



SNOWFLAKE cmd: NVL & COALESCE

NVL - Null value replacement

The primary purpose of the NVL function is to replace null values with a specified alternative value. This can be helpful when dealing with data that may contain null values

```
SELECT Name, NVL(Salary, 0) AS Salary FROM Employees;

-- The NVL function ensures that if the "Salary" column contains null,
-- it will be replaced with zero in the result.
```

Handling null concatenation: The NVL function is often used to concatenate strings, while handling null values appropriately. When concatenating strings, if any of the values being concatenated is null, the result will be null. NVL helps avoid this issue by substituting a non-null value.

```
SELECT First_Name || ' ' || NVL>Last_Name, '') AS Name FROM Customers;

-- The NVL function ensures that if the "Last_Name" column contains
-- null, it will be replaced with an empty string, resulting
-- in a complete full name.
```

COALESCE

The **COALESCE** function is similar to the **NVL** function, but it is more versatile as it can accept multiple arguments. It returns the first non-null value from the provided arguments. If all arguments are null, it returns null.

```

SELECT column1, column2, column3, coalesce(column1, column2, column3)
FROM (values
      (1, 2, 3 ),
      (null, 2, 3 ),
      (null, null, 3 ),
      (null, null, null),
      (1, null, 3 ),
      (1, null, null),
      (1, 2, null)
) v;

```

-- OUTPUT

| COLUMN1 | COLUMN2 | COLUMN3 | COALESCE(COLUMN1, COLUMN2, COLUMN3) |
|---------|---------|---------|-------------------------------------|
| 1 | 2 | 3 | 1 |
| NULL | 2 | 3 | 2 |
| NULL | NULL | 3 | 3 |
| NULL | NULL | NULL | NULL |
| 1 | NULL | 3 | 1 |
| 1 | NULL | NULL | 1 |
| 1 | 2 | NULL | 1 |

```

SELECT Name, COALESCE(Short_Description, Description) AS Summary
FROM Products;

```

-- The COALESCE function checks if the "Short_Description"
 -- column is null. If it is not null, it returns the
 -- "Short_Description" value. Otherwise, it returns
 -- the "Description" value.

Usage Notes

- If possible, pass in arguments of the same type. Avoid passing in arguments of different types.
- If one of the arguments is a number, the function coerces non-numeric string arguments (e.g. `'a string'`) and string arguments that are not constants to the type `NUMBER(18,5)`.

For numeric string arguments that are not constants, if `NUMBER(18,5)` is not sufficient to represent the numeric value, you should cast the argument to a type that can represent the value.

COMPARISON

| Feature | NVL | COALESCE |
|---------------------|---|---|
| Syntax | NVL(expression1, expression2) | COALESCE(expression1, expression2, ...) |
| Number of Arguments | Two arguments | Two or more arguments |
| Return Behavior | Returns expression2 if expression1 is null | Returns the first non-null expression from the provided list |
| Compatibility | Specific to Oracle SQL | Standard SQL function, available in multiple database systems |
| Usage Scenarios | Replacing null values with alternative values | Selecting the first non-null value from a list of expressions |

BOTH CAN BE USED IN SNOWFLAKE.