

PROJECT1

학과:컴퓨터공학과 학년:3학년 학번:20211524 이름:김준우

1.

er-diagram에서 entity는 총 6개로

1)Shipments=(Shipment_id, package_name,ship_care,est_time,destination)

2)loc_and_time=(date,time,location,vehicle_id,cur_destination)

3)Bill=(Bill_id,price,bill_address)

4)pay=(payment_method,paid_at)

5)Customers=(customer_id,customer_name,customer_address)

6)services=(services_id,weight,timeliness,package_type)

순서대로 설명을 하면 1)은 배송되는 물품에 관한 entityset으로 Shipment_id가 primary key다. 각각 attribute를 보면 우선 package_name은 이름 그대로 배송되는 물품의 이름이다. 같은 이름의 물품이라도 배송될 때 운송장의 번호는 다르기 때문에 primary key가 아니다. 그 다음으로 ship_care는 shipper가 고려할 상황에 대한 attribute다. 과제에서 주어진 조건인 위험물품인지, 세관 신고서가 필요한지, 둘 다 아닌 일반 물품인지를 구분하는 attribute다. 들어갈 수 있는 값이 3개이므로 각 tuple을 구분할 수 없기 때문에 primary key가 아니다. est_time은 배송이 완료되는 시점을 예상하는 attribute다. 마지막으로 destination은 최종 배송지를 의미한다. 이는 customer의 주소와 다를 수 있다.

2)번은 shipment의 weakentity set으로 shipment가 존재 할 때만 존재하는 entityset이다. 이 둘의 relation은 tracking으로 shipper가 배송품이 어디 있는지 확인할 수 있는 entityset이다. 우선 weak entity set이기 때문에 primary key는 존재하지 않는다. date는 날짜, time은 시간, location은 위치, vehicle_id는 운송수단의 id, cur_destination은 해당 시점의 목적지를 의미하는 attribute다. 날짜와 시간순으로 정렬을 해서 해당 시점에 어디에 있는지 어떤 운송수단으로 이동하고 있는지 tracking할 수 있도록 해주는 entityset이다.

3)번은 bill_id를 primary key로 가지고 price와 bill_address를 attribute로 가진다. price는 청구되는 돈을 bill_address는 청구서가 배송될 주소를 의미한다. shipments와 one to many의 relationship으로 여러 배송물이 하나의 청구서에 포함되어 있다. 그리고 customer와는 many to one으로 relationship set이 생긴다. 이는 pay로 다음번호에 설명 할 것이다. 하나의 청구서에 여러 물건이 포함될 수는 있지만 pay에 있는 attribute에 의해 한명에게 여러개의 청구서가 생길 수 있기 때문에 many to one의 관계다.

4)번 pay는 customer와 bill의 Relationship set으로 청구서는 언제 돈을 지불하는지 (선불 or 후불) 어떤 방식으로 돈을 지불하는지가 customer와 bill이 관계가 생길 때 존재하는 attribute기 때문에 relationship이 된 것이다. 각각 순서대로 paid_at과 payment_method다.

5)번 customers는 고객에 대한 정보로 customer_id를 primary key로 가지고 이름과 주소를 저장하는 attribute인 customer_name과 customer_address가 있다. shipments와 one to many의 관계를 가지고 있다. 한 명의 고객이 여러 물건을 주문할 수 있기 때문이다.

6)번 마지막으로 service는 weight timeliness package_type에 따라 달라지는 서비스에 대한 entityset이다. service_id를 primary key로 가진다. 나머지 attribute는 같은 값을 가질 수 있기 때문

에 service의 구분을 service_id로 구분했다. 이는 또한 shipment와 one to many의 관계를 가진다. 다른 물품이더라도 같은 service를 공유할 수 있기 때문이다.

2.

이제 모든 entity들의 relation에 대해 알아보자.

- 1) shipment와 customer의 관계인 order
- 2) customer와 bill의 관계인 pay 그리고 relationship set
- 3) shipment와 bill의 관계인 charge
- 4) shipments와 service의 관계인 service_type
- 5) shipments와 loc_and_time의 관계인 tracking

이렇게 5가지의 relation이 존재한다.

1)번은 order 즉, 주문의 관계로 한 명의 고객이 여러개의 물품을 구매해 배송품이 여러개인 경우가 있기 때문에 customer와 shipment는 one to many의 관계다. 또한 모든 등록된 고객이 배송품을 가지고 있지 않지만, 모든 배송품은 주문한 고객이 존재해야한다. 따라서 shipments는 total이다.

2)번은 pay 즉, 돈을 지불하는 문제에 관한 relation으로 고객이 돈을 지불하는 시점과 돈을 지불하는 방식을 정해야 해서 attribute를 가지는 relationship set을 만들어서 relation을 만들어 줬고 한 명의 고객에게 여러개의 청구서를 청구할 수 있기 때문에 one to many의 관계로 정의했다. 이 역시 물건을 주문한 고객에게만 청구서가생기기 때문에 주문을 하지 않은 고객이 존재하기 때문에 bill은 total이지만 customer는 total이 아니다.

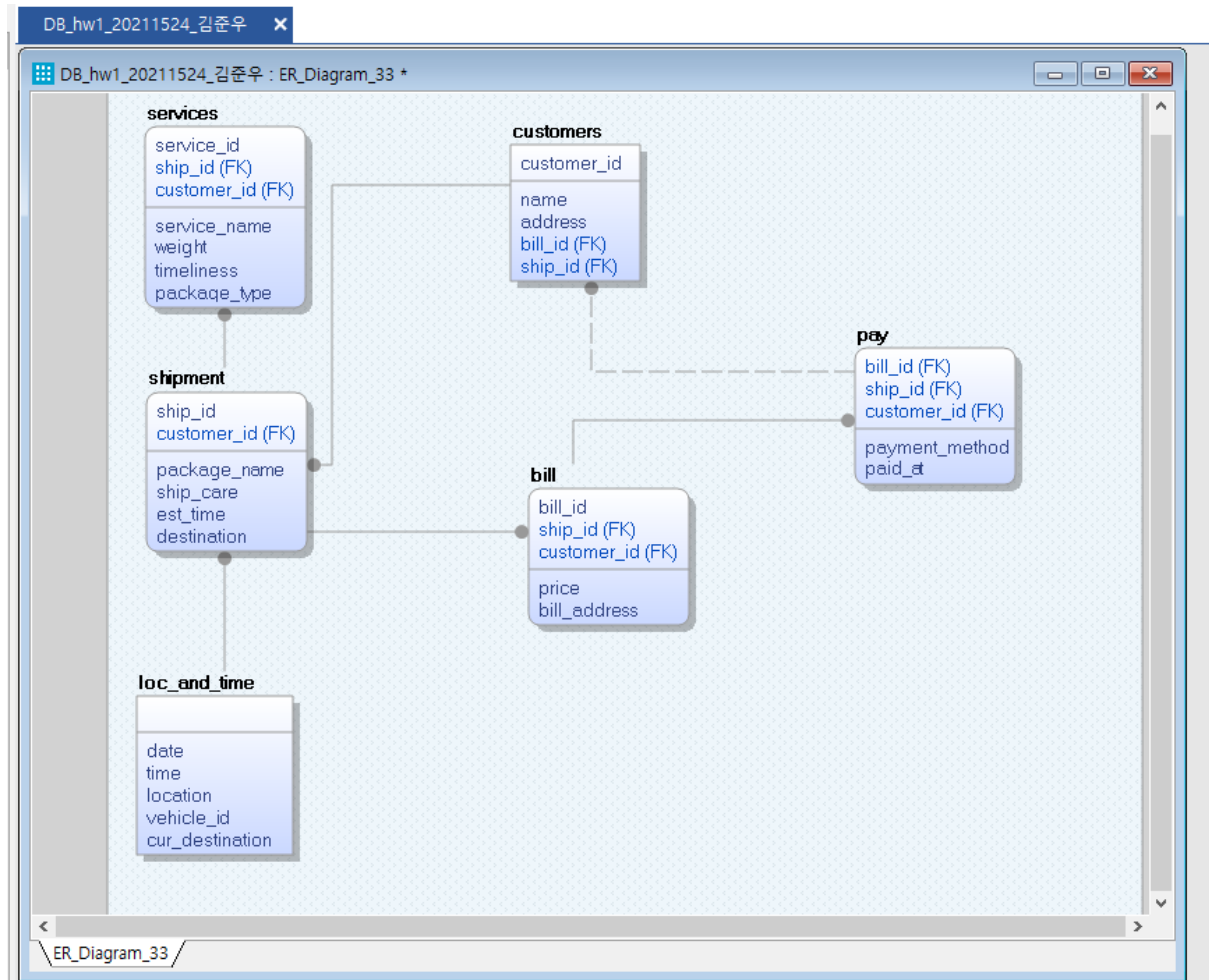
3)번은 shipment에 의해 청구서가 생기기 때문에 charge라는 relation을 만들었고 하나의 청구서에 여러 물건이 들어갈 수 있기 때문에 bill 과 shipment는 one to many의 관계로 정의했다. 이 relationship은 모든 물건에 bill이 생기고 bill에 모든 물건이 포함되어야 하기 때문에 둘 다 total이다.

4)번인 service_type은 각각의 배송품에 하나의 service가 정해진다. 그러나 서비스의 종류가 모든 물건에 따라 다른 것이 아니라 특정 범위 안의 무게, timeliness 그리고 포장방식으로 중복이 충분히 일어날 수 있는 것으로 정해지기 때문에 shipments와 service는 many to one의 관계로 정의내릴 수 있다.

5)마지막인 tracking은 weak entity set인 loc_and_time과 shipments의 관계다. 필수적으로 shipment가 있어야 tracking을 할 수 있기 때문에 weakentity set으로 정의내릴 수 있다.

3.

스키마 다이어그램에도 1번과 같이 총 6개의 entity set이 있다.



erwin으로 완성한 erd의 모습이다. 우선 loc_and_time은 weak entity set이므로 primary key가 없는것을 확인할 수 있다.

shipment는 customer와 many to one의 관계를 가져서 customer의 primary key인 customer_id를 foreign key로 가지고 있음을 확인 할 수 있다.

customer entity set은 다른 entity set과의 관계에서 one의 위치이므로 foreign key가 없다.

다음으로 services는 shipments와 one to many의 관계를 가지는데 service를 구분하는데 물건의 종류를 알아야 하므로 shipment의 primary key인 ship_id와 customer_id를 foreign key로 가져와 primary key로 이용하는 것을 확인할 수 있다.

다음으로 bill은 shipment에서 물건 종류와 어떤 고객에게 발송되어야 하는지를 알아야하기 때문에 ship_id와 customer_id를 foreign key로 가져와 primary key로 이용하는 모습을 확인할 수 있다.

마지막으로 pay는 원래의 primary key가 존재하지 않는 relationship set인데 구분을 하기 위해서 customer, bill의 primary key인 bill_id, ship_id, customer_id를 가져와 primary key로 사용한다.