

Database System Project 1

담당 교수: 정성원

이름: 김태규

학번: 20191243

1. 프로젝트 개요

가상의 Real Estate Office의 'DBA(Database Administrator)'가 되어, Manager의 요구사항을 만족시키는 데이터베이스를 설계하는 것이 본 프로젝트의 핵심이다. 이번 1차 프로젝트에서는 E-R Model과 Relational Database Schema를 통해 데이터베이스의 초기 디자인을 구축한다.

2. E-R Model

A. Requirements 분석

부동산 업무를 운영하는 애플리케이션을 디자인하고 구현해야 한다. 이 애플리케이션은 중개인, 구매자, 판매자, 시장에 있는 부동산, 최근에 판매된 부동산 등을 추적해야 하며, 이 부동산 업무는 서울의 주택에 중점을 두고 있다. 또한, studio 또는 one-bedroom 아파트에는 적어도 한 장의 실내 사진이 있어야 하며, multi-bedroom 아파트나 단독주택에는 적어도 한 장의 외부 사진과 한 장의 평면도가 있어야 한다.

B. Example Query 분석

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

이 쿼리는 마포구에 위치한, 가격이 10억원에서 15억원 사이인 주택의 주소를 찾는 것이다. 이를 위해서는 마포구에 해당하는 주택을 선택하고, 그 중에서 가격이 해당 범위에 있는 것을 필터링해야 한다. 이 때, 부동산의 가격은 고정되어 있는 것이 아니므로 최근 5년 간의 거래 가격의 평균을 부동산의 가격으로 정의하겠다.

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

이 쿼리는 8학군에 위치한 침실이 4개 이상이고 화장실이 2개 이상인 주택의 주소를 찾는 것이다. 이를 위해서는 8학군에 해당하는 주택을 선택하고,

침실과 화장실의 개수가 조건에 부합하는 것을 필터링해야 한다. 이 쿼리를 수월하게 수행하기 위해서는 Property와 School_district의 두 entity set 간에 학군지를 나타낼 수 있도록 relationship이 정의되어야 한다.

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

이 쿼리는 2022년에 가장 많은 부동산을 총 거래 가치에 따라 판매한 중개인의 이름을 찾는 것이다. 이를 위해서는 2022년에 판매된 모든 부동산 거래를 중개인별로 그룹화하고, 각 중개인이 총 거래 가치를 계산한 후에 가장 큰 값을 가진 중개인을 찾아야 한다. 현재까지의 거래 내역을 기록할 수 있도록 하는 Transaction relationship에 (selling) Agent entity set이 참여하도록 하고 price를 해당 relationship의 attribute로 추가해줌으로써 쿼리문이 잘 수행될 수 있도록 한다.

Query d) For each agent, compute the average selling price of properties sold in 2022, and the average time the property was on the market.

이 쿼리는 각 중개인별로 2022년에 판매된 부동산의 평균 판매 가격과 평균 판매 기간을 계산하는 것이다. 이를 위해서는 각 중개인이 판매한 부동산의 판매 가격과 판매 기간을 계산하고, 해당 값을 평균하여야 합니다. 이 쿼리 또한 Transaction relationship에 매물로 등록된 날짜(registration_date)와 매물이 팔린 날짜(sold_date)라는 attribute가 포함되도록 하면 쿼리를 쉽게 수행할 수 있을 것으로 보인다.

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

이 쿼리는 데이터베이스에서 가장 비싼 스튜디오(원룸), 침실이 한 개 있는 아파트, 다수의 침실이 있는 아파트, 그리고 단독 주택의 사진을 보여주는 것이다. 이를 위해서는 각 유형의 부동산 중에서 최고 가격의 부동산을 선택하

고 해당 부동산의 사진을 표시해야 한다. 우선 최종 목표가 사진을 보여주는 것이므로 Property entity에 사진이 포함되도록 해야 한다. 종류 별로 가장 비싼 집을 찾는 것은 이전 쿼리문 분석에 기술했던 Transaction relationship에서의 price와 Property entity가 집의 종류를 내포할 수 있도록 하여 쿼리문을 수행할 것이다.

Query 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.

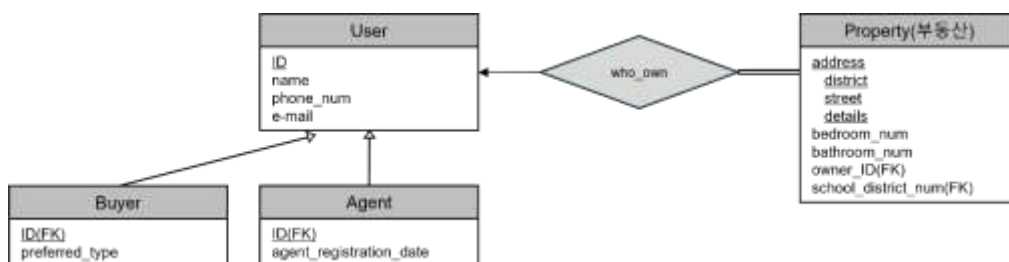
이 쿼리는 판매 가능한 부동산을 판매하는 거래를 기록하는 것이다. 이를 위해서는 판매 가격, 구매자, 판매 중개인, 구매자의 중개인(있는 경우), 그리고 거래 날짜를 저장해야 한다. 이 쿼리문은 전적으로 Transaction relationship의 존재 이유와 일치한다. 따라서 Transaction relationship을 구현함으로써 쉽게 수행할 수 있을 것이다.

Query g) Add a new agent to the database.

이 쿼리는 데이터베이스에 새로운 중개인을 추가하는 것이다. 이를 위해서는 새로운 중개인의 정보(이름, 연락처 등)를 데이터베이스에 삽입해야 한다. 이는 User의 child entity set인 Agent를 구현하여 agent 데이터를 추가하는 쿼리문으로 쉽게 수행할 수 있을 것으로 보인다.

C. E-R Model 구축

1. User 부분



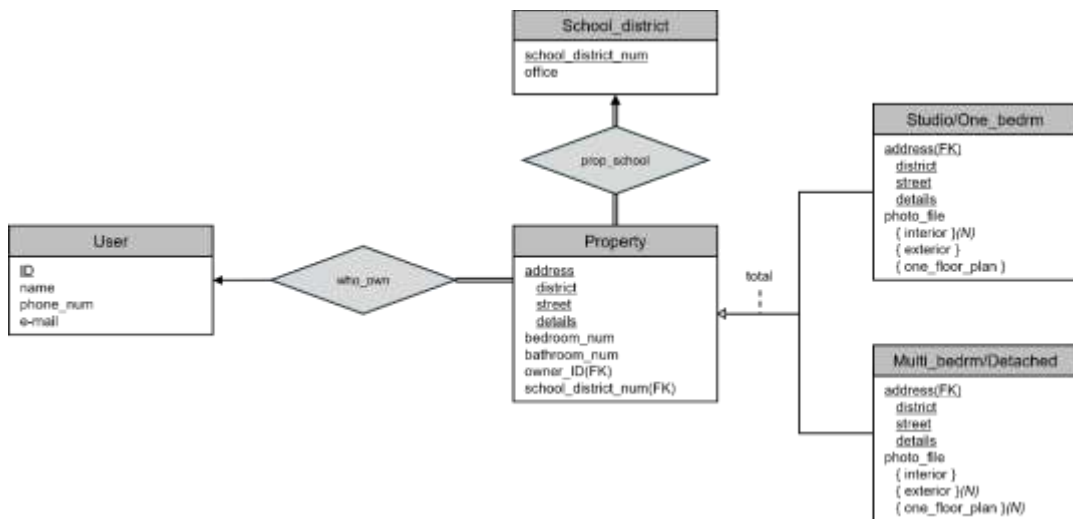
먼저 이용자가 Buyer, Agent, Seller 모두 될 수 있기 때문에 User라는 상위

entity set을 두고 Buyer, Agent라는 하위 entity set을 두었다. Buyer가 Agent 역할 또한 할 수 있기에 overlapping specialization을 적용하였다.

Seller entity set이 없는 이유는 다음과 같다. 먼저 User와 Property entity set 간에 "who_own"이라는 relationship set을 통해 부동산의 소유주를 기록할 수 있도록 하였다. 부동산의 소유주만이 부동산을 판매할 수 있으므로 이를 통해 Seller entity set을 대체할 수 있다고 생각하였다. 그럼에도 불구하고 Seller entity set을 추가할까 했지만 data redundancy를 고려하여 User의 child entity로 Buyer와 Agent만 추가하였다.

User의 attribute 중 phone_num은 multi-value를 가질 수 있지만 현재 requirement 중 특별한 언급이 없었기 때문에 single attribute라고 가정하고 들어갈 것이다.

2. Property 부분



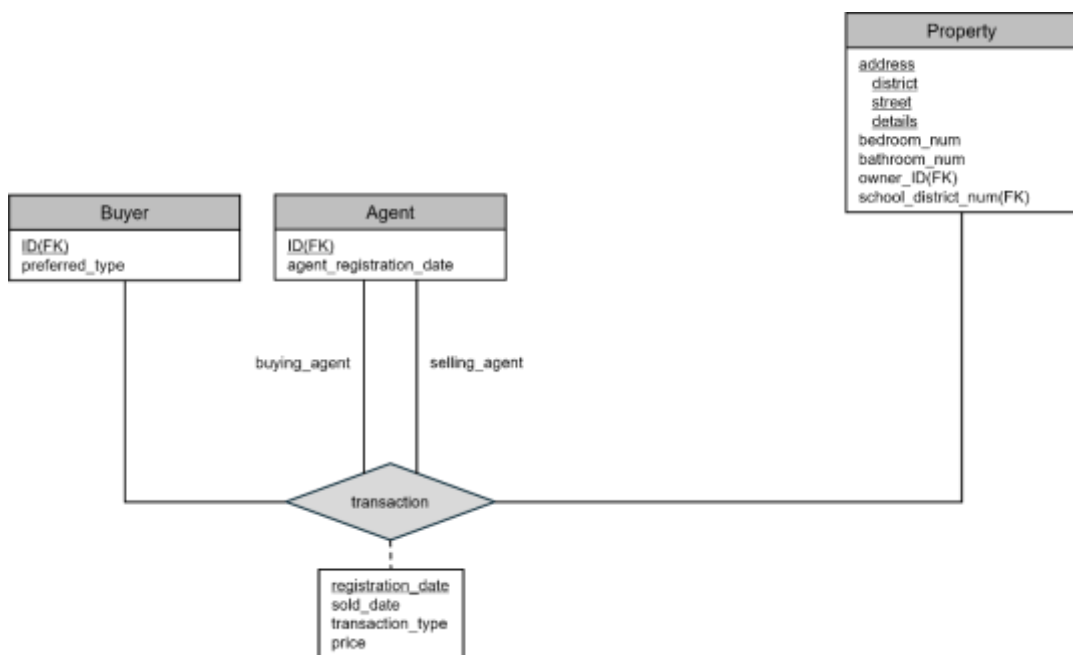
Description에 의하면 Property의 종류에는 studio, one-bedroom apartment, multi-bedroom apartment, 그리고 detached house 이렇게 네 가지로 나뉜다. 하지만 Client의 Requirement에 의하면 studio, one-bedroom apartment 구조를 가진 부동산은 최소 하나의 내부 사진이 있어야 하고, multi-bedroom apartment, detached house의 구조를 가진 부동산은 외부 사진과 단층도 각각 최소 하나씩 보유해야 한다고 한다. 따라서 해당 E-R Diagram에서는 total distinct specialization을 적용하여 Property entity set의 하위 entity set으로 Studio/One_bedrm과 Multi_bedrm/Detached 두 entity set을 두었다. 추후에 더

많은 requirement가 생긴다면 이에 따라 4개의 entity set으로 분리하는 것도 고려할 수 있다.

Studio/One_bedrm entity set의 interior attribute는 최소 하나의 value가 존재해야 하므로 NULL이 되는 것을 방지하였다. 마찬가지로 Multi_bedrm/Detached entity set에서는 exterior와 one_floor_plan attribute의 value가 NULL이 되는 것을 방지하였다.

또한 부동산에 학군지 정보도 포함시켜야 한다는 Requirement(query b)가 있었기에 School_district라는 학군 번호와 교육청 정보를 담고 있는 entity set을 추가하여 Property entity set과 relationship set을 이루도록 하였다. 현재 서울의 모든 구에 부동산이 최소 하나 존재하므로 total participation 관계를 이룬다.

3. Transaction 부분



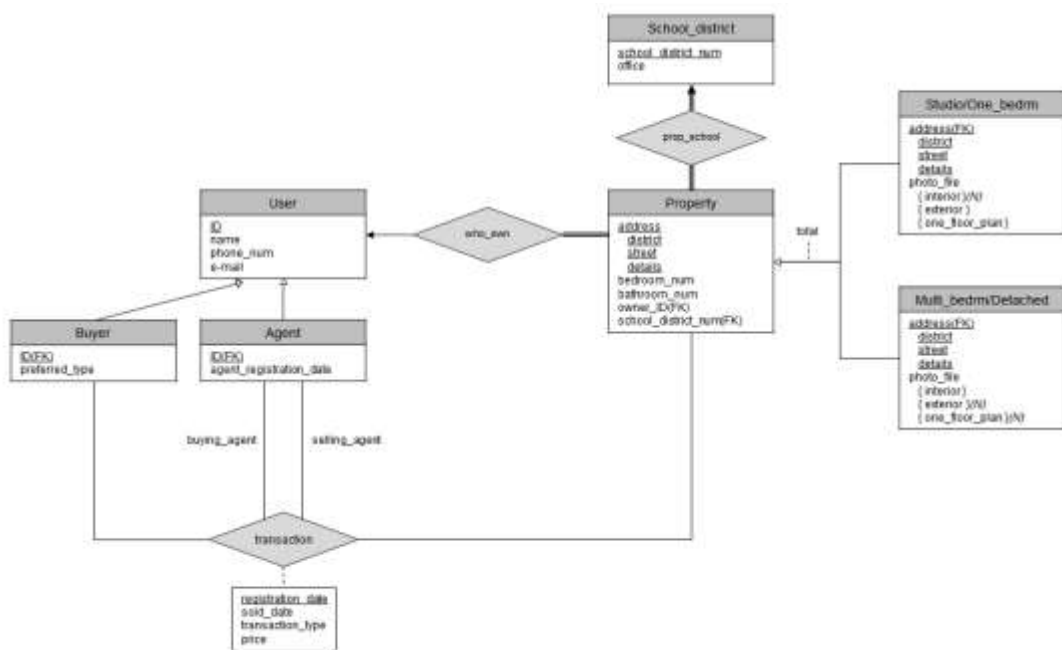
먼저 위의 Transaction relationship set은 총 4개의 entity set이 참여하는 relationship set이다. 수업 시간에 되도록이면 binary relationship을 가지도록 설계하라고 배웠고 그 이유는 cardinality를 정의하기 어려울 때가 있기 때문이었다. 하지만 위의 Transaction relationship set은 자명하게 many-to-many cardinality를 가진다. 또한, 수업 시간에 배웠던 대로 Transaction이라는 entity set을 새로 두어 다른 entity set들과 각각 binary relationship을 갖도록 할 수 있었지만, 본인은 entity set을 추가하는 것보다 Buyer, Agent, Property 간에 거래라는 "관계

(relationship)”를 가진다는 의미를 강조하고 싶어 relationship set으로 유지하였다.

Agent entity set은 해당 relationship set에 buying_agent와 selling_agent라는 역할로 두 번 참여한다. Transaction의 primary key는 4개의 entity set의 primary key들을 union한 결과이다. 하지만 부동산 거래에서 중개인이 항상 있을 것이라는 보장은 없다. 그렇게 되면 buying agent, selling agent attribute의 value가 NULL이 될 가능성이 있는데 이는 primary key 조건에 위배된다. 따라서 이러한 상황을 방지하기 위해 해당 DB를 launching 하자마자 Agent entity set에 NULL 값 대신 사용하게 될, 공란을 의미하는 entity를 추가한다고 가정하겠다.

또한, 같은 부동산에 대해 모두 같은 buyer, agent로 이루어진 transaction을 구별하기 위해 매물 등록 일자(registration_date)를 attribute로 두어 primary key로 설정하였다. 그리고 다른 attribute로 임대/매매 여부 정보를 담는 transaction type, 거래 가격(price), 거래 일자(sold_date)를 추가하였다.

이러한 과정을 통해 아래와 같은 E-R Diagram을 만들 수 있다.



3. Relational Schema Diagram

위의 과정을 통해 제작한 E-R Diagram을 Relational Schema Diagram으로 Reduction을 수행하였다.

A. Inheritance 관계

상속 관계 표현 방법은 크게 두 가지 방법이 있는데, Primary Key만을 상속하는 방법으로 구축하였다. Cardinality가 one-to-one/zero인 identifying relationship을 만들어 연결하였다. 상위 relation의 primary key가 하위 relation의 primary key이자, foreign key로 작동하게 된다. 위의 E-R Diagram에서 Buyer, Agent, Studio/One_bedrm, Multi_bedrm/Detached entity set이 이에 해당한다.

B. Multivalued 처리

위의 E-R Diagram에서 Studio/One_bedrm과 Multi_bedrm/Detached entity set의 interior, exterior, one_floor_plan attribute는 multivalued이다. 따라서 별도의 schema로 분리해서 address를 통해 연결해주었다. One-to-many의 cardinality를 적용하였다.

C. One-to-many 관계

Relation을 별도로 추가하지 않고 one 쪽의 entity set의 primary key를 many 쪽에 일반 attribute로 포함시켜주었다. 위의 E-R Diagram에서 Property entity set의 owner_ID, school_district_num이 이에 해당한다.

D. Many-to-many 관계

해당 relationship set의 이름과 같은 이름의 relation을 새로 만들고, relationship에 참여하는 entity set의 primary key들을 모두 가져와 해당 relation의 primary key로 취급하였다. 위의 E-R Diagram에서 Transaction relationship set이 이에 해당한다.

이러한 routine을 통해 수행한 Reduction의 결과는 아래와 같다. 이 때 total participation이 있는 경우 Reduction 시에 cardinality 하한을 1로 설정하는 것도 중요하다.

