Database system project1 report

Package Delivery System

컴퓨터공학과 20181618 김하늘

목차

1. 문제 상황 및 목표

운송 업체의 택배 배송, 추적 및 결제 청구에 대한 데이터베이스를 구축하는 것이 목표이다. 고객이 택배를 업체에 맡기는 순간부터 수령인이 수령하는 순간까지의 과정과 그에 관한 결제 정보를 관리해야 한다.

배송 과정 및 택배 위치의 domain은 일반적인 택배 배송 과정인

집화처리 : 판매자로부터 택배사가 물건 인수 후 중앙 HUB에 집화처리한다.

간선 상하차 : 배송지역의 HUB로 이동 후 지역별 트럭에 다시 싣는다.

배송 출고 : 목적지에 배송을 완료한다.

와 같은 형식을 참고하여, warehouse 위주의 위치 정보로 설계하였다.

2. 분석

우선 명세서에 있는 내용을 entity와 relationship으로 나누어 보았다. 밑줄이 entity고 기울어진 글씨가 relationship 개념이다.

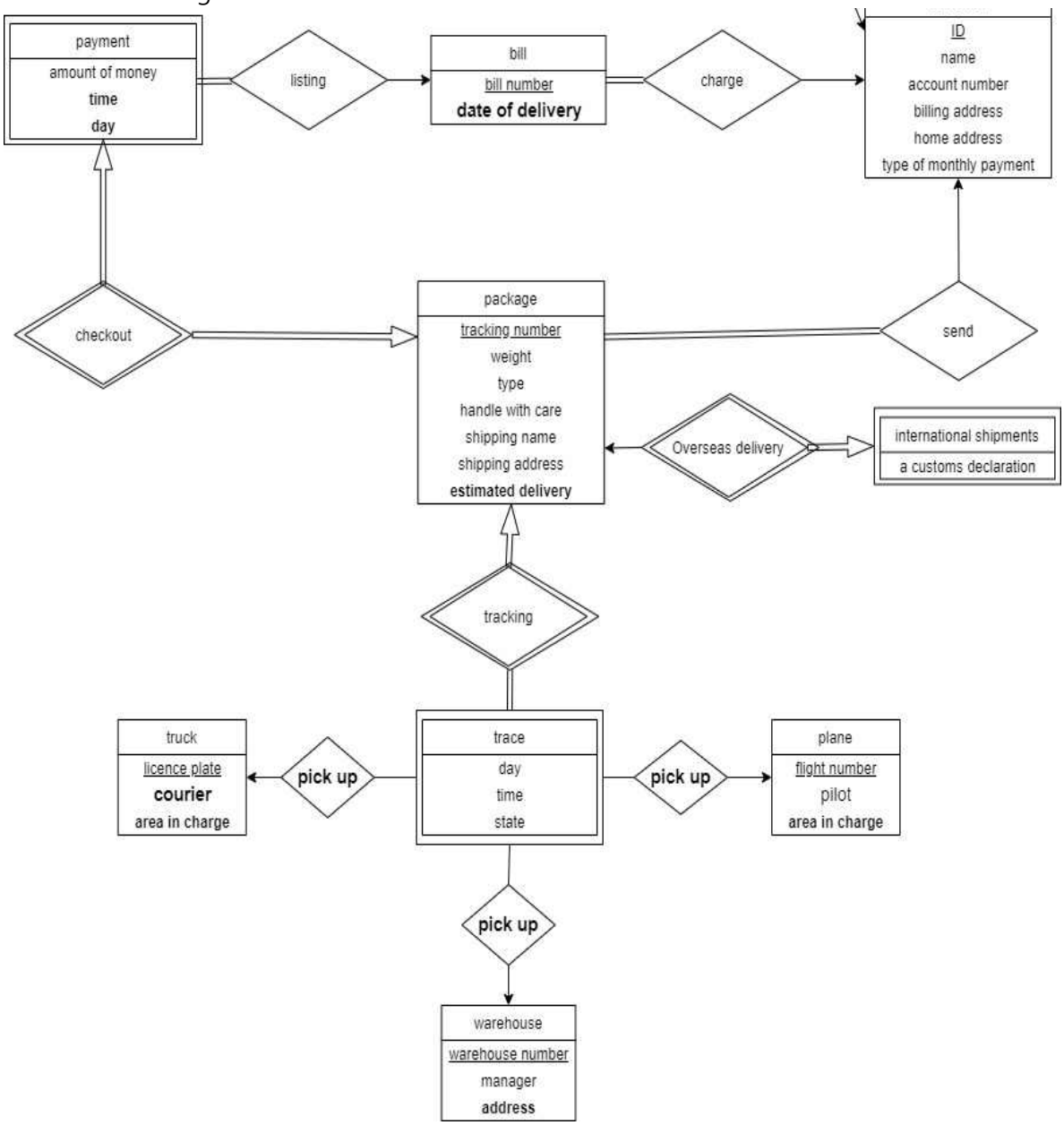
- 고객은 택배를 운송업체에 *맡긴다.*
- 대부분의 경우 신용카드를 이용해 배송비를 *결제*하며, 반송의 경우 선불로 결제한다.
- 정기구독 고객의 경우 일정 금액이 한 달에 한 번씩 결제된다.
- *청구*되는 영수증에는 결제에 대한 다양한 정보들이 나타날 수 있다.
- 배송 조회를 통해 *위치 추적*을 할 수 있으며, 현재까지의 경로와 예상 경로를 조회한다.
- 트럭, 비행기 또는 물류센터에 *적재*된 택배의 정보들이 기록된다.
- *해외로 배송*되는 국제 선적의 경우 세관 신고서가 필요하다.

가장 기본이 되는 package와 customer는 독립적인 정보로도 중요하므로 entity로 설정하였고, 결제에 대한 정보가 기록되는 payment, monthly payment와 해당 payment들을 고객에게 청구하는 bill에 대한 entity도 설정하였다. 배송추적에 필요한 위치 정보는 package에 속성으로 넣을 경우 package의 다른 변하지 않는 속성들(weight, type...)의 불필요한 중복이 생기므로 package에 종속되는 trace라는 weak entity도 설정하였다. 그리고 package의 정확한 위치 정보가 될 수 있는 truck, plane과 같은 운송기기와 warehouse의 정보가 필요하므로 각각 entity로 나누어주었다. 또한 package가 해외배송일 경우 세관 신고서에 대한 weak entity를 설정하였다.

customer는 package를 발송하며, 이에 대한 package의 payment가 생긴다. 이 payment 또는 monthly payment는 bill에 나열되며 이 bill은 customer에게 청구된다. trace는 한 package의 경로가 추적되어있으며 해당 위치 정보는 truck, plane, warehouse에게서 알 수 있다.

3. 구현

3.1. e-r diagram



3.1.1. customer

고객은 동명이인의 경우가 있으므로 'ID'를 부여하여 구분한다. 일반 고객일 경우 카드 번호에 해당하는 'billing address'의 정보가 필요하고, 정기 구독 고객일 경우 계좌 번호인 'account number'가 필요하다. 이러한 것은 'type of monthly payment'로 구분하며 반송될 때 필요한 'home address'에 대한 속성이 있다.

customer와 bill의 relation은 one to many의 관계이며, bill은 total, customer는 partial이다. 한 고객이 여러번 결제하여 여러개의 청구서를 받을 수 있으며, 모든 청구서는 고객에게 청구되지만 결제를 하지 않은 고객이 있을 수 있기 때문이다.

customer와 monthly_bill의 relation은 one to many의 관계이며, monthly_bill은 total, customer는 partial이다. bill과 마찬가지로 정기 결제 청구서는 한 사람에게 여러 번 올 수 있고, 모든 청구서는 고객에게 전달되지만 정기 구독을 하지 않은 고객은 해당 entity와 relation을 갖지 않는다.

customer와 package의 relation은 one to many의 관계이며, package는 total, customer는 partial이다. 한 고객이 여러개의 택배를 보낼 수 있으며, 모든 택배는 고객이 보낸 것이지만 택배를 한 번도 보내지 않은 고객이 있을 수 있기 때문이다.

3.1.2. package

택배는 고유한 운송장 번호 'tracking number'가 부여된다. 택배의 'type'과 'weight'에 대한 정보가 있으며, 취급 주의 여부를 알 수 있는 'handle with care' 그리고 배송지, 수령인의 속성이 있다. 요청받은 배송 예정 날짜를 뜻하는 'estimated delivery'의 속성 또한 존재한다.

package와 payment의 relation은 one to one의 관계이며, payment는 total, package도 total이다. 일반적인 결제에 속하는 payment는 하나의 택배에 이어진 결제내역이고, 모든 택배는 모두 결제 내역으로 날짜 및 시간이 처리 되기 때문이다. 정기 구독 고객의 경우 해당 payment가 생기는 이유는 실제로 청구되지는 않지만 배송을 맡긴 시점의 정보를 알 수 있고, 정기 구독으로 인해 얼마나 더 이익을 볼 수 있었는지에 대한 정보를 구할 수도 있기 때문이다.

package와 trace의 relation은 one to many의 관계이며, trace와 package 모두 total이다. 하나의 택배에 대해 날짜 및 시간대별 경로가 여러 번 측정되며, 모든 택배에 대한 경로 추적이 필요하기 때문이다.

package와 international shipments의 relation은 one to one의 관계이며, international shipments는 total, package는 partial이다. 도착지에 대한 속성인 estimated delivery가 해외로 입력되어 있으면 이 relation이 생겨 해당 택배에 대한 세관 신고서의 정보가 저장된다.

3.1.3. international shipments

해외배송의 경우 세관 신고서가 필요하다. 고객이 작성한 세관 신고서에 대한 정보만을 다루며, 이 역시 package의 속성으로 넣을 경우 해외배송 전문 업체가 아닌 이상 불필요한 Null이 많이 사용 될 것이라 생각하여 package에 dependent한 weak entity로 설정해주었다.

3.1.4. trace

택배의 경로 및 상태에 대한 정보가 저장되어있다. 택배라는 entity에 경로에 대한 속성이 들어간다면 변화하지 않는 다른 속성들의 중복이 매우 많아지므로, package에 dependent한 weak entity로 만들어 주었다. 날짜, 시간, 배송 상태라는 속성이 존재한다.

trace는 간선 상하차에 관해 각각 3가지의 entity와 relation을 가진다. truck, plane, warehouse 모두 many to one의 cardinality를 가지며 모두 partial이다. 특정 택배의 위치는 운송차량 안에 적재되어있거나 물류창고에서 보관 및 분류되어 있는 경우로 존재한다. 시간대별 배송상태가 기록되어 3가지의 relation을 통해 어디에 택배가 있었는지 알

수 있다.

3.1.5. truck

운송 차량인 트럭에 대한 entity이다. 차량 번호 'licence plate'와 운전수 'courier', 담당 구역인 'area in charge'이 속성으로 존재한다.

3.1.6. plane

해외 배송을 위한 비행기 entity이다. 비행기 번호인 'flight number'와 기장 'pilot', 담당 구역인 'area in charge'이 속성으로 존재한다.

3.1.7. warehouse

물류창고에 대한 entity이다. 창고를 식별하는 'warehouse number'와 해당 창고의 실제 주소인 'address'가 속성으로 존재한다.

3.1.8. payment

정기 결제를 제외한 모든 결제에 대한 정보가 저장된다. 택배가 있어야 결제가 이루어지므로, package에 dependent한 weak entity가 된다. 결제 금액과 날짜 및 시간이 속성으로 존재한다.

payment와 bill의 relation은 many to one의 관계이며 payment는 total, bill은 partial이다. 특정 기간 내의 여러 개의 택배에 대한 결제 내역이 하나의 청구서에 기록이 될 수 있고, 모든 결제 내역은 청구서에 나열되어서 고객에게 청구되어야 하기 때문이다.

3.1.9. bill

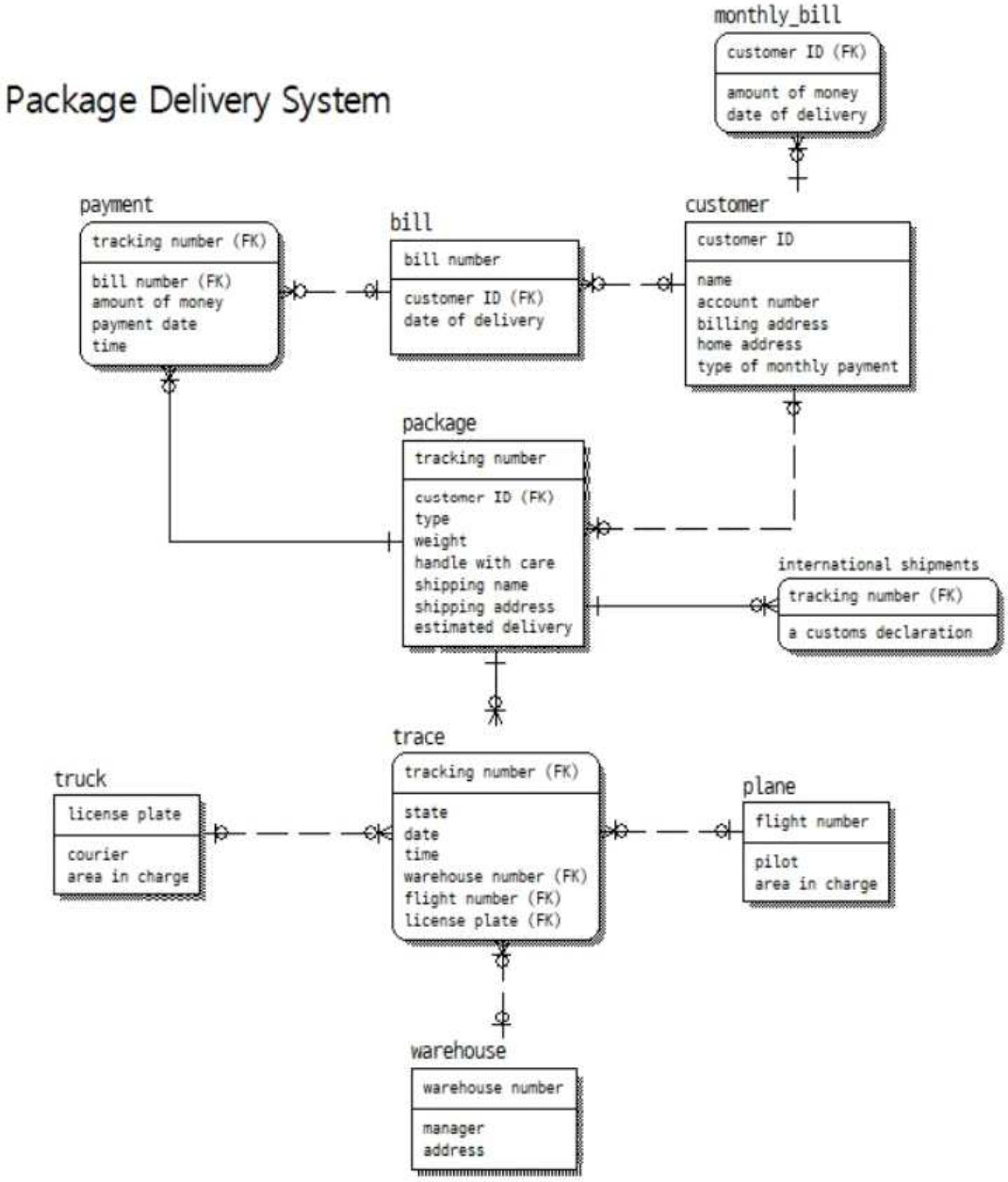
실제 배송되는 택배의 결제 정보들을 고객에게 청구하기 위한 entity이다. 각각의 bill을 식별하기 위한 bill number와 청구되는 날짜인 'date of delivery'가 속성으로 존재한다.

3.1.10. monthly_bill

정기 구독 고객에게 청구되는 bill entity이다. 정기 구독중인 고객이 있어야 구독료가 청구되므로 customer에 dependent한 weak entity이다.

3.2. relation schema diagram

E-R diagram의 relation을 reduce해주었다. relation에 해당하는 새로운 entity를 생성하진 않았다. cardinality가 one to many이면 one쪽 스키마의 PK를 many쪽 스키마의 FK로 속성을 추가시켰고 weak entity의 경우 부모 entity PK를 FK로 가져와서 자신의 PK로 추가해주었다.



< Attribute 분석 >

Entity	Attribute	Note	비고
customer	customer ID	PK	
	name		
	account number		
	billing address		
	home address		
	type of monthly payment	Y/N	
package	tracking number	PK	handle with care는 취급 주의 여부이다. (화학 물질, 유리 등 다양한 값이 올 수 있음)
	customer ID	FK	
	type		
	weight		
	handle with care		
	shipping name		
	shipping address		
	estimated delivery	YYYY-MM-DD	
international shipments	tracking number	PK, FK	
	a customs declaration		
trace	tracking number	PK, FK	트럭, 비행기, 창고 속성은 모두 NULL일 수 있다(배송완료된 경우)
	state		
	date	YYYY-MM-DD	
	time	HH.MM	
	warehouse number	FK	
	flight number	FK	
	license plate	FK	
truck	license plate	PK	
	courier		
	area in charge		
plane	flight number	PK	
	pilot		
	area in charge		
warehouse	warehouse number	PK	
	manager		
	address		
payment	tracking number	PK, FK	
	bill number	FK	
	amount of money		
	payment date	YYYY-MM-DD	
	time	HH.MM	
bill	bill number	PK	
	customer ID	FK	
	date of delivery	YYYY-MM-DD	
monthly_bill	customer ID	PK, FK	
	amount of money		
	date of delivery	YYYY-MM-DD	

4. 적용

▶ 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.

1721번 트럭이 2020년 4월 9일 15:00에 사고가 났다고 가정했을 때,

1. 사고가 난 트럭에 있던 상품의 발송인과 수령인

```
SELECT customer ID, name, p.shipping name
FROM customer
WHRER customer ID in (SELECT customer ID
        FROM trace as t, package as p
        WHERE date = '2020-04-09'
        AND state = '배송중'
        AND t.tracking number=p.tracking number)
```

trace와 package을 조인한 뒤, 해당 날짜에 적재한 모든 package중 state가 아직 배송중인 package의 발송인과 수령인을 select하면 찾을 수 있다.

2. 사고가 난 트럭에서 가장 마지막에 배송완료한 택배

```
SELECT tracking number
FROM trace as tt, package.pp
WHRER CONVERT(float, tt.time) = (SELECT max(CONVERT(float, t.time))
        FROM trace as t, package as p
        WHERE t.date = '2020-04-09'
        AND t.state = '배송완료'
        AND t.tracking number=p.tracking number)
AND tt.tracking number=pp.tracking number
```

time은 'HH.MM' 형태의 문자열로 저장되어있다. 이 값을 CONVERT 함수를 이용해 소수점으로 변환한 값 중 가장 큰 값이 제일 최근에 배송완료한 시점이다.