Create table

1)

```
SELECT * FROM capstone.table1;
CREATE TABLE table1 (
  Sno INT,
  Address VARCHAR(255),
  City VARCHAR(255),
  Rental_Price int,
  deposit DECIMAL,
);
```

2)

```
CREATE TABLE table2 (
  Sno INT,
  no_of_bed INT,
  no_of_bathroom INT,
  Area VARCHAR(255),
  pets_allowed BOOLEAN
);
```

3)

```
CREATE TABLE table1 (
  Sno INT,
  Washer_Dryer BOOLEAN,
  AC BOOLEAN,
  Parking BOOLEAN,
  Fireplace BOOLEAN,
  Dishwasher BOOLEAN,
  Hardwood_floors BOOLEAN,
  Roofdeck BOOLEAN,
  Storage BOOLEAN
);
```

****

TABLE 1

1) Write a SQL query to order records by a rental price column in ascending order.

```
SELECT *
FROM your_table
ORDER BY rental_price ASC;
```

2) Write a SQL query to select unique combinations of City and State with their average Rental Price

```
SELECT City, AVG(Rental_Price) AS Average_Rental_Price
FROM your_table
GROUP BY City;
```

3) Write a SQL query to select the top 5 highest deposit amounts with corresponding Address and City

```
SELECT Address, City, Maintenance_Amount
FROM your_table
ORDER BY Maintenance_Amount DESC
LIMIT 5;
```

4) Write a SQL query to select the count of records for each Country along with the total deposit amount

```
SELECT City, COUNT(*) AS Record_Count, SUM(Maintenance_Amount) AS
Total_Maintenance_Amount
```

```
FROM your_table

GROUP BY City;
```

5) Write a SQL query to select records with a Rental Price higher than the average Rental Price across all records

```
SELECT *

FROM your_table

WHERE Rental_Price > (SELECT AVG(Rental_Price) FROM your_table);
```

TABLE 2

1) Write a SQL query to select the average area for each number of bedrooms.

```
SELECT No_of_bed,

    AVG(Area) AS Average_Area

FROM your_table

GROUP BY No_of_bed;
```

2) Write a SQL query to select records with more than one bathroom and pets allowed

```
SELECT *

FROM your_table

WHERE No_of_bathroom > 1;
```

3) Write a SQL query to select the top 3 records with the highest total area (bedrooms + bathrooms)

```
SELECT *,

    (No_of_bed * Bedroom_Area + No_of_bathroom * Bathroom_Area) AS Total_Area

FROM your_table

ORDER BY Total_Area DESC

LIMIT 3;
```

4) Write a SQL query to select the count of records for each combination of bedrooms and bathrooms

```
SELECT No_of_bed, No_of_bathroom, COUNT(*) AS Record_Count
FROM your_table
GROUP BY No_of_bed, No_of_bathroom;
```

5) Write a SQL query to select records with the largest area where pets are allowed

```
SELECT *
FROM your_table
WHERE Pets_allowed = 'Yes'
ORDER BY Area DESC
LIMIT 1;
```

TABLE 3

1) Write a SQL query to Select records where both Washer/Dryer and AC are available, and order by Sno.

```
SELECT *
FROM your_table
WHERE Furnishing = 'available'
ORDER BY Sno;
```

2) Write a SQL query to Select records where Hardwood floors are available but neither Roofdeck nor Storage is present, and order by Sno in descending order

```
SELECT *
FROM your_table
WHERE Total_Floors IS NOT NULL
ORDER BY Sno DESC;
```

3) Write a SQL query to Select records where at least four amenities (AC, Parking, Dishwasher, Fireplace) are available, and order by Sno

```
SELECT *
FROM your_table
```

```
WHERE AC = 'available'
  AND Parking = 'available'
  AND Dishwasher = 'available'
  AND Fireplace = 'available'
ORDER BY Sno;
```

4) Write a SQL query to Select records where neither Roofdeck nor Storage is available, and include the count of such records.

```
SELECT COUNT(*) AS Record_Count
FROM your_table
WHERE Furnishing != 'available'
  AND Furnishing != 'Not available';
```

5) Write a SQL query to Select records with Parking and either Fireplace or Dishwasher, and include the count of records for each condition

```
SELECT
    SUM(CASE WHEN Parking = 'available' THEN 1 ELSE 0 END) AS Parking_Count,
    SUM(CASE WHEN Fireplace = 'available' THEN 1 ELSE 0 END) AS Fireplace_Count,
    SUM(CASE WHEN Dishwasher = 'available' THEN 1 ELSE 0 END) AS Dishwasher_Count
FROM
    your_table
WHERE
    Parking = 'available'
    AND (Fireplace = 'available' OR Dishwasher = 'available');
```

Join SQL Queries using all 3 tables

1) Write a SQL subquery to find records with more than the average area and related details using table 1 and table 2

```
SELECT t1.*, t2.Area
FROM Table1 t1
JOIN Table2 t2 ON t1.Property_ID = t2.Property_ID
WHERE t2.Area > (SELECT AVG(Area) FROM Table2);
```

2) Write a subquery to find records in table1 based on conditions pets allowed is 'YES' and no of bed is greater than 3 in table2

```
SELECT *

FROM Table1

WHERE Property_ID IN (

   SELECT t2.Property_ID

   FROM Table2 t2

   WHERE t2.Pets_allowed = 'YES'

   AND t2.No_of_bed > 3

);
```

3) Write a SQL subquery using both tables (2 and 3) to find records in Table2 with more than 2 bedrooms and related details from Table3 where AC is present

```
SELECT t2.*, t3.*

FROM Table2 t2

JOIN Table3 t3 ON t2.Property_ID = t3.Property_ID

WHERE t2.No_of_bed > 2

AND t3.AC = 'present';
```

4) Write a sql subquery to find records in Table2 with pets allowed and a Dishwasher, and include related details from Table3

```
SELECT t3.*

FROM Table3 t3

JOIN (

   SELECT t2.Property_ID

   FROM Table2 t2

   WHERE t2.Pets_allowed = 'Yes'

   AND t2.Dishwasher = 'available'

) subquery

ON t3.Property_ID = subquery.Property_ID;
```

5) Write a subquery to find records in Table2 with the highest area and related details from Table3 where roofdeck is present.

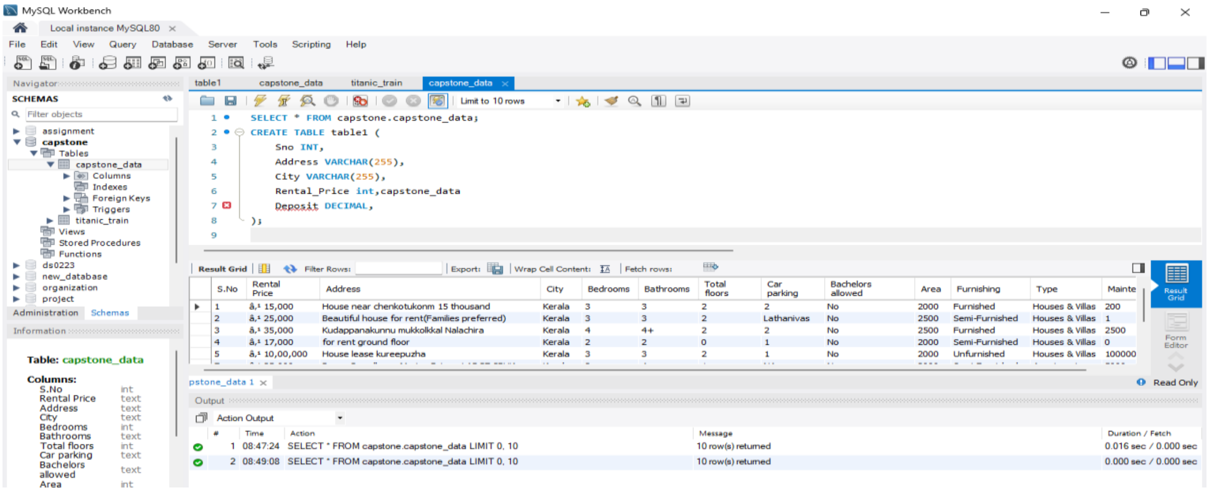
```
SELECT t3.*
FROM Table3 t3
JOIN (
  SELECT t2.Property_ID
  FROM Table2 t2
  WHERE t2.Area = (SELECT MAX(Area) FROM Table2)
  AND t2.Roofdeck = 'present'
) subquery
ON t3.Property_ID = subquery.Property_ID;
```

6) Write a sql Inner Join to combine information from table1 and table 2

```
SELECT t1.*, t2.*
FROM Table1 t1
INNER JOIN Table2 t2 ON t1.Property_ID = t2.Property_ID;
```

7) Write SQL Subquery to find records in table1 with pets allowed and a Washer/Dryer, and include details from table2 and table3

```
SELECT t2.*, t3.*
FROM Table2 t2
JOIN Table3 t3 ON t2.Property_ID = t3.Property_ID
WHERE t2.Property_ID IN (
  SELECT t1.Property_ID
  FROM Table1 t1
  WHERE t1.Pets_allowed = 'Yes'
  AND t1.Washer_Dryer = 'available'
);
```



```
SELECT * FROM capstone.capstone_data;
CREATE TABLE table1 (
    Sno INT,
    Address VARCHAR(255),
    City VARCHAR(255),
    Rental_Price int,capstone_data
    Deposit DECIMAL,
);
```

Table: capstone_data

Columns:

S.No	int
Rental Price	text
Address	text
City	text
Bedrooms	int
Bathrooms	text
Total floors	int
Car parking	text
Bachelors allowed	text
Area	int

	S.No	Rental Price	Address	City	Bedrooms	Bathrooms	Total floors	Car parking	Bachelors allowed	Area	Furnishing	Type	Mainte
▶	1	â‚¹ 15,000	House near chenkotukonm 15 thousand	Kerala	3	3	2	2	No	2000	Furnished	Houses & Villas	200
	2	â‚¹ 25,000	Beautiful house for rent(Families preferred)	Kerala	3	3	2	Lathanivas	No	2500	Semi-Furnished	Houses & Villas	1
	3	â‚¹ 35,000	Kudappanakunnu mukkolkkal Nalachira	Kerala	4	4+	2	2	No	2500	Furnished	Houses & Villas	2500
	4	â‚¹ 17,000	for rent ground floor	Kerala	2	2	0	1	No	2000	Semi-Furnished	Houses & Villas	0
	5	â‚¹ 10,00,000	House lease kureepuzha	Kerala	3	3	2	1	No	2000	Unfurnished	Houses & Villas	100000

	#	Time	Action	Message	Duration / Fetch
✓	1	08:47:24	SELECT * FROM capstone.capstone_data LIMIT 0, 10	10 row(s) returned	0.016 sec / 0.000 sec
✓	2	08:49:08	SELECT * FROM capstone.capstone_data LIMIT 0, 10	10 row(s) returned	0.000 sec / 0.000 sec