7 Join SQL Queries using all 3 tables

1 Retrieve properties from table1 that have a higher price per square foot than the average price per square foot in table2.

```sql
SELECT t1.*
FROM Table1 t1
JOIN Table2 t2 ON t1.property_id = t2.property_id  -- Assuming property_id is the common key
WHERE t1.price_per_sqft > (SELECT AVG(price_per_sqft) FROM Table2);
```

2 Find the properties in table1 that are located in cities where the average price per square foot in table2 is higher than the overall average price per square foot.

```sql
SELECT t1.*
FROM Table1 t1
JOIN (
    SELECT city, AVG(price_per_sqft) AS avg_price_per_sqft
    FROM Table2
    GROUP BY city
) t2_avg ON t1.city = t2_avg.city
WHERE t2_avg.avg_price_per_sqft > (SELECT AVG(price_per_sqft) FROM Table2);
```

3. Retrieve properties from table1 with a certain landmark that have a lower price per square foot than the average price per square foot for properties with the same landmark in table2. (Choose landmark on our own)

```sql
SELECT t1.*
FROM Table1 t1
JOIN Table2 t2 ON t1.landmark = t2.landmark  -- Assuming 'landmark' is the common field
WHERE t1.price_per_sqft < (SELECT AVG(price_per_sqft) FROM Table2 WHERE landmark = t1.landmark);
```

4 Retrieve properties from table2 with a price per square foot higher than the average booking amount in table3

```sql
SELECT *
FROM Table2
WHERE price_per_sqft > (SELECT AVG(booking_amount) FROM Table3);
```

5 Count the number of properties in table2 with more bedrooms than the maximum number of bedrooms in table3

```sql
SELECT COUNT(*) AS property_count
FROM Table2
WHERE num_bedrooms > (SELECT MAX(num_bedrooms) FROM Table3);
```

6

Find the cities where the average booking amount in table3 is higher than the overall average booking amount, and retrieve properties from table1 located in those cities

```sql
SELECT t1.*
FROM Table1 t1
JOIN (
    SELECT city
    FROM Table3
    GROUP BY city
    HAVING AVG(booking_amount) > (SELECT AVG(booking_amount) FROM Table3)
) t3_avg ON t1.city = t3_avg.city;
```

7

Retrieve properties from tablel with a furnished status of 'Unfurnished' and a facing direction that does not exist in table3.

```sql
SELECT *
FROM Table1 t1
WHERE furnished_status = 'Unfurnished' AND facing_direction NOT IN (SELECT DISTINCT facing_direction FROM Table3);
```