Total of Each Day of Week

Gets the total of each day of the week and then organizes the information with each day of the week in columns and the total weekly sum underneath using the pivot function.

```
;WITH cte AS (
  SELECT
    DATENAME(dw,releasetime) as DayOfWeekName
    ,COUNT(*) OVER () as TotalCount
  FROM
    @release
  WHERE
    releasetime >= DATEADD(DAY,- DATEPART(dw,GETDATE()) +
                     1,CAST(GETDATE() AS DATE))
    AND releasetime < DATEADD(DAY,7 - DATEPART(dw,GETDATE()) +
                        1,CAST(GETDATE() AS DATE))
)

SELECT *
FROM
  cte
  PIVOT (
    COUNT(DayOfWeekName)
    FOR DayOfWeekName IN (Sunday,Monday,Tuesday,Wednesday,Thursday,Friday,Saturday)
  ) p
```

Reverse Cumulative

Creates a reverse cumulative (minus the cumulative column on each row to the next)

```
I have this table:

id  ID  [Date]  Cumulative
----------------------------
1   x   Jan-10     10
3   x   Feb-10     40
7   x   Apr-10     60
9   x   May-10    100
2   y   Jan-10     20
6   y   Mar-10     40
8   y   Apr-10     60
10  y   May-10    100


I need to Reverse the "Cumulative" in MS SQL Server Query to be as the following:

id  ID  [Date]    Cumulative  Reversed
------------------------------------
1   x   Jan-10     10         10
3   x   Feb-10     40         30    -- 10 - 40 = 30
7   x   Apr-10     60         20    -- 40 - 60 = 20
9   x   May-10    100         40    -- 60 - 100 = 40
2   y   Jan-10     20         20    -- 40 - 20 = 20
6   y   Mar-10     40         20    -- 20 - 40 = 20
8   y   Apr-10     60         20    -- 40 - 60 = 20
10  y   May-10    100         40    -- 60 - 100 = 40
```

```sql
/*
For below SQL server 2012 using recursive CTE for reverse running total.
*/
DECLARE @t TABLE(id INT,IDs VARCHAR(20),Dates VARCHAR(20),Cumulative INT)
INSERT INTO @t VALUES
(1,'x','Jan-10',  10)
,(3,'x','Feb-10',  40)
,(7,'x','Apr-10',  60)
,(9,'x','May-10', 100)
,(2,'y','Jan-10',  20)
,(6,'y','Mar-10',  40)
,(8,'y','Apr-10',  60)
,(10,'y','May-10',100)
```

```sql
;WITH CTE AS
(SELECT *,row_number()OVER(PARTITION BY ids ORDER BY id)rn
FROM @t
)
,CTE1 AS
(SELECT id,ids,dates, Cumulative,rn,Cumulative Reversed
FROM cte WHERE rn=1
UNION ALL
SELECT c.id,c.ids,c.Dates,c.Cumulative,c.rn
,c.Cumulative - c1.Cumulative
FROM cte c
    INNER JOIN cte c1 ON c.ids=c1.ids
WHERE c.rn = c1.rn+1
)
SELECT * FROM cte1


/*
You can use lag to get the value in the previous row and
subtract from the current row's value to get the reversed value.
*/
SELECT t.*, Cumulative - COALESCE(lag(Cumulative) OVER(PARTITION BY id ORDER BY
[Date]),0) AS Reversed
FROM tablename t

/*
You can use lag(Cumulative,1,0) instead of coalesce.
*/
SELECT t.*, Cumulative-lag(Cumulative,1,0) OVER(PARTITION BY id ORDER BY [Date])
                                                                            AS Reversed
FROM tablename t
```