

Introducing SQL Query Verifier Plugin

IBM Application Runtime Expert for i

Document version: 1.0

To download the master version of this document, visit product home site:
<http://www.ibm.com/systems/power/software/i/are/index.html>

Table of content

1 Introduction.....	1
2 How to use.....	2
2.1 Specify the SQL query.....	2
2.2 Print query result.....	3
2.3 Add result verification.....	3
2.3.1 The number of records.....	3
2.3.2 Column values.....	5
3 Sample.....	6
3.1 Sample database table lppdb.score.....	6
3.2 Steps.....	7

1 Introduction

This document introduces the feature to verify the content and integrity of a database on IBM i. IBM Application Runtime Expert for i (ARE) has been enabled with the ability to access database information using SQL. The SQL Query plugin can verify database information using SQL statements. A SQL query statement can be specified to be run on the target system. The results of this SQL query can be verified, including specific column data and the number of records. The complete query results can also be returned as part of the ARE report. The overview of this plugin is shown in Figure 1.

[Home](#) > [Plugin Selection and Customization](#) > [SQL Query Verifier](#) > [Customize SQL Query Verifier](#)

Customize SQL Query Verifier

Specify the SQL Query and Verifications Details



Collection name:

Select statement [? Learn more...](#)

Example: `SELECT * FROM myDB.myTable`

Print the result set of SQL query to the verification report:

Check the number of records returned for the SQL query [? Learn more...](#)

Column value verification conditions [? Learn more...](#)

Column name	Check class	Expected value	Comparison	Result type	Match
Example id	Numeric	6	Equals	All	False
Example name	String	John	Contains substring	Any	True

This table is empty.

Figure 1: Overview

2 How to use

2.1 Specify the SQL query

Input a SQL query statement in the text area shown below. Only `SELECT` statement is acceptable. For example, "`SELECT * FROM myDB.myTable`". Other statements are not allowed. It is best to copy and paste a statement into this text box to help reduce issues encountered with typing errors or mal-formed SQL.

Select statement [? Learn more...](#)

Example: `SELECT * FROM myDB.myTable`

Figure 2: Specify the SQL query

2.2 Print query result

By default the query result is recorded into report. This means that all the data from the SQL is preserved in the verification report. If you do not care about the SQL query result, you can select “No” of option “Print the results set of SQL query to the verification report”, which is highlighted in Figure 3.

Specify the SQL Query and Verifications Details

Collection name:

Select statement [? Learn more...](#)

*Example: SELECT * FROM myDB.myTable*

Print the result set of SQL query to the verification report:

Check the number of records returned for the SQL query [? Learn more...](#)

Expect value:

Comparison:

Severity:

Column value verification conditions [? Learn more...](#)

Column name	Check class	Expected value	Comparison	Result type
<i>Example id</i>	<i>Numeric</i>	<i>6</i>	<i>Equals</i>	<i>All</i>
<i>Example name</i>	<i>String</i>	<i>John</i>	<i>Contains substring</i>	<i>Any</i>
<i>This table is empty.</i>				

Figure 3: Print the SQL result

2.3 Add result verification

There are lots of options available for verifying the results from the SQL query. You can verify the number of records returned, Column details and other filtering and verification items.

2.3.1 The number of records

A simple check is to verify the number of records returned by the SQL query, determine if the number returned is equal to, less than, or does not equal a specified number. To

check whether the number of records returned from a result set meets your expectation, you should follow the procedure below:

1. Check the box "**Check the number of records returned for the SQL query**".
2. Then, configure your expected results in the following fields:

Expect value: Input the expected value for the number of the records.

Comparison: Four options offered:

- *Equals
- *Does not equal
- *Greater than
- *Less than

Severity: The following keywords denote the severity levels of a problem found during the verification.

- *Error
- *Warning
- *Info

Customize SQL Query Verifier

Specify the SQL Query and Verifications Details

Collection name:

Select statement [? Learn more...](#)

*Example: SELECT * FROM myDB.myTable*

Print the result set of SQL query to the verification report:

Check the number of records returned for the SQL query [? Learn more...](#)

Expect value:

Comparison:

Severity:

Column value verification conditions [? Learn more...](#)

Column id	Column name	Check class	Expected value	Comparison	Result type	Match
Example id	Numeric	6	Equals	All	False	
Example name	String	John	Contains substring	Any	True	

This table is empty.

Figure 4: Configuration for checking the number of the records

2.3.2 Column values

To verify the specific contents of the results returned by the query, you can specify very specific verification criteria. You can verify the contents of a specific column and its values. These checks have a number of advanced features like String, Numeric, and Case sensitivity.

By clicking the add button, you can add a column verification.

Column name: The column name to be checked.

Check class: The type of the expect value, two options:

- *Numeric
- *String

Expected value: Input the expected value for the specified column.

Comparison: The comparison type. For example, Equals, Contains, Starts with, etc.

Result type: The select box has two options:

- *All: All records must pass the specified check.
- *Any: At least one record must pass the specified check.

Match case: Whether it is a case sensitive comparison or not.

Severity: The following keywords denote the severity levels of a problem what was found during the verification.

- *Error
- *Warning
- *Info

Description: Describe this check.

Customize SQL Query Verifier

Specify the SQL Query and Verifications Details

Collection name:

Select statement [? Learn more...](#)

`SELECT * FROM lppdb.ticket`

Example: `SELECT * FROM myDB.myTable`

Print the result set of SQL query to the verification report:

Check the number of records returned for the SQL query [? Learn more...](#)

Expect value:

Comparison:

Severity:

Column value verification conditions [? Learn more...](#)

Column name	Check class	Expected value	Comparison	Result type	Match case	Severity	Description
Example id	Numeric	6	Equals	All	False	Error	All ids should equal to 6
Example name	String	John	Contains substring	Any	True	Error	Names returned should contain 'John'
<input checked="" type="checkbox"/>	<input type="text"/>	<input type="button" value="String"/>	<input type="text"/>	<input type="button" value="Equals"/>	<input type="button" value="Any"/>	<input type="button" value="True"/>	<input type="button" value="Error"/>

Figure 5: Column values

3 Sample

To help understand how to set up the verification checking we have provided an example query and verification details.

3.1 Sample database table lppdb.score

Id (int, key)	Name (varchar)	Score (float)
1	Bob	78
2	Justin	69
3	Peter	75

Simple table of values returned from our example SQL query.

3.2 Steps

The image below contains a number of steps with the details for each step to follow.

Customize SQL Query Verifier

Specify the SQL Query and Verifications Details

Collection name:

Select statement [? Learn more...](#)
 step1

Example: `SELECT * FROM myDB.myTable`

Print the result set of SQL query to the verification report: **step2**

Check the number of records returned for the SQL query [? Learn more...](#) **step3**

Expect value:

Comparison:

Severity:

Column value verification conditions [? Learn more...](#)

Column name	Check class	Expected value	Comparison	Result type	Match case	Severity	Description
Example id	Numeric	6	Equals	All	False	Error	All ids should equal to 6
Example name	String	John	Contains substring	Any	True	Error	Names returned should contain 'John'
<input checked="" type="checkbox"/> id	<input type="button" value="Numeric"/>	<input type="text" value="3"/>	<input type="button" value="Equals"/>	<input type="button" value="Any"/>	<input type="button" value="True"/>	<input type="button" value="Error"/>	one record's id=3
<input checked="" type="checkbox"/> name	<input type="button" value="String"/>	<input type="text" value="Peter"/>	<input type="button" value="Contains substring"/>	<input type="button" value="Any"/>	<input type="button" value="True"/>	<input type="button" value="Error"/>	one record's name is pete
<input checked="" type="checkbox"/> score	<input type="button" value="Numeric"/>	<input type="text" value="60"/>	<input type="button" value="Greater than"/>	<input type="button" value="All"/>	<input type="button" value="True"/>	<input type="button" value="Error"/>	All record's scores greater

Step1: Specify the SQL Query statement.

Input the select statement
`SELECT * FROM lppdb.score`

Step2: Specify to have all the query results displayed in the verification report.

Set the "print the result set of SQL query to verification report" to yes.

Step3: Verify the number of records returned

Check the number of the records, make sure the amount of the records returned equals to 3.

Check the number of records returned for the SQL query [? Learn more...](#)

Expect value:

Comparison:

Severity:

Step4: Verify the contents of a specific value returned.

Use check the column value

	Column name	Check class	Expected value	Comparison	Result type	Match case	Severity	Description
	Example id	Numeric	6	Equals	All	False	Error	All ids should equal to 6
	Example name	String	John	Contains substring	Any	True	Error	Names returned should contain 'John'
<input checked="" type="checkbox"/>	id	Numeric	3	Equals	Any	True	Error	one record's id=3
<input checked="" type="checkbox"/>	name	String	Peter	Contains substring	Any	True	Error	one record's name is peter
<input checked="" type="checkbox"/>	score	Numeric	60	Greater than	All	True	Error	All record's scores greater

- 1) A record whose id is 3 must exist.
- 2) A record which contains substring "Peter" must exist.
- 3) The score field of all records must be greater than 60.

<End of document>