

Lesson 2 Transcript: Part 2 of 2 - The DB2 Environment

Slide 1: Cover

Welcome to Lesson 2 of the DB2 on Campus lecture series. Today we're talking about the DB2 environment, and this is part 2 of 2 parts. My name is Raul Chong and I'm the DB2 on Campus Program Manager.

Slide 2: Agenda

In the first part we talked about instances, and now we're going to talk about databases, the DAS, and DB2 configuration. So let's start with databases.

Slide 3: The DB2 Environment

First of all, I want to mention that in this course we're not explaining relational theory concepts, so we're basically teaching how to work with DB2. So, we're not teaching what a database is, what a table is, what a view is, an index is, etc. etc. Now what we have here is a database called mydb1 in instance db2, and as I said before, instances are independent environments. So I could create another database called mydb1, on a different instance. There would be no problem because these are two independent environments, and if you want to access a given database from a client you can access it through the port of the instance, which is a unique number, and that's how you can uniquely identify which database you want to connect to. When you create a database, by default, there are several objects created, so there are table spaces, there is a buffer pool, and there are log files. Let's explain first what a table space is all about.

Graph demo

So for this, we're going to use a simple graph. So let's say you have two tables, table-a, and you have another table, table-b, and table-a is a hot table, so if you want table-a to use the fastest disk, let's say it's disk 1, in this chart. And then you have another disk, which is slower, which is disk 2. Over here on the right, you have your memory of RAM and this is called a buffer pool, which is cache in memory for the database. So, table-a is a hot table, so it's used a lot and you want this table to have very good response in terms of performance. So one of the things that you can do, to make sure that table-a is stored on disk 1 and uses the most amount of memory, is to first create a table space.

So we're going to create a table space, I'm just going to copy/paste from here that I had prepared it, and we're going to paste the table space right here; so it's basically a layer between your logical tables, and your physical disks, and your physical memory. So when you create a table space, let's call it xyz; let's say this is a table space xyz. When you create a table space, you have to associate this table space to the syntax. To the create table space syntax to a directory or path or device and you associate it as well to a buffer pool, which is basically as I said before, cache in memory, and then, once you have created this table space and you create a table, you can create a table, and you can say "create this table in tablespace xyz". So by doing this, you are basically relating table-a with disk1 and with this buffer pool, or area of memory. So again, a table space is just a layer, as I said before, let's go back to the presentation.

Slide 4: The DB2 Environment

Now, there are three table spaces created by default, the first one is called SYSCATSPACE. This table space is used to store the catalog, which is also known as the dictionary, or also known as the metadata, which means data about data. So basically, it contains tables that store information like how many columns there are on all the tables, how many views there are, how many indexes or what type of indexes they are, etc. etc. Every object that starts with SYS is normally something that DB2 has created, so it's a system object, and therefore, you should not manipulate this object because you may corrupt the database. If you play with this object, you may eventually cause the database to become unusable.

The second table space is called TEMPSPACE1 and it is used for example in the case you're doing a sort or a join and there's not enough space in memory so DB2 will create temporary tables in this TEMPSPACE1 table space and they will be used to complete or to continue processing the sort or join and once they are finished, those tables that were used temporarily would be deleted.

USERSPACE1 is a table space in the case where you issue a create a table space and you do not specify which table space that table will reside. Then DB2 will use USERSPACE1 as the table space. So those are the 3 table spaces created by default, and this is one of the reasons why it takes a few minutes to create a database in DB2 because DB2 has to create these table spaces and populate the tables in syscatspace.

As part of the creation of a database as well, a buffer pool is created, and as I said before, a buffer pool is cached for the database. You can create several buffer pools, but the one that is created by default is called IBMDEFAULTBP. There are also some log files that are created for recovery purposes. As part of the creation of a database as well, there is some configuration that is done using a tool called **autoconfigure** and we will talk more about this in a following lesson.

Slide 4: The DB2 Environment (right side)

Now I'm showing on the right side, whenever you create a database these are the same objects that are created. So that's why we also say that databases are independent between themselves, because each of them have these objects and the first one which is SYSCATSPACE is the table space that contains the catalog or the dictionary, also known as the metadata, which is data about data. So if you delete SYSCATSPACE, you're pretty much corrupting the database, or making it unusable. Anything that starts with SYS, like in SYSCATSPACE is created by the system and normally you should not mess with those objects.

Now, once I have my own database, I can create my own table space, called, in this case, mytbls1, and in there, I can create my tables, my views, my indexes, etc. etc. Now in this example here, I'm showing that I can also create another database within the same instance, so, it's not like in Oracle, for example, where you can only create one database per instance, there's a one-to-one mapping, in DB2 you can create many database for (one) instance. And in this case, SAMPLE, I'm not showing you that it has also SYSCATSPACE, TEMPSPACE1, USERSPACE1, etc. etc. because I just didn't have space, but SAMPLE would also have those five objects mentioned before, because as I said before, again, databases are also independent between each other.

Now if databases are independent between each other that means I can create a table space in the SAMPLE database, which has the same name as the one used before, called mytbls1, so mytbls1 used in the SAMPLE database and mytbls1 used in database mydb1 will not conflict because mydb1 and sample are independent.

Slide 5: Creating and connecting to a database

To create a database, I showed you how to create a database on part 1 of this presentation, and to connect to a database, you have to use the syntax: **connect to dbalias**, the database alias, which if you don't provide the alias, it will be the same name as the database name and then you can optionally specify a user with a userid using a given password. We will show you some connection examples later on in these lessons.

Slide 6: Quicklab #2 - Creating a New Database

So now I suggest that you pause and start working with Quicklab #2, which provides an exercise that allows you to create a new database.

Slide 7: Agenda

Moving on to the next section, which is the DAS.

Slide 8: The DB2 Administration Server (DAS)

The DAS stands for the DB2 administration server, or some people call it the database administration server. This is a daemon process. You may not use it too much; it is created automatically for you when you install DB2, and on Windows it's called DB2DAS00 normally, and on Linux it's called DASUSR1. The DAS is normally used for remote GUI administration, so what that means is let's say, I'm in Toronto right now and I want to administer a database in China, so if I want to remotely administer this server, using the GUI, then the DAS must be running in the server in China, so that's the main purpose.

Slide 9: DAS commands

Now, these are some DAS commands to create and to drop the DAS in either Windows or Linux, normally you won't have to manipulate or work too much with the DAS. And to start and stop the DAS you use **db2admin start**, and **db2admin stop**.

Slide 10: DAS data server big picture

There's only one DAS per server, so even if you have many DB2 instances, or many instances, there will only be one DAS.

Slide 11: Agenda

Finally we move on to the last section of this presentation, which is DB2 configuration.

Slide 12: DB2 Configuration

There are different levels in which you can configure DB2; the first level is environment variables.

Slide 13: Environment Variables

The most important one is probably db2instance, which we discussed before, which is used to switch between the different instances in your server.

Slide 14: DB2 Configuration

The next level of configuration is called the database manager configuration file, which is configuration at the instance level, also known as the database manager.

Slide 15: Database Manager Configuration file (dbm cfg)

To work with the database manager configuration file, you can use the GUI, and we will show you how to do it in the next lesson when we talk about tools, but you can also use these commands here, and I'm going to quickly show you how to work with these commands.

Demo

So if we go to Start > Run, and open a db2cmd window, if you want to work with configuration, you can type **db2 get dbm cfg**, and that will show you all the parameters for that current instance. If you want to update the given parameters, let's say you want to update INTRA_PARALLEL using the command line, you would say **db2 update dbm cmd using intraparallel** and set it to yes. We press enter, and then the configuration is changed. So quickly we do a **dbm cfg**, we should see here the new value of INTRA_PARALLEL set to yes. Now some of these parameters are dynamic, which means you don't need to start and stop the instance again. Some of them are not dynamic, which means you do have to stop and start the instance using the **db2stop** and **db2start** command for the change to be effective.

Slide 16: DBM CFG from the Control Center

I'll show you this later when we use the Control Center, but you can do the same using the Control Center.

Slide 17: DB2 Configuration

The next level of configuration is the database configuration file or db cfg, which is configuring within the database.

Slide 18: Database Configuration file (db cfg)

And the commands are very similar.

Demo

So going to this screen, we could first connect to the database, so I **db2 connect to sample**. And then I can take a look at the configuration file by using **db2 get db cfg** for sample and then I get the result and from here I could change any parameter in a very similar way as in the other case, so for example, let's say I'm going to change the parameter logfilesz. So what I'm going to do is **update db cfg using logfilesz 2000**. And then in this case, it's warning you, so in this particular one is not a dynamic parameter, which is what I explained before.

Now in the case of databases, a dynamic parameter means that, if you make a change to the value, you don't need to disconnect all the connections and reconnect again for the parameter change to take effect. In the case where it is not dynamic, then you do have to make sure that

you have no connections on the database, and then on the first connection that takes place, then the new value will take effect. To make sure you don't have any connections, you can issue a **db2 force applications all** command, and that will force all applications or all connections. And sometimes you may have to do this a few times, because this is an asynchronous command. But then if you issue the **db2 list applications all** command, or **list db2 applications** command, in this case it shows you there are no connections, so you basically force all connections. If you had a connection, like in this case, **db2 connect to sample**, and then you issued a **db2 list applications** command, then it will show you that there is a connection, and this is the name of the application. db2dp is the name of this application, and it shows you the connection, application id, in the case of a remote connection it would not say local, but it would provide you the name of the TCP/IP information. Ok, great!

Now that I've shown you this, because I connected again the new value that I provided for the sample database should now be in effect. So if I take a look now again at the parameter that I modified, I think it was a little bit below for the log size, it should have changed to the new value. So here it is, logfilesize is now changed to 2000.

Slide 19: Database Configuration file (db cfg)

Ok, so let's go back to the presentation.

Slide 20: DB CFG - From the Control Center

You can also make this change also from the GUI tools, from the Control Center, and I'll show you this later on.

Slide 21: Database Configuration file (db cfg)

The last place for configuration is the DB2 profile registry, and this has no relationship to the Windows registry.

Slide 22: The DB2 Registry

And you can change the DB2 registry using the command **db2set**, and **db2set -all** will show you all the values that are currently set.

Demo

So if I use **db2set -all**, then I can see the current values set, if I issue **db2set -lr** it will show you all the possible, all the parameters that I can possibly set. And **db2set**; going back to **db2set -all**, you could set this parameter at different levels, at the environment level, at the instance level, or globally, which means for all the instances available.

Slide 23: Example of DB2 Registry variables

Here are some common DB2 registry variables, like db2com. We will use this variable or this registry variable later on in these lessons especially for the section on connectivity.

Slide 24: Quicklab #3 - Working with instances, databases and configuration

And now I would suggest that you pause and work on Quicklab #3, which will allow you to practice working with instances, databases, and configuration.

Slide 25: What's Next?

Great! So you have reached the end of Part 2 of this Lesson 2 for the DB2 environment, and as to what is next, I would recommend that you continue with Lesson 3, Tools and Scripting.

Thank-you and have a good day.