Unified Functional Testing

DB Check Point

Lesson Objectives

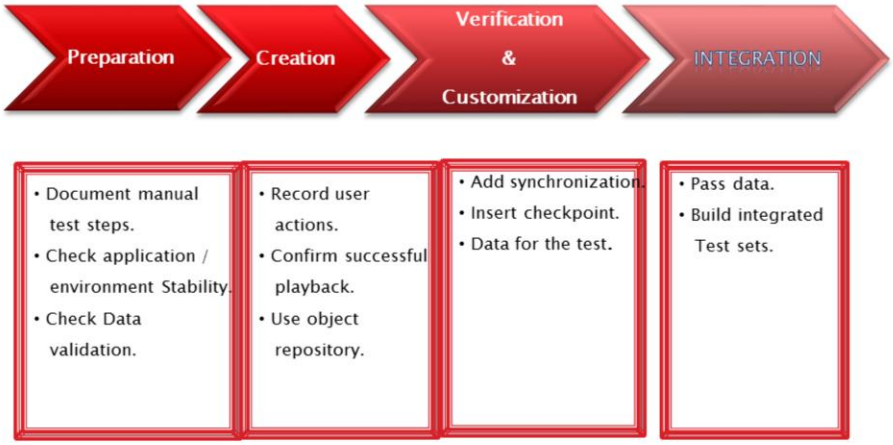
By the end of this Lesson you will be able to:

- Identify the purpose of a database checkpoint.
- Create a Structured Query Language (SQL) statement.
- Create an SQL query using Microsoft Query.
- Create a database checkpoint.
- Parameterize a database checkpoint.

Topics

1. Purpose of DB checkpoint
2. SQL overview
3. Insert a DB checkpoint
4. Parameterize a DB checkpoint

Fourth Phase – Integration



How does the DB check work?

When executing a database checkpoint, the following events occur:

- The test arrives at the database checkpoint step and connects to the database.
- UFT sends a query to the database to retrieve the actual data.
- UFT gathers the result set of the query, which provides the actual data.
- UFT compares the actual data with the expected data, which is stored in the database checkpoint, to determine if the test passed or failed.

Types of SQL Statements

Statement	Description	Syntax
SELECT	extracts data from a database	***SELECT*** *column_name(s)* ***FROM*** *table_name*
INSERT	inserts new data into a database	***INSERT INTO*** *table_name (column1, column2,...)* ***VALUES*** *(value1, value2,...)*
DELETE	deletes data from a database	***DELETE FROM*** *table_name* ***WHERE*** *column=value*
UPDATE	updates data in a database	***UPDATE*** *table_name* ***SET*** *column1=value, column2=value2,...* ***WHERE*** *column=value*

SELECT Statement

- SELECT statement contain 4 clauses:

Clause	Description	Mandatory\Optional?
SELECT	Specify the columns of a table that you want to retrieve.	Mandatory
FROM	Specify the table from which you want to retrieve data.	Mandatory
WHERE	Specify the rows that you want to retrieve from a table.	Optional
ORDER BY	specify sort order for the columns.	Optional

- **SELECT Syntax**

SELECT *column_name(s)* ***FROM*** *table_name* ***WHERE*** *column_name operator value*
ORDER BY *column_name(s) ASC|DESC*

Operators Allowed in the WHERE Clause

Operator	Description
=	Equal
<>	Not equal
>	Greater than
<	Less than
>=	Greater than or equal
<=	Less than or equal
BETWEEN	Between an inclusive range
LIKE	Search for a string pattern that contain the % wild card character
IN	If you know the exact value you want to return for at least one of the columns

Note: If the **WHERE** clause is not specified in a **SELECT** statement, the query retrieves

all rows from the table.

Result Set

Example of SQL Statement:

```
SELECT  customer_id  , external_id , customer_type ,cust_sub_type
FROM customer
WHERE external_id like '8-%'
ORDER BY customer_id desc
```

Result Set of an SQL Statement:

Row#	CUSTOMER_ID	EXTERNAL_ID	CUSTOMER_TYPE	CUST_SUB_TYPE
1	273	8-0099905	B	T
2	272	8-0099904	R	E
3	271	8-0099903	R	T
4	270	8-0099902	R	T
5	269	8-0099901	R	T
6	268	8-1122334488	R	T

Creating a DB Checkpoint

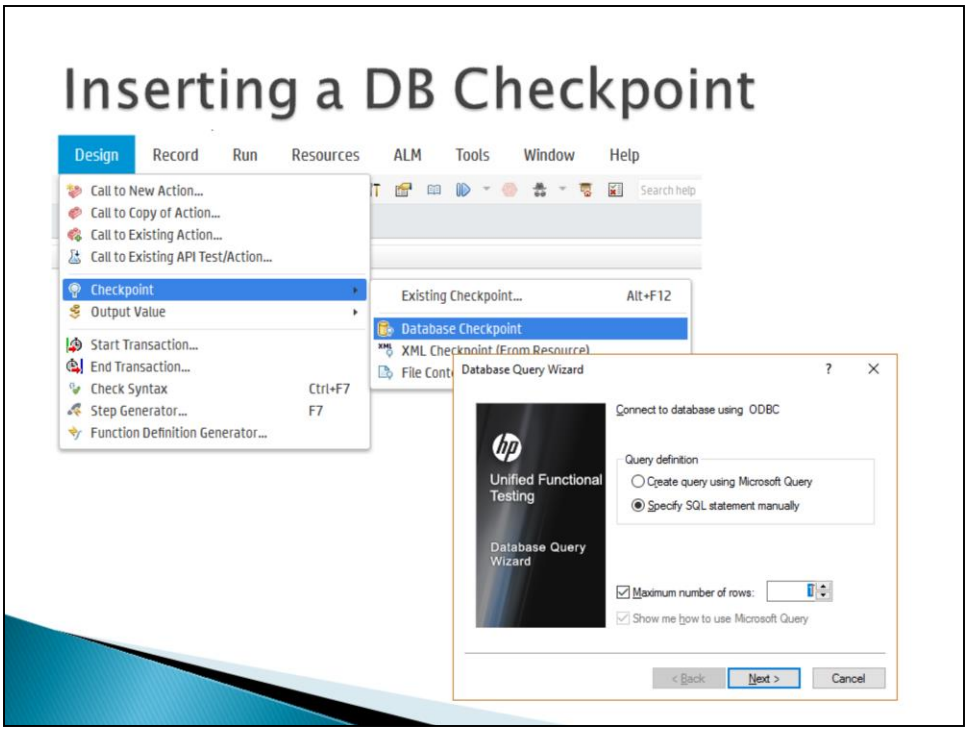
- To create a database checkpoint in a test:
  - Identify the checks that you want to use to verify data in a database.
  - Identify the step in a test after which you want to insert the database checkpoint.
  - Create an SQL query to retrieve data from the database.
  - Specify the expected data.
  - Run the test to verify the database checkpoint.

Example:

```
DbTable("DbTable").Check CheckPoint("DbTable")
```

Common DB Checks

- You can define database checkpoints in a test to check if:
  - The data is saved to the correct tables and columns of the database.
  - The data is updated in the database when a record is inserted, updated, or deleted in the AUT. This process is also known as data persistence.
  - The data entered using the AUT is represented correctly in the database. Some of the properties that you check for are data type, format, length, and spacing.
  - The data entered into the database is not duplicate.

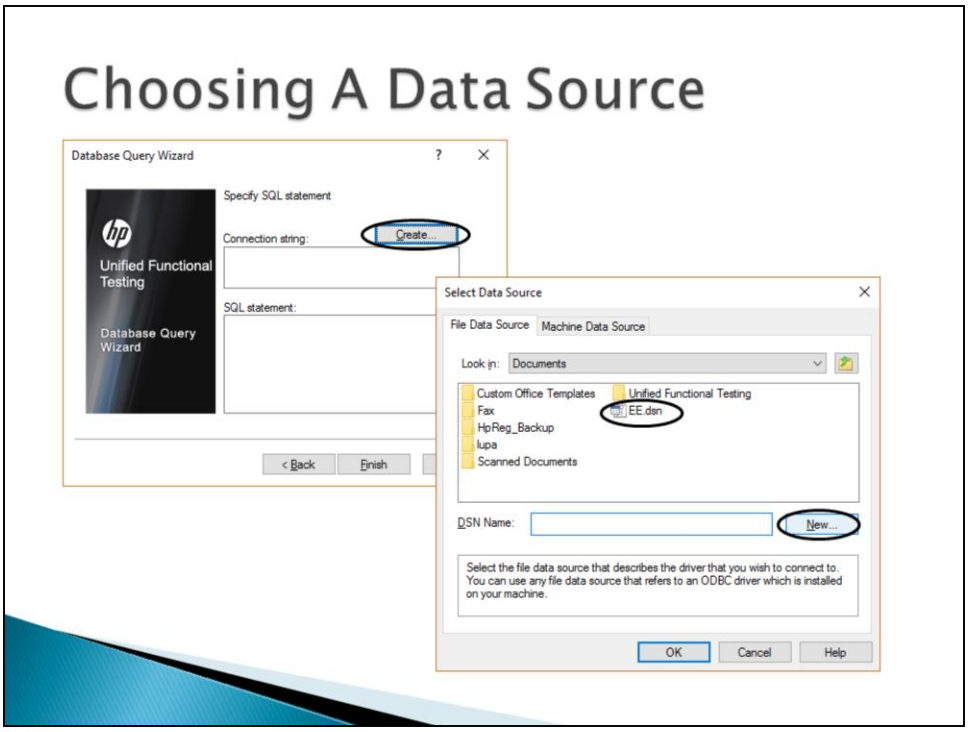


To insert a database checkpoint:

1. From the UFT menu bar, select **DESIGN→CHECKPOINT→DATABASE CHECKPOINT ,**The DATABASE QUERY WIZARD appears.
2. In DATABASE QUERY WIZARD, in the QUERY DEFINITION section, select one of the following options:

   a) **CREATE QUERY USING MICROSOFT QUERY:** To create a query by using the Microsoft Query tool.

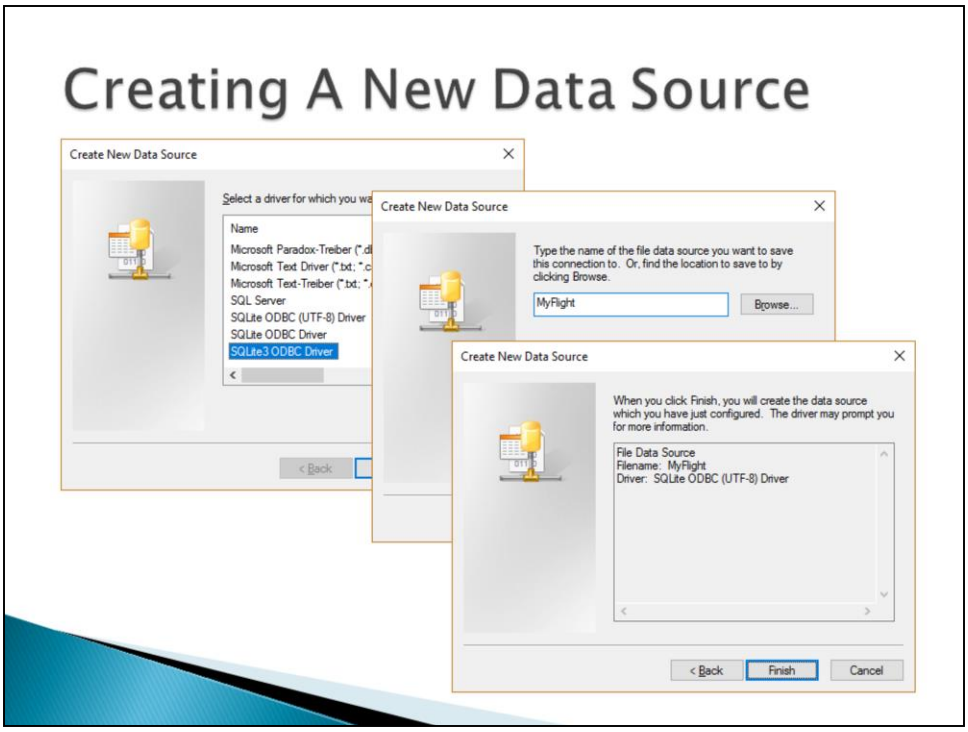
   b) **SPECIFY SQL STATEMENT MANUALLY:** To manually create an SQL query in the wizard.
3. Check the **MAXIMUM NUMBER OF ROWS** check box to set the maximum number of rows that you want to retrieve.
4. Click **NEXT.** The **INSTRUCTIONS FOR MICROSOFT QUERY** dialog box appears. The dialog box provides instructions for using the Microsoft Query tool.
5. Click **OK.**



Define connection string to the required DB and a query.

For first DB check you need to define a DSN containing the DB connection details.

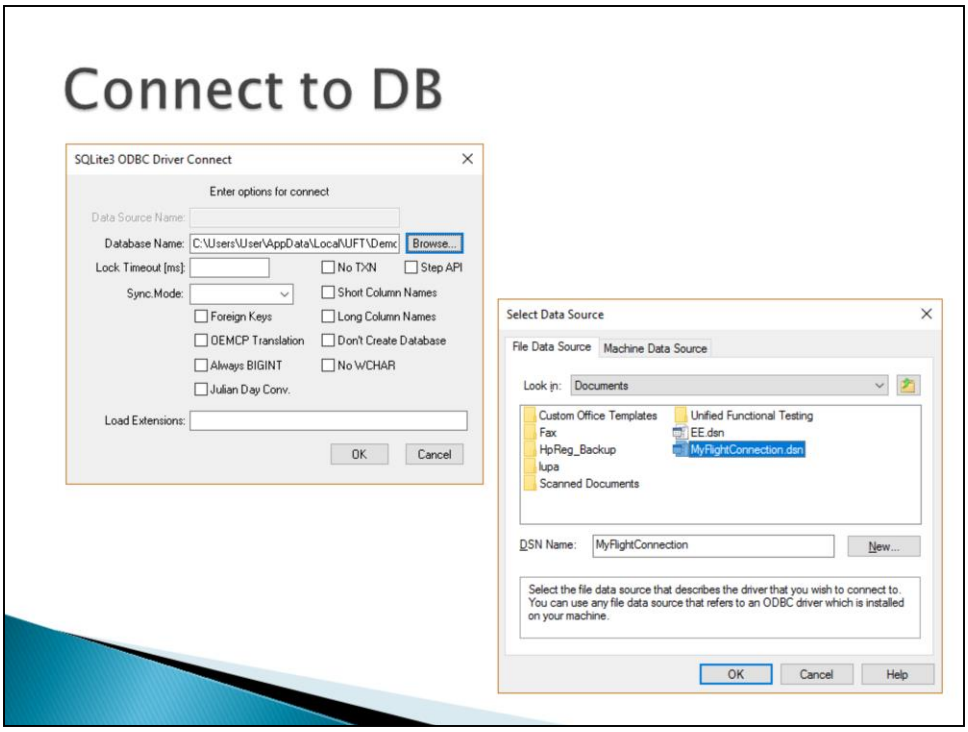
For the next DB checks you can use the DSN you created.



To create a new data source:

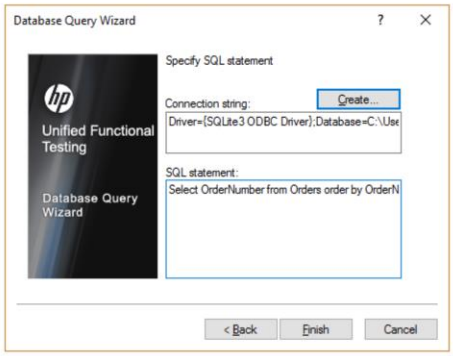
Select the correct driver according to the DB type of your application

Set DSN name

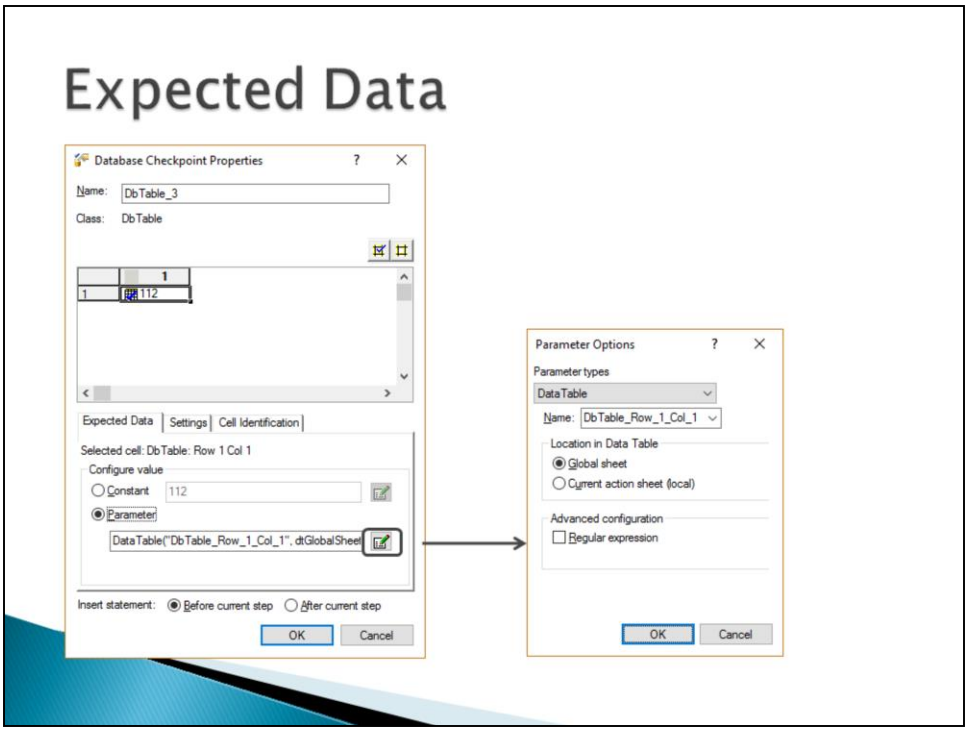


Browse to your DB file location

Write the query



Browse to your DB file location



The Database Checkpoint Properties dialog box displays all the columns and rows that match a query.

You can select specific columns and rows from the dialog box by using the **ADD TO CHECK** or **REMOVE FROM CHECK** buttons. The cells that are checked are the database checkpoint's expected results.

Now you need to define an output parameter for the Order Number field and store the data in the output parameter. The output parameter enables you to verify that the new

order number is entered into the database.

To store the expected data in a parameter:

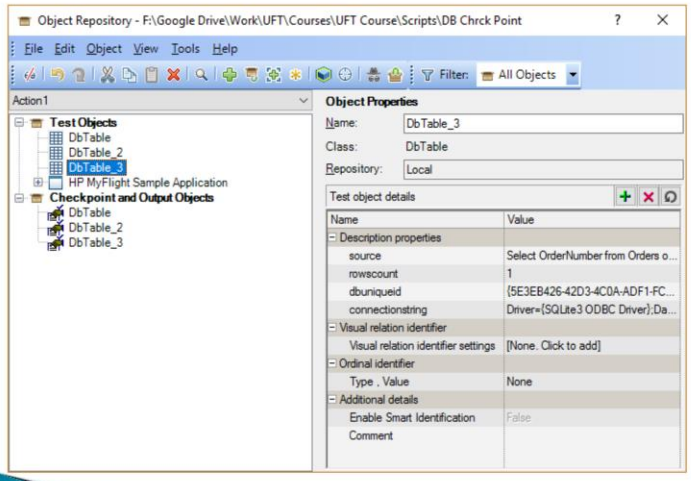
1. In the **DATABASE CHECKPOINT PROPERTIES** dialog box, select the **PARAMETER** option.

2. Click the **PARAMETER OPTIONS** button**. The PARAMETER OPTIONS** dialog box appears.

3. From the NAME list, select a parameter, and click **OK.**

4. In the **DATABASE CHECKPOINT PROPERTIES** dialog box, press on the **OK** button.

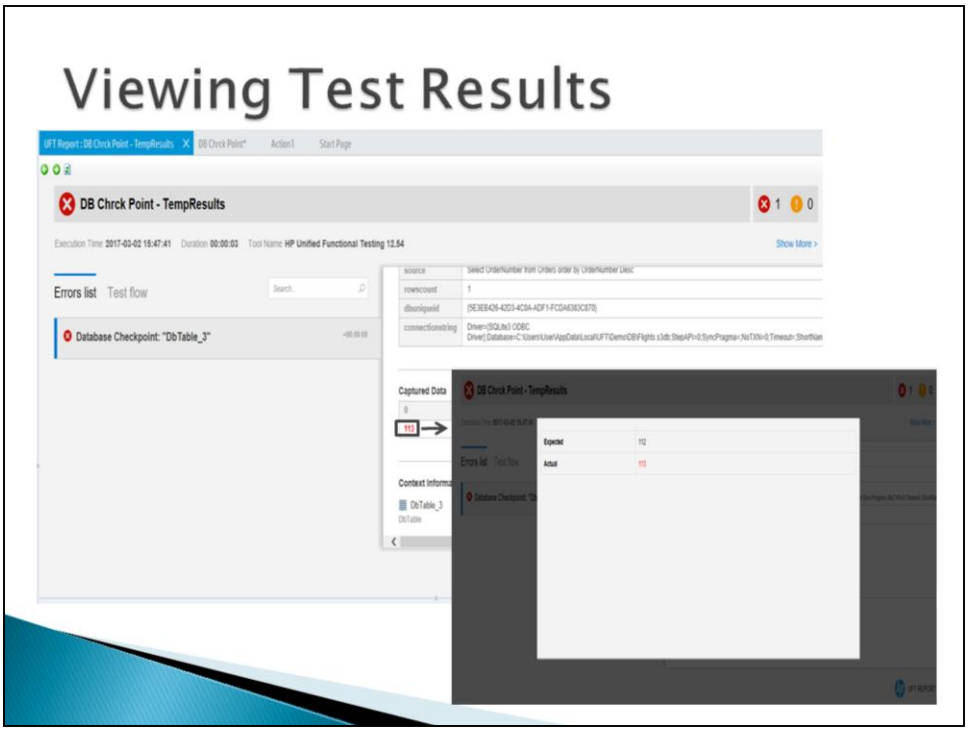
Viewing DB Checkpoint Properties



You can view the properties of a database checkpoint in **OBJECT REPOSITORY**, which you open from the UFT **RESOURCES** menu.

The properties of a database checkpoint are:

- **SOURCE:** Specifies the SQL query used to retrieve data from the database.
- **ROWSCOUNT:** Specifies the number of rows retrieved by the query.
- **DBUNIQUEID:** Specifies the unique identification number of the database.
- **CONNECTIONSTRING:** Specifies the connection string used to connect to the database.



After you create a database checkpoint, you can run the test and view the test results in the TEST RESULTS window. To view detailed results for a test,select the database checkpoint in the TEST SUMMARY tree.

If a test succeeds, the test result reports PASSED in the right pane of the TEST RESULTS

window and shows the evaluated fields. If a test fails, the test result reports FAILED in

the right pane of the TEST RESULTS window.

If a test fails, an additional area in the right pane displays the columns that failed. You can click the **NEXT MISMATCH** button to view the instances of mismatches between the expected value and the actual value. You can click the **COMPARE VALUES** button to view the expected values stored in the database checkpoint and the actual values stored in the database.

What's Next?

- Review Questions
- Exercise
- Next Lesson
  - The next lesson in the course is:
    Handle Exceptions

End of Lesson