

Bekleyen Sepetlerdeki Urunlerin Fiyatinin Guncellenebilmesi Uyarlamasi

Fiyatlar iskontolu, koşullu vb durumda geldiğinde bekleyen sepette saklanır.

Bu sebeple; ürün fiyatında değişiklik yapıldığında da sepette beklemekte olan ürünlere bir yansıma olmaz.(sepet fiyatlarının bozulmaması için)

Bu durum aşağıdaki örnek uyarlamalar ile özelleştirilebilir:

1) Trigger kullanımı

Aşağıdaki örnek ürün stok kartı FİYAT1 alanında güncelleme yapıldığında, ilgili ürün bulunan bekleyen sepetlerin durumlarını güncelleyerek sepete tekrar giriş yapıldığında yeni fiyattan hesaplama yapılmasını sağlar:

Trigger Örnek - Fiyat1 Güncelleme olduğunda tetiklenir

```
CREATE TRIGGER [dbo].[B2B_U_TBLSTSABIT_PRICE]
ON [dbo].[TBLSTSABIT]
FOR UPDATE NOT FOR REPLICATION
AS
BEGIN
    SET NOCOUNT ON

    DECLARE @STOK_KODU varchar (35)
    DECLARE @SATIS_FIAT1 decimal

    DECLARE @OLD_STOK_KODU varchar (35)
    DECLARE @OLD_SATIS_FIAT1 decimal

    DECLARE CURSOR_I CURSOR FOR
    SELECT
        STOK_KODU ,SATIS_FIAT1

    FROM INSERTED FOR READ ONLY
    OPEN CURSOR_I
    FETCH NEXT FROM CURSOR_I INTO
        @STOK_KODU , @SATIS_FIAT1

    DECLARE CURSOR_D CURSOR FOR
    SELECT
        STOK_KODU, SATIS_FIAT1

    FROM DELETED FOR READ ONLY
    OPEN CURSOR_D
    FETCH NEXT FROM CURSOR_D INTO
        @OLD_STOK_KODU , @OLD_SATIS_FIAT1

    WHILE @@FETCH_STATUS <> -1
    BEGIN
        IF (@STOK_KODU IS NOT NULL)
        BEGIN
            IF ((ISNULL(@SATIS_FIAT1, '') <> ISNULL(@OLD_SATIS_FIAT1, '')))
            BEGIN
                UPDATE PRT_BASKETITEM SET STATUS = 0
                WHERE ID IN (
                    SELECT BI.ID FROM PRT_BASKETITEM BI (NOLOCK)
                    JOIN PRT_BASKET B (NOLOCK) ON B.ID = BI.BASKET_ID AND B.STATUS = 1
                    WHERE BI.STOCK_CODE = @STOK_KODU
                )

                UPDATE PRT_BASKET SET NEEDS_RECALCULATION=1, PRICE_EDIT_STATUS=0
                WHERE ID IN (
                    SELECT B.ID FROM PRT_BASKETITEM BI (NOLOCK)
                    JOIN PRT_BASKET B (NOLOCK) ON B.ID = BI.BASKET_ID AND B.STATUS = 1
                    WHERE BI.STOCK_CODE = @STOK_KODU
                )
            END
        END
    END
```

```

        )

    END
END

    FETCH NEXT FROM CURSOR_I INTO

        @STOK_KODU, @SATIS_FIAT1

    FETCH NEXT FROM CURSOR_D INTO
        @STOK_KODU , @SATIS_FIAT1
END

CLOSE CURSOR_I
CLOSE CURSOR_D
DEALLOCATE CURSOR_I
DEALLOCATE CURSOR_D

SET NOCOUNT OFF
END

```

2) Günlük SQL Job kullanımı

Aşağıdaki örnek her sabah saat 05:00'te çalışan bir SQL job oluşturur. Böylece tüm sepetler tekrar hesaplanacak şekilde güncellenir, fiyat değişikliği varsa yeni fiyat geçerli olur:

SQL Günlük Job Örnek

```

USE [msdb]
GO

/*
Not: DATABASE ismi için B2BTEST olan yerlere ilgili veritabanı yazılmalıdır.
*/

/***** Object:  Job [Her Sabah - Bekleyen Sepetlerdeki Ürünlerin Fiyat Güncellemesi Alabilmesi Salanr]
Script Date: 28.06.2018 15:46:06 *****/
BEGIN TRANSACTION
DECLARE @ReturnCode INT
SELECT @ReturnCode = 0
/***** Object:  JobCategory [[Uncategorized (Local)]]      Script Date: 28.06.2018 15:46:06 *****/
IF NOT EXISTS (SELECT name FROM msdb.dbo.syscategories WHERE name=N'[Uncategorized (Local)]' AND
category_class=1)
BEGIN
EXEC @ReturnCode = msdb.dbo.sp_add_category @class=N'JOB', @type=N'LOCAL', @name=N'[Uncategorized (Local)]'
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback

END

DECLARE @jobId BINARY(16)
EXEC @ReturnCode =  msdb.dbo.sp_add_job @job_name=N'Her Sabah - Bekleyen Sepetlerdeki Ürünlerin Fiyat
Güncellemesi Alabilmesi Salanr',
        @enabled=1,
        @notify_level_eventlog=0,
        @notify_level_email=0,
        @notify_level_netsend=0,
        @notify_level_page=0,
        @delete_level=0,
        @description=N'PRT_BASKET ve PRT_BASKETITEM için STATUS ve NEEDS_RECALCULATION alanlar
varsayılana çekilerek, fiyat güncellemelerinin bekleyen sepetteki ürünler için de alınması salayan günlük
güncellemedir. Not: Plasiyer vb tarafından girilmiş iskontolar vb değişiklikler de edilerek tekrar hesaplama
yapar.',
        @category_name=N'[Uncategorized (Local)]',

```

```

        @owner_login_name=N'sa', @job_id = @jobId OUTPUT
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
/***** Object: Step [PRT_BASKET Update] Script Date: 28.06.2018 15:46:06 *****/
EXEC @ReturnCode = msdb.dbo.sp_add_jobstep @job_id=@jobId, @step_name=N'PRT_BASKET Update',
        @step_id=1,
        @cmdexec_success_code=0,
        @on_success_action=3,
        @on_success_step_id=0,
        @on_fail_action=2,
        @on_fail_step_id=0,
        @retry_attempts=0,
        @retry_interval=0,
        @os_run_priority=0, @subsystem=N'TSQL',
        @command=N'UPDATE PRT_BASKET
SET NEEDS_RECALCULATION=1,
PRICE_EDIT_STATUS=0
where STATUS = 1',
        @database_name=N'B2BTEST',
        @flags=0
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
/***** Object: Step [PRT_BASKETITEM Update] Script Date: 28.06.2018 15:46:06 *****/
EXEC @ReturnCode = msdb.dbo.sp_add_jobstep @job_id=@jobId, @step_name=N'PRT_BASKETITEM Update',
        @step_id=2,
        @cmdexec_success_code=0,
        @on_success_action=1,
        @on_success_step_id=0,
        @on_fail_action=2,
        @on_fail_step_id=0,
        @retry_attempts=0,
        @retry_interval=0,
        @os_run_priority=0, @subsystem=N'TSQL',
        @command=N'UPDATE PRT_BASKETITEM
SET STATUS = 0
WHERE ID IN
(
SELECT BI.ID FROM PRT_BASKETITEM BI (NOLOCK)

JOIN PRT_BASKET B (NOLOCK)
ON B.ID = BI.BASKET_ID WHERE
B.STATUS = 1
)',
        @database_name=N'B2BTEST',
        @flags=0
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
EXEC @ReturnCode = msdb.dbo.sp_update_job @job_id = @jobId, @start_step_id = 1
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
EXEC @ReturnCode = msdb.dbo.sp_add_jobschedule @job_id=@jobId, @name=N'HerSabah05:00',
        @enabled=1,
        @freq_type=4,
        @freq_interval=1,
        @freq_subday_type=1,
        @freq_subday_interval=0,
        @freq_relative_interval=0,
        @freq_recurrence_factor=0,
        @active_start_date=20180628,
        @active_end_date=99991231,
        @active_start_time=50000,
        @active_end_time=235959,
        @schedule_uid=N'a99ef6dd-20ff-4dd4-b19a-f33d16620195'
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
EXEC @ReturnCode = msdb.dbo.sp_add_jobserver @job_id = @jobId, @server_name = N'(local)'
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
COMMIT TRANSACTION
GOTO EndSave
QuitWithRollback:
        IF (@@TRANCOUNT > 0) ROLLBACK TRANSACTION
EndSave:
GO

```

