TP2-T1 (Programación de Funciones en Transact SQL)

```sql
use demo;
-- 1. Función para sumar dos números
CREATE FUNCTION Suma (@num1 INT, @num2 INT)
RETURNS INT
AS
BEGIN
    DECLARE @resultado INT
    SET @resultado = @num1 + @num2
    RETURN @resultado
END;

SELECT dbo.Suma(5, 3) AS SumaResultado;

-- 2. Función para obtener la ciudad de un proveedor
CREATE FUNCTION GetCiudad (@codigoProveedor INT)
RETURNS CHAR(2)
AS
BEGIN
    DECLARE @ciudad CHAR(2)
    SELECT @ciudad = ciud FROM prov WHERE cprv = @codigoProveedor
    RETURN @ciudad
END;

SELECT dbo.GetCiudad(1) AS CiudadProveedor;

-- 3. Función para obtener el nombre de un proveedor
CREATE FUNCTION GetNombre (@codigoProveedor INT)
RETURNS CHAR(40)
AS
BEGIN
    DECLARE @nombre CHAR(40)
    SELECT @nombre = nomb FROM prov WHERE cprv = @codigoProveedor
    RETURN @nombre
END;

SELECT dbo.GetNombre(2) AS NombreProveedor;

-- 4. Función para calcular los puntos de bonificación de un proveedor
CREATE FUNCTION CalcularPuntos (@codigoProveedor INT)
RETURNS INT
AS
BEGIN
    DECLARE @puntos INT
    DECLARE @sumaImporte DECIMAL(12, 2)

    SELECT @sumaImporte = SUM(impt) FROM sumi WHERE cprv = @codigoProveedor

    IF @sumaImporte BETWEEN 1 AND 20
        SET @puntos = 10
    ELSE IF @sumaImporte BETWEEN 21 AND 50
```

```
        SET @puntos = 15
    ELSE
        SET @puntos = 20

    RETURN @puntos
END;

SELECT dbo.CalcularPuntos(1) AS PuntosProveedor;

-- 5. Función para obtener el stock de un producto en una ciudad
CREATE FUNCTION GetStock (@codigoProducto INT, @ciudad CHAR(2))
RETURNS INT
AS
BEGIN
    DECLARE @stock INT
    SELECT @stock = SUM(cant) FROM sumi s INNER JOIN alma a ON s.calm = a.calm WHERE
cprd = @codigoProducto AND ciud = @ciudad
    RETURN @stock
END;

SELECT dbo.GetStock(1, 'CB') AS StockProductoCB;

-- 6. Función para obtener el inventario valorado de un producto
CREATE FUNCTION GetInven (@codigoProducto INT)
RETURNS DECIMAL(12, 2)
AS
BEGIN
    DECLARE @inventario DECIMAL(12, 2)
    SELECT @inventario = SUM(cant * prec) FROM sumi WHERE cprd = @codigoProducto
    RETURN @inventario
END;

SELECT dbo.GetInven(1) AS InventarioProducto1;

-- 7. Función para obtener los productos existentes en una ciudad
CREATE FUNCTION GetProdxCiud (@ciudad CHAR(2))
RETURNS TABLE
AS
RETURN
(
    SELECT DISTINCT p.* FROM prod p
    INNER JOIN sumi s ON p.cprd = s.cprd
    INNER JOIN alma a ON s.calm = a.calm
    WHERE a.ciud = @ciudad
);

SELECT * FROM dbo.GetProdxCiud('CB');

-- 8. Función para obtener los proveedores que suministraron algún producto
CREATE FUNCTION GetProvxProd ()
RETURNS TABLE
AS
RETURN
(
    SELECT DISTINCT p.* FROM prov p
    INNER JOIN sumi s ON p.cprv = s.cprv
);
```

```sql
SELECT * FROM dbo.GetProvxProd();

-- 9. Función para obtener los proveedores que todavía no suministraron productos
CREATE FUNCTION GetProvNoSumi ()
RETURNS TABLE
AS
RETURN
(
    SELECT * FROM prov WHERE cprv NOT IN (SELECT DISTINCT cprv FROM sumi)
);

SELECT * FROM dbo.GetProvNoSumi();

-- 10. Función para obtener los nombres de los proveedores que suministraron algún
producto de color rojo
CREATE FUNCTION GetProvSumi ()
RETURNS TABLE
AS
RETURN
(
    SELECT DISTINCT p.* FROM prov p
    INNER JOIN sumi s ON p.cprv = s.cprv
    INNER JOIN prod pr ON s.cprd = pr.cprd
    WHERE pr.colo = 'ROJO'
);

SELECT * FROM dbo.GetProvSumi();

-- 11. Función para obtener los productos suministrados por un proveedor en un
almacén específico
CREATE FUNCTION GetProdxProv (@codigoProveedor INT, @codigoAlmacen INT)
RETURNS TABLE
AS
RETURN
(
    SELECT DISTINCT p.* FROM prod p
    INNER JOIN sumi s ON p.cprd = s.cprd
    INNER JOIN alma a ON s.calm = a.calm
    WHERE s.cprv = @codigoProveedor AND a.calm = @codigoAlmacen
);

SELECT * FROM dbo.GetProdxProv(1, 1);

-- 12. Función para obtener los productos de color rojo suministrados por un
proveedor
CREATE FUNCTION GetProdxColor (@codigoProveedor INT)
RETURNS TABLE
AS
RETURN
(
    SELECT DISTINCT p.* FROM prod p
    INNER JOIN sumi s ON p.cprd = s.cprd
    WHERE s.cprv = @codigoProveedor AND p.colo = 'ROJO'
);

SELECT * FROM dbo.GetProdxColor(1);
```

```sql
-- 13. Función para obtener los nombres de los proveedores que suministraron todos
los productos
CREATE FUNCTION GetProvTodo ()
RETURNS TABLE
AS
RETURN
(
    SELECT p.* FROM prov p
    WHERE NOT EXISTS (
        SELECT * FROM prod WHERE NOT EXISTS (
            SELECT * FROM sumi WHERE sumi.cprv = p.cprv AND sumi.cprd = prod.cprd
        )
    )
);

SELECT * FROM dbo.GetProvTodo();

-- 14. Función para obtener los nombres de los proveedores que suministraron al
menos dos productos diferentes
CREATE FUNCTION GetProvTres ()
RETURNS TABLE
AS
RETURN
(
    SELECT p.* FROM prov p
    WHERE (SELECT COUNT(DISTINCT cprd) FROM sumi WHERE cprv = p.cprv) >= 2
);

SELECT * FROM dbo.GetProvTres();

-- 15. Función para obtener los nombres de los proveedores que suministraron algún
producto fuera de su ciudad
CREATE FUNCTION GetProvOutCiud ()
RETURNS TABLE
AS
RETURN
(
    SELECT DISTINCT p.* FROM prov p
    INNER JOIN sumi s ON p.cprv = s.cprv
    INNER JOIN alma a ON s.calm = a.calm
    WHERE p.ciud != a.ciud
);

SELECT * FROM dbo.GetProvOutCiud();

-- 16. Función para obtener la cantidad más alta suministrada de un producto en una
ciudad en particular
CREATE FUNCTION GetMaxCantxCiud (@codigoProducto INT, @ciudad CHAR(2))
RETURNS INT
AS
BEGIN
    DECLARE @maxCantidad INT
    SELECT TOP 1 @maxCantidad = cant FROM sumi s INNER JOIN alma a ON s.calm =
a.calm WHERE cprd = @codigoProducto AND ciud = @ciudad ORDER BY cant DESC
    RETURN @maxCantidad
END;

SELECT dbo.GetMaxCantxCiud(1, 'CB') AS MaxCantidadProductoCB;
```

```sql
-- 17. Función para obtener la última fecha en que se suministró un producto por un
proveedor en particular
CREATE FUNCTION GetUltFecxProv (@codigoProveedor INT)
RETURNS DATE
AS
BEGIN
    DECLARE @ultimaFecha DATE
    SELECT TOP 1 @ultimaFecha = ftra FROM sumi WHERE cprv = @codigoProveedor ORDER
BY ftra DESC
    RETURN @ultimaFecha
END;

SELECT dbo.GetUltFecxProv(1) AS UltimaFechaProveedor1;

-- 18. Función para obtener la fecha en que se suministró por primera vez algún
producto de color Rojo
CREATE FUNCTION GetPrimFecxColor ()
RETURNS DATE
AS
BEGIN
    DECLARE @primeraFecha DATE
    SELECT TOP 1 @primeraFecha = MIN(ftra) FROM sumi s INNER JOIN prod p ON s.cprd =
p.cprd WHERE p.colo = 'ROJO'
    RETURN @primeraFecha
END;

SELECT dbo.GetPrimFecxColor() AS PrimeraFechaColorRojo;

-- 19. Función para obtener el importe promedio de productos suministrados por un
proveedor
CREATE FUNCTION GetPromxProv (@codigoProveedor INT)
RETURNS DECIMAL(12, 2)
AS
BEGIN
    DECLARE @promedio DECIMAL(12, 2)
    SELECT @promedio = AVG(impt) FROM sumi WHERE cprv = @codigoProveedor
    RETURN @promedio
END;

SELECT dbo.GetPromxProv(1) AS PromedioImporteProveedor1;

-- 20. Función para verificar si hay stock disponible para un producto en un almacén
específico
CREATE FUNCTION HayStock (@codigoProducto INT, @codigoAlmacen INT)
RETURNS INT
AS
BEGIN
    DECLARE @hayStock INT
    SELECT @hayStock = CASE WHEN SUM(cant) > 0 THEN 1 ELSE 0 END FROM sumi WHERE
cprd = @codigoProducto AND calm = @codigoAlmacen
    RETURN @hayStock
END;

SELECT dbo.HayStock(1, 1) AS HayStockProducto1Almacen1;
```