DB Project1 Report

20191048 김도솔

1. E-R model

1-1. entity sets

1) agent (중개인)

중개인은 부동산 판매 과정에서 중개 역할을 수행하는 사람이다. 이 entity는 중개인의 id, 이름, 전화 번호, 고용 날짜를 담고 있다. 각 agent는 고용 시 고유의 id를 부여 받고 이를 통해 식별된다.

2) buyer (구매자)

구매자는 부동산을 구매하려는 개인이나 단체를 나타낸다. 이 entity는 구매자의 id, 이름, 전화번호 등의 개인 정보를 관리한다. 각 구매자는 고유한 id를 부여 받고 이를 통해 식별된다.

3) seller (판매자)

판매자는 부동산을 시장에 내놓는 개인이나 단체를 나타낸다. 이 entity는 판매자의 id, 이름, 전화번호 등의 개인 정보를 관리한다. 각 판매자는 고유한 id를 부여 받고 이를 통해 식별된다.

4) property (부동산)

부동산 entity는 매물로 나온 모든 부동산의 세부 사항을 저장한다. 이는 부동산의 id, 상세 주소(시, 구 제외 동부터 입력), 가격, 부동산 유형, 침실 및 욕실 수 등을 포함하며, 부동산의 판매 상태 및 시장에 나온 날짜 등도 관리한다. 각 부동산은 매물로 등록될 때 고유한 id를 부여 받고 이를 통해 식별 된다. 또한 seller의 id와 district의 id를 외래 키로 참조하여 seller_id를 통해 부동산을 판매하는 판매자의 정보, district_id를 통해 이 부동산이 속한 지구의 정보를 함께 기록할 수 있도록 하였다.

5) sale (판매 기록)

개별 부동산 거래의 세부 사항을 추적하고자 구현한 entity set이다. 여기에는 판매 기록의 id, 부동산의 실제 판매 가격, 판매 날짜가 저장된다. 각 판매 기록은 거래 성사 시 고유한 id를 부여 받고 이를 통해 식별 된다. 또한 property_id, buyer_id, selling_agent_id, buyers_agent_id를 외래 키로 참조해 매매에 참여한 부동산, 구매자, 판매 에이전트, 구매 에이전트의 정보가 매매 세부정보에 같이 기록될 수 있도록 구성했다. 이때 명세서의 query f)를 구현하기 위해 agent_id를 각각 다른 이름으로 두 번 참조해 selling_agent_id, buyers_agent_id라는 새로운 이름을 붙였다. 또한 구매 에이전트의 경우에는 거래에 참여한 구매 에이전트가 존재하는 경우에만 정보를 저장할 수 있도록 buyers_agent_id 속성에 null 값을 허용했다.

6) photo (사진)

부동산 매물의 사진을 관리하는 entity set이다. 이 곳에는 사진의 id, 유형, 이미지 파일이 저장된다. 각 사진은 매물을 시각적으로 보여주며, 사진의 유형은 내부, 외부, 평면도 등으로 구성된다. 또한 property_id를 외래 키로 참조해 각 사진이 어느 매물의 사진인지 알 수 있도록 하였다.

7) district (지구)

서울의 각 구에 대한 정보를 담고 있는 entity set이다. 각 지구의 id(이름)과 학군의 정보가 담겨 있다. 명세서의 query a), b)를 구현하기 위해 만들었다. 이때 지구와 학군을 property의 속성으로 저장하는 것보다 따로 관리하는 것이 데이터베이스의 공간을 효율적으로 사용하는 것이라 판단해 독립적인 entity set으로 구현하게 되었다.

1-2. relationship sets

1) buys

buyer와 sale이 참여하는 binary realationship set이다. 한 구매자가 여러 부동산을 구매할 수 있지만, 각 sale은 한 명의 구매자를 기록해야 한다. 따라서 일대다 관계가 형성된다. 또한 모든 판매는 항상 구매자가 존재해야 하지만 구매자의 경우 아직 판매에 참여하지 않은 구매자가 존재할 수 있기 때문에 sale은 이 관계에 total로 참여하고 buyer는 partial로 참여한다.

2) selling, buyers

둘 다 agent와 sale이 참여하는 binary realationship set이다. 중개인 한 명이 여러 건의 판매를 담당할 수 있으나, 각 판매 건은 한 명의 중개인에 의해 이루어진다. 따라서 일대다 관계가 형성된다. 이때 query f)에 따르면 모든 sale은 판매 중개인이 있고, 구매 중개인은 있을 수도, 없을 수도 있는 걸로 해석된다. 따라서 sale은 selling 관계에 total로 참여하고, buyers 관계에는 partial로 참여한다. 신입 에이전트의 경우 아직 판매에 참여하지 않았을 수도 있으므로 agent는 두 관계에 모두 partial로 참여한다.

3) sale_prop

property와 sale이 참여하는 binary realationship set이다. 한 부동산은 여러 번 판매(ex. 이전 판매와 새 판매)될 수 있다. 하지만 각 판매 기록은 한 번에 하나의 부동산만 참조하므로 이 관계는 일대다이다. 모든 판매 기록은 반드시 부동산 정보가 있어야 하기 때문에 sale은 이 관계에 total로 참여한다. 반면 아직 판매 기록이 없는 부동산이 존재할 수 있기 때문에 property는 이 관계에 partial로 참여한다.

4) prop_list

property와 seller가 참여하는 binary realationship set이다. 판매자 한 명이 여러 부동산을 시장에 내놓을 수 있다. 따라서 일대다 관계가 형성된다. 모든 부동산은 반드시 판매자가 존재해야 하기 때문에 property는 이 관계에 total로 참여한다. 반면 부동산 사무실에서 상담만 받고 아직 부동산을 리스트에 올리지 않은 판매자가 존재할 수 있기 때문에 seller는 이 관계에 partial로 참여한다.

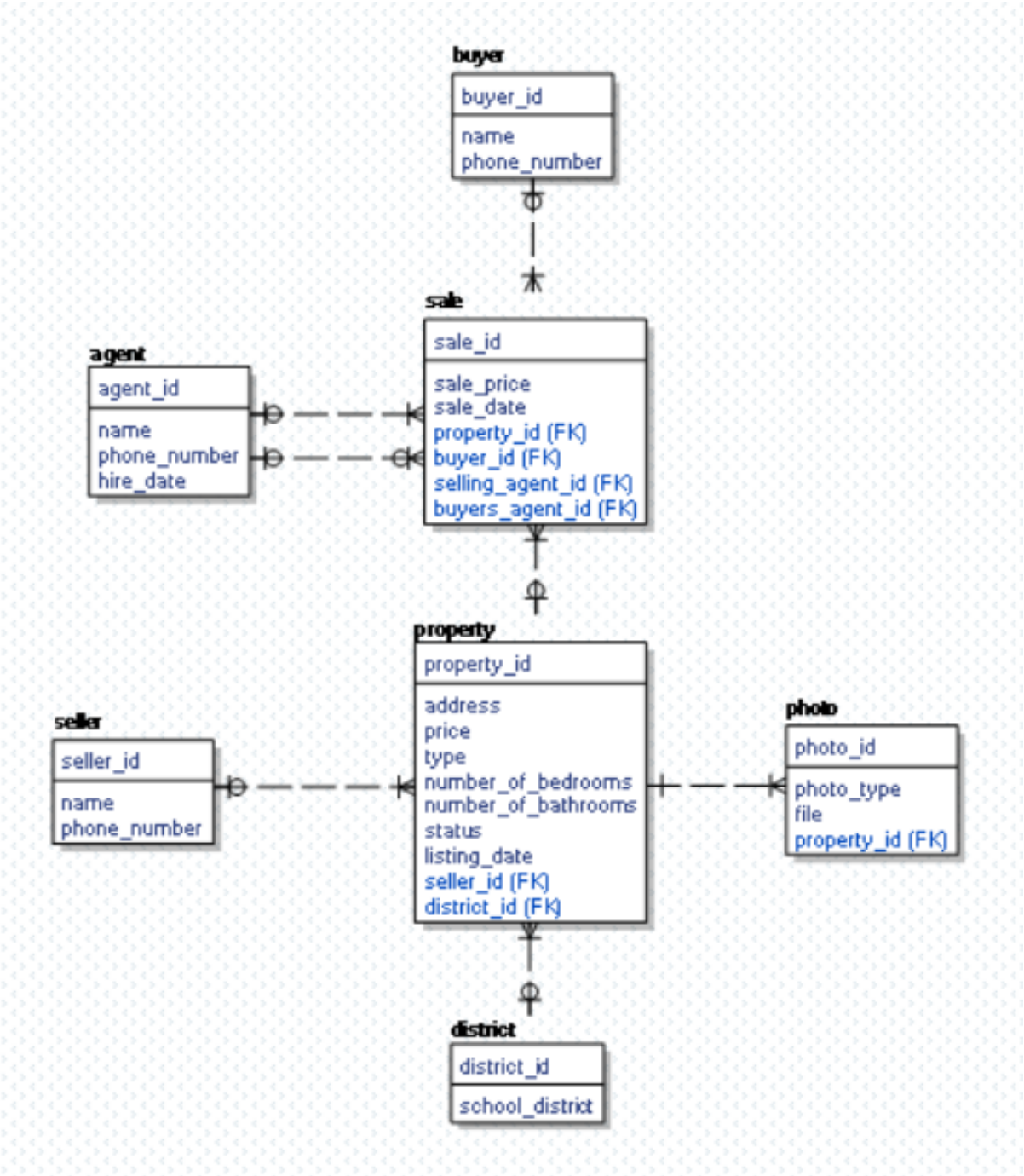
5) prop_photo

property와 photo가 참여하는 binary realationship set이다. 하나의 부동산에 여러 장의 사진이 있을 수 있기 때문에 일대다 관계가 형성된다. 이때 명세서에 따르면 studio나 one-bedroom 아파트의 경우 적어도 한 장의 실내 사진이 있어야 하고, multi-bedroom 아파트나 detached houses(단독주택)의 경우 적어도 한 장의 외부 사진과 한 장의 평면도가 있어야 하기에 property는 이 관계에 total로 참여한다. 또한 모든 사진은 해당되는 부동산이 존재해야하기 때문에 photo도 이 관계에 total로 참여한다.

6) prop_dist

property와 district가 참여하는 binary realationship set이다. 하나의 지구(district)에 여러 부동산이 위치할 수 있기 때문에 이 관계는 일대다이다. 모든 부동산은 반드시 하나의 지구에 속해야 하기 때문에 property는 이 관계에 total로 참여하고, 아직 부동산 정보를 가지고 있지 않은 지구가 있을 수 있기 때문에 district는 이 관계에 partial로 참여한다.

2. Schema diagram



2-1. Attribute detail

entity	attribute	data type, null option	상세 설명
agent	agent_id(PK)	varchar(18), not null	에이전트의 고유 식별자. 모든 에이전트를 유일하게 식별하는 데 사용된다.
	name	varchar(18), not null	에이전트의 이름
	phone_number	varchar(18), not null	에이전트의 업무 전화번호
	hire_date	date, not null	에이전트가 고용된 날짜. 에이전트의 경력을 추적하는 데 사용될 수 있다.
buyer	buyer_id(PK)	varchar(18), not null	구매자의 고유 식별자. 모든 구매자를 유일하게 식별하는 데 사용된다.
	name	varchar(18), not null	구매자의 이름
	phone_number	varchar(18), not null	구매자의 전화번호
seller	seller_id(PK)	varchar(18), not null	판매자의 고유 식별자. 모든 판매자를 유일하게 식별하는 데 사용된다.
	name	varchar(18), not null	판매자의 이름
	phone_number	varchar(18), not null	판매자의 전화번호
property	property_id(PK)	varchar(18), not null	부동산의 고유 식별자. 모든 부동산을 유일하게 식별하는 데 사용된다.
	address	varchar(200), not null	시, 구를 제외한 부동산의 상세 주소(동부터 입력)를 나타낸다. 정보의 중복을 방지하기 위해 상세 주소만으로 구성했다. query에서 주소 출력이 필요한 경우, 명세서에 이 office는 서울시의 매물만 다룬다고 나와있으므로 서울시는 고정이고, 구 정보는 district_id(FK)를 통해 참조할 수 있다.
	price	float, not null	부동산이 시장에 올라온 가격
	type	varchar(18), not null	부동산의 유형 (studio, one_bedroom, multi_bedroom, detached_house 등으로 구성)

	number_of_bedrooms	int, not null	부동산에 포함된 침실의 수
	number_of_bathrooms	int, not null	부동산에 포함된 욕실의 수
	status	varchar(18), not null	부동산의 판매 상태 (on_the_market, sold_out 등으로 구성)
	listing_date	date, not null	부동산이 시장에 등록된 날짜
	seller_id (FK)	varchar(18), not null	이 부동산을 판매하는 판매자의 id를 외래 키로 참조한다.
	district_id(FK)	varchar(18), not null	이 부동산이 속한 지구의 id를 외래키로 참조한다.
sale	sale_id(PK)	varchar(18), not null	판매 기록의 고유 식별자. 모든 판매 기록을 유일하게 식별하는 데 사용된다.
	sale_price	float, not null	부동산이 실제 거래된 가격
	sale_date	date, not null	판매가 완료된 날짜
	property_id(FK)	varchar(18), not null	판매된 부동산의 id를 외래 키로 참조한다.
	buyer_id (FK)	varchar(18), not null	부동산을 구매한 구매자의 id를 외래 키로 참조한다.
	selling_agent_id (FK)	varchar(18), not null	selling agent의 고유 식별자. 이 부동산의 판매를 담당한 agent의 id를 외래 키로 참조한다.
	buyers_agent_id (FK)	varchar(18), null	buyer's agent의 고유 식별자. 구매자를 대리한 에이전트의 id를 외래 키로 참조한다. 이때 명세서에 작성된 대로 buyer의 agent는 존재하는 경우에만 기록할 수 있게 구성했다.
photo	photo_id(PK)	varchar(50), not null	사진의 고유 식별자. 모든 사진을 유일하게 식별하는데 사용된다.
	photo_type	varchar(18), not null	사진의 유형 (interior, exterior, floor_plan 등으로 구성)
	file	image, not null	사진 파일이 blob 형태로 저장된다.
	property_id(FK)	varchar(18), not null	사진이 찍힌 부동산의 id를 외래 키로 참조한다.
district	district_id(PK)	varchar(18),	서울 지구의 고유 식별자 (Mapo,

		not null	Seodaemun, Nowon 등으로 구성)
	school_district	varchar(18), not null	각 지구가 속한 학군 (1st, 2nd, 3rd 등으로 구성)

2-2. 관계형 스키마 변환 과정

설계한 E-R 모델의 경우 각 entity set 은 모두 strong entity set 에 해당되기 때문에 각 개체 집합을 그대로 스키마로 표현할 수 있다. 또한 모든 relationship set 은 일대다 관계를 형성하고 있으므로 스키마의 결합으로 표현이 가능하다. 모든 관계 집합이 일대다이므로 모두 스키마 결합이 가능하기 때문에 각 관계 집합을 스키마로 변환하는 과정은 생략했다. 스키마 결합 시 한 릴레이션 스키마의 주 키를 다른 릴레이션 스키마의 외래 키로 추가하여 릴레이션 간의 관계를 나타낼 수 있다.

1 단계) 강한 개체 집합을 스키마로 나타내기

buyer(buyer_id, name, phone_number)
agent(agent_id, name, phone_number, hire_date)
sale(sale_id, sale_price, sale_date)
property(property_id, address, price, type, number_of_bedrooms, number_of_bathrooms, status, listing_date)
seller(seller_id, name, phone_number)
photo(photo_id, photo_type, file)
district(district_id, school_district)

2단계) 일대다 관계 집합들의 스키마 결합

buyer(buyer_id, name, phone_number)
agent(agent_id, name, phone_number, hire_date)
sale(sale_id, sale_price, sale_date, property_id(FK), buyer_id (FK), selling_agent_id (FK) , buyers_agent_id (FK))
property(property_id, address, price, type, number_of_bedrooms, number_of_bathrooms, status, listing_date, seller_id (FK), district_id(FK))
seller(seller_id, name, phone_number)
photo(photo_id, photo_type, file, property_id(FK))
district(district_id, school_district)

4. Queries

이번 프로젝트에서 설계한 모델을 기반으로 명세서에 나타난 질의문을 대략적으로 작성해보았다.

a) Find address of homes for sale in the district "Mapo" costing between ₩1,000,000,000 and ₩1,500,000,000.

```
SELECT p.district_id, p.address
FROM property p
JOIN district d ON p.district_id = d.district_id
WHERE d.district_id = 'Mapo'
  AND p.price BETWEEN 1000000000.0 AND 1500000000.0
  AND p.status = 'on_the_market';
```

b) Find the address of homes for sale in the 8th school district with 4 or more bedrooms and 2 bathrooms.

```
SELECT p.district_id, p.address
FROM property p
JOIN district d ON p.district_id = d.district_id
WHERE d.school_district = '8th'
  AND p.number_of_bedrooms >= 4
  AND p.number_of_bathrooms = 2
  AND p.status = 'on_the_market';
```

c) Find the name of the agent who has sold the most properties in the year 2022 by total won value.

```
SELECT a.name
FROM agent a
JOIN sale s ON a.agent_id = s.selling_agent_id
WHERE YEAR(s.sale_date) = 2022
GROUP BY a.agent_id, a.name
ORDER BY SUM(s.sale_price) DESC
LIMIT 1;
```

d) For each agent, compute the average selling price of properties sold in 2022, and the average time the property was on the market. (Note that this suggests the use of date attributes in your design)

```
SELECT a.name, AVG(s.sale_price), AVG(DATEDIFF(s.sale_date, p.listing_date))
FROM agent a
JOIN sale s ON a.agent_id = s.selling_agent_id
JOIN property p ON s.property_id = p.property_id
WHERE YEAR(s.sale_date) = 2022
GROUP BY a.agent_id, a.name;
```

e) Show photos of the most expensive studio, one-bedroom, multi-bedroom apartment(s), and detached house(s), respectively, from the database.

```
SELECT p.type, p.price, ph.photo_type, ph.file
FROM property p
JOIN photo ph ON p.property_id = ph.property_id
WHERE
    p.property_id IN (
        SELECT property_id
        FROM property
        WHERE type = 'studio'
        ORDER BY price DESC
        LIMIT 1
    )
    OR p.property_id IN (
        SELECT property_id
        FROM property
        WHERE type = 'one_bedroom'
        ORDER BY price DESC
        LIMIT 1
    )
    OR p.property_id IN (
        SELECT property_id
        FROM property
        WHERE type = 'multi_bedroom'
        ORDER BY price DESC
        LIMIT 1
    )
    OR p.property_id IN (
        SELECT property_id
        FROM property
        WHERE type = 'detached_house'
```

```
        ORDER BY price DESC
        LIMIT 1
    );
```

f) Record the sale of a property that had been listed as being available. This entails storing the sales price, the buyer, the selling agent, the buyer's agent(if any), and the date.

```
INSERT INTO sale (sale_id, sale_price, sale_date, property_id, buyer_id, selling_agent_id,
buyers_agent_id)
SELECT
    'SALE2206' AS sale_id
    p.property_id,
    3500000000.0 AS sale_price,
    'BUY3182' AS buyer_id,
    'AGT03' AS selling_agent_id,
    'AGT05' AS buyers_agent_id,
    CURRENT_DATE() AS sale_date
FROM
    property p
WHERE
    p.property_id = 'PROP1405' AND
    p.status = 'on_the_market';
```

g) Add a new agent to the database.

```
INSERT INTO agent (name, phone_number, hire_date)
VALUES ('dosol', '010-1234-5678', 2019-03-05);
```