## Output b

### b1. Address

SELECT COUNT(\*)

FROM mre\_address; -- 13,204 rows



Firstly, checked for duplicate primary key

SELECT COUNT(DISTINCT(address\_id))

FROM mre\_address; -- 13,204 rows

Secondly, checked for distinct street

SELECT COUNT(DISTINCT(street))

FROM mre\_address; -- 13,156 rows

Then observed rows with same street names

SELECT \*

FROM mre\_address

WHERE street IN (SELECT street FROM mre\_address GROUP BY street HAVING COUNT(\*) > 1) ORDER BY street;

From this, we noticed that a number of records have the same street but different suburb. As such, using the following, we were able to remove records with the same street, suburb, and postcode

DELETE FROM MRE\_Address

WHERE address\_id IN (

SELECT address\_id

FROM (SELECT address\_id,

street,

suburb,

postcode,

ROW\_NUMBER() OVER (PARTITION BY street, suburb, postcode ORDER BY street, suburb, postcode) as row\_num

FROM MRE\_Address) WHERE row\_num > 1); -- 43 rows deleted

Checking the number of rows again

SELECT COUNT(\*)

FROM MRE\_Address; -- 13,161 rows



### b2. Advertisement

SELECT COUNT(\*) FROM mre\_advertisement; -- 25 rows



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

SELECT COUNT(DISTINCT(advert\_id)) FROM mre\_advertisement; -- 25 rows

SELECT COUNT(DISTINCT(advert\_name)) FROM mre\_advertisement; -- 25 rows

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

### b3. Agent

SELECT COUNT(\*) FROM MRE\_Agent; -- 2,469 rows



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

SELECT COUNT(\*)

FROM (SELECT \* FROM MRE\_Person p, MRE\_Agent a

WHERE p.person\_id = a.person\_id); -- 2,468 rows

Using the following syntax, that row was deleted:

DELETE FROM MRE\_Agent WHERE NOT person\_id IN (SELECT person\_id FROM MRE\_person);

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

SELECT \* FROM mre\_agent WHERE salary < 0; -- 2 rows



SELECT \* FROM mre\_agent WHERE salary < 0; -- 2 rows deleted

Checking the number of rows again:

SELECT COUNT(\*) FROM MRE\_Agent; -- 2,466 rows



### b4. Agent\_Office

SELECT COUNT(\*) FROM mre\_agent\_office; -- 2,529 rows



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

SELECT COUNT(DISTINCT(person\_id)) FROM mre\_agent\_office; -- 2,467 rows

### b5. Client

SELECT COUNT(\*) FROM mre\_client; -- 3,339 rows



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

SELECT COUNT(\*) FROM mre\_person p, mre\_client c

WHERE p.person\_id = c.person\_id; -- 3,338 rows

As such, the extra client was deleted with

DELETE FROM mre\_client WHERE NOT person\_id IN (SELECT person\_id FROM mre\_person); -- 1 row deleted

Checking the number of rows again:

SELECT COUNT(\*) FROM mre\_client; -- 3,338 rows



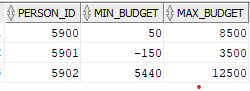
Checking the client budget:

SELECT \* FROM mre\_client WHERE min\_budget > max\_budget; -- 3 rows



The min budget should not more than max budget and must be more than or equal 0. Therefore, the min and max budget of these 3 rows should switch.

UPDATE mre\_client set max\_budget = min\_budget , min\_budget = max\_budget WHERE person\_id in(5900, 5901, 5902);



Then update the negative value to 0 for the person\_id 5901.

UPDATE mre\_client set min\_budget = 0 WHERE person\_id = 5901;



### b6. Client\_Wish

SELECT COUNT(\*) FROM mre\_client\_wish; -- 1,204 rows



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

### b7. Feature

SELECT COUNT(\*) FROM mre\_feature; -- 726 rows



Checking for duplicate feature\_id and feature\_description showed no duplicates.

### b8. Office

SELECT COUNT(\*) FROM mre\_office; -- 1,177 rows



Checking for duplicates of office\_id and office\_name showed no duplicates.

### b9. Person

SELECT COUNT(\*) FROM mre\_person; -- 7,000 rows



Looking for duplicates person\_id, 4 deplicates were found for person\_id = 6995

SELECT COUNT(DISTINCT(person\_id)) FROM MRE\_Person; -- 6,997 rows

SELECT person\_id FROM mre\_person GROUP BY person\_id HAVING COUNT(\*) > 1;



Using the following syntax, the duplicate records were deleted:

DELETE FROM MRE\_Person p

WHERE rowid > (SELECT MIN(rowid)FROM MRE\_Person p2

WHERE p.person\_id = p2.person\_id); -- 3 rows deleted

Checking through phone\_no and email showed no other duplicates, the final number is:

SELECT COUNT(\*) FROM mre\_person; -- 6,997 rows



### b10. Postcode

SELECT COUNT(\*) FROM mre\_postcode; -- 689 rows



No duplicate or nulls were found in the postcode table.

### b11. Property

SELECT COUNT(\*) FROM mre\_property; -- 6,226 rows



There were a large number of records that were duplicated

SELECT COUNT(DISTINCT(property\_id)) FROM mre\_property; -- 6,208 rows

The following syntax was used to delete the duplicate rows

DELETE FROM MRE\_Property p

WHERE rowid > (SELECT MIN(rowid)FROM MRE\_Property p2

WHERE p.property\_id = p2.property\_id); -- 18 rows deleted

The new number of rows is:

SELECT COUNT(\*) FROM mre\_property; -- 6,208 rows



### b12. Property\_Advert

SELECT COUNT(\*) FROM mre\_property\_advert; -- 3,646 rows



No duplicate records or null records were found in this table

### b13. Property\_Feature

SELECT COUNT(\*) FROM mre\_property\_feature; -- 30,373 rows



No duplicate or null records were found

### b14. Rent

SELECT COUNT(\*) FROM mre\_rent; --3,284 rows



Records were checked for duplicate and non were found but one record was found for rent\_end\_date before rent\_start\_date

SELECT \* FROM mre\_rent WHERE rent\_end\_date <= rent\_start\_date;



The record was then deleted

DELETE FROM MRE\_Rent WHERE rent\_id IN (SELECT rent\_id FROM MRE\_Rent WHERE rent\_end\_date < rent\_start\_date); -- 1 row deleted

Checking again

SELECT COUNT(\*) FROM MRE\_Rent; -- 3,283 rows



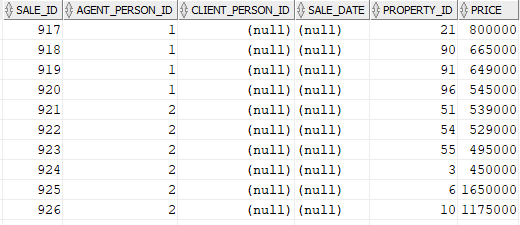
### b15. Sale

SELECT COUNT(\*) FROM mre\_sale; -- 2,925 rows



The following syntax was used and found that a number of records had null client\_person\_id and sale\_date and was deleted

SELECT \* FROM MRE\_Sale WHERE client\_person\_id IS NULL OR sale\_date IS NULL; -- 2,009 rows



DELETE FROM MRE\_Sale WHERE client\_person\_id IS NULL OR sale\_date IS NULL; -- 2,009 rows deleted

The new number of records is as follows:

SELECT COUNT(\*) FROM mre\_sale; -- 916 rows



### b16. State

SELECT \* FROM mre\_state; -- 9 rows

There were little number of rows so visual inspection was possible and one row was identified as NULL state\_code and UNKNOWN state\_name which was promptly deleted

DELETE FROM MRE\_State WHERE state\_code IS NULL; -- 1 row deleted

New number of records is:

SELECT COUNT(\*) FROM mre\_state; -- 8 rows



### b17. Visit

SELECT COUNT(\*) FROM mre\_visit; -- 575 rows



A record was found that the agent or client did not exist in mre\_agent or mre\_client

SELECT \* FROM mre\_visit WHERE NOT agent\_person\_id IN (SELECT person\_ID FROM mre\_agent) OR NOT client\_person\_id IN (SELECT person\_id FROM mre\_client); -- 1 row



That record was promptly deleted

DELETE FROM mre\_visit WHERE NOT agent\_person\_id IN (SELECT person\_ID FROM mre\_agent) OR NOT client\_person\_id IN (SELECT person\_id FROM mre\_client); -- 1 row deleted

Updated number of rows is:

SELECT COUNT(\*) FROM mre\_visit; -- 574 rows

