We're going to look at

* concatenating strings together
* condensing character string
* finding the length of a specific string
* replacing characters within a string
* searching for specific characters
* using the shift statement to move the contents of the field left and right
* splitting character strings
* using the sub field functions to access specific characters within a string.

The concatenate statement allows us to join two-character strings together to form a third string. And, it's fairly easy to understand and we start off by using the word, concatenate.

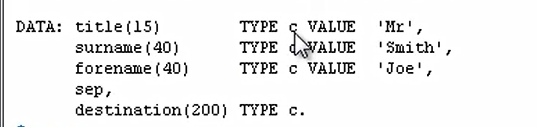
Concatenate f1 f2 into dl [separated by sep ].

And then it goes in the form of field 1, field 2 and we can carry on with field 3, field 4, field 5, etc. And then we choose a destination, that we want the output string to go to. So, then we can have destination one.

We can also add in an additional term called Separating bar, to allow us to insert a specific value in-between each field into our destination field.

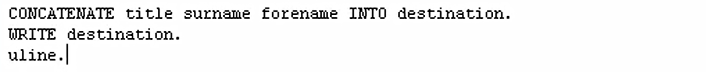
Now some things to note, if the destination field is shorter than the overall length of the input fields, the character string will be truncated to the length of the destination field.

So, make sure when using the concatenate statements, you are using string data types. Which as we have already discussed can hold more than 65,000 characters.



So, the first field is title, we're declaring a length of 15 characters. It is of data type C, the value of Mr. And we have surname. 40 characters long. And the surname will be Smith. And we have forename. 40 characters again. First name of Joe. And then we declare a separator field here.

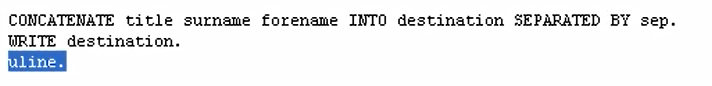
I've just called the SEP and you'll notice I haven't declared the length and I haven't declared a value or the type. So, this will take on the default that the system uses, which is a character string with a length of one character. And the last field is destination and I've just declared it 200 characters long on data type C.





MrSmithJoe with the underline. So you notice a couple of things here. All the characters have been concatenated together so we got no space and it's left a line, the result. And that's because we defined the result as a character field.

If we used a data type of n, a number everything would have been right lined.



we'll add in a blank space between each field when it gets populated into destination.

