-----------------------------------BASIC QUERIES----------------------------------

----Query 1: Select all columns and all rows from one table

SELECT \* FROM tenant;

A screenshot of a computer

Description automatically generated

----------------------------------------------------------------------------------

----Query 2: Select five columns and all rows from one table

SELECT apt\_id

,apt\_sqft

,apt\_bednum

,apt\_bathnum

,apt\_garageid

FROM apartment;

A screenshot of a computer

Description automatically generated

----------------------------------------------------------------------------------

----Query 3: Select all columns from all rows from one view

CREATE OR REPLACE VIEW

staff\_info AS

SELECT \* FROM staff;

SELECT \* FROM staff\_info;

A screenshot of a computer

Description automatically generated

----------------------------------------------------------------------------------

----Query 4: Using a join on 2 tables,

----select all columns and all rows from the tables without the use of a Cartesian product

SELECT \* FROM staff s

INNER JOIN request r

ON s.Stf\_ID = r.Stf\_ID

A screenshot of a computer

Description automatically generated

----------------------------------------------------------------------------------

----Query 5: Select and order data retrieved from one table

SELECT \* FROM REQUEST

ORDER BY Req\_IssueDate desc

A screenshot of a computer

Description automatically generated

----------------------------------------------------------------------------------

----Query 6: Using a join on 3 tables, select 5 columns from the 3 tables.

----Use syntax that would limit the output to 10 rows

SELECT a.Apt\_ID

,a.Apt\_Sqft

,t.Ten\_ID

,t.Ten\_FName

,l.les\_startDate

FROM apartment a

INNER JOIN lease l

ON a.apt\_id = l.apt\_id

INNER JOIN tenant t

ON t.ten\_id = l.ten\_id

LIMIT 10;

A screenshot of a computer

Description automatically generated

----------------------------------------------------------------------------------

----Query 7: Select distinct rows using joins on 3 tables

SELECT DISTINCT(s.Stf\_ID)

,a.apt\_id

,r.req\_issuedate

,r.req\_type

,r.req\_status

,r.req\_finishdate

FROM apartment a

INNER JOIN request r

ON a.apt\_id = r.apt\_id

INNER JOIN staff s

ON s.Stf\_ID = r.Stf\_ID

A screenshot of a computer

Description automatically generated

----------------------------------------------------------------------------------

----Query 8: Use GROUP BY and HAVING in a select statement using one or more tables

----Find Apartments that have raised multiple maintenance requests

SELECT apartment.apt\_id

,COUNT(request.req\_status) AS pending\_requests

FROM apartment

LEFT JOIN request

ON apartment.apt\_id = request.apt\_id

GROUP BY apartment.apt\_id

HAVING COUNT(request.req\_status) >1;

A screenshot of a computer

Description automatically generated

----------------------------------------------------------------------------------

----Query 9: Use IN clause to select data from one or more tables

SELECT \* FROM PAYMENT p

where p.les\_id IN

(SELECT l.les\_id FROM LEASE l

where l.les\_StartDate >= '2023-01-01'

AND l.les\_StartDate <= '2023-10-31')

AND p.Pay\_Method = 'Check'

A screenshot of a computer

Description automatically generated

----------------------------------------------------------------------------------

----Query 10: Select length of one column from one table (use LENGTH function)

SELECT S.Stf\_ID

,S.Stf\_FName

,LENGTH(S.Stf\_FName)

FROM STAFF S

A screenshot of a cellphone

Description automatically generated

----------------------------------------------------------------------------------

----Query 11:Delete one record from one table.

----Use select statements to demonstrate the table contents before and after the DELETE statement.

----Make sure you use ROLLBACK afterwards so that the data will not be physically removed

SELECT \* FROM PAYMENT P;

A screenshot of a computer

Description automatically generated

BEGIN;

DELETE FROM PAYMENT P

where P.les\_ID = 1

AND P.pay\_date = '2022-07-01';

SELECT \* FROM PAYMENT P

A screenshot of a computer

Description automatically generated

ROLLBACK;

END;

SELECT \* FROM PAYMENT P

A screenshot of a computer

Description automatically generated

----------------------------------------------------------------------------------

----Query 12: Update one record from one table.

----Use select statements to demonstrate the table contents before and after the UPDATE statement.

----Make sure you use ROLLBACK afterwards so that the data will not be physically removed

SELECT \* FROM STAFF S order by S.stf\_id;

A screenshot of a computer

Description automatically generated

BEGIN;

UPDATE STAFF SET stf\_role = 'Assistant Manager'

where stf\_id = 2;

SELECT \* FROM STAFF S order by S.stf\_id;

A screenshot of a computer

Description automatically generated

ROLLBACK;

END;

SELECT \* FROM STAFF S order by S.stf\_id;

A screenshot of a computer

Description automatically generated

-----------------------------------ADVANCE QUERIES----------------------------------

----1. Calculate the total payments made by each tenant for the year 2023, including fines.

----Business Implication: This query helps in understanding the financial contributions of

----each tenant for the year 2023, including lease payments and fines.

SELECT tenant.ten\_id

,ten\_fname

,ten\_lname

,SUM(pay\_amount + pay\_fine) AS total\_payments\_2023

FROM tenant

LEFT JOIN lease

ON tenant.ten\_id = lease.ten\_id

LEFT JOIN payment

ON lease.les\_id = payment.les\_id

WHERE EXTRACT(YEAR FROM pay\_date) = 2023

GROUP BY tenant.ten\_id

ORDER BY total\_payments\_2023 DESC;

A screenshot of a computer

Description automatically generated

----------------------------------------------------------------------------------

----2. Find the staff member who has completed the most requests, including In-Progress and completed requests.

----Business Implication: This query helps identify the most productive staff member in handling service requests, which can be used for performance evaluation.

SELECT staff.stf\_id

,stf\_fname

,stf\_lname

,COUNT(request.req\_issuedate) AS total\_requests

FROM staff

LEFT JOIN request

ON staff.stf\_id = request.stf\_id

WHERE (request.req\_status ILIKE ('Finished')

OR

request.req\_status ILIKE ('In-Progress'))

GROUP BY staff.stf\_id

ORDER BY total\_requests DESC

LIMIT 1;

A close up of a white background

Description automatically generated with medium confidence