

## Project 1. E-R design and Relational Schema design

# **Electronics Vendor company**

# 목차

## 1. 개요

## 2. E-R Diagram

### 2.1 Entity Sets

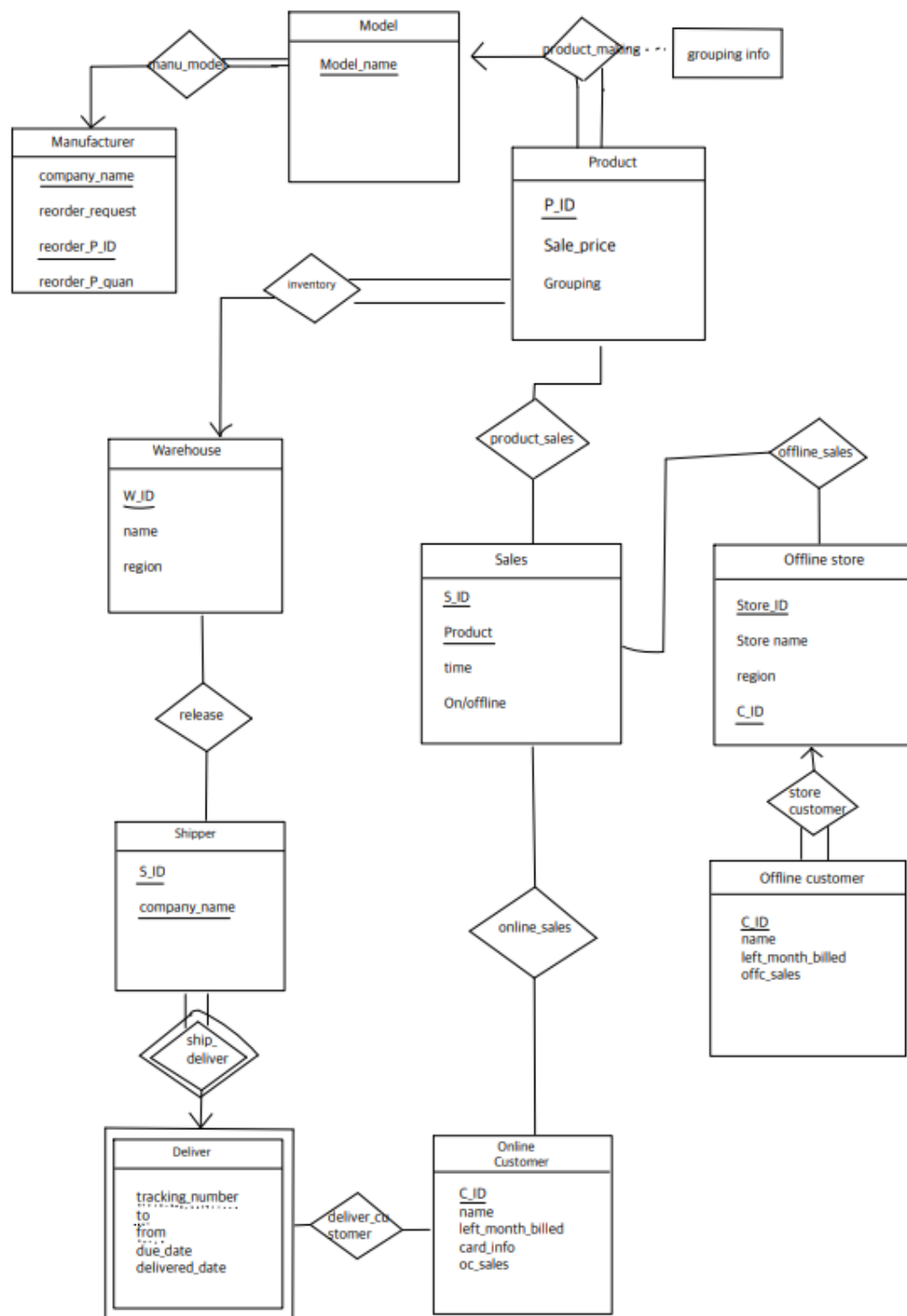
### 2.2 Relation Sets

## 3. Relational Schema Diagram

## 1. 개요

이번 프로젝트의 목표는 DBA가 되어 전자기기 판매 회사의 database를 설계하는 것이다. 전자기기 판매 회사는 온/오프라인 매장을 모두 가지고 있으며 전자기기는 회사별로 판매할 수 있으며 종류별로(노트북, 핸드폰) 판매할 수 있고, 세트로 판매할 수도 있다. 온라인 매장은 온라인 고객에게 배송하는 과정을 포함해야 하며 창고가 존재한다. 또한 여기에 몇가지 조건들과 추후 작성할 쿼리들을 고려하여 database를 작성했다.

### 2.1 E-R diagram



다음과 같이 ER diagram을 작성했다. Entity들과 설명은 다음과 같다.

#### **Sales(S\_ID, Product, time, On/offline)**

판매를 저장한다. 판매가 일어났을 때 해당 상품과 시간,온라인인지 오프라인인지를 기록한다. Sales data를 위해 매출이 일어난 시기를 저장하고 이것을 추후 season별/ time period/ reorder로 분석할 수 있다.

#### **Product(P\_ID, Sale\_price, Grouping)**

상품을 저장한다. 상품의 id와 판매 가격, 어떤 grouping을 저장한다.

#### **Model(Model\_name)**

Model name을 PKEY로 갖는 entity이다. 핸드폰, 노트북 등을 의미한다.

#### **Manufacturer(company\_name, reorder\_request, reorder\_P\_ID, reorder\_P\_quan)**

LG,삼성 등 electronic device의 제조사를 저장하는 entity이다. 핸드폰,노트북 등을 만드는 제조사이며 회사 이름을 PKEY로 가진다. 요구사항에 존재하는 reorder가 발생했을 때 제조사에게 주문을 넣어야 한다는 점을 고려하여 reorder\_request로 reorder\_request를 받았는지 여부를 저장하고, reorder\_P\_ID로 reorder를 한 상품의 ID를 저장하며 reorder\_P\_quan으로 reorder한 product의 수량을 저장한다.

#### **Warehouse(W\_ID, name, region)**

Inventory에 해당하는 entity이다. 이것은 online store을 위한 entity이고 W\_ID의 PKEY를 가지고 이름과 warehouse의 지역을 저장한다.

#### **Shipper(S\_ID, company\_name)**

고려사항에 존재하는 online sales에 필요한 배송 회사이다. Warehouse에서 출고된 상품을 받아가는 배송 회사이다. S\_ID를 PKEY로 가지고 회사 이름을 저장한다.

#### **Deliver(tracking\_number, to, from, due\_date, delivered\_date)**

이것은 Shipper가 존재하지 않으면 존재할 수 없는 weak entity set이다. Tracking number을 PKEY로 가지고 이것은 customer가 배송 정보를 확인할 때 사용될 수 있다. 발송자 to (Online\_customer을 의미하는 FKEY), 수신자 from(shipper을 의미하는 FKEY), 배송마감일 due\_date, 실제 배송일 delivered\_date를 통해서 추후 작성할 query에서 배송이 제때 이루어졌는지 확인하는 용도로 사용한다.

#### **Online\_customer(C\_ID, name, left\_month\_billed, card\_info, oc\_sales)**

온라인 고객의 정보를 저장하는 entity이다. C\_ID를 PKEY로 가지고, left\_month\_billed를 통해 요구사항에 존재하는, 어떤 고객은 회사와 할부 계약을 맺었다는 정보를 저장한다. 그리고 고객의 카드 정보와 고객 별 매출량을 저장할 수 있도록 oc\_sales를 추가했다.

#### **Offline\_store(Store\_ID, store\_name, region, C\_ID)**

Offline store의 정보를 저장하는 entity이다. Store\_ID를 PKEY로 가지고 오프라인 매장의 이름,지역, 매장의 고객(판매 고객)을 저장한다. 이것은 추후에 매장 별 매출을 집계할 때 사용된다.

### **Offline\_customer(C\_ID, name, left\_month\_billed, offc\_sales)**

Offline 고객의 정보를 저장한다. C\_ID를 PKEY로 가지고, online customer와 마찬가지로 잔여 할부금, 고객별 매출을 집계할 때 사용할 offc\_sales를 가진다.

2.2 Relation set에 대한 설명은 다음과 같다.

### **Manu\_model**

Manufacturer과 model간의 일대다 관계이다. 하나의 제조사는 여러 개의 model을 만들 수 있다.

### **Product\_making**

Product와 Model의 일대다 관계이다. 하나의 모델에는 여러 개의 상품이 있을 수 있다. 노트북 model 하나에 여러 개의 상품이 존재할 수 있기 때문이다.

### **Inventory**

Product와 Warehouse간의 일대다 관계이다. 하나의 warehouse에는 여러 개의 product가 존재할 수 있다.

### **Product\_sales**

Product와 sales간의 일대일 관계이다. 일단 하나의 상품은 한 번만 판매가 되기 때문이다.

### **Release**

Warehouse의 상품이 shipper에게 전달되는 출고 관계이다.

### **Offline\_sales**

Offline store에서 이루어지는 판매를 나타내는 relation이다.

### **Online\_sales**

Online customer에게 이루어지는 판매를 나타내는 relation이다.

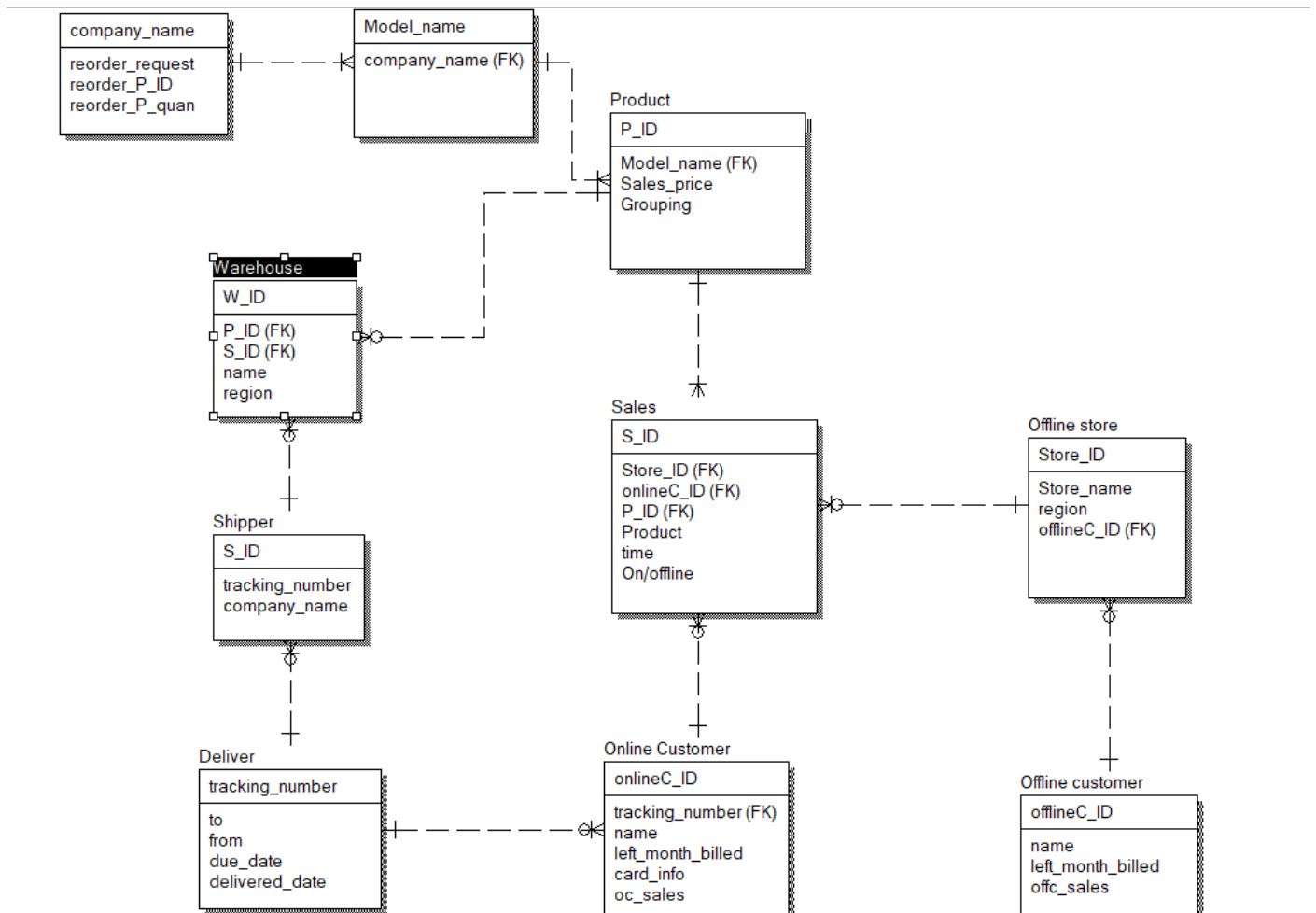
### **Ship\_deliver**

Shipper가 전달하는 배송 정보를 저장하는 관계이다.

### **Deliver\_customer**

Online customer와 배송 정보 간의 relation이다. Customer가 여러 상품을 주문해도 각 상품별 주문정보는 하나씩만 존재한다는 현실의 모습에 착안하여 일대일 관계로 설정했다.

3 위의 E-R diagram과 relation을 바탕으로 만든 Relation Schema diagram은 다음과 같다.



ER diagram에서 표시되지 않았던 foreign key들이 나타내어진 것을 확인할 수 있다.