Tutorial

A step by step walkthrough of QuerySurge™
Welcome to QuerySurge

QuerySurge is a testing tool from the developers at Real-Time Technology Solutions that automates the testing of Big Data implementations, Data Warehouses, the ETL process, Data Migrations, Database Upgrades, System Upgrades, Interface Testing and BI reports. By analyzing and pinpointing any differences QuerySurge ensures that the data extracted from sources remains intact in the target, and complies with transformation rules. QuerySurge ensures that data extracted from the source remains intact in the target by analyzing and pinpointing any undesired differences. QuerySurge is an essential asset to every data testing process.

Additional information about other QuerySurge features can be found online under Help.

Resources

QuerySurge Help is built into QuerySurge and provides immediate answers to questions that may arise during your usage of QuerySurge. Help can be found in the top toolbar in text form or in the bottom panel by the Help icon.

Knowledge Base and Community Forums provide a hub for extensive QuerySurge information and the answers to the most frequently asked questions about QuerySurge. To access the QuerySurge Knowledge Base, click here >>

- Video Library: The video library provides tutorials including demos and webinars on the usage QuerySurge usage. To access Video Library click here: http://www.queriesurge.com/product-tour/video-library
- Contact Us: The Contact Us page provides a form where you can submit comments, questions, or suggestions. http://www.queriesurge.com/company/contact-us

Note: This tutorial requires the installation of 'Tutorial + Sample Data’ in the QuerySurge installer. This component can be selected during the initial QuerySurge installation. If you are using the Trial in the Cloud the Sample Data is already installed. If you are using the Trial Download please select this option during installation.

Installation Requirements

The QuerySurge application server, the QuerySurge database server and QuerySurge Agents may be installed either on a single box or on separate boxes for trial installation. For increased performance, it is better to have more memory and more disk space. Deploying to a distributed environment with multiple boxes (or VM images) will positively impact the query size, data volume and QuerySurge performance that can be handled during the trial. For additional information on system requirements, click here QuerySurge System Requirements.
# Table of Contents

Welcome to QuerySurge ........................................................................................................... 2  
   Resources ......................................................................................................................... 2  
Tutorial .................................................................................................................................. 5  
   The Story ........................................................................................................................... 5  
   Overview ........................................................................................................................... 7  
First and Subsequent Logins to QuerySurge ....................................................................... 8  
Create QueryPairs using the Query Wizard for Column-Level Comparison ...................... 11  
Execute a QueryPair that fails ............................................................................................... 26  
Create and Execute a Test Suite ............................................................................................. 30  
Running a Detailed Scenario Report ...................................................................................... 36  
Create a QueryPair ............................................................................................................... 38  
Create a QueryPair using a Reusable Snippet ....................................................................... 44  
Add a Reusable Query Snippet to a QueryPair ..................................................................... 49  
Create a Test Suite and Schedule an Execution Scenario ...................................................... 52  
Create QueryPairs using the Query Wizard for Table-Level Comparison ............................ 60  
Review of Data Health Dashboard ......................................................................................... 71  
Summary ............................................................................................................................... 74  
Deleting the Tutorial Data ...................................................................................................... 75  
Appendix ................................................................................................................................. 78  
   Documentation ................................................................................................................ 78  
   Data Warehouse Testing ..................................................................................................... 78  
   The QuerySurge Testing Process ....................................................................................... 80  
   About the QuerySurge Architecture .................................................................................. 82  
About RTTS (developers of QuerySurge) ............................................................................... 85  
About QuerySurge™ .............................................................................................................. 85
Important Terms

**QueryPair**
A pair of SQL queries where one query retrieves data from a source file or database and another SQL query retrieves data from a target database, data warehouse or data mart.

**Agent**
Performs the query tasks. Agents execute queries against source and target data stores, and return the results to the QuerySurge database.

**Design-Time Run**
Allows you to execute a QueryPair test to make sure that it behaves as you expect. This is not intended for actual test execution purposes.

**Query Snippet**
Reusable piece of SQL code that can be embedded in one or many queries. The purpose of a Snippet is to minimize the number of places you need to make changes on the same code in different SQL calls.

**Test Suite**
A collection of QueryPairs used for execution. This feature allows you to group your querypairs execution purposes that is independent of your “Design Library” organization.

**Scenarios**
A collection of Suites that are scheduled for execution.

**Query Wizard**
A tool that allows you to generate QueryPairs automatically, requiring no manual SQL coding. The Query Wizard generates tests that can cover about 80% of all data in a data warehouse automatically.¹

**Widgets**
Project Widgets give you a real-time view into your project progress at all levels, from QueryPair development to execution and results.

**Command Line Integration**
Provides the ability to schedule Test Suites to run using Windows Task Scheduler or integrate with both Data Integration (ETL) and Continuous Build systems.

¹ A recent poll conducted by RTTS on targeted LinkedIn groups found that 80% of columns in data warehouse tables have no transformations.
QuerySurge

Tutorial
Consider This...

ZCity is a popular electronics corporation that is in the market for a new data warehouse so that corporate business personnel can assess the activities that are occurring throughout their sales regions. ZCity recently merged with one of its competitors, XMart, and now has two sales teams. The sales system environment for ZCity (a source database) resides on a MySQL database. The sales system environment for XMart resides on an old IBM mainframe system that can only output files (source files).

ZCity has decided that a data warehouse that runs nightly incremental ETLs will help them make more effective business decisions. The business analysts have created a mapping document that maps out each source field to its corresponding target field. This includes any data transformations that the business deems necessary to be entered into the warehouse. The corporate IT team has completed development of a MySQL data warehouse (the target database) and an ETL process that includes transformation logic (from the mapping document), to load the source data from each source into the data warehouse.

The business’ Quality Assurance team has been tasked with verifying the transformation logic and has purchased QuerySurge to validate 100% of the data being transformed from source to target. The Quality Assurance Manager has planned for the Quality Assurance team to review the mapping documents and build test cases, from the same logic that the developers used when creating the ETL code. The QuerySurge term for test cases is: QueryPairs. The team is confident that this approach will not only cover testing 100% of the data, but will also prove the mapping documents are comprehensive.

Their idea is that if two separate entities, the Development team and the Quality Assurance Team, are building queries separately and can attain the same results, then there is a good chance there will be no defects into the production environment.

That’s the plan, now, it’s time to get testing!
Overview

You will perform the following tasks in the tutorial. It is recommended that you use two monitors. It also might be easier to print the tutorial.

- Review the new Query Wizard and create a new Column-Level Comparison QueryPair
- Review existing and create new QueryPairs
- Review existing and create new Reusable Query Snippets
- Execute “Design-Time Runs”
- Investigate “Design-Time Run” failures
- Schedule and execute a Scenario
- Generate and review reports
- Generate QueryPairs using the “Query Wizards”
- Review Widgets

The tutorial is comprised of the following assets:

1. ZCITY Source
   - MySQL sales database (3 tables)
   - 27 QueryPairs
   - 1 Test Suite including all ZCITY QueryPairs
   - 1 Reusable Query Snippet

2. XMART Source
   - Mainframe sales CSV files (4 files)
   - 31 QueryPairs

3. Data Warehouse Target
   - MySQL data warehouse (Dimensional Model)
     - 5 dimension tables
     - 2 fact tables

4. Mapping and Model Document (printing and review of this document are recommended)

   The document can be downloaded here:

   - ZCITY mappings and data model
   - XMART mapping and file data model
   - Data warehouse data model
First and Subsequent Logins to QuerySurge

Upon QuerySurge installation one Administrative user is created for you to get started, or an Administrator has already provided you with your own login credentials.

**Note:** If you have already logged into QuerySurge, please review this section

1. On the installation machine, click **Start**
2. Under the **Start** menu select All Programs → QuerySurge → Launch QuerySurge
3. QuerySurge will launch in your default browser
4. If your QuerySurge install is local, enter your credentials or use the credentials admin/admin. If you are doing a Cloud trial, use the credentials clouduser/clouduser, click **Login**, and click the ‘Login’ button.
5. If this is your first time launching QuerySurge, a splash screen will display. If this is not the first time launching QuerySurge, the “Setup Wizard” is always available to you on the top toolbar.

Click “Setup Wizard” on the top toolbar.
Points of interest:

- **Setup Wizard** panel on the right
  - The panel is divided into 3 sections: 1. Configuring an Agent; 2. Creating a Connection; Optional Tasks
  - The panel shows that an Agent has been installed and enabled (The green circle with the check mark next to tasks ‘Install an Agent’ and ‘Enable your Agent in QuerySurge’ signifies a completed task)
  - Although connections have been established to the tutorial data, the Setup Wizard informs you that a connection has not been established to your databases (the red exclamation point signifies an incomplete task)
  - The ‘Create a Connection’ link brings you to the connection wizard
  - The ‘Optional Tasks’ section allows you to easily navigate to documentation and videos or to create additional connections

- **Data Health Dashboard** panel in the middle
  - The center panel displays QuerySurge’s Data Health Dashboard and Data Widgets, which is a configurable dashboard that provides a view into project information (more on this later)

6. Collapse the ‘Setup Wizard’ by clicking the button located on the top right corner
Create QueryPairs using the Query Wizard for Column-Level Comparison

A QueryPair is what QuerySurge refers to as a test case – two queries with the following characteristics:

- A SQL query that retrieves data from a source file or database and
- Another SQL query that retrieves data from a target big data store, data warehouse, data mart or database

This is a QueryPair.

The Query Wizard is a feature that allows you to generate QueryPairs automatically requiring no SQL coding. It is a fast and easy way to create QueryPairs for both skilled team members and beginners with less experience who want to increase testing speed.

Based on a LinkedIn poll of Data Warehouse experts, on average 80% of columns in a typical data warehouse have no transformations at all. The Query Wizards were developed to generate tests that can cover approximately 80% of all data in a data warehouse quickly and without writing SQL code. The Wizards are user friendly and provide amazing results for both beginners and experienced testers.

The Query Wizard can generate QueryPairs for:

- **Column-Level Comparison:** this is great for Big Data stores and Data Warehouses where tables will have some columns with transformations (20% on average) and some columns with no transformations (80% on average). The Column-Level Comparison is for the specific columns with no transformation. Columns with transformation will require SQL coding.

- **Table-Level Comparison:** this is great for data migrations and database upgrades with no transformation at all. Many tables can be compared simultaneously and quickly.

- **Row Count Comparison:** this works with all - Big Data stores, Data Warehouses, Data Migrations and Database Upgrades. Many tables can be compared simultaneously and quickly.
A folder can be created that contains all of the QueryPairs generated. These QueryPairs can also be added to a new Test Suite so they can be easily run together.

**Note:** This part of the tutorial illustrates the use of the Query Wizard for creating a QueryPair with Column-Level Comparisons

1. Click the dropdown menu in the “Design Menu” icon in the top toolbar and select “Launch Query Wizard”

2. Click the **Next** button.

3. Select “XMART_SOURCE – CUSTOMER” from the “Source Connection” drop down menu

4. Select “DW_TARGET” from the “Target Connection” drop down menu.

- **Points of interest:**
  - Metadata (or data about our data) pre-populates for the selected databases

5. Click **Next**

6. Keep the default “Column-Level Comparison” selected under the “Comparison Type” screen and click **Next**
Points of interest:

- **Column-Level Comparison** validates data points in the selected columns and allows for the addition of Filtering (WHERE Clause) and Sorting (ORDER BY)
- **Table-Level Comparison** validates each data point in the source table to its corresponding data point in the target table
- **Row Count Comparison** validates the number of rows in each of the selected tables

7. Click **Next**
Points of interest:

- Schemas available for selection are retrieved from the source and target databases that you previously selected
- One, many, or all schemas can be selected
- The ‘information_schema’ is the internal schemata used by MySQL databases

8. Select ‘PUBLIC’ under “Source Schemas” pane and ‘dw’ under “Target Schemas” and click Next
9. Select and drag ‘PUBLIC.XMART_CUSTOMER’ under “Source Tables” and drop it into it’s corresponding “Target Table” ‘dw.customer_dim’
Points of interest:

- As a source table is dropped onto its corresponding target table it’s ‘Query Mapping’ appears in the left hand column (if desired, you can double click to rename it)
- Mapped tables show a line connecting the source and target table with the selected ‘Query Mappings’ line highlighted green
- Auto generated SQL appears in the ‘Source Query’ and ‘Target Query’ panes at the bottom of the Query Wizard window

10. Select and drag the “CUSTOMER ID” under “Source Column” and drop it into its corresponding target column “SOURCE_ID”.
11. Select and drag the “FIRST” under “Source Column” and drop it onto “FIRST” in its corresponding “Target Column”.
12. Select and drag “LAST” under “Source Column” and drop it “LAST” in the corresponding ”Target Column”.
13. Select and drag “EMAIL” under “Source Column” and drop it into “EMAIL” in its corresponding “Target Column”.
14. Select and drag “PHONE NUMBER” under “Source Column” and drop it into “PHONE” in its corresponding “Target Column”.

Points of interest:

- As a source column is dropped onto its corresponding target column the auto-generated SQL will change to reflect your selections.
- Column relationships show a line connecting the source and target columns with the currently selected relationships line highlighted green.
- Relationships can be removed by clicking the ‘Remove Relationship’ icon.

15. At the middle of the screen under “Filtering and Sorting” click the “Add New Criteria” icon.
16. Under “Table” in the newly created Criteria Field click the dropdown menu and select “Source”.
17. Under “Column” click the dropdown menu and select “CUSTOMER_ID”.
18. Under “Operation” click the dropdown menu and select “Sort Direction”. Under “Value” click the dropdown menu and select “Ascending”. Click the “Add New Criteria” icon.
19. Under “Table” click the dropdown menu and select “Target”. Under column click the dropdown menu and select “SOURCE_SYSTEM”.
20. Under “Operation” click the dropdown menu and select.
21. Under “Value” type “XMART”.
22. Click the “Add New Criteria” icon.
23. Under “Table” click the dropdown menu and select “Target”.

```
SELECT "CUSTOMER_ID", FIRST, LAST, EMAIL
FROM PUBLIC.XMART_CUSTOMER;
```

```
SELECT SOURCE_ID, FIRST, LAST, EMAIL
FROM dev.customer_dim;
```
24. Under “Column” click the dropdown menu and select “SOURCE_ID”.
25. Under “Operation” click the dropdown menu and select “Sort Direction”.
26. Under “Value” click the dropdown menu and select “Ascending”.

Points of interest:

- Added criteria appear in the ‘Filtering (WHERE Clause) and Sorting (ORDER BY)’ section
- Auto generated SQL based on added filter criteria appear in the query panes
- Criteria can be removed by clicking the ‘Remove Criteria’ icon

27. Click Next
29. Under “Parent Folder” click the dropdown menu and select “QueryPairs”.
30. Type “WIZARD – COLUMN-LEVEL” in the “Folder Name” field
31. Under “Scheduling” select “Yes”.
32. Type “WIZARD – COLUMN-LEVEL” in the “Suite Name” field
Points of interest:

- The **Query Wizard** will automatically create the test assets:
  - A new folder in the ‘Design Library’ with the name entered in the ‘Folder Name’ field
  - All **QueryPair(s)** for testing the tables selected
  - A new **Test Suite** in the ‘Scheduling’ module with all of the generated **QueryPair(s)**

33. Click **Next**
Points of interest:

- A review screen appears with the details for your *Query Wizard* execution

34. Click **Create**
Points of interest:

- A test folder named ‘WIZARD – COLUMN-LEVEL’ was created in the ‘Design Library’ module
- A Test Suite named ‘WIZARD – COLUMN-LEVEL’ was created in the ‘Scheduling’ module
- One QueryPair was created

35. Click OK
Execute a *QueryPair* that passes

The QueryPair created in the previous chapter will be utilized for this section of the tutorial

1. In the “Design Library” panel to the left select “QueryPairs”

   ![Diagram of QuerySurge UI]

   **Points of interest:**
   - *QuerySurge* utilizes a folder structure similar to Microsoft Windows Explorer
   - Multiple sub-folders can be contained within folders
   - The left panel allows for simple navigation to folders
   - The tutorial has two main folders, one for each source system, and the folder you created while running the *Query Wizard*

2. Double click the “WIZARD - COLUMN-LEVEL” folder in the center panel
Points of interest:

- The QueryPair contained within the ‘WIZARD - COLUMN-LEVEL’ folder are displayed
- In this case, there is only one QueryPair that was created during the previous chapter

3. Double click the “Data (Column): PUBLIC.XMART_CUSTOMER - dw.customer_dim” QueryPair
Points of interest:

- The QueryPair was created to validate that the source customer data was loaded into the data warehouse. (See mapping 1.01 – 1.05 of the tutorial mapping document - a link to the document is on the ‘Setup Wizard’)
  - This QueryPair is validating the direct map (no logical transformation) of the data from Source, XMART, to target data warehouse
- The ‘QueryPair’ tab contains
  - ‘Source Query’ and ‘Connection’ – for storing the query with a specific database connection
  - ‘Target Query’ and ‘Connection’ – for storing the query with a specific database connection
  - ‘Save’ icons – for saving the source or target query
- The ‘Properties’ tab contains
  - QueryPair name, description, and mapping fields
  - Data Type Checking – Ability to broadly check data types (e.g. text vs numeric) or convert all types to varchar (String) before comparison
  - Row Count Options – to set reporting for row count differences
  - Shared Key Column Definition – to set a column from either the source or target query as key
  - Column Threshold Options – Ability to allow fields to pass if within a user defined threshold
  - Duplicate Row Options – to set whether or not QuerySurge uses a comparison algorithm to manage duplicate rows in your data. (Note: this is not a test for duplicates)
- The ‘Design-Time Run’ tab contains
  - Detailed results for each QueryPair. This tab allows you to execute a QueryPair test run to make sure that it behaves as you expect
- The ‘History’ tab shows the history of changes to the QueryPair
- The ‘DTR History’ tab shows pass/fail information from previous ‘Design-Time Runs’ (currently empty as no runs have been executed yet)

4. Click the “Design-Time Run” tab
Points of interest:

- The first time clicking on the ‘Design-Time Run’ will indicate that a run has not occurred previously for a QueryPair
- A specific Agent, if desired, can be selected in the drop down menu (not required)

5. Click the “Run” button to execute the QueryPair

Points of interest:

- When executing a ‘Design-Time Run’, QuerySurge goes through the following phases:
  - Loading – QuerySurge is loading the SQL pairs onto the Agent for execution
  - Running – The Agent is running the target and source queries
  - Analyzing – QuerySurge is comparing the results
- The QueryPair passes
- Both source and target returned 1250 rows, and all rows matched from source to target
- QuerySurge displays the following information about the last ‘Design-Time Run’
  - Source Rows – number of source rows returned
  - Target Rows – number of target rows returned
  - Row Count Difference – difference between the number of source and target rows returned
  - Failed Row Count – number of rows with data differences
  - Non-Matching Source Rows – number of source rows not found in the target (based on key column(s))
Non-Matching Target Rows – number of target rows not found in the source (based on key column(s))

- ‘Connections’ and ‘Query Performance’ metrics are displayed

6. Click “View Query Results”

![QueryPair Results Window]

Points of interest:

- The “QueryPair Results” window contains five tabs
  - Source Tab – displays the source result rows
  - Target Tab – displays the target result rows
  - Failures Tab – displays data failures (not populated in this example as no rows failed)
  - Source Query Tab – displays the source query
  - Target Query Tab – displays the target query

- Since this is a passing QueryPair, the source and target tabs display the same data (note: the column headers have been aliased)

7. Close the query results window

For more information on QueryPairs, please visit Design Library ➔ Working with QueryPairs in the Help guide
Executive a QueryPair that fails

Use QuerySurge to execute a QueryPair that fails.

1. In the “Design Library” tree (left panel), navigate to the folders ‘QueryPairs -> ZCITY -> ORDERS_FACT’

   ![QuerySurge interface showing the Design Library]

   **Points of interest:**
   - The QueryPairs contained within ‘ORDERS_FACT’ folder are displayed

2. Double click the ‘ZCITY – STATUS’ QueryPair.
Points of interest:

- The ‘ZCITY – STATUS’ QueryPair was created to validate that the source SALE_STATUS column was loaded into the data warehouse. (See tutorial mapping document link on the ‘Setup Wizard’)
  - This QueryPair is validating that the SALE_STATUS from source, ZCITY, to target data warehouse is loaded with the following logic:
    - Populate target ORDERS_FACT.STATUS using the following logic: if the Sale_Status value in the Sale table is ‘Incomplete’, then change to ‘Pending’. Otherwise leave as is

3. Click the “Design-Time Run” tab
4. Click ‘Run’ to execute the QueryPair

Points of interest:

- The ‘ZCITY – STATUS’ QueryPair fails. (if one or more fields do not match, then QuerySurge fails the QueryPair)
- Both source and target returned 6,919 rows
- 1,373 rows failed

5. Click View Query Results
6. Click the “Failures” tab
Points of interest:

- The rows with left-pointing green arrows are source rows.
- The rows with right-pointing blue arrows are target rows.
- Notice the source rows contain the value ‘Pending’ and the target rows contain the value ‘Incomplete’

7. Click the “Side By Side View” icon.
Points of interest:

- The ‘Side By Side View’ allows for easier viewing of your failures
  - Source rows on the left and target rows on the right
- All failures were ‘Data Failures’, as indicated by a red flag
  - If there were ‘Non-matching Rows’ they would appear with a yellow flag
- Failures are shown with a red background

8. Close the “ZCITY – STATUS” query results window.
Create and Execute a Test Suite

The Test Suites feature of QuerySurge lets you organize your QueryPairs in Test Suites - collections of QueryPairs for execution. This feature allows you a different level of organization for execution purposes that is independent of your ‘Design Library’ organization.

1. Click the “Scheduling” icon located on the bottom panel

Points of interest:

- A Test Suite has been created for you called ‘ZCITY – ALL’
  - ‘ZCITY – ALL’ contains all the QueryPairs associate with the ZCITY source
- The ‘WIZARD – COLUMN-LEVEL’ suite was created during this tutorial
- Left Panel – displays all the available Test Suites that can be run
- Middle Panel – displays tests associated with a Test Suite (currently no Test Suite is selected)
- Right Panel – displays your ‘QueryPair Library’ tree

2. Click “Create Suite”
Points of interest:

- The “Create New Test Suite” dialog contains:
  - Test Suite Name – a name for your suite
  - Description – a description for your suite

3. Enter “ZCITY – CUSTOMER_DIM” in the “Test Suite Name” field
4. Enter “To test the customer dimension fields” in the “Description” field
5. Click the Save.
Points of interest:

- Your new Test Suite is currently empty

6. Click “+” next to “QueryPairs” in “QueryPair Library” panel on the right and select “ZCITY”.

7. Drag and drop the “CUSTOMER_DIM” folder to the ZCITY – CUSTOMER_DIM” panel in the middle

Points of interest:

- The ‘ZCITY - CUSTOMER_DIM’ Test Suite now contains all QueryPairs from the ‘CUSTOMER_DIM’ folder

8. Click Save in the upper left corner of the middle panel to save your Test Suite

9. Right click on the “ZCITY – CUSTOMER_DIM” under “Test Suite” on the left panel and select “Run Now” which will execute the Test Suite as a single suite Scenario

10. QuerySurge will toggle you over to the ‘Run Dashboard’ and the Test Suite will execute
Points of interest:

- Left Panel – ‘Scenario Queue’ which includes all Scenarios including those previously run
  - The executed Scenario was created for you with a single test suite named ‘ZCITY – CUSTOMER_DIM’
  - Notice QuerySurge appended a date/time stamp to the end of the Scenario name
- Upper Middle Panel – the currently selected Scenario’s Test Suites and their progress
- Upper Right Panel – additional information and metrics
- Lower Middle/Right Panel – graphs of progress
- After execution has completed, review historical data by selecting a Scenario in the left panel

11. Double click the Test Suite “ZCITY – CUSTOMER_DIM” to see detailed data about the Test Suite execution results
12. Click on the “ZCITY – EMAIL” QueryPair
Points of interest:

- Notice the lower panel has a layout similar to the “Design-Time Run” execution results
- Lower panel contains
  - Source tab – shows the rows returned by the source query
  - Target tab – shows the rows returned by the target query
  - Failure tab – shows failures resulting from the source and target comparison
- Right panel contains
  - Source Row Count – displays number of source rows
  - Target Row Count – displays number of target rows
  - Row Count Difference – displays number of extra rows in source or target
  - Failed row count – displays number of data failures

- You can export results to Excel, CSV file, or an XML document using the ‘Export to’ dropdown menu located below the source tab on the lower pane

13. Close the “QueryPairs” Test Suite results window

For more information on Execution, please visit Scheduling → Running a Test Suite in the Help guide
Running a Detailed Scenario Report

The “Reporting Module” shows the formal reports offered by QuerySurge. Some reports have configurable options which are found once the report type has been selected and opened. Once executed, reports can be saved as a PDF or exported to Excel.

1. Click the “Reporting” icon in the bottom panel

Points of interest:

- A list of reports is shown in the ‘Reporting Center’ (left panel)

2. Select the “Scenario Detail” report under “Data Intelligence Reporting”

3. Click Open Report

4. Under “Select a Scenario” select ‘ZCITY – CUSTOMER_DIM dropdown list (a date time stamp was appended to the end of the Suite name from the previous exercise)

5. Click the Run Report
Points of interest:

- In the screenshot above the “Scenario Outcome” is “FAILED” since one QueryPair has failed
- The “Test Results” displays the number of passed and failed QueryPairs
- Additional information is displayed in the “Overview” section

6. Close the “Scenario Detail” report tab

For more information on Reports, please visit Reporting in the Help guide
Create a *QueryPair*

Use QuerySurge to create a new *QueryPair*.

1. Click the “Design Library” icon located on the bottom panel
2. In the left panel under “Design Library” select QueryPairs and select “XMART” and then “ADDRESS DIM”
3. Click “Create New QueryPair”
Points of interest:

- The “Create a New QueryPair” dialog contains:
  - QueryPair Name field – a name for your QueryPair
  - Description field – a description for your QueryPair
  - Mapping field – for reference back to a mapping document if applicable. (Data mapping is a process used in data warehousing by which different data models are linked to each other. Mappings are typically used for transformations in the ETL process or for consolidation of multiple databases and/or redundant columns.)
  - See the Tutorial Mappings document, which can be found by clicking on the Setup Wizard link at the top and then clicking on the PDF in the Setup Wizard pane, which will open on the right

4. Type “XMART – STREET (SHIPPING)” in the “QueryPair Name” field
5. Type “Extract the Street portion of XMART_CUSTOMER.SHIPPING ADDRESS by pulling all text before the first comma” in the “Description” field (This description was taken from the tutorial mapping document for XMART)
6. Enter “5.05” in the Mapping field
7. Click Save and the new QueryPair tab will open
8. Enter the following under “Source”:

```sql
select
    c."CUSTOMER ID" as source_id,
    substr(c."SHIPPING ADDRESS", 1, instr(c."SHIPPING ADDRESS", ',',)-1) as street
from
    XMART_CUSTOMER as c
order by
    source_id
```

9. Enter the following in the “Target” query text “Editor”:
```sql
select
c.SOURCE_ID as source_id,
a.STREET as street
from
CUSTOMER_DIM as c
left outer join
{
    select
    a.ADDRESS_ID as address_id,
a.STREET as street
    from
    ADDRESS_DIM as a
    where
    a.SOURCE_SYSTEM = 'XMART'
} a
on
c.SHIPPING_ADDR_ID = a.ADDRESS_ID
where
c.SOURCE_SYSTEM = 'XMART'
order by
c.SOURCE_ID
```

10. Under “Connection” select “XMART_SOURCE CUSTOMER” from the dropdown menu in the “Source” panel.
11. Under “Connection” select “DW_TARGET” in the “Target” panel
12. Click the “Save” icon in both the source and target queries
Points of interest:

- The XMART source is an old IBM mainframe that cannot be accessed directly. However files can be exported from the system into CSV files and then accessed directly.
- Connections to the exported files are created through the connection wizard by selecting ‘Flat File’ as the connection type.
- The flat file must reside in a location available to each Agent.
- Delimited flat files can be queried (see JStels documentation http://www.csvjdbc.com/stels_csv/doc.htm) as source or target data repositories, much like any database.
  - JStels utilizes H2 database technology.
  - H2 is a relational database management system written in Java.
- The ‘XMART – STREET (SHIPPING)’ QueryPair is implemented to validate that the source SHIPPING_ADDRESS column was loaded into the data warehouse correctly (see Tutorial Mapping document link on the ‘Setup Wizard’).
  - This QueryPair is validating that the SHIPPING_ADDRESS column from source, XMART, to target data warehouse is loaded with the following logic:
    - Extract the street name from "XMART_CUSTOMER.SHIPPING ADDRESS" by retrieving all of the text before the first comma.
13. Click the “Design Run Time” tab.

14. Click Run

15. Click View Query Results
Points of interest:

- The execution results of a flat file are displayed exactly the same as execution results from a database

16. Close the “XMART – STREET (SHIPPING)” query results window
Create a QueryPair using a Reusable Snippet

A query Snippet is a reusable piece of SQL code that can be embedded in one or more queries. The purpose of a Snippet is to make common SQL clauses as reusable as possible. A Snippet does not have to be a syntactically intact query (for example, it might just be a WHERE clause) but there is no reason why it can’t be a complete query (e.g. a reusable sub-query). This feature lets you develop libraries of ‘Reusable Query Snippets’ to be reused in multiple queries. If a Snippet shared by multiple queries needs to change, you can make the change once and all of the queries using the Snippet are changed automatically.

1. Under “Design Library” in the lefthand panel select “QueryPairs” and then select “ZCITY” and then “ORDERS_FACT”
2. Click Create New QueryPair
3. Type “ZCITY – COMMENTS” under “QueryPair Name”
4. Click the Save
5. Under the “Source” Table:

   select
   source_order_id,
   cast(group_concat(comments order by source_saleitem_id SEPARATOR ' | ') as char(100)) as comments
from
(
select
si.SALE_ID as source_order_id,
concat(si.NAME,: ',si.COMMENT) as comments,
si.SALEITEM_ID as source_saleitem_id
from
SALEITEM as si,
SALE as s
where
si.SALE_ID = s.SALE_ID
order by
source_order_id
) main
group by
source_order_id
6. Enter the following under the “Target” table:

```sql
select
    o.SOURCE_ORDER_ID as source_order_id,
    o.COMMENTS as comments
from
    ORDERS_FACT as o
where
    o.SOURCE_SYSTEM = 'ZCITY'
order by
    source_order_id
```

7. In the “Source” table select “ZCITY_SOURCE” from the “Connection” dropdown menu.
8. In the “Target” table Select “DW_TARGET” from the “Connection” dropdown menu.
9. Click the “Save” icon on both the source and target queries.

Points of interest:

- The “ZCITY – COMMENTS” QueryPair is to validate that the source comments column was loaded into the data warehouse (see tutorial mapping document link in the ‘Setup Wizard’)
  - This QueryPair is validating that the COMMENT column from source, ZCITY, to target data warehouse is loaded with the following logic:
- Populate COMMENTS column on target table ORDERS_FACT by concatenating columns from ZCITY source with the following logic:
  - saleitem.NAME + ': ' + saleitem.COMMENTS
  - For multiple comments separate source comments with ' | '

10. In the “Target” table highlight the following text: "o.SOURCE_SYSTEM = 'ZCITY'"

11. Click the “Save selected SQL as Reusable Query Snippet” icon.

---

Points of interest:

- The ‘Create New Reusable Query Snippet’ dialog contains:
  - Reusable Query Snippet Name field – a name for your Snippet
  - Description field – a description for your Snippet

12. Type “ZCITY ORDERS Filter” in the “Reusable Query Snippet Name” field

13. Click Save

14. Click the Save icon under the “Target” table
Points of interest:

- Notice ‘${ZCITY ORDERS Filter}’
- Hover your mouse over the Snippet to resolve, or see the actual query
  - You can click the “Query View” tab at the bottom of the pane to resolve the Snippet as well

Execute a Design Run by selecting the “Design Run” tab then click “View Query Results”

15. Click the “Failures” tab (expand the comments field to view the entire column)
Points of interest:

- Notice on the target side the development team is not using the correct delimiter
  - The target data warehouse is using ‘+’ as its delimiter between comments, but according to the mapping document ‘|’ was the specification


For more information on Reusable Snippets, please visit Design Library ➔ Working with Reusable Snippets in the Help guide
Add a Reusable Query Snippet to a QueryPair

Add a Snippet to an existing QueryPair

1. Under “Design Library” select “ZCITY” and then select “ORDERS_FACTS”
2. Double click “ZCITY - ORDER_DT”
3. Under the “Target” table highlight the following text: “o.SOURCE_SYSTEM = ‘ZCITY’”
4. Click the “Add Reusable Query Snippet” icon

Points of interest:

- The left panel utilizes a folder structure similar to Microsoft Windows Explorer (our Snippets are saved in the root folder, but can be grouped by subfolders manually)
- The top right panel will display the ‘Reusable Query Snippets’ contained within the selected folder once a folder is selected
- The bottom right panel displays the SQL of the selected Snippet

5. Click the “Reusable Query Snippets” folder in the left panel.
Points of interest:

- The top right pane displays the ‘Reusable Query Snippets’ including the one you created earlier

6. Select the “ZCITY ORDERS Filters” folder and click OK.

Points of interest:

- The “ZCITY ORDERS – Filters” replaces the highlighted text with the Reusable Query Snippet
• Now the Snippet is located in two locations. If you need to change that part of the query it can be done in one place instead of two. (What if it were in hundreds of QueryPairs? That would save quite a bit of time)

7. Click Save icon on the target query
Create a Test Suite and Schedule an Execution Scenario

QuerySurge Scenarios are collections of Test Suites that are scheduled for execution. Scenarios can be run immediately after creation, or scheduled for a future date/time, and are viewable in the ‘Run Dashboard’.

1. Click the “Scheduling” icon located on the bottom panel
2. Click the Create Suite.

3. Type “ALL – CUSTOMER_DIM” in the “Test Suite Name” field
4. Click Save
5. Expand the ‘XMART’ folder in the “QueryPair Library” panel on the right
6. Drag and drop the “CUSTOMER_DIM” folder to the “LL – CUSTOMER_DIM” panel in the middle
7. Expand the ‘ZCITY’ folder in the ‘QueryPair Library’ panel on the right
8. Drag and drop the “CUSTOMER_DIM” folder to the “ALL – CUSTOMER_DIM” panel in the middle
Points of interest:

- The “ALL – CUSTOMER_DIM” Test Suite now contains all QueryPairs from the “CUSTOMER_DIM” folders from both “XMART” and “ZCITY” (14 QueryPairs)
- Test Suites can contain a combination of QueryPairs from various sources (XMART and ZCITY) and varying connection types (flat file and database, respectively)

9. Click Save in the upper left corner of the middle panel to save your Test Suite

10. Click the Create Suite

11. Enter “ALL – ORDERS_FACT” in the “Test Suite Name” field

12. Click the Save

13. Expand the “XMART” folder in the “QueryPair Library” panel on the right

14. Drag and drop the “ORDERS_FACT” folder to the “ALL – ORDERS_FACT” panel in the middle

15. Expand the “ZCITY” folder in the “QueryPair Library” panel on the right

16. Drag and drop the “ORDERS_FACT” folder to the “ALL – ORDERS_FACT” panel in the middle
Points of interest:

- The ‘ALL – ORDERS_FACT’ Test Suite now contains all QueryPairs from the ‘ORDERS_FACT’ folders from both ‘XMART’ and ‘ZCITY’ (16 QueryPairs)

17. Click **Save** in the upper left corner of the middle panel to save your Test Suite

18. Click the **Schedule Scenario** in the “Test Suites” panel
Points of interest:

- The ‘Schedule Scenario’ window contains:
  - Scenario Name field – the name of the Scenario
  - All Dates to field – sets all Test Suites to run on a particular date and time
  - All Agents as field – sets all Test Suites to run on a particular Agent
  - A Test Suite schedule area with columns containing:
    - Test Suite – Test Suites for this Scenario
    - Agent – allows you to select a particular Agent
    - Scheduled Date & Time of Run – for selecting a specific date and time for a run
      - Useful for executing tests, e.g. during non-working hours

19. Enter “SCENARIO – XMART/ZCITY – CUSTOMERS AND ORDERS” into the “Scenario Name” field

20. Click the “Add Test Suite”

21. In the “Test Suite” column, click the drop down list and select the “ALL – CUSTOMER_DIM” Test Suite
22. Click the “Add Test Suite” again to add a second Test Suite and select the “ALL – ORDERS_FACT” Test Suite.

23. Click in the “Scheduled Date & Time of Run” field for the “ALL – ORDERS_FACT” Test Suite to enable the selection of a date and time for the run. Then click the “Date” icon.
24. Select “Today” for the date portion, set the time for five minutes from now (your current time), and click the **Save**

25. Click the “Run Dashboard” icon located on the bottom panel.

26. Click the **Refresh** at the bottom of the ‘Scenario Queue’ pane (left pane).

27. Click the “SCENARIO – XMART/ZCITY – CUSTOMERS AND ORDERS” Scenario in the “Scenario Queue” pane (this may take a few seconds to refresh and appear). Observe that the Scenario just created is now executing.

---

**Points of interest:**

- The ‘ALL – CUSTOMER_DIM’ Test Suite executes immediately while the ‘ALL – ORDERS_FACT’ Test Suite remains idle since it was set to execute in the next five minutes.
Points of interest:

- Even when the “ALL – CUSTOMER_DIM” Test Suite completes its execution, the “ALL – ORDERS_FACT” Test Suite remains idle
Points of interest:

- Only after the five minutes have elapsed will the ‘ALL – ORDERS_FACT’ Test Suite execute
Create QueryPairs using the Query Wizard for Table-Level Comparison

The Query Wizard is a tool that allows you to generate QueryPairs automatically. This is great for those with limited or no SQL experience, as well as for experts who are looking for a fast and easy way to create QueryPairs for data migrations, database upgrades, and other database comparisons that do not have transformations. The Query Wizard can generate QueryPairs for Row Count Comparison, Table-Level Comparison, and Column-Level Comparison validations. A folder can be created in the Wizard that contains all of the QueryPairs generated. These QueryPairs can also be added to a new Test Suite so they can be easily run together.

**Note**: This part of the tutorial illustrates the use of the Query Wizard for validating a database upgrade. In this case, the data warehouse was upgraded to a new version ‘Connection DW_UPGRADE’ from an older version ‘Connection DW_OLD’

1. In the “Design Menu” dropdown menu on the top toolbar select “Launch Query Wizard”
2. Click Next under the Query Wizard welcome screen
3. Select “DW_OLD” from the “Source Connection” drop down menu
4. Select “DW_UPGRADE” from the “Target Connection” drop down menu
Points of interest:

- Metadata (or data about our data) pre-populates for the selected databases

5. Click **Next**

6. Select the “Table-Level Comparison” radio button in the “Comparison Type” section of the *Query Wizard* window

Points of interest:

- *Column-Level Comparison* validates data points in the selected columns and allows for the addition of Filtering (WHERE Clause) and Sorting (ORDER BY)
- *Table-Level Comparison* validates each data point in the source table to its corresponding data point in the target table
- *Row Count Comparison* validates the number of rows in each of the selected tables

7. Click **Next**
Points of interest:

- Schemas available for selection are retrieved from the source and target databases that you previously selected
- One, many, or all schemas can be selected
- The ‘information_schema’ is the internal schemata used by MySQL databases

8. Select “dw_old” (the location for the data warehouse data before the upgrade) for the source schemata and “dw” (the location for the data warehouse data after the upgrade) for the target schemata

9. Click Next

10. Drag each individual source table and drop it onto its corresponding target table, scroll down in the ‘Source Tables’ and ‘Target Tables’ windows if needed
Points of interest:

- As each individual source table is dropped onto its corresponding target table its Query Mapping appears in the left hand column and can be renamed if desired
- Mapped tables show a line connecting the source and target tables with the selected Query Mapping line highlighted green
- Auto generated SQL appears in the ‘Source Query’ and ‘Target Query’ panes at the bottom of the Query Wizard window

11. Click Next
12. Select the “Create a New Folder” radio button in the ‘Design Library’ section of the Query Wizard window
13. In the “Parent Folder” drop down menu, select the “QueryPairs” folder
14. Enter “WIZARD – DW_UPGRADE TEST” in the “Folder Name” field
15. Select the ‘Yes’ in the “Scheduling” section of the Query Wizard window
16. Enter “WIZARD – DW_UPGRADE TEST” in the “Suite Name” field
Points of interest:

- The Query Wizard will automatically create the test assets:
  - A new folder in the 'Design Library' with the name entered in the 'Folder Name' field
  - All QueryPair(s) for testing the tables selected
  - A new Test Suite in the 'Scheduling' module with all of the generated QueryPairs

17. Click Next
Points of interest:

- A review screen appears with the details for your Query Wizard execution

18. Click Create
Points of interest:

- A test folder named ‘WIZARD – DW_UPGRADE TEST’ was created in the ‘Design Library’ module
- A Test Suite named ‘WIZARD – DW_UPGRADE TEST’ was created in the ‘Scheduling’ module
- Seven QueryPairs were created (one for each of the source and target tables mapped)

19. Click OK In the “Design Menu” dropdown menu on the top toolbar select “Library Explorer”

20. In the “Design Library” tree (left panel), navigate to the folder “QuerySurge Design -> QueryPairs -> WIZARD – DW_UPGRADE TEST”

Points of interest:

- The name of the folder is the name entered into the ‘Folder Name’ field on the Query Wizard
- The name of a QueryPair is in the following format
  - Validation Type: source schema.source table - target schema.target table

19. Double click the “Data (Table): dw_old.address_dim - dw.address_dim” QueryPair and notice the SQL that has been generated
Points of interest:

- The QueryPair is selecting all columns from the source connection and target connection
- During execution all columns will be compared between source and target

20. Execute a “Design-Time Run” by clicking on the tab above the “Source” pane and clicking the “Run” button
Points of interest:

- QueryPair compared 5,944 source and target rows, which passed
- Feel free to view the results

21. Click the “Scheduling” icon located on the bottom panel

22. Select the “WIZARD – DW_UPGRADE TEST” Test Suite in the left pane

![QueryPair Comparison Table]

Points of interest:

- The name of the Test Suite is the name entered into the ‘Name’ field in the Query Wizard
- The QueryPair(s) are included in the Test Suite upon generation

23. Right click the “WIZARD – DW_UPGRADE TEST” Test Suite and select “Run Now”

24. On “Run Dashboard” double click the “WIZARD – DW_UPGRADE TEST” row in the middle panel to review the results
### Points of interest:

- One *QueryPair* of the seven failed

25. Click the failed *QueryPair*

26. Select the “Failures” tab

### Points of interest:

- Four email addresses do not match from source to target
- Investigation needs to be done to determine the reason for the data differences that occurred during the upgrade
27. Close the “QueryPairs” query results window

28. Click the “Reporting” icon located on the bottom panel

29. Select the “Scenario Detail” report from the left panel

30. Click the ‘Open Report’ button from the middle panel

31. Select the ‘WIZARD – DW_UPGRADE TEST’ (a date time stamp will be appended to the end of the name in the ‘Select a Scenario’ dropdown list)

32. Click Run Report

Points of interest:

- Notice the number of individual field verifications that QuerySurge was able to accomplish, just under a quarter million, in less than 9 seconds on our tutorial instance (completion time will vary based on your hardware)

For more information on Reports, please visit Reporting in the Help guide
Review of Data Health Dashboard

The Data Health Dashboard gives you a real-time view into your project progress at all levels, from QueryPair development to execution and results. Data Widgets are configurable, so you can see the slice of your project that you want to see, in the way you want to see it.

1. Click the “Design Library” icon located on the bottom panel
2. Click the “Welcome” tab if it’s not already in view

Note: If the available widgets panel is not displayed, then click the ‘Add Widget’ Button

Points of interest:

- The “Welcome” tab defaults to one panel containing four Widgets (center panel)
- Click the ‘Add Widget’ button for a list of additional ‘Available Widgets’ (right panel)
- Panels can be configured by dragging and dropping Widgets as desired

3. Click the maximize button on the “Scenario Outcome / Data Reliability” Widget for a closer view
Points of interest:

- Pass/Fail Verifications of your recent Scenario Runs displayed as a bar graph by Pass/Fail Rate
- Data Reliability per Scenario metric displayed as a blue line
- The vertical axis for the bar graph is on the left, while the vertical axis for the blue line is on the right

4. Click the “Scenario Outcome / Data Reliability” Widget title to make it editable, type “Tutorial” at the beginning of the existing title and hit enter
5. Turn off the “Total Verifications Passed” data in the Widget by clicking on its legend
6. Zoom in on the data by clicking on the “SCENARIO” bar in the graph and dragging your mouse across to the ‘ZCITY – CU’ bar. Zoom occurs upon the release of the mouse button
Points of interest:

- Title changes are reflected above the graph and in the title bar
- ‘Total Verifications Passed’ are no longer reflected in the graph, add the data back by clicking on the greyed-out legend
- The graph is zoomed in on the two bars that were highlighted in the previous step and a ‘Reset zoom’ button now appears to revert your zoom selections
- You can point many of the Widgets at specific data by clicking on the ‘View Settings’ icon
- You can have multiple panels of Widgets by clicking on the ‘Add Panel’ button

For more information on Widgets, please visit Reporting → Project Widgets in the Help guide
Summary

You have just finished the QuerySurge tutorial. This is what you completed:

- Created a QueryPair using the Column-Level Comparison Query Wizard
- Reviewed existing and created new QueryPairs
- Created new Reusable Query Snippets
- Executed “Design-Time Runs”
- Investigated ‘Design-Time Run’ failures
- Scheduled and executed a Scenario
- Generated and reviewed reports
- Generated QueryPairs using the Table-Level Comparison Query Wizard
- Reviewed Widgets

There are additional examples within the sample data and QueryPairs, many of them more complex than the exercises you completed in this tutorial. Now that you have a solid foundation on using QuerySurge, try them out. You can access the mapping and data model document from the “Setup Wizard”. 
Deleting the Tutorial Data

This section is for removing the tutorial data from QuerySurge.

**Warning:** Deleting the tutorial data will permanently remove data. Please read this section carefully before proceeding!

The following entities will be permanently deleted: (original sample data)

- The ZCITY folder and all *QueryPairs* contained within
- The XMART folder and all *QueryPairs* contained within
- The *Snippet* named “ZCITY – Filter”
- The *Test Suite* named “ZCITY – ALL”

The following entities are not deleted: (can be deleted manually)

- *QueryPairs* created with the *Query Wizard*
- *Snippets* created during the tutorial
- *Test Suites* created during tutorial
- *Scenario* execution data and results

1. Click the “Administration” icon located on the bottom panel
2. Select “Data Retention” from the “Administration Tree” (left panel)
3. Click **Delete Sample** data from the center panel

![Delete Sample Data dialog]

---

**Warning:** All of the sample data will be permanently deleted! This includes any QueryPairs that were created within the XMART and ZCITY folders.

Are you sure you want to delete all of the sample data?

---

4. Click **OK** under the “Delete Sample Data” dialog

---

**Note:** The ‘Delete Sample Data’ button will remain inactive going forward.
Appendix

Documentation

- **System Requirements Guide** – hardware and software needed to minimally and optimally run QuerySurge
- Installation Guides
  - for **Windows (Single Machine)**
  - for **Windows (Multi Machine)**
  - for **Linux (Single Machine)**
  - for **Linux (Multi Machine)**
- **Configuration Guides** for a broad range of technologies.

**QuickStart User Guide** – everything you need to know to get started with QuerySurge

Data Warehouse Testing

Comprehensive testing of a data warehouse at every point throughout the ETL (extract, transform, and load) process is becoming increasingly important as more data is being collected and used for strategic decision-making.

Data warehouse projects are undertaken as a result of mergers and acquisitions, compliance and regulations, data consolidation, and the increased reliance on data-driven decision making (using Business Intelligence tools, etc.).

Any way you slice it, the data warehouse/business intelligence (BI) platform is complex and presents many data quality and testing problems to overcome. Some of the main challenges of data warehouse testing are:

- **Data Completeness.** Verifying that all data has been loaded from the sources to the target data warehouse
- **Data Transformation.** Ensuring that all data has been transformed correctly during the ETL process
- **Data Quality.** Ensuring that the ETL process correctly rejects, corrects, ignores, substitutes default values, and reports invalid data
- **Regression Testing.** Ensuring existing functionality remains intact each time a new release of code is completed
QuerySurge has five different modules:

**Design Library**

*Use the QuerySurge “Design Library” tools to create collections of powerful tests across your data warehouse architecture.*

- Create test *QueryPairs* between any two points (Source, Staging, Data Warehouse, Data Mart) in your architecture
- See query history for all of your queries
- Run *QueryPair* executions as you build queries to ensure they produce the required results
- Make your design flexible with Reusable Query *Snippet* libraries of query fragments that you can use to modularize your queries, helping to speed up the process of bulk *QueryPair* updates
- Run the ‘Query Wizard’, which provide information for (1) row count compares, (2) full table compares, and (3) column compares without writing any SQL code

**Scheduling**

*Schedule your testing by time and location for maximum productivity. Schedule your tests for the specific time and day or run tests when an event, like the ETL process, completes.*

- Mix-and-match test *QueryPairs* in QuerySurge *Test Suites* to meet specific project execution goals
- Build groups of QuerySurge *Test Suites* to test specific mappings, ETL logic or Data Warehouse Sources
- Build QuerySurge *Scenarios* for scheduling your execution runs at specific dates and times
- Use the execution API for event-based scheduling

**Run Dashboard**

*Visualize and track the real-time progress of your running Scenarios on the QuerySurge ‘Run Dashboard’. A graphical display helps you to follow the flow of your queries, and you can drill-down to see test details and data failures.*

- View query execution progress live via graphical run status displays
- Drill-down into data as QuerySurge executes to examine results as they become available during execution
- See real-time statistics for each *QueryPair* executed and for the *Scenario* execution as a whole
- Alert your team about the status of execution via custom email notifications
- Export detailed results in Excel, CSV, or XML formats to share with team members and other project stakeholders

**Reporting**

*Get a comprehensive audit and share detailed results with others. Use QuerySurge reports to share both high-level and detailed views of your testing with team members, managers, and business stakeholders.*

- Choose from a wide selection of configurable reports within the QuerySurge ‘Reporting Center’
- Built-in reports range from high-level summary reports to lower-level, detailed reports that contain a complete audit trail of test modifications
Configure reports for specific date ranges, asset types, or specific executions, to get a view into your results that you need
Export your reports as Excel or PDF files to share within your organization, or to archive for future audit needs

**Administration**

The administration module provides access to the control features of your QuerySurge installation. Here, you have the ability to create and manage user profiles, database connections, Agent configuration, data archiving and automated email notification options.

- Create and manage QuerySurge users and Agents
- Manage QuerySurge data storage with the included storage tools
- Create and manage connections to source and target data stores
- View QuerySurge server information, including configuration settings and your application licensing details
- Create automated email notifications to send results to your team at the end of a test run

**The QuerySurge Testing Process**

Testing and validating the ETL process is the key component to the success of a data warehouse implementation. Bad data caused by defects in the ETL process can cause data problems in reporting that can result in poor strategic decision making.

**Example**: A large fast food company depends on BI reports to determine how much raw beef to order on a monthly basis, by sales region and time of year. If these reports are not correct, then the company could order an incorrect amount which could cost the company millions of dollars in either lost revenue or wasted product.

Developers utilize ETL tools to accomplish the extracting, transformation and loading of data from sources into target systems.

**Test Points and ETL Legs**

- An “ETL Leg” refers to a single ETL process that moves/transforms data between two discrete points in the data warehouse architecture
- A full ETL process may have multiple legs
- Test points and single ETL leg: the verification is between the source and the target for that leg

**Example**: An operational source database (source test point) is extracted, transformed and loaded into a data warehouse (target test point). Testing is conducted across this ETL leg
• Test points and multiple ETL legs: The multi Leg approach is to validate multiple legs of the ETL process in a single test, ‘jumping over’ ETL legs in the process. A typical multi Leg test is to validate the entire ETL process from data sources to final data endpoint utilizing test points only at the data sources and at the final data endpoint.

**Data Mapping Document**
A source-to-target map is a “data mapping” document and is the most important artifact when building or testing a data warehouse. Data mapping documents are often created in Excel spreadsheets or Word documents. The document acts as a central listing of the “functional” requirements. Testers use the mapping document to verify that the data has been extracted from the source databases, data stores and files and into the target data warehouse and data marts correctly. The following information is contained within the mapping document:

- Source database Information
  - Source table
  - Source column
- Target database Information
  - Target table
  - Target column
- Data transformation logic
- Optional requirements
QuerySurge: The Process in a Nutshell

- Review mapping document
- Determine the optimum percent of coverage or amount of data that is required to be tested, based upon time and resources
- If not testing 100% of the data, determine the exact sampling of data needed
- Create test cases that exercise the requirements of the mapping document
- Create pairs of SQL queries (QueryPairs) – one aimed at the source database or file, one at the target data warehouse or data mart
- Bundle groups of QueryPairs (Test Suites) into test Scenarios
- Schedule Scenarios to run (a) immediately, (b) at a certain day and time, or (c) automatically after an event
- Analyze and drill down into your results and identify bad data and data defects with our robust reporting engine.
- Report defects in your defect tracking tool (i.e. HP Quality Center, IBM Rational Quality Manager, Jira, Bugzilla, etc.)
- Have reports sent automatically via email to team members

The QuerySurge Architecture
About the QuerySurge™ Architecture

QuerySurge is a locally installed, browser-based testing tool. Supporting all current browsers (Chrome, Firefox, IE, Safari, etc.), QuerySurge is perfect for teams that are geographically distributed. QuerySurge extracts the data-under-test to its own, separate infrastructure, which eliminates processing overhead on the source and target database servers in your data warehouse architecture. The QuerySurge architecture is comprised of an application server, database server and Agents.

**QuerySurge Application Server and Database**
QuerySurge utilizes a Tomcat application server and a MySQL database. The database is bundled with and embedded within the product.

**QuerySurge Agents**
QuerySurge Agents are the components of the architecture that execute queries against source and target data sources, returning the results to the QuerySurge database. The Agents execute SQL queries, validating each piece of data throughout the ETL process. This exposes all data mismatch failures, row count differences, and column type mismatch failures, affording you the ability to test to 100% of your data quickly.

Although the Agents issue queries to both the source and target databases, they do not reside on the physical source or target database boxes. QuerySurge Agents can be deployed on the same box as the QuerySurge Application Server and QuerySurge Database Server in a single box install or on any “satellite” boxes in the environment (often, these are test lab boxes or available desktop boxes). The QuerySurge Agent, when it receives a bundle of queries, can run multiple queries simultaneously (in parallel).

One of the features QuerySurge gives you is the ability to raise your ‘testing throughput’. The idea is to be able to fire off QueryPairs in bulk against your sources and targets so that you can execute at the highest level your environment can support. The QuerySurge Agent lets you do this, because you can deploy multiple Agents in your environment – on ‘satellite’ boxes surrounding your QuerySurge server (note that each Agent can run multiple query threads as well). QuerySurge allows you to install up to 10 Agents.

**More Agents = more queries = more throughput**

How many Agents are appropriate for your environment? The answer is – you find out by experimentation. Once you have built an initial test library, start with 2 or 3 Agents, and see how your sources and targets behave. Add additional Agents in a subsequent cycle, again monitoring the source and target behavior. As you continue to add Agents, the loads on sources and targets will grow with query volume – and source/target response times will start to grow as well. Once
you have identified the level where response times have started to increase, back down to the previous level. This level is roughly the maximum throughput your environment can support.

**Database / Data Warehouse Support**

In principle, QuerySurge can support any JDBC-compliant data source. QuerySurge currently ships with Oracle, MySQL, Teradata, Aster, SQL Server, DB2, Informix, Netezza, Sybase, in-memory database, flat file support, and supports many other data sources.
About RTTS

RTTS was founded in 1996, and has cultivated partnerships with the world's leading test tool vendors, including IBM, Microsoft, and HP. We are headquartered in New York City and our satellite locations are in Philadelphia, Atlanta, and Phoenix. Many of our consulting and education services are also offered through the cloud, so that no matter where you are, RTTS will ensure application functionality, performance, scalability, and security for your organization.

RTTS is the premier professional services organization that specializes in providing software quality for critical business applications. We offer the most comprehensive suite of quality assurance services. We've helped 400+ organizations drive positive results from their software development projects.

For more information, please visit www.rttsweb.com.

About QuerySurge™

QuerySurge is the only automated software tool built specifically for ETL testing. It can verify as much as 100% of all data from source systems, through the ETL process, to the target data warehouse and data marts. QuerySurge has increased test coverage and reduced test cycle time for numerous Fortune 500 organizations, helping them to mitigate risk and meet business requirements. For more information, please visit RTTS’ team of test experts developed QuerySurge (www.QuerySurge.com) to address the unique testing needs in the data warehousing and data migration area. It has been implemented on projects ranging from large data warehousing and ETL processes to data migrations, database upgrades, integration testing, data load testing and system patch testing.


Click here to contact us for more information.