

Project

Calculate timing hours for a given range of dates

```
CREATE PROCEDURE GetWorkingHours
    @Start_Date DATETIME,
    @End_Date DATETIME
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @CurrentDate DATE = @Start_Date;
    DECLARE @TotalHours INT = 0;

    WHILE @CurrentDate <= @End_Date
    BEGIN
        DECLARE @IsSunday BIT = CASE WHEN DATEPART(WEEKDAY,
@CurrentDate) = 1 THEN 1 ELSE 0 END;
        DECLARE @DayOfWeek INT = DATEPART(WEEKDAY, @CurrentDate); --
1=Sunday, 7=Saturday
        DECLARE @DayOfMonth INT = DAY(@CurrentDate);

        DECLARE @IsSaturday BIT = CASE WHEN @DayOfWeek = 7 THEN 1 ELSE
0 END;

        -- Get 1st and 2nd Saturdays
        DECLARE @MonthStart DATE = DATEFROMPARTS(YEAR(@CurrentDate),
MONTH(@CurrentDate), 1);
        DECLARE @SaturdayCount INT = 0;
        DECLARE @d DATE = @MonthStart;

        WHILE MONTH(@d) = MONTH(@CurrentDate)
        BEGIN
            IF DATEPART(WEEKDAY, @d) = 7
            BEGIN
                SET @SaturdayCount = @SaturdayCount + 1;
                IF @d = @CurrentDate AND @SaturdayCount <= 2
                    BREAK;
                IF @SaturdayCount > 2
                    BREAK;
            END
            @d = @d + 1;
        END
    END
```

```

        SET @d = DATEADD(DAY, 1, @d);
    END

    DECLARE @Is1stOr2ndSaturday BIT = CASE
        WHEN DATEPART(WEEKDAY, @CurrentDate) = 7 AND @SaturdayCount
<= 2 THEN 1
        ELSE 0
    END;

    IF (@IsSunday = 1 OR @Is1stOr2ndSaturday = 1)
    BEGIN
        -- Include if it's start or end date
        IF @CurrentDate = CAST(@Start_Date AS DATE) OR @CurrentDate =
CAST(@End_Date AS DATE)
            SET @TotalHours = @TotalHours + 24;
        END
    ELSE
    BEGIN
        SET @TotalHours = @TotalHours + 24;
    END

    SET @CurrentDate = DATEADD(DAY, 1, @CurrentDate);
    END

    -- Insert result into output table
    INSERT INTO counttotalworkinhours (START_DATE, END_DATE,
NO_OF_HOURS)
    VALUES (@Start_Date, @End_Date, @TotalHours);
    END;

```

Create Output Table

```

CREATE TABLE counttotalworkinhours (
    START_DATE DATETIME,
    END_DATE DATETIME,
    NO_OF_HOURS INT
);

```

Executing the Procedure

- First Execution: EXEC GetWorkingHours '2023-07-12', '2023-07-13';
- Second Execution: EXEC GetWorkingHours '2023-07-01', '2023-07-17';
- View Results: SELECT * FROM counttotalworkinhours;

99 %

3

0

Ln: 7Ch: 16TABS CRLF

Results

Messages

	START_DATE	END_DATE	NO_OF_HOURS
1	2023-07-12 00:00:00.000	2023-07-13 00:00:00.000	48
2	2023-07-01 00:00:00.000	2023-07-17 00:00:00.000	312

Query executed successfully.

DESKTOP-B4CBV0B (16.0 RTM) | DESKTOP-B4CBV0B\Hp (76) | master | 00:00:00 | 2 rows