1.For date filter use #mm/dd/yyyy#

2.select ID,Title,iif(Type=1,'A','B') as Type1,iif([Check]=1,'C','D') as Check1,iif(IsHome=1,'E','F') as Home1,[date] from XS\_news

3.SELECT A.[SUMMONS NO], A.[PLATE ID], A.DATE

FROM Summonses A

WHERE A.[DATE]=#10/15/2015#;

4. The Microsoft Access Month function returns the month (a number from 1 to 12) given a date value.

The Microsoft Access Now function returns the current system date and time.

5. WHERE ((IIf([Summonses]![Month]=Month(Now())-1,1,0))=1) AND ((IIf([Summonses]![Year]=Year(Now()-30),1,0))=1)

6. 微软官方对transform…pivot语句的说明：  
TRANSFORM 语句 (Microsoft Access SQL)  
创建交叉表查询。  
语法  
TRANSFORM aggfunction     selectstatement     PIVOT pivotfield [IN (value1[, value2[, ...]])]  
TRANSFORM 语句包含以下部分：  
aggfunction        对所选数据进行计算的 SQL 聚合函数。  
selectstatement        SELECT 语句。  
pivotfield        希望用于创建查询结果集中列标题的字段或表达式。  
value1、value2        用于创建列标题的固定值。

说明  
使用交叉表查询汇总数据时，将从作为列标题的指定字段或表达式中选择值，以便能够以一种比使用选择查询更紧凑的方式来查看数据。  
TRANSFORM 是可选的，但如果包括它，则应为 SQL 字符串中的第一个语句。它在指定作为行标题的字段的 SELECT 语句之前，在指定行分组方法的 GROUP BY 子句之前。您也可以包含其他子句（例如。指定其他选择或排序条件的 WHERE）。还可以在交叉表查询中使用子查询作为谓词，特别是在 WHERE 子句中。    
pivotfield 中返回的值作为查询结果集中的列标题。例如，在交叉表查询中如果依据月销售量来透视销售数据，将会创建 12 个列。可以约束 pivotfield 以便从可选 IN 子句中所列出的固定值（value1，value2）内选择标题。也可以包含固定值用于没有数据来创建其他列的情况。

TRANSFORM Sum(Summonses.OPEN) AS SumOfOPEN

SELECT Routes.[Merged Zone]

FROM Routes INNER JOIN Summonses ON Routes.[SUMMONS NO] = Summonses.[SUMMONS NO]

WHERE (iif([Summonses]![Month]=Month(Now())-1,1,0)=1 AND iif([Summonses]![Year]=Year(Now()-30),1,0)=1)

GROUP BY Routes.[Merged Zone]

PIVOT Summonses.Date;

7. Weekday( date [, firstdayofweek ] )

<https://support.office.com/en-US/article/Weekday-Function-05D360D7-2C3F-4691-9448-C96EA0351940>

Return to number 1 to 7 based on which first day you set, default is Sunday=1, like if you put 4=first day of the week which means Wed, so if the date is Wed, it will return to 1

**Settings**

The ***firstdayofweek*** argument has these settings:

| **Constant** | **Value** | **Description** |
| --- | --- | --- |
| **vbUseSystem** | 0 | Use the NLS API setting. |
| **vbSunday** | 1 | Sunday (default) |
| **vbMonday** | 2 | Monday |
| **vbTuesday** | 3 | Tuesday |
| **vbWednesday** | 4 | Wednesday |
| **vbThursday** | 5 | Thursday |
| **vbFriday** | 6 | Friday |
| **vbSaturday** | 7 | Saturday |

**Return Values**

The **Weekday** function can return any of these values:

| **Constant** | **Value** | **Description** |
| --- | --- | --- |
| **vbSunday** | 1 | Sunday |
| **vbMonday** | 2 | Monday |
| **vbTuesday** | 3 | Tuesday |
| **vbWednesday** | 4 | Wednesday |
| **vbThursday** | 5 | Thursday |
| **vbFriday** | 6 | Friday |
| **vbSaturday** | 7 | Saturday |

8.

Now()-7 refer the 7 days ago

Summonses!Date +(7-Weekday(Summonses!Date,4)) AS [Week End Date]

WHERE (((IIf([Summonses]![DATE]>= ((Now()-7)+(7-Weekday(Now()-7,4)) -28),1,0))=1))

TRANSFORM Sum(Summonses.OPEN) AS SumOfOPEN

SELECT Summonses.[Date] +(7-Weekday(Summonses.[Date],2)) AS [Week End Date],Summonses.[Date]

FROM Routes INNER JOIN Summonses ON Routes.[SUMMONS NO] = Summonses.[SUMMONS NO]

WHERE (((IIf([Summonses]![DATE]>= ((Now()-7)+(7-Weekday(Now()-7,2)) -21),1,0))=1)) and Routes.[Merged Zone] in ('503', '608', '970', '971')

GROUP BY Summonses.[Date] +(7-Weekday(Summonses.[Date],2)),Summonses.[Date]

PIVOT Routes.[Merged Zone];