CSE4110 – Database System

Project1. E-R design and Relational Schema design

Spring 2022

20161230 박재형

[목차]

1. Introduction

본 프로젝트의 목표는 관계형 데이터베이스와 관련된 응용프로그램의 개념 설계, 논리적 설계, 구현, 운영 및 유지에 대한 현실적인 경험을 제공하는 것이다.

본 프로젝트에서 설계할 데이터베이스의 대상은, 온라인과 여러 오프라인 매장들을 운영하는 전자제품 판매회사이다. 다음의 조건들을 고려하여 이 회사의 데이터베이스를 구축해보자.

- 다양한 카테고리로 그룹화할 수 있는 많은 상품들이 존재한다. 그리고 몇몇 고객들은 회사와 계약을 맺어 구매한 상품들의 금액을 매달 1번씩 한 번에 계좌로 지불한다. 그 외의 고객은 신용/직불 카드로 온라인에서 구매할 수 있으므로 카드 정보를 저장해야 한다.

- 온라인으로 구매한 상품은 배송회사를 통해 배송되어야 하며, 고객의 요구사항에 응답하기 위해 배송 추적 번호를 저장해야 한다.

- 오프라인 매장과 온라인 매장을 위한 창고에서 상품의 재고는 항상 정확해야 하며, 재고가 부족한 경우 제조사에 연락해서 재주문을 신청하고 이를 기록해야 한다. 그리고 재고가 다시 채워졌다면 재주문이 완료되었다는 표시를 해주어야 한다.

- 판매 데이터는 기업의 향후 계획에 중요하다. 따라서 마케터는 기간별, 상품별, 그룹별, 계절별, 지역별 등으로 상품의 판매 기록을 보기를 원한다.

2. Entity & Attribute

1) Product - 판매하는 모든 상품에 대한 정보를 제공한다.

Attribute name	Domain
product_ID	VARCHAR(20)
production_date	DATE
product_type	VARCHAR(20)
product_price	INT
manufacturer	VARCHAR(30)

product_ID : 제품의 고유 모델명을 의미한다.

production_date : 제품을 생산한 날짜를 의미한다. (ex. 20220410)

product_type : 제품의 유형을 의미한다. (ex. TV, 에어컨, 청소기)

product_price : 제품의 가격을 의미한다.

manufacturer : 제품을 생산한 제조사를 의미한다. (ex. 삼성, LG)

2) package - 여러 상품으로 구성되는 패키지에 대한 정보를 제공한다.

Attribute name	Domain
package_ID	VARCHAR(30)
package_price	INT

package_ID : 패키지의 고유 ID를 의미한다. (ex. Gateway PC)

package_price : 패키지의 가격을 의미한다.

3) customer - 고객에 대한 정보를 제공한다.

Attribute name	Domain
customer_ID	CHAR(12)
customer_name	VARCHAR(20)
phone_number	VARCHAR(15)
address	VARCHAR(30)

customer_ID : 고객의 고유 ID. 가입일자 6자리와 해당 날짜에 가입한 순서 6자리로 ID를 구성한다. 하루에 100만명 미만의 신규 고객이 존재한다고 가정한다. (ex. 220410 + 000001)

customer_name : 고객의 이름을 의미한다.

phone_number : 고객의 전화번호를 의미한다.

address : 고객이 사는 지역을 의미한다.

4) payment - 지불할 수단과 계좌/카드번호를 제공한다.

Attribute name	Domain
payment_number	VARCHAR(20)
payment_type	VARCHAR(10)

payment_number : 지불할 수단의 개인 번호를 의미한다. (계좌번호 or 카드번호)

payment_type : 지불할 수단을 의미한다. account 또는 credit 또는 debit을 속성값으로 가진다.

5) shipment - 온라인으로 구매한 제품의 배송 관련 정보를 제공한다.

Attribute name	Domain
tracking_number	CHAR(12)
shipping_company	VARCHAR(20)
send_time	TIMESTAMP
promised_deliver_time	TIMESTAMP, NULL

tracking_number : 운송 추적 번호를 의미한다. 이 번호를 통해 운송중인 상품의 정보를 유일하게 식별할 수 있다. 운송 시작일자 6자리와 해당 일에 배송한 순서 6자리로 구성된다. (220401 + 000001)

shipping_company : 제품의 운송을 맡은 운송 회사명을 의미한다.

send_time : 제품의 운송을 시작한 시간을 의미한다.

promised_deliver_time : 도착하기로 약속한 시간을 의미한다. 별도의 기한이 없다면 NULL값을 갖는다.

6) inventory - 현재 각 상점과 창고에 있는 상품들의 재고에 대한 정보를 제공한다.

Attribute name	Domain
inventory_ID	VARCHAR(6)
inventory_type	VARCHAR(10)
region	VARCHAR(30)

inventory_ID : 상점 또는 창고의 고유 번호를 의미한다.

inventory_type : 상점인지 창고인지를 의미한다.

region : 상점 또는 창고가 위치한 지역을 의미한다.

7) bill - 고객이 상품을 결제한 기록이 담긴 영수증에 대한 정보를 제공한다.

Attribute name	Domain
bill_ID	CHAR(14)
product_count	INT
purchase_date	TIMESTAMP

bill_ID : 영수증의 고유 번호를 의미한다. 한번에 여러 상품을 구매한 경우에, 구매한 각 모델에 대해서 별도로 저장한다. 구매일자 6자리 + count 8자리로 구성된다.

product_count : 상품의 구매 개수를 의미한다.

purchase_date : 상품을 구매한 날

8) reorder_list - 상점 또는 창고에서 요청한 재주문에 대한 정보를 제공한다. 각 상품 모델에 대해서 각각 저장한다.

Attribute name	Domain
reorder_ID	CHAR(12)
is_filled	VARCHAR(3)
fill_count	INT

reorder_ID : 재주문한 요청의 고유 번호를 의미한다.

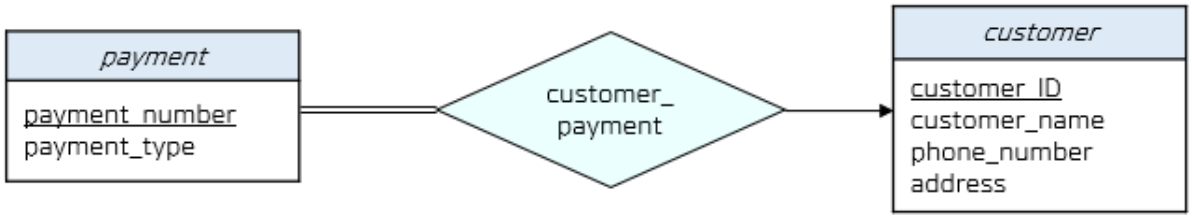
is_filled : 재주문 처리가 완료되었는지를 의미한다. 'YES' 또는 'NO'의 값을 갖는다.

fill_count : 채워야 하는 상품의 개수를 의미한다.

3. Relationship & Mapping Cardinality

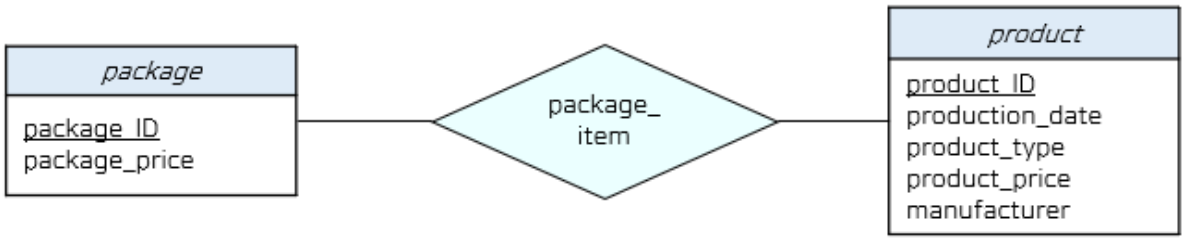
1) customer_payment

고객별로 지불하는 수단(계좌, 신용카드, 직불카드)과 그에 맞는 계좌/카드번호를 연결한다. 모든 지불 수단은 반드시 1명의 고객과 연결되어야 하고, 한 고객은 여러 지불 수단을 가질 수 있다. 따라서 many-to-one 관계로 이어주며, payment는 total이다.



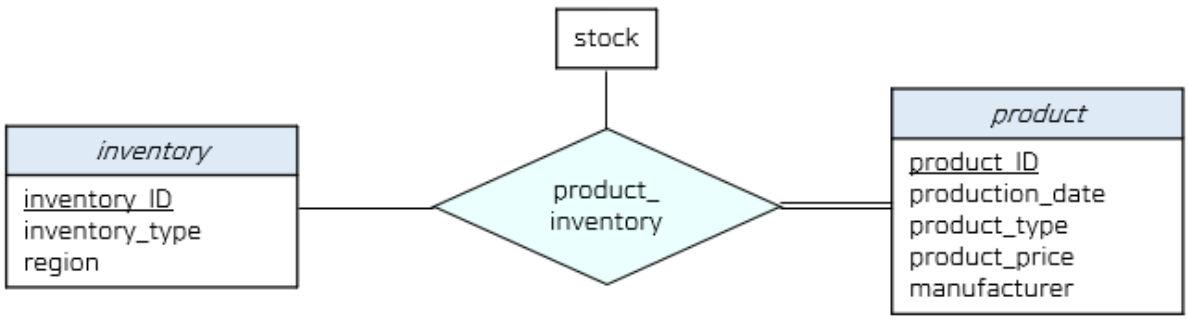
2) package_item

여러 상품을 모아서 하나의 패키지를 구성할 수 있으므로 상품과 패키지를 연결해준다. 각 상품은 여러 패키지에 들어있을 수 있고, 각 패키지에도 여러 상품이 포함되어 있으므로 many-to-many 관계로 이어준다.



3) product_inventory

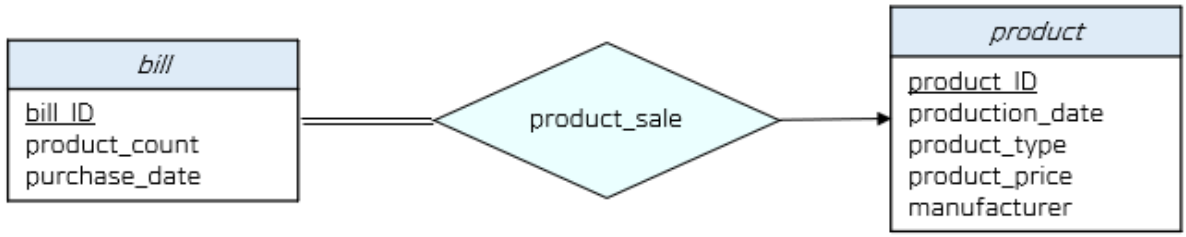
각 상점 또는 창고와 상품들을 연결해준다. 이를 통해 각 상점과 창고에 존재하는 상품의 정보를 알 수 있으며, stock이라는 relationship 속성을 통해 각 상품의 재고 현황을 파악할 수 있다. 각 상품은 반드시 창고 또는 상점의 물품 리스트에 포함되어 있어야 하므로 total이고, many-to-many 관계를 이룬다.



4) product_sale

결제 기록이 담긴 영수증과 상품을 연결한다. 이를 통해 판매된 상품에 대한 정보를 파악할 수 있다.

결제 기록은 동일한 모델에 대해서만 한 영수증으로 저장하기 때문에, 한 영수증에 한 상품만 포함될 수 있고, 여러 영수증에 동일한 상품이 존재할 수 있으므로 many-to-one 관계를 이루고, 각 영수증에 담긴 상품은 모두 product에 존재해야 하므로 total이다.



5) package_sale

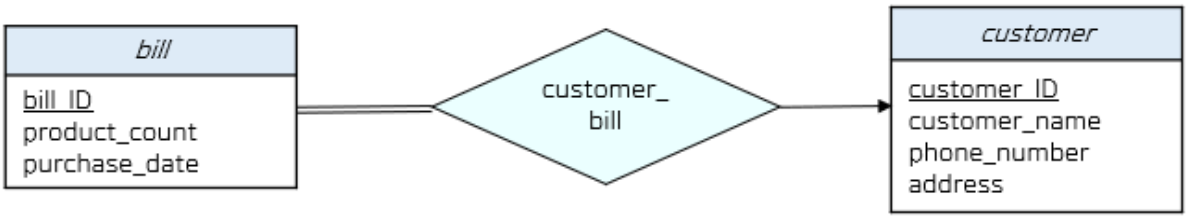
결제 기록이 담긴 영수증과 패키지를 연결한다. 이를 통해 판매된 패키지에 대한 정보를 파악할 수 있다.

결제 기록은 동일한 패키지에 대해서만 한 영수증으로 저장하기 때문에, 한 영수증에 한 패키지만 존재할 수 있고, 여러 영수증에 동일한 패키지가 존재할 수 있으므로 many-to-one 관계를 이루고, 각 영수증에 담긴 상품은 모두 package에 존재해야 하므로 total이다.

6) customer_bill

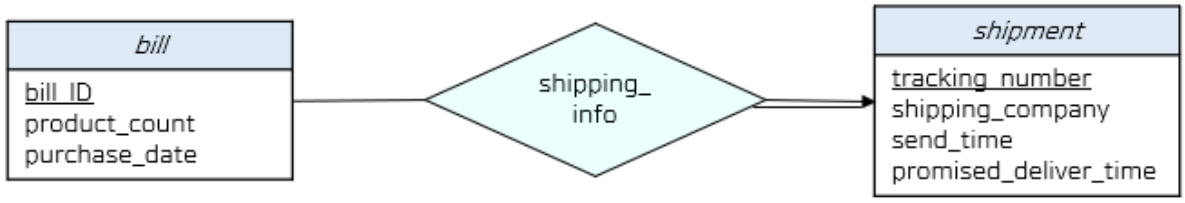
결제 기록이 담긴 영수증과 고객을 연결한다. 이를 통해 어떤 고객이 무슨 상품을 구매했는지를 파악할 수 있다.

한 고객이 여러 결제를 진행할 수 있으므로 many-to-one 관계를 이루며, 각 영수증에는 반드시 고객이 존재해야 하므로 total이다.



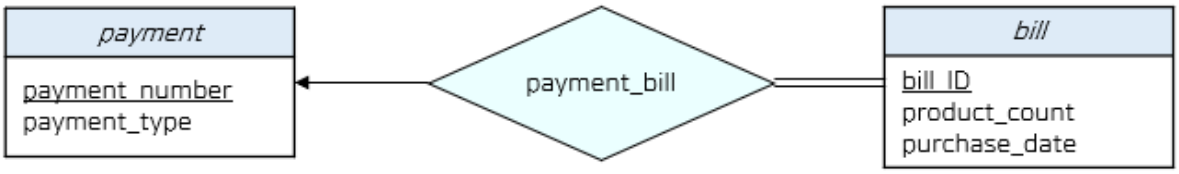
7) shipping_info

결제 중 온라인 결제에 대해서 배송과 연결해준다. 여러 상품을 동시에 구매한 고객에게 묶음으로 배송하기 때문에 여러 영수증이 하나의 배송에 연결될 수 있으므로 many-to-one 관계를 갖는다. 그리고 각 배송에 대해선 반드시 결제 내역이 존재해야 하므로 total이다.



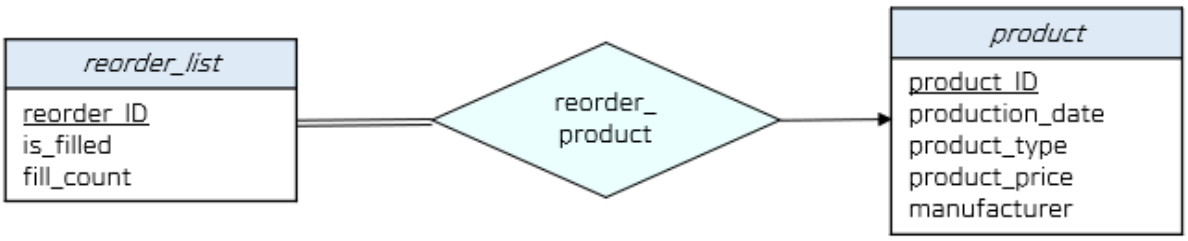
8) payment_bill

각 결제별로 이용한 결제 수단을 연결해준다. 각 결제 수단이 여러 상품 결제에서 사용될 수 있으므로 one-to-many 관계이며, 모든 결제는 반드시 결제 수단이 필요하므로 total이다.



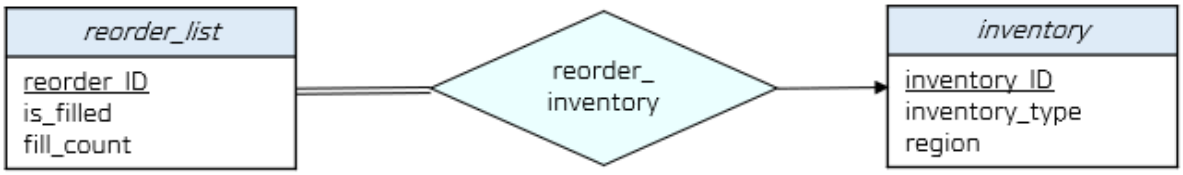
9) reorder_product

재주문 리스트와 상품을 연결해준다. 재주문한 상품에 대한 정보를 파악할 수 있다. 각 상품은 여러 번 재주문될 수 있으므로 many-to-one 관계를 이루며, 재주문 정보엔 반드시 상품이 존재해야 하므로 total이다.

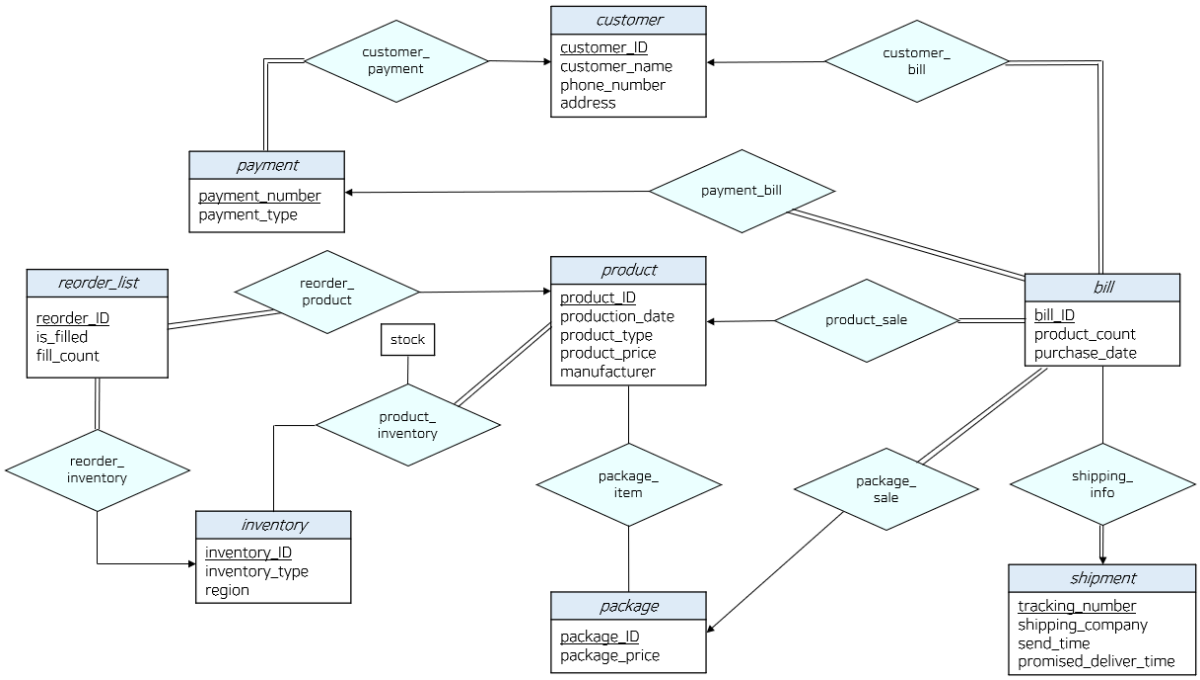


10) reorder_inventory

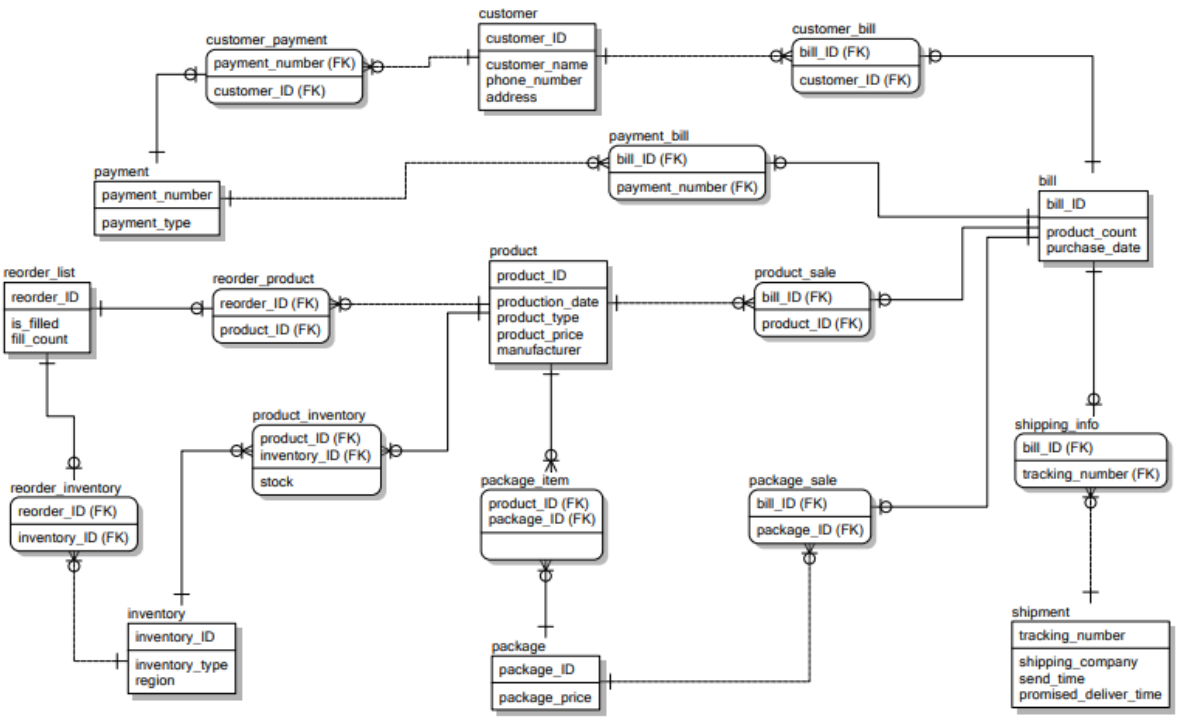
재주문 리스트와 상점/창고를 연결해준다. 재주문을 요청한 상점/창고에 대한 정보를 파악할 수 있다. 각 상점은 여러 번 재주문할 수 있으므로 many-to-one 관계를 이루며, 재주문 정보엔 반드시 상점/창고가 존재해야 하므로 total이다.



4. E-R Diagram



5. Relational Scheme Diagram



1) customer_payment

customer -> customer_payment : One or More

payment -> customer_payment : exactly One

각 고객별로 최소 1개의 결제 수단을 갖고, 각 결제 수단은 반드시 1명의 고객의 정보이어야 한다.

2) customer_bill

customer -> customer_bill : Zero, One or More

bill -> customer_bill : exactly One

각 고객별로 결제된 영수증이 0개 이상 존재할 수 있고, 각 영수증은 반드시 1명의 고객에 연결되어야 한다.

3) payment_bill

payment -> payment_bill : Zero, One or More

bill -> payment_bill : exactly One

각 결제수단별로 결제된 영수증이 0개 이상 존재할 수 있고, 각 영수증은 반드시 1개의 결제수단에 연결되어야 한다.

4) product_sale

product -> product_sale : Zero, One or More

bill -> product_sale : exactly One

각 상품별로 결제된 영수증이 0개 이상 존재할 수 있고, 각 영수증은 반드시 1개의 상품에 연결되어야 한다.

5) package_sale

package -> package_sale : Zero, One or More

bill -> package_sale : exactly One

각 패키지별로 결제된 영수증이 0개 이상 존재할 수 있고, 각 영수증은 반드시 1개의 패키지에 연결되어야 한다.

6) shipping info

bill -> shipping info : exactly One

shipment -> shipping info : One or More

각 영수증(상품)은 반드시 1개의 배송으로 이루어져야 하고, 각 배송은 여러 영수증(상품)을 한 번에 이루어질 수 있다.

7) package_item

product -> package_item : Zero, One or More

package -> package_item : One or More

각 상품은 패키지에 포함되지 않을 수도 있고, 여러 패키지에 포함될 수도 있다. 각 패키지는 반드시 1개 이상의 상품을 포함해야 한다.

8) product_inventory

product -> product_inventory : One or More

inventory -> product_inventory : One or More

각 상품은 여러 매장/창고에서 보관될 수 있으며, 각 매장/창고는 여러 상품들을 보관한다.

9) reorder_product

reorder_list -> reorder_product : exactly One

product -> reorder_product : Zero, One or More

각 재주문서는 반드시 한 상품만 작성되어야 하며, 각 상품은 여러 번 재주문 될 수 있다.

10) reorder_product

reorder_list -> reorder_inventory : exactly One

inventory -> reorder_product : Zero, One or More

각 재주문서는 반드시 한 상점/창고만 작성되어야 하며, 각 상점/창고는 여러 번 재주문 할 수 있다.