**Task 1.** Find the average temperature recorded for each device. The task is to calculate the average temperature recorded for each device in the dataset.

SELECT

device\_id,

AVG(temperature)

FROM cleaned\_environment

GROUP BY device\_id;

**Task 2.** Retrieve the top 5 devices with the highest average carbon monoxide levels. This task involves identifying the devices with the highest average carbon monoxide levels and retrieving the top 5 devices based on this metric.

SELECT device\_id, AVG(carbon\_monoxide) as avg\_co

FROM cleaned\_environment

GROUP BY device\_id

ORDER BY avg\_co

LIMIT 5;

**Task 3.** Calculate the average temperature recorded in the cleaned\_environment table. The objective is to Determine the average temperature recorded in the cleaned\_environment dataset.

SELECT AVG(temperature)

FROM cleaned\_environment;

**Task 4.** Find the timestamp and temperature of the highest recorded temperature for each device. This task requires identifying the highest recorded temperature for each device and retrieving the corresponding timestamp and temperature values.

SELECT timestamp, device\_id, MAX(temperature)

FROM cleaned\_environment

GROUP BY device\_id, timestamp;

**Task 5.** Identify devices where the temperature has increased from the minimum recorded temperature to the maximum recorded temperature. The goal is to Identify devices where the temperature has increased from the minimum recorded temperature to the maximum recorded temperature.

SELECT device\_id

FROM cleaned\_environment

GROUP BY device\_id

HAVING MIN(temperature) < MAX(temperature);

**Task 6.** Find the timestamps and devices where carbon monoxide level exceeds the average carbon monoxide level of all devices. The objective is to identify the timestamps and devices where the carbon monoxide level exceeds the average carbon monoxide level across all devices.

SELECT

timestamp,

device\_id

FROM cleaned\_environment

WHERE carbon\_monoxide > (SELECT

AVG(carbon\_monoxide)

FROM cleaned\_environment);

**Task 7.** Retrieve the devices with the highest average temperature recorded. The objective is to identify the devices that have recorded the highest average temperature among all the devices in the dataset.

SELECT device\_id, AVG(temperature)

FROM cleaned\_environment

GROUP BY device\_id

ORDER BY 2 desc;

**Task 8.** Calculate the average temperature for each hour of the day across all devices. The goal is to calculate the average temperature for each hour of the day, considering data from all devices.

SELECT hour(timestamp), AVG(temperature)

FROM cleaned\_environment

GROUP BY 1;

**Task 9.** Which device(s) in the cleaned environment dataset have recorded only a single distinct temperature value? The objective is to identify device(s) in the cleaned environment dataset have recorded only a single distinct temperature value.

SELECT device\_id

FROM cleaned\_environment

GROUP BY device\_id

HAVING COUNT(DISTINCT temperature) = 1;

**Task 10.** Find the devices with the highest humidity levels. The objective is to identify the devices that have recorded the highest humidity levels.

SELECT device\_id, MAX(humidity)

FROM cleaned\_environment

GROUP BY device\_id

ORDER BY 2 DESC

**Task 11.** Calculate the average temperature for each device, excluding outliers (temperatures beyond 3 standard deviations). This task requires calculating the average temperature for each device while excluding outliers, which are temperatures beyond 3 standard deviations from the mean.

SELECT

device\_id,

AVG(temperature) AS average\_temperature

FROM

cleaned\_environment

WHERE

temperature BETWEEN (

SELECT AVG(temperature) - 3 \* STDDEV(temperature)

FROM cleaned\_environment

) AND (

SELECT AVG(temperature) + 3 \* STDDEV(temperature)

FROM cleaned\_environment

)

GROUP BY

device\_id;

**Task 12.** Retrieve the devices that have experienced a sudden change in humidity (greater than 50% difference) within a 30-minute window. The goal is to identify devices that have undergone a sudden change in humidity, where the difference is greater than 50%, within a 30-minute time window.

WITH T1 AS (

SELECT

device\_id,

timestamp,

humidity,

LEAD(humidity) OVER (PARTITION BY device\_id ORDER BY timestamp) AS lead\_humidity

FROM

cleaned\_environment

),

T2 AS (

SELECT

device\_id,

timestamp,

humidity,

lead\_humidity,

ABS(lead\_humidity - humidity) AS change\_humidity

FROM

T1

)

SELECT

device\_id,

timestamp,

lead\_humidity

FROM

T2

WHERE

change\_humidity > 0.5

**Task 13.** Find the average temperature for each device during weekdays and weekends separately. This task involves calculating the average temperature for each device separately for weekdays and weekends.

SELECT

device\_id,

CASE

WHEN DAYOFWEEK(timestamp) < 6 THEN "Weekday"

ELSE "Weekend"

END AS day\_type,

AVG(temperature) AS average\_temperature

FROM cleaned\_environment

GROUP BY device\_id

**Task 14.** Calculate the cumulative sum of temperature for each device, ordered by timestamp limit to 10. The objective is to calculate the cumulative sum of temperature for each device, considering the records ordered by timestamp limit to 10.

SELECT

device\_id,

timestamp,

temperature,

SUM(temperature) OVER (PARTITION BY device\_id ORDER BY timestamp) AS cumulative\_temperature

FROM

cleaned\_environment

LIMIT 10