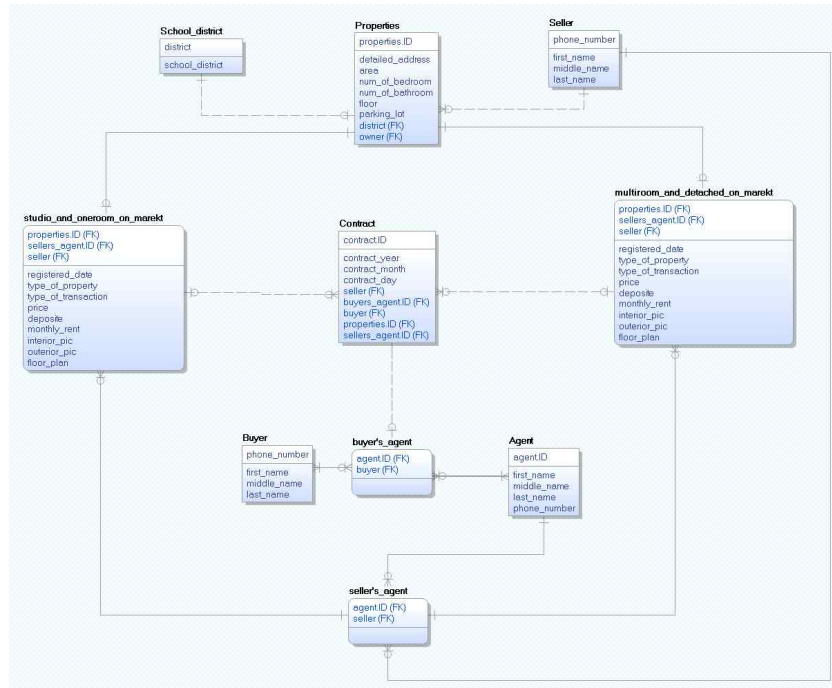


Database System Project 2 Report

20200977 한영태

1. BCNF Decomposition for Logical Schema



각 relation이 BCNF 조건을 만족하는지 확인 후 조건을 충족하지 못하는 relation에 대해 decomposition을 수행했다. 기존의 Properties relation에서 $\text{district} \rightarrow \text{school_district}$ 라는 Functional Dependency가 존재했다. 이는 non-trivial이면서 district가 primary key가 아니었다. 따라서 기존의 Properties를 Properties와 School_district(district, school_district)로 decompose하였다. 이외의 부분은 BCNF 조건을 만족하여 추가적인 decomposition을 진행하지 않았다.

2. Physical Schema

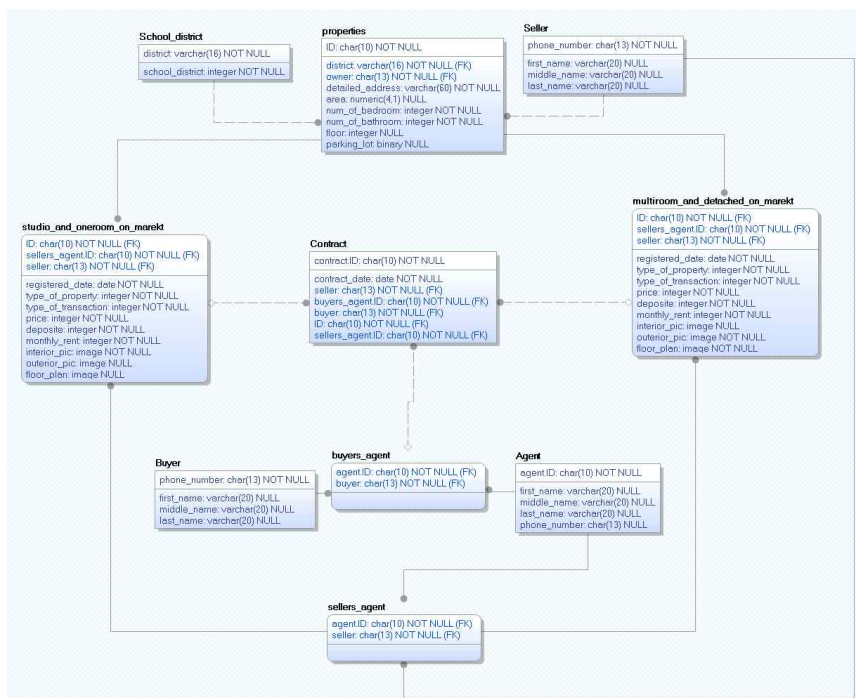


Table	Attribute	Key	Data Type	Not NULL	Note
School_district	district	PK	varchar(16)	O	
	school_district		integer	O	1~11
Properties	ID	PK	char(10)	O	
	district	FK	varchar(16)	O	School_district (district)
	owner	FK	char(13)	O	seller (phone_number)
	detailed_address		varchar(60)	O	도로명 주소
	area		numeric(4,1)		unit: 평
	num_of_bedroom		integer	O	
	num_of_bathroom		integer	O	
	floor		integer		
Seller	parking_lot		binary		
	phone_number	PK	char(13)	O	010-0000-0000
	first_name		varchar(20)		
	middle_name		varchar(20)		
studio_and_oneroom_on_market	last_name		varchar(20)		
	ID	PK, FK	char(10)	O	Properties(ID)
	sellers_agent.ID	PK, FK	char(10)	O	sellers_agent (agent.ID)
	seller	PK, FK	char(13)	O	Properties (owner)
	registered_date		date	O	
	type_of_property		integer	O	0: studio 1: oneroom
	type_of_transaction		integer	O	0: rent 1: buy
	price		integer	O	if rent: -1
	deposit		integer	O	if buy: -1
	monthly_rent		integer	O	if buy: -1
	interior_pic		image	O	
	exterior_pic		image		
	floor_plan		image		
multiroom_and_detached_on_market	ID	PK, FK	char(10)	O	Properties(ID)
	sellers_agent.ID	PK, FK	char(10)	O	sellers_agent (agent.ID)
	seller	PK, FK	char(13)	O	Properties (owner)
	registered_date		date	O	
	type_of_property		integer	O	2: multiroom 3: detached
	type_of_transaction		integer	O	0: rent 1: buy
	price		integer	O	if rent: -1
	deposit		integer	O	if buy: -1
	monthly_rent		integer	O	if buy: -1
	interior_pic		image		
	exterior_pic		image	O	
	floor_plan		image	O	
Contract	contract.ID	PK	char(10)	O	
	contract_date		date	O	
	seller	FK	char(13)	O	
	buyers_agent	FK	char(10)	O	
	buyer	FK	char(13)	O	
	ID	FK	char(10)	O	
	sellers_agent.ID	FK	char(10)	O	
Buyer	phone_number	PK	char(13)	O	010-0000-0000
	first_name		varchar(20)		
	middle_name		varchar(20)		
	last_name		varchar(20)		

Agent	agent.ID	PK	char(10)	O	
	first_name		varchar(20)		
	middle_name		varchar(20)		
	last_name		varchar(20)		
	phone_number		char(13)		010-0000-0000
buyers_agent	agent.ID	PK, FK	char(10)	O	Agent(agent.ID)
	buyer	PK, FK	char(13)	O	Buyer (phone_number)
sellers_agent	agent.ID	PK, FK	char(10)	O	Agent(agent.ID)
	seller	PK, FK	char(13)	O	Seller (phone_number)

3. ODBC Queries

0) TYPE Selection Menu

```

----- 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
Enter the number of TYPE: |

```

```

Enter the number of TYPE: 8
You entered a wrong number. Available numbers are 0 to 7.
----- 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
Enter the number of TYPE: |

```

왼쪽 이미지는 실행할 TYPE을 선택하는 메뉴이다. 1~7 사이의 정수 값을 입력하면 이에 해당하는 TYPE이 실행된다. 0을 입력할 경우 프로그램이 정상적으로 종료된다. 만약 0~7사이의 값이 아닌 다른 값이 입력될 경우 오른쪽 이미지와 같이 wrong input instruction message가 출력되고 TYPE을 다시 선택하도록 한다.

1) TYPE 1

```

1 • SELECT *
2 FROM properties
3 WHERE properties.ID NOT IN (SELECT property_ID FROM contract)
4 AND properties.district = 'Mapo';

```

ID	district	owner	detailed_address	area	num_of_bedroom	num_of_bathroom	floor	parking_lot
17-03-5227	Mapo	010-2508-1227	Maporo 11	52.0	5	2	4	1
22-77-3252	Mapo	010-0548-6252	Maporo 23	25.0	4	1	14	0
35-05-7202	Mapo	010-6517-1202	Maporo 8	60.0	2	2	19	0
64-38-6277	Mapo	010-0773-2277	Maporo 7	32.0	4	2	4	1

```

Enter the number of TYPE: 1
TYPE 1: Find the address of homes for sale in the district "Mapo".

Address
Maporo 11
Maporo 23-1
Maporo 18
Maporo 7
Maporo 13
Maporo 8-9
Maporo 8-11
Maporo 7-3

```

TYPE 1을 선택할 경우 왼쪽 이미지와 같은 쿼리가 MySQL 서버로 transport되며 결과를 receive해 출력한다. properties table에는 시장에 나왔던 모든 매물이 저장되어 있고 contract에는 계약이 체결된 매물 정보가 저장되어 있다. 따라서 contract에는 없고 properties에는 있는 매물 중 district가 “Mapo”인 매물의 주소를 출력하도록 쿼리를 작성했다.

1.0) SUBTYPE Selection Menu of TYPE 1

```

----- Subtype in TYPES 1 -----
1. TYPE 1-1
Enter a number of subtype or enter "0" to go back to the TYPE Selection menu: |

```

TYPE 1이 실행되면 Subtype을 실행할 수 있는 메뉴가 나온다. 0을 입력할 경우 TYPE Selection Menu로 돌아가고 1을 입력할 경우 TYPE 1-1이 실행된다. 이외의 값이 입력될 경우 아래와 같이 input instruction message가 출력된다.

```
Enter a number of subtype or enter "0" to go back to the TYPE Selection menu: 2
You entered a wrong number. Available numbers are 0 and 1.
```

1.1) TYPE 1-1

```
1 • USE new;
2 • SELECT properties.ID, detailed_address, price
3 FROM multiroom_and_detached
4 LEFT JOIN properties ON multiroom_and_detached.ID = properties.ID
5 WHERE multiroom_and_detached.ID IN (
6   SELECT ID
7   FROM properties
8   WHERE properties.ID NOT IN (
9     SELECT property_ID
10    FROM contract
11  ) AND properties.district = 'Mapo'
12 ) AND multiroom_and_detached.price<=1500000000 AND multiroom_and_detached.price>=1000000000;
```

ID	detailed_address	price
17-03-5227	Maporo 11	1500000000
25-05-2269	Maporo 18	1200000000
33-72-6213	Maporo 13	1100000000
61-47-0259	Maporo 8-11	1200000000

```
1 • USE new;
2 • SELECT properties.ID, detailed_address, price
3 FROM studio_and_oneroom
4 LEFT JOIN properties ON studio_and_oneroom.ID = properties.ID
5 WHERE studio_and_oneroom.ID IN (
6   SELECT ID
7   FROM properties
8   WHERE properties.ID NOT IN (
9     SELECT property_ID
10    FROM contract
11  ) AND properties.district = 'Mapo'
12 ) AND studio_and_oneroom.price<=1500000000 AND studio_and_oneroom.price>=1000000000;
```

ID	detailed_address	price
26-25-1197	Maporo 7	1000000000

price에 대한 정보를 properties table이 아닌 multiroom_and_detached와 studio_and_oneroom table에 따로 저장해두었기 때문에 쿼리를 각 table에 대해 2개 작성해 결과를 받았다. price가 저장되어 있는 multiroom_and_detached, studio_and_oneroom와 district 정보가 저장되어 있는 properties를 join한 후, TYPE 1의 schema의 조건에 가격에 대한 조건을 추가했다. 그 결과는 아래와 같이 출력된다.

```
Enter a number of subtype or enter "0" to go back to the TYPE Selection menu: 1
TYPE 1-1: Then find the costing between ₩1,000,000,000 and ₩1,500,000,000.
```

Properties ID	Address	Price
17-03-5227	Maporo 11	1500000000
25-05-2269	Maporo 18	1200000000
33-72-6213	Maporo 13	1100000000
61-47-0259	Maporo 8-11	1200000000
26-25-1197	Maporo 7	1000000000

2) TYPE 2

```
1 • SELECT p.ID, p.district, p.detailed_address, s.school_district
2 FROM properties p
3 LEFT JOIN school_dist s ON p.district=s.district
4 WHERE p.ID NOT IN(SELECT property_ID FROM contract)
5 AND s.school_district=8;
```

```
Enter the number of TYPE: 2
TYPE 2: Find the address of homes for sale in the 8th shcool district.
```

Properties ID	District	Address	School district
08-80-7292	Seocho	Seochoro 6	8
13-08-2193	Gangnam	Gangnamro 11	8
35-68-7268	Gangnam	Gangnamro 15	8
43-44-4243	Gangnam	Gangnamro 2	8
62-82-8242	Seocho	Seochoro 15	8
68-86-7192	Seocho	Seochoro 23	8
75-68-3218	Gangnam	Gangnamro 9	8
76-76-3267	Seocho	Seochoro 26	8
77-45-1293	Gangnam	Gangnamro 25	8
82-53-6217	Seocho	Seochoro 13	8

학군에 대한 정보가 shool_dist에 저장되어 있어 properties와 school_dist table을 join한 후 WHERE문으로 on sales 조건과 8학군 조건을 추가해 쿼리를 작성했다. 그 결과 오른쪽과 같이 판매중인 8학군 집 매물 주소를 확인할 수 있다.

2.0) SUBTYPE Selection Menu of TYPE 2

```
----- Subtype in TYPES 2 -----
1. TYPE 2-1
Enter a number of subtype or enter "0" to go back to the TYPE Selection menu: |
```

TYPE 2가 실행되면 Subtype을 실행할 수 있는 메뉴가 나온다. 0을 입력할 경우 TYPE Selection Menu로 돌아가고 1을 입력할 경우 TYPE 2-1이 실행된다. 이외의 값이 입력될 경우 TYPE1에서의 경우와 같이 input instruction message가 출력된다.

2.1) TYPE 2-1

```
1 • SELECT p.ID, p.detailed_address, s.school_district, p.num_of_bedroom, p.num_of_bathroom
2 FROM properties p
3 LEFT JOIN school_dist s ON p.district=s.district
4 WHERE p.ID NOT IN(SELECT property_ID FROM contract)
5 AND s.school_district=8
6 AND p.num_of_bedroom>=4
7 AND p.num_of_bathroom=2;
```

```
Enter a number of subtype or enter "0" to go back to the TYPE Selection menu: 1

TYPE 2-1: Then find properties with 4 or more bedrooms and 2 bathrooms.

Properties ID    Address          School district # of bedroom # of bathroom
13-08-2193      Gangnamro 11    8           4           2
43-44-4243      Gangnamro 2     8           5           2
76-76-3267      Seochoro 26     8           5           2
77-45-1293      Gangnamro 25    8           4           2
```

TYPE 2의 쿼리에서 WHERE문에 침실과 욕실에 대한 조건을 추가해 쿼리를 작성했다. 해당 쿼리로 오른쪽과 같이 8학군의 침실이 4개 이상이고 욕실이 2개인 매물에 대한 정보를 찾을 수 있다.

3) TYPE 3

```
1 • SELECT c.sellers_agent_ID, a.first_name, a.last_name, SUM(IFNULL(s.Price, 0) + IFNULL(m.Price, 0)) AS total_price_sum
2 FROM contract c
3 LEFT JOIN (SELECT ID, Price FROM studio_and_oneroom) s ON c.property_ID=s.ID
4 LEFT JOIN (SELECT ID, Price FROM multiroom_and_detached) m ON c.property_ID=m.ID
5 LEFT JOIN (SELECT ID, first_name, last_name FROM agent) a ON c.sellers_agent_ID=a.ID
6 WHERE YEAR(c.contract_date)=2022
7 GROUP BY c.sellers_agent_ID
8 ORDER BY total_price_sum DESC
9 LIMIT 1;
```

```
Enter the number of TYPE: 3

TYPE 3: Find the name of the agent who has sold the most properties in the year 2022 by total won value.

Agent ID      Name      Total Price
0000-00600    Gahye    Jang      2300000000
```

계약 내역이 저장되어 있는 contract table을 기준으로 studio_and_oneroom, multiroom_and_detached, agent table을 각각 property ID와 agent ID를 사용해 Join하였다. Join한 schema에서 2022년에 이루어진 계약만 추출하고 GROUP BY를 통해 agent별 판매액 합계를 구했다. 이후 ORDER BY를 이용해 판매액을 정렬했으며, LIMIT을 1로 설정해 가장 판매액이 높은 한 명의 agent 정보만 남도록 쿼리를 작성했다.

3.0) SUBTYPE Selection Menu of TYPE 3

```
----- Subtype in TYPES 3 -----
1. TYPE 3-1
2. TYPE 3-2

Enter a number of subtype or enter "0" to go back to the TYPE Selection menu: |
```

TYPE 3가 실행되면 Subtype을 실행할 수 있는 메뉴가 나온다. 0을 입력할 경우 TYPE Selection Menu로 돌아가고 1 또는 2를 입력할 경우 각각 TYPE 3-1, TYPE 3-2가 실행된다. 이외의 값이 입력될 경우 이전 SUBTYPE Selection Menu와 같이 input instruction message가 출력된다.

3.1) TYPE 3-1

```
char query3_1[999];
const char* query3_1_1 = "SELECT c.sellers_agent_ID, a.first_name,
strcpy(query3_1, query3_1_1);
strcat(query3_1, strk);
state = 0;
state = mysql_query(connection, query3_1);
```

```
----- TYPES 3-1 -----

** Then find the top k agents in the year 2023 by total won value. **
(if the number of agent under condition(n) is less than K, n tuples will be shown)
Which K? : 7

TYPE 3-1: Then find the top k agents in the year 2023 by total won value.

Agent ID      Name      Total Price
0000-00644    Adam      Lipa      1600000000
0000-00642    Luke      Jang      1500000000
0000-00691    Josua     Song      1500000000
0000-00630    Din       Son       1500000000
0000-00548    Adam      Kim       1500000000
0000-00880    Junseok   Willson   1400000000
0000-00538    Sungwon   Coman     1300000000
```

TYPE 3에서 2022년 조건을 2023년으로 바꾸고 LIMIT 조건을 input으로 받아서 LIMIT k를 실행하도록 쿼리를 작성했다. 왼쪽의 이미지는 k를 입력받아 기존의 쿼리와 합치는 코드이고 오른쪽은 해당 쿼리의 결과이다.

3.2) TYPE 3-2

```
1 • SELECT main.sellers_agent_ID, main.first_name, main.last_name, main.total_price_sum
2 FROM (
3 SELECT c.sellers_agent_ID, a.first_name, a.last_name,
4 SUM(IFNULL(s.Price, 0) + IFNULL(m.Price, 0)) AS total_price_sum,
5 ROW_NUMBER() OVER (ORDER BY SUM(IFNULL(s.Price, 0) + IFNULL(m.Price, 0)) ASC) AS row_num
6 FROM contract c
7 LEFT JOIN studio_and_oneroom s ON c.property_ID = s.ID
8 LEFT JOIN multiroom_and_detached m ON c.property_ID = m.ID
9 LEFT JOIN agent a ON c.sellers_agent_ID = a.ID
10 WHERE YEAR(c.contract_date) = 2021
11 GROUP BY c.sellers_agent_ID, a.first_name, a.last_name
12 ) AS main
13 WHERE main.row_num <= (
14 SELECT CEIL(COUNT(DISTINCT sellers_agent_ID) / 10)
15 FROM contract
16 WHERE YEAR(contract_date) = 2021
17 );
```

```
----- Subtype in TYPES 3 -----
1. TYPE 3-1
2. TYPE 3-2

Enter a number of subtype or enter "0" to go back to the TYPE Selection menu: 2

TYPE 3-2: And then find the bottom 10 0x0.00000000000000p+0agents in the year 2021 by total won value.

Agent ID      Name      Total Price
0000-00721    Josua     Willson   800000000
0000-00589    Gahye     Kane      800000000
```

TYPE 3-2의 main 서브쿼리에서는 합계 판매 금액인 total_price_sum을 기준으로 ROW_NUMBER()를 이용해 순위를 매긴다. WHERE에 있는 서브쿼리에서는 2021년 판매에 관여한 agent의 수를 계산해 하위 10%가 몇 명인지 확인한다. 쿼리의 실행 결과 오른쪽과 같은 결과가 나왔다. CRUD.txt를 통해 입력한 데이터에서는 해당 조건을 만족하는 agent가 27명이었고, 따라서 2명이 출력되었다.

4) TYPE 4

```
1 • SELECT
2     c.sellers_agent_ID,
3     AVG(s.price) AS avg_price,
4     AVG(DATEDIFF(c.contract_date, s.registered_date)) AS avg_time_on_market
5 FROM contract c
6 LEFT JOIN studio_and_oneroom s ON c.property_ID = s.ID
7 WHERE YEAR(c.contract_date) = 2022
8 GROUP BY c.sellers_agent_ID;
```

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.

Agent ID	avg Price	avg time on market(days)
0000-00595	150000000.0000	763.0000
0000-00657	100000000.0000	569.0000
0000-00600	115000000.0000	757.5000
0000-00708	160000000.0000	901.0000
0000-00650	120000000.0000	786.0000
0000-00671	120000000.0000	544.0000
0000-00666	160000000.0000	674.0000
0000-00555	90000000.0000	636.0000
0000-00603	80000000.0000	722.0000
0000-00718	120000000.0000	755.0000
0000-00518	160000000.0000	676.0000
0000-00528	110000000.0000	804.0000
0000-00899	150000000.0000	754.0000
0000-00609	140000000.0000	628.0000
0000-00679	130000000.0000	723.0000
0000-00760	100000000.0000	766.0000
0000-00666	130000000.0000	708.0000
0000-00872	160000000.0000	679.0000
0000-00707	150000000.0000	824.0000
0000-00826	150000000.0000	808.0000
0000-00685	160000000.0000	516.0000
0000-00725	140000000.0000	880.0000
0000-00602	110000000.0000	621.0000
0000-00515	140000000.0000	722.0000
0000-00702	160000000.0000	785.0000
0000-00644	90000000.0000	804.0000

contract table에서 2022년 체결된 계약만을 고려했다. GROUP BY를 통해 agent별 평균 계약 금액과 매물이 시장에 있었던 평균 시간을 구했다. 계약 체결 날짜와 매물 등록 날짜를 모두 DATE 데이터 타입으로 설정해놓아서 평균 시간은 DATEDIFF함수를 통해 계산했다. unit은 days이다.

4.0) SUBTYPE Selection Menu of TYPE 4

```
----- Subtype in TYPES 4 -----
1. TYPE 4-1
2. TYPE 4-2

Enter a number of subtype or enter "0" to go back to the TYPE Selection menu: |
```

TYPE 4가 실행되면 Subtype을 실행할 수 있는 메뉴가 나온다. 0을 입력할 경우 TYPE Selection Menu로 돌아가고 1 또는 2를 입력할 경우 각각 TYPE 4-1, TYPE 4-2가 실행된다. 이외의 값이 입력될 경우 이전 SUBTYPE Selection Menu와 같이 input instruction message가 출력된다.

4.1) TYPE 4-1

```
1 • SELECT
2     c.sellers_agent_ID,
3     MAX(s.price) AS max_price
4 FROM contract c
5 LEFT JOIN studio_and_oneroom s ON c.property_ID = s.ID
6 WHERE YEAR(c.contract_date) = 2023
7 GROUP BY c.sellers_agent_ID;
```

TYPE 4-1: Then compute the maximum selling price of properties sold in 2023 for each agent.

Agent ID	Max. Price
0000-00838	1200000000
0000-00873	9000000000
0000-00625	9000000000
0000-00835	9000000000
0000-00584	8000000000
0000-00805	1300000000
0000-00593	9000000000
0000-00508	1500000000
0000-00630	1500000000
0000-00691	1500000000
0000-00710	1100000000
0000-00808	1300000000
0000-00585	1200000000
0000-00571	1200000000
0000-00845	1100000000
0000-00725	9000000000
0000-00706	1200000000
0000-00880	1400000000
0000-00574	1300000000
0000-00732	9000000000
0000-00791	9000000000
0000-00528	9000000000
0000-00535	9000000000
0000-00644	1600000000
0000-00642	1500000000
0000-00538	1300000000

TYPE 4의 쿼리에서 WHERE문의 계약 연도를 2023년으로 교체하고, average price가 아닌 maximum price를 구했다.

4.2) TYPE 4-2

```
1 • SELECT
2     c.sellers_agent_ID,
3     MAX(DATEDIFF(c.contract_date, s.registered_date)) AS max_time_on_market
4 FROM contract c
5 LEFT JOIN studio_and_oneroom s ON c.property_ID = s.ID
6 GROUP BY c.sellers_agent_ID;
```

TYPE 4-2: And then compute the longest time the property was on the market for each agent.

Agent ID	Max. Time(days)
0000-00077	357
0000-00838	1157
0000-00525	763
0000-00873	1227
0000-00657	869
0000-00531	954
0000-00669	768
0000-00625	1159
0000-00835	927
0000-00681	152
0000-00584	1884
0000-00885	888
0000-00592	1097
0000-00708	901
0000-00548	1129
0000-00650	785
0000-00743	277

TYPE 4-2는 agent별 계약 매물 중 시장에 가장 오래 있었던 매물의 최장 시간을 계산하는 것이다. 명시된 계약일 조건이 없었기 때문에 TYPE 4 쿼리에서 계약일 조건을 삭제했다. 그리고 시장에 머물렀던 평균 시간을 최대 시간으로 바꾸어주었다. 시장에 등록되어 있던 시간의 단위는 day이다.

5) TYPE 5

```

SELECT ID, interior_pic, exterior_pic, floor_plan
FROM (
    SELECT ID, interior_pic, exterior_pic, floor_plan
    FROM studio_and_oneroom
    WHERE type_of_property = 0
    ORDER BY price DESC
    LIMIT 1
) AS t1
Union
SELECT ID, interior_pic, exterior_pic, floor_plan
FROM (
    SELECT ID, interior_pic, exterior_pic, floor_plan
    FROM multiroom_and_detached
    WHERE type_of_property = 2
    ORDER BY price DESC
    LIMIT 1
) AS t3
Union
SELECT ID, interior_pic, exterior_pic, floor_plan
FROM (
    SELECT ID, interior_pic, exterior_pic, floor_plan
    FROM studio_and_oneroom
    WHERE type_of_property = 1
    ORDER BY price DESC
    LIMIT 1
) AS t2
Union
SELECT ID, interior_pic, exterior_pic, floor_plan
FROM (
    SELECT ID, interior_pic, exterior_pic, floor_plan
    FROM multiroom_and_detached
    WHERE type_of_property = 3
    ORDER BY price DESC
    LIMIT 1
) AS t4;

```

TYPE 5는 4가지 종류의 집에 대해 각각 가장 비싼 property의 사진을 구한 후 union을 통해 하나의 schema로 합쳤다. 위의 4개의 코드는 왼쪽 위에서부터 시계방향으로 각각 studio, one-bedroom, detached house, multi-bedrooms 중 가장 비싼 property를 구한 코드이다. 쿼리 실행 결과는 아래와 같이 출력되며, 각 이미지는 질의응답을 참고해 “Just String”으로 대체했다.

TYPE 5: Showphotos of the most expensive studio, one - bedroom, multi - bedroom apartment(s), and detached house(s), respectively, from the database.

Property Type	Property ID	Interior pic	Exterior pic	Floor plan
Studio	04-52-4125	04-52-4125 interior picture	04-52-4125 exterior picture	04-52-4125 floor plan
Oneroom	08-30-2114	08-30-2114 interior picture	08-30-2114 exterior picture	08-30-2114 floor plan
Multiroom	01-33-3201	01-33-3201 interior picture	01-33-3201 exterior picture	01-33-3201 floor plan
Detached house	08-66-5220	08-66-5220 interior picture	08-66-5220 exterior picture	08-66-5220 floor plan

**images are replaced with "just string"

6) TYPE 6

TYPE 6는 listed as being available한 property의 sales(contract)를 기록하는 쿼리이다. 요구사항으로는 sales price, buyer, selling agent, buyer's agent, date가 기록되어야 한다. 해당 쿼리를 위해 available한, 즉 market에 등록은 되어있지만 아직 contract하지 않은 property 5개를 리스트화하였다. 해당 property에 대한 sales price와 selling agent는 이미 데이터베이스에 저장되어 있으므로, 사용자로부터 buyer와 date만 입력받아 데이터베이스에 해당 계약을 기록할 수 있도록 쿼리를 작성했다. buyer's agent는 입력을 받지 않는데 이는 buyer와 buyer's agent 간의 연결이 이미 데이터베이스에 존재하기 때문이다.

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.

Buyer candidates:

- 1) 010-2476-2301
- 2) 010-7673-0302
- 3) 010-3082-2303
- 4) 010-6835-2304
- 5) 010-3351-1305

Enter a buyer among buyer candidates (enter between 1 to 5): 2
Enter contract date (format: YYYY-MM-DD): 2022-03-24

A new contract is recorded to the DB successfully.

buyer 정보가 데이터베이스에 저장되어있어야 하므로 buyer candidate 5명을 사전에 리스트업해두었다. 사용자는 이 중에서 buyer를 골라 1~5사이의 정수로 buyer를 선택하고 “YYYY-MM-DD”형식에 맞춰 date를 입력하면 해당 계약이 데이터베이스에 저장되게 된다.

```
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.
```

```
No properties available to contract anymore.
```

사전에 property를 5개 리스팅해두었으므로, 만약 5개에 대해 모두 계약을 하게 된다면 프로그램은 available한 property가 존재하지 않는다는 메시지를 출력한다.

```
If you want to go back to the TYPE Selection Menu, enter "0", otherwise enter any other number: |
```

일련의 입력이 끝나면 프로그램은 contract를 계속할 기록할 것인지 확인한다. 만약 사용자가 0을 입력한다면 TYPE Selection Menu로 돌아가게 되고 0 이외의 정수를 입력하게 되면 TYPE 6 입력창으로 돌아가게 된다.

7) TYPE 7

```
TYPE 7: Add a new agent to the database.
```

```
Enter agent ID (Format: 0000-00000): 0000-00000
```

```
Enter agent's first name: Dua
```

```
Enter agent's last name: Lipa
```

```
A new agent is added to the DB successfully.
```

TYPE 7은 사용자로부터 agent에 대한 정보를 입력받아 데이터베이스에 추가한다. 프로그램은 사용자로부터 agent ID, agent의 first name과 last name을 입력받는다. 데이터베이스에 middle_name과 phone_number attribute도 존재하나, NULL이 허용되므로 필요한 정보를 제외한 나머지는 Default인 NULL을 입력하도록 했다. agent ID는 0000-00000 (4digits-dash-5digits) 형식이다. agent 정보가 제대로 입력될 경우 성공적으로 저장되었다는 메시지를 출력한다.

```
TYPE 7: Add a new agent to the database.
```

```
Enter agent ID (Format: 0000-00000): 0000-00000
```

```
Enter agent's first name: Olivia
```

```
Enter agent's last name: Rodrigo
```

```
The input agent ID is already in the database. Please try other IDs.
```

만약 이미 데이터베이스에 추가되어있는 ID를 사용자가 입력한다면 해당 데이터는 데이터베이스에 추가되지 않으며, ID가 중복되었다는 메시지를 출력한다.

```
If you want to go back to the TYPE Selection Menu, enter "0", otherwise enter any other number: |
```

일련의 입력이 끝나면 프로그램은 agent 추가를 계속할 것인지 확인한다. 만약 사용자가 0을 입력한다면 TYPE Selection Menu로 돌아가게 되고 0 이외의 정수를 입력하게 되면 agent 정보 입력창으로 돌아가게 된다.