

3. Создать 3 триггера:

а) Триггер любого типа на добавление корзины – если сумма товаров > 500р. и нет карты, то добавляем карту с 2%-ной скидкой, если сумма > 2000р. и нет карты, то добавляем карту с 4%-ной скидкой, если сумма > 2000р. и карта есть, то устанавливаем скидку в 4%

```
CREATE TRIGGER AddingCard
ON Sale
AFTER INSERT -- срабатывает после добавления записи
AS
BEGIN
    DECLARE @sale_id INT, @total_amount DECIMAL(12,2),
    @customer_id INT, @card_id INT, @new_card_id INT;

    -- создание курсора
    DECLARE sale_cursor CURSOR FOR
    SELECT id, total_amount, customer_id
    FROM inserted

    OPEN sale_cursor
    FETCH NEXT FROM sale_cursor INTO @sale_id, @total_amount, @customer_id

    -- цикл для чтения данных
    WHILE @@FETCH_STATUS = 0
    BEGIN
        -- получаем card_id из Customer (если клиент существует)
        IF @customer_id IS NOT NULL
        BEGIN
            SELECT @card_id = card_id
            FROM Customer
            WHERE id = @customer_id;

            -- обработка корзины
            IF @total_amount > 2000 AND (@card_id IS NULL OR @customer_id IS NULL)
            BEGIN
                -- добавляем карту с 4% скидкой
                INSERT INTO DiscountCard (discount_rate) VALUES (4.00)
                SET @new_card_id = SCOPE_IDENTITY();
                -- привязываем карту к клиенту (если клиент существует)
                IF @customer_id IS NOT NULL
                    UPDATE Customer SET card_id = @new_card_id WHERE id = @customer_id;
                PRINT 'Добавлена карта с 4% скидкой для корзины ID: ' + CAST(@sale_id AS VARCHAR);
            END
            ELSE IF @total_amount > 500 AND (@card_id IS NULL OR @customer_id IS NULL)
            BEGIN
                -- Добавляем карту с 2% скидкой
                INSERT INTO DiscountCard (discount_rate) VALUES (2.00)
                SET @new_card_id = SCOPE_IDENTITY();

                IF @customer_id IS NOT NULL
                    UPDATE Customer SET card_id = @new_card_id WHERE id = @customer_id;
                PRINT 'Добавлена карта с 2% скидкой для корзины ID: ' + CAST(@sale_id AS VARCHAR);
            END
            ELSE IF @total_amount > 2000 AND @card_id IS NOT NULL
            BEGIN
                -- Обновляем существующую карту на 4%
                UPDATE DiscountCard SET discount_rate = 4.00 WHERE id = @card_id;
                PRINT 'Обновлена карта на 4% скидку для корзины ID: ' + CAST(@sale_id AS VARCHAR);
            END
        END
        ELSE
            PRINT 'Корзина ID: ' + CAST(@sale_id AS VARCHAR) + ' - нет клиента, карта не выдана';

        -- чтение следующей записи или конец
        FETCH NEXT FROM sale_cursor INTO @sale_id, @total_amount, @customer_id;
    END

    CLOSE sale_cursor;
    DEALLOCATE sale_cursor;
END;
GO
```

```
INSERT INTO Sale (total_amount, discount_amount, final_amount, customer_id, employee_id) VALUES
(600, 0, 600, 15, 1), -- карта 2%
(2500, 0, 2500, 3, 1), -- карта 4%
(3000, 0, 3000, 1, 1), -- обновит карту на 4%
(1000, 0, 1000, NULL, 1); -- карта не выдана
```

Сообщения
Корзина ID: 19 - нет клиента, карта не выдана
(Затронута 1 строка)
Обновлена карта на 4% скидку для корзины ID: 18
(Затронута 1 строка)
Добавлена карта с 2% скидкой для корзины ID: 17
(Затронута 1 строка)
Обновлена карта на 4% скидку для корзины ID: 16
(Затронута 1 строка)
Добавлена карта с 2% скидкой для корзины ID: 15
(Затронута 1 строка)
(Затронуто строк: 4)

b) Последующий триггер на изменение цены продажи товара – если цена продажи меньше, чем закупочная цена, изменение отменяется, выводится соотв. сообщение

```
CREATE TRIGGER PriceChange
ON Products
AFTER UPDATE
AS
BEGIN
    IF EXISTS (SELECT 1
                FROM inserted i
                WHERE selling_price < purchase_price)
        BEGIN
            ROLLBACK TRANSACTION; -- отмена изменения
            RAISERROR('Цена продажи не может быть ниже закупочной цены! Изменение отменено.', 16, 1);
        END
    END;
GO

-- цена закупки = 45
UPDATE Products
SET selling_price = 40 -- ошибка
WHERE id = 1
```

Сообщения

```
Сообщение: 50000, уровень: 16, состояние: 1, процедура: PriceChange, строка: 11 [строка начала пакета: 324]
Цена продажи не может быть ниже закупочной цены! Изменение отменено.
Сообщение 3609, уровень 16, состояние 1, строка 326
Транзакция завершилась в триггере. Выполнение пакета прервано.
```

c) Замещающий триггер на операцию удаления – при удалении товара из корзины (возврат товара) вернуть его в супермаркет, пересчитать стоимость корзины

```
CREATE TRIGGER ReturnProduct
ON SaleItem
INSTEAD OF DELETE -- выполняется вместо задания(удаления)
AS
BEGIN
    -- возвращаем товары на склад
    UPDATE Products
    SET stock_quantity = stock_quantity + d.quantity
    FROM Products p
    JOIN deleted d ON p.id = d.product_id;

    -- удаляем товары из корзины
    DELETE FROM SaleItem
    WHERE id IN (SELECT id FROM deleted);

    -- пересчитываем стоимость корзины
    UPDATE Sale
    SET total_amount = ISNULL(si.total_price, 0),
        final_amount = ISNULL(si.total_price, 0) - s.discount_amount
    FROM Sale s
    JOIN (SELECT sale_id, SUM(total_price) as total_price
          FROM SaleItem
          GROUP BY sale_id) si ON s.id = si.sale_id
    WHERE s.id IN (SELECT sale_id FROM deleted);

    DECLARE @processed_count INT = (SELECT COUNT(*) FROM deleted);
    PRINT 'Удалено позиций: ' + CAST(@processed_count AS VARCHAR) +
        '. Товары возвращены на склад, корзина пересчитана.';

END;
GO

DELETE FROM SaleItem WHERE sale_id = 2 AND product_id IN (2, 4);
```

Сообщения

```
(Затронуто строк: 2)
(Затронуто строк: 2)
(Затронута 1 строка)
Удалено позиций: 2. Товары возвращены на склад, корзина пересчитана.
(Затронуто строк: 2)
```