NVL and COALESCE

By Dhandapani Yedappalli Krishnamurthi Sep 30, 2025

1. NVL Function

- **Syntax:**

  NVL(expr1, expr2)

- If expr1 is **NULL**, it returns expr2. Otherwise, it returns expr1.
- Both expressions **must be of the same datatype** (or Oracle must be able to implicitly convert them).

✅ **Example:**

```
SELECT NVL(commission, 0) AS commission_value
FROM employees;
```

- If commission is NULL, it will return 0.

2. COALESCE Function

- **Syntax:**
  COALESCE(expr1, expr2, expr3, ..., exprN)

- Returns the **first non-NULL value** in the list.
- It can take **multiple arguments** (unlike NVL, which takes only 2).
- Follows **ANSI SQL standard** (portable across databases).
- All arguments must be of the same datatype (or implicitly convertible).

✅ **Example:**

```
SELECT COALESCE(commission, bonus, 0) AS payment
FROM employees;
```

- If commission is not null → return commission
- Else, if bonus is not null → return bonus
- Else, return 0

3. Indian Names Sample Example

Suppose we have this table Employee:

EmpID	Name	Commission	Bonus
1	Kapil	NULL	5000
2	Sneha	2000	NULL
3	Tharun	NULL	NULL
4	Sangeetha	3000	2000

Using NVL:

```
SELECT Name,
       NVL(Commission, 0) AS Commission_Value
FROM Employee;
```

Output:

Name	Commission_Value
Kapil	0
Sneha	2000
Tharun	0
Sangeetha	3000

Using COALESCE:

```
SELECT Name,
       COALESCE(Commission, Bonus, 0) AS Payment
FROM Employee;
```

Output:

Name	Payment
Kapil	5000
Sneha	2000
Tharun	0
Sangeetha	3000

🔑 Key Differences:

Feature	NVL	COALESCE
Arguments	2 only	Multiple
SQL Standard	Oracle-specific	ANSI SQL (works in many DBs)
Data Type Handling	Implicit conversion	Stricter (all must match)
Flexibility	Limited	More powerful