MySQL Control Flow Function

MySQL Control Flow Function :

1. MySQL IF()
2. MySQL IFNULL()
3. MySQL NULLIF()
4. MySQL CASE

---

1. IF() Function

The IF() function is used to return one value if a condition is true and another value if it is false. It's a basic conditional statement.

Syntax:

```
IF(condition, true_value, false_value)
```

Query:

```
SELECT name, IF(age >= 18, 'Adult', 'Minor') AS status
FROM students;
```

=> Sample Data:

NAME	AGE
Alice	20
Bob	17
Carol	19

ex.

```
SELECT name, IF(age >= 18, 'Adult', 'Minor') AS status
FROM students;
```

=> Sample Output:

NAME	STATUS
Alice	Adult
Bob	Minor
Carol	Adult

=> Explanation: The status column shows 'Adult' if the age is 18 or older, and 'Minor' if younger.

2. IFNULL() Function :

The IFNULL() function returns a specified value if the expression is NULL; otherwise, it returns the expression itself. It is useful for handling NULL values.

Syntax:

IFNULL(expression, value_if_null)

=> Sample Data:

NAME	PHONE_NUMBER
John	1234567890
Jane	NULL
Mike	9876543210

ex.

```
SELECT name, IFNULL(phone_number, 'No phone number')
AS phone
FROM contacts;
```

=> Sample Output:

NAME PHONE
John 1234567890
Jane No phone number
Mike 9876543210

=> Explanation: The phone column replaces NULL values with 'No phone number'.

---

3. NULLIF() Function

The NULLIF() function compares two expressions and returns NULL if they are equal; otherwise, it returns the first expression.

NOTE :
=> Return NULL if both expression are equal
=> if both expression are not equal then return firts expression

Syntax:
NULLIF(expression1, expression2)

=> Sample Data:

NAME	SCORE
Alice	85
Bob	0
Carol	90

ex.

```
SELECT name, NULLIF(score, 0) AS adjusted_score
FROM results;
```

=> Sample Output:

NAME	adjusted_score
Alice	85
Bob	NULL
Carol	90

=> Explanation: The adjusted_score column returns NULL if score is 0, otherwise it returns the score.

4. CASE Expression :

The CASE expression is used for conditional logic. It allows you to evaluate a set of conditions and return a result based on which condition is true. It's more flexible and powerful than IF() for multiple conditions.

Syntax:

```
CASE
WHEN condition1 THEN result1
WHEN condition2 THEN result2
...
ELSE default_result
END
```

=> Sample Data:

NAME	SCORE
Alice	92
Bob	85
Carol	72
Dave	65

ex.

```
SELECT name,
CASE
WHEN score >= 90 THEN 'A'
WHEN score >= 80 THEN 'B'
WHEN score >= 70 THEN 'C'
ELSE 'D'
END AS grade
FROM exam_results;
```

=> Sample Output:

```
NAME    GRADE
Alice    A
Bob    B
Carol    C
Dave    D
```

=> Explanation: The grade column is determined based on the score value.

⭐Practice Questions that focus on Control Flow Function

1. Write a query to display all employees from the employees table with a column status that shows 'Active' if the last_login date is within the last 30 days, and 'Inactive' otherwise.

2. Write a query to retrieve all orders from the orders table, replacing any NULL values in the shipping_address column with 'Address not provided'.

3. Write a query to select all products from the products table and set the price to NULL if the price is exactly 0. Otherwise, display the actual price.

4. Write a query to list all students from the students table with their grade and a new column named performance that shows:

'Excellent' if the grade is 'A'
'Good' if the grade is 'B'
'Average' if the grade is 'C'
'Needs Improvement' if the grade is 'D' or below

👉 ANS:

1. Write a query to display all employees from the employees table with a column status that shows 'Active' if the last_login date is within the last 30 days, and 'Inactive' otherwise.

```
=> SELECT employee_name, IF(last_login >= CURDATE() - INTERVAL 30 DAY, 'Active', 'Inactive') AS status
FROM employees;
```

2. Write a query to retrieve all orders from the orders table, replacing any NULL values in the shipping_address column with 'Address not provided'.

```
=> SELECT order_name, IFNULL(shipping_address, 'Address not provided') AS Address
FROM orders;
```

3. Write a query to select all products from the products table and set the price to NULL if the price is exactly 0. Otherwise, display the actual price.

```
=> SELECT product, NULLIF(price, 0) AS Price
FROM products;
```

4. Write a query to list all students from the students table with their grade and a new column named performance that shows:

'Excellent' if the grade is 'A'
'Good' if the grade is 'B'
'Average' if the grade is 'C'
'Needs Improvement' if the grade is 'D' or below

```
=> SELECT student_name, grade,
CASE
WHEN grade = 'A' THEN 'Excellent'
WHEN grade = 'B' THEN 'Good'
WHEN grade = 'C' THEN 'Average'
ELSE 'Needs Improvement'
END AS performance
FROM students;
```