Article
Muhammad Waseem · Jan 11, 2022 2m read

Open Exchange

Use of %SQLDiag logging along with all-new LOAD DATA functionality

In this article I will explain the usage of %SQLDiag.Result and %SQLDiag.Message table along with all-new LOAD DATA functionality.

> It is recommended to go through LOAD DATA documentation first.

> **After successful operation LOAD DATA insert one record in %SQLDiag.Result table and details are inserted in %SQLDiag.Message table**

Below is the basic command when table is already created and source file does not contain header row.

```
LOAD DATA FROM FILE 'C://TEMP/mydata.txt'
INTO MyTable
```

The file name must include a .txt or .csv (comma-separated values) suffix and both source and target have the same sequence of data columns.

Loading from File Source: Header

To specify that the data file has a header row, use the header boolean parameter, as shown in the following example:

```
LOAD DATA FROM FILE 'C://TEMP/mydata.txt'
INTO Sample.Employees
USING {"from":{"file":{"header":"1"}}}
```

I am using same command in my Medical Datasets Application

```
SET qry = "LOAD DATA FROM FILE  '"_filename_"' INTO "_tableName_ " "_"USING {""from""
:{""file"":{""header"":""1""}}}"
SET rset = ##class(%SQL.Statement).%ExecDirect(,qry)
```

Here **filename** is a complete file path and **tableName** is a table name where data needed to be loaded.

%SQLDiag Logging

Please note that after every successful operation LOAD DATA will insert record into %SQLDiag.Result and %SQLDiag.Message tables.

%SQLDiag.Result table

Below is the structure of %SQLDiag.Result

Table: %SQL_Diag.Result ○ Table Info ◉ Fields ○ Maps/Indices ○ Triggers

Field Name	Datatype	Column #	Required	Unique	Collation	Hidden	MaxLen
ID	%Library.Integer	1	Yes	Yes		No	
createTime	%Library.PosixTime	2	No	No		No	
errorCount	%Library.Integer	3	No	No		No	
inputRecordCount	%Library.Integer	4	No	No		No	
maxErrorCount	%Library.Integer	5	No	No		No	
namespace	%Library.String	6	No	No	SQLUPPER	No	50
processId	%Library.String	7	No	No	SQLUPPER	No	50
resultId	%Library.Integer	8	Yes	Yes		No	
sqlcode	%Library.Integer	9	No	No		No	
user	%Library.String	10	No	No	SQLUPPER	No	50

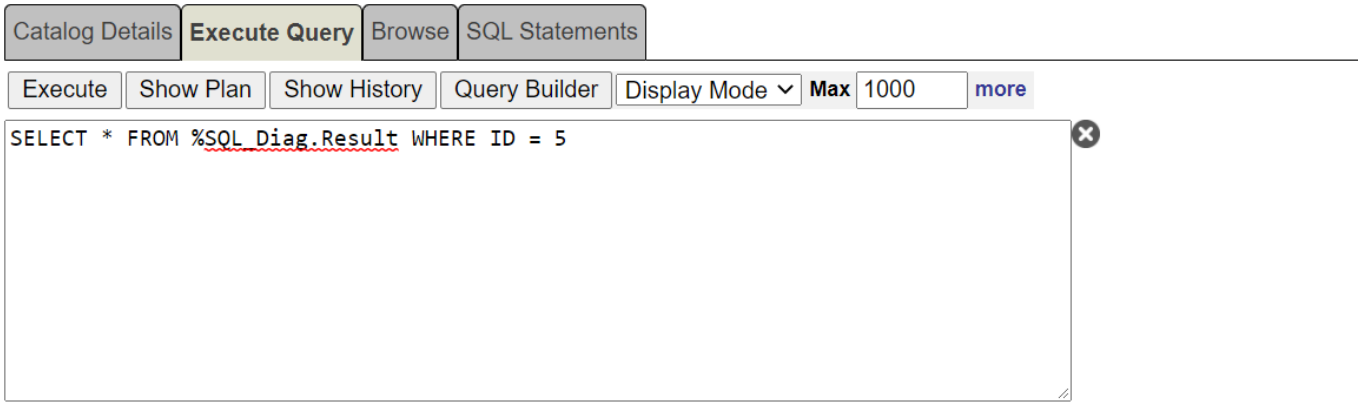
In order to get the detail we need to get maximum ID from %SQLDiag.Result table after LOAD DATA operation. Below is the SQL command to get the Maximum ID:

```
SET qry = "SELECT id FROM %SQL_Diag.Result WHERE ID = (SELECT MAX(ID) FROM %SQL_Diag.
Result )"
```

Let us suppose select of MAX(ID) return 5 from %SQLDiag.Result table which I demonstrated in my open exchange application preview.

We can use Management Portal SQL or $SYSTEM.SQL.Shell() to view the details

```
SELECT * FROM %SQL_Diag.Result WHERE ID = 5
```



Row count: **1** Performance: **0.002** seconds **323** global references **2375** commands executed **0** disk read latency (ms) Cached Query: %sqlcq.DATASETS.cls5
update: 2022-01-11 02:27:02.468

ID	createTime	errorCount	inputRecordCount	maxErrorCount	namespace	processId	resultId	sqlcode	user
5	01/11/2022 02:07:10.120011	140			DATASETS	619	5		irisowner

1 row(s) affected

%SQLDiag.Message table

In case of errors, system is saving number of errors in errorcount column of %SQL.Diag.Result table. These errors can be viewed in %SQLDiag.Message table by using %SQL.Diag.Result ID against %SQLDiag.Message diagResult column.

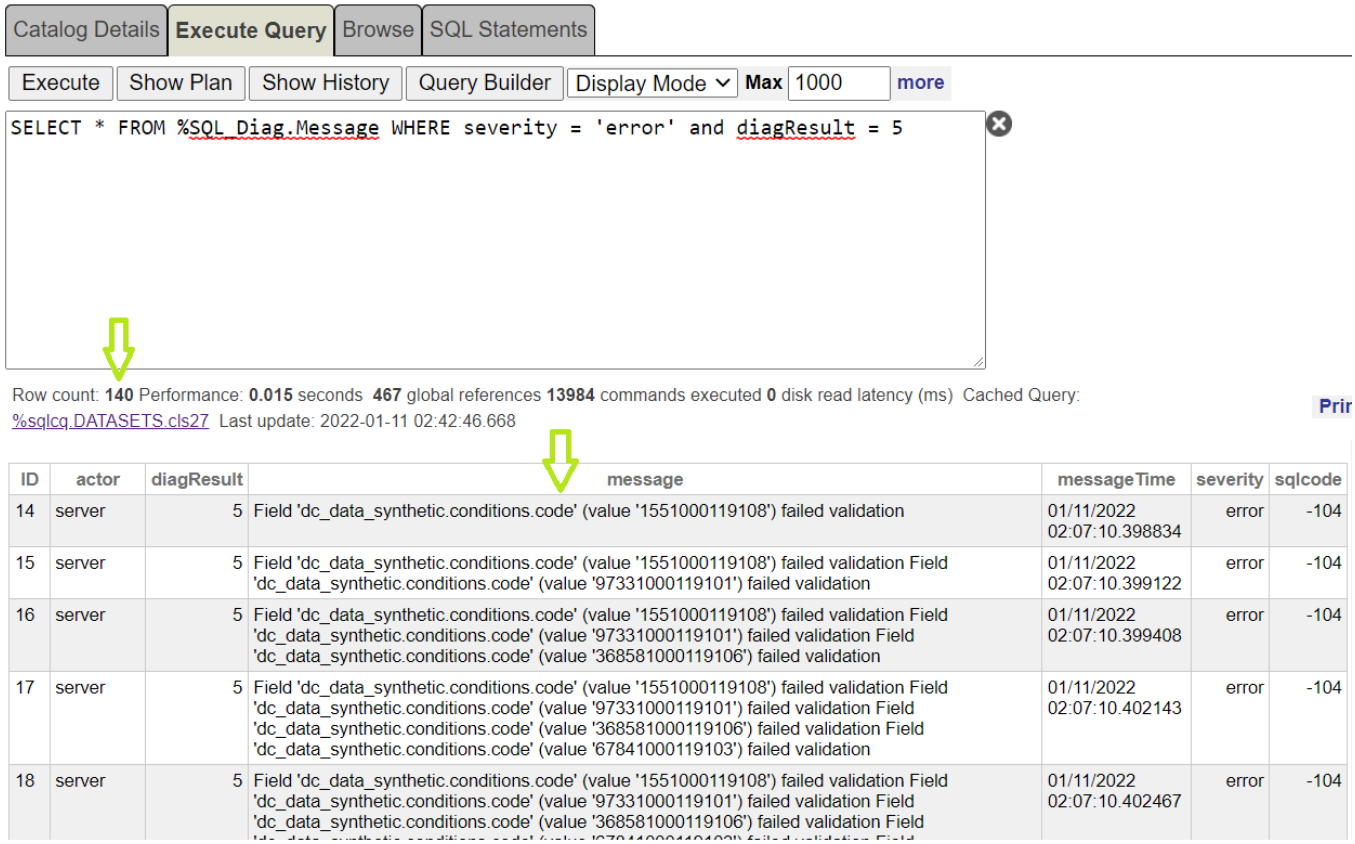
Below is the structure of %SQLDiag.Message table

Table: %SQL_Diag.Message ○ Table Info ◉ Fields ○ Maps/Indices ○ Triggers

Field Name	Datatype	Column #	Required	Unique	Collation	Hidden	MaxLen
ID	%Library.BigInt	1	Yes	Yes		No	
actor	%Library.String	2	No	No	SQLUPPER	No	50
diagResult	%Library.Integer	3	Yes	No		No	
message	%Library.String	4	No	No	SQLUPPER	No	
messageTime	%Library.PosixTime	5	No	No		No	
severity	%Library.Integer	6	No	No		No	
sqlcode	%Library.Integer	7	No	No		No	

Below is the command to check errors details:

```
SELECT * FROM %SQL_Diag.Message WHERE severity = 'error' and diagResult = 5
```



ID	actor	diagResult	message	messageTime	severity	sqlcode
14	server	5	Field 'dc_data_synthetic.conditions.code' (value '1551000119108') failed validation	01/11/2022 02:07:10.398834	error	-104
15	server	5	Field 'dc_data_synthetic.conditions.code' (value '1551000119108') failed validation Field 'dc_data_synthetic.conditions.code' (value '97331000119101') failed validation	01/11/2022 02:07:10.399122	error	-104
16	server	5	Field 'dc_data_synthetic.conditions.code' (value '1551000119108') failed validation Field 'dc_data_synthetic.conditions.code' (value '97331000119101') failed validation Field 'dc_data_synthetic.conditions.code' (value '368581000119106') failed validation	01/11/2022 02:07:10.399408	error	-104
17	server	5	Field 'dc_data_synthetic.conditions.code' (value '1551000119108') failed validation Field 'dc_data_synthetic.conditions.code' (value '97331000119101') failed validation Field 'dc_data_synthetic.conditions.code' (value '368581000119106') failed validation Field 'dc_data_synthetic.conditions.code' (value '67841000119103') failed validation	01/11/2022 02:07:10.402143	error	-104
18	server	5	Field 'dc_data_synthetic.conditions.code' (value '1551000119108') failed validation Field 'dc_data_synthetic.conditions.code' (value '97331000119101') failed validation Field 'dc_data_synthetic.conditions.code' (value '368581000119106') failed validation Field	01/11/2022 02:07:10.402467	error	-104

%SQLDiag.Message shows the details of the errors.

Thanks

#Databases #SQL #Tips & Tricks #InterSystems IRIS
Check the related application on InterSystems Open Exchange

Source URL:https://community.intersystems.com/post/use-sqldiag-logging-along-all-new-load-data-functionality