

```
In [1]: import sqlite3
import pandas as pd

# Example DataFrames
table1_df = pd.DataFrame({
    'Sno': [1, 2, 3],
    'Rental_Price': [1000, 1500, 1200],
    'City': ['A', 'B', 'C'],
    'State_Code': ['CA', 'NY', 'TX'],
    'Address': ['123 Main St', '456 Elm St'],
    'Deposit': [500, 700, 600],
    'Country': ['USA', 'Canada', 'USA']
})

table2_df = pd.DataFrame({
    'Sno': [1, 2, 3],
    'No_of_Bed': [2, 3, 1],
    'No_of_Bathroom': [1, 2, 1],
    'Pets_Allowed': ['Yes', 'No', 'Yes'],
    'Area': [800, 1200, 600]
})

table3_df = pd.DataFrame({
    'Sno': [1, 2, 3],
    'Washer_Dryer': ['Yes', 'No', 'Yes'],
    'AC': ['No', 'Yes', 'Yes'],
    'Hardwood_Floors': ['Yes', 'Yes', 'No'],
    'Roofdeck': ['No', 'Yes', 'No'],
    'Storage': ['No', 'No', 'Yes'],
    'Parking': ['Yes', 'No', 'Yes'],
    'Dishwasher': ['Yes', 'No', 'Yes'],
    'Fireplace': ['No', 'Yes', 'Yes']
})

# Step 1: Create SQLite connection
con = sqlite3.connect(':memory:')

# Step 2: Create Tables and Populate
table1_df.to_sql('Table1', con, index=False, if_exists='replace')
table2_df.to_sql('Table2', con, index=False, if_exists='replace')
table3_df.to_sql('Table3', con, index=False, if_exists='replace')

# Table1 Queries
print("Table1 Queries:")
avg_rental_price = pd.read_sql_query("SELECT AVG(Rental_Price) as avg_rental_price FROM Table1;", con)['avg_rental_price'][0]

# Query 1
query1_result = pd.read_sql_query(f"SELECT * FROM Table1 ORDER BY Rental_Price;", con)

# Query 2
query2_result = pd.read_sql_query("SELECT City, State_Code, AVG(Rental_Price) as avg_rental_price FROM Table1 GROUP BY City, State_Code;", con)

# Query 3
query3_result = pd.read_sql_query("SELECT Address, City, Deposit FROM Table1 ORDER BY Deposit DESC LIMIT 5;", con)

# Query 4
query4_result = pd.read_sql_query("SELECT Country, COUNT(Sno) as Record_Count, SUM(Deposit) as Total_Deposit FROM Table1 GROUP BY Country;", con)

# Query 5
query5_result = pd.read_sql_query(f"SELECT * FROM Table1 WHERE Rental_Price > {avg_rental_price};", con)

# Display Results for Table1 Queries
for i, result in enumerate([query1_result, query2_result, query3_result, query4_result, query5_result], start=1):
    print(f"Query {i}:")
    print(result)
    print()

# Table2 Queries
print("Table2 Queries:")

# Query 1
query1_result = pd.read_sql_query("SELECT No_of_Bed, AVG(Area) as avg_area FROM Table2 GROUP BY No_of_Bed;", con)

# Query 2
query2_result = pd.read_sql_query("SELECT * FROM Table2 WHERE No_of_Bathroom > 1 AND Pets_Allowed = 'Yes';", con)

# Query 3
query3_result = pd.read_sql_query("SELECT No_of_Bed, No_of_Bathroom, (No_of_Bed + No_of_Bathroom) as Total_Area FROM Table2 ORDER BY Total_Area DESC LIMIT 3;", con)

# Query 4
query4_result = pd.read_sql_query("SELECT No_of_Bed, No_of_Bathroom, COUNT(*) as Record_Count FROM Table2 GROUP BY No_of_Bed, No_of_Bathroom;", con)

# Query 5
query5_result = pd.read_sql_query("SELECT * FROM Table2 WHERE Pets_Allowed = 'Yes' ORDER BY Area DESC LIMIT 1;", con)

# Display Results for Table2 Queries
for i, result in enumerate([query1_result, query2_result, query3_result, query4_result, query5_result], start=1):
    print(f"Query {i}:")
    print(result)
    print()

# Table3 Queries
print("Table3 Queries:")

# Query 1
query1_result = pd.read_sql_query("SELECT * FROM Table3 WHERE Washer_Dryer = 'Yes' AND AC = 'Yes' ORDER BY Sno;", con)

# Query 2
query2_result = pd.read_sql_query("SELECT * FROM Table3 WHERE Hardwood_Floors = 'Yes' AND Roofdeck = 'No' AND Storage = 'No' ORDER BY Sno DESC;", con)

# Query 3
query3_result = pd.read_sql_query("SELECT * FROM Table3 WHERE AC = 'Yes' AND Parking = 'Yes' AND Dishwasher = 'Yes' AND Fireplace = 'Yes' ORDER BY Sno;", con)

# Query 4
query4_result = pd.read_sql_query("SELECT COUNT(Sno) as Record_Count FROM Table3 WHERE Roofdeck = 'No' AND Storage = 'No';", con)

# Query 5
query5_result = pd.read_sql_query("SELECT Parking, COUNT(Sno) as Record_Count FROM Table3 WHERE Fireplace = 'Yes' OR Dishwasher = 'Yes' GROUP BY Parking;", con)

# Display Results for Table3 Queries
for i, result in enumerate([query1_result, query2_result, query3_result, query4_result, query5_result], start=1):
    print(f"Query {i}:")
    print(result)
    print()

# 7 Join SQL Queries using all 3 tables
print("Join SQL Queries:")

# 1) SQL subquery to find records with more than the average area and related details using table 1 and table 2
join_query1 = f"SELECT * FROM Table1 WHERE Sno IN (SELECT Sno FROM Table2 WHERE Area > (SELECT AVG(Area) FROM Table2));"
result_join1 = pd.read_sql_query(join_query1, con)

# 2) Subquery to find records in table1 based on conditions pets allowed is 'YES' and no of bed is greater than 3 in table2
join_query2 = "SELECT * FROM Table1 WHERE Sno IN (SELECT Sno FROM Table2 WHERE Pets_Allowed = 'Yes' AND No_of_Bed > 3);"
result_join2 = pd.read_sql_query(join_query2, con)

# 3) 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
join_query3 = "SELECT Table2.*, Table3.* FROM Table2 JOIN Table3 ON Table2.Sno = Table3.Sno WHERE Table2.No_of_Bed > 2 AND Table3.AC = 'Yes';"
result_join3 = pd.read_sql_query(join_query3, con)

# 4) SQL subquery to find records in Table2 with pets allowed and a Dishwasher, and include related details from Table3
join_query4 = "SELECT Table2.*, Table3.* FROM Table2 JOIN Table3 ON Table2.Sno = Table3.Sno WHERE Table2.Pets_Allowed = 'Yes' AND Table3.Dishwasher = 'Yes';"
result_join4 = pd.read_sql_query(join_query4, con)

# 5) Subquery to find records in Table2 with the highest area and related details from Table3 where roofdeck is present
join_query5 = "SELECT Table2.*, Table3.* FROM Table2 JOIN Table3 ON Table2.Sno = Table3.Sno WHERE Table2.Area = (SELECT MAX(Area) FROM Table2) AND Table2.Roofdeck = 'Yes';"
result_join5 = pd.read_sql_query(join_query5, con)

# 6) SQL Inner Join to combine information from table1 and table 2
join_query6 = "SELECT * FROM Table1 INNER JOIN Table2 ON Table1.Sno = Table2.Sno;"
result_join6 = pd.read_sql_query(join_query6, con)

# 7) SQL Subquery to find records in table1 with pets allowed and a Washer/Dryer, and include details from table2 and table3
join_query7 = "SELECT Table1.*, Table2.*, Table3.* FROM Table1 JOIN Table2 ON Table1.Sno = Table2.Sno JOIN Table3 ON Table2.Sno = Table3.Sno WHERE Table1.Pets_Allowed = 'Yes' AND Table1.Washer_Dryer = 'Yes';"
result_join7 = pd.read_sql_query(join_query7, con)

# Display Results for Join SQL Queries
for i, result in enumerate([result_join1, result_join2, result_join3, result_join4, result_join5, result_join6, result_join7], start=1):
    print(f"Join Query {i}:")
    print(result)
    print()

# Close SQLite connection
con.close()
```

Table1 Queries:

Query 1:

	Sno	Rental_Price	City	State_Code	Address	Deposit	Country
0	1	1000	A	CA	123 Main St	500	USA
1	3	1200	C	TX	789 Oak St	600	USA
2	2	1500	B	NY	456 Elm St	700	Canada

Query 2:

	City	State_Code	avg_rental_price
0	A	CA	1000.0
1	B	NY	1500.0
2	C	TX	1200.0

Query 3:

	Address	City	Deposit
0	456 Elm St	B	700
1	789 Oak St	C	600
2	123 Main St	A	500

Query 4:

	Country	Record_Count	Total_Deposit
0	Canada	1	700
1	USA	2	1100

Query 5:

	Sno	Rental_Price	City	State_Code	Address	Deposit	Country
0	2	1500	B	NY	456 Elm St	700	Canada

Table2 Queries:

Query 1:

	No_of_Bed	avg_area
0	1	600.0
1	2	800.0
2	3	1200.0

Query 2:

Empty DataFrame

Columns: [Sno, No\_of\_Bed, No\_of\_Bathroom, Pets\_Allowed, Area]

Index: []

Query 3:

	No_of_Bed	No_of_Bathroom	Total_Area
0	3	2	5
1	2	1	3
2	1	1	2

Query 4:

	No_of_Bed	No_of_Bathroom	Record_Count
0	1	1	1
1	2	1	1
2	3	2	1

Query 5:

	Sno	No_of_Bed	No_of_Bathroom	Pets_Allowed	Area
0	1	2	1	Yes	800

Table3 Queries:

Query 1:

	Sno	Washer_Dryer	AC	Hardwood_Floors	Roofdeck	Storage	Parking	Dishwasher	\
0	3	Yes	Yes	No	No	Yes	Yes	Yes	
Fireplace									
0	Yes								

Query 2:

	Sno	Washer_Dryer	AC	Hardwood_Floors	Roofdeck	Storage	Parking	Dishwasher	\
0	1	Yes	No	Yes	No	No	Yes	Yes	
Fireplace									
0	No								

Query 3:

	Sno	Washer_Dryer	AC	Hardwood_Floors	Roofdeck	Storage	Parking	Dishwasher	\
0	3	Yes	Yes	No	No	Yes	Yes	Yes	
Fireplace									
0	Yes								

Query 4:

	Record_Count
0	1

Query 5:

	Parking	Record_Count
0	No	1
1	Yes	2

Join SQL Queries:

Join Query 1:

	Sno	Rental_Price	City	State_Code	Address	Deposit	Country
0	2	1500	B	NY	456 Elm St	700	Canada

Join Query 2:

Empty DataFrame

Columns: [Sno, Rental\_Price, City, State\_Code, Address, Deposit, Country]

Index: []

Join Query 3:

	Sno	No_of_Bed	No_of_Bathroom	Pets_Allowed	Area	Sno	Washer_Dryer	AC	\
0	2	3	2	No	1200	2	No	Yes	
Hardwood_Floors									
0	Yes		Yes	No	No	No	Yes		

Join Query 4:

	Sno	No_of_Bed	No_of_Bathroom	Pets_Allowed	Area	Sno	Washer_Dryer	AC	\
0	1	2	1	Yes	800	1	Yes	No	
1	3	1	1	Yes	600	3	Yes	Yes	
Hardwood_Floors									
0	Yes		No	No	Yes	Yes	No		
1	No		No	Yes	Yes	Yes	Yes		

Join Query 5:

	Sno	No_of_Bed	No_of_Bathroom	Pets_Allowed	Area	Sno	Washer_Dryer	AC	\
0	2	3	2	No	1200	2	No	Yes	
Hardwood_Floors									
0	Yes		Yes	No	No	No	Yes		

Join Query 6:

	Sno	Rental_Price	City	State_Code	Address	Deposit	Country	Sno	\
0	1	1000	A	CA	123 Main St	500	USA	1	
1	2	1500	B	NY	456 Elm St	700	Canada	2	
2	3	1200	C	TX	789 Oak St	600	USA	3	
No_of_Bed									
0	2		1	Yes	800				
1	3		2	No	1200				
2	1		1	Yes	600				
Join Query 7:									
	Sno	Rental_Price	City	State_Code	Address	Deposit	Country	Sno	\
0	1	1000	A	CA	123 Main St	500	USA	1	
1	3	1200	C	TX	789 Oak St	600	USA	3	
No_of_Bed									
0	2		1	...	800	1	Yes	No	
1	1		1	...	600	3	Yes	Yes	
Hardwood_Floors									
0	Yes		No	No	Yes	Yes	No		
1	No		No	Yes	Yes	Yes	Yes		
[2 rows x 21 columns]									