



FIT5195 MAJOR ASSIGNMENT – SEM 1/2020

BUSINESS INTELLIGENCE AND DATA WAREHOUSING

Group Members:

Krishna Thakkar- 31057128

Anmol Bagati- 30535808

GROUP ASSIGNMENT COVER SHEET

| Student ID Number | Surname | Given Names |
|------------------------------------------------------------------------------------------------|------------------------------|----------------------------------------------------|
| 30535808 | Bagati | Anmol |
| 31057128 | Thakkar | Krishna |
| <small>* Please include the names of all other group members.</small> | | |
| Unit name and code | | FIT5195 Business Intelligence and Data Warehousing |
| Title of assignment | Major Assignment | |
| Lecturer/tutor | Agnes Haryanto | |
| Tutorial day and time | Wednesday 4:00 PM to 6:00 PM | |
| Is this an authorised group assignment? — Yes | | Campus: Caulfield |
| Has any part of this assignment been previously submitted as part of another unit/course? — No | | |
| <i>Due Date: 15/06/2020</i> | | |
| | | Date submitted : 21 st June 2020 |

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

Extension granted until (date) 21/06/2020 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 The material used can be from any source (staff, students or the internet, published and unpublished works).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 Date.....

* delete (iii) if not applicable

Signature : Krishna

Date : 21st June 2020

Signature : Anmol

Date : 21st June 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.

Oracle details of Team Members:

Anmol Bagati - S30535808

Password - Student

Krishna Thakkar - S30157128

Password- Student

Contribution Declaration Form

1 NAME AND CONTRIBUTION DETAILS

| Student ID | Student Name | Contribution Percentage |
|------------|-----------------|-------------------------|
| 30535808 | Anmol Bagati | 50 |
| 31057128 | Krishna Thakkar | 50 |

2 DECLARATION

- 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

Krishna

Anmol

Day Month Year

21 / 06 / 2020

| Tasks | Anmol | Krishna |
|--------------------------------|-------|---------|
| Database building and cleaning | 50 | 50 |
| ER Diagram | 50 | 50 |
| Star Schema- Low Level | 50 | 50 |
| Star Schema- High Level | 50 | 50 |
| OLAP QUERIES- Low Level | 50 | 50 |
| OLAP QUERIES- High Level | 50 | 50 |

| | | |
|---------------|----|----|
| Documentation | 50 | 50 |
|---------------|----|----|

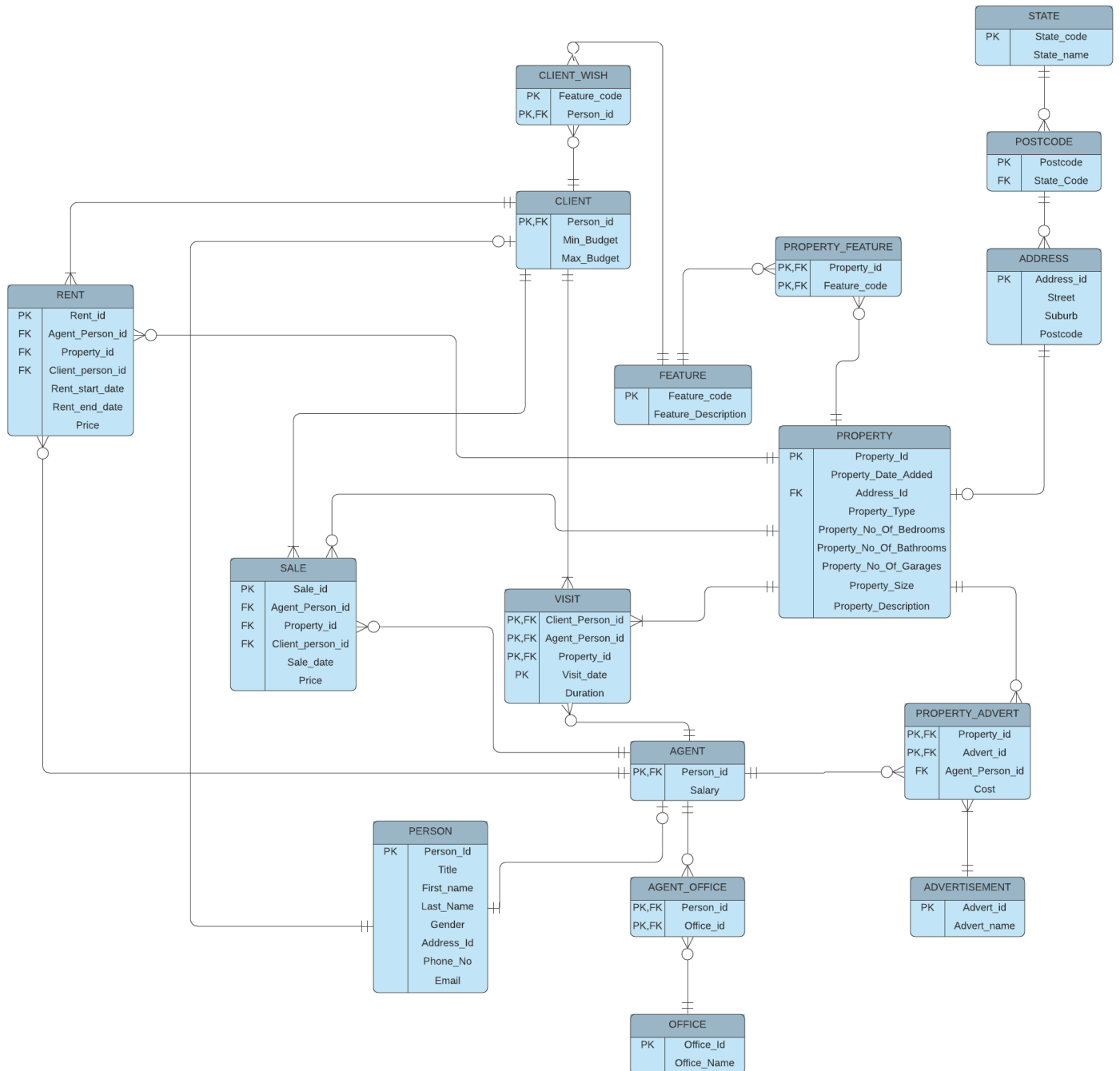
Note : The assignment has been done equally by both the group members together.

Task 1

The E/R diagram of the operational database

ERD - Major Assignment - BIDW

Anmol Bagati & Krishna Thakkar



Strategies used in the cleaning all the tables:

- Imported data from MONRE account using:

```
select * from monre.tablename;
```

- Count the number of rows to know if there are any duplicate data using:

```
select count (*) from (select attribute_name1, count (*) from MONRE.tablename group by attribute_name1 having count (*)>1);
```

- While creating our own database, we use select *distinct* to remove all duplicate line entries.
- Checking for null value

```
select * from monre.tablename where attribute_name is null;
```

- Checking for foreign key validity and presence of that key value in the parent table

```
select * from MonRE.tablename where attribute_name NOT IN (select attribute_name from monre.attribute_name);
```

- While creating our own operational database, we use *NOT IN* SQL command to remove all values in the child tables which are not present in the parent table.
- After following these steps tables are created for the operational database with correcting all the errors.

These strategies are followed for all the tables as shown below:

1.1 Address table **before**: (No Error)

| ADDRESS_ID | STREET | SUBURB | POSTCODE |
|------------|--------------------------|-----------------|----------|
| 1 | 526 63 Lindsay Street | Ashgrove | 4060 |
| 2 | 527 10 Flynn Place | Aspley | 4034 |
| 3 | 528 25 Attewood Avenue | Marsden | 4132 |
| 4 | 529 23/223 Tufnell Road | Banyo | 4014 |
| 5 | 530 17 Buxton Street | Ascot | 4007 |
| 6 | 531 1 Spoonbill Drive | Elimbah | 4516 |
| 7 | 532 16 Victoria Street | Indooroopilly | 4068 |
| 8 | 533 88 Railway Parade | Woodridge | 4114 |
| 9 | 534 311/18 Thorn Street | Kangaroo Point | 4169 |
| 10 | 535 4/24-26 Ferry Road | West End | 4101 |
| 11 | 536 2/24-26 Ferry Road | West End | 4101 |
| 12 | 537 130 Alastair Street | Lota | 4179 |
| 13 | 538 19 Jubilee Court | Eatons Hill | 4037 |
| 14 | 539 119 Cedar Street | Wynnum | 4178 |
| 15 | 540 7/702 Kingston Road | Loganlea | 4131 |
| 16 | 541 22 Borbridge Street | North Lakes | 4509 |
| 17 | 542 306-308 Uhlmann Road | Burpengary East | 4505 |
| 18 | 543 6/11 Crichton Street | Yeerongpilly | 4105 |
| 19 | 544 17 Glendale Street | Caboolture | 4510 |

1.2 Address table **after**:

| ADDRESS_ID | STREET | SUBURB | POSTCODE |
|------------|--------------------------|-----------------|----------|
| 1 | 526 63 Lindsay Street | Ashgrove | 4060 |
| 2 | 527 10 Flynn Place | Aspley | 4034 |
| 3 | 528 25 Attewood Avenue | Marsden | 4132 |
| 4 | 529 23/223 Tufnell Road | Banyo | 4014 |
| 5 | 530 17 Buxton Street | Ascot | 4007 |
| 6 | 531 1 Spoonbill Drive | Elimbah | 4516 |
| 7 | 532 16 Victoria Street | Indooroopilly | 4068 |
| 8 | 533 88 Railway Parade | Woodridge | 4114 |
| 9 | 534 311/18 Thorn Street | Kangaroo Point | 4169 |
| 10 | 535 4/24-26 Ferry Road | West End | 4101 |
| 11 | 536 2/24-26 Ferry Road | West End | 4101 |
| 12 | 537 130 Alastair Street | Lota | 4179 |
| 13 | 538 19 Jubilee Court | Eatons Hill | 4037 |
| 14 | 539 119 Cedar Street | Wynnum | 4178 |
| 15 | 540 7/702 Kingston Road | Loganlea | 4131 |
| 16 | 541 22 Borbridge Street | North Lakes | 4509 |
| 17 | 542 306-308 Uhlmann Road | Burpengary East | 4505 |
| 18 | 543 6/11 Crichton Street | Yeerongpilly | 4105 |

2.1 Advertisement table **before**: (No Error)

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

2.2 Advertisement table **after**:

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

3.1 Agent table before:

Error 1: Person_id 6997 not in person table but present in agent table.

SQL code for error detection

```

47
48 --Checking for foreign key validity and presence of that key value in the parent table
49 --Person_id 6997 not in person table but present in agent table. Delete after making Operational DB
50 select * from MonRE.agent where person_id NOT IN (select person_id from monre.person);
51

```

Error detected:

| PERSON_ID | SALARY |
|-----------|--------|
| 1 | 6997 0 |

Error 2: Person_id 6844, 6000 have negative salary

SQL code for error detection:

```

51
52 --checking if salary is 0 or negative
53 select * from monre.agent where salary <= -1;
54

```

Error detected:

Script Output x Query Result x Query Result 1 x Query Result 2 x Query Result 3 x Query Re... x x x x x x x x x x x x

SQL | All Rows Fetched: 2 in 0.023 seconds

| | PERSON_ID | SALARY |
|---|-----------|---------|
| 1 | 6844 | -100000 |
| 2 | 6000 | -120000 |

Count(*) before:

Worksheet

Query Builder

```

60 commit;
61 select count(*) from monre.agent;
62 select count(*) from agent;
63
64 --CLEANING AGENT_OFFICE TABLE
65 select * from MonRE.Agent_Office;
66 select count() from (select person_id, office_id, count() from MONRE.agent_office group by person_id, office_id having count(*)>1);
67
68

```

Script Output x

Query Result x

Query Result 1 x

Query Result 2 x

Query Result 3 x

Query Result 4 x

Query Result 5 x

Query Result 6 x

Query R... x

x

🔍

📄

🔗

🗑️

SQL

All Rows Fetched: 1 in 0.006 seconds

📉

COUNT(*)

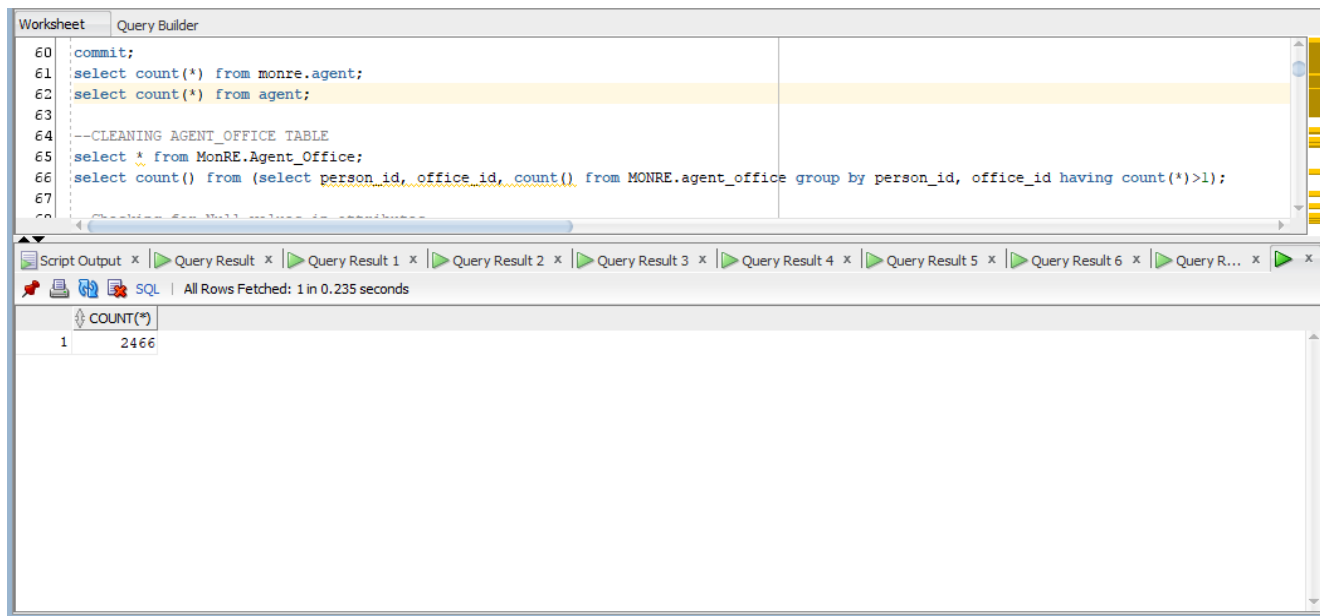
| | |
|---|------|
| 1 | 2469 |
|---|------|

3.2 Agent table after:

Removing both errors (3 entries removed)

| | PERSON_ID | SALARY |
|------|-----------|--------|
| 2449 | 587 | 210000 |
| 2450 | 845 | 190000 |
| 2451 | 894 | 175000 |
| 2452 | 1495 | 210000 |
| 2453 | 1516 | 175000 |
| 2454 | 1573 | 180000 |
| 2455 | 401 | 180000 |
| 2456 | 268 | 210000 |
| 2457 | 993 | 200000 |
| 2458 | 1096 | 190000 |
| 2459 | 510 | 175000 |
| 2460 | 644 | 200000 |
| 2461 | 761 | 200000 |
| 2462 | 870 | 180000 |
| 2463 | 1164 | 195000 |
| 2464 | 2114 | 190000 |
| 2465 | 2364 | 210000 |
| 2466 | 2451 | 200000 |

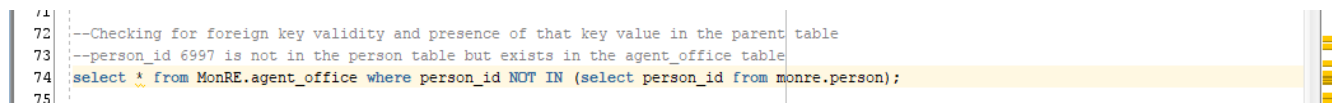
Count (*) after:



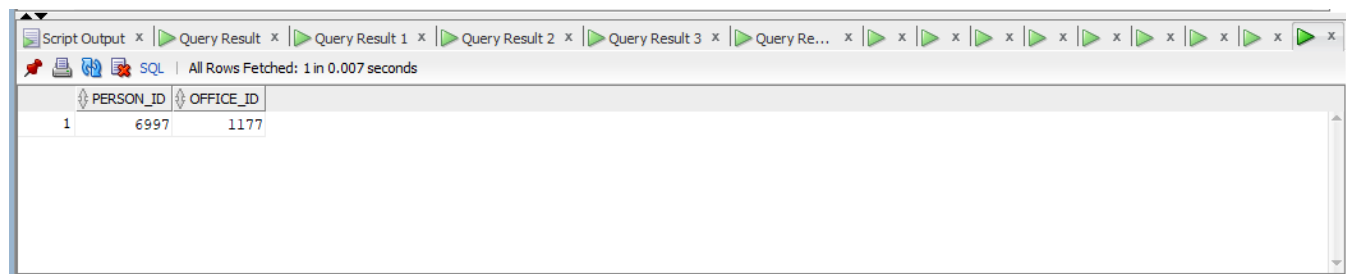
4.1 Agent_office table **before**:

Error 1: person_id 6997 is not in the person table but exists in the agent_office table

SQL code for error detection:



Error Detected:



Count (*) before:

Worksheet Query Builder

```

84
85 select count(*) from monre.agent_office;
86 select count(*) from agent_office;
87
88
89 --CLEANING CLIENT WISH TABLE

```

Script Output x Query Result x Query Result 1 x Query Result 2 x Query Result 3 x Query Result 4 x Qu... x x x x x x x x x x

SQL All Rows Fetched: 1 in 0.006 seconds

| | COUNT(*) |
|---|----------|
| 1 | 2529 |

4.2 Agent_office table **after**:
Error Corrected (one row deleted)

| | PERSON_ID | OFFICE_ID |
|------|-----------|-----------|
| 2515 | 1516 | 714 |
| 2516 | 1573 | 655 |
| 2517 | 401 | 717 |
| 2518 | 268 | 917 |
| 2519 | 993 | 10 |
| 2520 | 1096 | 3 |
| 2521 | 510 | 1108 |
| 2522 | 644 | 703 |
| 2523 | 761 | 176 |
| 2524 | 870 | 173 |
| 2525 | 1164 | 312 |
| 2526 | 2114 | 100 |
| 2527 | 2364 | 846 |
| 2528 | 2451 | 458 |

Count (*) after:

The screenshot shows a SQL Query Builder window with the following SQL code:

```

85 select count(*) from monre.agent_office;
86 select count(*) from agent_office;
87
88 --CLEANING CLIENT_WISH TABLE
89 select * from MonRE.Client_Wish;
90 select count() from (select person id, feature code, count() from MONRE.client wish group by person id, feature code having count(*)>1);

```

The query results are displayed in a table below the SQL editor:

| COUNT(*) |
|----------|
| 2528 |

The status bar at the bottom indicates: | Line 86 Column 35 | Insert | Modified | Windows: Cf

5.1 Client table before:

| | PERSON_ID | MIN_BUDGET | MAX_BUDGET |
|----|-----------|------------|------------|
| 1 | 3014 | 440100 | 537900 |
| 2 | 3020 | 490500 | 599500 |
| 3 | 3025 | 585000 | 715000 |
| 4 | 3029 | 607500 | 742500 |
| 5 | 3081 | 449100 | 548900 |
| 6 | 3087 | 1350000 | 1650000 |
| 7 | 3092 | 945000 | 1155000 |
| 8 | 3098 | 430200 | 525800 |
| 9 | 3103 | 467100 | 570900 |
| 10 | 3107 | 900000 | 1100000 |
| 11 | 3113 | 720000 | 880000 |
| 12 | 3162 | 179100 | 218900 |
| 13 | 3168 | 449100 | 548900 |
| 14 | 3173 | 450000 | 550000 |
| 15 | 3178 | 193500 | 236500 |
| 16 | 3184 | 720000 | 880000 |
| 17 | 3189 | 900000 | 1100000 |
| 18 | 3194 | 350100 | 427900 |
| 19 | 3244 | 512100 | 625900 |

Error 1: min_budget is lower than max_budget

SQL code for error detection:

```

192
193 --checking if min_budget is lower than max_budget
194 select * from monre.client where min_budget > max_budget;
195

```

Error detected:

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

Error 2: Person_id 7000 not in person table but present in client table




SQL code for error detection:

```
select * from MonRE.client where person_id NOT IN (select person_id from person);
select count(*) from client;
```

Error detected:

cript Output x

Query Result x

   SQL | All Rows Fetched: 2 in 0.022 seconds

| | PERSON_ID | MIN_BUDGET | MAX_BUDGET |
|---|-----------|------------|------------|
| 1 | 7001 | 7500 | 11250 |
| 2 | 7000 | 8500 | 15050 |

Count(*) before:

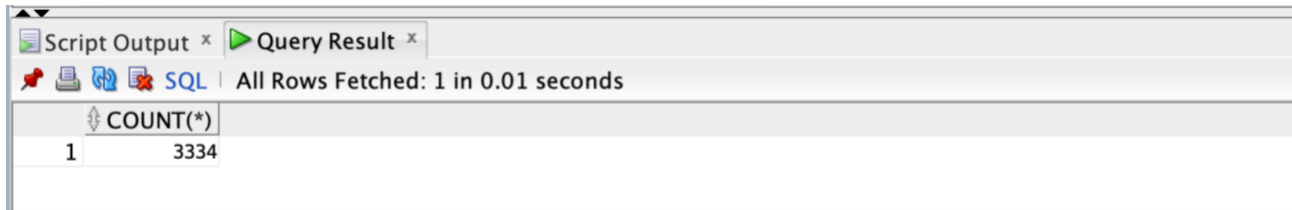
| | COUNT(*) |
|---|----------|
| 1 | 3339 |

5.2 Client table **after**:

Error corrected (4 Rows Deleted)

| | PERSON_ID | MIN_BUDGET | MAX_BUDGET |
|----|-----------|------------|------------|
| 1 | 2696 | 404100 | 493900 |
| 2 | 2697 | 258300 | 315700 |
| 3 | 2698 | 449100 | 548900 |
| 4 | 2700 | 495000 | 605000 |
| 5 | 2701 | 508500 | 621500 |
| 6 | 2702 | 810000 | 990000 |
| 7 | 2703 | 382500 | 467500 |
| 8 | 2704 | 630000 | 770000 |
| 9 | 2705 | 450000 | 550000 |
| 10 | 2706 | 540000 | 660000 |
| 11 | 2707 | 425500 | 531400 |

Count (*) after:

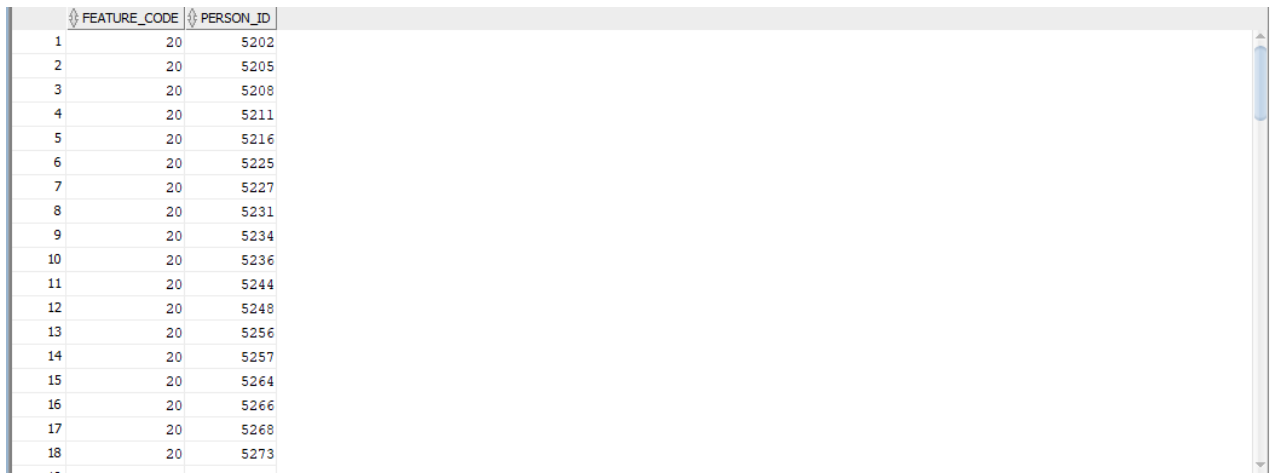


Script Output x Query Result x

SQL | All Rows Fetched: 1 in 0.01 seconds

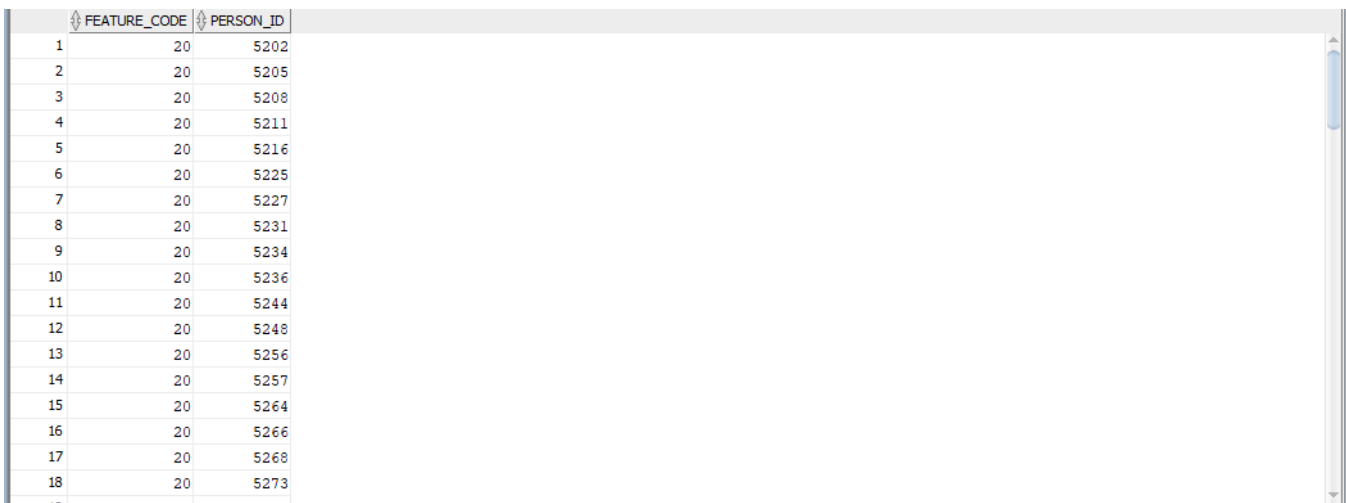
| | COUNT(*) |
|---|----------|
| 1 | 3334 |

6.1 Client_wish table **before**: (No Error)



| | 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 |

6.2 Client_wish table **after**:



| | 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 |

7.1 Feature table **before**: (No Error)

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

7.2 Feature table **after**:

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

8.1 Office table **before**: (No Error)

| OFFICE_ID | OFFICE_NAME |
|-----------|------------------------------------|
| 1 | 910 Ray White Manly QLD |
| 2 | 911 Ray White Mawson Lakes |
| 3 | 912 Ray White Meadowbank |
| 4 | 913 Ray White Metro West |
| 5 | 914 Ray White Moorooka |
| 6 | 915 Ray White Mordialloc |
| 7 | 916 Ray White Mount Gravatt |
| 8 | 917 Ray White Nerang |
| 9 | 918 Ray White New Farm |
| 10 | 919 Ray White Nolan & Iken |
| 11 | 920 Ray White North Adelaide |
| 12 | 921 Ray White North Ipswich |
| 13 | 922 Ray White North Lakes |
| 14 | 923 Ray White North Quays Sorrento |
| 15 | 924 Ray White Norwood |
| 16 | 925 Ray White Oakleigh |
| 17 | 926 Ray White Oatley |
| 18 | 927 Ray White Ormeau |
| 19 | 928 Ray White Pacific Pines |

8.2 Office table **after**:

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

9.1 Person table **before**:

Error 1: person_id 6995 has repeated entries

SQL code for error detection:

```

139
140 --person_id 6995 has repeated entries
141 select count() from (select person_id, count() from MONRE.person group by person_id having count(*)>1);
142 select count(), person_id from monre.person group by person_id having count(*)>1;
143 select * from monre.person where person_id = '6995';
144

```

Error detected:

| 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 2: Person_id 7001 has an address_id that is not present in the address table

SQL code for error detection:

```

Worksheet  Query Builder
158 --Checking for foreign key validity and presence of that key value in the parent table
159 --Person id 7001 has an address_id that is not present in the address table
160 select * from MonRE.person where address_id NOT IN (select address_id from monre.address);
161 select count(*) from person;
162

```

Error Detected:

| 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 3: Mismatch between title and gender

SQL code for error detection:

```

171 -- Updating gender and title discrepancies
172 select title, gender, count(*) from monre.person group by title,gender;
173 select title, gender, count(*) from person group by title,gender;
174 UPDATE person
175 SET title='Mr'
176 WHERE gender='Male' and title='Mrs';

```

Error detection:

| TITLE | GENDER | COUNT(*) |
|--------|--------|----------|
| 1 Mr | Male | 1747 |
| 2 Mr | Female | 606 |
| 3 Mrs | Female | 562 |
| 4 Dr | Male | 582 |
| 5 Ms | Female | 1741 |
| 6 Mrs | Male | 616 |
| 7 Dr | Female | 580 |
| 8 Ms | Male | 565 |
| 9 null | Male | 1 |

Count (*) before:

| COUNT(*) |
|----------|
| 1 7000 |

Line 137 Column 35 | Insert | Modified | Windows: C

9.2 Person table after:

Removing redundant data from the table:

| PERSON_ID | TITLE | FIRST_NAME | LAST_NAME | GENDER | ADDRESS_ID | PHONE_NO | EMAIL |
|-----------|----------|------------|------------|--------|------------|------------|-----------------------------|
| 1 | 982 Mr | Adolpho | Tregien | Male | 6641 | 8155537176 | atregienr9@goodreads.com |
| 2 | 989 Mr | Opaline | Fiske | Female | 6647 | 9944871590 | ofiskerg@gravatar.com |
| 3 | 1004 Ms | Ali | Ciotti | Female | 6659 | 3027340493 | aciotti3@goodreads.com |
| 4 | 1014 Ms | Krysta | Fearon | Female | 6667 | 1874628495 | kfearond@unesco.org |
| 5 | 1015 Mr | Henry | Buckbee | Male | 6668 | 1143871344 | hbuckbeee@yale.edu |
| 6 | 1018 Ms | Kimble | Nore | Male | 6671 | 1571127880 | knoreh@blog.com |
| 7 | 1024 Ms | Benson | Laverty | Male | 6677 | 7367997858 | blavertyn@aboutads.info |
| 8 | 1030 Ms | Onida | Clotworthy | Female | 6683 | 2303848991 | oclotworthyt@dion.ne.jp |
| 9 | 1056 Mrs | Debbie | Brace | Female | 6709 | 2261309945 | dbrace1j@ca.gov |
| 10 | 1058 Mrs | Pete | Roocroft | Male | 6711 | 8341978813 | proocroft11@hugedomains.com |
| 11 | 1081 Mr | Brandon | Witcombe | Male | 6729 | 8935483114 | bwitcombe28@prweb.com |
| 12 | 1085 Ms | Zacharias | Rodrigo | Male | 6732 | 4625601435 | zrodrigo2c@hibu.com |
| 13 | 1087 Dr | Alene | Kunath | Female | 6734 | 4421831984 | akunath2e@bloomberg.com |
| 14 | 9 Mr | Dasie | Mayston | Female | 6217 | 9221479008 | dmayston8@wix.com |
| 15 | 21 Ms | Lorine | Prinn | Female | 6229 | 2566429821 | lprinnk@google.ca |
| 16 | 34 Ms | Ada | Kitchaside | Female | 6242 | 9123291507 | akitchasidex@reuters.com |
| 17 | 62 Ms | Daffy | MacBey | Female | 6266 | 8249823482 | dmacbeylp@mac.com |
| 18 | 65 Dr | Roch | Storror | Female | 6268 | 3769037348 | rstorror1s@europa.eu |

updating gender and title discrepancies:

| TITLE | GENDER | COUNT(*) |
|-------|--------|----------|
| 1 Mr | Male | 2363 |
| 2 Mrs | Female | 1168 |
| 3 Ms | Female | 1741 |
| 4 Dr | Male | 582 |
| 5 Dr | Female | 577 |
| 6 Ms | Male | 565 |

Count(*) after:

| COUNT(*) |
|----------|
| 1 6996 |

10.1 Postcode table **before**: (No Error)

| | POSTCODE | STATE_CODE |
|----|----------|------------|
| 1 | 800 NT | |
| 2 | 2000 NSW | |
| 3 | 2007 NSW | |
| 4 | 2008 NSW | |
| 5 | 2009 NSW | |
| 6 | 2010 NSW | |
| 7 | 2011 NSW | |
| 8 | 2015 NSW | |
| 9 | 2016 NSW | |
| 10 | 2017 NSW | |
| 11 | 2018 NSW | |
| 12 | 2019 NSW | |
| 13 | 2020 NSW | |
| 14 | 2021 NSW | |
| 15 | 2022 NSW | |
| 16 | 2023 NSW | |
| 17 | 2024 NSW | |
| 18 | 2025 NSW | |
| 19 | 2026 NSW | |
| 20 | 2027 NSW | |

10.2 Postcode table **after**:

| | POSTCODE | STATE_CODE |
|----|----------|------------|
| 1 | 800 NT | |
| 2 | 2000 NSW | |
| 3 | 2007 NSW | |
| 4 | 2008 NSW | |
| 5 | 2009 NSW | |
| 6 | 2010 NSW | |
| 7 | 2011 NSW | |
| 8 | 2015 NSW | |
| 9 | 2016 NSW | |
| 10 | 2017 NSW | |
| 11 | 2018 NSW | |
| 12 | 2019 NSW | |
| 13 | 2020 NSW | |
| 14 | 2021 NSW | |
| 15 | 2022 NSW | |
| 16 | 2023 NSW | |
| 17 | 2024 NSW | |
| 18 | 2025 NSW | |
| 19 | 2026 NSW | |
| 20 | 2027 NSW | |

11.1 Property table **before**:

| | PROPERTY_ID | PROPERTY_DATE_ADDED | ADDRESS_ID | PROPERTY_TYPE | PROPERTY_NO_OF_BEDROOMS | PROPERTY_NO_OF_BATHROOMS | PROPERTY_NO_OF_GARAG |
|----|-------------|---------------------|------------|---------------|-------------------------|--------------------------|----------------------|
| 1 | 20 | 11/APR/20 | 20 | House | 4 | 2 | |
| 2 | 21 | 02/APR/20 | 21 | Townhouse | 4 | 2 | |
| 3 | 22 | 14/APR/20 | 22 | House | 3 | 2 | |
| 4 | 23 | 16/MAR/20 | 23 | House | 3 | 2 | |
| 5 | 24 | 01/DEC/19 | 24 | House | 3 | 2 | |
| 6 | 25 | 17/APR/20 | 25 | House | 4 | 2 | |
| 7 | 26 | 21/APR/20 | 26 | Townhouse | 4 | 3 | |
| 8 | 27 | 19/APR/20 | 27 | House | 4 | 2 | |
| 9 | 28 | 08/APR/20 | 28 | House | 4 | 2 | |
| 10 | 29 | 12/MAR/20 | 29 | House | 4 | 2 | |
| 11 | 30 | 13/FEB/20 | 30 | House | 4 | 2 | |
| 12 | 31 | 19/DEC/19 | 31 | House | 4 | 2 | |
| 13 | 32 | 14/APR/20 | 32 | House | 4 | 1 | |
| 14 | 33 | 24/DEC/19 | 33 | House | 4 | 2 | |
| 15 | 34 | 21/NOV/19 | 34 | House | 3 | 1 | |
| 16 | 35 | 12/APR/20 | 35 | House | 4 | 2 | |
| 17 | 36 | 05/MAR/20 | 36 | House | 5 | 2 | |
| 18 | 37 | 21/APR/20 | 37 | Townhouse | 3 | 2 | |
| 19 | 38 | 28/APR/20 | 38 | Townhouse | 2 | 1 | |

Error 1: 4 and 16 repeated entries of 2 different property_id

SQL code for error detection:

```

230 --CLEANING PROPERTY TABLE
231 -- 4 and 16 repeated entries of 2 different property_id
232 select * from MonRE.Property;
233 select count(*) from (select property_id, count(*) from MONRE.property group by property_id having count(*)>1);
234 select count(*), property_id from monre.property group by property_id having count(*)>1;
235 select * from monre.property where property_id = '6177' or property_id = '6179' order by property_id;
236

```

Error Detected:

| COUNT(*) | PROPERTY_ID |
|----------|-------------|
| 4 | 6177 |
| 16 | 6179 |

Null value in property_size attribute:

| JMS | PROPERTY_NO_OF_BATHROOMS | PROPERTY_NO_OF_GARAGES | PROPERTY_SIZE | PROPERTY_DESCRIPTION |
|-----|--------------------------|------------------------|---------------|----------------------------------------------------------------------------------|
| 1 | 4 | 2 | 2 | (null) Brilliantly designed and currently under construction these stunning arch |
| 2 | 3 | 2 | 2 | (null) This gorgeous single storey townhouse development is just a short stroll |
| 3 | 4 | 2 | 2 | (null) IMPORTANT NOTICE REGARDING THIS PROPERTY:This property is NOT available f |
| 4 | 4 | 2 | 1 | (null) You are able to view this property 7 days a week by private inspection. W |
| 5 | 3 | 2 | 2 | (null) 1 SOLD - 2 LEFT construction well under way lock up not too long!Are you |
| 6 | 2 | 1 | 2 | (null) PLEASE CALL TO BOOK YOUR PRIVATE INSPECTIONNB: All inspections will now b |
| 7 | 4 | 2 | 2 | (null) CONTACT US TODAY TO ARRANGE YOUR ONE-ON-ONE APPOINTMENTHeres your familys |
| 8 | 2 | 1 | 1 | (null) Be first in line for the new Geelong Flyer ferry to Melbourne, first to c |
| 9 | 2 | 2 | 1 | (null) This stylish, north-facing single level townhouse offers wonderful low ma |
| 10 | 4 | 2 | 2 | (null) CONTACT US TODAY TO ARRANGE YOUR ONE-ON-ONE APPOINTMENTIdeally located in |
| 11 | 3 | 1 | 1 | (null) Behind a picture-perfect picket fence is this ideal little entry point in |
| 12 | 3 | 1 | 6 | (null) PRIVATE VIEWINGS WELCOMED BY APPOINTMENT - BOOK NOWSituated in one of Ade |
| 13 | 2 | 1 | 1 | (null) Peace and quiet coupled with the effervescent Unley Road precinct makes t |
| 14 | 2 | 1 | 1 | (null) CONTACT US TODAY TO ARRANGE YOUR ONE-ON-ONE APPOINTMENTSnap up a rare opp |
| 15 | 3 | 2 | 1 | (null) Superbly located within walking distance to Waurm Ponds Shopping Centre, |
| 16 | 3 | 2 | 2 | (null) Pounce on this special opportunity to snap up a prime Newtown home site a |
| 17 | 3 | 2 | 2 | (null) Pounce on this special opportunity to snap up a prime Newtown home site a |
| 18 | 3 | 1 | 1 | (null) PAKO LIFESTYLE IS JUST AROUND THE CORNERCONTACT US TODAY TO ARRANGE YOUR |

Null value in Property_description:

| DDED | ADDRESS_ID | PROPERTY_TYPE | PROPERTY_NO_OF_BEDROOMS | PROPERTY_NO_OF_BATHROOMS | PROPERTY_NO_OF_GARAGES | PROPERTY_SIZE | PROPERTY_DESCRIPTION |
|------|------------|---------------|-------------------------|--------------------------|------------------------|---------------|----------------------|
| 1 | 3636 House | | 3 | 1 | 0 | (null) | (null) |
| 2 | 6198 House | | 3 | 1 | 2 | (null) | (null) |

Count (*) before:

| COUNT(*) |
|----------|
| 1 |
| 6226 |

11.2Property table after:

| PROPERTY_ID | PROPERTY_DATE_ADDED | ADDRESS_ID | PROPERTY_TYPE | PROPERTY_NO_OF_BEDROOMS | PROPERTY_NO_OF_BATHROOMS | PROPERTY_NO_OF_GARAGE |
|-------------|---------------------|-----------------------------|---------------|-------------------------|--------------------------|-----------------------|
| 1 | 28 08/APR/20 | 28 House | | 4 | 2 | |
| 2 | 36 05/MAR/20 | 36 House | | 5 | 2 | |
| 3 | 46 19/APR/20 | 46 House | | 4 | 2 | |
| 4 | 129 25/NOV/19 | 129 House | | 2 | 1 | |
| 5 | 131 19/NOV/19 | 131 House | | 3 | 1 | |
| 6 | 135 07/APR/20 | 135 House | | 3 | 1 | |
| 7 | 139 12/MAR/20 | 139 House | | 3 | 2 | |
| 8 | 49 13/APR/20 | 49 Townhouse | | 3 | 2 | |
| 9 | 53 04/APR/20 | 53 House | | 3 | 2 | |
| 10 | 60 12/APR/20 | 60 House | | 3 | 1 | |
| 11 | 4 05/APR/20 | 4 House | | 10 | 7 | |
| 12 | 5 15/DEC/19 | 5 House | | 6 | 2 | |
| 13 | 8 30/APR/20 | 8 House | | 3 | 2 | |
| 14 | 19 14/DEC/19 | 19 Apartment / Unit / Flat | | 2 | 2 | |
| 15 | 233 16/APR/20 | 233 House | | 2 | 1 | |
| 16 | 241 11/DEC/19 | 241 Apartment / Unit / Flat | | 2 | 1 | |
| 17 | 114 01/APR/20 | 114 Apartment / Unit / Flat | | 2 | 1 | |
| 18 | 216 22/APR/20 | 216 House | | 3 | 1 | |

Count(*) after

| COUNT(*) |
|----------|
| 1 6208 |

12.1 Property_Advert table **before:** (No Error)

| PROPERTY_ID | ADVERT_ID | AGENT_PERSON_ID | COST |
|-------------|-----------|-----------------|--------------|
| 1 | 2894 | 16 | 1345 1700000 |
| 2 | 2895 | 16 | 467 800000 |
| 3 | 2896 | 16 | 1345 735000 |
| 4 | 2897 | 16 | 774 680000 |
| 5 | 2898 | 16 | 95 500000 |
| 6 | 2899 | 16 | 2250 500000 |
| 7 | 2900 | 16 | 718 500000 |
| 8 | 2902 | 12 | 95 500000 |
| 9 | 2903 | 16 | 149 1100000 |
| 10 | 2300 | 16 | 1892 985000 |
| 11 | 2302 | 16 | 1887 977000 |
| 12 | 2303 | 16 | 1892 949000 |
| 13 | 2305 | 20 | 137 899000 |
| 14 | 2306 | 16 | 2304 898000 |
| 15 | 2307 | 16 | 142 879000 |
| 16 | 2506 | 16 | 2305 1500000 |
| 17 | 2507 | 16 | 1058 595000 |
| 18 | 2508 | 16 | 1058 690000 |
| 19 | 2509 | 16 | 1027 500000 |

12.2 Property_Advert table **after:**

| | PROPERTY_ID | ADVERT_ID | AGENT_PERSON_ID | COST |
|----|-------------|-----------|-----------------|---------|
| 1 | 2894 | 16 | 1345 | 1700000 |
| 2 | 2895 | 16 | 467 | 800000 |
| 3 | 2896 | 16 | 1345 | 735000 |
| 4 | 2897 | 16 | 774 | 680000 |
| 5 | 2898 | 16 | 95 | 500000 |
| 6 | 2899 | 16 | 2250 | 500000 |
| 7 | 2900 | 16 | 718 | 500000 |
| 8 | 2902 | 12 | 95 | 500000 |
| 9 | 2903 | 16 | 149 | 1100000 |
| 10 | 2300 | 16 | 1892 | 985000 |
| 11 | 2302 | 16 | 1887 | 977000 |
| 12 | 2303 | 16 | 1892 | 949000 |
| 13 | 2305 | 20 | 137 | 899000 |
| 14 | 2306 | 16 | 2304 | 898000 |
| 15 | 2307 | 16 | 142 | 879000 |
| 16 | 2506 | 16 | 2305 | 1500000 |
| 17 | 2507 | 16 | 1058 | 595000 |
| 18 | 2508 | 16 | 1058 | 690000 |

13.1 Property_feature table **before**: (No Error)

| | PROPERTY_ID | FEATURE_CODE |
|----|-------------|--------------|
| 1 | 9 | 1 |
| 2 | 9 | 2 |
| 3 | 9 | 3 |
| 4 | 9 | 4 |
| 5 | 9 | 5 |
| 6 | 9 | 6 |
| 7 | 9 | 7 |
| 8 | 9 | 8 |
| 9 | 9 | 9 |
| 10 | 9 | 10 |
| 11 | 9 | 11 |
| 12 | 9 | 12 |
| 13 | 9 | 117 |
| 14 | 11 | 1 |
| 15 | 11 | 2 |
| 16 | 11 | 5 |
| 17 | 11 | 6 |
| 18 | 11 | 7 |
| 19 | 11 | 9 |

13.2 Property_feature table **after**:

| | PROPERTY_ID | FEATURE_CODE |
|----|-------------|--------------|
| 1 | 9 | 1 |
| 2 | 9 | 2 |
| 3 | 9 | 3 |
| 4 | 9 | 4 |
| 5 | 9 | 5 |
| 6 | 9 | 6 |
| 7 | 9 | 7 |
| 8 | 9 | 8 |
| 9 | 9 | 9 |
| 10 | 9 | 10 |
| 11 | 9 | 11 |
| 12 | 9 | 12 |
| 13 | 9 | 117 |
| 14 | 11 | 1 |
| 15 | 11 | 2 |
| 16 | 11 | 5 |
| 17 | 11 | 6 |
| 18 | 11 | 7 |
| 19 | 11 | 9 |

14.1 Rent table before:

| | RENT_ID | AGENT_PERSON_ID | CLIENT_PERSON_ID | PROPERTY_ID | RENT_START_DATE | RENT_END_DATE | PRICE |
|----|---------|-----------------|------------------|-------------|-----------------|---------------|-------|
| 1 | 331 | 568 | 3713 | 6199 | 12/JAN/20 | 28/JUN/20 | 795 |
| 2 | 332 | 568 | 3714 | 6063 | 02/MAY/20 | 18/OCT/20 | 500 |
| 3 | 333 | 568 | 3715 | 6074 | 01/MAY/20 | 17/OCT/20 | 370 |
| 4 | 334 | 568 | 3716 | 6142 | 12/FEB/20 | 29/JUL/20 | 795 |
| 5 | 335 | 568 | 3717 | 6146 | 20/APR/20 | 06/OCT/20 | 595 |
| 6 | 336 | 570 | 3718 | 5373 | 27/APR/20 | 13/OCT/20 | 350 |
| 7 | 337 | 570 | 3719 | 5801 | 25/FEB/20 | 11/AUG/20 | 600 |
| 8 | 338 | 570 | 3720 | 5513 | 01/JAN/20 | 17/JUN/20 | 430 |
| 9 | 339 | 570 | 3721 | 5709 | 29/MAR/20 | 13/SEP/20 | 420 |
| 10 | 340 | 571 | 3722 | 5548 | 23/APR/20 | 09/OCT/20 | 520 |
| 11 | 341 | 571 | 3723 | 5901 | 01/MAY/20 | 17/OCT/20 | 330 |
| 12 | 342 | 571 | 3724 | 5724 | 01/MAY/20 | 17/OCT/20 | 500 |
| 13 | 343 | 571 | 3725 | 6035 | 30/APR/20 | 16/OCT/20 | 625 |
| 14 | 344 | 572 | 3726 | 5557 | 23/APR/20 | 09/OCT/20 | 815 |
| 15 | 345 | 572 | 3727 | 5621 | 21/APR/20 | 07/OCT/20 | 370 |
| 16 | 346 | 573 | 3728 | 5598 | 23/APR/20 | 09/OCT/20 | 495 |
| 17 | 347 | 574 | 3729 | 5386 | 18/MAR/20 | 02/SEP/20 | 1100 |
| 18 | 348 | 575 | 3730 | 5766 | 18/MAR/20 | 02/SEP/20 | 430 |
| 19 | 349 | 575 | 3731 | 6070 | 24/JAN/20 | 10/JUL/20 | 330 |

Null values in Client_person_id, Rent_start_date and Rent_end_date

| | RENT_ID | AGENT_PERSON_ID | CLIENT_PERSON_ID | PROPERTY_ID | RENT_START_DATE | RENT_END_DATE | PRICE |
|----|---------|-----------------|------------------|-------------|-----------------|---------------|-------|
| 1 | 2378 | 1155 | (null) | 5864 (null) | (null) | (null) | 590 |
| 2 | 2379 | 1155 | (null) | 5891 (null) | (null) | (null) | 350 |
| 3 | 2380 | 1155 | (null) | 5535 (null) | (null) | (null) | 1200 |
| 4 | 2381 | 1155 | (null) | 5989 (null) | (null) | (null) | 420 |
| 5 | 2382 | 1156 | (null) | 5571 (null) | (null) | (null) | 290 |
| 6 | 2383 | 1156 | (null) | 5851 (null) | (null) | (null) | 700 |
| 7 | 2384 | 1157 | (null) | 5962 (null) | (null) | (null) | 550 |
| 8 | 2385 | 1158 | (null) | 5970 (null) | (null) | (null) | 290 |
| 9 | 2386 | 1158 | (null) | 6144 (null) | (null) | (null) | 485 |
| 10 | 2387 | 1159 | (null) | 5671 (null) | (null) | (null) | 400 |
| 11 | 2388 | 1159 | (null) | 6095 (null) | (null) | (null) | 480 |
| 12 | 2389 | 1159 | (null) | 6153 (null) | (null) | (null) | 550 |
| 13 | 2390 | 1162 | (null) | 5873 (null) | (null) | (null) | 350 |
| 14 | 2391 | 1163 | (null) | 5933 (null) | (null) | (null) | 320 |
| 15 | 2392 | 1163 | (null) | 6034 (null) | (null) | (null) | 575 |
| 16 | 2393 | 1164 | (null) | 5699 (null) | (null) | (null) | 525 |
| 17 | 2394 | 1165 | (null) | 5234 (null) | (null) | (null) | 300 |
| 18 | 2395 | 1165 | (null) | 5235 (null) | (null) | (null) | 295 |

Error 1: agent_person_id 6002 is present in rent but not in agent table

SQL code for error detection:

```

311
312 --Checking for foreign key validity and presence of that key value in the parent table
313 -- agent_person_id 6002 is present in rent but not in agent table
314 select * from MonRE.rent where agent_person_id NOT IN ( select person_id from monre.agent);
315

```

Error detected:

| | 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 2: client_person_id 6001 is present in rent but not in client table

SQL code for error detection:

```

314 select * from MonRE.rent where agent_person_id NOT IN ( select person_id from monre.agent);
315
316 -- client_person_id 6001 is present in rent but not in client table
317 select * from MonRE.rent where client_person_id NOT IN ( select person_id from monre.client);
318
319

```

Error detected:

| | 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 3: Checking if start_date is before the end_date (1 discrepancy detected)

SQL code for error detection:

```

321
322 --Checking if start_date is before the end_date. 1 discrepancy detected
323 SELECT * FROM monre.rent WHERE rent_start_date > rent_end_date;
324 --Checking if there is only 1 entry with the combination of agent and client ids
325 SELECT * FROM monre.rent WHERE agent_person_id ='6002' and client_person_id ='6001';
326

```

Error detected:

| | 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 4: Checking if there is only 1 entry with the combination of agent and client ids

SQL code for error detection:

```

322 --Checking if start_date is before the end_date. 1 discrepancy detected
323 SELECT * FROM monre.rent WHERE rent_start_date > rent_end_date;
324 --Checking if there is only 1 entry with the combination of agent and client ids
325 SELECT * FROM monre.rent WHERE agent_person_id ='6002' and client_person_id ='6001';
326

```

Error detected:

| | 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 |

Count (*) before

| | COUNT(*) |
|---|----------|
| 1 | 3284 |

14.2 Rent table **after**:

| | RENT_ID | AGENT_PERSON_ID | CLIENT_PERSON_ID | PROPERTY_ID | RENT_START_DATE | RENT_END_DATE | PRICE |
|----|---------|-----------------|------------------|-------------|-----------------|---------------|-------|
| 1 | 1 | 104 | 3383 | 3049 | 27/APR/20 | 13/OCT/20 | 495 |
| 2 | 2 | 106 | 3384 | 2948 | 03/MAY/20 | 19/OCT/20 | 195 |
| 3 | 3 | 106 | 3385 | 3076 | 03/MAR/20 | 18/AUG/20 | 265 |
| 4 | 4 | 106 | 3386 | 3096 | 20/APR/20 | 06/OCT/20 | 265 |
| 5 | 5 | 107 | 3387 | 2954 | 21/APR/20 | 07/OCT/20 | 370 |
| 6 | 6 | 107 | 3388 | 3003 | 16/JAN/20 | 02/JUL/20 | 335 |
| 7 | 7 | 107 | 3389 | 3117 | 31/MAR/20 | 15/SEP/20 | 430 |
| 8 | 8 | 184 | 3390 | 3934 | 04/APR/20 | 19/SEP/20 | 500 |
| 9 | 9 | 185 | 3391 | 4419 | 18/JAN/20 | 04/JUL/20 | 630 |
| 10 | 10 | 185 | 3392 | 3973 | 29/APR/20 | 15/OCT/20 | 1100 |
| 11 | 11 | 186 | 3393 | 3885 | 12/JAN/20 | 28/JUN/20 | 650 |
| 12 | 12 | 186 | 3394 | 3895 | 17/MAR/20 | 01/SEP/20 | 320 |
| 13 | 13 | 186 | 3395 | 3943 | 30/APR/20 | 16/OCT/20 | 625 |
| 14 | 14 | 186 | 3396 | 4131 | 02/FEB/20 | 19/JUL/20 | 595 |
| 15 | 15 | 186 | 3397 | 3957 | 13/FEB/20 | 30/JUL/20 | 380 |
| 16 | 16 | 186 | 3398 | 4400 | 27/APR/20 | 13/OCT/20 | 540 |
| 17 | 17 | 188 | 3399 | 4127 | 24/APR/20 | 10/OCT/20 | 575 |
| 18 | 18 | 190 | 3400 | 4245 | 23/APR/20 | 09/OCT/20 | 600 |

Count (*) after:

| | COUNT(*) |
|---|----------|
| 1 | 3283 |

15.1 Sale table before:

| | SALE_ID | AGENT_PERSON_ID | CLIENT_PERSON_ID | SALE_DATE | PROPERTY_ID | PRICE |
|----|---------|-----------------|------------------|-----------|-------------|---------|
| 1 | 434 | 1077 | 2900 | 22/MAR/20 | 1964 | 1395000 |
| 2 | 435 | 1079 | 2901 | 16/JAN/20 | 1896 | 275000 |
| 3 | 436 | 1083 | 2902 | 20/MAR/20 | 1932 | 3490000 |
| 4 | 437 | 1084 | 2903 | 14/JAN/20 | 1998 | 799000 |
| 5 | 438 | 1087 | 2904 | 22/FEB/20 | 1943 | 2000000 |
| 6 | 439 | 1195 | 2905 | 29/JAN/20 | 5 | 1825000 |
| 7 | 440 | 1195 | 2906 | 14/MAR/20 | 67 | 380000 |
| 8 | 441 | 1195 | 2907 | 05/APR/20 | 72 | 1695000 |
| 9 | 442 | 1202 | 2908 | 19/JAN/20 | 121 | 495000 |
| 10 | 443 | 1202 | 2909 | 09/MAR/20 | 229 | 565000 |
| 11 | 444 | 1202 | 2910 | 09/MAR/20 | 159 | 545000 |
| 12 | 445 | 1202 | 2911 | 19/FEB/20 | 164 | 280000 |
| 13 | 446 | 1202 | 2912 | 01/MAR/20 | 202 | 520000 |
| 14 | 447 | 1202 | 2913 | 25/FEB/20 | 305 | 440000 |
| 15 | 448 | 1203 | 2914 | 02/MAR/20 | 217 | 470000 |
| 16 | 449 | 1203 | 2915 | 24/JAN/20 | 173 | 565000 |
| 17 | 450 | 1204 | 2916 | 25/JAN/20 | 241 | 390000 |
| 18 | 451 | 1207 | 2917 | 28/JAN/20 | 181 | 950000 |

Null value in client_Person_id and sale_date

| | SALE_ID | AGENT_PERSON_ID | CLIENT_PERSON_ID | SALE_DATE | PROPERTY_ID | PRICE |
|----|---------|-----------------|------------------|-----------|-------------|---------|
| 1 | 917 | 1 | (null) | (null) | 21 | 800000 |
| 2 | 918 | 1 | (null) | (null) | 90 | 665000 |
| 3 | 919 | 1 | (null) | (null) | 91 | 649000 |
| 4 | 920 | 1 | (null) | (null) | 96 | 545000 |
| 5 | 921 | 2 | (null) | (null) | 51 | 539000 |
| 6 | 922 | 2 | (null) | (null) | 54 | 529000 |
| 7 | 923 | 2 | (null) | (null) | 55 | 495000 |
| 8 | 924 | 2 | (null) | (null) | 3 | 450000 |
| 9 | 925 | 2 | (null) | (null) | 6 | 1650000 |
| 10 | 926 | 2 | (null) | (null) | 10 | 1175000 |
| 11 | 927 | 2 | (null) | (null) | 40 | 589000 |
| 12 | 928 | 2 | (null) | (null) | 47 | 559000 |
| 13 | 929 | 2 | (null) | (null) | 63 | 435000 |
| 14 | 930 | 2 | (null) | (null) | 74 | 1195000 |
| 15 | 931 | 2 | (null) | (null) | 75 | 1150000 |
| 16 | 932 | 2 | (null) | (null) | 85 | 726000 |
| 17 | 933 | 2 | (null) | (null) | 88 | 699000 |

15.2 Sale table after:

| | SALE_ID | AGENT_PERSON_ID | CLIENT_PERSON_ID | SALE_DATE | PROPERTY_ID | PRICE |
|----|---------|-----------------|------------------|----------------|-------------|---------|
| 1 | 434 | 1077 | | 2900 22/MAR/20 | 1964 | 1395000 |
| 2 | 435 | 1079 | | 2901 16/JAN/20 | 1896 | 275000 |
| 3 | 436 | 1083 | | 2902 20/MAR/20 | 1932 | 3490000 |
| 4 | 437 | 1084 | | 2903 14/JAN/20 | 1998 | 799000 |
| 5 | 438 | 1087 | | 2904 22/FEB/20 | 1943 | 2000000 |
| 6 | 439 | 1195 | | 2905 29/JAN/20 | 5 | 1825000 |
| 7 | 440 | 1195 | | 2906 14/MAR/20 | 67 | 380000 |
| 8 | 441 | 1195 | | 2907 05/APR/20 | 72 | 1695000 |
| 9 | 442 | 1202 | | 2908 19/JAN/20 | 121 | 495000 |
| 10 | 443 | 1202 | | 2909 09/MAR/20 | 229 | 565000 |
| 11 | 444 | 1202 | | 2910 09/MAR/20 | 159 | 545000 |
| 12 | 445 | 1202 | | 2911 19/FEB/20 | 164 | 280000 |
| 13 | 446 | 1202 | | 2912 01/MAR/20 | 202 | 520000 |
| 14 | 447 | 1202 | | 2913 25/FEB/20 | 305 | 440000 |
| 15 | 448 | 1203 | | 2914 02/MAR/20 | 217 | 470000 |
| 16 | 449 | 1203 | | 2915 24/JAN/20 | 173 | 565000 |
| 17 | 450 | 1204 | | 2916 25/JAN/20 | 241 | 390000 |
| 18 | 451 | 1207 | | 2917 28/JAN/20 | 181 | 950000 |

16.1 State table before:

| | 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 1: Null values in state code

SQL Code for error detection:

```

362  --Checking for Null values in attributes
363  select * from MONRE.state where state_code is null;
364  select * from monre.state where state_name is null;
365

```

Error detected:

| | STATE_CODE | STATE_NAME |
|---|------------|------------|
| 1 | (null) | Unknown |

Count(*) before:

| | COUNT(*) |
|---|----------|
| 1 | 9 |

16.2 State table **after**:

| 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 |

Count (*) after

| COUNT(*) |
|----------|
| 8 |

17.1 Visit table **before**:

| CLIENT_PERSON_ID | AGENT_PERSON_ID | PROPERTY_ID | VISIT_DATE | DURATION |
|------------------|-----------------|-------------|------------|----------|
| 1 5500 | 241 | 5741 | 13/APR/20 | 20 |
| 2 5568 | 241 | 5741 | 13/APR/20 | 20 |
| 3 5403 | 242 | 6102 | 13/APR/20 | 15 |
| 4 5520 | 242 | 6102 | 13/APR/20 | 15 |
| 5 5508 | 248 | 5585 | 26/MAR/20 | 15 |
| 6 5525 | 250 | 6206 | 14/APR/20 | 20 |
| 7 5529 | 252 | 5776 | 23/MAR/20 | 20 |
| 8 5399 | 253 | 5411 | 29/MAR/20 | 20 |
| 9 5462 | 253 | 5411 | 29/MAR/20 | 20 |
| 10 5474 | 253 | 5411 | 29/MAR/20 | 20 |
| 11 5498 | 253 | 5411 | 29/MAR/20 | 20 |

Error 1: person_id 6000 and 6001 are not present in their agent client tables.

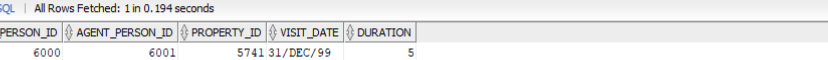
SQL code for error detection:

```

391
392 --Checking for foreign key validity and presence of that key value in the parent table
393 --Property_id 5741 has agent and client ids absent from the client and agent tables.
394 --person_id 6000 and 6001 are not present in their agent client tables.
395 select * from MonRE.visit where client_person_id NOT IN ( select person_id from monre.client);
396 select * from MonRE.visit where agent_person_id NOT IN ( select person_id from monre.agent);
397 select * from MonRE.visit where property_id NOT IN ( select property_id from monre.property);
398 select to_char(visit_date , 'yyyy') as visit_date from monre.visit where client_person_id = '6000';
399

```

Error detected:



The screenshot shows a database query result window. The title bar indicates multiple tabs for 'Script Output', 'Query Result', and several 'Query Result 1' through 'Query Result 3' tabs. The active tab is 'Query Result 1'. The status bar at the top shows 'All Rows Fetched: 1 in 0.194 seconds'. The query results are displayed in a table with the following columns: CLIENT_PERSON_ID, AGENT_PERSON_ID, PROPERTY_ID, VISIT_DATE, and DURATION. The first row of data shows values: 1, 6000, 6001, 5741.31/DEC/99, and 5.

| CLIENT_PERSON_ID | AGENT_PERSON_ID | PROPERTY_ID | VISIT_DATE | DURATION |
|------------------|-----------------|-------------|----------------|----------|
| 1 | 6000 | 6001 | 5741.31/DEC/99 | 5 |

Error 2: Property_id 5741 has agent and client ids absent from the client and agent tables.

| VISIT_DATE | |
|------------|--|
| 1 2599 | |

Count (*) before:

17.2 Visit table after:

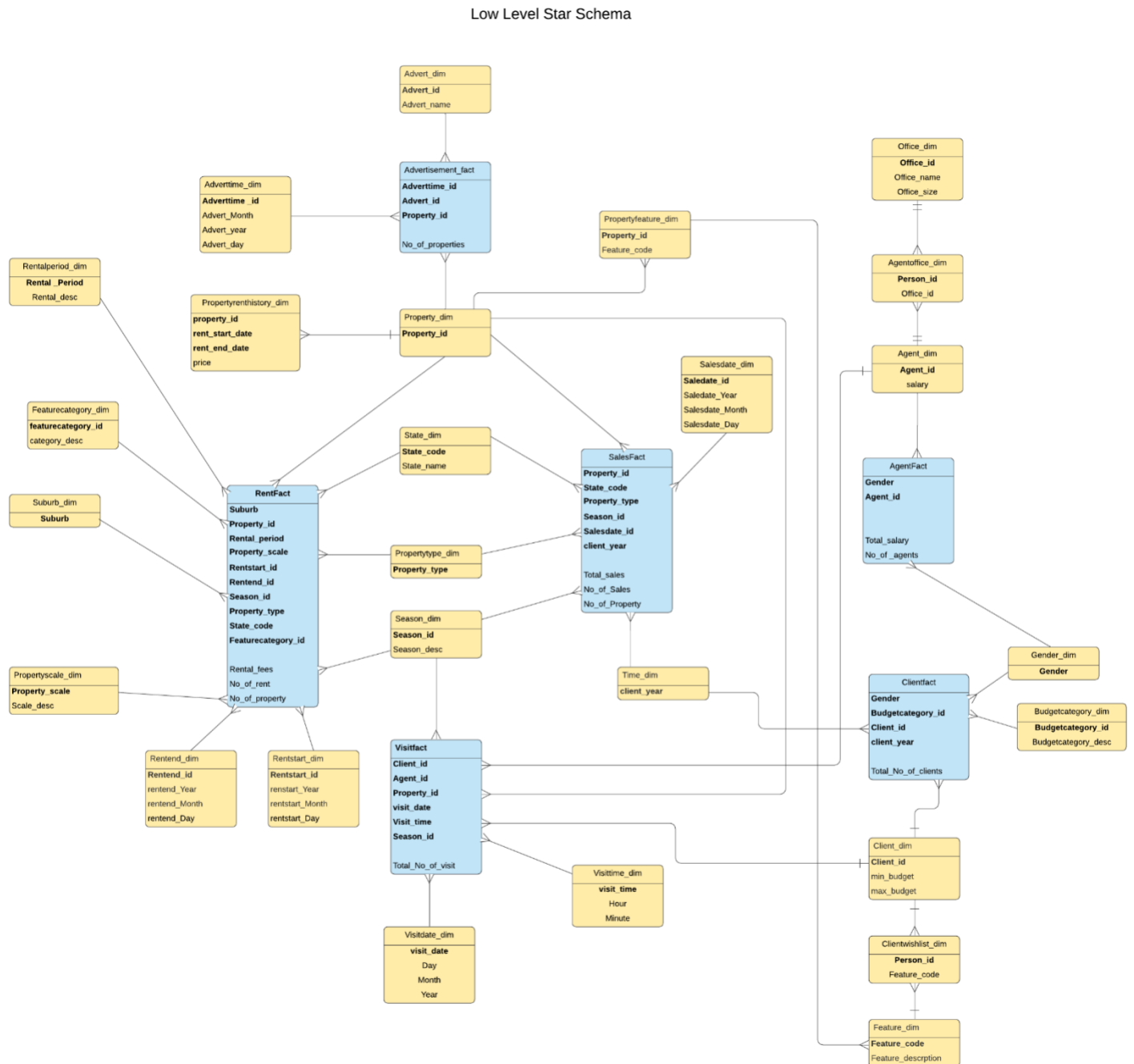
| | CLIENT_PERSON_ID | AGENT_PERSON_ID | PROPERTY_ID | VISIT_DATE | DURATION |
|----|------------------|-----------------|-------------|------------|----------|
| 1 | 5500 | 241 | 5741 | 13/APR/20 | 20 |
| 2 | 5568 | 241 | 5741 | 13/APR/20 | 20 |
| 3 | 5403 | 242 | 6102 | 13/APR/20 | 15 |
| 4 | 5520 | 242 | 6102 | 13/APR/20 | 15 |
| 5 | 5508 | 248 | 5585 | 26/MAR/20 | 15 |
| 6 | 5525 | 250 | 6206 | 14/APR/20 | 20 |
| 7 | 5529 | 252 | 5776 | 23/MAR/20 | 20 |
| 8 | 5399 | 253 | 5411 | 29/MAR/20 | 20 |
| 9 | 5462 | 253 | 5411 | 29/MAR/20 | 20 |
| 10 | 5474 | 253 | 5411 | 29/MAR/20 | 20 |
| 11 | 5498 | 253 | 5411 | 29/MAR/20 | 20 |
| 12 | 5585 | 255 | 5411 | 29/MAR/20 | 20 |

Count(*) after:

| | COUNT(*) |
|---|----------|
| 1 | 574 |

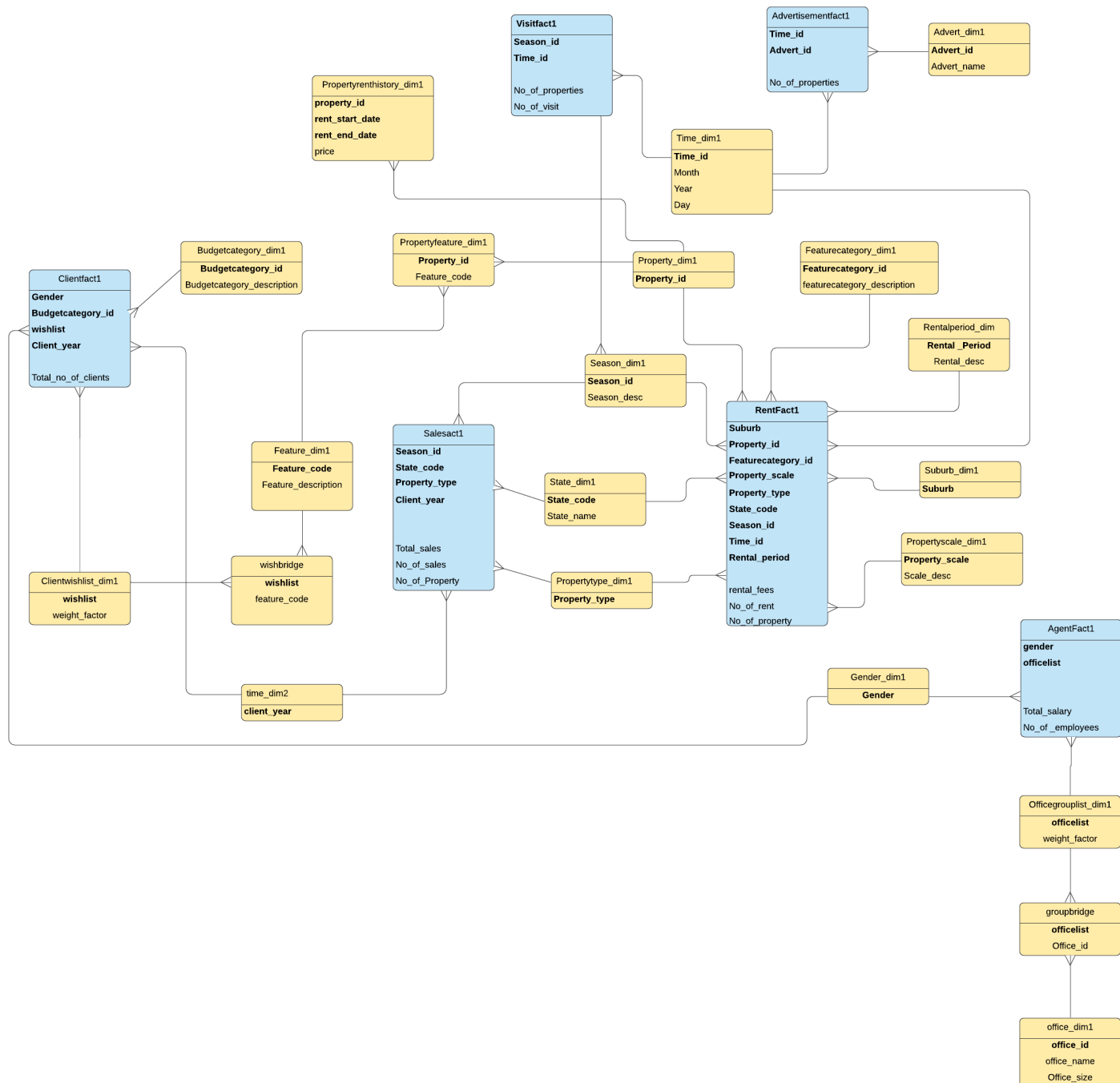
c) Two versions of star/snowflake schema diagrams

Low Level Star Schema:



High Level Star Schema:

High Level Start Schema



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

Ans:

In designing both the star schema, we have chosen the non-hierarchy dimension.

Reasons for doing so are:

Only one join is needed to join the dimension and the fact table.

In hierarchy, efficient query processing is required to query the information

Also, hierarchy does not offer roll up and drill down features

Identifier of parent dimension already covers the child dimension.

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

Ans:

We have chosen SCD type 4 by adding history table for property rent called the propertyrenthistory_dim. We have used this because, one table keeps the current data and the additional table we added keeps the record of all changes in the property rent if any. In our scenario the original table name is property and the history table are propertyhistory_dim.

Another reason for choosing it is that SCD type 0: Retaining original, SCD type 1: Overwrite, SCD type 2: Adding a new row, SCD type 3: Adding a new attribute will not work in our case. So, we can say that SCD type 4 can be used in our case for low level and high-level star schema.

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

Level 0-star schema is very detailed and almost like that of ERD. In our star schema, in case of client the granularity is on Client_id and there will be one value for each client so the granularity will be high, and aggregation will be low.

Level 2-star schema is less detailed. In this same case of client in level 2 the granularity is on client_wish and there can be more than one common wish_list between two clients or more so the granularity is low and so the aggregation is high.

Task 2

(a) SQL statements (e.g. create table, insert into, etc) to create the star/snowflake schema Version-1

```
--Creating dimension tables for making rentfact table
-- Creating rentalperiod_dim table
drop table rentalperiod_dim cascade constraints purge;
create table rentalperiod_dim (rental_period varchar(20),
```

```

rental_desc varchar(50));

insert into rentalperiod_dim values ('Short','< 6 months');
insert into rentalperiod_dim values ('Medium','6-12 months');
insert into rentalperiod_dim values ('Big','> 12 months');

--Creating suburb_dim table
drop table suburb_dim cascade constraints purge;
create table suburb_dim as
select distinct suburb from address;

--Creating property_dim table
drop table property_dim CASCADE constraints purge;
create table property_dim as
select property_id from property;

--Creating Propertyfeature_dim
drop table propertyfeature_dim cascade constraints purge;
create table propertyfeature_dim as
select * from property_feature;

--Creating slowly changing dimension propertyrenthistory_dim
drop table propertyrenthistory_dim cascade constraints purge;
create table propertyrenthistory_dim as
select property_id, rent_start_date, rent_end_date, price from rent;

--Creating state_dim table
drop table state_dim cascade constraints purge;
create table state_dim as
select distinct * from state;

--Creating Propertytype_dim table
drop table propertytype_dim cascade constraints purge;
create table propertytype_dim as
select distinct property_type from property;
--select * from propertytype_dim;
--select * from suburb_dim;

--Creating Featurecategory_dim table
drop table featurecategory_dim cascade constraints;
create table featurecategory_dim (featurecategory_id Varchar(15),
category_desc varchar(15));

```

```

insert into featurecategory_dim values ('Basic','< 10 features');
insert into featurecategory_dim values ('Standard','10-20 features');
insert into featurecategory_dim values ('Luxurious','> 20 features');

```

```
--Creating Propertyscale_dim table
```

```

drop table propertyscale_dim cascade constraints purge;
create table propertyscale_dim (property_scale Varchar(15),
scale_desc varchar(15));

```

```

insert into propertyscale_dim values ('Extra Small','<= 1 Bedroom');
insert into propertyscale_dim values ('Small','2-3 Bedrooms');
insert into propertyscale_dim values ('Medium','3-6 Bedrooms');
insert into propertyscale_dim values ('Large','6-10 Bedrooms');
insert into propertyscale_dim values ('Extra Large','> 10 Bedrooms');

```

```
--Creating Rentstart_dim table
```

```

drop table rentstart_dim cascade constraints purge;
create table rentstart_dim as
select distinct to_char(rent_start_date,'yyyyMMDD') as rentstart_id,
to_char(rent_start_date,'yyyy') as rentstart_year,
to_char(rent_start_date,'MM') as rentstart_month,
to_char(rent_start_date,'dd') as rentstart_date from rent where rent_start_date is not null;
--select * from rentstart_dim;

```

```
--Creating Rentend_dim table
```

```

drop table rentend_dim cascade constraints purge;
create table rentend_dim as
select distinct to_char(rent_end_date,'yyyyMon') as rentend_id,
to_char(rent_end_date,'yyyy') as rentend_year,
to_char(rent_end_date,'Mon') as rentend_month,
to_char(rent_end_date,'dd') as rentend_date from rent where rent_end_date is not null;

```

```
--select * from rentend_dim;
```

```
--Creating season_dim table
```

```

drop table SEASON_DIM cascade constraints purge;
CREATE TABLE SEASON_DIM
(
SEASON_ID VARCHAR(10),
Season_desc VARCHAR(50)
);
INSERT INTO SEASON_DIM VALUES ( 'AUTUMN', 'MAR-MAY' );

```

```

INSERT INTO SEASON_DIM VALUES ( 'SUMMER', 'DEC-FEB' );
INSERT INTO SEASON_DIM VALUES ( 'SPRING', 'SEP-NOV' );
INSERT INTO SEASON_DIM VALUES ( 'WINTER', 'JUN-AUG' );

```

```
--Select * from season_dim;
```

```
--purge recyclebin;
```

```
--Creating renttempfact table
```

```
drop table renttempfact cascade constraints purge;
```

```
create table renttempfact as
```

```
select a.suburb,
```

```
re.property_id,
```

```
p.property_type,
```

```
re.rent_start_date,
```

```
re.rent_end_date,
```

```
count(distinct(pf.feature_code)) as no_of_features,
```

```
to_char(re.rent_start_date,'yyyymmdd') as rentstart_id,
```

```
to_char(re.rent_end_date,'yyyymmdd') as rentend_id,
```

```
po.state_code,
```

```
p.property_no_of_bedrooms,
```

```
sum(((re.rent_end_date - re.rent_start_date)/7)*(re.price)) as Total_rent,
```

```
count(distinct(re.rent_id)) as No_of_rent,
```

```
count(distinct(re.property_id)) as No_of_property
```

```
from address a, property p, rent re, postcode po, property_feature pf
```

```
where a.address_id = p.address_id and
```

```
re.property_id = p.property_id(+) and
```

```
p.property_id = pf.property_id(+) and
```

```
po.postcode = a.postcode and rent_start_date is not null
```

```
group by a.suburb,
```

```
re.property_id,
```

```
re.rent_id,
```

```
p.property_type,
```

```
re.rent_start_date,
```

```
re.rent_end_date,
```

```
po.state_code,
```

```
p.property_no_of_bedrooms,
```

```
to_char(re.rent_start_date,'yyyymmdd'),
```

```
to_char(re.rent_end_date,'yyyymmdd');
```

```
--select count(*) from renttempfact;
```

```
--select * from renttempfact where property_id =3000;
```

--Altering and updating renttempfact table

```
ALTER TABLE renttempfact
ADD (
rental_period VARCHAR2(20),
season_id VARCHAR2(20),
property_scale varchar(15),
featurecategory_id varchar(15),
rental_difference number(4),
rental_fees number(10)
);
```

--Finding out the months for which properties were rented using months_between function.

```
update renttempfact set
rental_difference = months_between (rent_end_date, rent_start_date);
```

-- dividing rent by total features

```
update renttempfact set
rental_fees = total_rent/no_of_features
where no_of_features > 0;
```

```
update renttempfact set
rental_fees = total_rent
where no_of_features = 0;
```

--updating featurecategory_id table

```
update renttempfact set
featurecategory_id = 'Basic'
where no_of_features < 10;
```

```
update renttempfact set
featurecategory_id = 'Standard'
where no_of_features >= 10 and no_of_features <= 20;
```

```
update renttempfact set
featurecategory_id = 'Luxurious'
where no_of_features > 20;
```

-- UPDATING RENTAL_PERIOD

```
update renttempfact set
rental_period = 'Short'
where rental_difference <= 6;
```

```
update renttempfact set
```

```
rental_period = 'Medium'
where rental_difference > 6 and rental_difference < 12;
```

```
update renttempfact set
rental_period = 'Big'
where rental_difference >= 12 ;
```

```
-- UPDATING SEASON_ID
UPDATE renttempfact SET
SEASON_ID= 'WINTER'
WHERE TO_CHAR(rent_start_date,'MM') >= '06'
AND TO_CHAR(rent_start_date,'MM') <= '08';
```

```
UPDATE renttempfact SET
SEASON_ID='SPRING'
WHERE TO_CHAR(RENT_START_DATE,'MM') >= '09'
AND TO_CHAR(RENT_START_DATE,'MM') <= '11';
```

```
UPDATE renttempfact SET
SEASON_ID='AUTUMN'
WHERE TO_CHAR(RENT_START_DATE,'MM') >= '03'
AND TO_CHAR(RENT_START_DATE,'MM') <= '04';
```

```
UPDATE renttempfact SET
SEASON_ID = 'SUMMER'
WHERE SEASON_ID IS NULL;
```

```
-- UPDATING PROPERTY_SCALE
update renttempfact set
property_scale = 'Extra Small'
where property_no_of_bedrooms <= 1;
```

```
update renttempfact set
property_scale = 'Small'
where property_no_of_bedrooms >= 2 and property_no_of_bedrooms < 3;
```

```
update renttempfact set
property_scale = 'Medium'
where property_no_of_bedrooms >= 3 and property_no_of_bedrooms < 6;
```

```
update renttempfact set
property_scale = 'Large'
where property_no_of_bedrooms >= 6 and property_no_of_bedrooms < 10;
```

```

update renttempfact set
property_scale = 'Extra Large'
where property_no_of_bedrooms >= 10;

```

```

--Creating rentfact table
drop table rentfact cascade constraints purge;
create table rentfact as
select suburb,
property_id,
rental_period,
property_scale,
featurecategory_id,
rentstart_id,
rentend_id,
property_type,
state_code,
season_id,
rental_fees,
No_of_rent,
No_of_property
from renttempfact;

```

```

--select * from rentfact where property_id = 3000;
--select * from rent where property_id = 2965;
--select sum(no_of_property) from rentfact;
--select count(*) from rent where rent_start_date is not null;
--Both counts are equal hence rentfact is accurate
--
--
--
--END OF RENT FACT TABLE
--
--

```

```

--Creating salesdate_dim
drop table salesdate_dim cascade constraints purge;
create table salesdate_dim as
select distinct to_char(sale_date,'yyyymm') as saledate_id,
to_char(sale_date,'yyyy') as saledate_year,
to_char(sale_date,'mm') as saledate_month,
to_char(sale_date,'dd') as saledate_day from sale;

```



```
--Creating salestempfact table
drop table salestempfact cascade constraints purge;
create table salestempfact as
select p.property_id,
p.property_type,
pa.state_code,
to_char(s.sale_date,'yyyy') as client_year,
s.sale_date,
to_char(s.sale_date,'yyyymm') as saledate_id,
sum(s.price) as Total_sales,
count(s.sale_id) as No_of_sale,
count(s.property_id) as No_of_property
from property p, sale s, postcode pa, address a
where p.property_id = s.property_id and
a.postcode=pa.postcode and sale_date is not null and
a.address_id = p.address_id
group by p.property_id, p.property_type, pa.state_code, s.sale_date,
to_char(s.sale_date,'yyyy'),
to_char(s.sale_date,'yyyymm');
```

```
--select count(*) from salestempfact;
```

```
--Altering and updating season_id in salestempfact table
```

```
ALTER TABLE salestempfact
```

```
ADD
```

```
season_id VARCHAR2(20);
```

```
-- UPDATING SEASON_ID in salestempfact
```

```
UPDATE salestempfact SET
```

```
SEASON_ID= 'WINTER'
```

```
WHERE TO_CHAR(sale_date,'MM') >= '06'
```

```
AND TO_CHAR(sale_date,'MM') <= '08';
```

```
UPDATE salestempfact SET
```

```
SEASON_ID='SPRING'
```

```
WHERE TO_CHAR(sale_DATE,'MM') >= '09'
```

```
AND TO_CHAR(sale_DATE,'MM') <= '11';
```

```
UPDATE salestempfact SET
```

```
SEASON_ID='AUTUMN'
```

```
WHERE TO_CHAR(sale_DATE,'MM') >= '03'
```

```
AND TO_CHAR(sale_DATE,'MM') <= '04';
```

```
UPDATE salestempfact SET
SEASON_ID = 'SUMMER'
WHERE SEASON_ID IS NULL;
```

```
--select count(*) from salestempfact;
```

```
--Creating salesfact from salestempfact
drop table salesfact cascade constraints purge;
create table salesfact as
select property_id,
property_type,
state_code,
saledate_id,
client_year,
season_id,
Total_sales,
No_of_sale,
No_of_property from salestempfact;
```

```
--select * from salesfact;
--select count(*) from salesfact;
--select count(*) from sale where sale_date is not null;
--Both the counts are saem hence sale fact is accurate
```

```
--
--
```

```
--END OF SALES FACT TABLE
```

```
--
--
```

```
--select client_person_id as client_id, agent_person_id as agent_id, property_id from visit;
```

```
--Creating visitdate_dim
drop table visitdate_dim cascade constraints purge;
create table visitdate_dim as
select distinct to_char(visit_date, 'yyyymmday') as visit_date,
to_char(visit_date, 'yyyy') as Year,
to_char(visit_date, 'Mon') as Month,
to_char(visit_date, 'day') as Day
from visit;
```

```
--Creating visittime_dim
drop table visittime_dim cascade constraints purge;
create table visittime_dim as
select to_char(visit_date, 'hh:mi') as visit_time,
```

```
to_char(visit_date,'hh') as hour,
to_char(visit_date,'mi') as Minute
from visit;
```

```
--Creating client_dim table
drop table client_dim cascade constraints purge;
create table client_dim as
select person_id as client_id,
min_budget, max_budget from client;
```

```
--Creating agent_dim table
drop table agent_dim cascade constraints purge;
create table agent_dim as
select person_id as agent_id,
salary from agent;
```

```
--Creating visittempfact table
drop table visittempfact cascade constraints purge;
create table visittempfact as
select to_char(visit_date,'yyyymonday') as visit_date1,
visit_date,
to_char(visit_date,'hh:mi') as visit_time,
property_id,
agent_person_id as agent_id,
client_person_id as client_id,
count(property_id) as Total_no_of_visit
from visit
where visit_date is not null
group by client_person_id, agent_person_id, property_id, visit_date;
```

```
--Altering visittempfact table for season_id
ALTER TABLE visittempfact
ADD
season_id VARCHAR2(20);
```

```
--select * from visittempfact;
```

```
-- UPDATING SEASON_ID in visittempfact
UPDATE visittempfact SET
SEASON_ID= 'WINTER'
WHERE to_char(visit_date,'MM') >= '06'
AND TO_CHAR(visit_date,'MM') <='08';
```

```
UPDATE visittempfact SET
SEASON_ID='SPRING'
WHERE TO_CHAR(visit_DATE,'MM') >= '09'
AND TO_CHAR(visit_DATE,'MM') <= '11';
```

```
UPDATE visittempfact SET
SEASON_ID='AUTUMN'
WHERE TO_CHAR(visit_DATE,'MM') >= '03'
AND TO_CHAR(visit_DATE,'MM') <= '04';
```

```
UPDATE visittempfact SET
SEASON_ID = 'SUMMER'
WHERE SEASON_ID IS NULL;
```

```
--select count(*) from visittempfact;
```

```
--Creating visitfact table
drop table visitfact cascade constraints purge;
create table visitfact as
select client_id, agent_id,
property_id,
visit_date,
visit_time,
season_id,
total_no_of_visit
from visittempfact;
```

```
--select count(*) from visitfact;
--select count(*) from visit where visit_date is not null;
--Both the counts are the same hence visit fact is accurate
--
--END OF VISIT FACT
--
--
```

```
--Gender Dim--
drop table gender_dim cascade constraints purge;
create table gender_dim as
select distinct gender from person;
```

```
--Clientwishlist Dim--
```

```

drop table clientwishlist_dim cascade constraints purge;
create table clientwishlist_dim as
select * from client_wish;

--Creating feature_dim
Drop table feature_dim cascade constraints purge;
create table feature_dim as select distinct feature_code, feature_description from feature;

--Creating budgetcategory_dim table
Drop table budgetcategory_dim cascade constraints purge;
create table budgetcategory_dim (budgetcategory_id varchar(20), budgetcategory_desc
varchar(50));

insert into budgetcategory_dim values ('Low','0 to 1000');
insert into budgetcategory_dim values ('Medium','1001 to 100000');
insert into budgetcategory_dim values ('High','100001 to 10000000');

--Creating time_dim table with respect to sale_date
drop table time_dim cascade constraints purge;
create table time_dim as
select distinct to_char(sale_date,'yyyy') as client_year from sale where sale_date is not null;

--Select * from time_dim;

-- Creating Clienttempfact table --
Drop table clienttempfact cascade constraints purge;
Create table clienttempfact as select
p.gender,
c.person_id as client_id,
c.max_budget,
to_char(s.sale_date,'yyyy') as client_year,
count(c.person_id) as total_no_of_clients
from person p, client c, sale s
where c.person_id=p.person_id(+) and c.person_id = s.client_person_id(+)
group by p.gender, c.person_id, c.max_budget, to_char(s.sale_date,'yyyy')
union
select
p.gender,
c.person_id as client_id,
c.max_budget,
to_char(r.rent_start_date,'yyyy') as client_year,
count(distinct(c.person_id)) as total_no_of_clients
from person p, client c, rent r

```

```

where c.person_id = p.person_id(+) and c.person_id = r.client_person_id(+) and p.gender is
not null and r.rent_start_date is not null
group by p.gender, c.person_id, c.max_budget, to_char(r.rent_start_date,'yyyy')
union
select
p.gender,
c.person_id as client_id,
c.max_budget,
to_char(v.visit_date,'yyyy') as client_year,
count(distinct(c.person_id)) as total_no_of_clients
from person p, client c, visit v
where c.person_id=p.person_id(+) and c.person_id = v.client_person_id (+) and p.gender is
not null and v.visit_date is not null
group by p.gender, c.person_id, c.max_budget, to_char(v.visit_date,'yyyy');

```

--After and update clienttempfact table

```
Alter table clienttempfact add (budgetcategory_id VARCHAR(10));
```

-- update table clienttempfact to input budgetcategory_id

```
update clienttempfact set budgetcategory_id='Low' where max_budget>0 and max_budget<=
1000;
```

```
update clienttempfact set budgetcategory_id='Medium' where max_budget>1001 and
max_budget<= 100000;
```

```
update clienttempfact set budgetcategory_id='High' where max_budget>100001 and
max_budget<= 10000000;
```

```
--select count(*) from clienttempfact;
```

--Creating clientfact table from clienttempfact

```
drop table clientfact cascade constraints purge;
```

```
create table clientfact as select budgetcategory_id, client_year, gender, client_id,
total_no_of_clients from clienttempfact;
```

```
--select * from clientfact;
```

```
--select count(*) from clientfact;
```

--Both counts are the same hence clientfact is accurate

```
--
```

```
--
```

```
-- END OF CLIENT FACT TABLE
```

```
--
```

```
--
```

--Creating Adverttime_dim table

```
drop table adverttime_dim cascade constraints purge;
```

```
create table adverttime_dim as
select distinct to_char(property_date_added,'yyyymmdd') as adverttime_id,
to_char(property_date_added,'dd') as advert_day,
to_char(property_date_added,'mm') as advert_month,
to_char(property_date_added,'yyyy') as advert_year from property;
```

```
--Create Advert_dim table
drop table advert_dim cascade constraints purge;
create table advert_dim as
select * from advertisement;
```

```
--Creating final advertisement_fact table
drop table advertisement_fact cascade constraints purge;
create table advertisement_fact as
select to_char(property_date_added,'YYYYMMDD') as adverttime_id, pa.property_id,
a.advert_id,
count(distinct(pa.property_id)) as No_of_properties
from property p, property_advert pa, advertisement a
where p.property_id = pa.property_id and a.advert_id = pa.advert_id(+)
group by pa.property_id, to_char(property_date_added,'YYYYMMDD'),
a.advert_id;
```

```
--select count(distinct(property_id)) from advertisement_fact;
--select sum(no_of_properties) from advertisement_fact;
--select count(distinct(property_id)) from property_advert;
--select count(*) from advertisement_fact;
--select count(*) from property_advert;
```

```
--
--
-- END OF ADVERTISEMENT_FACT TABLE
```

```
--
--
```

```
--Creating agentoffice_dim table
drop table agentoffice_dim cascade constraints purge;
create table agentoffice_dim as
select * from agent_office;
```

```
--Creating tempoffice_dim table
drop table tempoffice_dim cascade constraints purge;
create table tempoffice_dim as
select o.office_id, o.office_name, count(ao.person_id) as no_of_employees
from office o, agent_office ao
where o.office_id = ao.office_id
```

```

group by o.office_id,o.office_name,ao.person_id;

alter table tempoffice_dim
add (office_size Varchar(20));

update tempoffice_dim
set office_size = 'Small'
where no_of_employees < 4;

update tempoffice_dim
set office_size = 'Medium'
where no_of_employees >= 4 and no_of_employees <=12;

update tempoffice_dim
set office_size = 'Big'
where no_of_employees > 12;

--select count(*) from tempoffice_dim;
--select * from tempoffice_dim;

--Creating office_dim from tempoffice_dim
drop table office_dim cascade constraints purge;
create table office_dim as
select office_id, office_name, office_size from tempoffice_dim;

--Creating agent_fact table
drop table agent_fact cascade constraints purge;
create table agent_fact as
select p.gender,
(a.person_id) as agent_id,
sum(a.salary) as total_salary,
count(distinct(a.person_id)) as no_of_agents
from agent a, person p
where a.person_id = p.person_id
group by p.gender, a.person_id;

--select sum(no_of_agents) from agent_fact;
--select count(*) from agent;

--
--
-- END OF AGENT FACT
--

```



```
--
commit;
/*
select * from agent_fact;

select * from clientfact;

select count(*) from agent;

select * from rentfact;

select * from salesfact;

select * from advertisement_fact;

select count(*) from property_advert;

select * from visitfact;
select count(*) from visit;

*/
```

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

```
--Creating Feature_dim1 table
drop table feature_dim1 cascade constraints purge;
create table feature_dim1 as
select distinct feature_code, feature_description from feature group by feature_code,
feature_description;

--Creating Season_dim1 table
drop table season_dim1 cascade constraints purge;
CREATE TABLE SEASON_DIM1
(
SEASON_ID VARCHAR(10),
Season_desc VARCHAR(50)
);

--Inserting Initial values into season_dim1
INSERT INTO SEASON_DIM1 VALUES ( 'AUTUMN', 'MAR-MAY' );
```

```

INSERT INTO SEASON_DIM1 VALUES ( 'SUMMER', 'DEC-FEB' );
INSERT INTO SEASON_DIM1 VALUES ( 'SPRING', 'SEP-NOV' );
INSERT INTO SEASON_DIM1 VALUES ( 'WINTER', 'JUN-AUG' );

```

```

--Creating property_dim1 from temporary dimension table for property
drop table property_dim1 cascade constraints purge;
create table property_dim1 as
select property_id from property;

```

```

--Creating state_dim1 table
drop table state_dim1 cascade constraints purge;
create table state_dim1 as
select distinct * from state;

```

```

--Creating Propertytype_dim1 table
drop table propertytype_dim1 cascade constraints purge;
create table propertytype_dim1 as
select distinct property_type from property;

```

```

--Creating Propertyfeature_dim1
drop table propertyfeature_dim1 cascade constraints purge;
create table propertyfeature_dim1 as
select * from property_feature;

```

```

--Creating time_dim2 table with respect to sale_date
drop table time_dim2 cascade constraints purge;
create table time_dim2 as
select distinct to_char(sale_date,'yyyy') as client_year from sale where sale_date is not null;

```

```

--creating salestempfact1 table
drop table salestempfact1 cascade constraints purge;
create table salestempfact1 as
select distinct s.property_id,
pa.state_code,
p.property_type,
to_char(s.sale_date,'yyyy') as client_year,
s.sale_date,
sum(s.price) as Total_sales,
count(distinct(s.sale_id)) as No_of_sale,
count(distinct(s.property_id)) as No_of_property
from property p, sale s, postcode pa, address a
where s.property_id = p.property_id(+) and --property_ids present in the sale table is needed
in the salesfact

```

```

a.postcode=pa.postcode and
a.address_id = p.address_id and s.sale_date is not null
group by s.property_id, p.property_type, pa.state_code, s.sale_date,
to_char(s.sale_date,'yyyy');

```

```
--select * from salestempfact1;
```

```
--Altering salestempfact1 table to input season_id
```

```
ALTER TABLE salestempfact1
```

```
ADD
```

```
season_id VARCHAR2(20);
```

```
-- UPDATING SEASON_ID in salestempfact1 according to sale_date
```

```
UPDATE salestempfact1 SET
```

```
SEASON_ID= 'WINTER'
```

```
WHERE TO_CHAR(sale_date,'MM') >= '06'
```

```
AND TO_CHAR(sale_date,'MM') <= '08';
```

```
UPDATE salestempfact1 SET
```

```
SEASON_ID='SPRING'
```

```
WHERE TO_CHAR(sale_DATE,'MM') >= '09'
```

```
AND TO_CHAR(sale_DATE,'MM') <= '11';
```

```
UPDATE salestempfact1 SET
```

```
SEASON_ID='AUTUMN'
```

```
WHERE TO_CHAR(sale_DATE,'MM') >= '03'
```

```
AND TO_CHAR(sale_DATE,'MM') <= '04';
```

```
UPDATE salestempfact1 SET
```

```
SEASON_ID = 'SUMMER'
```

```
WHERE SEASON_ID IS NULL;
```

```
--Creating salesfact1 from salestempfact1
```

```
drop table salesfact1 cascade constraints purge;
```

```
create table salesfact1 as
```

```
select property_id,
```

```
property_type,
```

```
client_year,
```

```
state_code,
```

```

season_id,
sum(Total_sales) as total_sales,
sum(No_of_sale) as no_of_sale,
sum(No_of_property) as no_of_property from salestempfact1
group by
property_id, property_type, state_code, season_id, client_year;

```

--Checking for correctness of the fact table with the Operational database

```
/*
```

```

select * from salesfact1;
select sum(total_sales) from salesfact1;
select sum(price) from sale where sale_date is not null;
select sum(no_of_property) from salesfact1;
select sum(no_of_sale) from salesfact1;
select count(property_id) from sale where sale_date is not null;
select count(no_of_property) from salesfact1;
select count(*) from sale;

```

```
*/
```

```
--
```

```
--
```

```
-- END OF SALES FACT TABLE
```

```
--
```

```
--
```

--Creating time_dim1 using union to add 5 different dates in one time dimension

drop table time_dim1 cascade constraints purge;

create table time_dim1 as

```

select distinct to_char(property_date_added,'yyyymmdd') as time_id,
to_char(property_date_added,'yyyy') as year,
to_char(property_date_added,'mm') as month,
to_char(property_date_added,'dd') as day from property
union

```

```

select distinct to_char(visit_date,'yyyymmdd') as time_id,
to_char(visit_date,'yyyy') as year,
to_char(visit_date,'mm') as month,
to_char(visit_date,'dd') as day from visit
union

```

```

select distinct to_char(sale_date,'yyyymmdd') as time_id,
to_char(sale_date,'yyyy') as year,
to_char(sale_date,'mm') as month,
to_char(sale_date,'dd') as day from sale
union

```

```

select distinct to_char(rent_start_date,'yyyymmdd') as time_id,

```

```

to_char(rent_start_date,'yyyy') as year,
to_char(rent_start_date,'mm') as month,
to_char(rent_start_date,'dd') as day from rent
union
select distinct to_char(rent_end_date,'yyyymmdd') as time_id,
to_char(rent_end_date,'yyyy') as year,
to_char(rent_end_date,'mm') as month,
to_char(rent_end_date,'dd') as day from rent;

```

```

--Create Advert_dim1 table
drop table advert_dim1 cascade constraints purge;
create table advert_dim1 as
select * from advertisement;

```

```

--Creating advertisementfact1 table
drop table advertisementfact1 cascade constraints purge;
create table advertisementfact1 as
select a.advert_id, to_char(p.property_date_added,'YYYYMMDD') as time_id,
count(pa.property_id) as no_of_properties
from property p, property_advert pa, advertisement a
where p.property_id = pa.property_id and a.advert_id = pa.advert_id(+) --requires all
advert_ids from the advertisement table hence the left outer join
group by a.advert_id, to_char(p.property_date_added,'YYYYMMDD');

```

```

--Checking for correctness of the fact table with the Operational database
--select sum(no_of_properties) from advertisementfact1;
--select * from advertisementfact1;
--select count(*) from property_advert;
--
--
-- END OF ADVERTISEMENT_FACT TABLE
--
--

```

```

-- Creating rentalperiod_dim1 table
drop table rentalperiod_dim1 cascade constraints;
create table rentalperiod_dim1 (rental_period varchar(20),
rental_desc varchar(50));

```

```

--Inserting initial dimention values and their description

```

```

insert into rentalperiod_dim1 values ('Short','< 6 months');
insert into rentalperiod_dim1 values ('Medium','6-12 months');
insert into rentalperiod_dim1 values ('Big','> 12 months');

```

```

--Creating suburb_dim1 table
drop table suburb_dim1 cascade constraints purge;
create table suburb_dim1 as
select distinct suburb from address;

--Creating slowly changing dimension propertyrenthistory_dim1
drop table propertyrenthistory_dim1 cascade constraints purge;
create table propertyrenthistory_dim1 as
select property_id, rent_start_date, rent_end_date, price from rent;

--Creating Propertyscale_dim1 table
drop table propertyscale_dim1 cascade constraints purge;
create table propertyscale_dim1 (property_scale Varchar(15),
scale_desc varchar(15));

--Inserting initial property_scale values and their description
insert into propertyscale_dim1 values ('Extra Small','<= 1 Bedroom');
insert into propertyscale_dim1 values ('Small','2-3 Bedrooms');
insert into propertyscale_dim1 values ('Medium','3-6 Bedrooms');
insert into propertyscale_dim1 values ('Large','6-10 Bedrooms');
insert into propertyscale_dim1 values ('Extra Large','> 10 Bedrooms');

--Creating Featurecategory_dim table
drop table featurecategory_dim1 cascade constraints;
create table featurecategory_dim1 (featurecategory_id Varchar(15),
category_desc varchar(15));

--Inserting initial categories and their description as given
insert into featurecategory_dim1 values ('Basic','< 10 features');
insert into featurecategory_dim1 values ('Standard','10-20 features');
insert into featurecategory_dim1 values ('Luxurious','> 20 features');

--Creating Rentstart_dim1 table
/*
create table rentstart_dim1 as
select distinct to_char(rent_start_date,'yyyMondd') as rentstart_id,
to_char(rent_start_date,'yyyy') as rentstart_year,
to_char(rent_start_date,'Mon') as rentstart_month,
to_char(rent_start_date,'dd') as rentstart_date from rent;
*/

--purge recyclebin;

```

```

--Creating renttempfact1 table
drop table renttempfact1 cascade constraints purge;
create table renttempfact1 as
select a.suburb,
re.property_id,
p.property_type,
re.rent_start_date,
re.rent_end_date,
count(distinct(pf.feature_code)) as no_of_features,
po.state_code,
p.property_no_of_bedrooms,
to_char(re.rent_start_date,'yyyymmdd') as time_id,
sum(((re.rent_end_date - re.rent_start_date)/7)*(re.price)) as Total_rent,
count(distinct(re.rent_id)) as No_of_rent,
count(distinct(re.property_id)) as No_of_property
from address a, property p, rent re, postcode po, property_feature pf
where a.address_id = p.address_id and
re.property_id = p.property_id(+) and -- we need all properties from rent table hence the left
outer join
p.property_id = pf.property_id(+) and -- all property_id values should be from the property
table to checking for features
po.postcode = a.postcode and rent_start_date is not null
group by a.suburb, re.property_id, re.rent_id, p.property_type, re.rent_start_date,
re.rent_end_date, po.state_code, p.property_no_of_bedrooms,
to_char(re.rent_start_date,'yyyymmdd'), re.rent_start_date,
'yyyymmdd', to_char(re.rent_end_date,'yyyymmdd'), re.rent_end_date, 'yyyymmdd',
to_char(re.rent_start_date,'yyyymmdd');

--Select count(*) from renttempfact1;

--Altering and updating renttempfact1 table
ALTER TABLE renttempfact1
ADD (
rental_period VARCHAR2(20),
season_id VARCHAR2(20),
property_scale varchar(15),
featurecategory_id varchar(15),
rental_difference number(4),
rental_fees number(10)
);

```

--Finding out the months for which properties were rented using months_between function.

```
update renttempfact1 set
rental_difference = months_between (rent_end_date, rent_start_date);
```

```
-- Dividing rent by total features
update renttempfact1 set
rental_fees = total_rent/no_of_features
where no_of_features > 0;
```

```
update renttempfact1 set
rental_fees = total_rent
where no_of_features = 0;
```

```
--updating featurecategory_id table
update renttempfact1 set
featurecategory_id = 'Basic'
where no_of_features < 10;
```

```
update renttempfact1 set
featurecategory_id = 'Standard'
where no_of_features >= 10 and no_of_features <= 20;
```

```
update renttempfact1 set
featurecategory_id = 'Luxurious'
where no_of_features > 20;
```

```
-- UPDATING RENTAL_PERIOD based on the rental_difference
update renttempfact1 set
rental_period = 'Short'
where rental_difference <= 6;
```

```
update renttempfact1 set
rental_period = 'Medium'
where rental_difference > 6 and rental_difference < 12;
```

```
update renttempfact1 set
rental_period = 'Big'
where rental_difference >= 12 ;
```



```
-- UPDATING SEASON_ID based on rent_start_date
UPDATE renttempfact1 SET
SEASON_ID= 'WINTER'
WHERE TO_CHAR(rent_start_date,'MM') >= '06'
AND TO_CHAR(rent_start_date,'MM') <= '08';
```

```
UPDATE renttempfact1 SET
SEASON_ID='SPRING'
WHERE TO_CHAR(RENT_START_DATE,'MM') >= '09'
AND TO_CHAR(RENT_START_DATE,'MM') <= '11';
```

```
UPDATE renttempfact1 SET
SEASON_ID='AUTUMN'
WHERE TO_CHAR(RENT_START_DATE,'MM') >= '03'
AND TO_CHAR(RENT_START_DATE,'MM') <= '04';
```

```
UPDATE renttempfact1 SET
SEASON_ID = 'SUMMER'
WHERE SEASON_ID IS NULL;
```

```
-- UPDATING PROPERTY_SCALE based on property_no_of_bedrooms
```

```
update renttempfact1 set
property_scale = 'Extra Small'
where property_no_of_bedrooms <= 1;
```

```
update renttempfact1 set
property_scale = 'Small'
where property_no_of_bedrooms >= 2 and property_no_of_bedrooms < 3;
```

```
update renttempfact1 set
property_scale = 'Medium'
where property_no_of_bedrooms >= 3 and property_no_of_bedrooms < 6;
```

```
update renttempfact1 set
property_scale = 'Large'
where property_no_of_bedrooms >= 6 and property_no_of_bedrooms < 10;
```

```
update renttempfact1 set
```

```
property_scale = 'Extra Large'
where property_no_of_bedrooms >= 10;
```

```
--Creating rentfact table from renttempfact1 table
```

```
drop table rentfact1 cascade constraints purge;
```

```
create table rentfact1 as
```

```
select suburb,
```

```
property_id,
```

```
featurecategory_id,
```

```
rental_period,
```

```
property_scale,
```

```
time_id,
```

```
property_type,
```

```
state_code,
```

```
season_id,
```

```
rental_fees,
```

```
No_of_rent,
```

```
No_of_property
```

```
from renttempfact1;
```

```
--select * from rentfact1;
```

```
--
```

```
--
```

```
--END OF RENT FACT
```

```
--
```

```
--
```

```
/*
```

```
select * from rentfact1;
```

```
select * from rent order by property_id;
```

```
select count(*) from rentfact1;
```

```
select * from rent order by property_id;
```

```
select * from rent where rent_start_date is null;
```

```
select sum(rental_fees) from rentfact1;
```

```
select sum(price) from rent where rent_start_date is not null;
```

```
select count(property_id) from rent where rent_start_date is not null;
```

```
select count(no_of_property) from rentfact1;
```

```
select count(no_of_rent) from rentfact1;
```

```
select count(*) from rent where rent_start_date is not null;
```

```
*/
```

```
/*
```

```

--Creating visitdate_dim1
create table visitdate_dim1 as
select to_char(visit_date, 'yyyymmdd') as visit_date,
to_char(visit_date,'yyyy') as Year,
to_char(visit_date,'mm') as Month,
to_char(visit_date,'dd') as Day
from visit;
*/

--Creating temporary visitfact table called visittempfact
drop table visittempfact1 cascade constraints purge;
create table visittempfact1 as
select
visit_date,
to_char(visit_date,'YYYYMONDAY') as time_id,
count(distinct(property_id)) as no_of_properties,
count(*) as Total_no_of_visit
from visit
where visit_date is not null
group by
visit_date,
to_char(visit_date,'YYYYMONDAY')
;
--select * from visittempfact1;

--Alter visittempfact1 to insert season_id
ALTER TABLE visittempfact1
ADD
season_id VARCHAR2(20);

--select * from visittempfact1;

-- UPDATING SEASON_ID in visittempfact1 based on visit_date
UPDATE visittempfact1 SET
SEASON_ID= 'WINTER'
WHERE to_char(visit_date,'MM') >= '06'
AND TO_CHAR(visit_date,'MM') <='08';

UPDATE visittempfact1 SET
SEASON_ID='SPRING'
WHERE TO_CHAR(visit_DATE,'MM') >= '09'
AND TO_CHAR(visit_DATE,'MM') <= '11';

```

```
UPDATE visittempfact1 SET
SEASON_ID='AUTUMN'
WHERE TO_CHAR(visit_DATE,'MM') >= '03'
AND TO_CHAR(visit_DATE,'MM') <= '04';
```

```
UPDATE visittempfact1 SET
SEASON_ID = 'SUMMER'
WHERE SEASON_ID IS NULL;
```

```
--select count(*) from visittempfact1;
```

```
--Creating visitfact1 table from visittempfact1
drop table visitfact1 cascade constraints purge;
```

```
create table visitfact1 as
select
time_id,
season_id,
count(no_of_properties) as no_of_properties,
sum(Total_no_of_visit) as total_no_of_visits
from visittempfact1
group by time_id, season_id;
```

```
--
--
```

```
--END OF VISIT FACT
```

```
--
--
```

```
/*
select count(distinct(property_id)) from visit;
select * from visitfact1;
select count(*) from visitfact1;
select count(*) from visit;
select sum(no_of_properties) from visitfact1;
select sum(total_no_of_visits) from visitfact1;
*/
```

```
--Creating tempoffice_dim1 table
drop table tempoffice_dim1 cascade constraints purge;
create table tempoffice_dim1 as
select distinct o.office_id, o.office_name, count (distinct (ao.person_id)) as no_of_employees
from office o, agent_office ao
where o.office_id = ao.office_id
```

```
group by o.office_id, o.office_name;
```

```
--Altering tempoffice_dim1 to add office_size as an attribute
alter table tempoffice_dim1 add
(office_size varchar(20));
```

```
--updating office_size attribute in tempoffice dimension table based on no_of_employees
update tempoffice_dim1
set office_size = 'Small'
where no_of_employees < 4;
```

```
update tempoffice_dim1
set office_size = 'Medium'
where no_of_employees >= 4 and no_of_employees <=12;
```

```
update tempoffice_dim1
set office_size = 'Big'
where no_of_employees > 12;
```

```
--select * from tempoffice_dim1;
```

```
--Creating office_dim1 from tempoffice_dim1 table
drop table office_dim1 cascade constraints purge;
create table office_dim1 as
select office_id, office_name, office_size from
tempoffice_dim1;
```

```
--Creating Gender_Dim1 table
drop table gender_dim1 cascade constraints purge;
create table gender_dim1 as select distinct gender from person;
```

```
--Creating agent_dim1 table
drop table agent_dim1 cascade constraints purge;
create table agent_dim1 as
select distinct a.person_id,salary, 1/count(*) as weight_factor,
listagg (ao.office_id,'_') within group (order by ao.office_id) as officelist --to increase level of
aggregation
from agent a, agent_office ao
where a.person_id = ao.person_id(+)
group by a.person_id,salary;
```

```
--select * from agent_dim1;
--select * from agent_office where person_id = 6;
```

```

--Creating officegrouplist_dim1 table
drop table officegrouplist_dim1 cascade constraints purge;
create table officegrouplist_dim1 as
select distinct officelist, weight_factor
from agent_dim1;

--select * from officegrouplist_dim1;

--Creating groupbridge table
drop table groupbridge cascade constraints purge;
create table groupbridge as
select distinct officeList, ao.office_ID
from agent_dim1 a, agent_office ao
where a.person_id = ao.person_id;
--select * from groupbridge;

--Creating agenttempfact1
drop table agenttempfact1 cascade constraints purge;
Create Table agentTempFact1 As
Select p.gender, a.person_id, a.salary,
count(a.person_id) As no_of_agents
From person p, agent a
Where p.person_ID = a.person_ID
group by p.gender, a.person_id, a.salary;

--Creating agenttempfact2 from agenttempfact1
drop table agenttempfact2 cascade constraints purge;
create table agentTempFact2
as select *
from agentTempFact1;

--Altering agenttempfact2 to add officelist
alter table agentTempFact2
add (officeList varchar(100));

--Updating value of officelist using listagg function
update agentTempFact2 T
set officeList = (
select listagg(ao.office_id, '_') within group -- to increase level of aggregation
(order by ao.office_id) as officeList
from agent_office ao
where t.person_id = ao.person_id

```

```

);
--select * from agenttempfact2;

--Creating agentfact1 from agenttempfact2
drop table agentfact1 cascade constraints purge;
create table agentFact1
as select
  gender, officeList,
  sum(no_of_agents) as No_of_agents
from agentTempFact2
group by gender, officeList;

--select * from agentfact1 order by officelist;
--select * from agent_office where office_id = 1;
--select * from agentfact1;
--
--
-- END OF AGENT FACT TABLE
--
--
-- Creating clientdim1 table, which is a temporary dimension to add weightfactor and wishlist
DROP TABLE client_dim1 cascade constraints purge;
CREATE TABLE client_dim1 AS SELECT
listagg (cw.feature_code, '_') within group (order by cw.feature_code) as WishList, --Higher
level of aggregation
c.person_id,c.min_budget,c.max_budget, 1.0/count(*) as weight_factor
FROM client c, client_wish cw
where c.person_id = cw.person_id(+)
group by c.person_id,c.min_budget,c.max_budget;

--select * from clientwishlist_dim1 where wishlist is not null;

--Creating clientwishlist_dim1 table from client_dim1
drop table clientwishlist_dim1 cascade constraints purge;
create table clientwishlist_DIM1 as select DISTINCT wishlist, weight_factor FROM
client_dim1;

--Creating budgetcategory_dim1 table
drop table budgetcategory_dim1 cascade constraints purge;
create table budgetcategory_dim1 (budgetcategory_id varchar(15),
budgetcategory_description varchar(50));

--Inserting initial budgetcategory_id values in dimension table

```

```

insert into budgetcategory_dim1 values ('Low','0 to 1000');
insert into budgetcategory_dim1 values ('Medium','1001 to 100000');
insert into budgetcategory_dim1 values ('High','100001 to 10000000');

```

```

--Creating wishBridge table
drop table wishbridge cascade constraints purge;
CREATE TABLE wishbridge AS SELECT DISTINCT
wishlist,
feature_code
FROM client_dim1 cd, client_wish cw
WHERE cd.person_id = cw.person_id;

```

```

--SELECT * FROM wishbridge;

```

```

--Creating clienttempfact1 table
DROP TABLE clienttempfact1 cascade constraints purge;
CREATE TABLE clienttempfact1 AS SELECT
p.gender, c.person_id, c.max_budget,
to_char(s.sale_date, 'yyyy') as client_year,
COUNT(distinct(c.person_id)) AS no_of_clients
FROM client c, person p, sale s
WHERE c.person_id = p.person_id (+) and p.person_id = s.client_person_id(+) and gender
is not null and sale_date is not null
GROUP BY p.gender, c.person_id, c.max_budget, to_char(s.sale_date, 'yyyy')
UNION
SELECT
p.gender, c.person_id, c.max_budget,
to_char(s.rent_start_date, 'yyyy') as client_year,
COUNT(distinct(c.person_id)) AS no_of_clients
FROM client c, person p, rent s
WHERE c.person_id = p.person_id (+) and p.person_id = s.client_person_id(+) and gender
is not null and rent_start_date is not null
GROUP BY p.gender, c.person_id, c.max_budget, to_char(s.rent_start_date, 'yyyy')
UNION
SELECT
p.gender, c.person_id, c.max_budget,
to_char(s.visit_date, 'yyyy') as client_year,
COUNT(distinct(c.person_id)) AS no_of_clients
FROM client c, person p, visit s
WHERE c.person_id = p.person_id (+) and p.person_id = s.client_person_id(+) and gender
is not null and visit_date is not null
GROUP BY p.gender, c.person_id, c.max_budget, to_char(s.visit_date, 'yyyy');

```



```

--SELECT * FROM clienttempfact1;

--Altering clienttempfact1 to add budgetcategory_id
ALTER TABLE clienttempfact1 ADD (budgetcategory_id VARCHAR2(10));

--Updating budgetcategory_id based on max_budet from client table
UPDATE clienttempfact1
SET budgetcategory_id = 'Low'
WHERE max_budget <= 1000;

UPDATE clienttempfact1
SET budgetcategory_id = 'Medium'
WHERE max_budget >= 1001 AND max_budget <= 100000;

UPDATE clienttempfact1
SET budgetcategory_id = 'High'
WHERE max_budget >= 100001;

--select * from clienttempfact1;
--Creating clienttempfact2 from clienttempfact1
drop table clienttempfact2 cascade constraints purge;
CREATE TABLE clienttempfact2 AS
SELECT distinct person_id, gender, max_budget, client_year, budgetcategory_id,
no_of_clients
FROM clienttempfact1 where client_year is not null;

--SELECT * FROM clienttempfact2;
--SELECT sum(no_of_clients) from clienttempfact2;

--Adding wishlist to clienttempfact2
ALTER TABLE clienttempfact2 ADD (wishlist VARCHAR(200));

--updating wishlist values using feature_code
UPDATE clienttempfact2 c
SET wishlist = ( SELECT LISTAGG(cw.feature_code, '_') WITHIN GROUP (ORDER BY
cw.feature_code) AS wishlist
FROM client_wish cw WHERE c.person_id = cw.person_id (+));

--select sum(no_of_clients) from clienttempfact2;

--Creating clientfact1 from clienttempfact2
drop table clientfact1 cascade constraints purge;

```

```
CREATE TABLE clientfact1 AS
SELECT gender, client_year, budgetcategory_id, wishlist,
SUM(no_of_clients) AS total_no_of_clients
FROM clienttempfact2
GROUP BY gender, budgetcategory_id, wishlist, client_year;
```

```
/*
select * from clientfact1;
select sum(total_no_of_clients) from clientfact1;
--select sum(total_no_of_clients) from clientfact1;
select count(distinct (person_id)) from client_wish;
*/
commit;
```

c) Screenshots of the tables that you have created; this includes the contents of each table that you have created.

Low level Tables:

rentalperiod_dim

| RENTAL_PERIOD | RENTAL_DESC |
|---------------|-------------|
| 1 Short | < 6 months |
| 2 Medium | 6-12 months |
| 3 Big | > 12 months |

Property_dim

| | PROPERTY_ID |
|----|-------------|
| 1 | 28 |
| 2 | 36 |
| 3 | 46 |
| 4 | 129 |
| 5 | 131 |
| 6 | 135 |
| 7 | 139 |
| 8 | 49 |
| 9 | 53 |
| 10 | 60 |
| 11 | 4 |
| 12 | 5 |
| 13 | 8 |
| 14 | 19 |
| 15 | 233 |
| 16 | 241 |
| 17 | 114 |

Propertyfeature_dim

| | PROPERTY_ID | FEATURE_CODE |
|----|-------------|--------------|
| 1 | 9 | 1 |
| 2 | 9 | 2 |
| 3 | 9 | 3 |
| 4 | 9 | 4 |
| 5 | 9 | 5 |
| 6 | 9 | 6 |
| 7 | 9 | 7 |
| 8 | 9 | 8 |
| 9 | 9 | 9 |
| 10 | 9 | 10 |
| 11 | 9 | 11 |
| 12 | 9 | 12 |
| 13 | 9 | 117 |
| 14 | 11 | 1 |
| 15 | 11 | 2 |
| 16 | 11 | 5 |
| 17 | 11 | 6 |
| 18 | 11 | 7 |

Propertyrenthistory_dim

| | PROPERTY_ID | RENT_START_DATE | RENT_END_DATE | PRICE |
|----|-------------|-----------------|---------------|-------|
| 1 | 3049 | 27/APR/20 | 13/OCT/20 | 495 |
| 2 | 2948 | 03/MAY/20 | 19/OCT/20 | 195 |
| 3 | 3076 | 03/MAR/20 | 18/ADG/20 | 265 |
| 4 | 3096 | 20/APR/20 | 06/OCT/20 | 265 |
| 5 | 2954 | 21/APR/20 | 07/OCT/20 | 370 |
| 6 | 3003 | 16/JAN/20 | 02/JUL/20 | 335 |
| 7 | 3117 | 31/MAR/20 | 15/SEP/20 | 430 |
| 8 | 3934 | 04/APR/20 | 19/SEP/20 | 500 |
| 9 | 4419 | 18/JAN/20 | 04/JUL/20 | 630 |
| 10 | 3973 | 29/APR/20 | 15/OCT/20 | 1100 |
| 11 | 3885 | 12/JAN/20 | 28/JUN/20 | 650 |
| 12 | 3895 | 17/MAR/20 | 01/SEP/20 | 320 |
| 13 | 3943 | 30/APR/20 | 16/OCT/20 | 625 |
| 14 | 4131 | 02/FEB/20 | 19/JUL/20 | 595 |
| 15 | 3957 | 13/FEB/20 | 30/JUL/20 | 380 |
| 16 | 4400 | 27/APR/20 | 13/OCT/20 | 540 |
| 17 | 4127 | 24/APR/20 | 10/OCT/20 | 575 |
| 18 | 4245 | 23/APR/20 | 09/OCT/20 | 600 |

State_dim

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

Propertytype_dim

| 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 |

Propertyscale_dim

| PROPERTY_SCALE | SCALE_DESC |
|----------------|---------------|
| 1 Extra Small | <= 1 Bedroom |
| 2 Small | 2-3 Bedrooms |
| 3 Medium | 3-6 Bedrooms |
| 4 Large | 6-10 Bedrooms |
| 5 Extra Large | > 10 Bedrooms |

Rentstart_dim

| RENTSTART_ID | RENTSTART_YEAR | RENTSTART_MONTH | RENTSTART_DATE |
|--------------|----------------|-----------------|----------------|
| 1 | 2020Apr | 2020 | Apr 23 |
| 2 | 2020Feb | 2020 | Feb 09 |
| 3 | 2020May | 2020 | May 02 |
| 4 | 2020Feb | 2020 | Feb 21 |
| 5 | 2020Jan | 2020 | Jan 23 |
| 6 | 2020Apr | 2020 | Apr 14 |
| 7 | 2020Mar | 2020 | Mar 23 |
| 8 | 2020Jan | 2020 | Jan 21 |
| 9 | 2020Jan | 2020 | Jan 08 |
| 10 | 2020Mar | 2020 | Mar 20 |
| 11 | 2020Feb | 2020 | Feb 18 |
| 12 | 2020Apr | 2020 | Apr 02 |
| 13 | 2020Mar | 2020 | Mar 01 |
| 14 | 2020Mar | 2020 | Mar 02 |
| 15 | 2020Apr | 2020 | Apr 27 |
| 16 | 2020Apr | 2020 | Apr 20 |
| 17 | 2020Jan | 2020 | Jan 16 |
| 18 | 2020Apr | 2020 | Apr 25 |

Rentend_dim

| RENTEND_ID | RENTEND_YEAR | RENTEND_MONTH | RENTEND_DATE |
|------------|--------------|---------------|--------------|
| 1 | 2020Jul | 2020 | Jul |
| 2 | 2020Jul | 2020 | Jul |
| 3 | 2020Jul | 2020 | Jul |
| 4 | 2020Jul | 2020 | Jul |
| 5 | 2020Sep | 2020 | Sep |
| 6 | 2020Aug | 2020 | Aug |
| 7 | 2020Aug | 2020 | Aug |
| 8 | 2020Jul | 2020 | Jul |
| 9 | 2020Sep | 2020 | Sep |
| 10 | 2020Oct | 2020 | Oct |
| 11 | 2020Aug | 2020 | Aug |
| 12 | 2020Sep | 2020 | Sep |
| 13 | 2020Jul | 2020 | Jul |
| 14 | 2020Aug | 2020 | Aug |
| 15 | 2020Jul | 2020 | Jul |
| 16 | 2020Oct | 2020 | Oct |
| 17 | 2020Oct | 2020 | Oct |
| 18 | 2020Aug | 2020 | Aug |

Season_dim

| SEASON_ID | SEASON_DESC |
|-----------|-------------|
| 1 | AUTUMN |
| 2 | SUMMER |
| 3 | SPRING |
| 4 | WINTER |

Featurecategory_dim

| FEATURECATEGORY_ID | CATEGORY_DESC |
|--------------------|---------------|
| 1 | Basic |
| 2 | Standard |
| 3 | Luxurious |

Rentfact

| SUBURB | PROPERTY_ID | RENTAL_PERIOD | PROPERTY_SCALE | RENTSTART_ID | RENTEND_ID | PROPERTY_TYPE | STATE_CODE | SEASON_ID | TOTAL_RE |
|--------|---------------|---------------|----------------|--------------|------------|-------------------------|------------|-----------|----------|
| 1 | Enfield | 2980 Short | Medium | 20200315 | 20200830 | House | SA | AUTUMN | 3 |
| 2 | Athelstone | 2987 Short | Medium | 20200418 | 20201004 | House | SA | AUTUMN | 4 |
| 3 | Linden Park | 3032 Short | Small | 20200223 | 20200809 | Apartment / Unit / Flat | SA | SUMMER | 3 |
| 4 | Adelaide | 3052 Short | Small | 20200221 | 20200807 | Apartment / Unit / Flat | SA | SUMMER | 5 |
| 5 | Athelstone | 3075 Short | Medium | 20200331 | 20200915 | House | SA | AUTUMN | 4 |
| 6 | Adelaide | 3102 Short | Small | 20200425 | 20201011 | Apartment / Unit / Flat | SA | AUTUMN | 5 |
| 7 | Belfield | 3729 Short | Medium | 20200418 | 20201004 | Townhouse | NSW | AUTUMN | 4 |
| 8 | Coogee | 3737 Short | Medium | 20200423 | 20201009 | Apartment / Unit / Flat | NSW | AUTUMN | 5 |
| 9 | Darlinghurst | 3746 Short | Extra Small | 20200401 | 20200916 | Apartment / Unit / Flat | NSW | AUTUMN | 5 |
| 10 | Elizabeth Bay | 3749 Short | Small | 20200206 | 20200723 | Apartment / Unit / Flat | NSW | SUMMER | 12 |
| 11 | North Sydney | 3758 Short | Extra Small | 20200223 | 20200809 | Apartment / Unit / Flat | NSW | SUMMER | 5 |
| 12 | Paddington | 3781 Short | Small | 20200410 | 20200925 | Apartment / Unit / Flat | NSW | AUTUMN | 7 |
| 13 | Surry Hills | 3794 Short | Extra Small | 20200409 | 20200924 | Apartment / Unit / Flat | NSW | AUTUMN | 5 |
| 14 | Stammore | 3795 Short | Small | 20200110 | 20200626 | Apartment / Unit / Flat | NSW | SUMMER | 5 |
| 15 | Woolloomooloo | 3816 Short | Small | 20200503 | 20201019 | Apartment / Unit / Flat | NSW | SUMMER | 6 |
| 16 | Paddington | 3847 Short | Small | 20200301 | 20200816 | House | NSW | AUTUMN | 7 |
| 17 | Vaucluse | 3861 Short | Small | 20200101 | 20200617 | Apartment / Unit / Flat | NSW | SUMMER | 6 |
| 18 | Stammore | 3863 Short | Extra Small | 20200318 | 20200902 | Apartment / Unit / Flat | NSW | AUTUMN | 4 |

| | RENTAL_PERIOD | PROPERTY_SCALE | RENTSTART_ID | RENTEND_ID | PROPERTY_TYPE | STATE_CODE | SEASON_ID | TOTAL_RENT | NO_OF_RENT | NO_OF_PROPERTY |
|----|---------------|----------------|--------------|------------|-------------------------|------------|-----------|------------|------------|----------------|
| 1 | ort | Medium | 20200315 | 20200830 | House | SA | AUTUMN | 350 | 1 | 1 |
| 2 | ort | Medium | 20200418 | 20201004 | House | SA | AUTUMN | 450 | 1 | 1 |
| 3 | ort | Small | 20200223 | 20200809 | Apartment / Unit / Flat | SA | SUMMER | 300 | 1 | 1 |
| 4 | ort | Small | 20200221 | 20200807 | Apartment / Unit / Flat | SA | SUMMER | 535 | 1 | 1 |
| 5 | ort | Medium | 20200331 | 20200915 | House | SA | AUTUMN | 425 | 1 | 1 |
| 6 | ort | Small | 20200425 | 20201011 | Apartment / Unit / Flat | SA | AUTUMN | 800 | 1 | 1 |
| 7 | ort | Medium | 20200418 | 20201004 | Townhouse | NSW | AUTUMN | 660 | 1 | 1 |
| 8 | ort | Medium | 20200423 | 20201009 | Apartment / Unit / Flat | NSW | AUTUMN | 990 | 1 | 1 |
| 9 | ort | Extra Small | 20200401 | 20200916 | Apartment / Unit / Flat | NSW | AUTUMN | 525 | 1 | 1 |
| 10 | ort | Small | 20200206 | 20200723 | Apartment / Unit / Flat | NSW | SUMMER | 1200 | 1 | 1 |
| 11 | ort | Extra Small | 20200223 | 20200809 | Apartment / Unit / Flat | NSW | SUMMER | 550 | 1 | 1 |
| 12 | ort | Small | 20200410 | 20200925 | Apartment / Unit / Flat | NSW | AUTUMN | 720 | 1 | 1 |
| 13 | ort | Extra Small | 20200409 | 20200924 | Apartment / Unit / Flat | NSW | AUTUMN | 575 | 1 | 1 |
| 14 | ort | Small | 20200110 | 20200626 | Apartment / Unit / Flat | NSW | SUMMER | 575 | 1 | 1 |
| 15 | ort | Small | 20200503 | 20201019 | Apartment / Unit / Flat | NSW | SUMMER | 850 | 1 | 1 |
| 16 | ort | Small | 20200301 | 20200816 | House | NSW | AUTUMN | 750 | 1 | 1 |
| 17 | ort | Small | 20200101 | 20200617 | Apartment / Unit / Flat | NSW | SUMMER | 850 | 1 | 1 |
| 18 | ort | Extra Small | 20200318 | 20200902 | Apartment / Unit / Flat | NSW | AUTUMN | 435 | 1 | 1 |

Salesdate_dim

| SALEDATE_ID | SALEDATE_YEAR | SALEDATE_MONTH | SALEDATE_DAY | |
|-------------|---------------|----------------|--------------|-----------|
| 1 | Jan2020 | 2020 | Jan | thursday |
| 2 | Jan2020 | 2020 | Jan | friday |
| 3 | Mar2020 | 2020 | Mar | wednesday |
| 4 | Mar2020 | 2020 | Mar | tuesday |
| 5 | Feb2020 | 2020 | Feb | friday |
| 6 | Apr2020 | 2020 | Apr | friday |
| 7 | Apr2020 | 2020 | Apr | tuesday |
| 8 | Dec2019 | 2019 | Dec | monday |
| 9 | Mar2020 | 2020 | Mar | sunday |
| 10 | Jan2020 | 2020 | Jan | sunday |
| 11 | Jan2020 | 2020 | Jan | saturday |
| 12 | Feb2020 | 2020 | Feb | monday |
| 13 | Apr2020 | 2020 | Apr | sunday |
| 14 | Feb2020 | 2020 | Feb | wednesday |
| 15 | Feb2020 | 2020 | Feb | sunday |
| 16 | Feb2020 | 2020 | Feb | thursday |
| 17 | Jan2020 | 2020 | Jan | monday |
| 18 | Apr2020 | 2020 | Apr | saturday |

Salesfact

| PROPERTY_ID | PROPERTY_TYPE | STATE_CODE | SALEDATE_ID | CLIENT_YEAR | SEASON_ID | TOTAL_SALES | NO_OF_SALE | NO_OF_PROPERTY |
|-------------|------------------------------|------------|-------------|-------------|-----------|-------------|------------|----------------|
| 1 | 550 Townhouse | QLD | 202001 | 2020 | SUMMER | 500000 | 1 | 1 |
| 2 | 556 House | QLD | 202003 | 2020 | AUTUMN | 579000 | 1 | 1 |
| 3 | 582 House | QLD | 202003 | 2020 | AUTUMN | 349000 | 1 | 1 |
| 4 | 585 House | QLD | 202002 | 2020 | SUMMER | 499000 | 1 | 1 |
| 5 | 601 Apartment / Unit / Flat | QLD | 202003 | 2020 | AUTUMN | 399000 | 1 | 1 |
| 6 | 608 House | QLD | 202001 | 2020 | SUMMER | 400000 | 1 | 1 |
| 7 | 613 Apartment / Unit / Flat | QLD | 202002 | 2020 | SUMMER | 820000 | 1 | 1 |
| 8 | 1422 Apartment / Unit / Flat | ACT | 202002 | 2020 | SUMMER | 305000 | 1 | 1 |
| 9 | 1435 Apartment / Unit / Flat | ACT | 202003 | 2020 | AUTUMN | 499000 | 1 | 1 |
| 10 | 1437 House | ACT | 202002 | 2020 | SUMMER | 570000 | 1 | 1 |
| 11 | 1456 House | ACT | 202003 | 2020 | AUTUMN | 499000 | 1 | 1 |
| 12 | 159 House | SA | 202003 | 2020 | AUTUMN | 545000 | 1 | 1 |
| 13 | 241 Apartment / Unit / Flat | SA | 202001 | 2020 | SUMMER | 390000 | 1 | 1 |
| 14 | 267 Apartment / Unit / Flat | SA | 202002 | 2020 | SUMMER | 199000 | 1 | 1 |
| 15 | 359 House | QLD | 202003 | 2020 | AUTUMN | 699000 | 1 | 1 |
| 16 | 396 Apartment / Unit / Flat | QLD | 202002 | 2020 | SUMMER | 247500 | 1 | 1 |

Visitdate_dim

Visittime_dim

| | VISIT_TIME | HOUR | MINUTE |
|----|------------|------|--------|
| 1 | 04:38 | 04 | 38 |
| 2 | 01:06 | 01 | 06 |
| 3 | 03:14 | 03 | 14 |
| 4 | 01:00 | 01 | 00 |
| 5 | 05:42 | 05 | 42 |
| 6 | 05:26 | 05 | 26 |
| 7 | 07:01 | 07 | 01 |
| 8 | 05:51 | 05 | 51 |
| 9 | 02:44 | 02 | 44 |
| 10 | 10:49 | 10 | 49 |
| 11 | 03:56 | 03 | 56 |
| 12 | 06:23 | 06 | 23 |
| 13 | 02:26 | 02 | 26 |
| 14 | 07:05 | 07 | 05 |
| 15 | 04:38 | 04 | 38 |
| 16 | 01:06 | 01 | 06 |
| 17 | 03:14 | 03 | 14 |
| 18 | 01:00 | 01 | 00 |

Client_dim

| | CLIENT_ID | MIN_BUDGET | MAX_BUDGET |
|----|-----------|------------|------------|
| 1 | 2702 | 810000 | 990000 |
| 2 | 2706 | 540000 | 660000 |
| 3 | 2707 | 426600 | 521400 |
| 4 | 2724 | 701955 | 857945 |
| 5 | 2729 | 593100 | 724900 |
| 6 | 2732 | 702000 | 858000 |
| 7 | 2737 | 332100 | 405900 |
| 8 | 2769 | 625500 | 764500 |
| 9 | 2772 | 719100 | 878900 |
| 10 | 2778 | 468000 | 572000 |
| 11 | 2780 | 314100 | 383900 |
| 12 | 2782 | 656100 | 801900 |
| 13 | 2797 | 526500 | 643500 |
| 14 | 2799 | 1795500 | 2194500 |
| 15 | 2816 | 810000 | 990000 |
| 16 | 2848 | 449100 | 548900 |
| 17 | 2850 | 1066500 | 1303500 |
| 18 | 2854 | 486000 | 594000 |

Agent_dim

| | AGENT_ID | SALARY |
|----|----------|--------|
| 1 | 1058 | 180000 |
| 2 | 362 | 175000 |
| 3 | 303 | 200000 |
| 4 | 330 | 200000 |
| 5 | 944 | 200000 |
| 6 | 495 | 200000 |
| 7 | 552 | 195000 |
| 8 | 566 | 180000 |
| 9 | 590 | 200000 |
| 10 | 1637 | 200000 |
| 11 | 1259 | 195000 |
| 12 | 698 | 195000 |
| 13 | 712 | 210000 |
| 14 | 823 | 175000 |
| 15 | 824 | 210000 |
| 16 | 860 | 210000 |
| 17 | 1439 | 190000 |
| 18 | 1446 | 195000 |

Visitfact

| | CLIENT_ID | AGENT_ID | PROPERTY_ID | VISIT_DATE | VISIT_TIME | SEASON_ID | TOTAL_NO_OF_VISIT |
|----|-----------|----------|-------------|------------|------------|-----------|-------------------|
| 1 | 5043 | 2243 | 1628 | 12/MAR/20 | 02:29 | AUTUMN | 1 |
| 2 | 5053 | 1089 | 1605 | 08/MAR/20 | 12:46 | AUTUMN | 1 |
| 3 | 5060 | 795 | 1631 | 24/MAR/20 | 01:06 | AUTUMN | 1 |
| 4 | 5071 | 1071 | 1494 | 13/MAR/20 | 03:14 | AUTUMN | 1 |
| 5 | 5079 | 1667 | 1623 | 25/MAR/20 | 08:35 | AUTUMN | 1 |
| 6 | 5092 | 496 | 1567 | 06/MAR/20 | 12:18 | AUTUMN | 1 |
| 7 | 5092 | 1089 | 1605 | 08/MAR/20 | 12:46 | AUTUMN | 1 |
| 8 | 5092 | 2004 | 1625 | 29/MAR/20 | 02:07 | AUTUMN | 1 |
| 9 | 5092 | 2341 | 1597 | 24/MAR/20 | 05:29 | AUTUMN | 1 |
| 10 | 5133 | 712 | 1642 | 15/MAR/20 | 04:25 | AUTUMN | 1 |
| 11 | 5139 | 2245 | 1382 | 18/MAR/20 | 06:23 | AUTUMN | 1 |
| 12 | 5165 | 782 | 1765 | 17/MAR/20 | 07:04 | AUTUMN | 1 |
| 13 | 5165 | 2023 | 1868 | 13/MAR/20 | 03:42 | AUTUMN | 1 |
| 14 | 5186 | 1665 | 1897 | 08/MAR/20 | 09:32 | AUTUMN | 1 |
| 15 | 5206 | 1683 | 2151 | 29/MAR/20 | 09:19 | AUTUMN | 1 |
| 16 | 5232 | 2331 | 1976 | 28/MAR/20 | 11:25 | AUTUMN | 1 |
| 17 | 5240 | 1088 | 1947 | 28/MAR/20 | 04:58 | AUTUMN | 1 |
| 18 | 5242 | 1077 | 2156 | 08/MAR/20 | 09:41 | AUTUMN | 1 |

Gender_dim

| GENDER |
|----------|
| 1 Male |
| 2 Female |

Clientwishlist_dim

| 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 |

Feature_dim

| FEATURE_CODE | FEATURE_DESCRIPTION |
|--------------|-------------------------------|
| 1 | 4 City Views |
| 2 | 5 Close to schools |
| 3 | 6 Close to shops |
| 4 | 23 Balcony |
| 5 | 34 Ducted Cooling |
| 6 | 35 Ducted Vacuum System |
| 7 | 36 Open Fireplace |
| 8 | 51 Split System Heating |
| 9 | 58 Garden Sheds |
| 10 | 67 Ocean Views |
| 11 | 73 Internal Laundry |
| 12 | 87 Broadband |
| 13 | 89 Workshop |
| 14 | 90 Deck with views |
| 15 | 111 Water Front |
| 16 | 114 Modern Bathroom |
| 17 | 130 Outdoor Entertaining Area |
| 18 | 145 Kitchenette |

Budgetcategory_dim

| BUDGETCATEGORY_ID | BUDGETCATEGORY_DESC |
|-------------------|---------------------|
| 1 Low | 0 to 1000 |
| 2 Medium | 1001 to 100000 |
| 3 High | 100001 to 10000000 |

Time_dim

| SALE_YEAR |
|-----------|
| 1 2019 |
| 2 2020 |

Clientfact

| BUDGETCATEGORY_ID | CLIENT_YEAR | GENDER | CLIENT_ID | TOTAL_NO_OF_CLIENTS |
|-------------------|-------------|--------|-----------|---------------------|
| 1 High | 2020 | Female | 2467 | 1 |
| 2 High | 2020 | Female | 2468 | 1 |
| 3 High | 2020 | Female | 2470 | 1 |
| 4 High | 2020 | Female | 2473 | 1 |
| 5 High | 2020 | Female | 2474 | 1 |
| 6 High | 2020 | Female | 2475 | 1 |
| 7 High | 2020 | Female | 2476 | 1 |
| 8 High | 2020 | Female | 2477 | 1 |
| 9 High | 2020 | Female | 2478 | 1 |
| 10 High | 2020 | Female | 2479 | 1 |
| 11 High | 2020 | Female | 2480 | 1 |
| 12 High | 2020 | Female | 2481 | 1 |
| 13 High | 2020 | Female | 2489 | 1 |
| 14 High | 2020 | Female | 2491 | 1 |
| 15 High | 2020 | Female | 2494 | 1 |
| 16 High | 2020 | Female | 2496 | 1 |

Adverttime_dim

| ADVERTTIME_ID | ADVERT_MONTH | ADVERT_YEAR |
|---------------|--------------|-------------|
| 1 202004 | 04 | 2020 |
| 2 202003 | 03 | 2020 |
| 3 202004 | 04 | 2020 |
| 4 201911 | 11 | 2019 |
| 5 201911 | 11 | 2019 |
| 6 202004 | 04 | 2020 |
| 7 202003 | 03 | 2020 |
| 8 202004 | 04 | 2020 |
| 9 202004 | 04 | 2020 |
| 10 202004 | 04 | 2020 |
| 11 202004 | 04 | 2020 |
| 12 201912 | 12 | 2019 |
| 13 202004 | 04 | 2020 |
| 14 201912 | 12 | 2019 |
| 15 202004 | 04 | 2020 |
| 16 201912 | 12 | 2019 |
| 17 202004 | 04 | 2020 |
| 18 202004 | 04 | 2020 |

Advert_dim

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

Advertisement_fact

| ADVERTTIME_ID | PROPERTY_ID | ADVERT_ID | NO_OF_PROPERTIES |
|---------------|-------------|-----------|------------------|
| 1 20200329 | 415 | 16 | 1 |
| 2 20200315 | 2807 | 12 | 1 |
| 3 20200423 | 1770 | 16 | 1 |
| 4 20200326 | 4157 | 4 | 1 |
| 5 20200428 | 4660 | 10 | 1 |
| 6 20200324 | 6202 | 4 | 1 |
| 7 20200321 | 6065 | 1 | 1 |
| 8 20200323 | 6119 | 1 | 1 |
| 9 20200327 | 392 | 12 | 1 |
| 10 20200421 | 569 | 16 | 1 |
| 11 20200315 | 572 | 23 | 1 |
| 12 20200324 | 1202 | 12 | 1 |
| 13 20200321 | 2697 | 12 | 1 |
| 14 20200322 | 2662 | 16 | 1 |
| 15 20200407 | 3857 | 1 | 1 |
| 16 20200413 | 3135 | 1 | 1 |
| 17 20200418 | 310 | 16 | 1 |

Agentoffice_dim

| | PERSON_ID | OFFICE_ID |
|----|-----------|-----------|
| 1 | 982 | 153 |
| 2 | 989 | 622 |
| 3 | 1004 | 481 |
| 4 | 1014 | 166 |
| 5 | 1015 | 533 |
| 6 | 1018 | 1093 |
| 7 | 1024 | 960 |
| 8 | 1030 | 668 |
| 9 | 1056 | 644 |
| 10 | 1058 | 871 |
| 11 | 1081 | 87 |
| 12 | 1085 | 955 |
| 13 | 1087 | 1027 |
| 14 | 9 | 200 |
| 15 | 21 | 214 |
| 16 | 34 | 757 |
| 17 | 62 | 779 |
| 18 | 65 | 83 |

Office_dim

| | OFFICE_ID | OFFICE_NAME | OFFICE_SIZE |
|----|-----------|------------------------------------|-------------|
| 1 | 127 | Blackshaw Belconnen | Small |
| 2 | 1070 | Stone Real Estate Newtown | Small |
| 3 | 34 | Ascott Real Estate | Small |
| 4 | 681 | McGrath St Kilda | Small |
| 5 | 376 | Greencliff Agency | Small |
| 6 | 135 | Blewitt Properties | Small |
| 7 | 1033 | Shellabears | Small |
| 8 | 532 | LIVREALTY | Small |
| 9 | 225 | City Residential Property | Small |
| 10 | 40 | Aurora Estate Agents | Small |
| 11 | 879 | Ray White Canberra | Small |
| 12 | 223 | City Realty | Small |
| 13 | 178 | Buxton Stonnington | Small |
| 14 | 133 | Blaze Real Estate | Small |
| 15 | 289 | Duke Realty | Small |
| 16 | 132 | Blackshaw Woden, Weston & Molonglo | Small |
| 17 | 1041 | Smith and Elliott Real Estate | Small |
| 18 | 129 | Blackshaw Gungahlin | Small |

agent_fact

| | GENDER | AGENT_ID | TOTAL_SALARY | NO_OF_AGENTS |
|----|--------|----------|--------------|--------------|
| 1 | Female | 1056 | 190000 | 1 |
| 2 | Female | 21 | 200000 | 1 |
| 3 | Female | 65 | 190000 | 1 |
| 4 | Female | 89 | 200000 | 1 |
| 5 | Female | 159 | 195000 | 1 |
| 6 | Female | 340 | 175000 | 1 |
| 7 | Male | 357 | 200000 | 1 |
| 8 | Male | 280 | 190000 | 1 |
| 9 | Male | 309 | 175000 | 1 |
| 10 | Male | 436 | 175000 | 1 |
| 11 | Female | 495 | 200000 | 1 |
| 12 | Male | 529 | 200000 | 1 |
| 13 | Male | 1103 | 195000 | 1 |
| 14 | Female | 1198 | 195000 | 1 |
| 15 | Male | 1272 | 195000 | 1 |
| 16 | Male | 670 | 210000 | 1 |
| 17 | Female | 846 | 195000 | 1 |
| 18 | Female | 1469 | 195000 | 1 |

Suburb_dim

| SUBURB | |
|------------------|--|
| 1 Loganlea | |
| 2 Brisbane City | |
| 3 Kingston | |
| 4 Tingalpa | |
| 5 Kedron | |
| 6 Holland Park | |
| 7 Eagleby | |
| 8 Chermside West | |
| 9 Casey | |
| 10 Downer | |
| 11 Belconnen | |
| 12 Latham | |
| 13 Ainslie | |
| 14 Forrest | |
| 15 Campbell | |
| 16 Leopold | |
| 17 Portarlington | |

High Level tables:

Feature_dim1

| FEATURE_CODE | FEATURE_DESCRIPTION |
|--------------|-------------------------|
| 1 | 4 City Views |
| 2 | 5 Close to schools |
| 3 | 6 Close to shops |
| 4 | 23 Balcony |
| 5 | 34 Ducted Cooling |
| 6 | 35 Ducted Vacuum System |
| 7 | 36 Open Fireplace |
| 8 | 51 Split System Heating |
| 9 | 58 Garden Sheds |
| 10 | 67 Ocean Views |
| 11 | 73 Internal Laundry |
| 12 | 87 Broadband |
| 13 | 89 Workshop |
| 14 | 90 Deck with views |
| 15 | 111 Water Front |

Season_dim1

| SEASON_ID | SEASON_DESC |
|-----------|-------------|
| 1 AUTUMN | MAR-MAY |
| 2 SUMMER | DEC-FEB |
| 3 SPRING | SEP-NOV |
| 4 WINTER | JUN-AUG |

State_dim1

Script Output x Query Result x

SQL | All Rows Fetched: 8 in 0.016 seconds

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

Property_dim1

Script Output x Query Result x

SQL | Fetched 50 rows in 0.661 seconds

| PROPERTY_ID |
|-------------|
| 1 28 |
| 2 36 |
| 3 46 |
| 4 129 |
| 5 131 |
| 6 135 |
| 7 139 |
| 8 49 |
| 9 53 |
| 10 60 |
| 11 4 |
| 12 5 |
| 13 8 |
| 14 19 |
| 15 233 |
| 16 241 |
| 17 114 |

Propertyfeature_dim1

Script Output x Query Result x

 | Fetched 50 rows in 0.015 seconds

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

Time_dim2

Script Output x Query Result x

SQL | All Rows Fetched: 2 in 0.042 seconds

| CLIENT_YEAR |
|-------------|
| 1 2019 |
| 2 2020 |

Salesfact1:

Script Output x Query Result x

SQL | Fetched 50 rows in 0.036 seconds

| PROPERTY_ID | PROPERTY_TYPE | CLIENT_YEAR | STATE_CODE | SEASON_ID | TOTAL_SALES | NO_OF_SALE | NO_OF_PROPERTY |
|-------------|-----------------------------|-------------|------------|-----------|-------------|------------|----------------|
| 1 | 1 House | 2020 | VIC | SUMMER | 650000 | 1 | 1 |
| 2 | 81 House | 2020 | VIC | SUMMER | 825000 | 1 | 1 |
| 3 | 121 House | 2020 | SA | SUMMER | 495000 | 1 | 1 |
| 4 | 126 House | 2020 | SA | SUMMER | 520000 | 1 | 1 |
| 5 | 130 House | 2020 | SA | SUMMER | 1350000 | 1 | 1 |
| 6 | 312 Apartment / Unit / Flat | 2020 | SA | AUTUMN | 1750000 | 1 | 1 |
| 7 | 320 Townhouse | 2020 | QLD | SUMMER | 349000 | 1 | 1 |
| 8 | 324 Apartment / Unit / Flat | 2020 | QLD | SUMMER | 550000 | 1 | 1 |
| 9 | 333 House | 2020 | QLD | SUMMER | 1000000 | 1 | 1 |
| 10 | 337 House | 2020 | QLD | AUTUMN | 799000 | 1 | 1 |
| 11 | 342 Apartment / Unit / Flat | 2020 | QLD | SUMMER | 650000 | 1 | 1 |
| 12 | 367 House | 2020 | QLD | SUMMER | 399000 | 1 | 1 |
| 13 | 376 House | 2020 | QLD | AUTUMN | 690000 | 1 | 1 |
| 14 | 384 Apartment / Unit / Flat | 2020 | QLD | SUMMER | 500000 | 1 | 1 |
| 15 | 388 House | 2020 | QLD | SUMMER | 439000 | 1 | 1 |

Time_dim1:

Script Output x Query Result x

SQL | Fetched 50 rows in 0.018 seconds

| TIME_ID | YEAR | MONTH | DAY |
|-------------|------|-------|-----|
| 1 20191115 | 2019 | 11 | 15 |
| 2 20191116 | 2019 | 11 | 16 |
| 3 20191117 | 2019 | 11 | 17 |
| 4 20191118 | 2019 | 11 | 18 |
| 5 20191119 | 2019 | 11 | 19 |
| 6 20191120 | 2019 | 11 | 20 |
| 7 20191121 | 2019 | 11 | 21 |
| 8 20191122 | 2019 | 11 | 22 |
| 9 20191123 | 2019 | 11 | 23 |
| 10 20191124 | 2019 | 11 | 24 |
| 11 20191125 | 2019 | 11 | 25 |
| 12 20191126 | 2019 | 11 | 26 |
| 13 20191127 | 2019 | 11 | 27 |
| 14 20191128 | 2019 | 11 | 28 |
| 15 20191129 | 2019 | 11 | 29 |

Advert_dim1:

Script Output x Query Result x

SQL | All Rows Fetched: 25 in 0.014 seconds

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

Advertisementfact1:

Script Output x Query Result x

SQL | Fetched 50 rows in 0.015 seconds

| ADVERT_ID | TIME_ID | NO_OF_PROPERTIES |
|-----------|-------------|------------------|
| 1 | 16 20200419 | 18 |
| 2 | 16 20200430 | 17 |
| 3 | 16 20200317 | 23 |
| 4 | 16 20200402 | 28 |
| 5 | 12 20200308 | 11 |
| 6 | 17 20200404 | 1 |
| 7 | 23 20200314 | 3 |
| 8 | 16 20200319 | 18 |
| 9 | 12 20200411 | 9 |
| 10 | 23 20200407 | 5 |
| 11 | 15 20200229 | 1 |
| 12 | 12 20200325 | 16 |
| 13 | 23 20200425 | 4 |
| 14 | 16 20200226 | 7 |
| 15 | 15 20200319 | 1 |

Rentalperiod_dim1:

Script Output x Query Result x

SQL | All Rows Fetched: 3 in 0.019 seconds

| RENTAL_PERIOD | RENTAL_DESC |
|---------------|-------------|
| 1 Short | < 6 months |
| 2 Medium | 6-12 months |
| 3 Big | > 12 months |

Suburb_dim1:

Script Output x Query Result x

SQL | Fetched 50 rows in 0.015 seconds

| SUBURB |
|------------------|
| 1 Loganlea |
| 2 Brisbane City |
| 3 Kingston |
| 4 Tingalpa |
| 5 Kedron |
| 6 Holland Park |
| 7 Eagleby |
| 8 Chermside West |
| 9 Casey |
| 10 Downer |
| 11 Belconnen |
| 12 Latham |
| 13 Ainslie |
| 14 Forrest |
| 15 Campbell |

Propertyrenthistory_dim1:

Script Output x Query Result x

SQL | Fetched 50 rows in 0.205 seconds

| PROPERTY_ID | RENT_START_DATE | RENT_END_DATE | PRICE |
|-------------|-----------------|---------------|-------|
| 1 | 3049 27/APR/20 | 13/OCT/20 | 495 |
| 2 | 2948 03/MAY/20 | 19/OCT/20 | 195 |
| 3 | 3076 03/MAR/20 | 18/AUG/20 | 265 |
| 4 | 3096 20/APR/20 | 06/OCT/20 | 265 |
| 5 | 2954 21/APR/20 | 07/OCT/20 | 370 |
| 6 | 3003 16/JAN/20 | 02/JUL/20 | 335 |
| 7 | 3117 31/MAR/20 | 15/SEP/20 | 430 |
| 8 | 3934 04/APR/20 | 19/SEP/20 | 500 |
| 9 | 4419 18/JAN/20 | 04/JUL/20 | 630 |
| 10 | 3973 29/APR/20 | 15/OCT/20 | 1100 |
| 11 | 3885 12/JAN/20 | 28/JUN/20 | 650 |
| 12 | 3895 17/MAR/20 | 01/SEP/20 | 320 |
| 13 | 3943 30/APR/20 | 16/OCT/20 | 625 |
| 14 | 4131 02/FEB/20 | 19/JUL/20 | 595 |
| 15 | 3957 13/FEB/20 | 30/JUL/20 | 380 |

Propertyscale_dim1:

Script Output x Query Result x

SQL | All Rows Fetched: 5 in 0.291 seconds

| PROPERTY_SCALE | SCALE_DESC |
|----------------|---------------|
| 1 Extra Small | <= 1 Bedroom |
| 2 Small | 2-3 Bedrooms |
| 3 Medium | 3-6 Bedrooms |
| 4 Large | 6-10 Bedrooms |
| 5 Extra Large | > 10 Bedrooms |

Featurecategory_dim1:

Script Output x Query Result x

SQL | All Rows Fetched: 3 in 0.017 seconds

| FEATURECATEGORY_ID | CATEGORY_DESC |
|--------------------|----------------|
| 1 Basic | < 10 features |
| 2 Standard | 10-20 features |
| 3 Luxurious | > 20 features |

Rentfact1:

Script Output x Query Result x

SQL | Fetched 50 rows in 0.023 seconds

| SUBURB | PROPERTY_ID | FEATURECATEGORY_ID | RENTAL_PERIOD | PROPERTY_SCALE | TIME_ID | PROPERTY_TYPE | STATE_CODE | SEASON_ID | RENTAL_FEES |
|---------|-------------|--------------------|---------------|----------------|----------|-----------------------------|------------|-----------|-------------|
| 1 City | 4701 | Basic | Short | Small | 20200320 | Apartment / Unit / Flat ACT | ACT | AUTUMN | 15600 |
| 2 City | 4752 | Basic | Short | Extra Small | 20200323 | Apartment / Unit / Flat ACT | ACT | AUTUMN | 15600 |
| 3 City | 4811 | Standard | Short | Extra Small | 20200426 | Apartment / Unit / Flat ACT | ACT | AUTUMN | 12313 |
| 4 City | 4830 | Basic | Short | Extra Small | 20200114 | Apartment / Unit / Flat ACT | ACT | SUMMER | 12000 |
| 5 City | 4831 | Basic | Short | Extra Small | 20191230 | Apartment / Unit / Flat ACT | ACT | SUMMER | 11520 |
| 6 City | 4839 | Basic | Short | Extra Small | 20200324 | Apartment / Unit / Flat ACT | ACT | AUTUMN | 11160 |
| 7 City | 4840 | Basic | Short | Extra Small | 20200420 | Apartment / Unit / Flat ACT | ACT | AUTUMN | 11347 |
| 8 City | 4849 | Standard | Short | Small | 20200412 | Apartment / Unit / Flat ACT | ACT | AUTUMN | 14880 |
| 9 City | 4859 | Standard | Short | Small | 20200214 | Apartment / Unit / Flat ACT | ACT | SUMMER | 16800 |
| 10 City | 4918 | Standard | Short | Extra Small | 20200311 | Apartment / Unit / Flat ACT | ACT | AUTUMN | 12000 |
| 11 City | 4925 | Basic | Short | Extra Small | 20200211 | Apartment / Unit / Flat ACT | ACT | SUMMER | 10560 |
| 12 City | 4978 | Basic | Short | Small | 20200428 | Apartment / Unit / Flat ACT | ACT | AUTUMN | 16417 |
| 13 City | 4989 | Basic | Short | Extra Small | 20200206 | Apartment / Unit / Flat ACT | ACT | SUMMER | 10800 |
| 14 City | 5049 | Basic | Short | Extra Small | 20200226 | House | ACT | SUMMER | 12000 |
| 15 City | 5106 | Basic | Short | Medium | 20200428 | Apartment / Unit / Flat ACT | ACT | AUTUMN | 21729 |

Visitfact1:

Script Output x Query Result x

SQL | All Rows Fetched: 14 in 0.016 seconds

| TIME_ID | SEASON_ID | NO_OF_PROPERTIES | TOTAL_NO_OF_VISITS |
|---------------------|-----------|------------------|--------------------|
| 1 2020MARFRIDAY | AUTUMN | 24 | 64 |
| 2 2020MARMONDAY | AUTUMN | 23 | 62 |
| 3 2020MARSATURDAY | AUTUMN | 38 | 77 |
| 4 2020APRFRIDAY | AUTUMN | 5 | 11 |
| 5 2020MARTHURSDAY | AUTUMN | 24 | 58 |
| 6 2020MARSUNDAY | AUTUMN | 22 | 50 |
| 7 2020APRTUESDAY | AUTUMN | 7 | 19 |
| 8 2020APRMONDAY | AUTUMN | 10 | 28 |
| 9 2020APRSATURDAY | AUTUMN | 11 | 30 |
| 10 2020APRTHURSDAY | AUTUMN | 5 | 12 |
| 11 2020APRWEDNESDAY | AUTUMN | 8 | 21 |
| 12 2020MARTUESDAY | AUTUMN | 27 | 64 |
| 13 2020MARWEDNESDAY | AUTUMN | 25 | 55 |
| 14 2020APRSUNDAY | AUTUMN | 7 | 23 |

Office_dim1:

Script Output x Query... x

SQL | Fetched 50 rows in 0.025 seconds

| OFFICE_ID | OFFICE_NAME | OFFICE_SIZE |
|-----------|------------------------------------------|-------------|
| 1 | 1070 Stone Real Estate Newtown | Small |
| 2 | 532 LIVREALTY | Small |
| 3 | 331 First National Real Estate David Kay | Small |
| 4 | 1052 Stanley Samuels Property | Small |
| 5 | 649 McGrath Bulimba | Small |
| 6 | 955 Ray White Upper Coomera | Medium |
| 7 | 231 Collins Simms | Small |
| 8 | 253 Coronis Lutwyche | Big |
| 9 | 654 McGrath Double Bay | Small |
| 10 | 800 Premier Capital Properties | Small |
| 11 | 41 Aurora Realty Brisbane | Small |
| 12 | 505 Jim Aitken + Partners Jordan Springs | Small |
| 13 | 937 Ray White Robina | Medium |
| 14 | 715 Nelson Alexander Brunswick | Medium |
| 15 | 626 Mantra Realty | Small |

Gender_dim1:

| GENDER |
|----------|
| 1 Male |
| 2 Female |

Officegroupolist_dim1:

Script Output x Query Result x

SQL | Fetched 50 rows in 0.034 seconds

| OFFICELIST | WEIGHT_FACTOR |
|------------|---------------|
| 1 238 | 1 |
| 2 255 | 1 |
| 3 241 | 1 |
| 4 25 | 1 |
| 5 27 | 1 |
| 6 944 | 1 |
| 7 792 | 1 |
| 8 1056 | 1 |
| 9 664 | 1 |
| 10 10 | 1 |
| 11 253 | 1 |
| 12 129 | 1 |
| 13 226 | 1 |
| 14 854 | 1 |
| 15 471 | 1 |

Groupbridge:

Script Output x Query Result x

50 rows | Fetched 50 rows in 0.539 seconds

| | OFFICELIST | OFFICE_ID |
|----|------------|-----------|
| 1 | 1027 | 1027 |
| 2 | 149 | 149 |
| 3 | 134 | 134 |
| 4 | 320 | 320 |
| 5 | 556 | 556 |
| 6 | 253 | 253 |
| 7 | 40 | 40 |
| 8 | 964 | 964 |
| 9 | 246_253 | 253 |
| 10 | 1157 | 1157 |
| 11 | 1106 | 1106 |
| 12 | 951 | 951 |
| 13 | 861 | 861 |
| 14 | 842 | 842 |
| 15 | 731 | 731 |

Agentfact1:

| | GENDER | OFFICELIST | NO_OF_AGENTS |
|----|--------|------------|--------------|
| 1 | Female | 263 | 1 |
| 2 | Female | 338 | 1 |
| 3 | Female | 729 | 2 |
| 4 | Female | 550 | 2 |
| 5 | Female | 102 | 3 |
| 6 | Male | 69 | 1 |
| 7 | Female | 335 | 1 |
| 8 | Male | 440 | 2 |
| 9 | Male | 785 | 6 |
| 10 | Female | 545 | 1 |
| 11 | Male | 145 | 1 |
| 12 | Female | 710 | 1 |
| 13 | Female | 183 | 1 |
| 14 | Female | 187 | 3 |
| 15 | Female | 836 | 1 |

Clientwishlist_dim1:

[illegible]

Budgetcategory_dim1:

| BUDGETCATEGORY_ID | BUDGETCATEGORY_DESCRIPTION |
|-------------------|----------------------------|
| 1 Low | 0 to 1000 |
| 2 Medium | 1001 to 100000 |
| 3 High | 100001 to 10000000 |

Wishbridge:

| WISHLIST | FEATURE_CODE |
|----------------------------------------------------------------------------------|--------------|
| 1 1_2_10_59_61_117 | 2 |
| 2 1_2_5_6_7_19_20_38_40_235_239 | 38 |
| 3 1_2_5_6_7_19_20_38_40_235_239 | 239 |
| 4 1_2_5_6_7_39_86_493 | 2 |
| 5 1_2_5_6_7_39_86_493 | 5 |
| 6 2_9_14_20_26 | 26 |
| 7 1_2_9_14_18_20_23_25_26_29_40_41_45_71_73_74_75_94_127_239 | 26 |
| 8 1_2_9_14_18_20_23_25_26_29_40_41_45_71_73_74_75_94_127_239 | 45 |
| 9 1_2_9_14_18_20_23_25_26_29_40_41_45_71_73_74_75_94_127_239 | 94 |
| 10 1_2_5_6_7_16_26_29_38_42_71_73_87_100_101_113_114_115_126_127_203_235_257_468 | 113 |
| 11 1_2_5_6_7_16_26_29_38_42_71_73_87_100_101_113_114_115_126_127_203_235_257_468 | 16 |
| 12 1_2_16_20_23_24_26_29_30_31_43_52_74_76_87_89_100_591 | 30 |
| 13 1_2_16_20_23_24_26_29_30_31_43_52_74_76_87_89_100_591 | 31 |
| 14 1_2_16_20_23_24_26_29_30_31_43_52_74_76_87_89_100_591 | 89 |
| 15 1_2_17_21_23_277 | 17 |

Clientfact1:

| GENDER | CLIENT_YEAR | BUDGETCATEGORY_ID | WISHLIST |
|-----------|-------------|-------------------|------------------------------------------------------------|
| 1 Male | 2019 | High | (null) |
| 2 Male | 2020 | Medium | (null) |
| 3 Male | 2020 | High | 1_14_20_22_26_64_86_87_120_213 |
| 4 Male | 2020 | High | 1_2_5_6_7_19_20 |
| 5 Male | 2020 | Low | 2 |
| 6 Female | 2020 | Medium | 2_5_6_7_9_19_23_25_26_33_45_72_73_74_127 |
| 7 Male | 2020 | High | 2 |
| 8 Female | 2020 | High | 1_2_17_22_86 |
| 9 Female | 2020 | Medium | (null) |
| 10 Male | 2020 | High | 2_9_14_20_26 |
| 11 Female | 2020 | High | 1_2_5_6_7_19_20_38_40_235_239 |
| 12 Male | 2020 | High | 1_17_22_86 |
| 13 Male | 2020 | High | 1_2_9_14_18_20_23_25_26_29_40_41_45_71_73_74_75_94_127_239 |
| 14 Male | 2020 | High | 1_2_5_6_7_19_20_21_23 |
| 15 Male | 2020 | Low | 1_2_18_23_26_43_52_126_127 |

TASK 3: Create the following reports using OLAP queries.

OLAP FOR LOW LEVEL

Report 1:

(a) Which are the top 3 property types in NSW with highest total sales?

(b) This query can help the management to analyse which is the most popular property type for sales in NSW. By doing so they know which property type people prefer in NSW and they can focus more on building those property types in that state to get revenue from sales of the property.

(c) SQL command for report 1

```
SELECT
*
FROM
(
SELECT
s.state_code,
p.property_type,
SUM(sf.Total_sales) as SALES,
DENSE_RANK() OVER
(
ORDER BY
SUM(sf.Total_sales) DESC
)
AS custom_rank
FROM state_dim s,
propertytype_dim
p, salesfact sf
WHERE s.state_code=sf.state_code
and p.property_type=sf.property_type
and s.state_code ='NSW'
GROUP BY s.state_code,p.property_type
)
WHERE CUSTOM_RANK <=3
;
```

(d) The screenshots of the query results

| | STATE_CODE | PROPERTY_TYPE | SALES | CUSTOM_RANK |
|---|------------|-------------------------|----------|-------------|
| 1 | NSW | House | 64575850 | 1 |
| 2 | NSW | Apartment / Unit / Flat | 32464788 | 2 |
| 3 | NSW | Semi-Detached | 3394000 | 3 |

Report 2:

(a) Top 50% of property type in NSW state with total sales

(C) SQL command for Report 2

```
SELECT *
FROM (
SELECT s.state_code,p.property_type,
SUM(sf.Total_sales) as SALES,
PERCENT_RANK() OVER (ORDER BY SUM(sf.Total_sales) DESC)
AS "PERCENT RANK"
FROM state_dim s, propertytype_dim p, salesfact sf
WHERE s.state_code=sf.state_code and p.property_type=sf.property_type and s.state_code
='NSW'
GROUP BY s.state_code,p.property_type
)
WHERE "PERCENT RANK" < 0.5;
```

(d) The screenshots of the query results

[illegible]

Report 3:

(a) The total sales of all property type in NSW

(c) SQL command for Report 3

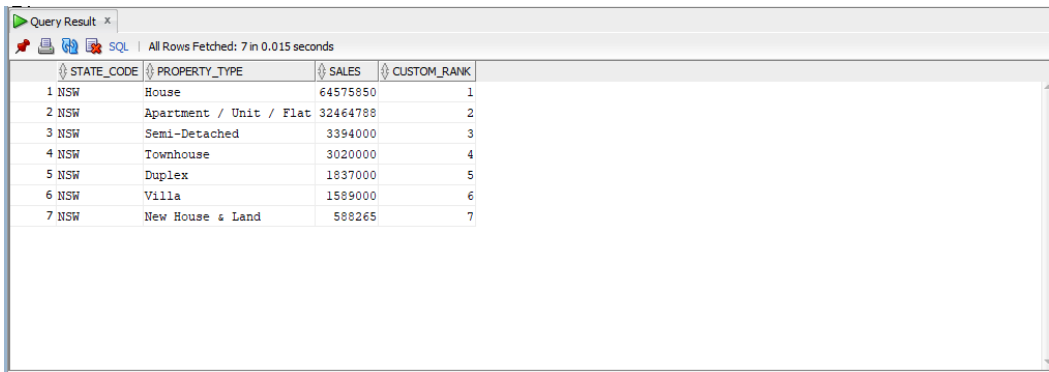
```
SELECT
  s.state_code,
  p.property_type,
  SUM (sf.Total_sales) as SALES,
  DENSE_RANK () OVER
  (
    ORDER BY SUM (sf.Total_sales) DESC
  )
```

```

AS custom_rank
FROM state_dim s,
     propertytype_dim p,
     salesfact sf
WHERE s.state_code=sf.state_code
and p.property_type=sf.property_type
and s.state_code ='NSW'
GROUP BY s.state_code,p.property_type;

```

(d) The screenshots of the query results



| STATE_CODE | PROPERTY_TYPE | SALES | CUSTOM_RANK |
|------------|-------------------------|----------|-------------|
| 1 NSW | House | 64575850 | 1 |
| 2 NSW | Apartment / Unit / Flat | 32464788 | 2 |
| 3 NSW | Semi-Detached | 3394000 | 3 |
| 4 NSW | Townhouse | 3020000 | 4 |
| 5 NSW | Duplex | 1837000 | 5 |
| 6 NSW | Villa | 1589000 | 6 |
| 7 NSW | New House & Land | 588265 | 7 |

REPORT 4 and REPORT 5:

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

Report 4:

(c) SQL command for Report 4

```

SELECT
    decode(GROUPING(s.suburb), 1, 'All_Suburbs', s.suburb) AS Suburb,
    decode(GROUPING(rs.rentstart_year), 1, 'All_Years', rs.rentstart_year) AS Year,
    decode(GROUPING(p.property_type), 1, 'All_Property_Types', p.property_type) AS
"Property Type",
    SUM(r.rental_fees) AS "Total Rental Fees"
FROM
    rentstart_dim    rs,
    suburb_dim       s,
    propertytype_dim p,
    rentfact         r
WHERE
    s.suburb = r.suburb
    AND rs.rentstart_id = r.rentstart_id

```

```

AND p.property_type = r.property_type
GROUP BY CUBE
( s.suburb,
  rs.rentstart_year,
  p.property_type) ;

```

(d) The screenshots of the query results

| | SUBURB | YEAR | Property Type | Total Rental Fees |
|----|-------------|-----------|-------------------------------|-------------------|
| 1 | All_Suburbs | All_Years | All_Property_Types | 22816569 |
| 2 | All_Suburbs | All_Years | House | 8060088 |
| 3 | All_Suburbs | All_Years | Villa | 116503 |
| 4 | All_Suburbs | All_Years | Duplex | 150738 |
| 5 | All_Suburbs | All_Years | Studio | 92534 |
| 6 | All_Suburbs | All_Years | Terrace | 124340 |
| 7 | All_Suburbs | All_Years | Penthouse | 12071 |
| 8 | All_Suburbs | All_Years | Townhouse | 1851567 |
| 9 | All_Suburbs | All_Years | Semi-Detached | 113824 |
| 10 | All_Suburbs | All_Years | Apartment / Unit / Flat | 12272824 |
| 11 | All_Suburbs | All_Years | New Apartments / Off the Plan | 22080 |
| 12 | All_Suburbs | 2019 | All_Property_Types | 309600 |
| 13 | All_Suburbs | 2019 | House | 164520 |
| 14 | All_Suburbs | 2019 | Apartment / Unit / Flat | 145080 |

Report 5:

(c) SQL command for Report 5

```

SELECT
  decode(GROUPING(s.suburb), 1, 'All_Suburbs', s.suburb) AS Suburb,
  decode(GROUPING(rs.rentstart_year), 1, 'All_Years', rs.rentstart_year) AS Year,
  decode(GROUPING(p.property_type), 1, 'All_Property_Types', p.property_type) AS
"Property Type",
  SUM(r.rental_fees) AS "Total Rental Fees"
FROM
  rentstart_dim    rs,
  suburb_dim       s,
  propertytype_dim p,
  rentfact         r
WHERE
  s.suburb = r.suburb
  AND rs.rentstart_id = r.rentstart_id
  AND p.property_type = r.property_type
GROUP BY s.suburb,
CUBE
  (rs.rentstart_year,
  p.property_type) ;

```


(d) The screenshots of the query results

| | SUBURB | YEAR | Property Type | Total Rental Fees |
|----|--------|-----------|-------------------------|-------------------|
| 1 | City | All_Years | All_Property_Types | 295806 |
| 2 | City | All_Years | House | 12000 |
| 3 | City | All_Years | Apartment / Unit / Flat | 283806 |
| 4 | City | 2019 | All_Property_Types | 11520 |
| 5 | City | 2019 | Apartment / Unit / Flat | 11520 |
| 6 | City | 2020 | All_Property_Types | 284286 |
| 7 | City | 2020 | House | 12000 |
| 8 | City | 2020 | Apartment / Unit / Flat | 272286 |
| 9 | Cook | All_Years | All_Property_Types | 25440 |
| 10 | Cook | All_Years | House | 25440 |
| 11 | Cook | 2019 | All_Property_Types | 12480 |
| 12 | Cook | 2019 | House | 12480 |
| 13 | Cook | 2020 | All_Property_Types | 12960 |
| 14 | Cook | 2020 | House | 12960 |

REPORT 6 and REPORT 7: Produce 2 other sub-totals reports that are useful for management using Roll-up and Partial Roll-up.

Report 6

- (a) What are the sub-total and total sales based on each season,property_type and state? (Using Roll-up)
- (b) Using this query, one can know that which state which property_type and in which season is the sales generated the most. This can help one decide that in that state and that season which property to prefer.
- (c) SQL code for the query:

```

SELECT
    decode(GROUPING(s.season_id), 1, 'All_Seasons', s.season_id) AS Season,
    decode(GROUPING(st.state_code), 1, 'All_States', st.state_code) AS State,
    decode(GROUPING(sf.property_type), 1, 'All_Property_type', sf.property_type) AS
Property_type,
    SUM(sf.total_sales) AS "Total Sales"
FROM

    season_dim s,
    State_dim st,
    salesfact sf,
    propertytype_dim p
WHERE

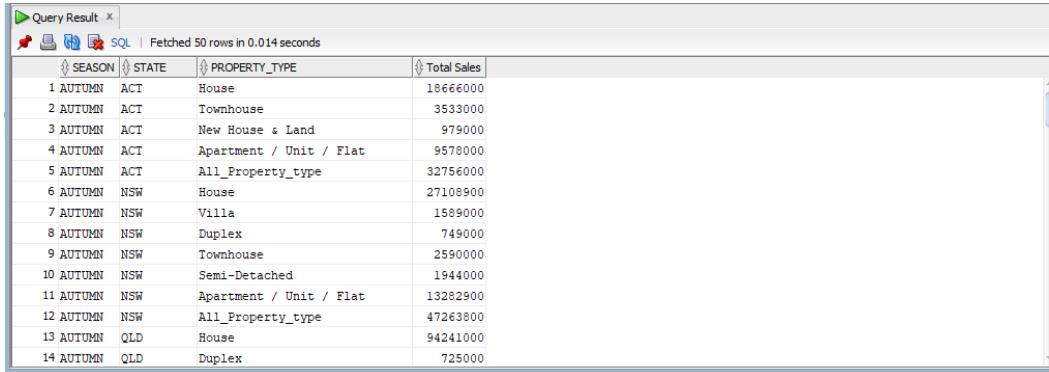
```

```

s.season_id = sf.season_id
AND st.state_code = sf.state_code
AND p.property_type = sf.property_type
GROUP BY Rollup
( s.season_id,st.state_code,sf.property_type ) ;

```

(d) The screenshots of the query results



| SEASON | STATE | PROPERTY_TYPE | Total Sales |
|-----------|-------|-------------------------|-------------|
| 1 AUTUMN | ACT | House | 18666000 |
| 2 AUTUMN | ACT | Townhouse | 3533000 |
| 3 AUTUMN | ACT | New House & Land | 979000 |
| 4 AUTUMN | ACT | Apartment / Unit / Flat | 9578000 |
| 5 AUTUMN | ACT | All_Property_type | 32756000 |
| 6 AUTUMN | NSW | House | 27108900 |
| 7 AUTUMN | NSW | Villa | 1589000 |
| 8 AUTUMN | NSW | Duplex | 749000 |
| 9 AUTUMN | NSW | Townhouse | 2590000 |
| 10 AUTUMN | NSW | Semi-Detached | 1944000 |
| 11 AUTUMN | NSW | Apartment / Unit / Flat | 13282900 |
| 12 AUTUMN | NSW | All_Property_type | 47263800 |
| 13 AUTUMN | QLD | House | 94241000 |
| 14 AUTUMN | QLD | Duplex | 725000 |

Report 7

- (a) What are the sub-total and total sales on each season,property_type and state?
(Using Partial Roll-up)
- (b) Using this query will keep the season constant and will show that which combination of property_type and state will generate the maximum sales. Hence one can know which property to prefer while selecting to buy a property in that state.
- (c) SQL code for the query

```

SELECT
    decode(GROUPING(s.season_id), 1, 'All_Seasons', s.season_id) AS Season,
    decode(GROUPING(st.state_code), 1, 'All_States', st.state_code) AS State,
    decode(GROUPING(sf.property_type), 1, 'All_Property_type', sf.property_type) AS
Property_type,
    SUM(sf.total_sales) AS "Total Sales"
FROM

    season_dim s,
    State_dim st,
    salesfact sf,
    propertytype_dim p

WHERE

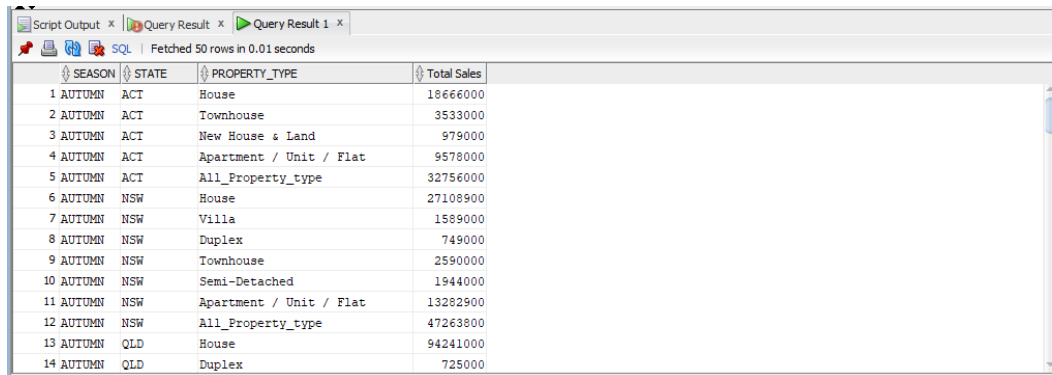
```

```

s.season_id = sf.season_id
AND st.state_code = sf.state_code
AND p.property_type = sf.property_type
GROUP BY s.season_id,
Rollup
(st.state_code,sf.property_type ) ;

```

(d) The screenshots of the query results



| SEASON | STATE | PROPERTY_TYPE | Total Sales |
|-----------|-------|-------------------------|-------------|
| 1 AUTUMN | ACT | House | 18666000 |
| 2 AUTUMN | ACT | Townhouse | 3533000 |
| 3 AUTUMN | ACT | New House & Land | 979000 |
| 4 AUTUMN | ACT | Apartment / Unit / Flat | 9578000 |
| 5 AUTUMN | ACT | All_Property_type | 32756000 |
| 6 AUTUMN | NSW | House | 27108900 |
| 7 AUTUMN | NSW | Villa | 1589000 |
| 8 AUTUMN | NSW | Duplex | 749000 |
| 9 AUTUMN | NSW | Townhouse | 2590000 |
| 10 AUTUMN | NSW | Semi-Detached | 1944000 |
| 11 AUTUMN | NSW | Apartment / Unit / Flat | 13282900 |
| 12 AUTUMN | NSW | All_Property_type | 47263800 |
| 13 AUTUMN | QLD | House | 94241000 |
| 14 AUTUMN | QLD | Duplex | 725000 |

Report 8

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

(c) SQL code for the query

```

select
c.budgetcategory_id,
c.client_year,
to_char (SUM(c.Total_no_of_clients), '9,999,999,999') AS Q_CLIENT,
TO_CHAR (SUM(SUM(c.Total_no_of_clients)) OVER
(ORDER BY c.budgetcategory_id,c.client_year
ROWS UNBOUNDED PRECEDING),
'9,999,999,999') AS CUM_CLIENT
FROM time_dim t, budgetcategory_dim b, clientfact c
where
c.client_year = t.client_year and
c.budgetcategory_id = b.budgetcategory_id and
b.budgetcategory_id='High'
and t.client_year in ('2019', '2020')
GROUP BY c.budgetcategory_id,c.client_year;

```

(d) Screenshot of the query result

| BUDGETCATEGORY_ID | CLIENT_YEAR | Q_CLIENT | CUM_CLIENT |
|-------------------|-------------|----------|------------|
| 1 High | 2019 | 23 | 23 |
| 2 High | 2020 | 892 | 915 |

Report 9

(a) What is the moving 3-year average of rent fees based on state and rent start year?

(b) This report will help the management get the average rental fees of 3 years which can help predict the next years rental fees and take necessary decisions.

(c) SQL code for the query

```
select r.state_code, rs.rentstart_year, to_char(sum(rental_fees), '9,999,999,999') as "Q rent",
to_char(AVG(SUM(rental_fees)) OVER(order by r.state_code, rs.rentstart_year
rows 2 preceding), '9,999,999,999') as three_year_moving_average
from
rentfact r, rentstart_dim rs
where r.rentstart_id=rs.rentstart_id and rs.rentstart_year in ('2018','2019','2020') and
r.state_code in ('TAS','NSW','VIC')
group by r.state_code, rs.rentstart_year;
```

(d) Screenshot of the query result

| STATE_CODE | RENTSTART_YEAR | Q rent | THREE_YEAR_MOVING_AVERAGE |
|------------|----------------|-----------|---------------------------|
| 1 NSW | 2019 | 84,000 | 84,000 |
| 2 NSW | 2020 | 6,842,443 | 3,463,222 |
| 3 VIC | 2019 | 90,000 | 2,338,814 |
| 4 VIC | 2020 | 5,792,437 | 4,241,627 |

Report 10

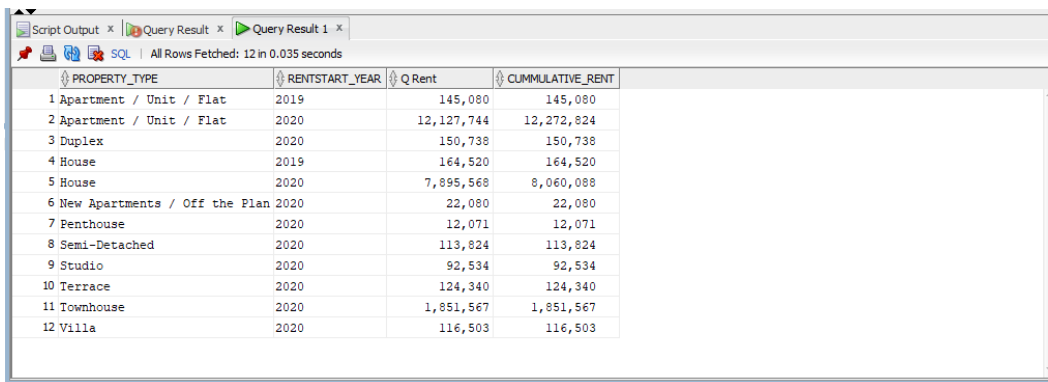
(a) Cumulative aggregate of rental fees based on property_type, rent start year

(b) This information can help analyze the property types which have higher increase in rental fees in succeeding years. Helps predict pattern in popular property types.

(C) SQL code for the query

```
r.property_type,
rs.rentstart_year,
to_char(sum(r.rental_fees),'9,999,999,999') as "Q Rent",
to_char(sum(sum(r.rental_fees)) over (
partition by r.property_type ORDER by
r.property_type, rs.rentstart_year
ROWS unbounded preceding),'9,999,999,999') as CUMMULATIVE_RENT
FROM
rentfact r, propertytype_dim p, rentstart_dim rs
where r.rentstart_id=rs.rentstart_id and
r.property_type= p.property_type
and rentstart_year in ('2019','2020')
group by r.property_type, rs.rentstart_year;
```

(d) Screenshot of the query result



| PROPERTY_TYPE | RENTSTART_YEAR | Q Rent | CUMMULATIVE_RENT |
|---------------------------------|----------------|------------|------------------|
| 1 Apartment / Unit / Flat | 2019 | 145,080 | 145,080 |
| 2 Apartment / Unit / Flat | 2020 | 12,127,744 | 12,272,824 |
| 3 Duplex | 2020 | 150,738 | 150,738 |
| 4 House | 2019 | 164,520 | 164,520 |
| 5 House | 2020 | 7,895,568 | 8,060,088 |
| 6 New Apartments / Off the Plan | 2020 | 22,080 | 22,080 |
| 7 Penthouse | 2020 | 12,071 | 12,071 |
| 8 Semi-Detached | 2020 | 113,824 | 113,824 |
| 9 Studio | 2020 | 92,534 | 92,534 |
| 10 Terrace | 2020 | 124,340 | 124,340 |
| 11 Townhouse | 2020 | 1,851,567 | 1,851,567 |
| 12 Villa | 2020 | 116,503 | 116,503 |

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 get the ranks of total sales of property based on property type, year and state. This can be beneficial to the management in factoring which property type is sold most in which year in which state. Strategic decisions on buying and selling of property can be made by this information.

(c) SQL code for the query

```
Select sf.property_type, t.client_year as Sales_Year, s.state_code,
```

```

to_char (sum(sf.total_sales)) as Sales,
Rank() over (partition by sf.property_type
order by sum (total_sales) desc) as Rank_by_property_type,
Rank() over (partition by t.client_year
order by sum (total_sales) desc) as Rank_by_year,
Rank() over (partition by s.state_code
order by sum (total_sales) desc) as Rank_by_state
From propertytype_dim p, time_dim t, state_dim s, salesfact sf
where
p.property_type=sf.property_type
and t.client_year=sf.client_year
and s.state_code=sf.state_code
group by sf.property_type,t.client_year, s.state_code;

```

(d)

| PROPERTY_TYPE | SALES_YEAR | STATE_CODE | SALES | RANK_BY_PROPERTY_TYPE | RANK_BY_YEAR | RANK_BY_STATE |
|---------------------------|------------|------------|-----------|-----------------------|--------------|---------------|
| 1 Apartment / Unit / Flat | 2020 | QLD | 57619500 | 1 | 4 | 2 |
| 2 Apartment / Unit / Flat | 2020 | VIC | 32899700 | 2 | 6 | 2 |
| 3 Apartment / Unit / Flat | 2020 | NSW | 32025788 | 3 | 8 | 2 |
| 4 Apartment / Unit / Flat | 2020 | ACT | 22330800 | 4 | 9 | 2 |
| 5 Apartment / Unit / Flat | 2020 | WA | 5843000 | 5 | 14 | 2 |
| 6 Apartment / Unit / Flat | 2020 | SA | 3359000 | 6 | 17 | 2 |
| 7 Apartment / Unit / Flat | 2019 | ACT | 1074000 | 7 | 7 | 5 |
| 8 Apartment / Unit / Flat | 2019 | NSW | 439000 | 8 | 10 | 9 |
| 9 Block of Units | 2020 | QLD | 4329000 | 1 | 15 | 4 |
| 10 Development Site | 2020 | VIC | 1300000 | 1 | 27 | 5 |
| 11 Duplex | 2020 | NSW | 1837000 | 1 | 24 | 5 |
| 12 Duplex | 2020 | QLD | 1333000 | 2 | 26 | 8 |
| 13 Duplex | 2019 | QLD | 1100000 | 3 | 6 | 9 |
| 14 Duplex | 2020 | WA | 300000 | 4 | 35 | 7 |
| 15 House | 2020 | QLD | 167877900 | 1 | 1 | 1 |
| 16 House | 2020 | VIC | 119785499 | 2 | 2 | 1 |
| 17 House | 2020 | NSW | 63725850 | 3 | 3 | 1 |

Report 12

- (a) What is the ranking of each property type based on total rent fees? In this, also show the ranking of each season based on total rent fees.
- (b) By using this query, we will get the ranking of total rent fees from the point of view of season and property type. This will give ranking of total rental fees. By doing so it can be identified that which property type has the maximum rent and which has the minimum.
- (c) SQL Query for Report 12

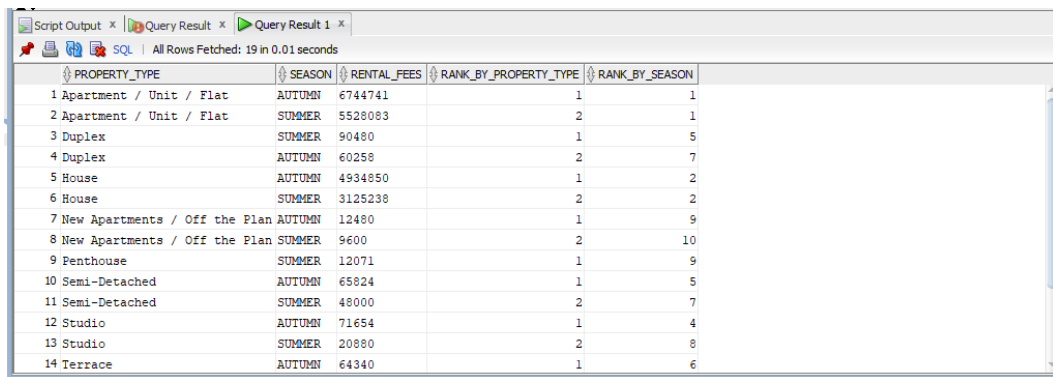
```
Select rf.property_type, s.season_id as season,
```

```

to_char (sum(rf.rental_fees)) as Rental_fees,
Rank() over (partition by rf.property_type
order by sum (rf.rental_fees) desc) as Rank_by_property_type,
Rank() over (partition by s.season_id
order by sum (rf.rental_fees) desc) as Rank_by_season
From propertytype_dim p, season_dim s, rentfact rf
where
p.property_type=rf.property_type
and s.season_id=rf.season_id
group by rf.property_type,s.season_id;

```

(d) Screenshot of the query result



| PROPERTY_TYPE | SEASON | RENTAL_FEES | RANK_BY_PROPERTY_TYPE | RANK_BY_SEASON |
|---------------------------------|--------|-------------|-----------------------|----------------|
| 1 Apartment / Unit / Flat | AUTUMN | 6744741 | 1 | 1 |
| 2 Apartment / Unit / Flat | SUMMER | 5528083 | 2 | 1 |
| 3 Duplex | SUMMER | 90480 | 1 | 5 |
| 4 Duplex | AUTUMN | 60258 | 2 | 7 |
| 5 House | AUTUMN | 4934850 | 1 | 2 |
| 6 House | SUMMER | 3125238 | 2 | 2 |
| 7 New Apartments / Off the Plan | AUTUMN | 12480 | 1 | 9 |
| 8 New Apartments / Off the Plan | SUMMER | 9600 | 2 | 10 |
| 9 Penthouse | SUMMER | 12071 | 1 | 9 |
| 10 Semi-Detached | AUTUMN | 65824 | 1 | 5 |
| 11 Semi-Detached | SUMMER | 48000 | 2 | 7 |
| 12 Studio | AUTUMN | 71654 | 1 | 4 |
| 13 Studio | SUMMER | 20880 | 2 | 8 |
| 14 Terrace | AUTUMN | 64340 | 1 | 6 |

OLAP FOR HIGH LEVEL

Report 1

```

(c)
SELECT * FROM (
SELECT s.state_code,
p.property_type,
SUM(sf.Total_sales) as SALES,
DENSE_RANK() OVER (ORDER BY SUM(sf.Total_sales) DESC)
AS custom_rank
FROM state_dim s,
propertytype_dim p,
salesfact sf
WHERE s.state_code=sf.state_code
and p.property_type=sf.property_type
and s.state_code ='NSW'
GROUP BY s.state_code,p.property_type)
WHERE CUSTOM_RANK <=3;

```

(d)

Query Result x

All Rows Fetched: 3 in 0.056 seconds

| STATE_CODE | PROPERTY_TYPE | SALES | CUSTOM_RANK |
|------------|-------------------------|----------|-------------|
| 1 NSW | House | 64575850 | 1 |
| 2 NSW | Apartment / Unit / Flat | 32464788 | 2 |
| 3 NSW | Semi-Detached | 3394000 | 3 |

Report 2

(c)

```
SELECT *
FROM (
SELECT s.state_code,
p.property_type,
SUM(sf.Total_sales) as SALES,
PERCENT_RANK() OVER (ORDER BY SUM(sf.Total_sales) DESC)
AS "PERCENT RANK"
FROM state_dim1 s, propertytype_dim1 p, salesfact1 sf
WHERE s.state_code=sf.state_code and p.property_type=sf.property_type and s.state_code
='NSW'
GROUP BY s.state_code,p.property_type
)
WHERE "PERCENT RANK" < 0.5;
```

(d)

[illegible]

Report 3

(c)

```
SELECT s.state_code,  
p.property_type,  
SUM(sf.Total_sales) as SALES,  
DENSE_RANK() OVER (ORDER BY SUM(sf.Total_sales) DESC)
```


AS custom_rank

FROM state_dim1 s, propertytype_dim1 p, salesfact1 sf

WHERE s.state_code=sf.state_code and p.property_type=sf.property_type and s.state_code='NSW'

GROUP BY s.state_code,p.property_type;

(d)

| STATE_CODE | PROPERTY_TYPE | SALES | CUSTOM_RANK |
|------------|-------------------------|----------|-------------|
| 1 NSW | House | 64575850 | 1 |
| 2 NSW | Apartment / Unit / Flat | 32464788 | 2 |
| 3 NSW | Semi-Detached | 3394000 | 3 |
| 4 NSW | Townhouse | 3020000 | 4 |
| 5 NSW | Duplex | 1837000 | 5 |
| 6 NSW | Villa | 1589000 | 6 |
| 7 NSW | New House & Land | 588265 | 7 |

Report 4

(c)

SELECT

decode(GROUPING(s.suburb), 1, 'All_Suburbs', s.suburb) AS Suburb,

decode(GROUPING(t.year), 1, 'All_Years', t.year) AS Year,

decode(GROUPING(p.property_type), 1, 'All_Property_Types', p.property_type) AS

"Property Type",

SUM(r.rental_fees) AS "Total Rental Fees"

FROM

time_dim1 t,

suburb_dim1 s,

propertytype_dim1 p,

rentfact1 r

WHERE

s.suburb = r.suburb

AND t.time_id = r.time_id

AND p.property_type = r.property_type

GROUP BY CUBE

(s.suburb,

t.year,

p.property_type) ;

(d)

Query Result x

SQL | Fetched 50 rows in 0.03 seconds

| | SUBURB | YEAR | Property Type | Total Rental Fees |
|----|-------------|-----------|-------------------------------|-------------------|
| 1 | All_Suburbs | All_Years | All_Property_Types | 22816569 |
| 2 | All_Suburbs | All_Years | House | 8060088 |
| 3 | All_Suburbs | All_Years | Villa | 116503 |
| 4 | All_Suburbs | All_Years | Duplex | 150738 |
| 5 | All_Suburbs | All_Years | Studio | 92534 |
| 6 | All_Suburbs | All_Years | Terrace | 124340 |
| 7 | All_Suburbs | All_Years | Penthouse | 12071 |
| 8 | All_Suburbs | All_Years | Townhouse | 1851567 |
| 9 | All_Suburbs | All_Years | Semi-Detached | 113824 |
| 10 | All_Suburbs | All_Years | Apartment / Unit / Flat | 12272824 |
| 11 | All_Suburbs | All_Years | New Apartments / Off the Plan | 22080 |
| 12 | All_Suburbs | 2019 | All_Property_Types | 309600 |
| 13 | All_Suburbs | 2019 | House | 164520 |
| 14 | All_Suburbs | 2019 | Apartment / Unit / Flat | 145080 |
| 15 | All_Suburbs | 2020 | All_Property_Types | 22506969 |
| 16 | All_Suburbs | 2020 | House | 7895568 |
| 17 | All_Suburbs | 2020 | Villa | 116503 |

Report 5

(c)

```

SELECT
    decode(GROUPING(s.suburb), 1, 'All_Suburbs', s.suburb) AS Suburb,
    decode(GROUPING(t.year), 1, 'All_Years', t.year) AS Year,
    decode(GROUPING(p.property_type), 1, 'All_Property_Types', p.property_type) AS
"Property Type",
    SUM(r.rental_fees) AS "Total Rental Fees"
FROM
    time_dim1      t,
    suburb_dim1    s,
    propertytype_dim1 p,
    rentfact1      r
WHERE
    s.suburb = r.suburb
    AND t.time_id = r.time_id
    AND p.property_type = r.property_type
GROUP BY s.suburb,
CUBE
    (t.year,
    p.property_type) ;

```

(d)

Query Result x

SQL | Fetched 50 rows in 0.033 seconds

| | SUBURB | YEAR | Property Type | Total Rental Fees |
|----|--------|-----------|-------------------------|-------------------|
| 1 | City | All_Years | All_Property_Types | 295806 |
| 2 | City | All_Years | House | 12000 |
| 3 | City | All_Years | Apartment / Unit / Flat | 283806 |
| 4 | City | 2019 | All_Property_Types | 11520 |
| 5 | City | 2019 | Apartment / Unit / Flat | 11520 |
| 6 | City | 2020 | All_Property_Types | 284286 |
| 7 | City | 2020 | House | 12000 |
| 8 | City | 2020 | Apartment / Unit / Flat | 272286 |
| 9 | Cook | All_Years | All_Property_Types | 25440 |
| 10 | Cook | All_Years | House | 25440 |
| 11 | Cook | 2019 | All_Property_Types | 12480 |
| 12 | Cook | 2019 | House | 12480 |
| 13 | Cook | 2020 | All_Property_Types | 12960 |
| 14 | Cook | 2020 | House | 12960 |
| 15 | Holt | All_Years | All_Property_Types | 27919 |
| 16 | Holt | All_Years | House | 13279 |
| 17 | Holt | All_Years | Duplex | 14640 |

Report 6

(c)

SELECT

decode(GROUPING(s.season_id), 1, 'All_Seasons', s.season_id) AS Season,
 decode(GROUPING(st.state_code), 1, 'All_States', st.state_code) AS State,
 decode(GROUPING(sf.property_type), 1, 'All_Property_type', sf.property_type) AS

Property_type,

SUM(sf.total_sales) AS "Total Sales"

FROM

season_dim1 s,
 State_dim1 st,
 salesfact1 sf,
 propertytype_dim1 p

WHERE

s.season_id = sf.season_id
 AND st.state_code = sf.state_code
 AND p.property_type = sf.property_type

GROUP BY Rollup

(s.season_id,st.state_code, sf.property_type) ;

(d)

Query Result x

SQL | Fetched 50 rows in 0.017 seconds

| SEASON | STATE | PROPERTY_TYPE | Total Sales |
|-----------|-------|-------------------------|-------------|
| 1 AUTUMN | ACT | House | 18666000 |
| 2 AUTUMN | ACT | Townhouse | 3533000 |
| 3 AUTUMN | ACT | New House & Land | 979000 |
| 4 AUTUMN | ACT | Apartment / Unit / Flat | 9578000 |
| 5 AUTUMN | ACT | All_Property_type | 32756000 |
| 6 AUTUMN | NSW | House | 27108900 |
| 7 AUTUMN | NSW | Villa | 1589000 |
| 8 AUTUMN | NSW | Duplex | 749000 |
| 9 AUTUMN | NSW | Townhouse | 2590000 |
| 10 AUTUMN | NSW | Semi-Detached | 1944000 |
| 11 AUTUMN | NSW | Apartment / Unit / Flat | 13282900 |
| 12 AUTUMN | NSW | All_Property_type | 47263800 |
| 13 AUTUMN | QLD | House | 94241000 |
| 14 AUTUMN | QLD | Duplex | 725000 |
| 15 AUTUMN | QLD | Townhouse | 588000 |
| 16 AUTUMN | QLD | Block of Units | 639000 |
| 17 AUTUMN | QLD | Apartment / Unit / Flat | 26120000 |

Report 7

(c)

SELECT

decode(GROUPING(s.season_id), 1, 'All_Seasons', s.season_id) AS Season,
 decode(GROUPING(st.state_code), 1, 'All_States', st.state_code) AS State,
 decode(GROUPING(sf.property_type), 1, 'All_Property_type', sf.property_type) AS

Property_type,

SUM(sf.total_sales) AS "Total Sales"

FROM

season_dim1 s,
 State_dim1 st,
 salesfact1 sf,
 propertytype_dim1 p

WHERE

s.season_id = sf.season_id
 AND st.state_code = sf.state_code
 AND p.property_type = sf.property_type

GROUP BY s.season_id,

Rollup

(st.state_code,sf.property_type) ;

(d)

Query Result x

SQL | Fetched 50 rows in 0.018 seconds

| SEASON | STATE | PROPERTY_TYPE | Total Sales |
|-----------|-------|-------------------------|-------------|
| 1 AUTUMN | ACT | House | 18666000 |
| 2 AUTUMN | ACT | Townhouse | 3533000 |
| 3 AUTUMN | ACT | New House & Land | 979000 |
| 4 AUTUMN | ACT | Apartment / Unit / Flat | 9578000 |
| 5 AUTUMN | ACT | All_Property_type | 32756000 |
| 6 AUTUMN | NSW | House | 27108900 |
| 7 AUTUMN | NSW | Villa | 1589000 |
| 8 AUTUMN | NSW | Duplex | 749000 |
| 9 AUTUMN | NSW | Townhouse | 2590000 |
| 10 AUTUMN | NSW | Semi-Detached | 1944000 |
| 11 AUTUMN | NSW | Apartment / Unit / Flat | 13282900 |
| 12 AUTUMN | NSW | All_Property_type | 47263800 |
| 13 AUTUMN | QLD | House | 94241000 |
| 14 AUTUMN | QLD | Duplex | 725000 |
| 15 AUTUMN | QLD | Townhouse | 588000 |
| 16 AUTUMN | QLD | Block of Units | 639000 |
| 17 AUTUMN | QLD | Apartment / Unit / Flat | 26120000 |

Report 8

(c)

(d)

Report 9

(c)

```
select r.state_code, t.year, to_char(sum(rental_fees), '9,999,999,999') as "Q rent",
to_char(AVG(SUM(rental_fees)) OVER(order by r.state_code, t.year
rows 2 preceding),'9,999,999,999') as three_year_moving_average
from
rentfact1 r, time_dim1 t
where r.time_id=t.time_id and t.year in ('2018','2019','2020') and r.state_code in
('TAS','NSW','VIC')
group by r.state_code, t.year;
```

(d)

Query Result x

SQL | All Rows Fetched: 4 in 0.109 seconds

| STATE_CODE | YEAR | Q rent | THREE_YEAR_MOVING_AVERAGE |
|------------|------|-----------|---------------------------|
| 1 NSW | 2019 | 84,000 | 84,000 |
| 2 NSW | 2020 | 6,842,443 | 3,463,222 |
| 3 VIC | 2019 | 90,000 | 2,338,814 |
| 4 VIC | 2020 | 5,792,437 | 4,241,627 |

Report 10

(c)

(d)

Report 11

(c)

```

Select sf.property_type, t.client_year as Sales_Year, s.state_code,
to_char (sum(sf.total_sales)) as Sales,
Rank() over (partition by sf.property_type
order by sum (total_sales) desc) as Rank_by_property_type,
Rank() over (partition by t.client_year
order by sum (total_sales) desc) as Rank_by_year,
Rank() over (partition by s.state_code
order by sum (total_sales) desc) as Rank_by_state
From propertytype_dim1 p, time_dim2 t, state_dim1 s, salesfact1 sf
where
p.property_type=sf.property_type
and t.client_year=sf.client_year
and s.state_code=sf.state_code
group by sf.property_type,t.client_year, s.state_code;

```

(d)

Query Result x

SQL | Fetched 50 rows in 0.038 seconds

| PROPERTY_TYPE | SALES_YEAR | STATE_CODE | SALES | RANK_BY_PROPERTY_TYPE | RANK_BY_YEAR | RANK_BY_STATE |
|---------------------------|------------|------------|-----------|-----------------------|--------------|---------------|
| 1 Apartment / Unit / Flat | 2020 | QLD | 57619500 | 1 | 4 | 2 |
| 2 Apartment / Unit / Flat | 2020 | VIC | 32899700 | 2 | 6 | 2 |
| 3 Apartment / Unit / Flat | 2020 | NSW | 32025788 | 3 | 8 | 2 |
| 4 Apartment / Unit / Flat | 2020 | ACT | 22330800 | 4 | 9 | 2 |
| 5 Apartment / Unit / Flat | 2020 | WA | 5843000 | 5 | 14 | 2 |
| 6 Apartment / Unit / Flat | 2020 | SA | 3359000 | 6 | 17 | 2 |
| 7 Apartment / Unit / Flat | 2019 | ACT | 1074000 | 7 | 7 | 5 |
| 8 Apartment / Unit / Flat | 2019 | NSW | 439000 | 8 | 10 | 9 |
| 9 Block of Units | 2020 | QLD | 4329000 | 1 | 15 | 4 |
| 10 Development Site | 2020 | VIC | 1300000 | 1 | 27 | 5 |
| 11 Duplex | 2020 | NSW | 1837000 | 1 | 24 | 5 |
| 12 Duplex | 2020 | QLD | 1333000 | 2 | 26 | 8 |
| 13 Duplex | 2019 | QLD | 1100000 | 3 | 6 | 9 |
| 14 Duplex | 2020 | WA | 300000 | 4 | 35 | 7 |
| 15 House | 2020 | QLD | 167877900 | 1 | 1 | 1 |
| 16 House | 2020 | VIC | 119785499 | 2 | 2 | 1 |
| 17 House | 2020 | NSW | 63725850 | 3 | 3 | 1 |

Report 12

(c)

```

Select rf.property_type, s.season_id as season,
to_char (sum(rf.rental_fees)) as Rental_fees,
Rank() over (partition by rf.property_type

```

```

order by sum (rf.rental_fees) desc) as Rank_by_property_type,
Rank() over (partition by s.season_id
order by sum (rf.rental_fees) desc) as Rank_by_season
From propertytype_dim1 p, season_dim1 s, rentfact1 rf
where
p.property_type=rf.property_type
and s.season_id=rf.season_id
group by rf.property_type,s.season_id;

```

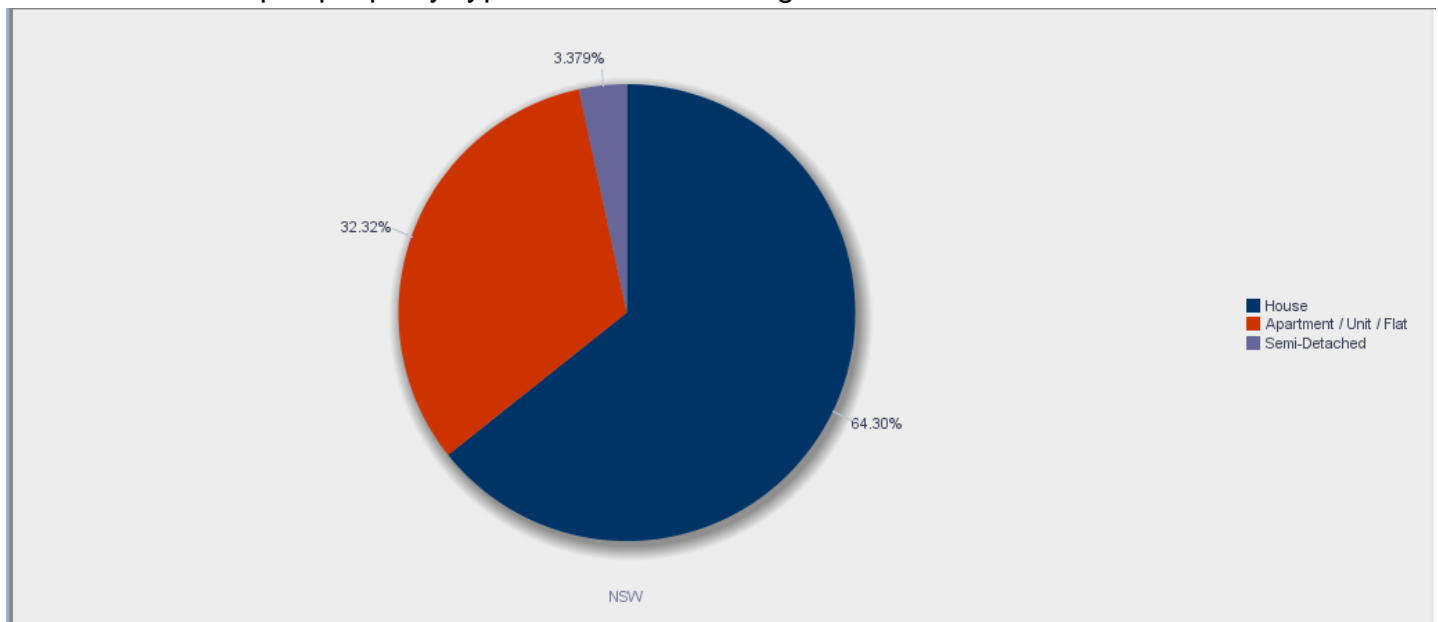
(d)

| PROPERTY_TYPE | SEASON | RENTAL_FEES | RANK_BY_PROPERTY_TYPE | RANK_BY_SEASON |
|---------------------------------|--------|-------------|-----------------------|----------------|
| 1 Apartment / Unit / Flat | AUTUMN | 6744741 | 1 | 1 |
| 2 Apartment / Unit / Flat | SUMMER | 5528083 | 2 | 1 |
| 3 Duplex | SUMMER | 90480 | 1 | 5 |
| 4 Duplex | AUTUMN | 60258 | 2 | 7 |
| 5 House | AUTUMN | 4934850 | 1 | 2 |
| 6 House | SUMMER | 3125238 | 2 | 2 |
| 7 New Apartments / Off the Plan | AUTUMN | 12480 | 1 | 9 |
| 8 New Apartments / Off the Plan | SUMMER | 9600 | 2 | 10 |
| 9 Penthouse | SUMMER | 12071 | 1 | 9 |
| 10 Semi-Detached | AUTUMN | 65824 | 1 | 5 |
| 11 Semi-Detached | SUMMER | 48000 | 2 | 7 |
| 12 Studio | AUTUMN | 71654 | 1 | 4 |
| 13 Studio | SUMMER | 20880 | 2 | 8 |
| 14 Terrace | AUTUMN | 64340 | 1 | 6 |
| 15 Terrace | SUMMER | 60000 | 2 | 6 |
| 16 Townhouse | AUTUMN | 1111719 | 1 | 3 |
| 17 Townhouse | SUMMER | 739848 | 2 | 3 |

TASK 4:

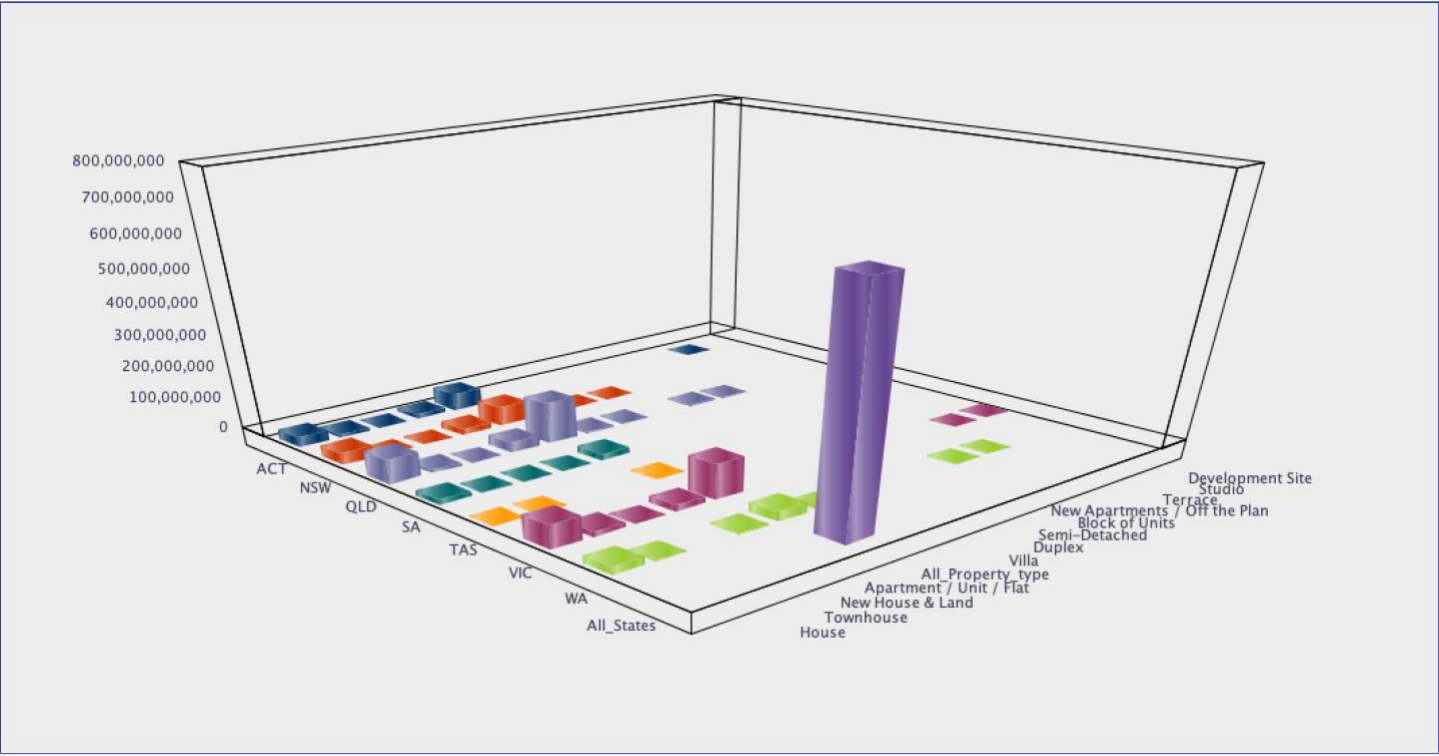
Report 1

Which are the top 3 property types in NSW with highest total sales



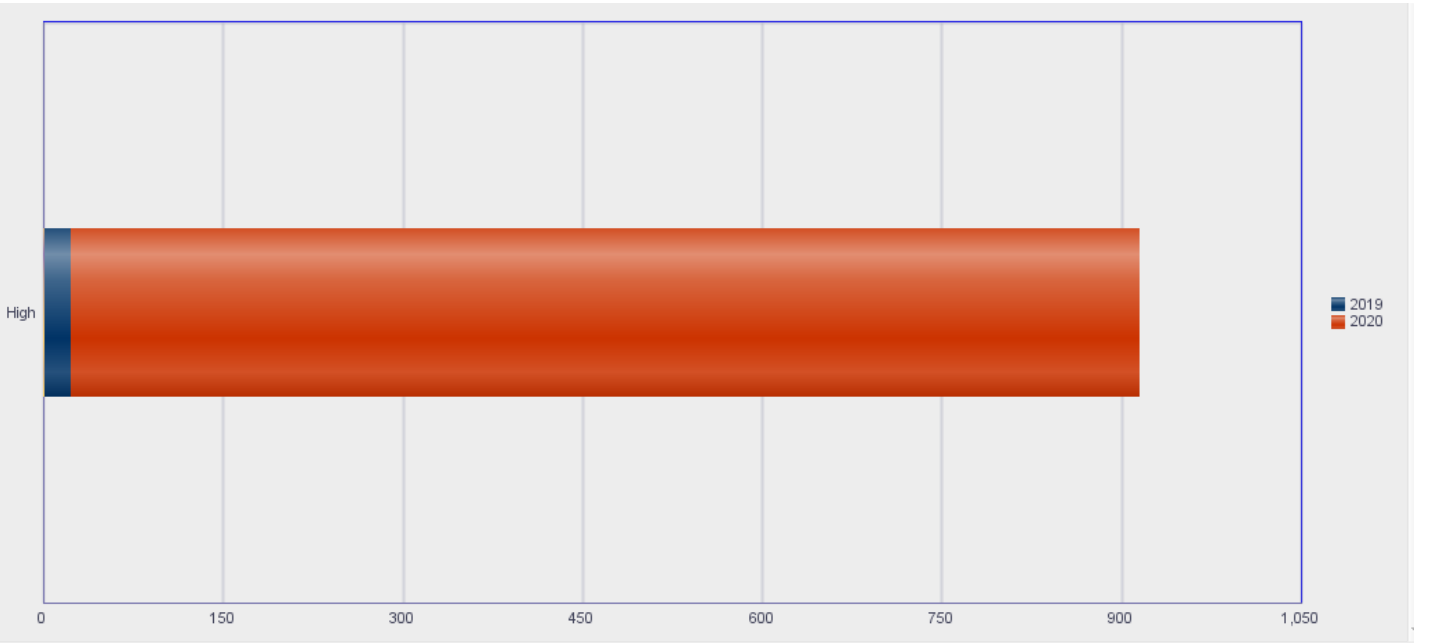
Report 2

sub-total and total sales based on property type and state



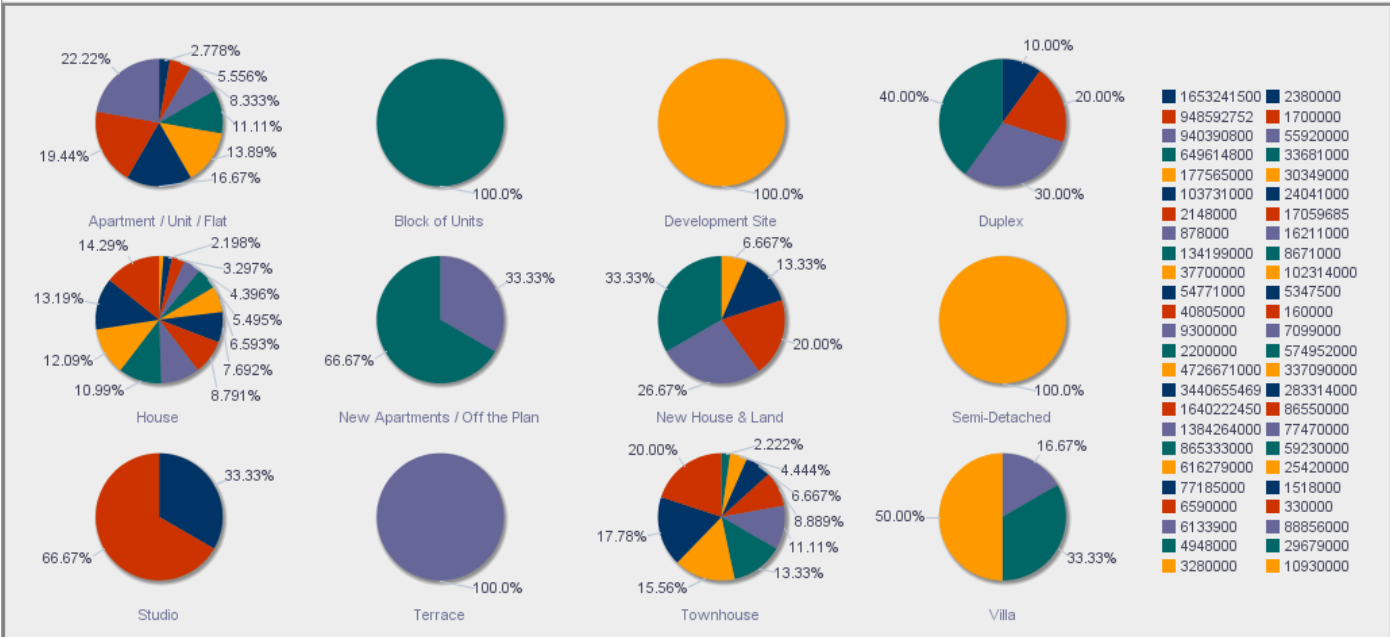
Report 3

Total number of clients and cumulative number of clients with a high budget in 2019 and 2020



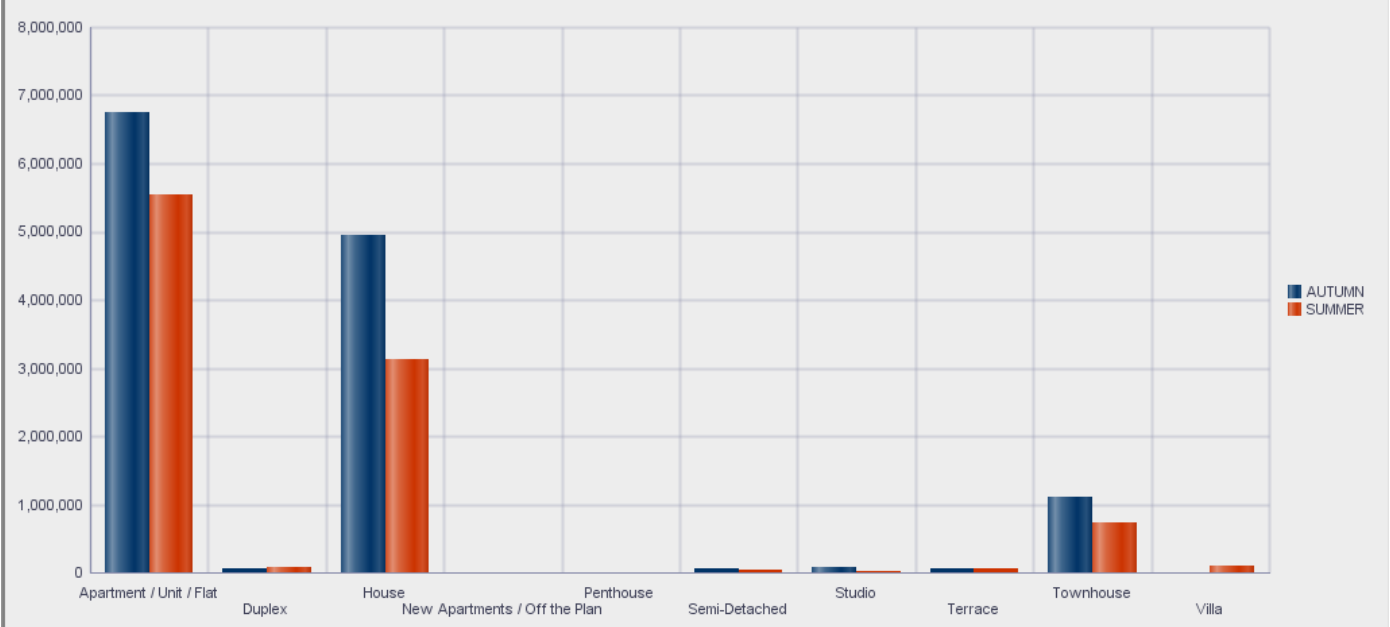
Report 4

Ranking of each property type based on yearly total number of sales



Report 5

Ranking of each property type based on total rental fees. This also shows the ranking of each season based on total rent



-----The End-----