Database System

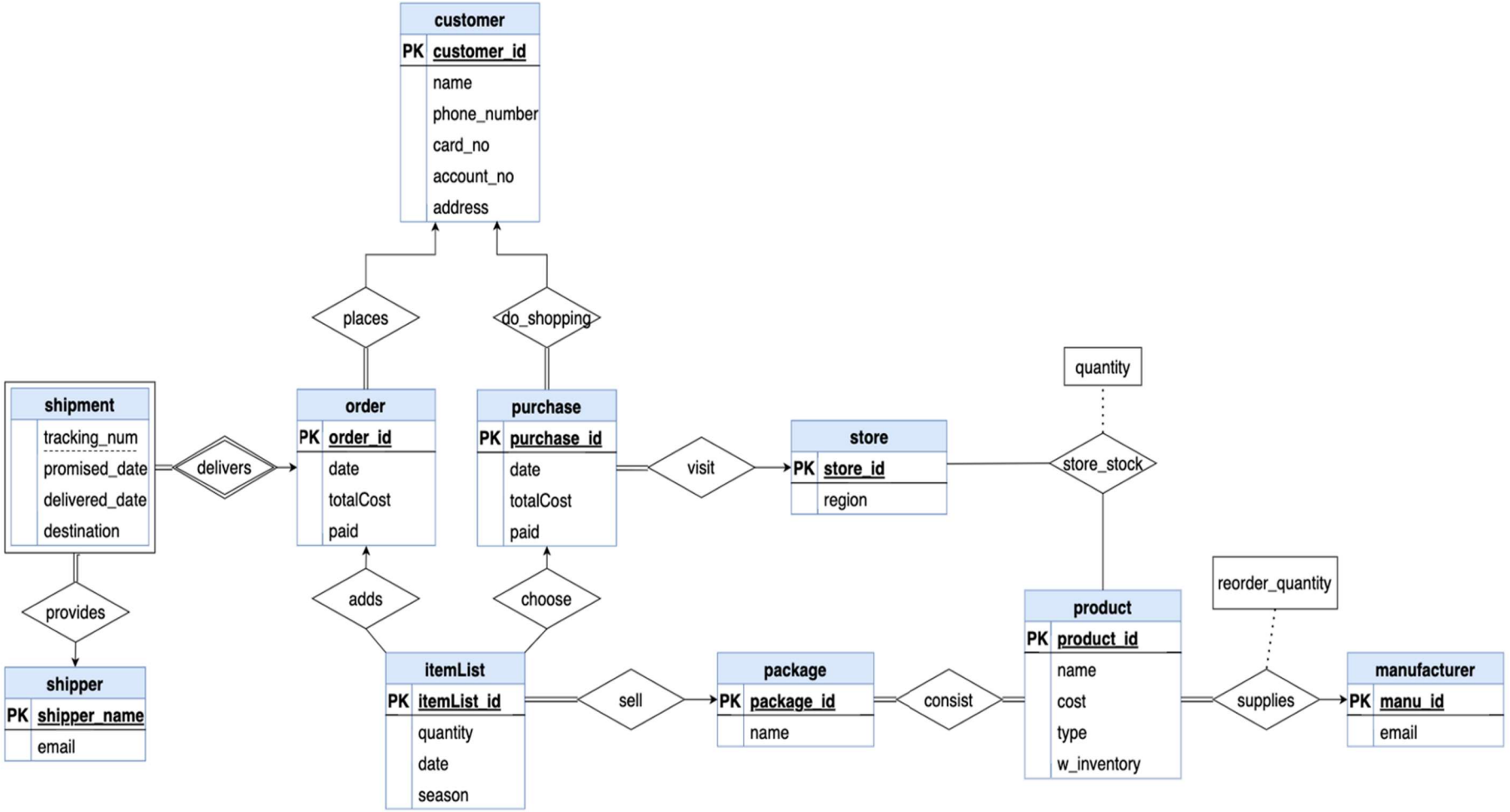
CSE4110-01

Project 1

학번: 20160169

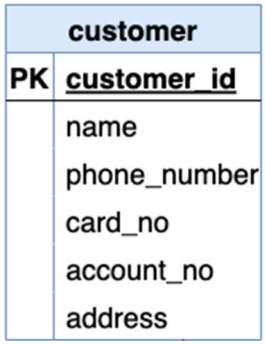
이름: 박성환

1. E-R Diagram


customer
PK customer_id
name
phone_number
card_no
account_no
address
places
do_shopping
shipment
tracking_num
promised_date
delivered_date
destination
delivers
order
PK order_id
date
totalCost
paid
purchase
PK purchase_id
date
totalCost
paid
visit
store
PK store_id
region
quantity
store_stock
provides
adds
choose
shipper
PK shipper_name
email
itemList
PK itemList_id
quantity
date
season
sell
package
PK package_id
name
consist
product
PK product_id
name
cost
type
w_inventory
reorder_quantity
supplies
manufacturer
PK manu_id
email

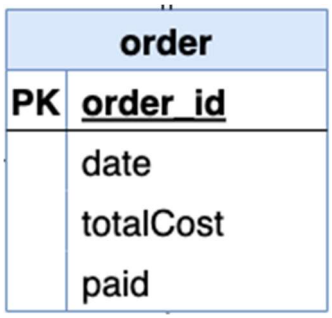
A. Entity

i. customer



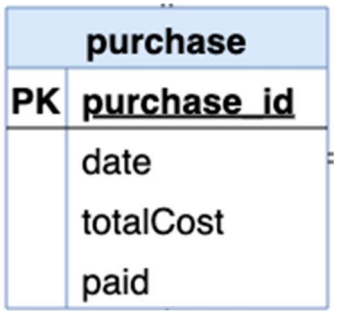
- 고객 정보를 저장한다. customer_id가 primary key이다.
- card_no, account_no는 null 값이 저장될 수 있다. account_no가 null 이 아닌 고객은 contract가 있는 고객이고 card_no가 있는 고객은 온라인 구매 경험이 있는 고객이다.

ii. order



- 온라인 주문 정보를 저장한다. order_id가 primary key이다.
- 주문 날짜, 총 주문 금액을 저장하고 paid는 고객이 계산을 했는지가 저장된다. 계산을 하지 않은 경우에는 달마다 bill로 청구된다.

iii. purchase



- 오프라인 구매에 대한 정보를 저장한다. purchase_id가 primary key이다. 저장하는 정보는 order entity와 동일하다.

iv. shipment

- 배송 정보를 저장한다.
- Weak entity로 order entity가 identifying entity 역할을 한다. 배송은 온라인 주문 없이는 존재할 수 없기 때문이다.
- 배송 추적 번호, 배송 희망 날짜, 실제 배송 날짜, 배송 목적지가 저장된다.

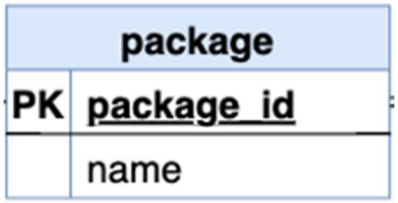
v. shipper

- 배송 업체를 저장한다. Shipper_name으로 배송 업체가 구분 가능하기 때문에 shipper_name이 primary key가 된다.
- 배송 업체의 연락처가 저장된다.

vi. itemList

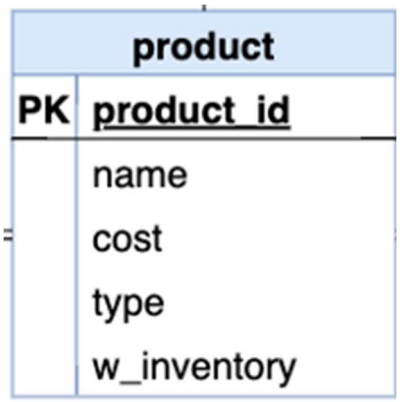
- order와 purchase entity의 각 주문 및 계산에 해당하는 물품 정보들을 저장한다. 예를 들어 order entity의 한 온라인 주문에 해당하는 주문서에서 각각의 물품 목록을 의미한다. itemList_id가 primary key 역할을 한다.
- 각 물품의 주문 수량, 날짜, season에 대한 정보를 저장한다.

vii. package



- 상품 패키지에 대한 정보를 저장한다. package_id가 primary key 역할을 한다.
- 패키지 이름을 추가로 저장한다.

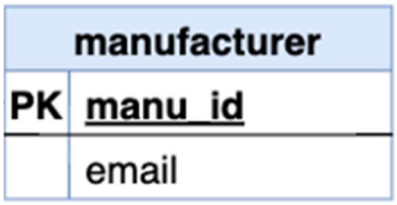
viii. product



- 상품 정보를 저장한다. product_id가 primary key 역할을 한다.

- 상품 이름, 상품 가격, 상품 종류를 저장하고 w_inventory는 warehouse에 있는 각 상품의 수량을 저장한다.

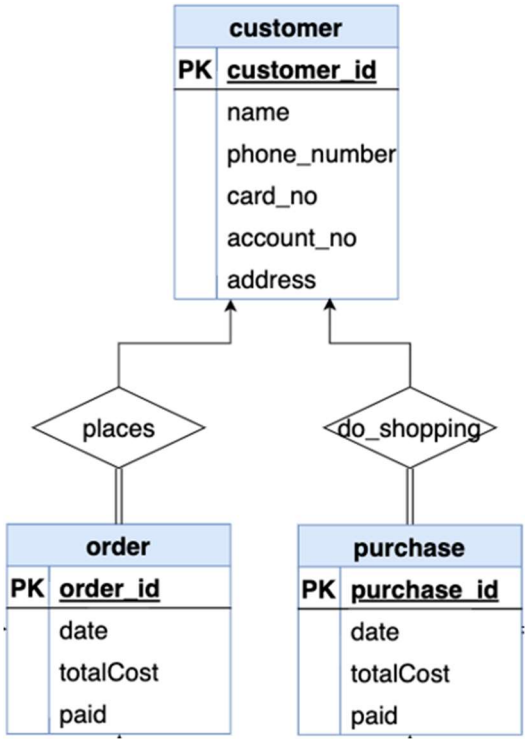
ix. manufacturer



- Manufacturer에 대한 정보를 저장한다. manu_id가 primary key 역할을 한다.
- Manufacturer의 연락처를 저장한다.

B. Relationship

i. places & do_shopping

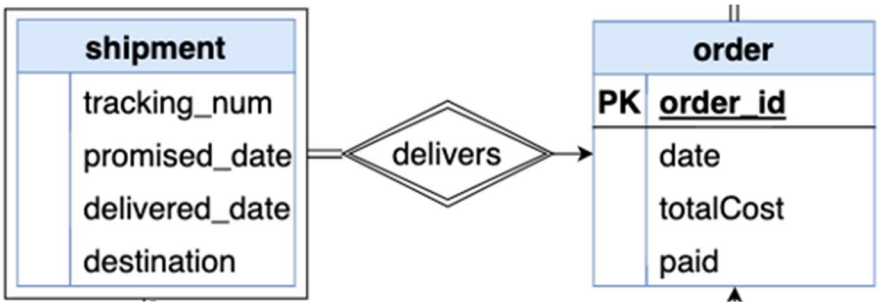


- order – customer, purchase – customer는 모두 고객과 고객의 주문을 의미한다. 온라인, 오프라인의 차이만 있다.
- customer - order는 one to many 관계이다. 모든 order는 customer에 하나씩 대응하고 customer는 order에 대응하지 않을 수도 있고 여러

order에 대응할 수 있기 때문이다. 모든 order가 customer에 대응해야 하므로 total participation이다.

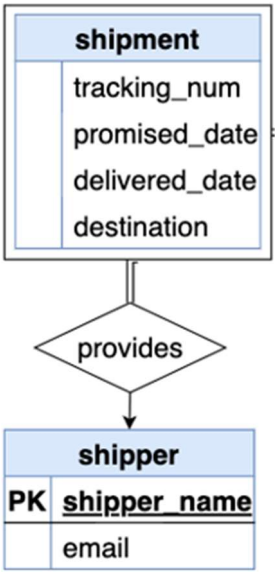
- purchase – customer 또한 order – customer와 마찬가지이다.

ii. delivers



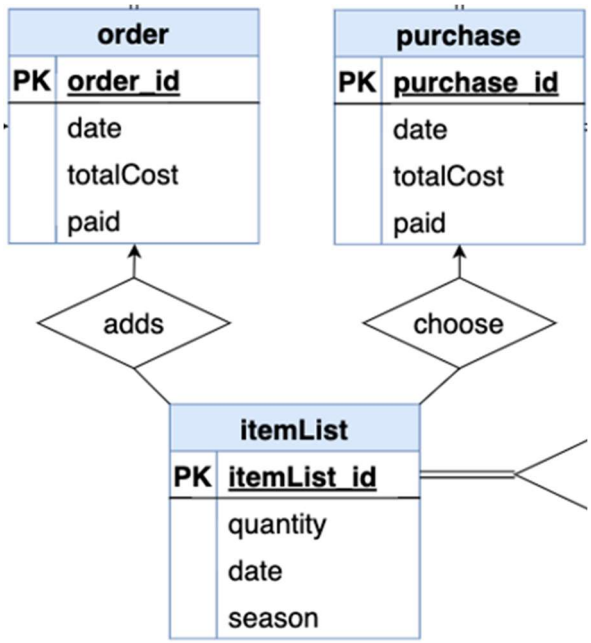
- weak entity set인 shipment가 identifying set인 order에 의존한다. shipment는 order 없이 존재할 수 없기 때문이다.
- 하나의 shipment가 여러 order를 담당할 수 있고 order는 shipment에 한 번씩만 참여하므로 order – shipment는 one – to – many 관계이다.

iii. provides



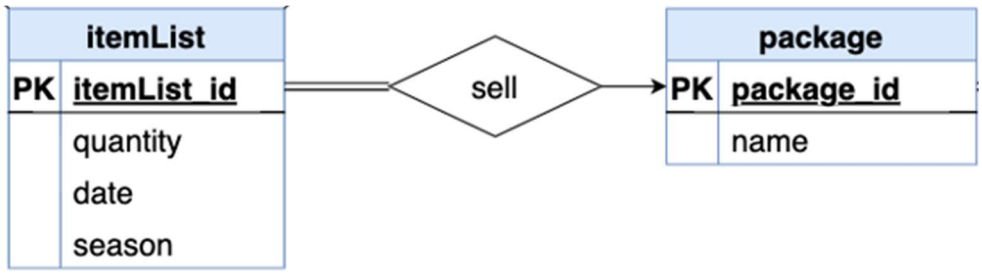
- shipment는 언제나 shipper를 필요로 하기 때문에 total participation이다. 동시에 shipper가 여러 shipment를 담당할 수 있기 때문에 shipment – shipper는 many – to – one 관계이다.

iv. adds, choose



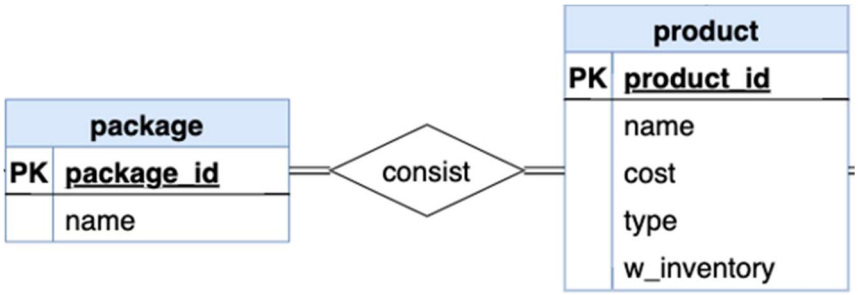
- adds, choose는 itemList로 하여금 온, 오프라인 상관 없이 모든 주문에 해당하는 물품 들을 저장할 수 있도록 한다.
- 각각의 order와 purchase가 여러 개의 itemList에 대응할 수 있으므로 order(purchase) – itemList 관계는 one – to many이다. 모든 itemList가 order 또는 purchase 한 쪽에 모두 대응할 필요는 없으므로 total participation이 아니다.

v. sell



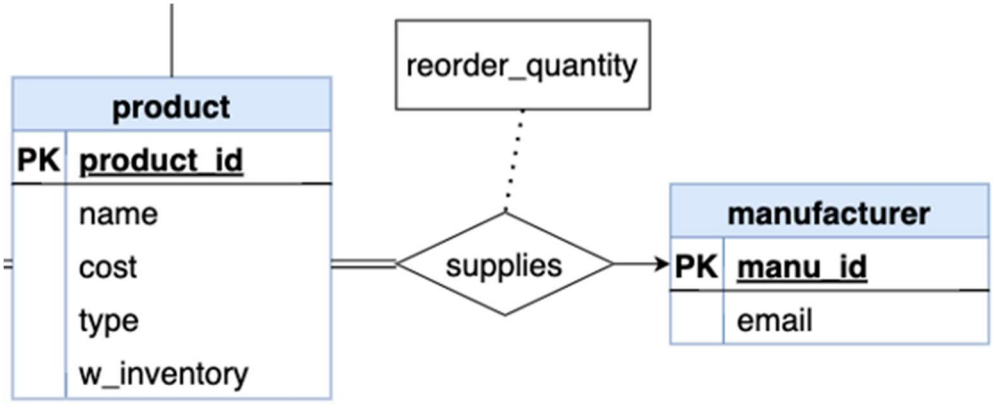
- 모든 itemList는 package에 대응해야 하므로 total participation이다. 반면 모든 package가 itemList에 대응할 필요는 없으므로 item List – package는 many – to – one 관계이다.

vi. consist



- Package와 product 모두 각자 주문이 가능한데 데이터베이스 관리의 편의를 위해 단품으로만 판매되는 product도 하나의 package로 취급하도록 하였다. 따라서 itemList에는 product_id가 필요하지 않고 package_id만 필요하다. 대신 product와 package의 관계를 나타내야 하므로 consist relationship을 만들었다.
- Package과 product는 각각 여러 개씩 대응될 수 있으므로 many – to many 관계이다. 모든 product는 최소 하나의 package에 대응하고 package는 product 없이 존재할 수 없으므로 양쪽 다 total participation이다.

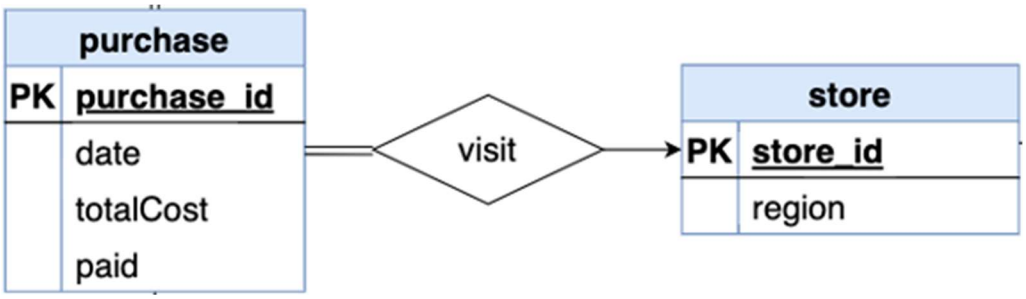
vii. supplies



- 모든 product는 manufacturer가 필요하므로 total participation이다. 동시에 하나의 manufacturer가 여러 product를 공급할 수 있으므로 product – manufacturer는 many – to – one 관계이다.
- reorder_quantity를 supplies relationship에 추가하였다. Reorder_quantity가 0이면 재고 신청이 필요 없는 상태이고 reorder_quantity가 양의 정수이면 manufacturer에게 재고 수급이 신

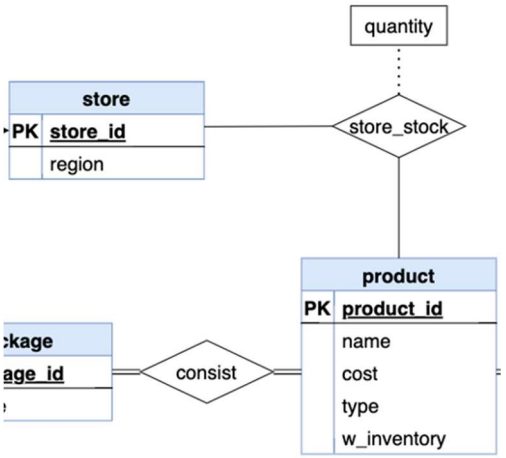
청된 상태이다.

viii. visit



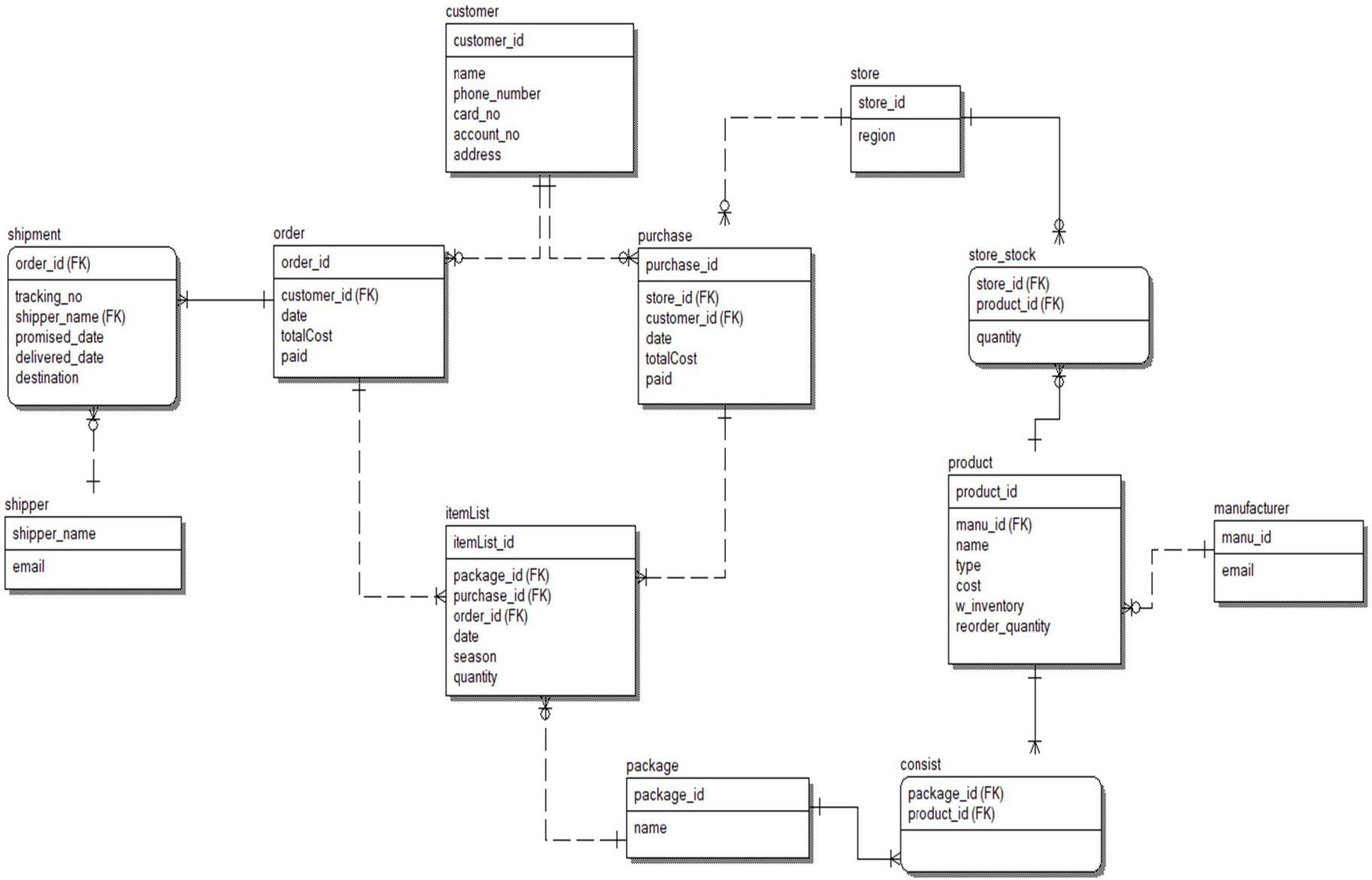
- 모든 오프라인 구매는 하나의 매장에 대응해야 하므로 total participation이다. 동시에 하나의 매장에서 여러 purchase가 발생할 수 있으므로 purchase – store는 many – to – one 관계이다.

ix. store_stock

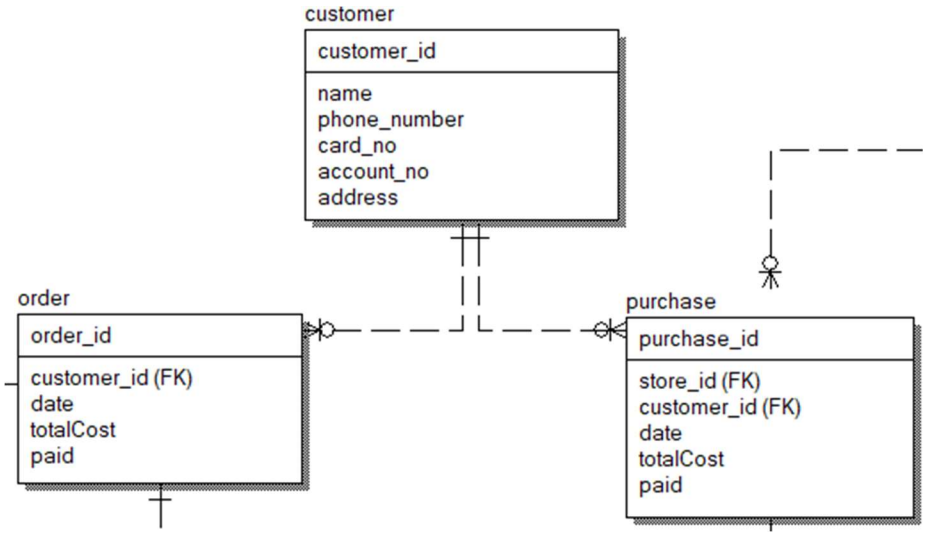


- store_stock은 매장 별 재고 현황을 나타내기 위해 만들었다. quantity attribute를 추가하여 재고 수를 표현한다.
- Store와 product는 서로 다수가 대응할 수 있으므로 many – to – many 관계이다.

2. Schema diagram


customer
customer_id
name
phone_number
card_no
account_no
address
store
store_id
region
shipment
order_id (FK)
tracking_no
shipper_name (FK)
promised_date
delivered_date
destination
order
order_id
customer_id (FK)
date
totalCost
paid
purchase
purchase_id
store_id (FK)
customer_id (FK)
date
totalCost
paid
store_stock
store_id (FK)
product_id (FK)
quantity
shipper
shipper_name
email
itemList
itemList_id
package_id (FK)
purchase_id (FK)
order_id (FK)
date
season
quantity
product
product_id
manu_id (FK)
name
type
cost
w_inventory
reorder_quantity
manufacturer
manu_id
email
package
package_id
name
consist
package_id (FK)
product_id (FK)

A. customer, order, purchase



i. customer는 Strong entity set이므로 ER diagram의 entity를 그대로 사용한다.

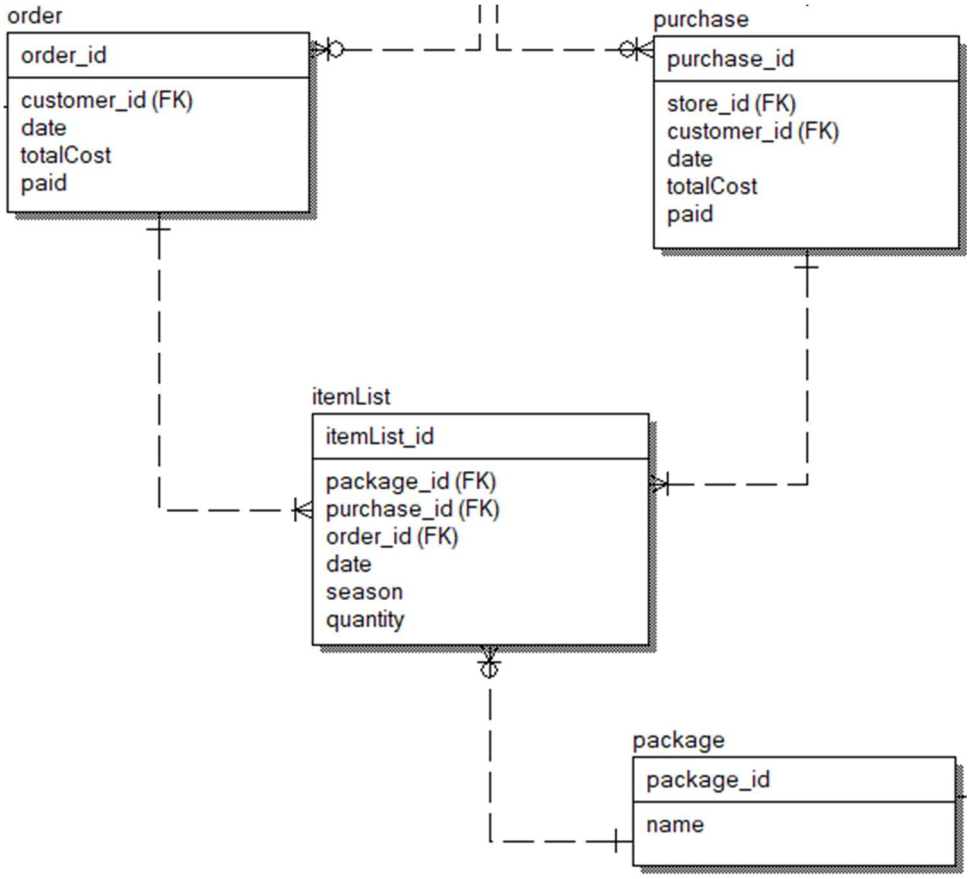
ii. order는 Customer와 many – to – one 관계이므로 customer_id를 foreign key로 추가하였다.

iii. purchase는 customer와 many – to – one 관계이므로 customer_id를 foreign key로 추가하였다.

iv. customer, order, purchase를 join하여 "Find the customer who has bought the most (by price) in the past year", "Generate the bill for each customer for the past month." 쿼리를 해결할 수 있다.

v. order는 항상 customer에 대응해야 하고 order가 없는 customer가 존재할 수 있으므로 customer – order의 cardinality는 one-to-zero-one-or-more이다. 이는 customer – purchase도 마찬가지다.

B. order, purchase, itemList, package



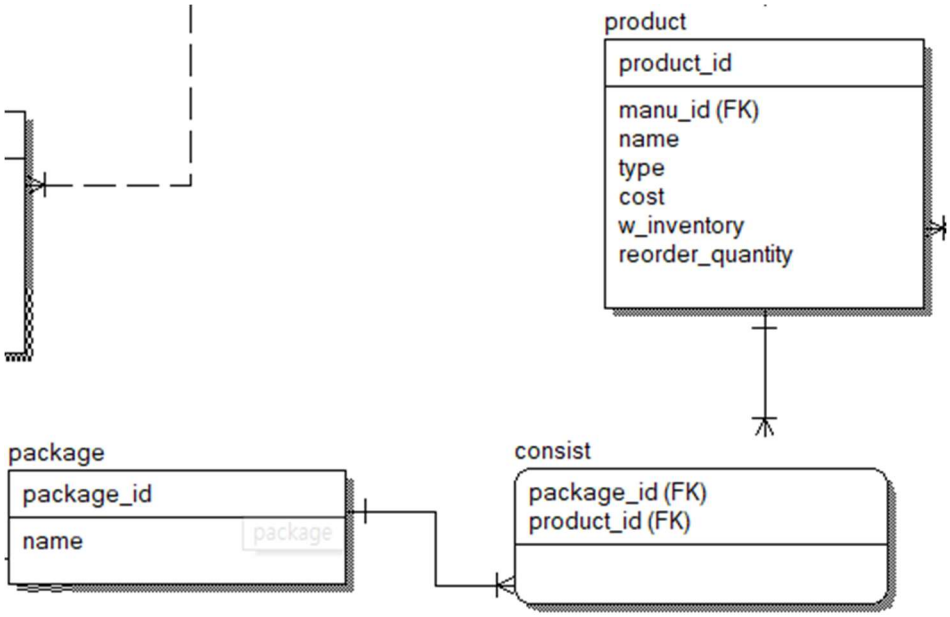
i. itemList는 order, purchase와 각각 many – to – one 관계이므로 order와 purchase의 primary key를 foreign key로 추가하였다.

ii. package와도 many – to – one 관계이므로 package의 primary key를 foreign key로 추가하였다.

iii. 모든 order는 itemList를 필요로 하기 때문에 order – itemList의 cardinality는 one-to-one-or-more이다. 이는 purchase – itemList 관계도 마찬가지다.

iv. 모든 itemList는 package와 대응해야하지만 모든 package는 itemList에 대응하지 않아도 되기 때문에 package – itemList의 cardinality는 one-to-zero-one-or-more 이다.

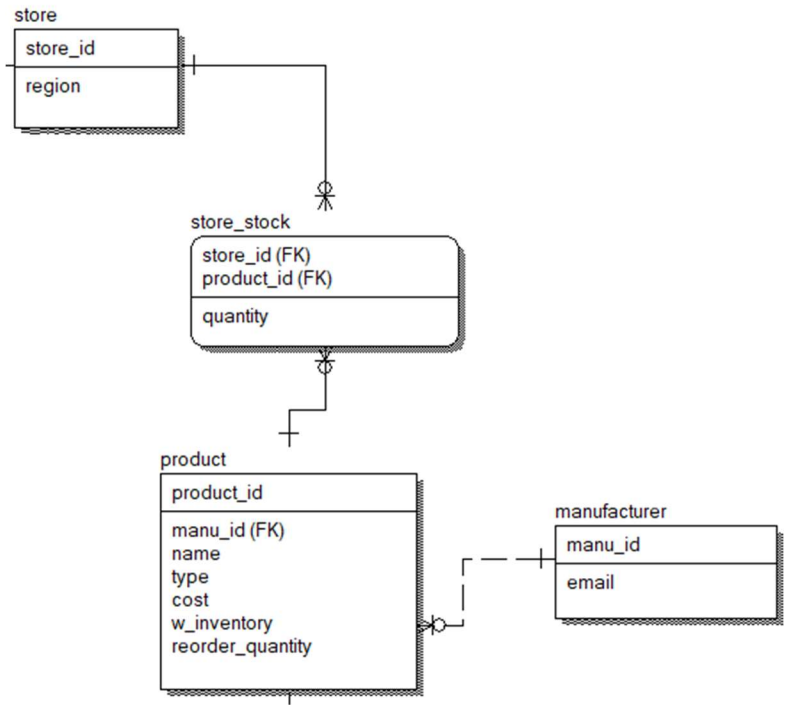
C. product, consist, package



i. consist는 package와 product의 many – to – many 관계를 표현한 relation 이다. 따라서 product와 package의 primary key가 consist의 primary key 가 된다. 모든 package와 product가 consist를 통해 대응한다. 따라서 package – consist, product – consist의 cardinality는 one-to-one-or-more 이다.

ii. consist와 itemList를 join하여 "Find the top 2 products by dollar-amount sold in the past year", "Find the top 2 products by unit sales in the past year." 쿼리를 해결할 수 있다.

D. product, store, store_stock, manufacturer



i. product와 manufacturer가 many – to – one 관계이므로 ER diagram의 product entity에 manufacturer entity의 primary key를 추가하였다. 동시에 supplies relationship의 reorder_quantity 또한 product에 추가하였다.

ii. store는 strong entity set이므로 ER diagram의 store entity를 그대로 사용한다.

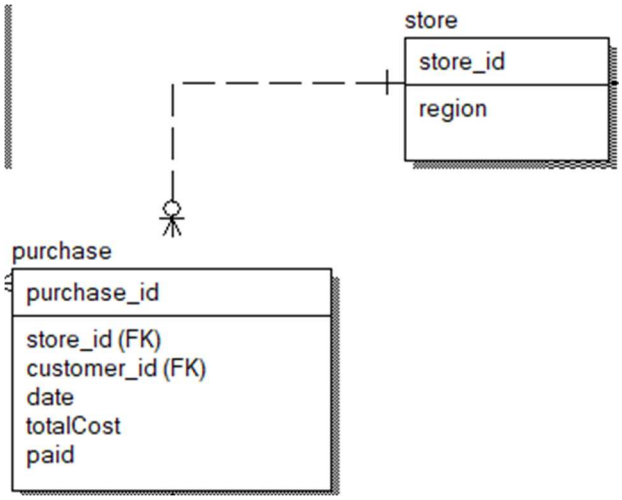
iii. store_stock은 store와 product의 many – to – many 관계를 나타내는 relation이므로 store와 product의 각각의 primary key를 primary key로 갖는다. 그리고 quantity attribute를 통해 재고 수량을 표현한다.

iv. 모든 product는 manufacturer를 필요로 하고 모든 manufacturer가 product에 대응할 필요는 없으므로 manufacturer – product의 cardinality는 one-to-zero-one-or-more이다.

v. store와 product는 store_stock에 각각 0 ~ n개까지 다양하게 대응할 수 있으므로 store – store_stock과 product – store_stock의 cardinality는 one-to-zero-one-or-more이다.

vi. store와 store_stock을 join하여 "Find those products that are out-of-stock at every store in California." 쿼리를 해결할 수 있다.

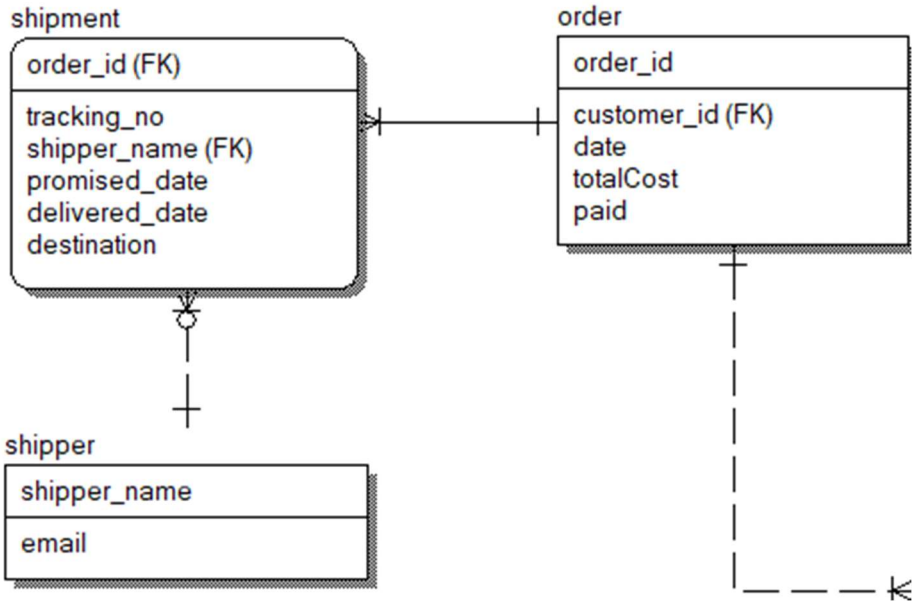
E. purchase, store



i. purchase – store가 many – to – one 관계이므로 store의 primary key를 purchase의 foreign key로 추가하였다.

ii. purchase는 하나의 store에만 대응하고 store는 purchase에 여러 개가 대응할 수도 있고 대응하지 않을 수도 있기 때문에 store – to – purchase의 cardinality는 one-to-zero-one-or-more이다.

F. shipment, shipper, order



i. shipment는 order의 weak entity set이므로 order_id를 primary key로 가진다. 하나의 shipment가 여러 order를 담당할 수는 있지만 하나의 order가 여러 shipment에 대응할 수 없기 때문에 shipment의 primary key로 order_id면 충분하다. 그리고 그에 따라 order – shipment의 cardinality는 one-to-one-or-more이다.

ii. shipment와 shipper는 many – to – one 관계이기 때문에 shipper의 primary key를 shipment의 foreign key로 추가하였다. 또 모든 shipment는 shipper를 필요로 하지만 모든 shipper는 shipment에 대응할 필요가 없으므로 shipper – shipment의 cardinality는 one-to-zero-one-or-more이다.

iii. shipment와 order, customer, itemList를 join하여 "• Assume the package shipped by USPS with tracking number 123456 is reported to have been destroyed in an accident. Find the contact information for the customer. Also, find the contents of that shipment and create a new shipment of replacement items." 쿼리를 해결할 수 있다.