/\*Task 17: Offset Functions

--Task 17 **ROUGH WORK[DOUBLE CHECK FOR ERRORS]** - **work [ REMOVE $ SIGN AND THEN EXECUTE THE QUERIES.]**

for example the query

**SELECT \* FROM FactSales$;**

Should be typed as

**SELECT \* FROM FactSales; \*/**

The OFFSET and FETCH clauses are the options of the ORDER BY clause. They allow you to limit the number of rows to be returned by a query.

Try the following queries and have a look at the results. What are the differences between them? Do they do what you expect?

There are two categories of offset functions in an SQL server namely, functions whose offset is relative to the current row (LAG and LEAD functions) and functions whose offset is relative to the start or the end of the window frame, such as First\_Value and Last\_Value.

LAG and LEAD

The LAG and LEAD functions allow you to return a value expression from a row in a window partition that is given an offset before (LAG) or after (LEAD) of the current row.

This example returns the SalesAmount for each employee as we as the values of the previous and next sales of that employee. \*/

select EmployeeID, FullDateTime, ProductCode, SalesAmount,

LAG(SalesAmount) Over (PARTITION by EmployeeID order by FullDateTime, ProductCode) as preSalesAmount,

Lead(SalesAmount) Over (PARTITION by EmployeeID order by FullDateTime, ProductCode) as postSalesAmount

from FactSales$

/\*FIRST\_VALUE and LAST\_VALUE

Given an ordered set of rows, FIRST\_VALUE returns the value of the specified expression with respect to the first row in the window frame. The LAST\_VALUE function returns the value of the expression with respect to the last row in the frame.

FIRST\_VALUE() function example to return product categories with the lowest sales volumes in 2019 and 2020.\*/

Select ProductType, YEAR(FullDateTime) AS [Year], Quantity,

FIRST\_VALUE(ProductType) OVER (

PARTITION BY YEAR(FullDateTime) ORDER BY Quantity

) As LowestSalesVolume

From DimProduct$ INNER JOIN FactSales$ on DimProduct$.ProductCode = FactSales$.ProductCode

Where YEAR(FullDateTime) BETWEEN 2019 and 2020;

/\*This example demonstrates how to return along with each employee’s sale, the current sale amount as well as the sale amounts from the employee’s first and last sales.

You do not need to use the range as ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW since this is the default range of rows but I added it here to demonstrate a good practise of “cheaper” or lighter query runtimes. \*/

select EmployeeID, FullDateTime, ProductCode, SalesAmount,

FIRST\_VALUE(SalesAmount) Over (PARTITION by EmployeeID order by FullDateTime, ProductCode ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) as FIRSTSale,

LAST\_VALUE(SalesAmount) Over (PARTITION by EmployeeID order by FullDateTime, ProductCode) as LastSale

from FactSales$