Lesson 7 Transcript: Data Movement

Slide 1: Cover
Welcome to Lesson 7 of the DB2 on Campus Lecture Series. Today we are going to talk about data movement. My name is Raul Chong, and I'm the DB2 on Campus program manager.

Slide 2: Agenda
This is the agenda for today.

Slide 3: Agenda
We will first have a look at data movement overview.

Slide 4: Data Movement Tools
So this slide shows the data movement tools. Some of them, we will cover some of those on the next few slides. So here, on the left side we have database A, on the right side database B. And then we can use the Export utility to export from a table in database A into a file. The file can have any of these formats: ASC stands for ASCII, DEL is delimited ASCII, so it's ASCII, but we use a deliminator; WSF is for Work-Sheet Format, normally used for Excel or Lotus-1-2-3; IXF stands for IBM eXchange Format, and this format is specific to IBM, but the nice thing of using this format IXF is that it includes the DDL of the table. So that means it will include the create table statement as part of the file. The other ones, on the other hand, do not include the DDL, but they are more standard, so you could move information from a DB2 table into these ASCII, delimited ASCII or WSF files and then you can move that, or you can import that later into another software product, if you need to.

Now, once you have stored in the file the information from a given table, you can do the opposite, and use the Import utility to read from a file and load it into a table in the database, in this case, database B. It doesn't have to be another database; it could be the same database A, into another table. But in this particular example we are inserting or we are importing into another table in database B. Now, if you are using format IXF, there is no need to create a table ahead of time because with IXF, the table will be created, since the DDL is exported in the file, the input file. But if you are using another format, like ASCII, delimited ASCII or WSF, then you do need to create manually the table on the target location ahead of time. Now, we talked about Import, but we also have another utility which is Load. A load is the same as import, but it's faster. And there's another tool called Set Integrity. We will talk more about these two tools or commands in the next few slides.

Slide 5: Data Movement Tools
So, in this slide we talk about export, import and load, we talk about how they can move from one table into a file and from one file into a table. Now, export, import and load are utilities that work at the table level. So it's per table. If you want to execute these on a bunch of tables, you could either create a script, or you can use a utility called **db2move**. So db2move is used

to work on several tables, and as mentioned on a previous lesson, for example, on the previous lesson you could use db2move and specify a given schema, so that you could move or export a given number of tables based on the schema, and then maybe you can import them to another database. So, you can use db2move with export, import and load, and it only works with the IXF format. So let me show you quickly... actually I’ll show you this in more detail in a few moments. And then we have another tool, called **db2look**. db2look is used to extract information like the DDL, permissions, database statistics and so on. So it basically allows me to maybe clone the structure of the database. We will provide an example in the next slides.

Slide 6: Agenda
So let's start with Export.

Slide 7: EXPORT
So the syntax for export is... there are many many choices but I'm just providing here one simple example. Now, you can think of export as doing a select behind the scenes, an SQL select, behind the scenes.

Demo
So, what we have here is that we could, for example, copy/paste the export utility here into notepad so it's easier to execute. So I have here the Command Editor. So I'm already connected to the SAMPLE database, and I'm just going to execute this export that I show you in power point. And when I execute it, I will have this output here that says this export utility finished exporting rows, etc., and the number of rows that was exported was 10. So it's successfully exported 10 rows to this file, employee.ixf. If you want to know where this file employee.ixf is, you could use an exclamation mark within the Command Editor, which is going to allow me to escape to the operating system, and if I issue a **dir** from here, and press Execute, then I will be able to see the directory information here. And in here, there should be the file that we just created. So, let's look for it here, employee.ixf. So, that file was created in the directory, which is here, right? Listed here. Now the IXF format doesn't allow you ... it's sort of binary, but you could sort of see some of the information, but it's not like a text file where you can see all the information there. But you may be able to read some.

Back to Slide 7: EXPORT
So basically we have selected 10 rows from employee table and store everything to that file.

Slide 8: Launching the EXPORT Table Dialog
Now, let's take a look here. You can also run the export utility by choosing a table, as shown in this chart, in this case the employee table, right clicking and choosing Export, and then just follow the wizard. It's fairly easy to do, so I will not cover this in detail.

Slide 9: Agenda
Then you have the Import utility.

Slide 10: IMPORT

The import utility: basically what it does behind the scenes is it's running an SQL insert. So, in this particular example, again, I'm not showing you all the possible cases, all the different syntax. I'm just showing you a simpler example here. But an SQL insert will activate triggers, will check all constraints, using the BUFFERPOOL. And in this case, we're importing from this file that we created before, of this format, which is IXF. And we are going to load this information into the table called employee_copy. It's another table. And we are using an option called REPLACE_CREATE. There are different options, but we are using the REPLACE_CREATE option, which means that if the file employee.... sorry, the table employee_copy already exists, I'm going to replace all of the contents. If the table doesn't exist, then we are going to create this table, and we are going to import and start loading the data.

Demo

Ok, so let me run this import, and let me first copy the syntax or the statement from my Notepad here. And I'm going to clear the results here, and let me just run import. And I'm going to execute. And then after I execute, I get the result that the number of rows inserted was 10. So that's good news. Things seem to be working well. And then what I'm going to do is I'm going to verify information that was stored in employee_copy. So I do **select * from employee_copy**. And then there should be 10 rows, because these were the 10 rows that were first exported into the employee.ixf file. Then we imported them into this new table called employee_copy, and these are the 10 rows that I have; so things are ok here.

Slide 11: Launching the IMPORT Dialog

If you want to use IMPORT from the Control Center, you can just choose any table, then choose Import, that is, right-click and choose Import, and then just follow the wizard.

Slide 12: Agenda

Ok, in the next section we talk about Load and Set Integrity.

Slide 13: LOAD

Load is the same thing as import, but it 's a lot faster and it would not check for constraints, or it will not activate triggers, and so on. So, it's because of this that it's going to be faster.

Demo

Now, let me just quickly show this to you. So, let's say what I have here is my DB2 engine. So this says this is my DB2 engine. And what I have over here is my database, stored in disk. And now, if I run just an import statement, what I'm doing behind the scene is actually doing an SQL insert, right? So, because I'm doing an SQL insert, I'm basically going to go through the engine, and I will be checking for triggers, checking for constraints, etc. etc. And once that is done, I will go and insert information into pages in my database. So I will start inserting information into pages. Now, what happens with the load? If I issue a load statement, what's going to happen is most of the things will be not checked in terms of triggers; triggers will not be activated, constraints will not be checked because load would normally go and try to escape the DB2 engine and just put information directly on the pages. But that means there may be some problems in terms of data integrity. So, if you want to use load, you probably should know that your table should be checked, or there should be integrating of your tables. And if you want this as well, you have the option to run the other tool, or the other command, which is **SET INTEGRITY**. So you can run this other tool after you run load to make sure that there's integrity in your data.

Slide 14: Launching the LOAD Dialog

Ok, you can try these commands yourself, and we are going to move on here. If you want to execute the **load** command from the Control Center, you can right-click on a given table and choose Load, and then just follow the load wizard.

Slide 15: Launching the SET INTEGRITY Dialog

The same thing with SET INTEGRITY, right-click on a given table, and choose Set Integrity and follow the wizard.

Slide 16: Agenda

Ok, so now we move on to the next section, which is **db2move**.

Slide 17: db2move

db2move, as I said before, allows you to either export, import or load a bunch of tables in one command. Because, if you just run export, import or load on its own, it's per table, but using db2move you can do it for a bunch of tables. Now here I'm providing a simpler example where I'm trying to move all the tables from the SAMPLE database. So I'm trying to export them.

Demo

So, what's going to happen is I will be generating many names for the given files. So let's start a command window. And if you are running on Linux, you can just work on the Linux shell. So, first I'm going to make a directory here, and I'm going to call it erase, because I'm going to erase it afterward. I'm going to change the directory, and then what I'm going to do is run the command that was shown before, so it's **db2move sample export**. The reason I created and changed to the erase directory is because this would generate a bunch of files and I don't want them to be combined with other existing files. So, if I press Enter to execute, it will start executing the **db2move** command or utility, and what it's doing is exporting all the tables from the sample database into files. Now, if I issue a **dir** command here, I will see a bunch of files, and these files are the corresponding ones to the tables that I was exporting. So, each of these files contain table information. If I want to take a look at the mapping between the tables and the files, I can take a look at these files called **db2move.lst**.

So if I open this file, which is just a text file, I will see that, for example, table AFCHONG.ACT maps to, or has been stored in tab1.ixf. And this one, the ones that end with msg, is a message file, so it's just more information about the given export for that particular table. For this table, ADVISE_INDEX, the data will be stored in tab2.ixf, and so on and so on.

And this same file will be used later by db2move, if you use db2move with the import option or the load option. Ok, so that was just a demonstration on db2move.

Slide 18: Agenda
And we move on to the next section, which is **db2look**.

Slide 19: db2look
So **db2look** allows me to extract all the DDL that I specify. There are many many options and I'm not going to put all the options here. But with db2look you can basically create a clone of the database in the sense you can get all the structure of the database. It will not get the data, but just the structure. Other things you can do with db2look is that you could use the mimic option, which is **-m**, when you want to, maybe, recreate a performance problem. So for example, you may get a customer who calls and says, oh, I have 10 million rows and I'm running this SQL and now it's running very slow. In the past, maybe, just a week ago, it was running very fast, but right now it's running very slow, and then I don't know what the problem is. So if you want to re-create the problem, let's say you are in a customer support, technical support, first of all, it would be hard to get 10 million rows from the customer, not only because it's hard to send them through the network, but it's also private information. So what you could do is ask the customer to run db2look with the mimic option, which is **-m**. That would generate a text file with a lot of update statements. These are updates on the statistics catalog tables. Then you can run db2look on your own computer and what this will do is it will simulate the environment that is at your customer site.

So even though you don't have 10 million rows, the statistics will say that you have 10 million rows, and when you try to recreate the problem by issuing the same SQL as your customer, the SQL will think that you have 10 million rows, because the engine in DB2, or the optimizer, which is the brain, let's say, of the DB2 to access data, will think it's a brain, or it's an engine, or it's an optimizer that will look at the statistics in your catalog tables, and based on the information there, it will determine which is the best access plan. So, you don't have to have the 10 million rows just because the statistics say that you have 10 million rows, DB2 will assume that there are 10 million rows. And it will try and create the best access plan based on these 10 million rows, and it will obviously not return anything back because there are no rows at all, but you could get a Visual Explain graph on this particular SQL. Visual Explain was briefly introduced when we are talking about tools, and when we are talking about the Command Editor. OK, so we can try and run this db2look, and what it's doing is getting information from the SAMPLE database. Getting the table information, extracting the DDL, and it's going to store the output in the file called sample.ddl.

Demo
So after we have executed this command, we are going to take a look at that file. So let's copy/paste from here. I'm going to execute **db2look** from a DB2 command window. First, let's change to the erase directory. And then I'm going to execute this command now. So it will start getting the DDL information, and now I'm going to take a look at the file. And the file was sample.ddl. So when I look at this file, as you can see, we will just first connect to

the SAMPLE database. It will have create statements for table spaces. It will have four other options, like sequences. And then you have create for tables, all the tables are altered to add primary keys or foreign key constraints for referential integrity, indices, you are creating indices here for another table, etc. etc. So as you can see, it's basically putting into a script file—it's a text file, which is script—all information related to this table. So you are basically getting all the structure of this database. Very well; let me just close this.

Now, when would you want to use db2look? As I said before, when you want to clone a database, but there is maybe a specific case, when you want to use it. So, for example, here, let's say you have two data servers, one is on Windows, and the other one is also on Windows. So as shown in this chart, you have the one on the left, and the one on the right. Now, let's say you want to clone a database that is on this DB2 server on Windows on the left side. Well, the easiest way to do it would be to issue a **backup** command, which I have not explained yet, or it's covered on later lessons, but let's just talk about it very quickly here. So you can run a backup. And then this backup, what it's going to do is it will create a file. Ok, so it will create a file. It's an internal binary file, and then from there I can take that file into the other computer and then I run another utility called **restore**. So I can run restore here. And then, that's it. I made a clone of my backup from one computer or from server to another server very quickly. Now, what happens, however, if the other server on the right side is not a DB2 server on Windows, but it's a DB2 server on Linux? So this is another platform. So in this particular case, I will not be able to do a backup and restore because they are in different platforms. Ok, so I cannot do this in the case where I have Windows and Linux. So, what I'll have to do is use some utility that I talked about before. So you probably have to issue first a db2look. Right?

So if you run **db2look**, you can extract all the DDL to get the structure of your database in this server, you have a DDL file here, it is a text file, it's a script file that you will move into the other computer. And then you will also run here, either open the Command Editor and execute the script, or if you are opening it from a script file, just using the command window you have to issue a command like this, right: **db2 -tvf** and the name of the file that has this DDL. So when you issue this statement from the command window, or if you open the file from the Command Editor and execute Run that will create the structure of your database on this server on Linux. But the database is empty. It just has all the definitions, it has all the tables created, all the indices created, but they are all empty.

So the next thing you need to do is you have to run a **db2move** command. And probably you will use it with export option, right? So this would generate a bunch of files. It would generate a bunch of files as I showed you before. It would generate many many files. So I'm just going to put 1, 2, 3, etc., etc., and then you would move all of these files into the other computer, and from there I want to be running the same utility, which is **db2move;** but now with the import, or maybe the load option to load all of this data of the tables into the server on Linux. So, by doing these two operations, we are basically cloning the database from DB2 server on Windows to DB2 server only on Linux. And again we have to do this because these were on different platforms.

Slide 20: db2look: Extracting DDL from Control Center

Ok, if you want to use db2look from the Control Center, you can right-click on the database that you are trying to work on, and choose Generate DDL. So it's not... you don't choose... there's no db2look option but **generate DDL**. It’s called **generate DDL**.

Slide 21: Extraction Options

And then you will have options as to what you want to extract.

Slide 22: QuickLab #8

Ok, so now I suggest that you to pause this presentation and take a look at QuickLab #8, where you can work on extracting DDL from the Express database.

Slide 23: What's Next

Ok, so now we have reached the end of this presentation of this lesson. So, congratulations for completing this Lesson 7, Data Movement. And as to what is next, it will be Lesson 8 on Security. Thank you.