**SQL Foundation**

Filtering dates info

© FDM Group Ltd 2017. All Rights Reserved.

Any unauthorised reproduction or distribution in part  
or in whole will constitute an infringement of copyright.

**Document change and version control**

|  |  |  |  |
| --- | --- | --- | --- |
| **Reason for Change** | **Author** | **Date of Change** | **New Version No** |
| Creation of document | Nick Lawton | 26/06/2017 | v1.0 |
| Rectified the inaccurate statement “when filtering by a fixed number of days from today’s date, work directly with SYSDATE” and more examples included to illustrate when to use TO\_CHAR and when not. | Nikola Ignjatovic | 11/11/2021 | v.1.1 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**SQL - Filtering dates info**

Sometimes filtering dates involves using TO\_CHAR, sometimes it doesn’t. Sometimes we use LIKE, sometimes we use ‘=’ or ‘>’ or ‘<’. This guide explains when to use each method.

**Storing date values**

Oracle stores values of DATE data type in seven separated fixed-length fields of NUMBER data type, each corresponding to century, year, month, day, hour, minute, and second

For some date values the time element is not important (e.g. date of birth or hire date).

In such cases, dates will be inserted into table columns without the time element, but Oracle will store them with the default time as midnight (00:00:00 or 12:00:00am)

In other cases, when the time element is important, the time will be provided along with the date when inserted (e.g. date of a financial transaction, flight departure date).

**Use TO\_CHAR in these situations:**

If the task requires ignoring the time part altogether or considering a section of the date & time value (e.g. YY, YYMM, YYMMDD, HH, HH:MI etc.), it will be necessary to use TO\_CHAR() conversion function to extract the part(s) od the date that we wish to compare or work with.

* This month
  + TO\_CHAR(column,’YYMM’) = TO\_CHAR(SYSDATE,’YYMM’)
* This year
  + TO\_CHAR(column,’YY’) = TO\_CHAR(SYSDATE,’YY’)
* This quarter
  + TO\_CHAR(column,’YYQ’) = TO\_CHAR(SYSDATE,’YYQ’)
* Last month
  + TO\_CHAR(column,’YYMM’) = TO\_CHAR(ADD\_MONTHS(SYSDATE,-1),’YYMM’)
* Last year
  + TO\_CHAR(column,’YY’) = TO\_CHAR(ADD\_MONTHS(SYSDATE,-12),’YY’)
* Last quarter
  + TO\_CHAR(column,’YYQ’) = TO\_CHAR(ADD\_MONTHS(SYSDATE,-3),’YYQ’)
* The month before last
  + TO\_CHAR(column,’YYMM’) = TO\_CHAR(ADD\_MONTHS(SYSDATE,-2),’YYMM’)
* The year before last
  + TO\_CHAR(column,’YY’) = TO\_CHAR(ADD\_MONTHS(SYSDATE,-24),’YY’)
* The quarter before last
  + TO\_CHAR(column,’YYQ’) = TO\_CHAR(ADD\_MONTHS(SYSDATE,-6),’YYQ’)
* The last 7 days (excluding the 7th day before today)
  + TO\_CHAR(column, 'YYYYMMDD') > TO\_CHAR(SYSDATE - 7, 'YYYYMMDD')
* The last 7 days (including the 7th day before today)
  + TO\_CHAR(column, 'YYYYMMDD') >= TO\_CHAR(SYSDATE - 7, 'YYYYMMDD')
* The last 30 days (excluding the 30th day before today)
  + TO\_CHAR(column, 'YYYYMMDD') > TO\_CHAR(SYSDATE - 30, 'YYYYMMDD')
* The last 30 days (including the 30th day before today)
  + TO\_CHAR(column, 'YYYYMMDD') >= TO\_CHAR(SYSDATE - 30, 'YYYYMMDD')
* Since this day last month (excluding this day last month)
  + TO\_CHAR(column,'YYMMDD') >

TO\_CHAR(ADD\_MONTHS(SYSDATE,-1),'YYMMDD')

* Since this day last month (including this day last month)
  + TO\_CHAR(column,'YYMMDD') >=

TO\_CHAR(ADD\_MONTHS(SYSDATE,-1),'YYMMDD')

* Since this day last quarter (excluding this day last quarter)
  + TO\_CHAR(column,'YYMMDD') >

TO\_CHAR(ADD\_MONTHS(SYSDATE,-3),'YYMMDD')

* Since this day last quarter (including this day last quarter)
  + TO\_CHAR(column,'YYMMDD') >=

TO\_CHAR(ADD\_MONTHS(SYSDATE,-3),'YYMMDD')

* Since this day last year (excluding this day last year)
  + TO\_CHAR(column,'YYMMDD') >

TO\_CHAR(ADD\_MONTHS(SYSDATE,-12),'YYMMDD')

* Since this day last year (including this day last year)
  + TO\_CHAR(column,'YYMMDD') >=

TO\_CHAR(ADD\_MONTHS(SYSDATE,-12),'YYMMDD')

* Within next month (until but excluding this day next month)
  + TO\_CHAR(column,'YYMMDD') <

TO\_CHAR(ADD\_MONTHS(SYSDATE,1),'YYMMDD')

* Within next month (until and including this day next month)
  + TO\_CHAR(column,'YYMMDD') <= TO\_CHAR(ADD\_MONTHS(SYSDATE,1),'YYMMDD')
* Today
  + TO\_CHAR(column, 'DD-MON-YY') = TO\_CHAR(SYSDATE, 'DD-MON-YY')
* Yesterday
  + TO\_CHAR(column, 'DD-MON-YY') = TO\_CHAR(SYSDATE-1, 'DD-MON-YY')
* Tomorrow
  + TO\_CHAR(column, 'DD-MON-YY') = TO\_CHAR(SYSDATE+1, 'DD-MON-YY')
* 2 days ago
  + TO\_CHAR(column, 'DD-MON-YY') = TO\_CHAR(SYSDATE-2, 'DD-MON-YY')
* 2 days from now
  + TO\_CHAR(column, 'DD-MON-YY') = TO\_CHAR(SYSDATE+2, 'DD-MON-YY')
* 7 days ago
  + TO\_CHAR(column, 'DD-MON-YY') = TO\_CHAR(SYSDATE-7, 'DD-MON-YY')
* 7 days from now
  + TO\_CHAR(column, 'DD-MON-YY') = TO\_CHAR(SYSDATE+7, 'DD-MON-YY')

**Don’t use TO\_CHAR in these situations:**

If the task requires considering the entire date part (DD-MM-YYYY) and the entire time part (HH:MI:SS), we need to work directly with the column(s) and/or SYSDATE in the WHERE or HAVING clause, without using TO\_CHAR.

In all of the examples below where date columns are being compared, it is assumed that all elements of DATE data type need to be compared (**up to and including the seconds**)

* Dates that are missing (have not been entered)

column IS NULL

* Dates in column1 later than the corresponding dates in column2

column1 > column2

* Dates in column1 later than or equal to the corresponding dates in column2

column1 >= column2

* Dates in column1 earlier than the corresponding dates in column2

column1 < column2

* Dates in column1 earlier than or equal to the corresponding dates in column2

column1 <= column2

* Dates in column in the last 12 hours (half a day)

column > SYSDATE - 0.5

* Dates in column1 between the corresponding dates from other two columns

column1 BETWEEN column2 AND column3

* Earliest recorded date in the last 12 months (365/366 days):

MIN(column) > ADD\_MONTHS(SYSDATE,-12)

* Latest recorded date in the last 12 months (365/366 days):

MAX(column) > ADD\_MONTHS(SYSDATE,-12)

You may use LIKE where searching for rows on one specific date:

* Today column LIKE SYSDATE
* Yesterday column LIKE SYSDATE -1
* 7 days ago column LIKE SYSDATE - 7

although a better solution is to use TO\_CHAR to specify what parts of the date to compare, as then comparison does not rely on the default date format (‘DD-MON-YY’).

**Filtering by a specific date**

TO\_CHAR(column,’DD-MM-YYYY’) = ’21-06-2021’

or

column LIKE TO\_DATE(’21-06-2021’,’DD-MM-YYYY’)

or

TO\_NUMBER(TO\_CHAR(column, 'YYYYMMDD')) = 20210621

Any of these would find dates matching 21 June 2021

**Filtering between two specific dates (inclusively)**

BETWEEN TO\_DATE('09/10/2021', 'DD/MM/YYYY') AND TO\_DATE('09/11/2021', 'DD/MM/YYYY')