

Lecture Notes – RANK()_DENSERANK()

Section 1: Lecture Summary

The lecture covers the `RANK()` and `DENSE_RANK()` window functions in SQL. It demonstrates their usage on the `Employees` table, starting with basic ranking, then ranking by `Salary`, and finally department-wise ranking using `partition by DeptID`. Key distinction: `RANK()` skips ranks for ties (e.g., 1,2,2,4), while `DENSE_RANK()` does not skip (e.g., 1,2,2,3).

Section 2: Key Concepts and Explanations

`RANK()` assigns ranks based on an `order by` clause within an `over()` window. Without ordering, all rows get rank 1. With `order by Salary`, rows sort by increasing `Salary`, and ties receive the same rank, skipping subsequent ranks (e.g., three ties at rank 2 lead to next rank 5). Adding `partition by DeptID` restarts ranking per department, still skipping on ties.

`DENSE_RANK()` works identically to `RANK()` but does not skip ranks after ties. For example, after two ties at rank 2, the next row gets rank 3. Partitioning by `DeptID` behaves the same way, with ranks restarting per partition. Both functions support ordering by any comparable column like `Salary`, `HireDate`, or `EmpID`.

Section 3: Example Code and Use Cases

Basic rank on all employees (no distinction, all rank 1):

```
SELECT *, RANK() OVER() AS EmpRank
FROM Employees;
```

Rank by salary (ties skip ranks):

```
SELECT *, RANK() OVER(ORDER BY Salary) AS SalaryRank
FROM Employees;
```

Department-wise rank by salary:

```
SELECT *, RANK() OVER(PARTITION BY DeptID ORDER BY Salary) AS SalaryRank
FROM Employees;
```

Basic dense rank (all rank 1):

```
SELECT *, DENSE_RANK() OVER() AS DenseRank
FROM Employees;
```

Dense rank by salary (no skipping):

```
SELECT *, DENSE_RANK() OVER(ORDER BY Salary) AS SalaryRank
FROM Employees;
```

Department-wise dense rank by salary:

```
SELECT *, DENSE_RANK() OVER(PARTITION BY DeptID ORDER BY Salary) AS
SalaryRank
FROM Employees;
```

Section 4: Key Takeaways

- Use **RANK()** when skipping ranks after ties is desired.
- Use **DENSE_RANK()** to avoid skipping ranks after ties.
- Always include **ORDER BY** in **OVER()** for meaningful ranks; **PARTITION BY** groups rankings (e.g., by **DeptID**).
- Both functions sort in increasing order by default and can rank on any column like **Salary**.