Queries to Demonstrate the Date/Time Functions in Access
3/14/2012

Most of the date fields in Voyager contain both a date and a time. When you use Microsoft Access to query the Voyager tables, it normally gives you the date and time combined. But sometimes you may want just a part of the date/time. There may be queries for which you want just the date, or just the year, or just the time, or the day of the week. There may be queries for which you want intervals of time, such as everything done in the last 30 days. This document shows a variety of functions and techniques for doing interesting things like these in Access.

In each section, the SQL is provided so that you can paste it into Access and modify it. The design view and the table of results will be useful to you as you read this document.

But first, there are a few date fields in Voyager that have only a date:

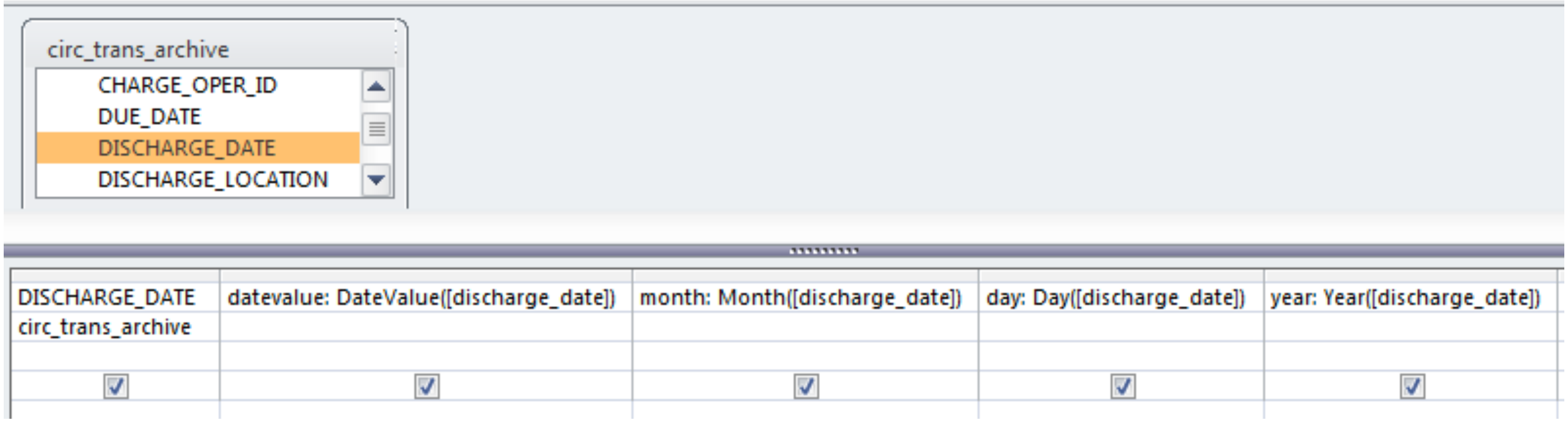
CALENDAR.CALENDAR_BEGIN_DATE
CALENDAR.CALENDAR_END_DATE
EXCEPTION_CALENDAR.EXCEPTION_DATE
FISCAL_PERIOD.START_DATE
MAINTENANCE.DATE_IN
MAINTENANCE.DATE_OUT
MONO_CLAIM.CLAIM_DATE
MONO_CLAIM_ARCHIVE.CLAIM_DATE
MONO_SUPPLIER_REPORT.REPORT_DATE
RESERVE_LIST.EFFECT_DATE
RESERVE_LIST.EXPIRE_DATE
SERIAL_CLAIM.CLAIM_DATE
SERIAL_CLAIM_ARCHIVE.CLAIM_DATE
SERIAL_ISSUES.EXPECTED_DATE
SERIAL_SUPPLIER_REPORT.REPORT_DATE

Some of the bibliographic fields in Voyager normally hold a date and time, but the data that came from your previous system have only a date. The time, for these, defaults to midnight on the morning of the date given:

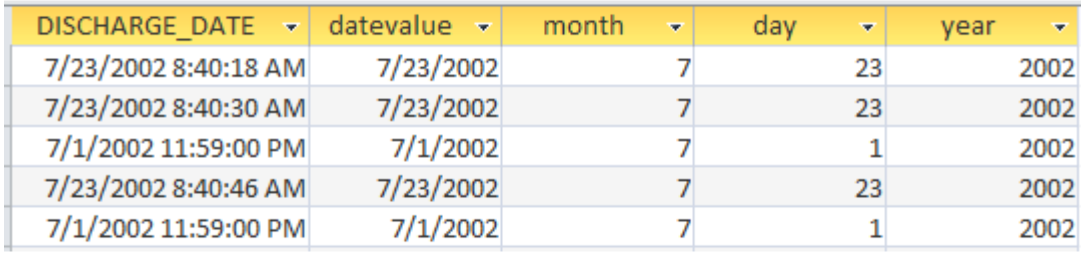
AUTH_HISTORY.ACTION_DATE
AUTH_MASTER.CREATE_DATE
BIB_HISTORY.ACTION_DATE
BIB_ITEM.ADD_DATE
BIB_MASTER.CREATE_DATE
HEADING.CREATE_DATE
ITEM.CREATE_DATE
ITEM_STATUS.ITEM_STATUS_DATE
MFHD_HISTORY.ACTION_DATE
MFHD_MASTER.CREATE_DATE
SUBDIVISION.CREATE_DATE

Most of the dates in the Voyager database actually have both the date and time recorded. This query shows how you can pull out each part of date into its own column. (We'll look at the time part of each date/time on the next page.)

```
SELECT TOP 5
      circ_trans_archive.DISCHARGE_DATE,
      DateValue([discharge_date]) AS datevalue,
      Month([discharge_date]) AS [month],
      Day([discharge_date]) AS [day],
      Year([discharge_date]) AS [year]
FROM circ_trans_archive;
```



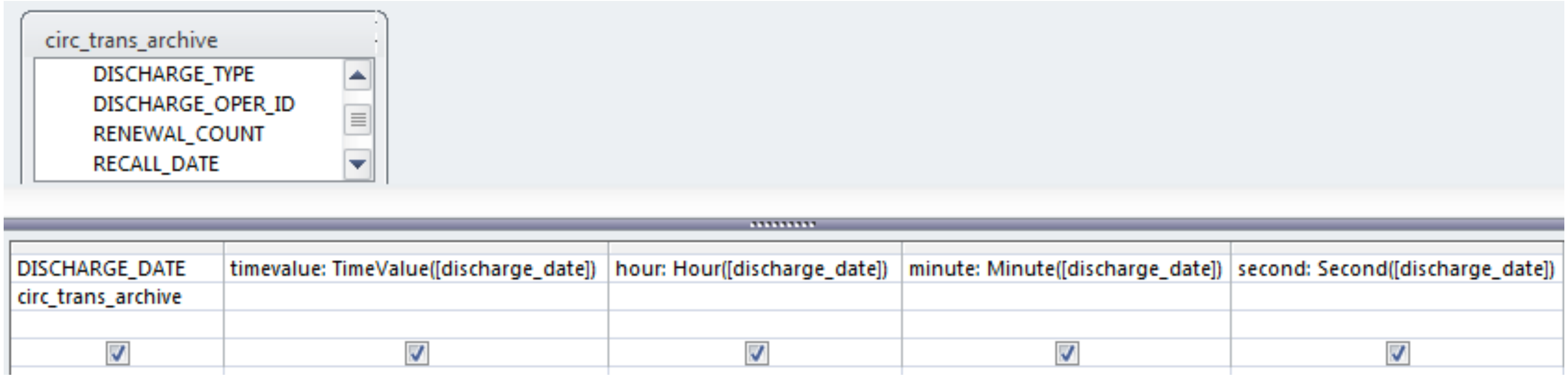
DISCHARGE_DATE	datevalue: DateValue([discharge_date])	month: Month([discharge_date])	day: Day([discharge_date])	year: Year([discharge_date])
circ_trans_archive				
☑	☑	☑	☑	☑



DISCHARGE_DATE	datevalue	month	day	year
7/23/2002 8:40:18 AM	7/23/2002	7	23	2002
7/23/2002 8:40:30 AM	7/23/2002	7	23	2002
7/1/2002 11:59:00 PM	7/1/2002	7	1	2002
7/23/2002 8:40:46 AM	7/23/2002	7	23	2002
7/1/2002 11:59:00 PM	7/1/2002	7	1	2002

This query shows you how to pull out each part of the time into its own column:

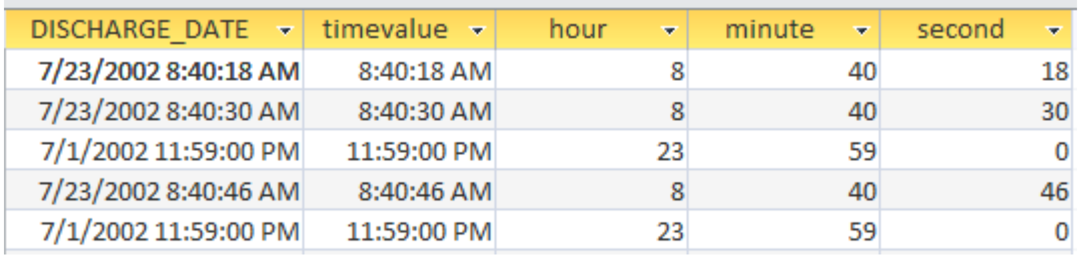
```
SELECT TOP 5
circ_trans_archive.DISCHARGE_DATE,
      TimeValue([discharge_date]) AS timevalue,
      Hour([discharge_date]) AS [hour],
      Minute([discharge_date]) AS [minute],
      Second([discharge_date]) AS [second]
FROM circ_trans_archive;
```



circ_trans_archive

- DISCHARGE_TYPE
- DISCHARGE_OPER_ID
- RENEWAL_COUNT
- RECALL_DATE

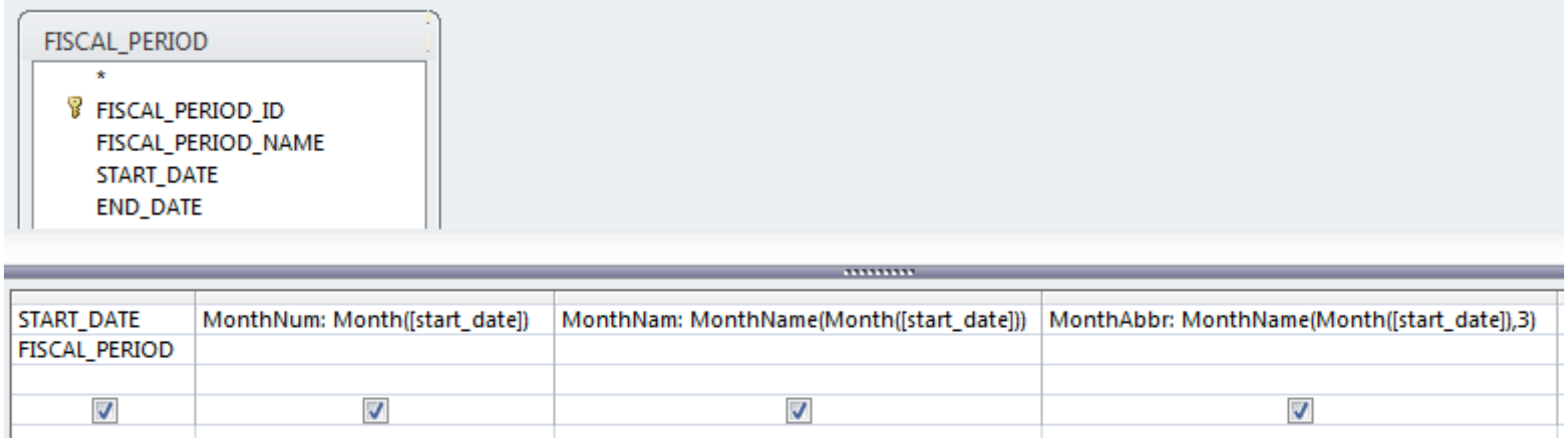
DISCHARGE_DATE	timevalue: TimeValue([discharge_date])	hour: Hour([discharge_date])	minute: Minute([discharge_date])	second: Second([discharge_date])
circ_trans_archive				
☑	☑	☑	☑	☑



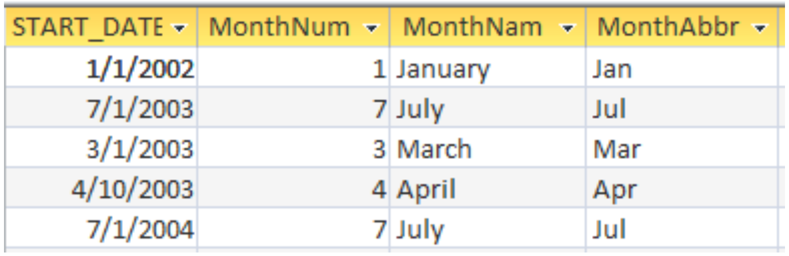
DISCHARGE_DATE	timevalue	hour	minute	second
7/23/2002 8:40:18 AM	8:40:18 AM	8	40	18
7/23/2002 8:40:30 AM	8:40:30 AM	8	40	30
7/1/2002 11:59:00 PM	11:59:00 PM	23	59	0
7/23/2002 8:40:46 AM	8:40:46 AM	8	40	46
7/1/2002 11:59:00 PM	11:59:00 PM	23	59	0

This query shows how to spell out the name of the month:

```
SELECT FISCAL_PERIOD.START_DATE,
Month([start_date]) AS MonthNum,
MonthName(Month([start_date])) AS MonthNam,
MonthName(Month([start_date]),3) AS MonthAbbr
FROM FISCAL_PERIOD;
```



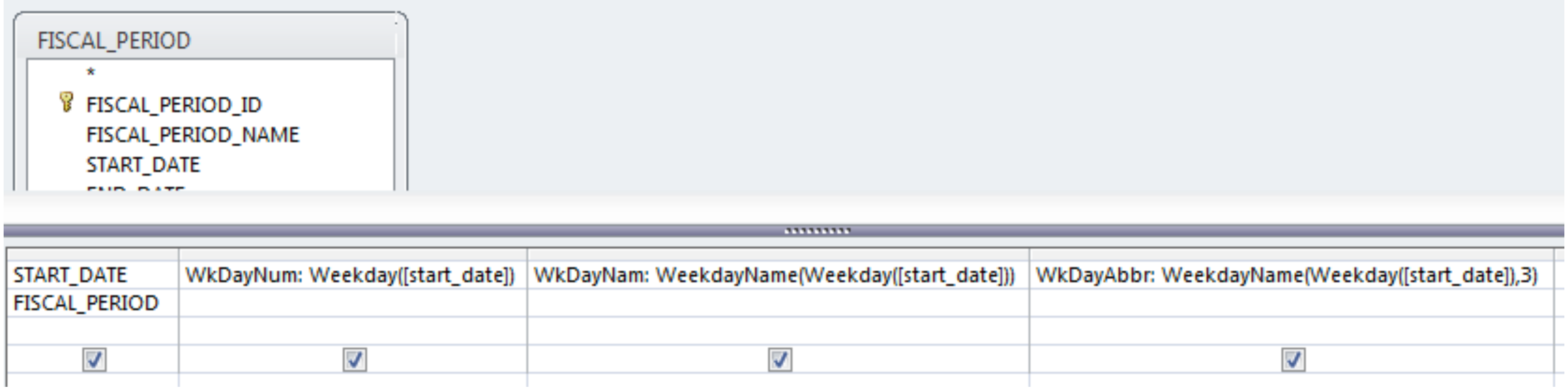
START_DATE	MonthNum: Month([start_date])	MonthNam: MonthName(Month([start_date]))	MonthAbbr: MonthName(Month([start_date]),3)
FISCAL_PERIOD			
☑	☑	☑	☑



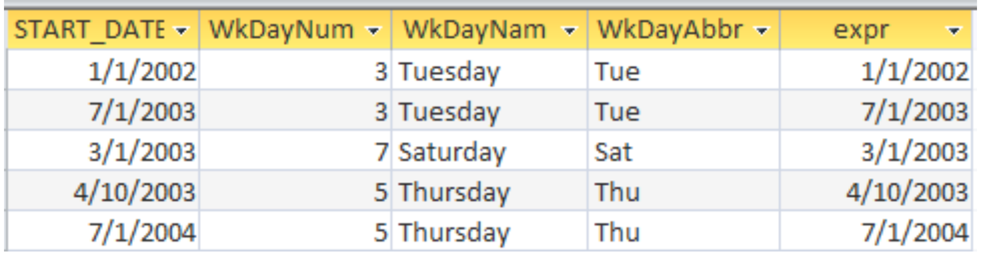
START_DATE	MonthNum	MonthNam	MonthAbbr
1/1/2002	1	January	Jan
7/1/2003	7	July	Jul
3/1/2003	3	March	Mar
4/10/2003	4	April	Apr
7/1/2004	7	July	Jul

This query shows how you can spell out the day of the week.

```
SELECT FISCAL_PERIOD.START_DATE,
Weekday([start_date]) AS WkDayNum,
WeekdayName(Weekday([start_date])) AS WkDayNam,
WeekdayName(Weekday([start_date]),3) AS WkDayAbbr
FROM FISCAL_PERIOD;
```



START_DATE	WkDayNum: Weekday([start_date])	WkDayNam: WeekdayName(Weekday([start_date]))	WkDayAbbr: WeekdayName(Weekday([start_date]),3)
FISCAL_PERIOD			
☑	☑	☑	☑



START_DATE	WkDayNum	WkDayNam	WkDayAbbr	expr
1/1/2002	3	Tuesday	Tue	1/1/2002
7/1/2003	3	Tuesday	Tue	7/1/2003
3/1/2003	7	Saturday	Sat	3/1/2003
4/10/2003	5	Thursday	Thu	4/10/2003
7/1/2004	5	Thursday	Thu	7/1/2004

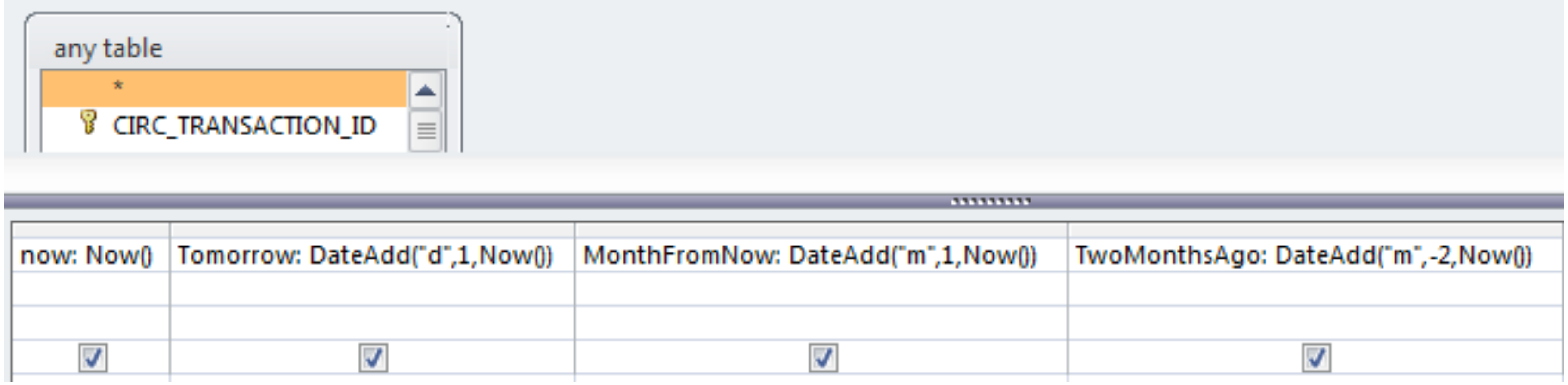
There are some more Access functions that let you do interesting things with dates:

Now() gives you today's date.

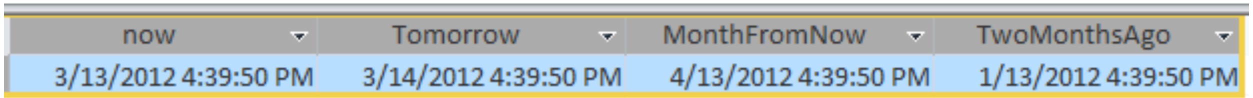
DateAdd lets you add or subtract days, weeks, or months from a date.

For example: DateAdd(now(),30,d) means "30 days from now"
            DateAdd(now(),-30,d) means "30 days ago"

```
SELECT TOP 1
      Now() AS now,
      DateAdd("d",1,Now()) AS Tomorrow,
      DateAdd("m",1,Now()) AS MonthFromNow,
      DateAdd("m",-2,Now()) AS TwoMonthsAgo
FROM CIRC_TRANS_ARCHIVE;
```



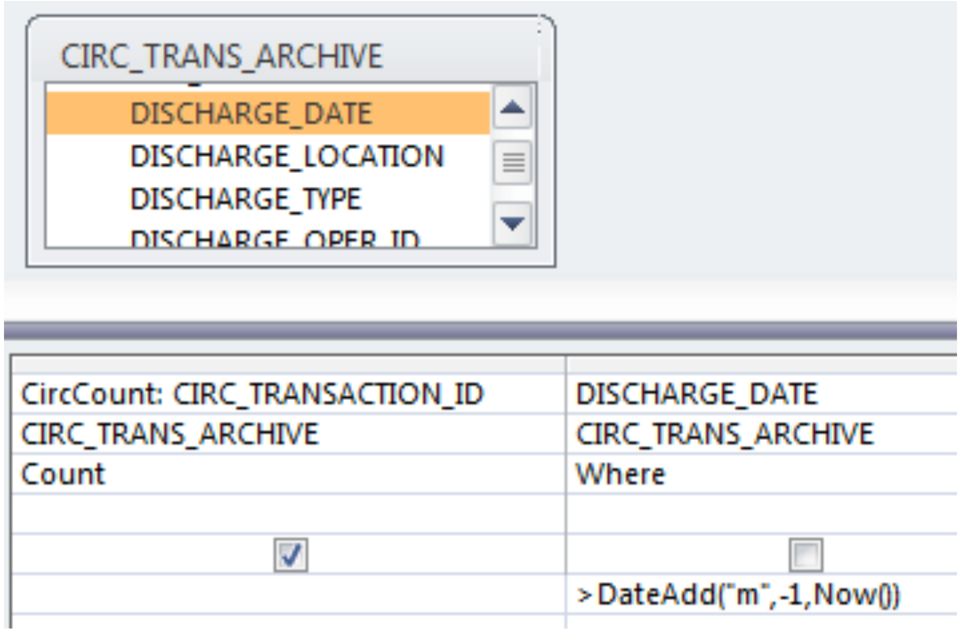
now: Now()	Tomorrow: DateAdd("d",1,Now())	MonthFromNow: DateAdd("m",1,Now())	TwoMonthsAgo: DateAdd("m",-2,Now())
☑	☑	☑	☑



now	Tomorrow	MonthFromNow	TwoMonthsAgo
3/13/2012 4:39:50 PM	3/14/2012 4:39:50 PM	4/13/2012 4:39:50 PM	1/13/2012 4:39:50 PM

Of course, it's not very interesting to ask Access what time it is. A more interesting question would be: How many discharges have we done in the past month?

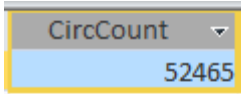
```
SELECT
      Count([CIRC_TRANS_ARCHIVE].[CIRC_TRANSACTION_ID]) AS CircCount
FROM CIRC_TRANS_ARCHIVE
WHERE (((CIRC_TRANS_ARCHIVE.DISCHARGE_DATE)>DateAdd("m",-1,Now())));
```



CIRC_TRANS_ARCHIVE

DISCHARGE_DATE
DISCHARGE_LOCATION
DISCHARGE_TYPE
DISCHARGE_OPER_ID

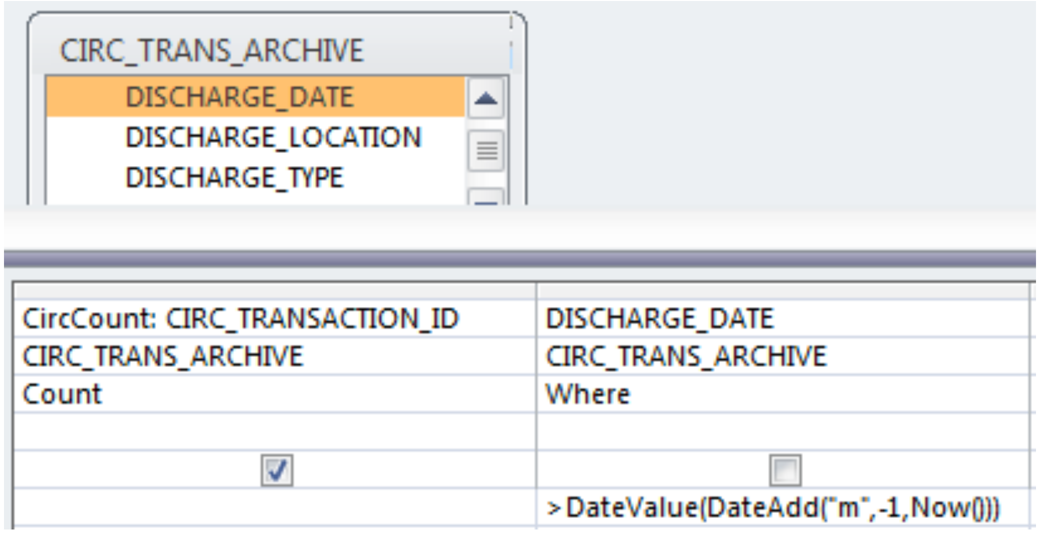
CircCount: CIRC_TRANSACTION_ID	DISCHARGE_DATE
CIRC_TRANS_ARCHIVE	CIRC_TRANS_ARCHIVE
Count	Where
☑	□
	>DateAdd("m",-1,Now())



CircCount
52465

The DateAdd function in the previous query is almost painfully literal. If you run it at 3pm today, you'll count discharges done since 3pm one month ago. Here's a query that shows how to count the discharges done since midnight one month ago, which is perhaps a cleaner way to count. Since DateValue extracts just the date from the date/time, Access treats it as though the time were midnight on the morning of that date:

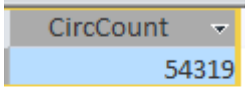
```
SELECT Count([CIRC_TRANS_ARCHIVE].[CIRC_TRANSACTION_ID]) AS CircCount
FROM CIRC_TRANS_ARCHIVE
WHERE (((CIRC_TRANS_ARCHIVE.DISCHARGE_DATE)>DateValue(DateAdd("m",-1,Now()))));
```



CIRC_TRANS_ARCHIVE

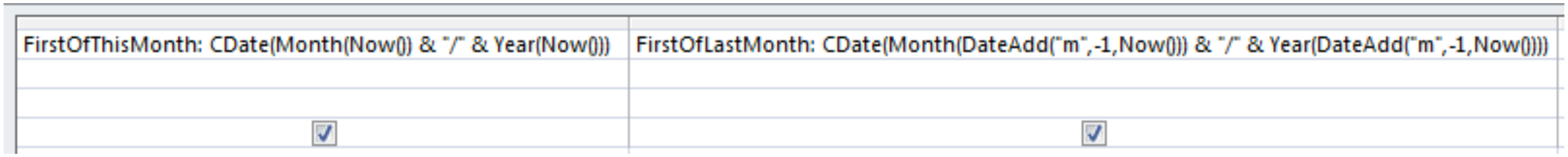
DISCHARGE_DATE
DISCHARGE_LOCATION
DISCHARGE_TYPE

CircCount: CIRC_TRANSACTION_ID	DISCHARGE_DATE
CIRC_TRANS_ARCHIVE	CIRC_TRANS_ARCHIVE
Count	Where
☑	☐
	>DateValue(DateAdd("m",-1,Now()))



CircCount
54319

We could build a query that will always give you the data from the previous calendar month. For example, if I run it on any day in March, it will give me the data for the entire month of February. The date range that I need to construct is February 1 of this year through March 1 of this year. Consider this snippet:

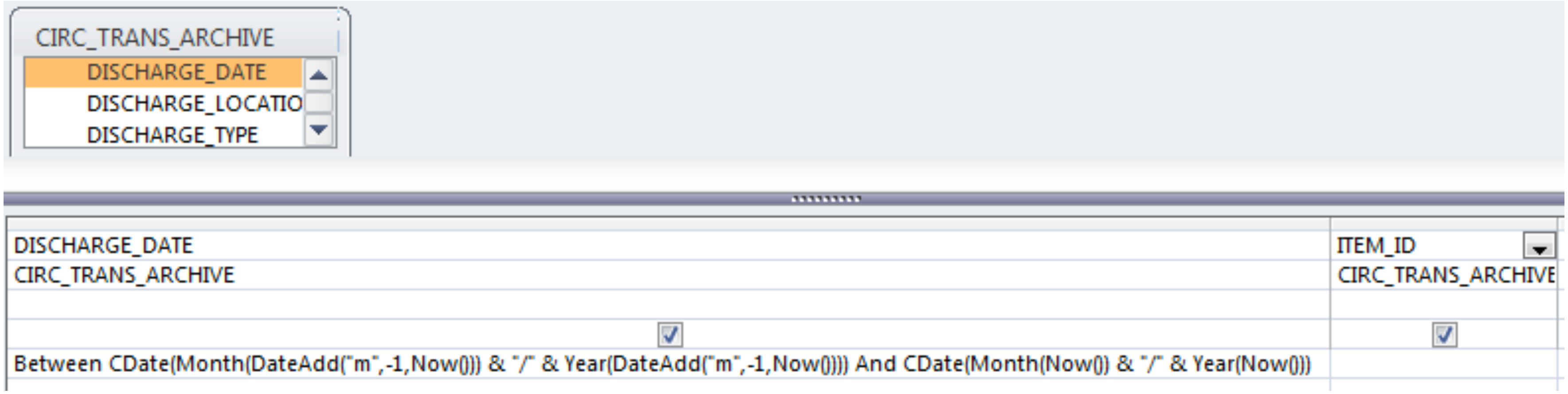


FirstOfThisMonth: CDate(Month(Now()) & "/" & Year(Now()))	FirstOfLastMonth: CDate(Month(DateAdd("m",-1,Now())) & "/" & Year(DateAdd("m",-1,Now())))
☑	☑

In the first column, you already know what Month(Now()) and Year(Now()) will do. I use an ampersand to stick them together with a slash in between. Then I use CDate to make it into a date. Since I didn't provide a day of the month, CDate assumes the first of the month. The second column is like the first except that I use DateAdd to subtract a month from Now().

Here's a query that lists items discharged in the previous calendar month:

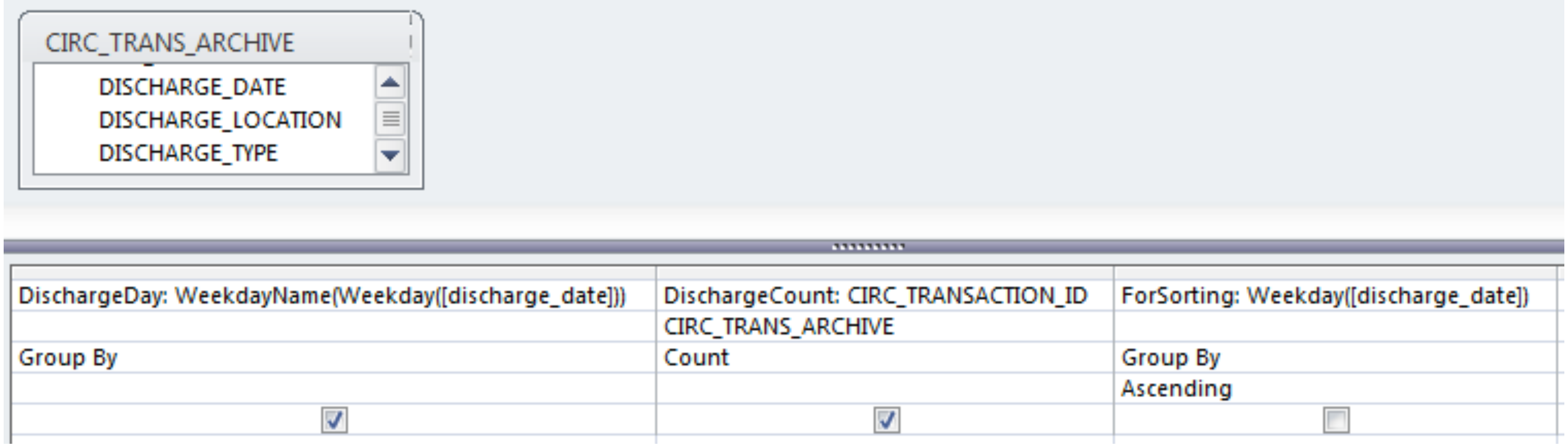
```
SELECT CIRC_TRANS_ARCHIVE.DISCHARGE_DATE, CIRC_TRANS_ARCHIVE.ITEM_ID
FROM CIRC_TRANS_ARCHIVE
WHERE (((CIRC_TRANS_ARCHIVE.DISCHARGE_DATE) Between
CDate(Month(DateAdd("m",-1,Now())) & "/" & Year(DateAdd("m",-1,Now()))) And
CDate(Month(Now()) & "/" & Year(Now()))));
```



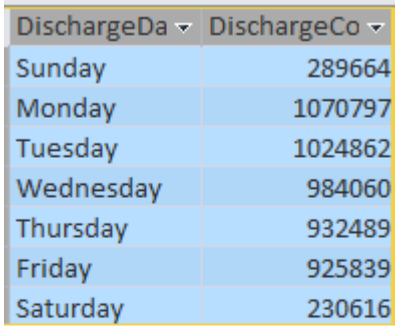
DISCHARGE_DATE	ITEM_ID
CIRC_TRANS_ARCHIVE	CIRC_TRANS_ARCHIVE
☑	☑
Between CDate(Month(DateAdd("m",-1,Now())) & "/" & Year(DateAdd("m",-1,Now()))) And CDate(Month(Now()) & "/" & Year(Now()))	

Another interesting question might be: On what day of the week do we do the most discharges? Notice that I sorted by Weekday, not WeekdayName. If I had sorted by WeekdayName, the days of the week would have been in alphabetical, not chronological, order.

```
SELECT
WeekdayName(Weekday([discharge_date])) AS DischargeDay,
Count(CIRC_TRANS_ARCHIVE.CIRC_TRANSACTION_ID) AS DischargeCount
FROM CIRC_TRANS_ARCHIVE
GROUP BY WeekdayName(Weekday([discharge_date])), Weekday([discharge_date])
ORDER BY Weekday([discharge_date]);
```



DischargeDay: WeekdayName(Weekday([discharge_date]))	DischargeCount: CIRC_TRANSACTION_ID	ForSorting: Weekday([discharge_date])
	CIRC_TRANS_ARCHIVE	
Group By	Count	Group By
		Ascending
☑	☑	☐



DischargeDa	DischargeCo
Sunday	289664
Monday	1070797
Tuesday	1024862
Wednesday	984060
Thursday	932489
Friday	925839
Saturday	230616