**Case statement**

SQL | Case Statement

Control statements form the heart of most languages since they control the execution of other sets of statements. These are found in SQL too, and should be exploited for uses such as query filtering and query optimization through careful selection of tuples that match our requirement. In this post, we explore the Case-Switch statement in SQL.  
The CASE statement is SQL’s way of handling if/then logic.  
**Syntax:**  
**There can be two valid ways of going about the case-switch statements.**

1. The first takes a variable called case\_value and matches it with some statement\_list.
2. CASE case\_value
3. WHEN when\_value THEN statement\_list
4. [WHEN when\_value THEN statement\_list] ...
5. [ELSE statement\_list]
6. END CASE
7. The second considers a search\_condition instead of variable equality and executes the statement\_list accordingly.
8. CASE
9. WHEN search\_condition THEN statement\_list
10. [WHEN search\_condition THEN statement\_list] ...
11. [ELSE statement\_list]
12. END CASE

**Examples:**

Say we have a relation, Faculty.

**Faculty Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **FACULTYID** | **NAME** | **DEPARTMENT** | **GENDER** |
| 001 | Aakash | CS | M |
| 002 | Sahil | EC | M |
| 003 | John | HSS | M |
| 004 | Shelley | CS | F |
| 005 | Anannya | CS | F |
| 006 | Sia | HSS | F |

Let’s say we would like to modify this table such that if the department name is ‘CS’, it gets modified to ‘Computer Science’, if it is ‘EC’ it gets modified to ‘Electronics and Communication’, and if it is ‘HSS’ it gets modified to ‘Humanities and Social Sciences’. This can be achieved using case statement.

**Sample Query:**  
Consider a variable, department\_name which is entered in the SQL code.

CASE department\_name

WHEN 'CS'

THEN UPDATE Faculty SET

department='Computer Science';

WHEN 'EC'

THEN UPDATE Faculty SET

department='Electronics and Communication';

ELSE UPDATE Faculty SET

department='Humanities and Social Sciences';

END CASE

Output:

The department name corresponding to the given input gets renamed.

Consider another query which selects all the fields corresponding to the Faculty table. Since the values written in the Gender field are single character values (M/F), we would like to present them in a more readable format.

SELECT FacultyID, Name, Department,

CASE Gender

WHEN'M' THEN 'Male'

WHEN'F' THEN 'Female'

END

FROM Faculty

**Output:**

|  |  |  |  |
| --- | --- | --- | --- |
| **FACULTYID** | **NAME** | **DEPARTMENT** | **GENDER** |
| 001 | Aakash | CS | Male |
| 002 | Sahil | EC | Male |
| 003 | John | HSS | Male |
| 004 | Shelley | CS | Female |
| 005 | Anannya | CS | Female |
| 006 | Sia | HSS | Female |

Consider yet another application of case-switch in SQL- custom sorting.

CREATE PROCEDURE GetFaculty(@ColToSort varchar(150)) AS

SELECT FacultyID, Name, Gender, Department

FROM Customers

ORDER BY

CASE WHEN @ColToSort='Department' THEN Department

WHEN @ColToSort='Name' THEN Name

WHEN @ColToSort='Gender' THEN Gender

ElSE FacultyID

END

Output:

The output gets sorted according to the provided field.

The above procedure (function) takes a variable of varchar data type as its argument, and on the basis of that, sorts the tuples in the Faculty table.

This article is contributed by **Anannya Uberoi**. If you like GeeksforGeeks and would like to contribute, you can also write an article using [contribute.geeksforgeeks.org](http://contribute.geeksforgeeks.org/) or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

**DECODE:**

* **DECODE()** : Facilitates conditional inquiries by doing the work of a CASE or IF-THEN-ELSE statement.  
  The DECODE function decodes an expression in a way similar to the IF-THEN-ELSE logic used in various languages. The DECODE function decodes expression after comparing it to each search value. If the expression is the same as search, result is returned.  
  If the default value is omitted, a null value is returned where a search value does not match any of the result values.

**Syntax –**

DECODE(col|expression, search1, result1

[, search2, result2,...,][, default])

Example –

SELECT last\_name, job\_id, salary,

DECODE(job\_id, ’IT\_PROG’, 1.10\*salary,

’ST\_CLERK’, 1.15\*salary,

’SA\_REP’, 1.20\*salary,salary)

REVISED\_SALARY FROM employees;

Output :  
