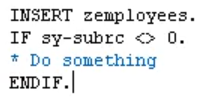
we'll take the scenario that we want to insert a brand-new record into our zemployees table. So, in its simplest form what we will do is write insert followed by our table name with a full stop at the end.



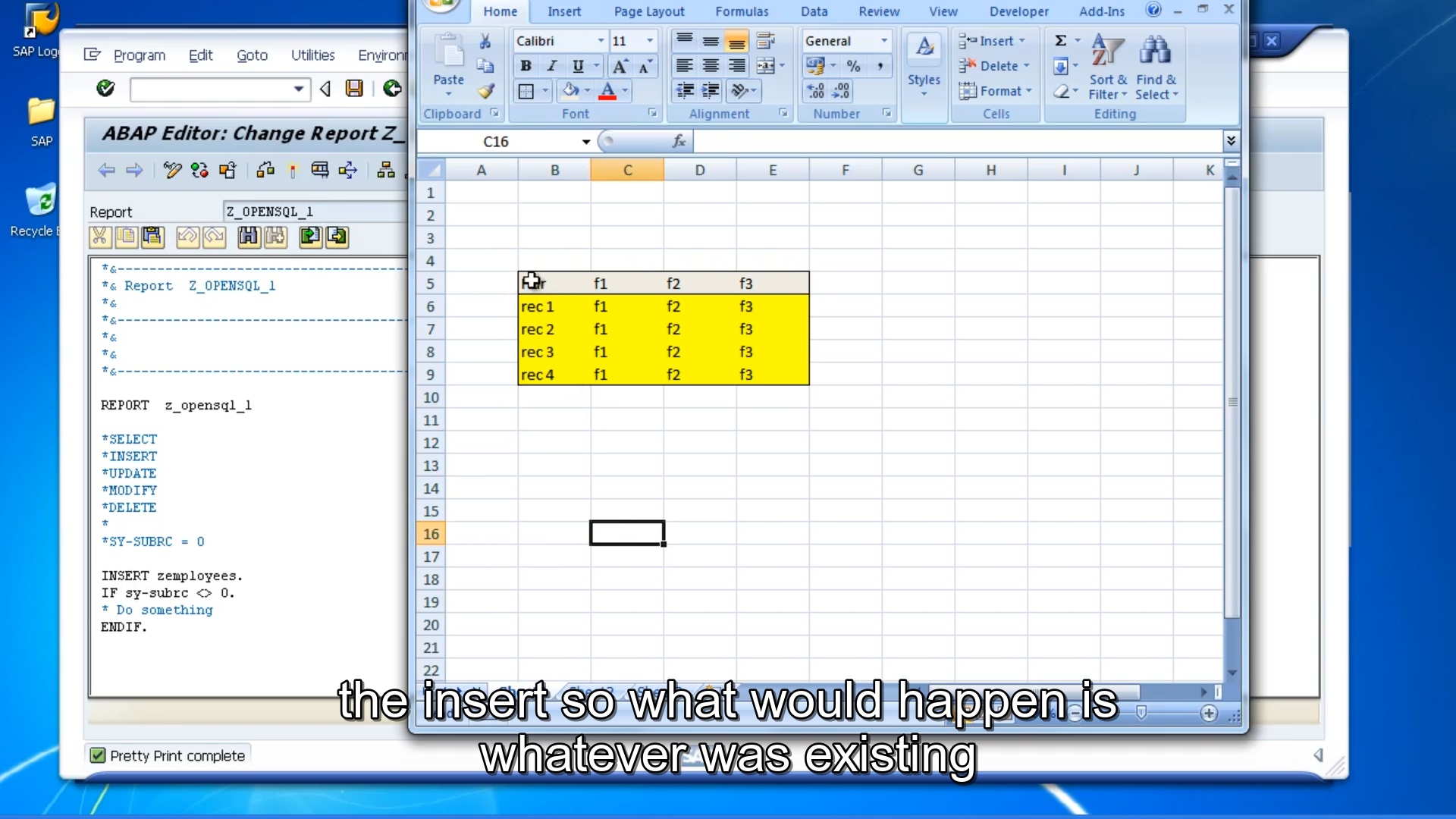
As you can see this is super simple. But one little thing to note is if you use this form of the statement, you always must specify the table name. You cannot use a variable instead.



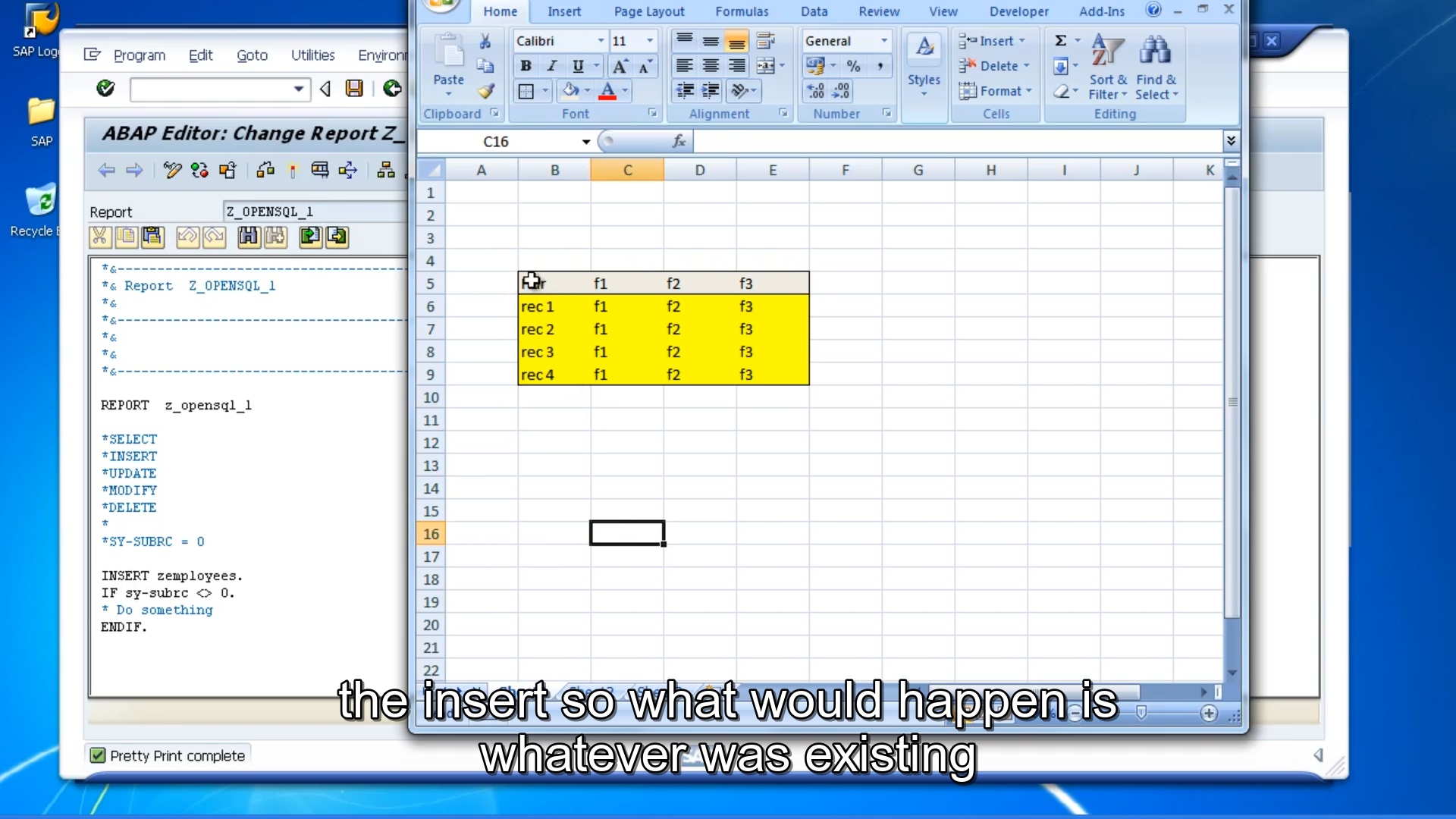
You might wonder, what are we actually inserting? Well, this is where we need to introduce a concept called a work area.

In the ABAP statement we have just written expects a work area to exist that was created when we declared an internal table. And this type of work area is often referred to as a header record.

So let me bring in Microsoft Excel to try and explain a header record and how it's attached to an internal table, and compare it against a standalone work area.



Then when we work with an internal table with a header record, we have this additional record up here.

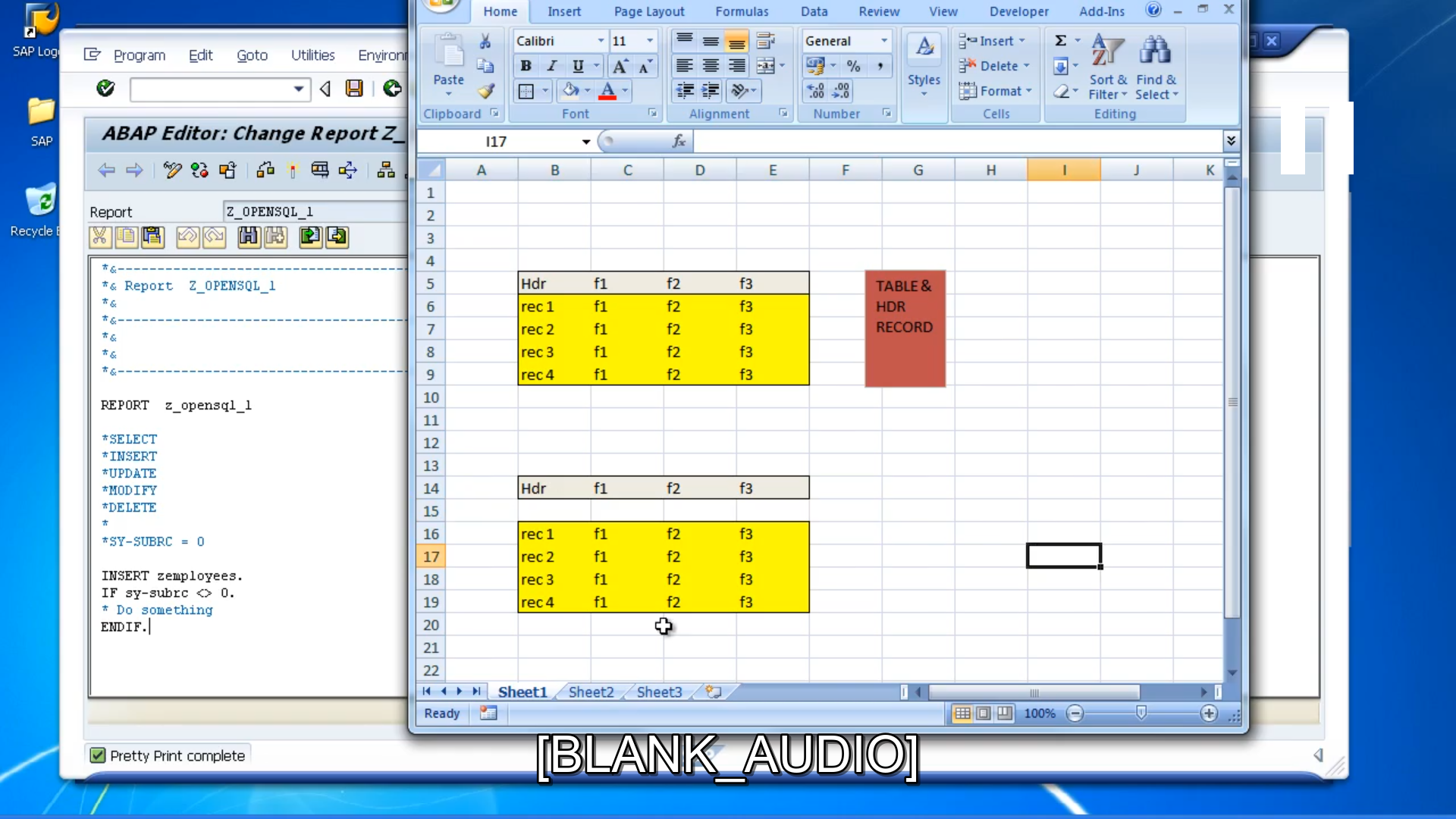
And this little guy is what gets populated when we reference. The table just by its table name or in our case where we chose the insert so what would happen is whatever was existing on this header record when the INSERT is executed, this record would be inserted into the table body.

This header record only exists in memory, whilst our program is being executed. It doesn't contain a record held in the table itself, it's just a place for us to work with a current record, and that's where the term work area comes from.

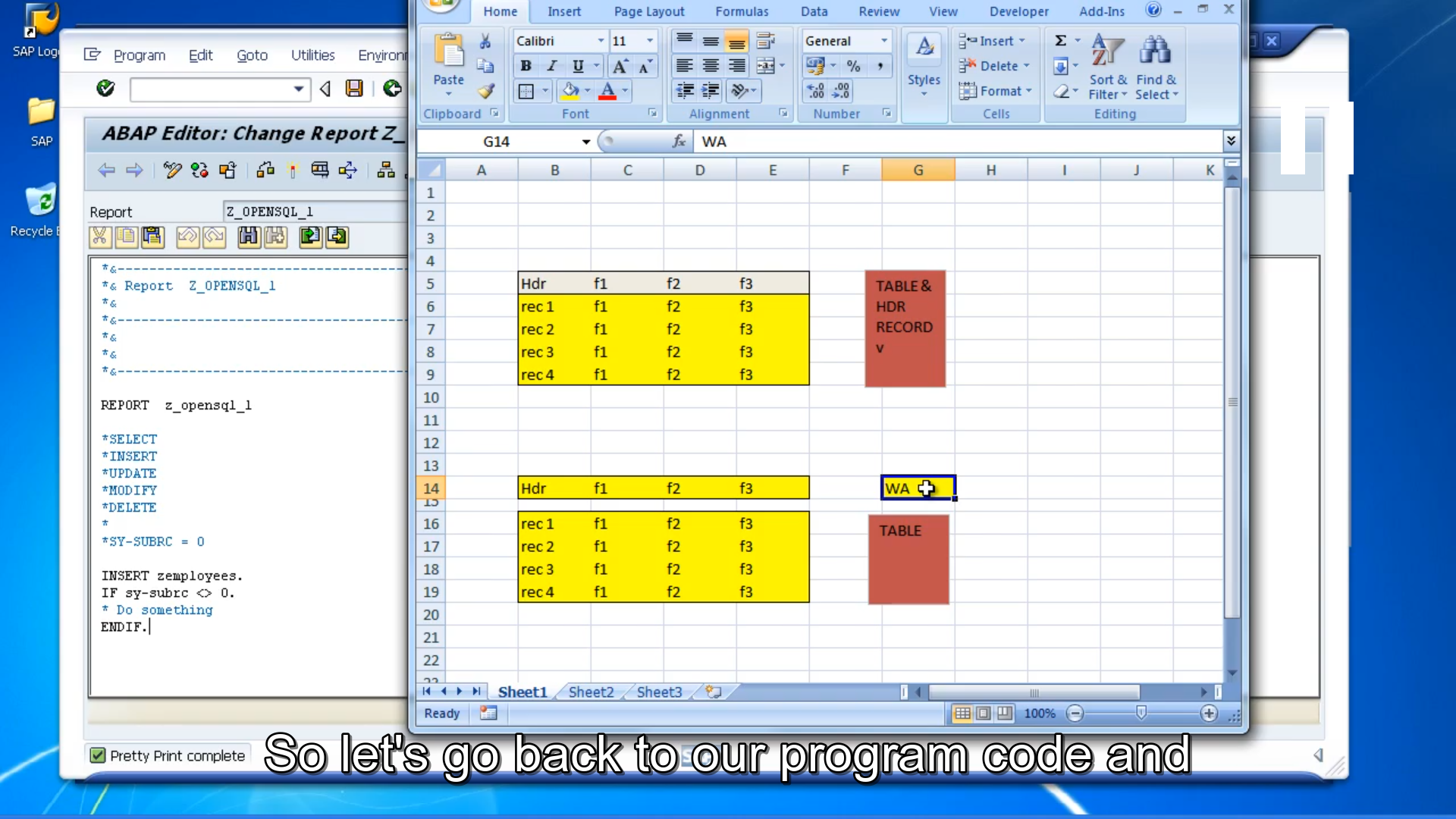
We just used it to hold the current record we are working with. So, when we are working with the data in our program, we're always referencing this header record just by using the table name. And it can become a little bit confusing. Because you think you're referencing the table, but actually no, you're working with a record held in memory that has the same structure as the table.

The fact that the table and its header line have the same name leads to confusion.

This way of referencing a work area for a table is no longer used, and you shouldn't create programs yourself using this type of work area. So, to avoid any confusion, when working with internal tables, you should make sure your programs work with separate work areas. And what do I mean by this? Well, it is similar in a way to what we have here. Let me pop it down here. If I do an insert. So, this example is a table with a header record. This one is an internal table without header record. But we have a separately defined work area. We'll call this WA. And this is our table.



So, when we write code, we can reference the table by itself. And, when we reference the work area where we want to work with a record in memory, we reference a completely separate work area. It will have its own name, so it will be clear that it does not belong to the table exclusively.



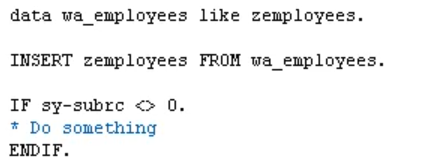
It's a separate structure that we will create in our program that sits by itself.

So, let's go back to our program code, it's just like a normal structure, all we do is data, and we'll give it a name, let's say wa\_employees. Then we need to give it a type.



But it won't be a table as such, it will be a record structure not a table structure. So, our work area will only store one record. So, once we declare our wa\_employees work area, we can then change our INSERT statement to use the work area to insert a record into the table. And we do this by editing the INSERT statement, instead of saying INSERT zemployees.

We will say INSERT zemployees FROM wa\_employees, which is our work area.



Now also, with this form of the statement, we could, actually use a variable here to identify the table. So instead of doing INSERT table name, we can do INSERT and put a variable there that we may have populated earlier with our zemployees table. And the only thing to remember is, if you do use a variable, you have to enclose it in brackets. So, our INSERT statement would look like, would look like this, INSERT, our table name in brackets.



