Up to this point we have talked about SELECT, FROM, PLAN/JOIN/WHERE clauses to extract data from the Millennium database. These clauses focus entirely on selecting the right tables and connecting then together to acquire the desired information from the database. They allow you to do almost anything you want. However, there are three additional clauses that allow you to order and group the data returned by a query, so it's easier to see.

**5.1 Order Clause**

The ORDER clause is an optional clause that follows the last WHERE or JOIN clause in a query. It allows you to sort data the query has extracted from the database. Expressions within the ORDER clause can either be columns from a table or user-defined variables from the SELECT clause (discussed in Chapter 2). CCL allows you to sort up to 10 levels deep, but you will rarely go beyond three or four.

To illustrate how the ORDER clause functions on three different field types, we will return to our SMITH example. Type the following query into DVDev and run it (CTRL + Q) to see how the data lists in your own environment. As always, if no patients exist with the last name SMITH, try another last name like TEST.

*select*

*p.name\_full\_formatted*

*, p.birth\_dt\_tm*

*, gender = uar\_get\_code\_display(p.sex\_cd)*

*from*

*person p*

*where p.name\_last\_key = "SMITH"*

*and p.sex\_cd > 0*

Machine generated alternative text:
Edit Query 
NAME FULL FORMATTED 
BIRTH DT TM 
10/14/82 
07/28/16 
10/27/ SS 
02/25/65 
09/23/75 
04/29/87 
09/14/32 
GENDER 
F emal e 
F emal e 
F emal e 
Mal e 
F ema I e 
F emal e 
F emal e 
SMITH, 
SMITH, 
SMITH, 
SMITH, 
SMITH, 
SMITH, 
SMITH, 
ABIGAIL 
ÄDÄLINE 
ADRIANA M 
ALAN L 
AMANDA 
AMANDA 
ANNE M 

The data is returned in the same order that it is written to the database. Let's order the data by gender. Add the ORDER clause after the last qualification and order by p.sex\_cd.

*select*

*p.name\_full\_formatted*

*, p.birth\_dt\_tm*

*, gender = uar\_get\_code\_display(p.sex\_cd)*

*from*

*person p*

*where p.name\_last\_key = "SMITH"*

*and p.sex\_cd > 0*

*order*

*p.sex\_cd*

As you can see, the data was completely re-ordered and none of the male SMITH's show up at the top.

Machine generated alternative text:
Edit Query 
NAME FULL FORMATTED 
BIRTH DT TM 
10/14/82 
07/28/16 
10/27/ SS 
09/23/75 
04/29/87 
09/14/32 
10/20/ SS 
GENDER 
F ema I e 
F emal e 
F ema I e 
F ema I e 
F emal e 
F ema I e 
F emal e 
SMITH, 
SMITH, 
SMITH, 
SMITH, 
SMITH, 
SMITH, 
SMITH, 
ABIGAIL 
ÄDÄLINE 
ADRIANA M 
AMANDA 
AMANDA 
ANNE M 
BILLIE 

To understand why female SMITHs listed first, remember that the SEX\_CD stores a code value from code set 57 for each gender. Females have a code value of 362 and males 363. The ORDER clause's default behavior is to list the returned data in ascending order. For \_CD, \_IND, \_FLAG, and \_ID fields that means listing the smallest number first. Thus, females are listed first.

Machine generated alternative text:
Edit Query 
CODE VALUE 
627757269 . oo 
362 . oo 
379924812.00 
363 .00 
364.00 
4040581 
CODE SET 
CDF MEANING 
AMBIGUOUS 
FEMALE 
INDETERMINAT 
MALE 
UNKNOWN 
UNSPECIFIED 
DISPLAY 
F emal e 
Inde t e Eminent 
Mal e 
un kn own 
Unspeci fled 

You can change the default behavior and list the data in descending order by typing DESC after an expression.

*select*

*p.name\_full\_formatted*

*, p.birth\_dt\_tm*

*, gender = uar\_get\_code\_display(p.sex\_cd)*

*, p.sex\_cd*

*from*

*person p*

*where p.name\_last\_key = "SMITH"*

*and p.sex\_cd > 0*

*order*

*p.sex\_cd desc*

Machine generated alternative text:
Edit Query 
NAME FULL FORMATTED 
363 .00 
363 .00 
363 .00 
363 .00 
363 .00 
.00 
BIRTH DT TM 
02/25/65 
11/12/49 
02/19/81 
06/15/26 
07/08/ so 
11/23/71 
GENDER 
Mal e 
Mal e 
Mal e 
Mal e 
Mal e 
Mal e 
SEX CD 
SMITH, 
SMITH, 
SMITH, 
SMITH, 
SMITH, 
SMITH, 
ALAN L 
WILLIE L 
BILLY 
BRADY O 
BRUCE 
cÄLv1N M 

If you order a date and time field the data is ordered chronologically. If you order a textual field, the data is displayed alphabetically.

Notice in the first query that we were ordering on the SEX\_CD field even though we are using a UAR on the SEX\_CD. This is entirely possible. You can have fields in an ORDER clause without them being in a SELECT clause. Another alternative to this query would be to use the user-defined variable gender in the ORDER clause. Writing the query this way is the same as using p.sex\_cd as a qualification. It's up to you to decide which way you prefer.

*select*

*p.name\_full\_formatted*

*, p.birth\_dt\_tm*

*, gender = uar\_get\_code\_display(p.sex\_cd)*

*from*

*person p*

*where p.name\_last\_key = "SMITH"*

*and p.sex\_cd > 0*

*order*

*gender*

If you want to order a query by the display of a code value, you *must* use a user-defined variable in the ORDER clause. This isn't possible by ordering the actual code value itself because numerical fields sort by size. Text values order alphabetically. The following query uses a user-defined variable, orderable, to store the text returned from the UAR. The ORDER clause uses the user-defined variable to sort the data alphabetically.

*select*

*p.person\_id*

*, p.name\_full\_formatted*

*, p.birth\_dt\_tm*

*, o.orig\_order\_dt\_tm*

*, orderable = uar\_get\_code\_display(oc.catalog\_cd)*

*from*

*person p*

*, encounter e*

*, orders o*

*, order\_catalog oc*

*plan p*

*where p.person\_id = 22822461*

*join e*

*where e.person\_id = p.person\_id*

*join o*

*where o.encntr\_id = e.encntr\_id*

*join oc*

*where oc.catalog\_cd = o.catalog\_cd*

*order by*

*orderable*

The ORDER clause can list up to 10 items to sort by. If we take the query above and add a second sort level on the ORIG\_ORDER\_DT\_TM what will happen is that it will list each order alphabetically. For each order that is repeated (a patient might have the same order more than once per encounter), it will list the orders chronologically.

*order by*

*orderable*

, o.orig\_order\_dt\_tm

Machine generated alternative text:
Edit Query 
PERSON ID 
22822461 
22822461 
22822461 
22822461 
22822461 
22822461 
22822461 
NAME FULL FORMATTED 
BIRTH DT TM 
07/28/16 
07/28/16 
07/28/16 
07/28/16 
07/28/16 
07/28/16 
07/28/16 
ORIG 
ORDER DT TM 
02/07/17 
06/19/17 
08/22/17 
11/14/17 
03/01/18 
07/16/18 
01/08/19 
ORDERABLE 
Fol I ow 
up 
up 
up 
up 
up 
up 
up 
SMITH, 
SMITH, 
SMITH, 
SMITH, 
SMITH, 
SMITH, 
SMITH, 
ÄDÄLINE 
ÄDÄLINE 
ÄDÄLINE 
ÄDÄLINE 
ÄDÄLINE 
ÄDÄLINE 
ÄDÄLINE 

**5.2 Group By Clause**

dfdfdadfa

There are three additional clauses that allow you to order the data and group the data

These clauses, while not all required, will be the majority of the queries you will ever need to write.

Returning constants

Returning conditionals

Group by

Having

With