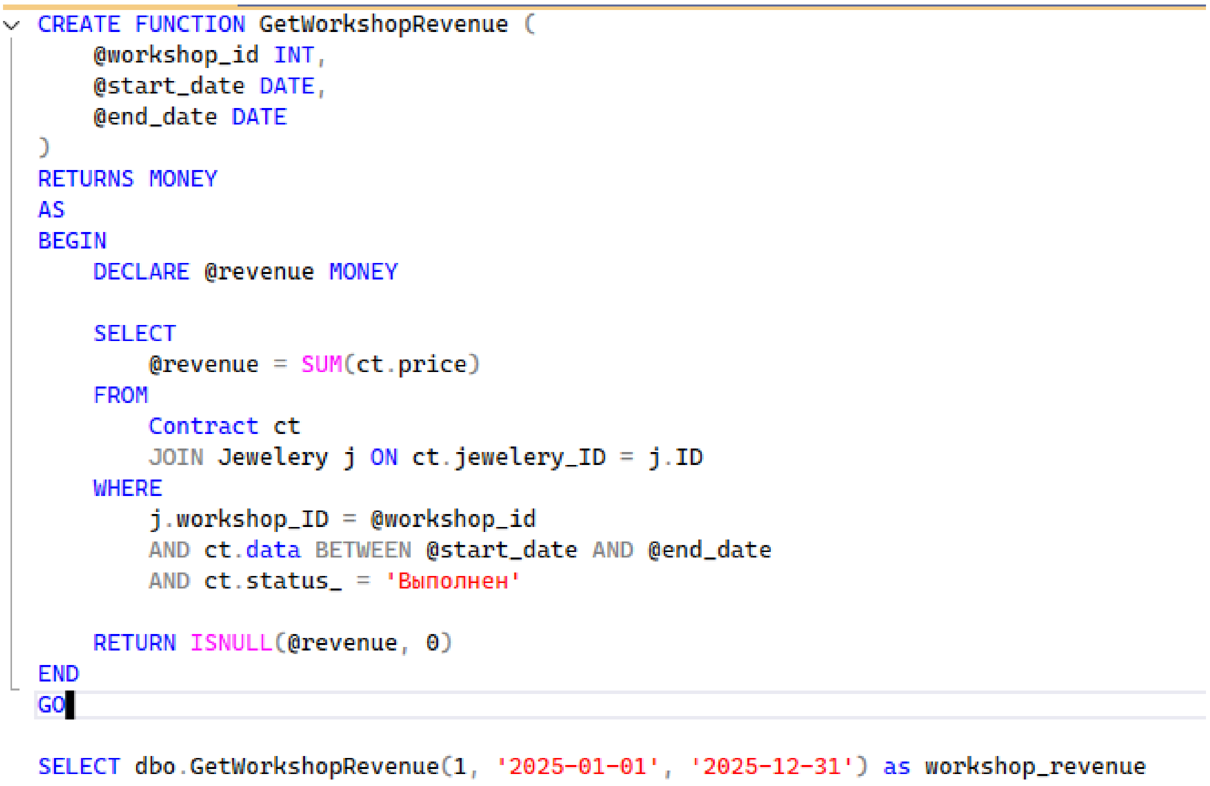
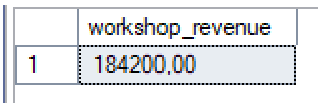
Пользовательские функции

А) Скалярная функция, подсчитывающая выручку мастерской в заданный период

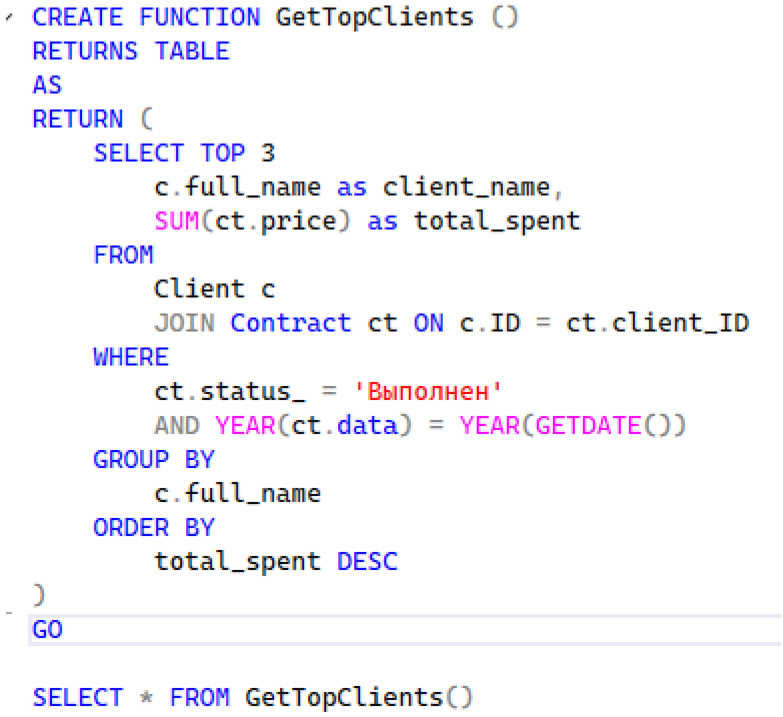


```
CREATE FUNCTION GetWorkshopRevenue (
    @workshop_id INT,
    @start_date DATE,
    @end_date DATE
)
RETURNS MONEY
AS
BEGIN
    DECLARE @revenue MONEY

    SELECT
        @revenue = SUM(ct.price)
    FROM
        Contract ct
        JOIN Jewelery j ON ct.jewelery_ID = j.ID
    WHERE
        j.workshop_ID = @workshop_id
        AND ct.data BETWEEN @start_date AND @end_date
        AND ct.status_ = 'Выполнен'

    RETURN ISNULL(@revenue, 0)
END
GO

SELECT dbo.GetWorkshopRevenue(1, '2025-01-01', '2025-12-31') as workshop_revenue
```

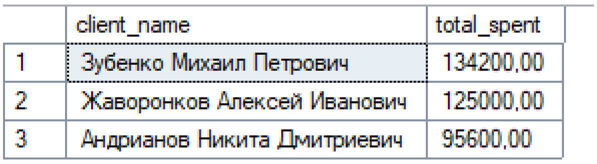


	workshop_revenue
1	184200,00

В) Inline-функция, возвращающая список клиентов, принесших наибольшую выручку мастерской с начала текущего года

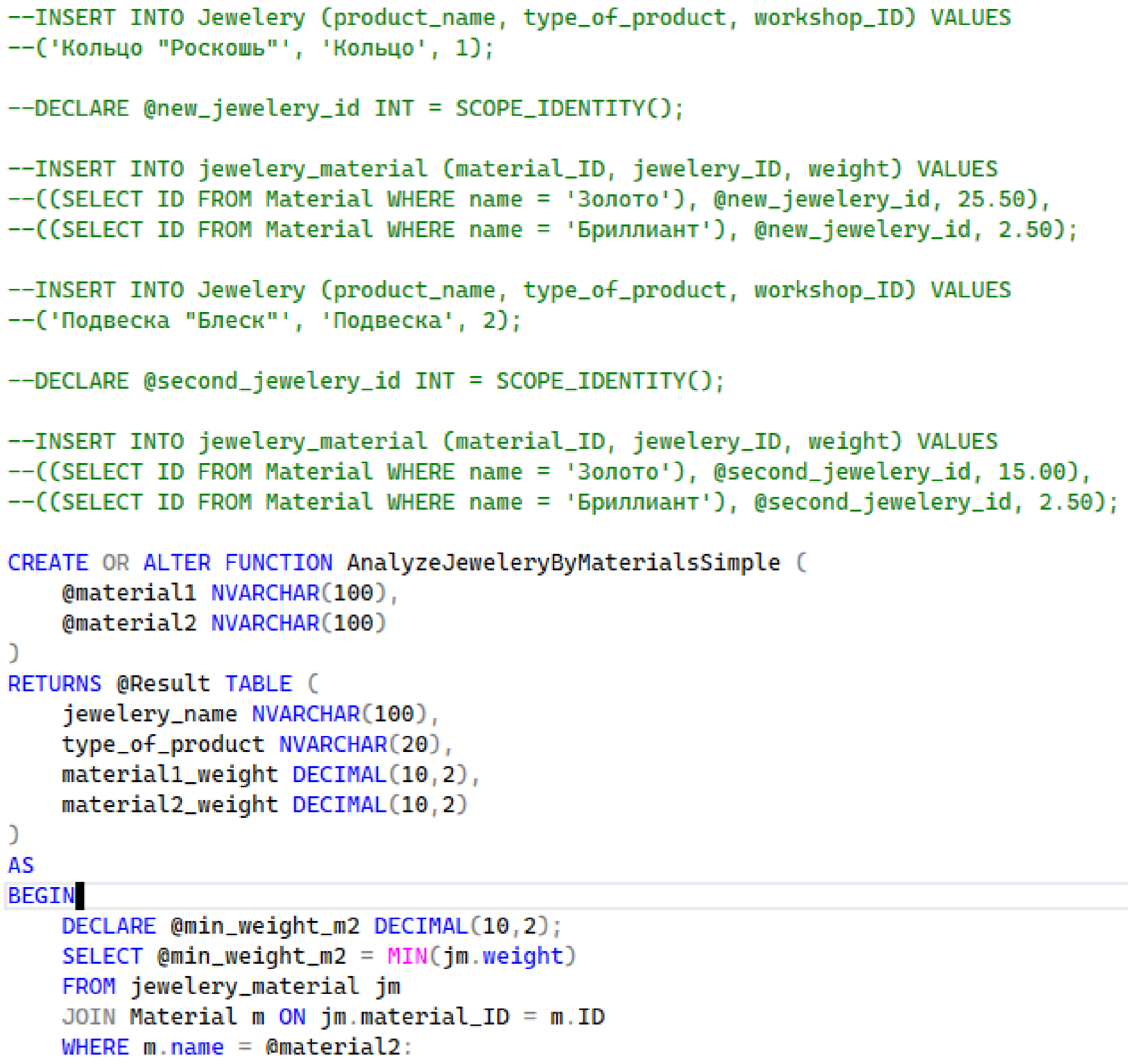


```
CREATE FUNCTION GetTopClients ()
RETURNS TABLE
AS
RETURN (
    SELECT TOP 3
        c.full_name as client_name,
        SUM(ct.price) as total_spent
    FROM
        Client c
        JOIN Contract ct ON c.ID = ct.client_ID
    WHERE
        ct.status_ = 'Выполнен'
        AND YEAR(ct.data) = YEAR(GETDATE())
    GROUP BY
        c.full_name
    ORDER BY
        total_spent DESC
)
GO

SELECT * FROM GetTopClients()
```

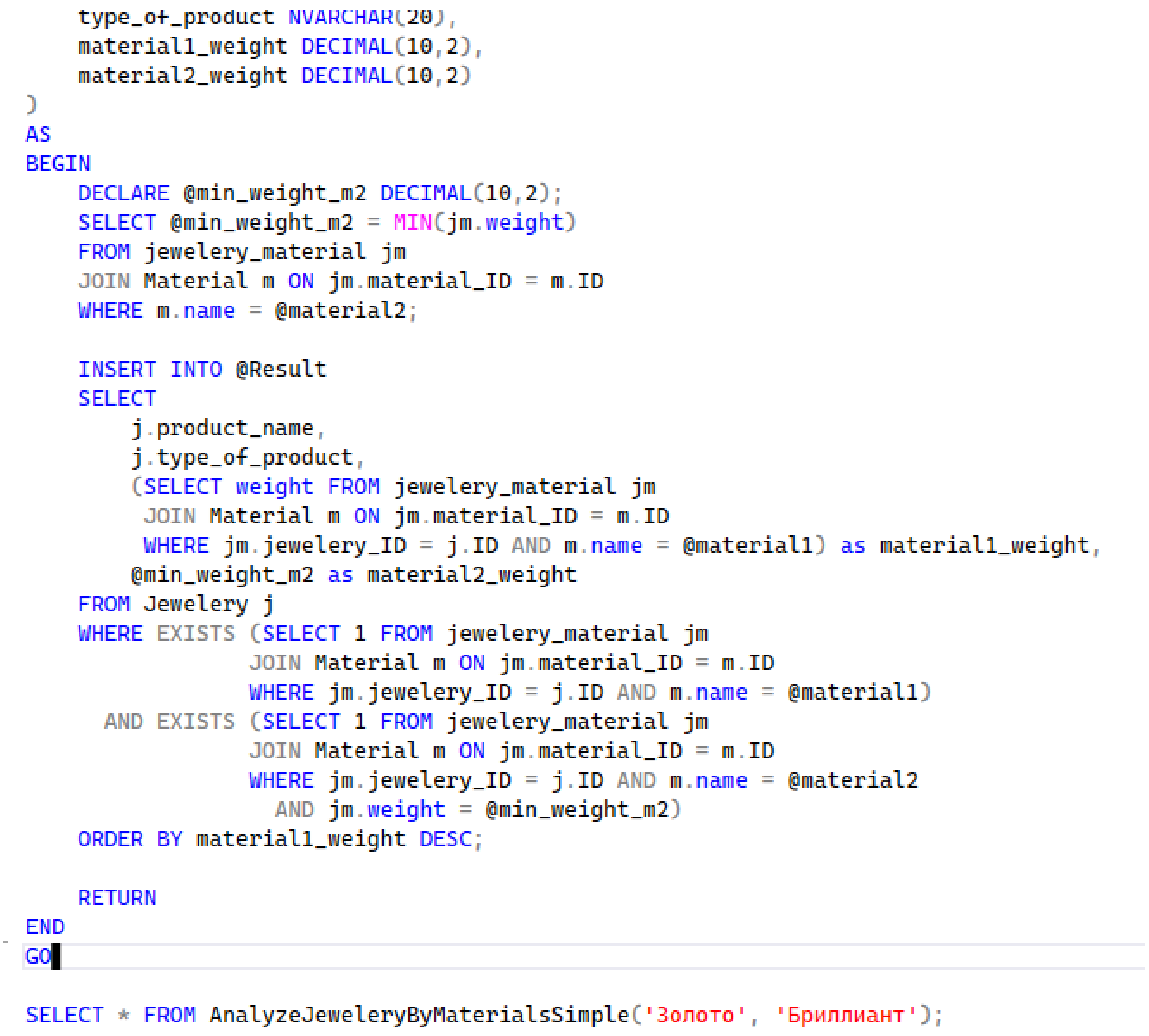


	client_name	total_spent
1	Зубенко Михаил Петрович	134200,00
2	Жаворонков Алексей Иванович	125000,00
3	Андрианов Никита Дмитриевич	95600,00

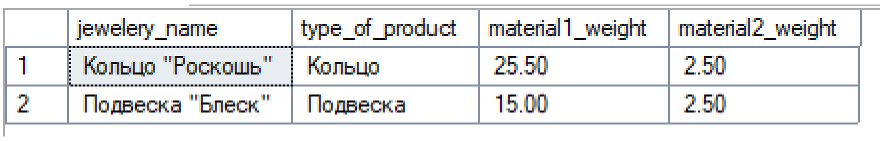
С) Multi-statement-функция, на входе получающая название 2-х материалов, и возвращающая список изделий из каталога, содержащих наибольшее количество материала 1 среди изделий, содержащих наименьшее количество материала 2



```sql
--INSERT INTO Jewelery (product_name, type_of_product, workshop_ID) VALUES
--('Кольцо "Роскошь"', 'Кольцо', 1);

--DECLARE @new_jewelery_id INT = SCOPE_IDENTITY();

--INSERT INTO jewelery_material (material_ID, jewelery_ID, weight) VALUES
--((SELECT ID FROM Material WHERE name = 'Золото'), @new_jewelery_id, 25.50),
--((SELECT ID FROM Material WHERE name = 'Бриллиант'), @new_jewelery_id, 2.50);

--INSERT INTO Jewelery (product_name, type_of_product, workshop_ID) VALUES
--('Подвеска "Блеск"', 'Подвеска', 2);

--DECLARE @second_jewelery_id INT = SCOPE_IDENTITY();

--INSERT INTO jewelery_material (material_ID, jewelery_ID, weight) VALUES
--((SELECT ID FROM Material WHERE name = 'Золото'), @second_jewelery_id, 15.00),
--((SELECT ID FROM Material WHERE name = 'Бриллиант'), @second_jewelery_id, 2.50);

CREATE OR ALTER FUNCTION AnalyzeJeweleryByMaterialsSimple (
    @material1 NVARCHAR(100),
    @material2 NVARCHAR(100)
)
RETURNS @Result TABLE (
    jewelery_name NVARCHAR(100),
    type_of_product NVARCHAR(20),
    material1_weight DECIMAL(10,2),
    material2_weight DECIMAL(10,2)
)
AS
BEGIN
    DECLARE @min_weight_m2 DECIMAL(10,2);
    SELECT @min_weight_m2 = MIN(jm.weight)
    FROM jewelery_material jm
    JOIN Material m ON jm.material_ID = m.ID
    WHERE m.name = @material2;
```



```
    type_of_product NVARCHAR(20),
    material1_weight DECIMAL(10,2),
    material2_weight DECIMAL(10,2)
)
AS
BEGIN
    DECLARE @min_weight_m2 DECIMAL(10,2);
    SELECT @min_weight_m2 = MIN(jm.weight)
    FROM jewelery_material jm
    JOIN Material m ON jm.material_ID = m.ID
    WHERE m.name = @material2;

    INSERT INTO @Result
    SELECT
        j.product_name,
        j.type_of_product,
        (SELECT weight FROM jewelery_material jm
         JOIN Material m ON jm.material_ID = m.ID
         WHERE jm.jewelery_ID = j.ID AND m.name = @material1) as material1_weight,
        @min_weight_m2 as material2_weight
    FROM Jewelery j
    WHERE EXISTS (SELECT 1 FROM jewelery_material jm
                 JOIN Material m ON jm.material_ID = m.ID
                 WHERE jm.jewelery_ID = j.ID AND m.name = @material1)
      AND EXISTS (SELECT 1 FROM jewelery_material jm
                 JOIN Material m ON jm.material_ID = m.ID
                 WHERE jm.jewelery_ID = j.ID AND m.name = @material2
                   AND jm.weight = @min_weight_m2)
    ORDER BY material1_weight DESC;

    RETURN
END
GO

SELECT * FROM AnalyzeJeweleryByMaterialsSimple('Золото', 'Бриллиант');
```



	jewelery_name	type_of_product	material1_weight	material2_weight
1	Кольцо "Роскошь"	Кольцо	25.50	2.50
2	Подвеска "Блеск"	Подвеска	15.00	2.50