StockTracking System

Presentation Highlights

Project Description
Flow Chart
Tables
Procedures Examples
Trigger Example
Views Examples
Functions Examples
Index Examples
Maintenance Plans
The Team

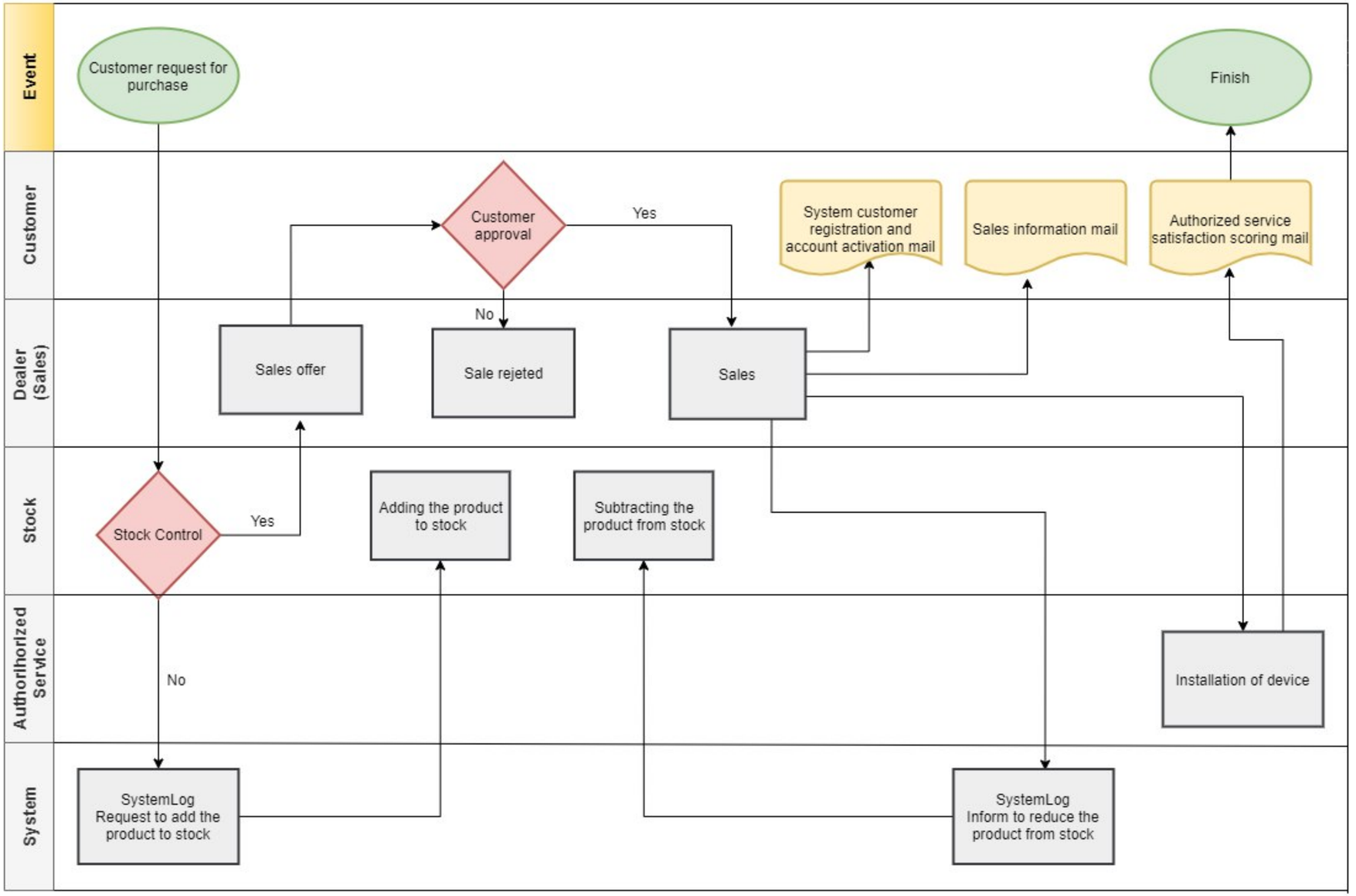
Project Description

By checking the stock, we can easily find the products purchased by the customers, how many products are left in the dealers, and which product is sold the most.

Flow Chart

Process Flow Diagram

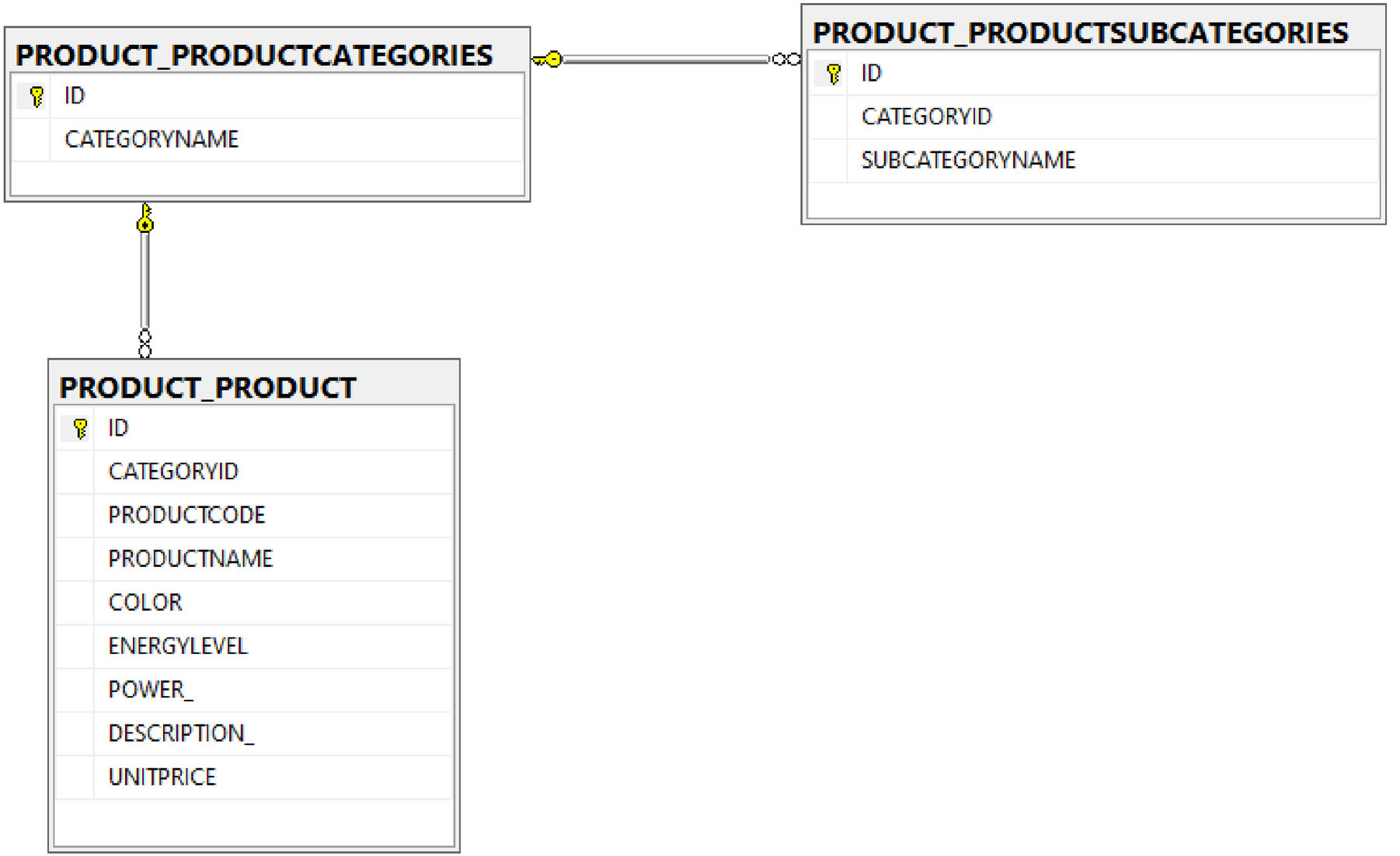
Sales Stock Tracking System



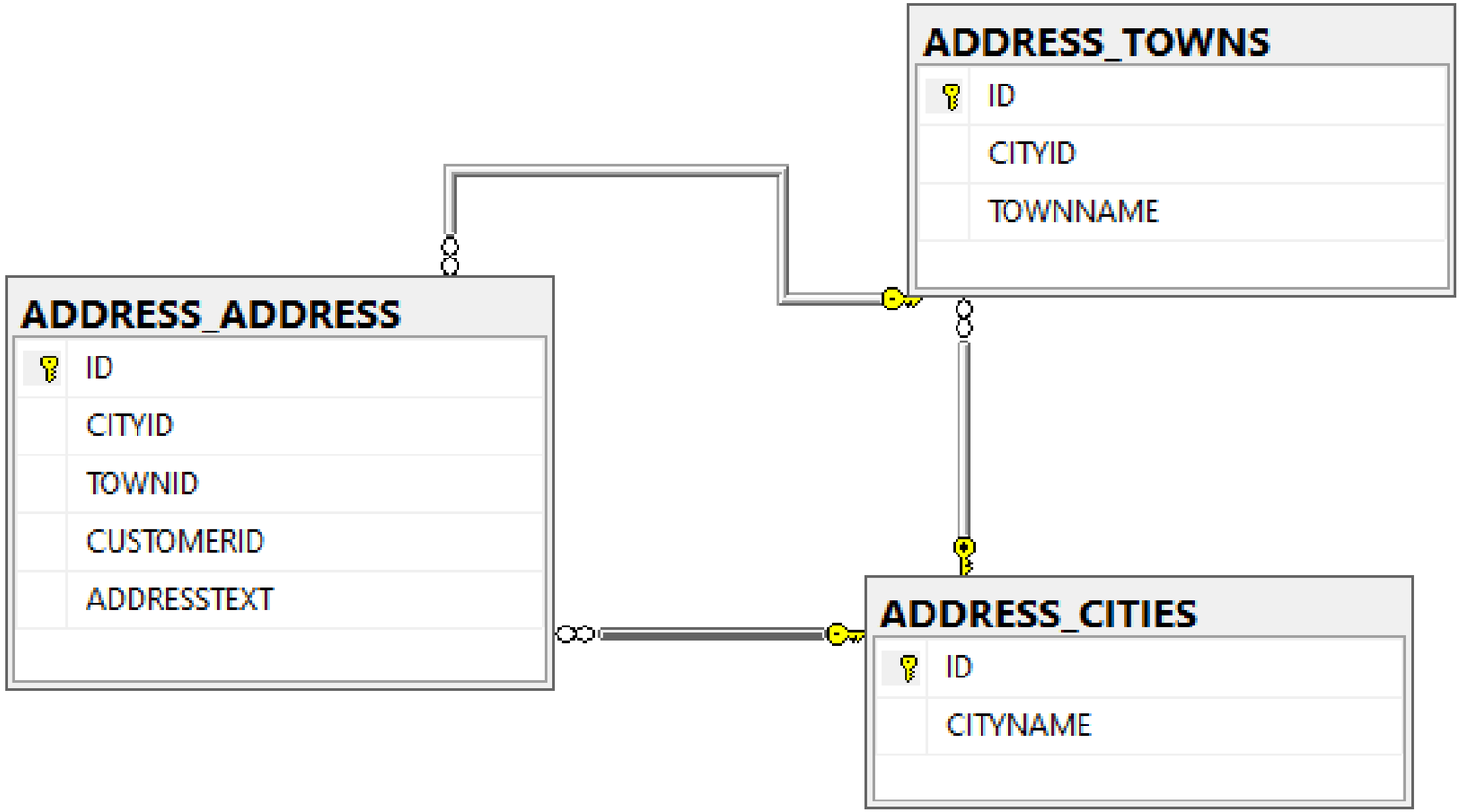
TABLES

- PRODUCT_PRODUCTCATEGORIES
- PRODUCT_PRODUCTSUBCATEGORIES
- PRODUCT_PRODUCT
- SALES_EMPLOYEEJOBCATEGORIES
- SALES_EMPLOYEES
- SALES_SALESDETAILS
- SALES_COMPLAINTS
- SALES_COMMENTS
- COMPANY_CAMPAIGN
- SALES_CUSTOMERS
- ADDRESS_CITIES
- ADDRESS_TOWNS
- SALES_POINTS
- COMPANY_DEALERS
- COMPANY_STOCKLOG
- SALES_SALES
- COMPANY_AUTHORIZEDSERVICEDETAILS
- ADDRESS_ADDRESS
- COMPANY_AUTHORIZEDSERVICE
- COMPANY_PRODUCTDEALER

```sql
CREATE TABLE PRODUCT_PRODUCTCATEGORIES(
ID INT IDENTITY(1,1) PRIMARY KEY,
CATEGORYNAME NVARCHAR(100)
)

CREATE TABLE PRODUCT_PRODUCTSUBCATEGORIES(
ID      INT IDENTITY(1,1) PRIMARY KEY,
CATEGORYID      INT,
SUBCATEGORYNAME NVARCHAR(100),

CONSTRAINT FK_PRODUCTSUBCATEGORIES_CATEGORY FOREIGN KEY(CATEGORYID) REFERENCES PRODUCT_PRODUCTCATEGORIES(ID)
)

CREATE TABLE PRODUCT_PRODUCT(
ID INT IDENTITY(1,1) PRIMARY KEY,
CATEGORYID INT,
PRODUCTCODE     CHAR(11),
PRODUCTNAME     NVARCHAR(100),
COLOR NVARCHAR(50),
ENERGYLEVEL     VARCHAR(50),
POWER_ INT,
DESCRIPTION_ NVARCHAR(200),
UNITPRICE DECIMAL,

CONSTRAINT FK_PRODUCT_CATEGORY FOREIGN KEY(CATEGORYID) REFERENCES PRODUCT_PRODUCTCATEGORIES(ID)
)
```


PRODUCT_PRODUCTCATEGORIES
ID
CATEGORYNAME
PRODUCT_PRODUCTSUBCATEGORIES
ID
CATEGORYID
SUBCATEGORYNAME
PRODUCT_PRODUCT
ID
CATEGORYID
PRODUCTCODE
PRODUCTNAME
COLOR
ENERGYLEVEL
POWER_
DESCRIPTION_
UNITPRICE

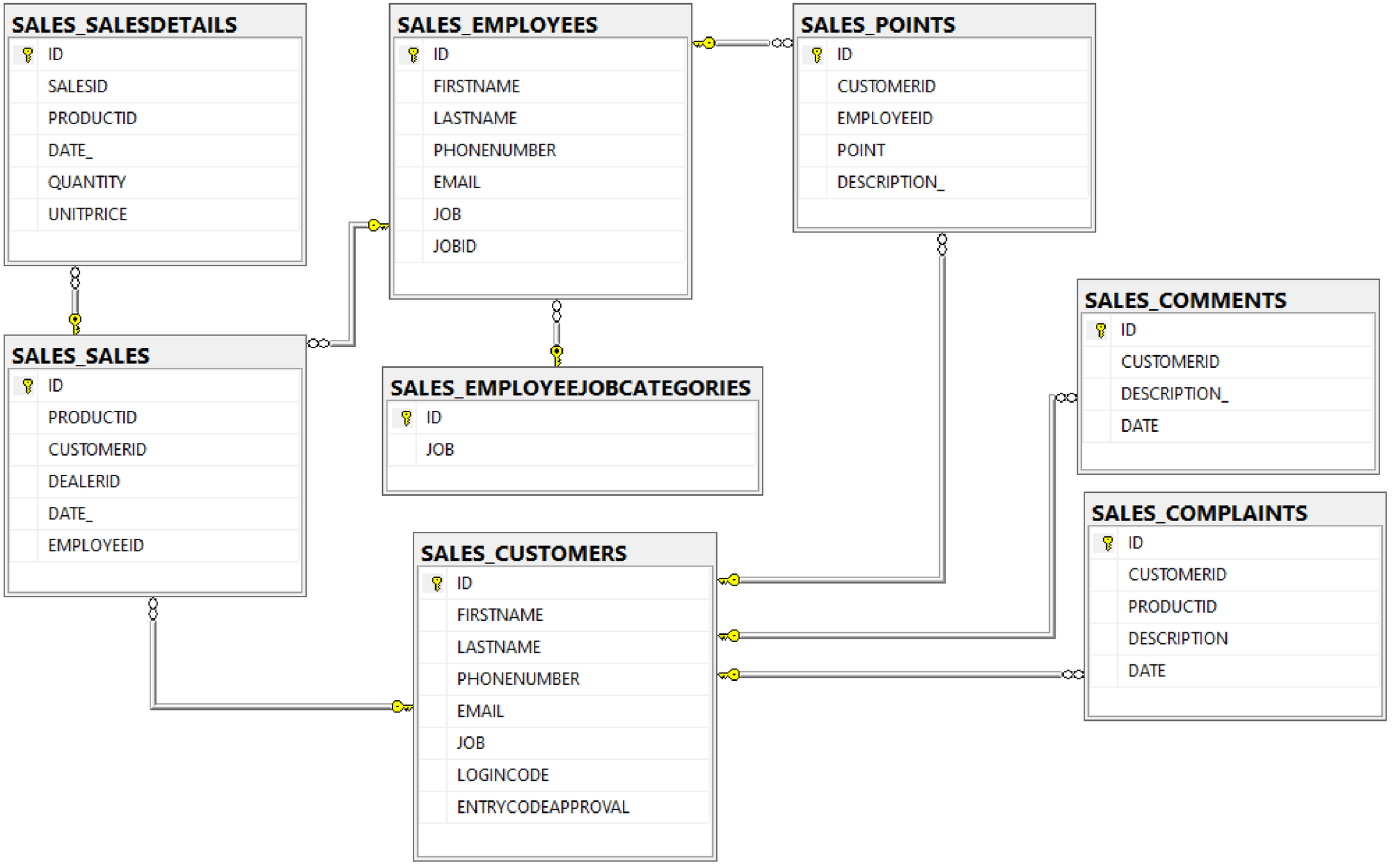
```sql
CREATE TABLE ADDRESS_CITIES(
ID TINYINT IDENTITY(1,1) PRIMARY KEY,
CITYNAME NVARCHAR(30)
)


CREATE TABLE ADDRESS_TOWNS(
ID      SMALLINT IDENTITY(1,1) PRIMARY KEY,
CITYID  TINYINT,
TOWNNAME NVARCHAR(50),

CONSTRAINT FK_TOWNS_CITY FOREIGN KEY(CITYID) REFERENCES ADDRESS_CITIES(ID)
)

CREATE TABLE ADDRESS_ADDRESS(
ID      INT IDENTITY(1,1) PRIMARY KEY,
CITYID  TINYINT,
TOWNID  SMALLINT,
CUSTOMERID      INT,
ADDRESSTEXT NVARCHAR(100),

CONSTRAINT FK_ADRESSS_CITY FOREIGN KEY(CITYID) REFERENCES ADDRESS_CITIES(ID),
CONSTRAINT FK_ADRESSS_TOWN FOREIGN KEY(TOWNID) REFERENCES ADDRESS_TOWNS(ID),
CONSTRAINT FK_ADDRESS_CUSTOMER FOREIGN KEY(CUSTOMERID) REFERENCES SALES_CUSTOMERS(ID)
)
```


ADDRESS_TOWNS
ID
CITYID
TOWNNAME
ADDRESS_ADDRESS
ID
CITYID
TOWNID
CUSTOMERID
ADDRESSTEXT
ADDRESS_CITIES
ID
CITYNAME

```sql
CREATE TABLE SALES_EMPLOYEEJOBCATEGORIES(
ID INT IDENTITY(1,1) PRIMARY KEY,
JOB NVARCHAR(50)
)

CREATE TABLE SALES_EMPLOYEES(
ID      INT IDENTITY(1,1) PRIMARY KEY,
JOBID INT,
FIRSTNAME NVARCHAR(50),
LASTNAME NVARCHAR(50),
PHONENUMBER     CHAR(11),
EMAIL VARCHAR(65) UNIQUE,
JOB NVARCHAR(50),

CONTRAINT FK_EMPLOYEE_JOB FOREIGN KEY(JOBID) REFERENCES SALES_EMPLOYEEJOBCATEGORIES(ID)
)

CREATE TABLE SALES_CUSTOMERS(
ID      INT IDENTITY(1,1) PRIMARY KEY,
FIRSTNAME       NVARCHAR(50),
LASTNAME NVARCHAR(50),
PHONENUMBER     CHAR(11),
EMAIL VARCHAR(65) UNIQUE,
JOB     NVARCHAR(50),
LOGINCODE INT,
ENTRYCODEAPPROVAL BIT
)
```

```sql
CREATE TABLE SALES_SALES(
ID      INT IDENTITY(1,1) PRIMARY KEY,
PRODUCTID INT,
CUSTOMERID      INT,
DEALERID INT,
DATE_   DATETIME,
EMPLOYEEID INT,

CONSTRAINT FK_SALES_PRODUCT FOREIGN KEY(PRODUCTID) REFERENCES PRODUCT_PRODUCT(ID),
CONSTRAINT FK_SALES_CUSTOMER FOREIGN KEY(CUSTOMERID) REFERENCES SALES_CUSTOMERS(ID),
CONSTRAINT FK_SALES_DEALER FOREIGN KEY(DEALERID) REFERENCES COMPANY_DEALERS(ID),
CONSTRAINT FK_SALES_EMPLOYEE FOREIGN KEY(EMPLOYEEID) REFERENCES SALES_EMPLOYEES(ID)
)

CREATE TABLE SALES_SALESDETAILS(
ID      INT IDENTITY(1,1) PRIMARY KEY,
SALESID INT,
PRODUCTID       INT,
DATE_   DATETIME,
QUANTITY INT,
UNITPRICE       DECIMAL,

CONSTRAINT FK_SALESDETAILS_SALES FOREIGN KEY(SALESID) REFERENCES SALES_SALES(ID),
CONSTRAINT FK_SALESDETAILS_PRODUCT FOREIGN KEY(PRODUCTID) REFERENCES PRODUCT_PRODUCT(ID)
)
```

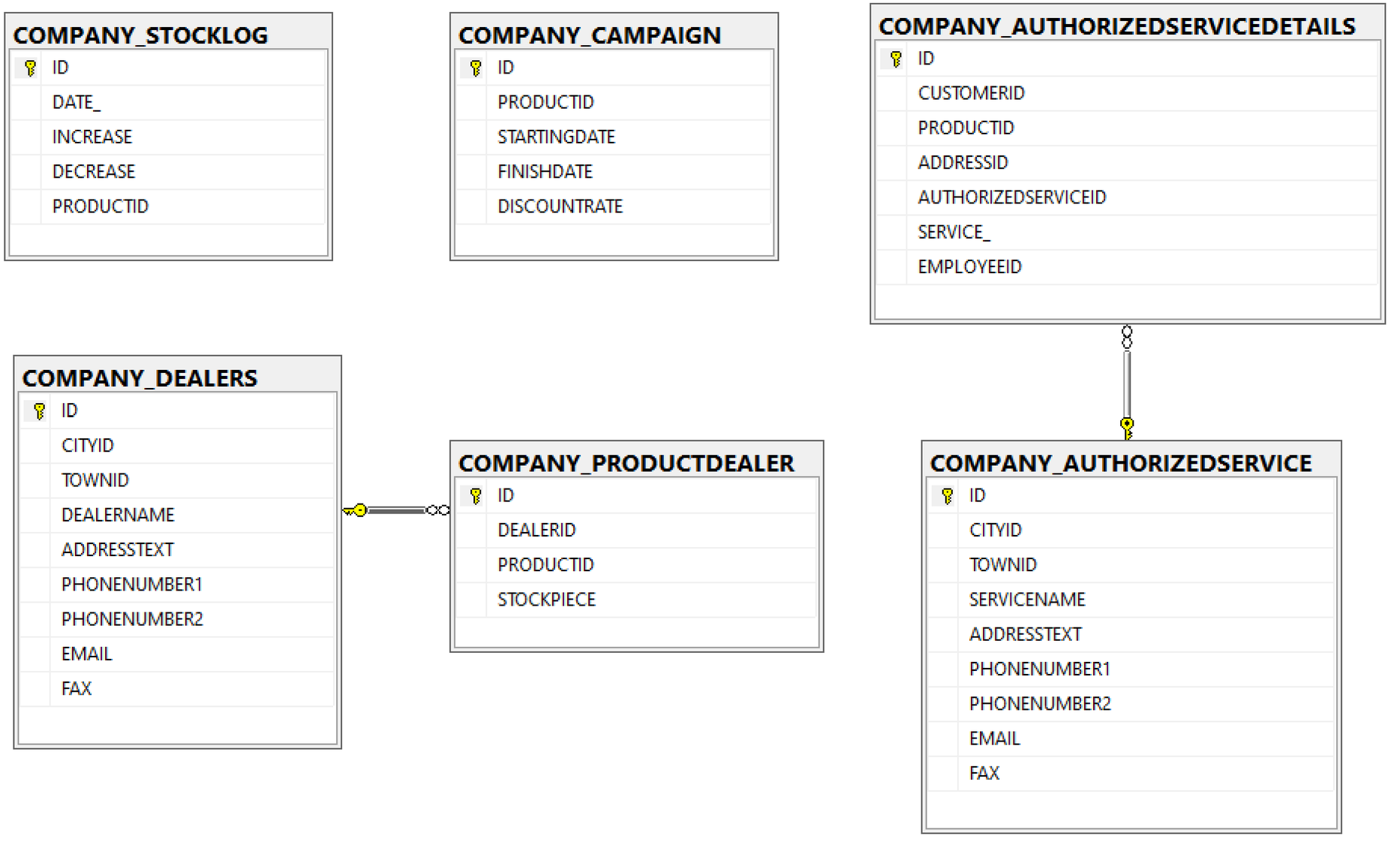
```
CREATE TABLE SALES_COMPLAINTS(
ID      INT IDENTITY(1,1),
CUSTOMERID INT,
PRODUCTID       INT,
DESCRIPTION NVARCHAR(500),
DATE DATETIME DEFAULT GETDATE(),

CONSTRAINT FK_COMPLAINTS_CUSTOMER FOREIGN KEY(CUSTOMERID) REFERENCES SALES_CUSTOMERS(ID),
CONSTRAINT FK_COMPLAINTS_PRODUCT FOREIGN KEY(PRODUCTID) REFERENCES PRODUCT_PRODUCT(ID)
)


CREATE TABLE SALES_COMMENTS(
ID      INT IDENTITY(1,1),
CUSTOMERID      INT,
DESCRIPTION_ NVARCHAR(500),
DATE DATETIME DEFAULT GETDATE(),

CONSTRAINT FK_COMMENTS_CUSTOMER FOREIGN KEY(CUSTOMERID) REFERENCES SALES_CUSTOMERS(ID)
)

CREATE TABLE SALES_POINTS(
ID      INT IDENTITY(1,1),
CUSTOMERID INT,
EMPLOYEEID      INT,
POINT INT NOT NULL,
DESCRIPTION_ NVARCHAR(100),

CONSTRAINT FK_COMMENTS_CUSTOMER FOREIGN KEY(CUSTOMERID) REFERENCES SALES_CUSTOMERS(ID),
CONSTRAINT FK_COMMENTS_EMPLOYEE FOREIGN KEY(EMPLOYEEID) REFERENCES SALES_EMPLOYEES(ID)
)
```


SALES_SALESDETAILS
ID
SALESID
PRODUCTID
DATE_
QUANTITY
UNITPRICE
SALES_EMPLOYEES
ID
FIRSTNAME
LASTNAME
PHONENUMBER
EMAIL
JOB
JOBID
SALES_POINTS
ID
CUSTOMERID
EMPLOYEEID
POINT
DESCRIPTION_
SALES_COMMENTS
ID
CUSTOMERID
DESCRIPTION_
DATE
SALES_SALES
ID
PRODUCTID
CUSTOMERID
DEALERID
DATE_
EMPLOYEEID
SALES_EMPLOYEEJOBCATEGORIES
ID
JOB
SALES_COMPLAINTS
ID
CUSTOMERID
PRODUCTID
DESCRIPTION
DATE
SALES_CUSTOMERS
ID
FIRSTNAME
LASTNAME
PHONENUMBER
EMAIL
JOB
LOGINCODE
ENTRYCODEAPPROVAL

```
CREATE TABLE COMPANY_DEALERS(
ID      INT IDENTITY(1,1) PRIMARY KEY,
CITYID  TINYINT,
TOWNID  SMALLINT,
DEALERNAME      NVARCHAR(100),
ADDRESSTEXT     NVARCHAR(100),
PHONENUMBER1    CHAR(11),
PHONENUMBER2    CHAR(11),
EMAIL VARCHAR(65),
FAX     VARCHAR(65),

CONSTRAINT FK_DEALER_CITY FOREIGN KEY(CITYID) REFERENCES ADDRESS_CITIES(ID),
CONSTRAINT FK_DEALER_TOWN FOREIGN KEY(TOWNID) REFERENCES ADDRESS_TOWNS(ID),
)


CREATE TABLE COMPANY_STOCKLOG(
ID INT IDENTITY(1,1) PRIMARY KEY,
PRODUCTID INT NOT NULL,
DATE_ DATETIME NOT NULL DEFAULT GETDATE(),
INCREASE INT,
DECREASE INT,

CONSTRAINT FK_STOCKLOG_PRODUCT FOREIGN KEY(PRODUCTID) REFERENCES PRODUCT_PRODUCT(ID)
)
```

```
CREATE TABLE COMPANY_PRODUCTDEALER(
ID      INT   IDENTITY(1,1) PRIMARY KEY,
DEALERID        INT,
PRODUCTID       INT,
STOCKPIECE      INT,

CONSTRAINT FK_PRODUCTDEALER_DEALER FOREIGN KEY(DEALERID) REFERENCES COMPANY_DEALERS(ID),
CONSTRAINT FK_PRODUCTDEALER_PRODUCT FOREIGN KEY(PRODUCTID) REFERENCES PRODUCT_PRODUCT(ID)
)


CREATE TABLE COMPANY_AUTHORIZEDSERVICE(
ID      INT IDENTITY(1,1) PRIMARY KEY,
CITYID  TINYINT,
TOWNID  SMALLINT,
SERVICENAME     NVARCHAR(100),
ADDRESSTEXT     NVARCHAR(100),
HONENUMBER1     CHAR(11),
HONENUMBER2     CHAR(11),
MAIL    VARCHAR(65),
AX      VARCHAR(65),

ONSTRAINT FK_AUTHORIZEDSERVICE_CITY FOREIGN KEY(CITYID) REFERENCES ADDRESS_CITIES(ID),
ONSTRAINT FK_AUTHORIZEDSERVICE_TOWN FOREIGN KEY(TOWNID) REFERENCES ADDRESS_TOWNS(ID),
```

```
CREATE TABLE COMPANY_AUTHORIZEDSERVICEDETAILS(
ID      INT IDENTITY(1,1) PRIMARY KEY,
CUSTOMERID      INT,
PRODUCTID INT,
ADDRESSID INT,
AUTHORIZEDSERVICEID INT,
SERVICE_        NVARCHAR(50),
EMPLOYEEID      INT,

CONSTRAINT FK_AUTHORIZEDSERVICEDETAILS_CUSTOMER FOREIGN KEY(CUSTOMERID) REFERENCES SALES_CUSTOMERS(ID),
CONSTRAINT FK_AUTHORIZEDSERVICEDETAILS_PRODUCT FOREIGN KEY(PRODUCTID) REFERENCES PRODUCT_PRODUCT(ID),
CONSTRAINT FK_AUTHORIZEDSERVICEDETAILS_ADDRESS FOREIGN KEY(ADDRESSID) REFERENCES ADDRESS_ADDRESS(ID),
CONSTRAINT FK_AUTHORIZEDSERVICEDETAILS_EMPLOYEE FOREIGN KEY(EMPLOYEEID) REFERENCES SALES_EMPLOYEES(ID),
CONSTRAINT FK_AUTHORIZEDSERVICEDETAILS_EMPLOYEE FOREIGN KEY(AUTHORIZEDSERVICEID) REFERENCES COMPANY_AUTHORIZEDSERVICE(ID)
)


CREATE TABLE COMPANY_CAMPAIGN(
ID      INT IDENTITY(1,1) PRIMARY KEY,
PRODUCTID       INT,
STARTINGDATE DATE,
FINISHDATE DATE,
DISCOUNTRATE FLOAT,

CONSTRAINT FK_CAMPAIGN_PRODUCT FOREIGN KEY(PRODUCTID) REFERENCES PRODUCT_PRODUCT(ID)
)
```


COMPANY_STOCKLOG
ID
DATE_
INCREASE
DECREASE
PRODUCTID
COMPANY_CAMPAIGN
ID
PRODUCTID
STARTINGDATE
FINISHDATE
DISCOUNTRATE
COMPANY_AUTHORIZEDSERVICEDETAILS
ID
CUSTOMERID
PRODUCTID
ADDRESSID
AUTHORIZEDSERVICEID
SERVICE_
EMPLOYEEID
COMPANY_DEALERS
ID
CITYID
TOWNID
DEALERNAME
ADDRESSTEXT
PHONENUMBER1
PHONENUMBER2
EMAIL
FAX
COMPANY_PRODUCTDEALER
ID
DEALERID
PRODUCTID
STOCKPIECE
COMPANY_AUTHORIZEDSERVICE
ID
CITYID
TOWNID
SERVICENAME
ADDRESSTEXT
PHONENUMBER1
PHONENUMBER2
EMAIL
FAX

SQLQuery1.sql - 1...eKUCUKDOGRU (54))*

```
INSERT INTO [dbo].[SALES_CUSTOMERS]
           ([FIRSTNAME]
           ,[LASTNAME]
           ,[PHONENUMBER]
           ,[EMAIL]
           ,[JOB]
           ,[LOGINCODE]
           ,[ENTRYCODEAPPROVAL])
     VALUES
           ('Merve'
           ,'KÜÇÜKDOĞRU'
           ,NULL
           ,'kucukdogrumerve@gmail.com'
           ,'Bilgisayar Mühendisi'
           ,NULL
           ,NULL)
```

Sistem Hesabı Aktifleştirme Kodu Gelen Kutusu ×

Merve KÜÇÜKDOĞRU <kucukdogrumerve@gmail.com> 23:55 (0 dakika önce)

Alıcı: ben

Sayın **Merve KÜÇÜKDOĞRU;**

Sisteme kayıt için gerekli tek kullanımlık şifreniz: **37599**

Tek kullanımlık şifrenizi kimseyle paylaşmayınız!!!

BOSCH

Yaşam için teknoloji

Procedures Examples

```
EXEC [dbo].[SALES]
        @PRODUCTID = 5,
        @CUSTOMERID = 1004,
        @DEALERID = 1,
        @PIECE = 5,
        @EMPLOYEEID = 55
```

Sipariş Bilgilendirme

MK Merve KÜÇÜKDOĞRU <kucukdogrumerve@gmail.com>

1:04 AM

To: m42erve@hotmail.com

Sayın **MERVE KÜÇÜKDOĞRU;**

Siparişiniz alınmıştır.

En kısa sürede kargoya verilecektir.

Aldığınız ürünler aşağıdaki gibidir:

Ürün Adı	Adet	Birim Fiyat	Toplam Fiyat
Hand blender	5	7472	5 x 7472 = 37360 TL

BOSCH

Yaşam için teknoloji

```sql
SELECT TOP 1 * FROM SALES_SALES S
ORDER BY S.ID DESC

GO

SELECT TOP 1 * FROM SALES_SALESDETAILS SD
ORDER BY SD.ID DESC

GO

SELECT TOP 1 * FROM COMPANY_STOCKLOG SL
ORDER BY SL.ID DESC
```

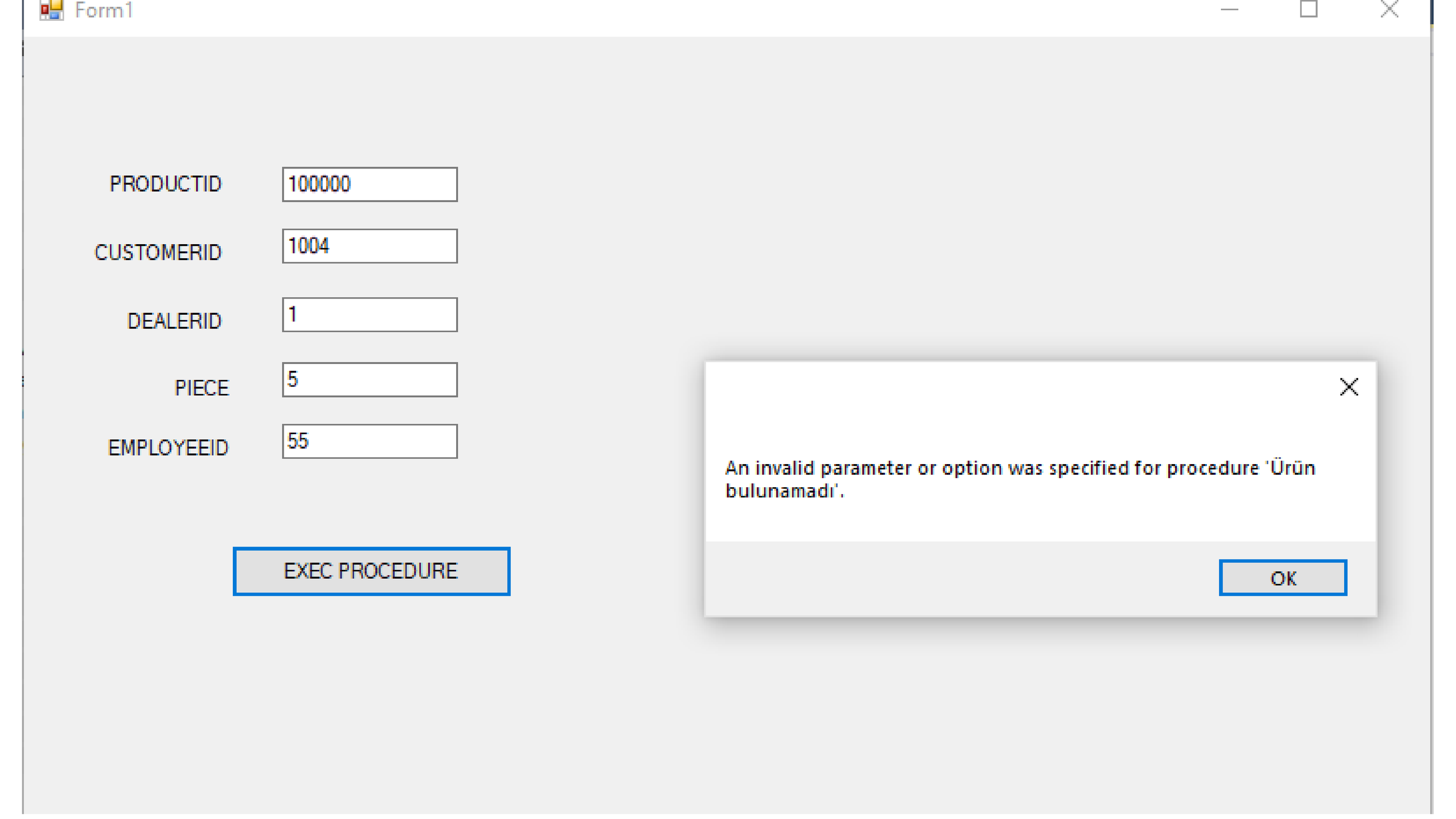
%

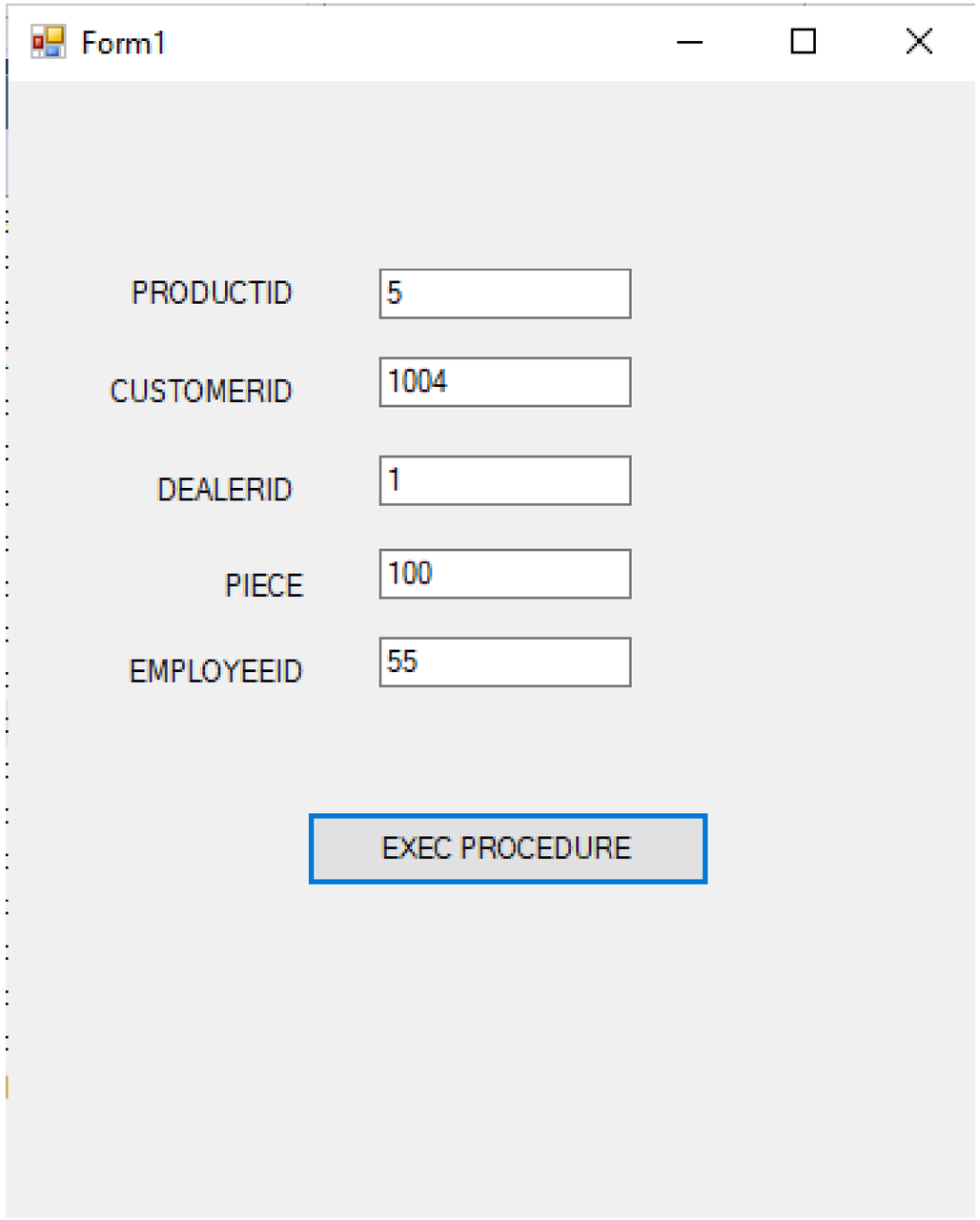
Results | Messages

ID	PRODUCTID	CUSTOMERID	DEALERID	DATE_	EMPLOYEEID
1118	5	1004	1	2021-01-24 01:04:45.713	55

ID	SALESID	PRODUCTID	DATE_	QUANTITY	UNITPRICE
1148	1118	5	2021-01-24 01:04:45.760	5	7472

ID	DATE_	INCREASE	DECREASE	PRODUCTID
68	2021-01-24 01:04:45.760	NULL	5	5


Form1
PRODUCTID
100000
CUSTOMERID
1004
DEALERID
1
PIECE
5
EMPLOYEEID
55
EXEC PROCEDURE
An invalid parameter or option was specified for procedure 'Ürün bulunamadı'.
OK



Stok Bilgilendirme

MK

Merve KÜÇÜKDOĞRU <kucukdogrumerve@gmail.com>

1:25 AM

To: m42erve@hotmail.com

Sayın **MERVE KÜÇÜKDOĞRU;**

Almak istediğiniz ürün stokta yoktur

Ürün geldiğinde size haber verilecektir.!!!

BOSCH

Yaşam için teknoloji

```sql
--KAMPANYADAKİ ÜRÜNLERİN SATIŞ MİKTARI ve stokta kalan ürün adeti

CREATE PROCEDURE SP_CAMPAING_SALES_STOCK_QUANTITY
@PRODUCTCODE NVARCHAR (20)
AS
SELECT PRODUCTCODE,PRODUCTNAME,
SUM(QUANTITY) AS TOTAL_QUANTITY,
SUM(STOCKPIECE) AS TOTALSTOCK
FROM COMPANY_CAMPAIGN C
LEFT JOIN COMPANY_PRODUCTDEALER PD ON PD.PRODUCTID=C.PRODUCTID
INNER JOIN PRODUCT_PRODUCT P ON P.ID=PD.PRODUCTID
INNER JOIN SALES_SALESDETAILS SD ON SD.PRODUCTID=C.PRODUCTID
WHERE PRODUCTCODE= @PRODUCTCODE
GROUP BY PRODUCTCODE,PRODUCTNAME
```

```sql
EXEC SP_CAMPAING_SALES_STOCK_QUANTITY @PRODUCTCODE='KGN76AIF0N'
```

100 %

Results | Messages

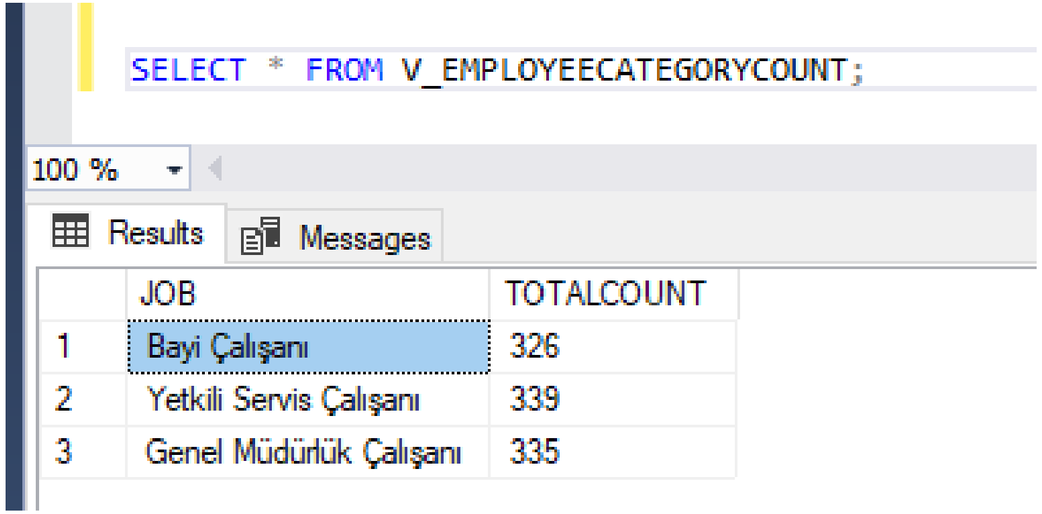
	PRODUCTCODE	PRODUCTNAME	TOTAL_QUANTITY	TOTALSTOCK
1	KGN76AIF0N	Alttan Donduruculu Buzdolabı	200	4500

Trigger Example

```
ALTER TRIGGER [dbo].[TRG_ADDSTOKLOG]
ON [dbo].[COMPANY_PRODUCTDEALER]
AFTER UPDATE
AS
BEGIN

DECLARE @PRODUCTID INT
DECLARE @INCREASE INT = 20
DECLARE @DATE DATETIME=GETDATE()

SELECT @PRODUCTID=PRODUCTID FROM COMPANY_PRODUCTDEALER WHERE STOCKPIECE<2 AND PRODUCTID=@PRODUCTID

IF @PRODUCTID IS NOT NULL
    BEGIN
    INSERT INTO [dbo].[COMPANY_STOCKLOG]
               ([DATE_]
               ,[INCREASE]
               ,[PRODUCTID])
         VALUES
               (@DATE
               ,@INCREASE
               ,@PRODUCTID
               )
    END
END
```

View Examples

```
CREATE VIEW V_EMPLOYEECATEGORYCOUNT
AS
SELECT JOB, COUNT(*) TOTALCOUNT FROM SALES_EMPLOYEES
        GROUP BY JOB

SELECT * FROM V_EMPLOYEECATEGORYCOUNT;
```



	JOB	TOTALCOUNT
1	Bayi Çalışanı	326
2	Yetkili Servis Çalışanı	339
3	Genel Müdürlük Çalışanı	335

```sql
--ÇALIŞANLAR TOPLAM NE KADAR SATIŞ YAPMIŞ

CREATE VIEW V_EMPLOYEESTOTALSALES
AS
SELECT S.EMPLOYEEID, E.FIRSTNAME + ' ' + E.LASTNAME EMPLOYEE, COUNT(*) TOTALSALES, SUM(UNITPRICE) SALESTOTALPRICE FROM SALES_SALES S
        INNER JOIN SALES_EMPLOYEES E ON E.ID=S.EMPLOYEEID
        INNER JOIN SALES_SALESDETAILS SD  ON SD.SALESID=S.ID
        GROUP BY S.EMPLOYEEID, E.FIRSTNAME, E.LASTNAME
```

```sql
SELECT * FROM V_EMPLOYEESTOTALSALES
ORDER BY 4 DESC;
```

100 %

Results | Messages

	EMPLOYEEID	EMPLOYEE	TOTALSALES	SALESTOTALPRICE
1	55	SÜHEYLA JALE	68	351876
2	175	BÜNYAMİN YILDIZ	4	50074
3	652	SEVDA İLHAN	3	43186
4	54	İSMAİL EVREN FATİH	4	42830
5	840	HAŞİM ONUR UMUT	3	28990
6	636	SERKAN İLKNUR	3	28346
7	87	FERDA SENEM	3	27018
8	49	ELİF İLKAY	3	26798
9	937	ELİF MURAT	3	26394
10	8	MEHMET SULTAN	5	26280

```sql
--Hangi ilde kaç tane bayi var?

CREATE VIEW V_TOTALDEALER
AS
SELECT D.CITYID, C.CITYNAME, COUNT(D.CITYID) TOTAL_DEALER FROM COMPANY_DEALERS D
        INNER JOIN ADDRESS_CITIES C ON C.ID=D.CITYID
        GROUP BY D.CITYID, C.CITYNAME;
```

```sql
SELECT * FROM V_TOTALDEALER;
```

100 %

Results | Messages

	CITYID	CITYNAME	TOTAL_DEALER
1	3	AFYONKARAHİSAR	1
2	6	ANKARA	25
3	11	BİLECİK	2
4	14	BOLU	2
5	16	BURSA	1
6	26	ESKİŞEHİR	9
7	32	ISPARTA	1
8	34	İSTANBUL	24
9	42	KONYA	22
10	43	KÜTAHYA	5
11	54	SAKARYA	5
12	56	SİİRT	1
13	70	KARAMAN	2

```sql
--Hangi kategoride kaç ürün var?

CREATE VIEW V_TOTALPRODUCT
AS
SELECT PC.CATEGORYNAME, COUNT(*) TOTALPRODUCT FROM PRODUCT_PRODUCT P
        INNER JOIN PRODUCT_PRODUCTCATEGORIES PC ON PC.ID=P.CATEGORYID
        GROUP BY PC.CATEGORYNAME;
```

```sql
SELECT * FROM V_TOTALPRODUCT;
```

100 %

Results | Messages

	CATEGORYNAME	TOTALPRODUCT
1	Bulaşık Makineleri	71
2	Buzdolapları & Derin Dondurucular	138
3	Çamaşır & Kurutma Makineleri	56
4	Elektrikli Süpürgeler	21
5	Kahvaltı Hazırlama Grubu	30
6	Kahve Makineleri	26
7	Klimalar & Ev Konforu	47
8	Mutfak Aletleri	83
9	Mutfak Makineleri	13
10	Pişirme Grubu	205
11	Ütüler	14
12	Yeni Ürünler	1

```sql
--ÜRÜNE GÖRE BAKIM SAYILARINI GETİREN VİEW

CREATE VIEW PRODUCT_MAINTENANCE
AS
SELECT PRODUCTNAME, PRODUCTCODE,
COUNT(SERVICE_) AS NUMBER_OF_MAINTENANCE
FROM PRODUCT_PRODUCT P
INNER JOIN COMPANY_AUTHORIZEDSERVICEDETAILS ASD ON ASD.PRODUCTID=P.ID
WHERE P.PRODUCTCODE IS NOT NULL
GROUP BY PRODUCTNAME, PRODUCTCODE
```

```sql
SELECT * FROM PRODUCT_MAINTENANCE
ORDER BY 3 DESC
```

Results | Messages

PRODUCTNAME	PRODUCTCODE	NUMBER_OF_MAINTENANCE
Alttan Donduruculu Ankastre Buzdolabı	KIS87AF30N	2
Solo Buzdolabı	KSV36VI30N	2
Solo Buzdolabı	KSV36VW30N	2
Clip door	KSZ1CVH00	2
Set Üstü Gazlı Ocak	PBP0C2B80L	2
Set Üstü Gazlı Ocak	PBP0C2B80O	2
Set Üstü Gazlı Ocak	PBP0C5B80L	2
Set Üstü Gazlı Ocak	PBP0C5B80O	2
Gazlı Ocak	PBP6C2B80L	2
Set Üstü Gazlı Ocak	POP0C2O10O	2
Set Üstü Gazlı Ocak	POP0C6B10L	2
Set Üstü Gazlı Ocak	POP0C6B10O	2
Set Üstü Gazlı Ocak	POP0C6O10O	2

Functions Examples

```sql
--ÇALIŞAN NUMARASI GİRİLEN ÇALIŞANIN SATMIŞ OLDUĞU ÜRÜNLERİ VE SATIŞ TARİHİNİ GETİREN FONKSİYON
CREATE FUNCTION EMPLOYEE_SALES (@EMPLOYEEID INT)
RETURNS TABLE
AS
RETURN
SELECT E.ID,FIRSTNAME,LASTNAME,PRODUCTNAME,SD.DATE_
FROM SALES_EMPLOYEES E
INNER JOIN SALES_SALES S ON S.EMPLOYEEID=E.ID
INNER JOIN SALES_SALESDETAILS SD ON SD.PRODUCTID=S.PRODUCTID
INNER JOIN PRODUCT_PRODUCT P ON P.ID=SD.PRODUCTID
WHERE E.ID=@EMPLOYEEID
GROUP BY E.ID,FIRSTNAME,LASTNAME,PRODUCTNAME,SD.DATE_
```

```sql
SELECT * FROM dbo.EMPLOYEE_SALES(5)
```

Results | Messages

ID	FIRSTNAME	LASTNAME	PRODUCTNAME	DATE_
5	YAŞAR GÖKHAN	AHMET GÖKHAN	Amerikan Tipi Buzdolapları için Su Filtresi	2017-01-02 00:00:00.000
5	YAŞAR GÖKHAN	AHMET GÖKHAN	Amerikan Tipi Buzdolapları için Su Filtresi	2019-01-21 00:00:00.000
5	YAŞAR GÖKHAN	AHMET GÖKHAN	Fırınlar için Lamba	2017-09-03 00:00:00.000
5	YAŞAR GÖKHAN	AHMET GÖKHAN	Fırınlar için Lamba	2017-11-24 00:00:00.000
5	YAŞAR GÖKHAN	AHMET GÖKHAN	Solo Bulaşık Makinesi	2017-09-07 00:00:00.000
5	YAŞAR GÖKHAN	AHMET GÖKHAN	Solo Bulaşık Makinesi	2019-09-07 00:00:00.000

```sql
--BAYİLERİN 0 STOĞA SAHİP ÜRÜNLERİNİ GETİREN FONKSİYON

CREATE FUNCTION DEALER_0_STOCK_PIECE()
RETURNS TABLE
AS
RETURN
SELECT DEALERID,DEALERNAME, PRODUCTCODE,PRODUCTNAME, STOCKPIECE FROM COMPANY_PRODUCTDEALER PD
INNER JOIN PRODUCT_PRODUCT P ON P.ID=PD.PRODUCTID
INNER JOIN COMPANY_DEALERS D ON D.ID=PD.DEALERID
WHERE STOCKPIECE=0
```

```sql
SELECT * FROM DBO.DEALER_0_STOCK_PIECE()
```

100 %

Results | Messages

	DEALERID	DEALERNAME	PRODUCTCODE	PRODUCTNAME	STOCKPIECE
1	1	BURDA ALISVERIS MERKEZI VE INSAAT	PCP6A5B90	Gazlı Ocak	0

Index Examples

```
1   Create NONCLUSTERED INDEX X_INDEX_NAME ON SALES_CUSTOMERS(FIRSTNAME);
```

```
SELECT * FROM SALES_CUSTOMERS
WHERE FIRSTNAME='Merve'
```

100 %

Results | Messages

	ID	FIRSTNAME	LASTNAME	PHONENUMBER	EMAIL	JOB	LOGINCODE	ENTRYCODEAPPROVAL
1	1001	Merve	KÜÇÜKDOĞRU	22155412252	kucukdogrumerve4288@gmail.com	Bilgisayar Mühendisi	65165	1
2	1002	MERVE	KÜÇÜKDOĞRU	25545585555	m42erve@gmail.com	Bilgisayar Mühendisi	49905	1
3	1004	MERVE	KÜÇÜKDOĞRU	25513265225	m42erve@hotmail.com	Bilgisayar Mühendisi	31382	1
4	1011	Merve	KÜÇÜKDOĞRU	NULL	kucukdogrumerve@gmail.com	Bilgisayar Mühendisi	37599	1

Query executed successfully. | 192.168.1.37 (15.0 RTM) | sa (65) | STOCKTRACKINGSYSTEM | 00:00:00 | 4 rows

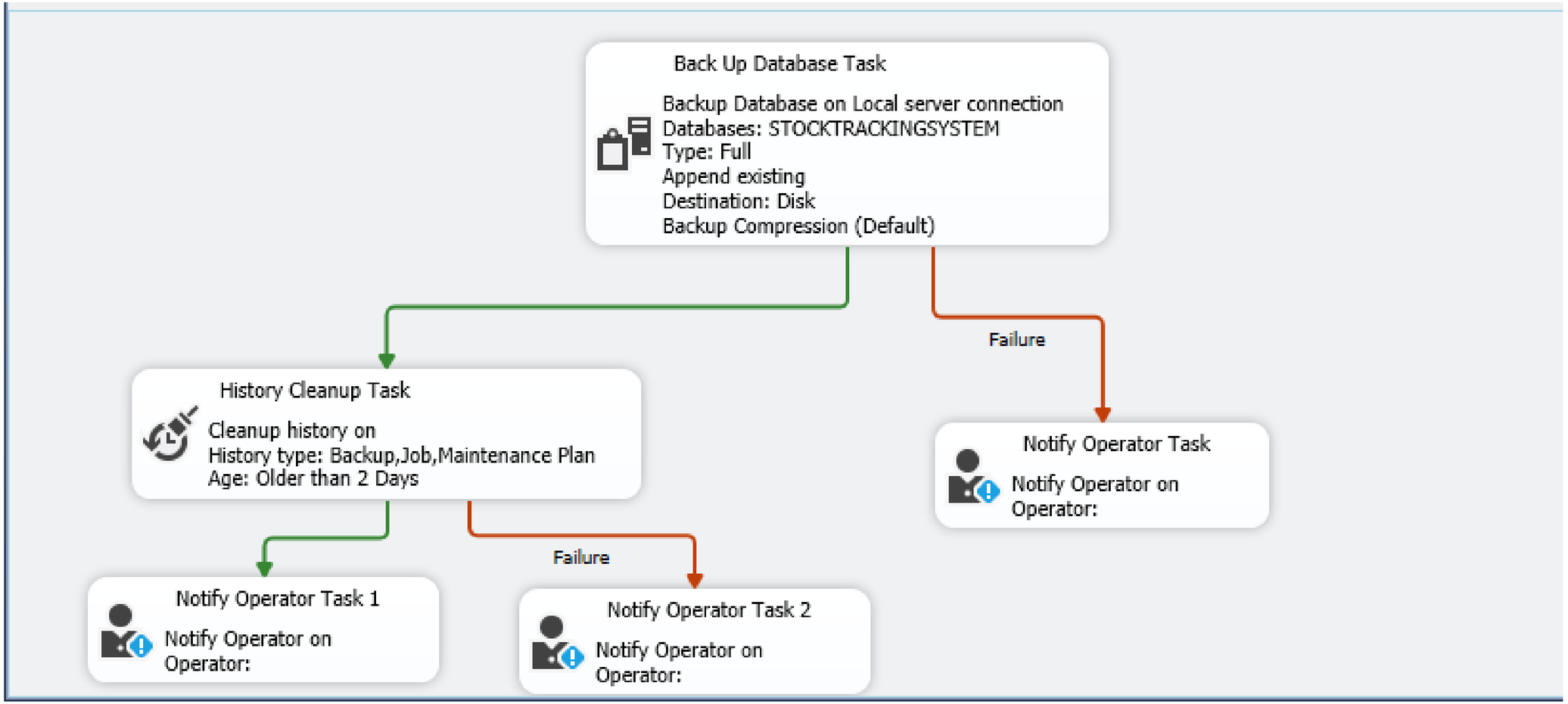
```
SELECT * FROM SALES_CUSTOMERS
WHERE FIRSTNAME='Merve'
```

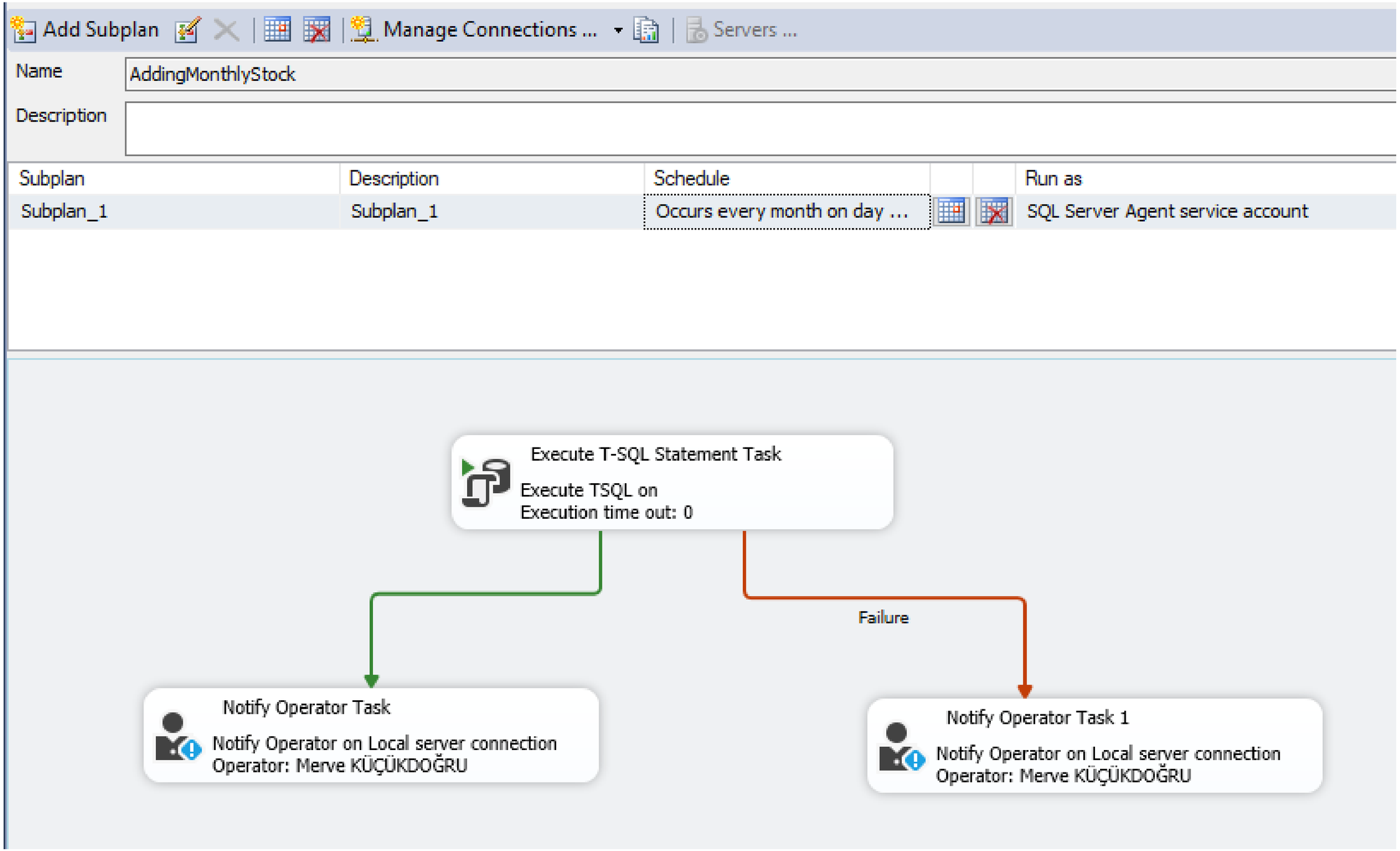
100 %

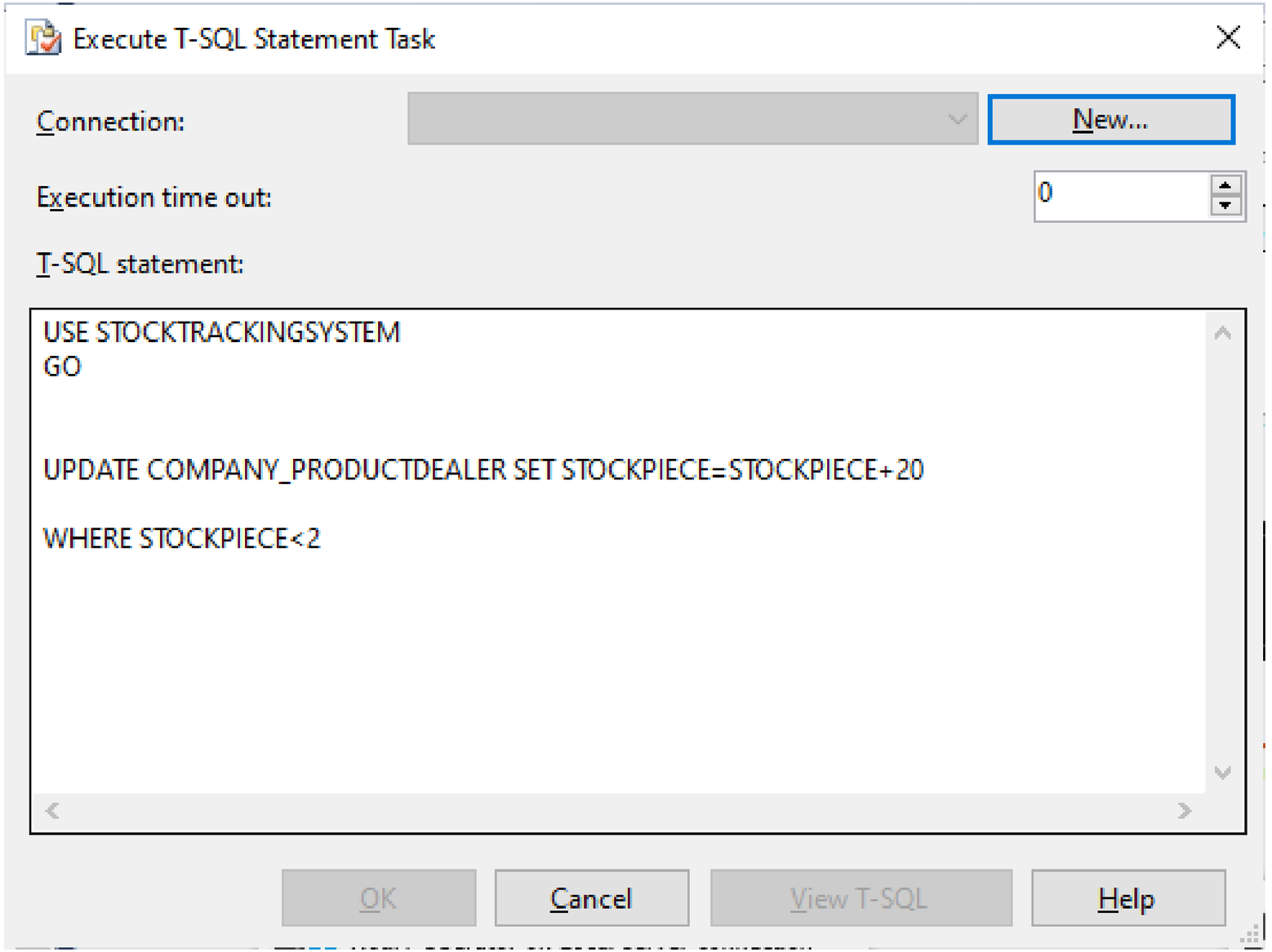
Results | Messages

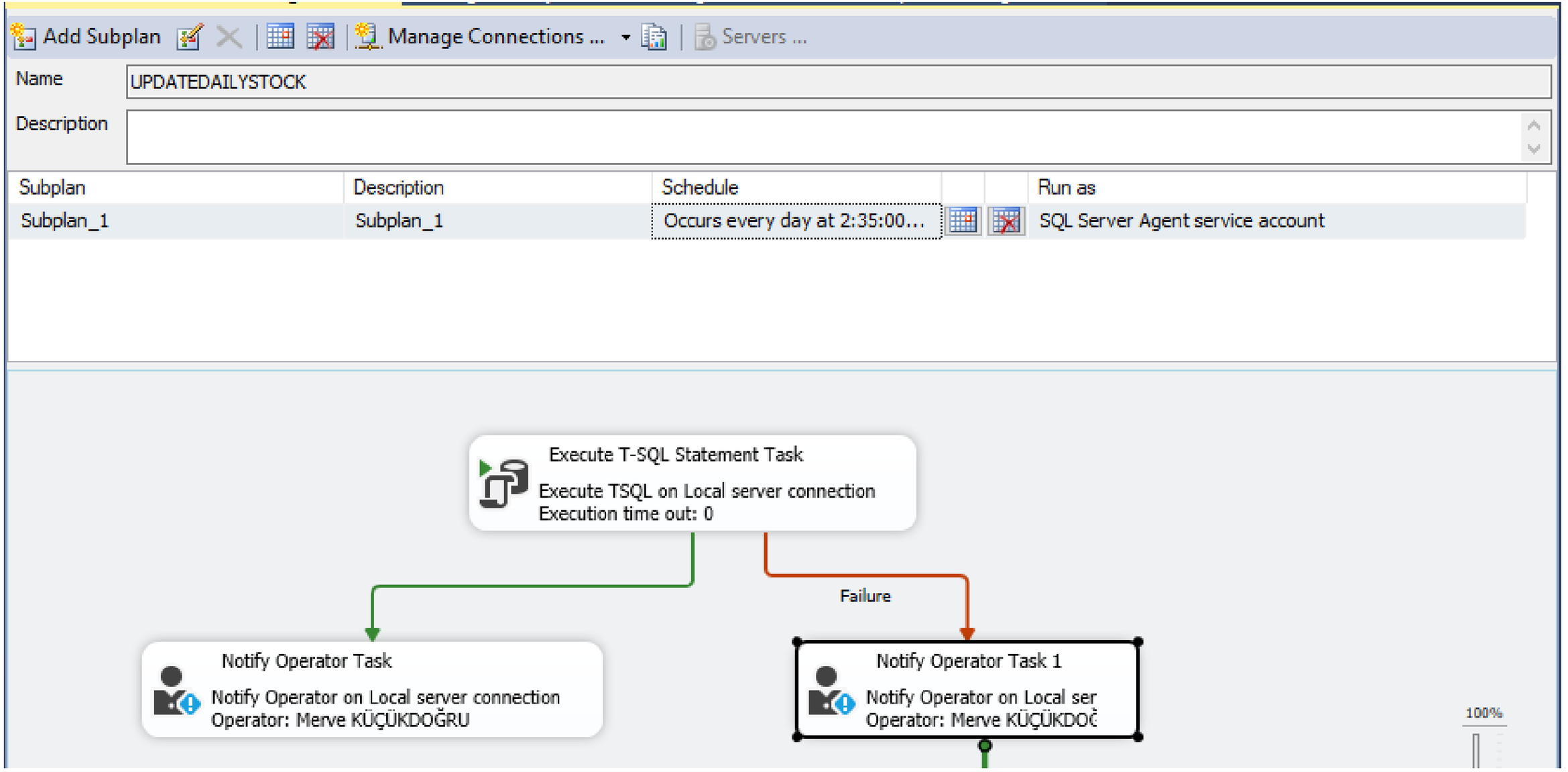
```
(4 rows affected)
Table 'SALES_CUSTOMERS'. Scan count 1, logical reads 10, physical reads 0, page server reads 0,
read-ahead reads 0, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0,
lob read-ahead reads 0, lob page server read-ahead reads 0.

Completion time: 2021-01-24T02:29:47.9695642+03:00
```

Maintenance Plans




Add Subplan
Manage Connections ...
Servers ...
Name
AddingMonthlyStock
Description
Subplan
Description
Schedule
Run as
Subplan_1
Subplan_1
Occurs every month on day ...
SQL Server Agent service account
Execute T-SQL Statement Task
Execute TSQL on
Execution time out: 0
Failure
Notify Operator Task
Notify Operator on Local server connection
Operator: Merve KÜÇÜKDOĞRU
Notify Operator Task 1
Notify Operator on Local server connection
Operator: Merve KÜÇÜKDOĞRU


Execute T-SQL Statement Task
Connection:
New...
Execution time out:
0
T-SQL statement:
USE STOCKTRACKINGSYSTEM
GO
UPDATE COMPANY_PRODUCTDEALER SET STOCKPIECE=STOCKPIECE+20
WHERE STOCKPIECE<2
OK
Cancel
View T-SQL
Help


Add Subplan
Manage Connections ...
Servers ...
Name
UPDATEDAILYSTOCK
Description
Subplan
Description
Schedule
Run as
Subplan_1
Subplan_1
Occurs every day at 2:35:00...
SQL Server Agent service account
Execute T-SQL Statement Task
Execute TSQL on Local server connection
Execution time out: 0
Failure
Notify Operator Task
Notify Operator on Local server connection
Operator: Merve KÜÇÜKDOĞRU
Notify Operator Task 1
Notify Operator on Local ser
Operator: Merve KÜÇÜKDOĞ
100%

The Team

Merve KÜÇÜKDOĞRU

TEAM LEADER
https://www.linkedin.com/in/merve-kucukdogru/

Fatmanur ACAR

https://www.linkedin.com/in/fatmanuracar/

Furkan AKTÜRK

https://www.linkedin.com/in/furkan66akturk67/