

## TALLER FUNCIONES

Nombre: Kevin Muñoz

### Parte 1:

```
DELIMITER $$
```

- ```
CREATE FUNCTION Obtener_Nombre_Completo(Cliente_id INT)
RETURNS VARCHAR(100)
DETERMINISTIC
BEGIN
    DECLARE Nombre_Completo VARCHAR(100);
    SELECT CONCAT(Nombre, " ", Apellido) INTO Nombre_Completo
    FROM Clientes
    WHERE Id = Cliente_id;
    RETURN Nombre_Completo;
END $$
```

```
DELIMITER ;
```

|   |                 |
|---|-----------------|
|   | Nombre_Completo |
| ▶ | Luis Martínez   |

- Esta función está diseñada para devolver el nombre completo de un cliente específico de la tabla Clientes, declarando un parámetro de entrada llamado Cliente\_id que es usado para identificar al cliente.

```
DELIMITER $$
```

- ```
CREATE FUNCTION Calcular_Descuento(Precio DECIMAL(10, 2), Descuento DECIMAL(10, 2))
RETURNS DECIMAL(10, 2)
DETERMINISTIC
BEGIN
    RETURN Precio - (Precio * Descuento / 100);
END $$
```

```
DELIMITER ;
```

	Precio_con_descuento
▶	900.00

- Esta función calcula el precio final de un producto después de aplicarle un descuento, en la función se declaran dos parámetros, el precio original del producto, y el porcentaje de descuento.

```
DELIMITER $$
```

- ```
CREATE FUNCTION Calcular_Total_Pedido(p_Pedido_id INT)
RETURNS DECIMAL(10, 2)
DETERMINISTIC
BEGIN
    DECLARE Total DECIMAL(10, 2);
    SELECT SUM(Cantidad * Precio_unitario) INTO Total
    FROM Detalles_Pedido
    WHERE Pedido_id = p_Pedido_id;
    RETURN Total;
END $$
```

```
DELIMITER ;
```

| Total_pedido |
|--------------|
| 699.99       |

- Esta función calcula el total de un pedido sumando el precio total de todos los productos en el pedido, la función tiene un parámetro que identificará el pedido para el cual se calculará el total

```
DELIMITER $$
```

- ```
CREATE FUNCTION Verificar_Disponibilidad(Producto_id INT, Cantidad_Solicitada INT)
RETURNS BOOLEAN
DETERMINISTIC
BEGIN
    DECLARE Stock_Actual INT;
    SELECT Stock INTO Stock_Actual
    FROM Productos
    WHERE Id = Producto_id;
    RETURN Stock_Actual >= Cantidad_Solicitada;
END $$
```

```
DELIMITER ;
```

	Disponible_1_0
▶	0

	Disponible_1_0
▶	1

- Esta función está hecha para verificar si hay suficiente stock disponible de un producto, además devuelve un valor booleano, en SQL será 1 = True, 0 = False, contiene dos parámetros, uno para identificar el producto que queremos (Producto\_id) y otro para solicitar la cantidad que se necesite.

DELIMITER \$\$

```

• CREATE FUNCTION Calcular_Antigüedad_Cliente(Cliente_id INT)
  RETURNS INT
  DETERMINISTIC
  BEGIN
    DECLARE Antigüedad INT;
    SELECT TIMESTAMPDIFF(YEAR, Fecha_registro, CURDATE()) INTO Antigüedad
    FROM Clientes
    WHERE Id = Cliente_id;
    RETURN Antigüedad;
  END $$

```

DELIMITER ;

	Antigüedad_del_cliente
▶	4

- La función calcula la antigüedad de un cliente, basada en la fecha de registro del cliente, este cálculo es la diferencia en años entre la fecha actual y la fecha de registro del cliente. Con el parámetro definido, se identificará al cliente el cual se requiere saber la antigüedad.

## Parte 2:

```
CREATE FUNCTION CalcularTotalOrden(id_orden INT)
RETURNS DECIMAL(10, 2)
DETERMINISTIC
BEGIN
    DECLARE total DECIMAL(10, 2);
    DECLARE iva DECIMAL(10, 2);

    SET iva = 0.15;

    SELECT SUM(P.precio * O.cantidad) INTO total
    FROM Ordenes O
    JOIN Productos P ON O.producto_id = P.ProductoID
    WHERE O.OrdenID = id_orden;

    SET total = total + (total * iva);

    RETURN total;
END $$
```

	TotalOrden
▶	105.15

- Esta función calcula el total de una orden aplicando un IVA del 15%, además se define un parámetro que sirve como el identificador de la orden que se desee calcular.

DELIMITER \$\$

```
CREATE FUNCTION CalcularEdad(fecha_nacimiento DATE)
RETURNS INT
DETERMINISTIC
BEGIN
    DECLARE edad INT;
    SET edad = TIMESTAMPDIFF(YEAR, fecha_nacimiento, CURDATE());
    RETURN edad;
END $$
```

DELIMITER ;

	Edad
▶	24

- La función calcula la edad de una persona, basándose en su fecha de nacimiento, la cual será ingresada gracias al parámetro definido en la función, y con el uso de TIMESTAMPDIF se hace el cálculo de la diferencia entre las fechas.

DELIMITER \$\$

CREATE FUNCTION VerificarStock(producto\_id INT)

RETURNS BOOLEAN

DETERMINISTIC

BEGIN

DECLARE stock INT;

SELECT Existencia INTO stock

FROM Productos

WHERE ProductoID = producto\_id;

IF stock > 0 THEN

RETURN TRUE;

ELSE

RETURN FALSE;

END IF;

END \$\$

DELIMITER ;

	StockDisponible
▶	1

- La función verifica si un producto tiene disponibilidad de stock, y devuelve True (1) o False (0) dependiendo de si hay o no stock para el producto, esto será identificado con el ingreso del parámetro definido que es de tipo INT, el cual pertenece al ID del producto que se quiere verificar.

```
DELIMITER $$
```

```
CREATE FUNCTION CalcularSaldo(id_cuenta INT)
```

```
RETURNS DECIMAL(10, 2)
```

```
DETERMINISTIC
```

```
BEGIN
```

```
    DECLARE saldo DECIMAL(10, 2);
```

```
    SELECT SUM(CASE
```

```
        WHEN tipo_transacción = 'deposito' THEN monto
```

```
        WHEN tipo_transacción = 'retiro' THEN -monto
```

```
        ELSE 0
```

```
    END) INTO saldo
```

```
    FROM Transacciones
```

```
    WHERE cuenta_id = id_cuenta;
```

```
    RETURN saldo;
```

```
END $$
```

```
DELIMITER ;
```

	SaldoTotal
►	500.00

- La función realiza el cálculo del saldo actual de una cuenta en base a las transacciones hechas como de depósito o retiro, con la ayuda del parámetro definido se podrá identificar la cuenta cuyo saldo se requiere hacer el cálculo. Se usa CASE para ajustar el monto según sea la transacción.