

1. 진행 순서

Electronics Vendor compnay 의 DBA 로서 설계한 E-R Schema Diagram 에 대해 살펴보기에 앞서, 보고서 진행 순서를 설명하도록 한다. 우선, 설계의 초기 단계에서 어떤 요구사항을 뽑아내었는지를 설명한 다음, 뽑아낸 요구사항들을 어떻게 entity 로 나누고, 속성을 배치하였는지를 설명한 뒤, 최종적으로 entity 들 간에 존재하는 relationship 들을 알아보도록 한다. 끝으로는 primary key를 알아보기 쉽도록 Relational schema에 대한 간략한 요약을 포함하도록 한다.

2. E-R Model - 사용자 요구사항 찾아내기

주어진 정보들을 바탕으로 사용자의 입장에서 필요한 최소한의 요구사항을 찾아낸 결과는 다음과 같다.

1. 제품 : 제품명, 가격, 제품유형, 제조업체, 패키지제품
2. 고객 : 이름, 계약번호, 주소 (온라인 고객 : 카드 정보, 휴대폰 번호)
3. 배송 : 배송업체, 배송추적번호, 발송일, 도착일
4. 재고 : 창고 재고, 가게 재고 (재주문 : 날짜, 품목, 수량, 도착여부)
5. 가게 : 이름, 번호, 지역명
6. 판매 : 제품명, 수량, 가격, 날짜, 지불방식

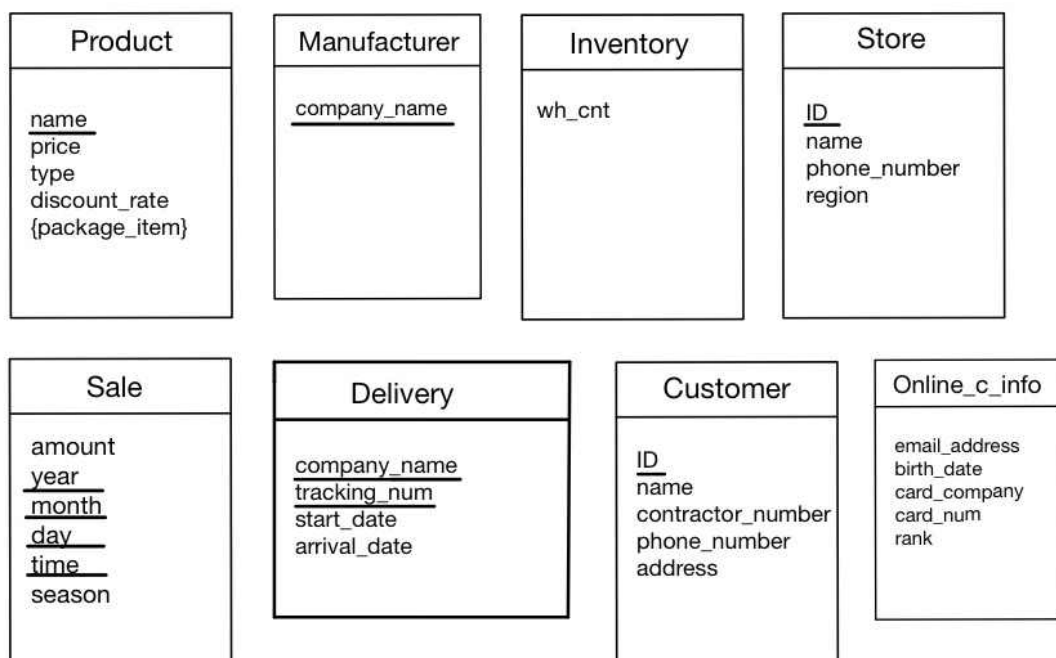
이에 본인의 전자제품 구매 경험을 살려, 제품에 대한 할인율을 표시할 수 있도록 할인 속성을 추가하고, 온라인 구매 고객의 구매 횟수에 따른 등급 속성을 추가해 추후에 혜택을 줄 수 있도록 했다. 이들을 종합하여, 요구사항을 최종적으로 정리하면 다음과 같다. 변경사항은 색상을 달리해 표시하였다.

1. 제품 : 제품명, 가격, 제품유형, 제조업체, 패키지제품, **할인율**
2. 고객 : 이름, 계약번호, 주소, 휴대폰번호 (온라인 고객 : 카드 정보, **출생일, 등급**)
3. 배송 : 배송업체, 배송추적번호, 발송일, 도착일
4. 재고 : 창고 재고, 가게 재고 (재주문 : 날짜, 품목, 수량, **도착날짜**)
5. 가게 : 이름, 번호, 지역명

6. 판매 : 제품명, 수량, 가격, 날짜

3. E-R Model - Entity sets

추려낸 사용자 요구사항을 바탕으로, Entity set 을 만들면 총 8개(제품, 제조사, 재고, 가게, 판매, 배송, 고객, 온라인 고객 정보)로 나눌 수 있다. 기존에 추려낸 6개의 항목에 제조사와, 온라인 고객 정보 항목이 추가되었다. 각각의 Entity set은 다음과 같다. 이들 간 관계에 대해서는 곧 이어 설명하도록 한다.



(1) 제품 (Product)

모든 제품에 대한 id를 부여하는 것은 불필요한 다른 속성들의 중복을 너무 많이 발생시켜, 제품명으로 모든 제품을 구분하도록 했다. 이외의 속성으로는 제품 가격, 제품 유형, 할인을, 패키지 항목을 두었다. 다른 속성은 모두 single, simple 값을 갖는데 반해서, **패키지 항목은 multivalue 속성으로 두었다**. 이는 패키지 항목이 생겼을 때, 제품 Entity에 그에 대한 패키지 명을 부여할 수 있도록 할 뿐만 아니라, 이외의 가격, 유형, 할인을 정보도 저장 가능하며, 패키지에 포함되는 제품명만 나열하면 되기에 저장이 용이하다고 파악했다.

(2) 제조사 (Manufacturer)

제품의 제조사를 또 다른 Entity set으로 추가해서 구분한 이유는, 특정 제조사가 많은 수많은 제품들이 같은 속성을 공유하게 될텐데, 이것은 정보의 중복을 유발하므로 따로 Entity set

으로 분리하였다. Entity 들은 제조사명을 통해 독립적 식별이 가능하다.

(3) 재고 (Inventory)

재고 Entity set 에서는 제품명에 대한 창고에 존재하는 재고수를 표시할 수 있도록 했다. 물론 재고 각각에 id를 부여하는 방법도 있겠지만, 이미 제품명으로 모든 제품을 식별할 수 있기 때문에 별도로 id를 만들지는 않았다. 가게가 갖는 재고는 가게 정보를 필요로 하기 때문에 다른 set에 분리해 저장하도록 하였다.

(4) 가게 (Store)

계열사 등으로 인해 가게명이 겹칠 경우를 고려하여, 모든 가게를 독립적으로 식별하기 위해 id를 부여하였다. 이에 더해 가게를 구성하기 위한 속성으로 가게 전화번호와 지역명을 추가했다.

(5) 판매 (Sale)

판매 Entity set 은 제품명과, 구매자 id, 구매날짜를 통해 모든 entity들을 식별할 수 있도록 했다. 이에 더해 구매 수량과 계절을 나타내는 속성을 저장하도록 했다.

(6) 배송 (Delivery)

배송은 온라인 고객의 주문을 배송하기 위해 만들어진 Entity set 이며, 배송업체와 배송추적 번호를 통해 모든 배송 목록을 독립적으로 식별할 수 있다. 이에 더해서 배송 시작일과 배송 도착일을 저장하도록 했다. 만약 도착일이 null 값을 갖는다면, 이에 대해 재배송을 진행하도록 구상하였다.

(7) 고객 (Customer)

고객은 이름은 중복가능성이 존재하므로 모두를 독립적으로 식별해낼 수 없기에, id를 부여에 식별할 수 있도록 했다. 추가적 속성으로는 이름, 계약을 했을 경우 계약번호, 휴대폰번호, 주소를 저장하도록 했다. 주소와 휴대폰번호의 경우 온라인 주문 고객의 배송을 위해 필수적이었다.

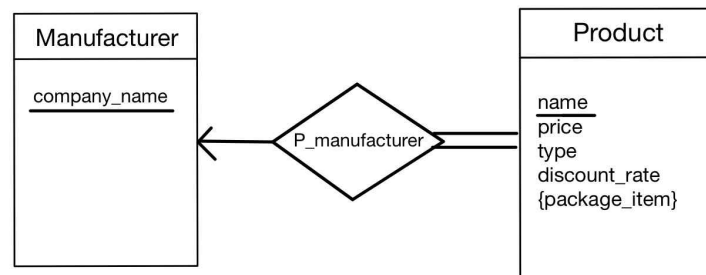
(8) 온라인 고객 정보 (Online_c_info)

온라인 고객의 추가적 정보의 경우, 모든 고객이 온라인 고객이 아니기 때문에 따로 분리해서 Entity set을 만들어 null 값이 많아짐을 방지하고자 했다. 이들은 기존의 고객 Entity set 의 주 키인 id를 통해 독립적 식별이 가능하다. 온라인 고객 정보에는 이메일 주소, 출생일, 카드회사, 카드번호, 등급 에 대한 속성을 갖도록 했다.

4. E-R Model - Relationship sets

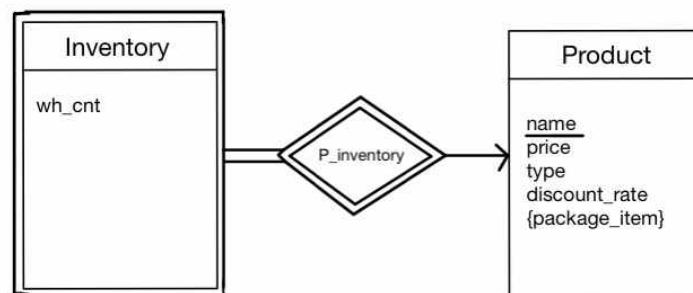
위에서 만든 Entity sets 들을 바탕으로 이들 간의 관계를 모두 표시하면 총 8개의 Relationship sets 을 구성할 수 있었다. 각각의 관계에 대해 살펴보면 다음과 같다.

(1) 제품 - 제조사 관계 (P_manufacturer)



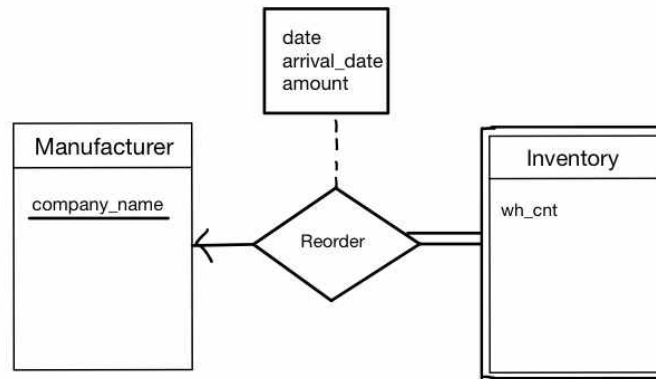
제조사가 생산하는 제품이 여러 가지가 있기 때문에, 제조사를 제품 Entity set 내의 속성으로 두면 정보의 중복이 발생한다. 이를 방지하고자, 제조사를 별도의 Entity set 으로 분리시키고, 제조사와 제품간 One to Many 관계를 이용해서 제품의 제조사를 표현할 수 있도록 했다. 이때, 모든 제품은 제조사를 갖게 되므로 Many 측인 제품 Entity set 은 total의 관계를 갖게 된다.

(2) 제품 - 재고 관계 (P_inventory)



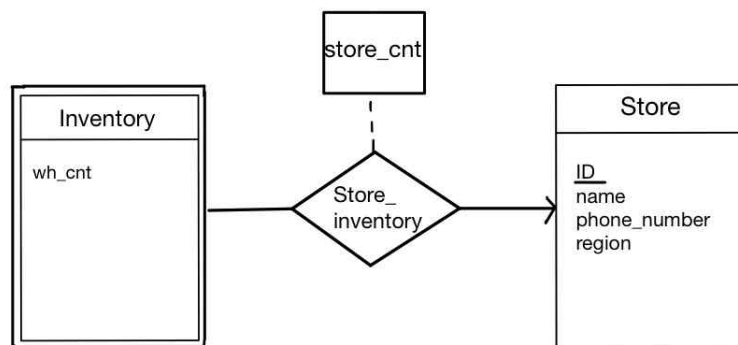
재고 Entity set 은 창고에 존재하는 재고의 개수를 나타내며, 재고의 개수는 제품명과 함께 있을 때 의미를 갖는다. 따라서, 제품과 재고 간에는 One to Many and total 의 관계가 성립하고, 모든 재고 수는 제품명이 있음으로서 비로소 독립적으로 식별이 가능해진다.

(3) 재고 - 제조사 관계 (Reorder)



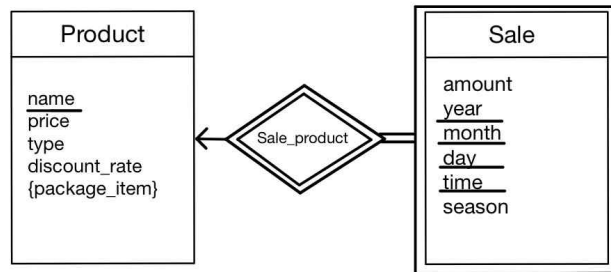
재고가 부족하게 되면, 제조사에 재주문을 넣어야 한다. 이 액션을 나타내기 위해 재고와 제조사간에 Reorder 라는 관계를 만들었으며, 재고 Entity set 은 부족한 제품명에 대해 제조사에 주문을 넣는다. 만약 재주문 관계가 있다면, 한 제조사에 여러개의 제품명에 대한 주문이 요청가능하며, 모든 재고는 제조사를 갖게되므로 이 둘간의 관계는 One to Many and total로 나타낼 수 있다. 또한, 이 관계는 관계 자체의 속성으로 주문 날짜, 주문 개수, 도착 날짜를 포함한다.

(4) 재고 - 가게 관계 (Store_inventory)



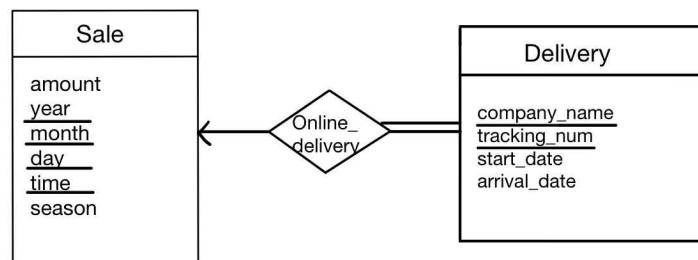
Inventory Entity set 은 창고 재고만을 나타내기 때문에, 가게의 재고량은 별도로 표시를 해야 한다. 이를 나타내기 위해 사용한 관계가 Store_inventory Relationship set 이다. 이 관계는 관계가 있다면, 한 가게에 대해 여러 가지 제품명의 재고를 나타낼 수 있으며, 재고 개수를 나타내기 위해 관계의 속성인 store_cnt 를 사용하였다.

(5) 제품 - 판매 관계 (Sale_product)



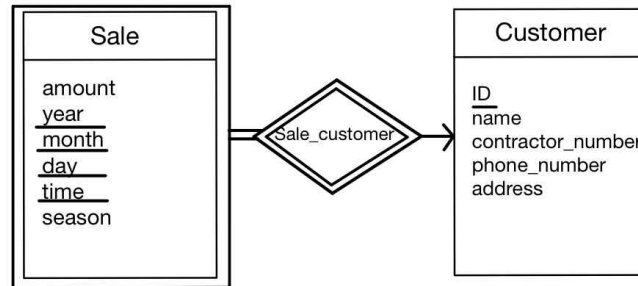
제품에 대한 판매를 나타내기 위해 Sale_product 관계를 사용하였다. 이는 One to Many and total 관계를 갖는다. 왜냐하면, 제품 하나에 대해 여러 구매가 발생할 수 있으며, 모든 구매는 제품을 갖기 때문이다. 게다가, 판매 Entity set 은 weak Entity set 으로 나타내었는데, 이는 제품명이 판매내역을 독립적으로 식별하는데 반드시 필요한 요소이기 때문이다.

(6) 판매 - 배송 관계 (Online_delivery)



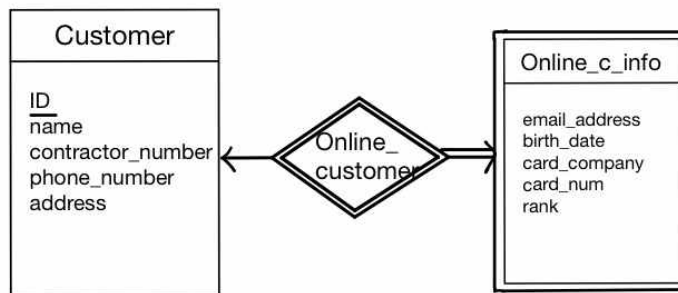
Online-delivery 관계를 별도로 표시한 이유는, 온라인 판매건에 대해 배송이라는 액션이 취해지기 때문이다. 관계가 있다면 모든 배송은 판매를 갖게 되므로 One to Many and total 의 관계로 표시하였다.

(7) 판매 - 고객 관계 (Sale_customer)



판매 Entity set에 존재하는 entity 들을 독립적으로 식별하기 위해서는 제품 정보를 포함해 당연하게도 고객의 정보가 존재해야만 한다. 이 같은 판매와 고객 사이의 관계를 나타내기 위해 Sale_customer 관계를 생성하였다. 앞서 이야기했듯이, 판매 내역 식별을 위해서는 고객 정보가 필수적이며 따라서 판매 Entity set 은 weak Entity set 이다. 그리고 관계가 있다면 모든 판매는 각 건당 고객을 갖게 되므로, One to Many and total 의 관계로 표시하였다.

(8) 고객 - 온라인 고객 정보 관계 (Online_customer)

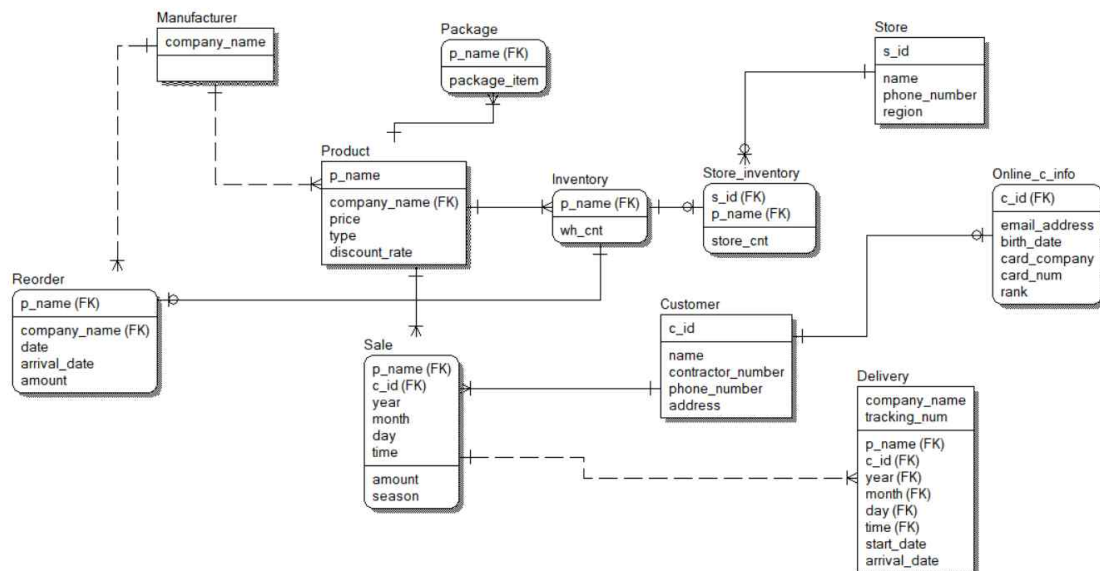


온라인 고객의 경우 별도의 정보를 저장해두었기 때문에 고객과 이 정보를 연결하는 관계가 필요하다. 그것이 바로 Online_customer Relationship set 이다. 이 둘은 관계가 존재한다면 1:1로 대응되며, 온라인 고객의 정보는 모두 고객과 관계를 맺으므로 total 관계를 갖는다. 즉, One to One and total 의 관계를 갖게 된다. 또한, 온라인 고객 정보를 식별하기 위해서는 고객의 id 가 필요하기 때문에 온라인 고객 정보 Entity set 은 weak Entity set 이 된다.

5. Reduction to Relational Schema Diagram

앞서 살펴본 E-R Model 의 Entity sets 과 Relationship sets 을 Relational model 을 통해 요약하면 다음과 같다.

- ◇ Product (name, company_name, price, type, discount_rate)
- ◇ Manufacturer (company_name)
- ◇ Package (name, package_item)
- ◇ Inventory (p_name, wh_cnt)
- ◇ Reorder (p_name, company_name, amount, date, arrival_date)
- ◇ Store (s_id, name, phone_number, region)
- ◇ Store_inventory (s_id, p_name, store_cnt)
- ◇ Customer (c_id, name, contractor_number, phone_number, address)
- ◇ Online_c_info (c_id, email_address, birth_date, card_company, card_num, rank)
- ◇ Sale (p_name, c_id, year, month, day, time, season, amount)
- ◇ Delivery (company_name, tracking_num, p_name, c_id, year, month, day, time, start_date, arrival_date)



E-R Schema diagram 이 어떻게 해서 위와 같은 Relational model을 갖게 되었는지를 설명하도록 한다. 전체적인 틀은 둘 간에 큰 차이가 없다. 다만, Relationship set을 mapping cardinality 에 따라 어떻게 table 로 표현했는지와, primary key를 명확히 볼 수 있다는데 차이가 존재한다.

1. Product (name, company_name, price, type, discount_rate)

E-R Schema diagram에서의 product Entity set 과 Relational model 의 차이는, company_name 이 새로 추가된 것과, 다중 속성인 {package_item} 이 사라진 것이다. company_name 이 추가된 이유는, Manufacturer Entity set 과 Product Entity set 간에 존재하는 One to Many and total 의 관계를 나타내기 위해 One 측의 주 키인 company_name을 Many 측인 Product Entity set 의 속성으로 추가했기 때문이다. 다중속성은 Relational schema에서 기존의 Entity set 과는 별개의 relation schema를 생성하므로 삭제되었다.

2. Manufacturer (company_name)

Manufacturer relation은 strong entity set 이므로 그대로 relation 으로 변환된다.

3. Package (p_name(FK), package_item)

패키지는 기존의 Product Entity set 의 다중 속성이있으므로 별도의 relation schema로 분리해 표시하였다. 패키지는 Product relation 의 주 키인 name을 주 키로 갖고, 기존의 다중 속성을 본인의 속성으로 갖는 relation 이다. 제품 Entity set 과 패키지 Entity set 은 각 패키지명에 대해 여러개의 제품명 속성을 가지므로, One to Many and total 의 관계로 어이진다. 따라서 제품명(패키지명)에 대해 여러개의 제품명을 갖게 될 수 있다.

4. Inventory (p_name(FK), wh_cnt)

Inventory relation은 Product 과 Many & total to One 의 관계를 가지므로, One 측의 주 키인 p_name을 본인의 속성으로 가져오게 되는데, 이때 Invenotry 가 weak Entity set 이므로 p_name 은 Inventory relation 의 주 키가 된다.

5. Reorder (p_name(FK), company_name(FK), amount, date, arrival_date)

Manufacturer 와 Inventory 간의 관계인 Reorder는 One to Many & total 의 관계이기 때문에 Many 측에 One 측의 주 키를 속성으로 포함시켜 나타낼 수도 있지만, Reorder 가 가진 별도의 속성이 존재하므로 따로 relation schema 로 분리해 나타내었다. 이는 두 Entity set 의 주 키를 본인의 주 키로 갖게 된다. Manufacturer 와 Reorder는 하나의 제조사에 여러 제품을 주문할 수 있으므로 일대다의 관계를 가지며, Inventory 와 Reorder는 각 제품에 대한 주문을 넣는 것이기 때문에 일대일의 관계로 나타내었다.

6. Store (s_id, name, phone_number, region)

Store relation 은 strong entity set 이므로 그대로 relation 으로 변환된다.

7. Store_inventory (s_id(FK), p_name(FK), store_cnt)

Store_inventory relation 은 Store 와 Inventory 간에 존재하는 One to Many 의 관계이지만, 본인만의 속성을 별도로 가지므로 이를 보여주기 위해 별도의 relation 으로 표현하였다. Store relation 은 연결하는 두 relation 의 주 키를 본인의 주 키로 갖는다. Inventory 와 Store_inventory 는 각 제품명에 대해 재고수를 나타내게 되므로 1:1의 관계를 갖고, Store 와 Store_inventory 는 일대다의 관계를 갖는다.

8. Customer (c_id(FK), name, contractor_number, phone_number, address)

Customer relation 은 strong entity set 이므로 그대로 relation 으로 변환된다.

9. Online_c_info (c_id(FK), email_address, birth_date, card_company, card_num, rank)

Online_c_info 는 Customer 와 1:1의 관계를 맺고 있으므로 어는 것이든 Many 측으로 취급해 관계를 나타낼 수 있으나, Online_c_info 는 weak Entity set 이므로, 각 Entity 들의 독립적 식별을 위해서는 Customer relation 의 주 키를 필요로 한다. 따라서 Online_c_info relation은 주 키로 c_id를 갖게 된다.

10. Sale (p_name(FK), c_id(FK), year, month, day, time, season, amount)

Sale relation 은 기존의 속성들을 모두 갖고 오면서, 두 Entity set 의 weak Entity set 이기에 이들의 주 키인 p_name 과 c_id 도 자신의 주 키로 갖고 온다.

11. Delivery (company_name, tracking_num, p_name(FK), c_id(FK), year(FK), month(FK), day(FK), time(FK), start_date, arrival_date)

Delivery relation 은 기존의 Entity set 의 속성을 모두 포함하면서, 동시에 Sale Entity set 과 One to Many & total 의 관계에 있으므로 One 측인 Sale relation 의 주 키를 본인의 속성으로 갖게 된다.

6. E-R Schema Diagram

