SATIFIYATI | ALISFIYATI | KAR

SELECT BIRIMFIYAT,

**(**SELECT TOP 1 BIRIMFIYAT FROM ALIMLAR WHERE MALZEMEKODU= SAT.MALZEMEKODU AND TARIH<= SAT.TARIH ORDER BY TARIH DESC**)** ALISFIYATI,

(BIRIMFIYAT-(SELECT TOP 1 BIRIMFIYAT FROM ALIMLAR WHERE MALZEMEKODU= SAT.MALZEMEKODU AND TARIH<= SAT.TARIH ORDER BY TARIH DESC)) AS KAR,

FROM SATISLAR SAT WHERE MALZEMEKODU= '42340'

====FONKSİYON TANIMLAMA====

CREATE FUNCTION DBO.ALISFIYATIBUL2(@MALZEMEKODU AS VARCHAR(30), @TARIH AS DATETIME)

RETURNS FLOAT

AS

BEGIN

DECLARE @SONUC AS FLOAT

SELECT TOP 1 @SONUC=BIRIMFIYAT FROM ALIMLAR WHERE MALZEMEKODU= @MALZEMEKODU AND TARIH<= @TARIH ORDER BY TARIH DESC

RETURN @SONUC

END

=====DINAMIK VIEW===

SELECT SATISFIYATI,ALISFIYATI, SATISFIYATI-ALISFIYATI AS KAR,\* FROM

(

SELECT BIRIMFIYAT SATISFIYATI, DBO.ALISFIYATIBUL(SAT.MALZEMEKODU,SAT.TARIH) ALISFIYATI, \*

FROM SATISLAR SAT WHERE MALZEMEKODU= '32382'

) DINAMIKVIEWDENEM

**===DINAMIKVIEW2===(URUNADI|TOPLAMSATIS|ORTALAMAFIYAT|TOPLAMKAR|ORTALAMAKAR|KARYUZDE)**

Herbir ğrğn için yukarıdaki kolonları oluşturacak dinamik view .

SELECT MALZEMEKODU,MALZEMEADI,

SUM(SATISFIYATI\*MIKTAR) AS TOPLAMSATIS,

SUM(SATISFIYATI\*MIKTAR)/SUM(MIKTAR) ORTALAMASATIS,

SUM(KAR\*MIKTAR) AS TOPLAMKAR,

SUM(KAR\*MIKTAR)/SUM(MIKTAR) AS ORTALAMAKAR

FROM

(

SELECT

SATISFIYATI-ALISFIYATI AS KAR, \* FROM

(

SELECT TOP 10000 BIRIMFIYAT SATISFIYATI, DBO.ALISFIYATIBUL(SAT.MALZEMEKODU,SAT.TARIH) ALISFIYATI, \*

FROM SATISLAR SAT -- WHERE MALZEMEKODU= '32382'

) DINAMIKVIEWDENEME

) DINAMIKVIEWDENEME2 GROUP BY MALZEMEKODU, MALZEMEADI

====== ÇEKTİĞİMİZ BİR SORGUYU AYRI BİR TABLOYA YAZAN SORGU ===

1. Ekleyeceğimiz bir tablomuz yoksa, yeni bir tablo oluşturup verileri raya yazan SQL

🡪 SELECT \* INTO TABLO1 FROM TABLOADI

1. Hali hazırda bir tablo varsa ona ekleyeceksek

🡪 INSERT INTO TABLO1 SELECT \* FROM TABLOADI

AMAÇ : Yukarıdaki binlerce satırlık veride yazdığımız fonksiyon binlerce kez çalışıyor hızımız düşüyor. Bundan dolayı, bir defalık bu tablaya çekme işlemini yapıyoruz. Sonra bütün işlemleri daha hızlı bir şekilde, bu tablo üzerinden hallediyoruz.

=====TRIGGER=====

1. Tabloya ekleme, silme, update işlemlerinde yapılması gereken şeyleri yaptırırız. Örneğin, Bir tabloda bir değişiklik olduğu zaman buişlemlerin kaydını tutmak istiyorsak, bu değişikliği kim değiştirdi, ne zaman kim tarafından, hangi uygulama tarafından yapıldı bilgilerini tutmak istiyorsak kullanırız.

ALTER TRIGGER TRG\_LG\_015\_ITEMS\_DELETE

ON LG\_015\_ITEMS

AFTER DELETE --after yerine INSTEAD OF DELETE yazsaydık tabloda silmeyecekti

AS

BEGIN

DECLARE @ACTIONTYPE AS VARCHAR(100)='DELETE'

DECLARE @DATE AS DATETIME

SET @DATE = GETDATE()

DECLARE @USERNAME AS VARCHAR(100)

DECLARE @APPNAME AS VARCHAR(100)

DECLARE @HOSTNAME AS VARCHAR(100)

SELECT @USERNAME=loginame,@APPNAME=program\_name,@HOSTNAME=hostname FROM MASTER.DBO.SYSPROCESSES WHERE SPID=@@SPID

INSERT INTO DBLOG.DBO.LG\_015\_ITEMS\_LOG

SELECT \*,@ACTIONTYPE,@DATE,@APPNAME,@USERNAME, @HOSTNAME FROM DELETED

END

\*\*TRIGGER çalıştığında 2 tane tablo kontrol ederiz. -–INSERTED, DELETED tabloları—

1-Delete işlemleri olduğunda, DELETED tablosunda tutulur eski bilgiler. INSERTED YOK.

2-Insert işlemleri olduğunda, INSERTED tablosunda tutulur. DELETED YOK.

3-Update işlemleri olduğunda, DELETED tablosunda eski bilgiler, INSERTED tablosunda yeni bilgiler tutulur.

=======STORED PROCEDURE============

CREATE PROC SPDENEME

@YIL AS INT =2015,

@AY AS VARCHAR(20)='01.OCAK'

AS

SELECT TOP \* FROM SATISLAR2015 WHERE YIL=@YIL AND AY=@AY

\*\*\*18 SANİYE BU SORGU

\*\*\*\*ÇAĞIRMA---🡪>>>EXEC SPDENEME @YIL=2015 , @AY='02.ŞUBAT'

Parametrelerin yazılma sırası önemli değildir.

\*\*Hız Önemli olduğu için view kullanmak antıklı değildir.View select komutunu doğrudan selct cümleciğini stored procedure içine yazmak daha mantıklıdır.

ALTER PROC [dbo].[SPDENEME]

@YIL AS INT =2015,

@AY AS VARCHAR(20)='01.OCAK'

AS

SELECT

STL.LOGICALREF,

CONVERT(VARCHAR,DATEPART(YEAR,STL.DATE\_))+'-'+CONVERT(VARCHAR,STL.STFICHEREF) AS STFICHEREF,

DATEPART(YEAR,STL.DATE\_) AS YIL,

dbo.getMonthName(STL.DATE\_) AS AY,

STL.DATE\_ AS TARIH,WHS.NAME AS SUBEADI,

ITM.LOGICALREF AS ITEMREF,

ITM.CODE AS MALZEMEKODU,ITM.NAME AS MALZEMEADI,

CATMAIN.DESCR AS ANAKATEGORI,

CAT.DESCR AS KATEGORI,

MRK.DESCR AS MARKA,STL.AMOUNT AS MIKTAR,STL.LINENET AS TUTAR,

USL.NAME AS BIRIM,

STL.PRICE AS BIRIMFIYAT

FROM LG\_015\_01\_STLINE STL

JOIN L\_CAPIWHOUSE WHS ON WHS.NR=STL.SOURCEINDEX AND WHS.FIRMNR=15

JOIN LG\_015\_ITEMS ITM ON ITM.LOGICALREF=STL.STOCKREF

JOIN LG\_015\_MARK MRK ON MRK.LOGICALREF=ITM.MARKREF

JOIN LG\_015\_UNITSETL USL ON USL.LOGICALREF=STL.UOMREF

LEFT JOIN LK\_015\_01\_PURCHCATEGORY CAT ON CAT.CODE=ITM.STGRPCODE

LEFT JOIN LK\_015\_01\_PURCHCATEGORY CATMAIN ON SUBSTRING(CAT.CODE,1,3)=CATMAIN.CODE

WHERE STL.TRCODE=7 AND DATEPART(YEAR,STL.DATE\_)=@YIL

AND dbo.getMonthName(STL.DATE\_)=@AY

\*\*\*\*BU SORGU 10 SN VIEW KULLANMADIĞIMIZ İÇİNN\*\*\*\*\*

\*\*\*Biraz daha hızlandırmak için fonksiyon kullanmamalıyızz\*\*\*\*

ALTER PROC [dbo].[SPDENEME]

@YIL AS INT =2015,

@AY AS VARCHAR(20)='01.OCAK'

AS

SELECT

STL.LOGICALREF,

CONVERT(VARCHAR,DATEPART(YEAR,STL.DATE\_))+'-'+CONVERT(VARCHAR,STL.STFICHEREF) AS STFICHEREF,

DATEPART(YEAR,STL.DATE\_) AS YIL,

dbo.getMonthName(STL.DATE\_) AS AY,

STL.DATE\_ AS TARIH,WHS.NAME AS SUBEADI,

ITM.LOGICALREF AS ITEMREF,

ITM.CODE AS MALZEMEKODU,ITM.NAME AS MALZEMEADI,

CATMAIN.DESCR AS ANAKATEGORI,

CAT.DESCR AS KATEGORI,

MRK.DESCR AS MARKA,STL.AMOUNT AS MIKTAR,STL.LINENET AS TUTAR,

USL.NAME AS BIRIM,

STL.PRICE AS BIRIMFIYAT

FROM LG\_015\_01\_STLINE STL

JOIN L\_CAPIWHOUSE WHS ON WHS.NR=STL.SOURCEINDEX AND WHS.FIRMNR=15

JOIN LG\_015\_ITEMS ITM ON ITM.LOGICALREF=STL.STOCKREF

JOIN LG\_015\_MARK MRK ON MRK.LOGICALREF=ITM.MARKREF

JOIN LG\_015\_UNITSETL USL ON USL.LOGICALREF=STL.UOMREF

LEFT JOIN LK\_015\_01\_PURCHCATEGORY CAT ON CAT.CODE=ITM.STGRPCODE

LEFT JOIN LK\_015\_01\_PURCHCATEGORY CATMAIN ON SUBSTRING(CAT.CODE,1,3)=CATMAIN.CODE

WHERE STL.TRCODE=7

AND STL.DATE\_ BETWEEN '20150201' AND '20150228'

\*\*BU SORGU 2 SN SÜRDÜ STL.DATE\_ YAPTIKTAN SONRA

\*\*\*\*DAHADA HIZLANDIRMAK İÇİN AY VE YILI AYRI AYRI DEĞİLDE AYNI SÜTUNDA KONTROL EDERSEK YİNE HIZLANDIRMIŞ OLURUZ.

ALTER PROC [dbo].[SPDENEME]

@YIL AS INT =2015,

@AY AS INT =3

AS

DECLARE @BASTARIH AS DATETIME

DECLARE @BITTARIH AS DATETIME

SET @BASTARIH =DATEADD(MONTH,@AY-1,'20150101')

SET @BITTARIH =DATEADD(MONTH,@AY,'20150101')

SELECT

STL.LOGICALREF,

CONVERT(VARCHAR,DATEPART(YEAR,STL.DATE\_))+'-'+CONVERT(VARCHAR,STL.STFICHEREF) AS STFICHEREF,

DATEPART(YEAR,STL.DATE\_) AS YIL,

dbo.getMonthName(STL.DATE\_) AS AY,

STL.DATE\_ AS TARIH,WHS.NAME AS SUBEADI,

ITM.LOGICALREF AS ITEMREF,

ITM.CODE AS MALZEMEKODU,ITM.NAME AS MALZEMEADI,

CATMAIN.DESCR AS ANAKATEGORI,

CAT.DESCR AS KATEGORI,

MRK.DESCR AS MARKA,STL.AMOUNT AS MIKTAR,STL.LINENET AS TUTAR,

USL.NAME AS BIRIM,

STL.PRICE AS BIRIMFIYAT

FROM LG\_015\_01\_STLINE STL

JOIN L\_CAPIWHOUSE WHS ON WHS.NR=STL.SOURCEINDEX AND WHS.FIRMNR=15

JOIN LG\_015\_ITEMS ITM ON ITM.LOGICALREF=STL.STOCKREF

JOIN LG\_015\_MARK MRK ON MRK.LOGICALREF=ITM.MARKREF

JOIN LG\_015\_UNITSETL USL ON USL.LOGICALREF=STL.UOMREF

LEFT JOIN LK\_015\_01\_PURCHCATEGORY CAT ON CAT.CODE=ITM.STGRPCODE

LEFT JOIN LK\_015\_01\_PURCHCATEGORY CATMAIN ON SUBSTRING(CAT.CODE,1,3)=CATMAIN.CODE

WHERE STL.TRCODE=7

AND STL.DATE\_ >=@BASTARIH AND STL.DATE\_<@BITTARIH

\*\*BU SORGU 0 SN SÜRDÜ STL.DATE\_ YAPTIKTAN SONRA

ARANOT:: SP\_UPDATESTATS --🡪>>Güncellemeleri kontrol etmek için

ARANOT2:: SP\_HELPTEXT SPDENEME 🡪> Stored procedure textini getirir.

===================TRANSACTION=======================

BEGIN TRAN

INSERT INTO TESTDATE1 (ITEMCODE,AMOUNT,DATE\_) VALUES('SEL.KBDF',0,GETDATE())

IF @@ERROR <> 0

ROLLBACK

ELSE

UPDATE TESTDATE1 SET AMOUNT = 15 WHERE AMOUNT = 0

IF @@ERROR<>0 san

ROLLBACK

ELSE

COMMIT

NOTTT:: Eğer transactıon yada benzeri işlemlerden dolayı sorgular bekletiliyorsa, beklemek istemiyor sorgumuzun sonuna WITH (NOLOCK) yazarak işlem gerçekleştirilebilir.

=========== CASE----WHEN/THEN KULLANIMIII===================

**SELECT** (**CASE** KolonAdi

**WHEN** 'Kolondan Gelen Değer' **THEN** 'Result'da gosterilecek değer'

end) **FROM** TabloAdi

🡪>

SELECT TOP 50000 (CASE TRCODE

WHEN 7 THEN 'SATIŞ'

WHEN 1 THEN 'ALIM'

ELSE 'DİĞER' END) AS TRCODE,\* FROM LG\_015\_01\_STLINE

======================IF-ELSE===========================

**IF**(koşul)

**BEGIN**

*--Eğer koşulumuz doğru ise bu alandaki ifademiz çalışır.*

**END**

**ELSE**

**BEGIN**

*--Eğer koşulumuz doğru değilse bu alandaki ifademiz çalışır.*

**END**