MONASH University
Information Technology

GROUP ASSIGNMENT COVER SHEET

Student ID Number	Surname	Given Names
30379660	Yadav	**Trilochan**
30105234	Banga	**Rishabh**
30501725	Tomar	**Utkarsh**

***** Please include the names of all other group members.

Unit name and code	Business Intelligence And Data Warehousing	
Title of assignment	Major Assignment	
Lecturer/tutor	Agnes Haryanto	
Tutorial day and time	Thursday 1-3pm Lab 11, Thursday 10-12 am Lab 7 , Wednesday 10-12 Lab 13	**Campus - Caulfield**
Is this an authorised group assignment? ☒ **Yes** ☐ **No**		
Has any part of this assignment been previously submitted as part of another unit/course? ☐ **Yes** ☒ **No**		
Due Date *15/06/2020*		**Date submitted** *17/06/2020*

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 The material used can be from any source (staff, students or the internet, published and unpublished works).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
  - ii. submit it to a text matching software; and/or
  - iii. 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 .....Trilochan,Rishabh,Utkarsh *Date* 15/06/2020

* delete (iii) if not applicable

Signature Trilochan Yadav Date: 17/06/2020 Signature Rishabh Banga Date: 17/06/2020

Signature Utkarsh Tomar Date: 17/06/2020

Privacy Statement

The information on this form is collected for the primary purpose of assessing your assignment and ensuring the academic integrity requirements of the University are met. Other purposes of collection include recording your plagiarism and collusion declaration, attending to course and administrative matters and statistical analyses. If you choose not to complete all the questions on this form it may not be possible for Monash University to assess your assignment. You have a right to access personal information that Monash University holds about you, subject to any exceptions in relevant legislation. If you wish to seek access to your personal information or inquire about the handling of your personal information, please contact the University Privacy Officer: privacyofficer@adm.monash.edu.au

Contribution Declaration Form
(to be completed by all team members)

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
30379660	Trilochan Yadav Designing Star Schema,Data Cleaning,E/R Design, OLAP , Screenshots	33.33
30105234	Rishabh Banga Designing Star Schema,Data Cleaning,E/R Design, OLAP , BI Reports	33.33
30501725	Utkarsh Tomar Designing Star Schema,Data Cleaning,E/R Design, OLAP , Report Writing	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 SIGNATURE

Signatures

Trilochan Yadav

Rishabh Banga

Utkarsh Tomar

Date

Day Month Year

17 /06 /2020

FIT5195

BUSINESS INTELLIGENCE AND DATA WAREHOUSING

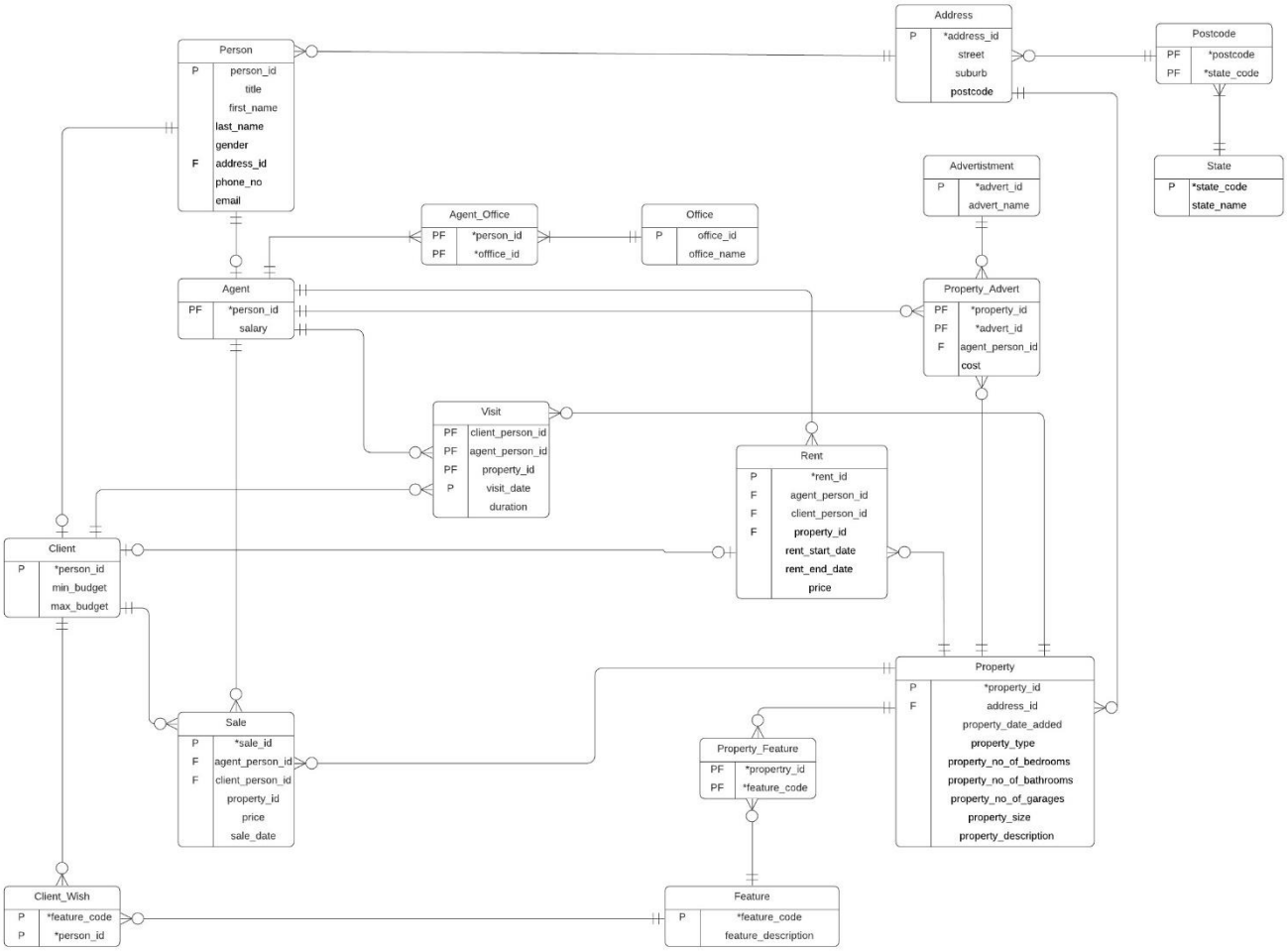
ASSIGNMENT

SUBMITTED BY :
ID : 30379660
30501725
30105234

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

a) Designing E/R diagram of the operational database , as can be seen from the operational database there are 17 tables , the database is mainly about properties, their sales history, rent history and visit history as well as client and agent details. Agents have salary and work in an agent office. A property has property features , client also has wish which contains feature which they desire in a property.

The E/R diagram was designed using Lucidchart.



b) Data Exploration and Data Cleaning

Data errors are common in most operational databases and should be handled carefully before making Business Intelligence Reports to get the correct results. Before data cleaning , we need to explore the operational database.

Checking Person Table –

SELECT * FROM MONRE.PERSON;

The data looks good as seen from output.

SQL | Fetched 50 rows in 0.144 seconds

	PERSON_ID	TITLE	FIRST_NAME	LAST_NAME	GENDER	ADDRESS_ID	PHONE_NO	EMAIL
1	977	Mr	Burton	Jonsson	Male	6637	7261979891	bjonssonr4@yellowpages.com
2	978	Mr	Gustave	Adamolli	Male	6638	7382918072	gadamollir5@hud.gov
3	980	Mrs	Niall	Thormann	Male	6639	3736835748	nthormannr7@ucoz.com
4	981	Dr	Franky	Plowman	Male	6640	5803253575	fplowmanr8@guardian.co.uk
5	982	Mr	Adolpho	Tregien	Male	6641	8155537176	atregienr9@goodreads.com
6	983	Ms	Kate	De la Yglesias	Female	6642	5442002141	kdelayglesiasra@archive.org
7	984	Ms	Elisha	Scroxton	Female	6643	1546936656	escroxtonrb@newyorker.com
8	986	Dr	Haven	Insko	Male	6644	9991758807	hinskord@nyu.edu
9	987	Dr	Bidget	Delhay	Female	6645	3118984485	bdelhayre@tmall.com
10	988	Ms	Valle	Vedekhin	Male	6646	5464335197	vvedekhinrf@ezinearticles.com
11	989	Mr	Opaline	Fiske	Female	6647	9944871590	ofiskerg@gravatar.com
12	991	Ms	Rozina	Oats	Female	6648	6608059959	roatsri@people.com.cn
13	992	Ms	Kirbie	Causier	Female	6649	6867932714	kcausierrj@sciencedirect.com
14	994	Ms	Thea	Hatrick	Female	6651	4382435040	thatrickrl@hp.com
15	995	Mrs	Roby	Gaylord	Female	6652	5826900770	rgaylordrm@auda.org.au
16	997	Mrs	Saloma	Wagge	Female	6653	4914988506	swaggero@amazonaws.com
17	998	Mrs	Stanfield	Iacobetto	Male	6654	8273280612	siacobettorp@phpbb.com
18	999	Dr	Toby	Hawking	Female	6655	9431798589	thawkingrq@economist.com

Now , we need to count the number of records in Person table .

SELECT COUNT(*) FROM MONRE.PERSON;

Query Result

SQL | All Rows Fetched: 1 in 0.01 s

	COUNT(*)
1	7000

Now ,checking for duplicate records using the below command:

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

Query Result

SQL | All Rows Fetched: 1 in 0.009 seconds

	PERSON_ID	COUNT(*)
1	6995	4

Now seeing the duplicate records, using this command.

SELECT * FROM MONRE.PERSON WHERE PERSON_ID = 6995;

Query Result

SQL | All Rows Fetched: 4 in 0.01 seconds

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

Error – 1 We have found duplicate records in Monre.Person table with Person_id = 6995.

Now looking for other errors in Person table like invalid Address as each person has an address associated with them where they live , by using this command.

After cleaning Person table , the results are following.

SELECT COUNT(*) FROM PERSON;

Script Output | Query Result

SQL | All Rows Fetched: 1 in 0.01 secor

	COUNT(*)
1	6996

SELECT * FROM PERSON WHERE PERSON_ID = 7001;

Script Output | Query Result

SQL | All Rows Fetched: 0 in 0.008 seconds

PERSON...	TITLE	FIRST_...	LAST_N...	GENDER	ADDRES...	PHONE_...	EMAIL

SELECT ADDRESS_ID FROM MONRE.PERSON WHERE ADDRESS_ID NOT IN (Select DISTINCT ADDRESS_ID FROM MONRE.ADDRESS);

Query Result

All Rows Fetched: 1 in 0.021 seconds

	ADDRESS_ID
1	13205

SELECT * FROM MONRE.PERSON WHERE ADDRESS_ID = 13205;

Query Result

All Rows Fetched: 1 in 0.006 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

Error - 2 We have found person with id = 7001 has no name, invalid phone number , invalid address and no email so it's an unnecessary record.

Now doing data cleaning on Person table.

CREATE TABLE PERSON AS SELECT DISTINCT * FROM MONRE.PERSON WHERE PERSON_ID <> 7001;

Now exploring MONRE.Address table, by using the following command :

SELECT ADDRESS_ID ,COUNT(*) FROM MONRE.ADDRESS GROUP BY ADDRESS_ID HAVING COUNT(*) > 1;

Script Output Query Result

All Rows Fetched: 0 in 0.012 seconds

ADDRES...	COUNT(*)

There are no errors found in Address table by using other commands also.

Now exploring MONRE.POSTCODE table by using the following commands:

SELECT POSTCODE FROM MONRE.POSTCODE WHERE STATE_CODE NOT IN (SELECT DISTINCT STATE_CODE FROM MONRE.STATE);

Script Output Query Result

All Rows Fetched: 0 in 0.007 seconds

POSTCO...	STATE_...

No errors were found in MONRE.postcode table as well.

Exploring STATE table now , using the following commands:

SELECT * FROM MONRE.STATE;

Script Output × Query Result ×

SQL | All Rows Fetched: 9 in 0.008 seconds

	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
9	(null)	Unknown

Error – 3 A state with null code and unknown name is not possible in real world scenario so has to be removed .

Now doing data cleaning for Monre.state table by using following commands.

CREATE TABLE STATE AS SELECT * FROM MONRE.STATE WHERE STATE_CODE IS NOT NULL;

After cleaning,the results are below:

SELECT * FROM STATE;

Script Output × Query Result ×

SQL | All Rows Fetched: 8 in 0.005 seconds

	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

Exploring Monre.Agent table now by using following commands:

SELECT COUNT(*) FROM MONRE.AGENT;

Script Output × Query Result ×

SQL | All Rows Fetched: 1 in 0.006 seconds

	COUNT(*)
1	2469

SELECT COUNT(DISTINCT PERSON_ID) FROM MONRE.AGENT;

Script Output × Query Result ×

SQL | All Rows Fetched: 1 in 0.007 second

	COUNT(DISTINCTPERSON_ID)
1	2469

So , far Agent table looks good , now checking for salary for Agents.

SELECT PERSON_ID FROM MONRE.AGENT WHERE SALARY < 0;

Script Output × Query Result ×

SQL | All Rows Fetched: 2 in 0.007

	PERSON_ID
1	6844
2	6000

Error – 4 Two agents have negative salary which is not possible as salary can not be negative in real world and maybe zero sometimes 0 for interns , so these agents should be removed , as they affect aggregate values also .

Cleaning Agent table , using following command:

CREATE TABLE AGENT AS SELECT DISTINCT * FROM MONRE.AGENT WHERE SALARY >= 0;

After cleaning , there are no agents with negative salary.

SELECT PERSON_ID FROM MONRE.AGENT WHERE SALARY < 0;

Script Output | Query Result

SQL | All Rows Fetched: 0 in 0.006 seconds

PERSON...

Exploring Agent_Office table, using following commands we found no errors :

SELECT COUNT(*) FROM MONRE.AGENT_OFFICE WHERE PERSON_ID = 6844 OR PERSON_ID = 6000 ;

No errors found in Agent_Office table.

Exploring Office table we did not find any errors.

Exploring Visit table , using following commands, we see there is a visit in future.

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

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

Both these command give the same output.

Script Output | Query Result

SQL | All Rows Fetched: 1 in 0.007 seconds

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

Error – 5 We have found a visit whose agent_id , client_id are not in agent and client tables and visit_date is very far in future and duration is also defined which is not possible in real world .

Cleaning the visit table using this command :

CREATE TABLE VISIT AS SELECT * FROM MONRE.VISIT WHERE CLIENT_PERSON_ID <> 6000;

After cleaning Visit table , we have the following results:

SELECT * FROM VISIT WHERE CLIENT_PERSON_ID = 6000;

Script Output | Query Result

SQL | All Rows Fetched: 0 in 0.009 seconds

CLIENT_...	AGENT_...	PROPER...	VISIT_D...	DURATION

Now exploring Client table , using following commands:

SELECT * FROM MONRE.CLIENT WHERE MIN_BUDGET > MAX_BUDGET;

Script Output | Query Result

SQL | All Rows Fetched: 3 in 0.006 seconds

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

Error – 6 We have found 3 clients with minimum budget greater than max budget which is illogical , so these records have to be removed.

Doing further exploration

SELECT * FROM MONRE.CLIENT WHERE PERSON_ID NOT IN(SELECT PERSON_ID FROM MONRE.PERSON);

Script Output | Query Result

SQL | All Rows Fetched: 1 in 0.007 seconds

	PERSON_ID	MIN_BUDGET	MAX_BUDGET
1	7000	8500	15050

Error – 7 As a client has to be present in person table i.e. be a living person , so this record is not acceptable.

Cleaning Client table , using the following commands:

CREATE TABLE CLIENT AS SELECT * FROM MONRE.CLIENT WHERE MIN_BUDGET <= MAX_BUDGET AND PERSON_ID <> 7000;

After Cleaning , the results are below:

SELECT * FROM CLIENT WHERE MIN_BUDGET > MAX_BUDGET;

Script Output | Query Result

SQL | All Rows Fetched: 0 in 0.006 seconds

PERSON...	MIN_BU...	MAX_BU...

SELECT * FROM CLIENT WHERE PERSON_ID = 7000;

Script Output × Query Result ×

SQL | All Rows Fetched: 0 in 0.004 seconds

PERSON...	MIN_BU...	MAX_BU...

Exploring Monre.sale table using the following commands:

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

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

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

SELECT * FROM MONRE.SALE WHERE SALE_DATE > SYSDATE;

SELECT * FROM MONRE.SALE WHERE PRICE < 0 ;

No errors were found in Monre.sale , so we use the table as it is .

Exploring Rent table ,using following commands :

SELECT * FROM MONRE.RENT WHERE RENT_START_DATE > RENT_END_DATE;

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

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

Script Output × Query Result ×

SQL | All Rows Fetched: 1 in 0.004 seconds

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

Error – 8 We have found a rental record that has invalid client and agent id as well as start date is later than end date.

Using the following command to clean the Rent table

CREATE TABLE RENT AS SELECT * FROM MONRE.RENT WHERE RENT_START_DATE < RENT_END_DATE;

After cleaning , the result is shown below:

SELECT * FROM RENT WHERE RENT_START_DATE > RENT_END_DATE;

Script Output × Query Result ×

SQL | All Rows Fetched: 0 in 0.009 seconds

RENT_ID	AGENT_...	CLIENT_...	PROPER...	RENT_S...	RENT_E...	PRICE

Exploring Client Wish , Feature and Property Feature using following commands we found no error:

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

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

SELECT COUNT(*) FROM MONRE.FEATURE;

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

Now exploring Property table using following commands :

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

Query... x

SQL | All Rows Fetched: 2 in 0.057 seconds

	PROPERTY_ID	COUNT(*)
1	6177	4
2	6179	16

Error – 9 There are duplicate properties in property table which need to be removed.

Cleaning the property table , we get :

CREATE TABLE PROPERTY AS SELECT DISTINCT * FROM MONRE.PROPERTY;

After cleaning , there are no duplicate records.

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

Script Output x Query Result x

SQL | All Rows Fetched: 0 in 0.017 seconds

PROPER...	COUNT(*)

Exploring Property_Advert and Advertisement tables using the following commands no errors were found :

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

SELECT ADVERT_ID FROM MONRE.PROPERTY_ADVERT WHERE ADVERT_ID NOT IN (SELECT ADVERT_ID FROM MONRE.ADVERTISEMENT);

SELECT * FROM MONRE.PROPERTY_ADVERT WHERE COST < 0;

SELECT COUNT(*) FROM MONRE.ADVERTISEMENT;

SELECT COUNT(DISTINCT ADVERT_ID) FROM MONRE.ADVERTISEMENT;
```

Creating Property_Advert and Advertisement as it is.

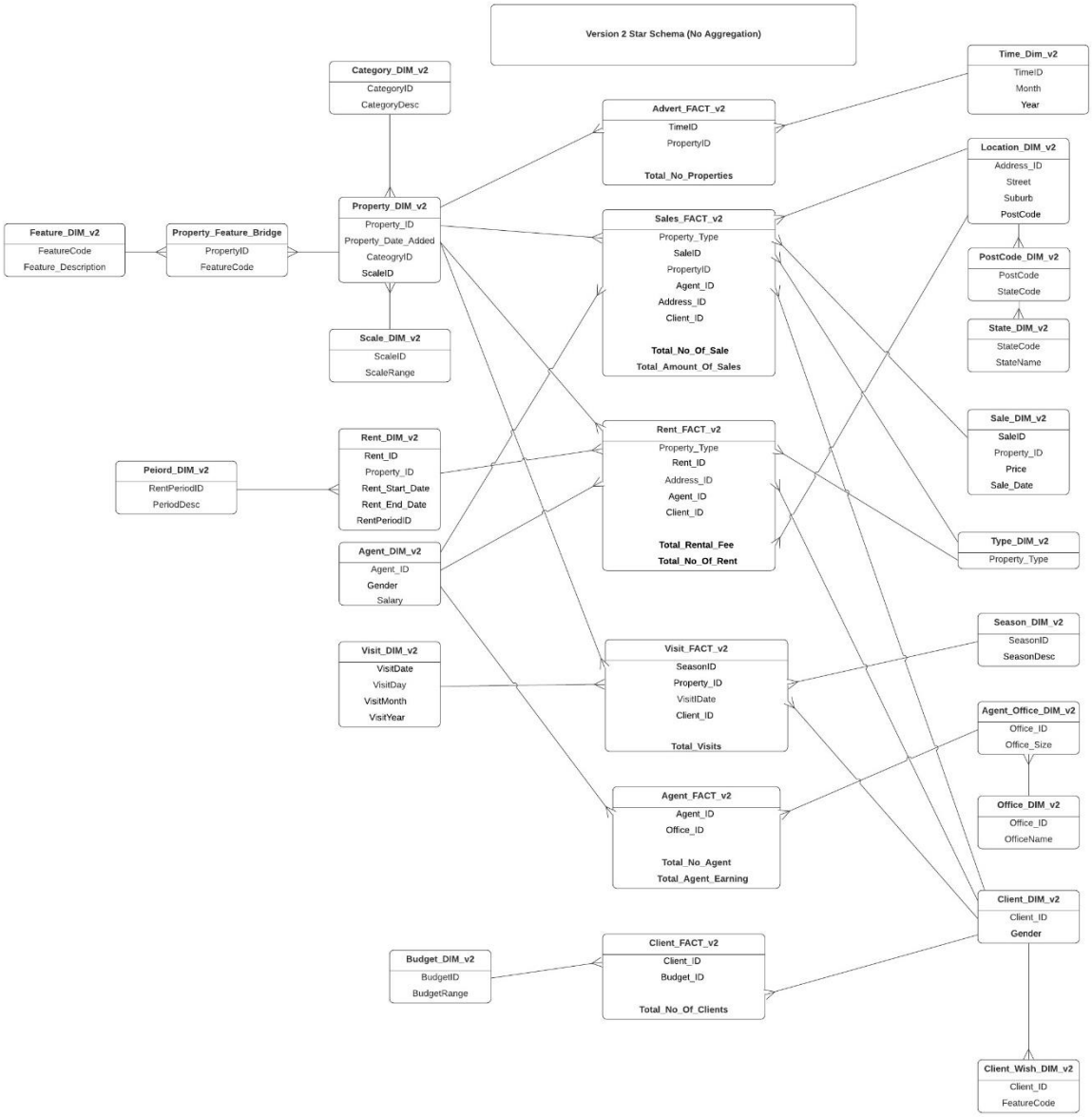
```
CREATE TABLE PROPERTY_ADVERT AS SELECT * FROM MONRE.PROPERTY_ADVERT;

CREATE TABLE ADVERTISEMENT AS SELECT * FROM MONRE.ADVERTISEMENT;
```

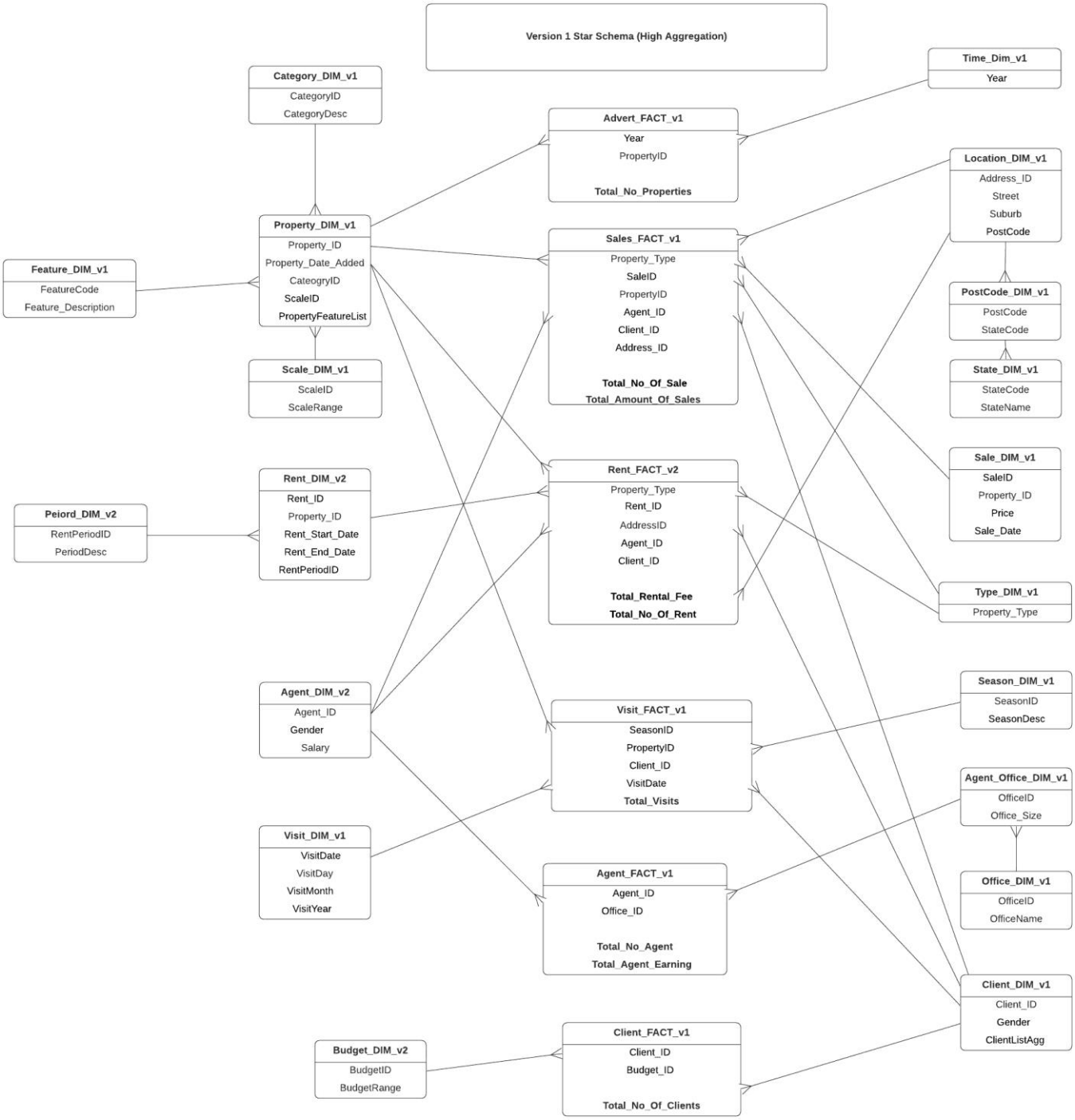
c) Two Versions Of The Star Schema

Below is the Star Schema for Level – 0 (No Aggregation) . We have 6 fact tables named :

Advert_Fact_v2 , Sales_Fact_v2,Rent_Fact_v2, Visit_Fact_v2,Agent_Fact_v2 and Client_Fact_v2 to handle different topics of aggregation.



Below is the star schema for version – 1 (High Aggregation) :



d) We chose Hierarchy in our Star Schema as it gets our dimension table in normalised which results in no redundancy of data and also helps in simplified query processing , we have used hierarchy for Location ,

postcode and state . We have also used hierarchy for Office and AgentOffice and also hierarchy for Scale dimension and Category dimension with Property_Dim.

e) We have taken Rent_Dim and Sale_Dim as Type 2 SCD as they record all history for a property's sale and rent since start with a new rent_id and sale_id.

f) The difference between Version 2 (No Aggregation) and Version 1 (High Aggregation) is that Time_dim for version 1 has only Year attribute as compared to timeID for version 2. Property_Dim v1 has PropertyFeatureList instead of Property_Feature_Bridge table in Version 2. There is no Client_Wish dimension ,instead client has clientlistagg.

TASK C.1 Implement the two versions star/snowflake schema using SQL

a) SQL statements to create Version-1 Star Schema

First All Dimension tables are created ,the commands are below:

```
--Creating Version 1 Star Schema

--Creating Feature Dimension

CREATE TABLE Feature_Dim_V1 AS

SELECT * FROM monre.FEATURE;

--Creating Property Dimension

DROP TABLE Property_Dim_temp_v1;

CREATE TABLE Property_Dim_temp_v1 AS

SELECT property.PROPERTY_ID,PROPERTY_DATE_ADDED,count(f.property_id) as
no_of_features,NVL(LISTAGG (F.Feature_code, '_') Within Group (Order By f.feature_code),'No
Features')  As PropertyFeatureList,

property_no_of_bedrooms FROM PROPERTY full outer join property_feature f on  property.property_id =
f.property_id

group by property.PROPERTY_ID,PROPERTY_DATE_ADDED,property_no_of_bedrooms ;

alter table Property_Dim_temp_v1  add(CategoryID VARCHAR(20));

alter table Property_Dim_temp_v1 add(ScaleID VARCHAR(20));

UPDATE Property_Dim_temp_v1

set ScaleID =

(case when property_no_of_bedrooms <= 1 then 'Extra Small'

  when property_no_of_bedrooms >= 2 and property_no_of_bedrooms < 3  then 'Small'

  when property_no_of_bedrooms >= 3 and property_no_of_bedrooms <= 6 then 'Medium'

  when property_no_of_bedrooms >= 6 and property_no_of_bedrooms <= 10 then 'Large'

  when property_no_of_bedrooms > 10  then 'Extra Large' end);UPDATE Property_Dim_temp_v1

set CategoryID =

(case when no_of_features >= 0 and no_of_features <= 9 then 'Very Basic'
```

```
  when no_of_features >= 10 and no_of_features <= 20 then 'Standard'
  when no_of_features > 20  then 'Luxurious' end);

CREATE TABLE Property_Dim_v1 as select * from Property_Dim_temp_v1;
--Creating Table PropertyCategory Dimension
--This table defines property category based on number of features it has
CREATE TABLE Category_Dim_v1
( CategoryID VARCHAR(15) ,
  Cat_Feature_Range VARCHAR(15));

--Populating PropertyCategory Dimension
INSERT INTO Category_Dim_V1 VALUES('Very Basic','0-10');
INSERT INTO Category_Dim_V1 VALUES('Standard','10-20');
INSERT INTO Category_Dim_V1 VALUES('Luxurious','More Than 20');
-- Creating Table Scale Dimension
--This table specifies the Property Scale based on Property Size
CREATE TABLE Scale_Dim_V1
( ScaleID VARCHAR(15) ,
  ScaleRange VARCHAR(15));

--Populating ScaleDim
INSERT INTO Scale_Dim_V1 VALUES('Extra Small','0-1');
INSERT INTO Scale_Dim_V1 VALUES('Medium','3-6');
INSERT INTO Scale_Dim_V1 VALUES('Large','6-10');
INSERT INTO Scale_Dim_V1 VALUES('Extra Large','More Than 10');
--Creating Rental Dimension
CREATE TABLE RENT_DIM_V1 AS
SELECT RENT_ID,PROPERTY_ID,RENT_START_DATE,RENT_END_DATE FROM RENT;

ALTER TABLE RENT_DIM_V1 ADD(RENTPERIODID VARCHAR(20));

UPDATE RENT_DIM_V1 SET RENTPERIODID = 'Short' WHERE ((RENT_END_DATE -
RENT_START_DATE) > 0 AND (RENT_END_DATE - RENT_START_DATE)  < 180);
```

```
UPDATE RENT_DIM_V1 SET RENTPERIODID = 'Medium' WHERE ((RENT_END_DATE -
RENT_START_DATE) >= 180 AND (RENT_END_DATE - RENT_START_DATE)  < 365);

UPDATE RENT_DIM_V1 SET RENTPERIODID = 'Short' WHERE ((RENT_END_DATE -
RENT_START_DATE) >= 365);


--Creating Location Dimension

CREATE TABLE LOCATION_Dim_V1 AS

SELECT * FROM ADDRESS;


--Creating Table PostCode Dimension

CREATE TABLE POSTCODE_Dim_V1 AS

SELECT * FROM POSTCODE;

--Creating Table State Dimension

CREATE TABLE STATE_Dim_V1 AS

SELECT * FROM State;

--Creating Type Dimension

--This table defines Property type like Apartment / House

CREATE TABLE Type_Dim_V1 AS

SELECT DISTINCT PROPERTY_TYPE FROM PROPERTY;

--Creating TimeDim

--this table stores advertisement month and year for properties

CREATE TABLE TIME_DIM_V1 AS

SELECT DISTINCT TO_CHAR(property_date_added,'YYYY') AS Year FROM PROPERTY;

--Creating Table SEASON  Dimension

--This table contains Season  data

CREATE TABLE SEASON_DIM_V1

(SeasonID VARCHAR(20),

SeasonDesc VARCHAR(20));



--Populating Season Dimension

INSERT INTO SEASON_DIM_V1 VALUES('Spring','Sep,Oct,Nov');

INSERT INTO SEASON_DIM_V1 VALUES('Summer','Dec,Jan,Feb');
```

```
INSERT INTO SEASON_DIM_V1 VALUES('Autumn','Mar,Apr,May');

INSERT INTO SEASON_DIM_V1 VALUES('Winter','Jun,Jul,Aug');

--Creating ClientBudget Dimension

CREATE TABLE BUDGET_DIM_V1

( BUDGETID VARCHAR(20),

BUDGETRANGE VARCHAR(20));

--Populating Budget_dimesion

INSERT INTO BUDGET_DIM_V1 VALUES('Low','0-1000');

INSERT INTO BUDGET_DIM_V1 VALUES('Medium','1001-100000');

INSERT INTO BUDGET_DIM_V1 VALUES('High','100001- 10000000');

--Creating Agent_Office DImension

CREATE TABLE AGENT_OFFICE_DIM_V1 AS

SELECT distinct OFFICE_ID,count(person_id) no_of_agents  FROM AGENT_OFFICE group by
office_id;

alter table agent_office_dim_v1 add(Office_Size Varchar(20));

UPDATE agent_office_dim_v1

set Office_Size =

(case when no_of_agents  >= 0 and  no_of_agents <= 3 then 'Small'

  when  no_of_agents >= 4 or no_of_agents <= 12 then 'Medium'

  when no_of_agents > 12 then 'Big'  end);

-- Creating Office Dimension

CREATE TABLE OFFICE_DIM_V1 AS

SELECT * FROM OFFICE;

--Creating Agent Dimension

CREATE TABLE AGENT_DIM_V1 AS

SELECT AGENT.PERSON_ID AS AGENT_ID,SALARY,GENDER FROM AGENT,PERSON WHERE
AGENT.PERSON_ID = PERSON.PERSON_ID;

--Creating Ssle Dimension

CREATE TABLE SALE_DIM_V1 AS SELECT

SALE_ID,SALE_DATE,PROPERTY_ID,PRICE FROM SALE WHERE CLIENT_PERSON_ID IS NOT
NULL;

--Creating Client Dimension
```

```
CREATE TABLE CLIENT_DIM_V1 AS SELECT CLIENT.PERSON_ID AS
CLIENT_ID,GENDER,NVL(LISTAGG (W.Feature_code, '_') Within Group (Order By
W.feature_code),'No Wishes')  As ClientListAgg

FROM  CLIENT LEFT JOIN PERSON ON CLIENT.PERSON_ID = PERSON.PERSON_ID  LEFT JOIN
CLIENT_WISH W ON client.person_id = W.person_id

group by CLIENT.PERSON_ID ,GENDER ;
```

Now All Fact tables are created ,the commands are below:

```
--Creating FACT TABLES NOW

-- Creating Advert_Fact_V2

-- This table contains aggeregate values for property advertisements

CREATE TABLE ADVERT_FACT_V1 AS

SELECT TO_CHAR(property_date_added,'YYYY') As Year,property_id , Count(property_id) AS
Total_No_Properties

from property group by TO_CHAR(property_date_added,'YYYY') ,property_id;

--Creating Temp Fact For Client

CREATE TABLE CLIENTTEMPFACT_v1 AS SELECT PERSON_ID AS CLIENT_ID ,MAX_BUDGET
FROM CLIENT;

ALTER TABLE CLIENTTEMPFACT_v1 ADD(BUDGETID VARCHAR(20));

UPDATE CLIENTTEMPFACT_v1

set BudgetID =

(case when Max_budget >= 0 and Max_Budget <= 1000 then 'Low'

  when max_budget >= 1001 and max_budget <= 100000 then 'Medium'

  when max_budget >= 100001 and max_budget <= 10000000 then 'High' end);


--Creating Client fact

CREATE TABLE CLIENT_FACT_V1 AS

SELECT BUDGETID,CLIENT_ID ,COUNT(CLIENT_ID) AS TOTAL_NO_CLIENTS FROM
CLIENTTEMPFACT_v1 GROUP BY BUDGETID,CLIENT_ID;


SELECT BUDGETID,COUNT(TOTAL_NO_CLIENTS) AS TC FROM CLIENT_FACT_V1 GROUP BY
BUDGETID;


drop table visittempfact_v1;

--Creating Visit Temp fact

CREATE TABLE VISITTEMPFACT_v1 AS
```

```
SELECT VISIT_DATE,PROPERTY_ID,CLIENT_PERSON_ID AS CLIENT_ID FROM VISIT;

ALTER TABLE VISITTEMPFACT_v1 ADD(SEASONID VARCHAR(20));

UPDATE VISITTEMPFACT_v1
set SeasonID =
(case when to_char(visit_date,'mm') >= 9 and to_char(visit_date,'mm') <= 11 then 'Spring'
  when to_char(visit_date,'mm') >= 12 or to_char(visit_date,'mm') <= 02 then 'Summer'
  when to_char(visit_date,'mm') >= 3 and to_char(visit_date,'mm') <= 5 then 'Autumn'
  when to_char(visit_date,'mm') >= 6 and to_char(visit_date,'mm') <= 8 then 'Winter' end);

--creating  visitfact table
CREATE TABLE VISIT_FACT_V1 AS
SELECT SEASONID,VISIT_DATE,PROPERTY_ID,CLIENT_ID,COUNT(*) AS TOTAL_VISITS
FROM VISITTEMPFACT  GROUP BY SEASONID,VISIT_DATE,PROPERTY_ID,CLIENT_ID;
--Creating Agent Fact Table
CREATE TABLE AGENT_FACT_V1 AS
SELECT OFFICE_ID,agent.person_id as agent_id,count(agent.person_id) as total_no_agent,sum(salary) as
total_agent_earning from
agent full outer join agent_office  on agent.person_id = agent_office.person_id group by
office_id,agent.person_id;
--Creating Temp Rental Fact Table
CREATE TABLE TEMPRENTFACT_v1 AS
SELECT property.property_id,rent_id,property.address_id,property_type,rent.agent_person_id as agent_id ,
rent.client_person_id  as client_id ,
sum(round((rent_end_date - rent_start_date) * (price/7),2)) as total_rental_fee,count(rent_id) as
total_no_of_rent
from property ,rent where property.property_id = rent.rent_id
group by
property.property_id,rent_id,property.address_id,property_type,rent.agent_person_id,rent.client_person_id ;
--Creating rent fact version 2 (low aggregation)
CREATE TABLE RENT_FACT_V1 AS SELECT * FROM TEMPRENTFACT_v1;
 --Creating Temp Sales fact Table
CREATE TABLE TEMPSALESFACT_v1 AS
```

```
SELECT property.property_id,sale_id,property.address_id,property_type,sale.agent_person_id as agent_id ,
sale.client_person_id  as client_id,sum(price) as tota_amount_of_sales

from property,sale where property.property_id = sale.property_id

and sale.client_person_id is not null

group by property.property_id,sale_id,property.address_id,property_type,sale.agent_person_id  ,
sale.client_person_id;

CREATE TABLE SALES_FACT_V1 AS SELECT * FROM TEMPSALESFACT_v1;
```

b) SQL statements to create Version-2 Star Schema

First All Dimension tables are created ,the commands are below:

```
--Creating Level 0 Star Schema

--Creating Feature Dimension

CREATE TABLE Feature_Dim_V2 AS

SELECT * FROM FEATURE;

--Creating Property Bridge Table

CREATE TABLE Property_Feature_Bridge AS

SELECT * FROM PROPERTY_FEATURE;

--Creating Property Dimension

CREATE TABLE Property_Dim_temp_v2 AS

SELECT property.PROPERTY_ID,PROPERTY_DATE_ADDED,count(property_feature.property_id) as
no_of_features,property_no_of_bedrooms FROM

PROPERTY full outer join property_feature on  property.property_id = property_feature.property_id

group by property.PROPERTY_ID,PROPERTY_DATE_ADDED,property_no_of_bedrooms ;

alter table Property_Dim_temp_v2  add(CategoryID VARCHAR(20));

alter table Property_Dim_temp_v2 add(ScaleID VARCHAR(20));

UPDATE Property_Dim_temp_v2

set ScaleID =

(case when property_no_of_bedrooms <= 1 then 'Extra Small'

  when property_no_of_bedrooms >= 2 and property_no_of_bedrooms < 3  then 'Small'

  when property_no_of_bedrooms >= 3 and property_no_of_bedrooms <= 6 then 'Medium'

  when property_no_of_bedrooms >= 6 and property_no_of_bedrooms <= 10 then 'Large'

  when property_no_of_bedrooms > 10  then 'Extra Large' end);
```

```
UPDATE Property_Dim_temp_v2
set CategoryID =
(case when no_of_features >= 0 and no_of_features <= 9 then 'Very Basic'
  when no_of_features >= 10 and no_of_features <= 20 then 'Standard'
  when no_of_features > 20  then 'Luxurious' end);
CREATE TABLE PERIOD_DIM_V2 (
RENTPERIODID VARCHAR(20) ,
PERIODDESC VARCHAR(20));
INSERT INTO PERIOD_DIM_V2 VALUES('Short','0-6');
INSERT INTO PERIOD_DIM_V2 VALUES('Medium','6-12');
INSERT INTO PERIOD_DIM_V2 VALUES('Long','More than 12');
--Creating Rental Dimension
CREATE TABLE RENT_DIM_V2 AS
SELECT RENT_ID,PROPERTY_ID,RENT_START_DATE,RENT_END_DATE FROM RENT;
ALTER TABLE RENT_DIM_V2 ADD(RENTPERIODID VARCHAR(20));
UPDATE RENT_DIM_V2 SET RENTPERIODID = 'Short' WHERE ((RENT_END_DATE -
RENT_START_DATE) > 0 AND (RENT_END_DATE - RENT_START_DATE)  < 180);
UPDATE RENT_DIM_V2 SET RENTPERIODID = 'Medium' WHERE ((RENT_END_DATE -
RENT_START_DATE) >= 180 AND (RENT_END_DATE - RENT_START_DATE)  < 365);
UPDATE RENT_DIM_V2 SET RENTPERIODID = 'Short' WHERE ((RENT_END_DATE -
RENT_START_DATE) >= 365);
--Creating Location Dimension
CREATE TABLE LOCATION_Dim_V2 AS
SELECT * FROM ADDRESS;
--Creating Table PostCode Dimension
CREATE TABLE POSTCODE_Dim_V2 AS
SELECT * FROM POSTCODE;
--Creating Table State Dimension
CREATE TABLE STATE_Dim_V2 AS
SELECT * FROM State;


--Creating Table PropertyCategory Dimension
--This table defines property category based on number of features it has
```

```
CREATE TABLE Category_Dim_v2
( CategoryID VARCHAR(15) ,
  Cat_Feature_Range VARCHAR(15));
  --Populating PropertyCategory Dimension
INSERT INTO Category_Dim_V2 VALUES('Very Basic','0-10');
INSERT INTO Category_Dim_V2 VALUES('Standard','10-20');
INSERT INTO Category_Dim_V2 VALUES('Luxurious','More Than 20');
SELECT * FROM CATEGORY_Dim_V2;
-- Creating Table Scale Dimension
--This table specifies the Property Scale based on Property Size
CREATE TABLE Scale_Dim_V2
( ScaleID VARCHAR(15) ,
  ScaleRange VARCHAR(15));
--Populating ScaleDim
INSERT INTO Scale_Dim_V2 VALUES('Extra Small','0-1');
INSERT INTO Scale_Dim_V2 VALUES('Medium','3-6');
INSERT INTO Scale_Dim_V2 VALUES('Large','6-10');
INSERT INTO Scale_Dim_V2 VALUES('Extra Large','More Than 10');
--Creating Type Dimension
--This table defines Property type like Apartment / House
CREATE TABLE Type_Dim_V2 AS
SELECT DISTINCT PROPERTY_TYPE FROM PROPERTY;
--Creating TimeDim
--this table stores advertisement month and year for properties
CREATE TABLE TIME_DIM_V2 AS
SELECT DISTINCT TO_CHAR(property_date_added,'MMYYYY') AS TimeID ,
TO_CHAR(property_date_added,'Mon') AS Month ,TO_CHAR(property_date_added,'YYYY') AS Year
FROM PROPERTY;


--Creating Table SEASON  Dimension
--This table contains Season  data
CREATE TABLE SEASON_DIM_V2
(SeasonID VARCHAR(20),
```

```
SeasonDesc VARCHAR(20));

--Populating Season Dimension

INSERT INTO SEASON_DIM_V2 VALUES('Spring','Sep,Oct,Nov');

INSERT INTO SEASON_DIM_V2 VALUES('Summer','Dec,Jan,Feb');

INSERT INTO SEASON_DIM_V2 VALUES('Autumn','Mar,Apr,May');

INSERT INTO SEASON_DIM_V2 VALUES('Winter','Jun,Jul,Aug');

--Creating ClientBudget Dimension

CREATE TABLE BUDGET_DIM_V2

( BUDGETID VARCHAR(20),

BUDGETRANGE VARCHAR(20));

--Populating Budget_dimesion

INSERT INTO BUDGET_DIM_V2 VALUES('Low','0-1000');

INSERT INTO BUDGET_DIM_V2 VALUES('Medium','1001-100000');

INSERT INTO BUDGET_DIM_V2 VALUES('High','100001- 10000000');

--Creating Agent_Office DImension

CREATE TABLE AGENT_OFFICE_DIM_V2 AS

SELECT distinct OFFICE_ID,count(person_id) no_of_agents  FROM AGENT_OFFICE group by
office_id;

alter table agent_office_dim_v2 add(Office_Size Varchar(20));

UPDATE agent_office_dim_v2

set Office_Size =

(case when no_of_agents  >= 0 and  no_of_agents <= 3 then 'Small'

  when  no_of_agents >= 4 or no_of_agents <= 12 then 'Medium'

  when no_of_agents > 12 then 'Big'  end);

-- Creating Office Dimension

CREATE TABLE OFFICE_DIM_V2 AS

SELECT * FROM OFFICE;

--Creating Agent Dimension

CREATE TABLE AGENT_DIM_V2 AS

SELECT AGENT.PERSON_ID AS AGENT_ID,SALARY,GENDER FROM AGENT,PERSON WHERE
AGENT.PERSON_ID = PERSON.PERSON_ID;

--Creating Ssle Dimension

CREATE TABLE SALE_DIM_V2 AS SELECT
```

SALE_ID,SALE_DATE,PROPERTY_ID,PRICE FROM SALE WHERE CLIENT_PERSON_ID IS NOT NULL;

--Creating Client Dimension

CREATE TABLE CLIENT_DIM_V2 AS SELECT CLIENT.PERSON_ID AS CLIENT_ID,GENDER FROM CLIENT left join PERSON on CLIENT.PERSON_ID = PERSON.PERSON_ID;

--Creating Client Wish Table

CREATE TABLE CLIENT_WISH_DIM_V2 AS SELECT * FROM CLIENT_WISH;

--Creating Visit_DIM_V2

CREATE TABLE VISIT_DIM_V2 AS SELECT

VISIT_DATE ,TO_CHAR(Visit_date,'Day') As VisitDay,TO_CHAR(Visit_date,'Mon') As Month,TO_CHAR(Visit_date,'YYYY') As Year from VISIT;

Now All Fact tables are created ,the commands are below:

--Creating FACT TABLES NOW

-- Creating Advert_Fact_V2

-- This table contains aggeregate values for property advertisements

CREATE TABLE ADVERT_FACT_V2 AS

SELECT TO_CHAR(property_date_added,'MMYYY') As TimeID,property_id , Count(property_id) AS Total_No_Properties

from property group by TO_CHAR(property_date_added,'MMYYY') ,property_id;

--Creating Temp Fact For Client

CREATE TABLE CLIENTTEMPFACT AS SELECT PERSON_ID AS CLIENT_ID ,MAX_BUDGET FROM CLIENT;

ALTER TABLE CLIENTTEMPFACT ADD(BUDGETID VARCHAR(20));

UPDATE CLIENTTEMPFACT

set BudgetID =

(case when Max_budget >= 0 and Max_Budget <= 1000 then 'Low'

  when max_budget >= 1001 and max_budget <= 100000 then 'Medium'

  when max_budget >= 100001 and max_budget <= 10000000 then 'High' end);

--Creating Client fact

CREATE TABLE CLIENT_FACT_V2 AS

SELECT BUDGETID,CLIENT_ID ,COUNT(CLIENT_ID) AS TOTAL_NO_CLIENTS FROM CLIENTTEMPFACT GROUP BY BUDGETID,CLIENT_ID;

SELECT BUDGETID,COUNT(TOTAL_NO_CLIENTS) AS TC FROM CLIENT_FACT_V2 GROUP BY BUDGETID;

--Creating Visit Temp fact

```
CREATE TABLE VISITTEMPFACT AS

SELECT VISIT_DATE,PROPERTY_ID,CLIENT_PERSON_ID AS CLIENT_ID FROM VISIT;

ALTER TABLE VISITTEMPFACT ADD(SEASONID VARCHAR(20));

UPDATE VISITTEMPFACT

set SeasonID =

(case when to_char(visit_date,'mm') >= 9 and to_char(visit_date,'mm') <= 11 then 'Spring'

  when to_char(visit_date,'mm') >= 12 or to_char(visit_date,'mm') <= 02 then 'Summer'

  when to_char(visit_date,'mm') >= 3 and to_char(visit_date,'mm') <= 5 then 'Autumn'

  when to_char(visit_date,'mm') >= 6 and to_char(visit_date,'mm') <= 8 then 'Winter' end);

--creating  visitfact table

CREATE TABLE VISIT_FACT_V2 AS

SELECT SEASONID,VISIT_DATE,PROPERTY_ID,CLIENT_ID,COUNT(*) AS TOTAL_VISITS FROM
VISITTEMPFACT  GROUP BY SEASONID,VISIT_DATE,PROPERTY_ID,CLIENT_ID;

SELECT * FROM VISIT_FACT_V2;

--Creating Agent Fact Table

CREATE TABLE AGENT_FACT_V2 AS

SELECT OFFICE_ID,agent.person_id as agent_id,count(agent.person_id) as total_no_agent,sum(salary)
as total_agent_earning from

agent full outer join agent_office  on agent.person_id = agent_office.person_id group by
office_id,agent.person_id;

--Creating Temp Rental Fact Table

CREATE TABLE TEMPRENTFACT AS

SELECT property.property_id,rent_id,property.address_id,property_type,rent.agent_person_id as agent_id
, rent.client_person_id  as client_id ,

sum(round((rent_end_date - rent_start_date) * (price/7),2)) as total_rental_fee,count(rent_id) as
total_no_of_rent

from property ,rent where property.property_id = rent.rent_id

group by
property.property_id,rent_id,property.address_id,property_type,rent.agent_person_id,rent.client_person_id
;

--Creating rent fact version 2 (low aggregation)

CREATE TABLE RENT_FACT_V2 AS SELECT * FROM TEMPRENTFACT;



 --Creating Temp Sales fact Table

CREATE TABLE TEMPSALESFACT AS
```

SELECT property.property_id,sale_id,property.address_id,property_type,sale.agent_person_id as agent_id , sale.client_person_id as client_id,sum(price) as tota_amount_of_sales

from property,sale where property.property_id = sale.property_id

and sale.client_person_id is not null

group by property.property_id,sale_id,property.address_id,property_type,sale.agent_person_id , sale.client_person_id;

CREATE TABLE SALES_FACT_V2 AS SELECT * FROM TEMPSALESFACT;

c) **Screenshots for Version – 1 Table :**

Feature_Dim_v1

Query Result

SQL | Fetched 50 rows in 0.004 seconds

	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

Category_Dim_V1

Script Output | Query Result

SQL | All Rows Fetched: 3 in 0.007 seconds

	CATEGORYID	CAT_FEATURE_RANGE
1	Very Basic	0-10
2	Standard	10-20
3	Luxurious	More Than 20

Scale_Dim_V1

Script Output | Query Result

SQL | All Rows Fetched: 4 in 0.007 seconds

	SCALEID	SCALERANGE
1	Extra Small	0-1
2	Medium	3-6
3	Large	6-10
4	Extra Large	More Than 10

Rent_dim_V1

Script Output × | Query Result ×

SQL | Fetched 50 rows in 0.005 seconds

	RENT_ID	PROPERTY_ID	RENT_START_DATE	RENT_END_DATE	RENTPERIODID
1	331	6199	12/JAN/20	28/JUN/20	Short
2	332	6063	02/MAY/20	18/OCT/20	Short
3	333	6074	01/MAY/20	17/OCT/20	Short
4	334	6142	12/FEB/20	29/JUL/20	Short
5	335	6146	20/APR/20	06/OCT/20	Short
6	336	5373	27/APR/20	13/OCT/20	Short
7	337	5801	25/FEB/20	11/AUG/20	Short
8	338	5513	01/JAN/20	17/JUN/20	Short
9	339	5709	29/MAR/20	13/SEP/20	Short
10	340	5548	23/APR/20	09/OCT/20	Short
11	341	5901	01/MAY/20	17/OCT/20	Short
12	342	5724	01/MAY/20	17/OCT/20	Short
13	343	6035	30/APR/20	16/OCT/20	Short
14	344	5557	23/APR/20	09/OCT/20	Short
15	345	5621	21/APR/20	07/OCT/20	Short
16	346	5598	23/APR/20	09/OCT/20	Short
17	347	5386	18/MAR/20	02/SEP/20	Short

LOCATION_Dim_V1

SQL | Fetched 50 rows in 0.033 seconds

	ADDRESS_ID	STREET	SUBURB	POSTCODE
1	527	10 Flynn Place	Aspley	4034
2	540	7/702 Kingston Road	Loganlea	4131
3	557	23 Eleventh Avenue	St Lucia	4067
4	559	4 Cliff Salisbury Court	Samford Village	4520
5	562	12 Amcord Place	Rothwell	4022
6	563	85 Gladstone Street	Coorparoo	4151
7	564	34 Seventh Avenue	St Lucia	4067
8	568	9/14 Hastings Street	Teneriffe	4005
9	571	105/28-46 Moriarty Street	Bald Hills	4036
10	577	24/21-29 Second Avenue	Marsden	4132
11	580	36/18 Defiance Road	Logan Central	4114
12	590	7 Cashel Street	Tingalpa	4173
13	593	11 Laconia Street	Logan Central	4114
14	596	91 Woodlands Blvd	Waterford	4133
15	620	46 Viney Street	Chermside West	4032
16	625	7/9 Norwood Street (east Wing)	Toowong	4066
17	1420	1a, 1b, 1c Morton Street	Weetangera	2614

postcode_Dim_V1

Script Output × | Query Result ×

SQL | Fetched 50 rows in 0.00

	POSTCODE	STATE_CODE
1	800	NT
2	2025	NSW
3	2027	NSW
4	2042	NSW
5	2069	NSW
6	2071	NSW
7	2075	NSW
8	2081	NSW
9	2090	NSW
10	2096	NSW
11	2111	NSW
12	2138	NSW
13	2144	NSW
14	2145	NSW
15	2150	NSW
16	2161	NSW
17	2163	NSW
18	2165	NSW

STATE_Dim_V1

Script Output × | Query Result ×

SQL | All Rows Fetched: 8 in 0.004 seconds

	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

Type_dim_v1

Script Output × | Query Result ×

SQL | All Rows Fetched: 14 in 0.004 seconds

	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

Season_Dim_V1

Script Output | Query Result

SQL | All Rows Fetched: 4 in 0.008 seconds

	SEASONID	SEASONDESC
1	Spring	Sep,Oct,Nov
2	Summer	Dec,Jan,Feb
3	Autumn	Mar,Apr,May
4	Winter	Jun,Jul,Aug

Budget_Dim_V1

Script Output | Query Result

SQL | All Rows Fetched: 3 in 0.007 seconds

	BUDGETID	BUDGETRANGE
1	Low	0-1000
2	Medium	1001-100000
3	High	100001- 10000000

Agent_Office_Dim_V1

Script Output | Query Result

SQL | Fetched 50 rows in 0.007 seconds

	OFFICE_ID	NO_OF_AGENTS	OFFICE_SIZE
1	837	1	Small
2	1132	1	Small
3	667	5	Medium
4	1133	4	Medium
5	902	6	Medium
6	817	5	Medium
7	554	1	Small
8	781	1	Small
9	237	2	Small
10	893	1	Small
11	281	4	Medium
12	1039	1	Small
13	173	9	Medium
14	592	1	Small
15	57	1	Small
16	924	3	Small
17	400	3	Small

Office_Dim_V1

Script Output | Query Result

SQL | Fetched 50 rows in 0.006 seconds

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

Agent_dim_v1

Script Output | Query Result

SQL | Fetched 50 rows in 0.006 seconds

	AGENT_ID	SALARY	GENDER
1	982	190000	Male
2	989	190000	Female
3	1004	175000	Female
4	1014	200000	Female
5	1015	200000	Male
6	1018	195000	Male
7	1024	200000	Male
8	1030	175000	Female
9	1056	190000	Female
10	1058	180000	Male
11	1081	210000	Male
12	1085	180000	Male
13	1087	200000	Female
14	9	190000	Female
15	21	200000	Female
16	34	210000	Female
17	62	190000	Female

Sale_dim_v1

Script Output × Query Result ×

SQL | Fetched 50 rows in 0.008 seconds

	SALE_ID	SALE_DATE	PROPERTY_ID	PRICE
1	434	22/MAR/20	1964	1395000
2	435	16/JAN/20	1896	275000
3	436	20/MAR/20	1932	3490000
4	437	14/JAN/20	1998	799000
5	438	22/FEB/20	1943	2000000
6	439	29/JAN/20	5	1825000
7	440	14/MAR/20	67	380000
8	441	05/APR/20	72	1695000
9	442	19/JAN/20	121	495000
10	443	09/MAR/20	229	565000
11	444	09/MAR/20	159	545000
12	445	19/FEB/20	164	280000
13	446	01/MAR/20	202	520000
14	447	25/FEB/20	305	440000
15	448	02/MAR/20	217	470000
16	449	24/JAN/20	173	565000
17	450	25/JAN/20	241	390000

client_dim_v1

Script Output × Query Result ×

SQL | Fetched 50 rows in 0.007 seconds

	CLIENT_ID	GENDER	CLIENTLISTAGG
1	5300	Female	20
2	5500	Female	6_7_9_23_42_48_64_73_127
3	5052	Male	1_2_10_59_61_117
4	5056	Female	1_2_5_6_7_19_20_38_40_235_239
5	5065	Male	1_20
6	5067	Female	1_2_5_6_7_39_86_493
7	5073	Male	2_9_14_20_26
8	5074	Male	2_9_14_20_26
9	5079	Female	20
10	5083	Male	20
11	5108	Male	1_20
12	5109	Female	16_20
13	5110	Male	1_2_5_6_7_39_86_493
14	5121	Female	2_9_14_20_26
15	5124	Male	1_14_20_22_26_64_86_87_120_213
16	5128	Male	1_2_9_14_18_20_23_25_26_29_40_41_45_71_73_74_75_94_127_239
17	5133	Male	16_20

Creating Fact tables

Client_fact_v1

Script Output × | Query Result ×

SQL | Fetched 50 rows in 0.006 seconds

	BUDGETID	CLIENT_ID	TOTAL_NO_CLIENTS
1	High	3103	1
2	High	3107	1
3	High	3330	1
4	High	2849	1
5	High	3142	1
6	High	3301	1
7	High	3368	1
8	Low	3965	1
9	Medium	3974	1
10	Low	3981	1
11	Low	3994	1
12	Low	3998	1
13	Low	4007	1
14	Low	4011	1
15	Low	4022	1
16	Low	4032	1
17	Low	4057	1

Visit_fact_v1

Script Output × | Query Result ×

SQL | Fetched 50 rows in 0.008 seconds

	SEASONID	PROPERTY_ID	CLIENT_ID	TOTAL_VISITS
1	Autumn	5937	5626	1
2	Autumn	5439	5471	1
3	Autumn	5632	5450	1
4	Autumn	5568	5423	1
5	Autumn	5389	5389	1
6	Autumn	5389	5489	1
7	Autumn	5391	5389	1
8	Autumn	5772	5389	1
9	Autumn	5275	5325	1
10	Autumn	5454	5527	1
11	Autumn	5615	5456	1
12	Autumn	6112	5414	1
13	Autumn	6112	5574	1
14	Autumn	5674	5373	1
15	Autumn	5857	5556	1
16	Autumn	5857	5592	1
17	Autumn	5511	5519	1

Agent_fact_v1

Script Output | Query Result

SQL | Fetched 50 rows in 0.005 seconds

	OFFICE_ID	AGENT_ID	TOTAL_NO_AGENT	TOTAL_AGENT_EARNING
1	1069	365	1	175000
2	235	964	1	200000
3	1070	1898	1	180000
4	332	380	1	175000
5	622	989	1	190000
6	667	2239	1	200000
7	833	406	1	175000
8	103	1316	1	190000
9	764	1025	1	190000
10	557	116	1	175000
11	1157	1961	1	190000
12	558	127	1	200000
13	556	1637	1	200000
14	62	1036	1	210000
15	495	2296	1	180000
16	324	471	1	175000
17	13	1669	1	200000

Rent_fact_v1

Query Result

SQL | Fetched 50 rows in 0.073 seconds

	PROPERTY_ID	RENT_ID	ADDRESS_ID	PROPERTY_TYPE	AGENT_ID	CLIENT_ID	TOTAL_RENTAL_FEE	TOTAL_NO_OF_RENT
1	28	28	28	House	196	3410	530	1
2	60	60	60	House	222	3442	525	1
3	19	19	19	Apartment / Unit / Flat	190	3401	1350	1
4	89	89	89	House	244	3471	340	1
5	264	264	264	House	544	3646	400	1
6	505	505	505	House	860	3887	400	1
7	354	354	354	Villa	576	3736	535	1
8	355	355	355	Apartment / Unit / Flat	578	3737	460	1
9	598	598	598	House	908	3980	410	1
10	600	600	600	House	909	3982	520	1
11	735	735	735	House	1154	4117	630	1
12	854	854	854	House	1185	4236	510	1
13	879	879	879	House	1381	4261	725	1
14	797	797	797	Apartment / Unit / Flat	1170	4179	640	1
15	1052	1052	1052	House	1484	4434	510	1
16	970	970	970	Apartment / Unit / Flat	1444	4352	310	1
17	1100	1100	1100	House	1716	4482	900	1
18	1133	1133	1133	House	1744	4515	650	1

Sales_Fact_v1

Query Result

SQL | Fetched 50 rows in 0.016 seconds

	PROPERTY_ID	SALE_ID	ADDRESS_ID	PROPERTY_TYPE	AGENT_ID	CLIENT_ID	TOTAL_AMOUNT_OF_SALES	TOTAL_NO_OF_SALE
1	320	453	320	Townhouse	1209	2919	349000	1
2	191	563	191	House	1516	3029	675000	1
3	564	348	564	House	941	2814	1800000	1
4	609	692	609	House	1842	3158	369000	1
5	765	694	765	Apartment / Unit / Flat	1844	3160	559000	1
6	1235	501	1235	House	1279	2967	799000	1
7	1361	156	1361	House	402	2622	780000	1
8	2722	746	2722	House	1935	3212	880000	1
9	2923	511	2923	House	1292	2977	625000	1
10	2379	62	2379	Townhouse	117	2528	640000	1
11	2025	764	2025	House	1972	3230	680000	1
12	1779	672	1779	House	1687	3138	465000	1
13	2259	642	2259	House	1647	3108	870000	1
14	2304	887	2304	House	2304	3353	920000	1
15	1834	779	1834	House	2006	3245	290000	1
16	223	227	223	House	624	2693	379000	1
17	331	816	331	House	2165	3282	775000	1
18	259	228	259	House	624	2694	279000	1

Version 2

Feature_Dim_v2

Script Output | Query Result

SQL | Fetched 50 rows in 0.006 seconds

	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

Property_dim_v2

Script Output | Query Result

SQL | Fetched 50 rows in 0.004 seconds

	PROPERTY_ID	PROPERTY_DATE_ADDED	NO_OF_FEATURES	PROPERTY_NO_OF_BEDROOMS	CATEGORYID	SCALEID
1	28	08/APR/20	25	4	Luxurious	Medium
2	31	19/DEC/19	10	4	Standard	Medium
3	90	02/MAR/20	8	2	Very Basic	Small
4	92	04/DEC/19	4	3	Very Basic	Medium
5	153	25/NOV/19	1	3	Very Basic	Medium
6	154	25/MAR/20	12	3	Standard	Medium
7	168	13/MAR/20	14	4	Standard	Medium
8	177	15/APR/20	11	4	Standard	Medium
9	194	14/APR/20	12	3	Standard	Medium
10	204	21/APR/20	7	3	Very Basic	Medium
11	209	26/APR/20	9	4	Very Basic	Medium
12	219	20/APR/20	5	2	Very Basic	Small
13	223	24/FEB/20	14	4	Standard	Medium
14	260	02/JAN/20	7	4	Very Basic	Medium
15	261	06/MAR/20	18	3	Standard	Medium
16	290	09/APR/20	9	3	Very Basic	Medium
17	310	18/APR/20	10	3	Standard	Medium
18	330	06/DEC/19	9	1	Very Basic	Extra Small

Rent_Dim_v2

Script Output | Query Result

SQL | Fetched 50 rows in 0.007 seconds

	RENT_ID	PROPERTY_ID	RENT_START_DATE	RENT_END_DATE	RENTPERIODID
1	331	6199	12/JAN/20	28/JUN/20	Short
2	332	6063	02/MAY/20	18/OCT/20	Short
3	333	6074	01/MAY/20	17/OCT/20	Short
4	334	6142	12/FEB/20	29/JUL/20	Short
5	335	6146	20/APR/20	06/OCT/20	Short
6	336	5373	27/APR/20	13/OCT/20	Short
7	337	5801	25/FEB/20	11/AUG/20	Short
8	338	5513	01/JAN/20	17/JUN/20	Short
9	339	5709	29/MAR/20	13/SEP/20	Short
10	340	5548	23/APR/20	09/OCT/20	Short
11	341	5901	01/MAY/20	17/OCT/20	Short
12	342	5724	01/MAY/20	17/OCT/20	Short
13	343	6035	30/APR/20	16/OCT/20	Short
14	344	5557	23/APR/20	09/OCT/20	Short
15	345	5621	21/APR/20	07/OCT/20	Short
16	346	5598	23/APR/20	09/OCT/20	Short
17	347	5386	18/MAR/20	02/SEP/20	Short

Location_Dim_v2

Script Output × Query Result ×

SQL | Fetched 50 rows in 0.007 seconds

	ADDRESS_ID	STREET	SUBURB	POSTCODE
1	527	10 Flynn Place	Aspley	4034
2	540	7/702 Kingston Road	Loganlea	4131
3	557	23 Eleventh Avenue	St Lucia	4067
4	559	4 Cliff Salisbury Court	Samford Village	4520
5	562	12 Amcord Place	Rothwell	4022
6	563	85 Gladstone Street	Coorparoo	4151
7	564	34 Seventh Avenue	St Lucia	4067
8	568	9/14 Hastings Street	Teneriffe	4005
9	571	105/28-46 Moriarty Street	Bald Hills	4036
10	577	24/21-29 Second Avenue	Marsden	4132
11	580	36/18 Defiance Road	Logan Central	4114
12	590	7 Cashel Street	Tingalpa	4173
13	593	11 Laconia Street	Logan Central	4114
14	596	91 Woodlands Blvd	Waterford	4133
15	620	46 Viney Street	Chermside West	4032
16	625	7/9 Norwood Street (east Wing)	Toowong	4066
17	1420	1a, 1b, 1c Morton Street	Weetangera	2614

Postcode_dim_v2

Script Output × Query Result ×

SQL | Fetched 50 rows

	POSTCODE	STATE_CODE
1	800	NT
2	2025	NSW
3	2027	NSW
4	2042	NSW
5	2069	NSW
6	2071	NSW
7	2075	NSW
8	2081	NSW
9	2090	NSW
10	2096	NSW
11	2111	NSW
12	2138	NSW
13	2144	NSW
14	2145	NSW
15	2150	NSW
16	2161	NSW
17	2163	NSW

State_dim_v2

Script Output × Query Result ×

SQL | All Rows Fetched: 8 in 0.005 seconds

	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

Category_dim_v2

Script Output × Query Result ×

SQL | All Rows Fetched: 3 in 0.00

	CATEGORYID	CAT_FEATURE_RANGE
1	Very Basic	0-10
2	Standard	10-20
3	Luxurious	More Than 20

Scale_dim_v2

Script Output × Query Result ×

SQL | All Rows Fetched: 4 in 0.007 seconds

	SCALEID	SCALERANGE
1	Extra Small	0-1
2	Medium	3-6
3	Large	6-10
4	Extra Large	More Than 10

Type_Dim_v2

Script Output × Query Result ×

SQL | All Rows Fetched: 14 in 0

	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

Season_dim_v2

Script Output | Query Result

SQL | All Rows Fetche

	SEASONID	SEASONDESC
1	Spring	Sep,Oct,Nov
2	Summer	Dec,Jan,Feb
3	Autumn	Mar,Apr,May
4	Winter	Jun,Jul,Aug

Budget_Dim_v2

Script Output | Query Result

SQL | All Rows Fetched: 3 ir

	BUDGETID	BUDGETRANGE
1	Low	0-1000
2	Medium	1001-100000
3	High	100001- 10000000

Agent_office_dim_v2

Script Output | Query Result

SQL | Fetched 50 rows in 0.005 seconds

	OFFICE_ID	NO_OF_AGENTS	OFFICE_SIZE
1	837	1	Small
2	1132	1	Small
3	667	5	Medium
4	1133	4	Medium
5	902	6	Medium
6	817	5	Medium
7	554	1	Small
8	781	1	Small
9	237	2	Small
10	893	1	Small
11	281	4	Medium
12	1039	1	Small
13	173	9	Medium
14	592	1	Small
15	57	1	Small
16	924	3	Small
17	400	3	Small

Office_dim_v2

Script Output | Query Result

SQL | Fetched 50 rows in 0.007 seconds

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

Agent_dim_v2

Script Output | Query Result

SQL | Fetched 50 rows in 0.008 seconds

	AGENT_ID	SALARY	GENDER
1	982	190000	Male
2	989	190000	Female
3	1004	175000	Female
4	1014	200000	Female
5	1015	200000	Male
6	1018	195000	Male
7	1024	200000	Male
8	1030	175000	Female
9	1056	190000	Female
10	1058	180000	Male
11	1081	210000	Male
12	1085	180000	Male
13	1087	200000	Female
14	9	190000	Female
15	21	200000	Female
16	34	210000	Female
17	62	190000	Female

Sale_Dim_v2

Query Result

SQL | Fetched 50 rows in 0.012 seconds

	SALE_ID	SALE_DATE	PROPERTY_ID	PRICE
1	434	22/MAR/20	1964	1395000
2	435	16/JAN/20	1896	275000
3	436	20/MAR/20	1932	3490000
4	437	14/JAN/20	1998	799000
5	438	22/FEB/20	1943	2000000
6	439	29/JAN/20	5	1825000
7	440	14/MAR/20	67	380000
8	441	05/APR/20	72	1695000
9	442	19/JAN/20	121	495000
10	443	09/MAR/20	229	565000
11	444	09/MAR/20	159	545000
12	445	19/FEB/20	164	280000
13	446	01/MAR/20	202	520000
14	447	25/FEB/20	305	440000
15	448	02/MAR/20	217	470000
16	449	24/JAN/20	173	565000
17	450	25/JAN/20	241	390000

Client_Dim_v2

Query Result

SQL | Fetched 5

	CLIENT_ID	GENDER
1	2703	Male
2	2717	Female
3	2719	Female
4	2728	Male
5	2739	Female
6	2740	Male
7	2754	Female
8	2768	Male
9	2769	Male
10	2777	Female
11	2784	Female
12	2785	Male
13	2786	Female
14	2789	Female
15	2804	Male
16	2805	Male
17	2813	Male

Client_Wish_dim_v2

Query Result

SQL | Fetched 50 rows in 0

	FEATURE_CODE	PERSON_ID
1	20	5202
2	20	5205
3	20	5208
4	20	5211
5	20	5216
6	20	5225
7	20	5227
8	20	5231
9	20	5234
10	20	5236
11	20	5244
12	20	5248
13	20	5256
14	20	5257
15	20	5264
16	20	5266
17	20	5268
18	20	5273

Visit_Dim_v2

Query Result

SQL | Fetched 50 rows in 0.019 seconds

	VISIT_DATE	VISITDAY	MONTH	YEAR
1	13/APR/20	Monday	Apr	2020
2	13/APR/20	Monday	Apr	2020
3	13/APR/20	Monday	Apr	2020
4	13/APR/20	Monday	Apr	2020
5	26/MAR/20	Thursday	Mar	2020
6	14/APR/20	Tuesday	Apr	2020
7	23/MAR/20	Monday	Mar	2020
8	29/MAR/20	Sunday	Mar	2020
9	29/MAR/20	Sunday	Mar	2020
10	29/MAR/20	Sunday	Mar	2020
11	29/MAR/20	Sunday	Mar	2020
12	29/MAR/20	Sunday	Mar	2020
13	31/MAR/20	Tuesday	Mar	2020
14	31/MAR/20	Tuesday	Mar	2020
15	31/MAR/20	Tuesday	Mar	2020
16	31/MAR/20	Tuesday	Mar	2020
17	31/MAR/20	Tuesday	Mar	2020
18	24/MAR/20	Tuesday	Mar	2020

Advert_fact_v2

Query Result

SQL | Fetched 50 rows in 0.01 seconds

	TIMEID	PROPERTY_ID	TOTAL_NO_PROPERTIES
1	04020	28	1
2	03020	139	1
3	04020	49	1
4	04020	8	1
5	02020	88	1
6	03020	269	1
7	04020	498	1
8	12019	408	1
9	03020	513	1
10	04020	435	1
11	03020	778	1
12	02020	764	1
13	04020	843	1
14	03020	854	1
15	04020	874	1
16	03020	818	1
17	01020	1049	1

client_fact_v2

Query Result

SQL | Fetched 50 rows in 0.013 seconds

	BUDGETID	CLIENT_ID	TOTAL_NO_CLIENTS
1	High	3103	1
2	High	3107	1
3	High	3330	1
4	High	2849	1
5	High	3142	1
6	High	3301	1
7	High	3368	1
8	Low	3965	1
9	Medium	3974	1
10	Low	3981	1
11	Low	3994	1
12	Low	3998	1
13	Low	4007	1
14	Low	4011	1
15	Low	4022	1
16	Low	4032	1
17	Low	4057	1
18	Low	4062	1

Visit_fact_v2

Query Result

SQL | Fetched 50 rows in 0.005 seconds

	SEASONID	VISIT_DATE	PROPERTY_ID	CLIENT_ID	TOTAL_VISITS
1	Autumn	20/MAR/20	5937	5626	1
2	Autumn	08/APR/20	5439	5471	1
3	Autumn	27/MAR/20	5632	5450	1
4	Autumn	02/APR/20	5568	5423	1
5	Autumn	10/APR/20	5389	5389	1
6	Autumn	10/APR/20	5389	5489	1
7	Autumn	21/MAR/20	5391	5389	1
8	Autumn	08/APR/20	5772	5389	1
9	Autumn	21/MAR/20	5275	5325	1
10	Autumn	30/MAR/20	5454	5527	1
11	Autumn	26/MAR/20	5615	5456	1
12	Autumn	12/APR/20	6112	5414	1
13	Autumn	12/APR/20	6112	5574	1
14	Autumn	30/MAR/20	5674	5373	1
15	Autumn	26/MAR/20	5857	5556	1
16	Autumn	26/MAR/20	5857	5592	1
17	Autumn	12/APR/20	5511	5519	1

Agent_fact_v2

Query Result

SQL | Fetched 50 rows in 0.007 seconds

	OFFICE_ID	AGENT_ID	TOTAL_NO_AGENT	TOTAL_AGENT_EARNING
1	1069	365	1	175000
2	235	964	1	200000
3	1070	1898	1	180000
4	332	380	1	175000
5	622	989	1	190000
6	667	2239	1	200000
7	833	406	1	175000
8	103	1316	1	190000
9	764	1025	1	190000
10	557	116	1	175000
11	1157	1961	1	190000
12	558	127	1	200000
13	556	1637	1	200000
14	62	1036	1	210000
15	495	2296	1	180000
16	324	471	1	175000
17	13	1669	1	200000

Rent_fact_v2

Query Result

SQL | Fetched 50 rows in 0.014 seconds

	PROPERTY_ID	RENT_ID	ADDRESS_ID	PROPERTY_TYPE	AGENT_ID	CLIENT_ID	TOTAL_RENTAL_FEE	TOTAL_NO_OF_RENT
1	28	28	28	House	196	3410	12795.71	1
2	60	60	60	House	222	3442	12600	1
3	19	19	19	Apartment / Unit / Flat	190	3401	32400	1
4	89	89	89	House	244	3471	8208.57	1
5	264	264	264	House	544	3646	9600	1
6	505	505	505	House	860	3887	9657.14	1
7	354	354	354	Villa	576	3736	12840	1
8	355	355	355	Apartment / Unit / Flat	578	3737	11040	1
9	598	598	598	House	908	3980	9840	1
10	600	600	600	House	909	3982	12480	1
11	735	735	735	House	1154	4117	15120	1
12	854	854	854	House	1185	4236	12240	1
13	879	879	879	House	1381	4261	17400	1
14	797	797	797	Apartment / Unit / Flat	1170	4179	15360	1
15	1052	1052	1052	House	1484	4434	12312.86	1
16	970	970	970	Apartment / Unit / Flat	1444	4352	7484.29	1
17	1100	1100	1100	House	1716	4482	21600	1

Sales_fact_v2

Script Output | Query Result

SQL | Fetched 50 rows in 0.012 seconds

	PROPERTY_ID	SALE_ID	ADDRESS_ID	PROPERTY_TYPE	AGENT_ID	CLIENT_ID	TOTA_AMOUNT_OF_SALES
1	320	453	320	Townhouse	1209	2919	349000
2	191	563	191	House	1516	3029	675000
3	564	348	564	House	941	2814	1800000
4	609	692	609	House	1842	3158	369000
5	765	694	765	Apartment / Unit / Flat	1844	3160	559000
6	1235	501	1235	House	1279	2967	799000
7	1361	156	1361	House	402	2622	780000
8	2722	746	2722	House	1935	3212	880000
9	2923	511	2923	House	1292	2977	625000
10	2379	62	2379	Townhouse	117	2528	640000
11	2025	764	2025	House	1972	3230	680000
12	1779	672	1779	House	1687	3138	465000
13	2259	642	2259	House	1647	3108	870000
14	2304	887	2304	House	2304	3353	920000
15	1834	779	1834	House	2006	3245	290000
16	223	227	223	House	624	2693	379000
17	331	816	331	House	2165	3282	775000

TASK C.3 Create the following reports using OLAP queries.

a) Simple Reports

i) Report -1 Top K

a) Question – Display the ranking of agents having top rental fees from houses property type.

b) Management should be interested because they might like to incentivise top 10 agents who help in renting houses in all locations.

c) SQL Commands are below:

--For version 1(High Aggregation)

Select * from (select r.property_type, r.agent_id, sum(TOTAL_RENTAL_FEE) as total_fees,

DENSE_RANK() Over (Order By Sum(TOTAL_RENTAL_FEE) desc) AS Rank_num

```
From RENT_fact_v1 r,type_dim_v1 t,agent_dim_v1 a
where r.property_type like 'House'
and r.property_type = t.property_type
and r.agent_id = a.agent_id
Group By r.property_type, r.agent_id) where Rank_num < 11;

--For version 2(Low Aggregation)
Select  * from ( select r.property_type, r.agent_id, sum(TOTAL_RENTAL_FEE) as total_fees,
DENSE_RANK() Over (Order By Sum(TOTAL_RENTAL_FEE) desc) AS Rank_num
From RENT_fact_v2 r,type_dim_v2 t,agent_dim_v2 a
where r.property_type like 'House'
and r.property_type = t.property_type
and r.agent_id = a.agent_id
Group By r.property_type, r.agent_id) where Rank_num < 11;
```

d) Outputs for both versions of star schema

For Version – 1

Query Result

SQL | All Rows Fetched: 10 in 0.01 seconds

	PROPERTY_TYPE	AGENT_ID	TOTAL_FEES	RANK_NUM
1	House	594	193222.14	1
2	House	2047	188830.71	2
3	House	500	187073.58	3
4	House	1114	156817.14	4
5	House	1774	155972.14	5
6	House	1806	155220	6
7	House	1783	130169.29	7
8	House	1154	117341.43	8
9	House	1812	116019.29	9
10	House	1772	103637.15	10

For version – 2

Query Result

All Rows Fetched: 10 in 0.01 seconds

	PROPERTY_TYPE	AGENT_ID	TOTAL_FEES	RANK_NUM
1	House	594	193222.14	1
2	House	2047	188830.71	2
3	House	500	187073.58	3
4	House	1114	156817.14	4
5	House	1774	155972.14	5
6	House	1806	155220	6
7	House	1783	130169.29	7
8	House	1154	117341.43	8
9	House	1812	116019.29	9
10	House	1772	103637.15	10

ii) Report – 2 Top – N%

a) Question - Show the 10 % agents by agent earning

b) Management should be interested in knowing the top 10 % agents by earning.

c) Commands for the queries are shown below:

```
--For Version 1

SELECT * from ( select agent_fact_v1.agent_id,sum(total_agent_earning) as agent_earning,

round(percent_rank() over(order by sum(total_agent_earning) desc),3) as percent_rank from agent_fact_v1
,agent_dim_v1

where  agent_fact_v1.agent_id = agent_dim_v1 .agent_id

group by agent_fact_v1.agent_id) where percent_rank < 0.1;

--For Version 2

SELECT * from ( select agent_fact_v2.agent_id,sum(total_agent_earning) as agent_earning,

round(percent_rank() over(order by sum(total_agent_earning) desc),3) as percent_rank from agent_fact_v2
,agent_dim_v2

where  agent_fact_v2.agent_id = agent_dim_v2 .agent_id

group by agent_fact_v2.agent_id) where percent_rank < 0.1;
```

d) Screenshots for the queries are shown below :

Query Result

SQL | Fetched 50 rows in 0.007 seconds

	AGENT_ID	AGENT_EARNING	PERCENT_RANK
1	1541	630000	0
2	241	630000	0
3	1721	600000	0.001
4	1383	585000	0.001
5	1850	570000	0.002
6	500	525000	0.002
7	781	420000	0.002
8	6	420000	0.002
9	1786	420000	0.002
10	1794	420000	0.002
11	1968	420000	0.002
12	1619	420000	0.002
13	364	420000	0.002
14	2426	420000	0.002
15	1057	420000	0.002
16	1248	420000	0.002
17	551	420000	0.002

iii) Report – 3

a) Show All

Question- Show the total number of female clients in each budget range .

b) Management might be interested in knowing the range of female clients as they are the ones that have a say in renting or buying properties in the family rather than men.

c) SQL Commands :

For Version – 1

--For Version 1

select c.budgetid,gender,count(total_no_clients) from client_fact_v1 c,budget_dim_v1,client_dim_v1 where c.budgetid = budget_dim_v1.budgetid

and c.client_id = client_dim_v1.client_id and client_dim_v1.gender like 'Female' group by

c.budgetid,gender;

For Version – 2

select c.budgetid,gender,count(total_no_clients) from client_fact_v2 c,budget_dim_v2,client_dim_v2 where c.budgetid = budget_dim_v2.budgetid

and c.client_id = client_dim_v2.client_id and client_dim_v2.gender like 'Female' group by

c.budgetid,gender;

d) Screenshots :

For Version -1

Query Result

All Rows Fetched: 3 in 0.01 seconds

	BUDGETID	GENDER	COUNT(TOTAL_NO_CLIENTS)
1	High	Female	652
2	Medium	Female	238
3	Low	Female	760

For Version – 2

Query Result

All Rows Fetched: 3 in 0.01 seconds

	BUDGETID	GENDER	COUNT(TOTAL_NO_CLIENTS)
1	High	Female	652
2	Medium	Female	238
3	Low	Female	760

b) Reports with proper sub-totals:

i) Report – 4

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

b) Management should be interested to know the rental fees each suburbs generate by each house type in a time period and how much all suburbs generate in a time period

c) Commands for the question:

--For Version 1

```
SELECT DISTINCT L.SUBURB,TO_CHAR(R.RENT_START_DATE,'MMYYYY') AS TIME_PERIOD
,F.PROPERTY_TYPE,SUM(TOTAL_RENTAL_FEE) AS TOTAL_RENTAL_FEE

FROM RENT_FACT_V1 F,ADDRESS L  ,RENT_DIM_V1 R,TYPE_DIM_V1 T

where F.ADDRESS_ID = L.ADDRESS_ID AND F.PROPERTY_TYPE = T.PROPERTY_TYPE

and f.rent_id = r.rent_id

GROUP BY CUBE(
L.SUBURB,TO_CHAR(R.RENT_START_DATE,'MMYYYY'),F.PROPERTY_TYPE)

ORDER BY L.SUBURB;
```

```
--For Version 2

SELECT DISTINCT L.SUBURB,TO_CHAR(R.RENT_START_DATE,'MMYYYY') AS TIME_PERIOD
,F.PROPERTY_TYPE,SUM(TOTAL_RENTAL_FEE) AS TOTAL_RENTAL_FEE

FROM RENT_FACT_V2 F,ADDRESS L  ,RENT_DIM_V2 R,TYPE_DIM_V2 T

where F.ADDRESS_ID = L.ADDRESS_ID AND F.PROPERTY_TYPE = T.PROPERTY_TYPE

and f.rent_id = r.rent_id

GROUP BY CUBE(
L.SUBURB,TO_CHAR(R.RENT_START_DATE,'MMYYYY'),F.PROPERTY_TYPE)

ORDER BY L.SUBURB;
```

d) Screenshots for the query output:

For Version – 1

Query Result

SQL | Fetched 50 rows in 0.083 seconds

	SUBURB	TIME_PERIOD	PROPERTY_TYPE	TOTAL_RENTAL_FEE
1	Aberfoyle Park	032020	House	580
2	Aberfoyle Park	032020	(null)	580
3	Aberfoyle Park	052020	House	975
4	Aberfoyle Park	052020	(null)	975
5	Aberfoyle Park	(null)	House	1555
6	Aberfoyle Park	(null)	(null)	1555
7	Acacia Gardens	042020	House	655
8	Acacia Gardens	042020	(null)	655
9	Acacia Gardens	(null)	House	655
10	Acacia Gardens	(null)	(null)	655
11	Acacia Ridge	042020	House	820
12	Acacia Ridge	042020	(null)	820
13	Acacia Ridge	(null)	House	820
14	Acacia Ridge	(null)	(null)	820
15	Acton	042020	Apartment / Unit / Flat	1500
16	Acton	042020	(null)	1500
17	Acton	(null)	Apartment / Unit / Flat	1500

For Version – 2

Query Result

SQL | Fetched 50 rows in 0.083 seconds

	SUBURB	TIME_PERIOD	PROPERTY_TYPE	TOTAL_RENTAL_FEE
1	Aberfoyle Park	032020	House	580
2	Aberfoyle Park	032020	(null)	580
3	Aberfoyle Park	052020	House	975
4	Aberfoyle Park	052020	(null)	975
5	Aberfoyle Park	(null)	House	1555
6	Aberfoyle Park	(null)	(null)	1555
7	Acacia Gardens	042020	House	655
8	Acacia Gardens	042020	(null)	655
9	Acacia Gardens	(null)	House	655
10	Acacia Gardens	(null)	(null)	655
11	Acacia Ridge	042020	House	820
12	Acacia Ridge	042020	(null)	820
13	Acacia Ridge	(null)	House	820
14	Acacia Ridge	(null)	(null)	820
15	Acton	042020	Apartment / Unit / Flat	1500
16	Acton	042020	(null)	1500
17	Acton	(null)	Apartment / Unit / Flat	1500

ii) Report – 5

a) What are the sub-total and total rental fees from each suburb, time period, and property type? (Partial Cube)

b) Management should be interested to know the rental fees each suburbs generate in a time period and how much all suburbs generate in a time period

c) Commands for the question:

--For Version 1

```
SELECT DISTINCT L.SUBURB,TO_CHAR(R.RENT_START_DATE,'MMYYYY') AS TIME_PERIOD
,F.PROPERTY_TYPE,SUM(TOTAL_RENTAL_FEE) AS TOTAL_RENTAL_FEE

FROM RENT_FACT_V1 F,ADDRESS L  ,RENT_DIM_V1 R,TYPE_DIM_V1 T

where F.ADDRESS_ID = L.ADDRESS_ID AND F.PROPERTY_TYPE = T.PROPERTY_TYPE

and f.rent_id = r.rent_id

GROUP BY CUBE(
L.SUBURB,TO_CHAR(R.RENT_START_DATE,'MMYYYY')),F.PROPERTY_TYPE

ORDER BY L.SUBURB;
```

--For Version 2

```
SELECT DISTINCT L.SUBURB,TO_CHAR(R.RENT_START_DATE,'MMYYYY') AS TIME_PERIOD
,F.PROPERTY_TYPE,SUM(TOTAL_RENTAL_FEE) AS TOTAL_RENTAL_FEE

FROM RENT_FACT_V2 F,ADDRESS L  ,RENT_DIM_V2 R,TYPE_DIM_V2 T
```

```
where F.ADDRESS_ID = L.ADDRESS_ID AND F.PROPERTY_TYPE = T.PROPERTY_TYPE
and f.rent_id = r.rent_id
GROUP BY CUBE(
L.SUBURB,TO_CHAR(R.RENT_START_DATE,'MMYYYY')),F.PROPERTY_TYPE
ORDER BY L.SUBURB;
```

d) Screenshots for Report – 5

For Version – 1

Query Result

SQL | Fetched 50 rows in 0.022 seconds

	SUBURB	TIME_PERIOD	PROPERTY_TYPE	TOTAL_RENTAL_FEE
1	Aberfoyle Park	(null)	House	1555
2	Aberfoyle Park	052020	House	975
3	Aberfoyle Park	032020	House	580
4	Acacia Gardens	(null)	House	655
5	Acacia Gardens	042020	House	655
6	Acacia Ridge	(null)	House	820
7	Acacia Ridge	042020	House	820
8	Acton	(null)	Apartment / Unit / Flat	1500
9	Acton	042020	Apartment / Unit / Flat	1500
10	Adelaide	022020	Apartment / Unit / Flat	540
11	Adelaide	(null)	Apartment / Unit / Flat	1590
12	Adelaide	042020	Apartment / Unit / Flat	455
13	Adelaide	042020	Townhouse	565
14	Adelaide	032020	Townhouse	400
15	Adelaide	032020	Apartment / Unit / Flat	595
16	Adelaide	(null)	Townhouse	1615
17	Adelaide	022020	Townhouse	650

For Version – 2

Query Result

SQL | Fetched 50 rows in 0.022 seconds

	SUBURB	TIME_PERIOD	PROPERTY_TYPE	TOTAL_RENTAL_FEE
1	Aberfoyle Park	(null)	House	1555
2	Aberfoyle Park	052020	House	975
3	Aberfoyle Park	032020	House	580
4	Acacia Gardens	(null)	House	655
5	Acacia Gardens	042020	House	655
6	Acacia Ridge	(null)	House	820
7	Acacia Ridge	042020	House	820
8	Acton	(null)	Apartment / Unit / Flat	1500
9	Acton	042020	Apartment / Unit / Flat	1500
10	Adelaide	022020	Apartment / Unit / Flat	540
11	Adelaide	(null)	Apartment / Unit / Flat	1590
12	Adelaide	042020	Apartment / Unit / Flat	455
13	Adelaide	042020	Townhouse	565
14	Adelaide	032020	Townhouse	400
15	Adelaide	032020	Apartment / Unit / Flat	595
16	Adelaide	(null)	Townhouse	1615
17	Adelaide	022020	Townhouse	650

iii) Report – 6

a) Question – What are the total number of male and female agents in each agent office ?

b) Management would like to know the number of male and female agents in each office to promote gender equality

c) Commands to get the result are :

--For Version 1

```
SELECT DECODE(GROUPING(f.office_id),1,'f.office_id',f.office_id)As
office_id,DECODE(GROUPING(n.office_name),1,'n.office_name',n.office_name)as
office_name,DECODE(GROUPING(a.GENDER),1,'All Genders',a.GENDER) AS
GENDER,COUNT(Total_No_Agent) AS No_Of_Agents from

agent_fact_v2 f ,agent_office_dim_v2 o,office_dim_v2 n,agent_dim_v2 a

where f.office_id = o.office_id and o.office_id = n.office_id and f.agent_id = a.agent_id

group by rollup (f.OFFICE_ID,n.OFFICE_NAME,a.GENDER);
```

--For Version 2

```
SELECT DECODE(GROUPING(f.office_id),1,'f.office_id',f.office_id)As
office_id,DECODE(GROUPING(n.office_name),1,'n.office_name',n.office_name)as
office_name,DECODE(GROUPING(a.GENDER),1,'All Genders',a.GENDER) AS
GENDER,COUNT(Total_No_Agent) AS No_Of_Agents from

agent_fact_v1 f ,agent_office_dim_v1 o,office_dim_v1 n,agent_dim_v1 a

where f.office_id = o.office_id and o.office_id = n.office_id and f.agent_id = a.agent_id

group by rollup (f.OFFICE_ID,n.OFFICE_NAME,a.GENDER);
```

d) Screenshots for the query output are below :

Version – 1

	OFFICE_ID	OFFICE_NAME	GENDER	NO_OF_AGENTS
1	1	ACTON Coogee	Male	4
2	1	ACTON Coogee	Female	2
3	1	ACTON Coogee	All Genders	6
4	1	n.office_name	All Genders	6
5	2	ACTON North	Female	1
6	2	ACTON North	All Genders	1
7	2	n.office_name	All Genders	1
8	3	ADDISONS PROPERTY MANAGEMENT	Female	1
9	3	ADDISONS PROPERTY MANAGEMENT	All Genders	1
10	3	n.office_name	All Genders	1
11	4	AIREY REAL ESTATE	Male	4
12	4	AIREY REAL ESTATE	Female	3
13	4	AIREY REAL ESTATE	All Genders	7
14	4	n.office_name	All Genders	7
15	5	AM REALTY	Female	1
16	5	AM REALTY	All Genders	1
17	5	n.office_name	All Genders	1
18	6	AO Properties	Male	1

Version – 2

	OFFICE_ID	OFFICE_NAME	GENDER	NO_OF_AGENTS
1	1	ACTON Coogee	Male	4
2	1	ACTON Coogee	Female	2
3	1	ACTON Coogee	All Genders	6
4	1	n.office_name	All Genders	6
5	2	ACTON North	Female	1
6	2	ACTON North	All Genders	1
7	2	n.office_name	All Genders	1
8	3	ADDISONS PROPERTY MANAGEMENT	Female	1
9	3	ADDISONS PROPERTY MANAGEMENT	All Genders	1
10	3	n.office_name	All Genders	1
11	4	AIREY REAL ESTATE	Male	4
12	4	AIREY REAL ESTATE	Female	3
13	4	AIREY REAL ESTATE	All Genders	7
14	4	n.office_name	All Genders	7
15	5	AM REALTY	Female	1
16	5	AM REALTY	All Genders	1
17	5	n.office_name	All Genders	1
18	6	AO Properties	Male	1

iv) Report – 7

a) Question - What are the total number of properties sold in each state, postcode ?

b) Management might be interested in knowing the number of properties sold in each state and postcode to analyse client behaviour.

c) Commands for the question are below:

--For Version 1

```
SELECT p.state_code,l.postcode,f.address_id,sum(total_no_of_sale) as total_properties_sold from
sales_fact_v1 f,location_dim_v1 l,postcode_dim_v1 p,state_dim_v1 s

where f.address_id = l.address_id and l.postcode = p.postcode and p.state_code = s.state_code

group by rollup (p.state_code,l.postcode),f.address_id ;
```

--For Version 2

```
SELECT p.state_code,l.postcode,f.address_id,sum(total_no_of_sale) as total_properties_sold from
sales_fact_v2 f,location_dim_v2 l,postcode_dim_v2 p,state_dim_v2 s

where f.address_id = l.address_id and l.postcode = p.postcode and p.state_code = s.state_code

group by rollup (p.state_code,l.postcode),f.address_id ;
```

d) Screenshots of query output are shown below:

For Version – 1

SQL | Fetched 50 rows in 0.029 seconds

	STATE_CODE	POSTCODE	ADDRESS_ID	TOTAL_PROPERTIES_SOLD
1	VIC	3215	1	1
2	VIC	(null)	1	1
3	(null)	(null)	1	1
4	VIC	3220	5	1
5	VIC	(null)	5	1
6	(null)	(null)	5	1
7	VIC	3220	11	1
8	VIC	(null)	11	1
9	(null)	(null)	11	1
10	VIC	3216	13	1
11	VIC	(null)	13	1
12	(null)	(null)	13	1
13	VIC	3223	18	1
14	VIC	(null)	18	1
15	(null)	(null)	18	1
16	VIC	3220	19	1
17	VIC	(null)	19	1
18	(null)	(null)	19	1

For Version – 2

Fetched 50 rows in 0.029 seconds

	STATE_CODE	POSTCODE	ADDRESS_ID	TOTAL_PROPERTIES_SOLD
1	VIC	3215	1	1
2	VIC	(null)	1	1
3	(null)	(null)	1	1
4	VIC	3220	5	1
5	VIC	(null)	5	1
6	(null)	(null)	5	1
7	VIC	3220	11	1
8	VIC	(null)	11	1
9	(null)	(null)	11	1
10	VIC	3216	13	1
11	VIC	(null)	13	1
12	(null)	(null)	13	1
13	VIC	3223	18	1
14	VIC	(null)	18	1
15	(null)	(null)	18	1
16	VIC	3220	19	1
17	VIC	(null)	19	1
18	(null)	(null)	19	1

c) Reports with moving and cumulative aggregates :

i) Report – 8

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

b) Management will be interested to know the trend in number of clients with high budget.

c) Commands are shown below:

--For Version 1

select year,sum(no_of_clients)as total_clients,sum(sum(no_of_clients))over (order by Year rows unbounded preceding) AS Cumulative_Clients from

(SELECT TO_CHAR(v.VISIT_DATE,'YYYY') As Year,sum(total_visits) as no_of_clients

from visit_fact_v1 v,client_fact_v1 a ,client_dim_v1 c,budget_dim_v1 b ,visit_dim_v1

where a.client_id = c.client_id and a.budgetid = b.budgetid and b.budgetid like 'High' and v.client_id = c.client_id and v.visit_date = visit_dim_v1.visit_date

group by TO_CHAR(v.VISIT_DATE,'YYYY'),b.budgetid

union

SELECT TO_CHAR(rent_dim_v1.rent_start_DATE,'YYYY') As Year,count(r.client_id) as no_of_clients

from rent_fact_v1 r,client_fact_v1 a ,client_dim_v1 c,budget_dim_v1 b ,rent_dim_v1

```
where a.client_id = c.client_id and a.budgetid = b.budgetid and b.budgetid like 'High' and r.client_id =
c.client_id and r.rent_id = rent_dim_v1.rent_id

group by TO_CHAR(rent_dim_v1.rent_start_DATE,'YYYY'),b.budgetid

union

SELECT TO_CHAR(sale_dim_v1.sale_DATE,'YYYY') As Year ,count(s.client_id) as no_of_clients

from sales_fact_v1 s,client_fact_v1 a ,client_dim_v1 c,budget_dim_v1 b ,sale_dim_v1

where a.client_id = c.client_id and a.budgetid = b.budgetid and b.budgetid like 'High' and s.client_id =
c.client_id and s.sale_id = sale_dim_v1.sale_id

group by TO_CHAR(sale_dim_v1.sale_DATE,'YYYY'),b.budgetid) group by year;


--For Version 2

select year,sum(no_of_clients)as total_clients,sum(sum(no_of_clients))over (order by Year rows
unbounded preceding) AS Cumulative_Clients  from

(SELECT TO_CHAR(v.VISIT_DATE,'YYYY') As Year,sum(total_visits) as no_of_clients

from visit_fact_v2 v,client_fact_v2 a ,client_dim_v2 c,budget_dim_v2 b ,visit_dim_v2

where a.client_id = c.client_id and a.budgetid = b.budgetid and b.budgetid like 'High' and v.client_id =
c.client_id and v.visit_date = visit_dim_v2.visit_date

group by TO_CHAR(v.VISIT_DATE,'YYYY'),b.budgetid

union

SELECT TO_CHAR(rent_dim_v2.rent_start_DATE,'YYYY') As Year,count(r.client_id) as no_of_clients

from rent_fact_v2 r,client_fact_v2 a ,client_dim_v2 c,budget_dim_v2 b ,rent_dim_v2

where a.client_id = c.client_id and a.budgetid = b.budgetid and b.budgetid like 'High' and r.client_id =
c.client_id and r.rent_id = rent_dim_v2.rent_id

group by TO_CHAR(rent_dim_v2.rent_start_DATE,'YYYY'),b.budgetid

union

SELECT TO_CHAR(sale_dim_v2.sale_DATE,'YYYY') As Year ,count(s.client_id) as no_of_clients

from sales_fact_v2 s,client_fact_v2 a ,client_dim_v2 c,budget_dim_v2 b ,sale_dim_v2

where a.client_id = c.client_id and a.budgetid = b.budgetid and b.budgetid like 'High' and s.client_id =
c.client_id and s.sale_id = sale_dim_v2.sale_id

group by TO_CHAR(sale_dim_v2.sale_DATE,'YYYY'),b.budgetid) group by year;
```

d) Screenshots for the query output are below :

For Version -1

Query Result

SQL | All Rows Fetched: 2 in 0.033 seconds

	YEAR	TOTAL_CLIENTS	CUMULATIVE_CLIENTS
1	2019	23	23
2	2020	2018	2041

For Version – 2

Query Result

SQL | All Rows Fetched: 2 in 0.033 seconds

	YEAR	TOTAL_CLIENTS	CUMULATIVE_CLIENTS
1	2019	23	23
2	2020	2018	2041

ii) Report – 9

a) What is the total number of sales for each suburb and cumulative sales for each suburb?

b) Management can analyse suburbs by the no of sales in each month and cumulative sales for each suburb to get suburbs where sales are high or sales are low.

c) Commands are shown below:

```
--For Version 1

SELECT l.suburb,to_char(sale_date,'Mon') as year ,sum(total_no_of_sale) as monthly_no_sales ,

sum(sum(total_no_of_sale)) over (partition by l.suburb order by l.suburb,to_char(sale_date,'Mon') rows
unbounded preceding) As Cumulative_Sales

from sales_fact_v1 s,sale_dim_v1 d,location_dim_v1 l where

s.address_id = l.address_id and s.sale_id = d.sale_id

group by l.suburb,to_char(sale_date,'Mon');


--For Version 2

SELECT l.suburb,to_char(sale_date,'Mon') as year ,sum(total_no_of_sale) as monthly_no_sales ,

sum(sum(total_no_of_sale)) over (partition by l.suburb order by l.suburb,to_char(sale_date,'Mon') rows
unbounded preceding) As Cumulative_Sales

from sales_fact_v2 s,sale_dim_v2 d,location_dim_v2 l where

s.address_id = l.address_id and s.sale_id = d.sale_id

group by l.suburb,to_char(sale_date,'Mon');
```

d) Screenshots for the query are shown below:

For Version – 1

Query Result

SQL | Fetched 50 rows in 0.018 seconds

	SUBURB	YEAR	MONTHLY_NO_SALES	CUMULATIVE_SALES
1	Abbotsford	Jan	1	1
2	Acacia Ridge	Jan	2	2
3	Acton	Apr	1	1
4	Adelaide	Mar	2	2
5	Aitkenvale	Jan	1	1
6	Albanvale	Mar	1	1
7	Albany Creek	Jan	1	1
8	Albert Park	Apr	2	2
9	Albion	Jan	1	1
10	Alderley	Feb	1	1
11	Alexandria	Mar	1	1
12	Allambie Heights	Feb	1	1
13	Alphington	Jan	1	1
14	Andrews Farm	Mar	1	1
15	Annandale	Feb	1	1
16	Annandale	Jan	1	2
17	Annerley	Jan	2	2
18	Anzac Hills	Feb	1	1

For Version – 2

Query Result

SQL | Fetched 50 rows in 0.018 seconds

	SUBURB	YEAR	MONTHLY_NO_SALES	CUMULATIVE_SALES
1	Abbotsford	Jan	1	1
2	Acacia Ridge	Jan	2	2
3	Acton	Apr	1	1
4	Adelaide	Mar	2	2
5	Aitkenvale	Jan	1	1
6	Albanvale	Mar	1	1
7	Albany Creek	Jan	1	1
8	Albert Park	Apr	2	2
9	Albion	Jan	1	1
10	Alderley	Feb	1	1
11	Alexandria	Mar	1	1
12	Allambie Heights	Feb	1	1
13	Alphington	Jan	1	1
14	Andrews Farm	Mar	1	1
15	Annandale	Feb	1	1
16	Annandale	Jan	1	2
17	Annerley	Jan	2	2
18	Anzac Hills	Feb	1	1

iii) Report – 10

a) Question - Show the number of properties added by each month and year and cumulative number of properties added ?

b) Management will be interested to know the number of properties added in each month and each year to get yearly analysis.

c) Commands for the queries are shown below :

```
--For Version 1

SELECT TO_CHAR(property_date_added,'MM') as Month,TO_CHAR(property_date_added,'YYYY') as
Year ,sum(total_no_properties) as total_properties_added,

sum(sum(total_no_properties)) over (order by
TO_CHAR(property_date_added,'YYYY'),TO_CHAR(property_date_added,'MM') rows unbounded
preceding) AS Cumulative_Prop_Added

from advert_fact_v1 ,property_dim_v1 where

advert_fact_v1.property_id = property_dim_v1.property_id

group by TO_CHAR(property_date_added,'YYYY'),TO_CHAR(property_date_added,'MM')

order by  TO_CHAR(property_date_added,'YYYY'),TO_CHAR(property_date_added,'MM');


--For Version 2

SELECT TO_CHAR(property_date_added,'MM') as Month,TO_CHAR(property_date_added,'YYYY') as
Year ,sum(total_no_properties) as total_properties_added,

sum(sum(total_no_properties)) over (order by
TO_CHAR(property_date_added,'YYYY'),TO_CHAR(property_date_added,'MM') rows unbounded
preceding) AS Cumulative_Prop_Added

from advert_fact_v2 ,property_dim_v2 where

advert_fact_v2.property_id = property_dim_v2.property_id

group by TO_CHAR(property_date_added,'YYYY'),TO_CHAR(property_date_added,'MM')

order by  TO_CHAR(property_date_added,'YYYY'),TO_CHAR(property_date_added,'MM');
```

d) Screenshots for the question is shown below:

for version – 1

Query Result

SQL | All Rows Fetched: 6 in 0.013 seconds

	MONTH	YEAR	TOTAL_PROPERTIES_ADDED	CUMULATIVE_PROP_ADDED
1	11	2019	304	304
2	12	2019	573	877
3	01	2020	566	1443
4	02	2020	545	1988
5	03	2020	2041	4029
6	04	2020	2179	6208

For version – 2

Query Result

SQL | All Rows Fetched: 6 in 0.013 seconds

	MONTH	YEAR	TOTAL_PROPERTIES_ADDED	CUMULATIVE_PROP_ADDED
1	11	2019	304	304
2	12	2019	573	877
3	01	2020	566	1443
4	02	2020	545	1988
5	03	2020	2041	4029
6	04	2020	2179	6208

d) Reports with Partitions:

i) Report - 11

a) Show ranking of each property type based on the yearly total number of sales and the ranking of each state based on the yearly total number of sales.

b) Management will be interested in knowing property sales ranked by property type and yearly property sales in each state to get geographical analysis for sales.

c) Commands for the query is shown below:

--Version 1

SELECT to_char(d.sale_date,'yyyy'), s.property_type,state_name as Year,SUM(total_amount_of_sales) as total_sales,

DENSE_RANK() over (PARTITION BY s.property_type ORDER BY SUM(total_amount_of_sales) desc) as rank_by_type,

dense_rank() over (PARTITION BY state_name ORDER BY SUM(total_amount_of_sales) desc) as rank_by_state

from sales_fact_v1 s,sale_dim_v1 d,type_dim_v1 t,location_dim_v1 l,postcode_dim_v1 p,state_dim_v1 s

where s.sale_id = d.sale_id and s.property_type = t.property_type and s.address_id = l.address_id

and l.postcode = p.postcode and p.state_code = s.state_code group by to_char(d.sale_date,'yyyy'),s.property_type, state_name

order by to_char(d.sale_date,'yyyy') ;

--Version 2

SELECT to_char(d.sale_date,'yyyy'), s.property_type,state_name as Year,SUM(total_amount_of_sales) as total_sales,

DENSE_RANK() over (PARTITION BY s.property_type ORDER BY SUM(total_amount_of_sales) desc) as rank_by_type,

dense_rank() over (PARTITION BY state_name ORDER BY SUM(total_amount_of_sales) desc) as rank_by_state

from sales_fact_v2 s,sale_dim_v2 d,type_dim_v2 t,location_dim_v2 l,postcode_dim_v2 p,state_dim_v2 s

where s.sale_id = d.sale_id and s.property_type = t.property_type and s.address_id = l.address_id

and l.postcode = p.postcode and p.state_code = s.state_code group by to_char(d.sale_date,'yyyy'),s.property_type, state_name

order by to_char(d.sale_date,'yyyy') ;

d) Screenshots for the query are shown below:

For version 1

Query Result

SQL | Fetched 50 rows in 0.023 seconds

	TO_CHAR(D.SALE_DATE,'YYYY')	PROPERTY_TYPE	YEAR	TOTAL_SALES	RANK_BY_TYPE	RANK_BY_STATE
1	2019	Studio	Victoria	80000	2	8
2	2019	House	New South Wales	850000	13	7
3	2019	Townhouse	Queensland	165000	9	11
4	2019	Townhouse	Victoria	759000	8	7
5	2019	House	South Australia	1190000	12	4
6	2019	Duplex	Queensland	1100000	3	9
7	2019	House	Victoria	2474000	10	4
8	2019	Apartment / Unit / Flat	New South Wales	439000	8	9
9	2019	House	Queensland	3066950	8	6
10	2019	House	Western Australia	3295000	7	3
11	2019	Apartment / Unit / Flat	Australian Capital Territory	1074000	7	5
12	2019	House	Australian Capital Territory	1640000	11	4
13	2020	House	Victoria	119785499	2	1
14	2020	House	New South Wales	63725850	3	1
15	2020	House	Australian Capital Territory	48806000	4	1
16	2020	House	Western Australia	32241000	5	1
17	2020	House	South Australia	21371000	6	1
18	2020	House	Tasmania	2885000	9	1

For Version – 2

Query Result

SQL | Fetched 50 rows in 0.023 seconds

	TO_CHAR(D.SALE_DATE,'YYYY')	PROPERTY_TYPE	YEAR	TOTAL_SALES	RANK_BY_TYPE	RANK_BY_STATE
1	2019	Studio	Victoria	80000	2	8
2	2019	House	New South Wales	850000	13	7
3	2019	Townhouse	Queensland	165000	9	11
4	2019	Townhouse	Victoria	759000	8	7
5	2019	House	South Australia	1190000	12	4
6	2019	Duplex	Queensland	1100000	3	9
7	2019	House	Victoria	2474000	10	4
8	2019	Apartment / Unit / Flat	New South Wales	439000	8	9
9	2019	House	Queensland	3066950	8	6
10	2019	House	Western Australia	3295000	7	3
11	2019	Apartment / Unit / Flat	Australian Capital Territory	1074000	7	5
12	2019	House	Australian Capital Territory	1640000	11	4
13	2020	House	Victoria	119785499	2	1
14	2020	House	New South Wales	63725850	3	1
15	2020	House	Australian Capital Territory	48806000	4	1
16	2020	House	Western Australia	32241000	5	1
17	2020	House	South Australia	21371000	6	1
18	2020	House	Tasmania	2885000	9	1

ii) Report – 12

a) Show ranking of property_type based on total_no_rents and ranking of states based on total_no_rents

b) Management would like to which property_type and states have highest number of rents by clients to see property trending states and consumer behaviour for each property type.

c) Command for query are below:

```
SELECT r.property_type ,state_name,sum(total_no_of_rent) as no_of_rents,

DENSE_RANK() over (PARTITION BY r.property_type ORDER BY sum(total_no_of_rent) desc)  as rank_by_type,

dense_rank() over (PARTITION BY state_name ORDER BY sum(total_no_of_rent) desc)  as rank_by_state

from rent_fact_v1 r,rent_dim_v1 d,type_dim_v1 t,location_dim_v1 l,postcode_dim_v1 p,state_dim_v1 s

where r.rent_id = d.rent_id and r.property_type = t.property_type and r.address_id = l.address_id

and l.postcode = p.postcode and p.state_code = s.state_code group by r.property_type, state_name;
```

--version 2

```
SELECT r.property_type ,state_name,sum(total_no_of_rent) as no_of_rents,

DENSE_RANK() over (PARTITION BY r.property_type ORDER BY sum(total_no_of_rent) desc)  as rank_by_type,

dense_rank() over (PARTITION BY state_name ORDER BY sum(total_no_of_rent) desc)  as rank_by_state

from rent_fact_v2 r,rent_dim_v2 d,type_dim_v2 t,location_dim_v2 l,postcode_dim_v2 p,state_dim_v2 s

where r.rent_id = d.rent_id and r.property_type = t.property_type and r.address_id = l.address_id

and l.postcode = p.postcode and p.state_code = s.state_code group by r.property_type, state_name;
```

d) Screenshots of query for both the versions are below:

For Version – 1

Query Result | Query Result 1 | Query Result 2

SQL | All Rows Fetched: 43 in 0.011 seconds

	PROPERTY_TYPE	STATE_NAME	NO_OF_RENTS	RANK_BY_TYPE	RANK_BY_STATE
1	Apartment / Unit / Flat	Queensland	161	1	2
2	Apartment / Unit / Flat	Australian Capital Territory	129	2	2
3	Apartment / Unit / Flat	New South Wales	102	3	2
4	Apartment / Unit / Flat	Western Australia	26	4	2
5	Apartment / Unit / Flat	South Australia	26	4	2
6	Apartment / Unit / Flat	Victoria	13	5	3
7	Apartment / Unit / Flat	Northern Territory	2	6	1
8	Block of Units	Queensland	4	1	5
9	Duplex	Western Australia	2	1	6
10	Duplex	Australian Capital Territory	1	2	5
11	Duplex	Queensland	1	2	7
12	Duplex	New South Wales	1	2	7
13	House	Queensland	313	1	1
14	House	South Australia	169	2	1
15	House	Western Australia	164	3	1
16	House	Australian Capital Territory	130	4	1
17	House	New South Wales	125	5	1
18	House	Victoria	83	6	1

For Version – 2

Query Result | Query Result 1 | Query Result 2

SQL | All Rows Fetched: 43 in 0.011 seconds

	PROPERTY_TYPE	STATE_NAME	NO_OF_RENTS	RANK_BY_TYPE	RANK_BY_STATE
1	Apartment / Unit / Flat	Queensland	161	1	2
2	Apartment / Unit / Flat	Australian Capital Territory	129	2	2
3	Apartment / Unit / Flat	New South Wales	102	3	2
4	Apartment / Unit / Flat	Western Australia	26	4	2
5	Apartment / Unit / Flat	South Australia	26	4	2
6	Apartment / Unit / Flat	Victoria	13	5	3
7	Apartment / Unit / Flat	Northern Territory	2	6	1
8	Block of Units	Queensland	4	1	5
9	Duplex	Western Australia	2	1	6
10	Duplex	Australian Capital Territory	1	2	5
11	Duplex	Queensland	1	2	7
12	Duplex	New South Wales	1	2	7
13	House	Queensland	313	1	1
14	House	South Australia	169	2	1
15	House	Western Australia	164	3	1
16	House	Australian Capital Territory	130	4	1
17	House	New South Wales	125	5	1
18	House	Victoria	83	6	1

TASK C.4 Business Intelligence (BI) Reports.

REPORT 4

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

--For Version 1

SELECT DISTINCT L.SUBURB,TO_CHAR(R.RENT_START_DATE,'MMYYYY') AS TIME_PERIOD ,SUM(TOTAL_RENTAL_FEE) AS TOTAL_RENTAL_FEE,F.PROPERTY_TYPE

FROM RENT_FACT_V1 F,ADDRESS L ,RENT_DIM_V1 R,TYPE_DIM_V1 T

where F.ADDRESS_ID = L.ADDRESS_ID AND F.PROPERTY_TYPE = T.PROPERTY_TYPE

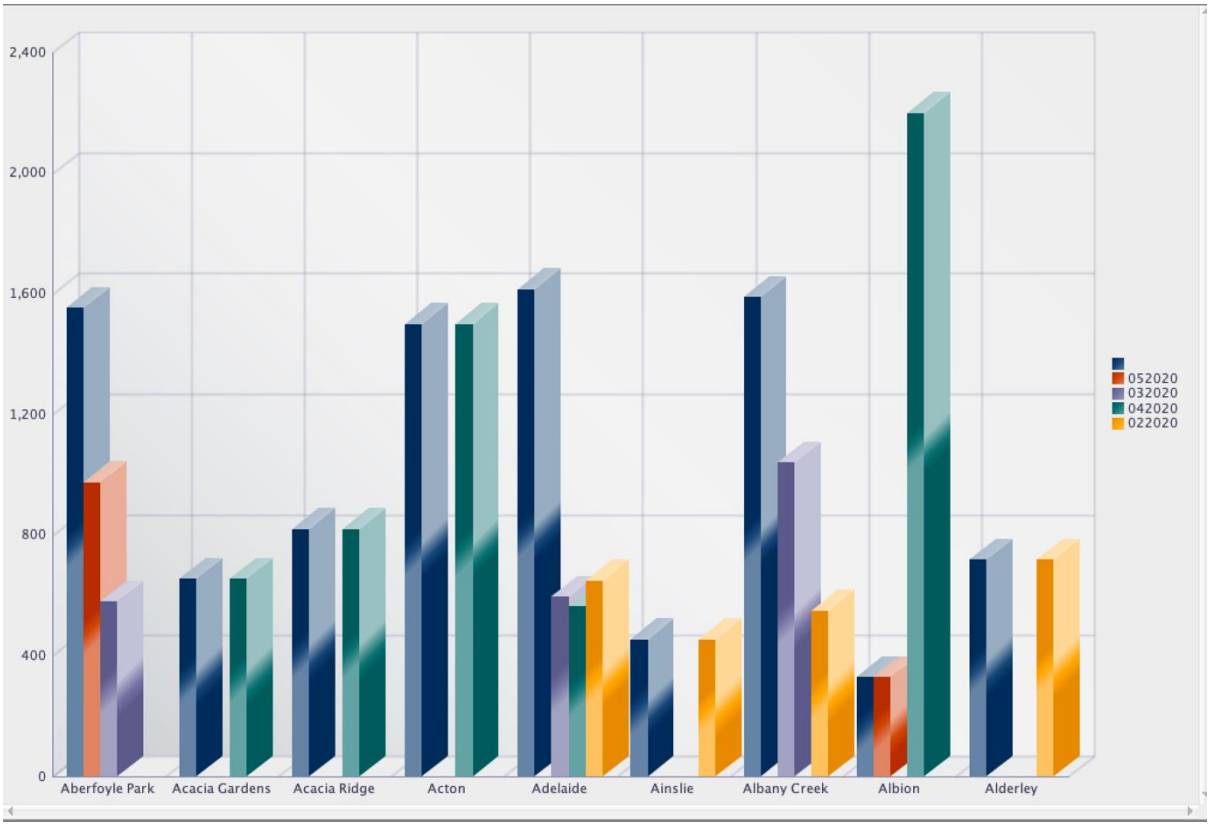
and f.rent_id = r.rent_id

GROUP BY CUBE(L.SUBURB,TO_CHAR(R.RENT_START_DATE,'MMYYYY')),F.PROPERTY_TYPE

ORDER BY L.SUBURB;

```
--What are the sub-total and total rental fees from each suburb, time period, and property type? (Using Cube)

--For Version 1
SELECT DISTINCT L.SUBURB,TO_CHAR(R.RENT_START_DATE,'MMYYYY') AS TIME_PERIOD ,SUM(TOTAL_RENTAL_FEE) AS TOTAL_RENTAL_FEE,F.PROPERTY_TYPE
FROM RENT_FACT_V1 F,ADDRESS L  ,RENT_DIM_V1 R,TYPE_DIM_V1 T
where F.ADDRESS_ID = L.ADDRESS_ID AND F.PROPERTY_TYPE = T.PROPERTY_TYPE
and f.rent_id = r.rent_id
GROUP BY CUBE( L.SUBURB,TO_CHAR(R.RENT_START_DATE,'MMYYYY')),F.PROPERTY_TYPE
ORDER BY L.SUBURB;
```



REPORT 6

--What are the total number of male and female agents in each agent office ?

--For Version 2

SELECT DECODE(GROUPING(a.GENDER),1,'All Genders',a.GENDER) AS GENDER,
DECODE(GROUPING(n.office_name),1,

'n.office_name',n.office_name)as office_name, COUNT(Total_No_Agent) AS No_Of_Agents,

DECODE(GROUPING(f.office_id),1,'f.office_id',f.office_id)As office_id from

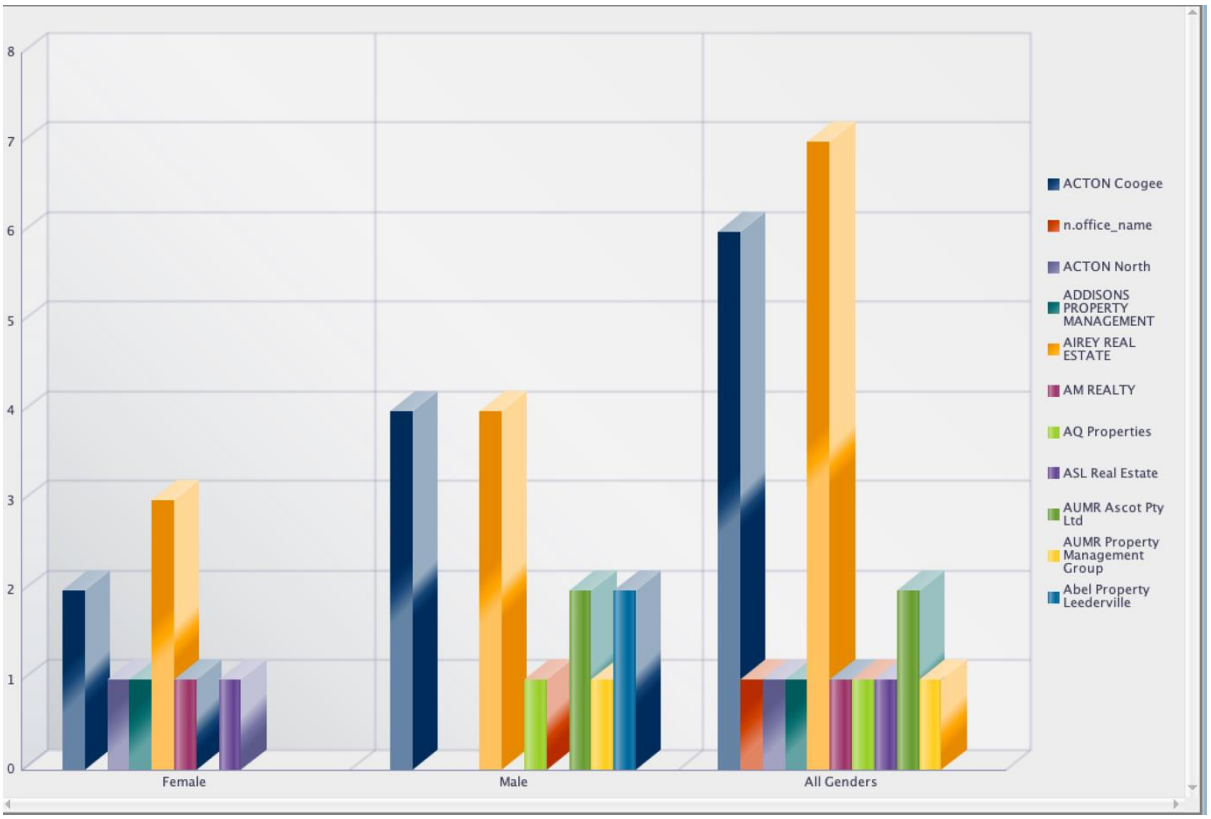
agent_fact_v1 f ,agent_office_dim_v1 o,office_dim_v1 n,agent_dim_v1 a

where f.office_id = o.office_id and o.office_id = n.office_id and f.agent_id = a.agent_id

group by rollup (f.OFFICE_ID,n.OFFICE_NAME,a.Gender)

order by f.office_id;

```
--What are the total number of male and female agents in each agent office ?
--For Version 2
SELECT DECODE(GROUPING(a.GENDER),1,'All Genders',a.GENDER) AS GENDER, DECODE(GROUPING(n.office_name),1,
'n.office_name',n.office_name)as office_name, COUNT(Total_No_Agent) AS No_Of_Agents,
DECODE(GROUPING(f.office_id),1,'f.office_id',f.office_id)As office_id from
agent_fact_v1 f ,agent_office_dim_v1 o,office_dim_v1 n,agent_dim_v1 a
where f.office_id = o.office_id and o.office_id = n.office_id and f.agent_id = a.agent_id
group by rollup (f.OFFICE_ID,n.OFFICE_NAME,a.Gender)
order by f.office_id;
```



REPORT 9

-- What is the total number of sales for each suburb and cumulative sales for each suburb?

--For Version 2

SELECT l.suburb,to_char(sale_date,'Mon') as year ,sum(total_no_of_sale) as monthly_no_sales ,

sum(sum(total_no_of_sale)) over

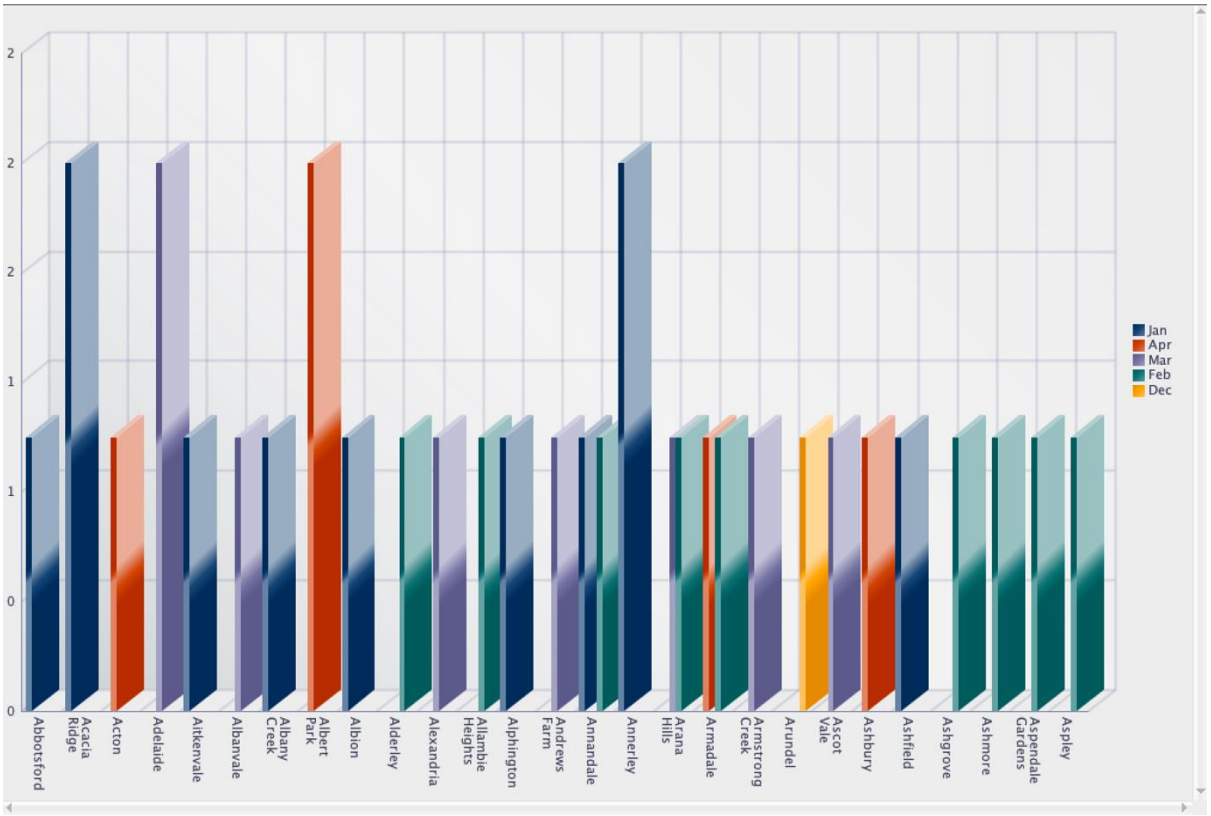
(partition by l.suburb order by l.suburb,to_char(sale_date,'Mon') rows unbounded preceding) As Cumulative_Sales

from sales_fact_v2 s,sale_dim_v2 d,location_dim_v2 l where

s.address_id = l.address_id and s.sale_id = d.sale_id

group by l.suburb,to_char(sale_date,'Mon');

```
-- What is the total number of sales for each suburb and cumulative sales for each suburb?
--For Version 2
SELECT l.suburb,to_char(sale_date,'Mon') as year ,sum(total_no_of_sale) as monthly_no_sales ,
sum(sum(total_no_of_sale)) over
(partition by l.suburb order by l.suburb,to_char(sale_date,'Mon') rows unbounded preceding) As Cumulative_Sales
from sales_fact_v2 s,sale_dim_v2 d,location_dim_v2 l where
s.address_id = l.address_id and s.sale_id = d.sale_id
group by l.suburb,to_char(sale_date,'Mon');
```



REPORT 10

--Q Show the number of properties added by each month and year and cumulative number of properties added ?

--For Version 1

SELECT TO_CHAR(property_date_added,'MM') as Month,TO_CHAR(property_date_added,'YYYY') as Year ,sum(total_no_properties) as total_properties_added,

sum(sum(total_no_properties)) over (order by TO_CHAR(property_date_added,'YYYY'),TO_CHAR(property_date_added,'MM') rows unbounded preceding) AS Cumulative_Prop_Added

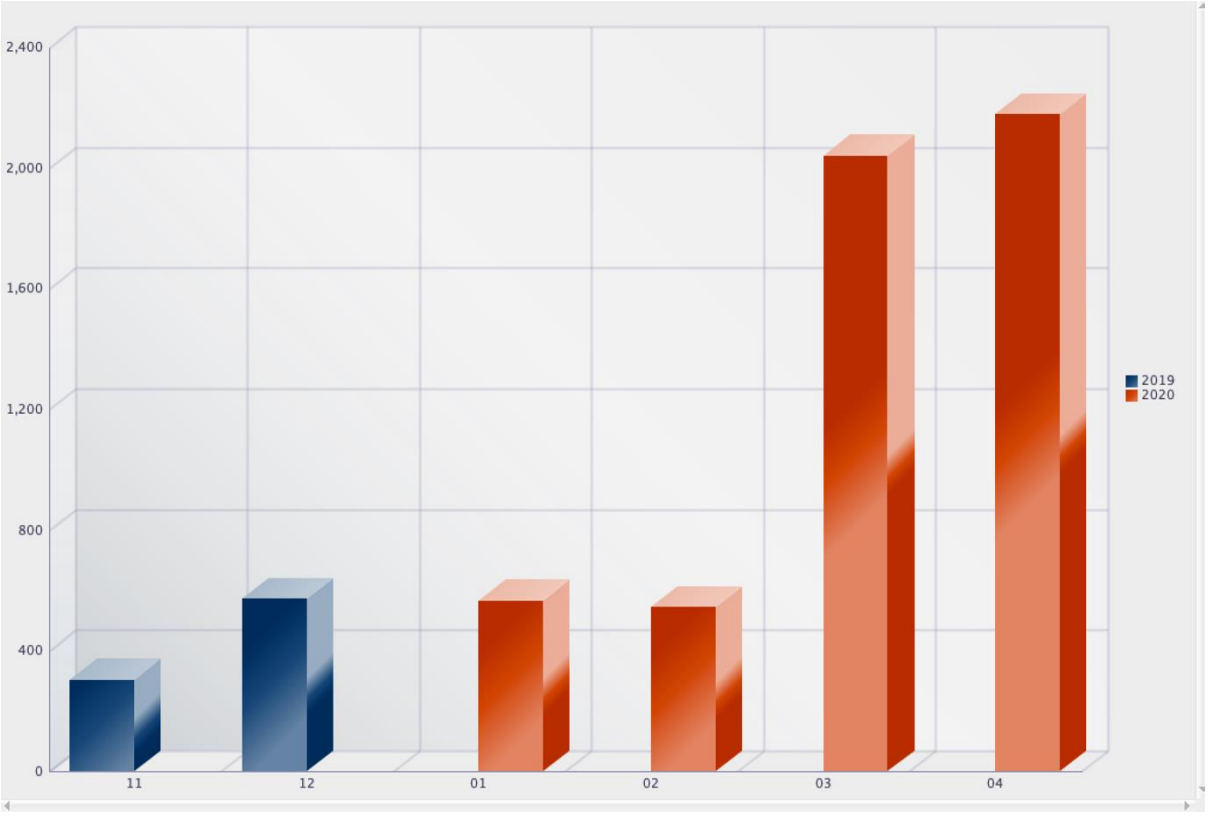
from advert_fact_v1 ,property_dim_v1 where

advert_fact_v1.property_id = property_dim_v1.property_id

group by TO_CHAR(property_date_added,'YYYY'),TO_CHAR(property_date_added,'MM')

order by TO_CHAR(property_date_added,'YYYY'),TO_CHAR(property_date_added,'MM');

```
--Q Show the number of properties added by each month and year and cumulative number of properties added ?
--For Version 1
SELECT TO_CHAR(property_date_added,'MM') as Month,TO_CHAR(property_date_added,'YYYY') as Year,
sum(total_no_properties) as total_properties_added, sum(sum(total_no_properties)) over (order by TO_CHAR(property_date_added,'YYYY'),
TO_CHAR(property_date_added,'MM') rows unbounded preceding) AS Cumulative_Prop_Added
from advert_fact_v1 ,property_dim_v1 where
advert_fact_v1.property_id = property_dim_v1.property_id
group by TO_CHAR(property_date_added,'YYYY'),TO_CHAR(property_date_added,'MM')
order by  TO_CHAR(property_date_added,'YYYY'),TO_CHAR(property_date_added,'MM');
```



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 2

SELECT to_char(d.sale_date,'yyyy') as Year, state_name ,

DENSE_RANK() over (PARTITION BY s.property_type ORDER BY SUM(total_amount_of_sales) desc) as rank_by_type,

dense_rank() over (PARTITION BY state_name ORDER BY SUM(total_amount_of_sales) desc) as rank_by_state,

s.property_type, SUM(total_amount_of_sales) as total_sales

from sales_fact_v2 s,sale_dim_v2 d,type_dim_v2 t,location_dim_v2 l,postcode_dim_v2 p,state_dim_v2 s

where s.sale_id = d.sale_id and s.property_type = t.property_type and s.address_id = l.address_id

and l.postcode = p.postcode and p.state_code = s.state_code group by to_char(d.sale_date,'yyyy'),s.property_type, state_name

order by to_char(d.sale_date,'yyyy') ;

```
-- 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 2
SELECT to_char(d.sale_date,'yyyy') as Year, state_name ,
DENSE_RANK() over (PARTITION BY s.property_type ORDER BY SUM(total_amount_of_sales) desc)  as rank_by_type,
dense_rank() over (PARTITION BY state_name ORDER BY SUM(total_amount_of_sales) desc)  as rank_by_state,
s.property_type, SUM(total_amount_of_sales) as total_sales
from sales_fact_v2 s,sale_dim_v2 d,type_dim_v2 t,location_dim_v2 l,postcode_dim_v2 p,state_dim_v2 s
where s.sale_id = d.sale_id and s.property_type = t.property_type and s.address_id = l.address_id
and  l.postcode = p.postcode and p.state_code = s.state_code group by to_char(d.sale_date,'yyyy'),s.property_type, state_name
order by to_char(d.sale_date,'yyyy') ;
```