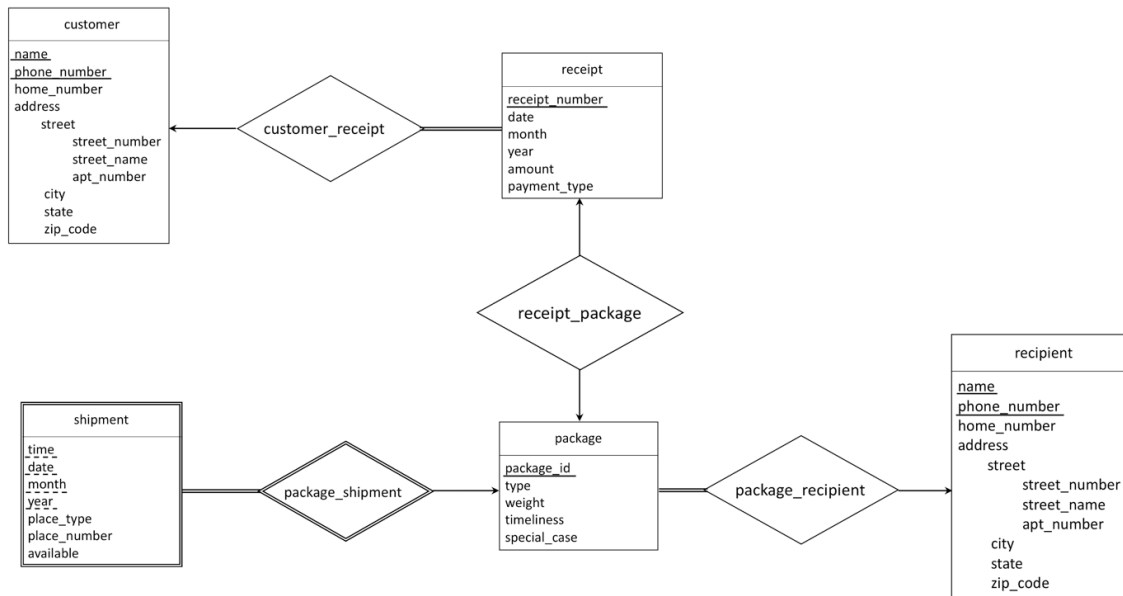
또한 shipment의 place_type='Truck', place_number='1721'인 항목에 대해 주어진 사고 시점에서 가장 가까운(사고 시점보다 이전 시점중 가장 큰) 시점에서 place_type='Recipient'인 항목의 package_id를 찾으면 해당 택배의 정보를 얻을 수 있다.

- Find the customer who has shipped the most packages in the past year.
: receipt entity는 접수 정보와 접수자(발송자) 정보를 저장하고 있다. 따라서 receipt의 tuple을 customer 정보에 의해 grouping하여 주문 번호의 수가 max인 customer를 select 하면 가장 많은 package를 보낸 customer를 찾을 수 있다.
- Find the customer who has spent the most money on shipping in the past year.
: receipt entity는 접수 정보와 접수자(발송자) 정보를 저장하고 있다. 따라서 receipt의 tuple을 customer 정보에 의해 grouping하고, payment_type<>'Collectible'인 항목에 대해 각 customer마다 amount의 합을 구하여 max를 select하면 가장 많은 돈을 소비한 customer를 찾을 수 있다.
- Find those packages that were not delivered within the promised time.
: shipment에서 place_type='Recipient'인 tuple을 찾으면 배송 완료 시점을 찾을 수 있다. 해당 shipment의 package_id를 가지는 receipt entity는 해당 택배가 접수된 날짜를 가지고 있으므로 이를 배송 시작 시점으로 생각할 수 있다. 배송 시작 시점과 완료 시점의 차이가 해당 package_id를 가지는 택배의 timeliness를 초과하는지를 비교하면 배송 약속일 이후에 도착한 택배를 찾을 수 있다.
- Generate the bill for each customer for the past month. Consider creating several types of bills.
 - A simple bill: customer, address, and amount owed.
: receipt entity에서 month가 저번 달에 해당하는 항목을 찾고, 그 항목의 payment_type이 'Collectible'이라면 recipient가, 이외의 경우에는 customer가 지불한 것이므로 지불한 사람의 정보를 찾는다. customer 정보는 customer entity의 PK가 receipt entity에 attribute로 저장되어 있으므로 찾을 수 있고, recipient 정보는 package entity의 PK인 package_id가 receipt entity에 attribute로 저장되어 있으므로 package_id에 해당하는 recipient 정보를 찾으면 수취자를 찾을 수 있다. 발송자/수취자의 주소는 각각의 발송자/수취자 entity에 존재하며, 지불한 금액은 receipt의 amount에 기록되어 있다.
 - A bill listing charges by type of service.
: receipt entity에서 month가 저번 달에 해당하는 항목을 찾고, 그 항목의

package_id를 통해 package entity를 검색하면 해당 package의 type을 찾을 수 있다. 배송비는 receipt의 amount에 기록되어 있다.

- An itemize billing listing each individual shipment and the charges for it.
- : receipt entity에서 month가 저번 달에 해당하는 항목을 찾고, 그 항목의 package_id를 통해 package entity를 검색하면 해당 package의 정보를 찾을 수 있다. 배송비는 receipt의 amount에 기록되어 있다.

5. E-R diagram



6. Relational Schema diagram

