House Rental Data Warehouse Report

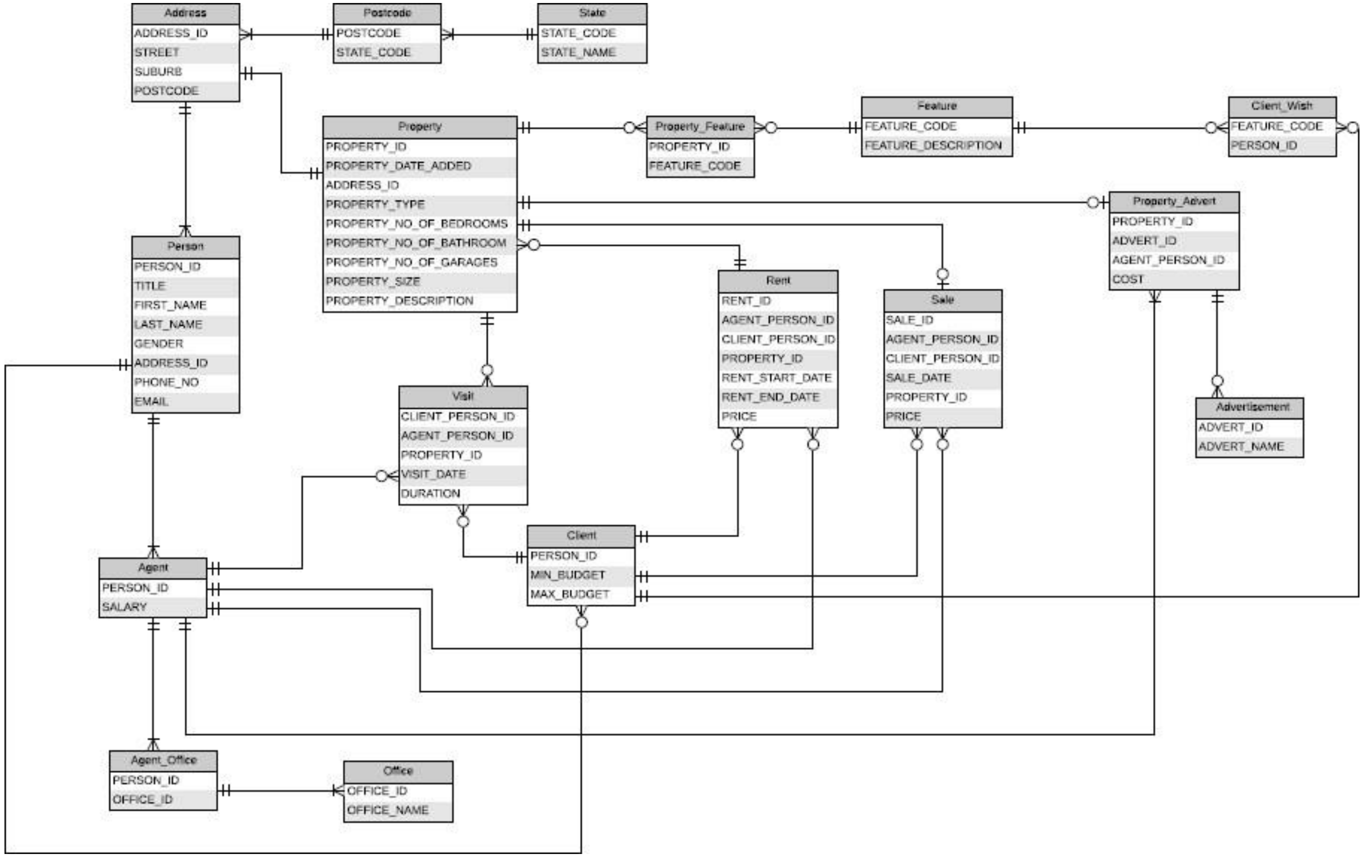
Pradeep Govindan

Contents

C.1

1. Design of Data Warehouse

1.1 The E/R diagram of the operational database



1.2 Data cleaning strategies

The below strategies were used to perform the data cleaning

1.2.1 Identify the data error:

1). Copy the tables from MonRE to local database

2). Check for duplicate records in all the tables local database

3). Check for null records in all the tables in local database

4). Check for invalid or incorrect records in all the tables in local database.

1.2.2 Clean the data error:

Using the star schema as shown in (1.6 star schema) section, the required tables are cleaned using the below steps:

1). Delete duplicate records in the Person table

2). Delete null records from the Person table

3). Delete invalid “phonenumber” record from Person table

3). Delete duplicate records in the Property table

4). Delete null records from State_Code table

5). Delete invalid records from Agent table

6). Delete invalid records from Client table

7). Delete invalid “start date and end date” record from Rent table

8). Delete invalid record from visit table

9). Delete invalid records from Client table

1.3 SQL Commands used to perform Data Cleaning:

1.3.1 Finding invalid records

Duplicate records:

```
SELECT COUNT(*) FROM ADDRESS GROUP BY ADDRESS_ID HAVING COUNT(*) >1;

SELECT COUNT(*) FROM ADVERTISEMENT GROUP BY ADVERT_ID HAVING
COUNT(*) >1;

SELECT COUNT(*) FROM AGENT GROUP BY PERSON_ID HAVING COUNT(*) >1;

SELECT COUNT(*) FROM AGENT_OFFICE GROUP BY PERSON_ID,OFFICE_ID
HAVING COUNT(*) >1;

SELECT COUNT(*) FROM CLIENT GROUP BY PERSON_ID HAVING COUNT(*) >1;
SELECT COUNT(*) FROM CLIENT_WISH GROUP BY PERSON_ID,FEATURE_CODE
HAVING COUNT(*) >1;

SELECT COUNT(*) FROM FEATURE GROUP BY FEATURE_CODE HAVING COUNT(*)
>1;

SELECT COUNT(*) FROM OFFICE GROUP BY OFFICE_ID HAVING COUNT(*) >1;

SELECT COUNT(*) FROM PERSON GROUP BY PERSON_ID HAVING COUNT(*) >1;

SELECT PERSON_ID FROM PERSON GROUP BY PERSON_ID HAVING COUNT(*) >1;

SELECT * FROM PERSON WHERE PERSON_ID=6995;

SELECT COUNT(*) FROM POSTCODE GROUP BY POSTCODE HAVING COUNT(*) >1;

SELECT COUNT(*) FROM PROPERTY GROUP BY PROPERTY_ID HAVING COUNT(*)
>1;

SELECT PROPERTY_ID FROM PROPERTY GROUP BY PROPERTY_ID HAVING
COUNT(*) >1;

SELECT * FROM PROPERTY WHERE PROPERTY_ID IN (6177, 6179) ORDER BY
PROPERTY_ID;

SELECT * FROM PROPERTY WHERE PROPERTY_ID IN (6177, 6179) ORDER BY
PROPERTY_ID;

SELECT COUNT(*) FROM PROPERTY_ADVERT GROUP BY
PROPERTY_ID,ADVERT_ID HAVING COUNT(*) >1;

SELECT COUNT(*) FROM PROPERTY_FEATURE GROUP BY
PROPERTY_ID,FEATURE_CODE HAVING COUNT(*) >1;

SELECT COUNT(*) FROM RENT GROUP BY RENT_ID HAVING COUNT(*) >1;

SELECT COUNT(*) FROM SALE GROUP BY SALE_ID HAVING COUNT(*) >1;

SELECT COUNT(*) FROM STATE GROUP BY STATE_CODE HAVING COUNT(*) >1;

SELECT COUNT(*) FROM VISIT GROUP BY
CLIENT_PERSON_ID,AGENT_PERSON_ID,PROPERTY_ID HAVING COUNT(*) >1;
```

Null Records:

- SALE TABLE

```
SELECT * FROM SALE;
SELECT * FROM SALE WHERE SALE_ID IS NULL;
SELECT * FROM SALE WHERE PRICE IS NULL;
SELECT * FROM SALE WHERE AGENT_PERSON_ID IS NULL;
SELECT * FROM SALE WHERE SALE_DATE IS NULL;
SELECT * FROM SALE WHERE PROPERTY_ID IS NULL;
```

- RENT TABLE

```
SELECT * FROM RENT WHERE RENT_ID IS NULL;
SELECT * FROM RENT WHERE AGENT_PERSON_ID IS NULL;
SELECT * FROM RENT WHERE PROPERTY_ID IS NULL;
SELECT * FROM RENT WHERE PRICE IS NULL;
```

- PROPERTY TABLE

```
SELECT * FROM PROPERTY WHERE PROPERTY_ID IS NULL;
SELECT * FROM PROPERTY WHERE PROPERTY_DATE_ADDED IS NULL;
SELECT * FROM PROPERTY WHERE ADDRESS_ID IS NULL;
SELECT * FROM PROPERTY WHERE PROPERTY_TYPE IS NULL;
SELECT * FROM PROPERTY WHERE PROPERTY_NO_OF_BEDROOMS IS NULL;
SELECT * FROM PROPERTY WHERE PROPERTY_NO_OF_BATHROOMS IS NULL;
SELECT * FROM PROPERTY WHERE PROPERTY_NO_OF_GARAGES IS NULL;
```

- PERSON TABLE

```
SELECT * FROM PERSON WHERE TITLE IS NULL OR LENGTH(TRIM(TITLE)) = 0 OR
TITLE='NULL' OR TITLE = 'Null' OR TITLE = 'null';
SELECT * FROM PERSON WHERE FIRST_NAME IS NULL OR
LENGTH(TRIM(FIRST_NAME)) = 0 OR FIRST_NAME='NULL'
OR FIRST_NAME = 'Null' or FIRST_NAME = 'null';
SELECT * FROM PERSON WHERE LAST_NAME IS NULL OR
LENGTH(TRIM(LAST_NAME)) = 0 OR LAST_NAME='NULL'
OR LAST_NAME = 'Null' or LAST_NAME = 'null';
SELECT * FROM PERSON WHERE GENDER IS NULL OR LENGTH(TRIM(GENDER)) = 0
OR GENDER='NULL';
SELECT * FROM PERSON WHERE EMAIL IS NULL OR LENGTH(TRIM(EMAIL)) = 0 OR
EMAIL='NULL';
SELECT * FROM PERSON WHERE PHONE_NO IS NULL OR
LENGTH(TRIM(PHONE_NO)) != 10 OR PHONE_NO='NULL';
SELECT * FROM PERSON WHERE PERSON_ID = NULL;
SELECT * FROM PERSON WHERE ADDRESS_ID = NULL;
```

- ADDRESS TABLE

```
SELECT * FROM ADDRESS WHERE ADDRESS_ID IS NULL;
SELECT * FROM ADDRESS WHERE STREET IS NULL OR LENGTH(TRIM(STREET)) = 0
OR STREET='NULL' OR STREET = 'Null' or STREET = 'null';
SELECT * FROM ADDRESS WHERE SUBURB IS NULL OR LENGTH(TRIM(SUBURB)) =
0
OR SUBURB='NULL' OR SUBURB = 'Null' or SUBURB = 'null';
SELECT * FROM POSTCODE WHERE POSTCODE IS NULL;
```

- ADVERTISMENT TABLE

```
SELECT * FROM ADVERTISEMENT WHERE ADVERT_ID IS NULL;
SELECT * FROM ADVERTISEMENT WHERE ADVERT_NAME IS NULL
OR LENGTH(TRIM(ADVERT_NAME)) = 0 OR ADVERT_NAME='NULL'
OR ADVERT_NAME = 'Null' or ADVERT_NAME = 'null';
```

- AGENT TABLE

```
SELECT * FROM AGENT WHERE PERSON_ID IS NULL;
SELECT * FROM AGENT WHERE SALARY IS NULL;
SELECT * FROM AGENT WHERE SALARY = 0;
```

- AGENT OFFICE TABLE

```
SELECT * FROM AGENT_OFFICE WHERE PERSON_ID IS NULL;
SELECT * FROM AGENT_OFFICE WHERE OFFICE_ID IS NULL;
```

- CLIENT TABLE

```
SELECT * FROM CLIENT WHERE PERSON_ID IS NULL;
SELECT * FROM CLIENT WHERE MIN_BUDGET IS NULL;
SELECT * FROM CLIENT WHERE MAX_BUDGET IS NULL;
```

- CLIENT WISH TABLE

```
SELECT * FROM CLIENT_WISH WHERE FEATURE_CODE IS NULL;
SELECT * FROM CLIENT_WISH WHERE PERSON_ID IS NULL;
```

- FEATURE TABLE

```
SELECT * FROM FEATURE WHERE FEATURE_DESCRIPTION IS NULL
OR LENGTH(TRIM(FEATURE_DESCRIPTION)) = 0
OR FEATURE_DESCRIPTION='NULL' OR FEATURE_DESCRIPTION = 'Null'
OR FEATURE_DESCRIPTION = 'null';
SELECT * FROM FEATURE WHERE FEATURE_CODE = NULL;
```

- OFFICE TABLE

```
SELECT * FROM OFFICE WHERE OFFICE_NAME IS NULL OR
LENGTH(TRIM(OFFICE_NAME)) = 0
OR OFFICE_NAME='NULL' OR OFFICE_NAME = 'Null'
OR OFFICE_NAME = 'null';
SELECT * FROM OFFICE WHERE OFFICE_ID = NULL;
```

- POSTCODE TABLE

```
SELECT * FROM POSTCODE WHERE STATE_CODE IS NULL OR
LENGTH(TRIM(STATE_CODE)) = 0
OR STATE_CODE='NULL' OR STATE_CODE = 'Null'
OR STATE_CODE = 'null';
SELECT * FROM POSTCODE WHERE POSTCODE IS NULL;
```

- PROPERTY_ADVERT

```
SELECT * FROM PROPERTY_ADVERT WHERE PROPERTY_ID IS NULL;
SELECT * FROM PROPERTY_ADVERT WHERE ADVERT_ID IS NULL;
SELECT * FROM PROPERTY_ADVERT WHERE AGENT_PERSON_ID IS NULL;
SELECT * FROM PROPERTY_ADVERT WHERE COST IS NULL;
```

- PROPERTY_FEATURE TABLE

```
SELECT * FROM PROPERTY_FEATURE WHERE PROPERTY_ID IS NULL OR
FEATURE_CODE IS NULL;
```

- VISIT TABLE

```
SELECT * FROM VISIT WHERE CLIENT_PERSON_ID IS NULL;
SELECT * FROM VISIT WHERE AGENT_PERSON_ID IS NULL;
SELECT * FROM VISIT WHERE PROPERTY_ID IS NULL;
SELECT * FROM VISIT WHERE VISIT_DATE IS NULL;
SELECT * FROM VISIT WHERE DURATION IS NULL;
```

- STATE TABLE

```
SELECT * FROM STATE WHERE STATE_CODE IS NULL;
```

Invalid records:

FINDING POSTCODES THAT ARE NOT IN POSTCODE TABLE

```
SELECT * FROM ADDRESS WHERE POSTCODE NOT IN
(SELECT POSTCODE FROM POSTCODE);
```

FINDING PERSON RECORDS THAT ARE NOT PRESENT IN THE ADDRESS TABLE

```
SELECT * FROM PERSON WHERE ADDRESS_ID NOT IN
(SELECT ADDRESS_ID FROM ADDRESS);
```

FINDING PROPERTY RECORDS THAT ARE NOT PRESENT IN THE ADDRESS TABLE

```
SELECT * FROM PROPERTY WHERE ADDRESS_ID NOT IN
(SELECT ADDRESS_ID FROM ADDRESS);
```

FINDINGPROPERTY ADVERT THAT IS NOT PRESENT IN THE ADD TABLE

```
SELECT * FROM PROPERTY_ADVERT WHERE ADVERT_ID NOT IN
(SELECT ADVERT_ID FROM ADVERTISEMENT);
```

FINDING PERSON ID IN THE AGENT OFFICE THAT IS NOT PRESENT IN THE AGENT TABLE

```
SELECT * FROM AGENT_OFFICE WHERE PERSON_ID NOT IN
(SELECT PERSON_ID FROM AGENT);
```

FINDING OFFICE ID IN THE AGENT OFFICE TABLE THAT IS NOT PRESENT IN THE OFFICE TABLE

```
SELECT * FROM AGENT_OFFICE WHERE OFFICE_ID NOT IN
(SELECT OFFICE_ID FROM OFFICE);
```

FINDING FEATURE CODE IN THE CLIENT WISH WHICH IS NOT PRESENT IN THE FEATURE TABLE

```
SELECT * FROM CLIENT_WISH WHERE FEATURE_CODE NOT IN
(SELECT FEATURE_CODE FROM FEATURE);
```

FINDING PERSON IN THE CLIENT WISH WHICH IS NOT PRESENT IN THE CLIENT TABLE

```
SELECT * FROM CLIENT_WISH WHERE PERSON_ID NOT IN
(SELECT PERSON_ID FROM CLIENT);
```

FINDING AGENT RECORDS WHICH DOES NOT HAVE ANY ENTRY IN PERSON TABLE

```
SELECT PERSON_ID,COUNT(PERSON_ID) FROM AGENT WHERE PERSON_ID NOT IN
(SELECT PERSON_ID FROM PERSON) GROUP BY PERSON_ID;
```

```
SELECT * FROM PERSON WHERE PERSON_ID = 6997;
SELECT * FROM AGENT WHERE PERSON_ID = 6997;
```

FINDING CLIENT RECORDS IN THE CLIENT TABLE WHICH IS NOT PRESENT IN THE PERSON TABLE

```
SELECT PERSON_ID,COUNT(PERSON_ID) FROM CLIENT WHERE PERSON_ID NOT IN
(SELECT PERSON_ID FROM PERSON) GROUP BY PERSON_ID;
```

```
SELECT * FROM PERSON WHERE PERSON_ID = 7000;
SELECT * FROM CLIENT WHERE PERSON_ID = 7000;
```

FINDING CLIENT RECORD IN THE CLIENT_WISH WHICH IS NOT PRESENT IN THE PERSON TABLE

```
SELECT PERSON_ID,COUNT(PERSON_ID) FROM CLIENT_WISH WHERE PERSON_ID
NOT IN
(SELECT PERSON_ID FROM PERSON) GROUP BY PERSON_ID;
```

FINDING AGENTS IN THE PROPERTY ADVERT TABLE WHICH IS NOT IN THE AGENT TABLE

```
SELECT * FROM PROPERTY_ADVERT WHERE AGENT_PERSON_ID NOT IN
(SELECT PERSON_ID FROM AGENT);
```

FINDING PROPERTY ID IN THE RENT TABLE WHICH IS NOT PRESENT IN THE PROPERTY TABLE

```
SELECT * FROM RENT WHERE PROPERTY_ID NOT IN
(SELECT PROPERTY_ID FROM PROPERTY);
```

FINDING PERSON ID IN THE RENT TABLE WHICH IS NOT PRESENT IN THE AGENT TABLE

```
SELECT * FROM RENT WHERE AGENT_PERSON_ID NOT IN
(SELECT PERSON_ID FROM AGENT);
```

```
select * from rent where agent_person_id = 6002;
select * from agent where person_id = 6002;
```

FINDING PERSON ID IN THE RENT TABLE WHICH IS NOT PRESENT IN THE CLIENT TABLE

```
SELECT * FROM RENT WHERE CLIENT_PERSON_ID NOT IN
(SELECT PERSON_ID FROM CLIENT); -- 1 RECORD, DELETING IT
```

```
select * from rent where client_person_id = 6001;
select * from client where person_id = 6001;
```

FINDING RENT START DATE IS GREATER THAN THE END DATE

```
SELECT * FROM RENT WHERE RENT_START_DATE > RENT_END_DATE;
```

FINDING PROPERTY ID IN THE ADVERT TABLE WHICH IS NOT PRESENT IN THE PROPERTY TABLE

```
SELECT * FROM PROPERTY_ADVERT WHERE PROPERTY_ID NOT IN
(SELECT PROPERTY_ID FROM PROPERTY);
```

FINDING PROPERTY ID IN THE PROPERTY FEATURE TABLE WHICH IS NOT PRESENT IN THE PROPERTY TABLE

```
SELECT * FROM PROPERTY_FEATURE WHERE PROPERTY_ID NOT IN
(SELECT PROPERTY_ID FROM PROPERTY);
```

FINDING PROPERTY ID IN THE SALE TABLE WHICH IS NOT PRESENT IN THE PROPERTY TABLE

```
SELECT * FROM SALE WHERE PROPERTY_ID NOT IN
(SELECT PROPERTY_ID FROM PROPERTY);
```

FINDING PROPERTY ID IN THE VISIT TABLE WHICH IS NOT PRESENT IN THE PROPERTY TABLE

```
SELECT * FROM VISIT WHERE PROPERTY_ID NOT IN
(SELECT PROPERTY_ID FROM PROPERTY);
```

FINDING PERSON ID IN THE VISIT TABLE WHICH IS NOT PRESETN IN THE AGENT TABLE

```
SELECT * FROM VISIT WHERE AGENT_PERSON_ID NOT IN
(SELECT PERSON_ID FROM AGENT);
```

```
select * from agent where person_id = 6001;
```

FINDING PERSON ID IN THE VISIT TABLE WHICH IS NOT PRESENT IN THE CLIENT TABLE

```
SELECT * FROM VISIT WHERE CLIENT_PERSON_ID NOT IN
(SELECT PERSON_ID FROM CLIENT);
```

```
select * from client where person_id = 6000;
```

FINDING STATECODES THAT ARE NOT IN POSTCODE TABLE

```
SELECT * FROM POSTCODE WHERE STATE_CODE NOT IN
(SELECT STATE_CODE FROM STATE);
```

FINDING PERSON ID IN THE SALE TABLE WHICH IS NOT PRESENT IN THE AGENT TABLE

```
SELECT * FROM SALE WHERE AGENT_PERSON_ID NOT IN
(SELECT PERSON_ID FROM AGENT);
```

FINDING PERSON ID IN THE SALE TABLE WHICH IS NOT PRESENT IN THE CLIENT TABLE

```
SELECT * FROM SALE WHERE CLIENT_PERSON_ID NOT IN
(SELECT PERSON_ID FROM CLIENT);
```

FINDING PROPERTY ID IN THE SALE TABLE WHICH IS NOT PRESENT IN THE PROPERTY TABLE

```
SELECT * FROM SALE WHERE PROPERTY_ID NOT IN
(SELECT PROPERTY_ID FROM PROPERTY);
```

FINDING MINIMUM BUDGET IS GREATER THAN THE MAXIMUM BUDGET OR NEGATIVE VALUE

```
SELECT * FROM CLIENT WHERE MIN_BUDGET > MAX_BUDGET OR MAX_BUDGET <
0 OR MIN_BUDGET < 0;
```

FINDING CHECK IF A PROPERTY IN SALE TABLE IS ADDED BEFROE ADDING IN THE PROPERTY TABLE

```
SELECT P.PROPERTY_ID, P.PROPERTY_DATE_ADDED, S.SALE_DATE
FROM PROPERTY P, SALE S
WHERE P.PROPERTY_ID = S.PROPERTY_ID
AND S.SALE_DATE < P.PROPERTY_DATE_ADDED;
```

POST CODE VALIDATION

```
SELECT * FROM POSTCODE WHERE POSTCODE BETWEEN 3000 AND 3999
```

```
AND STATE_CODE NOT IN ('VIC');
```

POST CODE VALIDATION

```
SELECT * FROM POSTCODE WHERE POSTCODE BETWEEN 4000 AND 4999
AND STATE_CODE NOT IN ('QLD');
```

POST CODE VALIDATION

```
SELECT * FROM POSTCODE WHERE POSTCODE BETWEEN 5000 AND 5999
AND STATE_CODE NOT IN ('SA');
```

POST CODE VALIDATION

```
SELECT * FROM POSTCODE WHERE POSTCODE BETWEEN 6000 AND 6999
AND STATE_CODE NOT IN ('WA');
```

POST CODE VALIDATION

```
SELECT * FROM POSTCODE WHERE POSTCODE BETWEEN 2000 AND 2599
AND POSTCODE BETWEEN 2619 AND 2898
AND POSTCODE BETWEEN 2921 AND 2999
AND STATE_CODE NOT IN ('NSW');
```

POST CODE VALIDATION

```
SELECT * FROM POSTCODE WHERE POSTCODE BETWEEN 2600 AND 2618
AND POSTCODE BETWEEN 2900 AND 2920
AND STATE_CODE NOT IN ('ACT');
```

POST CODE VALIDATION

```
SELECT * FROM POSTCODE WHERE POSTCODE BETWEEN 7000 AND 7799
AND STATE_CODE NOT IN ('TAS');
```

POST CODE VALIDATION

```
SELECT * FROM POSTCODE WHERE POSTCODE BETWEEN 0800 AND 0899
AND STATE_CODE NOT IN ('NT');
```

GENDER VERIFICATION

```
SELECT * FROM PERSON WHERE GENDER NOT IN ('Male', 'Female');
```

PHONE NUMBER VALIDATION

```
SELECT * FROM PERSON WHERE LENGTH(PHONE_NO) > 10;
```

Data Cleaning:

The tables which are used in the star schema is identified and cleaned

DELETING DUPLICATE RECORDS IN PERSON TABLE

```
DELETE FROM PERSON
WHERE ROWID NOT IN (SELECT MIN(ROWID) FROM PERSON GROUP BY
PERSON_ID);
```

DELETING DUPLICATE RECORDS FROM PROPERTY TABLE

```
DELETE FROM PROPERTY
WHERE ROWID NOT IN (SELECT MIN(ROWID) FROM PROPERTY GROUP BY
PROPERTY_ID);
```

DELETING NULL RECORDS FROM PERSON TABLE

```
DELETE FROM PERSON WHERE TITLE = 'null';
DELETE FROM PERSON WHERE LENGTH(TRIM(PHONE_NO)) != 10;


DELETE FROM STATE WHERE STATE_CODE IS NULL;
```

DELETING INVALID AGENT RECORD TABLE

```
DELETE FROM AGENT WHERE PERSON_ID = 6997;
DELETE FROM AGENT WHERE PERSON_ID = 1536;
```

DELETING INVALID CLIENT RECORD TABLE

```
DELETE FROM CLIENT WHERE PERSON_ID = 7000;
```

DELETING THE INCORRECT RECORD IN RENT TABLE

```
DELETE FROM RENT WHERE RENT_ID = 3284;
```

DELETING INCORRECT RECORD FROM VISIT TABLE

```
DELETE FROM VISIT WHERE CLIENT_PERSON_ID = 6000;
```

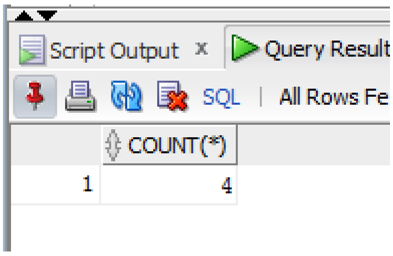
DELETING RECORDS IN CLIENT TABLE WHERE MINIMUM BUDGET AND MAX BUDGET ARE INVALID DATA

```
DELETE FROM CLIENT WHERE PERSON_ID IN (5900, 5901, 5902);
```

1.4 Screenshot of Operational Database before performing Data Cleaning:

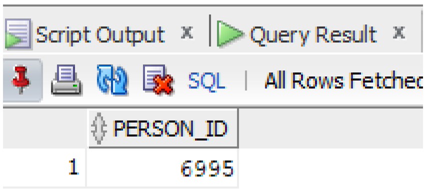
1.4.1). Duplicate records in Person table:

SELECT COUNT(*) FROM PERSON GROUP BY PERSON_ID HAVING COUNT(*) >1;



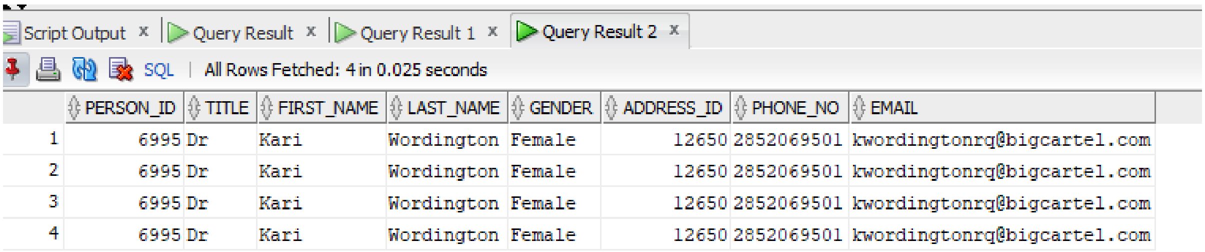
	COUNT(*)
1	4

SELECT PERSON_ID FROM PERSON GROUP BY PERSON_ID HAVING COUNT(*) >1;



	PERSON_ID
1	6995

SELECT * FROM PERSON WHERE PERSON_ID=6995;

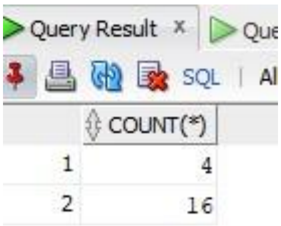


All Rows Fetched: 4 in 0.025 seconds

	PERSON_ID	TITLE	FIRST_NAME	LAST_NAME	GENDER	ADDRESS_ID	PHONE_NO	EMAIL
1	6995	Dr	Kari	Wordington	Female	12650	2852069501	kwordingtonrq@bigcartel.com
2	6995	Dr	Kari	Wordington	Female	12650	2852069501	kwordingtonrq@bigcartel.com
3	6995	Dr	Kari	Wordington	Female	12650	2852069501	kwordingtonrq@bigcartel.com
4	6995	Dr	Kari	Wordington	Female	12650	2852069501	kwordingtonrq@bigcartel.com

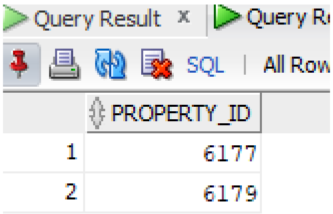
1.4.2). Duplicate records in Property table

SELECT COUNT(*) FROM PROPERTY GROUP BY PROPERTY_ID HAVING COUNT(*) >1;



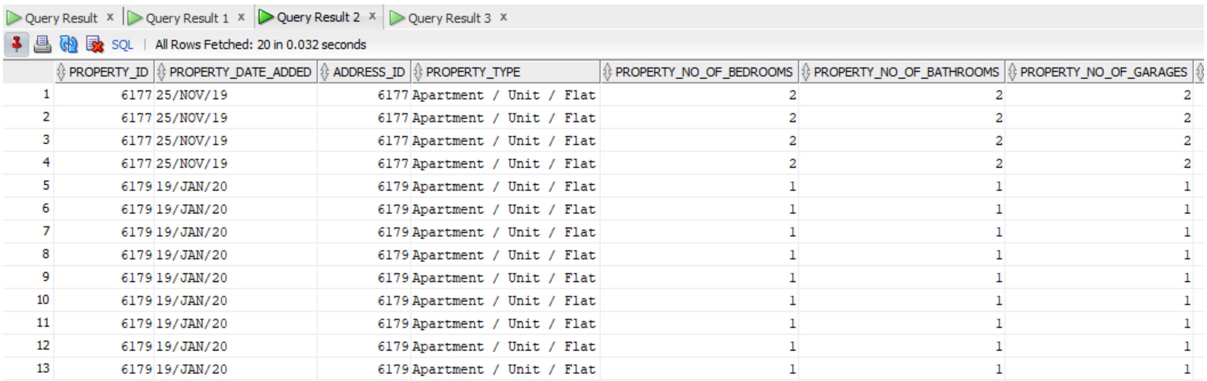
	COUNT(*)
1	4
2	16

SELECT PROPERTY_ID FROM PROPERTY GROUP BY PROPERTY_ID HAVING COUNT(*) >1;



	PROPERTY_ID
1	6177
2	6179

SELECT * FROM PROPERTY WHERE PROPERTY_ID IN (6177, 6179) ORDER BY PROPERTY_ID;



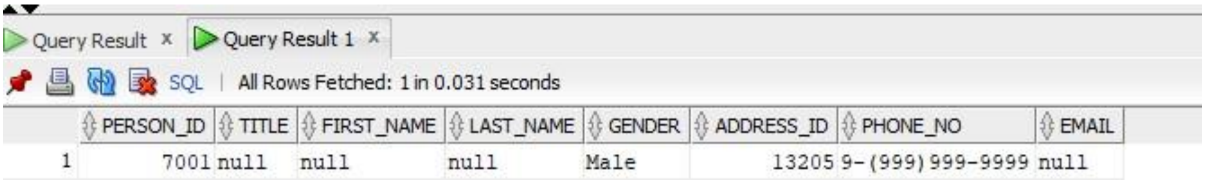
All Rows Fetched: 20 in 0.032 seconds

	PROPERTY_ID	PROPERTY_DATE_ADDED	ADDRESS_ID	PROPERTY_TYPE	PROPERTY_NO_OF_BEDROOMS	PROPERTY_NO_OF_BATHROOMS	PROPERTY_NO_OF_GARAGES
1	6177	25/NOV/19	6177	Apartment / Unit / Flat	2	2	2
2	6177	25/NOV/19	6177	Apartment / Unit / Flat	2	2	2
3	6177	25/NOV/19	6177	Apartment / Unit / Flat	2	2	2
4	6177	25/NOV/19	6177	Apartment / Unit / Flat	2	2	2
5	6179	19/JAN/20	6179	Apartment / Unit / Flat	1	1	1
6	6179	19/JAN/20	6179	Apartment / Unit / Flat	1	1	1
7	6179	19/JAN/20	6179	Apartment / Unit / Flat	1	1	1
8	6179	19/JAN/20	6179	Apartment / Unit / Flat	1	1	1
9	6179	19/JAN/20	6179	Apartment / Unit / Flat	1	1	1
10	6179	19/JAN/20	6179	Apartment / Unit / Flat	1	1	1
11	6179	19/JAN/20	6179	Apartment / Unit / Flat	1	1	1
12	6179	19/JAN/20	6179	Apartment / Unit / Flat	1	1	1
13	6179	19/JAN/20	6179	Apartment / Unit / Flat	1	1	1

Invalid Records

1.4.3). Invalid address in person table

SELECT * FROM PERSON WHERE ADDRESS_ID NOT IN (SELECT ADDRESS_ID FROM ADDRESS);

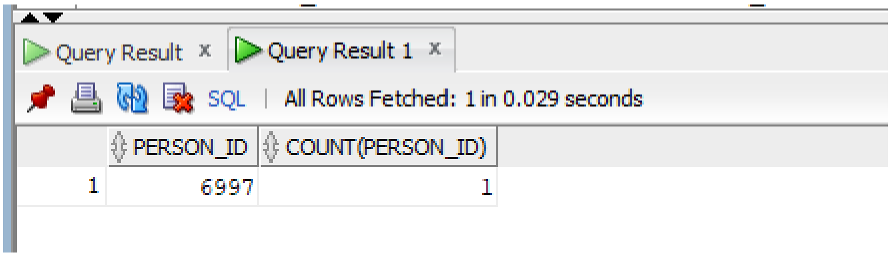


All Rows Fetched: 1 in 0.031 seconds

	PERSON_ID	TITLE	FIRST_NAME	LAST_NAME	GENDER	ADDRESS_ID	PHONE_NO	EMAIL
1	7001	null	null	null	Male	13205	9-(999)999-9999	null

1.4.4). Invalid Agent record – The agent's person_id is not present in the person table

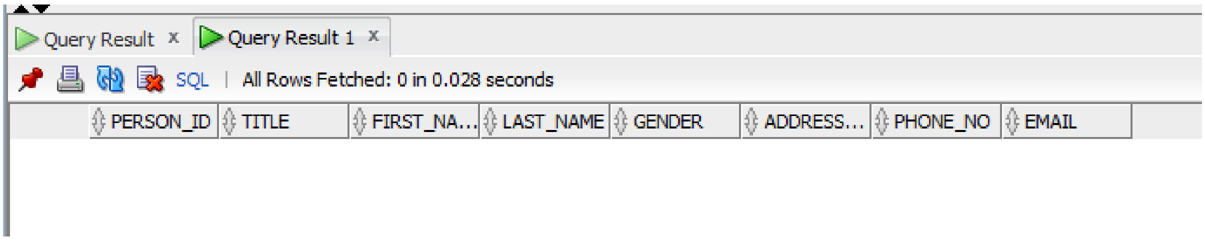
SELECT PERSON ID, COUNT(PERSON_ID) FROM AGENT WHERE PERSON_ID NOT IN (SELECT PERSON_ID FROM PERSON) GROUP BY PERSON_ID;



All Rows Fetched: 1 in 0.029 seconds

	PERSON_ID	COUNT(PERSON_ID)
1	6997	1

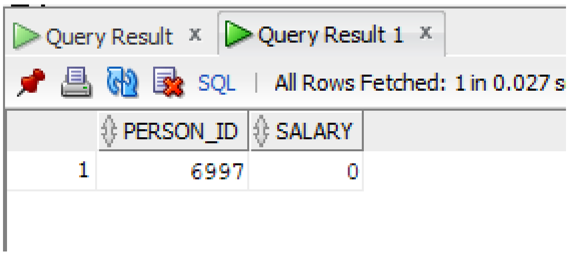
SELECT * FROM PERSON WHERE PERSON_ID = 6997;



All Rows Fetched: 0 in 0.028 seconds

PERSON_ID	TITLE	FIRST_NA...	LAST_NAME	GENDER	ADDRESS...	PHONE_NO	EMAIL

SELECT * FROM AGENT WHERE PERSON_ID = 6997;

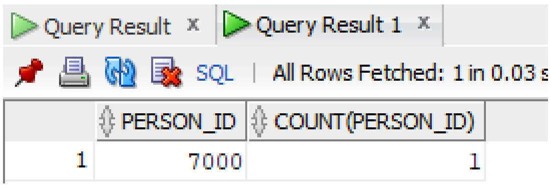


All Rows Fetched: 1 in 0.027 s

	PERSON_ID	SALARY
1	6997	0

1.4.5). Invalid Client record – The client's person_id is not present in the person table

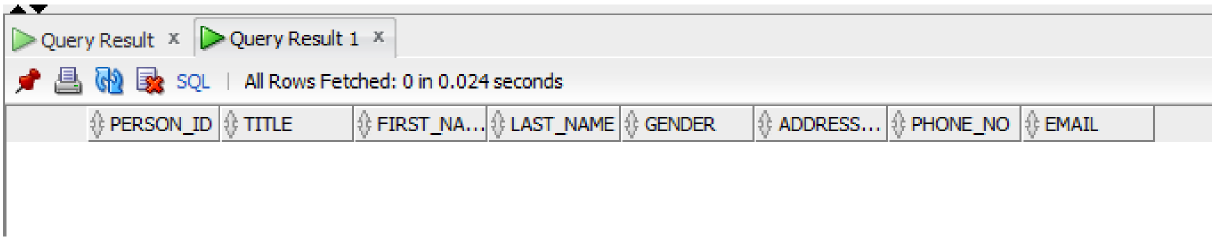
SELECT PERSON_ID,COUNT(PERSON_ID) FROM CLIENT WHERE PERSON_ID NOT IN (SELECT PERSON_ID FROM PERSON) GROUP BY PERSON_ID;



All Rows Fetched: 1 in 0.03 s

	PERSON_ID	COUNT(PERSON_ID)
1	7000	1

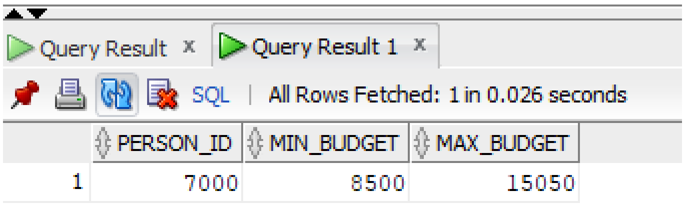
SELECT * FROM PERSON WHERE PERSON_ID = 7000;



All Rows Fetched: 0 in 0.024 seconds

PERSON_ID	TITLE	FIRST_NA...	LAST_NAME	GENDER	ADDRESS...	PHONE_NO	EMAIL

SELECT * FROM CLIENT WHERE PERSON_ID = 7000;

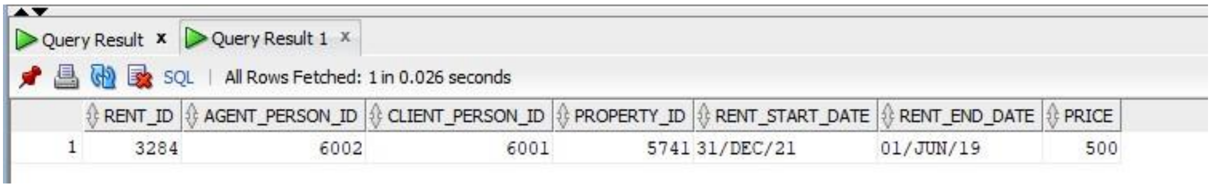


All Rows Fetched: 1 in 0.026 seconds

	PERSON_ID	MIN_BUDGET	MAX_BUDGET
1	7000	8500	15050

1.4.6). Invalid Rent record – The rent's agent_person_id is not present in the agent table

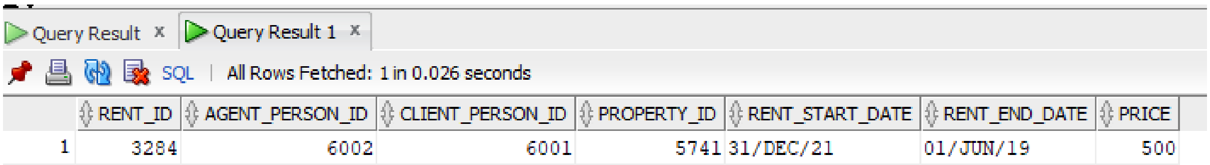
SELECT * FROM RENT WHERE AGENT_PERSON_ID NOT IN
(SELECT PERSON_ID FROM AGENT);



All Rows Fetched: 1 in 0.026 seconds

	RENT_ID	AGENT_PERSON_ID	CLIENT_PERSON_ID	PROPERTY_ID	RENT_START_DATE	RENT_END_DATE	PRICE
1	3284	6002	6001	5741	31/DEC/21	01/JUN/19	500

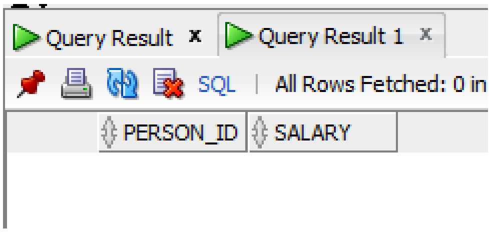
select * from rent where agent_person_id = 6002;



All Rows Fetched: 1 in 0.026 seconds

	RENT_ID	AGENT_PERSON_ID	CLIENT_PERSON_ID	PROPERTY_ID	RENT_START_DATE	RENT_END_DATE	PRICE
1	3284	6002	6001	5741	31/DEC/21	01/JUN/19	500

select * from agent where person_id = 6002;

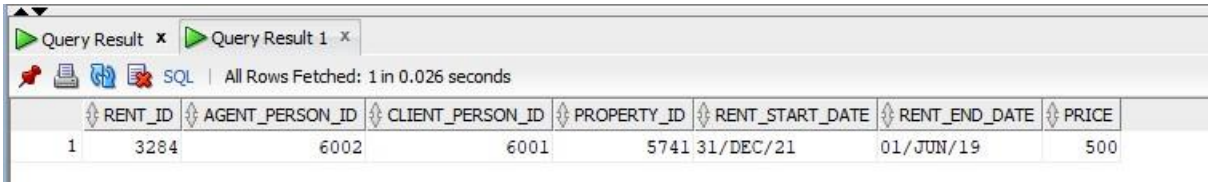


All Rows Fetched: 0 in

PERSON_ID	SALARY

1.4.7). Invalid Rent record – The rent's client_person_id is not present in the client table

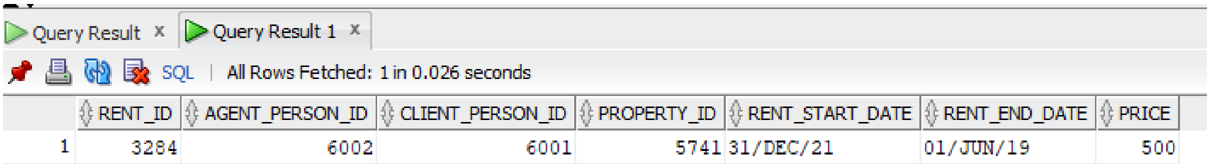
SELECT * FROM RENT WHERE CLIENT_PERSON_ID NOT IN
(SELECT PERSON_ID FROM CLIENT);



All Rows Fetched: 1 in 0.026 seconds

	RENT_ID	AGENT_PERSON_ID	CLIENT_PERSON_ID	PROPERTY_ID	RENT_START_DATE	RENT_END_DATE	PRICE
1	3284	6002	6001	5741	31/DEC/21	01/JUN/19	500

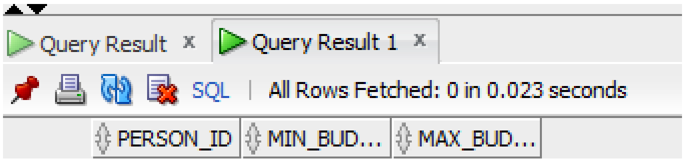
select * from rent where client_person_id = 6001;



All Rows Fetched: 1 in 0.026 seconds

	RENT_ID	AGENT_PERSON_ID	CLIENT_PERSON_ID	PROPERTY_ID	RENT_START_DATE	RENT_END_DATE	PRICE
1	3284	6002	6001	5741	31/DEC/21	01/JUN/19	500

select * from agent where person_id = 6001;

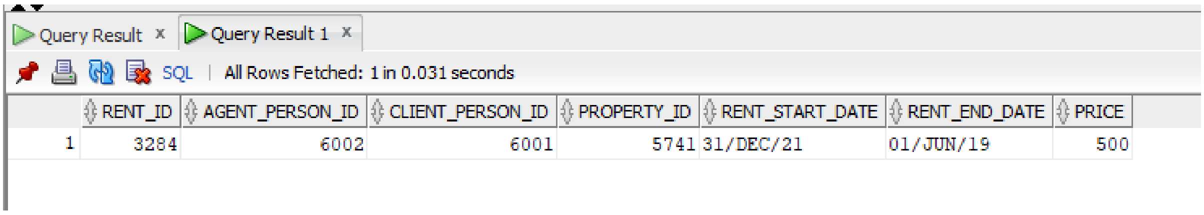


All Rows Fetched: 0 in 0.023 seconds

PERSON_ID	MIN_BUD...	MAX_BUD...

1.4.8). Invalid start and end date in rent table:

SELECT * FROM RENT WHERE RENT_START_DATE > RENT_END_DATE;

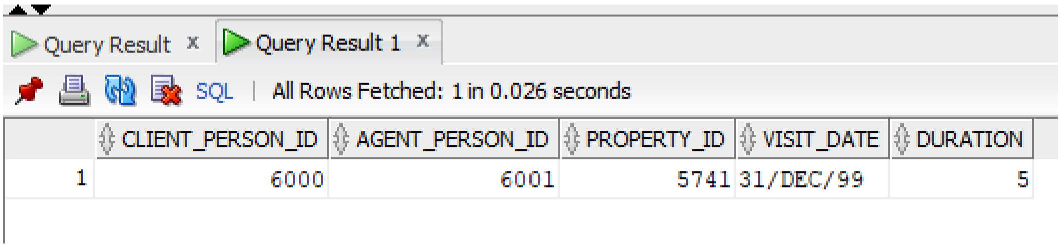


All Rows Fetched: 1 in 0.031 seconds

	RENT_ID	AGENT_PERSON_ID	CLIENT_PERSON_ID	PROPERTY_ID	RENT_START_DATE	RENT_END_DATE	PRICE
1	3284	6002	6001	5741	31/DEC/21	01/JUN/19	500

1.4.9). Invalid visit record – the agent_person_id who made a visit is not present in the agent table

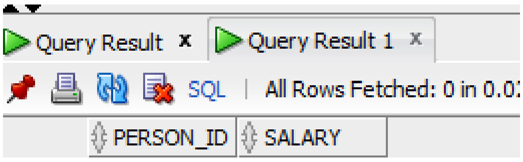
SELECT * FROM VISIT WHERE AGENT_PERSON_ID NOT IN(SELECT PERSON_ID FROM AGENT);



All Rows Fetched: 1 in 0.026 seconds

	CLIENT_PERSON_ID	AGENT_PERSON_ID	PROPERTY_ID	VISIT_DATE	DURATION
1	6000	6001	5741	31/DEC/99	5

select * from agent where person_id = 6001;

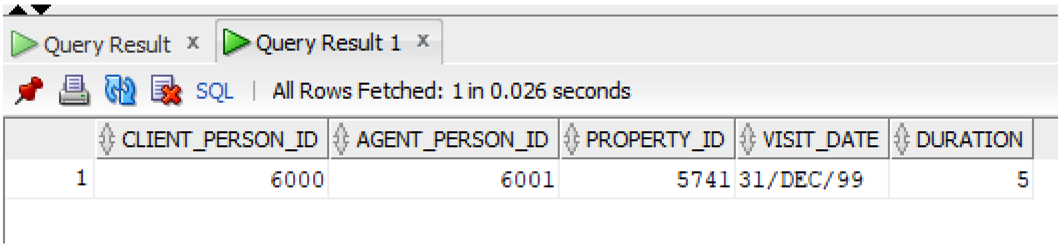


All Rows Fetched: 0 in 0.0

PERSON_ID	SALARY

1.4.10). Invalid visit record – the client_person_id who made a visit is not present in the client table

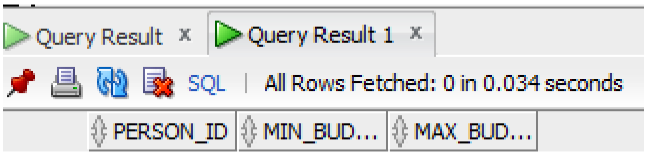
SELECT * FROM VISIT WHERE CLIENT_PERSON_ID NOT IN (SELECT PERSON_ID FROM CLIENT);



All Rows Fetched: 1 in 0.026 seconds

	CLIENT_PERSON_ID	AGENT_PERSON_ID	PROPERTY_ID	VISIT_DATE	DURATION
1	6000	6001	5741	31/DEC/99	5

select * from client where person_id = 6000;

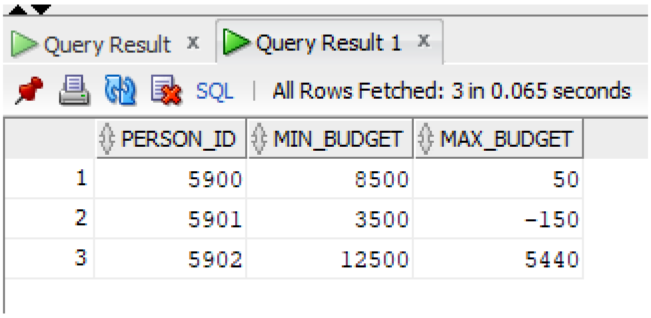


All Rows Fetched: 0 in 0.034 seconds

PERSON_ID	MIN_BUD...	MAX_BUD...

1.4.11). Invalid numeric values in the budget column of the client table

SELECT * FROM CLIENT WHERE MIN_BUDGET > MAX_BUDGET OR MAX_BUDGET < 0 OR MIN_BUDGET < 0

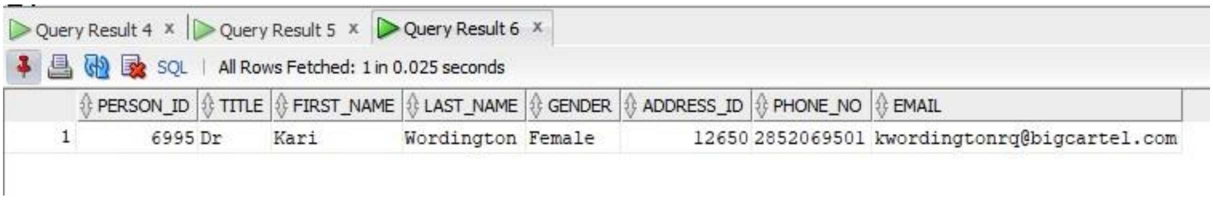


All Rows Fetched: 3 in 0.065 seconds

	PERSON_ID	MIN_BUDGET	MAX_BUDGET
1	5900	8500	50
2	5901	3500	-150
3	5902	12500	5440

1.5 Screenshot of Operational Database after performing Data Cleaning:

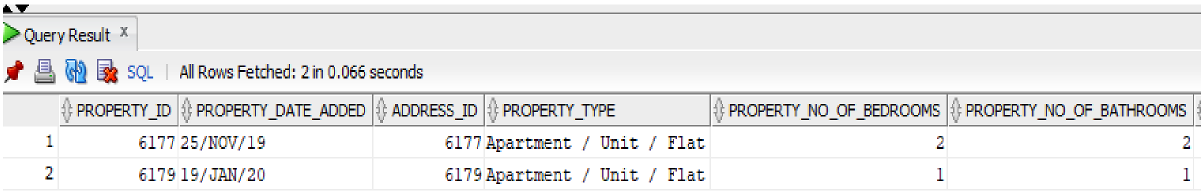
1.5.1). Removed the duplicate records in the Person table



All Rows Fetched: 1 in 0.025 seconds

	PERSON_ID	TITLE	FIRST_NAME	LAST_NAME	GENDER	ADDRESS_ID	PHONE_NO	EMAIL
1	6995	Dr	Kari	Wordington	Female	12650	2852069501	kwordingtonrq@bigcartel.com

1.5.2). Removed duplicate records in the PROPERTY table

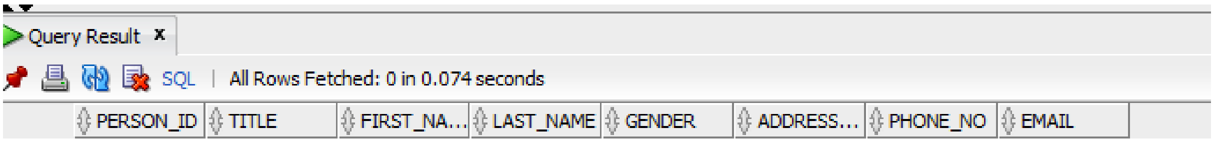


All Rows Fetched: 2 in 0.066 seconds

	PROPERTY_ID	PROPERTY_DATE_ADDED	ADDRESS_ID	PROPERTY_TYPE	PROPERTY_NO_OF_BEDROOMS	PROPERTY_NO_OF_BATHROOMS
1	6177	25/NOV/19	6177	Apartment / Unit / Flat	2	2
2	6179	19/JAN/20	6179	Apartment / Unit / Flat	1	1

1.5.3). Removed Invalid address in Person table

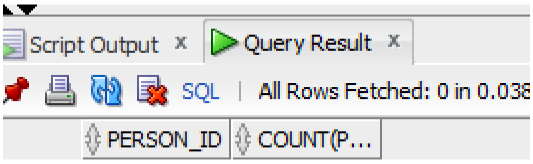
SELECT * FROM PERSON WHERE ADDRESS_ID NOT IN
(SELECT ADDRESS_ID FROM ADDRESS);



All Rows Fetched: 0 in 0.074 seconds

PERSON_ID	TITLE	FIRST_NA...	LAST_NAME	GENDER	ADDRESS...	PHONE_NO	EMAIL

1.5.4). Removed Invalid Agent record
SELECT PERSON_ID,COUNT(PERSON_ID) FROM AGENT WHERE PERSON_ID NOT IN
(SELECT PERSON_ID FROM PERSON) GROUP BY PERSON_ID;

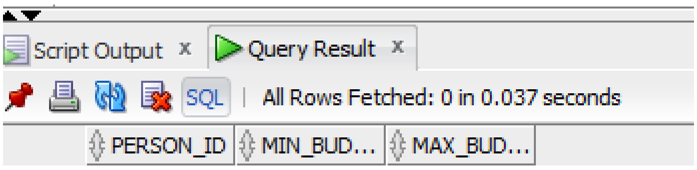


All Rows Fetched: 0 in 0.038

PERSON_ID	COUNT(P...

1.5.5). Removed invalid Client record

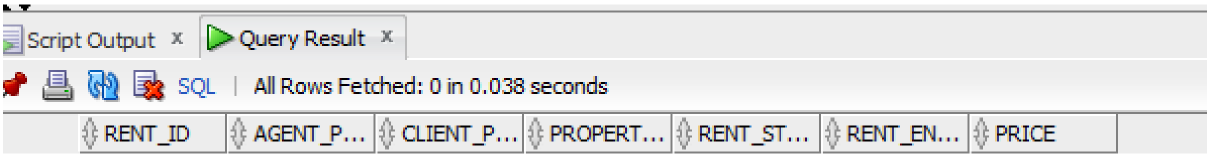
SELECT * FROM CLIENT WHERE MIN_BUDGET > MAX_BUDGET OR MAX_BUDGET < 0 OR MIN_BUDGET < 0



All Rows Fetched: 0 in 0.037 seconds

PERSON_ID	MIN_BUD...	MAX_BUD...

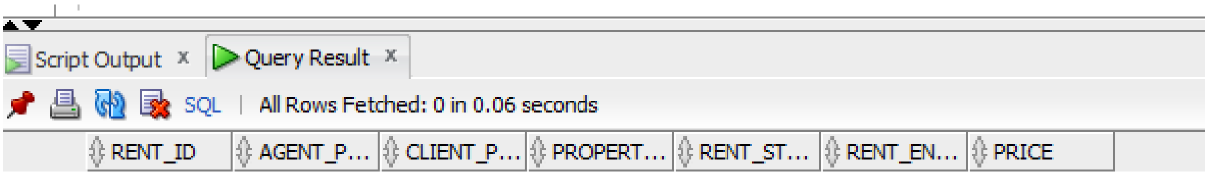
1.5.6). Removed invalid Rent record
SELECT * FROM RENT WHERE AGENT_PERSON_ID NOT IN
(SELECT PERSON_ID FROM AGENT);



All Rows Fetched: 0 in 0.038 seconds

RENT_ID	AGENT_P...	CLIENT_P...	PROPERT...	RENT_ST...	RENT_EN...	PRICE

SELECT * FROM RENT WHERE RENT_START_DATE > RENT_END_DATE;



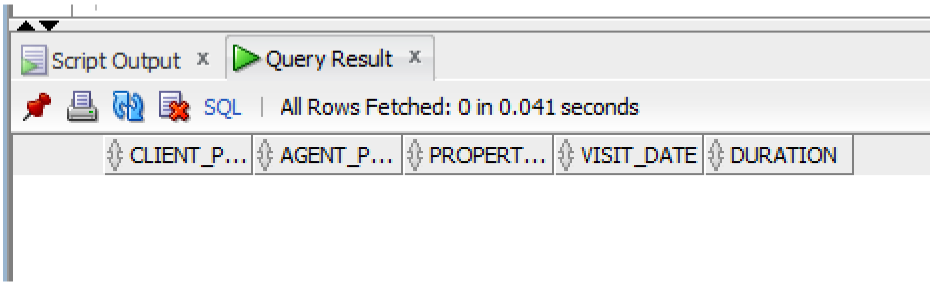
Script Output × Query Result ×

SQL | All Rows Fetched: 0 in 0.06 seconds

RENT_ID	AGENT_P...	CLIENT_P...	PROPERT...	RENT_ST...	RENT_EN...	PRICE

1.5.7). Removed invalid Visit record

SELECT * FROM VISIT WHERE AGENT_PERSON_ID NOT IN(SELECT PERSON_ID FROM AGENT);



Script Output × Query Result ×

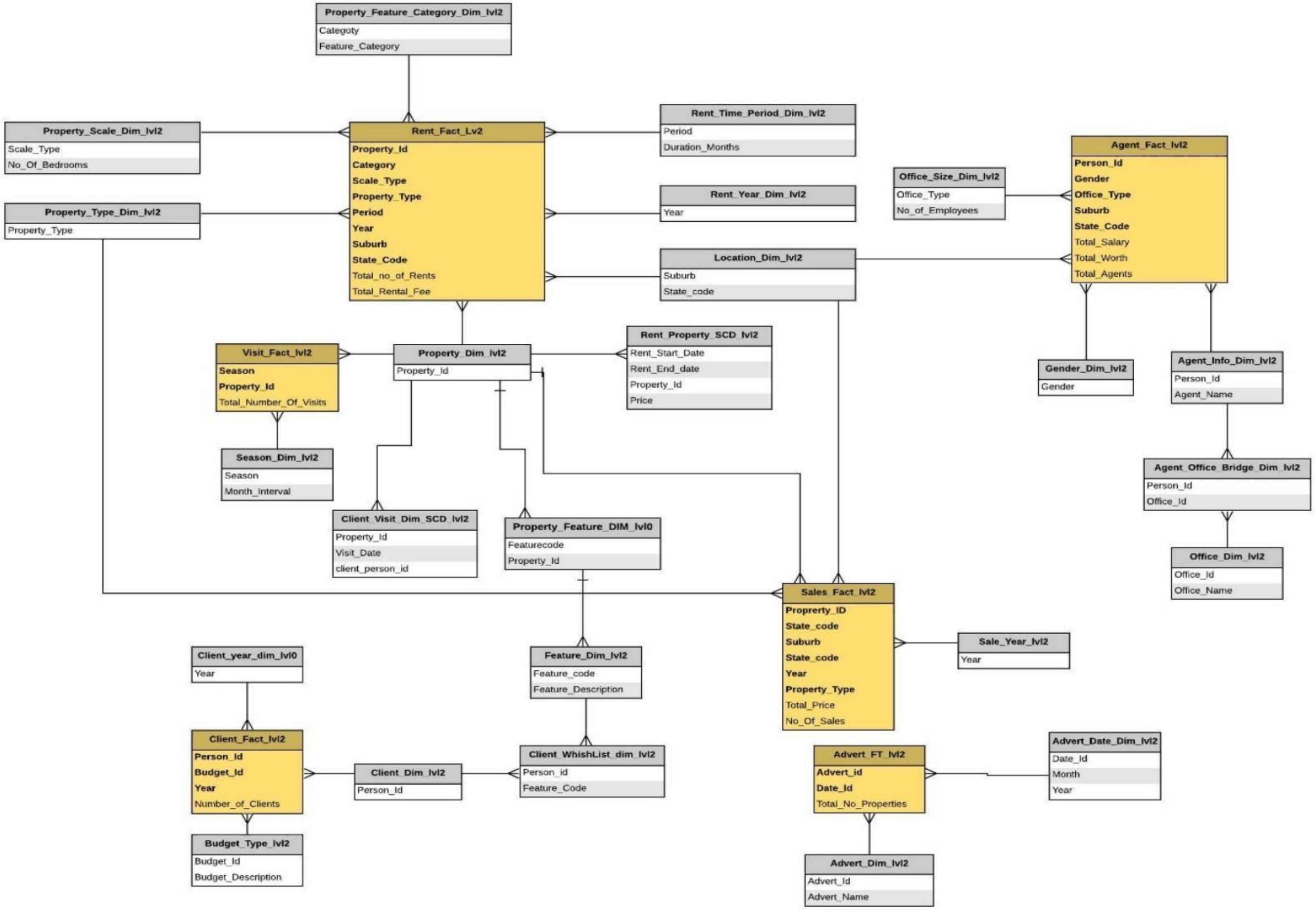
SQL | All Rows Fetched: 0 in 0.041 seconds

CLIENT_P...	AGENT_P...	PROPERT...	VISIT_DATE	DURATION

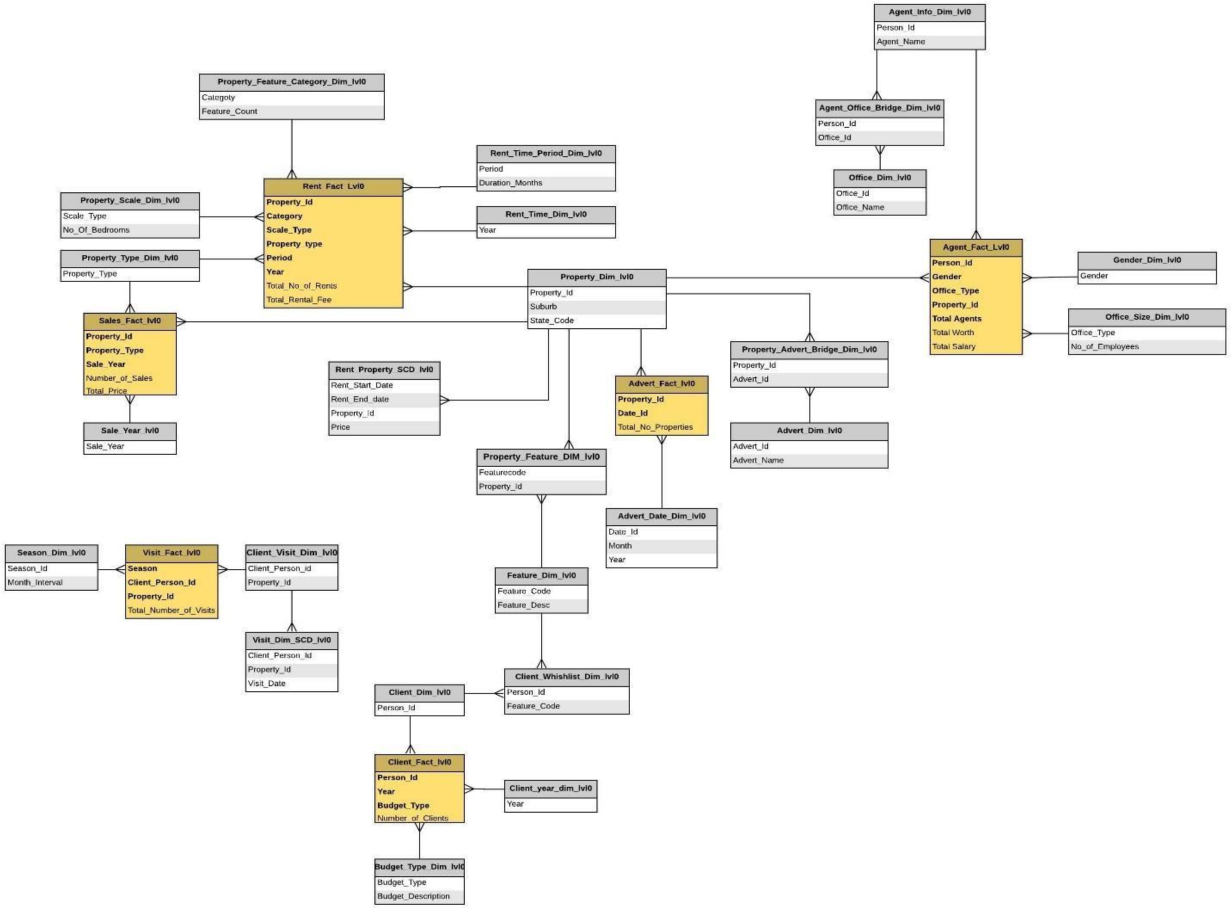
1.6 Design of Data warehouse

1.7 Schema

1.7.1 Version 1 (Level 2)



1.7.2 Version 2 (Level 0)



C2

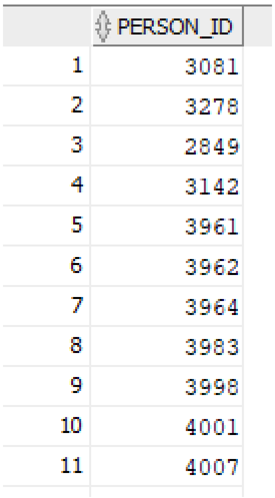
2. Implementation of Data warehouse

SQL statement to create star/snowflake schema Version – 1 (Level 2)

Client Dimensions and Fact creation

Client Dimension

```
CREATE TABLE client_dim_lvl2
   AS
      SELECT DISTINCT
         person_id
      FROM
         client;
```

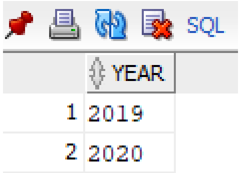


	PERSON_ID
1	3081
2	3278
3	2849
4	3142
5	3961
6	3962
7	3964
8	3983
9	3998
10	4001
11	4007

Client Year Dimension

```
CREATE TABLE client_year_dim_lvl2
   AS
      SELECT DISTINCT
```

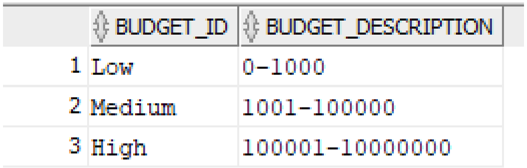
```
        year
    FROM
        (
            SELECT
                to_char(rent_start_date, 'yyyy') AS year
            FROM
                rent
            UNION
            SELECT
                to_char(sale_date, 'yyyy') AS year
            FROM
                sale
            UNION
            SELECT
                to_char(visit_date, 'yyyy') AS year
            FROM
                visit
        )
    WHERE
        NOT year IS NULL;
```



	YEAR
1	2019
2	2020

Client Budget Dimension

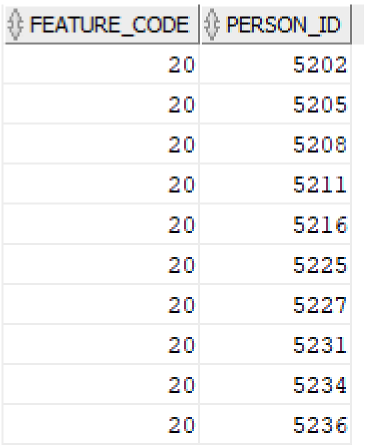
```
CREATE TABLE budget_type_dim_lvl2 (
    budget_id           VARCHAR(10),
    budget_description  VARCHAR(20)
);

INSERT INTO budget_type_dim_lvl2 VALUES (
    'Low',
    '0-1000'
);

INSERT INTO budget_type_dim_lvl2 VALUES (
    'Medium',
    '1001-100000'
);

INSERT INTO budget_type_dim_lvl2 VALUES (
    'High',
    '100001-10000000'
);
```



	BUDGET_ID	BUDGET_DESCRIPTION
1	Low	0-1000
2	Medium	1001-100000
3	High	100001-10000000

Client Wishlist Dimension

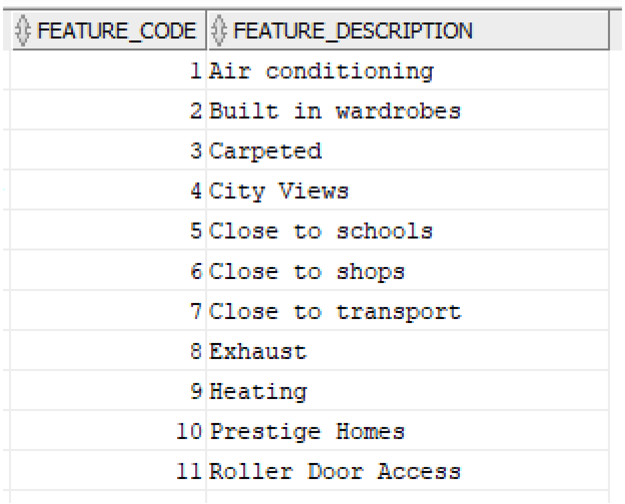
```
CREATE TABLE client_wishlist_dim_lvl2
    AS
        SELECT
            *
        FROM
            client_wish;
```



FEATURE_CODE	PERSON_ID
20	5202
20	5205
20	5208
20	5211
20	5216
20	5225
20	5227
20	5231
20	5234
20	5236

Feature Dimension

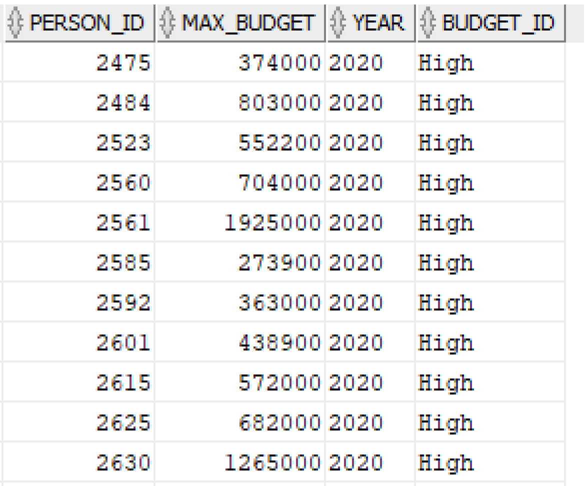
```
CREATE TABLE feature_dim_lvl2
    AS
        SELECT
            feature_code,
            feature_description
        FROM
            feature;
```



FEATURE_CODE	FEATURE_DESCRIPTION
1	Air conditioning
2	Built in wardrobes
3	Carpeted
4	City Views
5	Close to schools
6	Close to shops
7	Close to transport
8	Exhaust
9	Heating
10	Prestige Homes
11	Roller Door Access

Client Temp fact

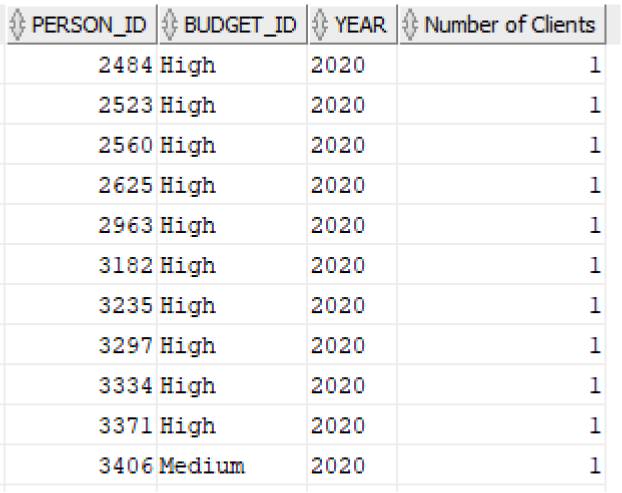
```
CREATE TABLE client_tempfact_lvl2
  AS
    SELECT
      person_id,
      max_budget,
      MIN(year) AS year
    FROM
      (
        SELECT
          person_id,
          max_budget,
          to_char(rent_start_date, 'yyyy') AS year
        FROM
          client c
          JOIN rent     r ON c.person_id = r.client_person_id
        UNION
        SELECT
          person_id,
          max_budget,
          to_char(sale_date, 'yyyy') AS year
        FROM
          client c
          JOIN sale     s ON c.person_id = s.client_person_id
        UNION
        SELECT
          person_id,
          max_budget,
          to_char(visit_date, 'yyyy') AS year
        FROM
          client c
          JOIN visit    v ON c.person_id = v.client_person_id
      )
    GROUP BY
      person_id,
      max_budget;

ALTER TABLE client_tempfact_lvl2 ADD budget_id VARCHAR(10);

UPDATE client_tempfact_lvl2
SET
  budget_id = 'Low'
WHERE
  max_budget BETWEEN 0 AND 1000;

UPDATE client_tempfact_lvl2
SET
  budget_id = 'Medium'
WHERE
  max_budget BETWEEN 1001 AND 100000;

UPDATE client_tempfact_lvl2
SET
  budget_id = 'High'
WHERE
  max_budget BETWEEN 100001 AND 10000000;
```



PERSON_ID	MAX_BUDGET	YEAR	BUDGET_ID
2475	374000	2020	High
2484	803000	2020	High
2523	552200	2020	High
2560	704000	2020	High
2561	1925000	2020	High
2585	273900	2020	High
2592	363000	2020	High
2601	438900	2020	High
2615	572000	2020	High
2625	682000	2020	High
2630	1265000	2020	High

Client Fact

```
CREATE TABLE client_fact_lvl2
  AS
    SELECT
      person_id,
      budget_id,
      year,
      COUNT(person_id) "Number of Clients"
    FROM
      client_tempfact_lvl2
    GROUP BY
      budget_id,
      person_id,
      year;
```

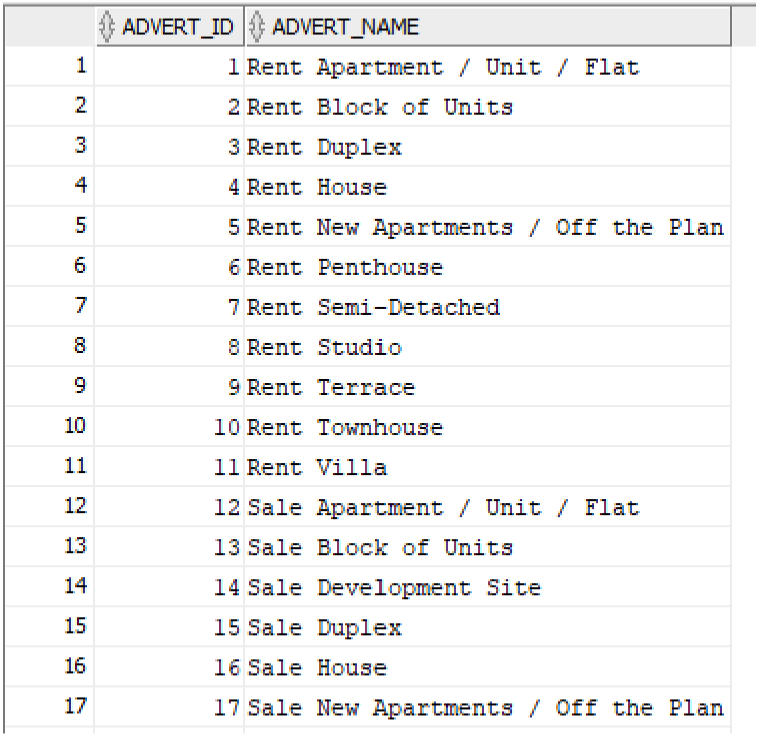


PERSON_ID	BUDGET_ID	YEAR	Number of Clients
2484	High	2020	1
2523	High	2020	1
2560	High	2020	1
2625	High	2020	1
2963	High	2020	1
3182	High	2020	1
3235	High	2020	1
3297	High	2020	1
3334	High	2020	1
3371	High	2020	1
3406	Medium	2020	1

Advertisement Dimensions and Fact creation

Advertisement Dimension

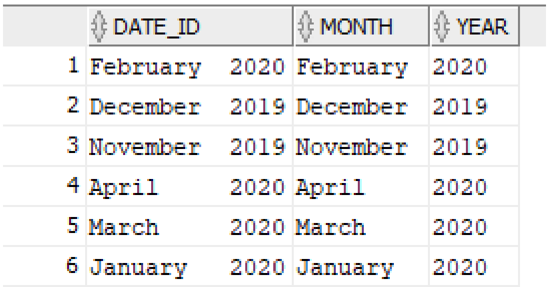
```
CREATE TABLE advert_dim_lvl2
  AS
    SELECT
        advert_id,
        advert_name
    FROM
        advertisement;
```



	ADVERT_ID	ADVERT_NAME
1	1	Rent Apartment / Unit / Flat
2	2	Rent Block of Units
3	3	Rent Duplex
4	4	Rent House
5	5	Rent New Apartments / Off the Plan
6	6	Rent Penthouse
7	7	Rent Semi-Detached
8	8	Rent Studio
9	9	Rent Terrace
10	10	Rent Townhouse
11	11	Rent Villa
12	12	Sale Apartment / Unit / Flat
13	13	Sale Block of Units
14	14	Sale Development Site
15	15	Sale Duplex
16	16	Sale House
17	17	Sale New Apartments / Off the Plan

Advertisement Date Dimension

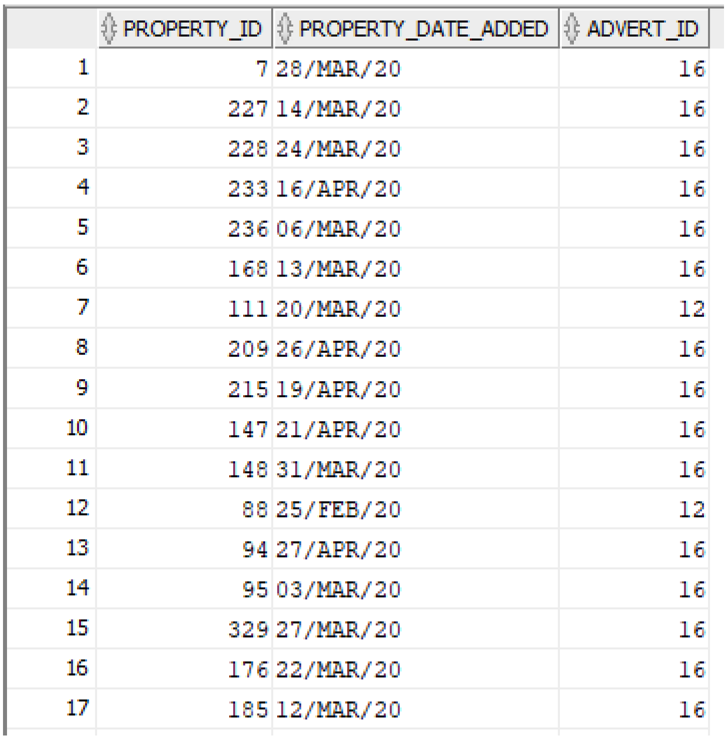
```
CREATE TABLE advert_date_dim_lvl2
    AS
        SELECT DISTINCT
            to_char(property_date_added, 'Month')
            || ' '
            || to_char(property_date_added, 'yyyy') date_id,
            to_char(property_date_added, 'Month') month,
            to_char(property_date_added, 'yyyy') year
        FROM
            property;
```



	DATE_ID	MONTH	YEAR
1	February 2020	February	2020
2	December 2019	December	2019
3	November 2019	November	2019
4	April 2020	April	2020
5	March 2020	March	2020
6	January 2020	January	2020

Advertisement Temp fact

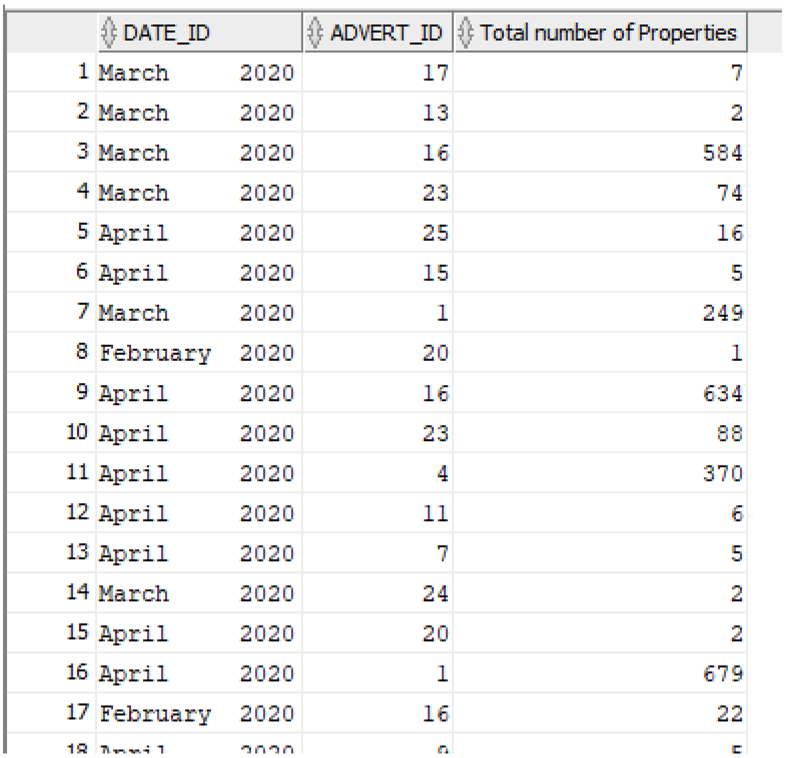
```
CREATE TABLE advertisement_tempfact_lvl2
    AS
        SELECT
            p.property_id,
            pd.property_date_added,
            a.advert_id
        FROM
            advertisement A
            join

        property_advert p
            ON a.advert_id = p.advert_id
        JOIN property pd
            ON p.property_id=pd.property_id
        GROUP BY
            p.property_id,pd.property_date_added,a.advert_id;
```



	PROPERTY_ID	PROPERTY_DATE_ADDED	ADVERT_ID
1	7	28/MAR/20	16
2	227	14/MAR/20	16
3	228	24/MAR/20	16
4	233	16/APR/20	16
5	236	06/MAR/20	16
6	168	13/MAR/20	16
7	111	20/MAR/20	12
8	209	26/APR/20	16
9	215	19/APR/20	16
10	147	21/APR/20	16
11	148	31/MAR/20	16
12	88	25/FEB/20	12
13	94	27/APR/20	16
14	95	03/MAR/20	16
15	329	27/MAR/20	16
16	176	22/MAR/20	16
17	185	12/MAR/20	16

Advertisement Fact

```
CREATE TABLE advertisement_fact_lvl2
    AS
        SELECT
            to_char(property_date_added, 'Month')
            || ' '
            || to_char(property_date_added, 'yyyy') date_id,
            advert_id,
            COUNT(property_id) "Total number of Properties"
        FROM
            advertisement_tempfact_lvl2
        GROUP BY
            to_char(property_date_added, 'Month')
            || ' '
            || to_char(property_date_added, 'yyyy'),
            advert_id;
```

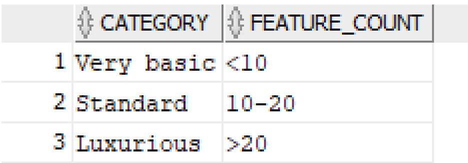


	DATE_ID	ADVERT_ID	Total number of Properties
1	March 2020	17	7
2	March 2020	13	2
3	March 2020	16	584
4	March 2020	23	74
5	April 2020	25	16
6	April 2020	15	5
7	March 2020	1	249
8	February 2020	20	1
9	April 2020	16	634
10	April 2020	23	88
11	April 2020	4	370
12	April 2020	11	6
13	April 2020	7	5
14	March 2020	24	2
15	April 2020	20	2
16	April 2020	1	679
17	February 2020	16	22
18	April 2020	9	5

Rent Dimensions and Fact creation

Property Feature Category Dimension

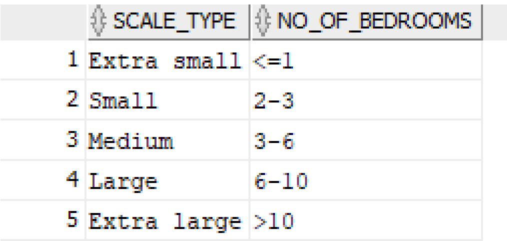
```
CREATE TABLE property_feature_category_dim_lvl2 (
    category        VARCHAR(20),
    feature_count  VARCHAR(20)
);

INSERT INTO property_feature_category_dim_lvl2 VALUES (
    'Very basic',
    '<10'
);

INSERT INTO property_feature_category_dim_lvl2 VALUES (
    'Standard',
    '10-20'
);

INSERT INTO property_feature_category_dim_lvl2 VALUES (
    'Luxurious',
    '>20'
);
```



	CATEGORY	FEATURE_COUNT
1	Very basic	<10
2	Standard	10-20
3	Luxurious	>20

Property Scale Dimension

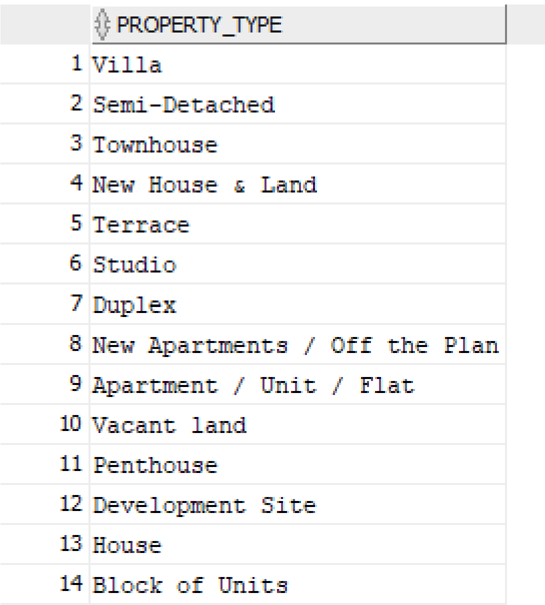
```
CREATE TABLE property_scale_dim_lvl2 (
    scale_type       VARCHAR(20),
    no_of_bedrooms  VARCHAR(20)
);

INSERT INTO property_scale_dim_lvl2 VALUES (
    'Extra small',
    '<=1'
);

INSERT INTO property_scale_dim_lvl2 VALUES (
    'Small',
    '2-3'
);

INSERT INTO property_scale_dim_lvl2 VALUES (
    'Medium',
    '3-6'
);

INSERT INTO property_scale_dim_lvl2 VALUES (
    'Large',
    '6-10'
);

INSERT INTO property_scale_dim_lvl2 VALUES (
    'Extra large',
    '>10'
);
```



	SCALE_TYPE	NO_OF_BEDROOMS
1	Extra small	<=1
2	Small	2-3
3	Medium	3-6
4	Large	6-10
5	Extra large	>10

Property Type Dimension

```
CREATE TABLE property_type_dim_lvl2
    AS
        SELECT DISTINCT
            property_type
        FROM
            property;
```

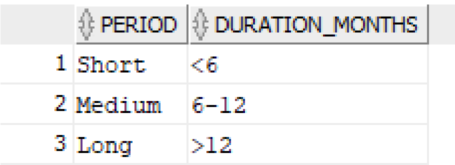


	PROPERTY_TYPE
1	Villa
2	Semi-Detached
3	Townhouse
4	New House & Land
5	Terrace
6	Studio
7	Duplex
8	New Apartments / Off the Plan
9	Apartment / Unit / Flat
10	Vacant land
11	Penthouse
12	Development Site
13	House
14	Block of Units

Rent Time Period Dimension

```
CREATE TABLE rent_time_period_dim_lvl2 (
    period           VARCHAR(20),
    duration_months  VARCHAR(20)
);

INSERT INTO rent_time_period_dim_lvl2 VALUES (
    'Short',
    '<6'
);
```

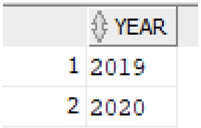
```
INSERT INTO rent_time_period_dim_lvl2 VALUES (
    'Medium',
    '6-12'
);

INSERT INTO rent_time_period_dim_lvl2 VALUES (
    'Long',
    '>12'
);
```



	PERIOD	DURATION_MONTHS
1	Short	<6
2	Medium	6-12
3	Long	>12

Rent Year Dimension

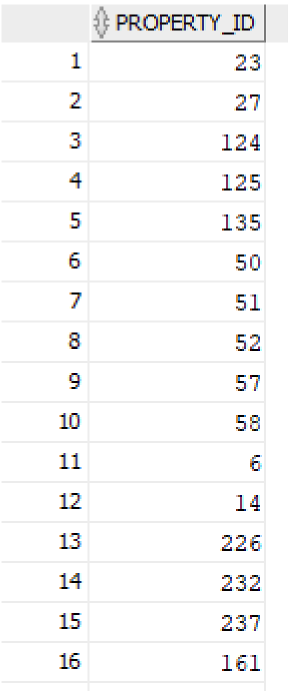
```
CREATE TABLE rent_year_dim_lvl2
    AS
        SELECT DISTINCT
            to_char(rent_start_date, 'yyyy') AS year
        FROM
            rent
        WHERE
            NOT rent_start_date IS NULL;
```



	YEAR
1	2019
2	2020

Property Dimension

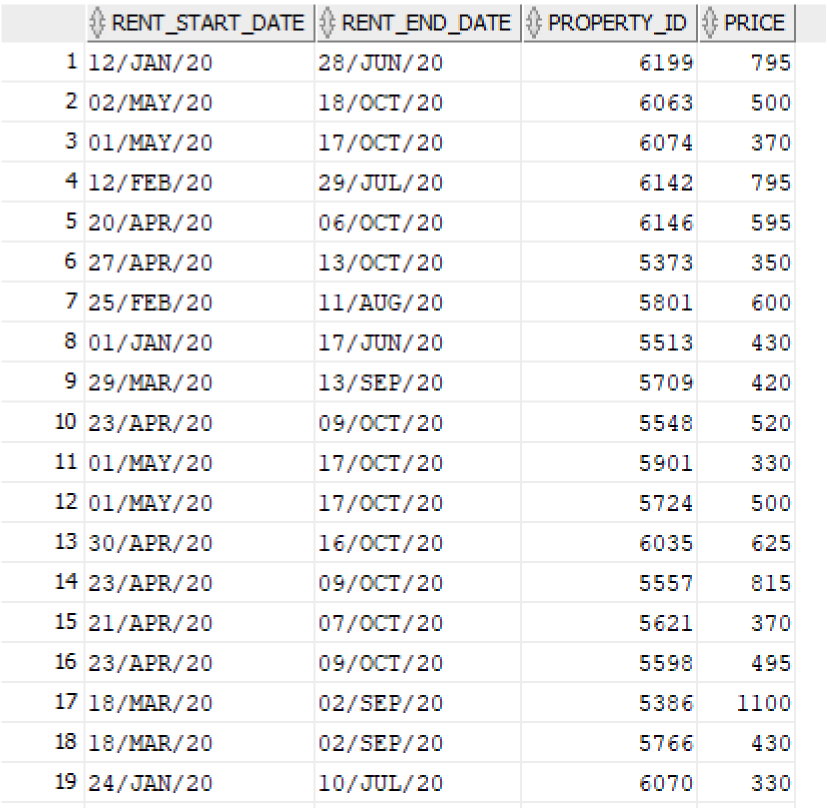
```
CREATE TABLE property_dim_lvl2
    AS
        SELECT DISTINCT
            p.property_id
        FROM
            Property p;
```



	PROPERTY_ID
1	23
2	27
3	124
4	125
5	135
6	50
7	51
8	52
9	57
10	58
11	6
12	14
13	226
14	232
15	237
16	161

Property Rent SCD Dimension

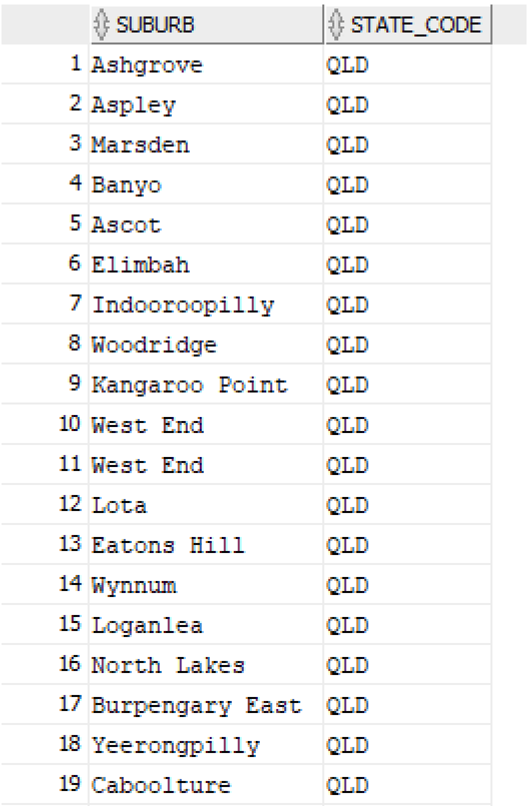
```
CREATE TABLE property_rent_scd_lvl2
    AS
        SELECT
            rent_start_date,
            rent_end_date,
            property_id,
            price
        FROM
            rent
        WHERE
            NOT rent_start_date IS NULL;
```



	RENT_START_DATE	RENT_END_DATE	PROPERTY_ID	PRICE
1	12/JAN/20	28/JUN/20	6199	795
2	02/MAY/20	18/OCT/20	6063	500
3	01/MAY/20	17/OCT/20	6074	370
4	12/FEB/20	29/JUL/20	6142	795
5	20/APR/20	06/OCT/20	6146	595
6	27/APR/20	13/OCT/20	5373	350
7	25/FEB/20	11/AUG/20	5801	600
8	01/JAN/20	17/JUN/20	5513	430
9	29/MAR/20	13/SEP/20	5709	420
10	23/APR/20	09/OCT/20	5548	520
11	01/MAY/20	17/OCT/20	5901	330
12	01/MAY/20	17/OCT/20	5724	500
13	30/APR/20	16/OCT/20	6035	625
14	23/APR/20	09/OCT/20	5557	815
15	21/APR/20	07/OCT/20	5621	370
16	23/APR/20	09/OCT/20	5598	495
17	18/MAR/20	02/SEP/20	5386	1100
18	18/MAR/20	02/SEP/20	5766	430
19	24/JAN/20	10/JUL/20	6070	330

Location Dimension

```
CREATE TABLE location_dim_lvl2
  AS
    SELECT
      a.suburb,
      p.state_code
    FROM
      address

A
join

postcode p ON a.postcode = p.postcode;
```

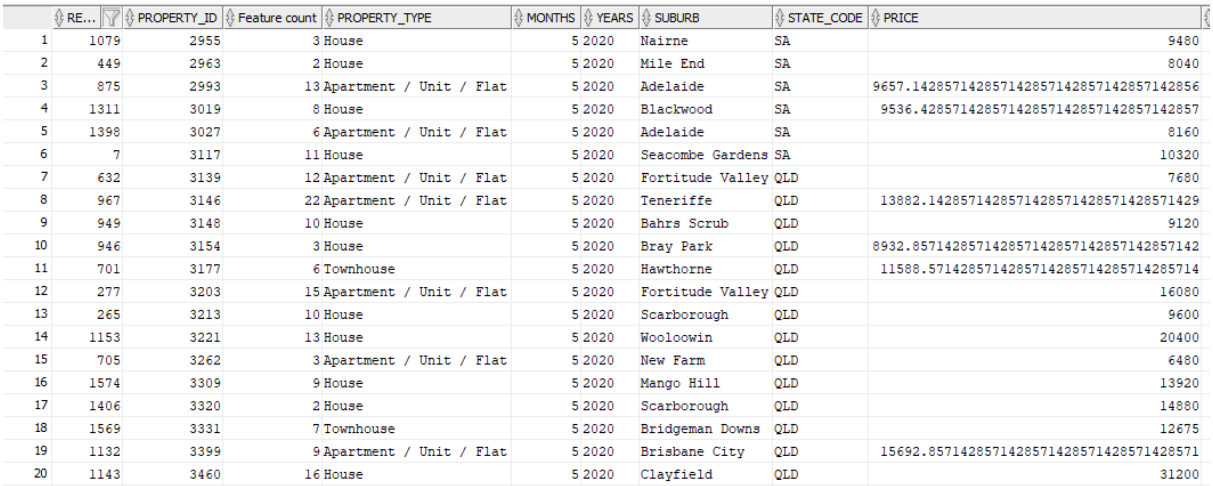


	SUBURB	STATE_CODE
1	Ashgrove	QLD
2	Aspley	QLD
3	Marsden	QLD
4	Banyo	QLD
5	Ascot	QLD
6	Elimbah	QLD
7	Indooroopilly	QLD
8	Woodridge	QLD
9	Kangaroo Point	QLD
10	West End	QLD
11	West End	QLD
12	Lota	QLD
13	Eatons Hill	QLD
14	Wynnum	QLD
15	Loganlea	QLD
16	North Lakes	QLD
17	Burpengary East	QLD
18	Yeerongpilly	QLD
19	Caboolture	QLD

Rent Temp Fact table

```
CREATE TABLE rent_tempfact_lvl2 AS
   SELECT rent_id, r.property_id, COUNT(feature_code) AS "Feature count", property_type,
floor(months_between(TO_DATE(rent_end_date,'dd-mm-
yyyy'),TO_DATE(rent_start_date,'dd-mm-yyyy'))) AS Months,
   to_char(rent_start_date,'yyyy') AS years, ad.suburb, pc.state_code,
price*((TO_DATE(rent_end_date,'dd-mm-yyyy')-TO_DATE(rent_start_date,'dd-mm-yyyy'))/7)
AS price, p.property_no_of_bedrooms    FROM rent r
   JOIN property p ON r.property_id=p.property_id
   JOIN property_feature pf ON p.property_id=pf.property_id
   JOIN address ad ON ad.address_id=p.address_id
   JOIN postcode pc ON ad.postcode=pc.postcode
   WHERE NOT r.rent_start_date IS NULL
   GROUP BY (rent_id, r.property_id, property_type,
months_between(TO_DATE(rent_end_date,'dd-mm-yyyy'),TO_DATE(rent_start_date,'dd-
mm-yyyy')),
      TO_CHAR(rent_start_date,'yyyy'), price*((TO_DATE(rent_end_date,'dd-mm-yyyy')-
TO_DATE(rent_start_date,'dd-mm-yyyy'))/7), p.property_no_of_bedrooms,ad.suburb,
pc.state_code);

ALTER TABLE rent_tempfact_lvl2 ADD category VARCHAR(20);

UPDATE rent_tempfact_lvl2 SET category='Very basic' WHERE "Feature count"<10;
UPDATE rent_tempfact_lvl2 SET category='Standard' WHERE "Feature count" BETWEEN
10 AND 20;
```

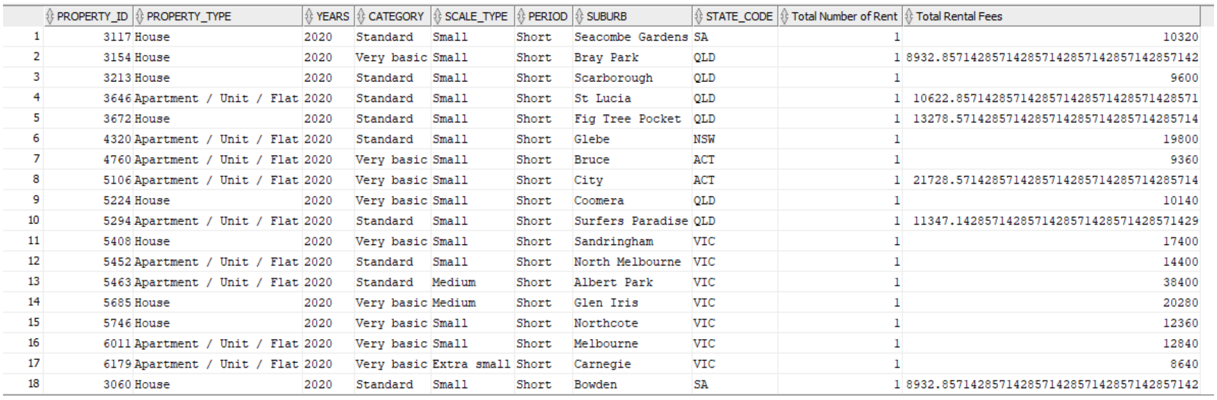
```
UPDATE rent_tempfact_lvl2 SET category='Luxurious' WHERE "Feature count">20;

ALTER TABLE rent_tempfact_lvl2 ADD scale_type VARCHAR(20);

UPDATE rent_tempfact_lvl2 SET scale_type='Extra small' WHERE
property_no_of_bedrooms<=1;
UPDATE rent_tempfact_lvl2 SET scale_type='Small' WHERE property_no_of_bedrooms
BETWEEN 2 AND 3;
UPDATE rent_tempfact_lvl2 SET scale_type='Medium' WHERE property_no_of_bedrooms
BETWEEN 4 AND 6;
UPDATE rent_tempfact_lvl2 SET scale_type='Large' WHERE property_no_of_bedrooms
BETWEEN 7 AND 10;
UPDATE rent_tempfact_lvl2 SET scale_type='Extra large' WHERE
property_no_of_bedrooms>10;

ALTER TABLE rent_tempfact_lvl2 ADD period VARCHAR(20);

UPDATE rent_tempfact_lvl2 SET period='Short' WHERE Months<6;
UPDATE rent_tempfact_lvl2 SET period='Medium' WHERE Months BETWEEN 6 AND 12;
UPDATE rent_tempfact_lvl2 SET period='Long' WHERE Months>12;
```



	RE...	PROPERTY_ID	Feature count	PROPERTY_TYPE	MONTHS	YEARS	SUBURB	STATE_CODE	PRICE
1	1079	2955	3	House	5	2020	Nairne	SA	9480
2	449	2963	2	House	5	2020	Mile End	SA	8040
3	875	2993	13	Apartment / Unit / Flat	5	2020	Adelaide	SA	9657.142857142857142857142857142857142856
4	1311	3019	8	House	5	2020	Blackwood	SA	9536.428571428571428571428571428571428857
5	1398	3027	6	Apartment / Unit / Flat	5	2020	Adelaide	SA	8160
6	7	3117	11	House	5	2020	Seacombe Gardens	SA	10320
7	632	3139	12	Apartment / Unit / Flat	5	2020	Fortitude Valley	QLD	7680
8	967	3146	22	Apartment / Unit / Flat	5	2020	Teneriffe	QLD	13882.142857142857142857142857142857142429
9	949	3148	10	House	5	2020	Bahrs Scrub	QLD	9120
10	946	3154	3	House	5	2020	Bray Park	QLD	8932.857142857142857142857142857142857142
11	701	3177	6	Townhouse	5	2020	Hawthorne	QLD	11588.571428571428571428571428571428571714
12	277	3203	15	Apartment / Unit / Flat	5	2020	Fortitude Valley	QLD	16080
13	265	3213	10	House	5	2020	Scarborough	QLD	9600
14	1153	3221	13	House	5	2020	Wooloowin	QLD	20400
15	705	3262	3	Apartment / Unit / Flat	5	2020	New Farm	QLD	6480
16	1574	3309	9	House	5	2020	Mango Hill	QLD	13920
17	1406	3320	2	House	5	2020	Scarborough	QLD	14880
18	1569	3331	7	Townhouse	5	2020	Bridgeman Downs	QLD	12675
19	1132	3399	9	Apartment / Unit / Flat	5	2020	Brisbane City	QLD	15692.857142857142857142857142857142857571
20	1143	3460	16	House	5	2020	Clayfield	QLD	31200

Rent Fact TABLE

```
CREATE TABLE rent_fact_lvl2
   AS
      SELECT
         property_id,
         property_type,
         years,
         category,
         scale_type,
         period,
         suburb,
         state_code,
         COUNT(rent_id) AS "Total Number of Rent",
         SUM(price) AS "Total Rental Fees"
      FROM
         rent_tempfact_lvl2
      GROUP BY (
         property_id,
         property_type,
         years,
         category,
         scale_type,
         period,
         suburb,
         state_code
      );
```

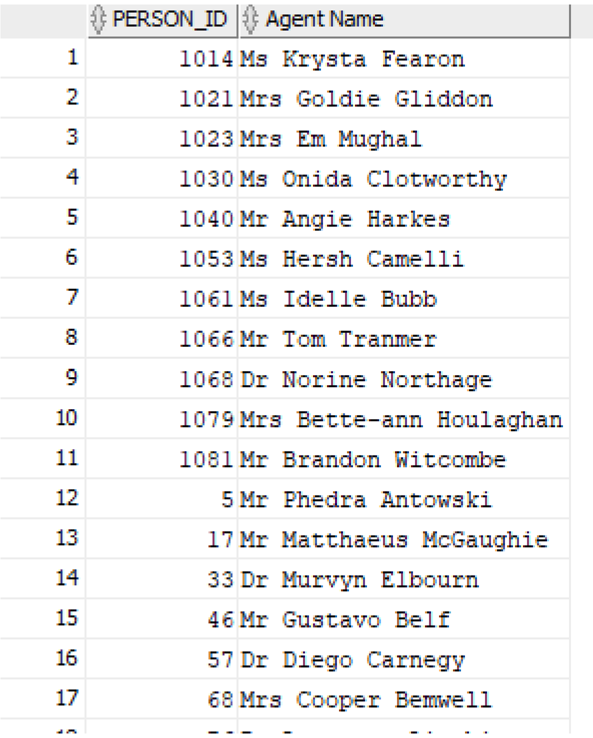


	PROPERTY_ID	PROPERTY_TYPE	YEARS	CATEGORY	SCALE_TYPE	PERIOD	SUBURB	STATE_CODE	Total Number of Rent	Total Rental Fees
1	3117	House	2020	Standard	Small	Short	Seacombe Gardens	SA	1	10320
2	3154	House	2020	Very basic	Small	Short	Bray Park	QLD	1	8932.857142857142857142857142857142857142
3	3213	House	2020	Standard	Small	Short	Scarborough	QLD	1	9600
4	3646	Apartment / Unit / Flat	2020	Standard	Small	Short	St Lucia	QLD	1	10622.857142857142857142857142857142857143
5	3672	House	2020	Standard	Small	Short	Fig Tree Pocket	QLD	1	13278.571428571428571428571428571428571428
6	4320	Apartment / Unit / Flat	2020	Standard	Small	Short	Glebe	NSW	1	19800
7	4760	Apartment / Unit / Flat	2020	Very basic	Small	Short	Bruce	ACT	1	9360
8	5106	Apartment / Unit / Flat	2020	Very basic	Small	Short	City	ACT	1	21728.571428571428571428571428571428571428
9	5224	House	2020	Very basic	Small	Short	Coomera	QLD	1	10140
10	5294	Apartment / Unit / Flat	2020	Standard	Small	Short	Surfers Paradise	QLD	1	11347.142857142857142857142857142857142857
11	5408	House	2020	Very basic	Small	Short	Sandringham	VIC	1	17400
12	5452	Apartment / Unit / Flat	2020	Standard	Small	Short	North Melbourne	VIC	1	14400
13	5463	Apartment / Unit / Flat	2020	Standard	Medium	Short	Albert Park	VIC	1	38400
14	5685	House	2020	Very basic	Medium	Short	Glen Iris	VIC	1	20280
15	5746	House	2020	Very basic	Small	Short	Northcote	VIC	1	12360
16	6011	Apartment / Unit / Flat	2020	Very basic	Small	Short	Melbourne	VIC	1	12840
17	6179	Apartment / Unit / Flat	2020	Very basic	Extra small	Short	Carnegie	VIC	1	8640
18	3060	House	2020	Standard	Small	Short	Bowden	SA	1	8932.857142857142857142857142857142857142

Agent Dimensions and Fact creation

Agent Information Dimension

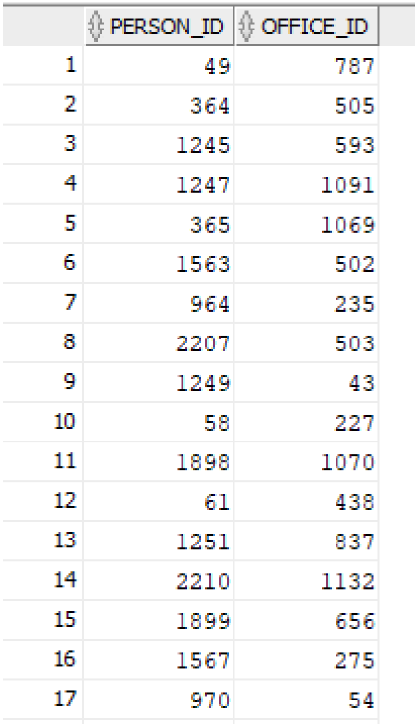
```
CREATE TABLE agent_info_dim_lvl2
    AS
        SELECT DISTINCT
            ( a.person_id ),
            p.title
            || ' '
            || p.first_name
            || ' '
            || p.last_name AS "Agent Name"
        FROM
            Agent A
join
person p ON a.person_id = p.person_id;
```



	PERSON_ID	Agent Name
1	1014	Ms Krysta Fearon
2	1021	Mrs Goldie Gliddon
3	1023	Mrs Em Mughal
4	1030	Ms Onida Clotworthy
5	1040	Mr Angie Harkes
6	1053	Ms Hersh Camelli
7	1061	Ms Idelle Bubb
8	1066	Mr Tom Tranmer
9	1068	Dr Norine Northage
10	1079	Mrs Bette-ann Houlaghan
11	1081	Mr Brandon Witcombe
12	5	Mr Phedra Antowski
13	17	Mr Matthaeus McGaughie
14	33	Dr Murvyn Elbourn
15	46	Mr Gustavo Belf
16	57	Dr Diego Carnegy
17	68	Mrs Cooper Bemwell

Agent Office Bridge Dimension

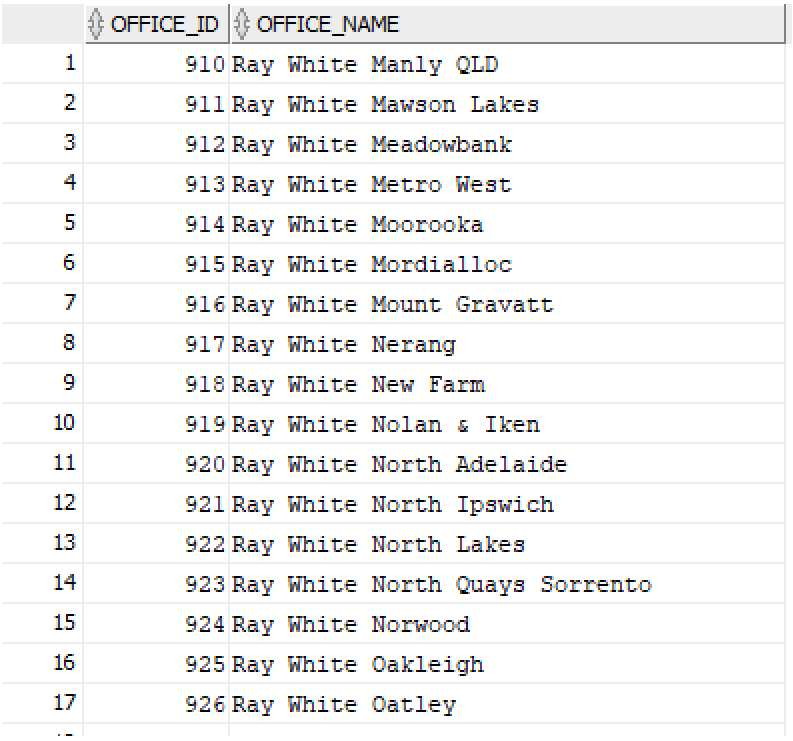
```
CREATE TABLE agent_office_bridge_dim_lvl2
   AS
      SELECT
         person_id,
         office_id
      FROM
         agent_office;
```



	PERSON_ID	OFFICE_ID
1	49	787
2	364	505
3	1245	593
4	1247	1091
5	365	1069
6	1563	502
7	964	235
8	2207	503
9	1249	43
10	58	227
11	1898	1070
12	61	438
13	1251	837
14	2210	1132
15	1899	656
16	1567	275
17	970	54

Office Dimension

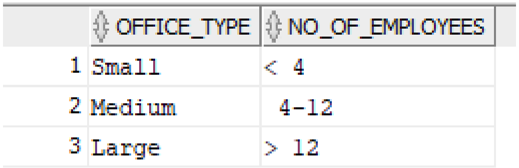
```
CREATE TABLE office_dim_lvl2
    AS
        SELECT
            office_id,
            office_name
        FROM
            office;
```



	OFFICE_ID	OFFICE_NAME
1	910	Ray White Manly QLD
2	911	Ray White Mawson Lakes
3	912	Ray White Meadowbank
4	913	Ray White Metro West
5	914	Ray White Moorooka
6	915	Ray White Mordialloc
7	916	Ray White Mount Gravatt
8	917	Ray White Nerang
9	918	Ray White New Farm
10	919	Ray White Nolan & Iken
11	920	Ray White North Adelaide
12	921	Ray White North Ipswich
13	922	Ray White North Lakes
14	923	Ray White North Quays Sorrento
15	924	Ray White Norwood
16	925	Ray White Oakleigh
17	926	Ray White Oatley

Office Size Dimension

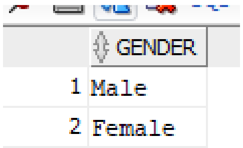
```
CREATE TABLE office_size_dim_lvl2 (
    office_type       VARCHAR2(30),
    no_of_employees  VARCHAR2(40)
);

INSERT INTO office_size_dim_lvl2 VALUES (
    'Small',
    '< 4'
);

INSERT INTO office_size_dim_lvl2 VALUES (
    'Medium',
    ' 4-12'
);

INSERT INTO office_size_dim_lvl2 VALUES (
    'Large',
    '> 12'
);
```



	OFFICE_TYPE	NO_OF_EMPLOYEES
1	Small	< 4
2	Medium	4-12
3	Large	> 12

Gender Dimension

```
CREATE TABLE gender_dim_lvl2
    AS
        SELECT DISTINCT
            gender
        FROM
            person;
```



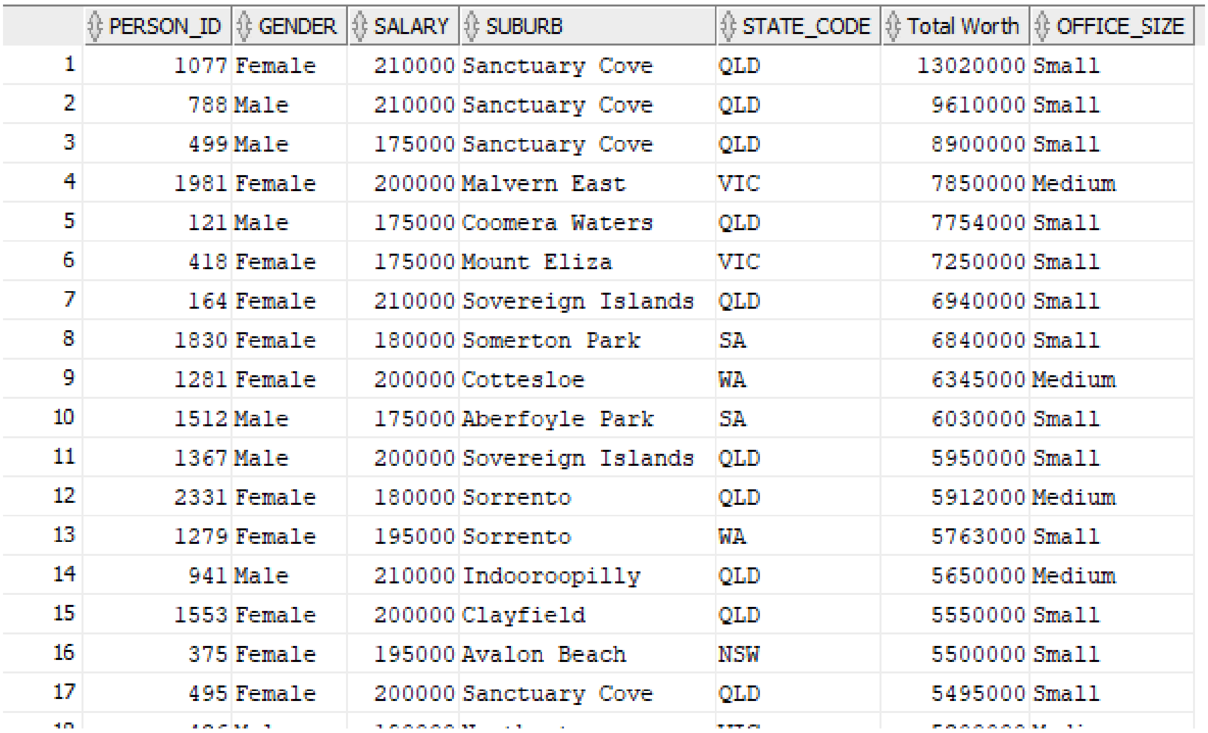
	GENDER
1	Male
2	Female

Agent Temp fact

```
CREATE TABLE agent_tempfact_lvl2
    AS
        SELECT
            person_id,
            gender,
            salary,
            suburb,
            state_code,
            SUM(price) "Total Worth"
        FROM
            (
                SELECT
                    a.person_id,
                    pe.gender,
                    a.salary,
                    ad.suburb,
                    pc.state_code,
                    s.price
                FROM
                    agent          a
                    LEFT JOIN sale          s ON a.person_id = s.agent_person_id
                    LEFT JOIN property      p ON s.property_id = p.property_id
                    LEFT JOIN address       ad ON p.address_id = ad.address_id
                    LEFT JOIN postcode      pc ON pc.postcode = ad.postcode
                    LEFT JOIN agent_office ao ON a.person_id = ao.person_id
                    LEFT JOIN person        pe ON a.person_id = pe.person_id
                UNION
                SELECT
                    a.person_id,
                    pe.gender,
                    a.salary,
                    ad.suburb,
                    pc.state_code,
                    r.price * ( r.rent_end_date - r.rent_start_date ) / 7
                FROM
                    agent        a
                    LEFT JOIN rent         r ON a.person_id = r.agent_person_id
                    LEFT JOIN property     p ON r.property_id = p.property_id
                    LEFT JOIN address      ad ON p.address_id = ad.address_id
                    LEFT JOIN postcode     pc ON pc.postcode = ad.postcode
                    LEFT JOIN agent_office ao ON a.person_id = ao.person_id
                    LEFT JOIN person       pe ON a.person_id = pe.person_id
            )
        WHERE
            price IS NOT NULL
        GROUP BY
            person_id,
            gender,
            suburb,
            state_code,
            salary
        ORDER BY
            SUM(price) DESC;

ALTER TABLE agent_tempfact_lvl2 ADD office_size VARCHAR(10);

SELECT
    a.person_id
FROM
    agent_tempfact_lvl2      a
    JOIN agent_office_bridge_dim b ON a.person_id = b.person_id
WHERE
    b.office_id IN (
        SELECT
            office_id
        FROM
            agent_office
        GROUP BY
            office_id
        HAVING
            COUNT(person_id) < 4
    );

UPDATE agent_tempfact_lvl2
SET
```

```
    office_size = 'Small'
WHERE
    person_id IN (
        SELECT
            a.person_id
        FROM
            agent_tempfact_lvl2 a
            JOIN agent_office          b ON a.person_id = b.person_id
        WHERE
            b.office_id IN (
                SELECT
                    office_id
                FROM
                    agent_office
                GROUP BY
                    office_id
                HAVING
                    COUNT(person_id) < 4
            )
    );

UPDATE agent_tempfact_lvl2
SET
    office_size = 'Medium'
WHERE
    person_id IN (
        SELECT
            a.person_id
        FROM
            agent_tempfact_lvl2 a
            JOIN agent_office          b ON a.person_id = b.person_id
        WHERE
            b.office_id IN (
                SELECT
                    office_id
                FROM
                    agent_office
                GROUP BY
                    office_id
                HAVING
                    COUNT(person_id) BETWEEN 4 AND 12
            )
    );

UPDATE agent_tempfact_lvl2
SET
    office_size = 'Big'
WHERE
```

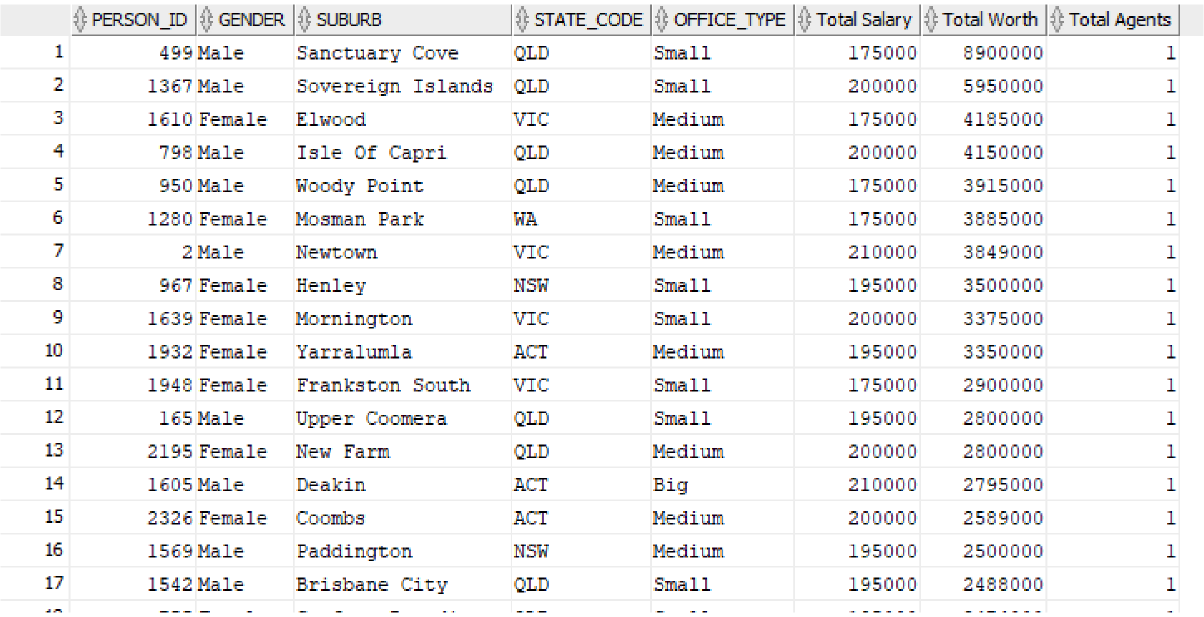
```
    person_id IN (
        SELECT
            a.person_id
        FROM
            agent_tempfact_lvl2 a
            JOIN agent_office          b ON a.person_id = b.person_id
        WHERE
            b.office_id IN (
                SELECT
                    office_id
                FROM
                    agent_office
                GROUP BY
                    office_id
                HAVING
                    COUNT(person_id) > 12
            )
    );
```



	PERSON_ID	GENDER	SALARY	SUBURB	STATE_CODE	Total Worth	OFFICE_SIZE
1	1077	Female	210000	Sanctuary Cove	QLD	13020000	Small
2	788	Male	210000	Sanctuary Cove	QLD	9610000	Small
3	499	Male	175000	Sanctuary Cove	QLD	8900000	Small
4	1981	Female	200000	Malvern East	VIC	7850000	Medium
5	121	Male	175000	Coomera Waters	QLD	7754000	Small
6	418	Female	175000	Mount Eliza	VIC	7250000	Small
7	164	Female	210000	Sovereign Islands	QLD	6940000	Small
8	1830	Female	180000	Somerton Park	SA	6840000	Small
9	1281	Female	200000	Cottesloe	WA	6345000	Medium
10	1512	Male	175000	Aberfoyle Park	SA	6030000	Small
11	1367	Male	200000	Sovereign Islands	QLD	5950000	Small
12	2331	Female	180000	Sorrento	QLD	5912000	Medium
13	1279	Female	195000	Sorrento	WA	5763000	Small
14	941	Male	210000	Indooroopilly	QLD	5650000	Medium
15	1553	Female	200000	Clayfield	QLD	5550000	Small
16	375	Female	195000	Avalon Beach	NSW	5500000	Small
17	495	Female	200000	Sanctuary Cove	QLD	5495000	Small

Agent Fact TABLE

```
CREATE TABLE agent_fact_lvl2
    AS
        SELECT
            person_id,
            gender,
            suburb,
            state_code,
            office_size AS office_type,
            SUM(salary) "Total Salary",
            SUM("Total Worth") "Total Worth",
            COUNT(DISTINCT person_id) "Total Agents"
        FROM
            agent_tempfact_lvl2
        GROUP BY (
            person_id,
            gender,
            suburb,
            state_code,
            office_size
        );
```

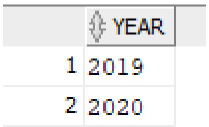


	PERSON_ID	GENDER	SUBURB	STATE_CODE	OFFICE_TYPE	Total Salary	Total Worth	Total Agents
1	499	Male	Sanctuary Cove	QLD	Small	175000	8900000	1
2	1367	Male	Sovereign Islands	QLD	Small	200000	5950000	1
3	1610	Female	Elwood	VIC	Medium	175000	4185000	1
4	798	Male	Isle Of Capri	QLD	Medium	200000	4150000	1
5	950	Male	Woody Point	QLD	Medium	175000	3915000	1
6	1280	Female	Mosman Park	WA	Small	175000	3885000	1
7	2	Male	Newtown	VIC	Medium	210000	3849000	1
8	967	Female	Henley	NSW	Small	195000	3500000	1
9	1639	Female	Mornington	VIC	Small	200000	3375000	1
10	1932	Female	Yarralumla	ACT	Medium	195000	3350000	1
11	1948	Female	Frankston South	VIC	Small	175000	2900000	1
12	165	Male	Upper Coomera	QLD	Small	195000	2800000	1
13	2195	Female	New Farm	QLD	Medium	200000	2800000	1
14	1605	Male	Deakin	ACT	Big	210000	2795000	1
15	2326	Female	Coombs	ACT	Medium	200000	2589000	1
16	1569	Male	Paddington	NSW	Medium	195000	2500000	1
17	1542	Male	Brisbane City	QLD	Small	195000	2488000	1

Sale Dimensions and Fact creation

Sale year Dimension

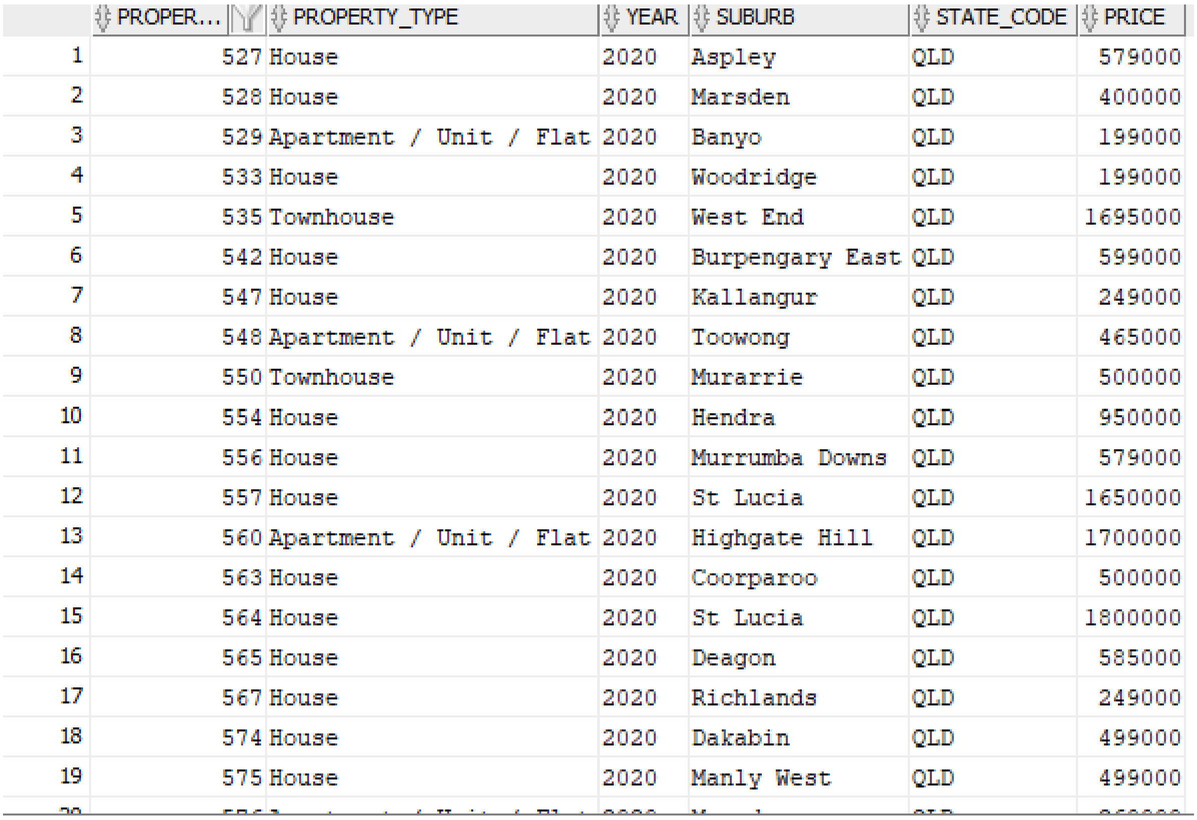
```
CREATE TABLE sale_year_dim_lvl2
  AS
    SELECT DISTINCT
        to_char(sale_date, 'yyyy') AS year
    FROM
        sale
    WHERE
        NOT sale_date IS NULL;
```



	YEAR
1	2019
2	2020

Sale Temp fact TABLE

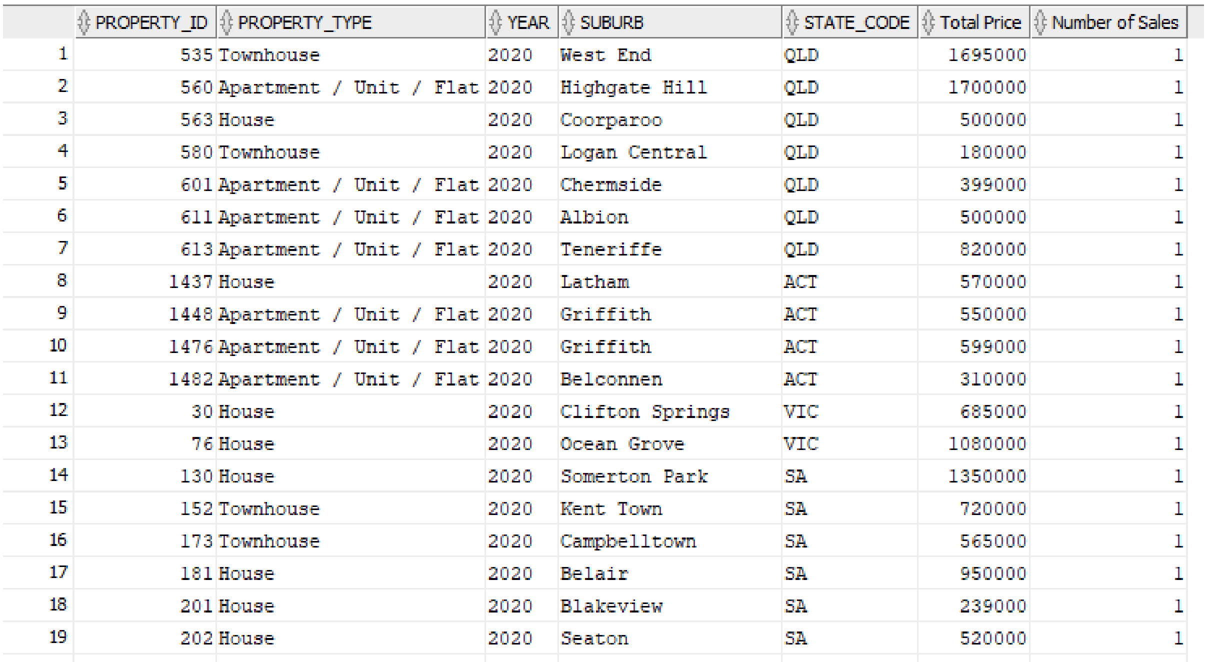
```
CREATE TABLE sale_tempfact_lvl2
  AS
    SELECT
      s.property_id,
      p.property_type,
      to_char(sale_date, 'yyyy') AS year,
      ad.suburb,
      pc.state_code,
      s.price
    FROM
      sale       s
      JOIN property p ON s.property_id = p.property_id
      JOIN address    ad ON ad.address_id = p.address_id
      JOIN postcode pc ON pc.postcode = ad.postcode
    WHERE
      NOT s.client_person_id IS NULL;
```



	PROPER...	PROPERTY_TYPE	YEAR	SUBURB	STATE_CODE	PRICE
1	527	House	2020	Aspley	QLD	579000
2	528	House	2020	Marsden	QLD	400000
3	529	Apartment / Unit / Flat	2020	Banyo	QLD	199000
4	533	House	2020	Woodridge	QLD	199000
5	535	Townhouse	2020	West End	QLD	1695000
6	542	House	2020	Burpengary East	QLD	599000
7	547	House	2020	Kallangur	QLD	249000
8	548	Apartment / Unit / Flat	2020	Toowong	QLD	465000
9	550	Townhouse	2020	Murarrie	QLD	500000
10	554	House	2020	Hendra	QLD	950000
11	556	House	2020	Murrumba Downs	QLD	579000
12	557	House	2020	St Lucia	QLD	1650000
13	560	Apartment / Unit / Flat	2020	Highgate Hill	QLD	1700000
14	563	House	2020	Coorparoo	QLD	500000
15	564	House	2020	St Lucia	QLD	1800000
16	565	House	2020	Deagon	QLD	585000
17	567	House	2020	Richlands	QLD	249000
18	574	House	2020	Dakabin	QLD	499000
19	575	House	2020	Manly West	QLD	499000

Sale Fact TABLE

```
CREATE TABLE sale_fact_lvl2
  AS
    SELECT
      property_id,
      property_type,
      year,
      suburb,
      state_code,
      SUM(price) "Total Price",
      COUNT(property_id) "Number of Sales"
    FROM
      sale_tempfact_lvl2
    GROUP BY
      property_id,
      property_type,
      year,
      suburb,
      state_code;
```



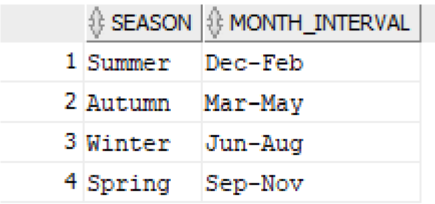
	PROPERTY_ID	PROPERTY_TYPE	YEAR	SUBURB	STATE_CODE	Total Price	Number of Sales
1	535	Townhouse	2020	West End	QLD	1695000	1
2	560	Apartment / Unit / Flat	2020	Highgate Hill	QLD	1700000	1
3	563	House	2020	Coorparoo	QLD	500000	1
4	580	Townhouse	2020	Logan Central	QLD	180000	1
5	601	Apartment / Unit / Flat	2020	Chermside	QLD	399000	1
6	611	Apartment / Unit / Flat	2020	Albion	QLD	500000	1
7	613	Apartment / Unit / Flat	2020	Teneriffe	QLD	820000	1
8	1437	House	2020	Latham	ACT	570000	1
9	1448	Apartment / Unit / Flat	2020	Griffith	ACT	550000	1
10	1476	Apartment / Unit / Flat	2020	Griffith	ACT	599000	1
11	1482	Apartment / Unit / Flat	2020	Belconnen	ACT	310000	1
12	30	House	2020	Clifton Springs	VIC	685000	1
13	76	House	2020	Ocean Grove	VIC	1080000	1
14	130	House	2020	Somerton Park	SA	1350000	1
15	152	Townhouse	2020	Kent Town	SA	720000	1
16	173	Townhouse	2020	Campbelltown	SA	565000	1
17	181	House	2020	Belair	SA	950000	1
18	201	House	2020	Blakeview	SA	239000	1
19	202	House	2020	Seaton	SA	520000	1

Visit Dimensions and Fact creation

Season dimension

```
CREATE TABLE season_dim_lvl2 (
    season          VARCHAR(10),
    month_interval  VARCHAR(20)
);

INSERT INTO season_dim_lvl2 VALUES (
    'Summer',
    'Dec-Feb'
);
```

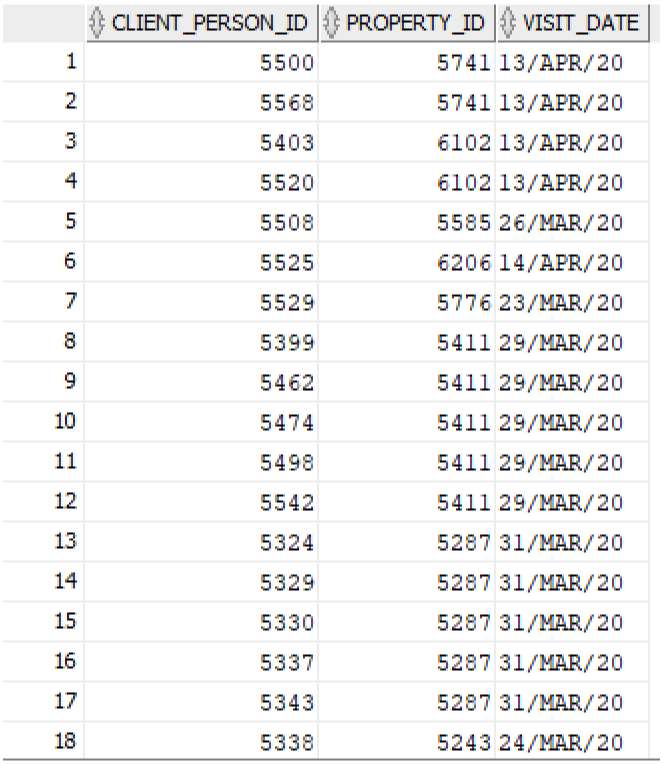
```
INSERT INTO season_dim_lvl2 VALUES (
    'Autumn',
    'Mar-May'
);

INSERT INTO season_dim_lvl2 VALUES (
    'Winter',
    'Jun-Aug'
);
```



	SEASON	MONTH_INTERVAL
1	Summer	Dec-Feb
2	Autumn	Mar-May
3	Winter	Jun-Aug
4	Spring	Sep-Nov

Client Visit SCD Dimension

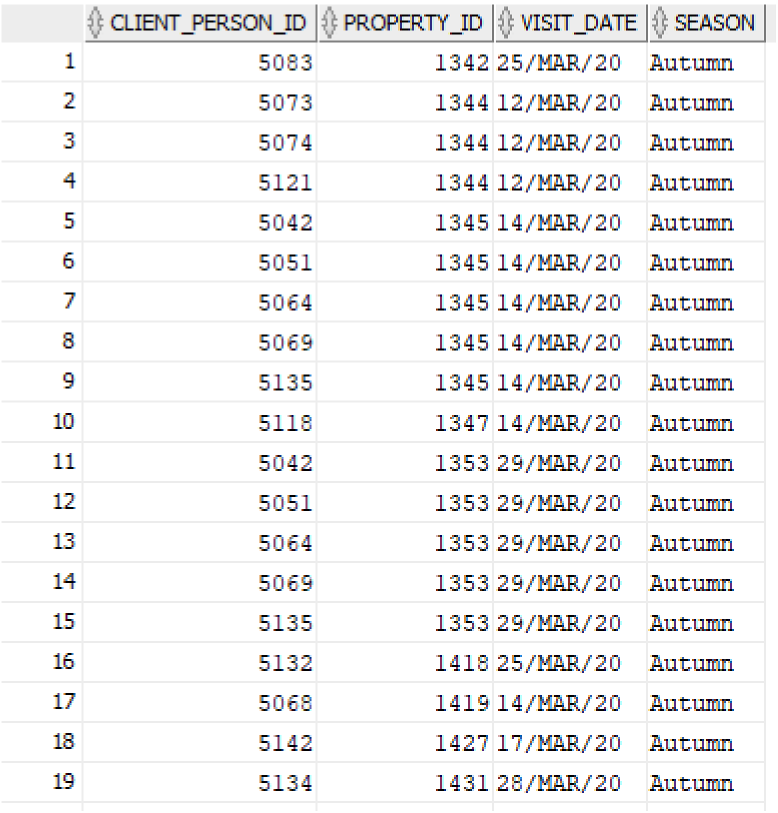
```
CREATE TABLE client_visit_dim_scd_lvl2 AS SELECT
    client_person_id,
    property_id, visit_date FROM
visit;
```



	CLIENT_PERSON_ID	PROPERTY_ID	VISIT_DATE
1	5500	5741	13/APR/20
2	5568	5741	13/APR/20
3	5403	6102	13/APR/20
4	5520	6102	13/APR/20
5	5508	5585	26/MAR/20
6	5525	6206	14/APR/20
7	5529	5776	23/MAR/20
8	5399	5411	29/MAR/20
9	5462	5411	29/MAR/20
10	5474	5411	29/MAR/20
11	5498	5411	29/MAR/20
12	5542	5411	29/MAR/20
13	5324	5287	31/MAR/20
14	5329	5287	31/MAR/20
15	5330	5287	31/MAR/20
16	5337	5287	31/MAR/20
17	5343	5287	31/MAR/20
18	5338	5243	24/MAR/20

Visit Temp fact TABLE

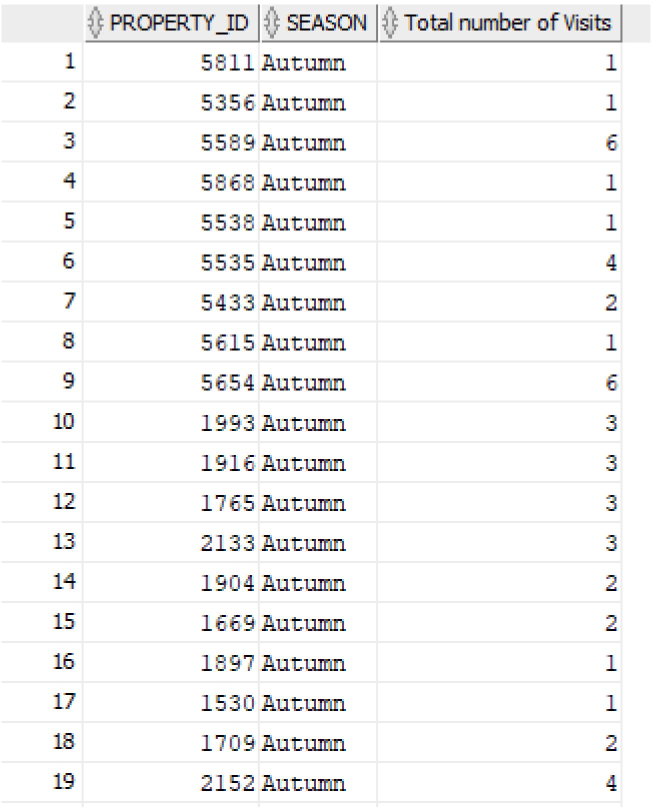
```
CREATE TABLE visit_tempfact_lvl2
    AS
    SELECT
      v.client_person_id,
      p.property_id,
      v.visit_date
    FROM
      visit      v
      JOIN property p ON p.property_id = v.property_id;

ALTER TABLE visit_tempfact_lvl2 ADD season VARCHAR(10);

UPDATE visit_tempfact_lvl2
SET
  season = 'Summer'
WHERE
  to_char(visit_date, 'mon') IN (
    'dec',
    'jan',
    'feb'
  );

UPDATE visit_tempfact_lvl2
SET
  season = 'Autumn'
WHERE
  to_char(visit_date, 'mon') IN (
    'mar',
    'apr',
    'may'
  );

UPDATE visit_tempfact_lvl2
SET
  season = 'Winter'
WHERE
  to_char(visit_date, 'mon') IN (
    'jun',
    'jul',
    'aug'
  );

UPDATE visit_tempfact_lvl2
SET
  season = 'Spring'
WHERE
  to_char(visit_date, 'mon') IN (
    'sep',
    'oct',
    'nov'
);
```



	CLIENT_PERSON_ID	PROPERTY_ID	VISIT_DATE	SEASON
1	5083	1342	25/MAR/20	Autumn
2	5073	1344	12/MAR/20	Autumn
3	5074	1344	12/MAR/20	Autumn
4	5121	1344	12/MAR/20	Autumn
5	5042	1345	14/MAR/20	Autumn
6	5051	1345	14/MAR/20	Autumn
7	5064	1345	14/MAR/20	Autumn
8	5069	1345	14/MAR/20	Autumn
9	5135	1345	14/MAR/20	Autumn
10	5118	1347	14/MAR/20	Autumn
11	5042	1353	29/MAR/20	Autumn
12	5051	1353	29/MAR/20	Autumn
13	5064	1353	29/MAR/20	Autumn
14	5069	1353	29/MAR/20	Autumn
15	5135	1353	29/MAR/20	Autumn
16	5132	1418	25/MAR/20	Autumn
17	5068	1419	14/MAR/20	Autumn
18	5142	1427	17/MAR/20	Autumn
19	5134	1431	28/MAR/20	Autumn

Visit Fact TABLE

```
CREATE TABLE visit_fact_lvl2
  AS
    SELECT
      property_id,
      season,
      COUNT(visit_date) "Total number of Visits"
    FROM
      visit_tempfact_lvl2
    GROUP BY
      property_id,
      season;
```



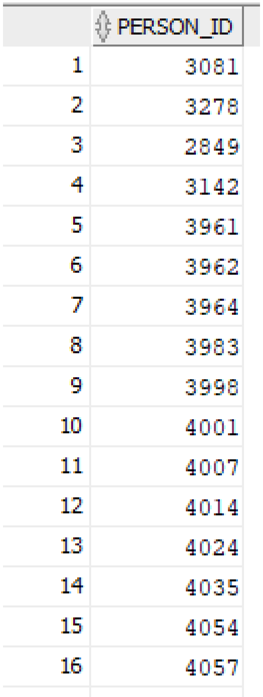
	PROPERTY_ID	SEASON	Total number of Visits
1	5811	Autumn	1
2	5356	Autumn	1
3	5589	Autumn	6
4	5868	Autumn	1
5	5538	Autumn	1
6	5535	Autumn	4
7	5433	Autumn	2
8	5615	Autumn	1
9	5654	Autumn	6
10	1993	Autumn	3
11	1916	Autumn	3
12	1765	Autumn	3
13	2133	Autumn	3
14	1904	Autumn	2
15	1669	Autumn	2
16	1897	Autumn	1
17	1530	Autumn	1
18	1709	Autumn	2
19	2152	Autumn	4

SQL statement to create star/snowflake schema Version – 2 (Level 0)

Client Dimensions and Fact creation

Client Dimension

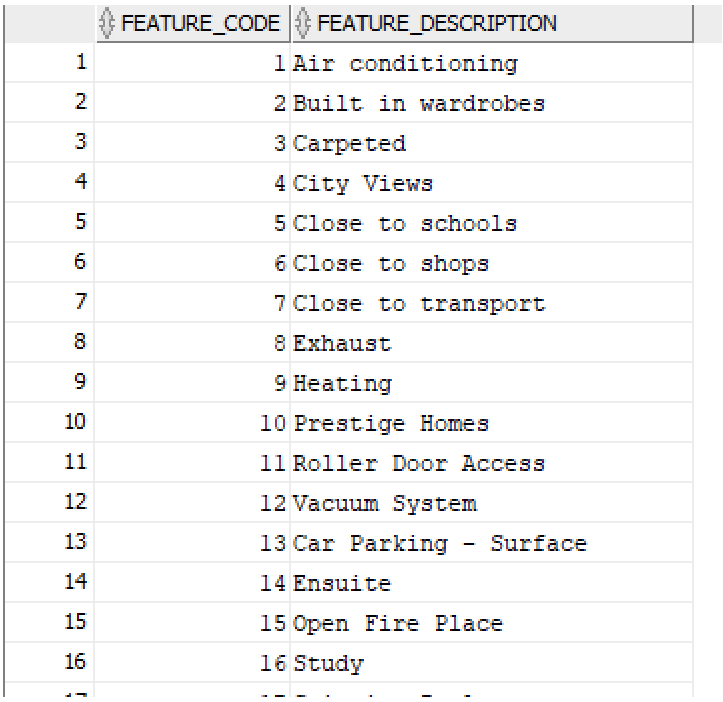
```
CREATE TABLE client_dim_lvl0
    AS
        SELECT DISTINCT
            person_id
        FROM
            client;
```



	PERSON_ID
1	3081
2	3278
3	2849
4	3142
5	3961
6	3962
7	3964
8	3983
9	3998
10	4001
11	4007
12	4014
13	4024
14	4035
15	4054
16	4057

Feature Dimension

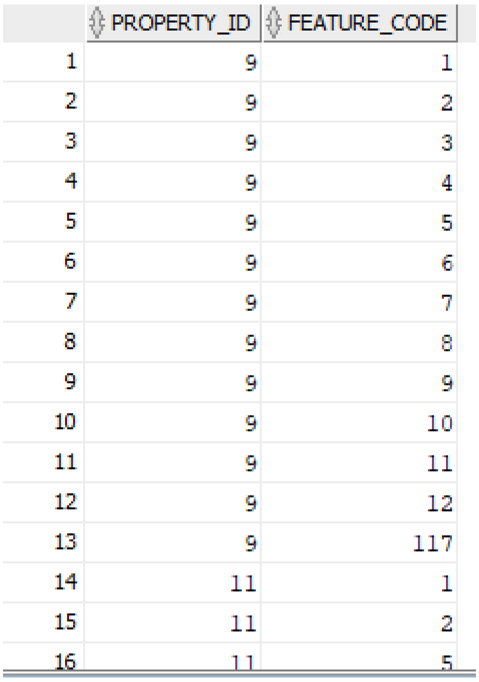
```
CREATE TABLE feature_dim_lvl0
   AS
      SELECT
         feature_code,
         feature_description
      FROM
         feature;
```



	FEATURE_CODE	FEATURE_DESCRIPTION
1	1	Air conditioning
2	2	Built in wardrobes
3	3	Carpeted
4	4	City Views
5	5	Close to schools
6	6	Close to shops
7	7	Close to transport
8	8	Exhaust
9	9	Heating
10	10	Prestige Homes
11	11	Roller Door Access
12	12	Vacuum System
13	13	Car Parking - Surface
14	14	Ensuite
15	15	Open Fire Place
16	16	Study

Property Feature Dimension

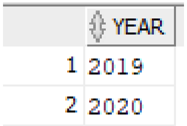
```
CREATE TABLE property_feature_dim_lvl0
    AS
        SELECT
            property_id,
            feature_code
        FROM
            property_feature;
```



	PROPERTY_ID	FEATURE_CODE
1	9	1
2	9	2
3	9	3
4	9	4
5	9	5
6	9	6
7	9	7
8	9	8
9	9	9
10	9	10
11	9	11
12	9	12
13	9	117
14	11	1
15	11	2
16	11	5

Client Year Dimension Table

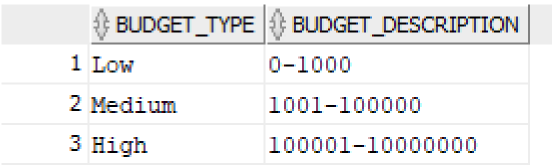
```
CREATE TABLE client_year_dim_lvl0
    AS
        SELECT DISTINCT
            year
        FROM
            (
                SELECT
                    to_char(rent_start_date, 'yyyy') AS year
                FROM
                    rent
                UNION
                SELECT
                    to_char(sale_date, 'yyyy') AS year
                FROM
                    sale
                UNION
                SELECT
                    to_char(visit_date, 'yyyy') AS year
                FROM
                    visit
            )
        WHERE
            NOT year IS NULL;
```



	YEAR
1	2019
2	2020

Budget type dimension

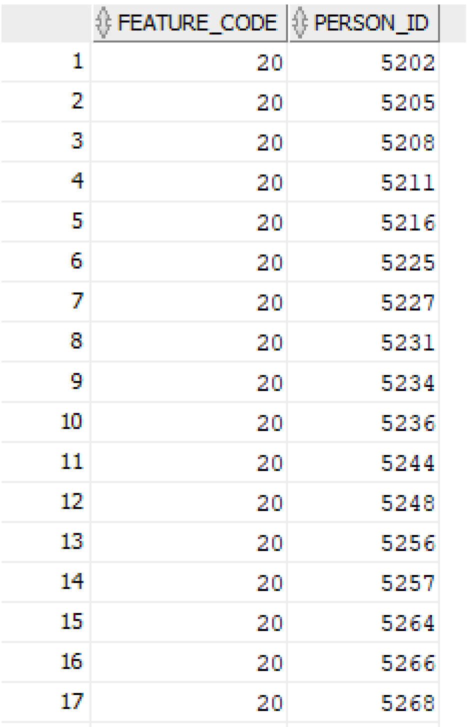
```
CREATE TABLE budget_type_dim_lvl0 (
    budget_type         VARCHAR(10),
    budget_description  VARCHAR(20)
);

INSERT INTO budget_type_dim_lvl0 VALUES (
    'Low',
    '0-1000'
);

INSERT INTO budget_type_dim_lvl0 VALUES (
    'Medium',
    '1001-100000'
);

INSERT INTO budget_type_dim_lvl0 VALUES (
    'High',
    '100001-10000000'
);
```



	BUDGET_TYPE	BUDGET_DESCRIPTION
1	Low	0-1000
2	Medium	1001-100000
3	High	100001-10000000

Client wish list dimension

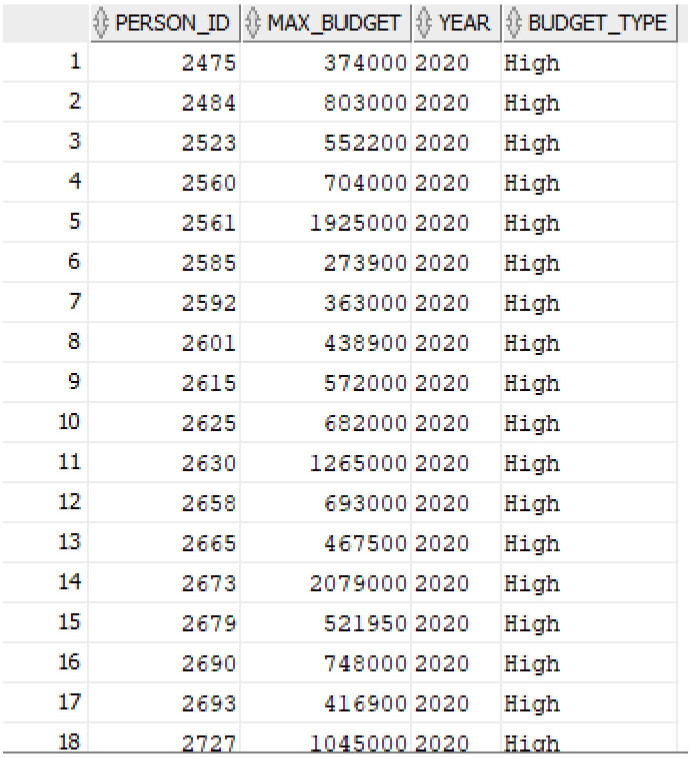
```
CREATE TABLE client_wishlist_dim_lvl0
    AS
        SELECT
            *
        FROM
            client_wish;
```



	FEATURE_CODE	PERSON_ID
1	20	5202
2	20	5205
3	20	5208
4	20	5211
5	20	5216
6	20	5225
7	20	5227
8	20	5231
9	20	5234
10	20	5236
11	20	5244
12	20	5248
13	20	5256
14	20	5257
15	20	5264
16	20	5266
17	20	5268

Client Temp fact Table

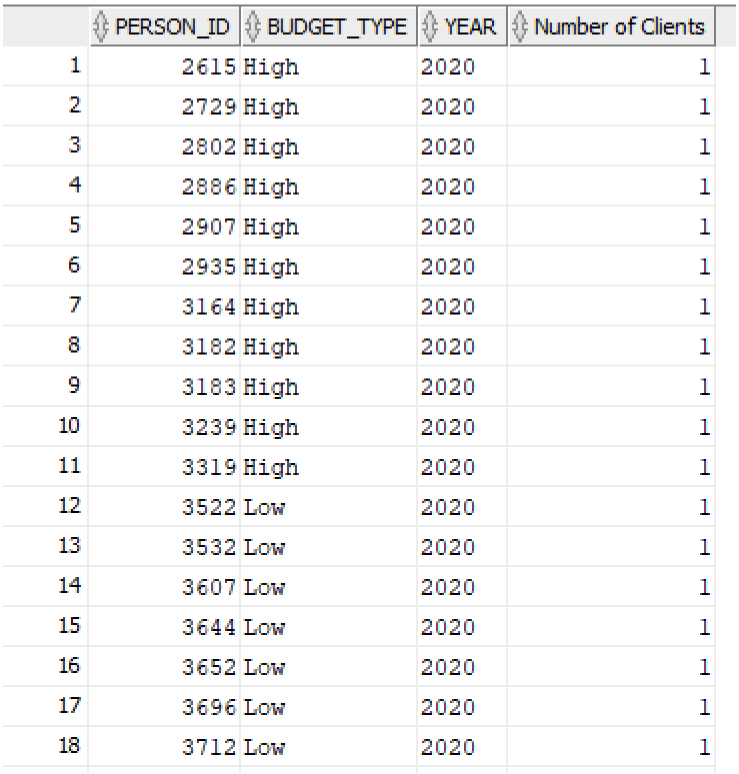
```
CREATE TABLE client_tempfact_lvl0
    AS
        SELECT
            person_id,
            max_budget,
            MIN(year) AS year
        FROM
            (
                SELECT
                    person_id,
                    max_budget,
                    to_char(rent_start_date, 'yyyy') AS year
                FROM
                    client c
                    JOIN rent     r ON c.person_id = r.client_person_id
                UNION
                SELECT
                    person_id,
                    max_budget,
                    to_char(sale_date, 'yyyy') AS year
                FROM
                    client c
                    JOIN sale     s ON c.person_id = s.client_person_id
                UNION
                SELECT
                    person_id,
                    max_budget,
                    to_char(visit_date, 'yyyy') AS year
                FROM
                    client c
                    JOIN visit    v ON c.person_id = v.client_person_id
            )
        GROUP BY
            person_id,
            max_budget;

ALTER TABLE client_tempfact_lvl0 ADD budget_type VARCHAR(10);

UPDATE client_tempfact_lvl0
SET
    budget_type = 'Low'
WHERE
    max_budget BETWEEN 0 AND 1000;

UPDATE client_tempfact_lvl0
SET
    budget_type = 'Medium'
WHERE
    max_budget BETWEEN 1001 AND 100000;

UPDATE client_tempfact_lvl0
SET
    budget_type = 'High'
WHERE
    max_budget BETWEEN 100001 AND 10000000;
```



	PERSON_ID	MAX_BUDGET	YEAR	BUDGET_TYPE
1	2475	374000	2020	High
2	2484	803000	2020	High
3	2523	552200	2020	High
4	2560	704000	2020	High
5	2561	1925000	2020	High
6	2585	273900	2020	High
7	2592	363000	2020	High
8	2601	438900	2020	High
9	2615	572000	2020	High
10	2625	682000	2020	High
11	2630	1265000	2020	High
12	2658	693000	2020	High
13	2665	467500	2020	High
14	2673	2079000	2020	High
15	2679	521950	2020	High
16	2690	748000	2020	High
17	2693	416900	2020	High
18	2727	1045000	2020	High

Client Fact Table

```
CREATE TABLE client_fact_lvl0
   AS
      SELECT DISTINCT
         ( person_id ),
         budget_type,
         year,
         COUNT(person_id) "Number of Clients"
      FROM
         client_tempfact_lvl0
      GROUP BY
         person_id,
         budget_type,
         year;
```

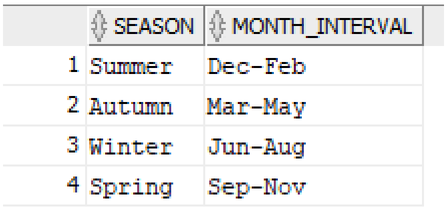


	PERSON_ID	BUDGET_TYPE	YEAR	Number of Clients
1	2615	High	2020	1
2	2729	High	2020	1
3	2802	High	2020	1
4	2886	High	2020	1
5	2907	High	2020	1
6	2935	High	2020	1
7	3164	High	2020	1
8	3182	High	2020	1
9	3183	High	2020	1
10	3239	High	2020	1
11	3319	High	2020	1
12	3522	Low	2020	1
13	3532	Low	2020	1
14	3607	Low	2020	1
15	3644	Low	2020	1
16	3652	Low	2020	1
17	3696	Low	2020	1
18	3712	Low	2020	1

Visit Dimensions and Fact creation

Season Dimension

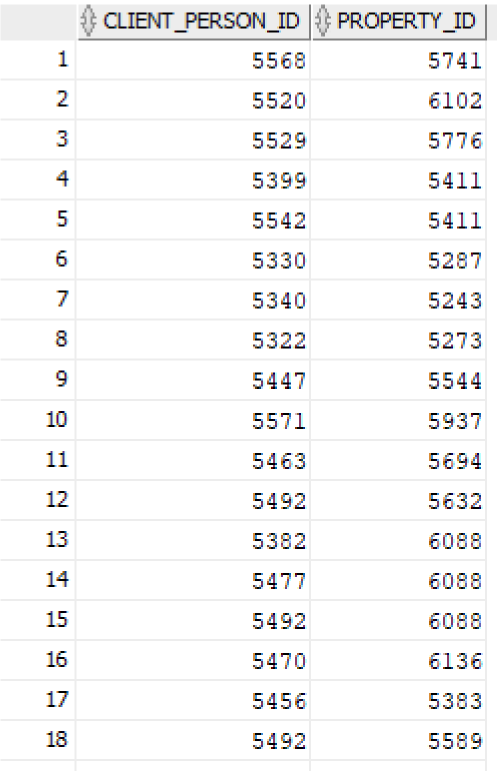
```
CREATE TABLE season_dim_lvl0 (
    season          VARCHAR(10),
    month_interval  VARCHAR(20)
);

INSERT INTO season_dim_lvl0 VALUES (
    'Summer',
    'Dec-Feb'
);

INSERT INTO season_dim_lvl0 VALUES (
    'Autumn',
    'Mar-May'
);

INSERT INTO season_dim_lvl0 VALUES (
    'Winter',
    'Jun-Aug'
);

INSERT INTO season_dim_lvl0 VALUES (
    'Spring',
    'Sep-Nov'
);
```



	SEASON	MONTH_INTERVAL
1	Summer	Dec-Feb
2	Autumn	Mar-May
3	Winter	Jun-Aug
4	Spring	Sep-Nov

Client Visit Dimension

```
CREATE TABLE client_visit_dim_lvl0
   AS
      SELECT DISTINCT
         client_person_id,
         property_id
      FROM
         visit;
```

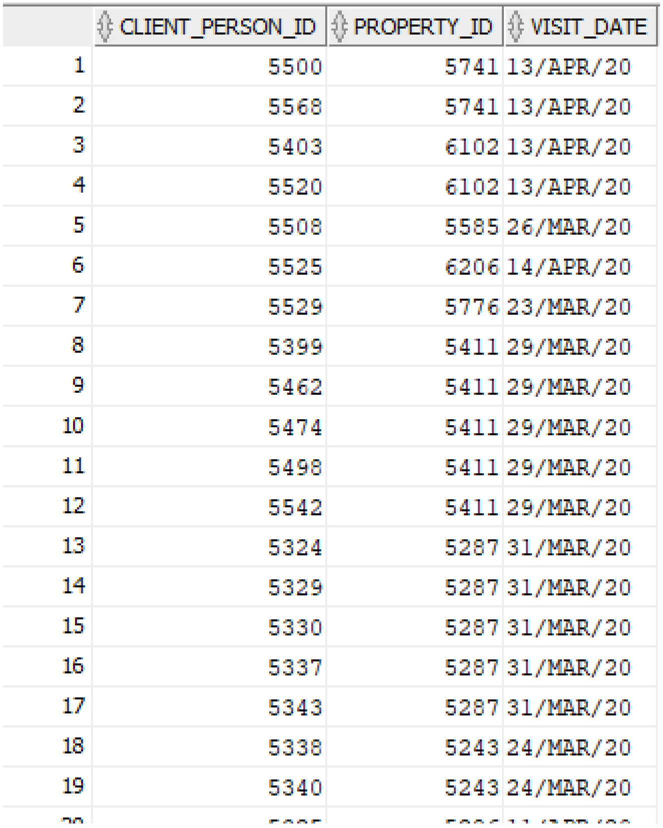


	CLIENT_PERSON_ID	PROPERTY_ID
1	5568	5741
2	5520	6102
3	5529	5776
4	5399	5411
5	5542	5411
6	5330	5287
7	5340	5243
8	5322	5273
9	5447	5544
10	5571	5937
11	5463	5694
12	5492	5632
13	5382	6088
14	5477	6088
15	5492	6088
16	5470	6136
17	5456	5383
18	5492	5589

Client Visit SCD Dimension

```
CREATE TABLE client_visit_dim_scd_lvl0
   AS
      SELECT
         client_person_id,
         property_id,
         visit_date
      FROM
```

```
        visit;
```

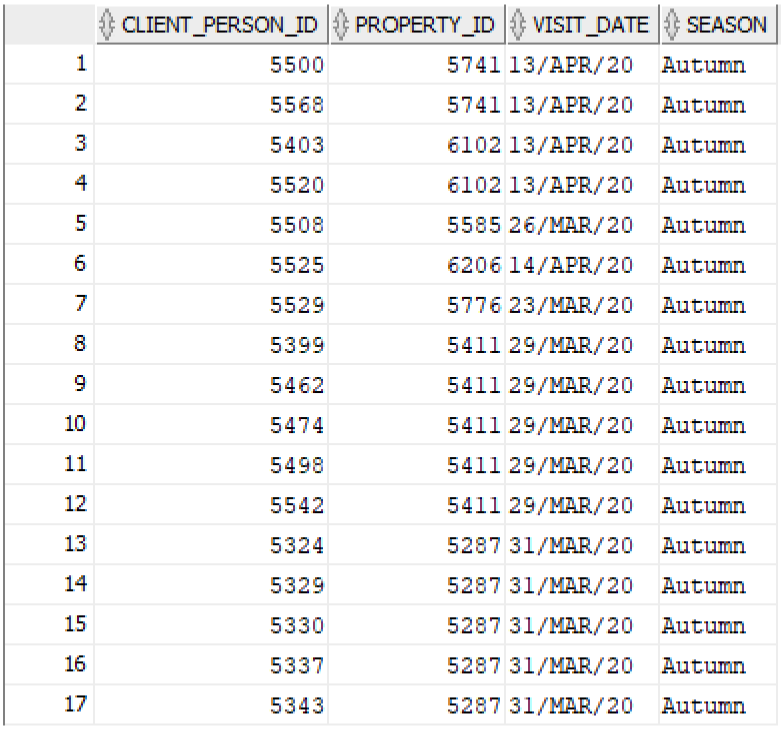


	CLIENT_PERSON_ID	PROPERTY_ID	VISIT_DATE
1	5500	5741	13/APR/20
2	5568	5741	13/APR/20
3	5403	6102	13/APR/20
4	5520	6102	13/APR/20
5	5508	5585	26/MAR/20
6	5525	6206	14/APR/20
7	5529	5776	23/MAR/20
8	5399	5411	29/MAR/20
9	5462	5411	29/MAR/20
10	5474	5411	29/MAR/20
11	5498	5411	29/MAR/20
12	5542	5411	29/MAR/20
13	5324	5287	31/MAR/20
14	5329	5287	31/MAR/20
15	5330	5287	31/MAR/20
16	5337	5287	31/MAR/20
17	5343	5287	31/MAR/20
18	5338	5243	24/MAR/20
19	5340	5243	24/MAR/20

Visit Temp fact Table

```
CREATE TABLE visit_tempfact_lvl0
    AS
        SELECT
            client_person_id,
            property_id,
            visit_date
        FROM
            visit;

ALTER TABLE visit_tempfact_lvl0 ADD season VARCHAR(10);

UPDATE visit_tempfact_lvl0
SET
    season = 'Summer'
WHERE
    to_char(visit_date, 'mon') IN (
        'dec',
        'jan',
        'feb'
    );

UPDATE visit_tempfact_lvl0
```

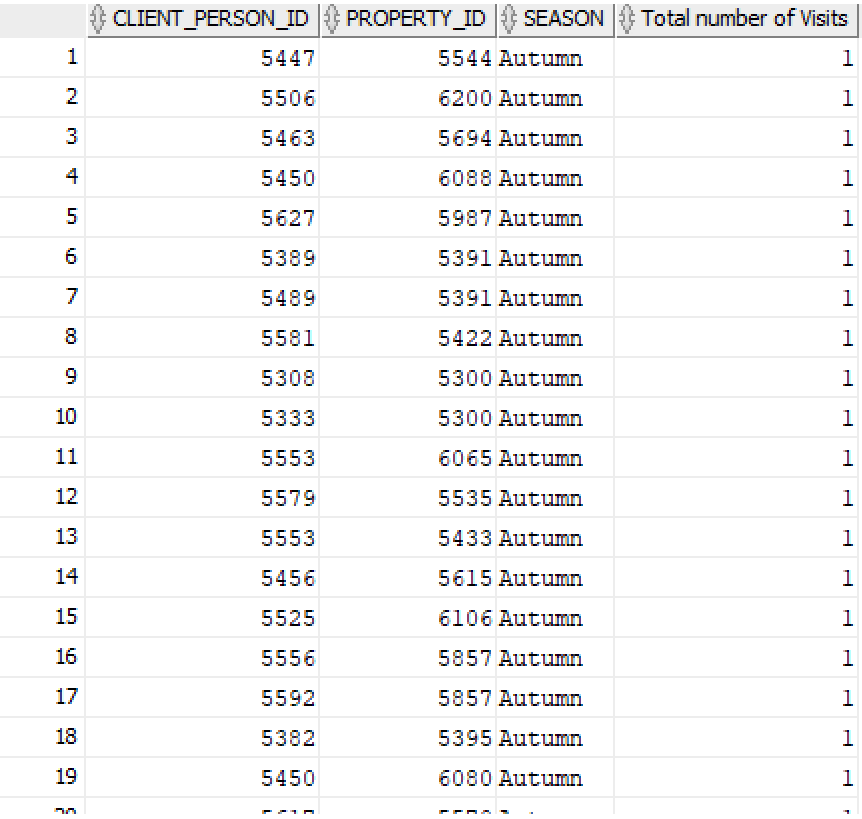
```
SET
    season = 'Autumn'
WHERE
    to_char(visit_date, 'mon') IN (
        'mar',
        'apr',
        'may'
    );

UPDATE visit_tempfact_lvl0
SET
    season = 'Winter'
WHERE
    to_char(visit_date, 'mon') IN (
        'jun',
        'jul',
        'aug'
    );

UPDATE visit_tempfact_lvl0
SET
    season = 'Spring'
WHERE
    to_char(visit_date, 'mon') IN (
        'sep',
        'oct',
        'nov'
    );
```



	CLIENT_PERSON_ID	PROPERTY_ID	VISIT_DATE	SEASON
1	5500	5741	13/APR/20	Autumn
2	5568	5741	13/APR/20	Autumn
3	5403	6102	13/APR/20	Autumn
4	5520	6102	13/APR/20	Autumn
5	5508	5585	26/MAR/20	Autumn
6	5525	6206	14/APR/20	Autumn
7	5529	5776	23/MAR/20	Autumn
8	5399	5411	29/MAR/20	Autumn
9	5462	5411	29/MAR/20	Autumn
10	5474	5411	29/MAR/20	Autumn
11	5498	5411	29/MAR/20	Autumn
12	5542	5411	29/MAR/20	Autumn
13	5324	5287	31/MAR/20	Autumn
14	5329	5287	31/MAR/20	Autumn
15	5330	5287	31/MAR/20	Autumn
16	5337	5287	31/MAR/20	Autumn
17	5343	5287	31/MAR/20	Autumn

Visit Fact Table

```
CREATE TABLE visit_fact_lvl0
   AS
      SELECT
         client_person_id,
         property_id,
         season,
         COUNT(visit_date) "Total number of Visits"
      FROM
         visit_tempfact_lvl0
      GROUP BY
         client_person_id,
         property_id,
         season;
```

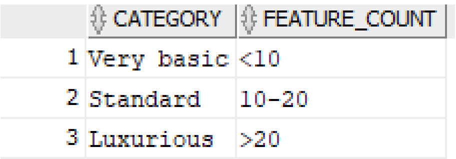


	CLIENT_PERSON_ID	PROPERTY_ID	SEASON	Total number of Visits
1	5447	5544	Autumn	1
2	5506	6200	Autumn	1
3	5463	5694	Autumn	1
4	5450	6088	Autumn	1
5	5627	5987	Autumn	1
6	5389	5391	Autumn	1
7	5489	5391	Autumn	1
8	5581	5422	Autumn	1
9	5308	5300	Autumn	1
10	5333	5300	Autumn	1
11	5553	6065	Autumn	1
12	5579	5535	Autumn	1
13	5553	5433	Autumn	1
14	5456	5615	Autumn	1
15	5525	6106	Autumn	1
16	5556	5857	Autumn	1
17	5592	5857	Autumn	1
18	5382	5395	Autumn	1
19	5450	6080	Autumn	1

Rent Dimensions and Fact creation

Property Feature Category Dimension

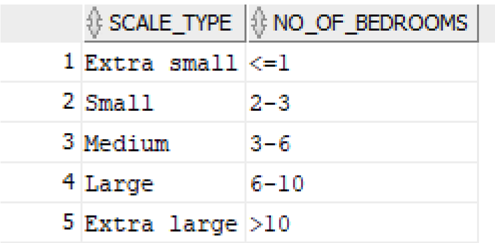
```
CREATE TABLE property_feature_category_dim_lvl0 (
   category  VARCHAR(20),
   feature_count  VARCHAR(20)
);

INSERT INTO property_feature_category_dim_lvl0 VALUES (
   'Very basic',
    '<10'
);

INSERT INTO property_feature_category_dim_lvl0 VALUES (
    'Standard',
    '10-20'
);

INSERT INTO property_feature_category_dim_lvl0 VALUES (
    'Luxurious',
    '>20'
);
```



	CATEGORY	FEATURE_COUNT
1	Very basic	<10
2	Standard	10-20
3	Luxurious	>20

Property Scale Dimension

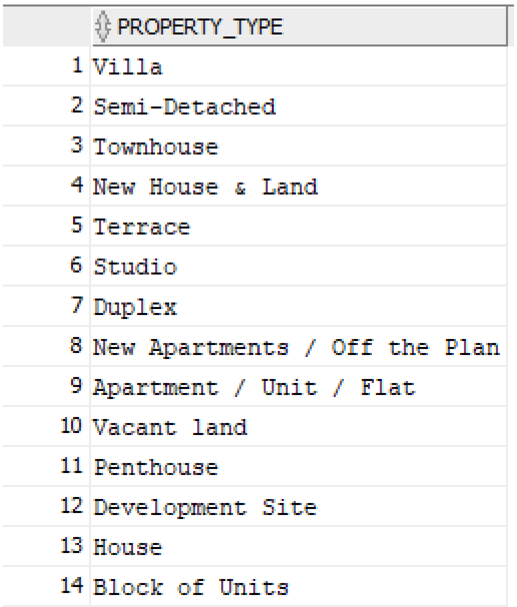
```
CREATE TABLE property_scale_dim_lvl0 (
    scale_type       VARCHAR(20),
    no_of_bedrooms  VARCHAR(20)
);

INSERT INTO property_scale_dim_lvl0 VALUES (
    'Extra small',
    '<=1'
);

INSERT INTO property_scale_dim_lvl0 VALUES (
    'Small',
    '2-3'
);

INSERT INTO property_scale_dim_lvl0 VALUES (
    'Medium',
    '3-6'
);

INSERT INTO property_scale_dim_lvl0 VALUES (
    'Large',
    '6-10'
);

INSERT INTO property_scale_dim_lvl0 VALUES (
    'Extra large',
    '>10'
);
```



	SCALE_TYPE	NO_OF_BEDROOMS
1	Extra small	<=1
2	Small	2-3
3	Medium	3-6
4	Large	6-10
5	Extra large	>10

Property Type Dimension

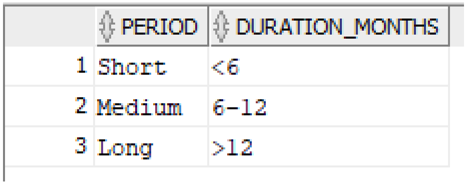
```
CREATE TABLE property_type_dim_lvl0
    AS
        SELECT DISTINCT
            property_type
        FROM
            monre.property;
```



	PROPERTY_TYPE
1	Villa
2	Semi-Detached
3	Townhouse
4	New House & Land
5	Terrace
6	Studio
7	Duplex
8	New Apartments / Off the Plan
9	Apartment / Unit / Flat
10	Vacant land
11	Penthouse
12	Development Site
13	House
14	Block of Units

Rent Time Period Dimension

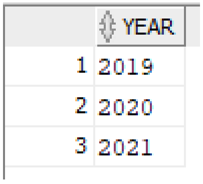
```
CREATE TABLE rent_time_period_dim_lvl0 (
    period           VARCHAR(20),
    duration_months  VARCHAR(20)
);

INSERT INTO rent_time_period_dim_lvl0 VALUES (
    'Short',
    '<6'
);

INSERT INTO rent_time_period_dim_lvl0 VALUES (
    'Medium',
    '6-12'
);

INSERT INTO rent_time_period_dim_lvl0 VALUES (
    'Long',
    '>12'
);
```



	PERIOD	DURATION_MONTHS
1	Short	<6
2	Medium	6-12
3	Long	>12

Rent Year Dimension

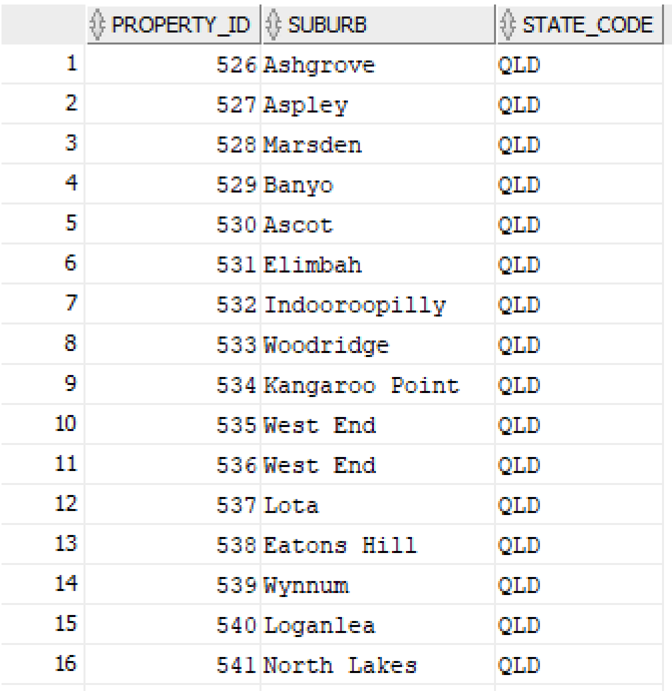
```
CREATE TABLE rent_year_dim_lvl0
    AS
        SELECT DISTINCT
            to_char(rent_start_date, 'yyyy') AS year
        FROM
            rent
        WHERE
            NOT ( to_char(rent_start_date, 'yyyy') ) IS NULL;
```



	YEAR
1	2019
2	2020
3	2021

Property Dimension

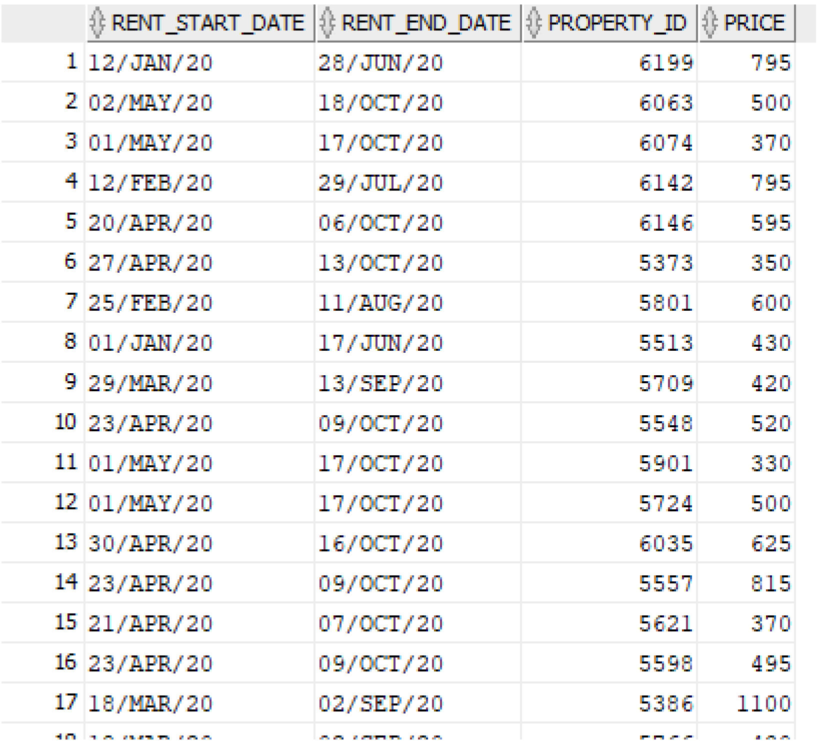
```
CREATE TABLE property_dim_lvl0
    AS
        SELECT
            p.property_id,
            ad.suburb,
            pc.state_code
        FROM
            property  p
            JOIN address    ad ON p.address_id = ad.address_id
            JOIN postcode pc ON ad.postcode = pc.postcode;
```



	PROPERTY_ID	SUBURB	STATE_CODE
1	526	Ashgrove	QLD
2	527	Aspley	QLD
3	528	Marsden	QLD
4	529	Banyo	QLD
5	530	Ascot	QLD
6	531	Elimbah	QLD
7	532	Indooroopilly	QLD
8	533	Woodridge	QLD
9	534	Kangaroo Point	QLD
10	535	West End	QLD
11	536	West End	QLD
12	537	Lota	QLD
13	538	Eatons Hill	QLD
14	539	Wynnum	QLD
15	540	Loganlea	QLD
16	541	North Lakes	QLD

Property Rent SCD Dimension

```
CREATE TABLE property_rent_scd_lvl0
   AS
      SELECT
         rent_start_date,
         rent_end_date,
         property_id,
         price
      FROM
         rent
      WHERE
         NOT rent_start_date IS NULL;
```

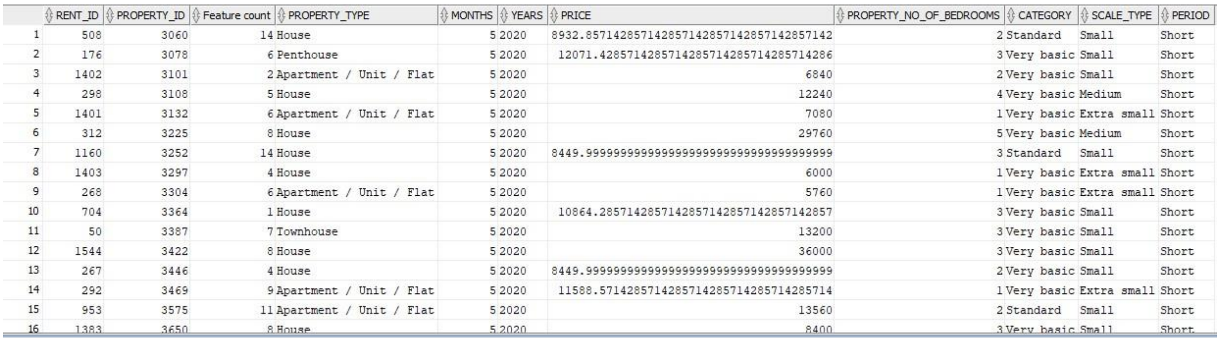


	RENT_START_DATE	RENT_END_DATE	PROPERTY_ID	PRICE
1	12/JAN/20	28/JUN/20	6199	795
2	02/MAY/20	18/OCT/20	6063	500
3	01/MAY/20	17/OCT/20	6074	370
4	12/FEB/20	29/JUL/20	6142	795
5	20/APR/20	06/OCT/20	6146	595
6	27/APR/20	13/OCT/20	5373	350
7	25/FEB/20	11/AUG/20	5801	600
8	01/JAN/20	17/JUN/20	5513	430
9	29/MAR/20	13/SEP/20	5709	420
10	23/APR/20	09/OCT/20	5548	520
11	01/MAY/20	17/OCT/20	5901	330
12	01/MAY/20	17/OCT/20	5724	500
13	30/APR/20	16/OCT/20	6035	625
14	23/APR/20	09/OCT/20	5557	815
15	21/APR/20	07/OCT/20	5621	370
16	23/APR/20	09/OCT/20	5598	495
17	18/MAR/20	02/SEP/20	5386	1100

Rent Temp Fact Table

```
CREATE TABLE rent_tempfact_lvl0
   AS
      SELECT
         rent_id,
         r.property_id,
         COUNT(feature_code) AS "Feature count",
         property_type,
         floor(months_between(to_date(rent_end_date, 'dd-mm-yyyy'),
to_date(rent_start_date, 'dd-mm-yyyy'))) AS months,
         to_char(rent_start_date, 'yyyy') AS years,
         price * ( ( to_date(rent_end_date, 'dd-mm-yyyy') - to_date(rent_start_date, 'dd-mm-
yyyy') ) / 7 ) AS price,
         p.property_no_of_bedrooms
      FROM
         rent

R

JOIN property p ON r.property_id = p.property_id
   JOIN property_feature pf ON p.property_id=pf.property_id
   JOIN address ad ON ad.address_id=p.address_id
   JOIN postcode pc ON ad.postcode=pc.postcode
   WHERE NOT r.rent_start_date IS NULL
```

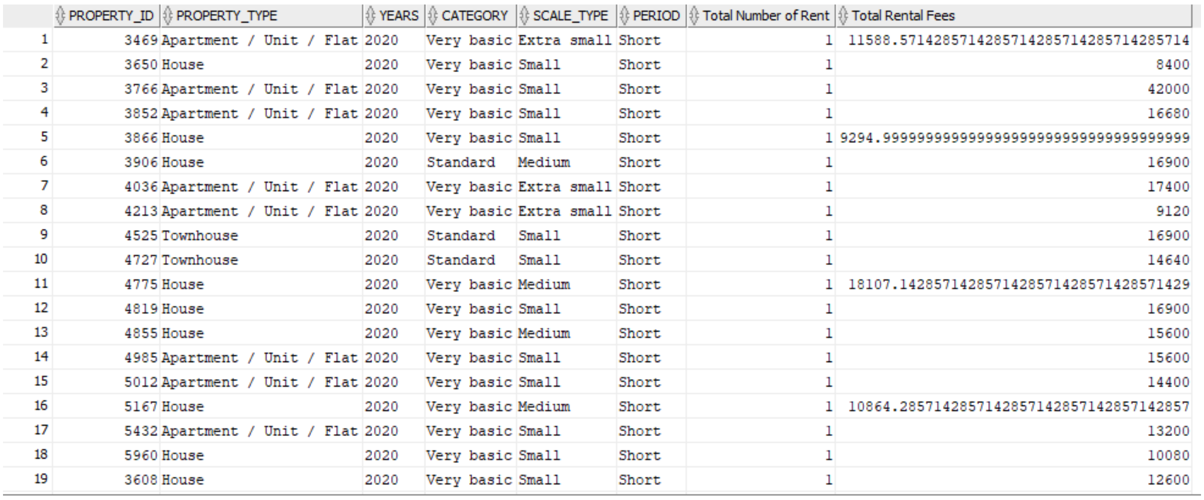
```
    GROUP BY (rent_id, r.property_id, property_type,
months_between(TO_DATE(rent_end_date,'dd-mm-yyyy'),TO_DATE(rent_start_date,'dd-mm-yyyy')),
        to_char(rent_start_date,'yyyy'), price*((TO_DATE(rent_end_date,'dd-mm-yyyy')-TO_DATE(rent_start_date,'dd-mm-yyyy'))/7), p.property_no_of_bedrooms);

ALTER TABLE rent_tempfact_lvl0 ADD category VARCHAR(20);

UPDATE rent_tempfact_lvl0 SET category='Very basic' WHERE "Feature count"<10;
UPDATE rent_tempfact_lvl0 SET category='Standard' WHERE "Feature count" between 10 and 20;
UPDATE rent_tempfact_lvl0 SET category='Luxurious' WHERE "Feature count">20;

ALTER TABLE rent_tempfact_lvl0 ADD scale_type VARCHAR(20);

UPDATE rent_tempfact_lvl0 SET scale_type='Extra small' WHERE property_no_of_bedrooms<=1;
UPDATE rent_tempfact_lvl0 SET scale_type='Small' WHERE property_no_of_bedrooms BETWEEN 2 AND 3;
UPDATE rent_tempfact_lvl0 SET scale_type='Medium' WHERE property_no_of_bedrooms BETWEEN 4 AND 6;
UPDATE rent_tempfact_lvl0 SET scale_type='Large' WHERE property_no_of_bedrooms BETWEEN 7 AND 10;
UPDATE rent_tempfact_lvl0 SET scale_type='Extra large' WHERE property_no_of_bedrooms>10;

ALTER TABLE rent_tempfact_lvl0 ADD period VARCHAR(20);

UPDATE rent_tempfact_lvl0 SET period='Short' WHERE Months<6;
UPDATE rent_tempfact_lvl0 SET period='Medium' WHERE Months BETWEEN 6 AND 12;
UPDATE rent_tempfact_lvl0 SET period='Long' WHERE Months>12;
```



	RENT_ID	PROPERTY_ID	Feature count	PROPERTY_TYPE	MONTHS	YEARS	PRICE	PROPERTY_NO_OF_BEDROOMS	CATEGORY	SCALE_TYPE	PERIOD
1	508	3060	14	House	5	2020	8932.857142857142857142857142857142857142	2	Standard	Small	Short
2	176	3078	6	Penthouse	5	2020	12071.428571428571428571428571428571428286	3	Very basic	Small	Short
3	1402	3101	2	Apartment / Unit / Flat	5	2020	6840	2	Very basic	Small	Short
4	298	3108	5	House	5	2020	12240	4	Very basic	Medium	Short
5	1401	3132	6	Apartment / Unit / Flat	5	2020	7080	1	Very basic	Extra small	Short
6	312	3225	8	House	5	2020	29760	5	Very basic	Medium	Short
7	1160	3252	14	House	5	2020	8449.999999999999999999999999999999999999	3	Standard	Small	Short
8	1403	3297	4	House	5	2020	6000	1	Very basic	Extra small	Short
9	268	3304	6	Apartment / Unit / Flat	5	2020	5760	1	Very basic	Extra small	Short
10	704	3364	1	House	5	2020	10864.285714285714285714285714285714142857	3	Very basic	Small	Short
11	50	3387	7	Townhouse	5	2020	13200	3	Very basic	Small	Short
12	1544	3422	8	House	5	2020	36000	3	Very basic	Small	Short
13	267	3446	4	House	5	2020	8449.999999999999999999999999999999999999	2	Very basic	Small	Short
14	292	3469	9	Apartment / Unit / Flat	5	2020	11588.571428571428571428571428571428285714	1	Very basic	Extra small	Short
15	953	3575	11	Apartment / Unit / Flat	5	2020	13560	2	Standard	Small	Short
16	1383	3650	8	House	5	2020	8400	3	Very basic	Small	Short

Rent Fact Table

```
CREATE TABLE rent_fact_lvl0
    AS
        SELECT
            property_id,
            property_type,
            years,
            category,
            scale_type,
            period,
            COUNT(rent_id) AS "Total Number of Rent",
            SUM(price) AS "Total Rental Fees"
        FROM
            rent_tempfact_lvl0
        GROUP BY (
            property_id,
            property_type,
            years,
            category,
            scale_type,
            period
        );
```



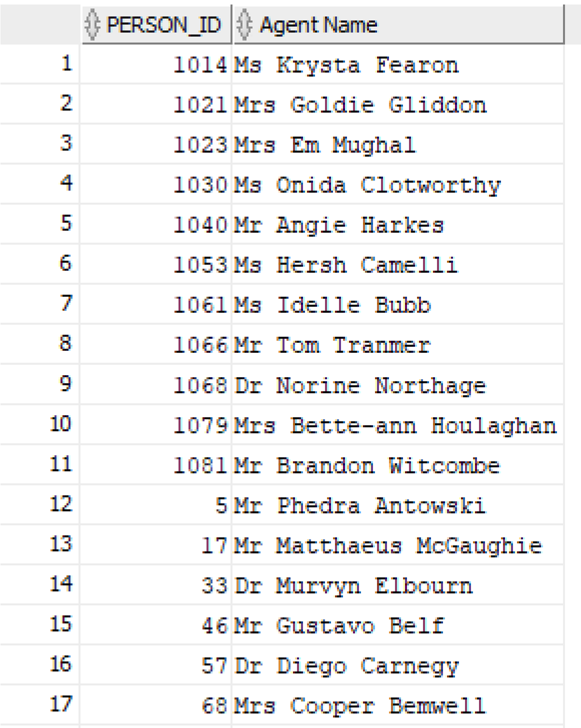
	PROPERTY_ID	PROPERTY_TYPE	YEARS	CATEGORY	SCALE_TYPE	PERIOD	Total Number of Rent	Total Rental Fees
1	3469	Apartment / Unit / Flat	2020	Very basic	Extra small	Short	1	11588.5714285714285714285714285714285714
2	3650	House	2020	Very basic	Small	Short	1	8400
3	3766	Apartment / Unit / Flat	2020	Very basic	Small	Short	1	42000
4	3852	Apartment / Unit / Flat	2020	Very basic	Small	Short	1	16680
5	3866	House	2020	Very basic	Small	Short	1	9294.999999999999999999999999999999999999
6	3906	House	2020	Standard	Medium	Short	1	16900
7	4036	Apartment / Unit / Flat	2020	Very basic	Extra small	Short	1	17400
8	4213	Apartment / Unit / Flat	2020	Very basic	Extra small	Short	1	9120
9	4525	Townhouse	2020	Standard	Small	Short	1	16900
10	4727	Townhouse	2020	Standard	Small	Short	1	14640
11	4775	House	2020	Very basic	Medium	Short	1	18107.1428571428571428571428571428571429
12	4819	House	2020	Very basic	Small	Short	1	16900
13	4855	House	2020	Very basic	Medium	Short	1	15600
14	4985	Apartment / Unit / Flat	2020	Very basic	Small	Short	1	15600
15	5012	Apartment / Unit / Flat	2020	Very basic	Small	Short	1	14400
16	5167	House	2020	Very basic	Medium	Short	1	10864.2857142857142857142857142857142857
17	5432	Apartment / Unit / Flat	2020	Very basic	Small	Short	1	13200
18	5960	House	2020	Very basic	Small	Short	1	10080
19	3608	House	2020	Very basic	Small	Short	1	12600

Agent Dimensions and Fact creation

Agent Information Dimension

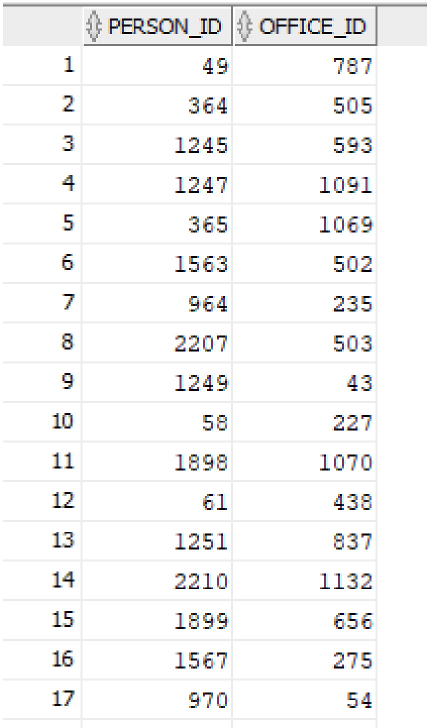
```
CREATE TABLE agent_info_dim_lvl0
  AS
    SELECT DISTINCT
        ( a.person_id ),
        p.title
        || ' '
        || p.first_name
        || ' '
        || p.last_name AS "Agent Name"
    FROM
        agent

A
join
```

person p ON a.person_id = p.person_id;



	PERSON_ID	Agent Name
1	1014	Ms Krysta Fearon
2	1021	Mrs Goldie Gliddon
3	1023	Mrs Em Mughal
4	1030	Ms Onida Clotworthy
5	1040	Mr Angie Harkes
6	1053	Ms Hersh Camelli
7	1061	Ms Idelle Bubb
8	1066	Mr Tom Tranmer
9	1068	Dr Norine Northage
10	1079	Mrs Bette-ann Houlaghan
11	1081	Mr Brandon Witcombe
12	5	Mr Phedra Antowski
13	17	Mr Matthaeus McGaughie
14	33	Dr Murvyn Elbourn
15	46	Mr Gustavo Belf
16	57	Dr Diego Carnegy
17	68	Mrs Cooper Bemwell

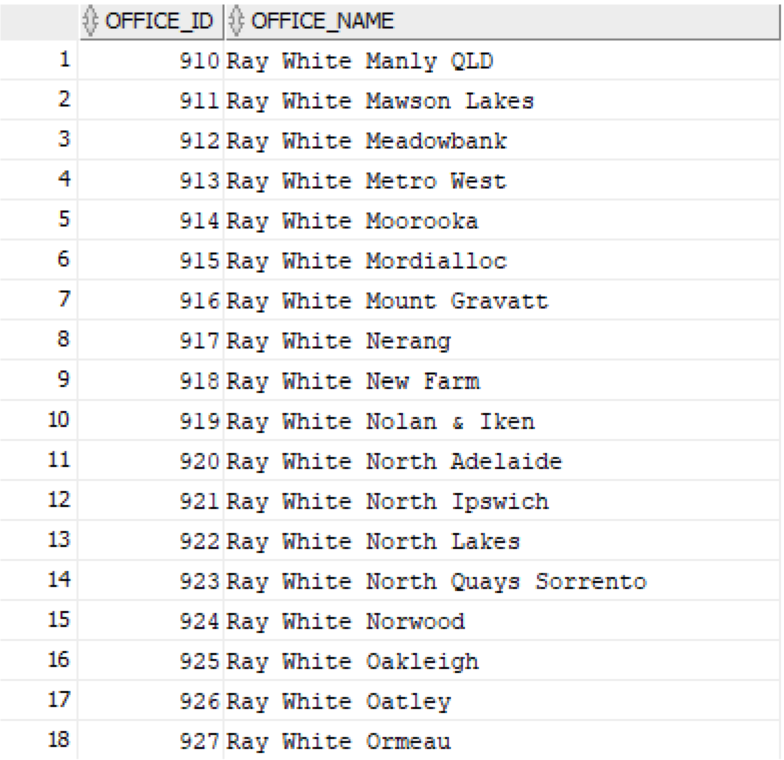
Agent Office Bridge Dimension
CREATE TABLE agent_office_bridge_dim_lvl0 AS SELECT person_Id, office_id FROM agent_office;



	PERSON_ID	OFFICE_ID
1	49	787
2	364	505
3	1245	593
4	1247	1091
5	365	1069
6	1563	502
7	964	235
8	2207	503
9	1249	43
10	58	227
11	1898	1070
12	61	438
13	1251	837
14	2210	1132
15	1899	656
16	1567	275
17	970	54

Office Dimension

CREATE TABLE office_dim_lvl0 AS SELECT office_id,office_name FROM Office;

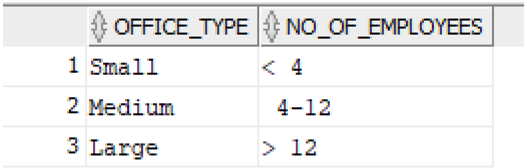


	OFFICE_ID	OFFICE_NAME
1	910	Ray White Manly QLD
2	911	Ray White Mawson Lakes
3	912	Ray White Meadowbank
4	913	Ray White Metro West
5	914	Ray White Moorooka
6	915	Ray White Mordialloc
7	916	Ray White Mount Gravatt
8	917	Ray White Nerang
9	918	Ray White New Farm
10	919	Ray White Nolan & Iken
11	920	Ray White North Adelaide
12	921	Ray White North Ipswich
13	922	Ray White North Lakes
14	923	Ray White North Quays Sorrento
15	924	Ray White Norwood
16	925	Ray White Oakleigh
17	926	Ray White Oatley
18	927	Ray White Ormeau

Office Size Dimension

CREATE TABLE office_size_dim_lvl0 (office_type VARCHAR2(30), no_of_employees VARCHAR2(40));

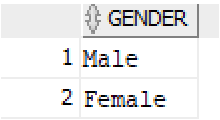
INSERT INTO office_size_dim_lvl0 VALUES('Small','< 4');
INSERT INTO office_size_dim_lvl0 VALUES('Medium',' 4-12');
INSERT INTO office_size_dim_lvl0 VALUES('Large','> 12');



	OFFICE_TYPE	NO_OF_EMPLOYEES
1	Small	< 4
2	Medium	4-12
3	Large	> 12

Gender Dimension

CREATE TABLE gender_dim_lvl0 AS SELECT DISTINCT gender FROM person;

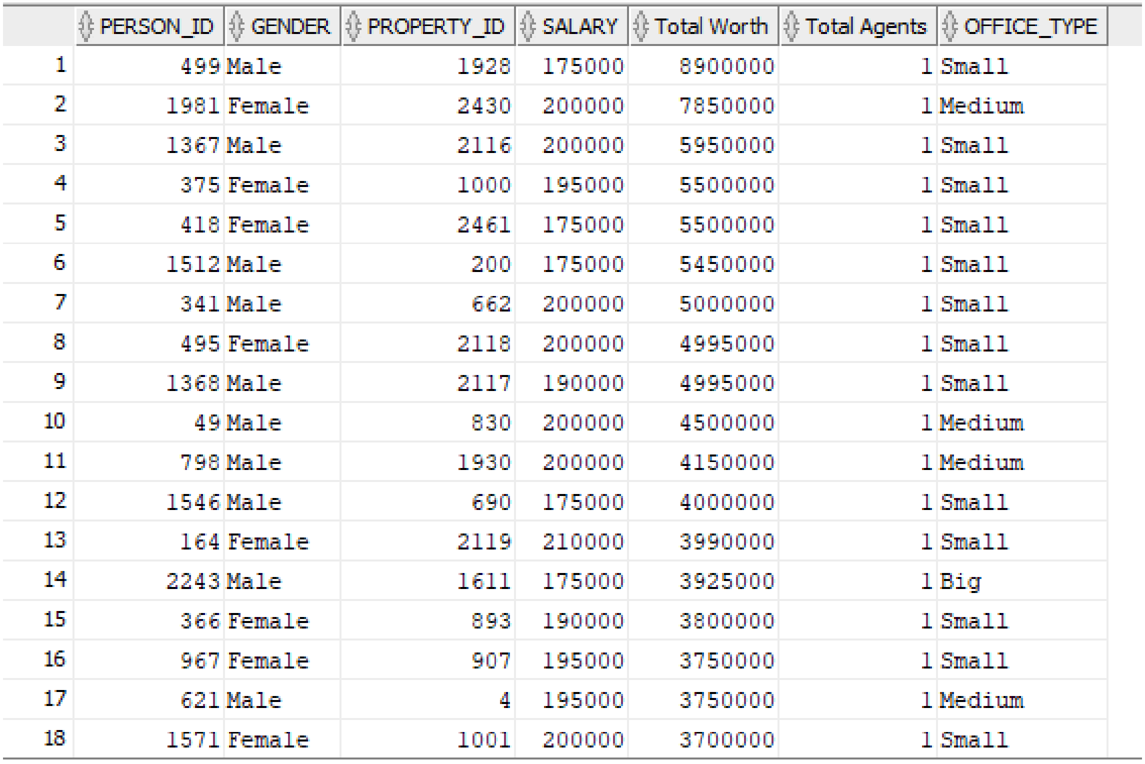


	GENDER
1	Male
2	Female

Agent Temp fact Table

```
CREATE TABLE agent_tempfact_lvl0
  AS
    SELECT
      person_id,
      gender,
      property_id,
      salary,
      SUM(price) "Total Worth",
      COUNT(person_id) "Total Agents"
    FROM
      (
        SELECT
          a.person_id,
          pe.gender,
          a.salary,
          p.property_id,
          s.price
        FROM
          agent        a
          LEFT JOIN sale          s ON a.person_id = s.agent_person_id
          LEFT JOIN property      p ON s.property_id = p.property_id
          LEFT JOIN address       ad ON p.address_id = ad.address_id
          LEFT JOIN agent_office ao ON a.person_id = ao.person_id
          LEFT JOIN person        pe ON a.person_id = pe.person_id
        UNION
        SELECT
          a.person_id,
          pe.gender,
          a.salary,
          p.property_id,
          r.price * ( r.rent_end_date - r.rent_start_date ) / 7
        FROM
          agent        a
          LEFT JOIN rent          r ON a.person_id = r.agent_person_id
          LEFT JOIN property      p ON r.property_id = p.property_id
          LEFT JOIN address       ad ON p.address_id = ad.address_id
          LEFT JOIN agent_office ao ON a.person_id = ao.person_id
          LEFT JOIN person        pe ON a.person_id = pe.person_id
      )
    WHERE
      price IS NOT NULL
    GROUP BY
      person_id,
      gender,
      property_id,
      salary
    ORDER BY
```

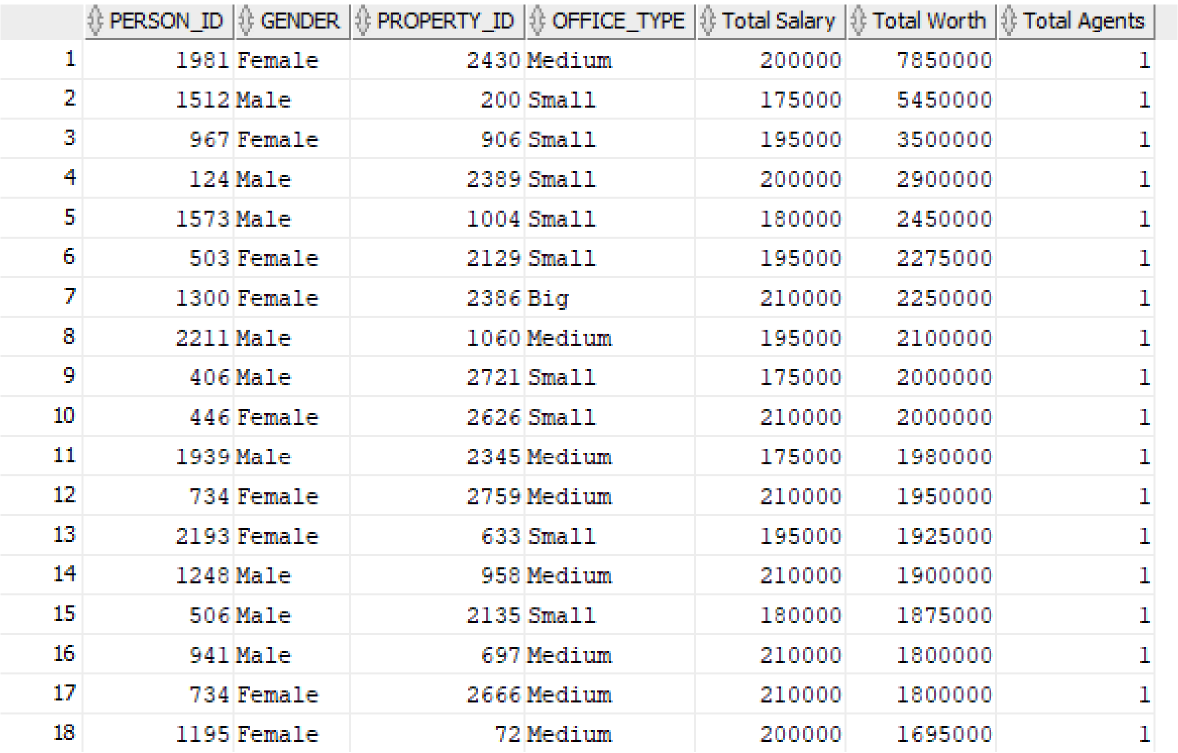
```
            SUM(price) DESC;

ALTER TABLE agent_tempfact_lvl0 ADD office_type VARCHAR(10);

UPDATE agent_tempfact_lvl0
SET
    office_type = 'Small'
WHERE
    person_id IN (
        SELECT
            a.person_id
        FROM
            agent_tempfact_lvl0 a
            JOIN agent_office          b ON a.person_id = b.person_id
        WHERE
            b.office_id IN (
                SELECT
                    office_id
                FROM
                    agent_office
                GROUP BY
                    office_id
                HAVING
                    COUNT(person_id) < 4
            )
    );

UPDATE agent_tempfact_lvl0
SET
    office_type = 'Medium'
WHERE
    person_id IN (
        SELECT
            a.person_id
        FROM
            agent_tempfact_lvl0 a
            JOIN agent_office          b ON a.person_id = b.person_id
        WHERE
            b.office_id IN (
                SELECT
                    office_id
                FROM
                    agent_office
                GROUP BY
                    office_id
                HAVING
                    COUNT(person_id) BETWEEN 4 AND 12
            )
    );

UPDATE agent_tempfact_lvl0
SET
    office_type = 'Big'
WHERE
    person_id IN (
        SELECT
            a.person_id
        FROM
            agent_tempfact_lvl0 a
            JOIN agent_office          b ON a.person_id = b.person_id
        WHERE
            b.office_id IN (
                SELECT
                    office_id
                FROM
                    agent_office
                GROUP BY
                    office_id
                HAVING
                    COUNT(person_id) > 12
            )
    );
```



	PERSON_ID	GENDER	PROPERTY_ID	SALARY	Total Worth	Total Agents	OFFICE_TYPE
1	499	Male	1928	175000	8900000	1	Small
2	1981	Female	2430	200000	7850000	1	Medium
3	1367	Male	2116	200000	5950000	1	Small
4	375	Female	1000	195000	5500000	1	Small
5	418	Female	2461	175000	5500000	1	Small
6	1512	Male	200	175000	5450000	1	Small
7	341	Male	662	200000	5000000	1	Small
8	495	Female	2118	200000	4995000	1	Small
9	1368	Male	2117	190000	4995000	1	Small
10	49	Male	830	200000	4500000	1	Medium
11	798	Male	1930	200000	4150000	1	Medium
12	1546	Male	690	175000	4000000	1	Small
13	164	Female	2119	210000	3990000	1	Small
14	2243	Male	1611	175000	3925000	1	Big
15	366	Female	893	190000	3800000	1	Small
16	967	Female	907	195000	3750000	1	Small
17	621	Male	4	195000	3750000	1	Medium
18	1571	Female	1001	200000	3700000	1	Small

Agent Fact Table

```
CREATE TABLE agent_fact_lvl0
  AS
    SELECT
      person_id,
      gender,
      property_id,
      office_type,
      SUM(salary) "Total Salary",
      SUM("Total Worth") "Total Worth",
      SUM("Total Agents") "Total Agents"
    FROM
      agent_tempfact_lvl0
    GROUP BY (
      person_id,
      gender,
      property_id,
      office_type
    );
```

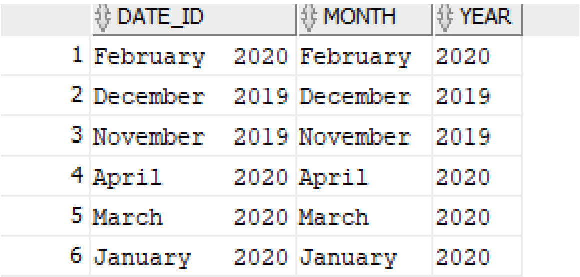


	PERSON_ID	GENDER	PROPERTY_ID	OFFICE_TYPE	Total Salary	Total Worth	Total Agents
1	1981	Female	2430	Medium	200000	7850000	1
2	1512	Male	200	Small	175000	5450000	1
3	967	Female	906	Small	195000	3500000	1
4	124	Male	2389	Small	200000	2900000	1
5	1573	Male	1004	Small	180000	2450000	1
6	503	Female	2129	Small	195000	2275000	1
7	1300	Female	2386	Big	210000	2250000	1
8	2211	Male	1060	Medium	195000	2100000	1
9	406	Male	2721	Small	175000	2000000	1
10	446	Female	2626	Small	210000	2000000	1
11	1939	Male	2345	Medium	175000	1980000	1
12	734	Female	2759	Medium	210000	1950000	1
13	2193	Female	633	Small	195000	1925000	1
14	1248	Male	958	Medium	210000	1900000	1
15	506	Male	2135	Small	180000	1875000	1
16	941	Male	697	Medium	210000	1800000	1
17	734	Female	2666	Medium	210000	1800000	1
18	1195	Female	72	Medium	200000	1695000	1

Advertisement Dimensions and Fact creation

Advertisement Date Dimension

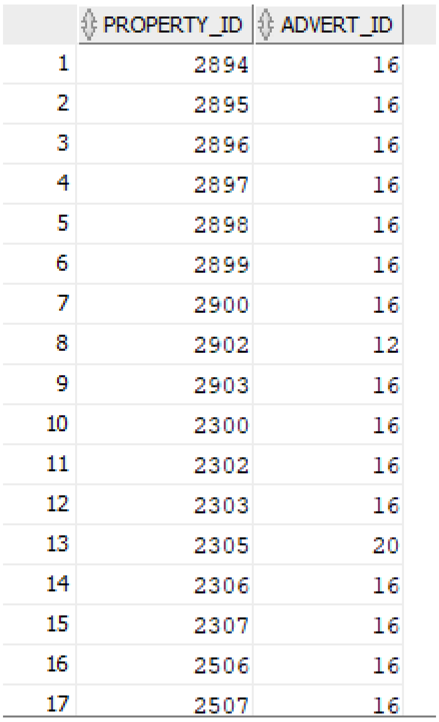
```
CREATE TABLE advert_date_dim_lvl0
   AS
      SELECT DISTINCT
         to_char(property_date_added, 'Month')
         || ' '
         || to_char(property_date_added, 'yyyy') AS date_id,
         to_char(property_date_added, 'Month') AS month,
         to_char(property_date_added, 'yyyy') AS year
      FROM
         property;
```



	DATE_ID	MONTH	YEAR
1	February 2020	February	2020
2	December 2019	December	2019
3	November 2019	November	2019
4	April 2020	April	2020
5	March 2020	March	2020
6	January 2020	January	2020

Property Advertisement Bridge Dimension

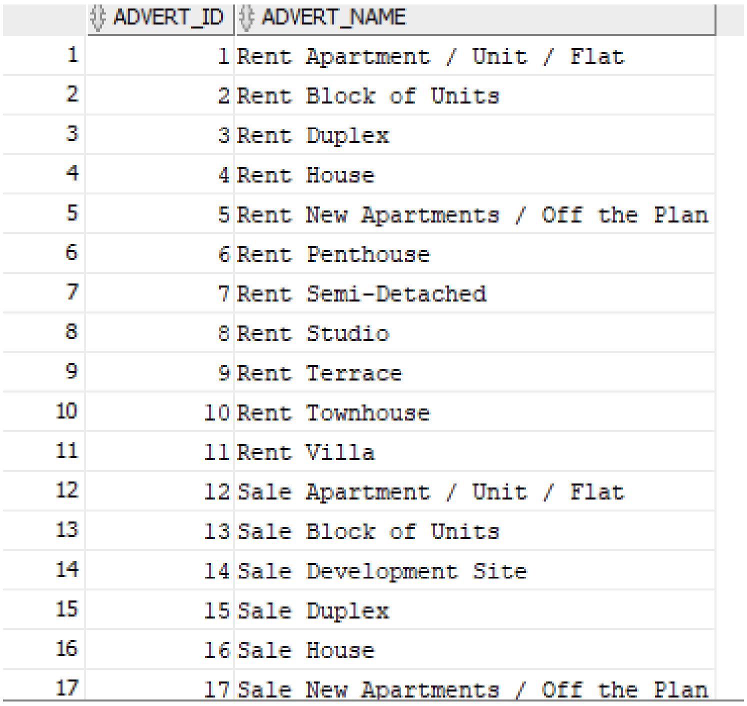
```
CREATE TABLE property_advert_bridge_dim_lvl0
   AS
      SELECT
         property_id,
         advert_id
      FROM
         property_advert;
```



	PROPERTY_ID	ADVERT_ID
1	2894	16
2	2895	16
3	2896	16
4	2897	16
5	2898	16
6	2899	16
7	2900	16
8	2902	12
9	2903	16
10	2300	16
11	2302	16
12	2303	16
13	2305	20
14	2306	16
15	2307	16
16	2506	16
17	2507	16

Advertisement Dimension

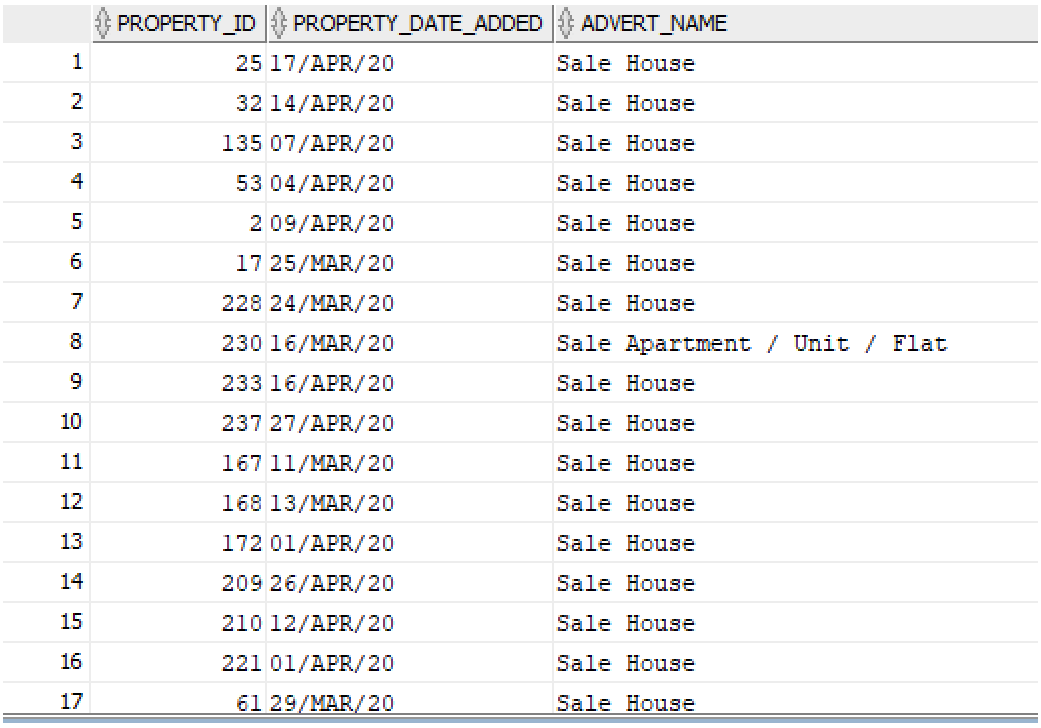
```
CREATE TABLE advert_dim_lvl0
   AS
      SELECT
         advert_id,
         advert_name
      FROM
         advertisement;
```



	ADVERT_ID	ADVERT_NAME
1	1	Rent Apartment / Unit / Flat
2	2	Rent Block of Units
3	3	Rent Duplex
4	4	Rent House
5	5	Rent New Apartments / Off the Plan
6	6	Rent Penthouse
7	7	Rent Semi-Detached
8	8	Rent Studio
9	9	Rent Terrace
10	10	Rent Townhouse
11	11	Rent Villa
12	12	Sale Apartment / Unit / Flat
13	13	Sale Block of Units
14	14	Sale Development Site
15	15	Sale Duplex
16	16	Sale House
17	17	Sale New Apartments / Off the Plan

Advertisement Temp Fact Table

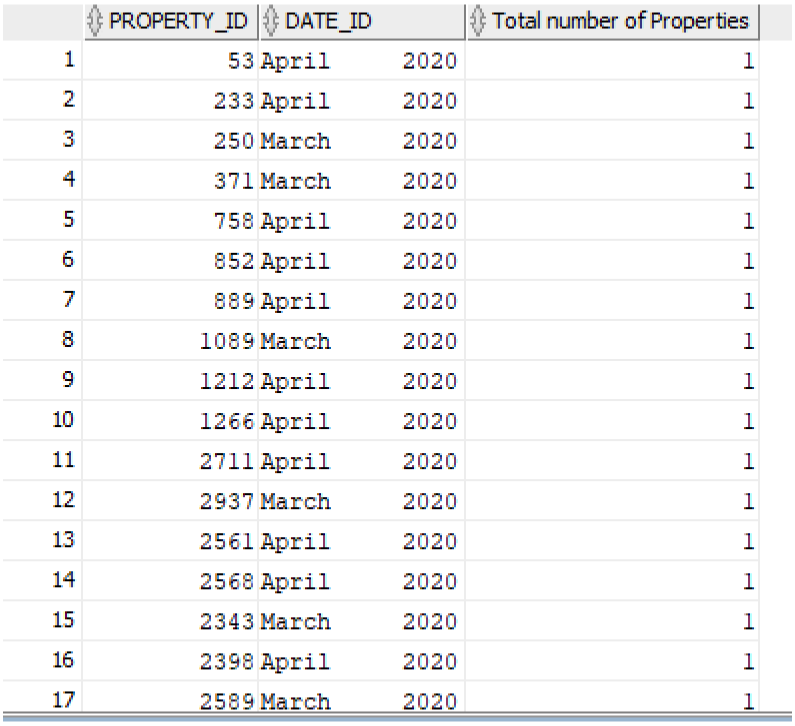
```
CREATE TABLE advertisement_tempfact_lvl0
    AS
        SELECT
            p.property_id,
            pd.property_date_added,
            a.advert_name
        FROM
            advertisement

A
join

property_advert p ON a.advert_id = p.advert_id
JOIN property pd ON p.property_id=pd.property_id
GROUP BY p.property_id,pd.property_date_added,a.advert_name;
```



	PROPERTY_ID	PROPERTY_DATE_ADDED	ADVERT_NAME
1	25	17/APR/20	Sale House
2	32	14/APR/20	Sale House
3	135	07/APR/20	Sale House
4	53	04/APR/20	Sale House
5	2	09/APR/20	Sale House
6	17	25/MAR/20	Sale House
7	228	24/MAR/20	Sale House
8	230	16/MAR/20	Sale Apartment / Unit / Flat
9	233	16/APR/20	Sale House
10	237	27/APR/20	Sale House
11	167	11/MAR/20	Sale House
12	168	13/MAR/20	Sale House
13	172	01/APR/20	Sale House
14	209	26/APR/20	Sale House
15	210	12/APR/20	Sale House
16	221	01/APR/20	Sale House
17	61	29/MAR/20	Sale House

Advertisement Fact Table

```
CREATE TABLE advertisement_fact_lvl0
    AS
        SELECT
            property_id,
            to_char(property_date_added, 'Month')
            || ' '
            || to_char(property_date_added, 'yyyy') date_id,
            COUNT(property_id) "Total number of Properties"
        FROM
            advertisement_tempfact_lvl0
        GROUP BY
            property_id,
            to_char(property_date_added, 'Month')
            || ' '
            || to_char(property_date_added, 'yyyy');
```

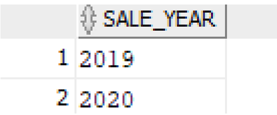


	PROPERTY_ID	DATE_ID	Total number of Properties
1	53	April 2020	1
2	233	April 2020	1
3	250	March 2020	1
4	371	March 2020	1
5	758	April 2020	1
6	852	April 2020	1
7	889	April 2020	1
8	1089	March 2020	1
9	1212	April 2020	1
10	1266	April 2020	1
11	2711	April 2020	1
12	2937	March 2020	1
13	2561	April 2020	1
14	2568	April 2020	1
15	2343	March 2020	1
16	2398	April 2020	1
17	2589	March 2020	1

Sales Dimensions and Fact creation

Sale Year DIMENSION

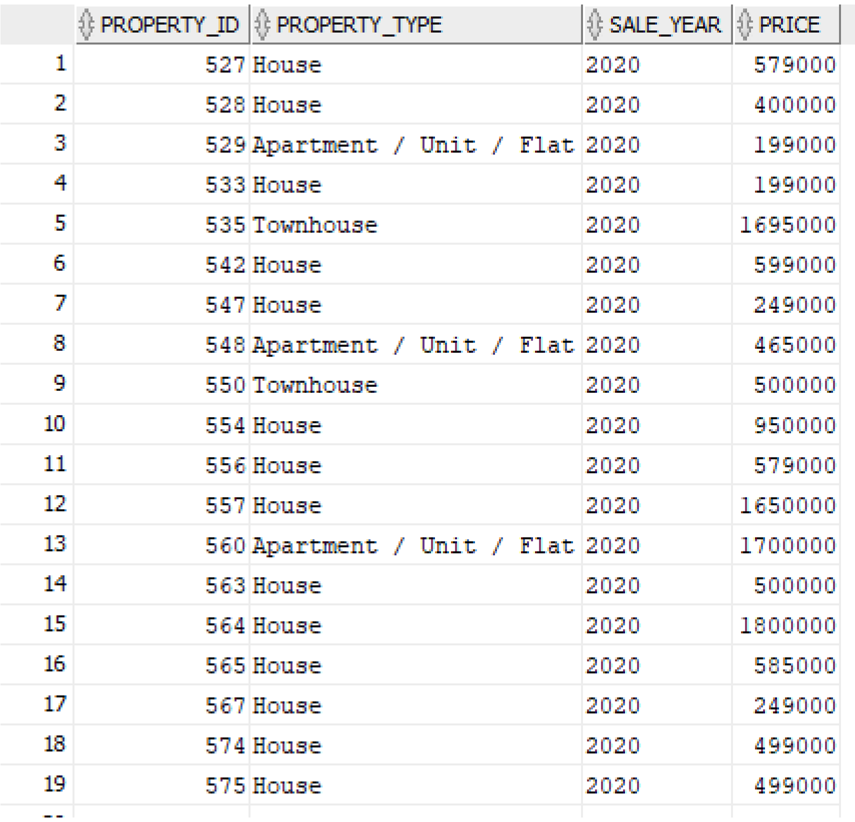
```
CREATE TABLE sale_year_lvl0
  AS
    SELECT DISTINCT
        ( to_char(sale_date, 'yyyy') ) AS sale_year
    FROM
        sale
    WHERE
        NOT ( to_char(sale_date, 'yyyy') ) IS NULL;
```



	SALE_YEAR
1	2019
2	2020

Sale Temp fact Table

```
CREATE TABLE sale_tempfact_lvl0
    AS
        SELECT
            s.property_id,
            p.property_type,
            to_char(s.sale_date, 'yyyy') AS sale_year,
            s.price
        FROM
            sale            s
            JOIN property        p ON s.property_id = p.property_id
            JOIN monre.address ad ON ad.address_id = p.address_id
            JOIN postcode        pc ON pc.postcode = ad.postcode
        WHERE
            NOT s.client_person_id IS NULL;
```

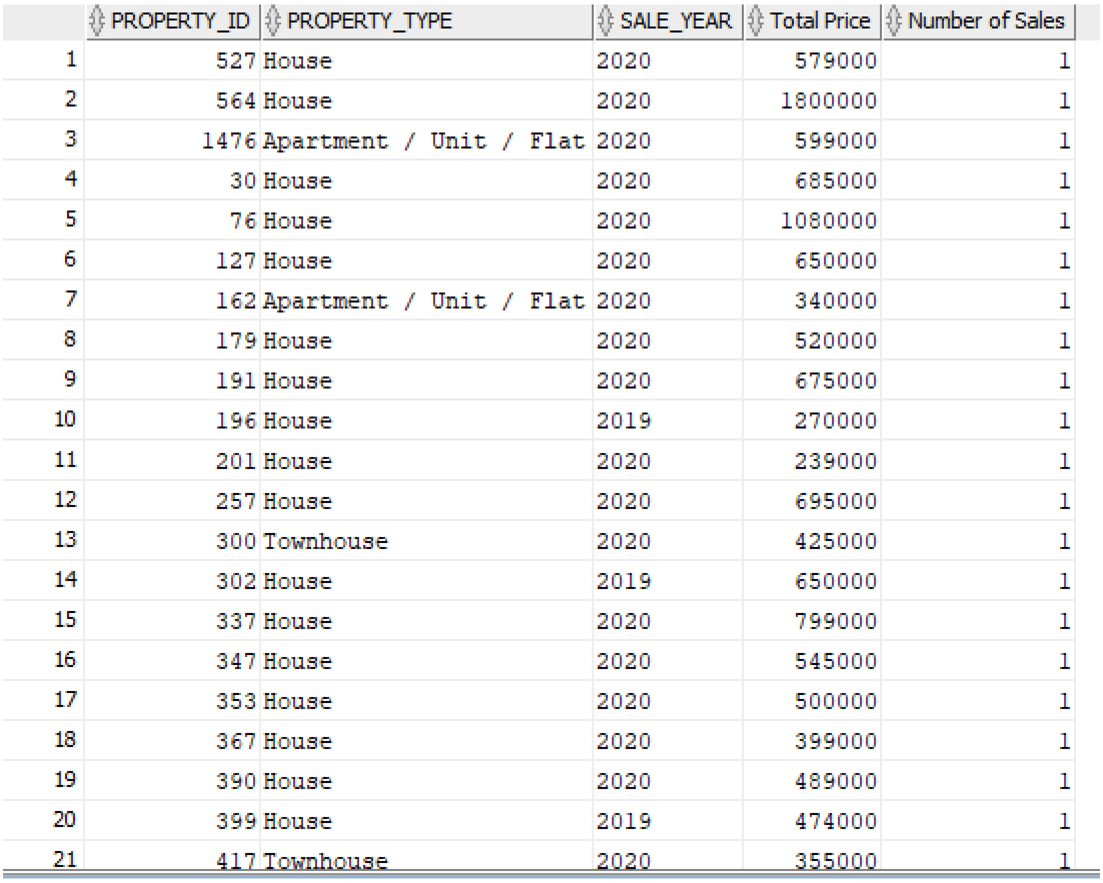


	PROPERTY_ID	PROPERTY_TYPE	SALE_YEAR	PRICE
1	527	House	2020	579000
2	528	House	2020	400000
3	529	Apartment / Unit / Flat	2020	199000
4	533	House	2020	199000
5	535	Townhouse	2020	1695000
6	542	House	2020	599000
7	547	House	2020	249000
8	548	Apartment / Unit / Flat	2020	465000
9	550	Townhouse	2020	500000
10	554	House	2020	950000
11	556	House	2020	579000
12	557	House	2020	1650000
13	560	Apartment / Unit / Flat	2020	1700000
14	563	House	2020	500000
15	564	House	2020	1800000
16	565	House	2020	585000
17	567	House	2020	249000
18	574	House	2020	499000
19	575	House	2020	499000

Sale Fact Table

```
CREATE TABLE sale_fact_lvl0
    AS
        SELECT
            property_id,
            property_type,
            sale_year,
            SUM(price) "Total Price",
            COUNT(property_id) "Number of Sales"
```

```
FROM
   sale_tempfact_lvl0
GROUP BY
   property_id,
   property_type,
   sale_year;
```



	PROPERTY_ID	PROPERTY_TYPE	SALE_YEAR	Total Price	Number of Sales
1	527	House	2020	579000	1
2	564	House	2020	1800000	1
3	1476	Apartment / Unit / Flat	2020	599000	1
4	30	House	2020	685000	1
5	76	House	2020	1080000	1
6	127	House	2020	650000	1
7	162	Apartment / Unit / Flat	2020	340000	1
8	179	House	2020	520000	1
9	191	House	2020	675000	1
10	196	House	2019	270000	1
11	201	House	2020	239000	1
12	257	House	2020	695000	1
13	300	Townhouse	2020	425000	1
14	302	House	2019	650000	1
15	337	House	2020	799000	1
16	347	House	2020	545000	1
17	353	House	2020	500000	1
18	367	House	2020	399000	1
19	390	House	2020	489000	1
20	399	House	2019	474000	1
21	417	Townhouse	2020	355000	1

C3

3. Report creation using OLAP queries

3 a). Simple Reports

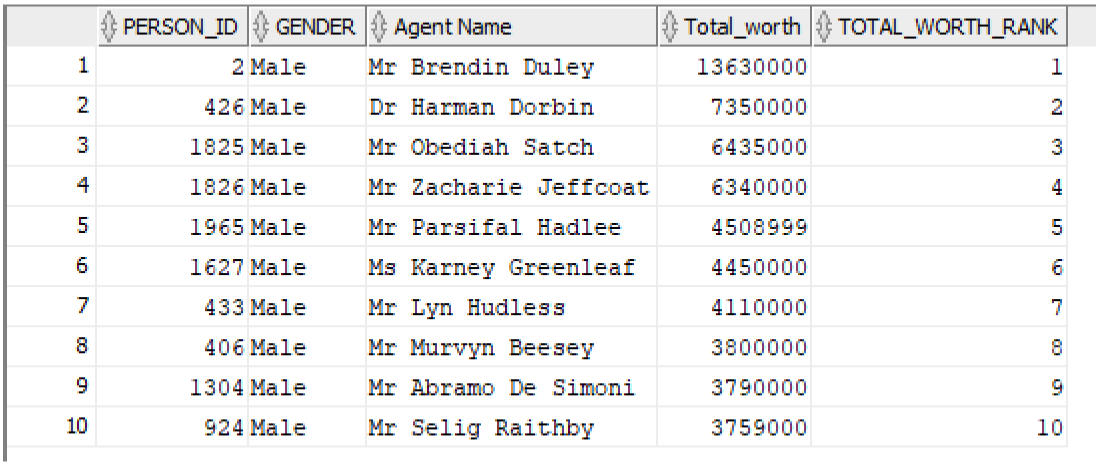
Report 1

WHO ARE THE TOP 10 MALE AGENTS IN VICTORIA

This query will help the management to find the male agents who work well for the Monash Real Estate. The ranking is calculated based on the total worth. This will enable management to appreciate the agents who work tirelessly for Monash Real Estate.

Version 1

```
SELECT
    *
FROM
    (
        SELECT
            a.person_id,
            a.gender,
            "Agent Name",
            SUM("Total Worth") AS "Total_worth",
            RANK() OVER(
                ORDER BY
                    SUM("Total Worth") DESC
            ) AS total_worth_rank
        FROM
            agent_fact_lvl2       a,
            agent_info_dim_lvl2  ai
        WHERE
            a.person_id=ai.person_id AND
            state_code = 'VIC'
            AND a.gender = 'Male'
        GROUP BY (
            a.person_id,
            a.gender,
            "Agent Name"
        )
    )
WHERE
    total_worth_rank <= 10;
```

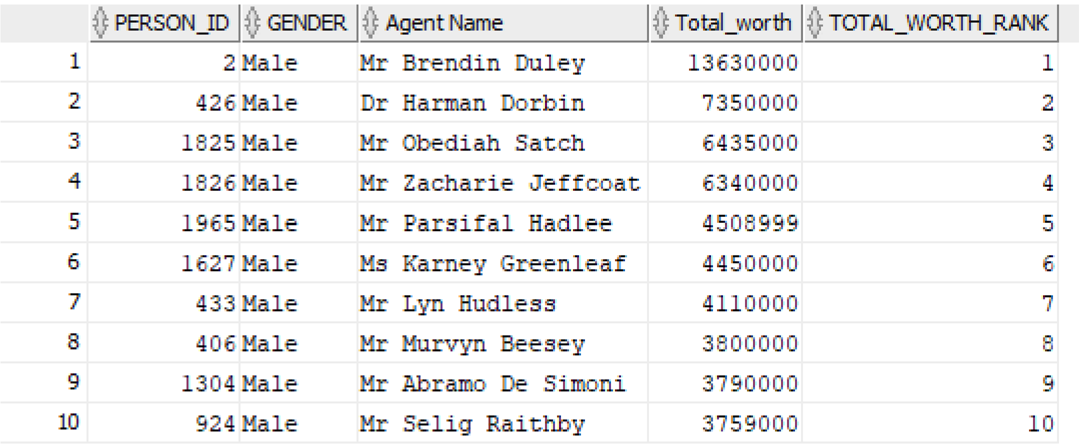


	PERSON_ID	GENDER	Agent Name	Total_worth	TOTAL_WORTH_RANK
1	2	Male	Mr Brendin Duley	13630000	1
2	426	Male	Dr Harman Dorbin	7350000	2
3	1825	Male	Mr Obediah Satch	6435000	3
4	1826	Male	Mr Zacharie Jeffcoat	6340000	4
5	1965	Male	Mr Parsifal Hadlee	4508999	5
6	1627	Male	Ms Karney Greenleaf	4450000	6
7	433	Male	Mr Lyn Hudless	4110000	7
8	406	Male	Mr Murvyn Beesey	3800000	8
9	1304	Male	Mr Abramo De Simoni	3790000	9
10	924	Male	Mr Selig Raithby	3759000	10

Version 2

```
SELECT
    *
```

```
FROM
    (
        SELECT
            a.person_id,
            a.gender,
            "Agent Name",
            SUM("Total Worth") AS "Total_worth",
            RANK() OVER(
                ORDER BY
                    SUM("Total Worth") DESC
            ) AS total_worth_rank
        FROM
            agent_fact_lvl0       a,
            property_dim_lvl0     p,
            agent_info_dim_lvl0  ai
        WHERE
            a.property_id = p.property_id
            AND a.person_id = ai.person_id
            AND p.state_code = 'VIC'
            AND a.gender = 'Male'
        GROUP BY (
            a.person_id,
            a.gender,
            "Agent Name"
        )
    )
WHERE
    total_worth_rank <= 10;
```



	PERSON_ID	GENDER	Agent Name	Total_worth	TOTAL_WORTH_RANK
1	2	Male	Mr Brendin Duley	13630000	1
2	426	Male	Dr Harman Dorbin	7350000	2
3	1825	Male	Mr Obediah Satch	6435000	3
4	1826	Male	Mr Zacharie Jeffcoat	6340000	4
5	1965	Male	Mr Parsifal Hadlee	4508999	5
6	1627	Male	Ms Karney Greenleaf	4450000	6
7	433	Male	Mr Lyn Hudless	4110000	7
8	406	Male	Mr Murvyn Beesey	3800000	8
9	1304	Male	Mr Abramo De Simoni	3790000	9
10	924	Male	Mr Selig Raithby	3759000	10

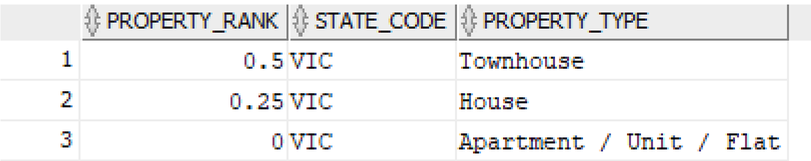
Report 2

WHAT ARE THE TOP PERCENTAGE OF PROPERTIES RENTED IN VICTORIA BASED ON DIFFERENT PROPERTY TYPES

This will help the management to figure out which property type is most liked by the residents of Victoria.

Version 1

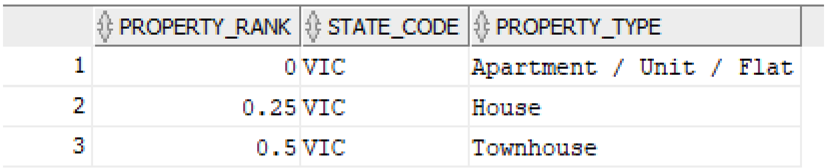
```
SELECT
    *
FROM
    (
        SELECT
            PERCENT_RANK() OVER(
                ORDER BY
                    SUM("Total Number of Rent") DESC
            ) AS property_rank,
            state_code,
            property_type
        FROM
            rent_fact_lvl2
        WHERE
            state_code = 'VIC'
        GROUP BY (
            state_code,
            property_type
        )
    )
WHERE
    property_rank <= 0.50
    Order by PROPERTY_RANK desc;
```



	PROPERTY_RANK	STATE_CODE	PROPERTY_TYPE
1	0.5	VIC	Townhouse
2	0.25	VIC	House
3	0	VIC	Apartment / Unit / Flat

Version 2

```
SELECT
    *
FROM
    (
        SELECT
            PERCENT_RANK() OVER(
                ORDER BY
                    COUNT("Total Number of Rent") DESC
            ) AS property_rank,
            p.state_code,
            r.property_type
        FROM
            rent_fact_lvl0      r,
            property_dim_lvl0  p
        WHERE
            p.property_id = r.property_id
            AND p.state_code = 'VIC'
        GROUP BY (
            p.state_code,
            r.property_type
        )
    )
WHERE
    property_rank <= 0.50;
```



	PROPERTY_RANK	STATE_CODE	PROPERTY_TYPE
1	0	VIC	Apartment / Unit / Flat
2	0.25	VIC	House
3	0.5	VIC	Townhouse

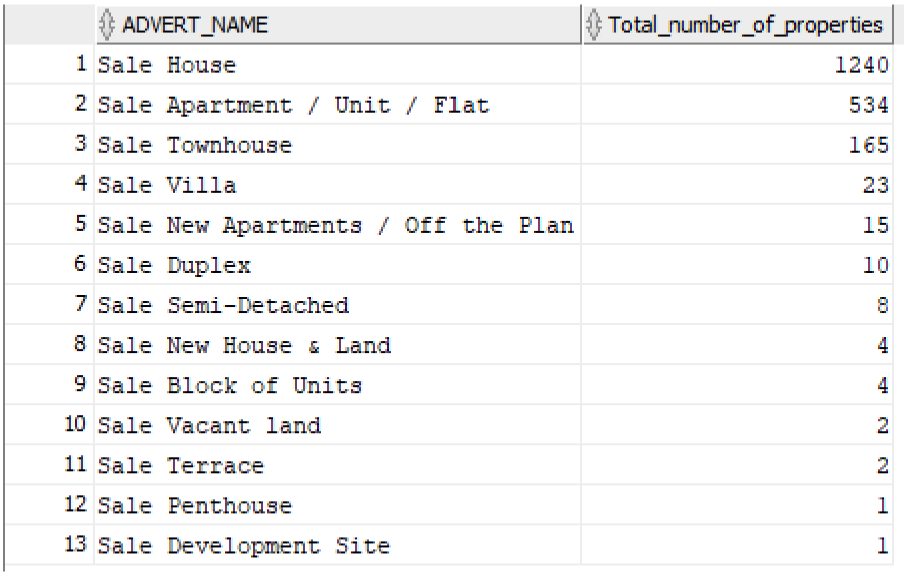
REPORT 3

<u>ALL THE ADVERTISEMENT NAME AND THE TOTAL NUMBER OF PROPERTIES ADVERTISED UNDER IT</u>

This will help the management to figure out how many properties are advertised under an advertisment name, so that they can categorise and give high importance to advertise name that has more properties.

Version 1:

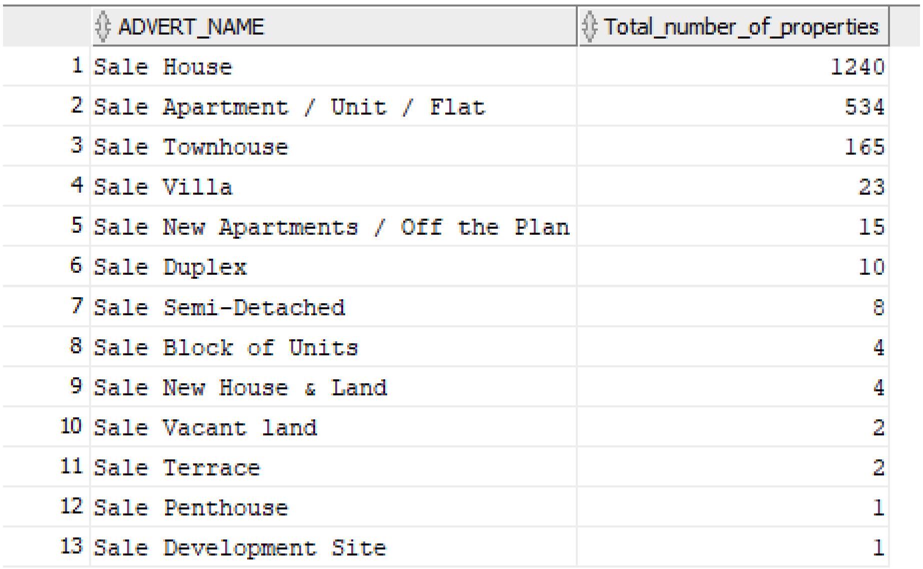
```
SELECT
    ad.advert_name,
    SUM("Total number of Properties") AS "Total_number_of_properties"
FROM
    advertisement_fact_lvl2 af,
    advert_dim_lvl2           ad
WHERE
    af.advert_id = ad.advert_id
    AND ad.advert_name LIKE 'Sale%'
GROUP BY (
    ad.advert_name
)
ORDER BY
    SUM("Total number of Properties") DESC;
```



	ADVERT_NAME	Total_number_of_properties
1	Sale House	1240
2	Sale Apartment / Unit / Flat	534
3	Sale Townhouse	165
4	Sale Villa	23
5	Sale New Apartments / Off the Plan	15
6	Sale Duplex	10
7	Sale Semi-Detached	8
8	Sale New House & Land	4
9	Sale Block of Units	4
10	Sale Vacant land	2
11	Sale Terrace	2
12	Sale Penthouse	1
13	Sale Development Site	1

Version 2

```
SELECT
    ad.advert_name,
    SUM("Total number of Properties") AS "Total_number_of_properties"
FROM
    advertisement_fact_lvl0          af,
    property_advert_bridge_dim_lvl0 pa,
    advert_dim_lvl0                  ad
WHERE
    af.property_id = pa.property_id
    AND pa.advert_id = ad.advert_id
    AND ad.advert_name LIKE 'Sale%'
GROUP BY (
    ad.advert_name
)
ORDER BY
    SUM("Total number of Properties") DESC;
```



	ADVERT_NAME	Total_number_of_properties
1	Sale House	1240
2	Sale Apartment / Unit / Flat	534
3	Sale Townhouse	165
4	Sale Villa	23
5	Sale New Apartments / Off the Plan	15
6	Sale Duplex	10
7	Sale Semi-Detached	8
8	Sale Block of Units	4
9	Sale New House & Land	4
10	Sale Vacant land	2
11	Sale Terrace	2
12	Sale Penthouse	1
13	Sale Development Site	1

3 b). Reports with proper subtotals

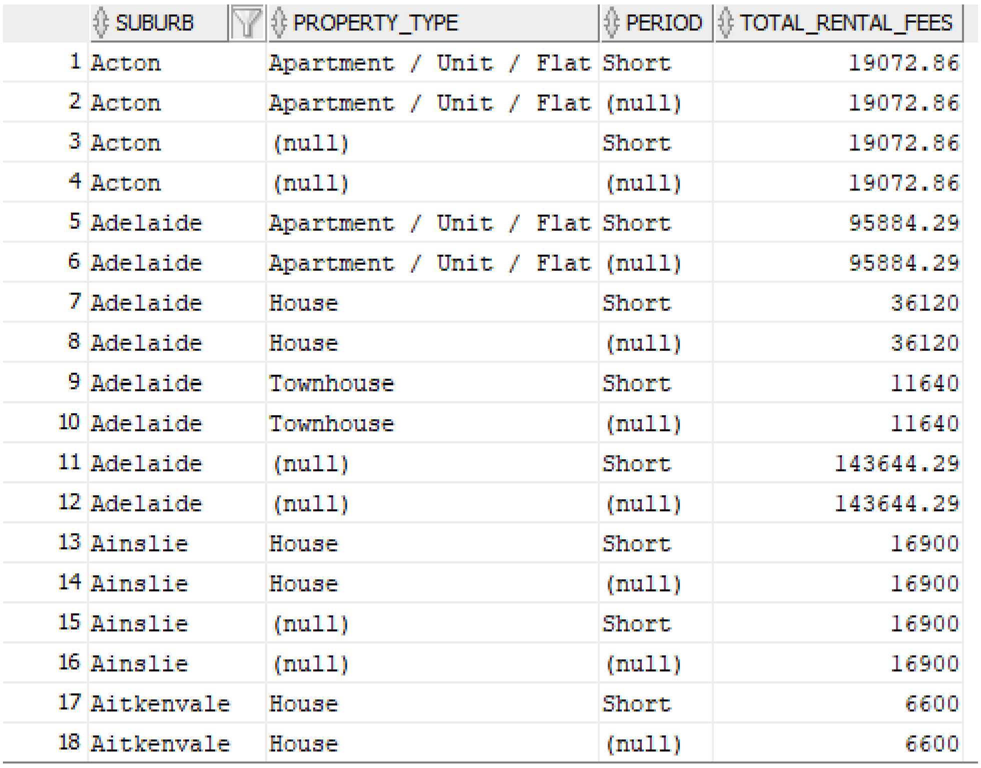
Report 4 and 5:

SUB-TOTAL AND TOTAL RENTAL FEES FROM EACH SUBURB, TIME PERIOD AND PROPERTY TYPE

Version 1

Cube

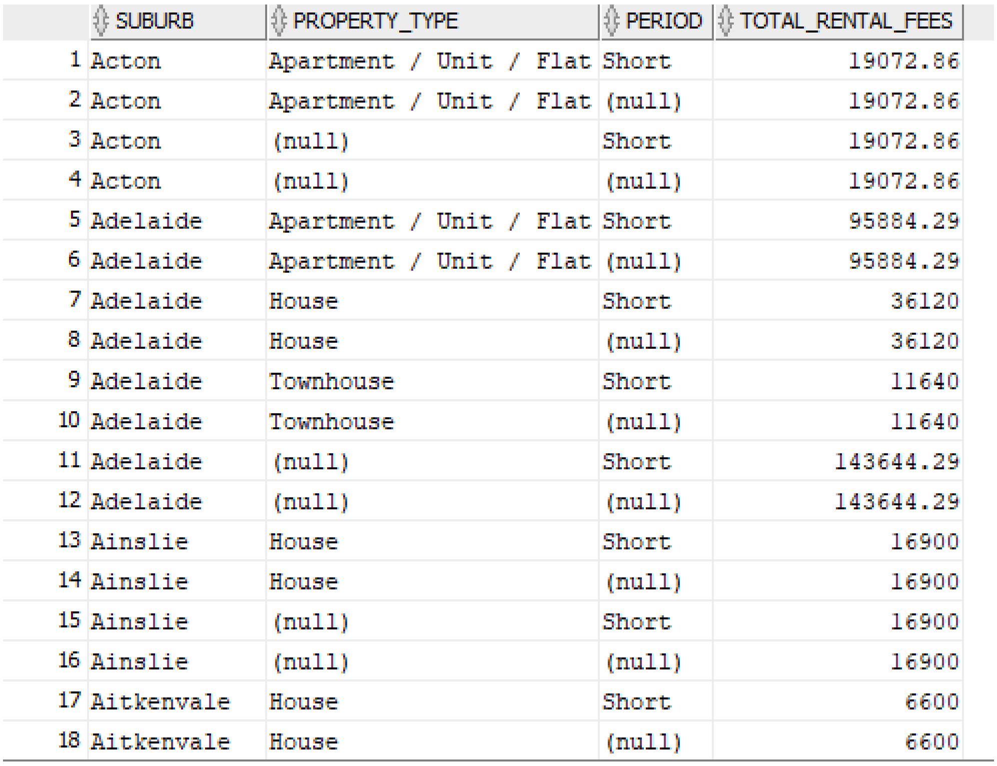
```
SELECT
   suburb,
   property_type,
   period,
   round(SUM("Total Rental Fees"), 2) AS total_rental_fees
FROM
   rent_fact_lvl2
GROUP BY
   CUBE(suburb,
       property_type,
       period)
ORDER BY
   suburb;
```



	SUBURB	PROPERTY_TYPE	PERIOD	TOTAL_RENTAL_FEES
1	Acton	Apartment / Unit / Flat	Short	19072.86
2	Acton	Apartment / Unit / Flat	(null)	19072.86
3	Acton	(null)	Short	19072.86
4	Acton	(null)	(null)	19072.86
5	Adelaide	Apartment / Unit / Flat	Short	95884.29
6	Adelaide	Apartment / Unit / Flat	(null)	95884.29
7	Adelaide	House	Short	36120
8	Adelaide	House	(null)	36120
9	Adelaide	Townhouse	Short	11640
10	Adelaide	Townhouse	(null)	11640
11	Adelaide	(null)	Short	143644.29
12	Adelaide	(null)	(null)	143644.29
13	Ainslie	House	Short	16900
14	Ainslie	House	(null)	16900
15	Ainslie	(null)	Short	16900
16	Ainslie	(null)	(null)	16900
17	Aitkenvale	House	Short	6600
18	Aitkenvale	House	(null)	6600

Partial Cube

```
SELECT
   suburb,
   property_type,
   period,
   round(SUM("Total Rental Fees"), 2) AS total_rental_fees
FROM
   rent_fact_lvl2
GROUP BY
   suburb,
   CUBE(property_type,
       period)
ORDER BY
   suburb;
```

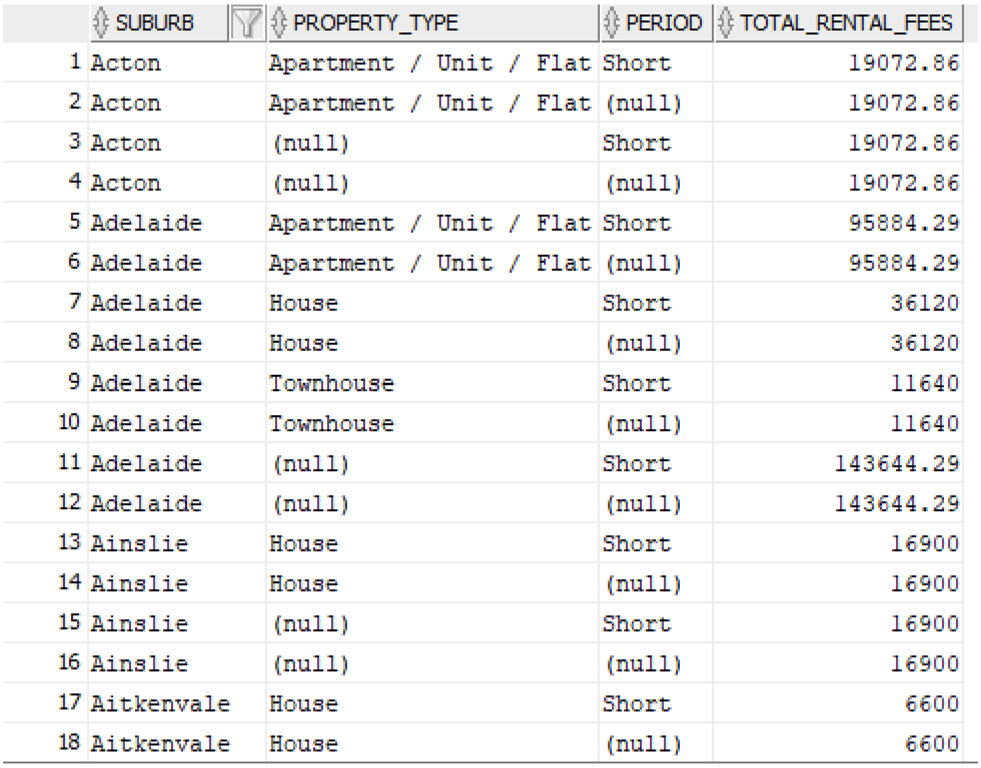


	SUBURB	PROPERTY_TYPE	PERIOD	TOTAL_RENTAL_FEES
1	Acton	Apartment / Unit / Flat	Short	19072.86
2	Acton	Apartment / Unit / Flat	(null)	19072.86
3	Acton	(null)	Short	19072.86
4	Acton	(null)	(null)	19072.86
5	Adelaide	Apartment / Unit / Flat	Short	95884.29
6	Adelaide	Apartment / Unit / Flat	(null)	95884.29
7	Adelaide	House	Short	36120
8	Adelaide	House	(null)	36120
9	Adelaide	Townhouse	Short	11640
10	Adelaide	Townhouse	(null)	11640
11	Adelaide	(null)	Short	143644.29
12	Adelaide	(null)	(null)	143644.29
13	Ainslie	House	Short	16900
14	Ainslie	House	(null)	16900
15	Ainslie	(null)	Short	16900
16	Ainslie	(null)	(null)	16900
17	Aitkenvale	House	Short	6600
18	Aitkenvale	House	(null)	6600

Version 2

Cube

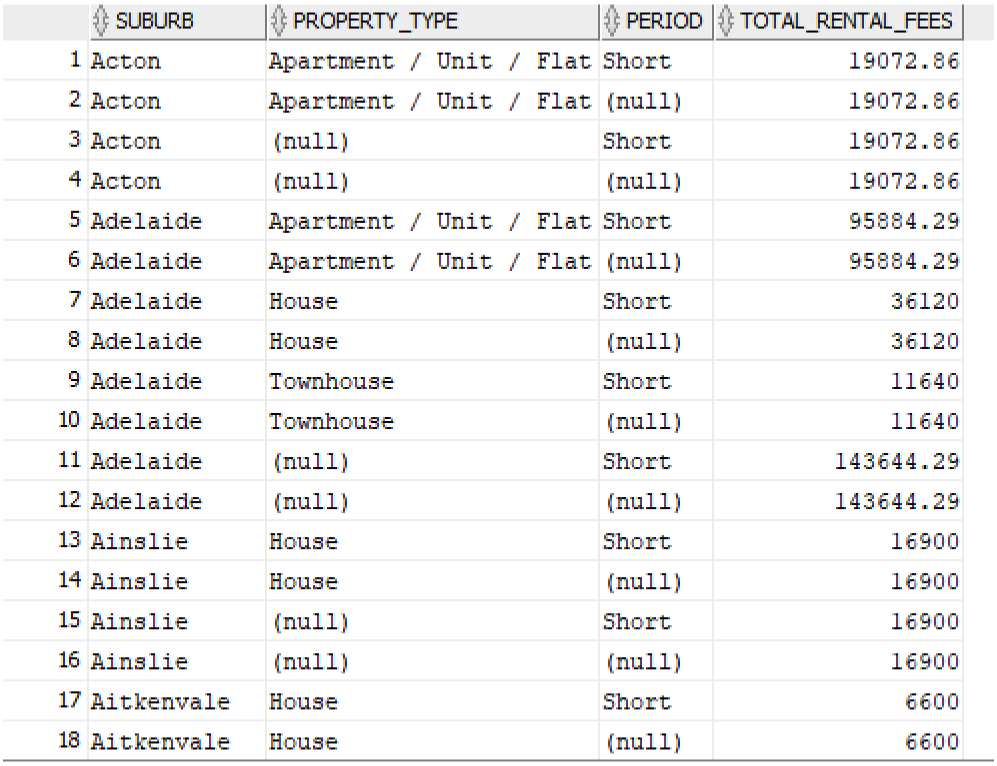
```
SELECT
    pd.suburb,
    r.property_type,
    r.period,
    round(SUM("Total Rental Fees"), 2) AS total_rental_fees
FROM
    rent_fact_lvl0      r,
    property_dim_lvl0  pd
WHERE
    pd.property_id = r.property_id
GROUP BY
    CUBE(pd.suburb,
         r.property_type,
         r.period)
ORDER BY
    pd.suburb;
```



	SUBURB	PROPERTY_TYPE	PERIOD	TOTAL_RENTAL_FEES
1	Acton	Apartment / Unit / Flat	Short	19072.86
2	Acton	Apartment / Unit / Flat	(null)	19072.86
3	Acton	(null)	Short	19072.86
4	Acton	(null)	(null)	19072.86
5	Adelaide	Apartment / Unit / Flat	Short	95884.29
6	Adelaide	Apartment / Unit / Flat	(null)	95884.29
7	Adelaide	House	Short	36120
8	Adelaide	House	(null)	36120
9	Adelaide	Townhouse	Short	11640
10	Adelaide	Townhouse	(null)	11640
11	Adelaide	(null)	Short	143644.29
12	Adelaide	(null)	(null)	143644.29
13	Ainslie	House	Short	16900
14	Ainslie	House	(null)	16900
15	Ainslie	(null)	Short	16900
16	Ainslie	(null)	(null)	16900
17	Aitkenvale	House	Short	6600
18	Aitkenvale	House	(null)	6600

Partial Cube

```
SELECT
    pd.suburb,
    r.property_type,
    r.period,
    round(SUM("Total Rental Fees"), 2) AS total_rental_fees
FROM
    rent_fact_lvl0      r,
    property_dim_lvl0  pd
WHERE
    pd.property_id = r.property_id
GROUP BY
    pd.suburb,
    CUBE(r.property_type,
         r.period)
ORDER BY
    pd.suburb;
```



	SUBURB	PROPERTY_TYPE	PERIOD	TOTAL_RENTAL_FEES
1	Acton	Apartment / Unit / Flat	Short	19072.86
2	Acton	Apartment / Unit / Flat	(null)	19072.86
3	Acton	(null)	Short	19072.86
4	Acton	(null)	(null)	19072.86
5	Adelaide	Apartment / Unit / Flat	Short	95884.29
6	Adelaide	Apartment / Unit / Flat	(null)	95884.29
7	Adelaide	House	Short	36120
8	Adelaide	House	(null)	36120
9	Adelaide	Townhouse	Short	11640
10	Adelaide	Townhouse	(null)	11640
11	Adelaide	(null)	Short	143644.29
12	Adelaide	(null)	(null)	143644.29
13	Ainslie	House	Short	16900
14	Ainslie	House	(null)	16900
15	Ainslie	(null)	Short	16900
16	Ainslie	(null)	(null)	16900
17	Aitkenvale	House	Short	6600
18	Aitkenvale	House	(null)	6600

Report 6 and 7

Sub total and Total Sales for each property type in different year in VIC and SA states

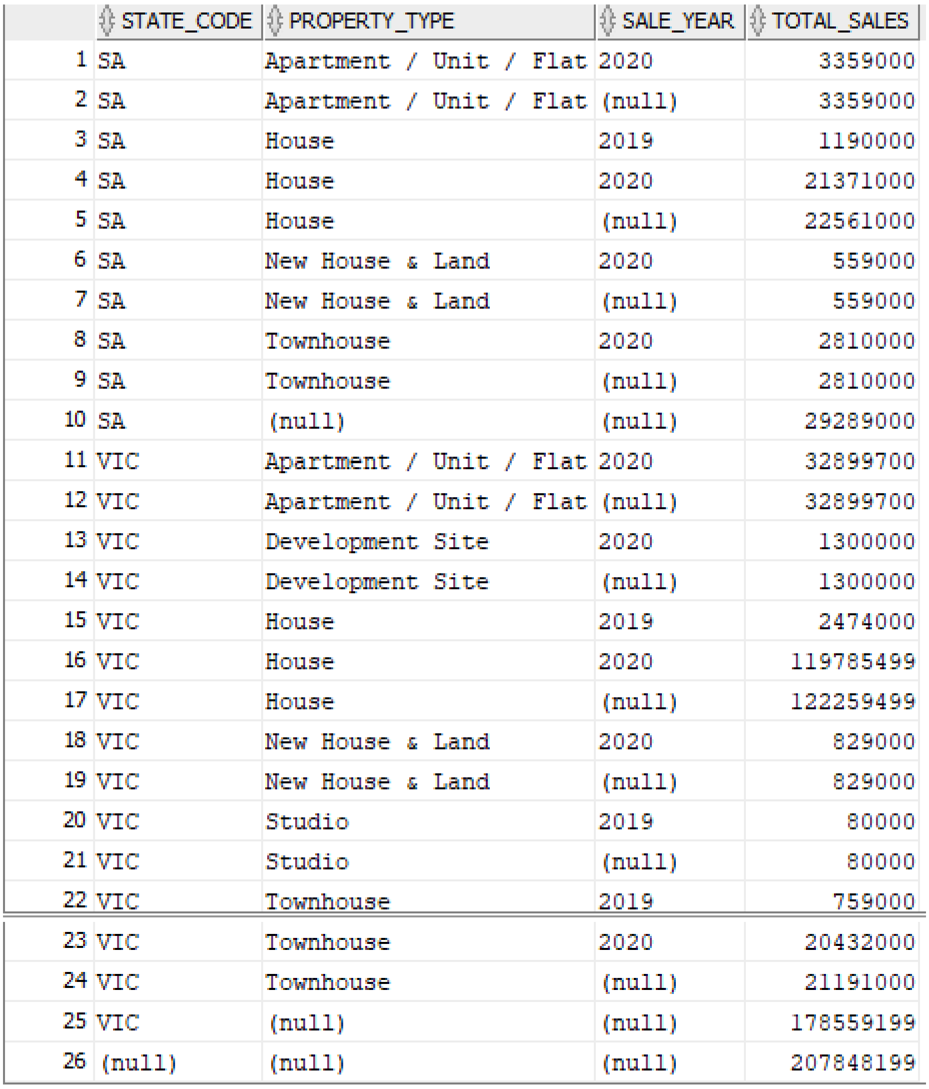
This query will help the management to find out the sub total and total sales of different property types sold in VIC and SA states based on year and property type. This will help the management to find out the popular type of property sold in each year.

Report 6 - ROLL UP

Version 1

```
SELECT
   state_code,
   property_type,
   year,
   round(SUM("Total Price"), 2) AS total_sales
FROM
   sale_fact_lvl2
WHERE
   state_code IN (
      'VIC',
      'SA'
```

```
    )
GROUP BY
    ROLLUP(state_code,
           property_type,
           year)
ORDER BY
    state_code;
```

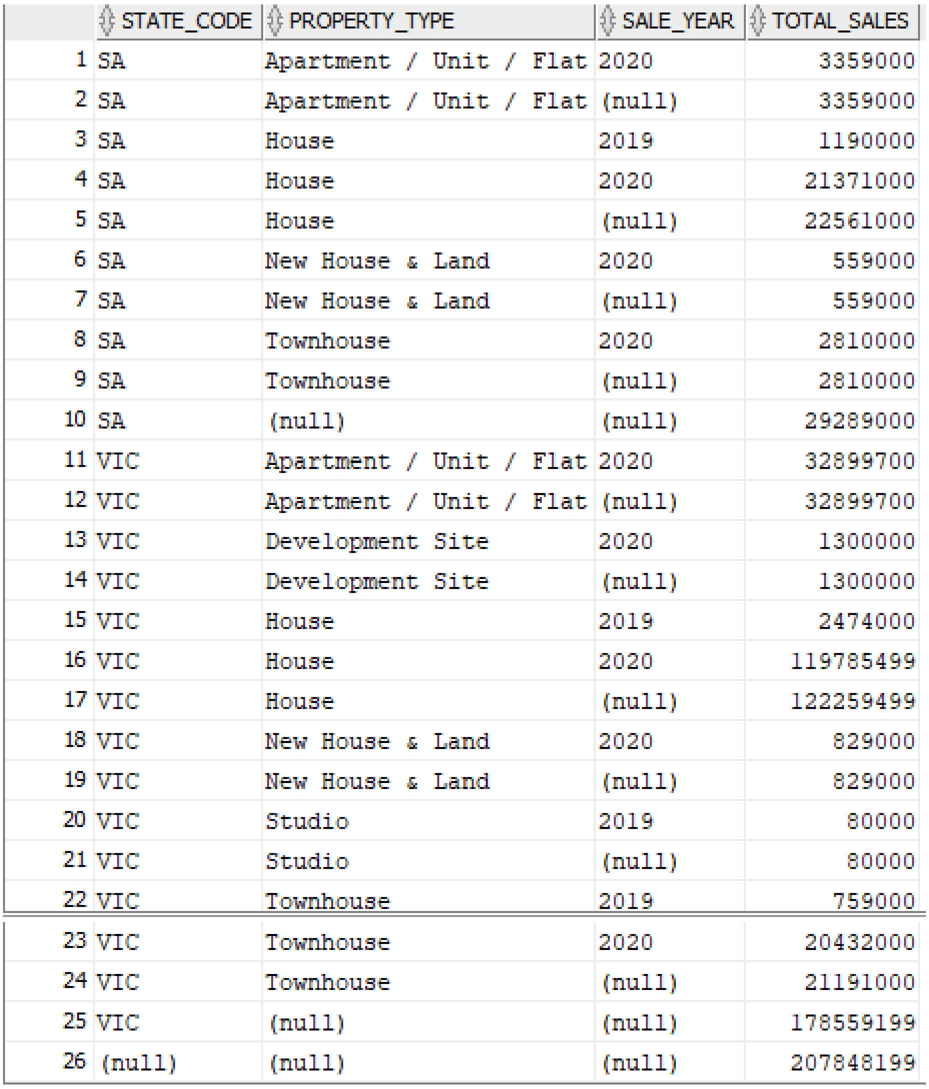


	STATE_CODE	PROPERTY_TYPE	SALE_YEAR	TOTAL_SALES
1	SA	Apartment / Unit / Flat	2020	3359000
2	SA	Apartment / Unit / Flat	(null)	3359000
3	SA	House	2019	1190000
4	SA	House	2020	21371000
5	SA	House	(null)	22561000
6	SA	New House & Land	2020	559000
7	SA	New House & Land	(null)	559000
8	SA	Townhouse	2020	2810000
9	SA	Townhouse	(null)	2810000
10	SA	(null)	(null)	29289000
11	VIC	Apartment / Unit / Flat	2020	32899700
12	VIC	Apartment / Unit / Flat	(null)	32899700
13	VIC	Development Site	2020	1300000
14	VIC	Development Site	(null)	1300000
15	VIC	House	2019	2474000
16	VIC	House	2020	119785499
17	VIC	House	(null)	122259499
18	VIC	New House & Land	2020	829000
19	VIC	New House & Land	(null)	829000
20	VIC	Studio	2019	80000
21	VIC	Studio	(null)	80000
22	VIC	Townhouse	2019	759000
23	VIC	Townhouse	2020	20432000
24	VIC	Townhouse	(null)	21191000
25	VIC	(null)	(null)	178559199
26	(null)	(null)	(null)	207848199

Version 2

```
SELECT
    pd.state_code,
    s.property_type,
    s.sale_year,
    round(SUM("Total Price"), 2) AS total_sales
FROM
    sale_fact_lvl0      s,
    property_dim_lvl0  pd
```

```
WHERE
    pd.property_id = s.property_id
    AND pd.state_code IN (
        'VIC',
        'SA'
    )
GROUP BY
    ROLLUP(pd.state_code,
           s.property_type,
           s.sale_year)
ORDER BY
    pd.state_code;
```

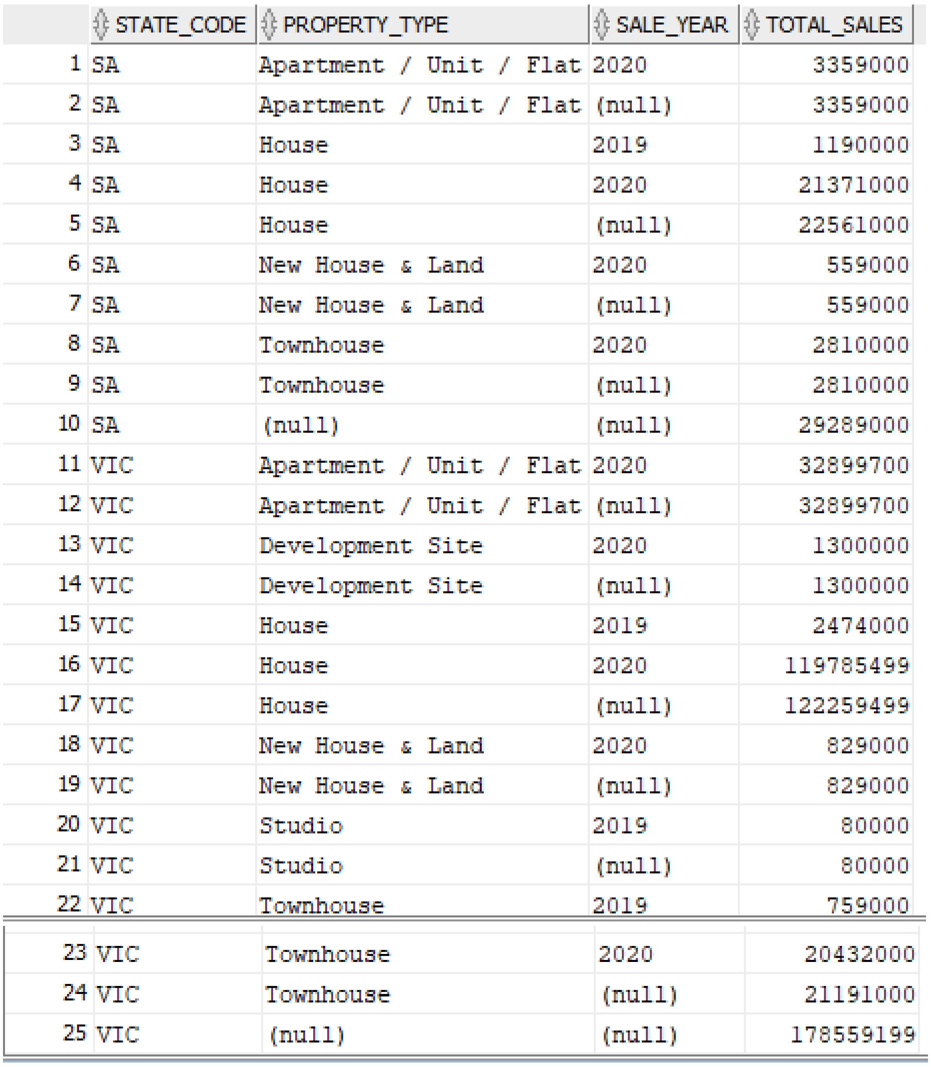


	STATE_CODE	PROPERTY_TYPE	SALE_YEAR	TOTAL_SALES
1	SA	Apartment / Unit / Flat	2020	3359000
2	SA	Apartment / Unit / Flat	(null)	3359000
3	SA	House	2019	1190000
4	SA	House	2020	21371000
5	SA	House	(null)	22561000
6	SA	New House & Land	2020	559000
7	SA	New House & Land	(null)	559000
8	SA	Townhouse	2020	2810000
9	SA	Townhouse	(null)	2810000
10	SA	(null)	(null)	29289000
11	VIC	Apartment / Unit / Flat	2020	32899700
12	VIC	Apartment / Unit / Flat	(null)	32899700
13	VIC	Development Site	2020	1300000
14	VIC	Development Site	(null)	1300000
15	VIC	House	2019	2474000
16	VIC	House	2020	119785499
17	VIC	House	(null)	122259499
18	VIC	New House & Land	2020	829000
19	VIC	New House & Land	(null)	829000
20	VIC	Studio	2019	80000
21	VIC	Studio	(null)	80000
22	VIC	Townhouse	2019	759000
23	VIC	Townhouse	2020	20432000
24	VIC	Townhouse	(null)	21191000
25	VIC	(null)	(null)	178559199
26	(null)	(null)	(null)	207848199

Report 7 - PARTIAL ROLLUP

Version 1

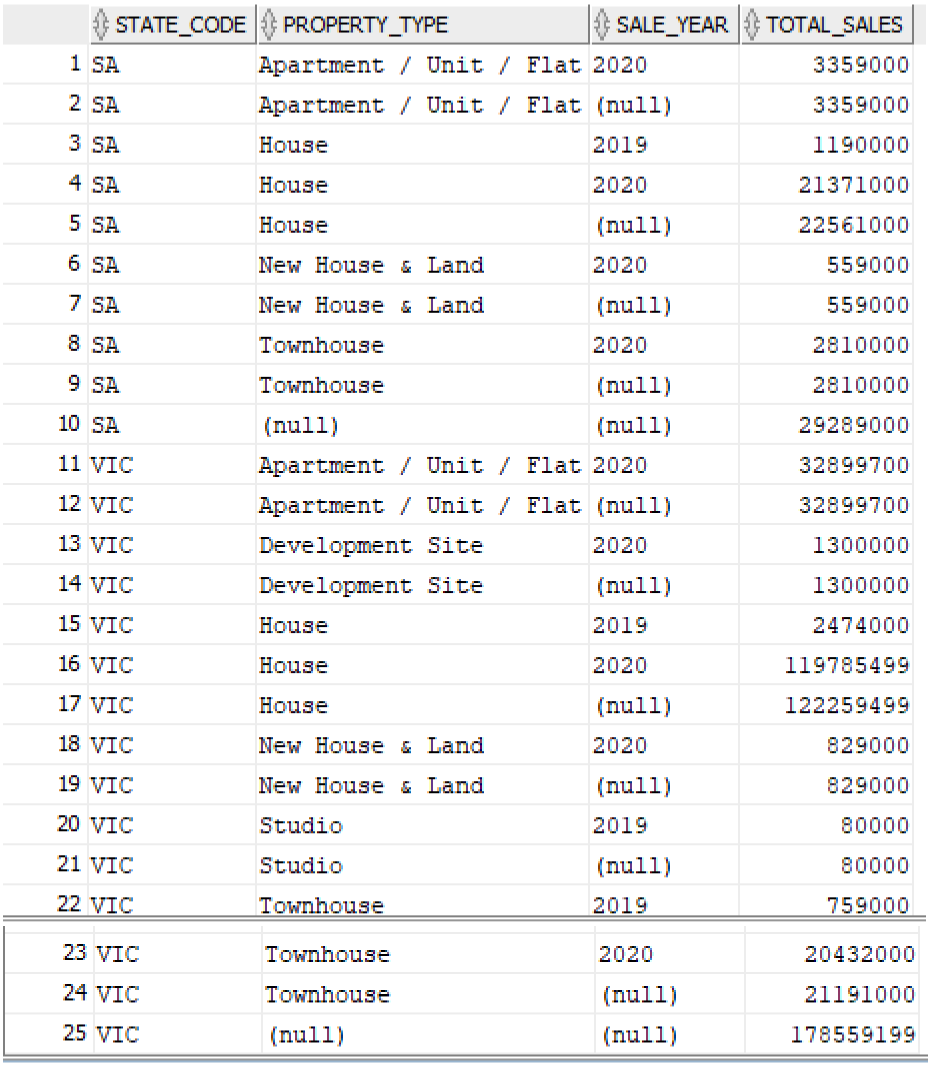
```
SELECT
    state_code,
    property_type,
    year,
    round(SUM("Total Price"), 2) AS total_sales
FROM
    sale_fact_lvl2
WHERE
    state_code IN (
        'VIC',
        'SA'
    )
GROUP BY
    state_code,
    ROLLUP(property_type,
          year)
ORDER BY
    state_code;
```



	STATE_CODE	PROPERTY_TYPE	SALE_YEAR	TOTAL_SALES
1	SA	Apartment / Unit / Flat	2020	3359000
2	SA	Apartment / Unit / Flat	(null)	3359000
3	SA	House	2019	1190000
4	SA	House	2020	21371000
5	SA	House	(null)	22561000
6	SA	New House & Land	2020	559000
7	SA	New House & Land	(null)	559000
8	SA	Townhouse	2020	2810000
9	SA	Townhouse	(null)	2810000
10	SA	(null)	(null)	29289000
11	VIC	Apartment / Unit / Flat	2020	32899700
12	VIC	Apartment / Unit / Flat	(null)	32899700
13	VIC	Development Site	2020	1300000
14	VIC	Development Site	(null)	1300000
15	VIC	House	2019	2474000
16	VIC	House	2020	119785499
17	VIC	House	(null)	122259499
18	VIC	New House & Land	2020	829000
19	VIC	New House & Land	(null)	829000
20	VIC	Studio	2019	80000
21	VIC	Studio	(null)	80000
22	VIC	Townhouse	2019	759000
23	VIC	Townhouse	2020	20432000
24	VIC	Townhouse	(null)	21191000
25	VIC	(null)	(null)	178559199

Version 2

```
SELECT
    pd.state_code,
    s.property_type,
    s.sale_year,
    round(SUM("Total Price"), 2) AS total_sales
FROM
    sale_fact_lvl0      s,
    property_dim_lvl0  pd
WHERE
    pd.property_id = s.property_id
    AND pd.state_code IN (
        'VIC',
        'SA'
    )
GROUP BY
    pd.state_code,
    ROLLUP(s.property_type,
           s.sale_year)
ORDER BY
    pd.state_code;
```



	STATE_CODE	PROPERTY_TYPE	SALE_YEAR	TOTAL_SALES
1	SA	Apartment / Unit / Flat	2020	3359000
2	SA	Apartment / Unit / Flat	(null)	3359000
3	SA	House	2019	1190000
4	SA	House	2020	21371000
5	SA	House	(null)	22561000
6	SA	New House & Land	2020	559000
7	SA	New House & Land	(null)	559000
8	SA	Townhouse	2020	2810000
9	SA	Townhouse	(null)	2810000
10	SA	(null)	(null)	29289000
11	VIC	Apartment / Unit / Flat	2020	32899700
12	VIC	Apartment / Unit / Flat	(null)	32899700
13	VIC	Development Site	2020	1300000
14	VIC	Development Site	(null)	1300000
15	VIC	House	2019	2474000
16	VIC	House	2020	119785499
17	VIC	House	(null)	122259499
18	VIC	New House & Land	2020	829000
19	VIC	New House & Land	(null)	829000
20	VIC	Studio	2019	80000
21	VIC	Studio	(null)	80000
22	VIC	Townhouse	2019	759000
23	VIC	Townhouse	2020	20432000
24	VIC	Townhouse	(null)	21191000
25	VIC	(null)	(null)	178559199

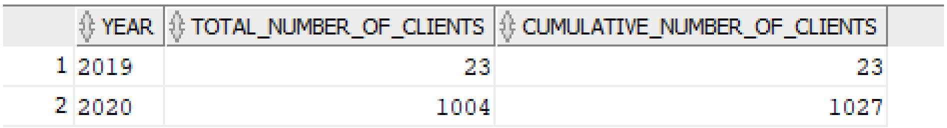
3 c). Reports with Moving and Cumulative aggregates

Report 8

Total number of clients and cumulative number of clients with a high budget in each year?

Version 1

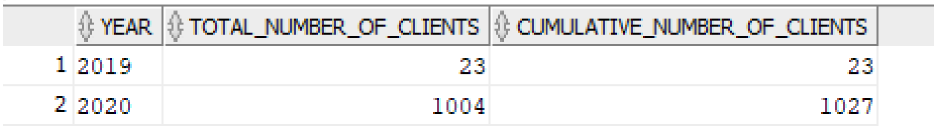
```
SELECT
    *
FROM
    client_fact_lvl2;

SELECT
    year,
    SUM("Number of Clients") AS total_number_of_clients,
    SUM(SUM("Number of Clients")) OVER(
        ORDER BY
            year
        ROWS UNBOUNDED PRECEDING
    ) AS cumulative_number_of_clients
FROM
    client_fact_lvl2
WHERE
    budget_id LIKE '%High%'
GROUP BY (
    year
);
```



	YEAR	TOTAL_NUMBER_OF_CLIENTS	CUMULATIVE_NUMBER_OF_CLIENTS
1	2019	23	23
2	2020	1004	1027

Version 2

```
SELECT
    year,
    SUM("Number of Clients") AS total_number_of_clients,
    SUM(SUM("Number of Clients")) OVER(
        ORDER BY
            year
        ROWS UNBOUNDED PRECEDING
    ) AS cumulative_number_of_clients
FROM
    client_fact_lvl0
WHERE
    budget_type LIKE '%High%'
GROUP BY (
    year
);
```



	YEAR	TOTAL_NUMBER_OF_CLIENTS	CUMULATIVE_NUMBER_OF_CLIENTS
1	2019	23	23
2	2020	1004	1027

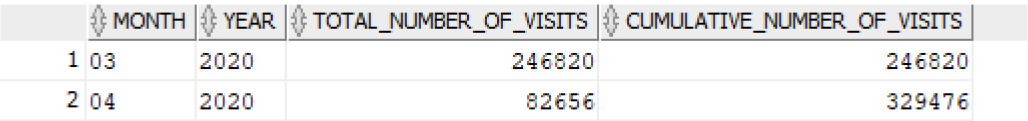
Report 9

Total number of visits and cumulative number of visits for each month in every year

This will help the management to know the total number of visits and how it has increased each month.

Version 1

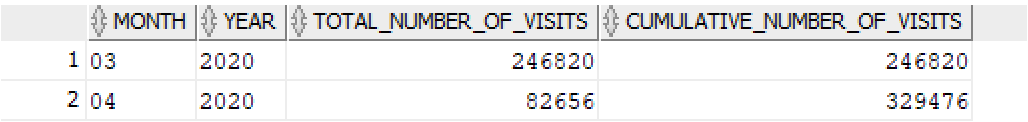
```
select to_char(v.visit_date, 'mm') as Month,
to_char(v.visit_date, 'YYYY') as Year,
sum("Total number of Visits") as Total_number_of_visits,
sum(sum("Total number of Visits")) over
(order by to_char(v.visit_date, 'mm'), to_char(v.visit_date, 'YYYY')
rows unbounded preceding) as Cumulative_number_of_visits
from visit_fact_lvl2, client_visit_dim_scd_lvl2 v
group by (to_char(v.visit_date, 'mm'), to_char(v.visit_date, 'YYYY'));
```



	MONTH	YEAR	TOTAL_NUMBER_OF_VISITS	CUMULATIVE_NUMBER_OF_VISITS
1	03	2020	246820	246820
2	04	2020	82656	329476

Version 2

```
SELECT
    to_char(v.visit_date, 'mm') AS month,
    to_char(v.visit_date, 'yyyy') AS year,
    SUM("Total number of Visits") AS total_number_of_visits,
    SUM(SUM("Total number of Visits")) OVER(
        ORDER BY
         to_char(v.visit_date, 'yyyy'),
            to_char(v.visit_date, 'mm')
        ROWS UNBOUNDED PRECEDING
    ) AS cumulative_number_of_visits
FROM
    visit_fact_l0,
    client_visit_dim_scd_l0 v
GROUP BY (
    to_char(v.visit_date, 'mm'),
    to_char(v.visit_date, 'yyyy')
);
```



	MONTH	YEAR	TOTAL_NUMBER_OF_VISITS	CUMULATIVE_NUMBER_OF_VISITS
1	03	2020	246820	246820
2	04	2020	82656	329476

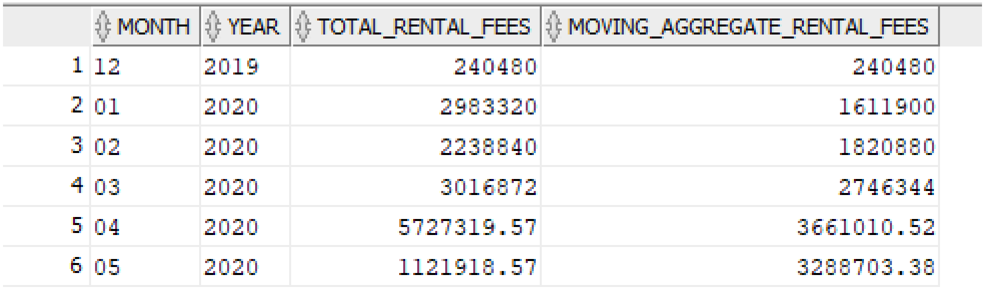
Report 10

TOTAL RENTAL AND MOVING AGGREGATE OF RENTAL FEE FOR EACH MONTH OF DIFFERENT YEARS

This report will help the management to find out the prior two months average rental fees received by the Monash real estate every month.

Version 1

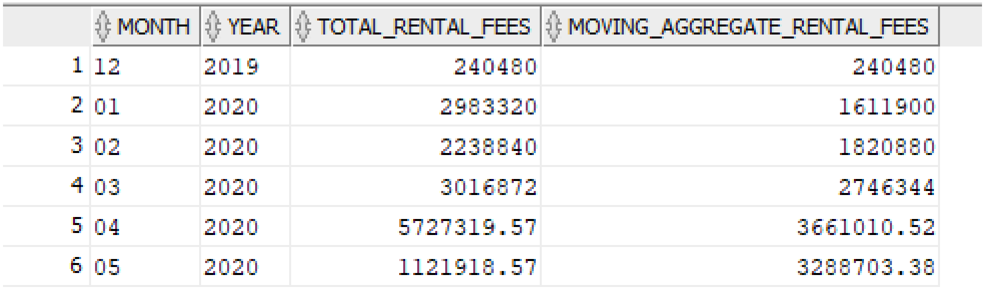
```
SELECT
    to_char(r.rent_start_date, 'mm') AS month,
    to_char(r.rent_start_date, 'yyyy') AS year,
    round(SUM("Total Rental Fees"), 2) AS total_rental_fees,
    round(AVG(SUM("Total Rental Fees")) OVER(
        ORDER BY
        to_char(r.rent_start_date, 'yyyy'),
            to_char(r.rent_start_date, 'mm')

        ROWS 2 PRECEDING
    ), 2) AS moving_aggregate_rental_fees
FROM
    rent_fact_lvl2          rf,
    property_rent_scd_lvl2  r
WHERE
    rf.property_id = r.property_id
GROUP BY (
    to_char(r.rent_start_date, 'mm'),
    to_char(r.rent_start_date, 'yyyy')
);
```



	MONTH	YEAR	TOTAL_RENTAL_FEES	MOVING_AGGREGATE_RENTAL_FEES
1	12	2019	240480	240480
2	01	2020	2983320	1611900
3	02	2020	2238840	1820880
4	03	2020	3016872	2746344
5	04	2020	5727319.57	3661010.52
6	05	2020	1121918.57	3288703.38

Version 2

```
SELECT
    to_char(r.rent_start_date, 'mm') AS month,
    to_char(r.rent_start_date, 'YYYY') AS year,
    round(SUM("Total Rental Fees"), 2) AS total_rental_fees,
    round(AVG(SUM("Total Rental Fees")) OVER(
        ORDER BY
            to_char(r.rent_start_date, 'YYYY'),
            to_char(r.rent_start_date, 'mm')


        ROWS 2 PRECEDING
    ), 2) AS moving_aggregate_rental_fees
FROM
    rent_fact_l0           rf,
    property_rent_scd_l0  r
WHERE
    rf.property_id = r.property_id
GROUP BY (
    to_char(r.rent_start_date, 'mm'),
    to_char(r.rent_start_date, 'YYYY')
);
```



	MONTH	YEAR	TOTAL_RENTAL_FEES	MOVING_AGGREGATE_RENTAL_FEES
1	12	2019	240480	240480
2	01	2020	2983320	1611900
3	02	2020	2238840	1820880
4	03	2020	3016872	2746344
5	04	2020	5727319.57	3661010.52
6	05	2020	1121918.57	3288703.38

3 d). Reports with Partitions

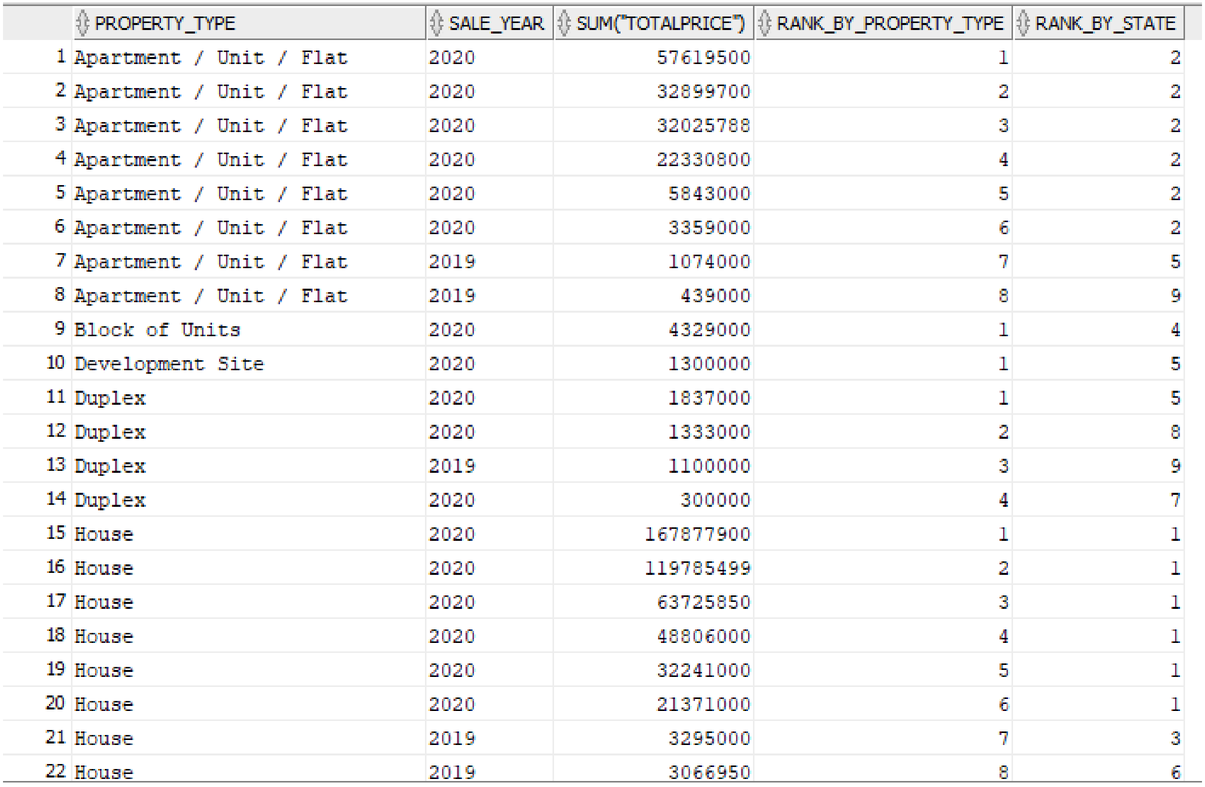
Report 11

<u>Ranking of each property type based on the yearly total number of sales and the ranking of each state based on the yearly total number of sales.</u>

Version 1

** Note: The total sale in this question is assumed as the total price

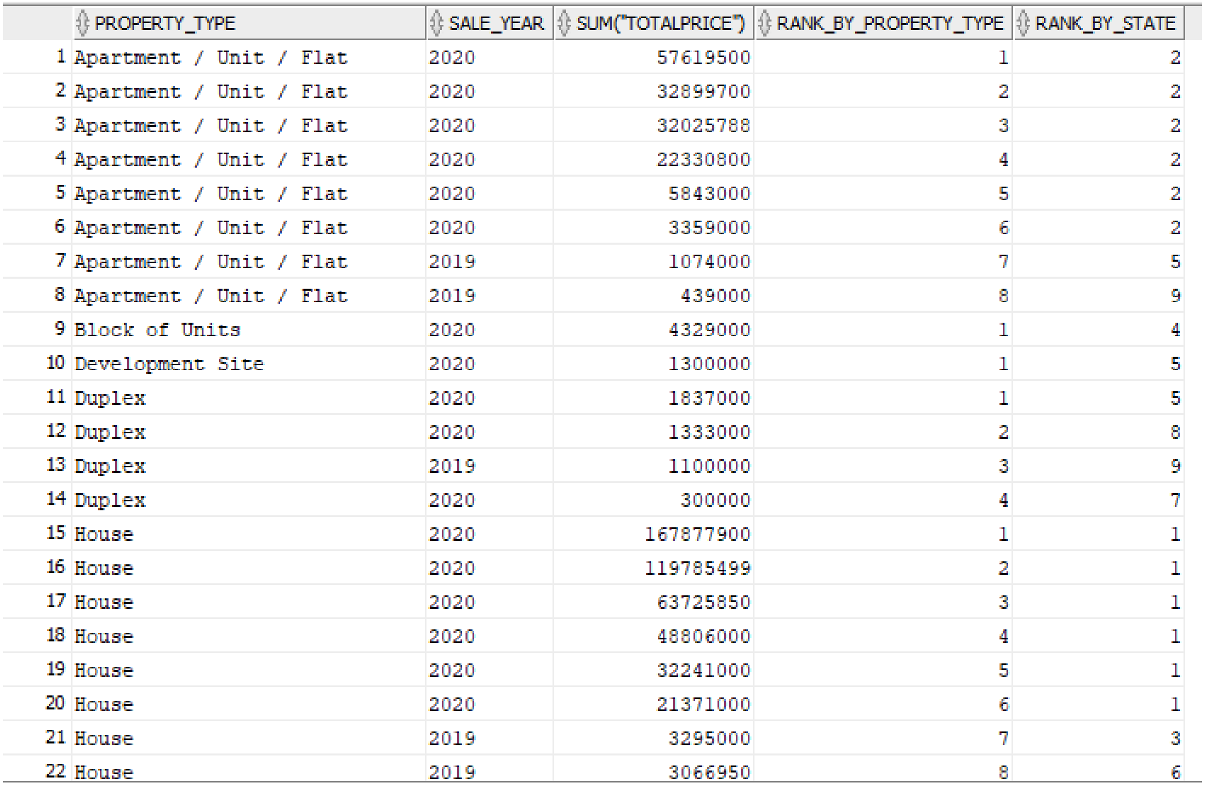
```
SELECT
    property_type,
    year,
    SUM("Total Price"),
    RANK() OVER(
        PARTITION BY property_type
        ORDER BY
            SUM("Total Price") DESC
    ) AS rank_by_property_type,
    RANK() OVER(
        PARTITION BY state_code
        ORDER BY
            SUM("Total Price") DESC
    ) AS rank_by_state
FROM
    sale_fact_lvl2
GROUP BY (
    property_type,
    year,
    state_code
);
```



	PROPERTY_TYPE	SALE_YEAR	SUM("TOTALPRICE")	RANK_BY_PROPERTY_TYPE	RANK_BY_STATE
1	Apartment / Unit / Flat	2020	57619500	1	2
2	Apartment / Unit / Flat	2020	32899700	2	2
3	Apartment / Unit / Flat	2020	32025788	3	2
4	Apartment / Unit / Flat	2020	22330800	4	2
5	Apartment / Unit / Flat	2020	5843000	5	2
6	Apartment / Unit / Flat	2020	3359000	6	2
7	Apartment / Unit / Flat	2019	1074000	7	5
8	Apartment / Unit / Flat	2019	439000	8	9
9	Block of Units	2020	4329000	1	4
10	Development Site	2020	1300000	1	5
11	Duplex	2020	1837000	1	5
12	Duplex	2020	1333000	2	8
13	Duplex	2019	1100000	3	9
14	Duplex	2020	300000	4	7
15	House	2020	167877900	1	1
16	House	2020	119785499	2	1
17	House	2020	63725850	3	1
18	House	2020	48806000	4	1
19	House	2020	32241000	5	1
20	House	2020	21371000	6	1
21	House	2019	3295000	7	3
22	House	2019	3066950	8	6

Version 2

```
SELECT
    s.property_type,
    s.sale_year,
    SUM("Total Price"),
    RANK() OVER(
        PARTITION BY s.property_type
        ORDER BY
            SUM("Total Price") DESC
    ) AS rank_by_property_type,
    RANK() OVER(
        PARTITION BY p.state_code
        ORDER BY
        SUM("Total Price") DESC
    ) AS rank_by_state
FROM
    sale_fact_lvl0      s,
    property_dim_lvl0  p
WHERE
    s.property_id = p.property_id
GROUP BY (
    s.property_type,
    s.sale_year,
    p.state_code
);
```



	PROPERTY_TYPE	SALE_YEAR	SUM("TOTALPRICE")	RANK_BY_PROPERTY_TYPE	RANK_BY_STATE
1	Apartment / Unit / Flat	2020	57619500	1	2
2	Apartment / Unit / Flat	2020	32899700	2	2
3	Apartment / Unit / Flat	2020	32025788	3	2
4	Apartment / Unit / Flat	2020	22330800	4	2
5	Apartment / Unit / Flat	2020	5843000	5	2
6	Apartment / Unit / Flat	2020	3359000	6	2
7	Apartment / Unit / Flat	2019	1074000	7	5
8	Apartment / Unit / Flat	2019	439000	8	9
9	Block of Units	2020	4329000	1	4
10	Development Site	2020	1300000	1	5
11	Duplex	2020	1837000	1	5
12	Duplex	2020	1333000	2	8
13	Duplex	2019	1100000	3	9
14	Duplex	2020	300000	4	7
15	House	2020	167877900	1	1
16	House	2020	119785499	2	1
17	House	2020	63725850	3	1
18	House	2020	48806000	4	1
19	House	2020	32241000	5	1
20	House	2020	21371000	6	1
21	House	2019	3295000	7	3
22	House	2019	3066950	8	6

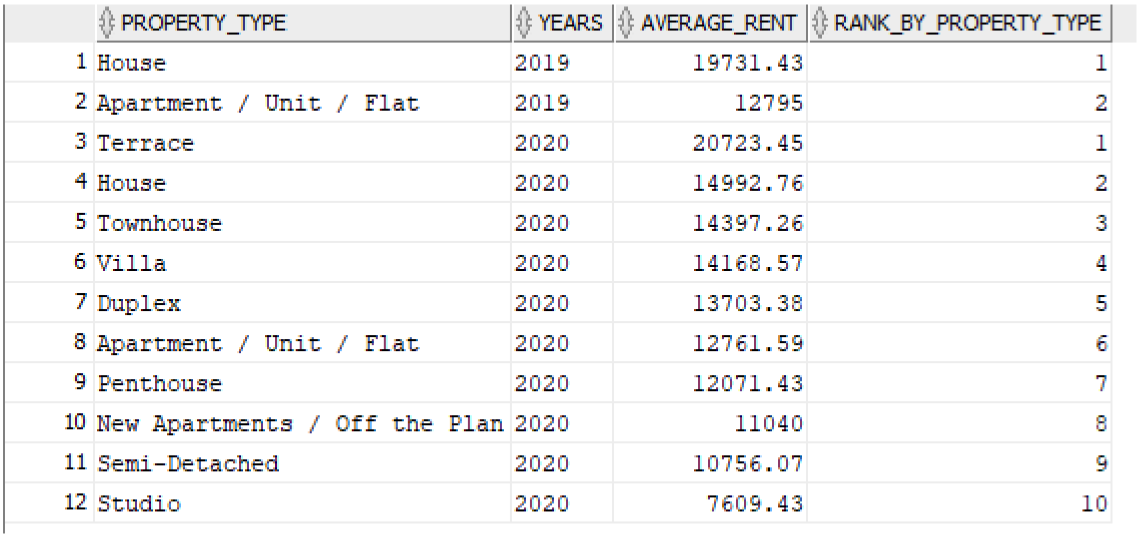
Report 12

SHOW THE RANK OF PROPERTY TYPES BASED ON AVERAGE RENT PARTITIONED BY YEAR

This query will help the management to find out the top property type based on average rent received in each year. So, this will make the management to decide the property type which is popular among the tenants.

Version 1

```
SELECT
    property_type,
    years,
    round(AVG("Total Rental Fees"), 2) AS average_rent,
    RANK() OVER(
        PARTITION BY years
        ORDER BY
            AVG("Total Rental Fees") DESC
    ) AS rank_by_property_type
FROM
    rent_fact_lvl2
GROUP BY (
    years,
    property_type
)
ORDER BY
    years;
```

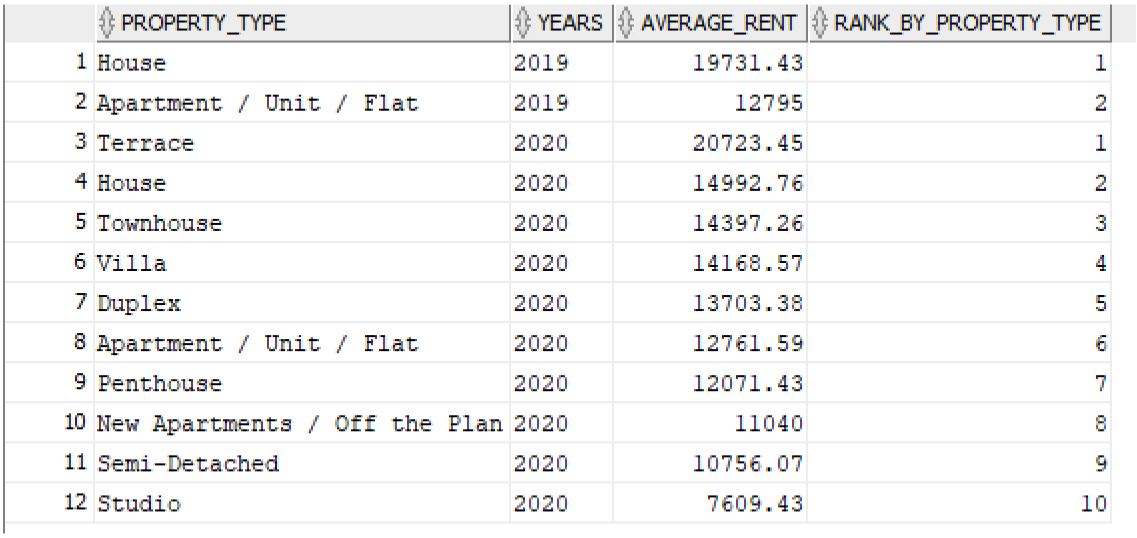


	PROPERTY_TYPE	YEARS	AVERAGE_RENT	RANK_BY_PROPERTY_TYPE
1	House	2019	19731.43	1
2	Apartment / Unit / Flat	2019	12795	2
3	Terrace	2020	20723.45	1
4	House	2020	14992.76	2
5	Townhouse	2020	14397.26	3
6	Villa	2020	14168.57	4
7	Duplex	2020	13703.38	5
8	Apartment / Unit / Flat	2020	12761.59	6
9	Penthouse	2020	12071.43	7
10	New Apartments / Off the Plan	2020	11040	8
11	Semi-Detached	2020	10756.07	9
12	Studio	2020	7609.43	10

Version 2

```
SELECT
    property_type,
    years,
    round(AVG("Total Rental Fees"), 2) AS average_rent,
    RANK() OVER(
        PARTITION BY years
        ORDER BY
            AVG("Total Rental Fees") DESC
    ) AS rank_by_property_type
FROM
```

```
    rent_fact_lvl0
GROUP BY (
    years,
    property_type
)
ORDER BY
    years;
```



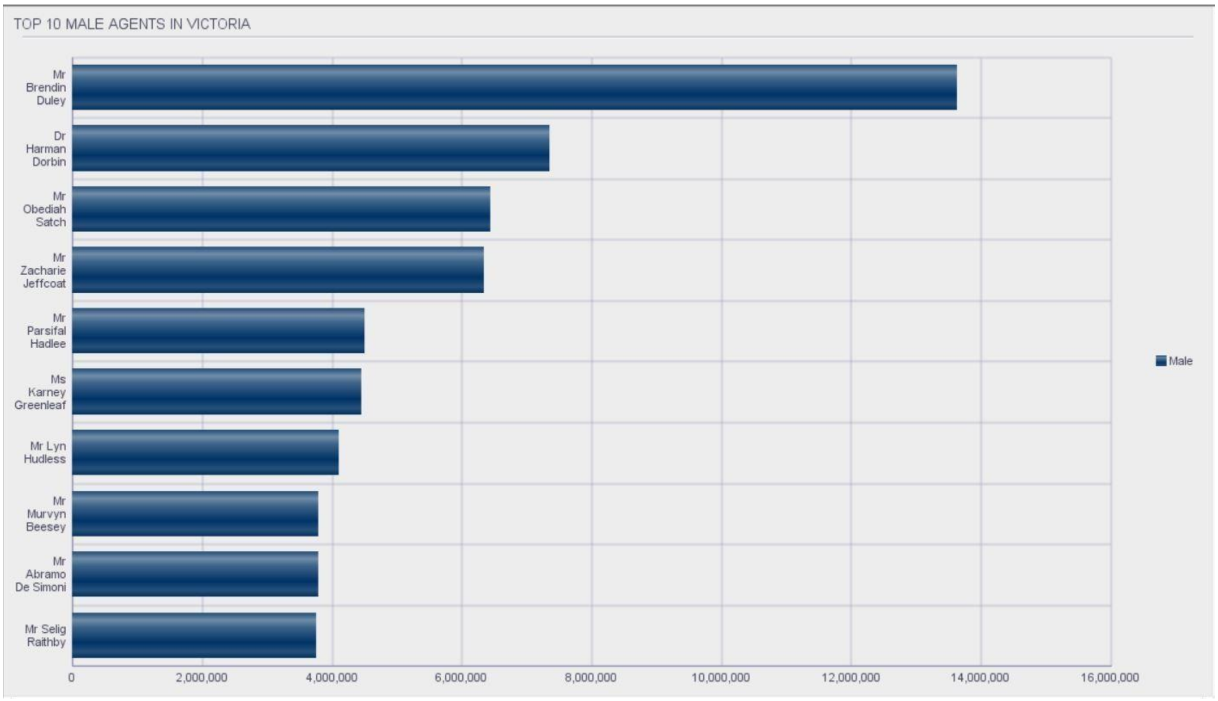
	PROPERTY_TYPE	YEARS	AVERAGE_RENT	RANK_BY_PROPERTY_TYPE
1	House	2019	19731.43	1
2	Apartment / Unit / Flat	2019	12795	2
3	Terrace	2020	20723.45	1
4	House	2020	14992.76	2
5	Townhouse	2020	14397.26	3
6	Villa	2020	14168.57	4
7	Duplex	2020	13703.38	5
8	Apartment / Unit / Flat	2020	12761.59	6
9	Penthouse	2020	12071.43	7
10	New Apartments / Off the Plan	2020	11040	8
11	Semi-Detached	2020	10756.07	9
12	Studio	2020	7609.43	10

C 4

Business Report

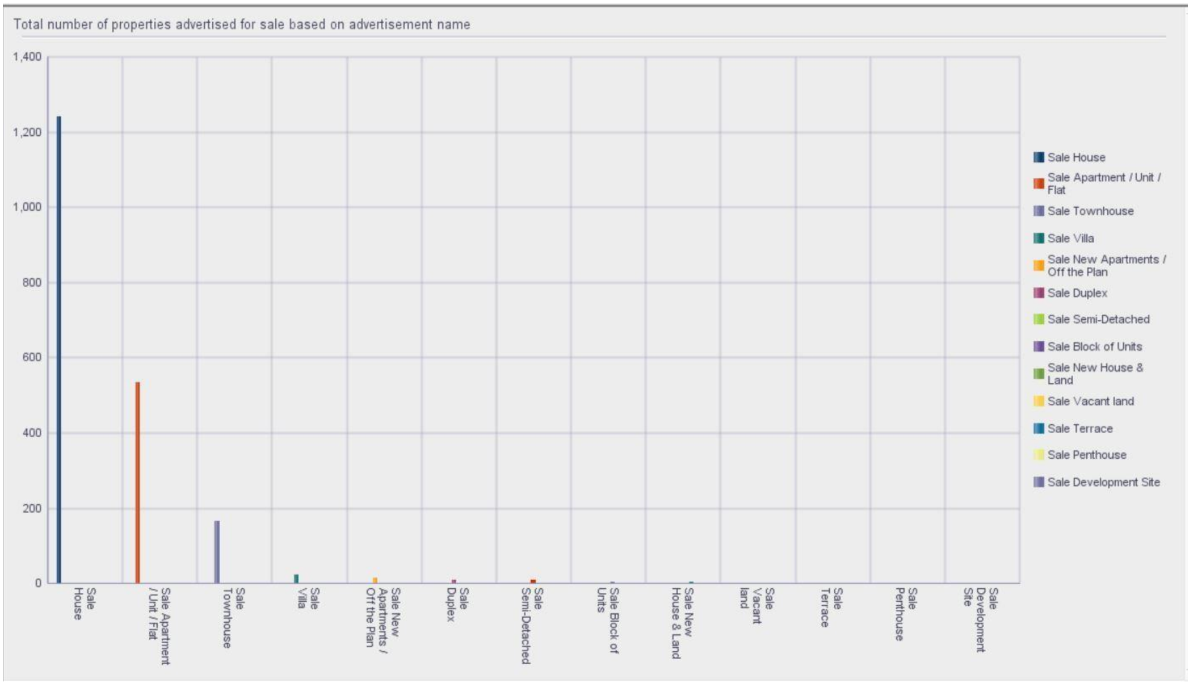
Report 1

TOP 10 MALE AGENTS IN VICTORIA



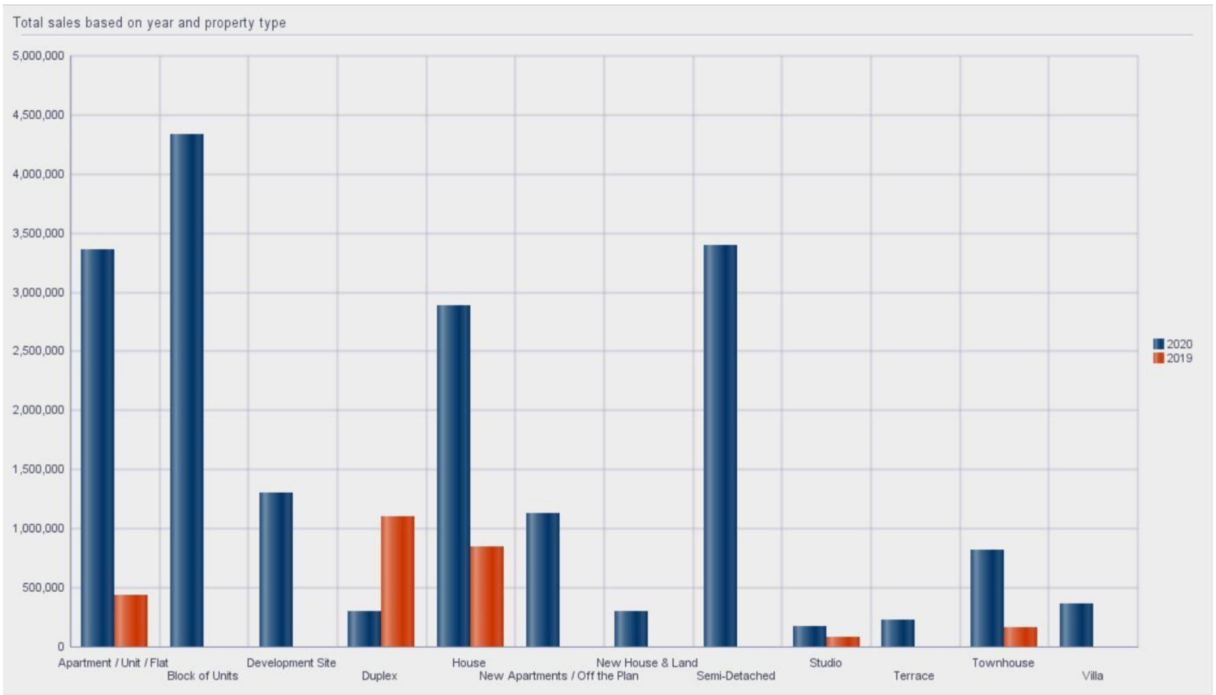
REPORT 3

ADVERTISEMENT NAME AND THE TOTAL NUMBER OF PROPERTIES ADVERTISED UNDER IT



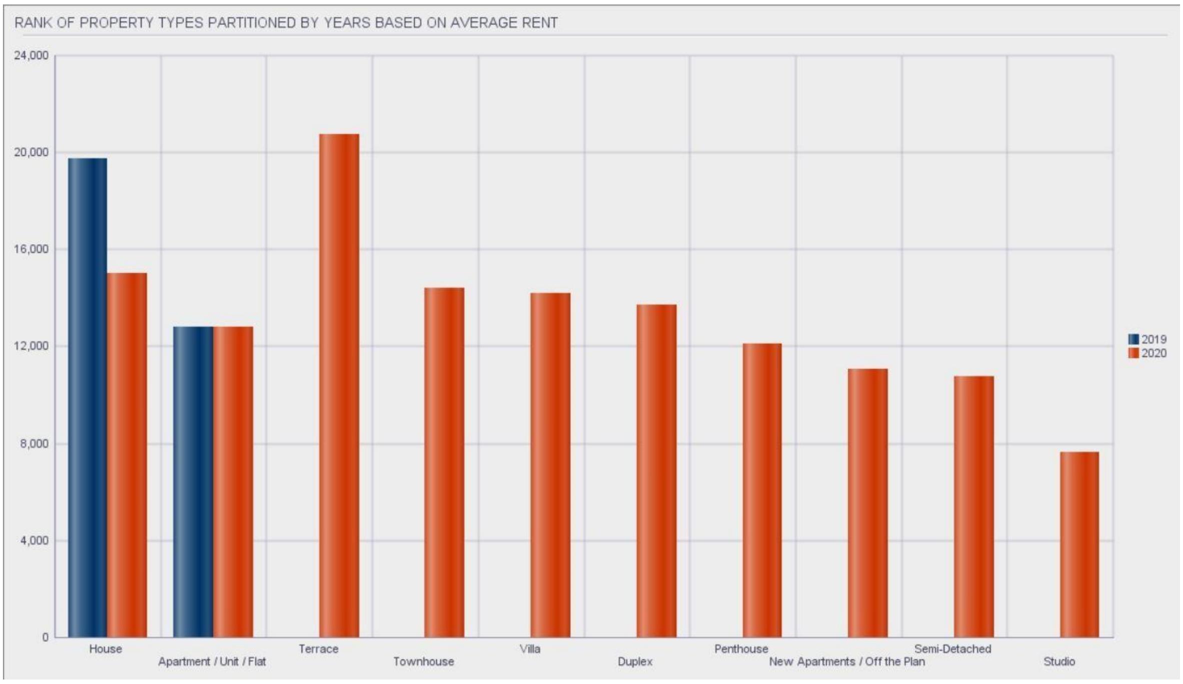
Report 6

Sub total and Total Sales for each property type in different year



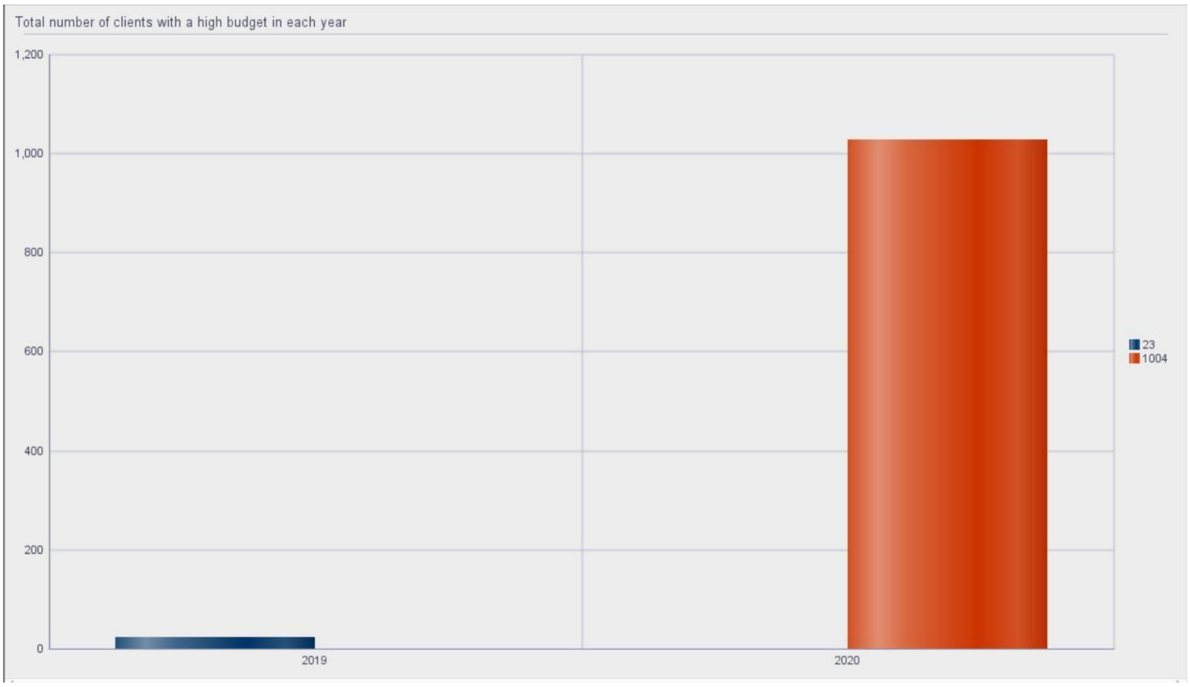
Report 12

RANK OF PROPERTY TYPES BASED ON AVERAGE RENT PARTITIONED BY YEAR



Report 8

Total number of clients and cumulative number of clients with a high budget in each year?


Total number of clients with a high budget in each year