



FUNCIONES

Funciones IF

EXISTEN EN MYSQL DIFERENTES FUNCIONES PARA SER IMPLEMENTADAS SEGÚN EL DATO ASIGNADO

If true



```
IF(expression, first, second)
```

Funciones IF

EXISTEN EN MYSQL DIFERENTES FUNCIONES PARA SER IMPLEMENTADAS SEGÚN EL DATO ASIGNADO

```
SELECT
    order_id,
    order_date,
    IF(
        YEAR(order_date) = YEAR(NOW()),
        'Active',
        'Archived')
FROM orders
```

ACTIVIDAD

06

REALIZAR LAS SIGUIENTES ACTIVIDADES

Funciones IF

	product_id	name	orders	frecuencia
▶	1	Tablet	3	Varias veces
	2	Audifonos	2	Varias veces
	3	Smartphone	4	Varias veces
	4	Mouse	2	Varias veces
	5	Tclado	2	Varias veces
	6	MousePad	2	Varias veces
	8	PC Gamer	1	una vez
	9	Laptop	1	una vez
	10	Smartwatch	1	una vez

Funciones CASE

EXISTEN EN MYSQL DIFERENTES FUNCIONES PARA SER IMPLEMENTADAS SEGÚN EL DATO ASIGNADO Y EN EL CASO DE QUE NO SEA UN RESULTADO BINARIO

```
SELECT
  order_id,
  CASE
    WHEN YEAR(order_date) = YEAR(NOW()) THEN 'Active'
    WHEN YEAR(order_date) = YEAR(NOW()) - 1 THEN 'Last Year'
    WHEN YEAR(order_date) < YEAR(NOW()) - 1 THEN 'Archived'
    ELSE 'Future'
  END AS category|
FROM orders
```


Actividad 06

REALIZAR LOS QUERYS QUE PERMITAN OBTENER LO SIGUIENTE

- AÑADIR UN CAMPO DE GENERACION Y SE CATALOGUE A CADA CUSTOMER

customer_id	first_name	last_name	birth_date	points	Generation
1	Barbara	Lopez	1986-03-28	2273	Milenial
2	Ines	Gomez	1986-04-13	947	Milenial
3	Alfredo	Borges	1985-02-07	2967	Milenial
4	Amara	Limon	1974-04-14	457	GenY
5	Clemente	Hernández	1973-11-07	3675	GenY
6	María	García	1991-09-04	3073	Milenial
7	Valeria	Martínez	1964-08-30	1672	GenY
8	Sofia	Pérez	1993-07-17	205	Milenial
9	Renata	Rodríguez	1992-05-23	1486	Milenial
10	Victoria	Sánchez	1969-10-13	796	GenY

*GenY = genX

Actividad 06

REALIZAR LOS QUERYS QUE PERMITAN OBTENER LO SIGUIENTE

- AÑADIR UN CAMPO DE CATEGORIA Y SE CATALOGUE A CADA CUSTOMER (BRONCE[<1000] , PLATA[1000-1999] Y ORO[>1999])

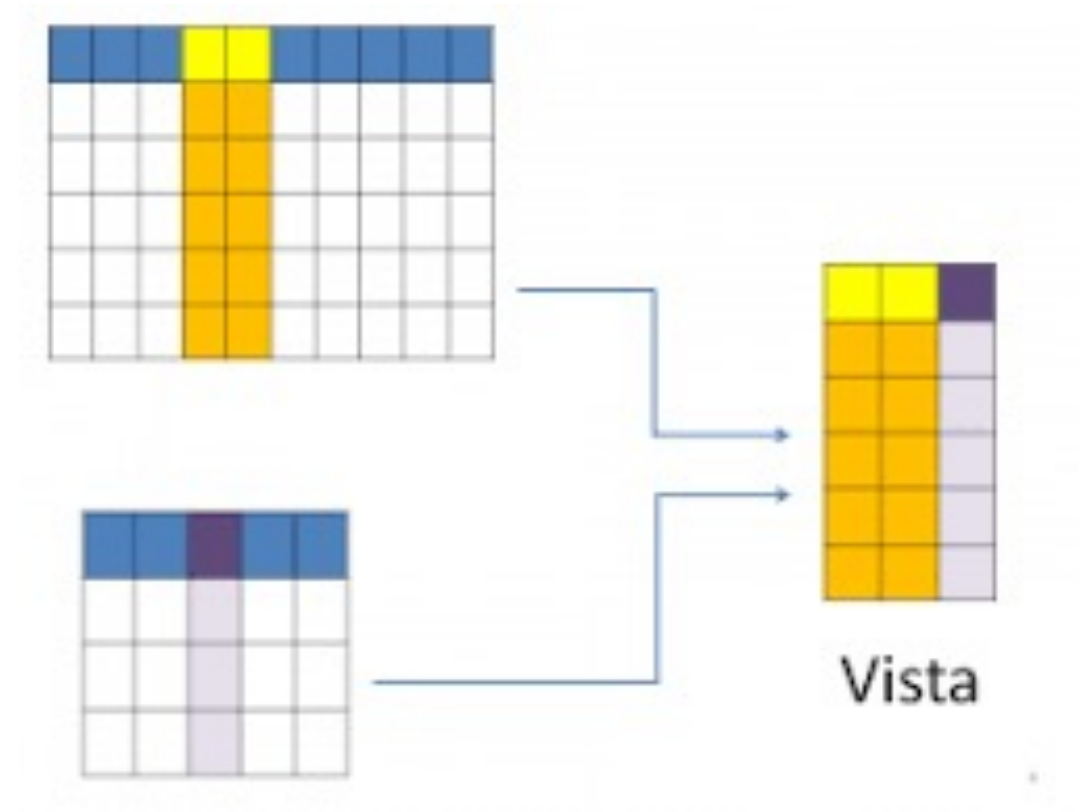
customer_id	first_name	last_name	birth_date	points	Category
1	Barbara	Lopez	1986-03-28	2273	oro
2	Ines	Gomez	1986-04-13	947	bronze
3	Alfredo	Borges	1985-02-07	2967	oro
4	Amara	Limon	1974-04-14	457	bronze
5	Clemente	Hernández	1973-11-07	3675	oro
6	María	García	1991-09-04	3073	oro
7	Valeria	Martínez	1964-08-30	1672	plata
8	Sofia	Pérez	1993-07-17	205	bronze
9	Renata	Rodríguez	1992-05-23	1486	plata
10	Victoria	Sánchez	1969-10-13	796	bronze



VIEWS

Vistas [views]

LAS VISTAS NOS PERMITEN ALMACENAR QUERYS COMPLEJOS PARA NO TENER QUE VOLVERLOS A ESTRUCTURAR



Vistas [views]

SELECT

c.client_id,

c.name,

SUM(invoice_total) **AS** total_sales

FROM clients c

JOIN invoices i **USING** (client_id)

GROUP BY client_id, name



UNIVERSIDAD
Panamericana

Vistas [views]

```
CREATE VIEW sales_by_client AS  
SELECT  
    c.client_id,  
    c.name,  
    SUM(invoice_total) AS total_sales  
FROM clients c  
JOIN invoices i USING (client_id)  
GROUP BY client_id, name
```

Actividad 06

REALIZAR LOS QUERYS QUE PERMITAN OBTENER LO SIGUIENTE

Crear una vista [clients_balance]

- Deberá mostrar el saldo por cliente el cual es calculado mediante el total de la factura y el pago realizado(por cliente)

client_id	name	Saldo
2	Carlos Muñoz	101.79
5	Carlos Name	841.63
3	Luis Slobotzky	557.46
1	Alfredo Adame	728.34

TAREA

04

REALIZAR LAS SIGUIENTES ACTIVIDADES
MEDIANTE EL USO DE FUNCIONES

TAREA 04

REALIZAR LOS QUERYS QUE PERMITAN OBTENER LO SIGUIENTE

Crear una vista [employees_above_salary]

- Obtener los empleados que ganan mas que el promedio

Crear una vista [clients_with_no_activity]

- Obtener los clientes que no tengan facturas

Crear una vista [payments_summary]

- Obtener el total de gastos generados por cada método de pago en payments