Cyberseminar Transcript

Date: May 23, 2018

Series: VIReC Corporate Data Warehouse

Session: Querying CDW Meta Views

Presenter: Margaret Gonsoulin, PhD

*This is an unedited transcript of this session. As such, it may contain omissions or errors due to sound quality or misinterpretation. For clarification or verification of any points in the transcript, please refer to the audio version posted at* [http://www.hsrd.research.va.gov/cyberseminars/catalog-archive.cfm](file:///C:\Users\VHASLCMyersK\AppData\Local\Microsoft\Windows\Temporary%20Internet%20Files\Content.IE5\73RXHSNF\l)

Heidi: Hira, can I turn things over to you?

Moderator: Thank you Heidi. Welcome everyone to VIReC’s Corporate Data Warehouse Cyberseminar series. Thank you to CIDER for the technical and promotional support. Today’s session is titled Querying CDW Meta Views, and it will be presented by Dr. Margaret Gonsoulin. Margaret is assistant director of VIReC, the VA Information Resource Center. She served as VIReC’s CDW data knowledge lead for the last four years, contributing to the core enterprise of understanding and articulating the nuances and use of VA data. She has also developed extensive training and knowledge resources related to the use of CDW data. Thank you for joining us today, Margaret.

Dr. Margaret Gonsoulin: Thank you for having me, and thank you everyone for being here. So let’s get started by talking about how to use meta views that are located on servers alongside CDW data to help you figure out what your data mean and when it’s appropriate to use it.

Before I begin, I'd like to take a moment to thank Richard Pham, Trinity Hall, Andy Kelly, and the whole team at BISL for the variety of support and assistance that they provided over the years.

And before we begin, I would like to take a moment to give just a little bit of background in the sense that, in general, before this our talks have focused on using these metadata but in a sort of manual search. So today we’ll take that and turn that into an electronic search using SQL query. So I'm going to remind you a little bit about that manual process of searching the metadata and how we’re now going to do this electronically and also remind you about some introductory talks that preceded this talk. So if some of this material is a little bit vague, these introductory talks may help you if you spend a little time with them and then you come back to this one.

So in previous talks, we’ve taken a look at the metadata as it exists in multiple places, but the primary place is on the Business Intelligence Service Line SharePoint site, which is BISL as you can see here. And if you come here to look for documentation of data you find in the Corporate Data Warehouse, then you’ll want to click on the link circled in red that reads CDW metadata. And that URL is here at the bottom of the screen if you’d like to navigate to that site in the future.

That will bring you to the metadata page in general, and in order to get to the actual documentation around the data, you would want to click the link that you see here in the purple box to launch the CDW metadata report.

And that will finally bring you to the metadata report itself. And this particular set of metadata for the Corporate Data Warehouse pertains to the production domains, which I’ll talk about in a little bit, and you can see them listed in alphabetical order. There are, last time I counted were over 60, and they are available for perusing manually on the website. When you do that, you might choose to click on the name of the domain itself to open up what is referred to as an entity relationship diagram, which is a diagram of what the domain looks like, so that each table appears in a box, and then you can see lines that show you how to connect the various tables together in this entity relationship diagram. Or you can click on the plus sign to the left of the domain name to open up a set of metadata for the tables located inside the domain, and then you can drill into the tables to see the columns and so on and so forth.

Now if any of the introductory querying or SQL information or just, in general, CDW information that we skim over today, goes a little too fast, you can find background information in the previous talks in this series. In the first three in the list about the conceptual overview, the manual search for locating documentation, and the basic SQL starter language would be the three that I would recommend if any of this is not totally clear.

Okay, so let’s get started looking at how to query CDW’s metadata views. So we’ll start first by finding the views themselves. They all begin with the same schema, which is M-E-T-A, or meta for metadata. And then we’ll go through examples of how to find information on a specific CDW column that you’re trying to use. Let’s say you have [inaudible 05:36] and you’re not sure what it means. How do you figure that out? Or capturing data on a specific topic, where you can search for that topic by key words, or searching for some information from a specific field or file from VistA that you know exists and you’re wondering if it’s [unintelligible 06:01] Corporate Data Warehouse. And then we’ll go through how to find information on the meaning of the linking keys that are scattered throughout the Corporate Data Warehouse and that you’ll use to connect the various tables and domains together to make your datasets. And finally we’ll take a quick look at some special columns that are intended to help improve the efficiency of SQL queries that you are writing against the warehouse.

So first let’s find the meta schema views. So as I mentioned, the part of the CDW that we are talking about when we look at these metadata, either manually or through these SQL queries against the meta views, pertains only to the part of the CDW referred to as production domain. And just in a really general way, that means the part of the CDW that has been modeled or architected to make it a bit easier to use. There is a whole other section of the CDW that is referred as raw domain, and they are a bit closer in nature to VistA, which is the most common origin for the data found in CDW. So they have less work done on them, and they are often a bit more of a learning curve for people to be able to use them. And just in case you’re not familiar with VistA, that’s the Veteran Information System and Technology Architecture, or the historic way of storing data that was built in-house in the VA over the last 20-plus years. And that’s where all the data from CPRS gets stored.

So getting to our meta views, first either from your desktop or from within VINCI you would open SQL Server Management Studio, or SSMS. That’s the software you’re going to write SQL in. And then you’re going to go to the server to which you have access. So for researchers that might be the VINCI server RbO2, or for operations users it might be the 801. Find your correct server. And then within the Object Explorer inside Microsoft SQL Server Management Studio, you will want to expand databases, then scroll down until you see CDWWork. Press the plus sign to the left of CDWWork to expand out its options. And then you will want to press the plus sign next to Views because all of us end users are using views rather than the actual tables, which preserves the integrity of the data in the tables and allows us to see a view of the table without disturbing the original data. And it works just the same as seeing the table from the point of view of end users as we discussed in previous parts of the series.

So once you’re in these views, all of the views are listed in alphabetical order. So we will need to do a bit of scrolling to get from A to M where we will find our metadata views. And we can see that there’s a whole set of them. And they contain a variety of information, and we’ll use several of these as we go through the demonstration in this talk.

So let’s use these meta views first to look for a description of a specific CDW column that you may want to use in your research. So we’re going to pick the meta view that is called Meta.DWViewField, and what the DW is is data warehouse, and it’s saying view for the views because we used view rather than table, and then field, meaning that it’s going to drill down to the level of the column information or the field information. As you can see to the left, there is a whole long set of fields or columns found in the view MetaDWViewField. Now we can’t see them all on the screen because there were too many. But it will give you a sense that you may not exactly be using every single piece of information found at this table at any given time, but you might want to be familiar enough with them to be able to choose which ones will help you on any given search you are trying to perform. So you’ll see that right away I'm going to reduce the number of these columns when I start trying to demonstrate the possible uses of DWViewField.

So one of the quickest things to do, as we discussed in previous Cyberseminars, would be to right click on the view of choice, Meta.DWViewField, to open up a dropdown menu that looks like the one you see here, and then get an automatic query using select top 1,000 rows from that menu. So that’s what I did, and inside SQL Server Management Studio this automatic query that you see on the top right resulted, and I removed a whole bunch of column names to only have those column names that I was particularly interested in for this demonstration. So that will tell me the schema for the view; the name of the view; the field name; a technical description if one exists for that particular field or column; codes that go with the values found in it, so if it has values one, two, and three, what does one, two, and three mean; source field description, which tells me the description of that field that came from the source data, and in most cases that’s VistA. And then I went ahead and I added a where statement to this little automatic query where the field name is such and such, whatever field or column I happen to be using in CDW and wanted to know about. That is what I will be filling in in this line of codes.

So let’s say I was using the column page in the Corporate Data Warehouse and I wanted to go and see what the definition was. Now this may seem like a no-brainer, right? And oftentimes I find myself tempted not to look at the metadata to see what something means because I feel like it must be obvious, but you’ll see why it’s kind of always a good practice to take a moment and look at your metadata. So I ran my little automatic top 1,000 query for DWViewField and I added my where statement, where my DWViewFieldName is equal to age. And I got the output that you see on the bottom right.

So the first thing I notice as I look down the first couple of columns for the schema and the ViewName is that I have columns called age for patient information, but I also have a column called age for staff information. So first off, was I looking for patient information or was I looking for staff information? So in this example I'm going to use patient information. But you will also notice that even for patient alone there are four rows of metadata. Now some of them are telling you about the table Patient.Patient and some are telling you about SPatient.SPatient. And you’ll notice that if you look at the column called ViewVersion, you can see why we end up with so many. So Patient.Patient has metadata for the version of the view that is number 63rd version of the view and a 178th version of the view. And the same thing happens with SPatients.

So ultimately it’s giving me quite a bit more to think about than I initially would have thought it would. When I take this metadata and I extract it into a table just for the purpose of our presentation today, I can get a good look at what it’s telling me. So it’s saying to me that age can be found, age for patients that is, can be found in two tables that we already mentioned. It is also telling me that there’s a technical description being provided by the architect of the CDW, and it’s coming in the form of SQL code. And so now I have to interpret what this means. And really what it’s saying is that the age may contain either a patient’s age as of today or the patient’s age at the time of death if there is a date of death noted in the file. So the interpretation of age can be two very different things. It doesn’t, of course, give me any field codes, and it doesn’t have a source field description coming from VistA because it was calculated inside the warehouse with that formula.

Well, what if I put in a column called patient period of service? In this case, I would find that it is also in two different views, OutpatVisit and OutpatWorkload. It gives me a technical description. But from the point of view of the end user, I'm not really finding my exact definition, which is what I was after, so I'm going to keep going. It had this note in field codes, but as an end user, I'm not really quite totally clear on what that means. And finally in the column called Source Field Description, I see the description that I have been seeking, and this is telling me about the period of service which best classifies this applicant. And whenever you’re looking at VistA data or thinking about the process of bringing people into care in the VA, you’ll note that a lot of the metadata will often call the patient the applicant because at some point they applied for care, and it wasn’t clear whether or not they were eligible, so at some points you’ll see them called applicants and others you’ll see them called patients. So I’ll know to think of that as also patient.

Another column that I'm going to give an example of here today is collection method, so you might have been using that. And you’ll find that a column of this name actually exists in three different views inside CDW, Dim.CollectionMethod, PatSub.PatientRace, and PatSub.PatientEthnicity. It doesn’t have a technical description, and it doesn’t come with any field codes. So when you read through the source field description, you'll understand that this is the name of a collection method that was used to obtain information about race or ethnicity, and that’s why you’ll find it in multiple tables having to do with race and ethnicity.

So what if we looked up kidney source? We can also find this in multiple tables, and in this case we see an example where it will tell us what field codes mean. So one means that the organ was from a live donor, or two, from a cadaver.

All right. So let’s move on to examples having to do with looking for metadata or content inside the CDW having to do with a topic we’re interested in. So we would search for key words inside the metadata views. And so in this case I'm still going to go ahead and use the metadata view for DWViewField because I would like to get that level of granularity. I want information about specific columns that exist in the CDW that may be helpful for me when I'm trying to find information on a topic. So I might search for that key word in a number of places. I might search for it in the name of the field, the name of the view, the technical description because as you’ve seen sometimes those can be really handy, field codes because it might be a value, or in the source descriptions because it might just be part of the description of the field that came from VistA or whatever other source might be pulling into the warehouse.

So I'm going to go back to that DWViewField, and I'm going to get my automatic query, and I'm going to need a where statement that allows me to look for my topic of interest. I'm going to have to write that line of code. So with all my different places that I suspect I should look in mind, I'm going to write a where statement that looks something like this where I might be interested in searching for the phrase OEF for Operation Enduring Freedom. And so I might look in the view name as I mentioned above or the field name or the technical description or the field codes or the source description. And so I would just keep looking all over for that. And of course you might have other search terms other than just OEF. But for the purposes of this demonstration, I was trying to keep it pretty straightforward.

So I got my automatic query inside SQL Server Management Studio, and you can see that I've selected certain fields from DWViewField instead of all the 30-plus that exist, and I added that where statement that we talked about in the previous slide. And what I'm learning here is that the most likely place for me to find information on this topic is in the table called PatSub OEFOIFService. And I can see that it’s going to give me a variety of columns related to this topic that is just in this table. And I could search through a bunch of field descriptions to try and understand what options I have inside my table or my view that I’ll use in CDW.

Okay, so let’s go to another example. Let’s imagine a scenario because I do get a lot of people who have experience in a variety of ways, either with CPRS or VistA, and they’re wondering whether that kind of content that they are aware of that exists out there in CPRS or in VistA is making its way into the Corporate Data Warehouse. So you can sort of search in a kind of reverse engineered way to find out whether or not that content is now part of the warehouse. So most often people who are using BISL may have been searching inside of the metadata for VistA itself, which is stored at the Data Architecture Repository. If you’re not familiar with that, this is a site that serves as a repository for metadata from a variety of places across the VA including, of course, VistA. And as I've been mentioning throughout this talk so far, VistA is one of the primary sources of CDW, so a lot of people are wondering which parts of VistA are now also part of the warehouse.

So referencing these files inside the Data Architecture Repository can also really, really help you understand the content of data you are seeing inside the CDW. And that’s just the kind of extra point that a lot of times when I'm in the Corporate Data Warehouse trying to use the data, the metadata, either searched manually or this way that we’re talking about today with QUERI, can often be a little bit opaque. And one of the ways I try to clear it up is to go to the Data Architecture Repository and look at metadata for VistA because it has a kind of context to it in its original place that can sometimes help me clear up my understanding of the content in the CDW. So this flows both ways.

But let’s say you’re here in the Data Architecture Repository looking at some metadata. You see something of interest, right? So the way to search in general in this report would be to start from the Data Architecture Repository homepage. You see the URL here at the bottom of the slide. Hover over VHA in the left-hand menu, and it will create the dropdown menu you see just to the right. VistA is the third option down on that dropdown menu, so you would select it, and you can search for specific kinds of information inside VistA’s metadata. So you would enter your search term in the general search box and then press that search VistA button to the right of it. Enter doesn’t work. You have to press the search VistA button.

So let’s say we searched for eligibility in our top box up there on the left. We would see lots of options come out as matches, so what you’re seeing in these matches are things, it’s telling you on the far left this is a file in VistA, its name is eligibility/benefit, and the number that goes with it is 2.322. And then you see one after that for eligibility 21.01. You see lots of great options in here, all potentially interesting matches to your topic.

So the one that I'm going to search for in CDW has the purple star by it near the bottom, and that’s a file called eligibility code, and its number is number eight inside the VistA system. So how do I search for this in the CDW?

Okay, just to give you a sense, I would have spent some time making sure I was interested in this file at CDW. So you can see the file’s content in terms of a list of fields, all their numbers, descriptions of all these fields, and help text inside the VistA metadata. And I would have said, yes, this is the one I'm hoping to find in the warehouse.

We would once again be going to the DWViewField meta view, and we would select the columns of interest. And this time I'm selecting a few more that I hadn’t selected before because I have different kinds of information. In this case, I have information about what the source entity name is, which is eligibility code. Right? Because that was the name of the file in VistA, and I also have source file number, which was number eight, so I could search these things. That’s what I'm trying to make that connection on. I still have the same things I searched for in the previous slides like the name of the schema at the top, the name, the view, the view version because we saw with patients sometimes there’s multiple versions in the metadata, the name of the field that might come out because I want to see what they are in CDW, any technical descriptions in case CDW made any kinds of notation about the work that they did, so on and so forth.

And so I added my where statement this time to search for CDW data where the source entity, or in other words the VistA file, was named eligibility code, and then I'm going to see whether or not it’s in the CDW when I run my query, and it is. So I can see that the output in SQL Server Management Studio is telling me that my VistA file eligibility code number eight can be found in the CDW table Dim.Eligibility. It’s in ViewVersion 20, and then just to the right of that I can see the name of all the different fields that I can find in here, and keep going right and I can see descriptions for these fields.

So let’s move on to the next topic for today’s discussion. How can you find information on the meaning of the linking keys that you see in CDW and that you use to join your tables together, because a lot of the times people are using these keys, but there are a lot of them, and the naming centers may seem to mean one thing, but it’s always really good if you know to check on the actual documentation to try and verify that the way we’re reading the name of that key is actually accurate and it is what we think it is.

Okay, so there’s two possible ways to think about this or two possible entry points depending on what you’re starting with. So on the one hand you might be in VistA looking at VistA’s metadata in the Data Architecture Repository and see what’s called a pointer field. Now these pointer fields often become linking keys inside CDW, and so they can tell you how you find information about this linking key, what did it become inside the CDW, so let’s do that example first.

So let’s say we’re out here in the Data Architecture Repository, and we’re looking at a VistA file called Patient Movement that has the number 405. You might see, if you look around in here, a bunch of pointers inside VistA, and one may catch your interest. So in this case we’re going to use the field in VistA called Ward Location that has the number .06, and it points out to tell you the ward to which this patient is being admitted or transferred. And you can see that VistA is calling this a pointer, and that’s telling me that I expect it to be a linking key inside the CDW.

So inside the CDW, the way that this is referred to is in the RelatedFileNumber and the RelatedFieldNumber. So we had file number 405 when we were looking at it in VistA, and we had field number .06 when we were looking at it in VistA. And whenever they make the linking key inside CDW, they would record this information in these metadata fields called RelatedFileNumber and RelatedFieldNumber because the linking key is not directly taken from VistA. It is related to the pointer in VistA, and so that’s how that got that name.

So we run this query, and we can see that we find a match because when we look at the related file number and field number, we see the 405 and the .06, and so we’ve found our pointer from VistA. And we can see when we look to the left where it is in the CDW and what it’s called. So we can see that actually this pointer can be found in Inpat Inpatient, Inpat Lodger, Inpat PatientTransfer, Inpat ProvisionalMovement. And then as you scroll right, you can see the various fields, and these are column names that it has. And then all of CDW, as you may recall from previous talks, the linking keys all end with the same three letters, SID. That stands for surrogate identifier, and they are created inside the warehouse. It’s part of what makes these production domains as part of the work that got done on them to make them a bit easier to use. So we can remember from our VistA search what the definition of this set of linking keys is, right? This is the ward to which the patient was admitted or transferred.

So let’s take it from the second entry point. Let’s say you are in the CDW, either using the data or looking at the metadata or looking at some documentation where you see this linking key, and you want to know what it is defined as. Okay, so in this case I'm going to take us on the manual search on the SharePoint site for CDW and pretend that we are looking at metadata for the inpatient 3.0 domain. We’ll click that plus sign to the left of the domain name to expand the list of tables.

Let’s say we got interested in the view called Inpat PatientTransfer, and we were particularly, of course, wanting to connect this transfer information to other information about the patient’s stay or the patient themselves or the staff members that provided the care. And so we would look to the far right and click on that link that shows you the linking keys. It shows you the relationship of this table to other tables in the warehouse. And by clicking on that link to the far right, we would open up the table of foreign keys and find particularly any key that we may want to know more about. So in this case I'm using AttendingPhysicianStaffSID, which is a foreign key in the Inpat PatientTransfer table that connects up staff, and I want to know the exact definition of this just to be sure I understand it. How do I find that?

So in order to write a query to do this, we’re going to need to use two of the meta views to help us. We’ll use Meta.DWViewField, the one we’ve been using through most of the talk, but we’ll also MetaFileManField. So this MetaFileManField is using this place, FileMan, which is a term indigenous to VistA, and it, in my layman’s terms, basically describes something to do with the control of the files in the VistA system. So basically what this is is information that is much closer to the information we’ve been looking at when we look at the Data Architecture Repository and we read the metadata from VistA. So this is going to have a lot more of a direct line to that metadata from VistA. And that’s why we’re going here.

While we get in here, I'm going to select some of these columns from the MetaFileManField view because, once again, the list is quite long. I might want to make some specific choices that will be most helpful for me in my query.

So from the original meta view that we’ve been using, the slide on DWViewField, I'm picking some familiar columns, the ViewName, the FieldName, the FileNumber, the FieldNumber. From the FileManField, right? The one that’s closer to VistA, I want to get the field names, but I also want to get a description because that’s what I'm here for. What does this linking key mean? Well, VistA is going to be the one with the definition. Simultaneously to write this query, remember I'm bringing two meta views together, and I'm joining several of their fields to make sure that I get the information linked there specifically. So I'm going to connect on the file number from VistA, which remember will be translated to the RelatedFileNumber in the CDW, and I'm going to connect on the FieldNumber from VistA, which will be the RelatedFieldNumber in CDW, and then just the linking key between the DWViewField and FileMan, which is SourceEntitySID inside the CDW views. And my search terms are going to be from what I was looking at in the CDW, the column called AttendingPhysicianStaffSID, which was my foreign key, and my view name, which was PatientTransfer from Inpat PatientTransfer.

So taking a look at these results, I would run this query inside SQL Server, I would get my one row of information back, and inside there I would see the description that would have come from the Data Architecture Repository from VistA metadata. And as I sort of made this bigger so we could read it, I can see that this is the supervising physician who is responsible for the care of patient. Then there’s more information that follows that.

So to our last topic, we’re going to take a quick look at how we can use metadata both on the SharePoint site and in a query to find columns that we can include in any query we are going to write about CDW that will hopefully help improve efficiency of the query and the speed with which you can find all your answers.

So the metadata view we’ll use for this is called Meta.DW, for warehouse again, Index. And you can see that I have located it in the Object Explorer on the far left of our screenshot here. And then I ran a little automatic query for the top 1,000 again, and I added a where statement again where I'm going to just look for information from the schema Outpat, which means I'm looking for information from the outpatient domain. And then I'm going to turn my attention to the results to look at those columns that are listed under IndexColumns, and these are the ones that are telling me use these columns to help you run your query more efficiently.

So what do we mean by this index term? Well, from just a quick Google search, you can get a basic answer, right? So an index is the kind of column in SQL that will help SQL Server process your request by finding the row or rows of values that you’re asking for in your query more quickly and efficiently. Sometimes these indices are clustered, which means that they will then go an extra step of sorting those data rows in a table, or a view, based on their values, which should make that process even faster if it’s clustered.

Also, you may want to pay attention when you’re inside the SharePoint site and you’re scrolling through the various metadata related to all those domains, they have a really important column here. It’s called Relevant Dates. And when you look at it, it always has information, at least on large fact tables, about partition keys. And it will tell you which one of the columns are acting as partition keys that can help your query find much more quickly and much more suitably against large fact tables in the CDW.

So what do we mean by partition keys? Partition is a way of dividing a table, a large table into smaller, more management parts without having to create a whole separate table for each part. So inside the CDW, they’re using this tool, right, to partition to the data into, sort groups of rows. And then whenever you use that key to sort of pull out a piece of data that you’re interested in, you will be able to find it much faster like an index in the back of the book telling you which page to go to and that sort of thing. So that’s what these keys are doing. And if you can use the metadata to identify them, then you can write much, much faster queries.

So I’ll take any questions.

Moderator: Hi. Thanks Margaret. We do have several questions here from the audience. I’ll just get started from the top. How often are descriptions in the metadata views updated? For example, in the age field example that you used, the description claims that the field date of birth was used, but that was renamed to birth date time months ago.

Dr. Margaret Gonsoulin: Right. So I may have, well, my example may be out of date. So I think it’s updated when they pull. It’s a little bit fuzzy to me because I think when they do work on the domain itself, for this database architect that is, they are pulling that metadata that goes with it as they’re building the domains. And so then also as they edit the domains, I suspect they’re pulling new ones, so it may be my fault that that’s named older because I may not have re-run that query right before this. I’ve been working on this talk a little while, and then it got delayed in being presented, so it may very well be updated at this point. I'm not quite sure. But it’s a little bit hard for me to know how continuous that work is behind the scenes or is it two separate processes, is it one process? But generally speaking as they build it they’re bringing in some metadata that goes with the data they built, and I think as they edit it they reload those metadata. I’m not behind the scenes on that.

Moderator: All right. Thanks Margaret. The next question: How do we know what the current version of a view is or should we just look at the record associated with the highest numbered view version?

Dr. Margaret Gonsoulin: Yes. I mean it goes in order, so as far as I understand it, the highest number is the most current, and so I think like the reason we’re seeing multiple versions for something like patient inside the metadata is because I think they probably left old and new versions of the metadata inside the metadata report for longer because although they did move patient 3.0, patient is so central to every single query people write. There are probably offices out there that are struggling to rewrite tons and tons of codes because of the changes to patient that they probably left it out there for longer, the older version of the view is in the metadata for longer for that purpose is, would be my guess.

Moderator: All right. Thank you. This next question: I would like to search CDW for VistA configurations such as user configurations on alerts, reminders, options selected when records were being created in FileMan. Is this possible?

Dr. Margaret Gonsoulin: I think that might require a bit more of a conversation. I'm not quite clear on what you mean by configuration. So to that requester out there, if you wouldn’t mind maybe a help desk question, we could have a call and talk a little bit more about what that means, and then I might be able to be more specific about it.

Moderator: All right. Thanks Margaret. To the audience, we still have about 13 minutes left in the presentation, so if you have any questions you would like to ask, please send those in now. This next one is a comment. Yesterday in SQL Office Hours we found the new version wouldn’t allow percent at the beginning and end of a like statement.

Dr. Margaret Gonsoulin: Like the new version of SQL Server?

Moderator: It doesn’t specify. Yes. The person just answered in yes.

Dr. Margaret Gonsoulin: Ah, the newest then, right? So if there are different [unintelligible 47:35], so I don’t know about the new because I don’t have it, but I assume that will be coming soon, so thank you. We’ll adjust accordingly as soon as possible.

Moderator: All right. Thank you Margaret. This person also said could that have been only in searching tables and not metadata?

Dr. Margaret Gonsoulin: Yeah. I mean it strikes me as odd that that would be a feature that would be removed wholly from the capability of SQL Server as a piece of software, so I'm wondering if there might not be more to it. And I know that searches, when you do searches on text fields and you use these wildcards and if people apply them to the wrong kind of table such a very large [unintelligible 48:38] notes or something like that, it can really take down the whole server for how much processing time it takes and that kind of thing. Computing power is not big enough for everybody to be doing that on large tables. So I'm wondering if it’s some kind of a restriction in relationship to large tables. That’s possible. And in this case a small table maybe you would be able to keep using it, but I guess I should have gone to SQL Office Hours yesterday, but I was too busy.

Moderator: Okay. You may have answered this question already, but someone else asked. How to make sure that I am using the correct view version for the given field when selecting the field column from the actual view?

Dr. Margaret Gonsoulin: So I mean I think in general that inside CDWWork and/or recently provisioned data from VINCI you would almost always be using the later version, but things do get a little bit tricky in there at times when they’re building or they are editing a table for particular reasons. Sometimes I do kind of notice a little bit of a stopgap like you’re really dealing with the earlier version of the table because the new version of the table is actually almost ready for release but hasn’t been ready for release, so there’s like a few days where it’s a little bit off. And so I mean oftentimes you just kind of say run against some column, and you realize it’s not in the metadata because the metadata hasn’t been updated or because you see it in the metadata because it’s about to be released to CDW production, but there’s a few more days when it’s [unintelligible 50:53] or something like that. And so you can’t actually see that column in the actual data, but most of the time even if it’s a little bit out of sync for the data that you are seeing both actually in the view of the data you’re using and also in the metadata themselves, the definition will still be valid regardless of the version of the view. But I do know what you mean. It’s just sometimes there is a little bit of a gap or a little bit of a time lag, a little bit of not being lined up quite right. But I still think most of the time it doesn’t present a huge issue in terms of accuracy of definitions and being able to understand what a column means. So that’s a little bit of a comfort.

Moderator: All right. Thank you Margaret. I’m seeing a lot of questions here from people about where you can access a copy of the handouts. To everyone who registered for this session, you should have gotten an e-mail before, and it has a link to the handouts in there. You will also be able to view the archived presentation once it’s on HSR&D’s Cyberseminar archives.

All right. Another question for you, Margaret. Should we link foreign keys in two tables which come from the common dim tables as primary key?

Dr. Margaret Gonsoulin: I'm sorry. Could you repeat that one?

Moderator: Sure. Should we link foreign keys in two tables which come from the common dim table as primary key?

Dr. Margaret Gonsoulin: Hmm. I’m not quite sure I understand the question. I mean I think like if you’re talking about having sort of two different foreign keys that will connect to the same primary key in the dim table, you would ask the question, then you would use the foreign key that makes sense for your question at the moment, if that’s what you’re asking. So like inside the inpatient data, you can find many foreign keys to the staff table, but one might be about a staff member that entered information. Another one might be [unintelligible 53:44] attending, or another one might be related to a specific encounter or action that got taken during that, and it just, there are a bunch of them, and they all go to the same primary key out in the staff table or out to like a dim table like Ward because it may be that this was the ward that they were admitted to or it may be just a different foreign key and different table, the ward they were transferred to or the ward they were discharged from, and it all goes to Dim.WardLocation. But you would use the one that you need in the moment based on the definition, if that’s what you were asking.

Moderator: All right. Thanks Margaret. I’ll see if that person sends in any clarifying information. I still have a couple more questions for you, and I think we have enough time to get through them. Who do we inquire about when a particular field will be added?

Dr. Margaret Gonsoulin: So there are two different things going on. I think in general there is a way to tell about the basic work that’s going on in terms of adding content to the CDW, and I’ll sort of navigate to it now just to show you where it is. So inside today’s talk, you have the link to the CDW SharePoint site or the Business Intelligence Service Line main page here. And you can see to the right-hand side here a link called CDW Domain State of [unintelligible 55:45] Priority. It’s in an Excel workbook. When you open it, it will summarize the work that is happening inside the CDW in terms of architect work of building content. And so if you look under column E here to the right, you can see that program integrity tool is being created right now because the green dot is on create table. And then on the second line you can see that the Master Veteran Index is with the ETL team right now, [unintelligible 56:27], and so on and so forth. So in general, in terms of new content that’s going in, you can find information like this summarized for you here on the SharePoint site. Also, I am a member of a data subcommittee for Capability Management Board that tries to hear requests. And the heads of that committee are Mark Love and Betsy Lancaster, and they hear requests for new content and sort of keep track of those requests. So you can always check with Mark Love or Betsy Lancaster for more information about new content going on or decisions that are being made behind the scenes. And I think those are the two primary ways that I'm aware of figuring out what’s coming with the CDW.

Moderator: Okay. Thank you. Last question: Oh wait, follow-up to that last one. Did you mention who you would e-mail to put in formal requests?

Dr. Margaret Gonsoulin: Mark Love and Betsy Lancaster, the\_

Moderator: Okay.

Dr. Margaret Gonsoulin: Yeah. The chairs of the data subcommittee for the Capability Management Board. Their job is to hear requests.

Moderator: All right. Thank you. Is there repository of SQL code for common types of searches?

Dr. Margaret Gonsoulin: Yes, there are actually in the BISL website for sure. I think they’re probably all sorts of places where people are doing this. Trying to remember where it’s at. I always have a little bit of trouble finding this, but I know it’s here on the SharePoint site. I feel like it’s in training somewhere. If I can’t find it really quickly, then I’ll poke around until I do. If you want to e-mail me, I can sort of locate it at that point. But I know they keep, ah here, script library under BISL resources. So BISL is doing it, but I think some regional data warehouses are also doing it and some COINs, like if I'm not mistaken I think Seattle’s COIN does, at Puget Sound I guess I should say. I think they might be doing a little bit of that. I feel like I heard Palo Alto COIN was doing a little bit of that, so I think it’s in bits here and there, but this is a central location that you could get to here in the script library. They have some great stuff in here.

Moderator: All right. Margaret, thank you so much for taking the time to present today’s session and answering everyone’s questions. We did get through most of them. To the audience, if your questions were not addressed during the presentation, you can contact the VIReC help desk at virec@va.gov. Please tune in for the next VIReC session. This session will be on Monday, June 4th, at 1 p.m. Eastern for VIReC’s Database and Methods Cyberseminar series. Dr. Susan Hasting, Elizabeth Mahanna, and Daniel Denhalter will be here to present on Working with EHR Data Using VistAWeb, VINCI ChartReview Tool, and Joint Legacy Viewer. We hope to see you there.

Heidi, can I turn it over to you?

Heidi: Sure. Thanks Hira. Margaret, I also want to thank you for presenting today. We really do appreciate the time you put into it. For the audience, I will be closing the meeting out in just a moment. When I do, you will be prompted with a feedback form. We appreciate you taking a couple moments to fill that out.

Thank you everyone for joining us for today’s HSR&D Cyberseminar, and we look forward to seeing you at a future session. Thank you.

[ END OF AUDIO ]