

MASTERING MySQL USER- DEFINED FUNCTIONS (UDFs)

* WHAT IS A UDF? *

A User-Defined Function (UDF) is a custom function you create and save in MySQL.

- Takes 0 or more inputs.
- Performs a calculation or logic.
- Must return a single value.
- Use it to: Simplify complex calculations and make queries more readable.

UDF VS. STORED PROCEDURE

Feature	UDF (Function)	Stored Procedure (SP)
Returns	1 single value	0, 1, or multiple values
How to call	Inside a query (e.g., SELECT)	With the CALL statement
Main use	Calculations, formatting	Complex actions, transactions

HOW TO CREATE A UDF (SYNTAX)

Use this template to build your function:

```
CREATE FUNCTION FunctionName(param_1_type)
RETURNS return_type
DETERMINISTIC
BEGIN
    -- Your logic here
    RETURN (your_single_value);
END;
```

- RETURNS: The data type of the single value you're sending back (e.g., INT, VARCHAR(20)).
- DETERMINISTIC: Tells MySQL the function always returns the same result for the same inputs (helps optimize).

EXAMPLE: GETREMARKS()

This function checks a score and returns 'Passed' or 'Failed'.

```
CREATE FUNCTION GetRemarks(score INT)
RETURNS VARCHAR(10)
DETERMINISTIC
BEGIN
    IF score >= 70 THEN
        RETURN 'Passed';
    ELSE
        RETURN 'Failed';
    END IF;
END;
```

HOW TO USE YOUR UDF

Call it in a query just like a built-in function (e.g., NOW()).

```
SELECT
    student_name,
    score,
    GetRemarks(score) AS Status
FROM
    students;
```

* VISUALIZING THE IMPACT *

Table: students | student_name | score | :--- | :--- | Anna Lim | 92 | Ben Tan | 68 |

BEFORE

This one is the Complex Query

```
SELECT
    student_name,
    CASE
        WHEN score >= 70 THEN 'Passed'
        ELSE 'Failed'
    END AS Status
FROM students;
```

AFTER

This one is a Clean Query with UDF

```
SELECT
    student_name,
    GetRemarks(score) AS Status
FROM students;
```

Result (Same for both!) | student_name | Status | :--- | :--- | Anna Lim | Passed | Ben Tan | Failed |