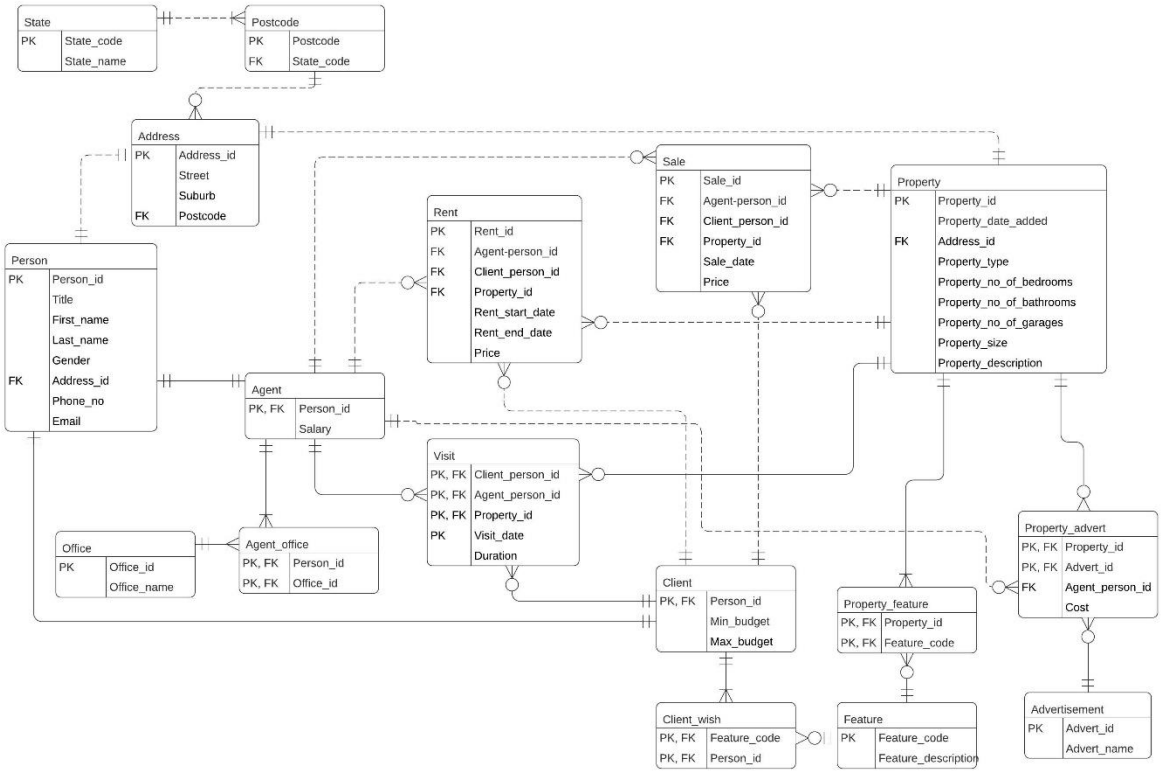
Table of Contents

Task C.1

Output a – E/R Diagram



Output b – Data Cleaning

b1. Address

```
SELECT COUNT(*)
     FROM mre_address; -- 13,204 rows
```

COUNT(*)
13204

Firstly, checked for duplicate address_id

```
SELECT COUNT(DISTINCT(address_id))
     FROM mre_address; -- 13,204 rows
```

Secondly, checked for distinct street

```
SELECT COUNT(DISTINCT(street))
     FROM mre_address; -- 13,156 rows
```

However, among the street records, it was found that a number of records contain the same street data but different suburbs, as such it was further evaluated based on distinct street, suburb and postcode combinations.

Using the following formula, there were no addresses that were not used in the Property and Person tables.

```
SELECT address_id
    FROM mre_address
          WHERE NOT address_id IN (SELECT address_id FROM
```

```
        mre_property)
        AND NOT address_id IN (SELECT address_id FROM
        mre_person);
```

As such, we derived that there are no records that are required to be deleted from the address table.

b2. Advertisement

```
SELECT COUNT(*) FROM mre_advertisement; -- 25 rows
```

COUNT(*)
25

Since that there were only 25 rows, visual inspection showed that there were no null records and the following two confirmed that there were no duplicate records:

```
SELECT COUNT(DISTINCT(advert_id))
     FROM mre_advertisement; -- 25 rows
SELECT COUNT(DISTINCT(advert_name))
     FROM mre_advertisement; -- 25 rows
```

We concluded that there needed no cleaning for the advertisement table.

b3. Agent

```
SELECT COUNT(*)
     FROM MRE_Agent; -- 2,469 rows
```

COUNT(*)
2469

When using the following query, we found that only 2,468 people are agents, meaning that one agent is non-existent

```
SELECT COUNT(*)
     FROM (SELECT * FROM MRE_Person p, MRE_Agent a
          WHERE p.person_id = a.person_id); -- 2,468 rows
SELECT *
     FROM mre_person
          WHERE NOT person_id IN (SELECT person_id FROM
          mre_person);
```

	PERSON_ID	SALARY
1	6997	0

Using the following syntax, that row was deleted:

```
DELETE FROM MRE_Agent
     WHERE NOT person_id IN (SELECT person_id FROM MRE_person);
     -- 1 row deleted
```

Subsequently, another error was found in 2 rows where salary was less than 0

```
SELECT * FROM mre_agent
     WHERE salary < 0; -- 2 rows
```

PERSON_ID	SALARY
6844	-100000
6000	-120000

```
DELETE FROM mre_agent
     WHERE salary < 0; -- 2 rows deleted
```

Checking the number of rows again:

```
SELECT COUNT(*)
     FROM MRE_Agent; -- 2,466 rows
```

COUNT(*)
2466

b4. Agent_Office

```
SELECT COUNT(*)
     FROM mre_agent_office; -- 2,529 rows
```

COUNT(*)
2529

While the following syntax shows that there are multiple records for the same agent, since an agent can work at multiple office, there is no error

```
SELECT COUNT(DISTINCT(person_id))
     FROM mre_agent_office; -- 2,467 rows
```

However, checking whether the agent exists, the agent deleted earlier was not found and was subsequently removed.

```
SELECT *
     FROM mre_agent_office
          WHERE NOT person_id IN (SELECT person_id FROM
mre_agent);
```

	PERSON_ID	OFFICE_ID
1	6997	1177

```
DELETE FROM mre_agent_office
     WHERE NOT person_id IN (SELECT person_id FROM mre_agent);
-- 1 row deleted
```

The number of rows in the end is

```
SELECT COUNT(*)
     FROM mre_agent_office; -- 2,528 rows
```

	COUNT(*)
1	2528

b5. Client

```
SELECT COUNT(*)
     FROM mre_client; -- 3,339 rows
```

COUNT(*)
3339

However, from the following syntax, there is a client that is not registered as a person

```
SELECT COUNT(*)
     FROM mre_person p, mre_client c
          WHERE p.person_id = c.person_id; -- 3,338 rows

SELECY *
     FROM mre_client
          WHERE NOT person_id IN (SELECT person_id FROM
mre_person);
```

	PERSON_ID	MIN_BUDGET	MAX_BUDGET
1	7000	8500	15050

As such, the extra client was deleted with

```
DELETE FROM mre_client
     WHERE NOT person_id IN (SELECT person_id FROM mre_person);
     -- 1 row deleted
```

Furthermore, it was found that few records had the max_budget lower than the min_budget, as well as min budget being negative. These rows were then deleted

```
SELECT *
     FROM mre_client
          WHERE max_budget < min_budget; -- 3 rows
```

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

```
DELETE FROM mre_client
     WHERE max_budget < min_budget; -- 3 rows deleted
```

Checking for any min_budget or max_budget being negative yield no results.

Checking the number of rows again:

```
SELECT COUNT(*)
     FROM mre_client; -- 3,338 rows
```

COUNT(*)
3335

b6. Client_Wish

```
SELECT COUNT(*)
     FROM mre_client_wish; -- 1,204 rows
```

COUNT(*)
1204

Since a single client can have many wishes, checking for distinct clients is useless. Checking for repeated person_id and feature code also proved no duplicates. As such, no changes were needed.

b7. Feature

```
SELECT COUNT(*)
     FROM mre_feature; -- 726 rows
```

COUNT(*)
726

Checking for duplicate feature_id and feature_description showed no duplicates.

b8. Office

```
SELECT COUNT(*)
     FROM mre_office; -- 1,177 rows
```

COUNT(*)
1177

Checking for duplicates of office_id and office_name showed no duplicates.

b9. Person

```
SELECT COUNT(*)
     FROM mre_person; -- 7,000 rows
```

COUNT(*)
7000

Looking for duplicates person_id, 4 deplicates were found for person_id = 6995

```
SELECT COUNT(DISTINCT(person_id))
     FROM MRE_Person; -- 6,997 rows
SELECT person_id
     FROM mre_person GROUP BY person_id HAVING COUNT(*) > 1;
```

PERSON_ID
6995

Using the following syntax, the duplicate records were deleted:

```
DELETE FROM MRE_Person p
     WHERE rowid > (SELECT MIN(rowid)FROM MRE_Person p2
          WHERE p.person_id = p2.person_id); -- 3 rows deleted
```

Checking through phone_no and email showed no other duplicates, the final number is:

```
SELECT COUNT(*)
     FROM mre_person; - 6,997 rows
```

COUNT(*)
6997

b10. Postcode

```
SELECT COUNT(*)
     FROM mre_postcode; -- 689 rows
```

COUNT(*)
689

No duplicate or nulls were found in the postcode table.

b11. Property

```
SELECT COUNT(*)
     FROM mre_property; -- 6,226 rows
```

COUNT(*)
6226

There were a large number of records that were duplicated

```
SELECT COUNT(DISTINCT(property_id)) FROM mre_property; -- 6,208
rows
SELECT *
    FROM mre_property p
        WHERE rowid > (SELECT MIN(rowid)FROM mre_property p2
          WHERE p.property_id = p2.property_id);
```

	PROPERTY_ID	PROPERTY_DATE_ADDED	ADDRESS_ID	PROPERTY_TYPE	PROPERTY_NO_OF_BEDROOMS	PROPERTY_NO_OF_BATHROOMS	PROPERTY_NO_OF_GARAGES	PROPERTY_SIZE	PROPERTY_DESCRIPTION
1	6179	19-JAN-20	6179	Apartment / Unit / Flat	1	1	1	(null)	Designer first flc
2	6179	19-JAN-20	6179	Apartment / Unit / Flat	1	1	1	(null)	Designer first flc
3	6177	25-NOV-19	6177	Apartment / Unit / Flat	2	2	2	(null)	DUE TO FEDERAL GOV
4	6177	25-NOV-19	6177	Apartment / Unit / Flat	2	2	2	(null)	DUE TO FEDERAL GOV
5	6177	25-NOV-19	6177	Apartment / Unit / Flat	2	2	2	(null)	DUE TO FEDERAL GOV
6	6179	19-JAN-20	6179	Apartment / Unit / Flat	1	1	1	(null)	Designer first flc
7	6179	19-JAN-20	6179	Apartment / Unit / Flat	1	1	1	(null)	Designer first flc
8	6179	19-JAN-20	6179	Apartment / Unit / Flat	1	1	1	(null)	Designer first flc
9	6179	19-JAN-20	6179	Apartment / Unit / Flat	1	1	1	(null)	Designer first flc
10	6179	19-JAN-20	6179	Apartment / Unit / Flat	1	1	1	(null)	Designer first flc
11	6179	19-JAN-20	6179	Apartment / Unit / Flat	1	1	1	(null)	Designer first flc
12	6179	19-JAN-20	6179	Apartment / Unit / Flat	1	1	1	(null)	Designer first flc
13	6179	19-JAN-20	6179	Apartment / Unit / Flat	1	1	1	(null)	Designer first flc
14	6179	19-JAN-20	6179	Apartment / Unit / Flat	1	1	1	(null)	Designer first flc
15	6179	19-JAN-20	6179	Apartment / Unit / Flat	1	1	1	(null)	Designer first flc

The following syntax was used to delete the duplicate rows

```
DELETE FROM MRE_Property p
     WHERE rowid > (SELECT MIN(rowid)FROM MRE_Property p2
          WHERE p.property_id = p2.property_id); -- 18 rows
deleted
```

The new number of rows is:

```
SELECT COUNT(*)
     FROM mre_property; -- 6,208 rows
```

COUNT(*)
6208

b12. Property_Advert

```
SELECT COUNT(*)
     FROM mre_property_advert; -- 3,646 rows
```

COUNT(*)
3646

No duplicate records or null records were found in this table

b13. Property_Feature

```
SELECT COUNT(*)
     FROM mre_property_feature; -- 30,373 rows
```

COUNT(*)
30373

No duplicate or null records were found

b14. Rent

```
SELECT COUNT(*)
     FROM mre_rent; --3,284 rows
```

COUNT(*)
3284

Records were checked for duplicate and non were found but one record was found for rent_end_date before rent_start_date

```
SELECT *
     FROM mre_rent
          WHERE rent_end_date <= rent_start_date;
```

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

The record was then deleted

```
DELETE FROM MRE_Rent WHERE rent_id IN (SELECT rent_id FROM MRE_Rent WHERE rent_end_date < rent_start_date); -- 1 row deleted
SELECT COUNT(*)
     FROM MRE_Rent; -- 3,283 rows
```

COUNT(*)
3283

b15. Sale

```
SELECT COUNT(*)
     FROM mre_sale; -- 2,925 rows
```

COUNT(*)
2925

The following syntax was used and found that a number of records had null client_person_id and sale_date. However, as these rows may be kept for record to show agent client relationships for unsold properties, the rows are temporarily kept.

b16. State

```
SELECT *
     FROM mre_state; -- 9 rows
```

There were little number of rows so visual inspection was possible and one row was identified as NULL state_code and UNKNOWN state_name which was promptly deleted

```
DELETE FROM MRE_State
     WHERE state_code IS NULL; -- 1 row deleted
```

New number of records is:

```
SELECT COUNT(*)
     FROM mre_state; -- 8 rows
```

COUNT(*)
8

b17. Visit

```
SELECT COUNT(*)
     FROM mre_visit; -- 575 rows
```

COUNT(*)
575

A record was found that the agent or client did not exist in mre_agent or mre_client

```
SELECT *
     FROM mre_visit
          WHERE NOT agent_person_id IN (SELECT person_ID FROM
          mre_agent)
          OR NOT client_person_id IN (SELECT person_id FROM
          mre_client); -- 1 row
```

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

That record was promptly deleted

```
DELETE
     FROM mre_visit
          WHERE NOT agent_person_id IN (SELECT person_ID FROM
          mre_agent)
          OR NOT client_person_id IN (SELECT person_id FROM
          mre_client); -- 1 row deleted
```

Updated number of rows is:

```
SELECT COUNT(*) FROM mre_visit; -- 574 rows
```

COUNT(*)
574

b18. Special Case

We checked for person records that did not reference any addresses. Using the following code, we found that there was a person record which address_id did not exist and the record contains majority of null fields. This record could not be detected through using 'IS NULL' because the fields while displaying null, was actually a string 'null'.

```
SELECT *
     FROM mre_person
WHERE NOT address_id IN (SELECT address_id FROM mre_address);
```

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

As a safety precaution, we checked through Agent, Client, Client_Wish to this person_id.

We found no such persin in the Agent table but there exists a normal record in the Client table.

PERSON_ID	MIN_BUDGET	MAX_BUDGET
7001	7500	11250

We continued to check the Client_Wish table and found one record of this person:

FEATURE_CODE	PERSON_ID
726	7001

Looking into the Feature table, we found that feature_code = 726 is labelled as Fake Feature.

FEATURE_CODE	FEATURE_DESCRIPTION
726	Fake Feature

After taking the above-mentioned factors into consideration, we decided to delete person_id = 7001 from the Person, Client and Client_Wish table as well as feature_code = 726 from the feature table.

b19. Summary

After analysing the operational database, the following errors were identified:

1. Agent table person_id NOT IN Person table person_id – 1 row deleted
2. Agent salary < 0 – 2 rows deleted
3. Agent person_id NOT IN Person table person_id – 1 row deleted
4. Client table person_id NOT IN Person table person_id – 1 row deleted
5. Client max_budget < min_budget – 3 rows deleted
6. Duplicate person record – 3 rows deleted
7. Duplicate property record – 18 rows deleted
8. Rent rent_end_date < rent_start_date – 1 row deleted
9. State state_code IS NULL – 1 row deleted
10. Visit table agent_person_id or client_person_id NOT IN Person table person_id – 1 row deleted
11. Special Case: Person address_id non-existent – 1 row deleted

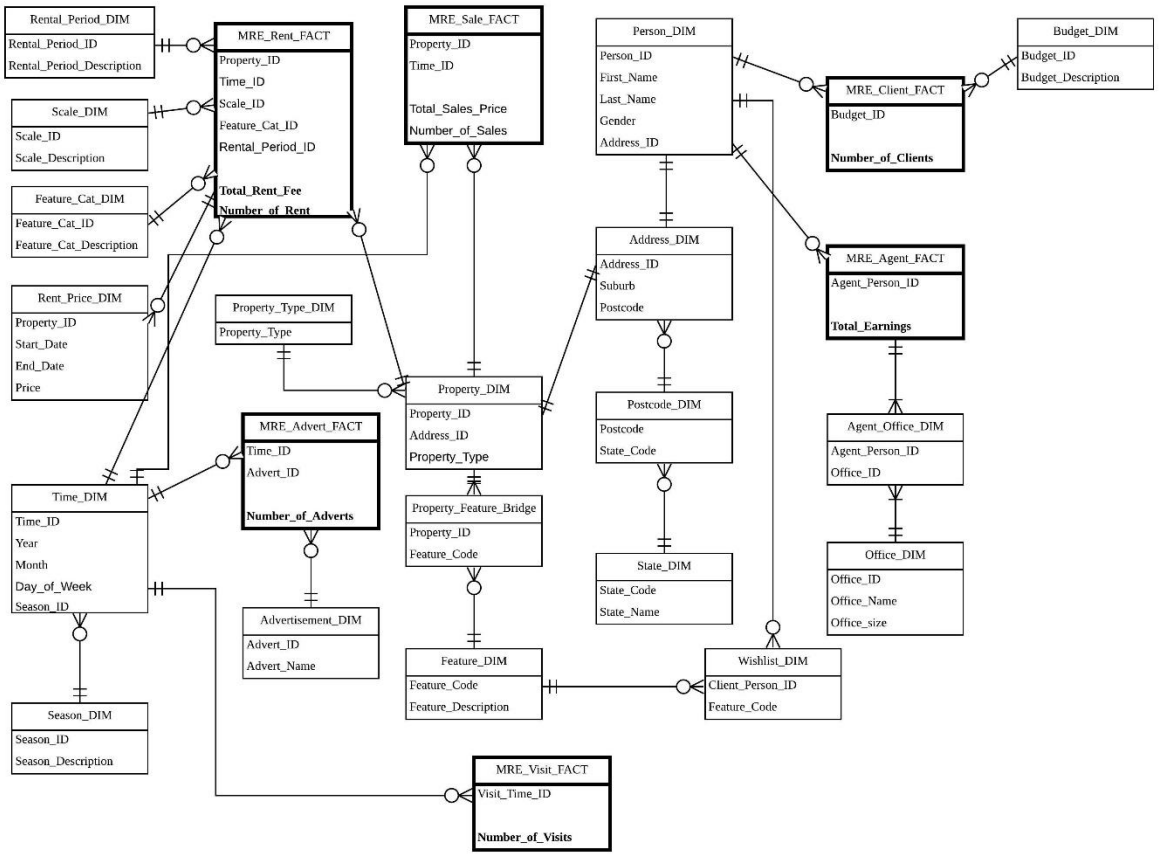
Client record referencing above – 1 row deleted

Client_wish record referencing above – 1 row deleted

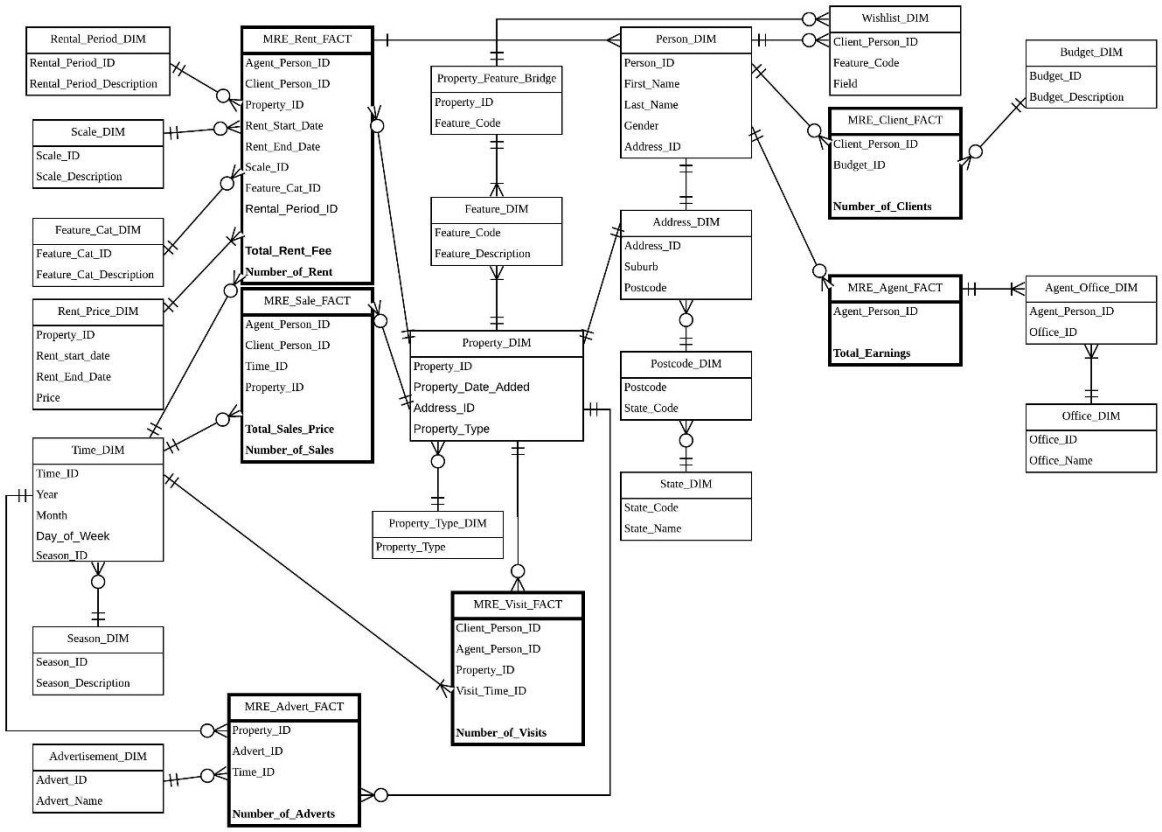
Feature record ‘Fake Feature’ – 1 row deleted

Output c – Star Schemas

c1 Level 2



c2 Level 0



Output d – Hierarchy or non-Hierarchy Explanation

We used hierarchy for address as it allows easy drill down into data for viewing different granularity. It is preferred compared to non-hierarchy as it allows optional usage of the lower level dimensions when needed but at the same time reduce the storage capacity in higher level tables as state names is longer than using a number to represent them.

Output e - Temporal Dimension SCD Type Explanation

SCD Type 4 was chosen as rental price may not necessarily be fixed for each rental agreement of the same property over time. Therefore, MRE_Rent_FACT does not need to keep the price information, but the property_id, rent_start_date and rent_end_date. The rent prices are kept separately in the Rent_Price_DIM, where the price can be seen for each rental agreement made with clients (after joining with property_id, rent_start_date, rent_end_date).

SCD Type 0 only stores the initial value of the rent price, which is not suitable for the fact measure. SCD Type 1 stores the latest rent price, but not the values before and therefore, it is not possible to view the change in rental prices over time.

SCD Type 2 was initially considered, but it had several redundancies such as adding sequence number to the property_id for each transaction, and the current flag to show whether it is the latest record. While SCD Type 3 has a unique property_id with no number sequencing, it uses current price and previous price to store the values. This is not suitable for storing rental prices

as the property may not be always out on rent, as there might be vacancies for some periods of time.

SCD Type 0 is suitable for storing property sale prices as once a property sale has been made, the original price value is recorded. There will be no further changes to the property price.

Output f – Differences between the Two Versions of Star/Snowflake Schema

Level 2 star/snowflake schema is the highest aggregation and level 0 star/snowflake schema has no aggregation. In level 2 MRE_Sale_Fact has 2 dimensions of Property_DIM and Time_DIM. In level 0, MRE_Sale_Fact dimension was further broken down to Property_DIM, Time_DIM, PropertyType_DIM and Person_DIM.

Level 2 – MRE_Sale_Fact

	PROPERTY_ID	TIME_ID	TOTAL_SALES_PRICE	NUMBER_OF_SALES
1	24	202001WED	769000	1
2	31	202002SUN	680000	1
3	241	202001SAT	390000	1
4	121	202001SUN	495000	1
5	324	202002FRI	550000	1
6	201	202002SUN	239000	1
7	305	202002TUE	440000	1
8	364	202004SUN	575000	1
9	300	202003MON	425000	1
10	459	202002WED	260000	1
11	626	201912MON	349950	1
12	631	202001WED	1025000	1

Looking at the data retrieved from Level 2 sale fact table, the management will only able to analyse the sales price and the number of times the property had been resold. However if the management were to drill down for further information regarding how well their agents are performing in term of sale, or some client insight this level 2 sale fact will no provide such data.

Level 0 – MRE_Sale_Fact

	AGENT_PERSON_ID	CLIENT_PERSON...	TIME_ID	PROPERTY_ID	TOTAL_SALES_PRICE	NUMBER_OF_SALES
1	1	2467	202003THU	50	549000	1
2	1	2468	202001SAT	92	639000	1
3	2	2469	202001SUN	1	650000	1
4	2	2470	202001TUE	19	895000	1
5	2	2471	202003WED	41	580000	1
6	2	2472	202001SUN	81	825000	1
7	6	2473	202001THU	129	249000	1
8	7	2474	202002SUN	260	439000	1
9	9	2475	202001TUE	162	340000	1
10	11	2476	202001FRI	208	1675000	1
11	11	2477	202001TUE	307	1034000	1
12	15	2478	202002SUN	179	520000	1

From Level 0 sale fact table, the management would able to know how well their agent performed. In this snapshot of data retrieved from level 0 sale fact table, agent ID 2 is performing well in selling the property. Therefore, this shows that the level of aggregation is low in level 0 MRE_Sale_Fact, where the fact was broken down to more detail.

For rent fact table, Level 2 has a higher aggregation compare to level 0. This means that the management will more likely to get more information from level 0 rent fact table.

Level 2 – MRE_Rent_Fact

	PROPERTY...	TIME_ID	SCALE_ID	FEATURE_CAT_ID	TOTAL_RENT_FEE	NUMBER_OF_RENT
1	2945	202001TUE	2	1	8640	1
2	2954	202004TUE	1	1	8932.857142857142857142857142857142857143	1
3	2955	202003FRI	3	1	9480	1
4	2957	202003MON	3	2	26279.999999999999999999999999999999999999	1
5	2963	202004THU	1	1	8040	1
6	2965	202005FRI	2	1	8450	1
7	2967	202004TUE	2	1	7242.857142857142857142857142857142857143	1
8	2971	202001WED	2	2	11640	1
9	2974	202004THU	2	2	14400	1
10	2975	202003TUE	1	1	10320	1
11	2983	202003SUN	2	1	8400	1
12	2986	202003WED	2	1	7679.999999999999999999999999999999999999	1

Looking at the data retrieved from Level 2 rent fact table, which consist of property id, time id, scale id, feature category id, total rent fee and number of rent. This fact table only provided limited information for the management. This table was aggregated with agent, client, rent start date, rent end date rental and rental period. For instance, looking at property id ‘ 2945’ there is only 1 time id ‘202001TUE’ in Level 2 star schema.

Level 0 – MRE_Rent_Fact

	AGENT_PERSON_ID	CLIENT_PERSON_ID	PROPERTY...	RENT_START_DATE	RENT_END_DATE	RENTAL_PERIOD_ID	SCALE_ID	FEATURE_CAT_ID	TOTAL_RENT_FEE	NUMBER_OF_RENT
1	730	3830	2945	202001TUE	202007TUE	1	2	1	8640	1

In Level 0 star schema rent fact, the time id was broken down into rent start date and rent end date. Agent and client was added into the table as well. The table records more details compare to level 2 star schema.

In term of client factor, Level 2 star schema client fact table only contain 3 rows of data with budget ID and total number of client. Looking at the data, budget ID of 1 which is low budget, consist of 1581 clients.

level 2 – MRE_Client _Fact

	BUDGET_ID	TOTAL_NUMBER_OF_CLIENT
1	1	1581
2	2	466
3	3	1287

However, in Level 0 star schema client fact table, budget ID 1 can be brake down into 1581 rows of data, which consists of a client ID. Level 0 star schema shows the budget category for each of every client.

Level 0 – MRE_Client fact

	CLIENT_PERSON_ID	BUDGET...	NUMBER_OF_CLIE...
1	5498	1	1
2	5501	1	1
3	5515	1	1
4	5524	1	1
5	5605	1	1
6	5630	1	1
7	5764	1	1
8	5790	1	1
9	3984	1	1
10	4001	1	1
11	4030	1	1
12	3996	1	1

In term of visit fact, Level 2 visit fact table contain visit time id and number of visits. To the management, this table only provide the number of visits in each visit time period. This table was constructed by aggregating the client, agent and property.

Level 2 – MRE_Visit_Fact

	VISIT_TIME_ID	NUMBER_OF_VISIT
1	202003THU	58
2	202004TUE	19
3	202004SAT	30
4	202004FRI	11
5	202003SUN	50
6	202004MON	28
7	202004THU	12
8	202003TUE	64
9	202004WED	21
10	202003FRI	64
11	202003MON	62
12	202003WED	55
13	202004SUN	23
14	202003SAT	77

In Level 0, the fact table brake down the aggregation back to client, agent and property. In level 2 visit fact table, visit time id '202003THU' has 58 number of visits. However, in level 0 visit fact table, time ID of '202003THU' was broken down based on client, agent and property. Therefore, for each different client, will have same agent id and property. The total of number visits will add up to 58 for the time ID '202003THU' in Level 0 visit fact table.

Level 0 – MRE_Visit Fact

	CLIENT_PERSON_ID	AGENT_PERSON_ID	PROPERTY_ID	TIME_ID	NUMBER_OF_VISITS
1	5617	1775	5857	202003THU	1
2	5037	161	1521	202003THU	1
3	5071	161	1521	202003THU	1
4	5044	797	1576	202003THU	1
5	5079	1689	1530	202003THU	1
6	5044	2243	1366	202003THU	1
7	5043	2243	1628	202003THU	1
8	5637	1155	5562	202003THU	1
9	5592	1775	5857	202003THU	1
10	5198	779	1756	202003THU	1
11	5257	788	2133	202003THU	1
12	5060	2242	1362	202003THU	1

For the advertisement fact, level 2 star schema shows that the time ID ‘202004WED’, advertisement ID ‘25’ has 4 advertisements. This fact table only show the number of advertisement as general, but the total number was aggregated with different property.

Level 2 – MRE_advert_fact

	TIME_ID	ADVERT...	NUMBER_OF_ADVERTS
1	202004WED	25	4
2	202004TUE	25	3
3	202004MON	25	2
4	202004SAT	25	3
5	202003THU	25	1
6	202003SUN	25	2
7	202004THU	25	1
8	202004SUN	25	3
9	202003SAT	25	1
10	202003FRI	25	2
11	202003TUE	25	1

In Level 0, for time ID ‘202004WED’ the data was broken down into 4 different rows with different property ID. This indicates that for the time id ‘202004WED’ the advertisement was published for 4 different property at the same time. Therefore, Level 0 advertisement fact table has no aggregation compared to level 2 advertisement fact of high aggregation.

Level 0 – MRE_advert_fact

	PROPERTY_ID	ADVERT_ID	TIME_ID	NUMBER_OF_ADVERTS
1	1266	25	202004WED	1
2	1323	25	202004WED	1
3	1125	25	202004WED	1
4	1164	25	202004WED	1

Task 2

Output a – SQL Statement for Level 2 Star Schema

```
-- Task c 2b)
-- Level 2 multi-fact star schema
DROP TABLE MRE_scale_DIM_l2 PURGE;
DROP TABLE MRE_feature_cat_DIM_l2 PURGE;
DROP TABLE MRE_property_dim_l2 PURGE;
DROP TABLE MRE_property_feature_bridge_l2 PURGE;
DROP TABLE MRE_feature_dim_l2 PURGE;
DROP TABLE MRE_property_type_dim_l2 PURGE;
DROP TABLE MRE_address_dim_l2 PURGE;
DROP TABLE MRE_postcode_dim_l2 PURGE;
DROP TABLE MRE_state_dim_l2 PURGE;
DROP TABLE MRE_advertisement_dim_l2 PURGE;
DROP TABLE MRE_person_dim_l2 PURGE;
DROP TABLE MRE_agent_office_dim_l2 PURGE;
DROP TABLE MRE_office_dim_l2 PURGE;
DROP TABLE MRE_budget_dim_l2 PURGE;
DROP TABLE MRE_rental_period_dim_l2 PURGE;
DROP TABLE MRE_wishlist_dim_l2 PURGE;
DROP TABLE MRE_rent_price_dim_l2 PURGE;
DROP TABLE MRE_temp_time_dim_l2 PURGE;
DROP TABLE MRE_time_dim_l2 PURGE;
DROP TABLE MRE_season_dim_l2 PURGE;
DROP TABLE MRE_agent_fact_l2 PURGE;
DROP TABLE MRE_temp_client_l2 PURGE;
DROP TABLE MRE_client_fact_l2 PURGE;
DROP TABLE MRE_temp_rent_fact_l2 PURGE;
DROP TABLE MRE_rent_fact_l2 PURGE;
DROP TABLE MRE_temp_visit_l2 PURGE;
DROP TABLE MRE_visit_fact_l2 PURGE;
DROP TABLE MRE_temp_sale_fact_l2 PURGE;
DROP TABLE MRE_sale_fact_l2 PURGE;
DROP TABLE MRE_temp_advert_l2 PURGE;
DROP TABLE MRE_advert_fact_l2 PURGE;

-- Dimension tables
-- Scale dimension
create table mre_scale_dim_l2 (
    scale_id numeric(1),
    scale_description char(20));

insert into mre_scale_dim_l2 values(1, 'extra small');
insert into mre_scale_dim_l2 values(2, 'small');
insert into mre_scale_dim_l2 values(3, 'medium');
insert into mre_scale_dim_l2 values(4, 'large');
insert into mre_scale_dim_l2 values(5, 'extra large');

-- Feature catagory dimension
create table mre_feature_cat_dim_l2(
```

```
    feature_cat_id numeric(1),
    feature_cat_description char(15));

insert into mre_feature_cat_dim_l2 values(1,'basic');
insert into mre_feature_cat_dim_l2 values(2,'standard');
insert into mre_feature_cat_dim_l2 values(3,'luxurious');


-- Property dimension
create table mre_property_dim_l2
    as select property_id, address_id, property_type
        from mre_property;

-- Property feature bridge
create table mre_property_feature_bridge_l2
    as select distinct *
        from mre_property_feature;


-- feature dim
create table mre_feature_dim_l2
    as select distinct *
        from mre_feature;

-- property type dimension
create table mre_property_type_dim_l2
    as select distinct(property_type)
        from mre_property;

-- Address dim
create table mre_address_dim_l2
    as select distinct address_id, suburb, postcode
        from mre_address;

-- postcode dim
create table mre_postcode_dim_l2
    as select distinct *
        from mre_postcode;

-- state dim
create table mre_state_dim_l2
    as select *
        from mre_state;

-- Advertisment dim
create table mre_advertisement_dim_l2
    as select distinct *
        from mre_advertisement;

-- person dim
create table mre_person_dim_l2
```

```
    as select person_id, first_name, last_name, gender,
address_id
        from mre_person;

-- agent office dim
create table mre_agent_office_dim_l2
    as select distinct person_id as agent_person_id, office_id
        from mre_agent_office;

-- office dim
create table mre_office_dim_l2
    as select *
        from mre_office;

alter table mre_office_dim_l2
    add office_size char(10);

update mre_office_dim_l2 t
    set office_size =
        (select case
                    when count(person_id) < 4 then 'small'
                    when count(person_id) between 4 and 12 then
'medium'
                    else 'big'
                end
                from mre_agent_office ao
                where t.office_id = ao.office_id);

-- Budget dimension
create table mre_budget_dim_l2(
    budget_id numeric(1),
    budget_description varchar(100));

insert into mre_budget_dim_l2 values (1, 'Budget between 0 and
1000');
insert into mre_budget_dim_l2 values (2, 'Budget between 1001
and 100000');
insert into mre_budget_dim_l2 values (3, 'Budget more than
100001');

-- Rental period DIM
create table mre_rental_period_dim_l2(
    rental_period_id numeric(2),
    rental_period_description varchar(50));

insert into mre_rental_period_dim_l2 values (1, 'short');
insert into mre_rental_period_dim_l2 values (2, 'medium');
insert into mre_rental_period_dim_l2 values (3, 'long');

-- whishlist dim
create table mre_wishlist_dim_l2
```

```
    as select distinct *
        from mre_client_wish;

-- Rent price dimension
create table mre_rent_price_dim_l2
    as  select  property_id,  rent_start_date,  rent_end_date,
price
        from mre_rent;

-- Time dimension
create table mre_temp_time_dim_l2
    as select *
        from (select distinct sale_date as dates from mre_sale
                    where sale_date is not null
                union
                select distinct rent_start_date from mre_rent
                    where rent_start_date is not null
                union
                select distinct rent_end_date from mre_rent
                    where rent_end_date is not null
                );

alter table mre_temp_time_dim_l2
    add (
        time_id varchar(20),
        Year numeric(4),
        Month numeric(2),
        Season_id numeric(1));

update mre_temp_time_dim_l2
    set time_id = to_char(dates, 'YYYYMMDY'),
        year = to_char(dates, 'YYYY'),
        month = to_char(dates, 'MM');

update mre_temp_time_dim_l2
    set season_id =
        case
            when month between 3 and 5 then 1
            when month between 6 and 8 then 2
            when month between 9 and 11 then 3
            else 4
        end;

create table mre_time_dim_l2
    as select DISTINCT(time_id), year, month, season_id
        from mre_temp_time_dim_l2;

-- Season DIM
create table mre_season_dim_l2(
    season_id numeric(1),
    season_description char(10));
```

```
insert into mre_season_dim_l2 values(1, 'Spring');
insert into mre_season_dim_l2 values(2, 'Summer');
insert into mre_season_dim_l2 values(3, 'Autumn');
insert into mre_season_dim_l2 values(4, 'Winter');

-- Fact tables
-- Agent fact table
create table mre_agent_fact_l2
as select a.person_id as agent_person_id, sum(nvl(s.price, 0))
+       nvl(sum(nvl(r.price,       0)/7*(r.rent_end_date       -
r.rent_start_date)), 0) as total_earnings
    from mre_agent a, mre_sale s, mre_rent r
        where a.person_id = s.agent_person_id (+)
        and a.person_id = r.agent_person_id (+)
            group by a.person_id;

-- client fact table
create table mre_temp_client_l2
    as select max_budget from mre_client;

alter table mre_temp_client_l2
    add budget_id numeric(1);

update mre_temp_client_l2
    set budget_id = case
        when max_budget between 0 and 1000 then 1
        when max_budget between 1001 and 100000 then 2
        else 3 end;

create table mre_client_fact_l2
    as select budget_id , count(*) as total_number_of_client
        from mre_temp_client_l2
            group by budget_id;

-- rent fact
create table mre_temp_rent_fact_l2
    as select distinct
        r.property_id ,
        r.rent_start_date as dates,
        p.property_no_of_bedrooms,
        COUNT(*) as number_of_features,
        r.price,
        r.rent_end_date,
        r.rent_start_date,
        count(distinct(rent_id)) as num_of_rent
            from     mre_rent     r,     mre_property     p,
mre_property_feature f
                where r.property_id = p.property_id
                    and p.property_id = f.property_id
                    and r.rent_start_date is not null
```

```
                        GROUP         BY         r.property_id,
p.property_no_of_bedrooms,      r.price,      r.rent_end_date,
r.rent_start_date;

alter table mre_temp_rent_fact_l2 add (
    time_id varchar(20),
    scale_id numeric(1),
    feature_cat_id numeric(1));

update mre_temp_rent_fact_l2
    set time_id = to_char(rent_start_date, 'YYYYMMDY'),
        scale_id =
            case
                when property_no_of_bedrooms between 0 and 1
then 1
                when property_no_of_bedrooms between 2 and 3
then 2
                when property_no_of_bedrooms between 4 and 6
then 3
                when property_no_of_bedrooms between 7 and 10
then 4
                else 5
            end;

update mre_temp_rent_fact_l2 t
       set feature_cat_id =
        (case when number_of_features < 10 then 1
                    when number_of_features between 10 and 20
then 2
                    else 3
                end);

create table mre_rent_fact_l2
    as select property_id, time_id, scale_id, feature_cat_id,
(price / 7 *(rent_end_date - rent_start_date)) as total_rent_fee,
num_of_rent as number_of_rent
        from mre_temp_rent_fact_l2;

-- visit fact
create table mre_temp_visit_l2
    as select visit_date
        from mre_visit;

alter table mre_temp_visit_l2
    add visit_time_id varchar(20);

update mre_temp_visit_l2
    set visit_time_id = to_char(visit_date, 'YYYYMMDY');

create table mre_visit_fact_l2
    as select visit_time_id, count(*) as number_of_visit
```

```
        from mre_temp_visit_l2
            group by visit_time_id;

-- sale fact
create table mre_temp_sale_fact_l2
    as  select  s.property_id,  s.sale_date,  p.property_type,
s.price
        from mre_sale s, mre_property p
            where s.property_id = p.property_id
                and sale_date is not null;

alter table mre_temp_sale_fact_l2 add (
    time_id varchar(20));

set define off;
update mre_temp_sale_fact_l2
    set time_id = to_char(sale_date, 'YYYYMMDY');

create table mre_sale_fact_l2
    as    select    property_id,    time_id,    sum(price)    as
total_sales_price, count(*) as number_of_sales
        from mre_temp_sale_fact_l2
            group by property_id, time_id;

-- Advert fact
create table mre_temp_advert_l2
    as select distinct a.advert_id, p.property_date_added
        from mre_property_advert a, mre_property p
            where p.property_id = a.property_id;

alter table mre_temp_advert_l2
    add time_id varchar(20);

update mre_temp_advert_l2
    set time_id = to_char(property_date_added, 'YYYYMMDY');

create table mre_advert_fact_l2
    as select time_id, advert_id, count(*) as number_of_adverts
        from mre_temp_advert_l2
            group by time_id, advert_id;

commit;
```

Output b – SQL Statement for Level 0 Star Schema

```
-- Task C 2b)
-- Level 0 multi-fact star schema
DROP TABLE MRE_Scale_DIM_l0 PURGE;
DROP TABLE MRE_Feature_Cat_DIM_l0 PURGE;
DROP TABLE MRE_Property_DIM_l0 PURGE;
DROP TABLE MRE_Property_Feature_Bridge_l0 PURGE;
DROP TABLE MRE_Feature_DIM_l0 PURGE;
DROP TABLE MRE_Wishlist_DIM_L0 PURGE;
DROP TABLE MRE_Property_Type_DIM_l0 PURGE;
DROP TABLE MRE_Address_DIM_l0 PURGE;
DROP TABLE MRE_Postcode_DIM_l0 PURGE;
DROP TABLE MRE_State_DIM_l0 PURGE;
DROP TABLE MRE_Advertisement_DIM_l0 PURGE;
DROP TABLE MRE_Person_DIM_l0 PURGE;
DROP TABLE MRE_Agent_Office_DIM_l0 PURGE;
DROP TABLE MRE_Office_DIM_l0 PURGE;
DROP TABLE MRE_Office_Size_DIM_L0 PURGE;
DROP TABLE MRE_Office_TempDIM_L0 PURGE;
DROP TABLE MRE_Budget_DIM_l0 PURGE;
DROP TABLE MRE_Rental_Period_DIM_l0 PURGE;
DROP TABLE MRE_Rent_Price_DIM_l0 PURGE;
DROP TABLE MRE_Season_DIM_l0 PURGE;
DROP TABLE MRE_Time_DIM_l0 PURGE;
DROP TABLE MRE_Sale_FACT_l0 PURGE;
DROP TABLE MRE_Rent_TempFACT_L0 PURGE;
DROP TABLE MRE_Rent_FACT_l0 PURGE;
DROP TABLE MRE_Client_TempFACT_L0 PURGE;
DROP TABLE MRE_Client_FACT_l0 PURGE;
DROP TABLE MRE_Agent_FACT_l0 PURGE;
DROP TABLE MRE_Visit_FACT_l0 PURGE;
DROP TABLE MRE_Advert_FACT_l0 PURGE;


--------------------------------
-- Implement dimension tables --
--------------------------------
-- MRE_Scale_DIM_L0
CREATE TABLE MRE_Scale_DIM_L0 (
    Scale_ID NUMBER,
    Scale_Description VARCHAR2(100)
);

INSERT INTO MRE_Scale_DIM_L0 VALUES (1, 'Extra small: <= 1
bedroom');
INSERT INTO MRE_Scale_DIM_L0 VALUES (2, 'Small: 2-3 bedrooms');
INSERT INTO MRE_Scale_DIM_L0 VALUES (3, 'Medium: 3-6 bedrooms');
INSERT INTO MRE_Scale_DIM_L0 VALUES (4, 'Large: 6-10 bedrooms');
INSERT INTO MRE_Scale_DIM_L0 VALUES (5, 'Extra large: > 10
bedrooms');
```

```
-- MRE_Feature_CAT_DIM_L0
CREATE TABLE MRE_Feature_CAT_DIM_L0 (
    Feature_CAT_ID NUMBER,
    Feature_CAT_Description VARCHAR2(100)
);

INSERT INTO MRE_Feature_CAT_DIM_L0 VALUES (1, 'Very basic: < 10
features');
INSERT INTO MRE_Feature_CAT_DIM_L0 VALUES (2, 'Standard: 10-20
features');
INSERT INTO MRE_Feature_CAT_DIM_L0 VALUES (3, 'Luxurious: > 20
features');

-- MRE_Property_DIM_L0
CREATE TABLE MRE_Property_DIM_L0 AS (
    SELECT
        p.Property_ID,
        p.Property_Date_Added,
        p.address_id,
        p.property_type
    FROM MRE_Property p
);

-- Property_Feature_Bridge_L0
CREATE TABLE MRE_Property_Feature_Bridge_L0 AS (
    SELECT DISTINCT * FROM MRE_Property_Feature
);

-- MRE_Feature_DIM_L0
CREATE TABLE MRE_Feature_DIM_L0 AS (
    SELECT DISTINCT * FROM MRE_Feature
);

-- MRE_Wishlist_DIM_L0
CREATE TABLE MRE_Wishlist_DIM_L0 AS (
    SELECT
        Person_ID AS Client_Person_ID,
        Feature_Code
    FROM MRE_Client_Wish
);

-- MRE_Property_Type_DIM_L0
CREATE TABLE MRE_Property_Type_DIM_L0 AS (
    SELECT DISTINCT(property_type)
        FROM mre_property
);

-- MRE_Address_DIM_L0
CREATE TABLE MRE_Address_DIM_L0 AS (
    SELECT DISTINCT
        Address_ID,
```

```
        Street,
        Suburb,
        Postcode
    FROM MRE_Address
);

-- MRE_Postcode_DIM_L0
CREATE TABLE MRE_Postcode_DIM_L0 AS (
    SELECT DISTINCT * FROM MRE_Postcode
);

-- MRE_State_DIM_L0
CREATE TABLE MRE_State_DIM_L0 AS (
    SELECT DISTINCT * FROM MRE_State
);

-- MRE_Advertisement_DIM_L0
CREATE TABLE MRE_Advertisement_DIM_L0 AS (
    SELECT DISTINCT * FROM MRE_Advertisement
);

-- MRE_Person_DIM_L0
CREATE TABLE MRE_Person_DIM_L0 AS (
    SELECT DISTINCT
        Person_ID,
        First_Name,
        Last_Name,
        Gender,
        Address_ID
    FROM MRE_Person
);

-- MRE_Agent_Office_DIM_L0
CREATE TABLE MRE_Agent_Office_DIM_L0 AS (
    SELECT DISTINCT
        Person_ID AS Agent_Person_ID,
        Office_ID
    FROM MRE_Agent_Office
);

-- MRE_Office_Size_DIM_L0
CREATE TABLE MRE_Office_Size_DIM_L0 (
    Office_Size_ID NUMBER,
    Office_Size_Description VARCHAR2(60)
);

INSERT INTO MRE_Office_Size_DIM_L0 VALUES (1, 'Small: < 4
employees');
INSERT INTO MRE_Office_Size_DIM_L0 VALUES (2, 'Medium: 4 - 12
employees');
```

```
INSERT INTO MRE_Office_Size_DIM_L0 VALUES (3, 'Big: > 12
employees');

-- MRE_Office_TempDIM_L0
CREATE TABLE MRE_Office_TempDIM_L0 AS (
    SELECT DISTINCT
        ao.Office_ID,
        o.Office_Name,
        COUNT(ao.Person_ID) AS Num_of_Employees
    FROM MRE_Office o, MRE_Agent_Office ao
    WHERE o.Office_ID = ao.Office_ID
    GROUP BY ao.Office_ID, o.Office_Name
);

ALTER TABLE MRE_Office_TempDIM_L0
ADD Office_Size_ID NUMBER;

UPDATE MRE_Office_TempDIM_L0
SET Office_Size_ID =
    (CASE
        WHEN Num_of_Employees < 4 THEN 1
        WHEN Num_of_Employees BETWEEN 4 AND 12 THEN 2
        WHEN Num_of_Employees > 12 THEN 3
     END);

CREATE TABLE MRE_Office_DIM_L0 AS (
    SELECT
        Office_ID,
        Office_Name,
        Office_Size_ID
    FROM MRE_Office_TempDIM_L0
);

-- MRE_Budget_DIM_L0
CREATE TABLE MRE_Budget_DIM_L0 (
    Budget_ID NUMBER,
    Budget_Description VARCHAR2(100),
    Min_Budget NUMBER,
    Max_Budget NUMBER
);

INSERT INTO MRE_Budget_DIM_L0 VALUES (1, 'Low [0 to 1,000]', 0,
1000);
INSERT INTO MRE_Budget_DIM_L0 VALUES (2, 'Medium [1,001 to
100,000]', 1001, 100000);
INSERT INTO MRE_Budget_DIM_L0 VALUES (3, 'High [100,001 to
10,000,000]', 100001, 10000000);

-- MRE_Rental_Period_DIM_L0
CREATE TABLE MRE_Rental_Period_DIM_L0 (
    Rental_Period_ID NUMBER,
```

```
    Rental_Period_Description VARCHAR2(60)
);

INSERT INTO MRE_Rental_Period_DIM_L0 VALUES (1, 'Short: < 6
months');
INSERT INTO MRE_Rental_Period_DIM_L0 VALUES (2, 'Medium: 6 - 12
months');
INSERT INTO MRE_Rental_Period_DIM_L0 VALUES (3, 'Long: > 12
months');

-- MRE_Rent_Price_DIM_L0
CREATE TABLE MRE_Rent_Price_DIM_L0 AS (
    SELECT DISTINCT
        Property_ID,
        Rent_Start_Date AS Start_date,
        Rent_End_Date AS End_date,
        Price
    FROM MRE_Rent
);

-- MRE_Season_DIM_L0
CREATE TABLE MRE_Season_DIM_L0 (
    Season_ID NUMBER,
    Season_Description VARCHAR2(10)
);

INSERT INTO MRE_Season_DIM_L0 VALUES (1, 'Summer');
INSERT INTO MRE_Season_DIM_L0 VALUES (2, 'Autumn');
INSERT INTO MRE_Season_DIM_L0 VALUES (3, 'Winter');
INSERT INTO MRE_Season_DIM_L0 VALUES (4, 'Spring');

-- MRE_Time_DIM_L0
CREATE TABLE MRE_Time_DIM_L0 AS (
    SELECT DISTINCT
        TO_CHAR(d.dates, 'YYYYMMDY') AS Time_ID,
        TO_CHAR(d.dates, 'YYYY') AS Year,
        TO_NUMBER(TO_CHAR(d.dates, 'MM'), '99') AS Month,
        TO_CHAR(d.dates, 'DY') AS Day_of_Week
    FROM (
        SELECT DISTINCT Sale_Date AS DATES FROM MRE_Sale
            WHERE Sale_Date IS NOT NULL
        UNION
        SELECT DISTINCT Rent_Start_Date FROM MRE_Rent
            WHERE Rent_Start_Date IS NOT NULL
        UNION
        SELECT DISTINCT Rent_End_Date FROM MRE_Rent
            WHERE Rent_End_Date IS NOT NULL
        ) d
);

ALTER TABLE MRE_Time_DIM_L0
```

```
ADD Season_ID NUMBER;

UPDATE MRE_Time_DIM_L0
SET Season_ID =
    (CASE
        WHEN Month = 12 OR Month BETWEEN 1 AND 2 THEN 1
        WHEN Month BETWEEN 3 AND 5 THEN 2
        WHEN Month BETWEEN 6 AND 8 THEN 3
        WHEN Month BETWEEN 9 AND 11 THEN 4
    END);

---------------------------
-- Implement fact tables --
---------------------------
-- MRE_Sale_FACT_L0
CREATE TABLE MRE_Sale_FACT_L0 AS (
    SELECT
        s.Agent_Person_ID,
        s.Client_Person_ID,
        TO_CHAR(s.Sale_Date, 'YYYYMMDY') AS Time_ID,
        s.Property_ID,
        s.Price AS Total_Sales_Price,
        COUNT(s.Sale_ID) AS Number_of_Sales
    FROM MRE_Sale s, MRE_Property p
    WHERE s.Property_ID = p.Property_ID
    AND s.Client_Person_ID IS NOT NULL
    AND s.Sale_Date IS NOT NULL
    GROUP     BY     s.Agent_Person_ID,     s.Client_Person_ID,
TO_CHAR(s.Sale_Date, 'YYYYMMDY'), s.Property_ID, s.Price
);

-- MRE_Rent_FACT_L0
CREATE TABLE MRE_Rent_TempFACT_L0 AS (
    SELECT
        r.Agent_Person_ID,
        r.Client_Person_ID,
        r.Property_ID,
        r.Rent_Start_Date,
        r.Rent_End_Date,
        p.Property_No_of_Bedrooms AS Number_of_bedrooms,
        COUNT(pf.Feature_Code) AS Number_of_features,
        ROUND((r.Price / 7) * (Rent_End_Date - Rent_Start_Date),
2) AS Total_Rent_Fee,
        COUNT(DISTINCT r.Rent_ID) AS Number_of_Rent
    FROM MRE_Rent r, MRE_Property p, MRE_Property_Feature pf
    WHERE r.Property_ID = p.Property_ID
    AND pf.Property_ID = p.Property_ID
    AND r.Client_Person_ID IS NOT NULL
    AND r.Rent_Start_Date IS NOT NULL
    AND r.Rent_End_Date IS NOT NULL
```

```
    GROUP     BY     r.Agent_Person_ID,     r.Client_Person_ID,
r.Property_ID, r.Rent_Start_Date, r.Rent_End_Date,
             p.Property_No_of_Bedrooms, ROUND((r.Price / 7) *
(Rent_End_Date - Rent_Start_Date), 2)
);

ALTER TABLE MRE_Rent_TempFACT_L0
ADD (Rental_Period_ID NUMBER,
     Scale_ID NUMBER,
     Feature_Cat_ID NUMBER);

UPDATE MRE_Rent_TempFACT_L0
SET Rental_Period_ID =
        (CASE
            WHEN MONTHS_BETWEEN(Rent_Start_Date, Rent_End_Date)
< 6 THEN 1
            WHEN MONTHS_BETWEEN(Rent_Start_Date, Rent_End_Date)
BETWEEN 6 AND 12 THEN 2
            WHEN MONTHS_BETWEEN(Rent_Start_Date, Rent_End_Date)
> 12 THEN 3
        END),
    Scale_ID =
        (CASE
            WHEN Number_of_bedrooms <= 1 THEN 1
            WHEN Number_of_bedrooms BETWEEN 2 AND 3 THEN 2
            WHEN Number_of_bedrooms BETWEEN 4 AND 6 THEN 3
            WHEN Number_of_bedrooms BETWEEN 7 AND 10 THEN 4
            WHEN Number_of_bedrooms > 10 THEN 5
        END),
    Feature_Cat_ID =
        (CASE
            WHEN Number_of_features < 10 THEN 1
            WHEN Number_of_features BETWEEN 10 AND 20 THEN 2
            WHEN Number_of_features > 20 THEN 3
        END)
;

CREATE TABLE MRE_Rent_FACT_L0 AS (
    SELECT
        Agent_Person_ID,
        Client_Person_ID,
        Property_ID,
        to_char(Rent_Start_Date, 'YYYYMMDY') as rent_start_date,
        to_char(Rent_End_Date, 'YYYYMMDY') as rent_end_date,
        Rental_Period_ID,
        Scale_ID,
        Feature_Cat_ID,
        Total_Rent_Fee,
        Number_of_Rent
    FROM MRE_Rent_TempFACT_L0
);
```

```
-- MRE_Client_FACT_L0
CREATE TABLE MRE_Client_TempFACT_L0 AS (
    SELECT
        Person_ID AS Client_Person_ID,
        Max_Budget,
        COUNT(Person_ID) AS Number_of_Clients
    FROM MRE_Client
    GROUP BY Person_ID, Min_Budget, Max_Budget
);

ALTER TABLE MRE_Client_TempFACT_L0
ADD Budget_ID VARCHAR2(2);

UPDATE MRE_Client_TempFACT_L0
SET Budget_ID =
    (CASE
        WHEN Max_Budget >= 0 AND Max_Budget <= 1000 THEN 1
        WHEN Max_Budget >= 1001 AND Max_Budget <= 100000 THEN 2
        WHEN Max_Budget >= 100001 AND Max_Budget <= 10000000
THEN 3
    END);

CREATE TABLE MRE_Client_FACT_L0 AS (
    SELECT
        Client_Person_ID,
        Budget_ID,
        Number_of_Clients
    FROM MRE_Client_TempFACT_L0
);

-- MRE_Agent_FACT_L0
CREATE TABLE MRE_Agent_FACT_L0 AS (
    SELECT * FROM
    (SELECT a.person_id as agent_person_id, SUM(nvl(s.price,
0)) + nvl(SUM(nvl(r.price, 0)/7*(r.rent_end_date -
r.rent_start_date)), 0) as total_earnings
    FROM mre_agent a, mre_sale s, mre_rent r
        WHERE a.person_id = s.agent_person_id (+)
        AND a.person_id = r.agent_person_id (+)
            GROUP BY a.person_id)
);

-- MRE_Visit_FACT_L0
CREATE TABLE MRE_Visit_FACT_L0 AS (
    SELECT DISTINCT
        Client_Person_ID,
        Agent_Person_ID,
        Property_ID,
        TO_CHAR(Visit_Date, 'YYYYMMDY') AS Time_ID,
        COUNT(*) AS Number_of_Visits
```

```
    FROM MRE_Visit
    GROUP BY Client_Person_ID, Agent_Person_ID, Property_ID,
TO_CHAR(Visit_Date, 'YYYYMMDY')
);

-- MRE_Advert_FACT_L0
CREATE TABLE MRE_Advert_FACT_L0 AS (
    SELECT DISTINCT
        pa.Property_ID,
        pa.Advert_ID,
        TO_CHAR(p.Property_Date_Added, 'YYYYMMDY') AS Time_ID,
        COUNT(pa.Advert_ID) AS Number_of_Adverts
    FROM MRE_Property_Advert pa, MRE_Property p
    WHERE pa.Property_ID = p.Property_ID
    GROUP BY pa.Property_ID, pa.Advert_ID,
TO_CHAR(p.Property_Date_Added, 'YYYYMMDY')
);


-----------------------------------------------------
-- Two-column methodology checking of fact tables --
-----------------------------------------------------
-- Numbers should be wrong since tested on non-cleaned data.

-- MRE_Sale_FACT_L0
SELECT SUM(Total_Sales_Price), SUM(Number_of_Sales) FROM
MRE_Sale_FACT_L0; -- 702,593,752 and 916
SELECT Agent_Person_ID, SUM(Total_Sales_Price),
SUM(Number_Of_Sales) FROM MRE_Sale_FACT_L0 GROUP BY
Agent_Person_ID ORDER BY Agent_Person_ID; -- 702,593,752 and 916
SELECT Client_Person_ID, SUM(Total_Sales_Price),
SUM(Number_Of_Sales) FROM MRE_Sale_FACT_L0 GROUP BY
Client_Person_ID ORDER BY Client_Person_ID; -- 702,593,752 and
916
SELECT Time_ID, SUM(Total_Sales_Price), SUM(Number_Of_Sales)
FROM MRE_Sale_FACT_L0 GROUP BY Time_ID ORDER BY Time_ID; --
702,593,752 and 916
SELECT Property_ID, SUM(Total_Sales_Price),
SUM(Number_Of_Sales) FROM MRE_Sale_FACT_L0 GROUP BY Property_ID
ORDER BY Property_ID; -- 702,593,752 and 916
SELECT Property_Type, SUM(Total_Sales_Price),
SUM(Number_Of_Sales) FROM MRE_Sale_FACT_L0 sf,
MRE_Property_Dim_L0 p WHERE sf.property_id = p.property_id GROUP
BY Property_Type ORDER BY Property_Type; -- 702,593,752 and 916

-- MRE_Rent_FACT_L0
SELECT SUM(Number_of_Rent) FROM MRE_Rent_FACT_L0; -- 1116
SELECT Agent_Person_ID, SUM(Number_Of_Rent) FROM
MRE_Rent_FACT_L0 GROUP BY Agent_Person_ID ORDER BY
Agent_Person_ID; -- 1116
```

```
SELECT Client_Person_ID, SUM(Number_Of_Rent) FROM
MRE_Rent_FACT_L0 GROUP BY Client_Person_ID ORDER BY
Client_Person_ID; -- 1116
SELECT Property_ID, SUM(Number_Of_Rent) FROM MRE_Rent_FACT_L0
GROUP BY Property_ID ORDER BY Property_ID; -- 1116
SELECT Rent_Start_Date, SUM(Number_Of_Rent) FROM
MRE_Rent_FACT_L0 GROUP BY Rent_Start_Date ORDER BY
Rent_Start_Date; -- 1116
SELECT Rent_End_Date, SUM(Number_Of_Rent) FROM MRE_Rent_FACT_L0
GROUP BY Rent_End_Date ORDER BY Rent_End_Date; -- 1116
SELECT Scale_ID, SUM(Number_Of_Rent) FROM MRE_Rent_FACT_L0
GROUP BY Scale_ID ORDER BY Scale_ID; -- 1116
SELECT Feature_Cat_ID, SUM(Number_Of_Rent) FROM
MRE_Rent_FACT_L0 GROUP BY Feature_Cat_ID ORDER BY
Feature_Cat_ID; -- 1116

-- MRE_Client_FACT_L0
SELECT SUM(Number_Of_Clients) FROM MRE_Client_FACT_L0; -- 3339
SELECT Client_Person_ID, SUM(Number_Of_Clients) FROM
MRE_Client_FACT_L0 GROUP BY Client_Person_ID ORDER BY
Client_Person_ID; -- 3339
SELECT Budget_ID, SUM(Number_Of_Clients) FROM
MRE_Client_FACT_L0 GROUP BY Budget_ID ORDER BY Budget_ID; --
3339

-- MRE_Agent_FACT_L0
SELECT SUM(Total_Earnings) FROM MRE_Agent_FACT_L0; --
477,290,000
SELECT Agent_Person_ID, SUM(Total_Earnings) FROM
MRE_Agent_FACT_L0 GROUP BY Agent_Person_ID ORDER BY
Agent_Person_ID; -- 477,290,000

-- MRE_Visit_FACT_L0
SELECT SUM(Number_of_Visits) FROM MRE_Visit_FACT_L0; -- 575
SELECT Client_Person_ID, SUM(Number_of_Visits) FROM
MRE_Visit_FACT_L0 GROUP BY Client_Person_ID ORDER BY
Client_Person_ID; -- 575
SELECT Agent_Person_ID, SUM(Number_of_Visits) FROM
MRE_Visit_FACT_L0 GROUP BY Agent_Person_ID ORDER BY
Agent_Person_ID; -- 575
SELECT Property_ID, SUM(Number_of_Visits) FROM
MRE_Visit_FACT_L0 GROUP BY Property_ID ORDER BY Property_ID; --
575
SELECT Time_ID, SUM(Number_of_Visits) FROM MRE_Visit_FACT_L0
GROUP BY Time_ID ORDER BY Time_ID; -- 575

-- MRE_Advert_FACT_L0
SELECT SUM(Number_of_Adverts) FROM MRE_Advert_FACT_L0; -- 3646
SELECT Property_ID, SUM(Number_of_Adverts) FROM
MRE_Advert_FACT_L0 GROUP BY Property_ID ORDER BY Property_ID; -
- 3646
```

```
SELECT Advert_ID, SUM(Number_of_Adverts) FROM
MRE_Advert_FACT_L0 GROUP BY Advert_ID ORDER BY Advert_ID; --
3646
SELECT Time_ID, SUM(Number_of_Adverts) FROM MRE_Advert_FACT_L0
GROUP BY Time_ID ORDER BY Time_ID; -- 3646

COMMIT;
```

Output c – Screenshots of Tables

Performed using:

```
SELECT *
     FROM [table_name];
```

c1 Level 2 Star Schema

MRE_Scale_DIM_L2

	SCALE_ID	SCALE_DESCRIPTION
1	1	extra small
2	2	small
3	3	medium
4	4	large
5	5	extra large

MRE_Feature_Cat_DIM_L2

	FEATURE_CAT_ID	FEATURE_CAT_DESCRIPTION
1	1	basic
2	2	standard
3	3	luxurious

MRE_Property_DIM_L2

	PROPERTY_ID	ADDRESS_ID	PROPERTY_TYPE
1	4	4	House
2	5	5	House
3	6	6	House
4	7	7	House
5	8	8	House
6	9	9	House
7	10	10	House
8	11	11	House
9	12	12	House
10	13	13	House

MRE_Property_Feature_Bridge_L2

	PROPERTY_ID	FEATURE_CODE
1	9	5
2	9	11
3	9	117
4	11	12
5	13	16
6	13	29
7	13	589
8	14	27
9	14	30
10	15	9
11	15	87
12	16	23
13	18	1
14	18	86
15	21	2
16	22	5
17	24	2

MRE_Feature_DIM_L2

	FEATURE_C...	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
17	17	Swimming Pool
18	18	Floorboards

MRE_Property_Type_DIM_L2

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

MRE_Address_DIM_L2

	ADDRESS_ID	SUBURB	POSTCODE
1	533	Woodridge	4114
2	535	West End	4101
3	537	Lota	4179
4	541	North Lakes	4509
5	544	Caboolture	4510
6	551	Murarrie	4172
7	552	Fortitude Valley	4006
8	563	Coorparoo	4151
9	564	St Lucia	4067
10	565	Deagon	4017
11	575	Manly West	4179
12	579	Fortitude Valley	4006
13	580	Logan Central	4114
14	604	Sunnybank	4109
15	608	Acacia Ridge	4110
16	620	Chermside West	4032
17	623	Mansfield	4122
18	1422	Macquarie	2614

MRE_Postcode_DIM_L2

	POSTCODE	STATE_CODE
1	2063	NSW
2	2068	NSW
3	2070	NSW
4	2090	NSW
5	2093	NSW
6	2100	NSW
7	2122	NSW
8	2153	NSW
9	2166	NSW
10	2194	NSW
11	2200	NSW
12	2204	NSW
13	2216	NSW
14	2218	NSW
15	2570	NSW
16	2650	NSW
17	2750	NSW
18	2904	ACT

MRE_State_DIM_L2

	STATE_CODE	STATE_NAME
1	ACT	Australian Capital Territory
2	NSW	New South Wales
3	NT	Northern Territory
4	QLD	Queensland
5	SA	South Australia
6	TAS	Tasmania
7	VIC	Victoria
8	WA	Western Australia

MRE_Advertisement_DIM_L2

	ADVERT_ID	ADVERT_NAME
1	18	Sale New House & Land
2	20	Sale Semi-Detached
3	23	Sale Townhouse
4	2	Rent Block of Units
5	3	Rent Duplex
6	4	Rent House
7	5	Rent New Apartments / Off the Plan
8	15	Sale Duplex
9	21	Sale Studio
10	9	Rent Terrace
11	17	Sale New Apartments / Off the Plan
12	25	Sale Villa
13	1	Rent Apartment / Unit / Flat
14	6	Rent Penthouse
15	7	Rent Semi-Detached
16	10	Rent Townhouse
17	14	Sale Development Site
18	13	Sale Block of Units

MRE_Person_DIM_L2

	PERSON_ID	FIRST_NAME	LAST_NAME	GENDER	ADDRESS_ID
1	977	Burton	Jonsson	Male	6637
2	978	Gustave	Adamolli	Male	6638
3	980	Niall	Thormann	Male	6639
4	981	Franky	Plowman	Male	6640
5	982	Adolpho	Tregien	Male	6641
6	983	Kate	De la Yglesias	Female	6642
7	984	Elisha	Scroxton	Female	6643
8	986	Haven	Insko	Male	6644
9	987	Bidget	Delhay	Female	6645
10	988	Valle	Vedekhin	Male	6646
11	989	Opaline	Fiske	Female	6647
12	991	Rozina	Oats	Female	6648
13	992	Kirbie	Causier	Female	6649
14	994	Thea	Hatrick	Female	6651
15	995	Roby	Gaylord	Female	6652
16	997	Saloma	Wagge	Female	6653
17	998	Stanfield	Iacobetto	Male	6654
18	999	Toby	Hawking	Female	6655

MRE_Agent_Office_DIM_L2

	AGENT_PERSON_ID	OFFICE_ID
1	61	438
2	2210	1132
3	1567	275
4	72	1006
5	711	1029
6	1607	607
7	2246	609
8	1940	1172
9	421	412
10	1314	364
11	1637	556
12	753	311
13	447	1058
14	1645	1076
15	1052	818
16	2015	199
17	1674	555
18	1679	324

MRE_Office_DIM_L2

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

MRE_Budget_DIM_L2

	BUDGET_ID	BUDGET_DESCRIPTION
1	l	Budget between 0 and 1000
2	m	Budget between 1001 and 100000
3	h	Budget more than 100001

MRE_Rental_Period_DIM_L2

	RENTAL_PERIOD_ID	RENTAL_PERIOD_DESCRIPTION
1	1	short
2	2	medium
3	3	long

MRE_Wishlist_DIM_L2

	FEATURE_CODE	PERSON_ID
1	20	5236
2	20	5268
3	20	5278
4	20	5322
5	22	5298
6	22	5540
7	23	5128
8	23	5182
9	23	5405
10	23	5534
11	23	5568
12	24	5152
13	24	5165
14	25	5325
15	25	5505
16	25	5538
17	26	5073
18	26	5124

MRE_Rent_Price_DIM_L2

	PROPERTY_ID	RENT_START_DATE	RENT_END_DATE	PRICE
1	6199	12/01/2020	28/06/2020	795
2	6063	02/05/2020	18/10/2020	500
3	6074	01/05/2020	17/10/2020	370
4	6142	12/02/2020	29/07/2020	795
5	6146	20/04/2020	06/10/2020	595
6	5373	27/04/2020	13/10/2020	350
7	5801	25/02/2020	11/08/2020	600
8	5513	01/01/2020	17/06/2020	430
9	5709	29/03/2020	13/09/2020	420
10	5548	23/04/2020	09/10/2020	520
11	5901	01/05/2020	17/10/2020	330
12	5724	01/05/2020	17/10/2020	500
13	6035	30/04/2020	16/10/2020	625
14	5557	23/04/2020	09/10/2020	815
15	5621	21/04/2020	07/10/2020	370
16	5598	23/04/2020	09/10/2020	495
17	5386	18/03/2020	02/09/2020	1100
18	5766	18/03/2020	02/09/2020	430

MRE_Temp_Time_DIM_L2

	DATES	TIME_ID	YEAR	MONTH	SEASON_ID
1	30-DEC-19	201912MON	2019	12	4
2	30-DEC-19	201912MON	2019	12	4
3	30-DEC-19	201912MON	2019	12	4
4	30-DEC-19	201912MON	2019	12	4
5	30-DEC-19	201912MON	2019	12	4
6	30-DEC-19	201912MON	2019	12	4
7	30-DEC-19	201912MON	2019	12	4
8	30-DEC-19	201912MON	2019	12	4
9	30-DEC-19	201912MON	2019	12	4
10	30-DEC-19	201912MON	2019	12	4

MRE_Time_DIM_L2

	TIME_ID	YEAR	MONTH	SEASON_ID
1	202001FRI	2020	1	4
2	202002SUN	2020	2	4
3	202004MON	2020	4	1
4	202004TUE	2020	4	1
5	202006TUE	2020	6	2
6	202007SUN	2020	7	2
7	202007MON	2020	7	2
8	202010FRI	2020	10	3
9	201912TUE	2019	12	4
10	202006MON	2020	6	2
11	202006SUN	2020	6	2
12	202008SUN	2020	8	2
13	202008THU	2020	8	2
14	202003TUE	2020	3	1
15	202003THU	2020	3	1
16	202004THU	2020	4	1
17	202004SUN	2020	4	1
18	202005SUN	2020	5	1

MRE_Season_DIM_L2

	SEASON_ID	SEASON_DESCRIPTION
1	1	Spring
2	2	Summer
3	3	Autumn
4	4	Winter

MRE_Agent_FACT_L2

	AGENT_P...	TOTAL_EARNIN...
1	574	26400
2	584	24840
3	604	30000.000...
4	606	18590
5	614	9840
6	729	9359.9999...
7	730	24600
8	810	13200
9	811	59880
10	815	9480
11	826	25680
12	851	16285.714...
13	904	14640
14	1013	30522.857...
15	1098	75185.000...
16	1110	16800
17	1159	26557.142...
18	1184	10622.857...

MRE_Temp_Client_L2

	MAX_BUDGET	BUDGET_ID
1	658900	h
2	988900	h
3	713900	h
4	1089000	h
5	207900	h
6	2145000	h
7	878900	h
8	1540000	h
9	412500	h
10	988900	h

MRE_Client_FACT_L2

	BUDGET_ID	TOTAL_NUMBER_OF_CLIENT
1	h	1287
2	l	1581
3	m	466

MRE_Temp_Rent_FACT_L2

	PROPERTY_ID	DATES	PROPERTY_NO_OF_BEDROOMS	NUMBER_OF_FEATURES	PRICE	RENT_END_DATE	RENT_START_DATE	NUM_OF_RENT	TIME_ID	SCALE_ID	FEATURE_CAT_ID
1	3262	01-APR-20	1	3	270	16-SEP-20	01-APR-20	1	202004WED	1	1
2	3281	23-APR-20	1	6	290	09-OCT-20	23-APR-20	1	202004THU	1	1
3	3034	17-FEB-20	3	1	380	03-AUG-20	17-FEB-20	1	202002MON	2	1
4	3057	11-APR-20	2	5	620	26-SEP-20	11-APR-20	1	202004SAT	2	1
5	3695	25-MAR-20	1	12	270	09-SEP-20	25-MAR-20	1	202003WED	1	2
6	3721	23-MAR-20	1	10	525	07-SEP-20	23-MAR-20	1	202003MON	1	2
7	4042	16-JAN-20	2	8	500	02-JUL-20	16-JAN-20	1	202001THU	2	1
8	4696	12-APR-20	1	3	445	27-SEP-20	12-APR-20	1	202004SUN	1	1
9	4731	22-APR-20	2	3	470	08-OCT-20	22-APR-20	1	202004WED	2	1
10	4739	30-APR-20	1	1	410	16-OCT-20	30-APR-20	1	202004THU	1	1

MRE_Rent_FACT_L2

	PROPERTY_ID	TIME_ID	SCALE_ID	FEATURE_CAT_ID	TOTAL_RENT_FEE	NUMBER_OF_RENT
1	3000	202002SAT	2	2	20399.999999999999999999999999999999999	1
2	3146	202004MON	2	3	13882.142857142857142857142857142857142857	1
3	3147	202003WED	2	2	7920	1
4	3524	202005SAT	3	1	8932.857142857142857142857142857142857143	1
5	3672	202004TUE	2	2	13278.571428571428571428571428571428571428	1
6	3691	202003THU	3	1	11880	1
7	3749	202002THU	2	2	28799.999999999999999999999999999999999999	1
8	4424	202004MON	1	1	15692.857142857142857142857142857142857571	1
9	4517	202004WED	2	1	10622.857142857142857142857142857142857571	1
10	4742	202002SAT	1	1	10800	1
11	5030	202003MON	1	1	14880	1
12	5070	202001MON	2	1	15120	1
13	5072	202004TUE	2	1	17382.857142857142857142857142857142857572	1
14	5117	202004TUE	1	1	9536.428571428571428571428571428571428572	1
15	5204	202004TUE	3	2	18107.142857142857142857142857142857142428571428	1
16	5256	202002TUE	2	1	15600	1
17	2955	202003FRI	3	1	9480	1
18	2995	202004SUN	2	1	15720	1

MRE_Temp_Visit_L2

	VISIT_DATE	VISIT_TIME_ID
1	29-MAR-20	202003SUN
2	29-MAR-20	202003SUN
3	29-MAR-20	202003SUN
4	12-MAR-20	202003THU
5	29-MAR-20	202003SUN
6	29-MAR-20	202003SUN
7	23-MAR-20	202003MON
8	23-MAR-20	202003MON
9	23-MAR-20	202003MON
10	23-MAR-20	202003MON

MRE_Visit_FACT_L2

	VISIT_TIME_ID	NUMBER_OF_VISIT
1	202004TUE	19
2	202004MON	28
3	202003SAT	77
4	202003THU	58
5	202003MON	62
6	202004THU	12
7	202003FRI	64
8	202004FRI	11
9	202003WED	55
10	202004SAT	30
11	202003TUE	64
12	202004WED	21
13	202003SUN	50
14	202004SUN	23

MRE_Temp_Sale_FACT_L2

	PROPERTY_ID	SALE_DATE	PROPERTY_TYPE	PRICE	TIME_ID
1	5	29-JAN-20	House	1825000	202001WED
2	11	14-FEB-20	House	1150000	202002FRI
3	13	25-FEB-20	House	1075000	202002TUE
4	18	06-JAN-20	House	900000	202001MON
5	19	28-JAN-20	Apartment / Unit / Flat	895000	202001TUE
6	24	15-JAN-20	House	769000	202001WED
7	30	29-MAR-20	House	685000	202003SUN
8	31	02-FEB-20	House	680000	202002SUN
9	33	07-FEB-20	House	665000	202002FRI
10	34	05-JAN-20	House	660000	202001SUN

MRE_Sale_FACT_L2

	PROPERTY_ID	TIME_ID	TOTAL_SALES_PRICE	NUMBER_OF_SALES
1	34	202001SUN	660000	1
2	132	202003WED	287000	1
3	19	202001TUE	895000	1
4	159	202003MON	545000	1
5	162	202001TUE	340000	1
6	220	202003MON	280000	1
7	89	202003WED	685000	1
8	191	202002SUN	675000	1
9	482	202002FRI	329000	1
10	343	202002WED	499000	1
11	567	202003FRI	249000	1
12	576	202003THU	260000	1
13	582	202003THU	349000	1
14	675	202001MON	859000	1
15	684	202002MON	500000	1
16	609	202001FRI	369000	1
17	611	202001FRI	500000	1
18	697	202003MON	1800000	1

MRE_Temp_Advert_L2

	ADVERT_ID	PROPERTY_DATE_ADDED	TIME_ID
1	16	28-MAR-20	202003SAT
2	16	27-APR-20	202004MON
3	16	14-MAR-20	202003SAT
4	23	25-MAR-20	202003WED
5	16	13-MAR-20	202003FRI
6	16	07-APR-20	202004TUE
7	23	13-APR-20	202004MON
8	16	13-MAR-20	202003FRI
9	12	04-MAR-20	202003WED
10	12	01-APR-20	202004WED

MRE_Advert_FACT_L2

	TIME_ID	ADVERT_ID	NUMBER_OF_ADVERTS
1	202003SAT	16	4
2	202004SUN	12	4
3	202004FRI	17	3
4	202004FRI	4	4
5	202003SAT	1	2
6	202004WED	4	5
7	202004FRI	7	1
8	202003MON	16	5
9	202004THU	25	1
10	202003FRI	12	4
11	202003SUN	25	2
12	202003SUN	1	2
13	202003TUE	17	2
14	202003TUE	20	3
15	202003SUN	4	2
16	202003TUE	1	2
17	202003MON	10	2
18	202004FRI	23	3

c2 Level 0 Star Schema

MRE_Scale_DIM_L0

	SCALE_ID	SCALE_DESCRIPTION
1	1	Extra small: <= 1 bedroom
2	2	Small: 2-3 bedrooms
3	3	Medium: 3-6 bedrooms
4	4	Large: 6-10 bedrooms
5	5	Extra large: > 10 bedrooms

MRE_Feature_Cat_DIM_L0

	FEATURE_CAT_ID	FEATURE_CAT_DESCRIPTION
1	1	Very basic: < 10 features
2	2	Standard: 10-20 features
3	3	Luxurious: > 20 features

MRE_Property_DIM_L0

	PROPERTY_ID	PROPERTY_DATE_ADDED	ADDRESS_ID	PROPERTY_TYPE
1	20	11-APR-20	20	House
2	21	02-APR-20	21	Townhouse
3	22	14-APR-20	22	House
4	23	16-MAR-20	23	House
5	24	01-DEC-19	24	House
6	25	17-APR-20	25	House
7	26	21-APR-20	26	Townhouse
8	27	19-APR-20	27	House
9	28	08-APR-20	28	House
10	29	12-MAR-20	29	House
11	30	13-FEB-20	30	House
12	31	19-DEC-19	31	House
13	32	14-APR-20	32	House
14	33	24-DEC-19	33	House
15	34	21-NOV-19	34	House
16	35	12-APR-20	35	House
17	36	05-MAR-20	36	House
18	37	21-APR-20	37	Townhouse

MRE_Property_Feature_Bridge_L0

	PROPERTY_ID	FEATURE_CODE
1	9	5
2	9	11
3	9	117
4	11	12
5	13	16
6	13	29
7	13	589
8	14	27
9	14	30
10	15	9
11	15	87
12	16	23
13	18	1
14	18	86
15	21	2
16	22	5
17	24	2
18	24	6
19	24	14
20	25	2
21	25	22

MRE_Feature_DIM_L0

	FEATURE_CODE	FEATURE_DESCRIPTION
1	3	Carpeted
2	10	Prestige Homes
3	14	Ensuite
4	18	Floorboards
5	24	Broadband Internet Available
6	32	Swimming Pool - In Ground
7	37	Solar panels
8	39	Security Alarm
9	52	Rumpus
10	61	Window Treatments
11	63	Side access
12	69	Panoramic View
13	71	Bath
14	89	Workshop
15	102	Swimming/Lap Pool
16	103	3.5KW Solar system
17	111	Water Front
18	118	Boat and Caravan Parking
19	131	PORCELAIN TILES THROUGH
20	134	STEPS TO OCEAN
21	138	Electric Hot Water
22	141	Life Style
23	150	MASSIVE POWERED SHED
24	159	Solar power

MRE_Wishlist_DIM_L0

	CLIENT_PERSON_ID	FEATURE_CODE
1	5202	20
2	5205	20
3	5208	20
4	5211	20
5	5216	20
6	5225	20
7	5227	20
8	5231	20
9	5234	20
10	5236	20
11	5244	20
12	5248	20
13	5256	20
14	5257	20
15	5264	20
16	5266	20
17	5268	20
18	5273	20
19	5278	20
20	5281	20
21	5283	20
22	5290	20
23	5291	20
24	5293	20

MRE_Property_Type_DIM_L0

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

MRE_Address_DIM_L0

	ADDRESS_ID	STREET	SUBURB	POSTCODE
1	535	4/24-26 Ferry Road	West End	4101
2	541	22 Borbidge Street	North Lakes	4509
3	545	23/9 Harpulia Court	Morayfield	4506
4	546	64 Freshwater Drive	Berrinba	4117
5	558	11/82 Boundary Street	Brisbane City	4000
6	562	12 Amcord Place	Rothwell	4022
7	567	2/19 Kathleen Street	Richlands	4077
8	576	3/21-29 Second Avenue	Marsden	4132
9	583	80 Minto Crescent	Arana Hills	4054
10	597	245 Oates Avenue	Holland Park	4121
11	599	76 Halpine Parade	Warner	4500
12	610	1 Akora Street	Macgregor	4109
13	613	55 Florence Street	Teneriffe	4005
14	615	106 Mitchell St	Acacia Ridge	4110
15	1425	14 Charlton Crescent	Gordon	2906
16	1427	168 Streeton Drive	Chapman	2611
17	1463	47 Holmes Crescent	Campbell	2612
18	1472	3 Giordano Street	Denman Prospect	2611
19	1474	42 Amaroo Street	Reid	2612
20	1476	41/5 Hely Street	Griffith	2603

MRE_Postcode_DIM_L0

	POSTCODE	STATE_CODE
1	2063	NSW
2	2068	NSW
3	2070	NSW
4	2090	NSW
5	2093	NSW
6	2100	NSW
7	2122	NSW
8	2153	NSW
9	2166	NSW
10	2194	NSW
11	2200	NSW
12	2204	NSW
13	2216	NSW
14	2218	NSW
15	2570	NSW
16	2650	NSW
17	2750	NSW
18	2904	ACT
19	2905	ACT
20	3040	VIC
21	3054	VIC

MRE_State_DIM_L0

	STATE_CODE	STATE_NAME
1	ACT	Australian Capital Territory
2	QLD	Queensland
3	TAS	Tasmania
4	NT	Northern Territory
5	WA	Western Australia
6	NSW	New South Wales
7	SA	South Australia
8	VIC	Victoria

MRE_Advertisement_DIM_L0

	ADVERT_ID	ADVERT_NAME
1	18	Sale New House & Land
2	20	Sale Semi-Detached
3	23	Sale Townhouse
4	2	Rent Block of Units
5	3	Rent Duplex
6	4	Rent House
7	5	Rent New Apartments / Off the Plan
8	15	Sale Duplex
9	21	Sale Studio
10	9	Rent Terrace
11	17	Sale New Apartments / Off the Plan
12	25	Sale Villa
13	1	Rent Apartment / Unit / Flat
14	6	Rent Penthouse
15	7	Rent Semi-Detached
16	10	Rent Townhouse
17	14	Sale Development Site
18	13	Sale Block of Units
19	19	Sale Penthouse
20	24	Sale Vacant land

MRE_Person_DIM_L0

	PERSON_ID	FIRST_NAME	LAST_NAME	GENDER	ADDRESS_ID
1	980	Niall	Thormann	Male	6639
2	988	Valle	Vedekhin	Male	6646
3	992	Kirbie	Causier	Female	6649
4	995	Roby	Gaylord	Female	6652
5	998	Stanfield	Iacobetto	Male	6654
6	1004	Ali	Ciotti	Female	6659
7	1016	Anet	Wilkenson	Female	6669
8	1031	Carroll	Eilers	Male	6684
9	1034	Sherman	Meadley	Male	6687
10	1048	Jacenta	Amsden	Female	6701
11	1067	Alanna	Trembey	Female	6718
12	1070	Emmy	Povey	Female	6720
13	1072	Coriss	Gadney	Female	6722
14	1076	Basile	Newton	Male	6725
15	1084	Stephen	Emney	Male	6731
16	1085	Zacharias	Rodrigo	Male	6732
17	1094	Ruttger	Letterick	Male	6739
18	14	Rockwell	Feige	Male	6222
19	19	Katerine	Barby	Female	6227
20	22	Keefe	Hauger	Male	6230
21	32	Eugenio	Tudgay	Male	6240

MRE_Agent_Office_DIM_L0

	AGENT_PERSON_ID	OFFICE_ID
1	61	438
2	2210	1132
3	1567	275
4	72	1006
5	711	1029
6	1607	607
7	2246	609
8	1940	1172
9	421	412
10	1314	364
11	1637	556
12	753	311
13	447	1058
14	1645	1076
15	1052	818
16	2015	199
17	1674	555
18	1679	324
19	2335	13
20	1695	186

MRE_Office_TempDIM_L0

	OFFICE_ID	OFFICE_NAME	NUM_OF_EMPLOYEES	OFFICE_SIZE_ID
1	685	McGrath Woden	4	2
2	687	McKean McGregor Real Estate Pty Ltd	3	1
3	690	Merrick Property Group	1	1
4	693	Mitchell Torre Real Estate	2	1
5	710	NGU REAL ESTATE RIPLEY	1	1
6	716	Nelson Alexander Northcote	4	2
7	724	Norwes Property	1	1
8	727	O'Brien Real Estate Carrum Downs	3	1
9	732	OBrien Real Estate Chelsea	3	1
10	737	Obsidian Property Pty Ltd	1	1

MRE_Office_DIM_L0

	OFFICE_ID	OFFICE_NAME	OFFICE_SIZE_ID
1	574	Laing+Simmons Kings Langley/Kings Park	1
2	288	Duet Property Group	2
3	217	Chisholm & Gamon Elwood	2
4	1133	Village Real Estate Seddon	2
5	364	Gary Peer	3
6	558	LJ Hooker Ormeau	1
7	495	Jellis Craig - Brunswick	2
8	818	Quest Realty Group	1
9	69	Belle Maison Realty	1
10	978	Red Brick Properties	1
11	99	Belle Property Robina	1
12	855	Raine & Horne Townsville	1
13	756	PM Realty	1
14	225	City Residential Property	1
15	357	GA Realty	1
16	1078	Sydney Cove Property	1
17	790	Place Estate Agents Woolloongabba	2
18	649	McGrath Bulimba	1
19	247	Coronis - North Lakes	2
20	814	Propertyworks QLD	1
21	925	Ray White Oakleigh	2
22	344	Fletchers Manningham	1

MRE_Office_Size_DIM_L0

	OFFICE_SIZE_ID	OFFICE_SIZE_DESCRIPTION
1	1	Small: < 4 employees
2	2	Medium: 4 - 12 employees
3	3	Big: > 12 employees

MRE_Budget_DIM_L0

	BUDGET_ID	BUDGET_DESCRIPTION	MIN_BUDGET	MAX_BUDGET
1	1	Low [0 to 1,000]	0	1000
2	3	Medium [1,001 to 100,000]	1001	100000
3	5	High [100,001 to 10,000,000]	100001	10000000

MRE_Rental_Period_DIM_L0

	RENTAL_PERIOD_ID	RENTAL_PERIOD_DESCRIPTION
1	1	Short: < 6 months
2	2	Medium: 6 - 12 months
3	3	Long: > 12 months

MRE_Rent_Price_DIM_L0

	PROPERTY_ID	START_DATE	END_DATE	PRICE
1	6142	12-FEB-20	29-JUL-20	795
2	5901	01-MAY-20	17-OCT-20	330
3	5621	21-APR-20	07-OCT-20	370
4	5386	18-MAR-20	02-SEP-20	1100
5	5856	18-APR-20	04-OCT-20	590
6	5673	01-JAN-20	17-JUN-20	450
7	5682	10-APR-20	25-SEP-20	410
8	6131	11-APR-20	26-SEP-20	1450
9	5039	20-APR-20	06-OCT-20	380
10	4739	30-APR-20	16-OCT-20	410
11	5099	23-MAR-20	07-SEP-20	560
12	4650	26-APR-20	12-OCT-20	450
13	5002	28-JAN-20	14-JUL-20	750
14	5093	10-FEB-20	27-JUL-20	920
15	4872	11-JAN-20	27-JUN-20	460
16	4587	24-APR-20	10-OCT-20	420
17	6078	03-MAR-20	18-AUG-20	410
18	4054	08-MAR-20	23-AUG-20	425
19	4159	30-DEC-19	15-JUN-20	550
20	3912	27-APR-20	13-OCT-20	675
21	3785	01-FEB-20	18-JUL-20	720

MRE_Season_DIM_L0

	SEASON_ID	SEASON_DESCRIPTION
1	1	Summer
2	2	Autumn
3	3	Winter
4	4	Spring

MRE_Time_DIM_L0

	TIME_ID	YEAR	MONTH	DAY_OF_WEEK	SEASON_ID
1	201912TUE	2019	12	TUE	1
2	202001WED	2020	1	WED	1
3	202002FRI	2020	2	FRI	1
4	202003THU	2020	3	THU	2
5	202006TUE	2020	6	TUE	3
6	202008MON	2020	8	MON	3
7	202009FRI	2020	9	FRI	4
8	202001SUN	2020	1	SUN	1
9	202001MON	2020	1	MON	1
10	202002THU	2020	2	THU	1
11	202003SUN	2020	3	SUN	2
12	202003SAT	2020	3	SAT	2
13	202004MON	2020	4	MON	2
14	202005SUN	2020	5	SUN	2
15	202006MON	2020	6	MON	3
16	202007THU	2020	7	THU	3
17	202007SAT	2020	7	SAT	3
18	202008WED	2020	8	WED	3
19	202008FRI	2020	8	FRI	3
20	202009SAT	2020	9	SAT	4
21	202010THU	2020	10	THU	4

MRE_Sale_FACT_L0

	AGENT_PERSON_ID	CLIENT_PERSON_ID	TIME_ID	PROPERTY_ID	TOTAL_SALES_PRICE	NUMBER_OF_SALES
1	1830	3148	202001SAT	130	1350000	1
2	626	2697	202003WED	132	287000	1
3	9	2475	202001TUE	162	340000	1
4	312	2568	202003SAT	166	429000	1
5	628	2698	202002MON	151	499000	1
6	1195	2906	202003SAT	67	380000	1
7	622	2692	202003TUE	83	780000	1
8	924	2809	202001MON	87	699000	1
9	18	2480	202001WED	333	1000000	1
10	1830	3149	202002THU	263	669000	1
11	1202	2913	202002TUE	305	440000	1
12	313	2569	202001WED	277	325000	1
13	36	2495	202001SAT	501	700000	1
14	644	2710	202004SUN	364	575000	1
15	1851	3164	202004WED	373	379000	1
16	2167	3285	202001SAT	388	439000	1
17	639	2704	202002WED	408	700000	1
18	31	2493	202002THU	348	300000	1
19	1536	3039	202001SAT	565	585000	1
20	658	2716	202003FRI	567	249000	1
21	1853	3167	202003THU	584	519000	1
22	2192	3300	202001WED	631	1025000	1
23	1228	2935	202002MON	684	500000	1

MRE_Rent_TempFACT_L0

	AGENT_PERSON_ID	CLIENT_PERSON_ID	PROPERTY_ID	RENT_START_DATE	RENT_END_DATE	NUMBER_OF_BEDROOMS	NUMBER_OF_FEATURES	TOTAL_RENT_FEE	NUMBER_OF_RENT	RENTAL_PERIOD_ID	SCALE_ID	FEATURE_CAT_ID
1	1760	4542	3252	03-MAY-20	19-OCT-20	3	14	8450	1	1	2	2
2	2089	4761	3313	28-JAN-20	14-JUL-20	1	3	6000	1	1	1	1
3	1312	4258	2995	05-APR-20	20-SEP-20	3	1	15720	1	1	2	1
4	233	3457	3044	04-APR-20	19-SEP-20	2	1	18000	1	1	2	1
5	545	3655	3435	30-JAN-20	16-JUL-20	3	7	11400	1	1	2	1
6	2261	4861	3131	02-MAR-20	17-AUG-20	3	6	10080	1	1	2	1
7	2090	4768	3147	18-MAR-20	02-SEP-20	3	14	7920	1	1	2	2
8	2405	4953	3579	17-FEB-20	03-AUG-20	3	10	13920	1	1	2	2
9	1407	4298	3847	01-MAR-20	16-AUG-20	2	10	18000	1	1	2	2
10	2357	4885	3961	20-APR-20	06-OCT-20	1	7	15210	1	1	1	1

MRE_Rent_FACT_L0

	AGENT_PERSON_ID	CLIENT_PERSON_ID	PROPERTY_ID	RENT_START_DATE	RENT_END_DATE	RENTAL_PERIOD_ID	SCALE_ID	FEATURE_CAT_ID	TOTAL_RENT_FEE	NUMBER_OF_RENT
1	553	3681	3118	202004WED	202010THU	1	2	1	10864.29	1
2	500	3580	3385	202002SAT	202008SAT	1	2	1	7200	1
3	552	3675	3470	202001THU	202007THU	1	1	2	8760	1
4	1745	4519	3651	202004THU	202010FRI	1	2	2	10864.29	1
5	1765	4548	3664	202001SUN	202007SUN	1	3	1	30000	1
6	1112	4046	3834	202004THU	202010FRI	1	2	1	19314.29	1
7	2370	4901	3997	202004TUE	202010WED	1	2	2	14485.71	1
8	845	3874	4372	202003THU	202009THU	1	1	2	11880	1
9	904	3971	4510	202001THU	202006THU	1	2	2	7440	1
10	284	3531	4611	202003THU	202009THU	1	3	3	10800	1
11	2137	4826	4661	202004WED	202010THU	1	2	1	10864.29	1
12	594	3772	4684	202004MON	202010TUE	1	1	1	9415.71	1
13	1179	4222	4848	202004TUE	202009TUE	1	2	1	14400	1
14	1178	4209	4865	202004SAT	202010SUN	1	2	1	13278.57	1
15	286	3539	4989	202002THU	202007THU	1	1	1	10800	1
16	1806	4659	5123	202003MON	202009MON	1	2	1	15120	1
17	1430	4332	5232	202003MON	202008MON	1	2	1	8160	1
18	261	3497	5354	202001THU	202007THU	1	2	1	10320	1
19	2110	4792	5941	202003FRI	202008FRI	1	1	1	8400	1
20	1459	4384	5988	202004THU	202010FRI	1	2	1	13278.57	1
21	2404	4951	3331	202004WED	202010THU	1	2	1	12675	1
22	1427	4322	3549	202001TUE	202007TUE	1	3	1	10800	1
23	1443	4351	3685	202002WED	202007WED	1	2	2	10080	1

MRE_Client_TempFACT_L0

	CLIENT_PERSON_ID	MAX_BUDGET	NUMBER_OF_CLIENTS	BUDGET_ID
1	2962	988900	1	5
2	2967	878900	1	5
3	2970	988900	1	5
4	2977	687500	1	5
5	2987	1314500	1	5
6	2997	1210000	1	5
7	3001	682000	1	5
8	3010	319000	1	5
9	3026	522500	1	5
10	3040	598400	1	5

MRE_Client_FACT_L0

	CLIENT_PERSON_ID	BUDGET_ID	NUMBER_OF_CLIENTS
1	3278	5	1
2	3356	5	1
3	2753	5	1
4	2532	5	1
5	3961	1	1
6	3962	1	1
7	3964	1	1
8	3969	1	1
9	3978	1	1
10	3982	1	1
11	3983	1	1
12	3991	1	1
13	3997	1	1
14	3999	1	1
15	4005	1	1
16	4016	1	1
17	4025	1	1
18	4030	1	1
19	4033	3	1
20	4034	1	1
21	4040	1	1
22	4042	1	1
23	4047	1	1
24	4065	3	1

MRE_Agent_FACT_L0

	AGENT_PERSON_ID	TOTAL_EARNINGS
1	574	26400
2	584	24840
3	604	30000.000000000000000000000000000000001
4	606	18590
5	614	9840
6	729	9359.999999999999999999999999999999999
7	730	24600
8	810	13200
9	811	59880
10	815	9480
11	826	25680
12	851	16285.714285714285714285714285714285143
13	904	14640
14	1013	30522.857142857142857142857142857142857
15	1098	75185.000000000000000000000000000000001
16	1110	16800
17	1159	26557.142857142857142857142857142857429
18	1184	10622.857142857142857142857142857142857
19	1313	10200
20	1386	77265.714285714285714285714285714285144
21	1387	14365
22	1399	18000
23	1403	12071.428571428571428571428571428571428

MRE_Visit_FACT_L0

	CLIENT_PERSON_ID	AGENT_PERSON_ID	PROPERTY_ID	TIME_ID	NUMBER_OF_VISITS
1	5474	253	5411	202003SUN	1
2	5470	569	5937	202003FRI	1
3	5627	584	6163	202003MON	1
4	5605	591	6088	202003WED	1
5	5470	616	6136	202004SAT	1
6	5626	616	6136	202004SAT	1
7	5605	868	5589	202004MON	1
8	5534	884	5488	202004WED	1
9	5325	887	5275	202003SAT	1
10	5538	1154	5538	202003TUE	1
11	5567	1155	5535	202003SAT	1
12	5627	1450	5406	202004SUN	1
13	5571	1462	6174	202004THU	1
14	5456	1467	5615	202003THU	1
15	5330	1469	5225	202003WED	1
16	5627	1478	6119	202003MON	1
17	5627	1773	5374	202004MON	1
18	5498	1774	6107	202004THU	1
19	5450	1778	5395	202004TUE	1
20	5477	1778	5511	202004SUN	1
21	5492	1778	6080	202004MON	1
22	5592	1779	5570	202004FRI	1
23	5333	1787	5307	202003SAT	1
24	5322	1787	5314	202003WED	1

MRE_Advert_FACT_L0

	PROPERTY_ID	ADVERT_ID	TIME_ID	NUMBER_OF_ADVERTS
1	22	16	202004TUE	1
2	28	16	202004WED	1
3	133	16	202004THU	1
4	140	16	202004WED	1
5	59	16	202004THU	1
6	2	16	202004THU	1
7	233	16	202004THU	1
8	237	16	202004MON	1
9	113	16	202003SUN	1
10	122	16	202003SUN	1
11	212	12	202004THU	1
12	213	12	202004SAT	1
13	218	16	202003SAT	1
14	221	16	202004WED	1
15	149	12	202003WED	1
16	91	23	202003MON	1
17	104	16	202004SUN	1
18	325	16	202003FRI	1
19	335	23	202003THU	1
20	336	16	202003TUE	1
21	246	16	202004SUN	1
22	253	16	202003MON	1
23	255	16	202004SUN	1
24	264	16	202003SAT	1

Task C.3

Simple Reports

Report 1

(a) The query questions written in English

Top 15 most rented property by scale and suburb.

(b) Your explanation on why such a query is necessary or useful for the management

This will allow the management to get the sentiment of the rental market.

(c) The SQL commands

```
SELECT *
     FROM
(SELECT   s.scale_description as Scale,
          a.suburb as Suburb,
          SUM(f.number_of_rent) as Number_of_Rents,
          ROW_NUMBER() OVER(ORDER BY SUM(f.number_of_rent)
          DESC) as RANK
     FROM    mre_rent_fact_l2    f,    mre_scale_dim_l2    s,
     mre_property_dim_l2 p, mre_address_dim_l2 a
          WHERE f.scale_id = s.scale_id
          AND f.property_id = p.property_id
          AND p.address_id = a.address_id
               GROUP BY s.scale_description, a.suburb
                    ORDER BY ROW_NUMBER() OVER(ORDER BY
                    SUM(f.number_of_rent) DESC) ASC)
          WHERE RANK <= 15;
```

(d) The screenshots of the query results (or part of the query results), including all the attribute names

	SCALE	SUBURB	NUMBER_OF_RENTS	RANK
1	small	Surfers Paradise	14	1
2	small	Kingston	12	2
3	small	Melbourne	10	3
4	extra small	City	10	4
5	extra small	Braddon	9	5
6	small	City	9	6
7	small	Brisbane City	8	7
8	extra small	St Kilda	8	8
9	extra small	Kingston	8	9
10	small	Adelaide	8	10
11	small	Manly	7	11
12	small	Collingwood	7	12
13	small	Griffith	7	13
14	extra small	Belconnen	7	14
15	extra small	Sydney	7	15

Report 2

(a) The query questions written in English

Top 15% sales based on time period and property type.

(b) Your explanation on why such a query is necessary or useful for the management

This might give the management an idea to focus on which suburb at what time to boast business performance.

(c) The SQL commands

```
SELECT *
    FROM (
SELECT    t.year as Year,
          t.month as Month,
          p.property_type as Property_Type,
          SUM(f.total_sales_price) as Total_Sales_Price,
          SUM(f.number_of_sales) as Number_of_Sales,
          PERCENT_RANK()         OVER         (ORDER         BY
          SUM(f.total_sales_price) DESC) as Revenue_Ranking
    FROM    mre_sale_fact_l2    f,    mre_property_dim_l2    p,
    mre_time_dim_l2 t
        WHERE f.time_id = t.time_id
            GROUP BY t.year, t.month, p.property_type)
        WHERE Revenue_Ranking >= 0.85
            ORDER BY Revenue_Ranking DESC;
```

(d) The screenshots of the query results (or part of the query results), including all the attribute names

	YEAR	MONTH	PROPERTY_TYPE	TOTAL_SALES_PRICE	NUMBER_OF_SALES	REVENUE_RANKING
1	2019	12	Vacant land	32265900	48	0.985507246376811594202898550724637681159 4
2	2019	12	Development Site	32265900	48	0.985507246376811594202898550724637681159 4
3	2019	12	Penthouse	48398850	72	0.971014492753623188405797101449275362318 8
4	2019	12	Block of Units	112930650	168	0.956521739130434782608695652173913043478 3
5	2020	4	Development Site	138551800	174	0.927536231884057971014492753623188405797 1
6	2020	4	Vacant land	138551800	174	0.927536231884057971014492753623188405797 1
7	2019	12	New House & Land	145196550	216	0.913043478260869565217391304347826086956 5
8	2020	4	Penthouse	207827700	261	0.898550724637681159420289855072463768115 9
9	2019	12	Terrace	225861300	336	0.884057971014492753623188405797101449275 4
10	2019	12	New Apartments / Off the Plan	338791950	504	0.869565217391304347826086956521739130434 8
11	2019	12	Studio	371057850	552	0.855072463768115942028985507246376811594 2

Report 3

(a) The query questions written in English

Total property visited by suburb and season.

(b) Your explanation on why such a query is necessary or useful for the management

This will give the management and insight of property investment of different locations and seasons

(c) The SQL commands

```
SELECT  t.year as Year,
        s.season_description as season,
        a.suburb as suburb,
        SUM(number_of_visits) as Number_of_Visits
    FROM mre_visit_fact_l0 f, mre_time_dim_l0 t,
    mre_season_dim_l0      s,      mre_property_dim_l0      p,
    mre_address_dim_l0 a
        WHERE f.time_id = t.time_id
        AND t.season_id = s.season_id
        AND f.property_id = p.property_id
        AND p.address_id = a.address_id
            GROUP BY t.year, s.season_description, a.suburb
               ORDER BY t.year, s.season_description,
               a.suburb;
```

(d) The screenshots of the query results (or part of the query results), including all the attribute names

	YEAR	SEASON	SUBURB	NUMBER_OF_VISITS
1	2020	Autumn	Albert Park	6
2	2020	Autumn	Annandale	6
3	2020	Autumn	Armadale	4
4	2020	Autumn	Balaclava	6
5	2020	Autumn	Barton	1
6	2020	Autumn	Belconnen	20
7	2020	Autumn	Benowa	9
8	2020	Autumn	Braddon	8
9	2020	Autumn	Brighton	16
10	2020	Autumn	Broadbeach	9
11	2020	Autumn	Broadbeach Waters	11
12	2020	Autumn	Brunswick	1
13	2020	Autumn	Bushland Beach	2
14	2020	Autumn	Campbell	2
15	2020	Autumn	Carnegie	1
16	2020	Autumn	Caulfield North	1

Reports with proper sub-totals

Report 4

(a) The query questions written in English

What are the sub-total and total rental fees from each suburb, time period, and property type?

(b) Your explanation on why such a query is necessary or useful for the management

This will give the management an idea of different combinations between suburb, time period and property type.

(c) The SQL commands

```
SELECT     t.year||t.month as Time_Period,
           a.suburb as Suburb,
           p.property_type as Property_Type,
           to_char(SUM(f.total_rent_fee), '9,999,999,999.99')
           as Rental_Fees,
           DECODE(GROUPING(t.year||t.month), 1, 'All Periods',
           t.year||t.month) as Period,
           DECODE(GROUPING(a.suburb), 1, 'All Suburbs',
           a.suburb) as Suburbs,
           DECODE(GROUPING(p.property_type), 1, 'All Types',
           p.property_type) as Types
      FROM     mre_rent_fact_l2     f,     mre_time_dim_l2     t,
      mre_property_dim_l2 p, mre_address_dim_l2 a
         WHERE f.time_id = t.time_id
         AND f.property_id = p.property_id
         AND p.address_id = a.address_id
           GROUP      BY      CUBE(t.year||t.month,      a.suburb,
           p.property_type);
```

(d) The screenshots of the query results (or part of the query results), including all the attribute names

	TIME_PERIOD	SUBURB	PROPERTY_TYPE	RENTAL_FEES	PERIOD	SUBURBS	TYPES
1	(null)	(null)	(null)	15,328,750.14	All Periods	All Suburbs	All Types
2	(null)	(null)	House	5,865,353.86	All Periods	All Suburbs	House
3	(null)	(null)	Villa	99,180.00	All Periods	All Suburbs	Villa
4	(null)	(null)	Duplex	150,737.14	All Periods	All Suburbs	Duplex
5	(null)	(null)	Studio	38,047.14	All Periods	All Suburbs	Studio
6	(null)	(null)	Terrace	124,340.71	All Periods	All Suburbs	Terrace
7	(null)	(null)	Penthouse	12,071.43	All Periods	All Suburbs	Penthouse
8	(null)	(null)	Townhouse	1,367,739.29	All Periods	All Suburbs	Townhouse
9	(null)	(null)	Semi-Detached	43,024.29	All Periods	All Suburbs	Semi-Detached
10	(null)	(null)	Apartment / Unit / Flat	7,606,176.29	All Periods	All Suburbs	Apartment / Unit / Flat
11	(null)	(null)	New Apartments / Off the Plan	22,080.00	All Periods	All Suburbs	New Apartments / Off the Plan
12	(null)	City	(null)	273,965.71	All Periods	City	All Types
13	(null)	City	House	12,000.00	All Periods	City	House
14	(null)	City	Apartment / Unit / Flat	261,965.71	All Periods	City	Apartment / Unit / Flat
15	(null)	Cook	(null)	12,480.00	All Periods	Cook	All Types
16	(null)	Cook	House	12,480.00	All Periods	Cook	House
17	(null)	Holt	(null)	27,918.57	All Periods	Holt	All Types
18	(null)	Holt	House	13,278.57	All Periods	Holt	House
19	(null)	Holt	Duplex	14,640.00	All Periods	Holt	Duplex
20	(null)	Lota	(null)	9,600.00	All Periods	Lota	All Types

Report 5

(a) The query questions written in English

What are the sub-total and total rental fees from each suburb, time period, and property type?

(b) Your explanation on why such a query is necessary or useful for the management

This will give the management an idea of different combinations between suburb, time period and property type.

(c) The SQL commands

```
SELECT    t.year||t.month as Time_Period,
          a.suburb as Suburb,
          p.property_type as Property_Type,
          to_char(SUM(f.total_rent_fee), '9,999,999,999.99')
          as Rental_Fees,
          DECODE(GROUPING(t.year||t.month), 1, 'All Periods',
          t.year||t.month) as Period,
          DECODE(GROUPING(a.suburb), 1, 'All Suburbs',
          a.suburb) as Suburbs,
          DECODE(GROUPING(p.property_type), 1, 'All Types',
          p.property_type) as Types
     FROM    mre_rent_fact_l2    f,    mre_time_dim_l2    t,
     mre_property_dim_l2 p, mre_address_dim_l2 a
          WHERE f.time_id = t.time_id
          AND f.property_id = p.property_id
          AND p.address_id = a.address_id
               GROUP BY a.suburb, CUBE(t.year||t.month,
               p.property_type);
```

(d) The screenshots of the query results (or part of the query results), including all the attribute names

	TIME_PERIOD	SUBURB	PROPERTY_TYPE	RENTAL_FEES	PERIOD	SUBURBS	TYPES
1	(null)	City	(null)	273,965.71	All Periods	City	All Types
2	(null)	City	House	12,000.00	All Periods	City	House
3	(null)	City	Apartment / Unit / Flat	261,965.71	All Periods	City	Apartment / Unit / Flat
4	20201	City	(null)	30,000.00	20201	City	All Types
5	20201	City	Apartment / Unit / Flat	30,000.00	20201	City	Apartment / Unit / Flat
6	20202	City	(null)	60,600.00	20202	City	All Types
7	20202	City	House	12,000.00	20202	City	House
8	20202	City	Apartment / Unit / Flat	48,600.00	20202	City	Apartment / Unit / Flat
9	20203	City	(null)	80,760.00	20203	City	All Types
10	20203	City	Apartment / Unit / Flat	80,760.00	20203	City	Apartment / Unit / Flat
11	20204	City	(null)	91,085.71	20204	City	All Types
12	20204	City	Apartment / Unit / Flat	91,085.71	20204	City	Apartment / Unit / Flat
13	201912	City	(null)	11,520.00	201912	City	All Types
14	201912	City	Apartment / Unit / Flat	11,520.00	201912	City	Apartment / Unit / Flat
15	(null)	Cook	(null)	12,480.00	All Periods	Cook	All Types
16	(null)	Cook	House	12,480.00	All Periods	Cook	House
17	201912	Cook	(null)	12,480.00	201912	Cook	All Types
18	201912	Cook	House	12,480.00	201912	Cook	House
19	(null)	Holt	(null)	27,918.57	All Periods	Holt	All Types
20	(null)	Holt	House	13,278.57	All Periods	Holt	House

Report 6

(a) The query questions written in English

What is the sub-total and total sale revenue from each state and time period for houses?

(b) Your explanation on why such a query is necessary or useful for the management

This will give the management an understanding of sale of one of their best sellers, houses, at different states and time periods.

(c) The SQL commands

```
SELECT     t.year||t.month as Time_period,
           st.state_name as State,
           SUM(s.total_sales_price) as Total_Revenue,
           DECODE(GROUPING(t.year||t.month), 1, 'All Periods',
           t.year||t.month) as Periods,
           DECODE(GROUPING(st.state_name), 1, 'All States',
           st.state_name) as States
      FROM mre_sale_fact_l2 s, mre_property_dim_l2 p,
      mre_address_dim_l2 a, mre_postcode_dim_l2 pc,
      mre_state_dim_l2 st, mre_time_dim_l2 t
           WHERE s.property_id = p.property_id
           AND p.address_id = a.address_id
           AND a.postcode = pc.postcode
           AND pc.state_code = st.state_code
           AND s.time_id = t.time_id
           AND p.property_type = 'House'
                GROUP BY ROLLUP (t.year||t.month,
                st.state_name);
```

(d) The screenshots of the query results (or part of the query results), including all the attribute names

	TIME_PERIOD	STATE	TOTAL_REVENUE	PERIODS	STATES
1	20201	Tasmania	1010000	20201	Tasmania
2	20201	Victoria	42626999	20201	Victoria
3	20201	Queensland	38936950	20201	Queensland
4	20201	New South Wales	14363950	20201	New South Wales
5	20201	South Australia	8336000	20201	South Australia
6	20201	Western Australia	9330000	20201	Western Australia
7	20201	Australian Capital Territory	18139000	20201	Australian Capital Territory
8	20201	(null)	132742899	20201	All States
9	20202	Tasmania	625000	20202	Tasmania
10	20202	Victoria	28712500	20202	Victoria
11	20202	Queensland	34699950	20202	Queensland
12	20202	New South Wales	22253000	20202	New South Wales
13	20202	South Australia	6567000	20202	South Australia
14	20202	Western Australia	10705000	20202	Western Australia
15	20202	Australian Capital Territory	12001000	20202	Australian Capital Territory
16	20202	(null)	115563450	20202	All States
17	20203	Tasmania	750000	20203	Tasmania
18	20203	Victoria	38661000	20203	Victoria
19	20203	Queensland	75689000	20203	Queensland
20	20203	New South Wales	13891950	20203	New South Wales

Report 7

(a) The query questions written in English

What is the sub-total and total sale revenue from each state and time period for houses?

(b) Your explanation on why such a query is necessary or useful for the management

This will give the management an understanding of sale of one of their best sellers, houses, at different states and time periods.

(c) The SQL commands

```
SELECT     t.year||t.month as Time_period,
           st.state_name as State,
           SUM(s.total_sales_price) as Total_Revenue,
           DECODE(GROUPING(t.year||t.month), 1, 'All Periods',
           t.year||t.month) as Periods,
           DECODE(GROUPING(st.state_name), 1, 'All States',
           st.state_name) as States
      FROM    mre_sale_fact_l2    s,    mre_property_dim_l2    p,
      mre_address_dim_l2     a,     mre_postcode_dim_l2     pc,
      mre_state_dim_l2 st, mre_time_dim_l2 t
           WHERE s.property_id = p.property_id
           AND p.address_id = a.address_id
           AND a.postcode = pc.postcode
           AND pc.state_code = st.state_code
           AND s.time_id = t.time_id
           AND p.property_type = 'House'
                GROUP BY st.state_name, ROLLUP
                (t.year||t.month);
```

(d) The screenshots of the query results (or part of the query results), including all the attribute names

	TIME_PERIOD	STATE	TOTAL_REVENUE	PERIODS	STATES
1	20201	Tasmania	1010000	20201	Tasmania
2	20202	Tasmania	625000	20202	Tasmania
3	20203	Tasmania	750000	20203	Tasmania
4	20204	Tasmania	500000	20204	Tasmania
5	(null)	Tasmania	2885000	All Periods	Tasmania
6	20201	Victoria	42626999	20201	Victoria
7	20202	Victoria	28712500	20202	Victoria
8	20203	Victoria	38661000	20203	Victoria
9	20204	Victoria	9785000	20204	Victoria
10	201912	Victoria	2474000	201912	Victoria
11	(null)	Victoria	122259499	All Periods	Victoria
12	20201	Queensland	38936950	20201	Queensland
13	20202	Queensland	34699950	20202	Queensland
14	20203	Queensland	75689000	20203	Queensland
15	20204	Queensland	18552000	20204	Queensland
16	201912	Queensland	3066950	201912	Queensland
17	(null)	Queensland	170944850	All Periods	Queensland
18	20201	New South Wales	14363950	20201	New South Wales
19	20202	New South Wales	22253000	20202	New South Wales
20	20203	New South Wales	13891950	20203	New South Wales

Reports with moving and cumulative aggregates

Report 8

a) **The query questions written in English**
What is the total number of clients and cumulative number of clients with a high budget in each year?

b) **Your explanation on why such a query is necessary or useful for the management**
This will give the management an understanding of how high budget clients have chosen MonRE as their preferred agency.

c) **The SQL commands**

```
SELECT     year,
           SUM(total_clients) as Number_of_Clients,
           SUM(SUM(total_clients)) OVER (ORDER BY year ROWS
           UNBOUNDED PRECEDING) as Cumulative_Total
      FROM
(SELECT *
      FROM
(SELECT t.year, SUM(f.number_of_clients) as total_clients
      FROM    mre_client_fact_l0    f,    mre_budget_dim_l0    b,
      mre_rent_fact_l0 rf, mre_time_dim_l0 t
           WHERE f.budget_id = b.budget_id
           AND f.client_person_id = rf.client_person_id
           AND rf.rent_start_date = t.time_id
           AND b.budget_description LIKE 'High%'
                GROUP BY t.year)
UNION
(SELECT t.year, SUM(f.number_of_clients) as total_clients
      FROM    mre_client_fact_l0    f,    mre_budget_dim_l0    b,
      mre_sale_fact_l0 sf, mre_time_dim_l0 t
           WHERE f.budget_id = b.budget_id
           AND f.client_person_id = sf.client_person_id
           AND sf.time_id = t.time_id
           AND b.budget_description LIKE 'High%'
                GROUP BY t.year))
           GROUP BY year
                ORDER BY year;
```

d) **The screenshots of the query results (or part of the query results), including all the attribute names**

	YEAR	NUMBER_OF_CLIENTS	CUMULATIVE_TOTAL
1	2019	23	23
2	2020	892	915

Report 9

(a) The query questions written in English

What is the total monthly number of visits and 3-month average number of visits?

(b) Your explanation on why such a query is necessary or useful for the management

Management may want to know more about why are there more visits in certain months. They can then drill down to find out which properties are having many visits from different visiting clients.

(c) The SQL commands

```
SELECT     t.year,
           t.month,
           SUM(number_of_visit) as Number_of_Visits,
           to_char(AVG(SUM(f.number_of_visit)) OVER (ORDER BY
           t.year, t.month ROWS 2 PRECEDING), '999,999') as
           Average_3_Month_Visits
      FROM mre_visit_fact_l2 f, mre_time_dim_l2 t
           WHERE f.visit_time_id =t.time_id
                GROUP BY t.year, t.month
                     ORDER BY year, month +0;
```

(d) The screenshots of the query results (or part of the query results), including all the attribute names

	YEAR	MONTH	NUMBER_OF_VISITS	AVERAGE_3_MONTH_VISITS
1	2020	3	430	430
2	2020	4	144	287

Report 10

(a) The query questions written in English

Cumulative monthly total number of rents based on properties that have been rented to clients.

(b) Your explanation on why such a query is necessary or useful for the management

Management would have a good indication of cash flow coming from rentals. They can use this information to make decisions on whether to increase spending on attracting tenants.

(c) The SQL commands

```
SELECT     t.year as Year,
           t.month as Month,
           SUM(Number_of_Rent) as Number_of_Rents,
           SUM(SUM(number_of_rent)) OVER (ORDER BY t.year,
           t.month ROWS UNBOUNDED PRECEDING) as
           Cumulative_Number_of_Rent
      FROM mre_rent_fact_l2 f, mre_time_dim_l2 t
           WHERE f.time_id = t.time_id
                GROUP BY t.year, t.month
                     ORDER BY year, month +0;
```

(d) The screenshots of the query results (or part of the query results), including all the attribute names

	YEAR	MONTH	NUMBER_OF_RENTS	CUMULATIVE_NUMBER_OF_RENT
1	2019	12	15	15
2	2020	1	217	232
3	2020	2	167	399
4	2020	3	221	620
5	2020	4	414	1034
6	2020	5	82	1116

Reports with Partitions

Report 11

(a) The query questions written in English

Show 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.

(b) Your explanation on why such a query is necessary or useful for the management

This information might to useful to management to see the property type with most sales as well as the state with most sales of each type.

(c) The SQL commands

```
SELECT     t.year as Year,
           p.property_type as Property_Type,
           s.state_name as State,
           SUM(f.number_of_sales) as Total_Number_of_Sales,
           RANK() OVER (PARTITION BY t.year ORDER BY
           SUM(f.number_of_sales) DESC) as RANK_BY_YEAR,
           RANK() OVER (PARTITION BY s.state_name ORDER BY
           SUM(f.number_of_sales) DESC) as RANK_BY_STATE
      FROM mre_sale_fact_l2 f, mre_property_dim_l2 p,
      mre_time_dim_l2       t,       mre_address_dim_l2       a,
      mre_postcode_dim_l2 pc, mre_state_dim_l2 s
           WHERE f.property_id = p.property_id
           AND f.time_id = t.time_id
           AND p.address_id = a.address_id
           AND a.postcode = pc.postcode
           AND pc.state_code = s.state_code
                GROUP BY t.year, p.property_type, s.state_name
                     ORDER BY SUM(f.number_of_sales) DESC;
```

(d) The screenshots of the query results (or part of the query results), including all the attribute names

	YEAR	PROPERTY_TYPE	STATE	TOTAL_NUMBER_OF_SALES	RANK_BY_YEAR	RANK_BY_STATE
1	2020	House	Queensland	202	1	1
2	2020	House	Victoria	121	2	1
3	2020	Apartment / Unit / Flat	Queensland	103	3	2
4	2020	House	New South Wales	61	4	1
5	2020	Apartment / Unit / Flat	Victoria	53	5	2
6	2020	House	Australian Capital Territory	50	6	1
7	2020	House	Western Australia	49	7	1
8	2020	Apartment / Unit / Flat	Australian Capital Territory	43	8	2
9	2020	Apartment / Unit / Flat	New South Wales	41	9	2
10	2020	House	South Australia	38	10	1
11	2020	Townhouse	Victoria	23	11	3
12	2020	Townhouse	Queensland	20	12	3
13	2020	Townhouse	Australian Capital Territory	17	13	3
14	2020	Apartment / Unit / Flat	Western Australia	13	14	2
15	2020	Apartment / Unit / Flat	South Australia	7	15	2
16	2019	House	Queensland	6	1	4

Report 12

(a) The query questions written in English

Show ranking of each advertisement type based on the yearly total number of adverts and the ranking of each state based on the yearly total number of adverts.

(b) Your explanation on why such a query is necessary or useful for the management

This information might be useful to management to compare the number of advertisements of different types and different states to observe sale patterns of certain areas.

(c) The SQL commands

```
SELECT     t.year as Year,
           ad.advert_name as Advertisement_Type,
           s.state_name as State,
           SUM(f.number_of_adverts) as
           Yearly_Total_Number_of_Adverts,
           RANK() OVER (PARTITION BY ad.advert_name ORDER BY
           SUM(f.number_of_adverts) DESC) as
           RANK_BY_ADVERT_TYPE,
           RANK() OVER (PARTITION BY s.state_name ORDER BY
           SUM(f.number_of_adverts) DESC) as RANK_BY_STATE
      FROM  mre_advert_fact_l0  f,  mre_advertisement_dim_l0  ad,
      mre_property_dim_l0      p,      mre_address_dim_l0      a,
      mre_postcode_dim_l0      pc,      mre_state_dim_l0      s,
      mre_time_dim_l0 t
           WHERE f.advert_id = ad.advert_id
           AND f.property_id = p.property_id
           AND p.address_id = a.address_id
           AND a.postcode = pc.postcode
           AND pc.state_code = s.state_code
           AND f.time_id = t.time_id
                GROUP BY t.year, ad.advert_name, s.state_name
                     ORDER BY SUM(f.number_of_adverts) DESC;
```

(d) The screenshots of the query results (or part of the query results), including all the attribute names

	YEAR	ADVERTISEMENT_TYPE	STATE	YEARLY_TOTAL_NUMBER_OF_ADVERTS	RANK_BY_ADVERT_TYPE	RANK_BY_STATE
1	2020	Sale House	Queensland	447	1	1
2	2020	Rent Apartment / Unit / Flat	New South Wales	312	1	1
3	2020	Sale House	Victoria	300	2	1
4	2020	Rent Apartment / Unit / Flat	Victoria	252	2	2
5	2020	Rent House	Queensland	198	1	2
6	2020	Sale Apartment / Unit / Flat	Queensland	178	1	3
7	2020	Rent Apartment / Unit / Flat	Australian Capital Territory	145	3	1
8	2020	Rent Apartment / Unit / Flat	Queensland	144	4	4
9	2020	Sale House	South Australia	128	3	1
10	2020	Sale House	New South Wales	125	4	2
11	2020	Rent House	Victoria	122	2	3
12	2020	Sale Apartment / Unit / Flat	Victoria	121	2	4
13	2020	Sale House	Western Australia	113	5	1
14	2020	Sale Apartment / Unit / Flat	Australian Capital Territory	110	3	2
15	2020	Sale House	Australian Capital Territory	109	6	3
16	2020	Sale Apartment / Unit / Flat	New South Wales	85	4	3

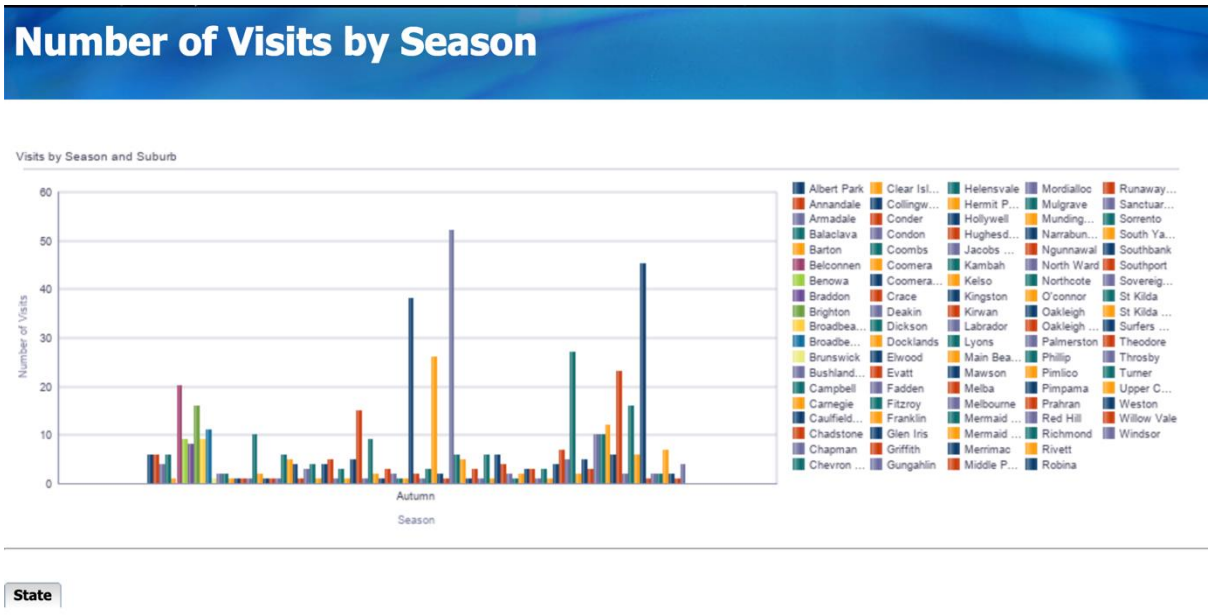
Task 4

Report 2

Top 15% sales based on time period and property type.

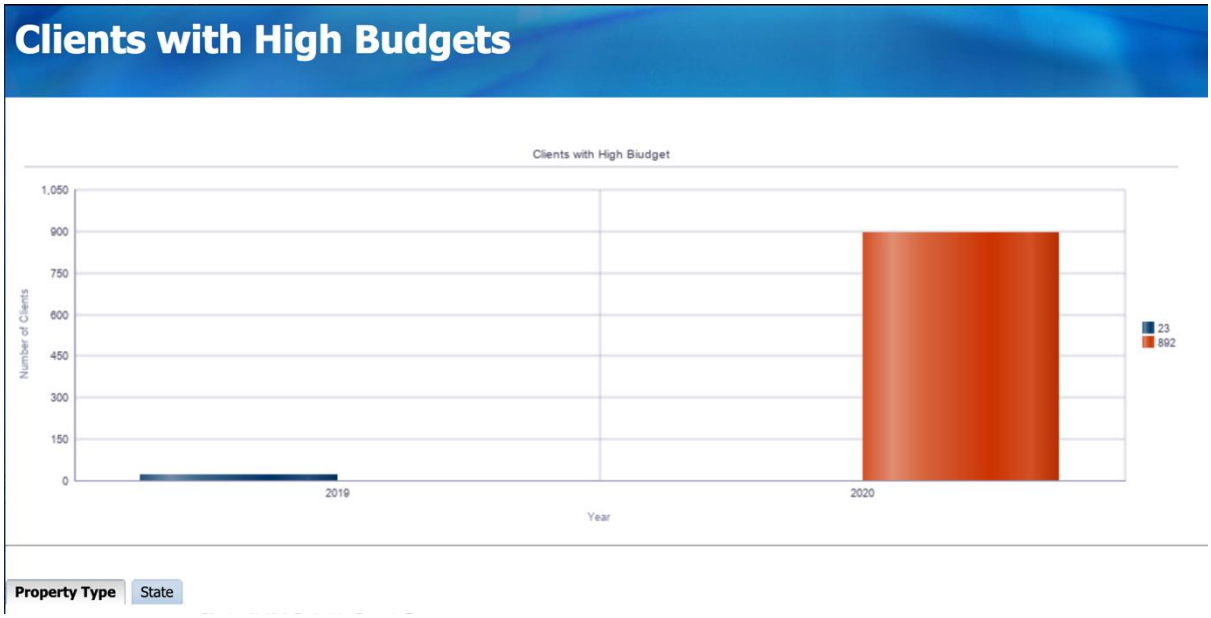
Report 3

Total property visited by suburb and season.



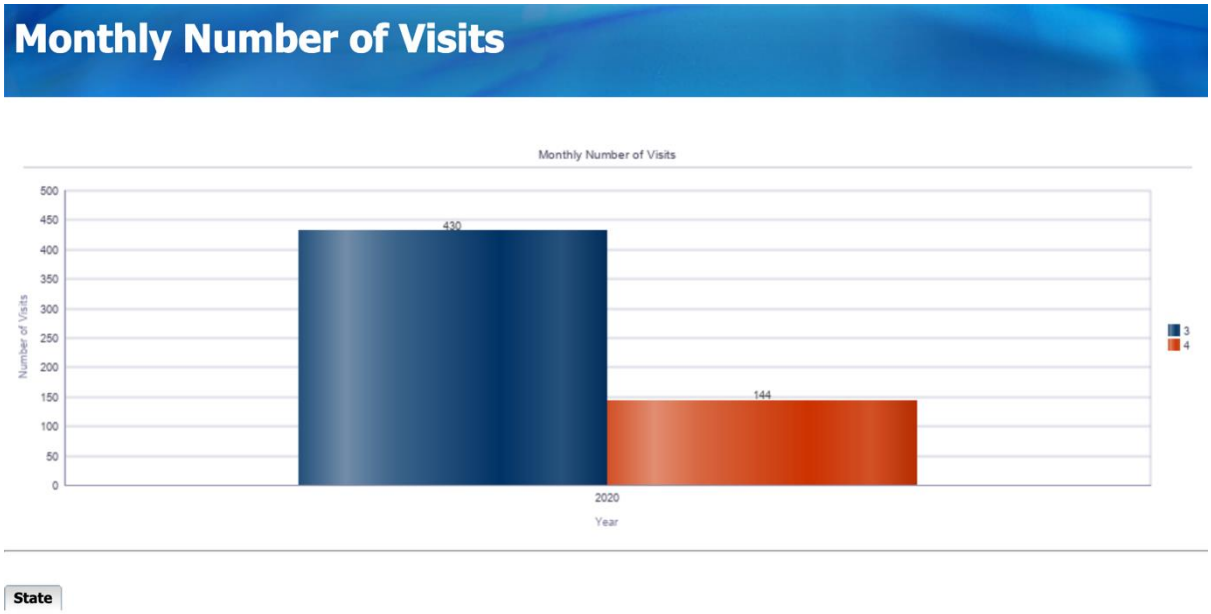
Report 8

What is the total number of clients and cumulative number of clients with a high budget in each year?



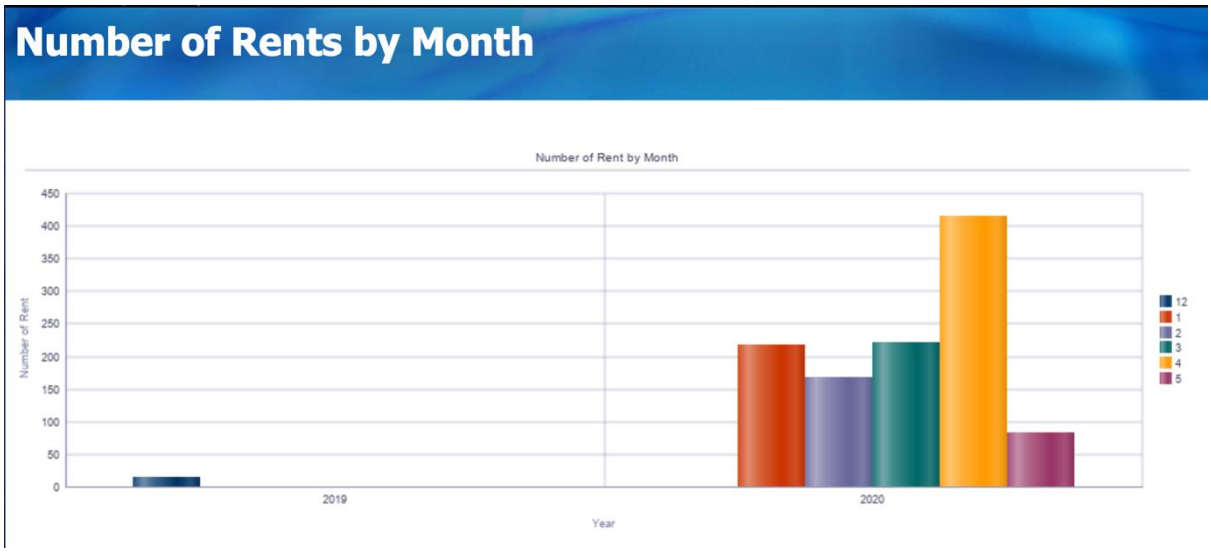
Report 9

What is the total monthly number of visits and 3-month average number of visits?



Report 10

Cumulative monthly total number of rents based on properties that have been rented to clients



Assumptions

- Given that the property scale given contained overlapping categories, the following categorisation was used for property scale:
    - <=1 Extra Small
    - 2-3 Small
    - 4-6 Medium
    - 7-10 Large
    - >10 Extra Large