



6/15/2020

FIT5195 – Business Intelligence and Data Warehousing

Semester 1, 2020

Major Assignment



Submitted by:
Abhilash Kale 30254140
Manali Choudhary 30151198
Vishal Gnanasekar 30382300

Tutorial: 14 & 22
Tutor Name: Agnes Haryanto
Arif Hidayat

GROUP ASSIGNMENT COVER SHEET

Student ID Number	Surname	Given Names
30254140	Kale	Abhilash Anil
30151198	Choudhary	Manali Prakash
30382300	Gnanasekar	Vishal

* Please include the names of all other group members.

Unit name and code	BUSINESS INTELLIGENCE AND DATA WAREHOUSING FIT5195	
Title of assignment	Major Assignment	
Lecturer/tutor	Agnes Haryanto, Arif Hidayat	
Tutorial day and time	Wed, 12 PM & Mon, 2 PM	Campus CAULFIELD
Is this an authorised group assignment?	<input type="checkbox"/> Yes <input type="checkbox"/> No	
Has any part of this assignment been previously submitted as part of another unit/course?		<input type="checkbox"/> Yes
<input type="checkbox"/> No		
Due Date: 16 th June 2020	Date submitted: 15 th June 2020	

All work must be submitted by the due date. If an extension of work is granted this must be specified with the signature of the lecturer/tutor.

Extension granted until (date) **Signature of lecturer/tutor**

.....

Please note that it is your responsibility to retain copies of your assessments.

Intentional plagiarism or collusion amounts to cheating under Part 7 of the Monash University (Council) Regulations

Plagiarism: Plagiarism means taking and using another person's ideas or manner of expressing them and passing them off as one's own. For example, by failing to give appropriate acknowledgement. The material used can be from any source (staff, students or the internet, published and unpublished works).

Collusion: Collusion means unauthorised collaboration with another person on assessable written, oral or practical work and includes paying another person to complete all or part of the work.

Where there are reasonable grounds for believing that intentional plagiarism or collusion has occurred, this will be reported to the Associate Dean (Education) or delegate, who may disallow the work concerned by prohibiting assessment or refer the matter to the Faculty Discipline Panel for a hearing.

Student Statement:

- I have read the university's Student Academic Integrity [Policy](#) and [Procedures](#).
- I understand the consequences of engaging in plagiarism and collusion as described in Part 7 of the Monash University (Council) Regulations <http://adm.monash.edu/legal/legislation/statutes>
- I have taken proper care to safeguard this work and made all reasonable efforts to ensure it could not be copied.
- No part of this assignment has been previously submitted as part of another unit/course.
- I acknowledge and agree that the assessor of this assignment may for the purposes of assessment, reproduce the assignment and:

i.provide to another member of faculty and any external marker; and/or

i.submit it to a text matching software; and/or

i.submit it to a text matching software which may then retain a copy of the assignment on its database for the purpose of future plagiarism checking.

- I certify that I have not plagiarised the work of others or participated in unauthorised collaboration when preparing this assignment.

Signature **Date**

* delete (iii) if not applicable

Signature: Abhilash Anil Kale

Date: 15 June 2020

Signature: Manali Prakash Choudhary

Date: 15 June 2020

Signature: Vishal Gnanasekar

Date: 15 June 2020

ORACLE ACCOUNT DETAILS

Abhilash Anil Kale

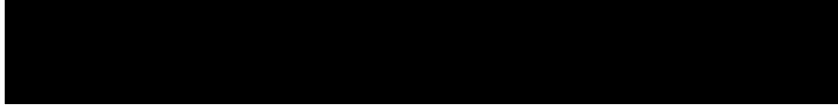
Username - S30254140
Password - student

Manali Prakash Choudhary

Username - S30151198
Password - student

Vishal Gnanasekar

Username - S30382300
Password - student



Please fill in the form with the contribution from each student towards the assignment.

1 NAME AND CONTRIBUTION DETAILS

Student ID	Student Name	Contribution Percentage
30254140	Abhilash Anil Kale	33.33%
30151198	Manali Prakash Choudhary	33.33%
30382300	Vishal Gnanasekar	33.33%

2 DECLARATION

We declare that:

- The information we have supplied in or with this form is complete and correct.
- We understand that the information we have provided in this form will be used for individual assessment of the assignment.

3 SIGNATURES

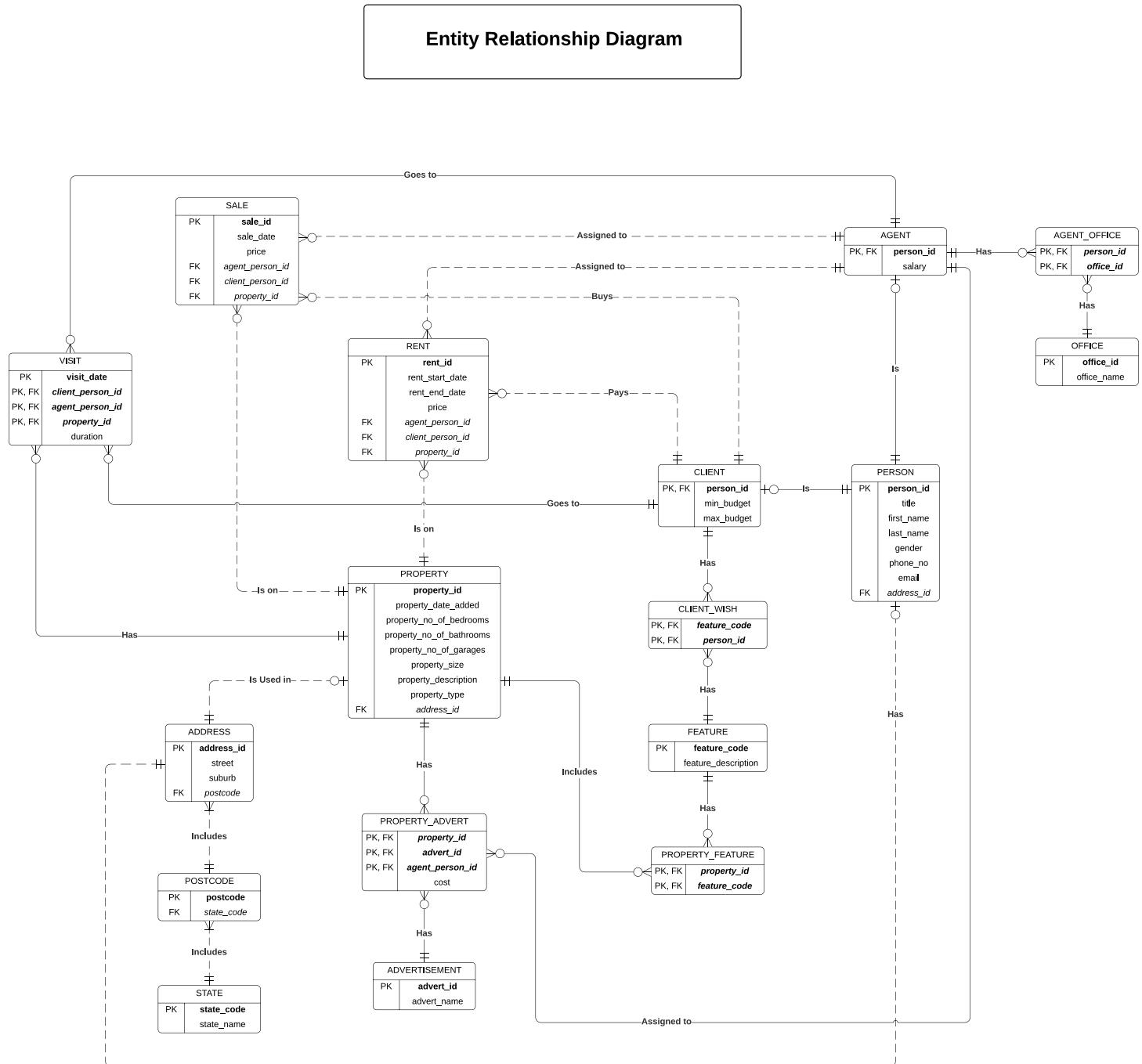
Signature

Date

15 June 2020

TASK C.1 Design a data warehouse for the above MonRE database

a. E/R diagram of operational database



b. Data cleaning

We've copied the operational database to our local database for cleaning the duplicate records. The below section includes the screenshots of the data error identified.

1. **Error:** Property table in the operational database had 20 duplicate entries for 2 property_id 6177, 6179 respectively.

```
select property_id, COUNT(*) from property group by property_id having count(*) > 1;
```

Query Result x

SQL | All Rows Fetched: 2 in 0.03 seconds

PROPERTY_ID	COUNT(*)
1 6177	4
2 6179	16

Strategy used: Use of "distinct * " to copy the table from operational database avoids the duplicate entries during the table creation itself in our database.

Query:

```
create table property as
select distinct * from monre.property;
```

2. **Error:** Person table in the operational database had 4 duplicate entries for the person id 6995.

```
83 select person_id, count(*)
84   from person
85   group by person_id
86   having count(*) > 1;
```

Script Output x Query Result x

SQL | All Rows Fetched

PERSON_ID	COUNT(*)
1 6995	4

Strategy used: Use of "distinct * " to copy the table from operational database avoids the duplicate entries during the table creation itself in our database.

Query:

```
create table person as
select distinct * from monre.person;
```

3. **Error:** The state_code table has a null value which does not provide any information.

<code>select * from state where state_code is null;</code>
Query Result x SQL All Rows Fetched: 9 in 0.028 seconds

STATE_CODE
1 QLD
2 SA
3 NSW
4 (null)
5 NT
6 WA
7 ACT
8 TAS
9 VIC

Strategy Used: Removal of null values that do not contribute any value to the State table after careful consideration of the states in Australia and the operational database entries.

Query:

```
delete
from state
where state_code is null;
```

After Cleaning:

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

4. **Error:** Agent table includes incorrect entries, 2 entries for negative salary and 1 entry with 0 salary.

<code>select * from agent</code>
<code>where salary <= 0;</code>

Query Result x
SQL All Rows Fetched: 3 in 0.292 seconds

PERSON_ID	SALARY
1	-100000
2	-120000
3	0

Strategy used: Deletion of the incorrect(negative) entries as the salary cannot be assumed but after careful consideration of the case study, the agent with salary “0” had been kept back assuming it to be an intern, but due to (Error: 6), it was removed by combining it in a single query as given.

Query:

```
delete
from agent
where salary <=0;
```

5. **Error:** person_id: 6997 in agent table is not present in the person table. This is an illegal/incorrect entry.

```
59 | select * from agent
60 | where person_id NOT IN
61 | (select person_id from person);
```

Script Output x Query Result x

SQL | All Rows Fetched: 1 in

	PERSON_ID	SALARY
1	6997	0

Strategy used: person_id: 6997 was cleaned by combining it in a single query as given in the previous data cleaning query itself (Error 4).

6. **Error:** person_id 6997 in agent_office table is not present in the person table.

```
229 | select * from agent_office
230 | where person_id NOT IN
231 | (select person_id from agent);
```

Query Result x

SQL | All Rows Fetched: 1 in

	PERSON_ID	OFFICE_ID
1	6997	1177

Strategy Used: person_id **6997** is already removed in agent table, hence it is meaningless for the person_id to be in agent_office table. Based on this, illegal/incorrect entry was deleted using the following query.

Query:

```
delete
from agent_office
where person_id not in
(select person_id from agent);
```

7. **Error:** Few records in client table had min_budget greater than max_budget which is practically not possible. 3 entries for person_id 5900, 5901, 5902 share the error.

```
53 | select * from client
54 | where min_budget > max_budget;
```

Script Output x Query Result x

SQL | All Rows Fetched: 3 in 0.026 sec

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

Strategy used: Deletion of the rows that had incorrect entries as there are no possibility of assuming the user's minimum and maximum budget.

Query:

```
delete
from client
where min_budget > max_budget;
```

8. **Error:** Client with person_id 5901 has a negative value for the max_budget.

```
126 | select * from client
127 | where max_budget <= 0;
```

All Rows Fetched: 1 in 0.035 sec

	PERSON_ID	MIN_BUDGET	MAX_BUDGET
1	5901	3500	-150

Strategy used: person_id 5901 is addressed in the previous cleaning process (**Error: 7**) which has been deleted in the previous cleaning query.

9. **Error:** person_id in the client table does not exist in the person table, which is an incorrect/illegal entry.

```
63 | select * from client
64 | where person_id NOT IN
65 | (select person_id from person);
```

All Rows Fetched: 1 in 0.025 sec

	PERSON_ID	MIN_BUDGET	MAX_BUDGET
1	7000	8500	15050

Strategy used: Deletion of the person_id that does not exist in the client table.

Query:

```
delete
from client
where person_id NOT IN
(select person_id from person);
```

10. **Error:** The visit table has client_person_id that does not exist in the client table. In this case client_person_id 6000 is incorrect/illegal entry

```
42 | select * from visit
43 | where client_person_id NOT IN
44 | (select person_id from client);
```

All Rows Fetched: 1 in 0.028 seconds

	CLIENT_PERSON_ID	AGENT_PERSON_ID	PROPERTY_ID	VISIT_DATE	DURATION
1	6000	6001	5741	31-12-2999 00:00:00	5

Strategy Used: Deletion of the entry from the visit table where person_id does not exist in the client table.

Query:

```
delete
from visit
where agent_person_id NOT IN
(select person_id from agent);
```

11. **Error:** The visit table has agent_person_id that does not exist in the agent table. In this case agent_person_id 6001 is an incorrect/illegal entry.

```
38 | select * from visit
39 | where agent_person_id NOT IN
40 | (select person_id from agent);
```

Query Result x

SQL | All Rows Fetched: 1 in 0.029 seconds

	CLIENT_PERSON_ID	AGENT_PERSON_ID	PROPERTY_ID	VISIT_DATE	DURATION
1	6000	6001	5741	31-12-2999 00:00:00	5

Strategy used: Removed from the previous cleaning process using the single data cleaning query. (**Error: 10).**

12. **Error:** Jargon data entry error - The visit data for the client_id 6000 has a jargon date that does not provide the any date value.

```
104 | select * from visit
105 | where visit_date > TO_DATE('2020', 'YYYY');
```

Query Result x

SQL | All Rows Fetched: 1 in 0.093 seconds

	CLIENT_PERSON_ID	AGENT_PERSON_ID	PROPERTY_ID	VISIT_DATE	DURATION
1	6000	6001	5741	31-12-2999 00:00:00	5

Strategy used: Removed from the previous cleaning process using the single data cleaning query. (**Error: 10).**

13. **Error:** Person table has illegal/incorrect address_id which does not exist in the Address table.

```
725 | select * from person
726 | where address_id NOT IN
727 | (select address_id from address);
```

Script Output x | Query Result x

SQL | All Rows Fetched: 1 in 0.053 seconds

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

Strategy used: Deletion of the entry with person_id 7001 as it has incorrect/illegal address_id.

Query:

```
delete
from person
where address_id NOT IN
(select address_id from address);
```

14. **Error:** Rent table has an illegal/incorrect entry for the client_person_id: 6001 that does not exist in person table.

```
52 | select * from rent
53 | where client_person_id NOT IN
54 | (select person_id from client);
```

Query Result x

SQL | All Rows Fetched: 1 in 0.03 seconds

	RENT_ID	AGENT_PERSON_ID	CLIENT_PERSON_ID	PROPERTY_ID	RENT_START_DATE	RENT_END_DATE	PRICE
1	3284	6002	6001	5741	31-12-2021 00:00:00	01-06-2019 00:00:00	500

Strategy used: Deletion of the entry as the client_person_id not in the person table for rent_id 3284.

Query:

```
delete
from rent
where client_person_id NOT IN
(select person_id from client);
```

15. Error: Rent table has an illegal/incorrect entry for the agent_person_id: 6002 that do not exist in the person table.

```
48 | select * from rent
49 | where agent_person_id NOT IN
50 | (select person_id from agent);
▼
```

Query Result x

SQL | All Rows Fetched: 1 in 0.031 seconds

RENT_ID	AGENT_PERSON_ID	CLIENT_PERSON_ID	PROPERTY_ID	RENT_START_DATE	RENT_END_DATE	PRICE
1	3284	6002	6001	5741 31-12-2021 00:00:00	01-06-2019 00:00:00	500

Strategy used: The agent_person_id: 6002 is addressed in the previous cleaning process (**Error: 14**) that involves the same rent_id: 3284, so the record was removed in the previous data cleaning query.

16. Error: Rent_end_date is prior to rent_start_date which is not practically possible.

```
97 | select *
98 | from rent
99 | where rent_end_date < rent_start_date;
▼
```

Query Result x

SQL | All Rows Fetched: 1 in 0.275 seconds

RENT_ID	AGENT_PERSON_ID	CLIENT_PERSON_ID	PROPERTY_ID	RENT_START_DATE	RENT_END_DATE	PRICE
1	3284	6002	6001	5741 31-12-2021 00:00:00	01-06-2019 00:00:00	500

Strategy used: rent_id: 3284 was addressed in the previous cleaning process (**Error: 14**) that has been already removed.

SQL Queries for copying and cleaning the database from MonRE:

```
DROP TABLE address CASCADE CONSTRAINTS PURGE;  
CREATE TABLE address AS  
SELECT * FROM monre.address;
```

```
DROP TABLE advertisement CASCADE CONSTRAINTS PURGE;  
CREATE TABLE advertisement AS  
SELECT * FROM monre.advertisement;
```

```
DROP TABLE agent CASCADE CONSTRAINTS PURGE;  
CREATE TABLE agent AS  
SELECT * FROM monre.agent;
```

```
DROP TABLE agent_office CASCADE CONSTRAINTS PURGE;  
CREATE TABLE agent_office AS  
SELECT * FROM monre.agent_office;
```

```
DROP TABLE client CASCADE CONSTRAINTS PURGE;  
CREATE TABLE client AS  
SELECT * FROM monre.client;
```

```
DROP TABLE client_wish CASCADE CONSTRAINTS PURGE;  
CREATE TABLE client_wish AS  
SELECT * FROM monre.client_wish;
```

```
DROP TABLE feature CASCADE CONSTRAINTS PURGE;  
CREATE TABLE feature AS  
SELECT * FROM monre.feature;
```

```
DROP TABLE office CASCADE CONSTRAINTS PURGE;  
CREATE TABLE office AS  
SELECT * FROM monre.office;
```

```
DROP TABLE person CASCADE CONSTRAINTS PURGE;  
CREATE TABLE person AS  
SELECT distinct * FROM monre.person;
```

```
DROP TABLE postcode CASCADE CONSTRAINTS PURGE;  
CREATE TABLE postcode AS  
SELECT * FROM monre.postcode;
```

```
DROP TABLE property CASCADE CONSTRAINTS PURGE;  
CREATE TABLE property AS  
SELECT distinct * FROM monre.property;
```

```
DROP TABLE property_advert CASCADE CONSTRAINTS PURGE;  
CREATE TABLE property_advert AS  
SELECT * FROM monre.property_advert;
```

```
DROP TABLE property_feature CASCADE CONSTRAINTS PURGE;  
CREATE TABLE property_feature AS  
SELECT * FROM monre.property_feature;
```

```
DROP TABLE rent CASCADE CONSTRAINTS PURGE;
```

```

CREATE TABLE rent AS
SELECT * FROM monre.rent;

DROP TABLE sale CASCADE CONSTRAINTS PURGE;
CREATE TABLE sale AS
SELECT * FROM monre.sale;

DROP TABLE state CASCADE CONSTRAINTS PURGE;
CREATE TABLE state AS
SELECT * FROM monre.state;

DROP TABLE visit CASCADE CONSTRAINTS PURGE;
CREATE TABLE visit AS
SELECT * FROM monre.visit;

DELETE
FROM state
WHERE state_code is null;

DELETE
FROM agent
WHERE salary <= 0;

DELETE
FROM agent_office
WHERE person_id NOT IN
(SELECT person_id FROM AGENT);

DELETE
FROM client
WHERE min_budget > max_budget;

DELETE
FROM client
WHERE person_id NOT IN
(SELECT person_id FROM person);

DELETE
FROM visit
WHERE client_person_id NOT IN
(SELECT person_id FROM client);

DELETE
FROM rent
WHERE client_person_id NOT IN
(SELECT person_id FROM client);

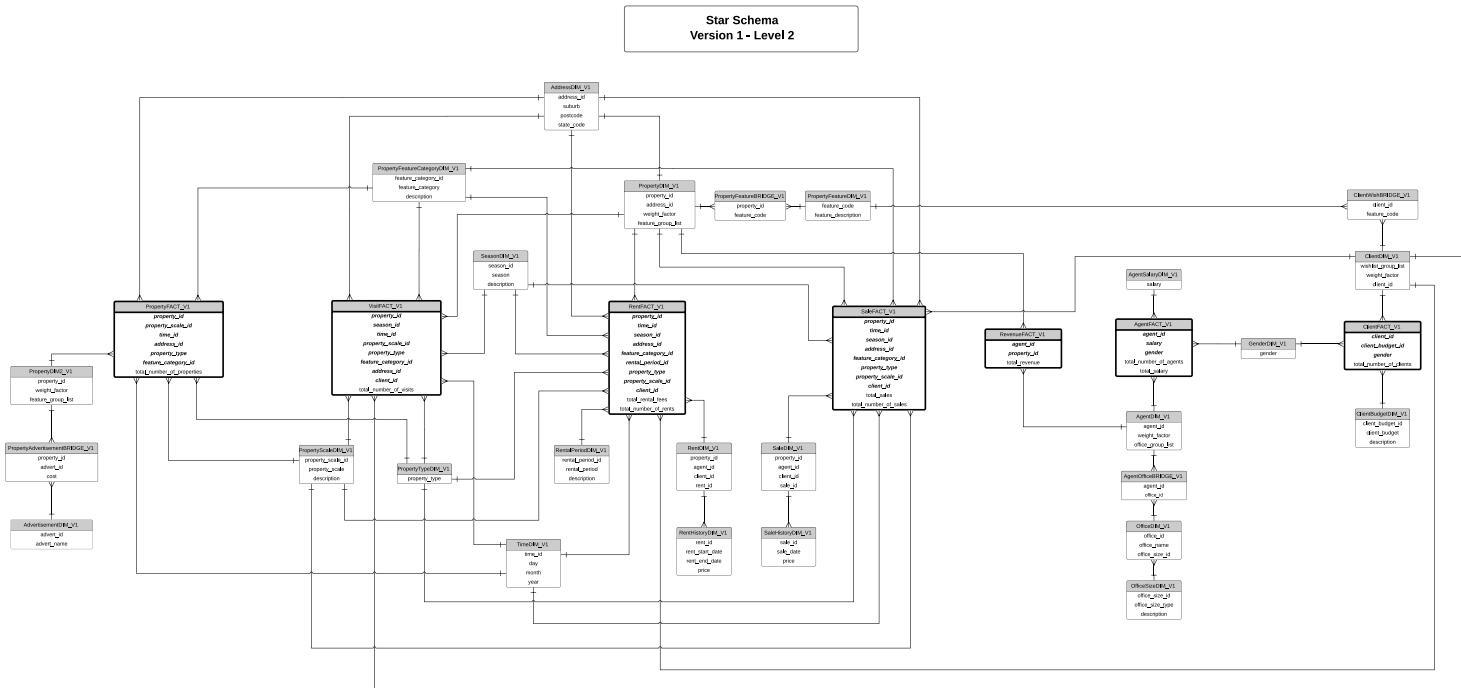
DELETE
FROM person
WHERE address_id NOT IN
(SELECT address_id FROM address);

DELETE
FROM client
WHERE person_id = 7001;

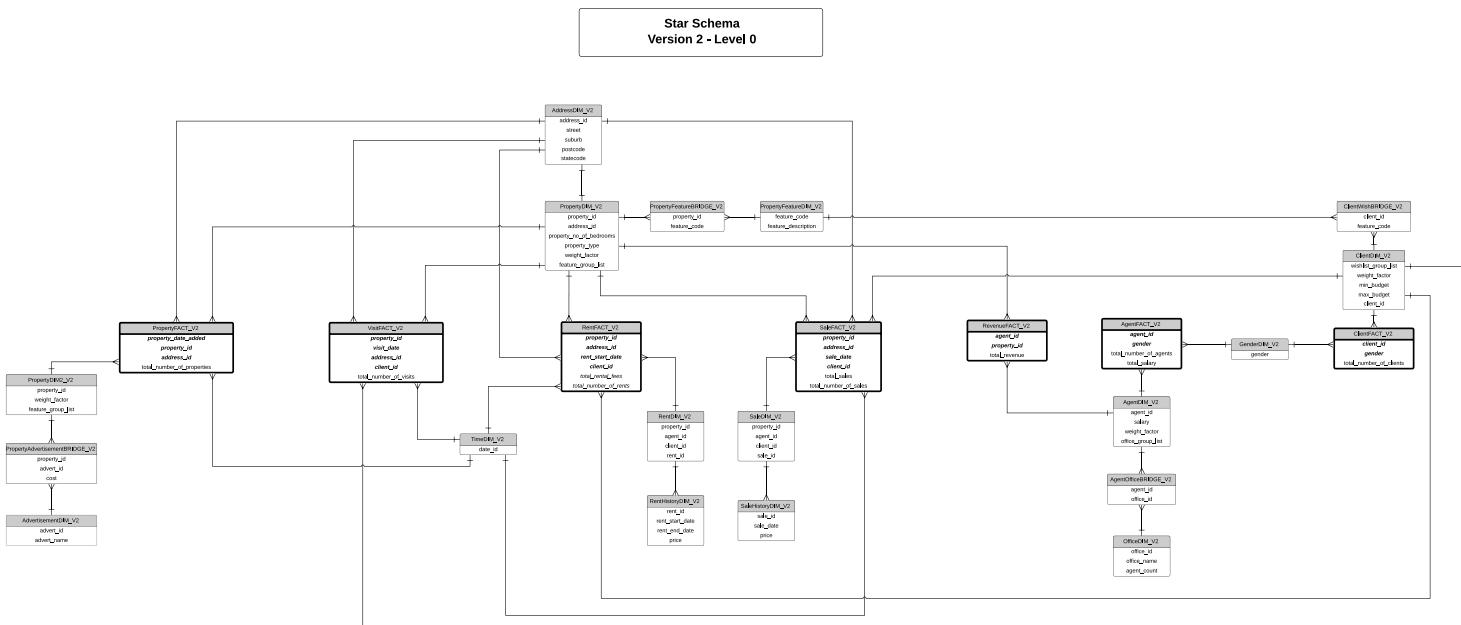
```

c) Two versions of star/snowflake schema diagrams

VERSION 1 – Level 2



VERSION 2 – Level 0



d) A short explanation of why you chose hierarchy or non-hierarchy

We have chosen the non-hierarchy strategy in both versions of the star schemas for the address dimension. As the address dimension has all the address data of all the properties from the operational database, it is useful for and linked with most fact tables like the PropertyFACT, VisitFACT, RentFACT and SaleFACT. Hence, for an increased efficiency in the queries, a less complex data warehouse structure, reduced number of joins while querying and to avoid redundancy of data, the non-hierarchy strategy was used. This ensures easy accessibility of all the attributes from the address dimension which play a crucial role for the Business Intelligence queries and reports.

Although, we have used the hierarchy strategy for the office size dimension along in relation with the office dimension in the 1st version of the star schema. As the office size dimension is a created dimension and needs to exist independently along with the size description and the actual office size of the offices, a hierarchical relationship between these 2 dimensions was necessary. Even though we could have combined both these dimensions, it would have created a rather unnecessary value redundant column of the size description column along with the office size column for each office record.

The choices of hierarchy and non-hierarchy strategies in the star schemas were completely dependent on that specific data, its type and its behaviour and importance in the data warehouse.

e) The reasons of the choice of SCD type for temporal dimension

We have chosen the Type 4 SCD for temporal dimensions in both levels of our data warehouse. The Rent and Sale “price” as temporal attribute for the RentFACT and the SaleFACT respectively has been chosen for keeping the temporal dimensions as they require to be kept track of the price for the price changes over time. Hence using the SCD Type 4, the temporal dimensions are RentHistory and SaleHistory Dimension. The main advantage of choosing Type 4 are as follows,

1. The entire history of the price changes is kept in case we want to analyse the price changes (increase or decrease) for properties over the time.
2. No need to have a different id for the same property.
3. As the complete history is kept, when creating the BI reports, accurate prices will be shown as the prices are recorded based on the time period/id.

f) A short explanation of the difference among the two versions of star/snowflake schema

Both the versions of our star schemas consist of 7 fact tables, namely, PropertyFACT, VisitFACT, RentFACT, SaleFACT, RevenueFACT, ClientFACT and AgentFACT. Each fact table concentrates on answering the management’s questions related to the respective specific topics. The RevenueFACT was added to star schemas to answer the revenue related questions for the management. The revenue for the management is the total amount gained through successful sales and rents towards the clients. The total revenue gained can now be reviewed by the management through the agents’ and properties’ perspectives.

The 1st version of the star schema, level 2, ensures maximum aggregation of data through various manually created dimensions and other dimensions individually as requested by the management. Dimensions like season, office size, property feature category, property scale, etc. are all crucial in aggregating the data into various categories and perspectives for the results. The level 2 star schema of the data warehouse ensures to answer the management’s questions efficiently and by pre-aggregating the data at the maximum level.

Whereas, the 2nd version of the star schema, level 0, makes sure that the data is non-aggregated and in its original form in the data warehouse. The level 0 star schema is similar to the entity relationship diagram of the operational database, while ensuring to answer all of the management’s questions in a non-aggregated manner. The topic related dimensions which existed separately in the 1st version, are combined in the 2nd version of the star schema with respect to the management’s interests. This ensures that all the actual data from the operational database is stored directly in the data warehouse within a topic related dimension for easy accessibility.

TASKC.2 Implement the two versions star/snowflake schema using SQL
a) SQL statements (e.g. create table, insert into, etc) to create the star/snowflake schema
Version-1

-----VERSION 1-----

```
--Create Dimensions
--Create SeasonDIM_V1
DROP TABLE SeasonDIM_V1 CASCADE CONSTRAINTS PURGE;
CREATE TABLE SeasonDIM_V1
(season_id    NUMBER(1),
season      VARCHAR2(10),
description  VARCHAR2(20)
);

--Insert Values into SeasonDIM_V1
INSERT INTO SeasonDIM_V1 VALUES (1, 'Summer', 'DEC-FEB');
INSERT INTO SeasonDIM_V1 VALUES (2, 'Autumn', 'MAR-MAY');
INSERT INTO SeasonDIM_V1 VALUES (3, 'Winter', 'JUN-AUG');
INSERT INTO SeasonDIM_V1 VALUES (4, 'Spring', 'SEP-NOV');

--Create PropertyDIM2_V1
DROP TABLE PropertyDIM2_V1 CASCADE CONSTRAINTS PURGE;
CREATE TABLE PropertyDIM2_V1 AS
SELECT
    P.property_id,
    1.0 / COUNT(PA.advert_id) AS weight_factor,
    LISTAGG(PA.advert_id, '_') WITHIN GROUP (ORDER BY PA.advert_id) AS advertisement_group_list
FROM
    property P,
    property_advert PA
WHERE P.property_id = PA.property_id
GROUP BY P.property_id;

--Create PropertyAdvertisementBRIDGE_V1
DROP TABLE PropertyAdvertisementBRIDGE_V1 CASCADE CONSTRAINTS PURGE;
CREATE TABLE PropertyAdvertisementBRIDGE_V1 AS
SELECT
    property_id,
    advert_id,
    cost
FROM property_advert;

--Create AdvertisementDIM_V1
DROP TABLE AdvertisementDIM_V1 CASCADE CONSTRAINTS PURGE;
CREATE TABLE AdvertisementDIM_V1 AS
SELECT *
FROM advertisement;

--Create PropertyScaleDIM_V1
DROP TABLE PropertyScaleDIM_V1 CASCADE CONSTRAINTS PURGE;
CREATE TABLE PropertyScaleDIM_V1
(property_scale_id    NUMBER(1),
property_scale      VARCHAR2(15),
```

```

description      VARCHAR2(20)
);

--Insert Values into PropertyScaleDIM_V1
INSERT INTO PropertyScaleDIM_V1 VALUES (1, 'Extra Small', '<= 1 bedroom');
INSERT INTO PropertyScaleDIM_V1 VALUES (2, 'Small', '2-3 bedrooms');
INSERT INTO PropertyScaleDIM_V1 VALUES (3, 'Medium', '4-6 bedrooms');
INSERT INTO PropertyScaleDIM_V1 VALUES (4, 'Large', '7-10 bedrooms');
INSERT INTO PropertyScaleDIM_V1 VALUES (5, 'Extra Large', '> 10 bedrooms');

--Create PropertyTypeDIM_V1
DROP TABLE PropertyTypeDIM_V1 CASCADE CONSTRAINTS PURGE;
CREATE TABLE PropertyTypeDIM_V1 AS
SELECT DISTINCT property_type
FROM property;

--Create PropertyFeatureCategoryDIM_V1
DROP TABLE PropertyFeatureCategoryDIM_V1 CASCADE CONSTRAINTS PURGE;
CREATE TABLE PropertyFeatureCategoryDIM_V1
(feature_category_id NUMBER(1),
 feature_category    VARCHAR2(15),
 description        VARCHAR2(20)
);

--Insert Values into PropertyFeatureCategoryDIM_V1
INSERT INTO PropertyFeatureCategoryDIM_V1 VALUES (1, 'Very basic', '< 10 features');
INSERT INTO PropertyFeatureCategoryDIM_V1 VALUES (2, 'Standard', '10-20 features');
INSERT INTO PropertyFeatureCategoryDIM_V1 VALUES (3, 'Luxurious', '> 20 features');

--Create RentalPeriodDIM_V1
DROP TABLE RentalPeriodDIM_V1 CASCADE CONSTRAINTS PURGE;
CREATE TABLE RentalPeriodDIM_V1
(rental_period_id   NUMBER(1),
 rental_period     VARCHAR2(15),
 description       VARCHAR2(20)
);

--Insert Values into RentalPeriodDIM_V1
INSERT INTO RentalPeriodDIM_V1 VALUES (1, 'Short', '< 6 months');
INSERT INTO RentalPeriodDIM_V1 VALUES (2, 'Medium', '6-12 months');
INSERT INTO RentalPeriodDIM_V1 VALUES (3, 'Long', '> 12 months');

--Create TempTimeDIM1_V1
DROP TABLE TempTimeDIM1_V1 CASCADE CONSTRAINTS PURGE;
CREATE TABLE TempTimeDIM1_V1 AS
SELECT DISTINCT
    TO_CHAR(property_date_added, 'DAY') AS day,
    TO_CHAR(property_date_added, 'MM') AS month,
    TO_CHAR(property_date_added, 'YYYY') AS year,
    TO_CHAR(property_date_added, 'YYYYMMDD') AS time_id
FROM property;

--Create TempTimeDIM2_V1
DROP TABLE TempTimeDIM2_V1 CASCADE CONSTRAINTS PURGE;
CREATE TABLE TempTimeDIM2_V1 AS
SELECT DISTINCT

```

```

TO_CHAR(visit_date, 'DAY') AS day,
TO_CHAR(visit_date, 'MM') AS month,
TO_CHAR(visit_date, 'YYYY') AS year,
TO_CHAR(visit_date, 'YYYYMMDAY') AS time_id
FROM visit;

--Create TempTimeDIM3_V1
DROP TABLE TempTimeDIM3_V1 CASCADE CONSTRAINTS PURGE;
CREATE TABLE TempTimeDIM3_V1 AS
SELECT DISTINCT
    TO_CHAR(sale_date, 'DAY') AS day,
    TO_CHAR(sale_date, 'MM') AS month,
    TO_CHAR(sale_date, 'YYYY') AS year,
    TO_CHAR(sale_date, 'YYYYMMDAY') AS time_id
FROM sale;

--Create TempTimeDIM4_V1
DROP TABLE TempTimeDIM4_V1 CASCADE CONSTRAINTS PURGE;
CREATE TABLE TempTimeDIM4_V1 AS
SELECT DISTINCT
    TO_CHAR(rent_start_date, 'DAY') AS day,
    TO_CHAR(rent_start_date, 'MM') AS month,
    TO_CHAR(rent_start_date, 'YYYY') AS year,
    TO_CHAR(rent_start_date, 'YYYYMMDAY') AS time_id
FROM rent;

--Create TimeDIM_V1
DROP TABLE TimeDIM_V1 CASCADE CONSTRAINTS PURGE;
CREATE TABLE TimeDIM_V1 AS
SELECT * FROM TempTimeDIM1_V1
UNION
SELECT * FROM TempTimeDIM2_V1
UNION
SELECT * FROM TempTimeDIM3_V1
UNION
SELECT * FROM TempTimeDIM4_V1;

--Create PropertyDIM_V1
DROP TABLE PropertyDIM_V1 CASCADE CONSTRAINTS PURGE;
CREATE TABLE PropertyDIM_V1 AS
SELECT
    P.property_id,
    P.address_id,
    (CASE WHEN COUNT(PF.feature_code) > 0 THEN (1.0 / COUNT(PF.feature_code)) ELSE 0 END) AS weight_factor,
    LISTAGG(PF.feature_code, '_') WITHIN GROUP (ORDER BY PF.feature_code) AS feature_group_list
FROM
    property P,
    property_feature PF
WHERE P.property_id = PF.property_id (+)
GROUP BY
    P.property_id,
    P.address_id;

--Create PropertyFeatureBRIDGE_V1
DROP TABLE PropertyFeatureBRIDGE_V1 CASCADE CONSTRAINTS PURGE;

```

```

CREATE TABLE PropertyFeatureBRIDGE_V1 AS
SELECT *
FROM property_feature;

--Create PropertyFeatureDIM_V1
DROP TABLE PropertyFeatureDIM_V1 CASCADE CONSTRAINTS PURGE;
CREATE TABLE PropertyFeatureDIM_V1 AS
SELECT *
FROM feature;

--Create SaleDIM_V1
DROP TABLE SaleDIM_V1 CASCADE CONSTRAINTS PURGE;
CREATE TABLE SaleDIM_V1 AS
SELECT
    property_id,
    agent_person_id AS agent_id,
    client_person_id AS client_id,
    sale_id
FROM sale;

--Create SaleHistoryDIM_V1
DROP TABLE SaleHistoryDIM_V1 CASCADE CONSTRAINTS PURGE;
CREATE TABLE SaleHistoryDIM_V1 AS
SELECT
    sale_id,
    sale_date,
    price
FROM sale;

--Create RentDIM_V1
DROP TABLE RentDIM_V1 CASCADE CONSTRAINTS PURGE;
CREATE TABLE RentDIM_V1 AS
SELECT
    property_id,
    agent_person_id AS agent_id,
    client_person_id AS client_id,
    rent_id
FROM rent;

--Create RentHistoryDIM_V1
DROP TABLE RentHistoryDIM_V1 CASCADE CONSTRAINTS PURGE;
CREATE TABLE RentHistoryDIM_V1 AS
SELECT
    rent_id,
    rent_start_date,
    rent_end_date,
    price
FROM rent;

--Create OfficeSizeDIM_V1
DROP TABLE OfficeSizeDIM_V1 CASCADE CONSTRAINTS PURGE;
CREATE TABLE OfficeSizeDIM_V1
(office_size_id    NUMBER(2),
 office_size_type  VARCHAR2(10),
 description       VARCHAR2(30));

```

```
--Insert Values into AgentOfficeSizeDIM_V1
INSERT INTO OfficeSizeDIM_V1 VALUES(1, 'Small', '< 4 employees');
INSERT INTO OfficeSizeDIM_V1 VALUES(2, 'Medium', '4-12 employees');
INSERT INTO OfficeSizeDIM_V1 VALUES(3, 'Big', '> 12 employees');

--Create GenderDIM_V1
DROP TABLE GenderDIM_V1 CASCADE CONSTRAINTS PURGE;
CREATE TABLE GenderDIM_V1 AS
SELECT DISTINCT gender
FROM person;

--Create AgentSalaryDIM_V1
DROP TABLE AgentSalaryDIM_V1 CASCADE CONSTRAINTS PURGE;
CREATE TABLE AgentSalaryDIM_V1 AS
SELECT DISTINCT salary
FROM agent;

--Create TempOfficeDIM_V1
DROP TABLE TempOfficeDIM_V1 CASCADE CONSTRAINTS PURGE;
CREATE TABLE TempOfficeDIM_V1 AS
SELECT
    O.office_id,
    O.office_name,
    COUNT(A.person_id) AS agent_count
FROM
    office O,
    agent_office A
WHERE O.office_id = A.office_id
GROUP BY
    O.office_id,
    O.office_name;

-- Alter TempOfficeDIM_V1
ALTER TABLE TempOfficeDIM_V1 ADD
(office_size_id NUMBER(1));

-- Update TempOfficeDIM_V1
UPDATE TempOfficeDIM_V1
SET office_size_id = 1
WHERE agent_count < 4;

UPDATE TempOfficeDIM_V1
SET office_size_id = 2
WHERE agent_count >= 4
AND agent_count <= 12;

UPDATE TempOfficeDIM_V1
SET office_size_id = 3
WHERE agent_count > 12;

--Create OfficeDIM_V1
DROP TABLE OfficeDIM_V1 CASCADE CONSTRAINTS PURGE;
CREATE TABLE OfficeDIM_V1 AS
SELECT
    office_id,
    office_name,
```

```

office_size_id
FROM TempOfficeDIM_V1;

--Create AgentOfficeBRIDGE_V1
DROP TABLE AgentOfficeBRIDGE_V1 CASCADE CONSTRAINTS PURGE;
CREATE TABLE AgentOfficeBRIDGE_V1 AS
SELECT
    person_id AS agent_id,
    office_id
FROM agent_office;

--Create AgentDIM_V1
DROP TABLE AgentDIM_V1 CASCADE CONSTRAINTS PURGE;
CREATE TABLE AgentDIM_V1 AS
SELECT
    A.person_id AS agent_id,
    1.0 / COUNT(AO.office_id) AS weight_factor,
    LISTAGG(AO.office_id, '_') WITHIN GROUP (ORDER BY AO.office_id) AS office_group_list
FROM
    agent A,
    agent_office AO
WHERE A.person_id = AO.person_id (+)
GROUP BY A.person_id;

--Create AddressDIM_V1
DROP TABLE AddressDIM_V1 CASCADE CONSTRAINTS PURGE;
CREATE TABLE AddressDIM_V1 AS
SELECT DISTINCT
    P.address_id,
    A.suburb,
    PC.postcode,
    PC.state_code
FROM
    property P,
    address A,
    postcode PC
WHERE P.address_id = A.address_id
AND A.postcode = PC.postcode;

--Create ClientBudgetDIM_V1
DROP TABLE ClientBudgetDIM_V1 CASCADE CONSTRAINTS PURGE;
CREATE TABLE ClientBudgetDIM_V1
(client_budget_id NUMBER(3),
client_budget VARCHAR2(20),
description VARCHAR2(30));

--Insert Values into ClientBudgetDIM_V1
INSERT INTO ClientBudgetDIM_V1 VALUES(1, 'Low', '0 to 1000');
INSERT INTO ClientBudgetDIM_V1 VALUES(2, 'Medium', '1001 to 100000');
INSERT INTO ClientBudgetDIM_V1 VALUES(3, 'High', '100001 to 1000000');

--Create ClientDIM_V1
DROP TABLE ClientDIM_V1 CASCADE CONSTRAINTS PURGE;
CREATE TABLE ClientDIM_V1 AS
SELECT
    C.person_id AS client_id,

```

```

(CASE WHEN COUNT(W.feature_code) > 0 THEN (1.0 / COUNT(W.feature_code)) ELSE 0 END) AS
weight_factor,
LISTAGG(W.feature_code, '_') WITHIN GROUP (ORDER BY W.feature_code) AS wishlist_group_list
FROM
client C,
client_wish W
WHERE C.person_id = W.person_id (+)
GROUP BY C.person_id;

--Create ClientWishBRIDGE_V1
DROP TABLE ClientWishBRIDGE_V1 CASCADE CONSTRAINTS PURGE;
CREATE TABLE ClientWishBRIDGE_V1 AS
SELECT
    person_id AS client_id,
    feature_code
FROM client_wish;

--Create Facts
--Create TempPropertyFACT_V1
DROP TABLE TempPropertyFACT_V1 CASCADE CONSTRAINTS PURGE;
CREATE TABLE TempPropertyFACT_V1 AS
SELECT
    P.property_id,
    to_char(P.property_date_added, 'YYYYMMMDAY') AS time_id,
    P.property_type,
    P.address_id,
    P.property_no_of_bedrooms,
    COUNT(PF.feature_code) AS feature_count
FROM
    property P,
    property_feature PF
WHERE P.property_id = PF.property_id (+)
GROUP BY
    P.property_id,
    to_char(P.property_date_added, 'YYYYMMMDAY'),
    P.property_type,
    P.address_id,
    P.property_no_of_bedrooms;

--Alter TempPropertyFACT_V1
ALTER TABLE TempPropertyFACT_V1 ADD (
property_scale_id NUMBER(1),
feature_category_id NUMBER(1));

--Update TempPropertyFACT_V1
UPDATE TempPropertyFACT_V1 SET property_scale_id = 1
WHERE property_no_of_bedrooms <= 1;

UPDATE TempPropertyFACT_V1 SET property_scale_id = 2
WHERE property_no_of_bedrooms >= 2
AND property_no_of_bedrooms <= 3;

UPDATE TempPropertyFACT_V1 SET property_scale_id = 3
WHERE property_no_of_bedrooms >= 4
AND property_no_of_bedrooms <= 6;

```

```
UPDATE TempPropertyFACT_V1 SET property_scale_id = 4
WHERE property_no_of_bedrooms >=7
AND property_no_of_bedrooms <= 10;
```

```
UPDATE TempPropertyFACT_V1 SET property_scale_id = 5
WHERE property_no_of_bedrooms > 10;
```

```
UPDATE TempPropertyFACT_V1 SET feature_count = 0
WHERE feature_count IS NULL;
```

```
UPDATE TempPropertyFACT_V1 SET feature_category_id = 1
WHERE feature_count < 10;
```

```
UPDATE TempPropertyFACT_V1 SET feature_category_id = 2
WHERE feature_count >= 10
AND feature_count <= 20;
```

```
UPDATE TempPropertyFACT_V1 SET feature_category_id = 3
WHERE feature_count > 20;
```

```
--Create PropertyFACT_V1
DROP TABLE PropertyFACT_V1 CASCADE CONSTRAINTS PURGE;
CREATE TABLE PropertyFACT_V1 AS
SELECT
    property_id,
    property_scale_id,
    time_id,
    property_type,
    feature_category_id,
    address_id,
    COUNT(*) AS total_number_of_properties
FROM TempPropertyFACT_V1
GROUP BY
    property_id,
    property_scale_id,
    time_id,
    property_type,
    feature_category_id,
    address_id;
```

```
--Create TempVisitFACT_V1
DROP TABLE TempVisitFACT_V1 CASCADE CONSTRAINTS PURGE;
CREATE TABLE TempVisitFACT_V1 AS
SELECT
    P.property_id,
    to_char(V.visit_date, 'YYYYMMDAY') AS time_id,
    to_char(V.visit_date) AS visit_date,
    P.property_type,
    P.address_id,
    P.property_no_of_bedrooms,
    V.client_person_id AS client_id,
    (CASE WHEN PD.weight_factor > 0 THEN ROUND(1 / PD.weight_factor) ELSE 0 END) AS
feature_count
FROM
    property P,
    PropertyDIM_V1 PD,
```

Visit V

```
WHERE V.property_id = PD.property_id (+)
AND V.property_id = P.property_id;
```

--Alter TempVisitFACT_V1

```
ALTER TABLE TempVisitFACT_V1 ADD (
season_id NUMBER(1),
property_scale_id NUMBER(1),
feature_category_id NUMBER(1));
```

--UPDATE TempVisitFACT_V1

```
UPDATE TempVisitFACT_V1 SET season_id = 1
WHERE EXTRACT( MONTH FROM TO_DATE( visit_date ) ) >= 01
AND EXTRACT( MONTH FROM TO_DATE( visit_date ) ) <= 02;
```

```
UPDATE TempVisitFACT_V1 SET season_id = 1
```

```
WHERE EXTRACT( MONTH FROM TO_DATE( visit_date ) ) = 12;
```

```
UPDATE TempVisitFACT_V1 SET season_id = 3
```

```
WHERE EXTRACT( MONTH FROM TO_DATE( visit_date ) ) >= 06
AND EXTRACT( MONTH FROM TO_DATE( visit_date ) ) <= 08;
```

```
UPDATE TempVisitFACT_V1 SET season_id = 2
```

```
WHERE EXTRACT( MONTH FROM TO_DATE( visit_date ) ) >= 03
AND EXTRACT( MONTH FROM TO_DATE( visit_date ) ) <= 05;
```

```
UPDATE TempVisitFACT_V1 SET season_id = 4
```

```
WHERE EXTRACT( MONTH FROM TO_DATE( visit_date ) ) >= 09
AND EXTRACT( MONTH FROM TO_DATE( visit_date ) ) <= 11;
```

```
UPDATE TempVisitFACT_V1 SET property_scale_id = 1
```

```
WHERE property_no_of_bedrooms <= 1;
```

```
UPDATE TempVisitFACT_V1 SET property_scale_id = 2
```

```
WHERE property_no_of_bedrooms >= 2
AND property_no_of_bedrooms <= 3;
```

```
UPDATE TempVisitFACT_V1 SET property_scale_id = 3
```

```
WHERE property_no_of_bedrooms >= 4
AND property_no_of_bedrooms <= 6;
```

```
UPDATE TempVisitFACT_V1 SET property_scale_id = 4
```

```
WHERE property_no_of_bedrooms >= 7
AND property_no_of_bedrooms <= 10;
```

```
UPDATE TempVisitFACT_V1 SET property_scale_id = 5
```

```
WHERE property_no_of_bedrooms > 10;
```

```
UPDATE TempVisitFACT_V1 SET feature_count = 0
```

```
WHERE feature_count IS NULL;
```

```
UPDATE TempVisitFACT_V1 SET feature_category_id = 1
```

```
WHERE feature_count < 10;
```

```
UPDATE TempVisitFACT_V1 SET feature_category_id = 2
```

```
WHERE feature_count >= 10
```

AND feature_count <= 20;

```
UPDATE TempVisitFACT_V1 SET feature_category_id = 3
WHERE feature_count > 20;
```

--Create VisitFACT_V1

```
DROP TABLE VisitFACT_V1 CASCADE CONSTRAINTS PURGE;
CREATE TABLE VisitFACT_V1 AS
SELECT
    property_id,
    season_id,
    time_id,
    property_type,
    address_id,
    property_scale_id,
    feature_category_id,
    client_id,
    COUNT(*) AS total_number_of_visits
FROM TempVisitFACT_V1
GROUP BY
    property_id,
    season_id ,
    time_id,
    property_type,
    address_id,
    property_scale_id,
    feature_category_id,
    client_id;
```

--Create TempRentFACT_V1

```
DROP TABLE TempRentFACT_V1 CASCADE CONSTRAINTS PURGE;
CREATE TABLE TempRentFACT_V1 AS
SELECT
    P.address_id,
    (CASE WHEN PD.weight_factor > 0 THEN ROUND(1 / PD.weight_factor) ELSE 0 END) AS
feature_count,
    to_char(R.rent_start_date, 'YYYYMMMDAY') AS time_id,
    R.price,
    P.property_id,
    R.rent_start_date,
    R.rent_end_date,
    P.property_type,
    P.property_no_of_bedrooms,
    R.client_person_id AS client_id
FROM
    rent R,
    property P,
    propertyDIM_V1 PD
WHERE R.property_id = P.property_id
AND R.property_id = PD.property_id (+)
AND R.client_person_id IS NOT NULL;
```

--Alter TempRentFACT_V1

```
ALTER TABLE TempRentFACT_V1 ADD
(season_id      NUMBER(1),
rental_period_id  NUMBER(1),
```

```

property_scale_id NUMBER(1),
feature_category_id NUMBER(1)
);

--Update TempRentFACT_V1
UPDATE TempRentFACT_V1
SET season_id = 2
WHERE TO_CHAR(rent_start_date, 'MM') >= 3
AND TO_CHAR(rent_start_date, 'MM') <= 5;

UPDATE TempRentFACT_V1
SET season_id = 3
WHERE TO_CHAR(rent_start_date, 'MM') >= 6
AND TO_CHAR(rent_start_date, 'MM') <= 8;

UPDATE TempRentFACT_V1
SET season_id = 4
WHERE TO_CHAR(rent_start_date, 'MM') >= 9
AND TO_CHAR(rent_start_date, 'MM') <= 11;

UPDATE TempRentFACT_V1
SET season_id = 1
WHERE season_id IS NULL;

UPDATE TempRentFACT_V1
SET rental_period_id = 1
WHERE (ROUND(MONTHS_BETWEEN(rent_end_date, rent_start_date))) < 06;

UPDATE TempRentFACT_V1
SET rental_period_id = 2
WHERE (ROUND(MONTHS_BETWEEN(rent_end_date, rent_start_date))) >= 06
AND (ROUND(MONTHS_BETWEEN(rent_end_date, rent_start_date))) <= 12;

UPDATE TempRentFACT_V1
SET rental_period_id = 3
WHERE (ROUND(MONTHS_BETWEEN(rent_end_date, rent_start_date))) > 12;

UPDATE TempRentFACT_V1
SET property_scale_id = 1
WHERE property_no_of_bedrooms <= 1;

UPDATE TempRentFACT_V1
SET property_scale_id = 2
WHERE property_no_of_bedrooms >= 2
AND property_no_of_bedrooms <= 3;

UPDATE TempRentFACT_V1
SET property_scale_id = 3
WHERE property_no_of_bedrooms >= 4
AND property_no_of_bedrooms <= 6;

UPDATE TempRentFACT_V1
SET property_scale_id = 4
WHERE property_no_of_bedrooms >= 7
AND property_no_of_bedrooms <= 10;

```

```
UPDATE TempRentFACT_V1
SET property_scale_id = 5
WHERE property_no_of_bedrooms > 10;
```

```
UPDATE TempRentFACT_V1
SET feature_count = 0
WHERE feature_count IS NULL;
```

```
UPDATE TempRentFACT_V1
SET feature_category_id = 1
WHERE feature_count < 10;
```

```
UPDATE TempRentFACT_V1
SET feature_category_id = 2
WHERE feature_count >= 10
AND feature_count <= 20;
```

```
UPDATE TempRentFACT_V1
SET feature_category_id = 3
WHERE feature_count > 20;
```

```
--Create RentFACT_V1
DROP TABLE RentFACT_V1 CASCADE CONSTRAINTS PURGE;
CREATE TABLE RentFACT_V1 AS
SELECT
    property_id,
    time_id,
    season_id,
    address_id,
    feature_category_id,
    rental_period_id,
    property_type,
    property_scale_id,
    client_id,
    COUNT(*) AS total_number_of_rents,
    SUM(price) AS total_rental_fees
FROM TempRentFACT_V1
GROUP BY
    address_id,
    time_id,
    season_id,
    feature_category_id,
    property_id,
    rental_period_id,
    property_type,
    price,
    property_scale_id,
    client_id;
```

```
--Create TempSaleFACT_V1
DROP TABLE TempSaleFACT_V1 CASCADE CONSTRAINTS PURGE;
CREATE TABLE TempSaleFACT_V1 AS
SELECT
    P.address_id,
    (CASE WHEN PD.weight_factor > 0 THEN ROUND(1 / PD.weight_factor) ELSE 0 END) AS
    feature_count,
```

```

to_char(S.sale_date, 'YYYYMMDD') AS time_id,
S.price,
S.sale_date,
P.property_id,
P.property_type,
P.property_no_of_bedrooms,
S.client_person_id AS client_id
FROM
    sale S,
    property P,
    propertyDIM_V1 PD
WHERE S.property_id = P.property_id
AND S.property_id = PD.property_id (+)
AND S.client_person_id IS NOT NULL;

--Alter TempSaleFACT_V1
ALTER TABLE TempSaleFACT_V1 ADD
(season_id      NUMBER(1),
property_scale_id  NUMBER(1),
feature_category_id  NUMBER(1)
);

--Update TempSaleFACT_V1
UPDATE TempSaleFACT_V1
SET season_id = 2
WHERE TO_CHAR(sale_date, 'MM') >= 3
AND TO_CHAR(sale_date, 'MM') <= 5;

UPDATE TempSaleFACT_V1
SET season_id = 3
WHERE TO_CHAR(sale_date, 'MM') >= 6
AND TO_CHAR(sale_date, 'MM') <= 8;

UPDATE TempSaleFACT_V1
SET season_id = 4
WHERE TO_CHAR(sale_date, 'MM') >= 9
AND TO_CHAR(sale_date, 'MM') <= 11;

UPDATE TempSaleFACT_V1
SET season_id = 1
WHERE season_id IS NULL;

UPDATE TempSaleFACT_V1
SET property_scale_id = 1
WHERE property_no_of_bedrooms <= 1;

UPDATE TempSaleFACT_V1
SET property_scale_id = 2
WHERE property_no_of_bedrooms >= 2
AND property_no_of_bedrooms <= 3;

UPDATE TempSaleFACT_V1
SET property_scale_id = 3
WHERE property_no_of_bedrooms >= 4
AND property_no_of_bedrooms <= 6;

```

```
UPDATE TempSaleFACT_V1
SET property_scale_id = 4
WHERE property_no_of_bedrooms >= 7
AND property_no_of_bedrooms <= 10;
```

```
UPDATE TempSaleFACT_V1
SET property_scale_id = 5
WHERE property_no_of_bedrooms > 10;
```

```
UPDATE TempSaleFACT_V1
SET feature_count = 0
WHERE feature_count IS NULL;
```

```
UPDATE TempSaleFACT_V1
SET feature_category_id = 1
WHERE feature_count < 10;
```

```
UPDATE TempSaleFACT_V1
SET feature_category_id = 2
WHERE feature_count >= 10
AND feature_count <= 20;
```

```
UPDATE TempSaleFACT_V1
SET feature_category_id = 3
WHERE feature_count > 20;
```

```
--Create SaleFACT_V1
DROP TABLE SaleFACT_V1 CASCADE CONSTRAINTS PURGE;
CREATE TABLE SaleFACT_V1 AS
SELECT
    property_id,
    time_id,
    season_id,
    address_id,
    feature_category_id,
    property_type,
    property_scale_id,
    client_id,
    COUNT(*) AS total_number_of_sales,
    SUM(price) AS total_sales
FROM TempSaleFACT_V1
GROUP BY
    address_id,
    time_id,
    season_id,
    feature_category_id,
    property_id,
    property_type,
    price,
    property_scale_id,
    client_id;
```

```
--Create TempClientFACT_V1
DROP TABLE TempClientFACT_V1 CASCADE CONSTRAINTS PURGE;
CREATE TABLE TempClientFACT_V1 AS
SELECT
```

```

C.person_id AS client_id,
C.max_budget AS client_budget,
P.gender
FROM client C,
person P,
client_wish CW
WHERE C.person_id = P.person_id
AND C.person_id = CW.person_id (+);

-- Alter TempClientFACT_V1
ALTER TABLE TempClientFACT_V1 ADD
(client_budget_id NUMBER(1));

-- Update TempClientFACT_V1
UPDATE TempClientFACT_V1
SET client_budget_id = 1
WHERE client_budget BETWEEN 0 AND 1000;

UPDATE TempClientFACT_V1
SET client_budget_id = 2
WHERE client_budget BETWEEN 1001 AND 100000;

UPDATE TempClientFACT_V1
SET client_budget_id = 3
WHERE client_budget BETWEEN 100001 AND 10000000;

--Create ClientFACT_V1
DROP TABLE ClientFACT_V1 CASCADE CONSTRAINTS PURGE;
CREATE TABLE ClientFACT_V1 AS
SELECT
    client_id,
    client_budget_id,
    gender,
    COUNT(DISTINCT client_id) AS total_number_of_clients
FROM TempClientFACT_V1
GROUP BY
    client_id,
    client_budget_id,
    gender;

--Create TempAgentFACT_V1
DROP TABLE TempAgentFACT_V1 CASCADE CONSTRAINTS PURGE;
CREATE TABLE TempAgentFACT_V1 AS
SELECT
    A.person_id as agent_id,
    A.salary,
    P.gender
FROM
    agent A,
    person P
WHERE A.person_id = P.person_id;

--Create AgentFACT_V1
DROP TABLE AgentFACT_V1 CASCADE CONSTRAINTS PURGE;
CREATE TABLE AgentFACT_V1 AS
SELECT

```

```

agent_id,
salary,
gender,
COUNT(agent_id) AS total_number_of_agents,
SUM(salary) AS total_salary
FROM
TempAgentFACT_V1
GROUP BY
agent_id,
salary,
gender;

--Create TempRevenueFACT1_V1
DROP TABLE TempRevenueFACT1_V1 CASCADE CONSTRAINTS PURGE;
CREATE TABLE TempRevenueFACT1_V1 AS
SELECT
agent_person_id AS agent_id,
property_id,
price AS total_revenue
FROM sale
WHERE client_person_id IS NOT NULL;

--Create TempRevenueFACT2_V1
DROP TABLE TempRevenueFACT2_V1 CASCADE CONSTRAINTS PURGE;
CREATE TABLE TempRevenueFACT2_V1 AS
SELECT
agent_person_id as agent_id,
property_id,
ROUND((price / 7) * (rent_end_date - rent_start_date), 2) AS total_revenue
FROM rent
WHERE client_person_id IS NOT NULL;

--Create RevenueFACT_V1
DROP TABLE RevenueFACT_V1 CASCADE CONSTRAINTS PURGE;
CREATE TABLE RevenueFACT_V1 AS
SELECT * FROM TempRevenueFACT1_V1
UNION
SELECT * FROM TempRevenueFACT2_V1;

```

b) SQL statements (e.g. create table, insert into, etc) to create the star/snowflake schema Version-2

-----VERSION 2-----

```
--Create Dimensions
--Create PropertyDIM2_V2
DROP TABLE PropertyDIM2_V2 CASCADE CONSTRAINTS PURGE;
CREATE TABLE PropertyDIM2_V2 AS
SELECT
    P.property_id,
    1.0 / COUNT(PA.advert_id) AS weight_factor,
    LISTAGG(PA.advert_id, '_') WITHIN GROUP (ORDER BY PA.advert_id) AS advertisement_group_list
FROM
    property P,
    property_advert PA
WHERE P.property_id = PA.property_id
GROUP BY P.property_id;

--Create PropertyAdvertisementBRIDGE_V2
DROP TABLE PropertyAdvertisementBRIDGE_V2 CASCADE CONSTRAINTS PURGE;
CREATE TABLE PropertyAdvertisementBRIDGE_V2 AS
SELECT
    property_id,
    advert_id,
    cost
FROM property_advert;

--Create AdvertisementDIM_V2
DROP TABLE AdvertisementDIM_V2 CASCADE CONSTRAINTS PURGE;
CREATE TABLE AdvertisementDIM_V2 AS
SELECT *
FROM advertisement;

--Create TempDateDIM_V2
DROP TABLE TempDateDIM_V2 CASCADE CONSTRAINTS PURGE;
CREATE TABLE TempDateDIM_V2 AS
SELECT DISTINCT
    property_date_added AS date_id
FROM property;

--Create TempVisitDateDIM_V2
DROP TABLE TempVisitDateDIM_V2 CASCADE CONSTRAINTS PURGE;
CREATE TABLE TempVisitDateDIM_V2 AS
SELECT DISTINCT visit_date AS date_id
FROM visit;

--Create TempSaleDateDIM_V2
DROP TABLE TempSaleDateDIM_V2 CASCADE CONSTRAINTS PURGE;
CREATE TABLE TempSaleDateDIM_V2 AS
SELECT DISTINCT sale_date AS date_id
FROM sale;

--Create TempRentStartDateDIM_V2
DROP TABLE TempRentStartDateDIM_V2 CASCADE CONSTRAINTS PURGE;
```

```

CREATE TABLE TempRentStartDateDIM_V2 AS
SELECT DISTINCT rent_start_date AS date_id
FROM rent;

--Create TimeDIM_V2
DROP TABLE TimeDIM_V2 CASCADE CONSTRAINTS PURGE;
CREATE TABLE TimeDIM_V2 AS
SELECT * FROM TempDateDIM_V2
UNION
SELECT * FROM TempVisitDateDIM_V2
UNION
SELECT * FROM TempSaleDateDIM_V2
UNION
SELECT * FROM TempRentStartDateDIM_V2;

--Create PropertyDIM_V2
DROP TABLE PropertyDIM_V2 CASCADE CONSTRAINTS PURGE;
CREATE TABLE PropertyDIM_V2 AS
SELECT
    P.property_id,
    P.address_id,
    P.property_no_of_bedrooms,
    P.property_type,
    (CASE WHEN COUNT(PF.feature_code) > 0 THEN (1.0 / COUNT(PF.feature_code)) ELSE 0 END) AS weight_factor,
    LISTAGG(PF.feature_code, '_') WITHIN GROUP (ORDER BY PF.feature_code) AS feature_group_list
FROM
    property P,
    property_feature PF
WHERE P.property_id = PF.property_id (+)
GROUP BY
    P.property_id,
    P.address_id,
    P.property_no_of_bedrooms,
    P.property_type;

--Create PropertyFeatureBRIDGE_V2
DROP TABLE PropertyFeatureBRIDGE_V2 CASCADE CONSTRAINTS PURGE;
CREATE TABLE PropertyFeatureBRIDGE_V2 AS
SELECT *
FROM property_feature;

--Create PropertyFeatureDIM_V2
DROP TABLE PropertyFeatureDIM_V2 CASCADE CONSTRAINTS PURGE;
CREATE TABLE PropertyFeatureDIM_V2 AS
SELECT *
FROM feature;

--Create SaleDIM_V2
DROP TABLE SaleDIM_V2 CASCADE CONSTRAINTS PURGE;
CREATE TABLE SaleDIM_V2 AS
SELECT
    property_id,
    agent_person_id AS agent_id,
    client_person_id AS client_id,
    sale_id

```

FROM sale;

```
--Create SaleHistoryDIM_V2
DROP TABLE SaleHistoryDIM_V2 CASCADE CONSTRAINTS PURGE;
CREATE TABLE SaleHistoryDIM_V2 AS
SELECT
    sale_id,
    sale_date,
    price
FROM sale;
```

```
--Create RentDIM_V2
DROP TABLE RentDIM_V2 CASCADE CONSTRAINTS PURGE;
CREATE TABLE RentDIM_V2 AS
SELECT
    property_id,
    agent_person_id AS agent_id,
    client_person_id AS client_id,
    rent_id
FROM rent;
```

```
--Create RentHistoryDIM_V2
DROP TABLE RentHistoryDIM_V2 CASCADE CONSTRAINTS PURGE;
CREATE TABLE RentHistoryDIM_V2 AS
SELECT
    rent_id,
    rent_start_date,
    rent_end_date,
    price
FROM rent;
```

```
--Create GenderDIM_V2
DROP TABLE GenderDIM_V2 CASCADE CONSTRAINTS PURGE;
CREATE TABLE GenderDIM_V2 AS
SELECT DISTINCT gender
FROM person;
```

```
--Create OfficeDIM_V2
DROP TABLE OfficeDIM_V2 CASCADE CONSTRAINTS PURGE;
CREATE TABLE OfficeDIM_V2 AS
SELECT
    O.office_id,
    O.office_name,
    COUNT(A.person_id) AS agent_count
FROM
    office O,
    agent_office A
WHERE O.office_id = A.office_id
GROUP BY
    O.office_id,
    O.office_name;
```

```
--Create AgentOfficeBRIDGE_V2
DROP TABLE AgentOfficeBRIDGE_V2 CASCADE CONSTRAINTS PURGE;
CREATE TABLE AgentOfficeBRIDGE_V2 AS
SELECT
```

```

person_id AS agent_id,
office_id
FROM agent_office;

--Create AgentDIM_V2
DROP TABLE AgentDIM_V2 CASCADE CONSTRAINTS PURGE;
CREATE TABLE AgentDIM_V2 AS
SELECT
    A.person_id AS agent_id,
    A.salary,
    1.0 / COUNT(AO.office_id) AS weight_factor,
    LISTAGG(AO.office_id, '_') WITHIN GROUP (ORDER BY AO.office_id) AS office_group_list
FROM
    agent A,
    agent_office AO
WHERE A.person_id = AO.person_id (+)
GROUP BY
    A.person_id,
    A.salary;

--Create AddressDIM_V2
DROP TABLE AddressDIM_V2 CASCADE CONSTRAINTS PURGE;
CREATE TABLE AddressDIM_V2 AS
SELECT DISTINCT
    P.address_id,
    A.street,
    A.suburb,
    PC.postcode,
    PC.state_code
FROM
    property P,
    address A,
    postcode PC
WHERE P.address_id = A.address_id
AND A.postcode = PC.postcode;

--Create ClientDIM_V2
DROP TABLE ClientDIM_V2 CASCADE CONSTRAINTS PURGE;
CREATE TABLE ClientDIM_V2 AS
SELECT
    C.person_id AS client_id,
    C.min_budget,
    C.max_budget,
    (CASE WHEN COUNT(W.feature_code) > 0 THEN (1.0 / COUNT(W.feature_code)) ELSE 0 END) AS
    weight_factor,
    LISTAGG(W.feature_code, '_') WITHIN GROUP (ORDER BY W.feature_code) AS wishlist_group_list
FROM
    client C,
    client_wish W
WHERE C.person_id = W.person_id (+)
GROUP BY
    C.person_id,
    C.min_budget,
    C.max_budget;

--Create ClientWishBRIDGE_V2

```

```

DROP TABLE ClientWishBRIDGE_V2 CASCADE CONSTRAINTS PURGE;
CREATE TABLE ClientWishBRIDGE_V2 AS
SELECT
    person_id AS client_id,
    feature_code
FROM client_wish;

--Create Facts
--Create PropertyFACT_V2
DROP TABLE PropertyFACT_V2 CASCADE CONSTRAINTS PURGE;
CREATE TABLE PropertyFACT_V2 AS
SELECT
    property_date_added,
    address_id,
    property_id,
    COUNT(*) AS total_number_of_properties
FROM property
GROUP BY
    property_date_added,
    address_id,
    property_id;

--Create VisitFACT_V2
DROP TABLE VisitFACT_V2 CASCADE CONSTRAINTS PURGE;
CREATE TABLE VisitFACT_V2 AS
SELECT
    P.property_id,
    V.visit_date,
    P.address_id,
    V.client_person_id AS client_id,
    COUNT(*) AS total_number_of_visits
FROM
    property P,
    Visit V
WHERE V.property_id = P.property_id
GROUP BY
    P.property_id,
    V.visit_date,
    P.address_id,
    V.client_person_id;

--Create RentFACT_V2
DROP TABLE RentFACT_V2 CASCADE CONSTRAINTS PURGE;
CREATE TABLE RentFACT_V2 AS
SELECT
    P.address_id,
    P.property_id,
    R.rent_start_date,
    R.client_person_id AS client_id,
    SUM(R.price) AS total_rental_fees,
    COUNT(*) AS total_number_of_rents
FROM
    rent R,
    property P
WHERE R.property_id = P.property_id
AND R.client_person_id IS NOT NULL

```

```

GROUP BY
P.address_id,
R.client_person_id,
P.property_id,
R.rent_start_date;

--Create SaleFACT_V2
DROP TABLE SaleFACT_V2 CASCADE CONSTRAINTS PURGE;
CREATE TABLE SaleFACT_V2 AS
SELECT
    P.address_id,
    S.sale_date,
    P.property_id,
    S.client_person_id AS client_id,
    SUM(S.price) AS total_sales,
    COUNT(*) AS total_number_of_sales
FROM
    sale S,
    property P
WHERE S.property_id = P.property_id
AND S.client_person_id IS NOT NULL
GROUP BY
    P.address_id,
    S.sale_date,
    P.property_id,
    S.client_person_id;

--Create ClientFACT_V2
DROP TABLE ClientFACT_V2 CASCADE CONSTRAINTS PURGE;
CREATE TABLE ClientFACT_V2 AS
SELECT
    C.person_id AS client_id,
    P.gender,
    COUNT(*) AS total_number_of_clients
FROM client C,
    person P
WHERE C.person_id = P.person_id
GROUP BY
    C.person_id,
    P.gender;

--Create AgentFACT_V2
DROP TABLE AgentFACT_V2 CASCADE CONSTRAINTS PURGE;
CREATE TABLE AgentFACT_V2 AS
SELECT
    A.person_id AS agent_id,
    P.gender,
    COUNT(*) AS total_number_of_agents,
    SUM(A.salary) AS total_salary
FROM
    agent A,
    person P
WHERE A.person_id = P.person_id
GROUP BY
    A.person_id,
    P.gender;

```

```
--Create TempRevenueFACT1_V2
DROP TABLE TempRevenueFACT1_V2 CASCADE CONSTRAINTS PURGE;
CREATE TABLE TempRevenueFACT1_V2 AS
SELECT
    agent_person_id AS agent_id,
    property_id,
    price AS total_revenue
FROM sale
WHERE client_person_id IS NOT NULL;

--Create TempRevenueFACT2_V2
DROP TABLE TempRevenueFACT2_V2 CASCADE CONSTRAINTS PURGE;
CREATE TABLE TempRevenueFACT2_V2 AS
SELECT
    agent_person_id as agent_id,
    property_id,
    ROUND((price / 7) * (rent_end_date - rent_start_date), 2) AS total_revenue
FROM rent
WHERE client_person_id IS NOT NULL;

--Create RevenueFACT_V2
DROP TABLE RevenueFACT_V2 CASCADE CONSTRAINTS PURGE;
CREATE TABLE RevenueFACT_V2 AS
SELECT * FROM TempRevenueFACT1_V2
UNION
SELECT * FROM TempRevenueFACT2_V2;
```

c) Screenshots of the tables that you have created; this includes the contents of each table that you have created. If the table is very big, you can only show the first part of the data.

Version 1

SeasonDIM_V1

	SEASON_ID	SEASON	DESCRIPTION
1	1	Summer	DEC-FEB
2	2	Autumn	MAR-MAY
3	3	Winter	JUN-AUG
4	4	Spring	SEP-NOV

PropertyDIM2_V1

	PROPERTY_ID	WEIGHT_FACTOR	ADVERTISEMENT_GROUP_LIST
1	2	1	16
2	3	1	16
3	4	1	16
4	6	1	16
5	7	1	16
6	8	1	16
7	9	1	16
8	10	1	16
9	12	1	16
10	14	1	16
11	15	1	16
12	16	1	16
13	17	1	16

PropertyAdvertisementBRIDGE_V1

	PROPERTY_ID	ADVERT_ID	COST
1	2894	16	1700000
2	2895	16	800000
3	2896	16	735000
4	2897	16	680000
5	2898	16	500000
6	2899	16	500000
7	2900	16	500000
8	2902	12	500000
9	2903	16	1100000
10	2300	16	985000
11	2302	16	977000

AdvertisementDIM_V1

ADVERT_ID	ADVERT_NAME
1	1 Rent Apartment / Unit / Flat
2	2 Rent Block of Units
3	3 Rent Duplex
4	4 Rent House
5	5 Rent New Apartments / Off the Plan
6	6 Rent Penthouse
7	7 Rent Semi-Detached
8	8 Rent Studio
9	9 Rent Terrace
10	10 Rent Townhouse

PropertyScaleDIM_V1

PROPERTY_SCALE_ID	PROPERTY_SCALE	DESCRIPTION
1	1 Extra Small	<= 1 bedroom
2	2 Small	2-3 bedrooms
3	3 Medium	4-6 bedrooms
4	4 Large	7-10 bedrooms
5	5 Extra Large	> 10 bedrooms

.PropertyTypeDIM_V1

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

PropertyFeatureCategoryDIM_V1

FEATURE_CATEGORY_ID	FEATURE_CATEGORY	DESCRIPTION
1	1 Very basic	< 10 features
2	2 Standard	10-20 features
3	3 Luxurious	> 20 features

RentalPeriodDIM_V1

	RENTAL_PERIOD_ID	RENTAL_PERIOD	DESCRIPTION
1		1 Short	< 6 months
2		2 Medium	6-12 months
3		3 Long	> 12 months

TimeDIM_V1

	DAY	MONTH	YEAR	TIME_ID
1	FRIDAY	01	2020	202001FRIDAY
2	FRIDAY	02	2020	202002FRIDAY
3	FRIDAY	03	2020	202003FRIDAY
4	FRIDAY	04	2020	202004FRIDAY
5	FRIDAY	05	2020	202005FRIDAY
6	FRIDAY	11	2019	201911FRIDAY
7	FRIDAY	12	2019	201912FRIDAY
8	MONDAY	01	2020	202001MONDAY
9	MONDAY	02	2020	202002MONDAY
10	MONDAY	03	2020	202003MONDAY
11	MONDAY	04	2020	202004MONDAY
12	MONDAY	11	2019	201911MONDAY

PropertyDIM_V1

PropertyFeatureBRIDGE_V1

PROPERTY_ID	FEATURE_CODE
1	9
2	9
3	9
4	9
5	9
6	9
7	9
8	9
9	9
10	9
11	9
12	9
13	9
14	11
15	11
16	11
17	11

PropertyFeatureDIM_V1

FEATURE_CODE	FEATURE_DESCRIPTION
1	1 Air conditioning
2	2 Built in wardrobes
3	3 Carpeted
4	4 City Views
5	5 Close to schools
6	6 Close to shops
7	7 Close to transport
8	8 Exhaust
9	9 Heating
10	10 Prestige Homes
11	11 Roller Door Access
12	12 Vacuum System
13	13 Car Parking - Surface
14	14 Ensuite
15	15 Open Fire Place
16	16 Study
17	17 Swimming Pool
18	18 Floorboards
19	19 Garden
20	20 Secure Parking
21	21 Terrace-Balcony
22	22 Alarm System
23	23 Balcony

SaleDIM_V1

PROPERTY_ID	AGENT_ID	CLIENT_ID	SALE_ID
1	1964	1077	2900
2	1896	1079	2901
3	1932	1083	2902
4	1998	1084	2903
5	1943	1087	2904
6	5	1195	2905
7	67	1195	2906
8	72	1195	2907
9	121	1202	2908
10	229	1202	2909
11	159	1202	2910
12	164	1202	2911
13	202	1202	2912
14	305	1202	2913
15	217	1203	2914
16	173	1203	2915
17	241	1204	2916
18	181	1207	2917
19	190	1208	2918
20	320	1209	2919
21	321	1210	2920
22	404	1215	2921
23	250	1217	2922

SaleHistoryDIM_V1

	SALE_ID	SALE_DATE	PRICE
1	434	22-03-20 04:41:47	1395000
2	435	16-01-20 15:37:51	275000
3	436	20-03-20 09:12:14	3490000
4	437	14-01-20 04:58:57	799000
5	438	22-02-20 15:31:38	2000000
6	439	29-01-20 18:16:23	1825000
7	440	14-03-20 22:45:36	380000
8	441	05-04-20 21:05:36	1695000
9	442	19-01-20 20:34:35	495000
10	443	09-03-20 17:54:46	565000
11	444	09-03-20 02:31:28	545000
12	445	19-02-20 11:01:35	280000
13	446	01-03-20 12:59:31	520000
14	447	25-02-20 00:32:40	440000
15	448	02-03-20 12:09:32	470000
16	449	24-01-20 07:10:46	565000
17	450	25-01-20 21:54:32	390000
18	451	28-01-20 17:39:01	950000
19	452	19-03-20 23:37:16	550000
20	453	15-02-20 00:08:42	349000
21	454	22-03-20 10:23:15	339000
22	455	04-02-20 06:28:13	480000
23	456	14-01-20 11:52:55	499000
24	457	18-03-20 02:13:34	500000
25	458	31-03-20 12:59:50	500000
26	459	25-03-20 08:28:26	639000
27	460	28-01-20 22:00:05	639000
28	461	14-02-20 20:59:10	329000
29	462	28-03-20 03:54:39	599000

RentDIM_V1

	PROPERTY_ID	AGENT_ID	CLIENT_ID	RENT_ID
1	6199	568	3713	331
2	6063	568	3714	332
3	6074	568	3715	333
4	6142	568	3716	334
5	6146	568	3717	335
6	5373	570	3718	336
7	5801	570	3719	337
8	5513	570	3720	338
9	5709	570	3721	339
10	5548	571	3722	340
11	5901	571	3723	341
12	5724	571	3724	342
13	6035	571	3725	343

RentHistoryDIM_V1

	RENT_ID	RENT_START_DATE	RENT_END_DATE	PRICE
1	331	12-01-20 13:36:37	28-06-20 13:36:37	795
2	332	02-05-20 13:02:11	18-10-20 13:02:11	500
3	333	01-05-20 17:47:48	17-10-20 17:47:48	370
4	334	12-02-20 05:09:13	29-07-20 05:09:13	795
5	335	20-04-20 20:25:23	06-10-20 20:25:23	595
6	336	27-04-20 00:36:56	13-10-20 00:36:56	350
7	337	25-02-20 00:56:05	11-08-20 00:56:05	600
8	338	01-01-20 08:26:33	17-06-20 08:26:33	430
9	339	29-03-20 04:04:47	13-09-20 04:04:47	420
10	340	23-04-20 14:59:06	09-10-20 14:59:06	520
11	341	01-05-20 06:15:09	17-10-20 06:15:09	330
12	342	01-05-20 08:28:29	17-10-20 08:28:29	500
13	343	30-04-20 01:49:10	16-10-20 01:49:10	625
14	344	23-04-20 19:53:04	09-10-20 19:53:04	815
15	345	21-04-20 18:50:41	07-10-20 18:50:41	370
16	346	23-04-20 01:21:35	09-10-20 01:21:35	495
17	347	18-03-20 18:20:03	02-09-20 18:20:03	1100
18	348	18-03-20 03:27:59	02-09-20 03:27:59	430
19	349	24-01-20 14:32:48	10-07-20 14:32:48	330
20	350	16-04-20 11:12:42	02-10-20 11:12:42	320
21	351	16-01-20 06:50:28	02-07-20 06:50:28	490
22	352	11-03-20 18:12:38	26-08-20 18:12:38	480
23	353	10-03-20 01:29:57	25-08-20 01:29:57	500

OfficeSizeDIM_V1

	OFFICE_SIZE_ID	OFFICE_SIZE_TYPE	DESCRIPTION
1	1	Small	< 4 employees
2	2	Medium	4-12 employees
3	3	Big	> 12 employees

GenderDIM_V1

	GENDER
1	Male
2	Female

AgentSalaryDIM_V1

	SALARY
1	180000
2	200000
3	195000
4	175000
5	190000
6	210000

OfficeDIM_V1

	OFFICE_ID	OFFICE_NAME	OFFICE_SIZE_ID
1	916	Ray White Mount Gravatt	1
2	919	Ray White Nolan & Iken	1
3	937	Ray White Robina	2
4	955	Ray White Upper Coomera	2
5	965	Ray White at The Entertainment Quarter	1
6	966	Rayner Real Estate	1
7	985	Rental Master Pty Ltd	1
8	989	RichardsElliot Surry Hills	1
9	992	Richardson & Wrench Double Bay	1
10	993	Richardson & Wrench Elizabeth Bay / Potts Point	2
11	1010	Rockpool Real Estate	1
12	1023	Sandy Funston	1
13	1024	Santa Cruz Realty	1
14	1046	Space Property Agency	1
15	1051	St George Property Agents - Penshurst	1
16	1052	Stanley Samuels Property	1
17	1053	Starr Partners Kellyville	1
18	1062	Stone McMahons Point	1
19	1064	Stone Real Estate - Coomera	1
20	1068	Stone Real Estate Manly	2

AgentOfficeBRIDGE_V1

	AGENT_ID	OFFICE_ID
1	49	787
2	364	505
3	1245	593
4	1247	1091
5	365	1069
6	1563	502
7	964	235
8	2207	503
9	1249	43
10	58	227
11	1898	1070
12	61	438
13	1251	837
14	2210	1132
15	1899	656
16	1567	275
17	970	54
18	971	810
19	371	1071
20	1904	1096

AgentDIM_V1

	AGENT_ID	WEIGHT_FACTOR	OFFICE_GROUP_LIST
1	1	1 658	
2	2	1 441	
3	3	1 441	
4	4	1 11	
5	5	1 523	
6	6	0.5 356_729	
7	7	1 963	
8	8	1 899	
9	9	1 200	
10	10	1 523	
11	11	1 1111	
12	12	1 200	
13	13	1 869	
14	14	1 523	
15	15	1 386	
16	16	1 238	
17	17	1 790	
18	18	1 842	
19	19	1 230	
20	20	1 523	

AddressDIM_V1

	ADDRESS_ID	SUBURB	POSTCODE	STATE_CODE
1	526	Ashgrove	4060	QLD
2	531	Elimbah	4516	QLD
3	537	Lota	4179	QLD
4	553	Bracken Ridge	4017	QLD
5	566	Morningside	4170	QLD
6	581	Bracken Ridge	4017	QLD
7	585	Everton Park	4053	QLD
8	595	Chermside	4032	QLD
9	596	Waterford	4133	QLD
10	597	Holland Park	4121	QLD
11	600	Albany Creek	4035	QLD
12	604	Sunnybank	4109	QLD
13	613	Teneriffe	4005	QLD
14	621	Nundah	4012	QLD
15	624	Caboolture	4510	QLD
16	1425	Gordon	2906	ACT
17	1443	Moncrieff	2914	ACT
18	1446	Mawson	2607	ACT
19	1457	Phillip	2606	ACT
~	~	~	~	~

ClientBudgetDIM_V1

	CLIENT_BUDGET_ID	CLIENT_BUDGET	DESCRIPTION
1	1	Low	0 to 1000
2	2	Medium	1001 to 100000
3	3	High	100001 to 10000000

ClientDIM_V1

CLIENT_ID	WEIGHT_FACTOR	WISHLIST_GROUP_LIST
2804	5270	0.25 1_17_22_86
2805	5271	0 (null)
2806	5272	0 (null)
2807	5273	0.02941176470... 1_2_4_5_6_7_9_10_14...
2808	5274	0 (null)
2809	5275	0.05882352941... 1_2_3_4_7_14_16_17...
2810	5276	0 (null)
2811	5277	0 (null)
2812	5278	0.02941176470... 1_2_4_5_6_7_9_10_14...
2813	5279	0.25 1_14_26_74
2814	5280	1 2
2815	5281	0.125 1_2_5_6_7_17_19_20
2816	5282	0 (null)
2817	5283	0.02941176470... 1_2_4_5_6_7_9_10_14...

ClientWishBRIDGE_V1

CLIENT_ID	FEATURE_CODE
1	5202
2	5205
3	5208
4	5211
5	5216
6	5225
7	5227
8	5231
9	5234
10	5236
11	5244

PropertyFACT_V1

PROPERTY_ID	PROPERTY_SCALE_ID	TIME_ID	PROPERTY_TYPE	FEATURE_CATEGORY_ID	ADDRESS_ID	TOTAL_NUMBER_OF_PROPERTIES
1	168	3.202003	House	2	168	1
2	366	2.202004	Townhouse	1	366	1
3	593	3.202003	House	1	593	1
4	764	3.202002	House	3	764	1
5	768	2.202002	Apartment / Unit / Flat	1	768	1
6	979	2.201912	Apartment / Unit / Flat	1	979	1
7	1102	3.202004	House	1	1102	1
8	1198	2.202004	Apartment / Unit / Flat	1	1198	1
9	1211	2.202004	House	2	1211	1
10	1677	3.202004	House	1	1677	1
11	1719	2.202003	House	1	1719	1
12	1820	3.202001	Duplex	1	1820	1
13	1825	3.201911	House	1	1825	1
14	1902	2.202004	House	1	1902	1
15	1980	3.202003	House	1	1980	1
16	2088	2.202003	House	1	2088	1

VisitFACT_V1

	PROPERTY_ID	SEASON_ID	TIME_ID	PROPERTY_TYPE	ADDRESS_ID	PROPERTY_SCALE_ID	FEATURE_CATEGORY_ID	CLIENT_ID	TOTAL_NUMBER_OF_VISITS
1	5300	2	202004SATURDAY	Apartment / Unit / Flat	5300	2	1	5308	1
2	5400	2	202004SATURDAY	House	5400	3	2	5580	1
3	1345	2	202003SATURDAY	Apartment / Unit / Flat	1345	2	1	5051	1
4	1345	2	202003SATURDAY	Apartment / Unit / Flat	1345	2	1	5135	1
5	1379	2	202003TUESDAY	Apartment / Unit / Flat	1379	1	1	5035	1
6	1380	2	202003TUESDAY	Apartment / Unit / Flat	1380	2	1	5136	1
7	1383	2	202003WEDNESDAY	Apartment / Unit / Flat	1383	1	1	5139	1
8	1431	2	202003SATURDAY	House	1431	3	3	5134	1
9	1454	2	202003WEDNESDAY	House	1454	3	2	5128	1
10	1477	2	202003MONDAY	Apartment / Unit / Flat	1477	2	1	5053	1
11	1530	2	202003THURSDAY	Apartment / Unit / Flat	1530	2	1	5079	1
12	1558	2	202003MONDAY	Apartment / Unit / Flat	1558	2	1	5059	1
13	1576	2	202003THURSDAY	Apartment / Unit / Flat	1576	1	1	5035	1
14	1618	2	202003WEDNESDAY	Apartment / Unit / Flat	1618	2	1	5064	1
15	1618	2	202003WEDNESDAY	Apartment / Unit / Flat	1618	2	1	5135	1

RentFACT_V1

	PROPERTY_ID	TIME_ID	SEASON_ID	ADDRESS_ID	FEATURE_CATEGORY_ID	RENTAL_PERIOD_ID	PROPERTY_TYPE	PROPERTY_SCALE_ID	CLIENT_ID	TOTAL_NUMBER_OF_RENTS	TOTAL_RENTAL_FEES
1	2945	202001	1	2945	1	2	Apartment / Unit / Flat	2	3830	1	360
2	2965	202005	2	2965	1	2	Apartment / Unit / Flat	2	4862	1	350
3	3057	202004	2	3057	1	1	Apartment / Unit / Flat	2	3665	1	620
4	3083	202002	1	3083	1	2	Apartment / Unit / Flat	2	4691	1	650
5	3140	202003	2	3140	1	1	Apartment / Unit / Flat	2	3598	1	445
6	3190	202004	2	3190	2	1	House	2	3442	1	525
7	3204	202004	2	3204	1	1	Apartment / Unit / Flat	2	4074	1	455
8	3227	202003	2	3227	1	1	House	3	3582	1	580
9	3258	202004	2	3258	2	1	House	2	4942	1	390
10	3275	202004	2	3275	1	2	House	2	4326	1	438
11	3276	202001	1	3276	2	2	House	2	3673	1	470
12	3280	202002	1	3280	1	2	House	3	3435	1	575
13	3297	202001	1	3297	1	2	House	1	4765	1	250
14	3308	202003	2	3308	1	1	House	2	4955	1	525
15	3309	202004	2	3309	1	1	House	3	4956	1	580
16	3330	202005	2	3330	2	2	House	2	4352	1	310

SaleFACT_V1

	PROPERTY_ID	TIME_ID	SEASON_ID	ADDRESS_ID	FEATURE_CATEGORY_ID	PROPERTY_TYPE	PROPERTY_SCALE_ID	CLIENT_ID	TOTAL_NUMBER_OF_SALES	TOTAL_SALES
1	89202003	2	89	1	House	1	2	2688	1	685000
2	1100	202001	1	1100	1	House	3	2724	1	779950
3	2100	202002	1	2100	1	Townhouse	2	2805	1	320000
4	153	202001	1	153	1	House	2	3278	1	750000
5	196201912	1	196	1	House	2	3279	1	270000	
6	282	202001	1	282	1	House	3	2701	1	565000
7	296	202003	2	296	1	Apartment / Unit / Flat	1	2482	1	115000
8	302	201912	1	302	1	House	2	2571	1	650000
9	405	202002	1	405	2	Apartment / Unit / Flat	2	2812	1	399000
10	408	202002	1	408	1	House	3	2704	1	700000
11	422	202002	1	422	2	Townhouse	2	3293	1	315000
12	482	202002	1	482	1	House	3	2927	1	329000
13	501	202001	1	501	2	House	2	2495	1	700000
14	522	202003	2	522	1	House	3	2705	1	500000
15	528	202003	2	528	1	House	3	3175	1	400000
16	567	202003	2	567	1	House	2	2716	1	249000

ClientFACT_V1

	CLIENT_ID	CLIENT_BUDGET_ID	GENDER	TOTAL_NUMBER_OF_CLIENTS
1	3824	1	Male	1
2	4050	2	Male	1
3	4063	1	Female	1
4	4854	1	Female	1
5	5437	2	Male	1
6	5076	3	Female	1
7	3542	1	Female	1
8	4828	1	Male	1
9	5402	2	Female	1
10	4970	2	Male	1
11	2847	3	Male	1
12	2518	3	Female	1
13	3836	1	Female	1

AgentFACT_V1

	AGENT_ID	SALARY	GENDER	TOTAL_NUMBER_OF_AGENTS	TOTAL_SALARY
1	1056	190000	Female	1	190000
2	1058	180000	Male	1	180000
3	65	190000	Female	1	190000
4	219	175000	Male	1	175000
5	224	190000	Female	1	190000
6	280	190000	Male	1	190000
7	495	200000	Female	1	200000
8	535	195000	Male	1	195000
9	566	180000	Female	1	180000
10	626	200000	Female	1	200000
11	1103	195000	Male	1	195000
12	1168	200000	Female	1	200000
13	1272	195000	Male	1	195000
14	659	175000	Male	1	175000
15	691	200000	Female	1	200000
16	860	210000	Female	1	210000
17	1437	200000	Female	1	200000
18	1479	200000	Male	1	200000
19	1526	200000	Female	1	200000
20	1580	175000	Male	1	175000

RevenueFACT_V1

	AGENT_ID	PROPERTY_ID	TOTAL_REVENUE
1	1	50	549000
2	1	92	639000
3	2	1	650000
4	2	19	895000
5	2	41	580000
6	2	81	825000
7	6	129	249000
8	7	260	439000
9	9	162	340000
10	11	208	1675000
11	11	307	1034000
12	15	179	520000
13	16	730	525000
14	18	333	1000000
15	18	742	3690000
16	21	296	115000
17	24	400	375000
18	24	717	730000
19	26	410	329000
20	28	417	355000

Version-2**PropertyDIM2_V2**

	PROPERTY_ID	WEIGHT_FACTOR	ADVERTISEMENT_GROUP_LIST
1	2	1	16
2	3	1	16
3	4	1	16
4	6	1	16
5	7	1	16
6	8	1	16
7	9	1	16
8	10	1	16
9	12	1	16
10	14	1	16
11	15	1	16
12	16	1	16
13	17	1	16
14	20	1	16
15	21	1	23
16	22	1	16
17	23	1	16
18	25	1	16
19	26	1	23
20	27	1	16
21	28	1	16

PropertyAdvertisementBRIDGE_V2

	PROPERTY_ID	ADVERT_ID	COST
1	2894	16	1700000
2	2895	16	800000
3	2896	16	735000
4	2897	16	680000
5	2898	16	500000
6	2899	16	500000
7	2900	16	500000
8	2902	12	500000
9	2903	16	1100000
10	2300	16	985000
11	2302	16	977000
12	2303	16	949000
13	2305	20	899000
14	2306	16	898000
15	2307	16	879000
16	2506	16	1500000
17	2507	16	595000
18	2508	16	690000
19	2509	16	500000
20	2510	16	475000
21	2511	16	540000

AdvertisementDIM_V2

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

TimeDIM_V2

DATE_ID
15-11-19 00:24:09
15-11-19 00:36:00
15-11-19 00:57:40
15-11-19 01:16:36
15-11-19 01:39:00
15-11-19 02:40:54
15-11-19 08:08:55
15-11-19 08:32:26
15-11-19 08:42:51
15-11-19 09:26:30
15-11-19 10:05:16
15-11-19 10:53:48
15-11-19 11:00:22
15-11-19 11:59:07
15-11-19 12:13:42
15-11-19 12:21:43
15-11-19 12:24:01
15-11-19 13:15:35
15-11-19 17:02:51
15-11-19 17:20:47
15-11-19 17:46:37

PropertyDIM_V2

PropertyFeatureBRIDGE_V2

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

PropertyFeatureDIM_V2

	FEATURE_CODE	FEATURE_DESCRIPTION
1	1	Air conditioning
2	2	Built in wardrobes
3	3	Carpeted
4	4	City Views
5	5	Close to schools
6	6	Close to shops
7	7	Close to transport
8	8	Exhaust
9	9	Heating
10	10	Prestige Homes
11	11	Roller Door Access
12	12	Vacuum System
13	13	Car Parking - Surface
14	14	Ensuite
15	15	Open Fire Place
16	16	Study
17	17	Swimming Pool
18	18	Floorboards
19	19	Garden
20	20	Secure Parking
21	21	Terrace-Balcony

SaleDIM_V2

	PROPER...	AGENT_ID	CLIENT_ID	SALE_ID
1	1964	1077	2900	434
2	1896	1079	2901	435
3	1932	1083	2902	436
4	1998	1084	2903	437
5	1943	1087	2904	438
6	5	1195	2905	439
7	67	1195	2906	440
8	72	1195	2907	441
9	121	1202	2908	442
10	229	1202	2909	443
11	159	1202	2910	444
12	164	1202	2911	445
13	202	1202	2912	446
14	305	1202	2913	447
15	217	1203	2914	448
16	173	1203	2915	449
17	241	1204	2916	450
18	181	1207	2917	451
19	190	1208	2918	452
20	320	1209	2919	453
21	321	1210	2920	454

SaleHistoryDIM_V2

	SALE_ID	SALE_DATE	PRICE
1	434	22-03-20 04:41:47	1395000
2	435	16-01-20 15:37:51	275000
3	436	20-03-20 09:12:14	3490000
4	437	14-01-20 04:58:57	799000
5	438	22-02-20 15:31:38	2000000
6	439	29-01-20 18:16:23	1825000
7	440	14-03-20 22:45:36	380000
8	441	05-04-20 21:05:36	1695000
9	442	19-01-20 20:34:35	495000
10	443	09-03-20 17:54:46	565000
11	444	09-03-20 02:31:28	545000
12	445	19-02-20 11:01:35	280000
13	446	01-03-20 12:59:31	520000
14	447	25-02-20 00:32:40	440000
15	448	02-03-20 12:09:32	470000
16	449	24-01-20 07:10:46	565000
17	450	25-01-20 21:54:32	390000
18	451	28-01-20 17:39:01	950000
19	452	19-03-20 23:37:16	550000
20	453	15-02-20 00:08:42	349000
21	454	22-03-20 10:23:15	339000

RentDIM_V2

	PROPERTY_ID	AGENT_ID	CLIENT_ID	RENT_ID
1	6199	568	3713	331
2	6063	568	3714	332
3	6074	568	3715	333
4	6142	568	3716	334
5	6146	568	3717	335
6	5373	570	3718	336
7	5801	570	3719	337
8	5513	570	3720	338
9	5709	570	3721	339
10	5548	571	3722	340
11	5901	571	3723	341
12	5724	571	3724	342
13	6035	571	3725	343
14	5557	572	3726	344
15	5621	572	3727	345
16	5598	573	3728	346
17	5386	574	3729	347
18	5766	575	3730	348
19	6070	575	3731	349
20	5966	576	3732	350
21	5586	576	3733	351

RentHistoryDIM_V2

RE...	Y	RENT_START_DATE	RENT_END_DATE	PRICE
1	331	12-01-20 13:36:37	28-06-20 13:36:37	795
2	332	02-05-20 13:02:11	18-10-20 13:02:11	500
3	333	01-05-20 17:47:48	17-10-20 17:47:48	370
4	334	12-02-20 05:09:13	29-07-20 05:09:13	795
5	335	20-04-20 20:25:23	06-10-20 20:25:23	595
6	336	27-04-20 00:36:56	13-10-20 00:36:56	350
7	337	25-02-20 00:56:05	11-08-20 00:56:05	600
8	338	01-01-20 08:26:33	17-06-20 08:26:33	430
9	339	29-03-20 04:04:47	13-09-20 04:04:47	420
10	340	23-04-20 14:59:06	09-10-20 14:59:06	520
11	341	01-05-20 06:15:09	17-10-20 06:15:09	330
12	342	01-05-20 08:28:29	17-10-20 08:28:29	500
13	343	30-04-20 01:49:10	16-10-20 01:49:10	625
14	344	23-04-20 19:53:04	09-10-20 19:53:04	815
15	345	21-04-20 18:50:41	07-10-20 18:50:41	370
16	346	23-04-20 01:21:35	09-10-20 01:21:35	495
17	347	18-03-20 18:20:03	02-09-20 18:20:03	1100
18	348	18-03-20 03:27:59	02-09-20 03:27:59	430
19	349	24-01-20 14:32:48	10-07-20 14:32:48	330
20	350	16-04-20 11:12:42	02-10-20 11:12:42	320
21	351	16-01-20 06:50:28	02-07-20 06:50:28	490

GenderDIM_V2

GENDER
1 Male
2 Female

OfficeDIM_V2

	OFFICE_ID	OFFICE_NAME	AGENT_COUNT
1	916	Ray White Mount Gravatt	1
2	919	Ray White Nolan & Iken	1
3	937	Ray White Robina	4
4	955	Ray White Upper Coomera	7
5	965	Ray White at The Entertainment Quarter	1
6	966	Rayner Real Estate	1
7	985	Rental Master Pty Ltd	1
8	989	RichardsElliot Surry Hills	1
9	992	Richardson & Wrench Double Bay	1
10	993	Richardson & Wrench Elizabeth Bay / Potts Point	5
11	1010	Rockpool Real Estate	3
12	1023	Sandy Funston	1
13	1024	Santa Cruz Realty	1
14	1046	Space Property Agency	1
15	1051	St George Property Agents - Penshurst	1
16	1052	Stanley Samuels Property	3
17	1053	Starr Partners Kellyville	1
18	1062	Stone McMahons Point	1
19	1064	Stone Real Estate - Coomera	1
20	1068	Stone Real Estate Manly	4
21	1070	Stone Real Estate Newtown	3

AgentOfficeBRIDGE_V2

	AGENT_ID	OFFICE_ID
1	49	787
2	364	505
3	1245	593
4	1247	1091
5	365	1069
6	1563	502
7	964	235
8	2207	503
9	1249	43
10	58	227
11	1898	1070
12	61	438
13	1251	837
14	2210	1132
15	1899	656
16	1567	275
17	970	54
18	971	810
19	371	1071
20	1904	1096
21	692	1054

AgentDIM_V2

	AGENT_ID	SALARY	WEIGHT_FA...	OFFICE_GROUP_LIST
1	1	210000	1 658	
2	2	210000	1 441	
3	3	175000	1 441	
4	4	180000	1 11	
5	5	210000	1 523	
6	6	210000	0.5 356_729	
7	7	195000	1 963	
8	8	190000	1 899	
9	9	190000	1 200	
10	10	180000	1 523	
11	11	200000	1 1111	
12	12	210000	1 200	
13	13	175000	1 869	
14	14	195000	1 523	
15	15	200000	1 386	
16	16	200000	1 238	
17	17	200000	1 790	
18	18	175000	1 842	
19	19	210000	1 230	
20	20	210000	1 523	
21	21	200000	1 214	

AddressDIM_V2

	ADDRE...	STREET	SUBURB	POSTCODE	STATE_CODE
1	541 22 Borbridge Street	North Lakes	4509 QLD		
2	552 4/592 Ann Street	Fortitude Valley	4006 QLD		
3	554 68 Pleystowe Crescent	Hendra	4011 QLD		
4	569 10 Mitchelson Place	Regents Park	4118 QLD		
5	575 12 Edenderry Street	Manly West	4179 QLD		
6	594 13 Topaz Crescent	Logan Reserve	4133 QLD		
7	608 15 Harpullia Street	Acacia Ridge	4110 QLD		
8	618 405/29 Bank St	West End	4101 QLD		
9	623 52 Mansfield Place	Mansfield	4122 QLD		
10	625 7/9 Norwood Street (east Wing)	Toowong	4066 QLD		
11	628 6/91 Emperor Street	Annerley	4103 QLD		
12	1420 1a, 1b, 1c Morton Street	Weetangera	2614 ACT		
13	1432 4 Burnell Place	Monash	2904 ACT		
14	1439 12 Currey Street	Gowrie	2904 ACT		
15	1445 904/6 Graziers Lane	Belconnen	2617 ACT		
16	1453 9/50 Peter Cullen Way	Wright	2611 ACT		
17	1455 60/234 Flemington Road	Harrison	2914 ACT		
18	1467 68 Ashburner Street	Higgins	2615 ACT		
19	1471 1 Kennerley Street	Curtin	2605 ACT		
20	1472 3 Giordano Street	Denman Prospect	2611 ACT		
21	1474 42 Amaroo Street	Reid	2612 ACT		

ClientDIM_V2

	CLIENT_ID	MIN_BUDGET	MAX_BUDGET	WEIGHT_FACTOR	WISHLIST_GROUP_LIST
1	2500	450000	550000		0 (null)
2	2600	438750	536250		0 (null)
3	2700	495000	605000		0 (null)
4	2800	495000	605000		0 (null)
5	2900	1255500	1534500		0 (null)
6	3000	495000	605000		0 (null)
7	3100	1035000	1265000		0 (null)
8	3200	206100	251900		0 (null)
9	3300	922500	1127500		0 (null)
10	3400	600	720		0 (null)
11	3500	1000	1200		0 (null)
12	3600	490	588		0 (null)
13	3700	420	504		0 (null)
14	3800	475	570		0 (null)
15	3900	800	960		0 (null)
16	4000	280	336		0 (null)
17	4100	550	660		0 (null)
18	4200	430	516		0 (null)
19	4300	620	744		0 (null)
20	4400	460	552		0 (null)
21	4500	380	456		0 (null)

ClientWishBRIDGE_V2

	CLIENT_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

PropertyFACT_V2

	PROPERTY_DATE_ADDED	ADDRESS_ID	PROPERTY_ID	TOTAL_NUMBER_OF_PROPERTIES
1	25-11-19 17:42:33	129	129	1
2	01-04-20 14:41:16	114	114	1
3	25-03-20 20:21:21	154	154	1
4	29-11-19 02:02:25	87	87	1
5	25-02-20 14:08:39	88	88	1
6	09-02-20 19:24:25	89	89	1
7	27-04-20 14:31:44	94	94	1
8	29-12-19 17:11:39	189	189	1
9	25-04-20 21:42:47	303	303	1
10	28-01-20 11:25:23	564	564	1
11	27-11-19 05:34:34	565	565	1
12	30-03-20 19:05:28	633	633	1
13	22-03-20 14:28:03	694	694	1
14	03-12-19 08:02:52	611	611	1
15	03-03-20 04:54:28	619	619	1
16	11-03-20 05:15:21	620	620	1
17	25-04-20 10:29:23	649	649	1
18	11-04-20 00:54:28	537	537	1
19	18-01-20 15:39:04	695	695	1
20	24-04-20 03:44:02	655	655	1
21	14-04-20 04:26:48	888	888	1

VisitFACT_V2

	PROPERTY_ID	VISIT_DATE	ADDRESS_ID	CLIENT_ID	TOTAL_NUMBER_OF_VISITS
1	1380	10-03-20 06:44:59	1380	5136	1
2	1991	27-03-20 17:59:01	1991	5239	1
3	5345	08-04-20 20:50:19	5345	5308	1
4	1362	26-03-20 05:42:11	1362	5037	1
5	5571	07-04-20 13:57:12	5571	5470	1
6	5433	22-03-20 11:37:58	5433	5616	1
7	5615	26-03-20 21:21:59	5615	5456	1
8	5272	07-04-20 12:38:58	5272	5333	1
9	5445	12-04-20 20:58:28	5445	5427	1
10	5538	31-03-20 16:46:45	5538	5538	1
11	6112	12-04-20 10:48:11	6112	5465	1
12	6112	12-04-20 10:48:11	6112	5574	1
13	1427	17-03-20 14:11:19	1427	5142	1
14	1947	28-03-20 04:58:21	1947	5210	1
15	1605	08-03-20 12:46:43	1605	5092	1
16	5570	03-04-20 21:12:21	5570	5556	1
17	5383	27-03-20 23:54:41	5383	5456	1
18	5439	08-04-20 16:25:03	5439	5544	1
19	2222	21-03-20 14:02:22	2222	5233	1
20	2026	10-03-20 00:32:03	2026	5288	1
21	1973	14-03-20 14:09:17	1973	5210	1

RentFACT_V2

	ADDRESS_ID	PROPERTY_ID	RENT_START_DATE	CLIENT_ID	TOTAL_RENTAL_FEES	TOTAL_NUMBER_OF_RENTS
1	3798	3798	31-01-20 07:51:50	4303	1500	1
2	3844	3844	20-04-20 20:59:14	4870	795	1
3	3848	3848	26-01-20 04:35:19	3841	675	1
4	3852	3852	06-04-20 02:02:51	4485	695	1
5	4380	4380	17-03-20 11:47:25	4020	800	1
6	3935	3935	25-04-20 22:55:56	4905	1475	1
7	3891	3891	30-03-20 05:55:11	4465	650	1
8	3048	3048	21-04-20 06:16:58	3833	330	1
9	3217	3217	24-04-20 15:06:24	4777	450	1
10	5429	5429	26-01-20 04:12:16	3498	395	1
11	3589	3589	30-04-20 05:40:18	3693	1200	1
12	3305	3305	16-02-20 08:23:37	4935	540	1
13	5643	5643	22-01-20 03:22:33	4642	350	1
14	5126	5126	25-03-20 16:19:11	4660	580	1
15	5205	5205	29-04-20 08:19:07	3502	750	1
16	3107	3107	07-04-20 02:20:30	3832	330	1
17	5136	5136	02-03-20 06:42:32	4847	700	1
18	4515	4515	23-02-20 05:53:18	3806	410	1
19	5162	5162	03-05-20 12:08:46	4001	230	1
20	4614	4614	21-04-20 00:41:21	3822	275	1
21	5697	5697	02-05-20 11:36:18	3510	425	1

SaleFACT_V2

	ADDRESS_ID	SALE_DATE	PROPERTY_ID	CLIENT_ID	TOTAL_SALES	TOTAL_NUMBER_OF_SALES
1	350	14-01-20 11:52:55	350	2922	499000	1
2	684	24-02-20 14:31:10	684	2935	500000	1
3	609	31-01-20 23:54:05	609	3158	369000	1
4	528	14-03-20 02:32:11	528	3175	400000	1
5	768	27-03-20 18:50:26	768	3174	500000	1
6	809	10-02-20 20:33:55	809	3034	699000	1
7	1017	16-02-20 11:19:51	1017	2831	750000	1
8	1123	22-02-20 03:36:14	1123	2735	595000	1
9	1348	13-03-20 03:56:23	1348	2848	499000	1
10	1276	21-02-20 11:14:22	1276	2971	500000	1
11	2640	09-02-20 10:40:43	2640	2787	1200000	1
12	2498	28-01-20 20:59:23	2498	2539	530000	1
13	2874	09-02-20 07:35:28	2874	3088	545000	1
14	2115	01-02-20 06:45:27	2115	3340	145000	1
15	331	29-02-20 04:26:34	331	3282	775000	1
16	426	15-02-20 10:01:00	426	3283	635000	1
17	635	03-04-20 04:02:38	635	3047	450000	1
18	1139	08-04-20 23:32:13	1139	3063	799000	1
19	1441	21-01-20 03:18:48	1441	2743	699000	1
20	2554	26-01-20 05:39:46	2554	3334	285000	1
21	2688	01-03-20 10:36:08	2688	2996	698000	1

ClientFACT_V2

	CLIENT_ID	GENDER	TOTAL_NUMBER_OF_CLIENTS
1	2769	Male	1
2	2789	Female	1
3	2813	Male	1
4	2980	Male	1
5	3003	Female	1
6	3018	Male	1
7	3112	Male	1
8	3116	Female	1
9	2940	Female	1
10	2571	Female	1
11	2673	Female	1
12	2680	Male	1
13	3461	Female	1
14	4156	Male	1
15	4178	Female	1
16	3740	Female	1
17	3782	Female	1
18	3792	Male	1
19	3180	Male	1
20	3212	Female	1
21	3261	Male	1

AgentFACT_V2

	AGENT_ID	GENDER	TOTAL_NUMBER_OF_AGENTS	TOTAL_SALARY
1	1085	Male	1	180000
2	65	Female	1	190000
3	83	Male	1	175000
4	167	Male	1	210000
5	168	Male	1	200000
6	366	Female	1	190000
7	188	Male	1	190000
8	226	Male	1	190000
9	330	Male	1	200000
10	590	Female	1	200000
11	626	Female	1	200000
12	1662	Male	1	200000
13	1207	Female	1	210000
14	787	Female	1	200000
15	1445	Male	1	200000
16	1462	Female	1	200000
17	1508	Female	1	200000
18	1585	Female	1	200000
19	1850	Female	1	190000
20	1910	Male	1	190000
21	1996	Female	1	175000

RevenueFACT_V2

	AGENT_ID	PROPERTY_ID	TOTAL_REVENUE
1	1	50	549000
2	1	92	639000
3	2	1	650000
4	2	19	895000
5	2	41	580000
6	2	81	825000
7	6	129	249000
8	7	260	439000
9	9	162	340000
10	11	208	1675000
11	11	307	1034000
12	15	179	520000
13	16	730	525000
14	18	333	1000000
15	18	742	3690000
16	21	296	115000
17	24	400	375000
18	24	717	730000
19	26	410	329000
20	28	417	355000
21	28	529	199000

TASK C.3 Create the following reports using OLAP queries

a. Simple reports:

REPORT 1: Top k

- (a) **What are the top 10 suburbs with lowest rent per week for Apartment Property Type?**
- (b) The Apartment Property Type is the most generic property type and hence there might arise frequent questions from the staff about the suburbs with the lowest rent possible.

VERSION 1

```

SELECT * FROM(
SELECT A.suburb AS SUBURB,
P.property_type AS "PROPERTY TYPE",
SUM(R.total_rental_fees) AS "RENTAL FEES",
DENSE_RANK() OVER(ORDER BY SUM(R.total_rental_fees)) AS "RANK BY RENT"
FROM rentfact_v1 R,
propertytypedim_v1 P,
addressdim_v1 A
WHERE R.property_type = P.property_type
AND R.address_id = A.address_id
AND R.property_type like '%Apartment%'
GROUP BY A.suburb,
P.property_type)
WHERE "RANK BY RENT" <= 10;
  
```

VERSION 2

```

SELECT * FROM(
SELECT A.suburb AS SUBURB,
P.property_type AS "PROPERTY TYPE",
SUM(R.total_rental_fees) AS "RENTAL FEES",
DENSE_RANK() OVER(ORDER BY SUM(R.total_rental_fees)) AS "RANK BY RENT"
FROM rentfact_v2 R,
propertydim_v2 P,
addressdim_v2 A
WHERE R.property_id = P.property_id
AND R.address_id = A.address_id
AND P.property_type like '%Apartment%'
GROUP BY A.suburb,
P.property_type)
WHERE "RANK BY RENT" <= 10;
  
```

REPORT 2: Top n%

- (a) What are the top 10% visits by property type and the months for the year 2020?**
 (b) The management might be interested in knowing the property type and the month where they get the top 10% responses as visits for the properties to focus their policies.

VERSION 1

```

SELECT *
FROM (
SELECT
P.property_type AS "PROPERTY TYPE",
T.MONTH AS "TIME PERIOD",
SUM(V.total_number_of_visits) AS "TOTAL VISITS",
round(PERCENT_RANK() OVER(ORDER BY SUM(V.total_number_of_visits))), 5) AS "PERCENTAGE
RANK"
FROM TimeDIM_V1 T,
visitfact_v1 V,
propertytypedim_v1 P
WHERE T.time_id = V.time_id
AND P.property_type = V.property_type
AND T.YEAR = 2020
GROUP BY T.MONTH,
P.property_type)
WHERE "PERCENTAGE RANK" >= 0.9
ORDER BY "PERCENTAGE RANK" DESC;
  
```

VERSION 2

```

SELECT *
FROM (
SELECT
P.property_type AS "PROPERTY TYPE",
TO_CHAR(V.visit_date, 'MM') AS "TIME PERIOD",
SUM(V.total_number_of_visits) AS "TOTAL VISITS",
round(PERCENT_RANK() OVER(ORDER BY SUM(V.total_number_of_visits))), 5) AS "PERCENTAGE
RANK"
FROM TimeDIM_V2 T,
visitfact_v2 V,
propertydim_v2 P
WHERE T.date_id = V.visit_date
AND P.property_id = V.property_id
AND TO_CHAR(V.visit_date, 'YYYY') = 2020
GROUP BY TO_CHAR(V.visit_date, 'MM'),
P.property_type)
WHERE "PERCENTAGE RANK" >= 0.9
ORDER BY "PERCENTAGE RANK" DESC;
  
```

REPORT 3: Show All.**(a) Show average sales by states and by years**

- (b) The management might need this useful information about the average sales happening in each state over the years to come to identify the issues and the areas of improvement for those states which are lower in sales.

VERSION 1

```
SELECT A.state_code AS STATE,
T.year as "YEAR",
Round(SUM(S.total_sales)/SUM(S.total_number_of_sales)) AS "AVERAGE SALES"
FROM timedim_v1 T,
salefact_v1 S,
rentfact_v1 R,
addressdim_v1 A
WHERE S.address_id = A.address_id
AND t.time_id = s.time_id
GROUP BY A.state_code,
T.year
ORDER BY STATE;
```

VERSION 2

```
SELECT A.state_code AS STATE,
to_char(S.sale_date, 'YYYY') AS "YEAR",
Round(SUM(S.total_sales)/SUM(S.total_number_of_sales)) AS "AVERAGE SALES"
FROM TimeDIM_V2 T,
salefact_v2 S,
addressdim_v2 A
WHERE S.address_id = A.address_id
AND t.date_id = s.sale_date
GROUP BY A.state_code,
to_char(S.sale_date, 'YYYY')
ORDER BY STATE;
```

(d) Results**VERSION 1****REPORT 1**

	SUBURB	PROPERTY TYPE	RENTAL FEES	RANK BY RENT
1	Gilles Plains	Apartment / Unit / Flat	195	1
2	Victoria Park	Apartment / Unit / Flat	220	2
3	Lutwyche	Apartment / Unit / Flat	235	3
4	Scarborough	Apartment / Unit / Flat	240	4
5	Bardon	Apartment / Unit / Flat	245	5
6	Rosslea	Apartment / Unit / Flat	250	6
7	Chifley	Apartment / Unit / Flat	250	6
8	Kingswood	Apartment / Unit / Flat	250	6
9	Churchlands	Apartment / Unit / Flat	250	6
10	Greenacres	Apartment / Unit / Flat	260	7
11	Mosman Park	Apartment / Unit / Flat	260	7
12	Payneham	Apartment / Unit / Flat	265	8
13	Felixstow	Apartment / Unit / Flat	265	8
14	Doubleview	Apartment / Unit / Flat	280	9
15	Kippa-ring	Apartment / Unit / Flat	280	9
16	Midland	Apartment / Unit / Flat	280	9
17	Belmont	Apartment / Unit / Flat	280	9
18	Oaks Estate	Apartment / Unit / Flat	280	9

REPORT 2

	PROPERTY TYPE	TIME PERIOD	TOTAL VISITS	PERCENTAGE RANK
1	Apartment / Unit / Flat	03	224	1
2	House	03	180	0.9

REPORT 3

	STATE	YEAR	AVERAGE SALES
1	ACT	2019	678500
2	ACT	2020	744949
3	NSW	2019	644500
4	NSW	2020	931403
5	QLD	2019	541494
6	QLD	2020	733888
7	SA	2019	396667
8	SA	2020	540365
9	TAS	2020	617500
10	VIC	2019	662600
11	VIC	2020	880634
12	WA	2019	1647500
13	WA	2020	589944

VERSION 2**REPORT 1**

	SUBURB	PROPERTY TYPE	RENTAL FEES	RANK BY RENT
1	Gilles Plains	Apartment / Unit / Flat	195	1
2	Victoria Park	Apartment / Unit / Flat	220	2
3	Lutwyche	Apartment / Unit / Flat	235	3
4	Scarborough	Apartment / Unit / Flat	240	4
5	Bardon	Apartment / Unit / Flat	245	5
6	Rosslea	Apartment / Unit / Flat	250	6
7	Chifley	Apartment / Unit / Flat	250	6
8	Kingswood	Apartment / Unit / Flat	250	6
9	Churchlands	Apartment / Unit / Flat	250	6
10	Greenacres	Apartment / Unit / Flat	260	7
11	Mosman Park	Apartment / Unit / Flat	260	7
12	Payneham	Apartment / Unit / Flat	265	8
13	Felixstow	Apartment / Unit / Flat	265	8
14	Oaks Estate	Apartment / Unit / Flat	280	9
15	Kippa-ring	Apartment / Unit / Flat	280	9
16	Midland	Apartment / Unit / Flat	280	9
17	Belmont	Apartment / Unit / Flat	280	9
18	Doubleview	Apartment / Unit / Flat	280	9

REPORT 2

	PROPERTY TYPE	TIME PERIOD	TOTAL VISITS	PERCENTAGE RANK
1	Apartment / Unit / Flat	03	224	1
2	House	03	180	0.9

REPORT 3

	STATE	YEAR	AVERAGE SALES
1	ACT	2019	678500
2	ACT	2020	744949
3	NSW	2019	644500
4	NSW	2020	931403
5	QLD	2019	541494
6	QLD	2020	733888
7	SA	2019	396667
8	SA	2020	540365
9	TAS	2020	617500
10	VIC	2019	662600
11	VIC	2020	880634
12	WA	2019	1647500
13	WA	2020	589944

b. Reports with proper sub-totals:

REPORT 4 and REPORT 5: What are the sub-total and total rental fees from each suburb, time period, and property type? (You must use the Cube and Partial Cube operator)

CUBE

VERSION 1

```

SELECT
decode(GROUPING(A.suburb), 1, 'All Suburbs',
A.suburb) AS SUBURB,
decode(GROUPING(T.year), 1, 'All Years',
T.year) AS "YEARS",
decode(GROUPING(T.month), 1, 'All Months',
T.month) AS "MONTHS",
decode(GROUPING(P.property_type), 1, 'All Property Types',
P.property_type) AS "PROPERTY TYPE",
SUM(total_rental_fees) AS "RENTAL FEES"
FROM rentfact_v1 R,
timedim_v1 T,
propertytypedim_v1 P,
addressdim_v1 A
WHERE R.time_id = T.time_id
AND R.property_type = P.property_type
AND R.address_id = A.address_id
GROUP BY CUBE(A.suburb,T.year,T.Month,P.property_type)
ORDER BY A.suburb;

```

VERSION 2

```

SELECT
decode(GROUPING(A.suburb), 1, 'All Suburbs',
A.suburb) AS SUBURB,
decode(GROUPING(to_char(R.rent_start_date, 'YYYY')), 1, 'All Years',
to_char(R.rent_start_date, 'YYYY')) AS "YEARS",
decode(GROUPING(to_char(R.rent_start_date, 'MM')), 1, 'All Months',
to_char(R.rent_start_date, 'MM')) AS "MONTHS",
decode(GROUPING(P.property_type), 1, 'All Property Types',
P.property_type) AS "PROPERTY TYPE",
SUM(total_rental_fees) AS "RENTAL FEES"
FROM rentfact_v2 R,
TimeDIM_V2 T,
propertydim_v2 P,
addressdim_v2 A
WHERE R.rent_start_date = T.date_id
AND R.property_id = P.property_id
AND R.address_id = A.address_id
GROUP BY CUBE(A.suburb,
to_char(R.rent_start_date, 'YYYY'),
to_char(R.rent_start_date, 'MM'),
P.property_type)
ORDER BY A.suburb;

```

PARTIAL CUBE**VERSION 1**

```

SELECT
decode(GROUPING(A.suburb), 1, 'All Suburbs',
A.suburb) AS SUBURB,
decode(GROUPING(T.year), 1, 'All Years',
T.year) AS "YEARS",
decode(GROUPING(T.month), 1, 'All Months',
T.month) AS "MONTHS",
decode(GROUPING(P.property_type), 1, 'All Property Types',
P.property_type) AS "PROPERTY TYPE",
SUM(R.total_rental_fees) AS "RENTAL FEES"
FROM rentfact_v1 R,
timedim_v1 T,
propertytypedim_v1 P,
addressdim_v1 A
WHERE R.time_id = T.time_id
AND R.property_type = P.property_type
AND R.address_id = A.address_id
GROUP BY A.suburb, CUBE(T.year,T.Month,P.property_type)
ORDER BY A.suburb;

```

VERSION 2

```

SELECT
decode(GROUPING(A.suburb), 1, 'All Suburbs',
A.suburb) AS SUBURB,
decode(GROUPING(to_char(R.rent_start_date, 'YYYY')), 1, 'All Years',
to_char(R.rent_start_date, 'YYYY')) AS "YEARS",
decode(GROUPING(to_char(R.rent_start_date, 'MM')), 1, 'All Months',
to_char(R.rent_start_date, 'MM')) AS "MONTHS",
decode(GROUPING(P.property_type), 1, 'All Property Types',
P.property_type) AS "PROPERTY TYPE",
SUM(R.total_rental_fees) AS "RENTAL FEES"
FROM rentfact_v2 R,
TimeDIM_V2 T,
propertydim_v2 P,
addressdim_v2 A
WHERE R.rent_start_date = T.date_id
AND R.property_id = P.property_id
AND R.address_id = A.address_id
GROUP BY A.suburb, CUBE(to_char(R.rent_start_date, 'YYYY'),
to_char(R.rent_start_date, 'MM'),P.property_type)
ORDER BY A.suburb;

```

REPORT 6 using Roll-up

- (a) **What is the total Revenue and sub totals of all Agent offices from all Suburbs and all Genders?**
(b) This report helps the management to judge the agent offices by the total revenue they generate on the basis of the suburbs for performance tracking.

VERSION 1

```

SELECT
decode(GROUPING(o.office_name), 1, 'All Offices',
o.office_name) AS "OFFICE NAME",
decode(GROUPING(A.suburb), 1, 'All Suburbs',
A.suburb) AS "SUBURB",
decode(GROUPING(G.gender), 1, 'All Genders',
G.gender) AS "GENDER",
SUM(R.total_revenue) AS "TOTAL REVENUE"
FROM revenuefact_v1 R,
officedim_v1 o,
genderdim_v1 G,
addressdim_v1 A,
agentdim_v1 AT,
propertydim_v1 P,
agentofficebridge_v1 ao
WHERE AT.agent_id = ao.agent_id
AND ao.office_id = o.office_id
AND R.agent_id = AT.agent_id
AND R.property_id = P.property_id
AND P.address_id = A.address_id
GROUP BY ROLLUP(o.office_name, A.suburb, G.gender)
ORDER BY o.office_name;

```

VERSION 2

```

SELECT
decode(GROUPING(o.office_name), 1, 'All Offices',
o.office_name) AS "OFFICE NAME",
decode(GROUPING(A.suburb), 1, 'All Suburbs',
A.suburb) AS "SUBURB",
decode(GROUPING(G.gender), 1, 'All Genders',
G.gender) AS "GENDER",
SUM(R.total_revenue) AS "TOTAL REVENUE"
FROM revenuefact_v2 R,
officedim_v2 o,
genderdim_v2 G,
addressdim_v2 A,
agentdim_v2 AT,
propertydim_v2 P,
agentofficebridge_v2 ao
WHERE AT.agent_id = ao.agent_id
AND ao.office_id = o.office_id
AND R.agent_id = AT.agent_id
AND R.property_id = P.property_id
AND P.address_id = A.address_id
GROUP BY ROLLUP(o.office_name, A.suburb, G.gender)

```

ORDER BY o.office_name;

REPORT 7 using Partial Roll-up.

- (a) **What are the total Properties and subtotals from all states and all Property Types by time period?**
- (b) This report is useful for the management to know the total properties by states and property types, if the number has increased or decreased over the time and to know which property type is posted the most.

VERSION 1

```

SELECT
decode(GROUPING(A.State_code), 1, 'All States',
A.State_code) AS STATE,
decode(GROUPING(P.property_type), 1, 'All Property Types',
p.property_type) AS "PROPERTY TYPE",
decode(GROUPING(T.year), 1, 'All Years',
T.year) AS "YEARS",
decode(GROUPING(T.month), 1, 'All Months',
T.month) AS "MONTHS",
SUM(PF.total_number_of_properties) AS "TOTAL PROPERTIES"
FROM TimeDIM_V1 T,
PropertyFACT_V1 PF,
PropertyTypeDIM_V1 P,
AddressDIM_V1 A
WHERE PF.time_id = T.time_id
AND A.address_id= PF.address_id
AND p.property_type = pf.property_type
GROUP BY A.State_code, ROLLUP(P.property_type,T.year,T.month)
ORDER BY A.State_code;

```

VERSION 2

```

SELECT
decode(GROUPING(A.State_code), 1, 'All States',
A.State_code) AS STATE,
decode(GROUPING(P.property_type), 1, 'All Property Types',
p.property_type) AS "PROPERTY TYPE",
decode(GROUPING(to_char(PF.property_date_added, 'YYYY')), 1, 'All Years',
to_char(PF.property_date_added, 'YYYY')) AS "YEARS",
decode(GROUPING(to_char(PF.property_date_added, 'MM')), 1, 'All Months',
to_char(PF.property_date_added, 'MM')) AS "MONTHS",
SUM(PF.total_number_of_properties) AS "TOTAL PROPERTIES"
FROM TimeDIM_V2 T,
PropertyFACT_V2 PF,
PropertyDIM_V2 P,
AddressDIM_V2 A
WHERE PF.property_date_added = T.date_id
AND A.address_id= PF.address_id
AND p.property_id = pf.property_id
GROUP BY A.State_code, ROLLUP(P.property_type,

```

```
to_char(PF.property_date_added, 'YYYY'),
to_char(PF.property_date_added, 'MM'))
ORDER BY A.State_code;
```

d) Results

VERSION 1

Report 4

SUBURB	YEARS	MONTHS	PROPERTY TYPE	RENTAL FEES
1 Abbotsford	2020	01	Apartment / Unit / Flat	525
2 Abbotsford	2020	01	All Property Types	525
3 Abbotsford	2020	02	Apartment / Unit / Flat	400
4 Abbotsford	2020	02	All Property Types	400
5 Abbotsford	2020	03	Apartment / Unit / Flat	490
6 Abbotsford	2020	03	All Property Types	490
7 Abbotsford	2020	04	House	500
8 Abbotsford	2020	04	All Property Types	500
9 Abbotsford	2020	All Months	Apartment / Unit / Flat	1415
10 Abbotsford	2020	All Months	House	500
11 Abbotsford	2020	All Months	All Property Types	1915
12 Abbotsford All Years	01		Apartment / Unit / Flat	525
13 Abbotsford All Years	01		All Property Types	525
14 Abbotsford All Years	02		Apartment / Unit / Flat	400
15 Abbotsford All Years	02		All Property Types	400
16 Abbotsford All Years	03		Apartment / Unit / Flat	490
17 Abbotsford All Years	03		All Property Types	490
18 Abbotsford All Years	04		House	500

Report 5

SUBURB	YEARS	MONTHS	PROPERTY TYPE	RENTAL FEES
1 Abbotsford	2020	01	Apartment / Unit / Flat	525
2 Abbotsford	2020	01	All Property Types	525
3 Abbotsford	2020	02	Apartment / Unit / Flat	400
4 Abbotsford	2020	02	All Property Types	400
5 Abbotsford	2020	03	Apartment / Unit / Flat	490
6 Abbotsford	2020	03	All Property Types	490
7 Abbotsford	2020	04	House	500
8 Abbotsford	2020	04	All Property Types	500
9 Abbotsford	2020	All Months	Apartment / Unit / Flat	1415
10 Abbotsford	2020	All Months	House	500
11 Abbotsford	2020	All Months	All Property Types	1915
12 Abbotsford All Years	01		Apartment / Unit / Flat	525
13 Abbotsford All Years	01		All Property Types	525
14 Abbotsford All Years	02		Apartment / Unit / Flat	400
15 Abbotsford All Years	02		All Property Types	400
16 Abbotsford All Years	03		Apartment / Unit / Flat	490
17 Abbotsford All Years	03		All Property Types	490
18 Abbotsford All Years	04		House	500

Report 6

OFFICE NAME	SUBURB	GENDER	TOTAL REVENUE
1 ACTON Coogee	Beeliar	Female	599000
2 ACTON Coogee	Beeliar	Male	599000
3 ACTON Coogee	Beeliar	All Genders	1198000
4 ACTON Coogee	Hamilton Hill	Female	185000
5 ACTON Coogee	Hamilton Hill	Male	185000
6 ACTON Coogee	Hamilton Hill	All Genders	370000
7 ACTON Coogee	Spearwood	Female	640000
8 ACTON Coogee	Spearwood	Male	640000
9 ACTON Coogee	Spearwood	All Genders	1280000
10 ACTON Coogee	All Suburbs	All Genders	2848000
11 ACTON North	Floreat	Female	995000
12 ACTON North	Floreat	Male	995000
13 ACTON North	Floreat	All Genders	1990000
14 ACTON North	All Suburbs	All Genders	1990000
15 ADDISONS PROPERTY MANAGEMENT Ryde		Female	13200
16 ADDISONS PROPERTY MANAGEMENT Ryde		Male	13200
17 ADDISONS PROPERTY MANAGEMENT Ryde		All Genders	26400
18 ADDISONS PROPERTY MANAGEMENT All Suburbs		All Genders	26400

Report 7

STATE	PROPERTY TYPE	YEARS	MONTHS	TOTAL PROPERTIES
1 ACT	Apartment / Unit / Flat	2019	11	25
2 ACT	Apartment / Unit / Flat	2019	12	41
3 ACT	Apartment / Unit / Flat	2019	All Months	66
4 ACT	Apartment / Unit / Flat	2020	01	46
5 ACT	Apartment / Unit / Flat	2020	02	51
6 ACT	Apartment / Unit / Flat	2020	03	157
7 ACT	Apartment / Unit / Flat	2020	04	150
8 ACT	Apartment / Unit / Flat	2020	All Months	404
9 ACT	Apartment / Unit / Flat	All Years	All Months	470
10 ACT	Duplex	2019	12	1
11 ACT	Duplex	2019	All Months	1
12 ACT	Duplex	2020	01	1
13 ACT	Duplex	2020	04	4
14 ACT	Duplex	2020	All Months	5
15 ACT	Duplex	All Years	All Months	6
16 ACT	House	2019	11	15
17 ACT	House	2019	12	26
18 ACT	House	2019	All Months	41

VERSION 2**Report 4**

SUBURB	YEARS	MONTHS	PROPERTY TYPE	RENTAL FEES
1 Abbotsford	2020	01	Apartment / Unit / Flat	525
2 Abbotsford	2020	01	All Property Types	525
3 Abbotsford	2020	02	Apartment / Unit / Flat	400
4 Abbotsford	2020	02	All Property Types	400
5 Abbotsford	2020	03	Apartment / Unit / Flat	490
6 Abbotsford	2020	03	All Property Types	490
7 Abbotsford	2020	04	House	500
8 Abbotsford	2020	04	All Property Types	500
9 Abbotsford	2020	All Months	Apartment / Unit / Flat	1415
10 Abbotsford	2020	All Months	House	500
11 Abbotsford	2020	All Months	All Property Types	1915
12 Abbotsford	All Years	01	Apartment / Unit / Flat	525
13 Abbotsford	All Years	01	All Property Types	525
14 Abbotsford	All Years	02	Apartment / Unit / Flat	400
15 Abbotsford	All Years	02	All Property Types	400
16 Abbotsford	All Years	03	Apartment / Unit / Flat	490
17 Abbotsford	All Years	03	All Property Types	490
18 Abbotsford	All Years	04	House	500

Report 5

SUBURB	YEARS	MONTHS	PROPERTY TYPE	RENTAL FEES
1 Abbotsford	2020	01	Apartment / Unit / Flat	525
2 Abbotsford	2020	01	All Property Types	525
3 Abbotsford	2020	02	Apartment / Unit / Flat	400
4 Abbotsford	2020	02	All Property Types	400
5 Abbotsford	2020	03	Apartment / Unit / Flat	490
6 Abbotsford	2020	03	All Property Types	490
7 Abbotsford	2020	04	House	500
8 Abbotsford	2020	04	All Property Types	500
9 Abbotsford	2020	All Months	Apartment / Unit / Flat	1415
10 Abbotsford	2020	All Months	House	500
11 Abbotsford	2020	All Months	All Property Types	1915
12 Abbotsford	All Years	01	Apartment / Unit / Flat	525
13 Abbotsford	All Years	01	All Property Types	525
14 Abbotsford	All Years	02	Apartment / Unit / Flat	400
15 Abbotsford	All Years	02	All Property Types	400
16 Abbotsford	All Years	03	Apartment / Unit / Flat	490
17 Abbotsford	All Years	03	All Property Types	490
18 Abbotsford	All Years	04	House	500

Report 6

OFFICE NAME	SUBURB	GENDER	TOTAL REVENUE
1 ACTON Coogee	Beeliar	Female	599000
2 ACTON Coogee	Beeliar	Male	599000
3 ACTON Coogee	Beeliar	All Genders	1198000
4 ACTON Coogee	Hamilton Hill	Female	185000
5 ACTON Coogee	Hamilton Hill	Male	185000
6 ACTON Coogee	Hamilton Hill	All Genders	370000
7 ACTON Coogee	Spearwood	Female	640000
8 ACTON Coogee	Spearwood	Male	640000
9 ACTON Coogee	Spearwood	All Genders	1280000
10 ACTON Coogee	All Suburbs	All Genders	2848000
11 ACTON North	Floreat	Female	995000
12 ACTON North	Floreat	Male	995000
13 ACTON North	Floreat	All Genders	1990000
14 ACTON North	All Suburbs	All Genders	1990000
15 ADDISONS PROPERTY MANAGEMENT Ryde		Female	13200
16 ADDISONS PROPERTY MANAGEMENT Ryde		Male	13200
17 ADDISONS PROPERTY MANAGEMENT Ryde		All Genders	26400
18 ADDISONS PROPERTY MANAGEMENT All Suburbs		All Genders	26400

Report 7

...	PROPERTY TYPE	YEARS	MONTHS	TOTAL PROPERTIES
1 ACT	Apartment / Unit / Flat	2019	11	25
2 ACT	Apartment / Unit / Flat	2019	12	41
3 ACT	Apartment / Unit / Flat	2019	All Months	66
4 ACT	Apartment / Unit / Flat	2020	01	46
5 ACT	Apartment / Unit / Flat	2020	02	51
6 ACT	Apartment / Unit / Flat	2020	03	157
7 ACT	Apartment / Unit / Flat	2020	04	150
8 ACT	Apartment / Unit / Flat	2020	All Months	404
9 ACT	Apartment / Unit / Flat	All Years	All Months	470
10 ACT	Duplex	2019	12	1
11 ACT	Duplex	2019	All Months	1
12 ACT	Duplex	2020	01	1
13 ACT	Duplex	2020	04	4
14 ACT	Duplex	2020	All Months	5
15 ACT	Duplex	All Years	All Months	6
16 ACT	House	2019	11	15
17 ACT	House	2019	12	26
18 ACT	House	2019	All Months	41

c. Reports with moving and cumulative aggregates:

REPORT 8: What is the total number of clients and cumulative number of clients with a high budget in each year?

VERSION 1

```

SELECT year, SUM(total_count) AS "TOTAL COUNT", SUM(SUM(total_count)) OVER (ORDER BY year
rows unbounded preceding) AS CUMULATIVE FROM
((SELECT td.year, SUM(cf.total_number_of_clients) AS Total_count,
SUM(SUM(cf.total_number_of_clients)) OVER (ORDER BY td.year rows unbounded preceding) AS
Cumulative
FROM ClientFACT_V1 cf, clientbudgetDIM_v1 cbd, clientDIM_V1 cd, SaleFACT_V1 sf, TimeDIM_V1 td
WHERE cf.client_budget_id = cbd.client_budget_id
AND cf.client_id = cd.client_id
AND cd.client_id = sf.client_id
AND sf.time_id = td.time_id
AND cf.client_budget_id = 3
GROUP BY td.year)
UNION
(SELECT td.year, SUM(cf.total_number_of_clients) AS Total_count,
SUM(SUM(cf.total_number_of_clients)) OVER (ORDER BY td.year rows unbounded preceding) AS
Cumulative
FROM ClientFACT_V1 cf, clientbudgetDIM_v1 cbd, clientDIM_V1 cd, rentFACT_V1 sf, TimeDIM_V1 td
WHERE cf.client_budget_id = cbd.client_budget_id
AND cf.client_id = cd.client_id
AND cd.client_id = sf.client_id
AND sf.time_id = td.time_id
AND cf.client_budget_id = 3
GROUP BY td.year)
UNION
(SELECT td.year, SUM(cf.total_number_of_clients) AS Total_count,
SUM(SUM(cf.total_number_of_clients)) OVER (ORDER BY td.year rows unbounded preceding) AS
Cumulative
FROM ClientFACT_V1 cf, clientbudgetDIM_v1 cbd, clientDIM_V1 cd, visitFACT_V1 sf, TimeDIM_V1 td
WHERE cf.client_budget_id = cbd.client_budget_id
AND cf.client_id = cd.client_id
AND cd.client_id = sf.client_id
AND sf.time_id = td.time_id
AND cf.client_budget_id = 3
GROUP BY td.year))
GROUP BY year;
    
```

VERSION 2

```

SELECT YEAR, SUM(total_count) AS "TOTAL COUNT", SUM(SUM(total_count)) OVER (ORDER BY year
rows unbounded preceding) AS CUMULATIVE FROM
((SELECT TO_CHAR(td.date_id, 'YYYY') AS year, SUM(cf.total_number_of_clients) AS Total_count,
SUM(SUM(cf.total_number_of_clients)) OVER (ORDER BY TO_CHAR(td.date_id, 'YYYY') rows
unbounded preceding) AS Cumulative
FROM ClientFACT_V2 cf, clientDIM_V2 cd, SaleFACT_V2 sf, TimeDIM_V2 td
WHERE cf.client_id = cd.client_id
AND cd.client_id = sf.client_id
AND sf.sale_date = td.date_id
    
```

```

AND cd.max_budget BETWEEN 100001 AND 10000000
GROUP BY TO_CHAR(td.date_id, 'YYYY')
UNION
(SELECT TO_CHAR(td.date_id, 'YYYY') AS year, SUM(cf.total_number_of_clients) AS Total_count,
SUM(SUM(cf.total_number_of_clients)) OVER (ORDER BY TO_CHAR(td.date_id, 'YYYY') rows
unbounded preceding) AS Cumulative
FROM ClientFACT_V2 cf, clientDIM_V2 cd, RentFACT_V2 sf, TimeDIM_V2 td
WHERE cf.client_id = cd.client_id
AND cd.client_id = sf.client_id
AND sf.rent_start_date = td.date_id
AND cd.max_budget BETWEEN 100001 AND 10000000
GROUP BY TO_CHAR(td.date_id, 'YYYY'))
UNION
(SELECT TO_CHAR(td.date_id, 'YYYY') AS year, SUM(cf.total_number_of_clients) AS Total_count,
SUM(SUM(cf.total_number_of_clients)) OVER (ORDER BY TO_CHAR(td.date_id, 'YYYY') rows
unbounded preceding) AS Cumulative
FROM ClientFACT_V2 cf, clientDIM_V2 cd, VisitFACT_V2 sf, TimeDIM_V2 td
WHERE cf.client_id = cd.client_id
AND cd.client_id = sf.client_id
AND sf.visit_date = td.date_id
AND cd.max_budget BETWEEN 100001 AND 10000000
GROUP BY TO_CHAR(td.date_id, 'YYYY')))
GROUP BY year;

```

REPORT 9

- (a) What is the total revenue and cumulative revenue for each property type and for each state?**
- (b) This is a very important report to check overall revenue that each state generates by property type so that important policies and strategies can be focused.**

VERSION 1

```

SELECT PROPERTY_TYPE, STATE_CODE, SUM(TOTAL_REVENUE) AS TOTAL_REVENUE,
SUM(SUM(TOTAL_REVENUE)) over (order by property_type rows unbounded preceding) AS
CUMULATIVE FROM
((select pd.property_type,a.state_code,
sum(r.total_revenue) as Total_revenue,
sum(sum(r.total_revenue)) over (order by pd.property_type rows unbounded preceding) as Cumulative
from RevenueFACT_V1 r, PropertyTypeDIM_V1 pd, AddressDIM_V1 a, RentFACT_V1 rf, PropertyDIM_V1
p
where rf.property_id=p.property_id
and p.address_id=a.address_id
and p.property_id=r.property_id
and rf.property_type =pd.property_type
group by pd.property_type,a.state_code)
UNION
(select pd.property_type,a.state_code,
sum(r.total_revenue) as Total_salary,
sum(sum(r.total_revenue)) over (order by pd.property_type rows unbounded preceding) as Cumulative
from RevenueFACT_V1 r, PropertyTypeDIM_V1 pd, AddressDIM_V1 a, SaleFACT_V1 sf, PropertyDIM_V1
p
where sf.property_id=p.property_id

```

```

and p.address_id=a.address_id
and p.property_id=r.property_id
and sf.property_type =pd.property_type
group by pd.property_type,a.state_code))
GROUP BY PROPERTY_TYPE, STATE_CODE
ORDER BY PROPERTY_TYPE;

```

VERSION 2

```

select pd.property_type,a.state_code,
sum(r.total_revenue) as Total_revenue,
sum(sum(r.total_revenue)) over (order by pd.property_type rows unbounded preceding) as Cumulative
from RevenueFACT_V2 r, PropertyDIM_V2 pd, AddressDIM_V2 a
where r.property_id=pd.property_id
and pd.address_id=a.address_id
group by pd.property_type,a.state_code;

```

REPORT 10

- (a) **What is the total rental fee and moving aggregate in 2020 for Victoria?**
 (b) The moving aggregate helps the management to check the rental amount for each month as well the totals till then which might be useful for record keeping of the calculations or audit

VERSION 1

```

select t.month as Month,
a.state_code,
sum(r.total_rental_fees) as Total_Rent_fees,
avg(sum(r.total_rental_fees)) over (order by a.state_code,t.month rows 2 preceding)
as Avg_3_Months
from RentFact_V1 r,
AddressDIM_V1 a,
TimeDIM_V1 t
where r.Time_id = t.time_id
and r.address_id=a.address_id
and t.year ='2020'
and a.state_code='VIC'
group by t.month,a.state_code;

```

VERSION 2

```

select to_char(r.rent_start_date, 'MM') AS "Month",
a.state_code,
sum(r.total_rental_fees) as Total_Rent_fees,
avg(sum(r.total_rental_fees)) over (order by a.state_code,to_char(r.rent_start_date, 'MM') rows 2
preceding)
as Avg_3_Months
from RentFact_V2 r,
AddressDIM_V2 a,
TimeDIM_V2 t
where r.rent_start_date = t.date_id
and r.address_id=a.address_id
and to_char(r.rent_start_date, 'YYYY') ='2020'
and a.state_code='VIC'

```

group by to_char(r.rent_start_date, 'MM'),a.state_code;

d)Results

VERSION 1

Report 8

	YEAR	TOTAL C...	CUMULATIVE
1	2019	23	23
2	2020	1198	1221

Report 9

	PROPERTY_TYPE	STATE_CODE	TOTAL_REVENUE	CUMULATIVE
1	Apartment / Unit / Flat	ACT	25363217.14	25363217.14
2	Apartment / Unit / Flat	NSW	37443818	62807035.14
3	Apartment / Unit / Flat	QLD	59110765.72	121917800.86
4	Apartment / Unit / Flat	SA	3658370.01	125576170.87
5	Apartment / Unit / Flat	VIC	36154167.44	161730338.31
6	Apartment / Unit / Flat	WA	6133275.71	167863614.02
7	Block of Units	QLD	4329000	172192614.02
8	Development Site	VIC	1300000	173492614.02
9	Duplex	ACT	31920	173524534.02
10	Duplex	NSW	1930195.71	175454729.73
11	Duplex	QLD	2451661.43	177906391.16
12	Duplex	WA	306960	178213351.16
13	House	ACT	51464540.73	229677891.89

Report 10

	M...	STATE_CODE	TOTAL_RENT_FEES	AVG_3_MONTHS
1	01	VIC	43860	43860
2	02	VIC	39000	41430
3	03	VIC	51549	44803
4	04	VIC	92145	60898
5	05	VIC	14295	52663

VERSION 2

Report 8

	YEAR	TOTAL C...	CUMULATIVE
1	2019	23	23
2	2020	1198	1221

Report 9

PROPERTY_TYPE	STATE_CODE	TOTAL_REVENUE	CUMULATIVE
1 Apartment / Unit / Flat	ACT	25363217.14	25363217.14
2 Apartment / Unit / Flat	NSW	37443818	62807035.14
3 Apartment / Unit / Flat	QLD	59110765.72	121917800.86
4 Apartment / Unit / Flat	SA	3658370.01	125576170.87
5 Apartment / Unit / Flat	VIC	36154167.44	161730338.31
6 Apartment / Unit / Flat	WA	6133275.71	167863614.02
7 Block of Units	QLD	4329000	172192614.02
8 Development Site	VIC	1300000	173492614.02
9 Duplex	ACT	31920	173524534.02
10 Duplex	NSW	1930195.71	175454729.73
11 Duplex	QLD	2451661.43	177906391.16
12 Duplex	WA	306960	178213351.16
13 House	ACT	51464540.73	229677891.89

Report 10

M...	STATE_CODE	TOTAL_RENT_FEES	AVG_3_MONTHS
1 01	VIC	43860	43860
2 02	VIC	39000	41430
3 03	VIC	51549	44803
4 04	VIC	92145	60898
5 05	VIC	14295	52663

d. Reports with Partitions:

REPORT 11: 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.

VERSION 1

```

select s.property_type,
a.state_code,
t.year,
to_char(sum(s.total_sales)) as Sales$,
DENSE_RANK() OVER (PARTITION BY s.property_type
order by sum(s.total_sales) DESC) AS RANK_BY_PROPERTY,
DENSE_RANK() OVER (PARTITION BY a.state_code
order by sum(s.total_sales) DESC) as RANK_BY_STATE
from SaleFACT_V1 s,
propertytypedim_v1 p,
AddressDIM_V1 a,
TimeDIM_V1 t
where s.property_type=p.property_type
and a.address_id=s.address_id
and s.time_id=t.time_id
group by s.property_type, a.state_code,t.year;

```

VERSION 2

```

select p.property_type,
a.state_code,
to_char(S.sale_date, 'YYYY') AS "YEAR",
to_char(sum(s.total_sales)) as Sales$,
DENSE_RANK() OVER (PARTITION BY p.property_type
order by sum(s.total_sales) DESC) AS RANK_BY_PROPERTY,
DENSE_RANK() OVER (PARTITION BY a.state_code
order by sum(s.total_sales) DESC) as RANK_BY_STATE
from SaleFACT_V2 s,
propertydim_v2 p,
AddressDIM_V2 a,
TimeDIM_V2 t
where s.property_id=p.property_id
and a.address_id=s.address_id
and s.sale_date=t.date_id
group by p.property_type,
a.state_code,
to_char(S.sale_date, 'YYYY');

```

REPORT 12:

- (a) **Show ranking of each advertisement type based on total number of properties and ranking of each state based on total number of properties.**
- (b) It is a very useful report as it tells the management the advertisement types most advertised by the number of properties and state so that the management can focus on the marketing strategies in that way.

VERSION 1

```

select a.advert_name, ad.state_code, to_char(sum(pf.total_number_of_properties)) as Total_properties,
DENSE_RANK() OVER(PARTITION BY a.advert_name
order by sum(pf.total_number_of_properties) DESC) AS RANK_BY_Advertisements,
DENSE_RANK() OVER(PARTITION BY ad.state_code
order by sum(pf.total_number_of_properties) DESC) AS RANK_BY_states
from PropertyFACT_V1 pf, PropertyDIM2_V1 pd, PropertyAdvertisementBRIDGE_V1 pfb,
AdvertisementDIM_V1 a, AddressDIM_V1 ad
where pf.property_id=pd.property_id
and pd.property_id=pfb.property_id
and pfb.advert_id=a.advert_id
and pf.address_id=ad.address_id
group by a.advert_name, ad.state_code;

```

VERSION 2

```

select a.advert_name, ad.state_code, to_char(sum(pf.total_number_of_properties)) as Total_properties,
DENSE_RANK() OVER(PARTITION BY a.advert_name
order by sum(pf.total_number_of_properties) DESC) AS RANK_BY_Advertisements,
DENSE_RANK() OVER(PARTITION BY ad.state_code
order by sum(pf.total_number_of_properties) DESC) AS RANK_BY_states
from PropertyFACT_V2 pf, PropertyDIM2_V2 pd, PropertyAdvertisementBRIDGE_V2 pfb,
AdvertisementDIM_V2 a, AddressDIM_V2 ad
where pf.property_id=pd.property_id
and pd.property_id=pfb.property_id
and pfb.advert_id=a.advert_id
and pf.address_id=ad.address_id
group by a.advert_name, ad.state_code;

```

d)Results**VERSION 1****Report 11**

PROPERTY_TYPE	STATE_CODE	YEAR	SALES\$	RANK_BY_PROPERTY	RANK_BY_STATE
1 Apartment / Unit / Flat	QLD	2020	57619500	1	2
2 Apartment / Unit / Flat	VIC	2020	32899700	2	2
3 Apartment / Unit / Flat	NSW	2020	32025788	3	2
4 Apartment / Unit / Flat	ACT	2020	22330800	4	2
5 Apartment / Unit / Flat	WA	2020	5843000	5	2
6 Apartment / Unit / Flat	SA	2020	3359000	6	2
7 Apartment / Unit / Flat	ACT	2019	1074000	7	5
8 Apartment / Unit / Flat	NSW	2019	439000	8	9
9 Block of Units	QLD	2020	4329000	1	4
10 Development Site	VIC	2020	1300000	1	5
11 Duplex	NSW	2020	1837000	1	5
12 Duplex	QLD	2020	1333000	2	8
13 Duplex	QLD	2019	1100000	3	9
14 Duplex	WA	2020	300000	4	7
15 House	QLD	2020	167877900	1	1
16 House	VIC	2020	119785499	2	1
17 House	NSW	2020	63725850	3	1
18 House	ACT	2020	48806000	4	1

Report 12

ADVERT_NAME	STATE_CODE	TOTAL_PROPERTIES	RANK_BY_ADVERTISEMENTS	RANK_BY_STATES
1 Rent Apartment / Unit / Flat NSW	NSW	312	1	1
2 Rent Apartment / Unit / Flat VIC	VIC	252	2	2
3 Rent Apartment / Unit / Flat ACT	ACT	145	3	1
4 Rent Apartment / Unit / Flat QLD	QLD	144	4	4
5 Rent Apartment / Unit / Flat SA	SA	46	5	3
6 Rent Apartment / Unit / Flat WA	WA	28	6	2
7 Rent Apartment / Unit / Flat NT	NT	1	7	2
8 Rent Block of Units	NSW	1	1	13
9 Rent Duplex	QLD	6	1	9
10 Rent Duplex	ACT	2	2	7
11 Rent Duplex	NSW	1	3	13
12 Rent Duplex	WA	1	3	9
13 Rent House	QLD	198	1	2
14 Rent House	VIC	122	2	3
15 Rent House	NSW	74	3	4
16 Rent House	ACT	58	4	4
17 Rent House	SA	53	5	2
18 Rent House	WA	26	6	3

VERSION 2**Report 11**

PROPERTY_TYPE	STATE_CODE	YEAR	SALES\$	RANK_BY_PROPERTY	RANK_BY_STATE
1 Apartment / Unit / Flat	QLD	2020	57619500	1	2
2 Apartment / Unit / Flat	VIC	2020	32899700	2	2
3 Apartment / Unit / Flat	NSW	2020	32025788	3	2
4 Apartment / Unit / Flat	ACT	2020	22330800	4	2
5 Apartment / Unit / Flat	WA	2020	5843000	5	2
6 Apartment / Unit / Flat	SA	2020	3359000	6	2
7 Apartment / Unit / Flat	ACT	2019	1074000	7	5
8 Apartment / Unit / Flat	NSW	2019	439000	8	9
9 Block of Units	QLD	2020	4329000	1	4
10 Development Site	VIC	2020	1300000	1	5
11 Duplex	NSW	2020	1837000	1	5
12 Duplex	QLD	2020	1333000	2	8
13 Duplex	QLD	2019	1100000	3	9
14 Duplex	WA	2020	300000	4	7
15 House	QLD	2020	167877900	1	1
16 House	VIC	2020	119785499	2	1
17 House	NSW	2020	63725850	3	1
18 House	ACT	2020	48806000	4	1

Report 12

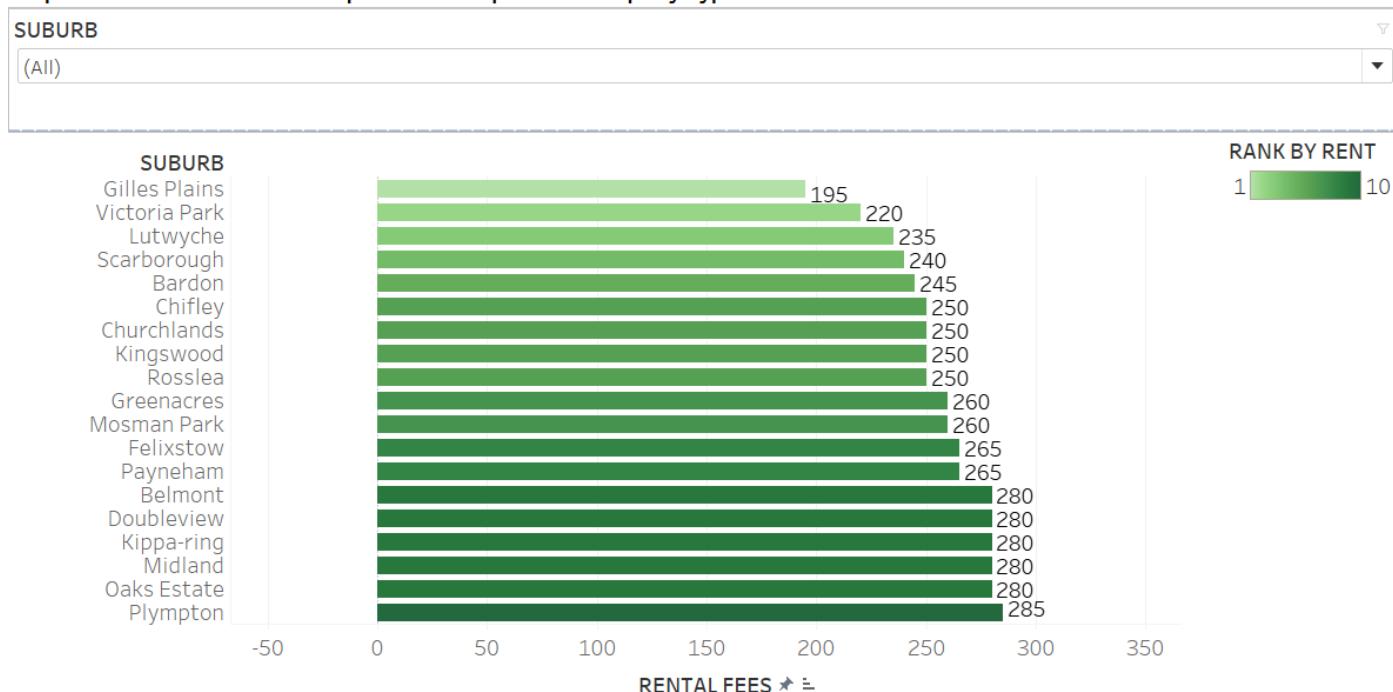
ADVERT_NAME	STATE_CODE	TOTAL_PROPERTIES	RANK_BY_ADVERTISEMENTS	RANK_BY_STATES
1 Rent Apartment / Unit / Flat NSW	NSW	312	1	1
2 Rent Apartment / Unit / Flat VIC	VIC	252	2	2
3 Rent Apartment / Unit / Flat ACT	ACT	145	3	1
4 Rent Apartment / Unit / Flat QLD	QLD	144	4	4
5 Rent Apartment / Unit / Flat SA	SA	46	5	3
6 Rent Apartment / Unit / Flat WA	WA	28	6	2
7 Rent Apartment / Unit / Flat NT	NT	1	7	2
8 Rent Block of Units	NSW	1	1	13
9 Rent Duplex	QLD	6	1	9
10 Rent Duplex	ACT	2	2	7
11 Rent Duplex	WA	1	3	9
12 Rent Duplex	NSW	1	3	13
13 Rent House	QLD	198	1	2
14 Rent House	VIC	122	2	3
15 Rent House	NSW	74	3	4
16 Rent House	ACT	58	4	4
17 Rent House	SA	53	5	2
18 Rent House	WA	26	6	3

Task C.4 Business Intelligence (BI) Reports

Reports used for generating the BI report:

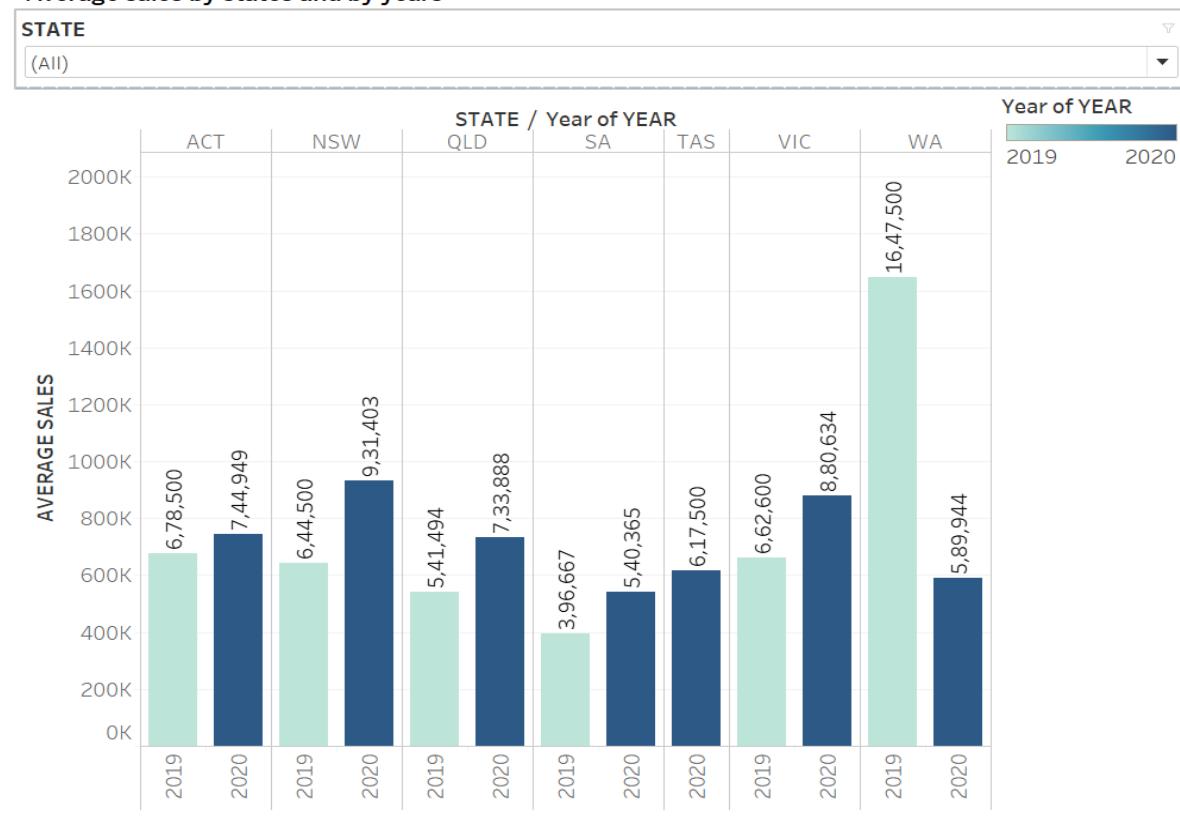
Report 1: Top 10 suburbs with lowest rent per week for Apartment Property Type

Top 10 suburbs with lowest rent per week for Apartment Property Type



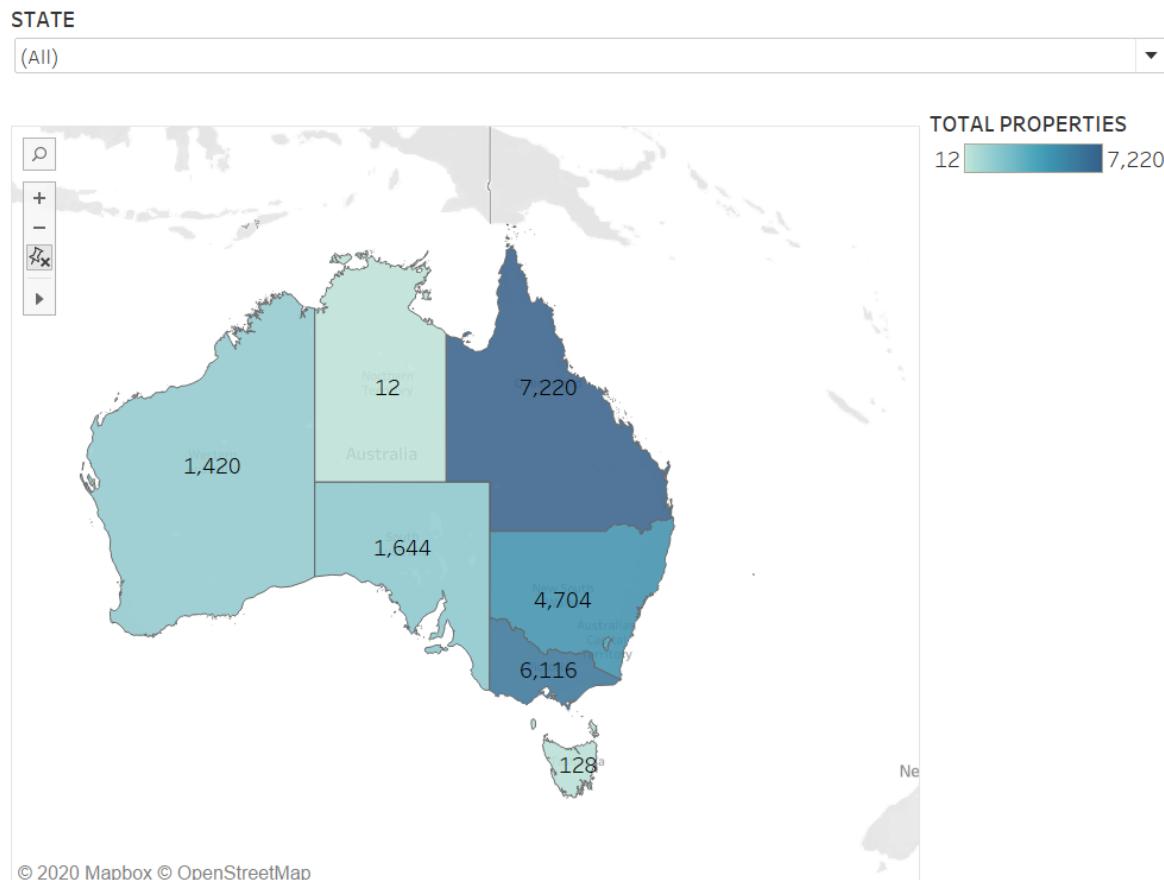
Report 3: Show avg sales by all states and years

Average sales by states and by years



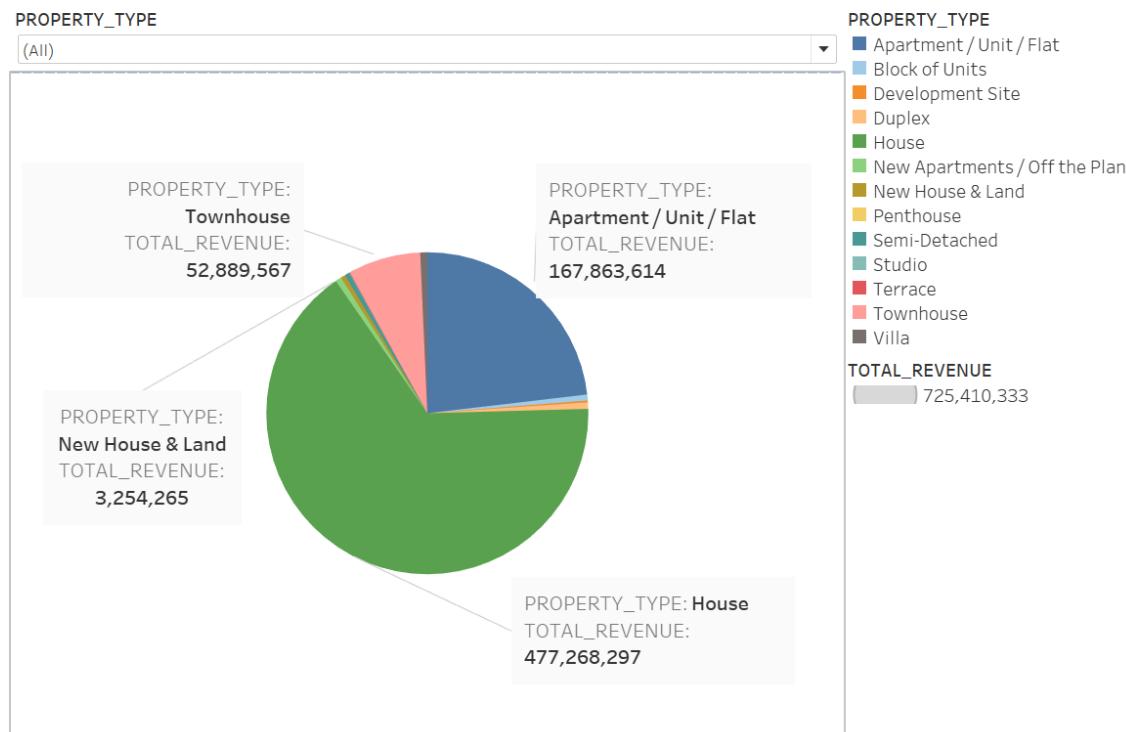
Report 7: Total Properties and subtotals from all states and all Property Types by time period.

Total Properties in Australia

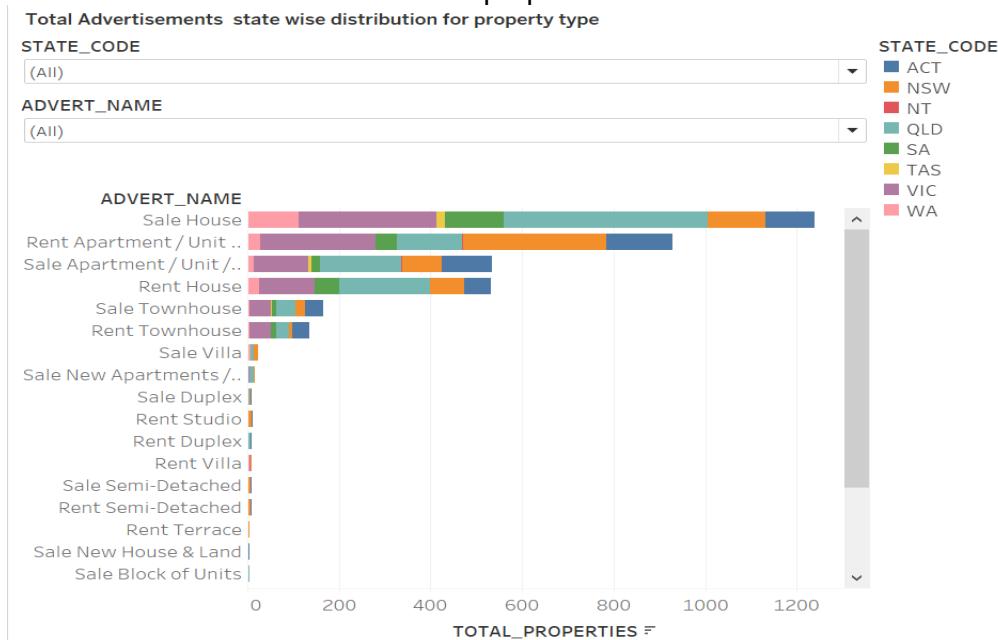


Report 9: What is the total revenue and cumulative revenue for each property type and for each state?

Total revenue based on property



Report 12: Show ranking of each advertisement type based on total number of properties and ranking of each state based on total number of properties.



Dashboard:

The Business Intelligence Report dashboard is as follows. Here, we have made the report dynamic where the user can select the year, state, suburb, and property type. As seen in the report, the user can generate the total revenue using all these filters which we found crucial for the management. The questions used for this report generations are as given above.

