Data Bhau

Class 5

Case Statement and CTE

What is a CASE Statement

The CASE statement goes through conditions and returns a value when the first condition is met (like an if-then-else statement). So, once a condition is true, it will stop reading and return the result. If no conditions are true, it returns the value in the ELSE clause.

If there is no ELSE part and no conditions are true, it returns NULL.

The CASE statement is followed by at least one pair of WHEN and THEN statements—SQL's equivalent of IF/THEN in Excel. Every CASE statement must end with the END statement.

The case statement helps you to add a new column to the existing database whose value is based on if-else like conditional statements

Syntax of CASE Statement

```
CASE
    WHEN condition1 THEN result1
    WHEN condition2 THEN result2
    WHEN conditionN THEN resultN
    ELSE result
END;
```

CASE Statement Example - 1

```
select *, CASE
    WHEN id >=1 and id <= 3 THEN 'The id is between 1-3'
    WHEN  id >=4 and id <= 6  THEN 'The id is between 4-6'
    ELSE 'The id is greater than 6'
END AS id_slab
from tutorial.city_populations
```

city	state	population_estimate_2012	id	id_slab
New York	NY	8336697	1	The id is between 1-3
Los Angeles	CA	3857799	2	The id is between 1-3
Chicago	IL	2714856	3	The id is between 1-3
Houston	TX	2160821	4	The id is between 4-6
Philadelphia	PA	1547607	5	The id is between 4-6
Phoenix	AZ	1488750	6	The id is between 4-6
San Antonio	TX	1382951	7	The id is greater than...
San Diego	CA	1338348	8	The id is greater than...
Dallas	TX	1241162	9	The id is greater than...
San Jose	CA	982765	10	The id is greater than...
Austin	TX	842592	11	The id is greater than...
Jacksonville	FL	836507	12	The id is greater than...

CASE Statement Example - 2

```
select id,destination_airport,coach_rev, CASE
   WHEN  coach_rev >=11000 and coach_rev <= 20000 THEN 'The revenue is between 11000-20000'
   WHEN   coach_rev >20000 and coach_rev <= 25000 THEN 'The revenue is between 20000-25000'
   ELSE 'The revenue is greater than 25000'
END AS revenue_slab
from tutorial.flight_revenue
```

id	destination_airport	coach_rev	revenue_slab
1	SFO	11782	The revenue is between 11000-200...
2	LAX	23363	The revenue is between 20000-250...
3	JFK	12549	The revenue is between 11000-200...
4	ANC	25099	The revenue is greater than 25000
5	LHR	23195	The revenue is between 20000-250...
6	ORD	11690	The revenue is between 11000-200...
7	DEN	29265	The revenue is greater than 25000
8	DFW	25007	The revenue is greater than 25000
9	ABQ	11110	The revenue is between 11000-200...
10	SEA	25618	The revenue is greater than 25000
11	PDX	20583	The revenue is between 20000-250...
12	MIA	24938	The revenue is between 20000-250...

COMMON TABLE EXPRESSION (CTE)

1. Common table expression (CTEs) is an SQL functionality that allows you to perform complex, multi-step transformations in a single, easy-to-read query. They are a helpful tool for beginners and experts alike because of their power, readability, and flexibility.
2. If we put this simply, CTE allows you to create temporary datasets that you can refer to in the future for a query.
3. These temporary datasets are "available" to use for the duration of the query itself, but they are not stored in your database. They are extinct once your query has been implemented.

Syntax of CTE Statement

```
with my_cte as

(select * from the table)

,

my_cte2 as

(select * from table)

select * from my_cte;
```

How to use CTE to add a new column to your database without actually creating a new table to calculate that column's value

```
WITH CTE_AVG_BONUS AS (
SELECT position, AVG(bonus) AS average_bonus_for_position
FROM employee
GROUP BY position)
SELECT b.employee_id, b.first_name, b.last_name, b.position, b.bonus,
ap.average_bonus_for_position
FROM employee_bonus b
JOIN avg_position ap
ON b.position = ap.position;
```

CTE with JOIN

employee_id	first_name	last_name	position	bonus	average_bonus_for_position
1	Max	Black	manager	2305.45	2428.725
2	Jane	Wolf	cashier	1215.35	1367.875
3	Kate	White	customer service specialist	1545.75	1863.6875
4	Andrew	Smart	customer service specialist	1800.55	1863.6875
5	John	Ruder	manager	2549.45	2428.725
6	Sebastian	Cornell	cashier	1505.25	1367.875
7	Diana	Johnson	customer service specialist	2007.95	1863.6875
8	Sofia	Blanc	manager	2469.75	2428.725
9	Jack	Spider	customer service specialist	2100.5	1863.6875
10	Maria	Le	cashier	1325.65	1367.875
11	Anna	Winfrey	manager	2390.25	2428.725
12	Marion	Spencer	cashier	1425.25	1367.875

Practice Question - 1

QUESTION

Write a query that includes a column flagged "RCB" when a player is from Bangalore, "CSK" when a player is from Chennai and classify the results with those players first.

SOLUTION

```
SELECT player_name,
      state,
      CASE WHEN state = 'BANGALORE'' THEN 'RCB'
      CASE WHEN state = CHENNAI THEN 'CSK'
       ELSE NULL END AS 'IPL'
 FROM IPL_Players
ORDER BY 3
```

Practice Question - 2

QUESTION

Write a query that includes players' names and a column that classifies them into four categories based on height.

SOLUTION

```
SELECT player_name,
    height,
    CASE WHEN height > 74 THEN 'over 74'
        WHEN height > 72 THEN '73-74'
        WHEN height > 70 THEN '71-72'
        ELSE 'under 70' END AS height_group
FROM height_table
```

Practice Question - 3

QUESTION

Write a query that calculates the combined weight of all underclass players (FR/SO) in California and the combined weight of all upperclass players (JR/SR) in calcutta.

SOLUTION

```
SELECT CASE WHEN year IN ('FR', 'SO') THEN 'underclass'
            WHEN year IN ('JR', 'SR') THEN 'upperclass'
            ELSE NULL END AS class_group,
       SUM(weight) AS combined_player_weight
 FROM weight_players
WHERE city = 'CA'
GROUP BY 1
```

Practice Question - 3

QUESTION

Write a query that shows the number of players at schools with the names beginning with A through M and the number of schools with the names beginning with N - Z.

SOLUTION

```
SELECT CASE WHEN school_name < 'n' THEN 'A-M'
            WHEN school_name >= 'n' THEN 'N-Z'
            ELSE NULL END AS school_name_group,
       COUNT(1) AS players
 FROM players
GROUP BY 1
```

Thanks for attending

DataBhau communication platforms:

- **Whatsapp :** https://chat.whatsapp.com/BaP2CAajm9597J8LwzYE3j
- **Linkedin Handle:** https://www.linkedin.com/company/databhau/
- **Instructors Linkedin Handles:**
  - Shrey Jain : https://www.linkedin.com/in/shrey-jain-74a90b13b/
  - Harsh Katyayan : https://www.linkedin.com/in/harsh-katyayan-a2248316b/