

Database System

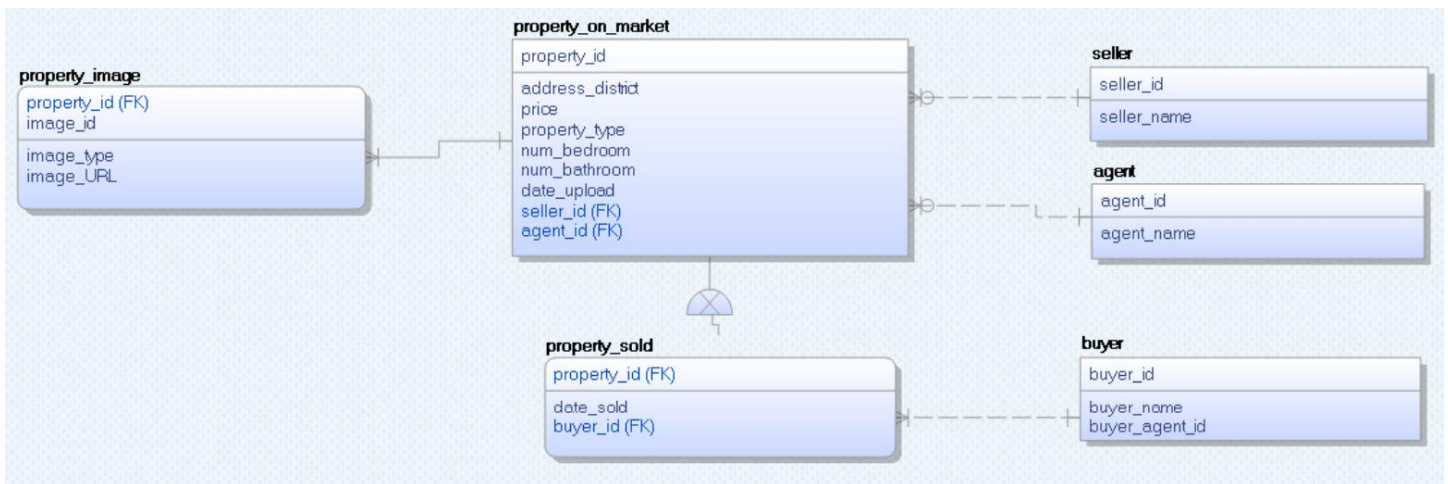
project 2

20191849 이다빈

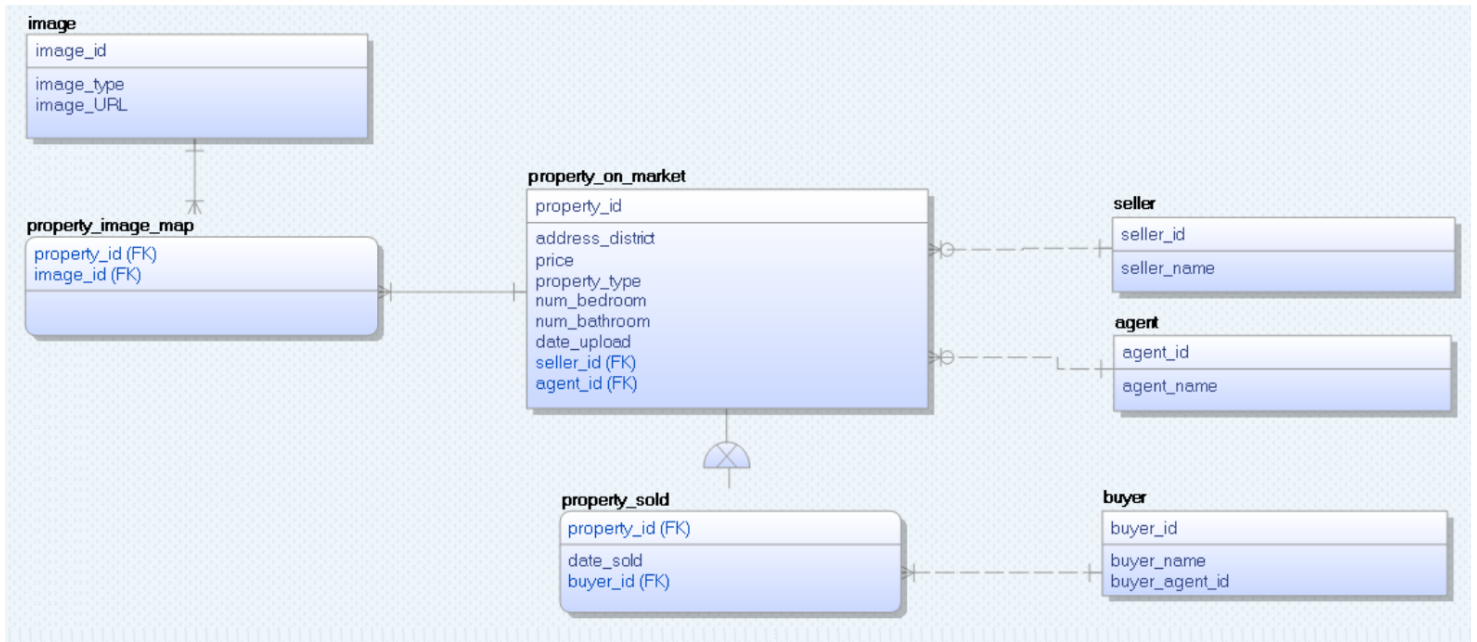
1. BCNF Decomposition

Relation R 에 있는 $a \rightarrow b$ 형태를 만족하는 모든 함수종속(FD)을 f 라고 할 때, BCNF 를 만족하려면 두 조건 중 하나를 만족해야 한다. 1) f 는 trivial dependency 이다. 2) a 는 superkey 이다.

아래는 순서대로 부동산 app 의 BCNF Decomposition 전 diagram 과 BCNF Decomposition 후 diagram 이다.



[사진 1. BCNF Decomposition 전의 diagram (project 1 과 미세하게 다른 부분이 있음)]



[사진 2. BCNF Decomposition 후의 diagram]

두 diagram 의 차이점은 'property_image' relation 이 'image' relation 과 'property_image_map' relation 으로 decompose 된 것뿐이다. 사진 1 diagram 의 property_image relation 에서 image_type 과 image_URL 은 PK 중 image_id 만 알아도 식별이 가능하다. 따라서 image_type 과 image_URL 은 PK 에 부분 함수 종속된 관계이고 이는 BCNF 를 만족하지 못한다. 이를 분해하여 BCNF 를 만족시키도록 만든 것이 사진 2 diagram 이다.

사진 2 diagram 의 모든 relation 이 BCNF 를 만족시킴을 보이자.

image : image_id -> { image_type, image_URL } 인 FD 만 존재한다. image_id 는 PK 이므로 BCNF 를 만족한다.

property_image_map : PK 인 property_id 와 image_id 를 제외하면 다른 FD 가 존재하지 않으므로 BCNF 를 만족한다.

property_on_market : property_id -> { address_district, price, property_type, num_bedroom, num_bathroom, date_upload, seller_id, agent_id } 인 FD 만 존재한다. property_id 는 PK 이므로 BCNF 를 만족한다.

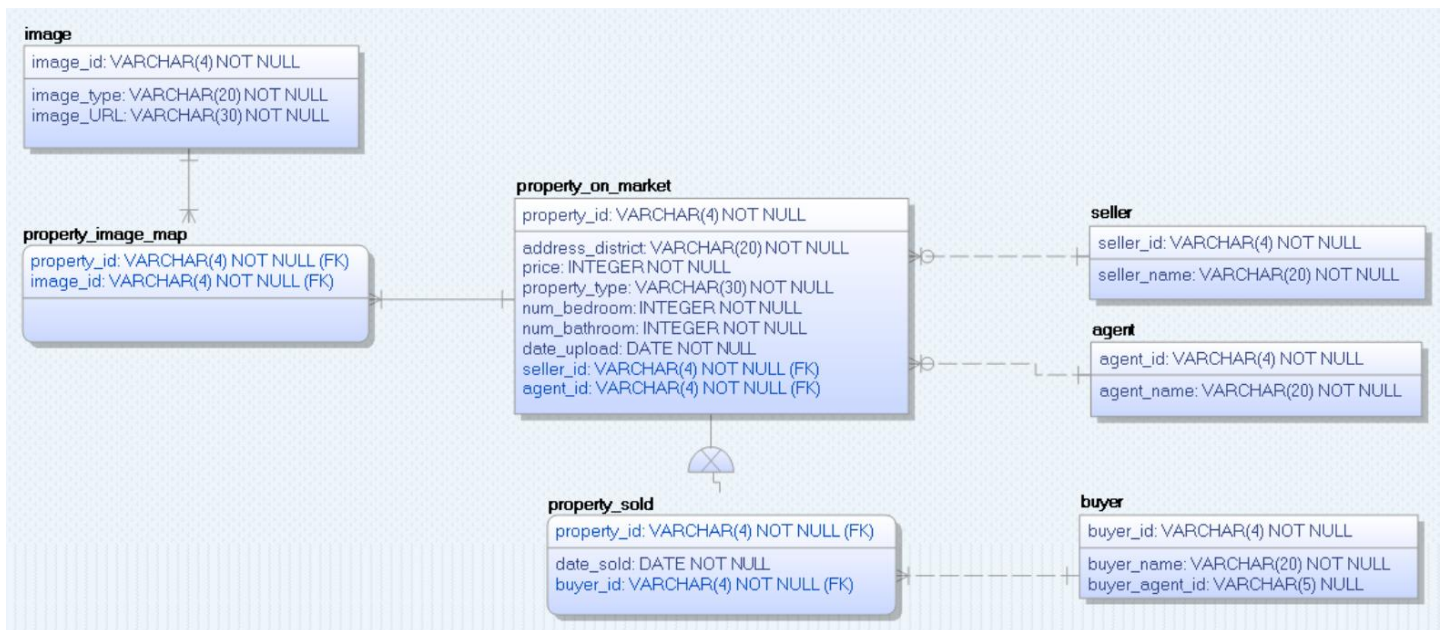
property_sold : property_id -> { address_district, price, property_type, num_bedroom, num_bathroom, date_upload, seller_id, agent_id, date_sold, buyer_id } 인 FD 만 존재한다. property_id 는 PK 이므로 BCNF 를 만족한다.

seller : seller_id -> { seller_name } 인 FD 만 존재한다. seller_id 는 PK 이므로 BCNF 를 만족한다.

agent : agent_id -> { agent_name } 인 FD 만 존재한다. agent_id 는 PK 이므로 BCNF Dependency 를 만족한다.

buyer : buyer_id -> { buyer_name, buyer_agent_id } 인 FD 만 존재한다. buyer_id 는 PK 이므로 BCNF 를 만족한다.

2. Physical Schema Diagram



[사진3. Physical Schema Diagram]

relational schema diagram의 buyer_agent_id를 제외한 모든 속성은 존재해야 하므로 not null로 설정하였다. 아래에서 각 relation에 대해 자세히 설명하겠다. 사진으로 파악할 수 있는 data type에 대해서는 필요한 부분이 아니면 아래에서 설명하지 않았다. 또한, FK는 해당 속성이 PK인 relation에서 자세히 설명하겠다.

① property_on_market

- property_id : P455의 형태로 PK이다.
- price : property의 가격이고 위 사진에서는 INTEGER 타입이지만, 더 큰 수를 표현할 수 있는 BIGINT 타입으로 지정하고 싶었으나, erwin이 BIGINT를 지원하지 않아 INTEGER로 표기한 것이다.
- property_type : studio, one bedroom apartment, multi bedroom apartment, detached house 중 하나의 값을 갖는다.
- num_bedroom : 침실 개수이다.
- num_bathroom : 화장실 개수이다.
- date_upload : property가 부동산 app 에 upload 된 날짜이다. 2021-01-01의 형태이다.
- seller_id, agent_id : FK이다.

② **property_sold**

- 상위 개체인 property_on_market의 PK인 property_id를 참조하여 FK이자 PK로 가지고 있다.
- date_sold : property가 부동산 app 에서 sold 된 날짜이다. 2021-01-01의 형태이다.
- buyer_id : FK이다.

③ **property_image_map**

- property_on_market과 다대일 관계이다. image와도 다대일 관계이다.
- image_id, property_id : FK이자 PK이다.

④ **image**

- image_id : I500의 형태로 PK이다.
- image_type : interior, exterior floor plan 중 하나의 값을 갖는다.
- image_URL : <http://image1.jpg> 의 형태로, 가상의 이미지 주소 값이다.

⑤ **seller**

- seller_id : S100의 형태로 PK이다.
- seller_name을 속성으로 가진다.

⑥ **agent**

- agent_id : A200의 형태로 PK이다.
- agent_name을 속성으로 가진다.

⑦ **buyer**

- buyer_id : B300의 형태로 PK이다.
- buyer_name을 속성으로 가진다.
- 속성 buyer_agent_id는 이 스키마 다이어그램에서 유일하게 null을 가질 수도 있는 속성이다.

3. Query 실행

먼저 txt 파일을 만들어 CRUD 를 진행하였다. 20191849_1.txt 는 table 이 남아있다면 전부 drop 을 하고 INSERT table 을 한다. 20191849_2.txt 는 table 을 drop 한다.

실행 예시 : 1 을 입력하면 type 1 에 대한 결과가 나오고, subtype 을 입력하라는 내용이 나온 후 1 을 또 입력하면 1-1 에 대한 결과가 나온다.

① TYPE 1

본래 쿼리는 address를 나타내는 거지만, database에서 address를 구만 설정하여 식별가능한 PK 인 property_id 를 결과로 보인다.

```
SELECT property_id
FROM property_on_market
WHERE address_district = 'Mapo'
```

```
Connection Succeed
----- SELECT QUERY TYPES -----

    1. TYPE 1
    2. TYPE 2
    3. TYPE 3
    4. TYPE 4
    5. TYPE 5
    6. TYPE 6
    7. TYPE 7
    0. QUIT

Select query : 1
---- TYPE 1 ----
** Find address of homes for sale in the district"Mapo"**
P456    P459    P461    P466    P467

----- Subtypes in TYPE 1 -----
    1. TYPE 1-1

Select query : |
```

1-1

```
SELECT property_id

FROM property_on_market

WHERE address_district = 'Mapo'

AND price BETWEEN 1000000000 AND 1500000000
```

```
----- Subtypes in TYPE 1 -----
      1. TYPE 1-1

Select query : 1

----- TYPE 1-1 -----
** Then find the costing between ¥ 1, 000, 000, 000 and ¥1, 500, 000, 000. **
P466      P467
----- SELECT QUERY TYPES -----

      1. TYPE 1
      2. TYPE 2
      3. TYPE 3
      4. TYPE 4
      5. TYPE 5
      6. TYPE 6
      7. TYPE 7
      0. QUIT

Select query :
```

② TYPE 2

본래 쿼리는 address를 나타내는 거지만, database에서 address를 구만 설정하여 식별가능한 PK 인 property_id 를 결과로 보인다.

```
SELECT property_id
FROM property_on_market
WHERE address_district IN ('Gangnam', 'Seocho')
```

```
Connection Succeed
----- SELECT QUERY TYPES -----

      1. TYPE 1
      2. TYPE 2
      3. TYPE 3
      4. TYPE 4
      5. TYPE 5
      6. TYPE 6
      7. TYPE 7
      0. QUIT

Select query : 2
---- TYPE 2 ----
** Find the address of homes for sale in the 8th school district. **
P464      P468      P469

----- Subtypes in TYPE 2 -----
      1. TYPE 2-1

Select query :
```

2-1

```
SELECT property_id
```

```
FROM property_on_market
```

```
WHERE address_district IN('Gangnam', 'Seocho') AND num_bedroom >= 4 AND
num_bathroom = 2
```

----- Subtypes in TYPE 2 -----

1. TYPE 2-1

Select query : 1

----- TYPE 2-1 -----

** Then find properties with 4 or more bedrooms and 2 bathrooms. **

P468 P469

----- SELECT QUERY TYPES -----

1. TYPE 1

2. TYPE 2

3. TYPE 3

4. TYPE 4

5. TYPE 5

6. TYPE 6

7. TYPE 7

0. QUIT

Select query :