

## Week-4: Assignment

### Q1). Weather Observation Station 8

```
Ans). SELECT DISTINCT CITY FROM STATION WHERE CITY REGEXP  
'^[aeiouAEIOU].*[aeiouAEIOU]$'
```

### Q2). Population Density Difernece

```
Ans). SELECT MAX(POPULATION) - MIN(POPULATION)  
FROM CITY;
```

### Q3). Weather Observation Station 19

```
Ans). SELECT ROUND(  
    SQRT(POWER(MAX(LAT_N) - MIN(LAT_N), 2) +  
        POWER(MAX(LONG_W) - MIN(LONG_W), 2)), 4)  
FROM STATION;
```

### Q4). Weather Observation Station 20

```
Ans). WITH Ordered AS (  
    SELECT LAT_N,  
           ROW_NUMBER() OVER (ORDER BY LAT_N) AS rn,  
           COUNT(*) OVER () AS total_rows  
    FROM STATION  
) ,  
MedianCalc AS (  
    SELECT LAT_N  
    FROM Ordered  
    WHERE rn = (total_rows + 1) / 2  
           OR rn = (total_rows + 2) / 2  
)  
SELECT ROUND(AVG(LAT_N), 4) AS median  
FROM MedianCalc;
```

### Q5). African Cities

```
Ans). SELECT C1.name  
FROM CITY as C1  
JOIN COUNTRY as C2
```

```
ON C1.COUNTRYCODE = C2.Code
WHERE C2.CONTINENT='Africa';
```

#### Q6). Report: Write a SQL Query to create a report

```
Ans).  SELECT
        CASE
            WHEN G.Grade < 8 THEN 'NULL'
            ELSE S.Name
        END AS Name,
        G.Grade,
        S.Marks
FROM Students S
JOIN Grades G
    ON S.Marks BETWEEN G.Min_Mark AND G.Max_Mark
ORDER BY
    G.Grade DESC,
    CASE
        WHEN G.Grade >= 8 THEN S.Name
        ELSE NULL
    END,
    CASE
        WHEN G.Grade < 8 THEN S.Marks
        ELSE NULL
    END;
END;
```

#### Q7). Top Competitors

```
Ans).  SELECT
        s.hacker_id,
        h.name
FROM Submissions s
JOIN Challenges c ON s.challenge_id = c.challenge_id
JOIN Difficulty d ON c.difficulty_level = d.difficulty_level
JOIN Hackers h ON s.hacker_id = h.hacker_id
WHERE s.score = d.score
GROUP BY s.hacker_id, h.name
HAVING COUNT(DISTINCT s.challenge_id) > 1
ORDER BY COUNT(DISTINCT s.challenge_id) DESC, s.hacker_id;
```

### Q8). Ollivander's Inventory

```
Ans).  SELECT w.id, wp.age, w.coins_needed, w.power
FROM Wands w
JOIN Wands_Property wp ON w.code = wp.code
JOIN (
    SELECT w2.power, wp2.age, MIN(w2.coins_needed) AS min_coins
    FROM Wands w2
    JOIN Wands_Property wp2 ON w2.code = wp2.code
    WHERE wp2.is_evil = 0
    GROUP BY w2.power, wp2.age
) AS best
ON w.power = best.power
AND wp.age = best.age
AND w.coins_needed = best.min_coins
WHERE wp.is_evil = 0
ORDER BY w.power DESC, wp.age DESC;
```

### Q9). Contest Leaderboard: Question based on joins

```
Ans).  SELECT
    h.hacker_id,
    h.name,
    SUM(max_scores.max_score) AS total_score
FROM Hackers h
JOIN (
    SELECT
        s.hacker_id,
        s.challenge_id,
        MAX(s.score) AS max_score
    FROM Submissions s
    GROUP BY s.hacker_id, s.challenge_id
) AS max_scores
ON h.hacker_id = max_scores.hacker_id
GROUP BY h.hacker_id, h.name
HAVING SUM(max_scores.max_score) > 0
ORDER BY total_score DESC, h.hacker_id ASC;
```