

Resolución TP UDF's Y CTE's

1-
CREATE FUNCTION ObtenerNombreCompleto(clienteID INT)
RETURNS VARCHAR(200)
DETERMINISTIC
BEGIN
RETURN (SELECT CONCAT(nombre, ', ', apellido)
FROM Clientes
WHERE cliente_id = clienteID);
END;

2-
CREATE FUNCTION PromedioVentasPorCliente(clienteID INT)
RETURNS DECIMAL(10, 2)
DETERMINISTIC
BEGIN
RETURN (SELECT AVG(valor)
FROM Ventas
WHERE cliente_id = clienteID);
END;

3-
CREATE FUNCTION CalcularAntiguedad(empleadoID INT)
RETURNS INT
DETERMINISTIC
BEGIN
RETURN YEAR(CURDATE()) - YEAR((SELECT fecha_contratacion
FROM Empleados
WHERE empleado_id = empleadoID));
END;

4-
CREATE FUNCTION CalcularComision(clienteID INT)
RETURNS DECIMAL(10, 2)
DETERMINISTIC
BEGIN
RETURN (SELECT SUM(valor) * 0.10
FROM Ventas
WHERE cliente_id = clienteID);
END;

5-
CREATE FUNCTION ObtenerJefe(departamentoID INT)
RETURNS VARCHAR(100)
DETERMINISTIC
BEGIN
RETURN (SELECT nombre
FROM Empleados
WHERE empleado_id = (SELECT jefe_id
FROM Departamentos
WHERE departamento_id = departamentoID));
END;

TUP

2do Año – Bases de Datos II

6-

```
CREATE FUNCTION TotalVentasPorFecha(fechaInicio DATE, fechaFin DATE)
RETURNS DECIMAL(10, 2)
DETERMINISTIC
BEGIN
RETURN (SELECT SUM(valor)
FROM Ventas
WHERE fecha_venta BETWEEN fechaInicio AND fechaFin);
END;
```

7- (no asustarse!)

```
CREATE FUNCTION PorcentajeSalarioDepartamento(empleadoID INT)
RETURNS DECIMAL(5, 2)
DETERMINISTIC
BEGIN
DECLARE salarioEmpleado DECIMAL(10, 2);
DECLARE totalDepartamento DECIMAL(10, 2);
SET salarioEmpleado = (SELECT salario
FROM Empleados
WHERE empleado_id = empleadoID);
SET totalDepartamento = (SELECT SUM(salario)
FROM Empleados
WHERE departamento_id = (SELECT departamento_id
FROM Empleados
WHERE empleado_id = empleadoID));
RETURN (salarioEmpleado / totalDepartamento) * 100;
END;
```

8-

```
CREATE FUNCTION ClienteSinVentas(clienteID INT)
RETURNS BOOLEAN
DETERMINISTIC
BEGIN
RETURN NOT EXISTS (SELECT 1
FROM Ventas
WHERE cliente_id = clienteID);
END;
```

9-

```
CREATE FUNCTION NumeroDeVentas(clienteID INT)
RETURNS INT
DETERMINISTIC
BEGIN
RETURN (SELECT COUNT(*)
FROM Ventas
WHERE cliente_id = clienteID);
END;
```

TUP
2do Año – Bases de Datos II

```
10-
CREATE FUNCTION ObtenerDepartamentoEmpleado(empleadoID INT)
RETURNS VARCHAR(100)
DETERMINISTIC
BEGIN
RETURN (SELECT nombre_departamento
FROM Departamentos
WHERE departamento_id = (SELECT departamento_id
FROM Empleados
WHERE empleado_id = empleadoID));
END;
```

```
1-
WITH VentasAltas AS (
SELECT *
FROM Ventas
WHERE valor > 1000
)
SELECT
COUNT(*) AS TotalVentas,
AVG(valor) AS ValorPromedio
FROM VentasAltas;
```

```
2-
WITH PromedioPorDepartamento AS (
SELECT
departamento_id,
AVG(salario) AS SalarioPromedio
FROM Empleados
GROUP BY departamento_id
)
SELECT
departamento_id,
SalarioPromedio
FROM PromedioPorDepartamento
WHERE SalarioPromedio > 4000;
```

```
3-
WITH AntiguedadEmpleados AS (
SELECT
empleado_id,
nombre,
DATEDIFF(CURDATE(), fecha_contratacion) AS DiasEnEmpresa
FROM Empleados
)
SELECT
empleado_id,
nombre,
DiasEnEmpresa
FROM AntiguedadEmpleados
WHERE DiasEnEmpresa > 5 * 365;
```

TUP
2do Año – Bases de Datos II

```
4-
WITH VentasClientes AS (
SELECT
c.cliente_id,
c.nombre,
c.apellido,
v.fecha_venta
FROM Clientes c
JOIN Ventas v ON c.cliente_id = v.cliente_id
WHERE YEAR(v.fecha_venta) = YEAR(CURDATE())
)
SELECT
cliente_id,
nombre,
apellido,
COUNT(*) AS TotalCompras
FROM VentasClientes
GROUP BY cliente_id, nombre, apellido
HAVING TotalCompras > 3;
```

```
5-
WITH TopEmpleados AS (
SELECT
nombre,
salario
FROM Empleados
ORDER BY salario DESC
LIMIT 10
)
SELECT
nombre,
salario
FROM TopEmpleados;
```

```
6-
WITH VentasPorMes AS (
SELECT
YEAR(fecha_venta) AS Año,
MONTH(fecha_venta) AS Mes,
SUM(valor) AS TotalVentas
FROM Ventas
GROUP BY YEAR(fecha_venta), MONTH(fecha_venta)
)
SELECT
Año,
Mes,
TotalVentas
FROM VentasPorMes
WHERE TotalVentas > 5000;
```

TUP

2do Año – Bases de Datos II

7- (solo a modo de ejemplo)

WITH RECURSIVE JerarquiaDepartamental AS (

-- Caso base: Seleccionar los departamentos raíz (sin jefe_id o jefe_id NULL)

SELECT

departamento_id,

nombre_departamento,

jefe_id,

1 AS Nivel

FROM Departamentos

WHERE jefe_id IS NULL

UNION ALL

-- llamado recursivo: Seleccionar los departamentos (no se termina, es solo para saber que se puede hacer)

8-

WITH ClientesDuplicados AS (

SELECT

nombre,

apellido,

COUNT(*) AS TotalDuplicados

FROM Clientes

GROUP BY nombre, apellido

HAVING COUNT(*) > 1

)

SELECT

nombre,

apellido,

TotalDuplicados

FROM ClientesDuplicados;

9-

WITH TotalVentasPorCliente AS (

SELECT

cliente_id,

SUM(valor) AS TotalVentas

FROM Ventas

GROUP BY cliente_id

),

ClientesDestacados AS (

SELECT

cliente_id,

TotalVentas

FROM TotalVentasPorCliente

WHERE TotalVentas > 10000

)

SELECT

c.cliente_id,

c.nombre,

c.apellido,

cd.TotalVentas

FROM Clientes c

JOIN ClientesDestacados cd ON c.cliente_id = cd.cliente_id;

TUP

2do Año – Bases de Datos II

10-

WITH VentasUltimoMes AS (

SELECT

cliente_id,

valor,

fecha_venta

FROM Ventas

WHERE fecha_venta >= DATE_SUB(CURDATE(), INTERVAL 1 MONTH)

)

SELECT

c.nombre,

c.apellido,

v.valor

FROM VentasUltimoMes v

JOIN Clientes c ON v.cliente_id = c.cliente_id;