Lecture Notes – FIRST_VALUE()LAST_VALUE()NTHVALUE()

Section 1: Lecture Summary

The lecture covers **FIRST_VALUE()**, **LAST_VALUE()**, and **NTH_VALUE()** window functions for accessing specific values within a result set. These build on **LAG()** and **LEAD()**, enabling retrieval of the first hire date, last hire date, or any specified nth hire date across the company or by department using the Employees table.

Section 2: Key Concepts and Explanations

FIRST_VALUE(hire_date) OVER (ORDER BY hire_date) returns the earliest hire date when rows are ordered by hire date ascending, shown for every employee row. Adding **PARTITION BY DeptID** computes the earliest hire date within each department separately.

LAST_VALUE(hire_date) requires an explicit frame clause like **RANGE BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING** after **ORDER BY hire_date** to capture the latest hire date across all rows; without it, the function defaults to the current row. Partitioning by **DeptID** applies this per department.

NTH_VALUE(hire_date, n) retrieves the nth hire date (e.g., 2 for second, 3 for third) in the ordered window, using similar partitioning and framing as above.

Section 3: Example Code and Use Cases

```
SELECT EmpID, FirstName, HireDate,
       FIRST_VALUE(HireDate) OVER (ORDER BY HireDate) AS
company_first_hire
FROM Employees;
```

Shows 2016-12-30 (earliest company hire date) for all rows.

```
SELECT EmpID, FirstName, DeptID, HireDate,
       FIRST_VALUE(HireDate) OVER (PARTITION BY DeptID ORDER BY HireDate)
AS dept_first_hire
FROM Employees;
```

Shows earliest hire date per **DeptID** for each employee in that department.

```
SELECT FirstName, HireDate,
       LAST_VALUE(HireDate) OVER (ORDER BY HireDate
                                  ROWS BETWEEN UNBOUNDED PRECEDING AND
UNBOUNDED FOLLOWING) AS company_last_hire
FROM Employees;
```

Shows 2021-11-03 (latest company hire date) for all rows.

```
SELECT EmpID, FirstName, DeptID, HireDate,
       LAST_VALUE(HireDate) OVER (PARTITION BY DeptID ORDER BY HireDate
                                  ROWS BETWEEN UNBOUNDED PRECEDING AND
UNBOUNDED FOLLOWING) AS dept_last_hire
FROM Employees;
```

Shows latest hire date per **DeptID** (e.g., 2021-05-12 for one department, 2021-11-03 for another).

Section 4: Key Takeaways

Order by **HireDate** ascending for **FIRST_VALUE()**; use frame clause **ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING** for **LAST_VALUE()** and **NTH_VALUE()**. Partition by **DeptID** for department-level results. These functions repeat the target value across all rows in the window.