SQL ASSESSMENT

Q1) (Use the Cricket data for this problem) Prepare the output in which you will be having the following column: venue,match_detail (column gives the data in string format team1 vs team2 and if the match is affected by rain then the format will be team1 vs team2 (match abandoned due to rain)),total_run_scored(total runs scored in a match),total_fall_of_wickets(total fall wicket in a match), winners.

```
SELECT
    Venue,
    CASE
        WHEN Winners = 'Rain' THEN CONCAT([_1st_Team], ' vs ', [_2nd_Team], ' (match
abandoned due to rain)')
        ELSE CONCAT([_1st_Team], ' vs ', [_2nd_Team])
    END AS match_detail,
    COALESCE(CAST([First_Innings_Score] AS INT), 0) +
COALESCE(CAST([Second_Innings_Score] AS INT), 0) AS total_run_scored,
    COALESCE(CAST([Fall_of_wickets_First_Innings] AS INT), 0) +
COALESCE(CAST([Fall_of_wickets_Second_Innings] AS INT), 0) AS total_fall_of_wickets,
    Winners
FROM [dbo].[Cricket];
```

Q2) (Use the Cricket data for this problem) Create a view that will give you the name of the team and the total runs scored by a team in the entire World Cup.

```
CREATE VIEW team_total_runs AS
SELECT
    [_1st_Team] AS team_name,
    SUM(CAST([First_Innings_Score] AS INT)) AS total_runs
FROM [dbo].[Cricket]
GROUP BY [_1st_Team]
UNION
SELECT
    [_2nd_Team] AS team_name,
    SUM(CAST([Second_Innings_Score] AS INT)) AS total_runs
FROM [dbo].[Cricket]
GROUP BY [_2nd_Team]

SELECT * FROM team_total_runs
```

Q3) (Use the Cricket data for this problem) Write a query that will give you the name of the player who became the player of the match for the most number of times.

```
SELECT TOP 1 "Player_Of_The_Match" AS PlayerName, COUNT(*) AS MatchCount
FROM dbo.Cricket
WHERE "Player_Of_The_Match" <> 'Rain'
GROUP BY "Player_Of_The_Match"
ORDER BY MatchCount DESC;
```

Q4) (Use the Cricket data for this problem) Write a query that will display the name of the venue of match where the average of first innings score is 6th lowest.

```
SELECT
   Venue
FROM (
   SELECT
      Venue,
      AVG(CAST(First_Innings_Score AS INT)) AS avg_first_innings_score,
      ROW_NUMBER() OVER (ORDER BY AVG(CAST(First_Innings_Score AS INT))) AS rn
   FROM Cricket
   GROUP BY Venue
) t
WHERE rn = 6;
```

Q5) Write a query that will help you to calculate that by how much the temperature is increasing or decreasing next day in Rajasthan's Ajmer

```
SELECT
   DISTINCT location_name,
   last_updated_epoch,
   CAST(temperature_celsius AS FLOAT) AS temperature_celsius,
   LEAD(CAST(temperature_celsius AS FLOAT), 1) OVER (PARTITION BY location_name
ORDER BY last_updated_epoch) AS next_day_temperature_celsius,
   ROUND(LEAD(CAST(temperature_celsius AS FLOAT), 1) OVER (PARTITION BY
location_name ORDER BY last_updated_epoch) - CAST(temperature_celsius AS FLOAT), 2)
AS temperature_change
FROM
   weather
WHERE
   location_name = 'Ajmer' AND
   region = 'Rajasthan'
ORDER BY
   last_updated_epoch;
```

Q6) Write a query that will give you hottest temperature for region's locations you need to display region,location_name and the date on which the hottest temperature was recorded

```
SELECT
   region,
   location_name,
   MAX(CAST(temperature_celsius AS FLOAT)) AS max_temperature_celsius,
   CONVERT(VARCHAR, DATEADD(SECOND, MAX(last_updated_epoch), '1970-01-01'), 120)
AS max_temperature_date
FROM
   Weather
GROUP BY
   region,
   location_name
ORDER BY
   max_temperature_celsius DESC;
```

Q7) Write a query that will give you the all details of region's location where the temperature is greater than the average temperature of that particular location

```
WITH average_temps AS (
   SELECT
      location_name,
      AVG(CAST(temperature_celsius AS FLOAT)) AS avg_temperature_celsius
   FROM
     Weather
   GROUP BY
      location_name
)
SELECT
   i.location_name,
   i.temperature_celsius,
   i.region,
   i.latitude,
   i.longitude,
   i.last_updated,
   i.condition_text
FROM
   Weather i
   JOIN average_temps a ON i.location_name = a.location_name
WHERE
   CAST(i.temperature_celsius AS FLOAT) > a.avg_temperature_celsius
```

```
ORDER BY
    i.temperature_celsius DESC;
```

Q8) Create a procedure that will take a number(N) as input and display the top N coldest dates,location in India for each location among the dataset additionally also display what is the weekday,month with there name.

```
CREATE PROCEDURE GetTopNColdestLocationss
    @N INT
AS
BEGIN
    SELECT
        location_name,
        CAST(temperature_celsius AS FLOAT) AS temperature_celsius,
        CONVERT(VARCHAR, DATEADD(SECOND, last_updated_epoch, '1970-01-01'), 120) AS record_date,
        DATENAME(WEEKDAY, DATEADD(SECOND, last_updated_epoch, '1970-01-01')) AS weekday,
        DATENAME(MONTH, DATEADD(SECOND, last_updated_epoch, '1970-01-01')) AS month
    FROM
        Weather
    ORDER BY
        temperature_celsius ASC
    OFFSET 0 ROWS
    FETCH FIRST @N ROWS ONLY;
END

EXEC GetTopNColdestLocationss @N = 1;
```