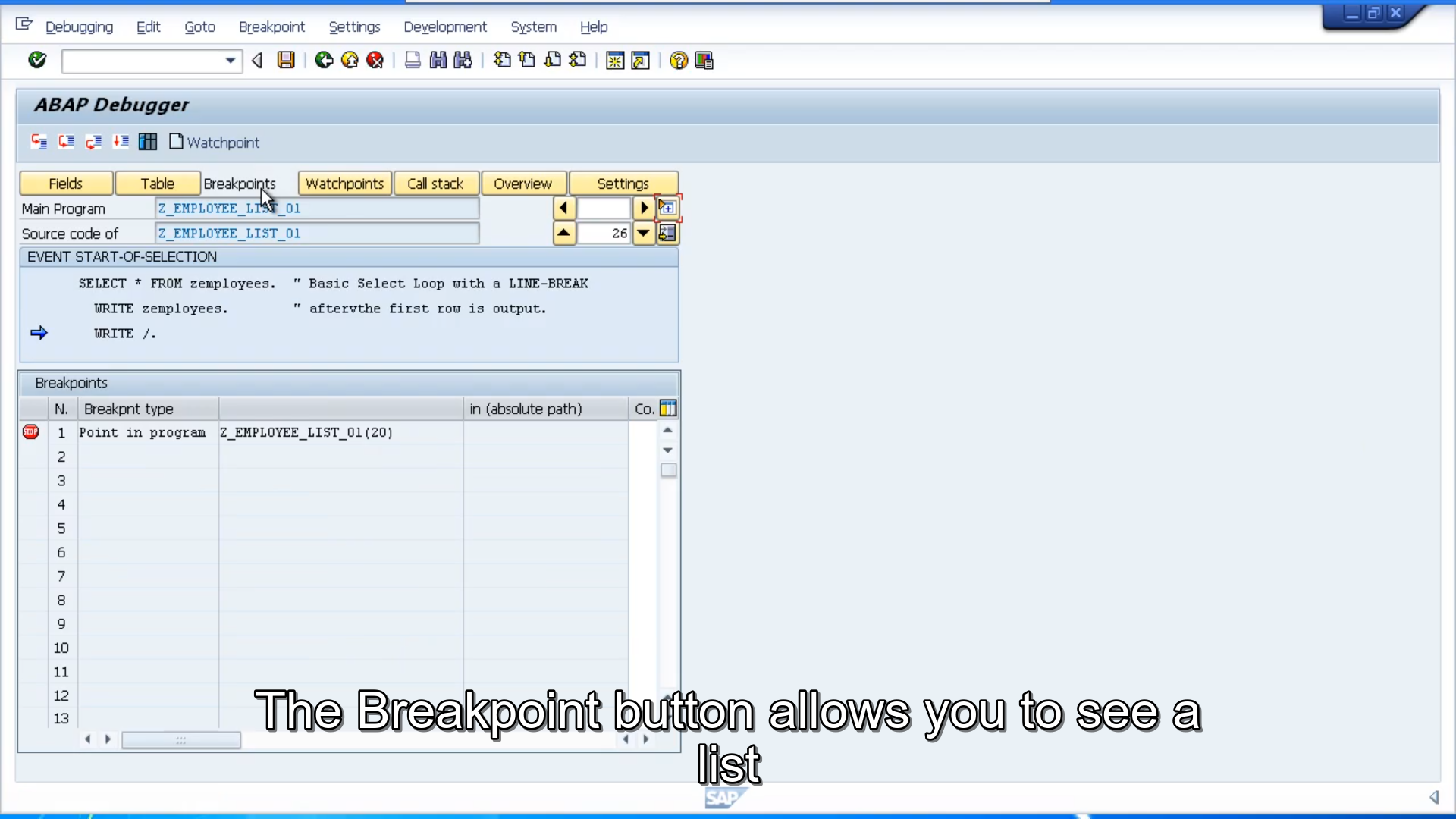
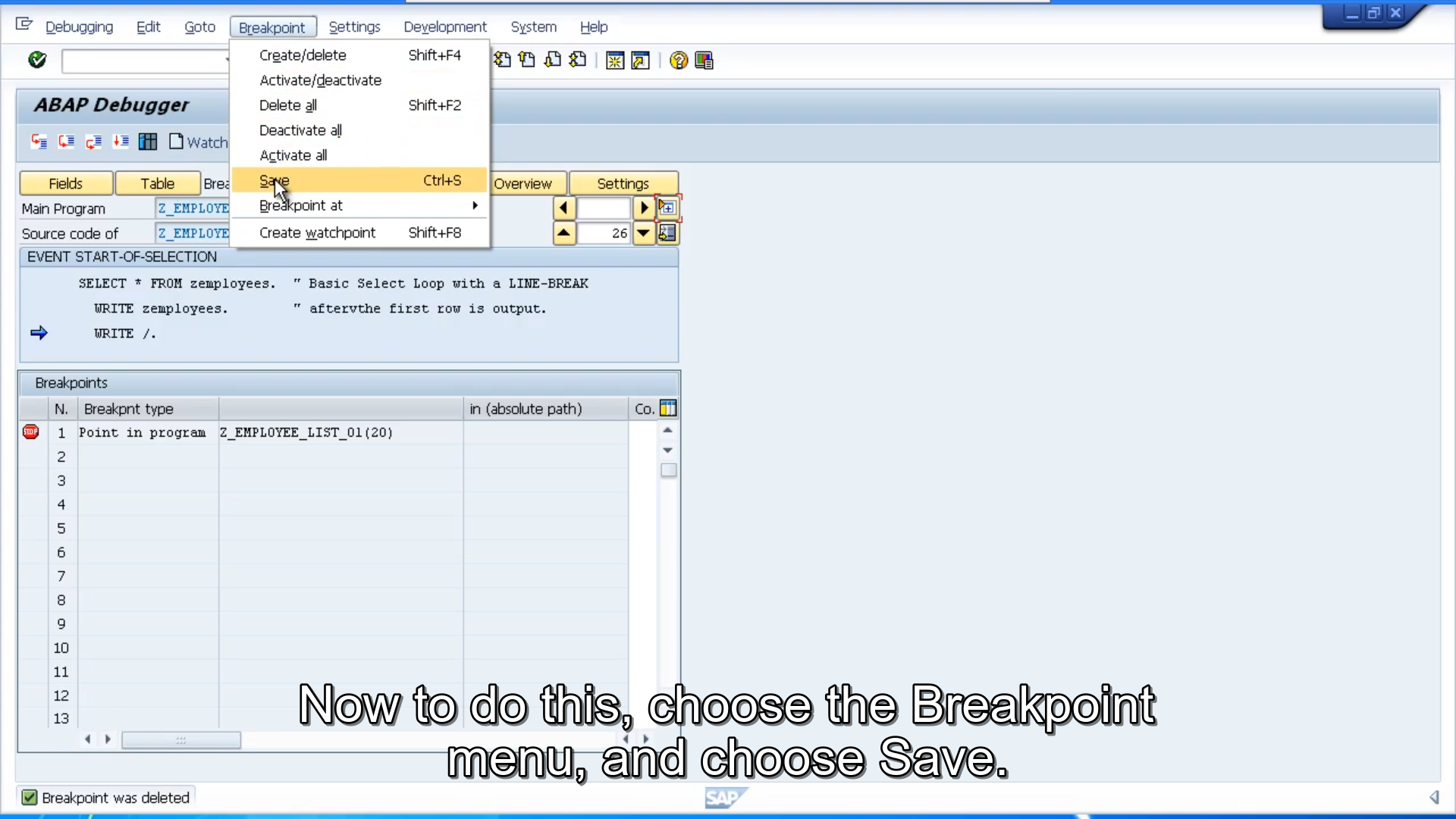
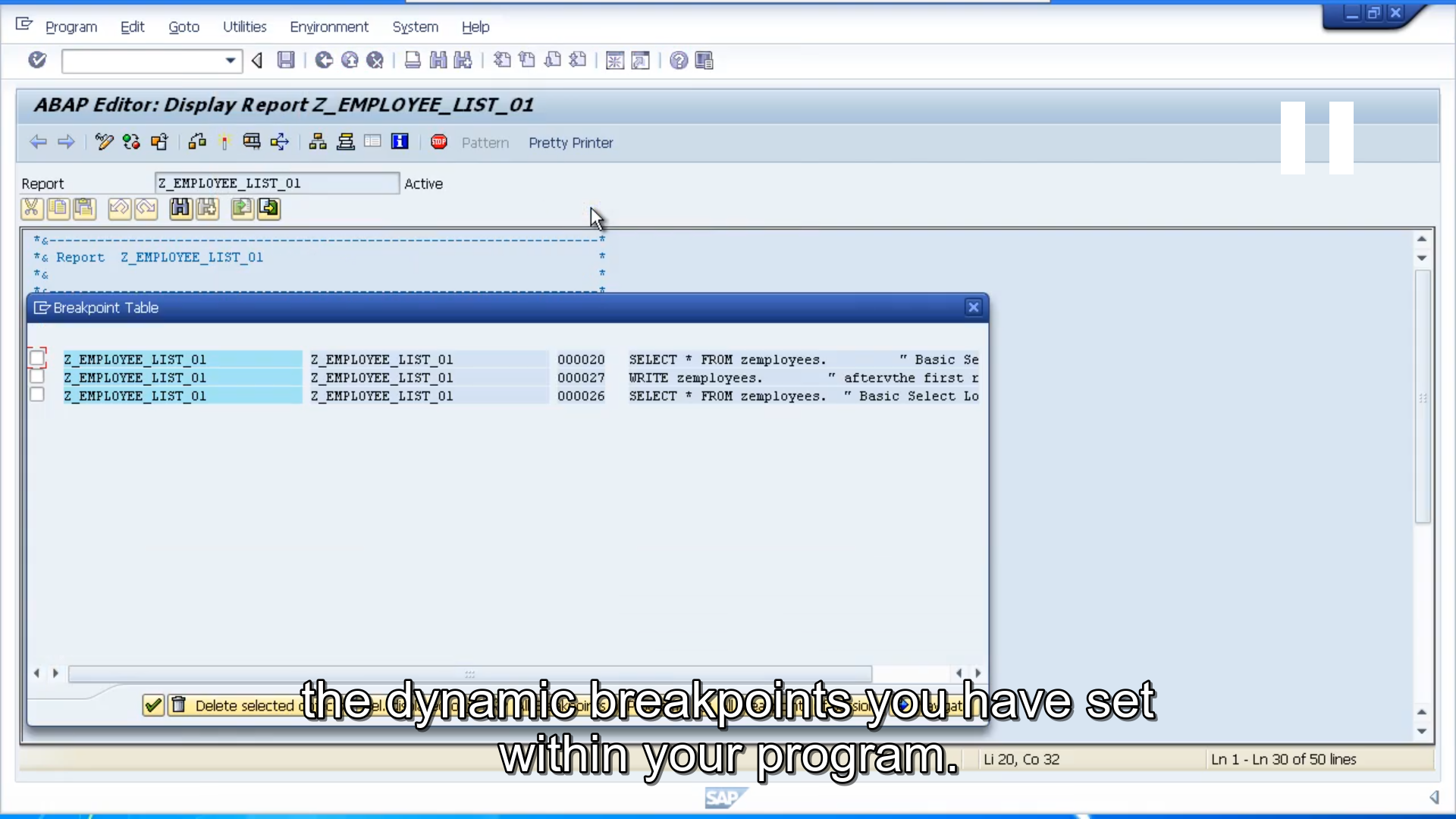
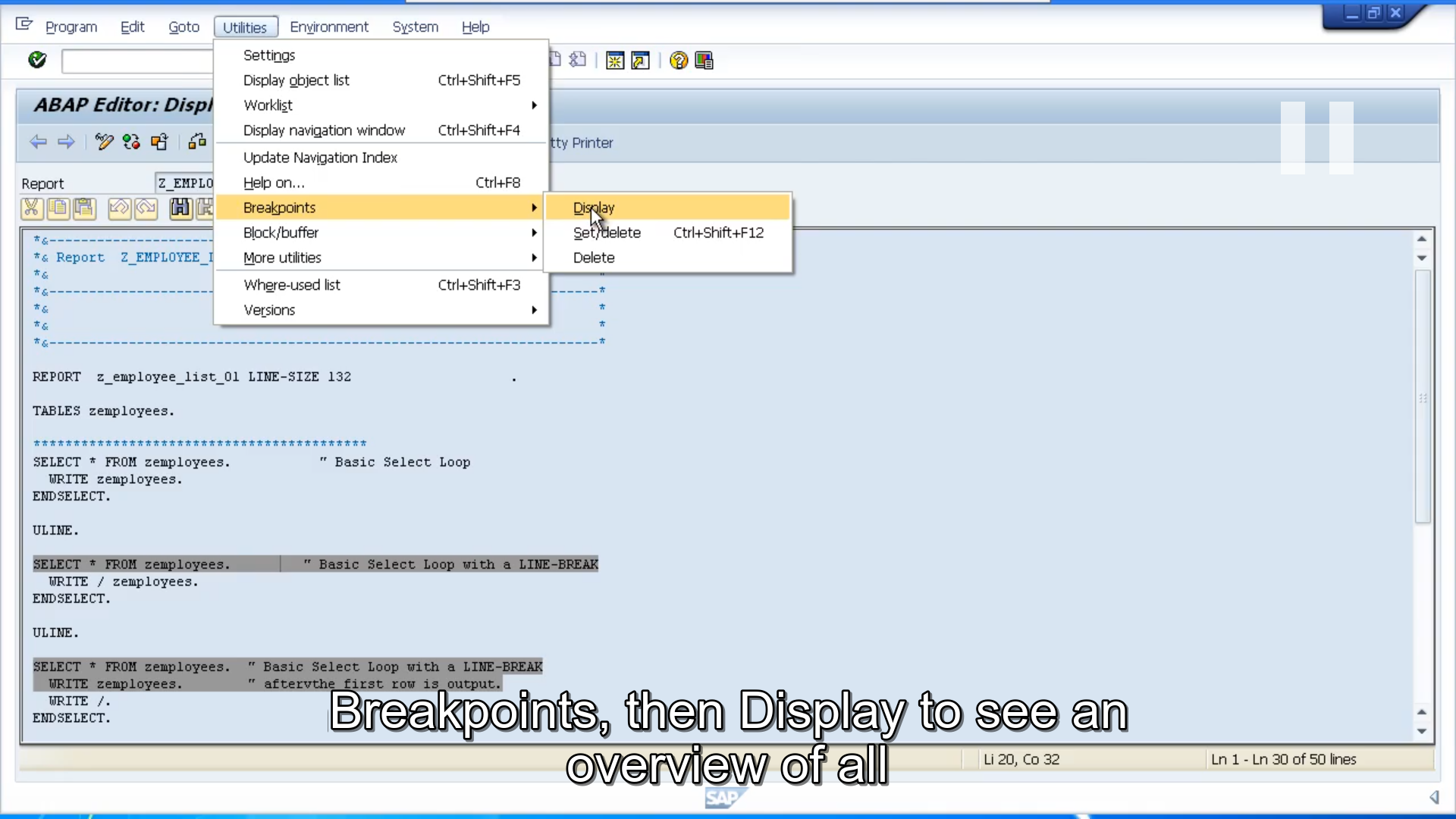
Let's have a look at breakpoints in more detail. The Breakpoint button allows you to see a list of individual breakpoints that you have set within your program.

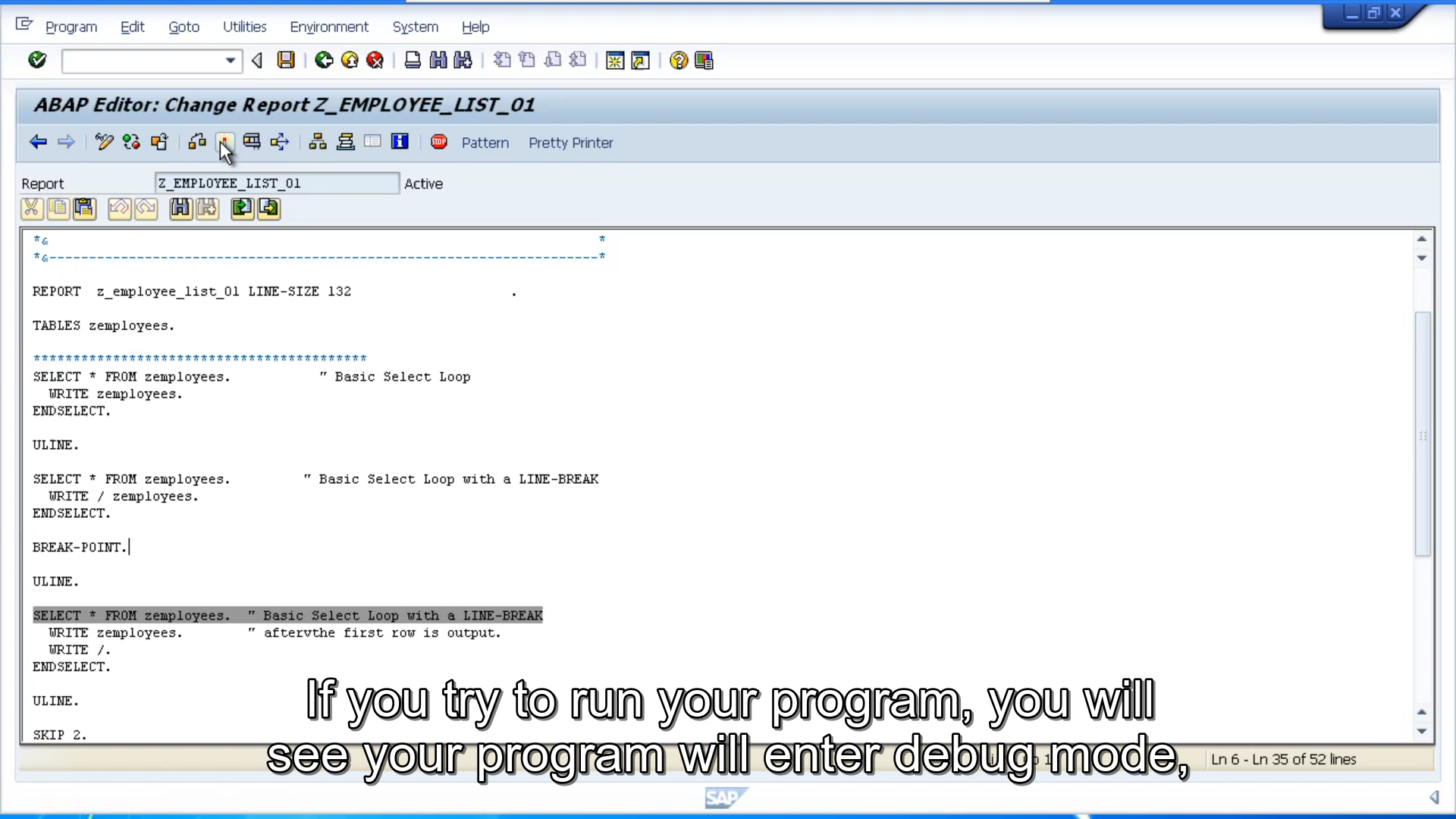


And by double-clicking on any breakpoint, entering the table, it will remove the break point. This breakpoint table is very handy. When you're in a large program, and you have to set lots of breakpoints to keep track of the different variables that are being used. And the table allows you to review the breakpoints sets, and when you don't want them, you can just choose which ones you want to remove. It's important to remember the lifetime of your breakpoint and watchpoints which we will go on to. It's only for the length of your current debug session. When you leave your debug session, your breakpoint will be deleted. There is an option that allows you to save the breakpoint. And watchpoints before closing your debug session. So that your breakpoints remain active for the next time you come to debug your program. This allows them to be set in place, and saved, so you don't have the hassle of recreating them. Now to do this, choose the Breakpoint menu, and choose Save. These breakpoints will remain intact, until you manually remove them or until you end your current SAP session.



If you are in the ABAP Editor, you can choose the Utilities menu, Breakpoints, then display to see an overview of all the dynamic breakpoints you have set within your program. You can then use the options available to delete your breakpoints. By double-clicking you will navigate to the location of the breakpoint in your program.

Earlier on you may recall I mentioned static break points. This is where you enter a line of code within your ABAP program itself. Forcing the program to enter debug mode at the specific line you choose. To do this, you need to insert the statement BREAK-POINT.

If you try to run your program, you will see your program will enter debug mode, and your cursor should be positioned at the line of code where you entered the breakpoint. Now by embedding this statement into your programs, it is active for all users. As you can imagine, this is undesirable for anyone else that tries to run your program that doesn't need to debug your code, so be careful with this, and ensure you don't leave this statement line, lying around in programs that will get transported to other systems, such as your q and a, and your live environment.