## Output b

### b1. Address

SELECT COUNT(\*)

FROM mre\_address; -- 13,204 rows



Firstly, checked for duplicate address\_id

SELECT COUNT(DISTINCT(address\_id))

FROM mre\_address; -- 13,204 rows

Secondly, checked for distinct street

SELECT COUNT(DISTINCT(street))

FROM mre\_address; -- 13,156 rows

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

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

SELECT address\_id

FROM mre\_address

WHERE NOT address\_id IN (SELECT address\_id FROM

mre\_property)

AND NOT address\_id IN (SELECT address\_id FROM mre\_person);

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

### b2. Advertisement

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



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

SELECT \*

FROM mre\_person

WHERE NOT person\_id IN (SELECT person\_id FROM mre\_person);



Using the following syntax, that row was deleted:

DELETE FROM MRE\_Agent

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

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

SELECT \* FROM mre\_agent

WHERE salary < 0; -- 2 rows



DELETE 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

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

SELECT \*

FROM mre\_agent\_office

WHERE NOT person\_id IN (SELECT person\_id FROM mre\_agent);



DELETE FROM mre\_agent\_office

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

The number of rows in the end is

SELECT COUNT(\*)

FROM mre\_agent\_office; -- 2,528 rows



### 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

SELECY \*

FROM mre\_client

WHERE NOT person\_id IN (SELECT person\_id FROM mre\_person);



As such, the extra client was deleted with

DELETE FROM mre\_client

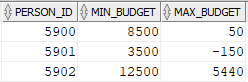
WHERE NOT person\_id IN (SELECT person\_id FROM mre\_person); -- 1 row deleted

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

SELECT \*

FROM mre\_client

WHERE max\_budget < min\_budget; -- 3 rows



DELETE FROM mre\_client

WHERE max\_budget < min\_budget; -- 3 rows deleted

Checking for any min\_budget or max\_budget being negative yield no results.

Checking the number of rows again:

SELECT COUNT(\*)

FROM mre\_client; -- 3,338 rows



### 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

SELECT \*

FROM mre\_property p

WHERE rowid > (SELECT MIN(rowid)FROM mre\_property p2

WHERE p.property\_id = p2.property\_id);



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

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. However, as these rows may be kept for record to show agent client relationships for unsold properties, the rows are temporarily kept.

### b16. State

SELECT \*

FROM mre\_state; -- 9 rows

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

DELETE FROM MRE\_State

WHERE state\_code IS NULL; -- 1 row deleted

New number of records is:

SELECT COUNT(\*)

FROM mre\_state; -- 8 rows



### 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



### b18. Special Case

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

SELECT \*

FROM mre\_person

WHERE NOT address\_id IN (SELECT address\_id FROM mre\_address);



As a safety precaution, we checked through Agent, Client, Client\_Wish to this person\_id.

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



We continued to check the Client\_Wish table and found one record of this person:



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



After taking the above-mentioned factors into consideration, we decided to delete person\_id = 7001 from the Person, Client and Client\_Wish table as well as feature\_code = 726 from the feature table.

### b19. Summary

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

1. Agent table person\_id NOT IN Person table person\_id – 1 row deleted
2. Agent salary < 0 – 2 rows deleted
3. Agent person\_id NOT IN Person table person\_id – 1 row deleted
4. Client table person\_id NOT IN Person table person\_id – 1 row deleted
5. Client max\_budget < min\_budget – 3 rows deleted
6. Duplicate person record – 3 rows deleted
7. Duplicate property record – 18 rows deleted
8. Rent rent\_end\_date < rent\_start\_date – 1 row deleted
9. State state\_code IS NULL – 1 row deleted
10. Visit table agent\_person\_id or client\_person\_id NOT IN Person table person\_id – 1 row deleted
11. Special Case: Person address\_id non-existent – 1 row deleted

Client record referencing above – 1 row deleted

Client\_wish record referencing above – 1 row deleted

Feature record ‘Fake Feature’ – 1 row deleted