**4.1 Overview**

The Millennium database contains a special table called the CODE\_VALUE table that stores every single code value. A code value is simply a numerical representation for some text. Text that repeats in Cerner is often stored as a code value because its far more efficient to store numbers in a database than text. Thus, the text is only stored one time and the code value is used to throughout the database to represent the text.

Every row on the PERSON table has a SEX\_CD. Instead of storing "Male" or "Female" millions of times repeatedly on the PERSON table, "Male" and "Female" are stored one time on code set 57. The SEX\_CD for each row on the PERSON table stores the code value associated with either "Male" or "Female".

Code values are comprised of five main fields:

|  |  |
| --- | --- |
| Field | Description |
| CODE\_VALUE | Unique, system assigned number. This is the value that is stored on other tables in \_CD fields. This is the primary key of the CODE\_VALUE table. |
| CODE\_SET | Code sets are logical groupings of code values. All gender code values have the same code set. |
| DISPLAY | The short description of the code value. This is generally what is used to display in applications. |
| DISPLAY\_KEY | Uppercase, alphanumeric version of DISPLAY (no special characters, spaces, or punctuation). |
| CDF\_MEANING | A standard meaning for the code value defined by the COMMON\_DATA\_FOUNDATION table. This field is not always populated. |
| DESCRIPTION | The description of the code value. |

**Exercise 4.1**

Load the PERSON table from the Tables/Fields tab in DVDev and find the SEX\_CD. Double clicking SEX\_CD will display the field properties. As you can see, the values on this field are stored in code set 57.

Machine generated alternative text:
name 
nation 
next r 
person 
person 
person 
purge_ 
race c 
religio 
reside 
rowid 
sex_ag 
T able Name: 
PERSON 
Code Set: 
Description: 
sex code value 
D efinition: 
ield Properti 
Field Name: 
SEX CD 
Lookup 
the sex/gender that the patient is considered to have for administration 
and record keeping purposes this is typically asserted by the patient 
when they present to administrative users this may not match the 
biological sex as determined by anatomy or genetics, or the individual's 
preferred identification (gender identityl 
Close 

Click on the Lookup button to view all of the code values on code set 57. Alternatively, you can right click on SEX\_CD from the Tables/Fields tab and select Code Lookup.

Machine generated alternative text:
Tables/ Fields 
v person 
Fields 
sex cd 
species cd 
9 
txn id text 
updt_appl 
updt_cnt 
9 
updt_dt tm 
updt_id 
Code... 
Output 
Add/Remove.. 
Remove All 
Filter Float... 
ndexes 
Uppercase 
Low e r case 
Properties... 
Code Lookup... 

Looking up the code set in this manner is synonymous to running the following query:

*select \* from code\_value where code\_set = 57*

Machine generated alternative text:
Edit Query 
CODE VALUE 
627757269 . 
362 . 
364. 
F emal e 
Mal e 
un kn own 
CDF MEANING 
AMBIGUOUS 
DESCRIPTION 
DISPLAY KEY 
AMBIGUOUS 
FEMALE 
UNKNOWN 
CKI 
CKI . CODEVÄLUE ! 2774 
. CODEVÄLUE ! 2773 
. CODEVÄLUE ! 277 S 
DEFINITION 
FEMALE 
MALE 
UNKNOWN 
Sex 
Sex 
Sex 
fema I e 
mal 
un kn own 

Now, let's look at a single PERSON row to see what is stored as the SEX\_CD. Run the following query in your environment.

*select sex\_cd, \* from person where sex\_cd > 0 with maxrec = 1*

Machine generated alternative text:
Edit Query 
SEX CD 
PERSON ID 
UPDT CNT 
362 .00 
S90EES 

The PERSON table stores code value 362 which represents "Female" from the CODE\_VALUE table (in my environment). Since 362 doesn't mean anything to an end user, we need a way to change the code value its display.

**4.2 UARs**

User access routines (UARs) are pieces of code CCL uses to gather information about code values. We have seen UAR\_GET\_CODE\_DISPLAY() already in previous examples, but there are several more useful ones. The following example shows four useful UARs.

*select*

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

*, display\_key = uar\_get\_displaykey(p.sex\_cd)*

*, description = uar\_get\_code\_description(p.sex\_cd)*

*, meaning = uar\_get\_code\_meaning(p.sex\_cd)*

*, p.sex\_cd*

*from*

*person p*

*where p.sex\_cd > 0*

*with maxrec = 5*

Machine generated alternative text:
SEX CD 
362 . oo 
362 . oo 
363 .00 
363 .00 
362 
Edit Query 
KEY 
DESCRIPTION 
MEANING 
FEMALE 
FEMALE 
FEMALE 
DISPLAY 
Femal 
Mal e 
Mal e 
F ema I e 
DISPLAY 
FEMALE 
FEMALE 
FEMALE 
Sex 
Sex 
Sex 
Sex 
Sex 
fema I e 
fema I e 
mal 
mal 
fema I e 

UARs return the text of the code value passed to them for the DISPLAY, DISPLAY\_KEY, DESCRIPTION, and CDF\_MEANING, respectively.

UARs never hit the database directly, therefore they are much more efficient than joining to the CODE\_VALUE table directly to pull the DISPLAY, DISPLAY\_KEY etc.

The same data could be returned by writing the following query. However, you should never join to the CODE\_VALUE table unless using a UAR will not work. There are scenarios where this is true, but they are rare enough that they are not worth mentioning.

*select*

*cv.display*

*, cv.display\_key*

*, cv.description*

*, cv.cdf\_meaning*

*, p.sex\_cd*

*from*

*person p*

*, code\_value cv*

*plan p*

*where p.sex\_cd > 0*

*join cv*

*where cv.code\_value = p.sex\_cd*

*with maxrec = 5*

**Exercise 4.2**

There is an easy way to use these UARs without having to memorize them.

Begin typing the following query in the text window. Make sure to leave out the UARs for the time being.

*select*

*display =*

*, display\_key =*

*, description =*

*, meaning =*

*, p.sex\_cd*

*from*

*person p*

*where p.sex\_cd > 0*

*with maxrec = 5*

Now, place your curser after "display = " and press CTRL + Spacebar. You should see a drop down window appear.

Machine generated alternative text:
select 
di spl ay 
spl ay 
descrip 
me an Ing 
p. sex c 
from 
person p 
here p. sex 
with maxrec 
accept 
aesclecrypt 
'e aesdecryptfile 
aesencrypt 
'e aesencryptfile 
alter 
'e alter2 
'e alterlist 

Begin typing "uar" and all of the UARs we previously discussed will be listed. Using the down arrow key on your keyboard, highlight uar\_get\_code\_display and then press "Tab."

Machine generated alternative text:
select 
display ua 
spl ay 
descript 
me an Ing 
p. sex cd 
from 
person p 
here p. sex 
ith maxrec 
'e uar get_code_by 
uar 
'e uar 
'e validate 
'e value 
variance 
'e video 
'4 warning 

Pressing "Tab" with an item selected gives you a short description of the item you selected. As you can see from this screen shot, the parameter accepted by this UAR is a code value.

Machine generated alternative text:
select 
display uar 
display key 
description 
me an Ing 
p. sex cd 
from 
person p 
here p. sex cd 
i th maxrec 
code 
display ( 
LIAR GET CODE DISPLAYßany_cb) 

Finish this line by adding p.sex\_cd between the brackets and complete the remaining lines with their appropriate UAR. Finally, run the query in your own environment.

When pressing CTRL + Spacebar like this or with a blank window, a list of many functions, UARs and reserved variables/keywords are listed. We will be using this strategy in the future to help us write more complex programs.

**4.3 Dynamically Using Code Values**

One of the most common problems in CCL scripts is hard coded code values. Hard coding data into a script means embedding data directly into a script that has to be changed every time that data changes. In this example, the PERSON\_ID is hard coded to only return orders for that person.

*select*

*p.person\_id*

*, p.name\_full\_formatted*

*, p.birth\_dt\_tm*

*, o.order\_mnemonic*

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

*from*

*person p*

*, encounter e*

*, orders o*

*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*

This is, of course, the desired behavior. But, what if we add one more qualification to limit the orders returned to orders with the CATALOG\_CD = 911502?

*select*

*p.person\_id*

*, p.name\_full\_formatted*

*, p.birth\_dt\_tm*

*, o.order\_mnemonic*

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

*from*

*person p*

*, encounter e*

*, orders o*

*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*

*and o.catalog\_cd = 911502.00*

Hard coding the PERSON\_ID is necessary to only returns orders for that person, but hard coding the CATALOG\_CD is a terrible practice that should never be done. **Code values should never be hard coded!** Code values are not always consistent between environments *even if your organization takes domain copies instead of doing manual build*. This means that the code values in your Production environment might not match other environments. Thus, your code may not work if you hard code!

The code value 911502 in my environment happens to correspond to a CBC with Differential orderable.

*select \* from code\_value where code\_set = 200 and display = "CBC\*"*

Machine generated alternative text:
Edit Query 
CODE VALUE 
€€47€1 
911 s02.oo 
CODE SET 
CDF 
MEANING 
DISPLAY 
CBC wi Chou t Differential 
CBC with Differential 
DISPLAY KEY 
CBCWITHCUTDIEEERENTIÄL 
CBCWITHDIFFERENTIÄL 
DESCRIPTION 
CBC wi Chou t Differential 
CBC with Differential 
200 
200 

Instead of hard coding 911502 as the CATALOG\_CD, you can use a UAR in combination with the function cnvtreal() to make this script dynamic.

*select*

*p.person\_id*

*, p.name\_full\_formatted*

*, p.birth\_dt\_tm*

*, o.order\_mnemonic*

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

*from*

*person p*

*, encounter e*

*, orders o*

*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*

*and o.catalog\_cd = cnvtreal(uar\_get\_code\_by("DISPLAYKEY",200,"CBCWITHDIFFERENTIAL")) ;911502.00*

The UAR is uar\_get\_code\_by() and it takes three arguments and each argument is separated by a comma.

* The first argument is either "DISPLAYKEY" or "MEANING".
* The second argument is the code set.
* The third argument is a lookup string corresponding to the first argument.

The cnvtreal() function converts a non-real expression to a real expression. This is necessary because of the way the UAR returns the data. Functions, including cnvtreal(), will be discussed in greater depth later in the book.

By writing the script this way, we have made it dynamic. Now, as long as all of the environments have the same DISPLAYKEY for the orderable, the query will work. If the orderable name is changed, it could possibly change the DISPLAYKEY, so that would be the only instance you would need to modify this query.

In the previous example, the DISPLAYKEY was only used because there was no CDF\_MEANING. **If there is a CDF\_MEANING, use it.** To illustrate this, the following query returns all General Laboratory orderables on a patient.

Machine generated alternative text:
Edit Query 
CODE VALUE 
€92 . 
General 
CDF 
Lab 
MEANING 
DESCRIPTION 
General Laboratory 
DISPLAY KEY 
GENERÄLLÄB 

*select*

*p.person\_id*

*, p.name\_full\_formatted*

*, p.birth\_dt\_tm*

*, o.order\_mnemonic*

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

*from*

*person p*

*, encounter e*

*, orders o*

*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*

*and o.activity\_type\_cd = cnvtreal(uar\_get\_code\_by("MEANING",106,"GLB"))*

**4.4 Global Code Cache**

UARs return their information from something called the Global Code Cache. The Code Cache Query (scp 52) and Code Cache Refresh Server (scp 49) work in tandem to keep code sets and code values up to date. They maintain a shared memory cache of both active code sets and code values that are used by many processes including UARs.

The "refresh delay" property on the refresh server controls how often the code cache is refreshed (in seconds). This means that if a user builds a code value, the UARs might not work until 10 minutes after it is built.

Machine generated alternative text:
scp:- s 
server 49 on node Cbaycf1app5J in domain Cc30J 
description: 
path : 
parameters : 
nstances : 
restart : 
login as: 
properti es : 
CPM code cache Refresh 
cer_exe/cpm_srvcacherefresh 
enabl ed 
system 
DependencyGroup O 
refresh delay 600 
user Name d c30 

**4.5 Core Code Builder**

In addition to looking up code values and code sets in DVDev by either writing a query or using the code value lookup functionality, the application CoreCodeBuilder.exe can be used to do the same thing. This book is by no means a build or a Core book, but it is useful to know your way around this tool, so I recommend spending a couple of minutes looking at all of the search capabilities within it.

Machine generated alternative text:
Code Value Management Code Set Management 
Code 57 : SEX 
Code Value Data (read-onb') 
Code Value Group 
Code Value Aias 
Code Value Out Bound 
Code Value 
Display Key 
MALE 
UNKNOWN 
UNSPECIFIED 
INDETERMINENT 
AMBIGUOUS 
Code Set 
Code Value 
382 
363 
364 
4040581 
37+24812 
627757269 
COF Meaning 
FEMALE 
MALE 
UNKNOWN 
UNSPECIFIED 
INDETERMINAT 
AMBIGUOUS 
Display 
Male 
Unknown 
Unspecifi ed 
Indeterminent 
Ambiguous 
Description 
Sexis female 
Sex is male 
Sex is unknown 
Unspecifi ed 
Indeterminent 
Ambiguous 
Definition 
FEMALE 
MALE 
UNKNOW 
Unspecifie 
Indetermin 
Ambiguou: 

The search capabilities of CoreCodeBuilder.exe changed in Millennium 2018 and may change in future versions, so I'm not going to go in depth into how to use this application. I recommend getting comfortable with this application because I have used it many times.