

# Lesson 10 Transcript: Maintenance

## Slide 1: Cover

Welcome to Lesson 10 of the DB2 on Campus lecture series. Today we are going to talk about maintenance tasks. 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: Overview of maintenance utilities

And we will start with an overview of maintenance utilities.

## Slide 4: REORG, RUNSTATS, REBIND

So what you see in this slide are the main utilities we are going to talk about today. We have REORG, RUNSTATS and REBIND. So normally you may want to do a REORG to reorganize your tables, and we'll cover this in more detail in the next a few slides. Then after doing a REORG, it is recommended to do a RUNSTATS and recommended to do a REBIND. And maybe after 2 weeks or 3 weeks, depending how active or how many changes are performed in your database, you want to do again another REORG, then another RUNSTATS, and another REBIND and so on, and so on. These are maintenance tasks, which mean you, have to run them every now and then, to keep your database in good shape.

## Slide 5: Database Operational View in Control Center

From the Control Center, as well, if you click on a given database, like in this example, you have the sample database highlighted; you can see at the bottom of right corner that you can have some maintenance information about the database.

## Slide 6: REORG

Ok, so let's move on to the next section, which is REORG.

## Slide 7: Table Reorganization (REORG)

A REORG is a reorganization and basically what happens is over time your tables and indexes will get fragmented, like in the file system. So let's say you are doing inserts, you are doing deletes, you are doing updates and therefore, behind the scenes down where your information is being stored, the pages where this information is being stored may not be contiguous anymore, or would be spread all over the place. So by doing a REORG, you basically are reclaiming wasted space and reorganizing all of these pages or data, so that it's more efficient to retrieve them later on.

So obviously the tables that you are updating constantly or changing constantly would benefit the most from a REORG. So you don't need to do a REORG on all the tables of your database, but maybe on the tables that are the most active ones. A REORG can be done either online or offline. This also is similar to what we explain for backup. So an online operation of REORG means that while you are reorging your table or index, the database is being used by other users or there are connections basically on the database. And an offline REORG means that there would be no connections on the database where you are performing the REORG.

## Slide 8: REORG

To perform the REORG, you can follow this simple syntax that I provided here. Again I am just providing the simplest syntax. You can take a look at the manuals for the full syntax of the REORG command. But as an example, you can issue REORG TABLE on the employee table, which is on the SAMPLE database.

### Demo: Switch to Command Editor

So let's try that right now. First of all, you need to click Add to connect to the SAMPLE database or just issue a connect statement from here to connect to the sample database. We need to connect to the sample database because the employee table is in the SAMPLE database. Now we are going to do a **reorg table employee** then we press the execute button. And as you can see from the bottom, the REORG was successful. Now you may be wondering which tables you need to reorg. How you can determine that? Well, there is another command called **reorg check (reorgchk)**. So if you execute this command, it basically goes through the entire set of tables in your database. It first performs a RUNSTATS, which we will talk about in a few minutes, what it is. And then it will do some calculations; and the formulas are provided as well as part of the output. So here is the formula that you will use and based on these formulas, it will determine... , see the formulas are here, right? The output of the formulas, and based on these formulas, this command will determine which table needs to be reorg'd and which ones should not be reorg'd. Anyway, you can read more about the **reorg check** command from the online manuals or the hardcopy portion of the manuals.

## Slide 9: Agenda -RUNSTATS

Moving on to the next section, we are now going to talk about RUNSTATS.

### Slide 10: Statistics (RUNSTATS)

So what is RUNSTATS? Basically, it's a utility to keep the catalog statistics up-to-date. So in DB2, what we have is something called an optimizer. You can think of it as a black box, let's say. And you are going to input some query, some SQL, and you pass it to this optimizer black box, which is going to magically find a way to get your data the fastest. Now this optimizer is cost-based, so it will try to determine which is the lowest cost. And it is going to calculate the cost, looking at the statistics on your table.

So let's say you are doing a **select \* from employee** where employee number equals to ten, ok? Assuming that you have maybe, let's say, at the beginning you have five rows in your table, then DB2 will say, "ok you want to find one row where employee equals to ten" but you only have five rows, so even though you have an index defined, I am not going to use the index, I'm just going to do a table scan which means I am going to go row by row until I find the record that you want. That's because this optimizer went to the statistics, to the catalog statistics and saw that you only had five rows, so it decided to do a table scan because, as you know, five rows a very simple to go one by one or very quickly.

Now let's say after one week, you decided to or you had to insert 10 million rows into the same table, and then you had to execute the same query, right? If the catalog statistics were not updated after you inserted the 10 million rows, then DB2 will still think that you have only 5 rows. So it will still use the same access plan which is means it's going to be a table scan and it will start touching or accessing each of these rows until it finds the one you want, and this will be very time-consuming in the case of 10 million rows, right? And that's because DB2 is looking at the statistics and they are showing that you only have 5 rows, so

that's why it was using this access plan. So it is very important to have the statistics of your database up-to-date so that DB2 can calculate the best access plan. Now you may think, ok, why doesn't DB2 automatically updates statistics whenever I do an update, a delete and so on, right? Well, you can imagine that, you know, in a database management system, you may have maybe, let's say, if it's a bank, for example, you may have maybe one thousand users working on the database at the same time, and they are doing updates, deletes, inserts, etc etc.

As you know, you can imagine how much pressure it will put into the database system, and the database server to update these statistics every time one of this 1000 users performs operation. So by design, DB2 was designed like this so that it doesn't update things on the fly dynamically when people make changes, but we run this command RUNSTATS, so that you can manually run it, or a DBA can manually run this command and then, at that time is when all the statistics will be collected. So that's how, or that's the reason why RUNSTATS needs to be run. And the frequency when you have to run RUNSTATS will depend on how often your data changes in your table, right? Ok.

### **Slide 11: RUNSTATS**

So let's move on to the next section or slide. And here is where you have the syntax of the RUNSTATS command. So you say RUNSTATS ON TABLE and then the schema.tablename. For this particular command, you do need to put the schema explicitly.

### **Demo: Switch to the Command Editor**

So let's run RUNSTATS from the Command Editor, which I have here. And I will do **runstats on table**. Let's use the same table. And as I said before, you have to put the schema. In my case is arfchong. and let's use employee 'runstats on table arfchong . employee'. And when I execute this, I get that my operation was successful, right?

### **Switch back to ppt**

So we can now move on to...

### **Slide 12: RUNSTATS and REORG - Manually**

This next slide: RUNSTATS and REORG can also be performed from the Control Center. So you have to highlight a given table. For example, here I am highlighting table staff, then I am going to right-click and I am going to get this menu. And from there I can choose Reorganize and also Run Statistics. Also we can Reorganize Indexes by themselves.

### **Slide 13: Agenda - REBIND**

And before moving on to REBIND, one more thing about RUNSTATS and the optimizer is that you should know.

### **Switch to Command Editor**

From the Command Editor, and I showed this when I was talking about tools and showed the Command Editor; you could do for example, **select \* from dept where deptno = 'E01'** for example, something like this, and if you highlight this and click on this third button. What that would do is it will show you what we call a Visual Explain graph, which shows you the access plan that the optimizer selected for this particular query. So in this case, we have this table, we are using a primary key index, so it's an index which happens to be taken from the primary key. And I am doing an index scan, and then once I am doing an index scan, I am scanning that index; then I am going to fetch the actual data that I want to get from the table. And the cost for index scan is 0.03. The cost of the fetch, and all of these combinations is

going to be 7.6. And the final cost in timerons is 7.6.

Timerons is a special unit in DB2 and it depends on the speed of your CPU and the speed of your disk, your memory and so on. So depending on which computer you are using, you may get different results for your timerons. But it is a good measure for getting a reference. If you get a lower amount in timerons for this same query from the same computer, you basically did something that improved the performance on this particular query. For example, you did a reorg, and then you run the same query and then it performs better, so then you can tell the reorg was helpful for this purpose of improving the performance. Anyway, so this is just like a related topic with respect to RUNSTATS. So RUNSTATS is very very important. Let's talk about the next section in this presentation which is REBIND.

#### **Slide 14: Bind/Rebind Packages**

REBIND normally applies to embedded SQL applications and SQL PL stored procedures, and let's explain rebind by using the following example. So let's say you have an embedded C program, which has an extension of .sqc, and the embedded C program means that it is a C program where sql is embedded in the program. And normally you will go through a pre-compile process here where the file is divided into two parts, or into two different files. One file will be a C file with that .c extension and another will be a bind file with that .bnd extension.

The .c file will basically contain the C code and the SQL that was embedded would basically be commented out. And the bind file basically contains the SQL. Now these files are obtained through a pre-compile process and then, once you get the two files, you will compile the C file using the C compiler as usual, and you will get an executable, and then for the bind file, what you do is you bind the file, the .bnd file, and that's like compiling the SQL, but that will also perform or obtain the access plan for that SQL. And that will be stored in a package within or inside the database. And the executable and the package have some token to establish the relationship between this executable and this package.

Now the bind file, when you do the bind, it is at that time when, as said before, you will get the access plan. So that means at the time you are doing the bind is when DB2 or when you were actually looking at the statistics to calculate the best access plan. So what happens when you are doing the bind plan, your table only, let's say, it has 5 rows as in the previous example, right? So maybe if because the catalog statistics at the time for that table said that you have 5 rows, then the access plan that is stored in the package will say that it is good to just use a table scan, which means just go one row at a time. Now let's say after 2 or 3 days, you insert 10 million rows, right? And then you do run RUNSTATS, so the statistics are updated, however, even though you run RUNSTATS, this package will not be automatically updated. So what you need to do is do a rebind, so that this access plan is updated where the new catalog statistics are taken into account, so that now you say, oh, ok there are 10 million rows, there are not 5 rows any more. Therefore, probably I will have to use, or the optimizer will have to use a different access plan, like using an index. Ok so that's basically what a rebind is.

#### **Slide 15: REBIND**

And you can execute the **db2rbind** command to rebind all the packages in a given database, so you could run, for example, **db2rbind sample -l mylog.txt**

### **Demo: Switch to cmd window**

So we can run it from a command window. So if I open here, I had a shortcut to open the command window. I can do **db2rbind sample -l** and then I will put here any name, for example erase.txt And I execute and it quickly did a rebind on the database SAMPLE.

### **Switch to command editor**

You can also do a rebind from the Command Editor. Just use the exclamation mark to exit to the operating system, and then do rebind from here, so you do **db2rbind sample -l erase.txt**. This file is just a log file, so just store information about the rebind, and then you can execute and as you can see from here as well. It executed that command successfully. Ok, that's the rebind now.

### **Switch to the Windows Explorer**

If you want more information about this precompilation process, you can go to this, in Windows, or you can go to this directory: C:\Program Files\IBM\SQLLIB\samples, well I chose the C folder which is for the C program language, but you can also use C, CLI, CLP, COBOL, etc etc. These are all sample programs. And for example, in the C folder you can see the bldapp.bat, and if you open that file, you will see that, let's look for precompile, there is a step here, so you basically can run these batch files and they have the information that will take a .sqc file, and precompile it, and then create those 2 files and then compile etc etc. So you can read more or take a look at these files and use them to work with the existing sample files here. And it would be a good way to learn more about programming in C or any of these other languages, but anyway we are talking about precompilation. I just mention this because we were working on this REBIND.

### **Switch back to ppt: Slide 16 – Agenda - Automatic Maintenance**

Ok, so now we are going to move one to the last section of this presentation, which is Automatic Maintenance.

### **Slide 17: Maintenance Choices**

There are different maintenance choices that you can follow. The first one is automated maintenance which is basically that you tell DB2, or you provide DB2 with the window where DB2 will automatically do a REORG, RUNSTATS or BACKUP. You can do a manual maintenance where you can do it manually, where you say, ok this problem happens, so I have to do a manual RUNSTATS or a manual REORG. Or you can create a script, or scripts, and you can run them. And you can schedule when you want to run them.

### **Slide 18: Automated Maintenance**

For automatic maintenance, all you need to do is basically specify a maintenance window, and then DB2 will perform the maintenance required, only if it's needed. So even though you specify, let's say, Saturday from 9 p.m. to 12 p.m. is my maintenance window, it doesn't mean that DB2 will always perform the maintenance at that time. If there is no maintenance required based on what DB2 calculates, then DB2 will not do anything during that window, but if something is required, then DB2 will do it during that window. Now there are two windows that you have to specify: an **online** and an **offline** window. So online means you'll have to specify this window for online operations, which means the database is active at the time; there are connections on that database. And offline means there are no connections, so DB2 may take action so that ensure that there are no connections before executing the operations during this window.

**Slide 19: Configure Automated Maintenance Wizard (1)**

Ok, if you want to enable or Configure Automatic Maintenance, you can right-click on the database. In this case, it's the SAMPLE database. And choose Configure Automatic Maintenance.

**Slide 20 – Configure Automated Maintenance Wizard (2)**

And then from there, you will go to a wizard where you just have to follow the steps. And you will work on this on one of your labs. And from here you can see how you basically have to configure automatic maintenance and it gives you all the information required in this page.

**Slide 21: Scheduled Maintenance Using Task Center**

The second way to do maintenance is to use the Task Center and to do scheduling. So this is the one where you basically, the third option that I mentioned, where you can create scripts and then you can use the Task Center to schedule these scripts so that they are run at a given time. And the second choice which I have not talked right now, but the basically the second choice is to do it manually, where a problem arises and then you actually do the maintenance work at that time.

**Slide 22: QuickLab #11 – Configuring Automated Maintenance**

Ok, so now you can work, maybe just pause this presentation and work on quicklab #11 to configure automated maintenance for the SAMPLE database.

**Slide 23-24: What's Next?**

Ok, so... Congratulations, because you have just completed Lesson 10, Maintenance, which is the last lesson within the database administration part of these e-learning courses or e-learning series. So with this, as to what is next, maybe to move to the next set of courses and starting with Lesson 11, which is Concurrency and Locking and this will be now part of the Application Development e-learning part. So with this, I thank you very much for attending and I hope you will continue with Lesson 11. Thank you.