

4 этап.

7.

7.1.

```
CREATE PROCEDURE Данные_страны
AS
SELECT Наименование_страны FROM Страны
WHERE Код_страны = 1
GO
```

```
CREATE PROCEDURE Данные_цены_товара
AS
SELECT Цена FROM Товары
WHERE Код_товара = 1
GO
```

```
CREATE PROCEDURE Данные_заказчика
AS
SELECT Адрес FROM Заказчики
WHERE Код_заказчика = 1
GO
```

7.2.

```
CREATE FUNCTION Страны (Наименование CHAR(20))
RETURNS CHAR(50) DETERMINISTIC
RETURN CONCAT('Страна => ',Наименование);
```

```
CREATE FUNCTION Данные_заказчика
RETURNS one
BEGIN
    DECLARE @Адрес one
    SELECT @Адрес = Адрес FROM Заказчики
```

```
WHERE Код_заказчика
RETURN @Адрес
END
GO
```

```
CREATE FUNCTION Выбор_заказов
RETURNS TABLE
AS
RETURN
SELECT Общая_стоимость, Дата_заказа FROM Заказы З INNER JOIN Товары
Т
ON З.Код_товара = Т.Код_товара
WHERE Т.Цена = @Цена

WHERE Код_товара = 1
GO
```

```
CREATE FUNCTION Данные_страны
RETURNS @CountryRegions TABLE
(
    Код_страны          INT(100),
    Наименование_страны TEXT
)
BEGIN
DECLARE @rowset TABLE (
    Код_страны          INT(100),
    Город_штаб_квартиры TEXT,
    Наименование_страны TEXT
)

INSERT INTO @rowset (Код_страны, Город_штаб_квартиры,
Наименование_страны)
    SELECT Код_страны, Город_штаб_квартиры, Наименование_страны
        FROM Страна
            WHERE Country = @Country;

UPDATE @rowset SET Данные_страны = 1 WHERE Наименование_страны = '';

INSERT INTO @CountryRegions
    SELECT Код_страны, Наименование_страны
        FROM @rowset

RETURN
END
GO
```

7.3.

```
CREATE TRIGGER One BEFORE INSERT ON Товары
FOR EACH ROW SET @sum = @sum + NEW.Цена;
```

```
SET @sum = 0;
INSERT INTO Товары VALUES(4, 4, 'feltre_4', "", 600);
```

```
SELECT @sum AS 'Новый товар';
```

```
CREATE TRIGGER Two BEFORE INSERT ON Заказчики
FOR EACH ROW SET @sum = @sum + NEW.Телефон;
```

```
SET @sum = 0;
INSERT INTO Заказчики VALUES(4, 'Иванушкин', 'Варшава', '375298675312');
```

```
SELECT @sum AS 'Новый заказчик';
```

```
CREATE TRIGGER Three BEFORE INSERT ON Виды_товаров
FOR EACH ROW SET @sum = @sum + NEW.Наименование;
```

```
SET @sum = 0;
INSERT INTO Виды_товаров VALUES(4, 'Стол', "");
```

```
SELECT @sum AS 'Новый вид товара';
```

7.4.

```
CREATE TRIGGER Products
ON Товары
FOR UPDATE AS
IF UPDATE(Цена)
BEGIN
    DECLARE @ProductCode INT, @Price MONEY, @PriceBYN MONEY
    DECLARE myCursor CURSOR LOCAL STATIC FOR
        SELECT inserted.Код_товара, inserted.Цена,
            inserted.Цена * Currencies.Rate
        FROM inserted INNER JOIN Currencies
            ON inserted.Currency = Currencies.CurrencyID
```

```
OPEN myCursor
  FETCH FIRST FROM myCursor INTO @ProductCode, @Price, @PriceBYN
  WHILE @@FETCH_STATUS = 0
    BEGIN
      UPDATE Currency
      SET Cost = Quantity * @Price,
          CostBYN = Quantity * @PriceBYN
      WHERE Product = @ProductCode
      FETCH NEXT FROM myCursor INTO @ProductCode, @Price,
@PriceBYN
    END
  CLOSE myCursor
  DEALLOCATE myCursor
END
GO
```