eBOX

BY AMPHISOFT

Conditional & Null Related Functions

In this session, you will learn:

- List of Conditional and Null Related Functions with example queries

COALESCE()

- This function returns the first non-NULL value of a list, or NULL if there are no non-NULL values.

```
SELECT  COALESCE(NULL,'A','B') result FROM dual;
RESULT: A
```

Syntax

```
COALESCE(value1,value2,value3,...)
```

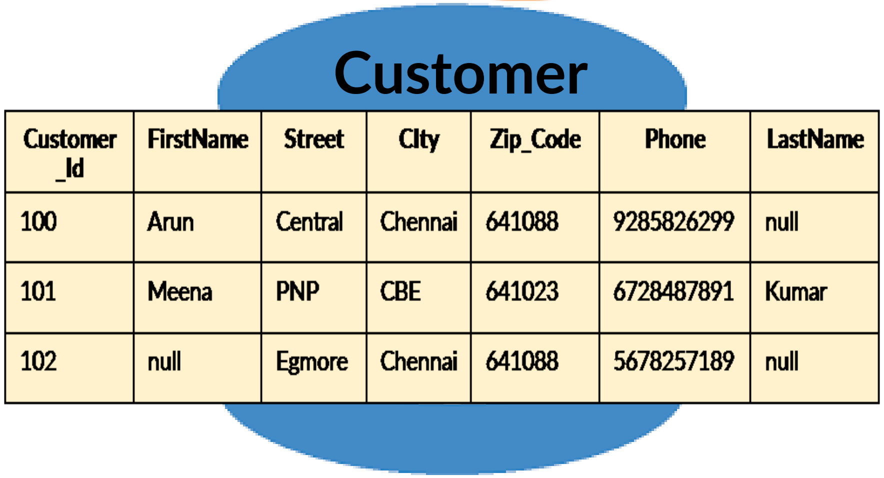
↓

```
IF value1 is not NULL THEN
    result = value1;
ELSIF value2 is not NULL THEN
    result = value2;
ELSIF value3 is not NULL THEN
    result = value3;
ELSE
    result = NULL;
END IF;
```

Example

```
SELECT   COALESCE(NULL,'A','B') result FROM dual;
RESULT: A
```

```
SELECT Customer_Id,
COALESCE(LastName,FirstName,Phone) AS Contact
from Customer
```



Customer

Customer _Id	FirstName	Street	City	Zip_Code	Phone	LastName
100	Arun	Central	Chennai	641088	9285826299	null
101	Meena	PNP	CBE	641023	6728487891	Kumar
102	null	Egmore	Chennai	641088	5678257189	null

→

Customer

Customer _Id	Contact
100	Arun
101	Kumar
102	5678257189

DECODE()

- This function has the functionality of an IF-THEN-ELSE statement.

Syntax

DECODE(expression , search1 , result1 [, search2 , result2]... [, default])

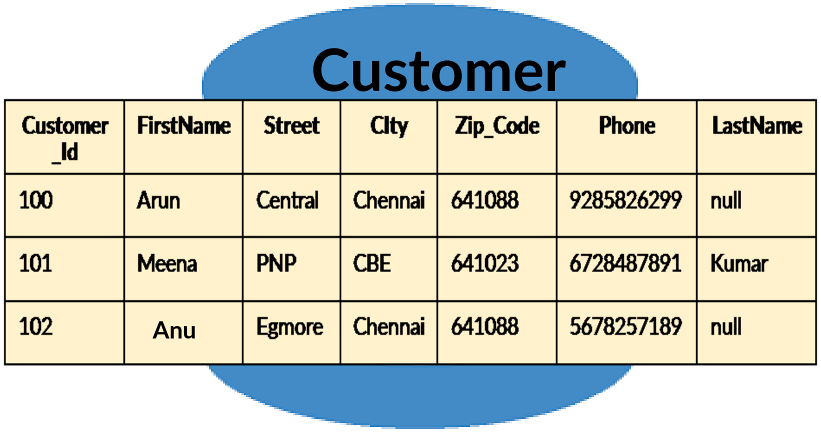
```
SELECT   DECODE(2, 1, 'One',  2, 'Two') FROM   dual;
Result: Two
```

```
SELECT DECODE(3, 1, 'One',  2, 'Two', 'Not one or two') FROM dual;
Result: Not one or two
```

Example

```
SELECT FirstName,
DECODE(Customer_Id, 100,’9285826299’,
101,’6728487891’,
‘5678257189’) AS Contact
from Customer
```

```
IF Customer_Id = 100 THEN
Contact := '9285826299';
ELSIF Customer_Id = 101 THEN
Contact := '6728487891';
ELSE
Contact := '5678257189 ';
END IF;
```



Customer

Customer _Id	FirstName	Street	CIty	Zip_Code	Phone	LastName
100	Arun	Central	Chennai	641088	9285826299	null
101	Meena	PNP	CBE	641023	6728487891	Kumar
102	Anu	Egmore	Chennai	641088	5678257189	null

Customer

FirstName	Contact
Arun	9285826299
Meena	6728487891
Anu	null

NULLIF()

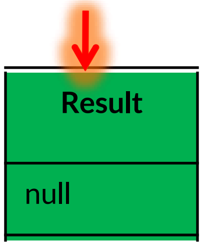
- NULLIF() compares expr1 and expr2. If expr1 and expr2 are equal, then this function returns NULL. Else returns expr1.

Syntax

```
NULLIF(expr1 , expr2);
```

Example

```
SELECT NULLIF(12,12) AS RESULT FROM dual;
```



Result
null

The Oracle NULLIF() function accepts two arguments. It returns a null value if the two arguments are equal. In case the arguments are not equal, the NULLIF() function returns the first argument.

NULLIF()

- NULLIF() compares expr1 and expr2. If expr1 and expr2 are equal, then this function returns NULL. Else returns expr1.

Syntax

```
NULLIF(expr1 , expr2);
```

Example

```
SELECT NULLIF(12,14) AS RESULT FROM dual;
```

Result
12

CASE

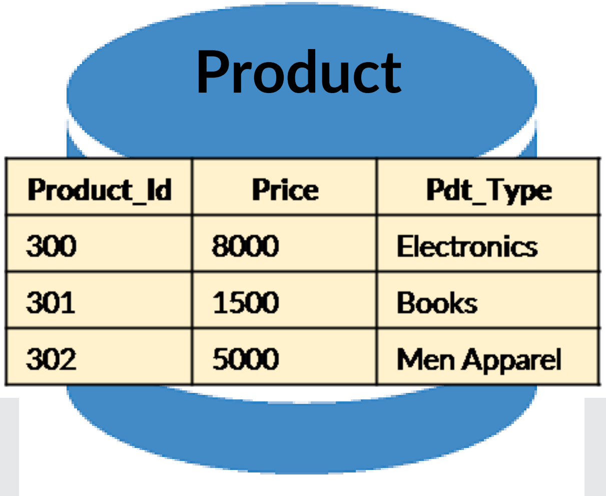
- CASE statement has the functionality of an IF-THEN-ELSE statement.

Syntax

```
CASE
      WHEN condition_1 THEN result_1
      WHEN condition_2 THEN result_2
      ...
      WHEN condition_n THEN result_n
      ELSE result
END
```

Example

```
SELECT Product_Id,
CASE WHEN Price < 3000 THEN ‘Low Price Product’
        WHEN Price BETWEEN 3000 AND 5000 THEN ‘Medium Price Product’
        ELSE ‘High Price Product’
        END AS Price_Range
from Product
```



Product_Id	Price	Pdt_Type
300	8000	Electronics
301	1500	Books
302	5000	Men Apparel

→

Product

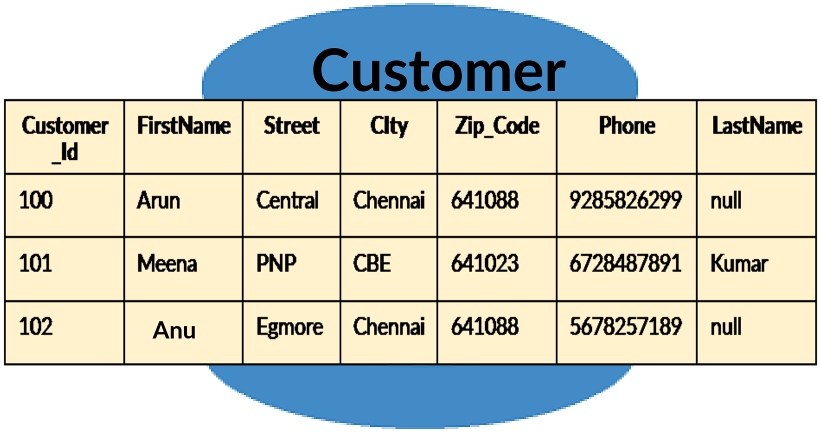
Product _Id	Price_Range
300	High Price Product
301	Low Price Product
301	Medium Price Product

NVL()

- Lets you substitute a value when a null value is encountered.

Syntax

NVL(string1 , replace_with);



Customer

Customer _Id	FirstName	Street	CIty	Zip_Code	Phone	LastName
100	Arun	Central	Chennai	641088	9285826299	null
101	Meena	PNP	CBE	641023	6728487891	Kumar
102	Anu	Egmore	Chennai	641088	5678257189	null

Example

SELECT Customer_Id, FirstName,
NVL(LastName,Phone) AS LastName FROM Customers;

↓

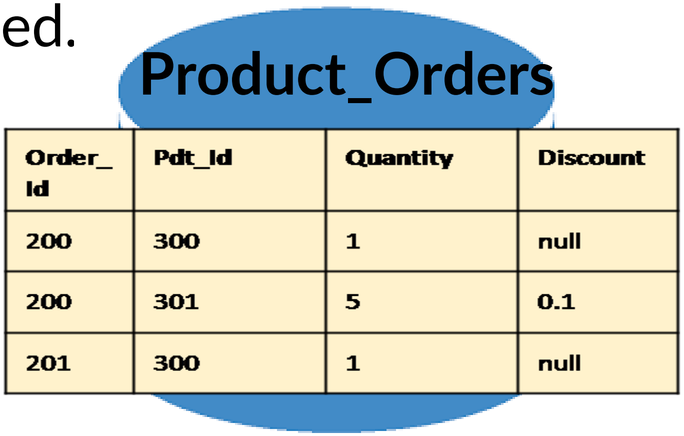
Customer_Id	FirstName	LastName
100	Arun	9285826299
101	Meena	Kumar
102	Anu	56782527189

NVL stands for "Null Value"

The NVL() function accepts two arguments. If e1 evaluates to null, then NVL() function returns e2. If e1 evaluates to non-null, the NVL() function returns e1.

NVL2()

- lets you substitutes a value when a null value is encountered as well as when a non-null value is encountered.



Product_Orders

Order_Id	Pdt_Id	Quantity	Discount
200	300	1	null
200	301	5	0.1
201	300	1	null

Syntax

NVL2(string1, value_if_not_null, value_if_null)

Example

```
SELECT Order_Id, Pdt_Id,
NVL2(Discount,’Discount Available’,’Not Available’) AS Info
FROM Product_Orders;
```

Order_Id	Pdt_Id	Info
200	300	Not Available
200	301	Discount Available
201	300	Not Available

Comparison Function – GREATEST() and LEAST()

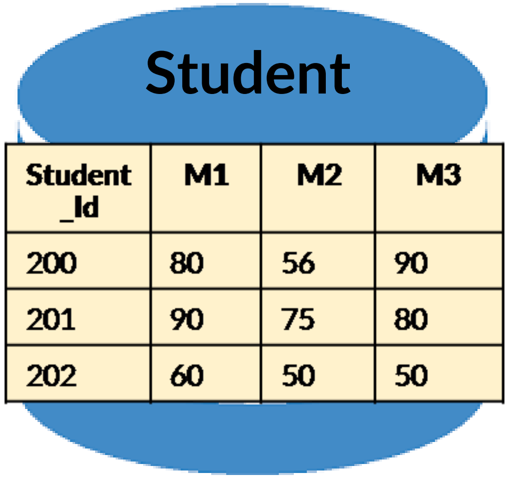
- Takes *n* arguments and return the greatest and least values of the n arguments respectively.

Syntax

```
GREATEST(value1,value2,value3,...)
LEAST(value1,value2,value3,...)
```

Example

```
SELECT Student_Id,
LEAST(M1,M2,M3) AS Low_Mark,
GREATEST(M1,M2,M3) AS High_Mark
from Student;
```



Student

Student_Id	M1	M2	M3
200	80	56	90
201	90	75	80
202	60	50	50

Student_Id	Low_Mark	High_Mark
200	56	90
201	75	90
202	50	60

THANKS

eBOX
BY AMPHISOFT