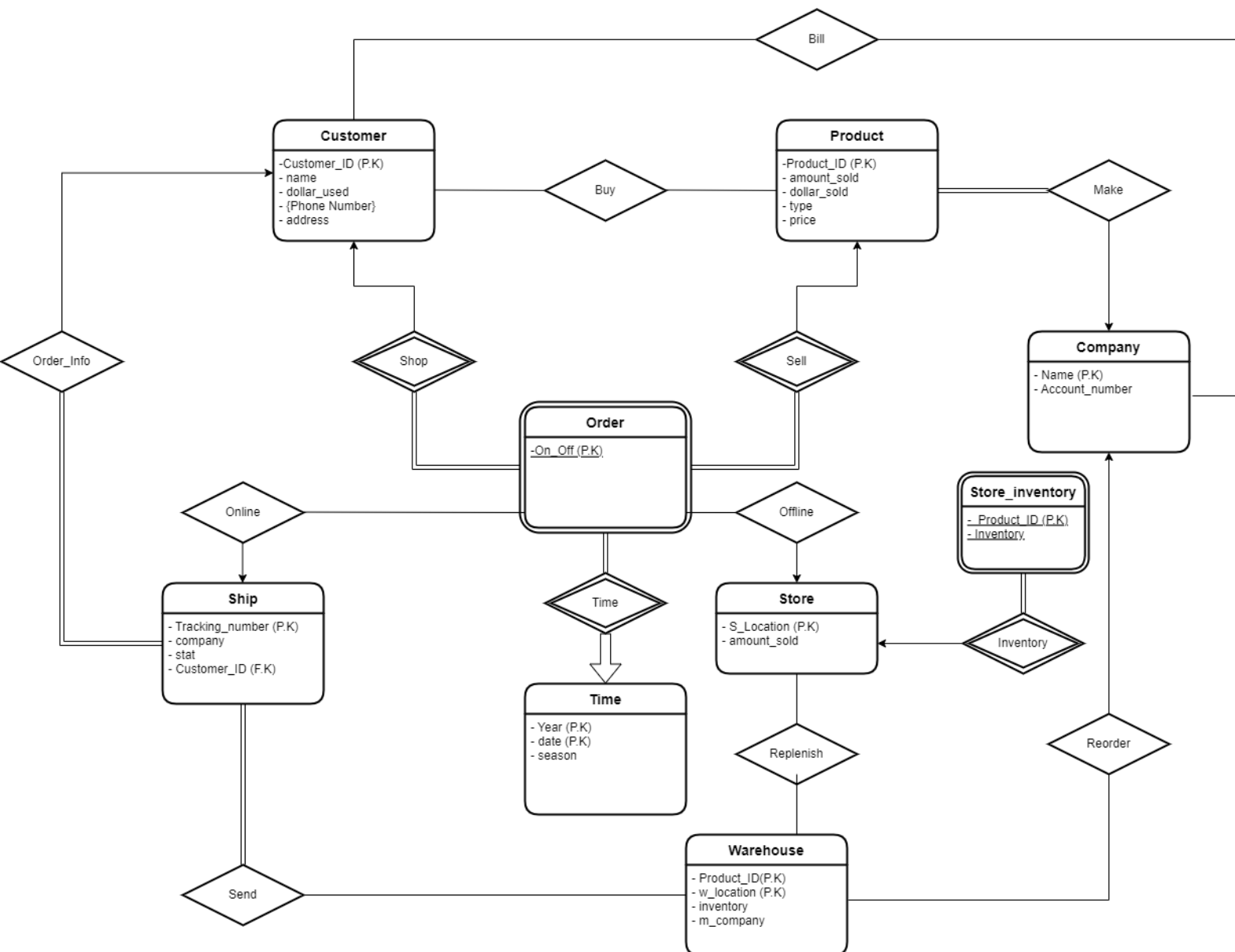


# 데이터베이스시스템(CSE4110) Project 1

20181632 박성현

## 1. E-R model



## A. Entity

- i. Customer – 기본적인 고객의 정보이다. 각 고객을 식별하는 Customer ID를 Primary Key로 가지고 있으며 고객의 name, dollar used(누적 사용금액), Phone number(multivalued), 그리고 address를 가지고 있다.
- ii. Product – 상품의 기본정보이다. 한 종류의 상품을 나타내며 상품의 Product ID를 Primary Key로 가지고 있으며 amount sold(누적 판매 수량), dollar sold(누적 판매 금액), type을 attributes로 가진다.
- iii. Company – 상품을 만든 회사에 대한 정보이다. 회사의 Name을 Primary Key로 가지고 있으며 이 회사와 계약한 고객들이 매월 돈을 이체하는 회사의 계좌번호 account number이 있다.
- iv. Order – 상품의 구매이력들을 저장하는 Entity이다. 이 Entity는 그 구매가 온라인으로 이루어진 구매인지 오프라인으로 이루어진 구매인지 판별하는 On\_off를 Primary Key로 가지고 있다. 이 entity는 weak entity set으로써 Time entity의 year 과 date, Customer의 Customer ID, Product의 Product ID를 Primary Key로 상속 받아야 각 tuple들을 unique하게 define할 수 있다. .
- v. Ship – 상품의 배송 정보이다. 온라인 고객들만 가지고 있는 정보이며 Tracking\_number을 Primary Key로 가진다. 상품의 회사와 배송 상태, 그리고 그 배송을 받는 Customer ID를 Customer을 referencing하는 Foreign Key로 가진다.
- vi. Time – Order이 일어난 시간을 저장하며 년도, 날짜를 Primary Key로 가지며 계절을 기본 속성으로 가진다.
- vii. Store – 오프라인 고객들이 가지고 있는 정보들이다. S\_Location, 즉 가게의 위치를 Primary Key로 가진다.
- viii. Store\_inventory – 가게에 있는 물품의 재고이다. Weak entity로써 store이 identifying 해준다. Store의 S\_Location을 가져와야 각 tuple을 정의할

수 있다.

- ix. Warehouse – 창고를 나타낸다. 각 물품의 ID, 그리고 창고의 위치를 Primary Key로 가지고 있으며 재고와 생산 업체를 기본 속성으로 가진다.

## B. Relationship sets

- i. Buy (Customer\_ID, Product\_ID) – Customer과 Product를 연결하는 관계로써 누가 어떤 물건을 샀는지 알려준다. Many-to-Many 관계이다.
- ii. Bill (Customer\_ID, Name) – Customer과 Company를 연결하는 관계로 매달 정기적으로 회사에 돈을 내도록 계약한 고객과 그 회사의 계좌번호이다. Many-to-Many 관계이다.
- iii. Make (Product\_ID, Name) – Product 와 Company를 이어주는 관계로 어떤 물품이 어디서 생산됐는지 알려주는 관계이다. One-to-Many 이며 Product쪽은 total이다.
- iv. Shop, Sell, Time (Customer\_ID, On\_Off, Product\_ID, year, date) – 이 세개의 관계는 모두 같은 값을 가진다. Order이 3개의 entity로의 weak한 entity이기에 같은 값을 가진다. 이 물건이 누가 언제 샀는지에 대한 정보이다. 하나의 relationship이 여러 entity와 관계를 갖는 것을 배제하기 위해 order이란 임의의 entity를 만들어 binary relationship으로 구성했다. One-to-Many 관계이며 Order쪽이 모두 total이다. Time relation은 양쪽 다 total이다.
- v. Online(Customer\_ID, On\_Off, Product\_ID, year, date, Tracking\_number) – order과 ship을 연결한다. 어떤 주문이 온라인으로 이루어 졌을 때, 배송 정보와 주문을 나타낸다. One-to-Many관계이다.
- vi. Offline(Customer\_ID, On\_Off, Product\_ID, year, date, S\_Location) – Order과 Store을 연결한다. 어떤 주문이 오프라인으로 이루어 졌을 때, 가게의 위치와 주문 정보를 나타낸다. One-to Many관계이다.
- vii. Inventory( Product\_ID, S\_Location) – Store과 Store\_inventory를 연결하는 관계이다. 한 가게와 그 가게의 하나의 물품을 나타내며 이 관계로 store\_inventory에서 그 물품의 재고를 알 수 있다. One-to-Many관계이며 Store\_inventory쪽이 total이다.

- viii. Replenish(S\_Location, Product\_ID, w\_location) – Store 과 Warehouse를 연결하는 관계로 가게의 물품이 부족할 때, 창고로 물품을 주문하는 관계를 나타낸다. Many-to-Many관계이다.
- ix. Reorder(Name, Product\_ID, w\_location) – Company와 Warehouse를 연결하는 관계로 창고에 물품이 부족할 때, 생산업체로 주문을 넣는 관계이다. Many-to-One이다. 예를 들어 아이폰은 apple회사에서만 만들기에 특정 id를 가진 물품은 하나의 회사에서만 주문할 수 있다.
- x. Send (Tracking\_number, Product\_ID, w\_location) – Warehouse와 Ship를 연결하는 관계로써, 온라인 주문이 있을 때, 창고에서 배송업체로 물품을 보내는 관계를 나타낸다. Many-to-Many이며 ship쪽이 total이다.
- xi. Order\_Info (Customer\_Id, Tracking\_number) – Customer과 Ship을 연결하는 관계로 배송과 한명의 고객을 연결한다. 이 관계로 이 배송이 어느 고객의 배송인지 알아 낼 수 있다. One-to-Many관계이며 ship쪽이 total이다.

## 2. Relational Schema Diagram

Erwin 프로그램을 사용해 ER model을 Relational Schema Diagram으로 변환시켰다. 가장 큰 차이점은 Entity들 간의 relationship set들을 처리하는 방식이다. 기본적으로 모든 attribute들은 not Null의 값들을 할당해 줬으며 각 attribute의 속성에 따라 varchar 혹은 Integer 값들을 설정했다. 먼저 Many-to-Many관계의 relationship set들은 relationship shema를 보존하며 양쪽의 Primary Key들을 가져와 relationship의 Primary Key로 할당하였다. Bill, Replenish, Send, Buy는 Many-to-Many관계이기에 relationship set을 그대로 유지하며 양쪽의 entity의 Primary Key들을 가져왔다. One-to-Many의 관계에서는 One쪽의 Primary Key를 Many쪽의 Foreign key로 가져오면서 relationship set을 제거하였다. Shop, sell, time, online, offline, inventory, Reorder, Order\_Info, Make Relationship들을 제거해주고 Many쪽에 One쪽의 Primary Key를 가져와 Foreign Key로 할당했다. 또한 Customer의 Multivalued-attribute 인 Phone\_number은 Customer entity에서 나오는 One-to-Many의 새로운 relationship을 만들어줬다. Customer\_Phone 이라는 Relationship은 Customer의 Primary Key인 Customer\_ID와 multivalued-attribute인 Phone\_number을 Primary key로 가진다. 그 결과로 Customer에서

Phone\_number은 제거되었다. 또한 Order entity의 F.K중 Tracking\_Number과 S\_Location은 고객이 온라인 혹은 오프라인으로 구매를 진행했는지에 따라 NULL 값을 가질 수 있기에 not null을 할당하지 않았다. 이들을 Relational Schema로 나타내면 다음과 같다.

Customer(Customer\_ID, name, dollar\_used, address)

Product(Product\_ID, amount\_sold, dollar\_sold, type, price, Name (F.K))

Order(On\_Off, Product\_ID, (F.K) Customer\_ID (F.K), Year (F.K), date (F.K), Tracking\_Number (F.K), S\_Location (F.K))

Store(S\_Location, amount\_sold)

Store\_inventory(Product\_ID, S\_Location (F.K), inventory)

Warehouse(Product\_ID, w\_location, inventory, m\_company, Name(F.K))

Company(Name, Account\_number)

Ship(Tracking\_Number, Company, stat, Customer\_ID (F.K))

Time(Year, date, season)

Bill(Customer\_ID, Name)

Buy(Product\_ID, Customer\_ID)

Replenish(Product\_ID, S\_Location, w\_location)

Send(Tracking\_number, Product\_ID, w\_location)

Customer\_Phone(Customer\_ID (F.K), Phone Number)