**Part 01 - Join Analysis Problem**

**1)** Carry out analysis of the proposed LEFT JOIN – note that there are 8640 records in the TimestampTable and 4619 records in the WindSpeedTable. Use SQL queries to verify these figures.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SELECT  (SELECT COUNT(\*) FROM timestamp) AS 'Timestamp Table',  (SELECT COUNT(\*) FROM windspeed WHERE sampleTime like '2008%') AS 'WindSpeed Table'; | |  |  | | --- | --- | | Timestamp Table | WindSpeed Table | | 8640 | 4619 | |

**2)** Explain, based on your understanding of the data – why there are 8640 records in the TimestampTable and why there are 4619 records in the WindSpeedTable (give a written explanation showing calculations).

|  |  |
| --- | --- |
| TimeStamp:  1 record every 10 minutes (x 6)  6 records per hour (x 24)  144 records per day (x (31 days in Jan + 29 days in Feb = 60))  TOTAL of 8640 records within Jan and Feb | WindSpeed:  1 record every 10 minutes (x 6)  6 records per hour (x 24)  144 records per day (x 32 turbines)  Total of 4608 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SELECT DISTINCT(  SUBSTRING(sampleTime, 1, 7))AS Months,  COUNT(\*) AS Total  FROM windspeed  GROUP BY SUBSTRING(sampleTime, 1, 7); | |  |  | | --- | --- | | Months | Total | | 2008-01 | 4608 | | 2008-03 | 11 | | 2009-01 | 2 | |

**3)**Write SQL to verify that the TimestampTable contains data for Jan/Feb 2008. Each Day (31 in Jan and 29 in Feb) stores 24 hours with 6 samples per hr (i.e. each at 10min intervals)

**THIS QUERY HAS TO BE REVIEWED**

Select count(timestamp)

from timestamp

where month(timestamp) between 1 and 2 ;

**4)** Based on an understanding of how the LEFT JOIN operation will work, and before executing any query, write a short calculation of the number of records there should be in the output of the left join. Compare this to what you actually get after you execute a query.

We expected 4608 records, however we got an output of 13104 because carrying a left join query we got duplicated values of timestamps and these values were marked as null: 8496 null (duplicated values) + 4608 (correct values) = 13104 in the output.

**-USING DISTINCT-**

SELECT p.turbineID AS ID, timestamp.timestamp as TimeStamp,

p.sampleTime as SampleTime, p.val as Value

FROM timestamp

LEFT JOIN p ON p.sampleTime = timestamp.timestamp

WHERE (select distinct sampleTime)

ORDER BY turbineID, sampleTime;

**5)** Write SQL to find out where exactly the problem is. Give a brief written explanation to accompany your SQL.

The timestamp table has dates from january and february 2008 only, while the windspeed table contains data from march 2008 and also a few outputs of 2009. Analysing the tables and executing the query below, we found out that 13 records are incorrect in the table.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SELECT sampleTime,  COUNT(\*)  FROM windspeed  GROUP BY(sampleTime)  HAVING COUNT(\*) <> 32; | |  |  | | --- | --- | | sampleTime | COUNT(\*) | | 2008-03-31 23:30:00 | 1 | | 2008-03-31 23:40:00 | 1 | | 2008-03-31 23:50:00 | 9 | | 2009-01-02 13:30:00 | 1 | | 2009-01-02 13:50:00 | 1 | |

**6)**Write SQL to solve the problem for Timestamp and Windspeed (i.e. a windspeed for each time-stamp) and give a brief description of how the overall problem of getting each variable (pitch, yaw, etc.) all lined up correctly (or “squarely” as it has been termed) would work.

To have the tables timeStamp and windSpeed squared, we executed queries to fill each timestamp row with the correct windspeed value or a null value when the sample was not collected. The same procedure should be repeated in order to have all the other variables lined up correctly.

**7)** Use the supplied alldata.sql to get each variable (pitch, yaw, etc) and create a fully joined (fully “square”) solution.