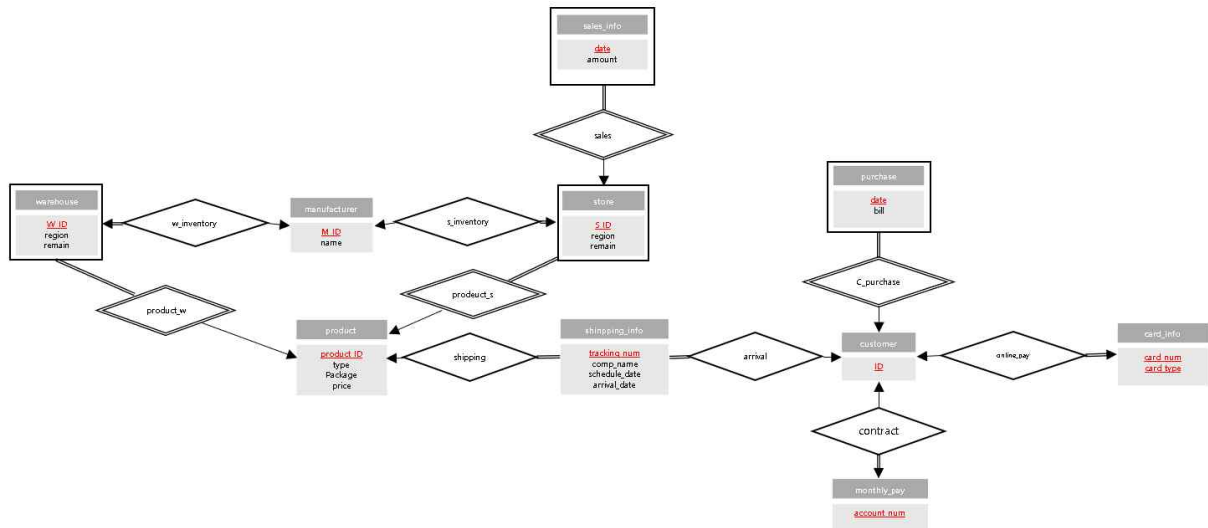


# 데이터베이스 시스템 project1 보고서

컴퓨터공학과 20181687 정지석

## 1. E-R diagram



### 1-1. entity

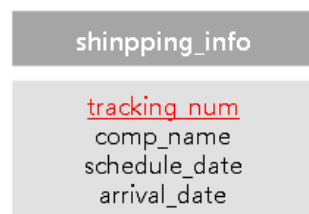
#### \*product



판매하는 상품에 대한 정보가 들어있는 entity이다.

각 물품을 구분할 수 있게 해주는 product\_ID가 primary key가 된다. 그 외의 속성으로 각 상품의 가격(price), 종류(type), 패키지(package, ex: 모니터 + 키보드 + 마우스)가 들어온다.

#### \*shipping\_info



온라인으로 주문한 상품의 배송 정보가 담겨있는 entity이다.

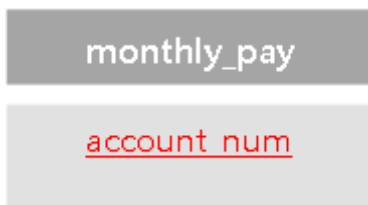
어떤 상품이 고객에게 배송되었는지를 구분하기 위해 tracking\_num(tracking number)를 primary key로 지정하였다. 그 외에도 어떤 회사가 배송을 진행 중인지를 알려주는 comp\_name(company name), 예정 도착일을 알려주는 schedule\_date, 그리고 실제로 고객에게 도착한 날짜인 arrival\_date를 속성으로 가지고 있다.

\*customer



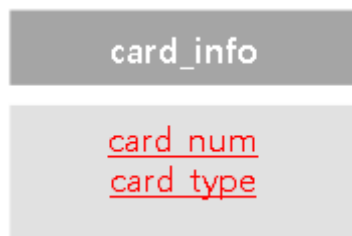
고객에 대한 정보가 들어있는 entity이다. 각각의 고객을 구분하기 위해 ID를 primary key로 가진다.

\*monthly\_pay



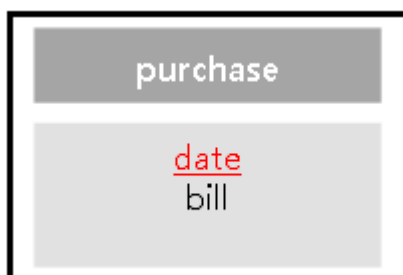
회사와 계약을 맺어 매달 계좌 이체로 한번에 값을 지불하는 고객들의 계좌 번호가 들어있는 entity이다. 계약을 한 고객들의 계좌 번호(account\_num)가 primary key로 들어간다.

\*card\_info



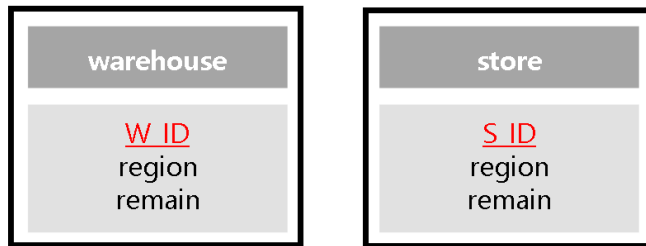
온라인으로 결제를 하는 고객들의 카드 정보가 들어있는 entity이다. 고객들의 카드 번호(card\_num)와 카드 종류(card\_type, credit 인지 debit 인지)가 primary key로 들어온다.

\*purchase



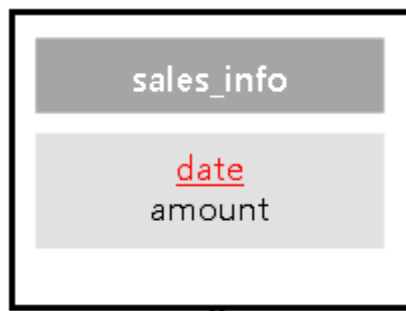
특정 고객이 특정일에 얼마만큼을 결제했는지 보여주는 entity이다. weak entity로 고객의 ID를 customer entity로부터 가지고 온다. 고객과 날짜를 모두 구분할 수 있어야 하기 때문에 상품의 ID(customer\_ID)와 date(날짜)가 primary key로 들어온다. bill(결제한 금액)은 같은 중복값이 나올 수 있기 때문에 primary key에 해당하지 않는다.

\*store, warehouse



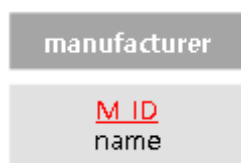
특정 상품이 특정 매장/창고에 얼마나 남아있는지에 대한 entity이다. weak entity로 상품의 ID를 product entity로부터 가지고 온다. 상품과 매장/창고를 구분할 수 있어야 하기 때문에 상품의 ID(product\_ID)와 매장/창고의 ID(S\_ID/W\_ID)를 primary key로 가진다. 매장/창고의 지역(region)과 남아있는 재고의 수량(remain)은 중복값이 나올 수 있기 때문에 primary key에 해당하지 않는다.

\*sales\_info



특정 상품이 특정일에 특정 매장에서 얼마나 팔렸는지에 대한 entity이다. weak entity로 상품의 ID와 매장의 ID를 store entity로부터 가지고 온다. 상품과 날짜, 매장을 모두 구분할 수 있어야 하기 때문에 상품의 ID(product\_ID)와 date, S\_ID(store ID, 매장의 ID) primary key로 가진다. 팔린 개수(amount)는 중복값이 나올 수 있기 때문에 primary key에 해당하지 않는다.

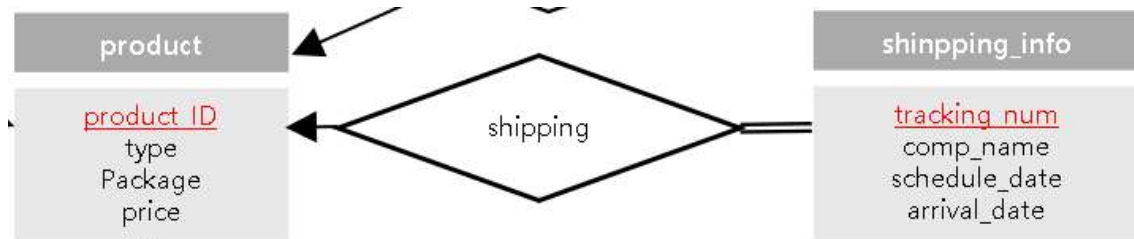
\*manufacturer



상품의 제조사에 대한 정보가 들어있는 entity이다. 각 제조사를 구분할 수 있게 해주는 M\_ID(manufacturer ID)가 primary key로 들어온다.

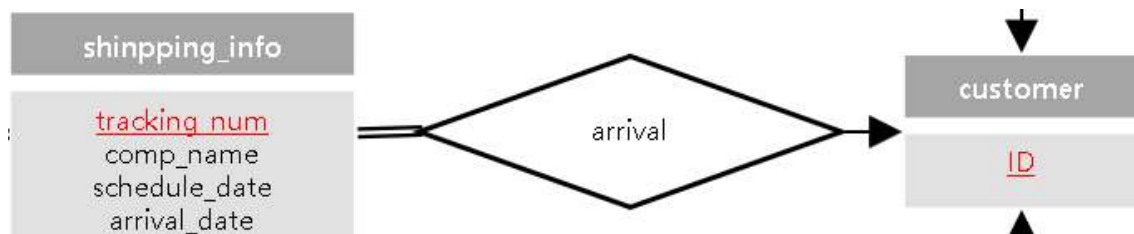
## 1-2. relationship

\*product-shipping-shippong\_info



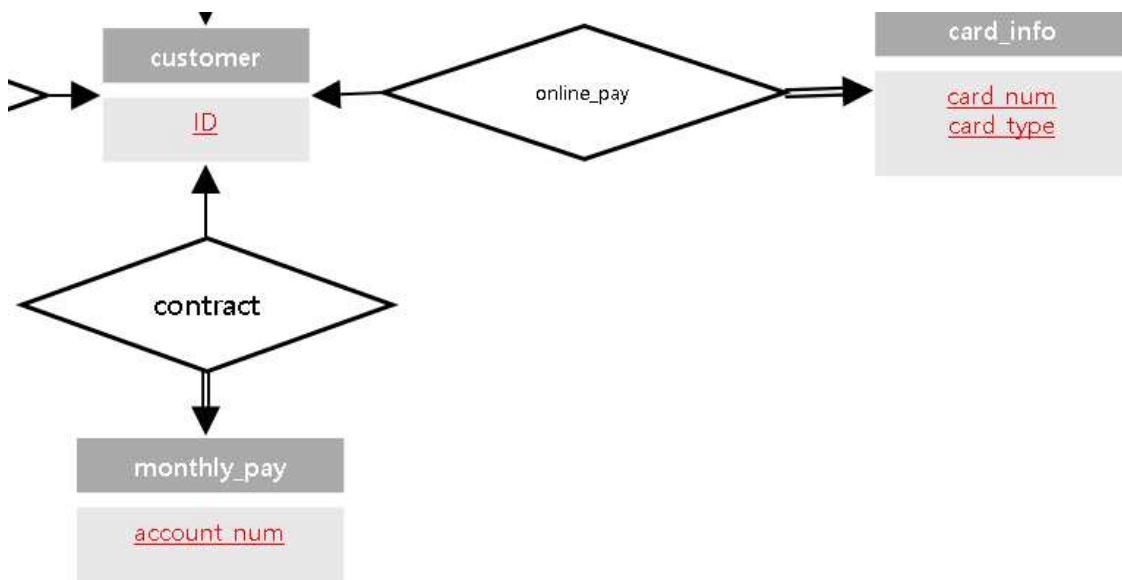
하나의 상품이 여러 고객에게 배송될 수 있기 때문에 one to many의 관계를 지닌다. 또한 온라인으로 배송된 상품은 모두 tracking\_num을 가지기 때문에 shipping\_info는 total이 된다.

\*shippong\_info-arrival-customer



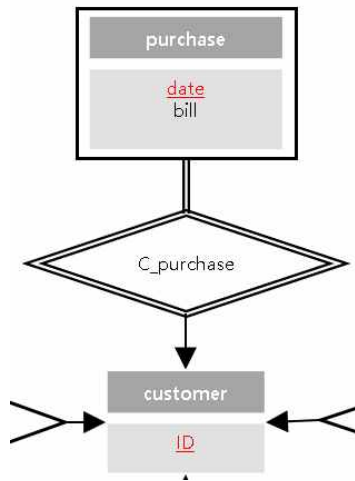
한 명의 고객이 온라인으로 여러번 주문을 할 수 있기 때문에 many to one의 관계를 지닌다. 모든 고객이 온라인으로 상품을 구매하는 것은 아니기에 customer 쪽은 total이 되지 않는다.

\*customer-contract-monthly\_pay/  
customer-online\_pay-card\_info



하나의 고객이 하나의 계좌/카드정보를 가지기에 둘다 one to one 관계이다. 모든 고객이 회사와 계약을 맺지 않았기 때문에/모든 고객이 온라인으로 상품을 구매하지 않기 때문에, customer 쪽은 total이 되지 않는다.

\*customer-C\_purchase-purchase



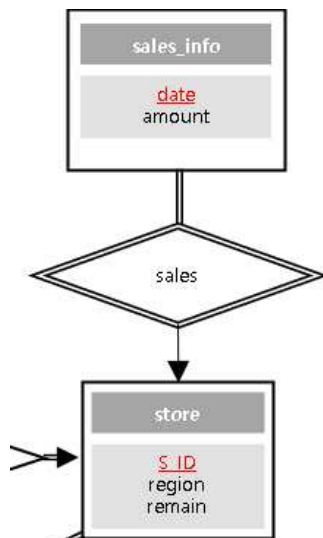
weak entity와 identifying entity를 연결해주는 relationship이다. 한 명의 고객이 각 날짜에 결제한 금액을 가지기 때문에 one to many의 관계이다. 아무 것도 구입하지 않은 고객이 있을 수 있기 때문에 customer는 total이 되지 않는다.

\*product-product\_w-warehouse/  
product-product\_s-store



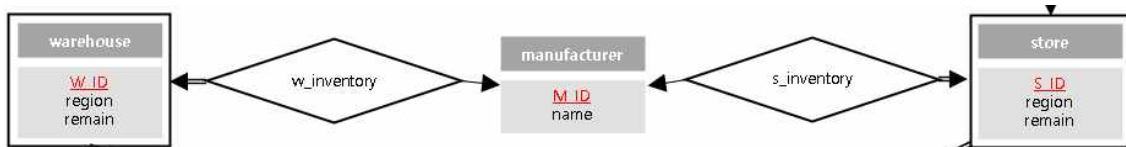
weak entity와 identifying entity를 연결해주는 relationship이다. 하나의 상품이 여러 개의 창고/매장에 존재할 수 있기 때문에 one to many의 관계를 가진다. 특정 상품의 경우 특정 매장/창고에 존재하지 않을 수도 있기 때문에 product는 total이 되지 않는다.

\*store-sales-sales\_info



하나의 상품이 여러 날짜에 여러 매장에서 팔리기 때문에 one to many의 관계를 가진다. 한번도 팔린적 없는 상품이 존재할 수 있기 때문에 store는 total이 되지 않는다.

\*manufacturer-w\_inventory-warehouse/  
manufacturer-s\_inventory-store



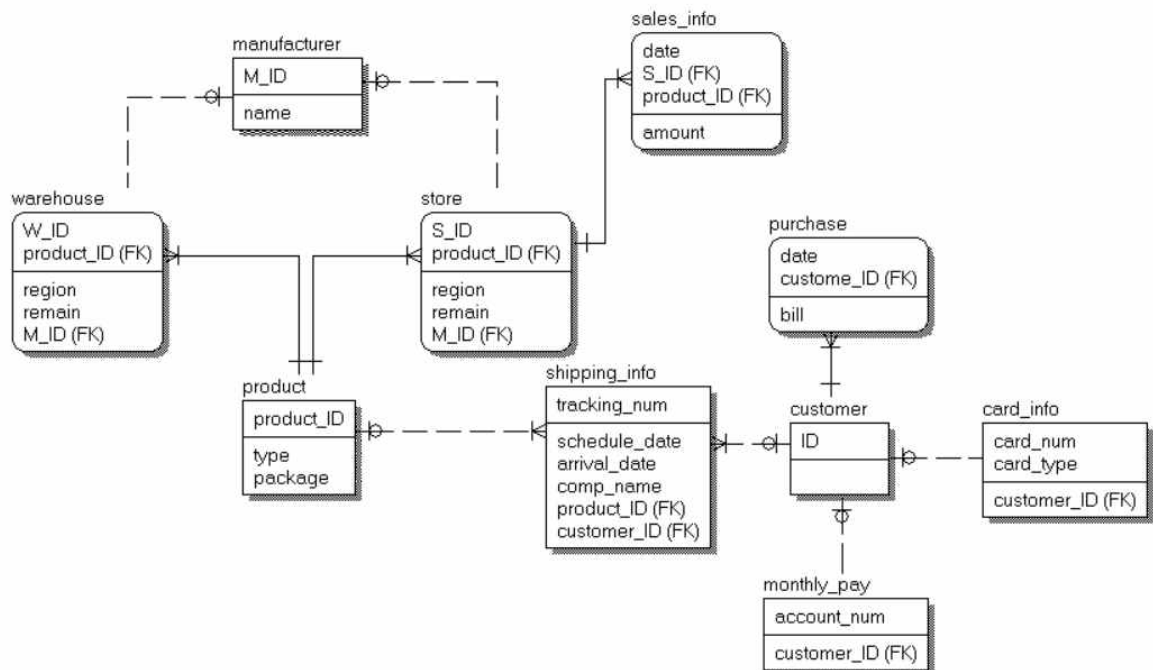
각 창고/매장에 있는 각 상품은 하나의 제조사를 가지고 있기 때문에 one to one의 관계이다. 모든 상품은 그 상품을 제조한 제조사가 존재하므로 warehouse/store는 total이 된다.

## 2. Schema Diagram

ER diagram을 schema diagram으로 변환할 때 one to many relation은 one 쪽의 primary key를 many 쪽의 속성에 추가했다. one to many relation 중 many 쪽이 weak entity일 경우에는 one 쪽(identifying entity)의 primary key를 many 쪽(weak entity)의 primary key에 추가해 주었다.

one to one relation의 경우 양쪽 중 아무 것이나 many의 역할을 수행 할 수 있다. 하지만 total이 아닌 one이 many의 역할을 수행할 경우 null 값이 들어올 수 있기 때문에, total인 one이 many의 역할을 수행하게 하였다.

이와 같은 과정을 거쳐 ERWIN을 통해 작성된 schema diagram은 다음과 같다.



## 2-1. entity

### \*product

information	attribute	domain
상품의 ID(primary key)	product_ID	Number
상품 종류	type	String
상품 package	package	String

### \*shipping\_info

information	attribute	domain
배송 번호(primary key)	tracking_num	Number
예정 배송일	schedule_date	Datetime
실제 배송일	arrival_date	Datetime
배송 회사명	comp_name	String
상품의 ID(foreign key)	product_ID	Number
고객의 ID(foreign key)	customer_ID	Number

### \*customer

information	attribute	domain
고객의 ID(primary key)	ID	Number

### \*monthly\_pay

information	attribute	domain
계좌 번호(primary key)	account_num	Number
고객의 ID(foreign key)	customer_ID	Number

### \*card\_info

information	attribute	domain
카드 번호(primary key)	card_num	Number
카드 종류(primary key)	card_type	String
고객의 ID(foreign key)	customer_ID	Number

### \*purchase

information	attribute	domain
날짜(primary key)	date	Datetime
고객의 ID(pk,fk)	customer_ID	Number
결제 금액	bill	Number

### \*warehouse

information	attribute	domain
창고의 ID(primary key)	W_ID	Number
상품의 ID(pk,fk)	product_ID	Number
지역	region	String
남은 수량	remain	Number
제조사 ID(foreign key)	M_ID	Number

\*store

information	attribute	domain
매장의 ID(primary key)	S_ID	Number
상품의 ID(pk,fk)	product_ID	Number
지역	region	String
남은 수량	remain	Number
제조사 ID(foreign key)	M_ID	Number

\*sales\_info

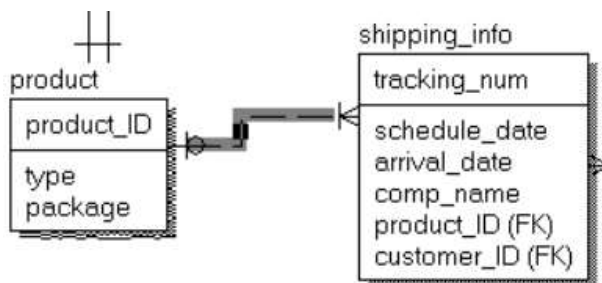
information	attribute	domain
날짜(primary key)	date	Datetime
매장의 ID(pk,fk)	S_ID	Number
상품의 ID(pk,fk)	product_ID	Number
판매 수량	amount	Number

\*manufacturer

information	attribute	domain
제조사 ID(primary key)	M_ID	Number
제조사 이름	name	String

## 2-2. relationship

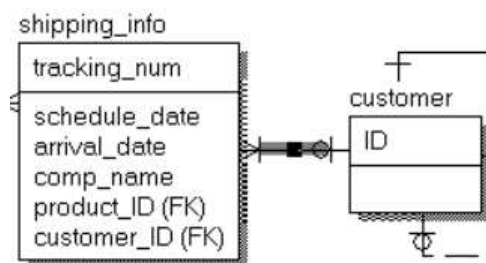
\*shipping



product to shipping\_info

cardinality: zero-or-one to one-or-more

\*arrival

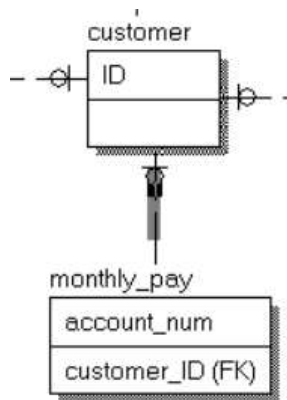


customer to shipping\_info

cardinality: zero-or-one to one-or-more



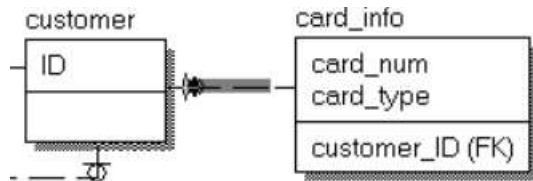
\*contract



customer to monthky\_pay

cardinality: zero-or-one to exactly one

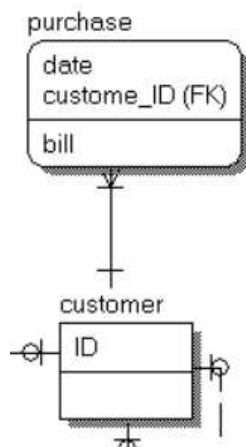
\*online\_pay



customer to card\_info

cardinality: zero-or-one to exactly one

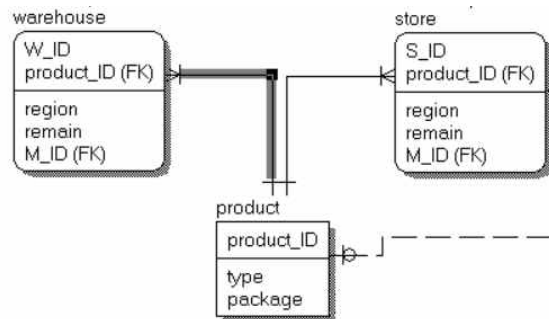
\*C\_purchase(identifying relation)



customer to purchase

cardinality: one to one or more

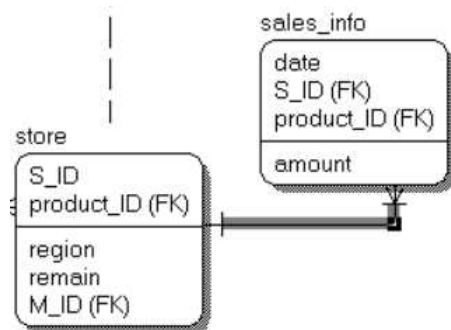
\*product\_w/product\_s(identifying relation)



product to warehouse/store

cardinality: one to one or more

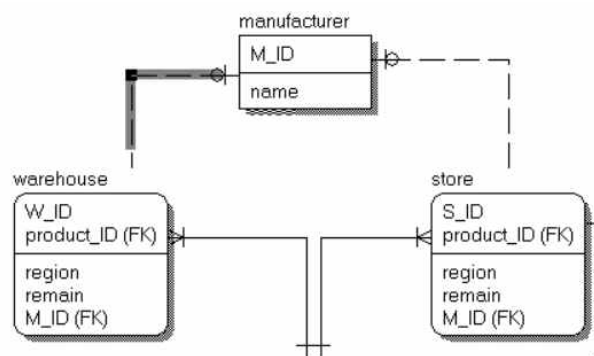
\*sales(identifying relation)



store to sales\_info

cardinality: one to one or more

\*w\_inventory/s\_inventory



manufacturer to warehouse/store

cardinality: zero or one to exactly one