Задача:

В таблице хранятся сведения о еженедельных продажах продукта.

Необходимо настроить автоматическое преобразование еженедельных значений в ежемесячные. Продажи в переходных неделях (часть недели в одном месяце, часть в другом) необходимо распределить по будним дням (исключая выходные дни - сб, вск).

В данном примере продажи за неделю, окончившуюся 05.03.2013, необходимо распределить следующим образом 2 дня на февраль, 3 дня на март.

Результатом решения задачи является SQL код процедуры, которая позволит автоматически получать преобразованные данные в формате «Таблица результат» (месяц;сумма) по введенному параметру MonthNumber (номер месяца). Если параметр не указан, выводятся вся таблица.

**Таблица с данными – условное название «Test».**

|  |  |
| --- | --- |
| Дата | Сумма |
| 26.02.2013 | 312,00 |
| 05.03.2013 | 833,00 |
| 12.03.2013 | 225,00 |
| 19.03.2013 | 453,00 |
| 26.03.2013 | 774,00 |
| 02.04.2013 | 719,00 |
| 09.04.2013 | 136,00 |
| 16.04.2013 | 133,00 |
| 23.04.2013 | 157,00 |
| 30.04.2013 | 850,00 |
| 07.05.2013 | 940,00 |
| 14.05.2013 | 933,00 |
| 21.05.2013 | 422,00 |
| 28.05.2013 | 952,00 |
| 04.06.2013 | 136,00 |
| 11.06.2013 | 701,00 |
|  | 8 676,00 |

**КОД ПРОЦЕДУРЫ**

USE [aero]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER PROCEDURE [dbo].[ToMonth]

@MonthNumber real = 0

AS

BEGIN

SET NOCOUNT ON;

WITH cte AS

(

SELECT ROW\_NUMBER() OVER (ORDER BY (SELECT 1)) - 1 AS [Incrementor]

FROM [master].[sys].[columns] sc1

CROSS JOIN [master].[sys].[columns] sc2

),

Data\_table as (Select months, CAST(sum(payments) AS DECIMAL(15, 2)) as payments

from

(Select MONTH(day) months,

case

when payment\_sum is null

then CAST((max(payment\_sum) OVER(ORDER BY MONTH(day) ASC ROWS BETWEEN CURRENT ROW AND 4 FOLLOWING)) AS DECIMAL(15, 2))/5

else (case

when lag(MONTH(day)) OVER(ORDER BY MONTH(day) ASC) is not null

then CAST(payment\_sum AS DECIMAL(15, 2))/5

else CAST(payment\_sum AS DECIMAL(15, 2))

end)

end payments

from payments a

right join

(SELECT DATEADD(DAY, cte.[Incrementor], (SELECT min(payment\_dt) FROM payments)) as day

FROM cte

WHERE DATEADD(DAY, cte.[Incrementor], (SELECT min(payment\_dt) FROM payments)) <= (SELECT max(payment\_dt) FROM payments)) b

on a.payment\_dt=b.day

where DATEPART(WEEKDAY, day) not in (6, 7)) m

group by months)

SELECT \* from Data\_table

where months in

(select case

when @MonthNumber in (select months from Data\_table)

then @MonthNumber

else months

end filter

from Data\_table)

order by months

END

**ПРОЦЕДУРА:**

****

**РЕЗУЛЬТАТЫ**

EXEC[ToMonth] 2

|  |  |
| --- | --- |
| months | payments |
| 2 | 645.20 |

EXEC[ToMonth] 15

|  |  |
| --- | --- |
| months | payments |
| 2 | 645.20 |
| 3 | 2383.20 |
| 4 | 1563.60 |
| 5 | 3328.60 |
| 6 | 755.40 |