





Here are some syntax rules that you should follow when you are creating DDL source. 1st

keywords must be all uppercase or all lowercase or in lowercase with an uppercase initial letter.

For example, here 'SELECT' all letters are uppercase then 'Select' first letter is uppercase and all other lowercase

and 'select' all letters are

lowercase.

So these kinds of combinations of letters are allowed for keywords and the combinations that are not

allowed is mixed uppercase and lowercase.

For example, here in this 'SeLect' , the first letter is uppercase and the 'L' is uppercase so such kind of

combinations are not allowed for keywords. want to comment a single line

then use two // and if you want to comment out multiple lines then in that case you

have to inclose those multiple lines that want to comment out by character /\* and \*/.

CDS view names and SQL view

names are not case-sensitive and the maximum length is 30 characters.

Number literals must always be specified in full and decimal point use as a decimal separator if

necessary.

For example, here 1, 2.0, 0.5

all these numbers are allowed but if you are writing number like .5 then that number is not allowed.

If you are using characters literals then those character literals must include in ( ' ).

Now let's see what are the naming rules that you have to follow.

First DDL source name.

The maximum character length in DDL source name is 30 and all the letters must be always the uppercase case.

CDS view name.

Maximum length of character is 30.

CDS view name is not case-sensitive and it can be different from the DDL source name.

but it's not recommended to have CDS view name different from the DDL source name. maximum 16

characters

you can have in SQL view name. The SQL view name is not case-sensitive and it should be different

from CDS view name.

let's quickly go to the last topic that I want to cover in this video.

When you define CDS view you have two option to specify field list. Option 1, after a keyword SELECT

and before keyword FROM this corresponds to SQL standards syntax.

The second option you have is, you can inclose field list in { } after FROM clause of SELECT statement.

This is speciality of CDS syntax and allows improved auto-completion support.

Hence it is recommended to use second CDS specific syntax. Hence in this course

whenever I'll write code for CDS, I will use CDS specific syntax.

I hope you have understood, what are the syntax and naming conventions that you are to follow when you

are writing code for Core Data Services.