

Триггеры

- а) Триггер любого типа на добавление договора - если клиент не 1-й раз делает заказ, то делаем ему скидку: если он был у нас 4 и более раз - 10%, если 2 и более - 5%.

```
CREATE TRIGGER TR_Contract_Insert
ON Contract
AFTER INSERT
AS
BEGIN
    DECLARE @client_id INT, @contract_count INT, @new_discount_id INT
    DECLARE client_cursor CURSOR FOR
        SELECT DISTINCT client_ID FROM inserted

    OPEN client_cursor
    FETCH NEXT FROM client_cursor INTO @client_id

    WHILE @@FETCH_STATUS = 0
    BEGIN
        SELECT @contract_count = COUNT(*)
        FROM Contract
        WHERE client_ID = @client_id AND status_ = 'Выполнен'

        IF @contract_count >= 4
            SET @new_discount_id = (SELECT ID FROM Discount WHERE denomination = 'Постоянный')
        ELSE IF @contract_count >= 2
            SET @new_discount_id = (SELECT ID FROM Discount WHERE denomination = 'Новичок')
        ELSE
            SET @new_discount_id = (SELECT ID FROM Discount WHERE denomination = 'Новичок')

        --или ELSE
        --      SET @new_discount_id = NULL тк тогда будет скидка 0 при 1 заказе, но у меня базовая новичок
        --если с NULL то дальше так IF @new_discount_id IS NOT NULL
        --BEGIN обновляем скидку если она есть

        UPDATE Contract
        SET discount_ID = @new_discount_id
        WHERE client_ID = @client_id AND ID IN (SELECT ID FROM inserted)

        FETCH NEXT FROM client_cursor INTO @client_id
    END

    CLOSE client_cursor
    DEALLOCATE client_cursor
END
GO
```

- б) Последующий триггер на изменение веса материала для какого-то изделия – если есть клиенты, для которых это изделие находится на этапе изготовления, изменение не производим

```
CREATE OR ALTER TRIGGER TR_JeweleryMaterial_Update
ON jewelery_material
AFTER UPDATE
AS
BEGIN
    IF UPDATE(weight)
    BEGIN
        DECLARE @jewelery_id INT, @material_id INT, @old_weight DECIMAL(10,2), @new_weight DECIMAL(10,2)
        DECLARE update_cursor CURSOR FOR
            SELECT i.jewelery_ID, i.material_ID, d.weight, i.weight
            FROM inserted i
            JOIN deleted d ON i.jewelery_ID = d.jewelery_ID AND i.material_ID = d.material_ID

        OPEN update_cursor
        FETCH NEXT FROM update_cursor INTO @jewelery_id, @material_id, @old_weight, @new_weight

        WHILE @@FETCH_STATUS = 0
        BEGIN
            IF EXISTS (SELECT 1 FROM Contract WHERE jewelery_ID = @jewelery_id AND status_ = 'В работе')
            BEGIN
                UPDATE jewelery_material
                SET weight = @old_weight
                WHERE jewelery_ID = @jewelery_id AND material_ID = @material_id

                PRINT 'Ошибка: Изделие ID ' + CAST(@jewelery_id AS NVARCHAR(10)) +
                    ' находится в работе, изменение веса отменено'
            END
            ELSE
            BEGIN
                PRINT 'Вес материала успешно обновлен для изделия ID: ' + CAST(@jewelery_id AS NVARCHAR(10))
            END

            FETCH NEXT FROM update_cursor INTO @jewelery_id, @material_id, @old_weight, @new_weight
        END

        CLOSE update_cursor
        DEALLOCATE update_cursor
    END
    WHILE @@FETCH_STATUS = 0
    BEGIN
        IF EXISTS (SELECT 1 FROM Contract WHERE jewelery_ID = @jewelery_id AND status_ = 'В работе')
        BEGIN
            UPDATE jewelery_material
            SET weight = @old_weight
            WHERE jewelery_ID = @jewelery_id AND material_ID = @material_id

            PRINT 'Ошибка: Изделие ID ' + CAST(@jewelery_id AS NVARCHAR(10)) +
                ' находится в работе, изменение веса отменено'
        END
        ELSE
        BEGIN
            PRINT 'Вес материала успешно обновлен для изделия ID: ' + CAST(@jewelery_id AS NVARCHAR(10))
        END

        FETCH NEXT FROM update_cursor INTO @jewelery_id, @material_id, @old_weight, @new_weight
    END

    CLOSE update_cursor
    DEALLOCATE update_cursor
END
END
GO
SELECT * FROM jewelery_material;
UPDATE jewelery_material SET weight = 12.00 WHERE jewelery_ID = 3 AND material_ID = 1;
```

	material_ID	jewelery_ID	weight
1	1	1	12.00
2	1	2	15.75
3	1	3	5.50
4	2	3	25.30
5	3	4	8.20
6	2	5	18.60
7	4	6	12.40
8	1	7	22.10
9	2	8	12.75
10	5	8	30.25
11	3	9	9.80
12	2	10	11.35
13	4	11	7.90
14	5	12	14.20
15	1	14	25.50
16	4	14	2.50
17	1	15	15.00
18	4	15	2.50

	material_ID	jewelery_ID	weight
1	1	1	12.00
2	1	2	15.75
3	1	3	12.00
4	2	3	25.30
5	3	4	8.20
6	2	5	18.60
7	4	6	12.40
8	1	7	22.10
9	2	8	12.75
10	5	8	30.25
11	3	9	9.80
12	2	10	11.35
13	4	11	7.90
14	5	12	14.20
15	1	14	25.50
16	4	14	2.50
17	1	15	15.00
18	4	15	2.50

- с) Замещающий триггер на операцию удаления изделия из каталога – если доход от изготовления этого изделия минимален, и оно не находится на этапе изготовления для какого-то клиента, то удаление производим, в противном случае – выдаем сообщение о невозможности удаления

```
CREATE TRIGGER TR_Jewelry_Delete
ON Jewelry
INSTEAD OF DELETE
AS
BEGIN
    DECLARE @jewelry_id INT, @product_name NVARCHAR(100), @total_revenue MONEY
    DECLARE delete_cursor CURSOR FOR
        SELECT ID, product_name FROM deleted

    OPEN delete_cursor
    FETCH NEXT FROM delete_cursor INTO @jewelry_id, @product_name

    WHILE @@FETCH_STATUS = 0
    BEGIN
        SELECT @total_revenue = SUM(price)
        FROM Contract
        WHERE jewelry_ID = @jewelry_id AND status_ = 'Выполнен'

        IF (@total_revenue IS NULL OR @total_revenue < 50000)
            AND NOT EXISTS (SELECT 1 FROM Contract WHERE jewelry_ID = @jewelry_id AND status_ = 'В работе')
        BEGIN
            DELETE FROM jewelry_material WHERE jewelry_ID = @jewelry_id
            DELETE FROM Contract WHERE jewelry_ID = @jewelry_id
            DELETE FROM Jewelry WHERE ID = @jewelry_id

            PRINT 'Изделие "' + @product_name + '" успешно удалено'
        END
        ELSE
        BEGIN
            PRINT 'Невозможно удалить изделие "' + @product_name + '". Причина: '
            IF EXISTS (SELECT 1 FROM Contract WHERE jewelry_ID = @jewelry_id AND status_ = 'В работе')
                PRINT '- Изделие находится в работе'
            IF @total_revenue >= 50000
                PRINT '- Доход от изделия превышает допустимый порог'
        END
    END

    FETCH NEXT FROM delete_cursor INTO @jewelry_id, @product_name
END

CLOSE delete_cursor
DEALLOCATE delete_cursor
END
GO
--находим какие можно удалить
SELECT
    j.ID,
```

```

CLOSE delete_cursor
DEALLOCATE delete_cursor
END
GO
--находим какие можно удалить
SELECT
    j.ID,
    j.product_name,
    ISNULL(SUM(CASE WHEN c.status_ = 'Выполнен' THEN c.price ELSE 0 END), 0) as total_revenue,
    CASE
        WHEN EXISTS (SELECT 1 FROM Contract WHERE jewelery_ID = j.ID AND status_ = 'В работе')
        THEN 'ДА'
        ELSE 'НЕТ'
    END as in_work
FROM Jewelery j
LEFT JOIN Contract c ON j.ID = c.jewelery_ID
GROUP BY j.ID, j.product_name
HAVING ISNULL(SUM(CASE WHEN c.status_ = 'Выполнен' THEN c.price ELSE 0 END), 0) < 50000
    AND NOT EXISTS (SELECT 1 FROM Contract WHERE jewelery_ID = j.ID AND status_ = 'В работе');
-- Нашли напр возьмем 7 и удалим
SELECT 'ДО удаления:' as stage;
SELECT * FROM Jewelery WHERE ID = 7;

DELETE FROM Jewelery WHERE ID = 7;

SELECT 'ПОСЛЕ удаления:' as stage;
SELECT * FROM Jewelery WHERE ID = 7;

```

	stage
1	ДО удаления:

	ID	product_name	type_of_product	workshop_ID
1	7	Цепочка "Рэперская"	Цепочка	5

	stage
1	ПОСЛЕ удаления:

	ID	product_name	type_of_product	workshop_ID