

CASE Clause

SQL provides CASE clause to perform **conditional operations**. This is similar to the **switch case** / **if-else** conditions in other programming languages.

Let's learn more about the usage of CASE clause using the given database

Database

The IMDb dataset which consists of **movies**, **actors** and **cast**. You can refer to the database in the code playground for a better understanding.

movie	cast	actor
id	actor_id	actor_id
name	movie_id	name
genre	role	age
budget_in_cr		
collection_in_cr		
rating		
release_date		

CASE Clause

Each condition in the

CASE clause is evaluated and results in corresponding value when the first condition is met.

Syntax

```
1 SELECT c1, c2
2 CASE
3     WHEN condition1 THEN value1
4     WHEN condition2 THEN value2
5     ...
6     ELSE value
7     END AS cn
8 FROM table;
```

SQL

 Note

1. In CASE clause, if no condition is satisfied, it returns the value in the ELSE part. If we do not specify the ELSE part,

CASE clause results in NULL
2. We can use CASE in various clauses like SELECT, WHERE, HAVING, ORDER BY and GROUP BY.

Example

Calculate the tax amount for all movies based on the profit. Check the following table for tax percentages.

profit	tax_percentage
<=100 crores	10% of profit
100< <=500 crores	15% of profit
> 500 crores	18% of profit

[SQL](#)

```
1  SELECT id, name,
2     CASE
3         WHEN collection_in_cr - budget_in_cr <= 100 THEN collection_in_cr - budget_in_cr *
4         WHEN (collection_in_cr - budget_in_cr > 100
5         AND collection_in_cr - budget_in_cr < 500) THEN collection_in_cr - budget_in_cr *
6         ELSE collection_in_cr - budget_in_cr * 0.18
7     END AS tax_amount
8  FROM
9     movie;
```

Output

id	name	tax_amount
1	The Matrix	45.8
2	Inception	82.08
3	The Dark Knight	98.7
...

Try it Yourself

Question 1

Categorise movies as following.

rating	category
< 5	Poor
5 <= _ <= 7	Average
7 <	Good

CASE with Aggregates

CASE statements can also be used together with aggregate functions

Example

- Get the number of movies with rating greater than or equal to 8, and the movies with rating less than 8, and are released between 2015 and 2020.

```
1  SELECT
2    count(
3      CASE
4        WHEN rating >= 8 THEN 1
5      END
6    ) AS above_eight,
7    count(
8      CASE
9        WHEN rating < 8 THEN 1
10     END
```

SQL

Expand

Output

above_eight	below_eight
4	2

Try it Yourself!

- Get the number of movies with collection greater than or equal to 100 crores, and the movies with collection less than 100 crores.

Output

above_100_cr	below_100_cr
13	7



MARKED AS COMPLETE