So, this to be stated that the idea behind each and every of these solutions are my own I have only used resources from ChatGPT to refine and work on my weakness. Also the results could be achieved from a different queries.

Ans.1- So, we have multiple ways to get this query resolved. I am taking the most simple way according to me. We simply get two queries and run them one inside the other. Basically, getting the profile\_id from tenancy\_history and then using it in the outer quary to get the required details from the profiles.

SELECT CONCAT(first\_name, ' ', last\_name) AS full\_name, profile\_id, phone FROM Profiles

where profile\_id in

(

select top 1 profile\_id

from tenancy\_history

order by Datediff(Day, move\_in\_date, move\_out\_date) DESC

)

Ans.2- We can have simple two subqueries running like….

SELECT CONCAT(first\_name, ' ', last\_name) AS full\_name, profile\_id, phone FROM Profiles

where profile\_id in

(

select profile\_id from Tenancy\_History

where rent> 9000

)

AND marital\_status <>'N'

Ans.3-

Ans.4- select CONCAT(first\_name,'',last\_name) Name,

email\_id, phone,city,referral\_code

from Profiles

where profile\_id in

(

select profile\_id

from Referral

group by profile\_id

having Count(profile\_id)>1

)

select sum(referrer\_bonus\_amount) Total\_bonus from Referral

where referral\_valid=1 and profile\_id in

(

select profile\_id

from Referral

group by profile\_id

having Count(profile\_id)>1

)

Again here whe need to get the two outputs in a single query.

Ans.5- So here are two different approachs to the solution

**Case A:**

SELECT distinct city,

SUM(rent) OVER (PARTITION BY city) AS City\_wise\_rent

FROM Tenancy\_History

FULL OUTER JOIN

Addresses ON Tenancy\_History.house\_id = Addresses.house\_id

GROUP BY

city, rent;

SELECT SUM(rent) total\_rent

FROM Tenancy\_History

The above will get us the requested result but is not the ideal single query output.

Case B

Ans.6- CREATE VIEW VW\_tennant AS

select Tenancy\_History.profile\_id,

Tenancy\_History.rent,

Tenancy\_History.move\_in\_date,

Houses.house\_type,

Houses.beds\_vacant,

Addresses.description,

Addresses.city

from Tenancy\_History

JOIN Houses ON Tenancy\_History.house\_id=Houses.house\_id

JOIN Addresses ON Houses.house\_id=Addresses.house\_id

where beds\_vacant >0;

select \* from VW\_tennant;

Ans.7-

select distinct(profile\_id),valid\_till as old\_date, DATEADD(month,1, valid\_till) as extended\_date

from Referral

where profile\_id in

(

SELECT profile\_id

FROM Referral

GROUP BY profile\_id

HAVING COUNT(profile\_id) > 1

)

Ans.8-

ALTER TABLE tenancy\_history

ADD customer\_segment VARCHAR(8);

UPDATE Tenancy\_History

SET customer\_segment =

CASE

WHEN rent > 10000 THEN 'GRADE A'

WHEN rent BETWEEN 7500 AND 10000 THEN 'GRADE B'

ELSE 'GRADE C'

END;

select profiles.profile\_id,

CONCAT(first\_name,' ', last\_name) Full\_Name,

phone,

customer\_segment

FROM Profiles

JOIN Tenancy\_History ON

Profiles.profile\_id=Tenancy\_History.profile\_id;

Ans.9-

select CONCAT(first\_name,'',last\_name) Name,

email\_id, phone,city

from Profiles

where profile\_id in

(

select profile\_id

from Referral

where referral\_valid=0

)

Ans.10-

To get this ansdwer we use the houses and tenancy history tables, such as we get the details of the house from the house table but we get the house\_id from the tenancy history table of the house with the longest time people staying in it, by an inspection we can check that it is the house with id 14 but to make a more sustainable query we use the MIN function.

select \* from Houses where house\_id =

(

select house\_id from Tenancy\_History where move\_in\_date= (select MIN(move\_in\_date) from Tenancy\_History)

)