# SQL Assessment Questions with Points

## 🟢 Beginner Level – Total: 20 Points

These questions test basic SQL SELECT and JOIN skills.

1. (6 points)  
Write an SQL query to list all site names (AssetTitle) where the ONM\_Project\_status is 'In operation'.

SELECT AssetTitle

FROM Business.vwSites

WHERE ONM\_Project\_status = 'In operation'

2. (7 points)  
Write an SQL query to list all weather station names (AssetTitle) that belong to the site called Sarnia Solar.

SELECT w.AssetTitle

FROM Business.vwSites s

LEFT JOIN Business.vwWeatherStations w

ON s.AssetID = w.SiteAssetId

WHERE s.AssetTitle = 'Sarnia Solar'

3. (7 points)  
Write an SQL query to retrieve the site name and the associated inverter names and models (AssetTitle and Make\_Model) for all inverters located at the Sarnia Solar site.

SELECT s.AssetTitle, i.AssetTitle, i.Make\_Model

FROM Business.vwSites s

LEFT JOIN Business.vwInverters i

ON s.AssetID = i.SiteAssetID

WHERE s.AssetID = 48

## 🟡 Intermediate Level – Total: 30 Points

These questions require grouping and aggregation.

4. (10 points)  
Write an SQL query to list all site names that are currently in operation, along with the total number of inverters at each site.

SELECT s.AssetTitle, COUNT(i.AssetTitle) AS InverterCount

FROM Business.vwSites s

LEFT JOIN Business.vwInverters i

ON s.AssetID = i.SiteAssetID

WHERE s.ONM\_Project\_Status = 'In operation'

GROUP BY s.AssetTitle

5. (10 points)  
Write an SQL query to list all site names that are currently in operation, along with the total number of weather stations at each site.  
Note: Make sure to use appropriate table joins and grouping.

SELECT s.AssetTitle, COUNT(w.AssetTitle) AS InverterCount

FROM Business.vwSites s

LEFT JOIN Business.vwWeatherStations w

ON s.AssetID = w.SiteAssetID

WHERE s.ONM\_Project\_Status = 'In operation'

GROUP BY s.AssetTitle

## 🔴 Advanced Level – Total: 50 Points

These questions involve complex joins, filtering, grouping, and window functions.

6. (25 points)  
Write an SQL query for all sites that are currently operational. For each site, return:  
- Site ID and Site Name  
- Average tilt angle and azimuth angle (ensure values are cast as float and not blank)  
- Maximum and minimum tracking limit angles  
- Mount type: return 'Tracker' if the average tilt angle is -1, otherwise return 'Fix Mount'

Requirements:  
Only include records where tilt and azimuth angles are not NULL or blank. Use joins between the site, block details, and tracker controller tables.

SELECT

s.AssetID,

s.AssetTitle,

AVG(CAST(b.Modules\_Tilt\_Angle AS FLOAT)) AS Tilt\_Angle,

AVG(CAST(b.Modules\_Azimuth\_Angle AS FLOAT)) AS Azimuth\_Angle,

t.[MaximumTrackingLimitAngle],

t.[MinimumTrackingLimitAngle],

CASE

WHEN AVG(CAST(b.Modules\_Tilt\_Angle AS FLOAT)) = -1 THEN “Tracker”

ELSE “Fix Mount”

END AS Mount\_Type

FROM Business.vwSites s

LEFT JOIN Business.vwBlockDetails b

ON s.AssetID = b.SiteAssetID

LEFT JOIN Business.vwTrackerController t

ON s.AssetID = t.SiteAssetID

WHERE s.OnM\_Project\_Status = 'In Operation'

AND b.Modules\_Tilt\_Angle IS NOT NULL

AND b.Modules\_Azimuth\_Angle IS NOT NULL

AND b.Modules\_Tilt\_Angle NOT LIKE ''

AND b.Modules\_Azimuth\_Angle NOT LIKE ''

GROUP BY s.AssetID, s.AssetTitle, t.[MaximumTrackingLimitAngle], t.[MinimumTrackingLimitAngle]

7. (25 points)  
Assume there is a temporary table called #production that contains timestamps and output readings.  
Write an SQL query to retrieve the most recent production record (based on timestamp) for each date. Use a common table expression (CTE) and ROW\_NUMBER() window function.

--Create a temporary local table

CREATE TABLE #production (

[timestamp] DATETIME,

[date] DATE,

total\_output INT

);

-- Insert data into the temporary table

INSERT INTO #production ([timestamp], [date], total\_output) VALUES

('2025-04-15 17:00', '2025-04-15', 185),

('2025-04-15 17:30', '2025-04-15', 192),

('2025-04-16 17:15', '2025-04-16', 188),

('2025-04-17 17:37', '2025-04-17', 186),

('2025-04-18 17:14', '2025-04-18', 176),

('2025-04-18 17:50', '2025-04-18', 183),

('2025-04-19 17:16', '2025-04-19', 188),

('2025-04-19 17:44', '2025-04-19', 196);

WITH RankedProduction AS (

SELECT

[timestamp],

[date],

total\_output,

ROW\_NUMBER() OVER (PARTITION BY [date] ORDER BY [timestamp] DESC) AS rn

FROM #production

)

SELECT

[timestamp],

[date],

total\_output

FROM RankedProduction

WHERE rn = 1;

✅ Total Marks: 100 Points