

```
CREATE OR REPLACE FUNCTION obtener_ranking_productos_mas_vendidos(  
    fecha_inicio DATE,  
    fecha_final DATE  
)  
RETURNS TABLE (  
    id_producto VARCHAR,  
    nombre_producto VARCHAR,  
    cantidad_vendida INT,  
    ranking BIGINT  
) AS $$  
BEGIN  
    RETURN QUERY  
    SELECT  
        p.id_producto,  
        p.nombre_producto,  
        CAST(SUM(v.cantidad_vendida) AS INT) AS cantidad_vendida,  
        dense_rank() OVER (ORDER BY SUM(v.cantidad_vendida) DESC) AS ranking  
    FROM  
        VENTA v  
    JOIN  
        PRODUCTO p ON v.id_producto = p.id_producto  
    JOIN  
        TICKET t ON v.Id_ticket = t.Id_ticket  
    WHERE  
        t.Fecha_ticket BETWEEN fecha_inicio AND fecha_final  
    GROUP BY  
        p.id_producto, p.nombre_producto  
    ORDER BY  
        ranking;  
END;  
$$ LANGUAGE plpgsql;
```

```
CREATE OR REPLACE FUNCTION resumen_ventas_farmacia(
    fecha_inicio DATE,
    fecha_fin DATE
)
RETURNS TABLE (
    nombre_farmacia VARCHAR(10),
    fecha_ticket DATE,
    metodo_pago VARCHAR(10),
    total_venta NUMERIC
) AS $$
BEGIN
    RETURN QUERY
    SELECT
        f.nombre_farmacia,
        t.fecha_ticket,
        t.metodo_pago,
        SUM(t.total) AS total_venta
    FROM
        venta v
    NATURAL JOIN
        ticket t
    NATURAL JOIN
        farmacia f
    NATURAL JOIN
        producto p
    WHERE
        (t.fecha_ticket BETWEEN fecha_inicio AND fecha_fin OR (fecha_inicio IS
NULL AND fecha_fin IS NULL))
    GROUP BY
        ROLLUP(t.fecha_ticket, f.nombre_farmacia), t.metodo_pago
    ORDER BY fecha_ticket DESC;
END;
$$ LANGUAGE plpgsql;
```

```
CREATE OR REPLACE FUNCTION desglose_ventas_producto(  
    fecha_inicio DATE,  
    fecha_fin DATE  
)  
RETURNS TABLE (  
    nombre_producto VARCHAR(20),  
    nombre_categoria VARCHAR(20),  
    total_cantidad_vendida NUMERIC,  
    total_venta NUMERIC  
) AS $$  
BEGIN  
    RETURN QUERY  
    SELECT  
        p.nombre_producto,  
        c.nombre_categoria,  
        SUM(v.cantidad_vendida) AS total_cantidad_vendida,  
        SUM(v.total_venta) AS total_venta  
    FROM  
        venta v  
    NATURAL JOIN  
        producto p  
    INNER JOIN  
        categoria c ON p.id_categoria = c.id_categoria  
    INNER JOIN  
        ticket t ON v.id_ticket = t.id_ticket  
    WHERE  
        (t.fecha_ticket BETWEEN fecha_inicio AND fecha_fin OR (fecha_inicio IS  
NULL AND fecha_fin IS NULL))  
    GROUP BY  
        CUBE (c.nombre_categoria, p.nombre_producto)  
    ORDER BY total_cantidad_vendida DESC;  
END;  
$$ LANGUAGE plpgsql;
```

```
CREATE OR REPLACE FUNCTION obtener_ranking_productos_mas_vendidos_notdense(
    fecha_inicio DATE,
    fecha_final DATE
)
RETURNS TABLE (
    id_producto VARCHAR,
    nombre_producto VARCHAR,
    cantidad_vendida INT,
    ranking BIGINT
) AS $$
BEGIN
    RETURN QUERY
    SELECT
        p.id_producto,
        p.nombre_producto,
        CAST(SUM(v.cantidad_vendida) AS INT) AS cantidad_vendida,
        rank() OVER (ORDER BY SUM(v.cantidad_vendida) DESC) AS ranking
    FROM
        VENTA v
    JOIN
        PRODUCTO p ON v.id_producto = p.id_producto
    JOIN
        TICKET t ON v.Id_ticket = t.Id_ticket
    WHERE
        t.Fecha_ticket BETWEEN fecha_inicio AND fecha_final
    GROUP BY
        p.id_producto, p.nombre_producto
    ORDER BY
        ranking;
END;
$$ LANGUAGE plpgsql;
```