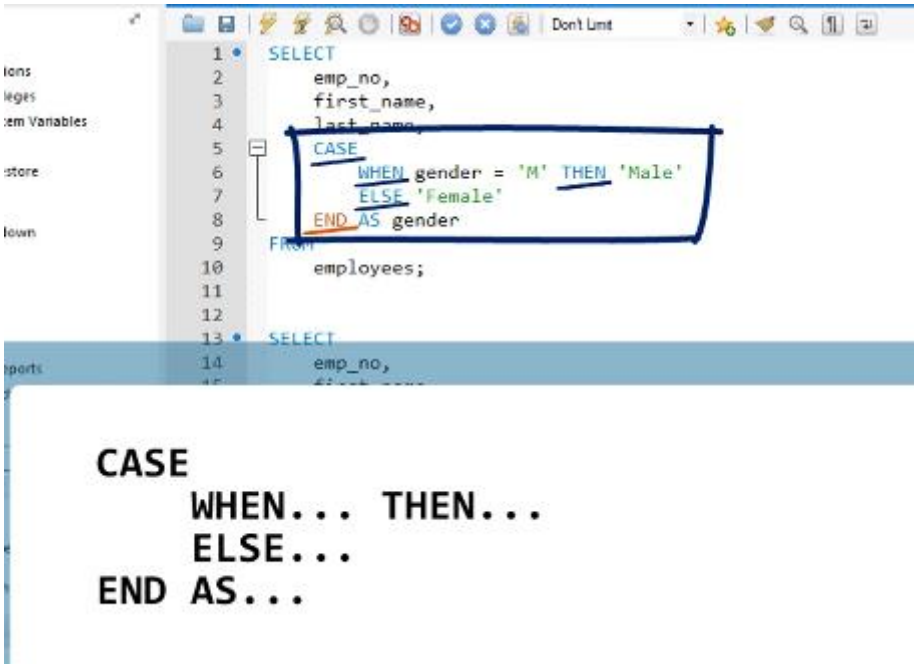
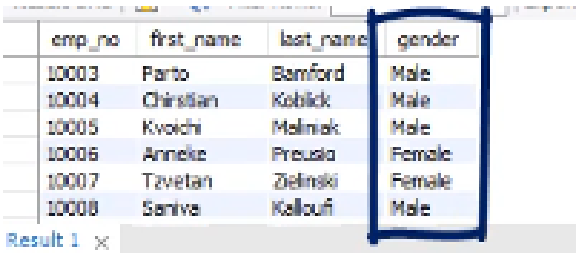
## THE CASE STATEMENT

SQL

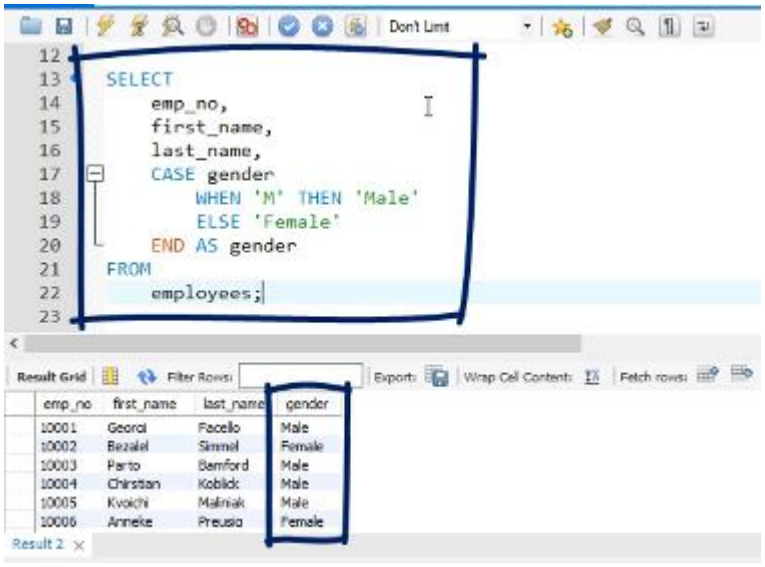
```
SELECT
    column_name(s)
    CASE
        WHEN condition_1 THEN result_1
        WHEN condition_2 THEN result_2
        ...
        ELSE
    END AS
FROM
    table_name;
```

The Case Statement is used within a Select Statement when we want to return a specific value based on a condition. Its syntax can vary depending on what we want to show.




In this example, when the value of the column is M, it will return "Male", and if it is F, "Female". As simple as that. The syntax of the case constructs starts with the keyword "Case" followed by "When", and a conditional expression containing the word "Then". After that, we have "Else" as a final expression if all conditions mentioned turn out false. Furthermore, "End" is an obligatory part of the syntax.



Now, to explore another way in which we can rewrite the Case Statement. We can obtain the same result by putting the name of the column once right after the word "Case". Then, we should write the corresponding value after the "WHEN" keyword without using the equals operator. When using the IF NOT NULL syntax, we can't do what we did here and place 'gender' after 'CASE' – the screenshot below is an example of when we can't do this but its not that deep – we're pretty advanced at reading SQL code now.

```
25  SELECT
26      e.emp_no,
27      e.first_name,
28      e.last_name,
29      CASE
30          WHEN  dm.emp_no IS NOT NULL THEN 'Manager'
31          ELSE 'Employee'
32      END AS is_manager
33  FROM
34      employees e
35          LEFT JOIN
36      dept_manager dm ON dm.emp_no = e.emp_no
37  WHERE
38      e.emp_no > 109990;
```

```
39
40
41  SELECT
42      emp_no,
43      first_name,
44      last_name,
45      IF(gender = 'M', 'Male', 'Female') AS gender
46  FROM
47      condition
48
```

IF()

The first expression within the parentheses is the condition which we want to be true. If it is true, then the return value will be the second expression of this construct. If it's false, the return value will be the one written in the third place, being 'Female', so this structure is an unspoken IF THEN ELSE setup.


```
50  SELECT
51      dm.emp_no,
52      e.first_name,
53      e.last_name,
54      MAX(s.salary) - MIN(s.salary) AS salary_difference,
55      CASE
56          WHEN MAX(s.salary) - MIN(s.salary) > 30000 THEN 'Salary was raised by more than $30,000'
57          WHEN MAX(s.salary) - MIN(s.salary) BETWEEN 20000 AND 30000 THEN
58                              'Salary was raised by more than $20,000 but less than $30,000'
59          ELSE 'Salary was raised by less than $20,000'
60      END AS salary_increase
61  FROM
62      dept_manager dm
63          JOIN
64      employees e ON e.emp_no = dm.emp_no
65          JOIN
66      salaries s ON s.emp_no = dm.emp_no
67  GROUP BY s.emp_no;
```

## The SQL CASE Statement - Exercise #1

Retrieve the employee number ( `emp_no` ), first name ( `first_name` ), and last name ( `last_name` ) of all employees from the `employees` table whose employee number is greater than 10005. Join this information with the data from the department manager `dept_manager` table to add a fourth column named `is_manager` , containing the string *'Manager'* if the employee number of the given employee is not a null value, and *'Employee'* otherwise.

This topic is covered in

**Lecture 258: The SQL CASE Statement**

```sql
SELECT
e.emp_no,
e.first_name,
e.last_name,
CASE
WHEN dm.emp_no IS NOT NULL THEN 'Manager'
ELSE 'Employee'
END AS is_manager
FROM
employees e
LEFT JOIN
dept_manager dm
ON e.emp_no = dm.emp_no
WHERE e.emp_no > 10005;
```

Run query Reset

Result Success

## The SQL CASE Statement - Exercise #2

Your analytics task is to decide if the salary raises of all managers whose employee numbers over 10005 have been significant. To do this, you need to retrieve the following table containing eight columns:

| emp_no | first_name | last_name | hire_date | min_salary | max_salary | salary_difference | salary_raise |
|---|---|---|---|---|---|---|---|
| 10006 | Anneke | Preusig | 1989-06-02 | 40000 | 60098 | 20098 | significant |
| 10007 | Tzvetan | Zielinski | 1989-02-10 | 56724 | 88070 | 31346 | significant |
| 10008 | Saniya | Kalloufi | 1994-09-15 | 46671 | 52668 | 5997 | insignificant |

1. `emp_no` from the `dept_manager` table.
2. `first_name` , `last_name` , and `hire_date` from the `employees` table.
3. `min_salary` , `max_salary` , and `salary_difference` from the `salaries` table.

`salary_raise` to indicate whether the raise is

```sql
SELECT
dm.emp_no,
e.first_name,
e.last_name,
e.hire_date,
MIN(s.salary) AS min_salary,
MAX(s.salary) AS max_salary,
MAX(s.salary) - MIN(s.salary) AS salary_difference,
CASE
WHEN MAX(s.salary) - MIN(s.salary) <= 10000 AND MAX(s.salary) - MIN(s.salary) > 0
THEN 'insignificant'
WHEN MAX(s.salary) - MIN(s.salary) > 10000 THEN 'significant'
ELSE 'salary decrease'
END as salary_raise
FROM
dept_manager dm
JOIN
employees e ON dm.emp_no = e.emp_no
JOIN
salaries s ON s.emp_no = dm.emp_no
```

Run query Reset

Result Success

The full syntax is on MYSQL

## The SQL CASE Statement - Exercise #3

Retrieve the employee number ( `emp_no` ), first name ( `first_name` ), and last name ( `last_name` ) of all employees from the `employees` table who also have records in the department employees table `dept_emp` . Add a fourth column named `current_status` displaying *"Currently working"* if their contract in the `dept_emp` table ends on or after January 1, 2025, or later. Otherwise, display *"No longer with the company"*. Use GROUP BY on the employee number, first name, and last name to obtain the desired result.

This topic is covered in

**Lecture 258: The SQL CASE Statement**

```sql
SELECT
e.emp_no,
e.first_name,
e.last_name,
CASE
WHEN MAX(de.to_date) > '2024-12-31' THEN "Currently working"
ELSE "No longer with the company"
END as current_status
FROM
employees e
JOIN
dept_emp de
ON e.emp_no = de.emp_no
GROUP BY e.emp_no, e.first_name, e.last_name;
```

Run query Reset

Result Success