
Article

[Heloisa Paiva](#) · Mar 2, 2023 3m read

Quick sample database tutorial

Introduction

This is a simple tutorial on the quickest way I found to create a sample database for any purposes such as testing, making samples for tutorials, etc.

Creating a namespace

1. Open the terminal
2. Write the command "D \$SYSTEM.SQL.Shell()"
3. Write "CREATE DATABASE " and the name you want for your namespace.

Now you have a new namespace in a faster way than creating it from the Management Portal - which of course offers way more configuration options.

To select it, quit the SQL Shell writing "q" and then typing 'zn "namespaceName"'. Now everything you do in the terminal is in the scope of this namespace.

Creating a table

1. Open the terminal
2. Open the SQL Shell with "D \$SYSTEM.SQL.Shell()"
3. Enter the multiline mode by pressing enter
4. Write the code such as the following example:

```
[SQL]SAMPLE>>  << entering multiline statement mode, 'GO' to execute >>
1>>CREATE TABLE Sample.PersitentData (
2>>Name %String,
3>>Age %Integer )
4>>GO
```

You can specify the properties with the names you want and with types from IRIS (such as %String) or from SQL (such as VARCHAR), choose the one you're more comfortable with. You can also define many configurations within that code, as you can see in [CREATE TABLE \(SQL\)](#), but the objective of this article is to make everything as simple as possible.

Now you can see your class in the Management Portal, Studio, VS Code, or access it from the Terminal, with the name and package you put instead of "Sample.PersitentData" in the example.

Populating the table

Once you have a table created, access the source code of the class through Studio or VS Code and add %Populate to the "Extends" parameters.

The class signature should look like this:

```
Class Sample.PersistentData Extends (%Persistent, %Populate)
```

of course, with any other classes you might want to extend - %JSON.Adaptor and %XML.Adaptor might be useful for exporting data, viewing it in the productions portal, etc.

Now you can open the terminal and execute the following code, substituting "100" for the number of rows you want:

```
D #class(Sample.PersistentData).Populate(100)
```

Notice how in this example I've only specified the age as Integers, so I'll have unreal results because the Populate() method will only create random Integers, not worrying if they're adequate ages for people.

Here's what I got as a result:

ID	Age	Name
1	592435561	Garcia,Quentin L.
2	890770823	Mastrolito,Alexandra I.
3	389916366	Long,Bart Y.
4	2100006	Minichillo,Michael Q.
5	633969908	Hanson,Kyra A.

You can make a more complex code when creating the table, but since I'm doing everything in the simplest way possible, I ran the following code:

```
UPDATE Sample.PersistentData  
SET Age = Age # 120
```

Conclusion

Although the tutorial might not seem that small, having learned all those steps I use them for testing ideas I want to bring to my company and to create samples for the articles I write here - you might have read something I wrote! - and it takes me only around a minute to have a well structured database and focus on developing the content itself.

[#Innovatium](#) [#SQL](#) [#Tutorial](#) [#InterSystems](#) [IRIS](#)

Source URL: <https://community.intersystems.com/post/quick-sample-database-tutorial>
