

```

---Query 1

drop PROC GET_CUST
go

CREATE PROC GET_CUST
    @ID NVARCHAR,
    @COMPANY NVARCHAR(20) OUTPUT,
    @FIO NVARCHAR(20) OUTPUT,
    @CITY NVARCHAR(20) OUTPUT

AS

SELECT @COMPANY = COMPANY
FROM CUSTOMERS
WHERE CUSTOMERS.CUST_NUM = @ID
SELECT @CITY = CITY
FROM OFFICES JOIN CUSTOMERS ON CUSTOMERS.CUST_NUM = OFFICES.CUST_NUM
WHERE CUSTOMERS.CUST_NUM = @ID
SELECT @FIO = FIO
FROM CUSTOMERS
    JOIN SALESPERS ON CUSTOMERS.CUST_REP = SALESPERS.EMPL_NUM
WHERE CUSTOMERS.CUST_NUM = @ID

DROP PROC GET_CUST1
GO
CREATE PROC GET_CUST1
    @ID1 NVARCHAR(20)
AS
DECLARE @COMP NVARCHAR(20), @CIT NVARCHAR(20), @FI NVARCHAR(20), @ID NVARCHAR(20)
SET @ID = @ID1
EXEC GET_CUST @ID,@COMP OUTPUT,@FI OUTPUT,@CIT OUTPUT
GO

SELECT @ID AS 'ID',
    @COMP as 'COMPANY',
    @FI as 'FIO',
    @CIT as 'CITY'
GO

EXEC GET_CUST1 211
GO
---

---Query 2

DROP PROC CHK_TOT
GO

```

```

CREATE PROC CHK_TOT
    @ID NVARCHAR(20)
AS
DECLARE @SUM INT
SELECT @SUM = sum(AMOUNT)
FROM CUSTOMERS
where CUST_NUM = @ID
UPDATE OFFICES
    SET STATUS = IIF(@SUM > 30000, 'Большой объем заказов', 'Небольшой объем заказов')
    WHERE OFFICES.CUST_NUM = @ID
RETURN @SUM
GO

exec CHK_TOT 211
GO
---

---Query 3
DROP PROC INSERT_INTO_OFFICES
GO
CREATE PROC INSERT_INTO_OFFICES
    @IDOFF BIGINT,
    @TARGET FLOAT,
    @CITY NVARCHAR(10),
    @CUST_NUM NVARCHAR(20),
    @STATUS NVARCHAR(30)
AS
BEGIN TRANSACTION
IF @CUST_NUM IN (SELECT CUST_NUM
FROM CUSTOMERS)
BEGIN
    INSERT INTO OFFICES
        (IDOFF, TARGET, CITY, CUST_NUM, STATUS)
    VALUES
        (@IDOFF, @TARGET, @CITY, @CUST_NUM, @STATUS)
END
ELSE
BEGIN
    ROLLBACK TRANSACTION
    RETURN
END
DECLARE @TA INT
EXEC @TA = CHK_TOT @CUST_NUM
IF @TA > CONVERT(money, 20000)

```

```

BEGIN
    ROLLBACK TRANSACTION
END
ELSE
begin
    UPDATE SALESPERS

SET QUOTA = QUOTA + @TARGET
WHERE SALESPERS.EMPL_NUM IN (
    SELECT DISTINCT CUST_REP
    FROM CUSTOMERS
    WHERE CUSTOMERS.CUST_NUM = @CUST_NUM)
    COMMIT TRANSACTION
END
GO

EXEC INSERT_INTO_OFFICES 5, 10, 'Лондон', 211, null
GO
---

---Query 4
CREATE TRIGGER EDITED_EMPL
ON SALESPERS
AFTER UPDATE
AS
UPDATE CUSTOMERS
SET CUST_REP = (SELECT EMPL_NUM
FROM inserted)
WHERE CUSTOMERS.CUST_REP = (SELECT EMPL_NUM
FROM deleted)
GO

UPDATE SALESPERS
SET EMPL_NUM = 5
WHERE FIO = 'Петров'
GO
---

---Query 5
CREATE TRIGGER CHK_DATE
ON CUSTOMERS
AFTER INSERT , UPDATE
as
BEGIN
    IF (SELECT DATEPART(DAY, (SELECT DATA_ORDER

```

```
FROM inserted))) > 15
BEGIN
    ROLLBACK TRANSACTION
END
END
GO

INSERT INTO CUSTOMERS
    (IDCUST,CUST_NUM,COMPANY,CUST_REP,CREDIT_LIM,DATA_ORDER,AMOUNT)
VALUES(5, 214, 'Шредер', 121, 15, '2004-17-12', 1000)
GO
---
```