

Project2

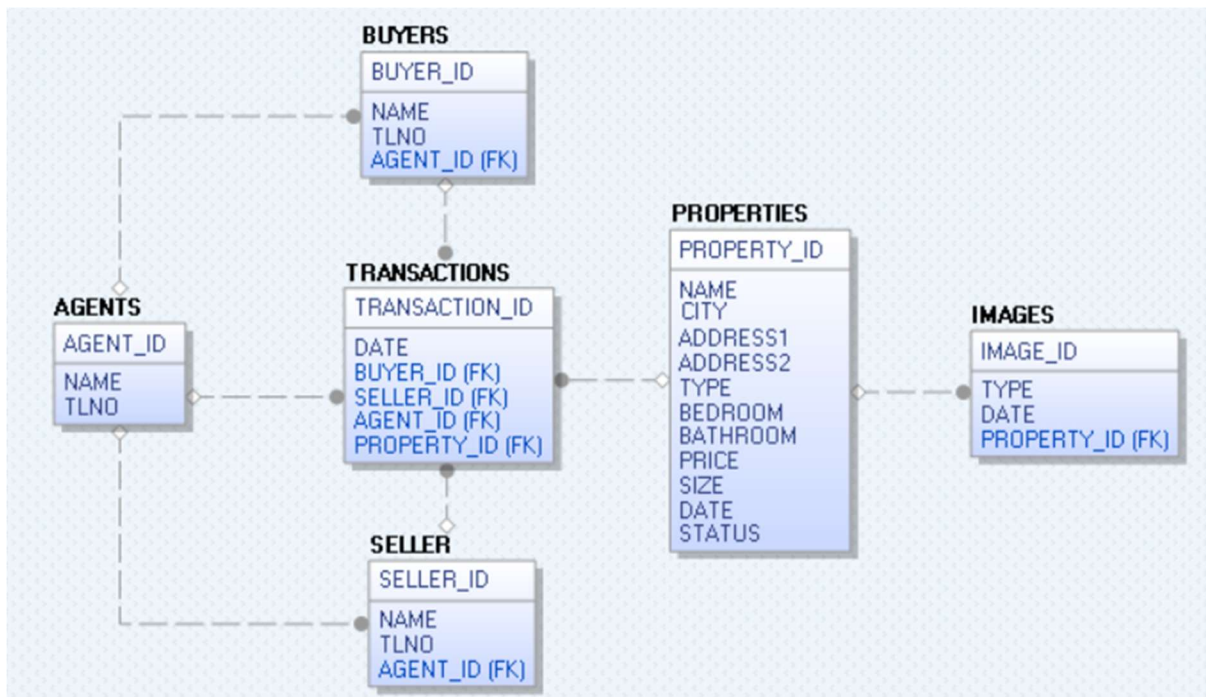
20181360 함형원

0. 개요

프로젝트 1에서 설계한 relational schema에 BCNF를 하고, 얻은 physical schema diagram을 통해 ODBC를 수행한다.

1. Schema Diagram

프로젝트 1에서 작성한 logical schema diagram은 다음과 같다. E-R model을 relation schema diagram으로 변환한다. 모든 관계가 one-to-many(total participation)이므로 many 측 entity set에 one 측 primary key를 추가한다. 추가한 primary key는 foreign key이다. relational schema diagram에서 검은색 점은 many, 흰색 점은 one을 의미한다.



각 table에 대한 설명은 다음과 같다.

1) BUYERS(BUYER_ID(PK), NAME, TLNO, AGENT_ID(FK))

부동산을 구매하는 사람을 의미한다. BUYER_ID는 구매인의 ID로 primary key이며, NAME은 구매인의 이름, TLNO는 구매인의 전화번호를 의미한다. AGENT_ID는 foreign key로 AGENTS와 BUYERS에 대해 1:N의 관계를 가진다.

2) SELLERS(SELLER_ID(PK), NAME, TLNO, AGENT_ID(FK))

부동산을 판매하는 사람을 의미한다. SELLER_ID는 판매인의 ID로 primary key이며, NAME은 판매인의 이름, TLNO는 판매인의 전화번호를 의미한다. AGENT_ID는 foreign key로 AGENTS와 SELLERS에 대해 1:N의 관계를 가진다.

3) AGENTS(AGENTS_ID(PK), NAME, TLNO)

부동산 구매인과 판매인의 중개인을 의미한다. AGENT_ID는 중개인의 ID로 primary key이며, NAME은 중개인의 이름, TLNO는 중개인의 전화번호를 의미한다.

4) PROPERTIES(PROPERTY_ID(PK), NAME, CITY, ADDRESS1, ADDRESS2, TYPE, BEDROOM, BATHROOM, PRICE, SIZE, DATE, STATUS)

부동산의 정보를 의미한다. PROPERTY_ID는 부동산의 ID로 primary key이다. CITY는 구(ex. 마포구), ADDRESS1은 구를 제외한 도로명 주소, ADDRESS2는 상세 주소를 의미한다. TYPE은 부동산의 종류(ex. studio, one-bedroom apartments), BEDROOM은 침실의 개수, BATHROOM은 욕실의 개수, PRICE는 부동산의 가격, DATE는 부동산의 등록일을 의미한다. STATUS는 판매 중 또는 거래 완료를 의미한다.

5) IMAGES(IMAGE_ID(PK), TYPE, DATE, PROPERTY_ID(FK))

사진이나 평면도를 의미한다. IMAGE_ID는 이미지의 ID로 primary key이다. TYPE은 이미지의 종류로 실내 사진, 실외 사진, 평면도가 있다. DATE는 이미지의 등록일을 의미한다. PROPERTY_ID는 foreign key로 IMAGES와 PROPERTIES에 대해 1:N의 관계를 가진다.

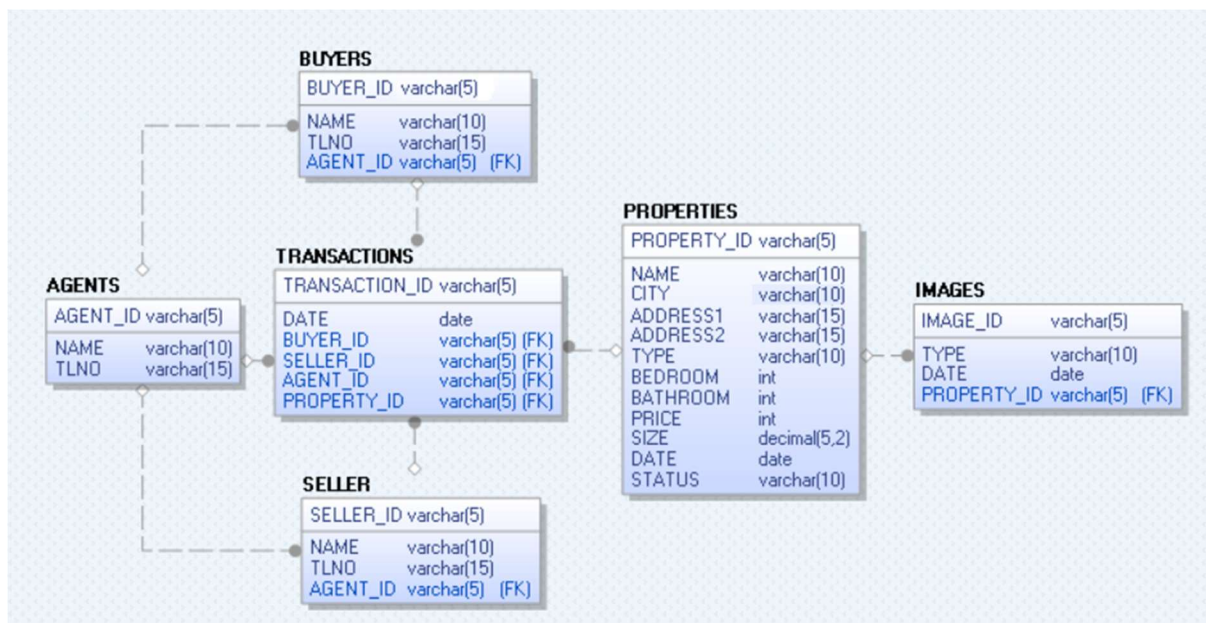
6) TRANSACTIONS(TRANSACTION_ID(PK), DATE, AGENT_ID(FK), BUYER_ID(FK), SELLER_ID(FK), PROPERTY_ID(FK))

부동산의 거래를 의미한다. TRANSACTION_ID는 이미지의 ID로 primary key이다. DATE는 거래일을 의미한다. AGENT_ID는 foreign key로 AGENTS와 TRANSACTIONS에 대해 1:N의 관계를 가진다. BUYER_ID는 foreign key로 BUYERS와 TRANSACTIONS에 대해 1:N의 관계를 가진다. SELLER_ID는 foreign key로 SELLER와 TRANSACTION에 대해 1:N의 관계를 가진다. PROPERTY_ID는 foreign key로 PROPERTIES와 TRANSACTIONS에 대해 1:N의 관계를 가진다.

각 table에서 functional dependency를 확인하자. BUYERS에서 BUYER_ID → NAME, TLNO, AGENT_ID이므로, BUYERS는 이미 BCNF를 만족한다. SELLERS에서 SELLER_ID → NAME, TLNO, AGENT_ID이므로, SELLERS는 이미 BCNF를 만족한다. AGENTS에서 AGENT_ID → NAME, TLNO이므로, AGENTS는 이미 BCNF를 만족한다. PROPERTIES에서 PROPERTY_ID → NAME, CITY, ADDRESS1, ADDRESS2, TYPE, BEDROOM, BATHROOM, PRICE, SIZE, DATE, STATUS이므로, PROPERTIES는 이미 BCNF를 만족한다. IMAGES에서 IMAGE_ID → TYPE, DATE, PROPERTY_ID이므로, IMAGES는 이미 BCNF를 만족한다. TRANSACTIONS에서 TRANSACTION_ID → DATE, AGENT_ID, BUYER_ID, SELLER_ID,

PROPERTY_ID이므로, TRANSACTIONS는 이미 BCNF를 만족한다. 따라서 모든 table이 BCNF를 만족하므로 추가적인 BCNF decomposition이 필요하지 않다.

logical schema diagram에 data type을 추가하여 physical schema diagram을 설계한다. physical schema diagram은 다음과 같다. E-R model을 relation schema diagram으로 변환한다. 모든 관계가 one-to-many(total participation)이므로 many 측 entity set에 one 측 primary key를 추가한다. 추가한 primary key는 foreign key이다. relational schema diagram에서 검은색 점은 many, 흰색 점은 one을 의미한다.



각 table에 대해 설명하겠다.

1) AGENTS

AGENTS table 생성은 다음과 같다.

```
CREATE TABLE AGENTS (AGENT_ID VARCHAR(5) NOT NULL, NAME VARCHAR(10) NOT NULL, TLNO VARCHAR(15) NOT NULL, PRIMARY KEY (AGENT_ID));
```

AGENTS의 primary key는 agent_id이다. 모든 속성은 NULL이 허용되지 않고, VARCHAR을 사용했다.

2) PROPERTIES

PROPERTIES table 생성은 다음과 같다.

```
CREATE TABLE PROPERTIES (PROPERTY_ID VARCHAR(5) NOT NULL, NAME VARCHAR(10) NOT NULL, CITY VARCHAR(10) NOT NULL, ADDRESS1 VARCHAR(15) NOT NULL, ADDRESS2 VARCHAR(15) NOT NULL, TYPE VARCHAR(10) NOT NULL, BEDROOM INT NOT NULL, BATHROOM INT NOT NULL, PRICE INT NOT NULL, SIZE DECIMAL(5,2) NOT NULL, DATE DATE NOT NULL, STATUS VARCHAR(10) NOT NULL, PRIMARY KEY (PROPERTY_ID));
```

PROPERTIES의 primary key는 property_id이다. 모든 속성은 NULL이 허용되지 않는다. property_id,

name, city, address1, address2, type, status는 VARCHAR을 사용했다. type에는 'house', 'apt1', 'aptN', 'studio'가 있고, status에는 'on sale', 'sold out'이 있다. bedroom, bathroom, price는 INT를 사용했다. price의 단위는 ₩10,000이다. size는 DECIMAL을 사용했고, date는 DATE를 사용했다.

3) BUYERS

BUYERS table 생성은 다음과 같다.

```
CREATE TABLE BUYERS (BUYER_ID VARCHAR(5) NOT NULL, NAME VARCHAR(10) NOT NULL, TLNO VARCHAR(15) NOT NULL, AGENT_ID VARCHAR(5) NOT NULL, PRIMARY KEY (BUYER_ID));  
ALTER TABLE BUYERS ADD CONSTRAINT AIDB FOREIGN KEY (AGENT_ID) REFERENCES AGENTS(AGENT_ID) ON DELETE CASCADE;
```

BUYERS의 primary key는 buyer_id이다. 모든 속성은 NULL이 허용되지 않고, VARCHAR을 사용했다. foreign key는 agent_id이며 AGENTS의 agent_id를 참조한다.

4) SELLERS

SELLERS table 생성은 다음과 같다.

```
CREATE TABLE SELLERS (SELLER_ID VARCHAR(5) NOT NULL, NAME VARCHAR(10) NOT NULL, TLNO VARCHAR(15) NOT NULL, AGENT_ID VARCHAR(5) NOT NULL, PRIMARY KEY (SELLER_ID));  
ALTER TABLE SELLERS ADD CONSTRAINT AIDS FOREIGN KEY (AGENT_ID) REFERENCES AGENTS(AGENT_ID) ON DELETE CASCADE;
```

SELLERS의 primary key는 seller_id이다. 모든 속성은 NULL이 허용되지 않고, VARCHAR을 사용했다. foreign key는 agent_id이며 AGENTS의 agent_id를 참조한다.

5) IMAGES

IMAGES table 생성은 다음과 같다.

```
CREATE TABLE IMAGES (IMAGE_ID VARCHAR(5) NOT NULL, TYPE VARCHAR(10) NOT NULL, DATE DATE NOT NULL, PROPERTY_ID VARCHAR(5) NOT NULL, PRIMARY KEY (IMAGE_ID));  
ALTER TABLE IMAGES ADD CONSTRAINT PIDI FOREIGN KEY (PROPERTY_ID) REFERENCES PROPERTIES(PROPERTY_ID) ON DELETE CASCADE;
```

IMAGES의 primary key는 image_id이다. 모든 속성은 NULL이 허용되지 않는다. image_id, type, property_id는 VARCHAR을 사용했다. type에는 plan(평면도), exterior, interior가 있다. date는 date를 사용했다. foreign key는 property_id이며 PROPERTIES의 property_id를 참조한다.

6) TRANSACTIONS

TRANSACTIONS table 생성은 다음과 같다.

```
CREATE TABLE TRANSACTIONS (TRANSACTION_ID VARCHAR(5) NOT NULL, DATE DATE NOT NULL, BUYER_ID VARCHAR(5) NOT NULL, SELLER_ID VARCHAR(5) NOT NULL, AGENT_ID VARCHAR(5) NOT NULL, PROPERTY_ID VARCHAR(5) NOT NULL, PRIMARY KEY (TRANSACTION_ID));  
ALTER TABLE TRANSACTIONS ADD CONSTRAINT BID FOREIGN KEY (BUYER_ID) REFERENCES BUYERS(BUYER_ID) ON DELETE CASCADE;  
ALTER TABLE TRANSACTIONS ADD CONSTRAINT SID FOREIGN KEY (SELLER_ID) REFERENCES
```

```

SELLERS(SELLER_ID) ON DELETE CASCADE;
ALTER TABLE TRANSACTIONS ADD CONSTRAINT AIDT FOREIGN KEY (AGENT_ID) REFERENCES AGENTS(AGENT_ID)
ON DELETE CASCADE;
ALTER TABLE TRANSACTIONS ADD CONSTRAINT PIDT FOREIGN KEY (PROPERTY_ID) REFERENCES
PROPERTIES(PROPERTY_ID) ON DELETE CASCADE;

```

TRANSACTIONS의 primary key는 transaction_id이다. 모든 속성은 NULL이 허용되지 않는다. transaction_id, buyer_id, seller_id, agent_id, property_id는 VARCHAR을 사용했다. date는 date를 사용했다. foreign key는 buyer_id, seller_id, agent_id, property_id이며, 각각 BUYERS의 buyer_id, SELLERS의 seller_id, AGENTS의 agent_id, PROPERTIES의 property_id를 참조한다.

2. ODBC Implementation

설계한 schema를 통해 명세서에 있는 쿼리를 작성한다. 프로그램을 실행하면 모든 table이 생성되며 조회된다.

[AGENTS]		
ID	이름	전화번호
10001	AA	010-1111-1111
10002	AB	010-1111-1112
10003	AC	010-1111-1113
10004	AD	010-1111-1114
10005	AE	010-1111-1115
10006	AF	010-1111-1116
10007	AG	010-1111-1117
10008	AH	010-1111-1118
10009	AI	010-1111-1119
10010	AJ	010-1111-1120
10011	AK	010-1111-1121
10012	AL	010-1111-1122
10013	AM	010-1111-1123
10014	AN	010-1111-1124

[PROPERTIES]											
ID	이름	구	동	상세주소	유형	침실	화장실	가격	크기	등록일	상태
40001	BA	MAPO	MPA	111-11	HOUSE	3	1	90000	10.00	2022-01-01	ON SALE
40002	BB	MAPO	MPA	111-12	HOUSE	3	1	100000	10.00	2022-01-02	ON SALE
40003	B	MAPO	MPA	111-13	HOUSE	3	1	150000	10.00	2022-01-03	ON SALE
40004	BD	MAPO	MPA	111-14	HOUSE	3	1	160000	10.00	2022-01-04	ON SALE
40005	BE	MAPO	MPA	111-15	HOUSE	3	1	100000	10.00	2022-01-05	SOLD OUT
40006	BFA	GANGNAM	GNA	211-11	APT1	1	1	200000	10.00	2023-01-06	ON SALE
40007	BGA	GANGNAM	GNA	211-12	APTN	4	1	300000	10.00	2023-01-07	ON SALE
40008	BHA	GANGNAM	GNA	211-13	STUDIO	3	2	300000	10.00	2023-01-08	ON SALE
40009	BIA	GANGNAM	GNA	211-14	STUDIO	4	2	300000	10.00	2023-01-09	ON SALE
40010	BJA	GANGNAM	GNA	211-15	STUDIO	4	2	300000	10.00	2023-01-10	SOLD OUT
40016	BFB	SEOCHO	SCA	212-11	APT1	1	1	400000	10.00	2023-01-06	ON SALE
40017	BGB	SEOCHO	SCA	212-12	APTN	4	1	600000	10.00	2023-01-07	ON SALE
40018	BHB	SEOCHO	SCA	212-13	STUDIO	3	2	300000	10.00	2023-01-08	ON SALE
40019	BIB	SEOCHO	SCA	212-14	APTN	4	2	300000	10.00	2023-01-09	ON SALE
40020	BJB	SEOCHO	SCA	212-15	STUDIO	4	2	600000	10.00	2023-01-10	SOLD OUT
40021	BKA	GANGDONG	GDA	311-11	STUDIO	4	2	100000	10.00	2021-01-11	SOLD OUT
40022	BKB	GANGDONG	GDA	311-11	STUDIO	4	2	110000	10.00	2022-01-11	SOLD OUT
40023	BKC	GANGDONG	GDA	311-11	STUDIO	4	2	120000	10.00	2023-01-11	SOLD OUT
40024	BKD	GANGDONG	GDA	311-12	STUDIO	4	2	130000	10.00	2021-01-11	SOLD OUT
40025	BKE	GANGDONG	GDA	311-12	STUDIO	4	2	140000	10.00	2022-01-11	SOLD OUT
40026	BKF	GANGDONG	GDA	311-12	STUDIO	4	2	150000	10.00	2023-01-11	SOLD OUT
40027	BLA	GANGSEO	GSA	411-11	STUDIO	4	2	160000	10.00	2021-01-11	SOLD OUT
40028	BLB	GANGSEO	GSA	411-12	STUDIO	4	2	170000	10.00	2021-01-11	SOLD OUT
40029	BLC	GANGSEO	GSA	411-13	STUDIO	4	2	180000	10.00	2021-01-11	SOLD OUT
40030	BLD	GANGSEO	GSA	411-14	STUDIO	4	2	190000	10.00	2021-01-11	SOLD OUT
40031	BLE	GANGSEO	GSA	411-15	STUDIO	4	2	200000	10.00	2021-01-11	SOLD OUT
40032	BLF	GANGSEO	GSA	411-16	STUDIO	4	2	210000	10.00	2021-01-11	SOLD OUT
40033	BLG	GANGSEO	GSA	411-17	STUDIO	4	2	220000	10.00	2021-01-11	SOLD OUT
40034	BLH	GANGSEO	GSA	411-18	STUDIO	4	2	230000	10.00	2021-01-11	SOLD OUT
40035	BLI	GANGSEO	GSA	411-19	STUDIO	4	2	240000	10.00	2021-01-11	SOLD OUT

[BUYERS]			
ID	이름	전화번호	중개인
20001	CA	010-2222-1111	10001
20002	CB	010-2222-1112	10002
20003	CC	010-2222-1113	10003
20004	CD	010-2222-1114	10004
20005	CE	010-2222-1115	10005

[SELLERS]			
ID	이름	전화번호	중개인
30001	DA	010-3333-1111	10001
30002	DB	010-3333-1112	10002
30003	DC	010-3333-1113	10003
30004	DD	010-3333-1114	10004
30005	DE	010-3333-1115	10005

[IMAGES]

ID	유형	촬영일	부동산
50001	PLAN	2022-01-01	40001
50002	EXTERIOR	2022-01-01	40001
50003	PLAN	2022-01-01	40001
50004	EXTERIOR	2022-01-01	40002
50005	PLAN	2022-01-01	40001
50006	EXTERIOR	2022-01-01	40003
50007	PLAN	2022-01-01	40001
50008	EXTERIOR	2022-01-01	40004
50009	PLAN	2022-01-01	40001
50010	EXTERIOR	2022-01-01	40005
50011	INTERIOR	2022-01-01	40006
50012	PLAN	2022-01-01	40007
50013	EXTERIOR	2022-01-01	40007
50014	INTERIOR	2022-01-01	40008
50015	INTERIOR	2022-01-01	40009
50016	INTERIOR	2022-01-01	40010
50017	INTERIOR	2022-01-01	40016
50018	PLAN	2022-01-01	40017
50019	EXTERIOR	2022-01-01	40017
50020	INTERIOR	2022-01-01	40018
50021	INTERIOR	2022-01-01	40019
50022	INTERIOR	2022-01-01	40020
50023	INTERIOR	2022-01-01	40021
50024	INTERIOR	2022-01-01	40022
50025	INTERIOR	2022-01-01	40023
50026	INTERIOR	2022-01-01	40024
50027	INTERIOR	2022-01-01	40025
50028	INTERIOR	2022-01-01	40026
50029	INTERIOR	2022-01-01	40027
50030	INTERIOR	2022-01-01	40028
50031	INTERIOR	2022-01-01	40029
50032	INTERIOR	2022-01-01	40030
50033	INTERIOR	2022-01-01	40031
50034	INTERIOR	2022-01-01	40032
50035	INTERIOR	2022-01-01	40033
50036	INTERIOR	2022-01-01	40034
50037	INTERIOR	2022-01-01	40035

[TRANSACTIONS]					
ID	거래일	구매인	판매인	중개인	부동산
60001	2022-01-06	20001	30001	10001	40005
60002	2023-01-12	20002	30002	10002	40010
60003	2023-01-13	20003	30003	10003	40020
60004	2021-01-15	20004	30004	10004	40021
60005	2022-01-16	20004	30004	10004	40022
60006	2023-01-17	20004	30004	10004	40023
60007	2021-01-18	20004	30004	10005	40024
60008	2022-01-19	20004	30004	10005	40025
60009	2023-01-20	20004	30004	10005	40026
60010	2021-01-21	20004	30004	10006	40027
60011	2021-01-22	20004	30004	10007	40028
60012	2021-01-23	20004	30004	10008	40029
60013	2021-01-24	20004	30004	10009	40030
60014	2021-01-25	20004	30004	10010	40031
60015	2021-01-26	20004	30004	10011	40032
60016	2021-01-27	20004	30004	10012	40033
60017	2021-01-28	20004	30004	10013	40034
60018	2021-01-29	20004	30004	10014	40035

다음으로, 0~7까지 입력을 통해 type을 선택할 수 있다.

```
[SELECT QUERY TYPE]
1. TYPE 1
2. TYPE 2
3. TYPE 3
4. TYPE 4
5. TYPE 5
6. TYPE 6
7. TYPE 7
0. QUIT
```

0이나 나머지를 입력하면 프로그램이 종료된다. 이때, 모든 table이 drop된다.

```
[SELECT QUERY TYPE]
1. TYPE 1
2. TYPE 2
3. TYPE 3
4. TYPE 4
5. TYPE 5
6. TYPE 6
7. TYPE 7
0. QUIT
0

C:\Users\User\Desktop\Project2\x64\Debug\Project2.exe(프로세스 17772개)이(가) 종료되었습니다(코드: 0개).
```

type이 입력되면 subtype에 관계없이 반드시 쿼리가 출력된다. subtype은 올바른 입력만 처리되며, 나머지 입력은 초기 화면으로 복귀한다. 쿼리가 출력되면 초기 화면으로 복귀한다.

1) type 1

type 1은 city가 'MAPO', status가 'ON SALE'인 tuple을 찾는 쿼리이다.

```
query = "SELECT CITY, ADDRESS1, ADDRESS2 FROM PROPERTIES WHERE CITY = 'MAPO'";
```

city, address1, address2를 조회하여 부동산의 전체 주소를 확인할 수 있다.

```
[TYPE 1]
Find address of homes for sale in the district 'MAPO'.
주소
MAPO MPA 111-11
MAPO MPA 111-12
MAPO MPA 111-13
MAPO MPA 111-14
MAPO MPA 111-15
```

type 1-1은 ₩1,000,000,000~₩1,500,000,000인 type 1을 찾는 쿼리이다. price의 단위는 ₩10,000인 것을 유의하자. type 1에 price >= 100000 and price <= 150000를 추가한다.

```
query = "SELECT CITY, ADDRESS1, ADDRESS2 FROM PROPERTIES WHERE CITY = 'MAPO' AND PRICE >= 100000
AND PRICE <= 150000 AND STATUS = 'ON SALE'";
```

city, address1, address2를 조회하여 부동산의 전체 주소를 확인할 수 있다.

```
[SELECT QUERY SUBTYPE]
1. TYPE 1-1
1

[TYPE 1-1]
Find address of homes for sale in the district 'MAPO' costing between ₩1,000,000,000 and ₩1,500,000,000.
주소
MAPO MPA 111-12
MAPO MPA 111-13
```

2) type 2

type 2는 city가 'GANGNAM' 또는 'SEOCHO', status가 'ON SALE'인 tuple을 찾는 쿼리이다. 8학군은 강남구, 서초구이다.

```
query = "SELECT CITY, ADDRESS1, ADDRESS2 FROM PROPERTIES WHERE CITY IN ('GANGNAM', 'SEOCHO') AND
STATUS = 'ON SALE'";
```

city, address1, address2를 조회하여 부동산의 전체 주소를 확인할 수 있다.

```
[TYPE 2]
Find the address of homes for sale in the 8th school district.
주소
GANGNAM GNA 211-11
GANGNAM GNA 211-12
GANGNAM GNA 211-13
GANGNAM GNA 211-14
SEOCHO SCA 212-11
SEOCHO SCA 212-12
SEOCHO SCA 212-13
SEOCHO SCA 212-14
```

type 2-1은 침실이 4개 이상, 화장실이 2개인 type 2를 찾는 쿼리이다. type 2에 bedroom >= 4 and bathroom = 2를 추가한다.

```
query = "SELECT CITY, ADDRESS1, ADDRESS2 FROM PROPERTIES WHERE CITY IN ('GANGNAM', 'SEOCHO') AND
BEDROOM >= 4 AND BATHROOM = 2 AND STATUS = 'ON SALE'";
```

city, address1, address2를 조회하여 부동산의 전체 주소를 확인할 수 있다.

```
[TYPE 2-1]
Then find properties with 4 or more bedrooms and 2 bathrooms.
주소
GANGNAM GNA 211-14
SEOCHO SCA 212-14
```

3) type 3

type 3은 2022년에 가장 많은 부동산을 판매한 중개인을 찾는 쿼리이다. 부동산의 수(the most properties)가 기준인지, 판매액(by total won value)이 기준인지 명확하지 않으므로 후자를 선택하여 쿼리를 작성했다.

```
query = "SELECT NAMEW
FROM AGENTSW
WHERE AGENT_ID = (SELECT AGENT_IDW
FROM PROPERTIES PW
LEFT JOIN TRANSACTIONS TW
ON P.PROPERTY_ID = T.PROPERTY_IDW
WHERE YEAR(T.DATE) = 2022 AND STATUS = 'SOLD OUT'W
GROUP BY AGENT_IDW
ORDER BY SUM(PRICE) DESCW
LIMIT 1)";
```

PROPERTIES와 TRANSACTIONS을 조인하고 agent_id를 기준으로 그룹화하여 sum(price)를 기준으로 내림차순 정렬했다. limit 1을 통해 가장 많은 부동산을 판매한 중개인을 찾을 수 있다. name을 조회하여 해당하는 중개인의 이름을 확인할 수 있다.

```
[TYPE 3]
Find the name of the agent who has sold the most properties in the year 2022 by total won value.
이름
AE
```

type 3-1는 2023년에 많은 부동산을 판매한 상위 k 중개인을 찾는 쿼리이다. 마찬가지로, 금액을 기준으로 쿼리를 작성했다.

```
query = "SELECT AGENT_IDW
        FROM PROPERTIES PW
        LEFT JOIN TRANSACTIONS TW
        ON P.PROPERTY_ID = T.PROPERTY_IDW
        WHERE YEAR(T.DATE) = 2023 AND STATUS = 'SOLD OUT'W
        GROUP BY AGENT_IDW
        ORDER BY SUM(PRICE) DESC";
```

PROPERTIES와 TRANSACTIONS을 조인하고 agent_id를 기준으로 그룹화하여 sum(price)를 기준으로 내림차순 정렬했다. k에 대한 입력을 받고, while loop에서 k만큼 tuple을 출력하도록 구현했다.

```
while ((sql_row = mysql_fetch_row(sql_result)) != NULL) {
    printf("%sWn", sql_row[0]);
    cnt += 1;
    if (cnt == k)
        break;
}
```

agent_id를 조회하여 해당하는 중개인을 확인할 수 있다.

```
[TYPE 3-1]
Then find the top k agents in the year 2023 by total won value.
k: 2
ID
10003
10002
```

type 3-2는 2021년에 많은 부동산을 판매한 상위 10% 중개인을 찾는 쿼리이다. 마찬가지로, 금액을 기준으로 쿼리를 작성했다.

```
query = "WITH TEMP AS (W
        SELECT AGENT_ID, SUM(PRICE), PERCENT_RANK() OVER (ORDER BY
SUM(PRICE)) AS PERCENTILE_RANKW
        FROM TRANSACTIONS TW
        LEFT JOIN PROPERTIES PW
        ON P.PROPERTY_ID = T.PROPERTY_IDW
        WHERE YEAR(T.DATE) = 2021W
        GROUP BY AGENT_IDW
    )W
    SELECT AGENT_IDW
    FROM TEMPW
    WHERE PERCENTILE_RANK <= 0.1";
```

PROPERTIES와 TRANSACTIONS을 조인하고 sum(price)를 기준으로 percent_rank를 사용하여 백분위를 계산할 수 있다. 백분위를 포함한 temp에서 하위 10%에 해당하는 중개인을 찾는다.

agent_id를 조회하여 해당하는 중개인을 확인할 수 있다.

```
[TYPE 3-2]
And then find the bottom 10% agents in the year 2021 by total won value.
ID
10004
10005
```

4) type 4

type 4는 2022년을 기준으로 각 중개인에 대해 평균 판매액, 평균 거래일을 찾는 쿼리이다. 거래일은 PROPERTIES의 date와 TRANSACTIONS의 date의 차이를 통해 계산할 수 있다.

```
query = "SELECT AGENT_ID, AVG(PRICE), AVG(DATEDIFF(T.DATE, P.DATE))W
        FROM TRANSACTIONS TW
        LEFT JOIN PROPERTIES PW
        ON T.PROPERTY_ID = P.PROPERTY_IDW
        WHERE YEAR(T.DATE) = 2022W
        GROUP BY AGENT_ID";
```

PROPERTIES와 TRANSACTIONS를 조인하고, agent_id를 기준으로 그룹화한다. agent_id, avg(price), avg(datediff(t.date, p.date))를 조회하여 해당하는 정보를 확인할 수 있다. 평균 판매액의 단위는 ₩10,000이다.

```
[TYPE 4]
For each agent, compute the average selling price of properties sold in 2022, and the average time the property was on the market.
중개인 | 평균 판매액 | 평균 거래일
10001 | 100000.0000 | 1.0000
10004 | 110000.0000 | 5.0000
10005 | 140000.0000 | 8.0000
```

type 4-1은 2023년을 기준으로 각 중개인에 대해 최고 판매액을 찾는 쿼리이다.

```
query = "SELECT AGENT_ID, MAX(PRICE)W
        FROM TRANSACTIONS TW
        LEFT JOIN PROPERTIES PW
        ON T.PROPERTY_ID = P.PROPERTY_IDW
        WHERE YEAR(T.DATE) = 2023W
        GROUP BY AGENT_ID";
```

PROPERTIES와 TRANSACTIONS를 조인하고, agent_id를 기준으로 그룹화한다. agent_id, max(price)를 조회하여 해당하는 정보를 확인할 수 있다. 최고 판매액의 단위는 ₩10,000이다.

```
[TYPE 4-1]
Then compute the maximum selling price of properties sold in 2023 for each agent.
중개인 | 최고 판매액
10002 | 300000
10003 | 600000
10004 | 120000
10005 | 150000
```

type 4-2는 각 중개인에 대해 최장 거래일을 찾는 쿼리이다.

```
query = "SELECT AGENT_ID, MAX(DATEDIFF(T.DATE, P.DATE))W
```

```

FROM TRANSACTIONS TW
LEFT JOIN PROPERTIES PW
ON T.PROPERTY_ID = P.PROPERTY_IDW
GROUP BY AGENT_ID";

```

PROPERTIES와 TRANSACTIONS를 조인하고, agent_id를 기준으로 그룹화한다. agent_id, max(datediff(t.date, p.date))를 조회하여 해당하는 정보를 확인할 수 있다.

```

[TYPE 4-2]
And then compute the longest time the property was on the market for each agent.
중개인 | 최장 거래일
10001 | 1
10002 | 2
10003 | 3
10004 | 6
10005 | 9
10006 | 10
10007 | 11
10008 | 12
10009 | 13
10010 | 14
10011 | 15
10012 | 16
10013 | 17
10014 | 18

```

5) type 5

type 5는 가장 비싼 'studio', 'apt1', 'aptN', 'house'의 이미지를 찾는 쿼리이다.

```

query = "SELECT I.IMAGE_ID, P.TYPEW
        FROM IMAGES IW
        LEFT JOIN PROPERTIES PW
        ON I.PROPERTY_ID = P.PROPERTY_IDW
        WHERE (P.TYPE, P.PRICE) IN (SELECT TYPE, MAX(PRICE)W
                                   FROM PROPERTIESW
                                   GROUP BY TYPE)";

```

서브쿼리에서 type을 기준으로 그룹화하고, type과 최고 price를 조회한다. PROPERTIES와 IMAGES를 조인하여 PROPERTIES의 type과 price가 서브쿼리에 있는지 확인한다. IMAGES의 image_id, PROPERTIES의 type을 조회하여 해당하는 정보를 확인할 수 있다.

```

[TYPE 5]
Show photos of the most expensive studio, one - bedroom, multi - bedroom apartment(s), and detached house(s), respectively, from the database.
이미지 | 타입
50008 | HOUSE
50017 | APT1
50018 | APTN
50019 | APTN
50022 | STUDIO

```

6) type 6

type 6를 찾는 쿼리는 간단하게 TRANSACTIONS에서 찾을 수 있다.

```

query = "SELECT PROPERTY_ID, DATE, BUYER_ID, SELLER_ID, AGENT_ID FROM TRANSACTIONS";

```

```
[TYPE 6]
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.
부동산 | 날짜 | 구매인 | 판매인 | 중개인
40005 | 2022-01-06 | 20001 | 30001 | 10001
40010 | 2023-01-12 | 20002 | 30002 | 10002
40020 | 2023-01-13 | 20003 | 30003 | 10003
40021 | 2021-01-15 | 20004 | 30004 | 10004
40022 | 2022-01-16 | 20004 | 30004 | 10004
40023 | 2023-01-17 | 20004 | 30004 | 10004
40024 | 2021-01-18 | 20004 | 30004 | 10005
40025 | 2022-01-19 | 20004 | 30004 | 10005
40026 | 2023-01-20 | 20004 | 30004 | 10005
40027 | 2021-01-21 | 20004 | 30004 | 10006
40028 | 2021-01-22 | 20004 | 30004 | 10007
40029 | 2021-01-23 | 20004 | 30004 | 10008
40030 | 2021-01-24 | 20004 | 30004 | 10009
40031 | 2021-01-25 | 20004 | 30004 | 10010
40032 | 2021-01-26 | 20004 | 30004 | 10011
40033 | 2021-01-27 | 20004 | 30004 | 10012
40034 | 2021-01-28 | 20004 | 30004 | 10013
40035 | 2021-01-29 | 20004 | 30004 | 10014
```

7) type 7

type 7은 AGENTS에 새로운 중개인을 추가하는 쿼리이다. 중개인의 agent_id, name, tlno를 입력 받아 sprintf를 통해 쿼리를 구성한다.

```
printf("ID: ");
scanf("%s", aID);
printf("이름: ");
scanf("%s", aNAME);
printf("전화번호: ");
scanf("%s", aTLNO);
sprintf(q, "INSERT INTO AGENTS VALUES('%s', '%s', '%s')", aID, aNAME, aTLNO);
```

AGENTS에 tuple을 추가하고 AGENTS table을 조회한다.

```
[TYPE 7]
Add a new agent to the database.
ID: 11111
이름: ZZ
전화번호: 010-1234-5678
[AGENTS]
ID | 이름 | 전화번호
10001 | AA | 010-1111-1111
10002 | AB | 010-1111-1112
10003 | AC | 010-1111-1113
10004 | AD | 010-1111-1114
10005 | AE | 010-1111-1115
10006 | AF | 010-1111-1116
10007 | AG | 010-1111-1117
10008 | AH | 010-1111-1118
10009 | AI | 010-1111-1119
10010 | AJ | 010-1111-1120
10011 | AK | 010-1111-1121
10012 | AL | 010-1111-1122
10013 | AM | 010-1111-1123
10014 | AN | 010-1111-1124
11111 | ZZ | 010-1234-5678
```

마지막으로, CRUD의 내용을 첨부한다.

```
CREATE DATABASE project;
USE project;
CREATE TABLE AGENTS (AGENT_ID VARCHAR(5) NOT NULL, NAME VARCHAR(10) NOT NULL, TLNO
VARCHAR(15) NOT NULL, PRIMARY KEY (AGENT_ID));
INSERT INTO AGENTS VALUES ('10001', 'AA', '010-1111-1111'), ('10002', 'AB', '010-1111-1112'),
('10003', 'AC', '010-1111-1113'), ('10004', 'AD', '010-1111-1114'), ('10005', 'AE', '010-1111-
1115'), ('10006', 'AF', '010-1111-1116'), ('10007', 'AG', '010-1111-1117'), ('10008', 'AH',
'010-1111-1118'), ('10009', 'AI', '010-1111-1119'), ('10010', 'AJ', '010-1111-1120'), ('10011',
'AK', '010-1111-1121'), ('10012', 'AL', '010-1111-1122'), ('10013', 'AM', '010-1111-1123'),
('10014', 'AN', '010-1111-1124');
CREATE TABLE PROPERTIES (PROPERTY_ID VARCHAR(5) NOT NULL, NAME VARCHAR(10) NOT NULL, CITY
VARCHAR(10) NOT NULL, ADDRESS1 VARCHAR(15) NOT NULL, ADDRESS2 VARCHAR(15) NOT NULL, TYPE
VARCHAR(10) NOT NULL, BEDROOM INT NOT NULL, BATHROOM INT NOT NULL, PRICE INT NOT NULL, SIZE
DECIMAL(5,2) NOT NULL, DATE DATE NOT NULL, STATUS VARCHAR(10) NOT NULL, PRIMARY KEY
(PROPERTY_ID));
INSERT INTO PROPERTIES VALUES ('40001', 'BA', 'MAPO', 'MPA', '111-11', 'HOUSE', 3, 1, 90000,
10.00, '2022-01-01', 'ON SALE'), ('40002', 'BB', 'MAPO', 'MPA', '111-12', 'HOUSE', 3, 1,
100000, 10.00, '2022-01-02', 'ON SALE'), ('40003', 'B', 'MAPO', 'MPA', '111-13', 'HOUSE', 3, 1,
150000, 10.00, '2022-01-03', 'ON SALE'), ('40004', 'BD', 'MAPO', 'MPA', '111-14', 'HOUSE', 3,
1, 160000, 10.00, '2022-01-04', 'ON SALE'), ('40005', 'BE', 'MAPO', 'MPA', '111-15', 'HOUSE',
3, 1, 100000, 10.00, '2022-01-05', 'SOLD OUT'), ('40006', 'BFA', 'GANGNAM', 'GNA', '211-11',
'APT1', 1, 1, 200000, 10.00, '2023-01-06', 'ON SALE'), ('40007', 'BGA', 'GANGNAM', 'GNA', '211-
12', 'APT1', 4, 1, 300000, 10.00, '2023-01-07', 'ON SALE'), ('40008', 'BHA', 'GANGNAM', 'GNA',
'211-13', 'STUDIO', 3, 2, 300000, 10.00, '2023-01-08', 'ON SALE'), ('40009', 'BIA', 'GANGNAM',
'GNA', '211-14', 'STUDIO', 4, 2, 300000, 10.00, '2023-01-09', 'ON SALE'), ('40010', 'BJA',
'GANGNAM', 'GNA', '211-15', 'STUDIO', 4, 2, 300000, 10.00, '2023-01-10', 'SOLD OUT'), ('40016',
```

```

'BFB', 'SEOCHO', 'SCA', '212-11', 'APT1', 1, 1, 400000, 10.00, '2023-01-06', 'ON SALE'),
('40017', 'BGB', 'SEOCHO', 'SCA', '212-12', 'APT1', 4, 1, 600000, 10.00, '2023-01-07', 'ON
SALE'), ('40018', 'BHB', 'SEOCHO', 'SCA', '212-13', 'STUDIO', 3, 2, 300000, 10.00, '2023-01-
08', 'ON SALE'), ('40019', 'BIB', 'SEOCHO', 'SCA', '212-14', 'APT1', 4, 2, 300000, 10.00,
'2023-01-09', 'ON SALE'), ('40020', 'BJB', 'SEOCHO', 'SCA', '212-15', 'STUDIO', 4, 2, 600000,
10.00, '2023-01-10', 'SOLD OUT'), ('40021', 'BKA', 'GANGDONG', 'GDA', '311-11', 'STUDIO', 4, 2,
100000, 10.00, '2021-01-11', 'SOLD OUT'), ('40022', 'BKB', 'GANGDONG', 'GDA', '311-11',
'STUDIO', 4, 2, 110000, 10.00, '2022-01-11', 'SOLD OUT'), ('40023', 'BKC', 'GANGDONG', 'GDA',
'311-11', 'STUDIO', 4, 2, 120000, 10.00, '2023-01-11', 'SOLD OUT'), ('40024', 'BKD',
'GANGDONG', 'GDA', '311-12', 'STUDIO', 4, 2, 130000, 10.00, '2021-01-11', 'SOLD OUT'),
('40025', 'BKE', 'GANGDONG', 'GDA', '311-12', 'STUDIO', 4, 2, 140000, 10.00, '2022-01-11',
'SOLD OUT'), ('40026', 'BKF', 'GANGDONG', 'GDA', '311-12', 'STUDIO', 4, 2, 150000, 10.00,
'2023-01-11', 'SOLD OUT'), ('40027', 'BLA', 'GANGSEO', 'GSA', '411-11', 'STUDIO', 4, 2, 160000,
10.00, '2021-01-11', 'SOLD OUT'), ('40028', 'BLB', 'GANGSEO', 'GSA', '411-12', 'STUDIO', 4, 2,
170000, 10.00, '2021-01-11', 'SOLD OUT'), ('40029', 'BLC', 'GANGSEO', 'GSA', '411-13',
'STUDIO', 4, 2, 180000, 10.00, '2021-01-11', 'SOLD OUT'), ('40030', 'BLD', 'GANGSEO', 'GSA',
'411-14', 'STUDIO', 4, 2, 190000, 10.00, '2021-01-11', 'SOLD OUT'), ('40031', 'BLE', 'GANGSEO',
'GSA', '411-15', 'STUDIO', 4, 2, 200000, 10.00, '2021-01-11', 'SOLD OUT'), ('40032', 'BLF',
'GANGSEO', 'GSA', '411-16', 'STUDIO', 4, 2, 210000, 10.00, '2021-01-11', 'SOLD OUT'), ('40033',
'BLG', 'GANGSEO', 'GSA', '411-17', 'STUDIO', 4, 2, 220000, 10.00, '2021-01-11', 'SOLD OUT'),
('40034', 'BLH', 'GANGSEO', 'GSA', '411-18', 'STUDIO', 4, 2, 230000, 10.00, '2021-01-11', 'SOLD
OUT'), ('40035', 'BLI', 'GANGSEO', 'GSA', '411-19', 'STUDIO', 4, 2, 240000, 10.00, '2021-01-
11', 'SOLD OUT');

```

```

CREATE TABLE BUYERS (BUYER_ID VARCHAR(5) NOT NULL, NAME VARCHAR(10) NOT NULL, TLNO
VARCHAR(15) NOT NULL, AGENT_ID VARCHAR(5) NOT NULL, PRIMARY KEY (BUYER_ID));
ALTER TABLE BUYERS ADD CONSTRAINT AIDB FOREIGN KEY (AGENT_ID) REFERENCES AGENTS(AGENT_ID) ON
DELETE CASCADE;
INSERT INTO BUYERS VALUES ('20001', 'CA', '010-2222-1111', '10001'), ('20002', 'CB', '010-2222-
1112', '10002'), ('20003', 'CC', '010-2222-1113', '10003'), ('20004', 'CD', '010-2222-1114',
'10004'), ('20005', 'CE', '010-2222-1115', '10005');
CREATE TABLE SELLERS (SELLER_ID VARCHAR(5) NOT NULL, NAME VARCHAR(10) NOT NULL, TLNO
VARCHAR(15) NOT NULL, AGENT_ID VARCHAR(5) NOT NULL, PRIMARY KEY (SELLER_ID));
ALTER TABLE SELLERS ADD CONSTRAINT AIDS FOREIGN KEY (AGENT_ID) REFERENCES AGENTS(AGENT_ID) ON
DELETE CASCADE;
INSERT INTO SELLERS VALUES ('30001', 'DA', '010-3333-1111', '10001'), ('30002', 'DB', '010-
3333-1112', '10002'), ('30003', 'DC', '010-3333-1113', '10003'), ('30004', 'DD', '010-3333-
1114', '10004'), ('30005', 'DE', '010-3333-1115', '10005');
CREATE TABLE IMAGES (IMAGE_ID VARCHAR(5) NOT NULL, TYPE VARCHAR(10) NOT NULL, DATE DATE
NOT NULL, PROPERTY_ID VARCHAR(5) NOT NULL, PRIMARY KEY (IMAGE_ID));
ALTER TABLE IMAGES ADD CONSTRAINT PIDI FOREIGN KEY (PROPERTY_ID) REFERENCES
PROPERTIES(PROPERTY_ID) ON DELETE CASCADE;
INSERT INTO IMAGES VALUES ('50001', 'PLAN', '2022-01-01', '40001'), ('50002', 'EXTERIOR',
'2022-01-01', '40001'), ('50003', 'PLAN', '2022-01-01', '40001'), ('50004', 'EXTERIOR', '2022-
01-01', '40002'), ('50005', 'PLAN', '2022-01-01', '40001'), ('50006', 'EXTERIOR', '2022-01-01',
'40003'), ('50007', 'PLAN', '2022-01-01', '40001'), ('50008', 'EXTERIOR', '2022-01-01',
'40004'), ('50009', 'PLAN', '2022-01-01', '40001'), ('50010', 'EXTERIOR', '2022-01-01',
'40005'), ('50011', 'INTERIOR', '2022-01-01', '40006'), ('50012', 'PLAN', '2022-01-01',
'40007'), ('50013', 'EXTERIOR', '2022-01-01', '40007'), ('50014', 'INTERIOR', '2022-01-01',
'40008'), ('50015', 'INTERIOR', '2022-01-01', '40009'), ('50016', 'INTERIOR', '2022-01-01',
'40010'), ('50017', 'INTERIOR', '2022-01-01', '40016'), ('50018', 'PLAN', '2022-01-01',
'40017'), ('50019', 'EXTERIOR', '2022-01-01', '40017'), ('50020', 'INTERIOR', '2022-01-01',
'40018'), ('50021', 'INTERIOR', '2022-01-01', '40019'), ('50022', 'INTERIOR', '2022-01-01',
'40020'), ('50023', 'INTERIOR', '2022-01-01', '40021'), ('50024', 'INTERIOR', '2022-01-01',
'40022'), ('50025', 'INTERIOR', '2022-01-01', '40023'), ('50026', 'INTERIOR', '2022-01-01',
'40024'), ('50027', 'INTERIOR', '2022-01-01', '40025'), ('50028', 'INTERIOR', '2022-01-01',

```



```
'40026'), ('50029', 'INTERIOR', '2022-01-01', '40027'), ('50030', 'INTERIOR', '2022-01-01', '40028'), ('50031', 'INTERIOR', '2022-01-01', '40029'), ('50032', 'INTERIOR', '2022-01-01', '40030'), ('50033', 'INTERIOR', '2022-01-01', '40031'), ('50034', 'INTERIOR', '2022-01-01', '40032'), ('50035', 'INTERIOR', '2022-01-01', '40033'), ('50036', 'INTERIOR', '2022-01-01', '40034'), ('50037', 'INTERIOR', '2022-01-01', '40035');
```

```
CREATE TABLE TRANSACTIONS (TRANSACTION_ID VARCHAR(5) NOT NULL, DATE DATE NOT NULL, BUYER_ID VARCHAR(5) NOT NULL, SELLER_ID VARCHAR(5) NOT NULL, AGENT_ID VARCHAR(5) NOT NULL, PROPERTY_ID VARCHAR(5) NOT NULL, PRIMARY KEY (TRANSACTION_ID));
```

```
ALTER TABLE TRANSACTIONS ADD CONSTRAINT BID FOREIGN KEY (BUYER_ID) REFERENCES BUYERS(BUYER_ID) ON DELETE CASCADE;
```

```
ALTER TABLE TRANSACTIONS ADD CONSTRAINT SID FOREIGN KEY (SELLER_ID) REFERENCES SELLERS(SELLER_ID) ON DELETE CASCADE;
```

```
ALTER TABLE TRANSACTIONS ADD CONSTRAINT AIDT FOREIGN KEY (AGENT_ID) REFERENCES AGENTS(AGENT_ID) ON DELETE CASCADE;
```

```
ALTER TABLE TRANSACTIONS ADD CONSTRAINT PIDT FOREIGN KEY (PROPERTY_ID) REFERENCES PROPERTIES(PROPERTY_ID) ON DELETE CASCADE;
```

```
INSERT INTO TRANSACTIONS VALUES ('60001', '2022-01-06', '20001', '30001', '10001', '40005'), ('60002', '2023-01-12', '20002', '30002', '10002', '40010'), ('60003', '2023-01-13', '20003', '30003', '10003', '40020'), ('60004', '2021-01-15', '20004', '30004', '10004', '40021'), ('60005', '2022-01-16', '20004', '30004', '10004', '40022'), ('60006', '2023-01-17', '20004', '30004', '10004', '40023'), ('60007', '2021-01-18', '20004', '30004', '10005', '40024'), ('60008', '2022-01-19', '20004', '30004', '10005', '40025'), ('60009', '2023-01-20', '20004', '30004', '10005', '40026'), ('60010', '2021-01-21', '20004', '30004', '10006', '40027'), ('60011', '2021-01-22', '20004', '30004', '10007', '40028'), ('60012', '2021-01-23', '20004', '30004', '10008', '40029'), ('60013', '2021-01-24', '20004', '30004', '10009', '40030'), ('60014', '2021-01-25', '20004', '30004', '10010', '40031'), ('60015', '2021-01-26', '20004', '30004', '10011', '40032'), ('60016', '2021-01-27', '20004', '30004', '10012', '40033'), ('60017', '2021-01-28', '20004', '30004', '10013', '40034'), ('60018', '2021-01-29', '20004', '30004', '10014', '40035');
```

```
$$$
```

```
DROP TABLE AGENTS;
```

```
DROP TABLE PROPERTIES;
```

```
DROP TABLE BUYERS;
```

```
DROP TABLE SELLERS;
```

```
DROP TABLE IMAGES;
```

```
DROP DATABASE project;
```