



Now, let's talk about internal tables, we use structured dates or objects in database tables and internal

tables.

Now, this means if we declare a structure as a global object in the data dictionary, then this can

be used in database tables and internal tables.

If we declare structures in our program only, they are only local.

So then they can only be used as internal tables in our programs, not in database tables.

One of the nice things about the Arab language is it allows us to copy data between two structures,

even if they are not the same type.

And this is all done based on the actual component name of the structure.

So when we have internal tables and database tables, we can copy data from one field in one structure

to the same field name in a different structure, and we do this using the move corresponding statement.

So as you can see in the example here, we have moved corresponding from A to Structure B, the system

takes into account the field names that we have defined in our structures.

And as you can see here, only like name components of a structure are copied.

All the others are ignored.

And bear in mind, the automatic conversion will be done by the system, if at all possible.

So here on the right hand side, we have a structure, a we field one, two, three, four, structure

B with one, two, three.

When we use the move, corresponding statement structure B will be filled with the contents from field

one field to field three of structure.

But Field four will not be touched.

There's nowhere for it to go because that doesn't appear in our structure.

B.

Now, when we are creating internal tables in our programs, there are a few things to keep in mind.

And the most basic are what exactly do you need in your internal table if you want to read data from

a database table?

There's actually no need most of the time to create a structure that your internal table will be based

upon with all the files from the database table, because more often than not, you don't need all the

data.

So think is it best to create your own structure with only the fields that you're going to need?

Also, keep in mind, the memory usage, as in about programa, it is up to you to make sure your programs

are as efficient as possible.

And if you're reading data from a database table, more often than not, you do not need all the records

and Oldfield's of the database table to be read into an internal table all in one go just so that the

records you want now when you're creating your internal table.

Also keep in mind the organizational structure of it.

So ask yourself these questions.

What Salt-Water do I need?

What logical structure do I need to create?

Does it make sense?

The key fields and if you do, will these have to be unique and this will help you determine whether

you create standard internal tables, sorted internal tables or even hashed internal tables.

Remember, depending on whether key fields are required or not will help make the decision of what type

of internal table you need to create.

Which also has an effect on the performance of your programs.

So let's go through the different steps for creating internal tables.

First of all, at a high level, we define the type for our table.

And as we've seen in a previous video, we can declare strictures that define the time for us or we

can use the dictionary that will have Global Times already defined.

Once we have the time defined, we can then instantiate we create our internal table using the type

structure that we have selected.

And as I've just mentioned, we have three types of internal tables, standard, sorted and hashed,

and you can see there are various things to bear in mind here.

Standard are similar to database tables in that we can access the records one row at a time or by using

a table key.

For salted internal tables, the data is structured differently.

The data is stored using this specific case we define when creating our table.

But again, we can read through the table one row at a time and by the keys, then finally the hash

table.

This is specific and most often this is the fastest way of accessing data in an internal table.

If it's defined as hushed because Hashd uses a specific algorithm and only use it unique keys to access

the data.

We don't have the facility to read the data line by line.