Case Statements and Handling NULL Values

- Searched Case Expression Syntax:

```
CASE num_value
When when_value THEN state_list
[WHEN when_value THEN state_list]..
[ELSE state_list]
End;
```

- In the above screenshot num_value is a column.
- In simple terms whenever you see a column name next to the CASE keyword it is called as a SIMPLE CASE EXPRESSION as you are locking the logic to only one column which indicates that all the conditions you will be writing will only be on that specific column.
- Whenever, you don’t see any column name next to the CASE keyword it is called as SEARCHED CASE EXPRESSION which indicates that you are not locking the logic only to one column you can give the logic on any column as per your requirement.

- Consider this employees table.

```
SELECT *,
CASE JOB_ID WHEN 'AD_PRES' THEN SALARY*0.1 + SALARY
            WHEN 'IT_PROG' THEN SALARY*0.2 + SALARY
            ELSE SALARY
            END
FROM EMPLOYEES;
```

- Let's suppose I want to increase the salary on the basis of JOB_ID.
    1. All the people whose job_id is AD_PRES their salary needs to be increased by 10%
    2. All the people whose job_id is IT_PROG their salary needs to be increased by 20%
    3. Everyone else's salary should remain as it is.

- As the question above clearly indicates that the salary has to be increased based on JOB_ID. It means I need to write a query based on the JOB_ID i.e. on the basis of a single column so we can say that we will be writing a SIMPLE CASE EXPRESSION.

employee_id	first_name	last_name	email	phone_number	HIRE_DATE	JOB_ID	salary
100	Steven	King	SKING	515.123.4567	1987-06-17	AD_PRES	90000.00
101	N	Kochhar	NKOCHHAR	515.123.4568	1989-09-21	AD_VP	17000.00
102	Lex	De Haan	LDEHAAN	515.123.4569	1993-01-13	AD_VP	17000.00
103	Alexander	Hunold	AHUNOLD	590.423.4567	1990-01-03	IT_PROG	9000.00
104	Bruce	Ernst	BERNST	590.423.4568	1991-05-21	IT_PROG	6000.00
105	David	Austin	DAUSTIN	590.423.4569	1997-06-25	IT_PROG	4800.00
106	Valli	Pataballa	VPATABAL	590.423.4560	1998-02-05	IT_PROG	4800.00
107	Diana	Lorentz	DLORENTZ	590.423.5567	1999-02-07	IT_PROG	4200.00

- Consider this employees table.

employee_id	first_name	last_name	email	phone_number	HIRE_DATE	JOB_ID	salary
100	Steven	King	SKING	515.123.4567	1987-06-17	AD_PRES	90000.00
101	N	Kochhar	NKOCHHAR	515.123.4568	1989-09-21	AD_VP	17000.00
102	Lex	De Haan	LDEHAAN	515.123.4569	1993-01-13	AD_VP	17000.00
103	Alexander	Hunold	AHUNOLD	590.423.4567	1990-01-03	IT_PROG	9000.00
104	Bruce	Ernst	BERNST	590.423.4568	1991-05-21	IT_PROG	6000.00
105	David	Austin	DAUSTIN	590.423.4569	1997-06-25	IT_PROG	4800.00
106	Valli	Pataballa	VPATABAL	590.423.4560	1998-02-05	IT_PROG	4800.00
107	Diana	Lorentz	DLORENTZ	590.423.5567	1999-02-07	IT_PROG	4200.00

- Let’s suppose I want to increase the salary on the basis of JOB_ID.
  1. All the people whose job_id is AD_PRES their salary needs to be increased by 10%
  2. Employee whose employee_id is 104 need to increase that person’s salary by 20%
  3. Everyone else’s salary should remain as it is.

- As the question above clearly indicates that the salary has to be increased based on JOB_ID and Employee_id.It means I need to write a query based on the JOB_ID and Employee_id i.e. on the basis of multiple column so we can say that we will be writing a SEARCHED CASE EXPRESSION.

```
SELECT *,
CASE WHEN JOB_ID = 'AD_PRES' THEN SALARY*0.1 + SALARY
     WHEN EMPLOYEE_ID = 104 THEN SALARY*0.2 + SALARY
     ELSE SALARY
     END
FROM EMPLOYEES;
```

NULL

- NULL, In database context, is the total absence of a value in a certain field and it means that the field value is unknown.
- NULL is not same as ZERO value for a numerical field, text field or space.
- NULL implies that a database field value has not been stored.

FUNCTIONS TO HANDLE NULL VALUES

- Below are the list of functions available in MySQL that can help you to deal with NULL.

1. ISNULL
2. IFNULL
3. COALESCE

ISNULL:

Consider the below table where you can see a column by the name commission_pct where you see some values as NULL and there are some real values present in the column.

employee_id	first_name	last_name	commission_...
132	TJ	Olson	NULL
133	Jason	Mallin	NULL
134	Michael	Rogers	NULL
135	Ki	Gee	NULL
136	Hazel	Philtanker	NULL
137	Renske	Ladwig	NULL
138	Stephen	Stiles	NULL
139	John	Seo	NULL
140	Joshua	Patel	NULL
141	Trenna	Rajs	NULL
142	Curtis	Davies	NULL
143	Randall	Matos	NULL
144	Peter	Vargas	NULL
145	John	Russell	0.40
146	Karen	Partners	0.30
147	Alberto	Errazuriz	0.30
148	Gerald	Cambrault	0.30
149	Eleni	Zlotkey	0.20
150	P	Tucker	0.30
151	David	Bernstein	0.25

. If we write the below query
select *, isnull(commission_pct) from employees;
. The output will look something like screenshot on right
. All the places where there is NULL it will show as 1 which means TRUE.
. All the places where there is NO NULL it will shows as 0 which means FALSE.

employee_id	first_name	last_name	commission_...	isnull(commission_p...
132	TJ	Olson	NULL	1
133	Jason	Mallin	NULL	1
134	Michael	Rogers	NULL	1
135	Ki	Gee	NULL	1
136	Hazel	Philtanker	NULL	1
137	Renske	Ladwig	NULL	1
138	Stephen	Stiles	NULL	1
139	John	Seo	NULL	1
140	Joshua	Patel	NULL	1
141	Trenna	Rajs	NULL	1
142	Curtis	Davies	NULL	1
143	Randall	Matos	NULL	1
144	Peter	Vargas	NULL	1
145	John	Russell	0.40	0
146	Karen	Partners	0.30	0
147	Alberto	Errazuriz	0.30	0
148	Gerald	Cambrault	0.30	0
149	Eleni	Zlotkey	0.20	0
150	P	Tucker	0.30	0
151	David	Bernstein	0.25	0

IFNULL:

Let's suppose you want to replace these NULL with 0 or some other value given or decided by the business you can make use of IFNULL function.

IFNULL takes two arguments.

1. First argument, is the column on which you would like to apply this function.
2. Second argument, is the value that you need to replace the NULLs with.

When I write the below query you can see that the rows where it was NULL a new column has been added where the NULL is showing as zero (as I have written a SELECT statement) and wherever, there was a value it remained as it is.

```
select *, ifnull(commission_pct,0) from employees;
```

employee_id	first_name	last_name	commission_...	ifnull(commission_pc...
135	Ki	Gee	NULL	0.00
136	Hazel	Philtanker	NULL	0.00
137	Renske	Ladwig	NULL	0.00
138	Stephen	Stiles	NULL	0.00
139	John	Seo	NULL	0.00
140	Joshua	Patel	NULL	0.00
141	Trenna	Rajs	NULL	0.00
142	Curtis	Davies	NULL	0.00
143	Randall	Matos	NULL	0.00
144	Peter	Vargas	NULL	0.00
145	John	Russell	0.40	0.40
146	Karen	Partners	0.30	0.30
147	Alberto	Errazuriz	0.30	0.30
148	Gerald	Cambrault	0.30	0.30
149	Eleni	Zlotkey	0.20	0.20
150	P	Tucker	0.30	0.30
151	David	Bernstein	0.25	0.25
152	Peter	Hall	0.25	0.25
153	Christopher	Olsen	0.20	0.20
154	Nanette	Cambrault	0.20	0.20

If in case you want to change or replace the null in the same column you can make use of UPDATE statement.

Update table_name
Set column_name = ifnull(column_name,0);

COALESCE:

Unlike IFNULL, Coalesce can take multiple arguments. The thing that we need to keep in mind is the datatype of the column as all the values or arguments that you provide to replace for NULL for the column should be of the same datatype as the column.

Consider the below example of the Locations table.

As you can see in the above screenshot the first row for the state province column the value is NULL. I want to pick the value from city column wherever the state province is NULL. If in case for that row the city column value is also NULL I want to pick the data from street address column and put it in state province column.

So, the query will be something like this:

Select *, coalesce(state_province,city,street_address) from locations;

location_id	street_address	postal_code	city	state_province	country_id
1000	1297 Via Cola di Rie	00989	Roma	NULL	IT
1100	93091 Calle della Testa	10934	Venice	NULL	IT
1200	2017 Shinjuku-ku	1689	Tokyo	Tokyo Prefecture	JP
1300	9450 Kamiya-cho	6823	Hiroshima	NULL	JP
1400	2014 Jabberwocky Rd	26192	Southlake	Texas	US
1500	2011 Interiors Blvd	99236	South San Francisco	California	US