Testing DB2 applications without a DB2 subsystem

Andrew Jepeal

Increased Testing Pressure

- Agile – Testing is done at end of sprint. Every two weeks not every 3 months
- Test Driven Development – Tests are created first and run more frequently, sometimes daily
- Shifting Testing Left – do it earlier, do it more often

Increased Testing Challenges

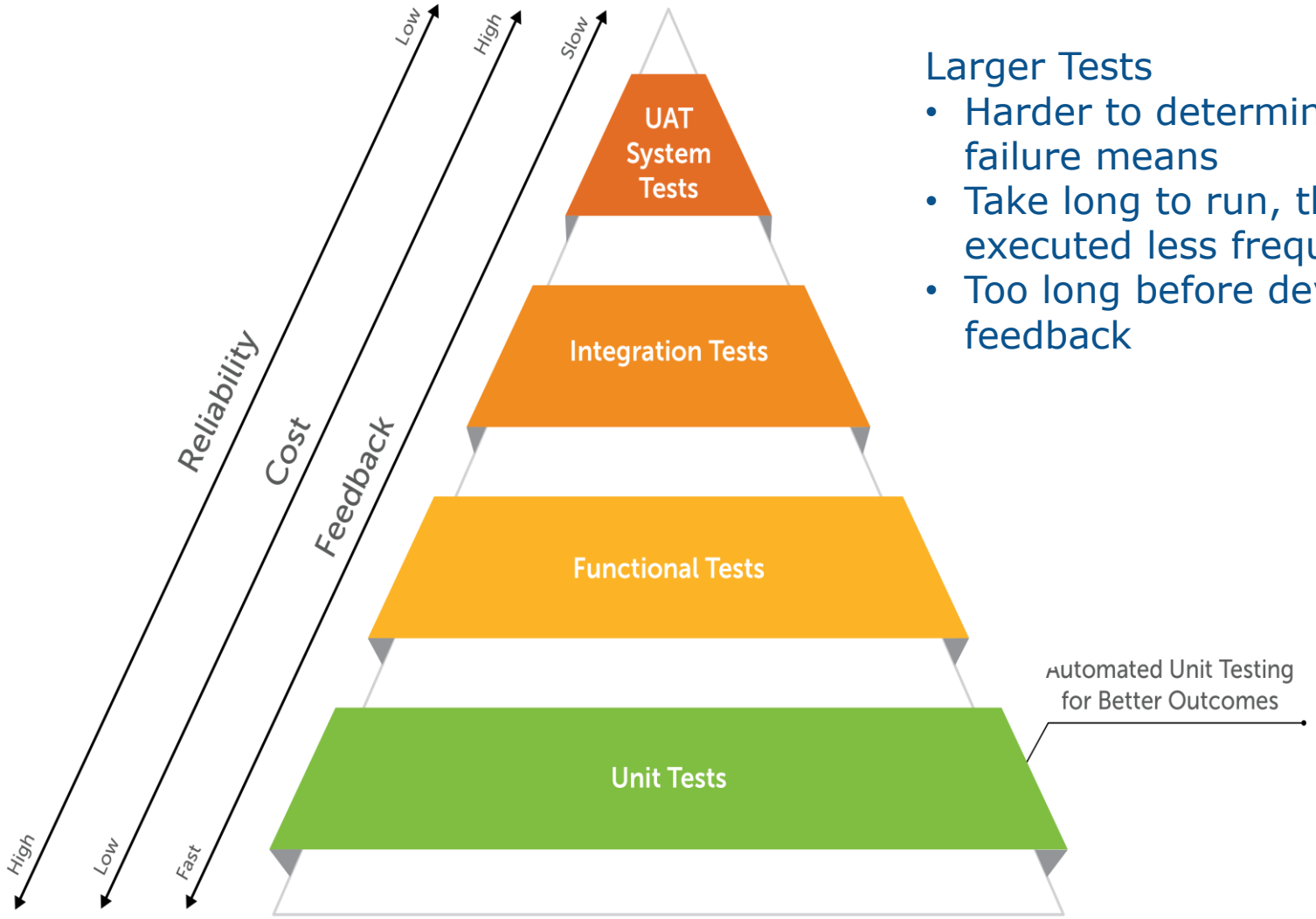
- There's more and they are run more often
  - At the end of every sprint
  - Every day

- Data (tables, databases, datasets) have to be set up in each LPAR
- Data must be refreshed before each test cycle
- New data has to be added to test new functionality
- Barely manageable before the increase in testing

Increased Testing Solutions?

- Add more staff?
- Reduce testing?
- Automate data refresh?
- Use a Unit test framework that allows:
  - Most of the early testing to be done without needing the DB2 subsystem
  - Testing without needing to manually refresh data
  - Developers to change data to test new functionality.

Testing



Larger Tests

- Harder to determine what a failure means
- Take long to run, therefore executed less frequently
- Too long before developer gets feedback

UAT and System Testing

- What – User Acceptance / Performance / Load
- When – Right before release
- Who – Business Unit, Performance team
- Where – UAT LPAR
- Size – Production copy or production size data
- Why – Gets Business unit acceptance, tests application for performance problems.

Integration Testing

- What – Whole application or multiple applications
- When – Before release boundary
- Who – Testing Group or QA
- Where – Test LPARs or UAT LPAR
- Size – Usually a subset of production
- Why – Tests the whole application and how the components integrate with each other and possibly other applications.

Functional Testing

- What – Business function comprised of multiple modules
- When – When all modules are finished
- Who – Developers or Testing Group
- Where – Test LPARs
- Size – Usually a subset of production
- Why – Tests a complete business function but must wait for all changes to be completed. Tests environment also.

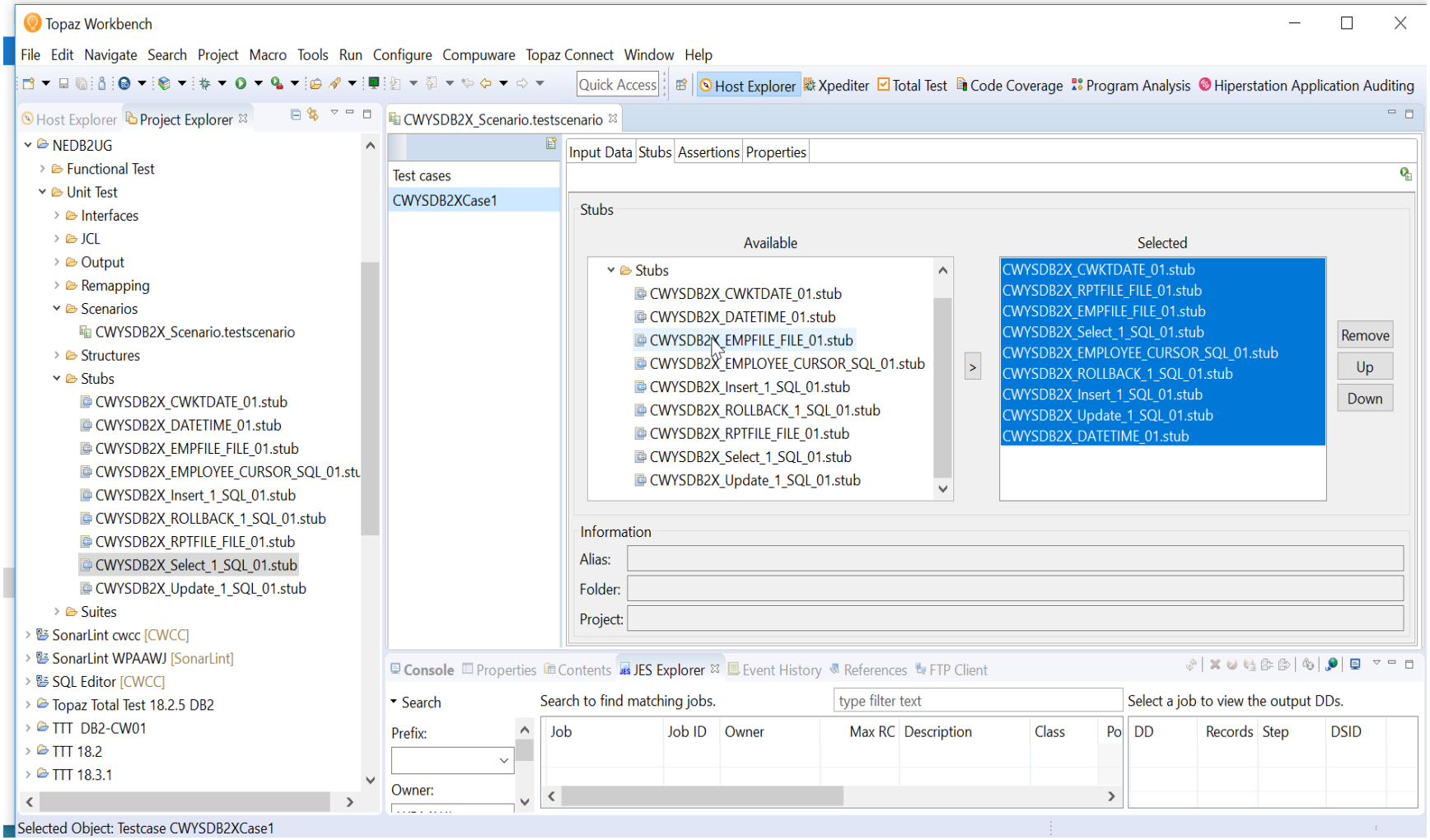
Unit Testing

- What – Individual Unit of code (a module)
- When – Right after code development
- Who – Usually Developers
- Where – Development LPARs and at promotion
- Size – Small number of records – just enough to test all code paths
- Why – Small tests used to check the code logic / algorithms.

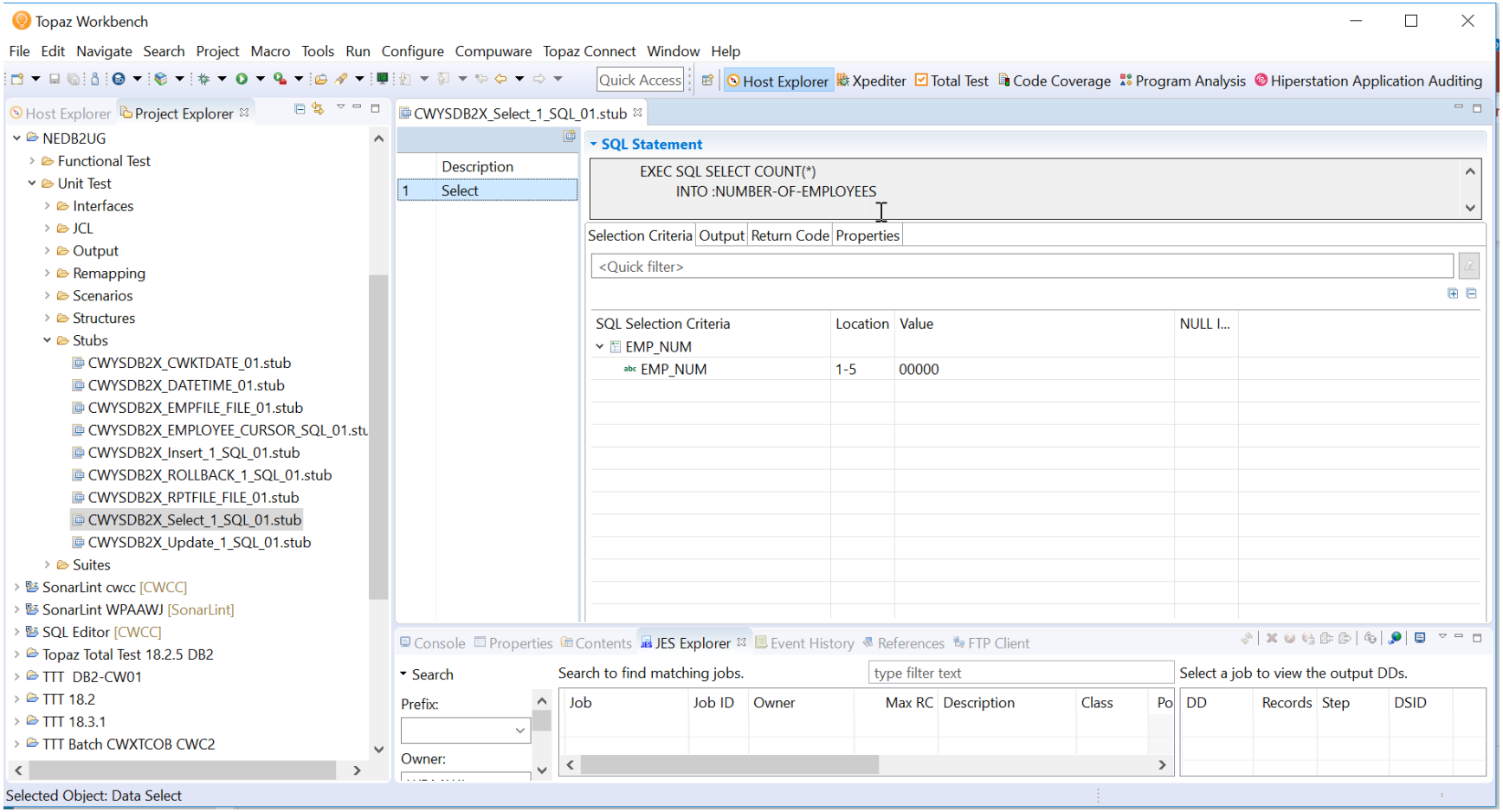
Regression vs New Function

- Not a phase – Done at each phase of testing
- Regression – Rerun tests that have been run before. Make sure working code has not been broken.
- New Function – New tests that make sure the new functions added are working properly.

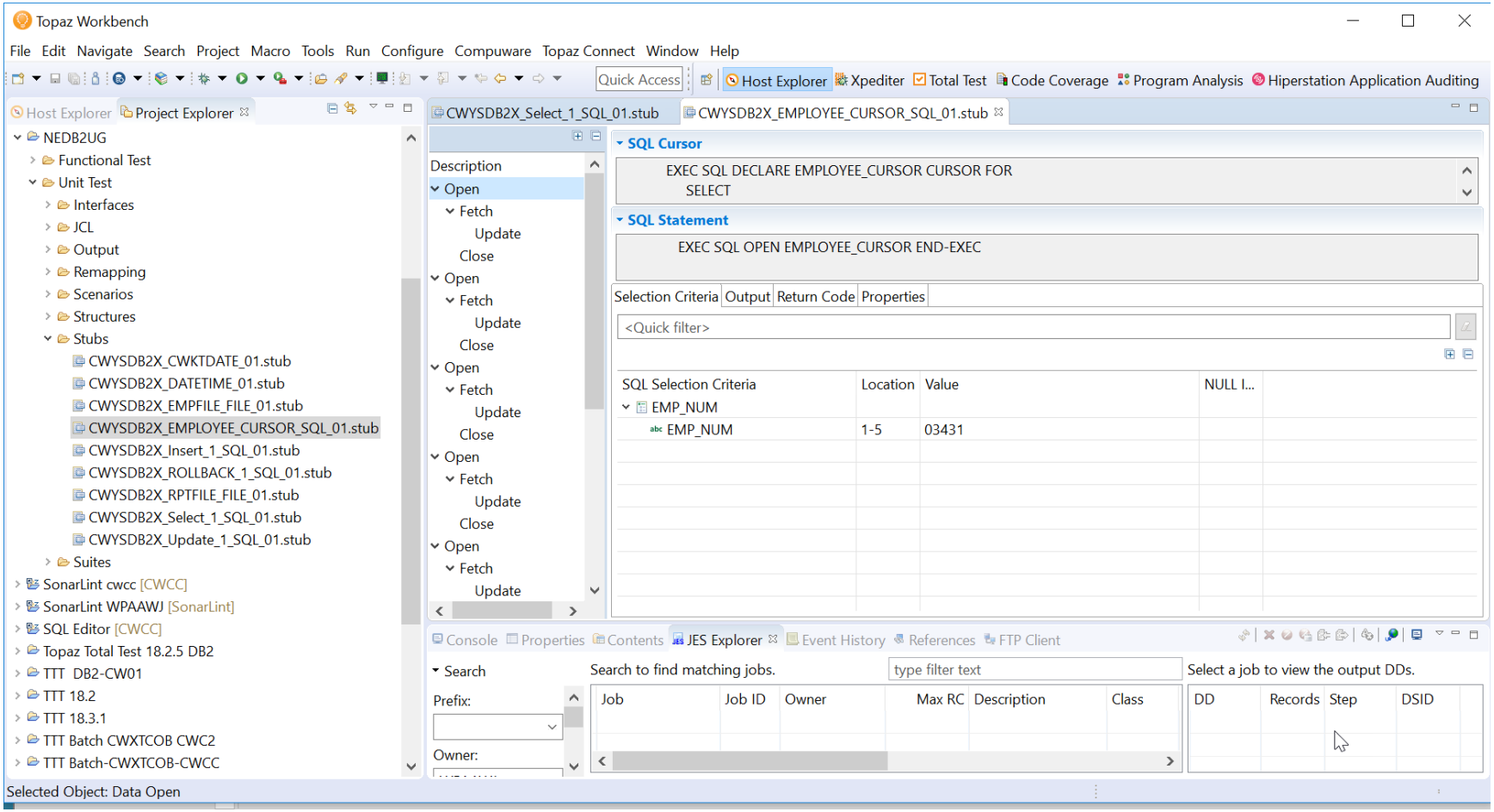
A Unit test framework that captures SQL calls and creates stubs.



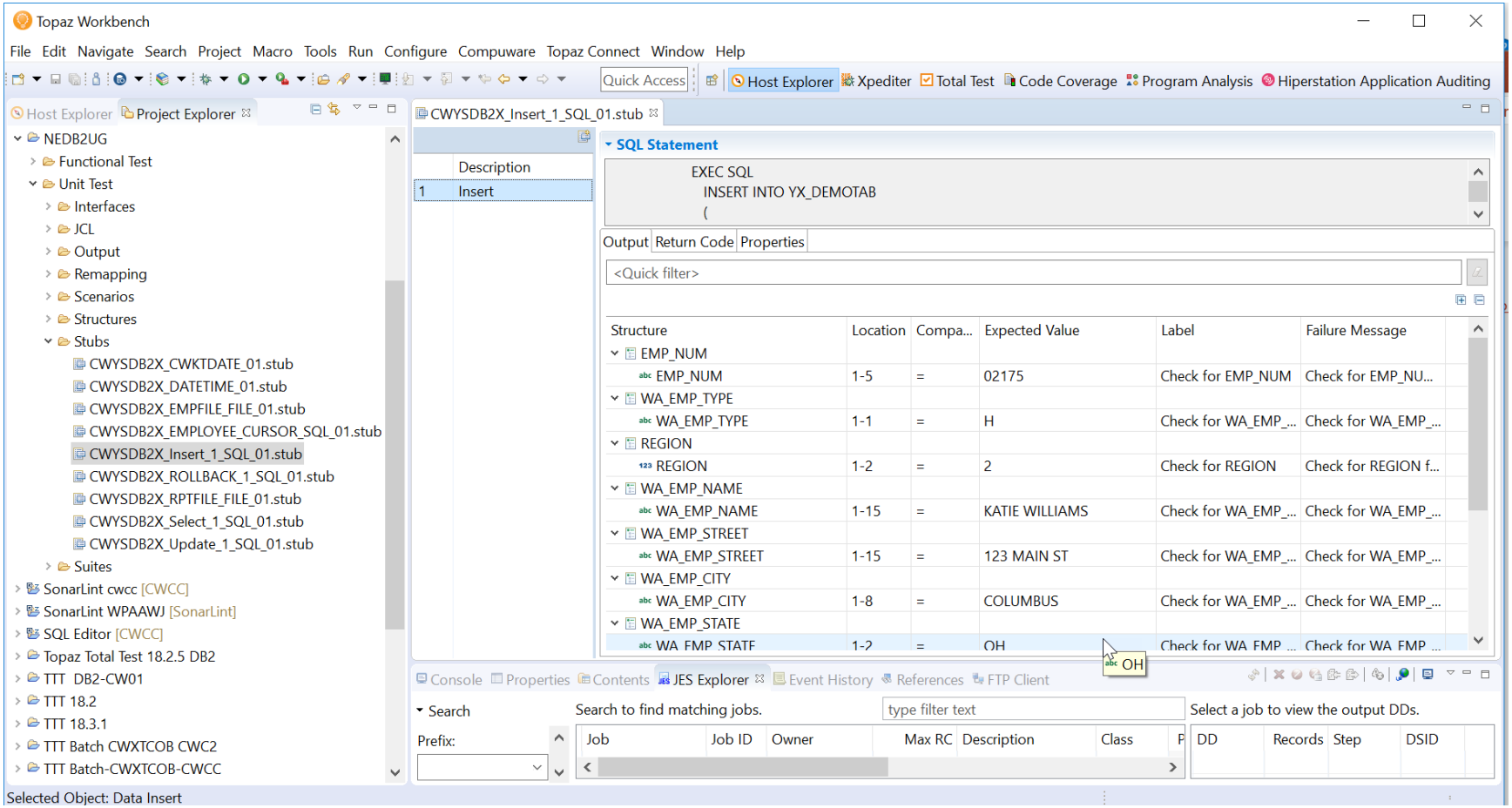
A framework that captures Selects



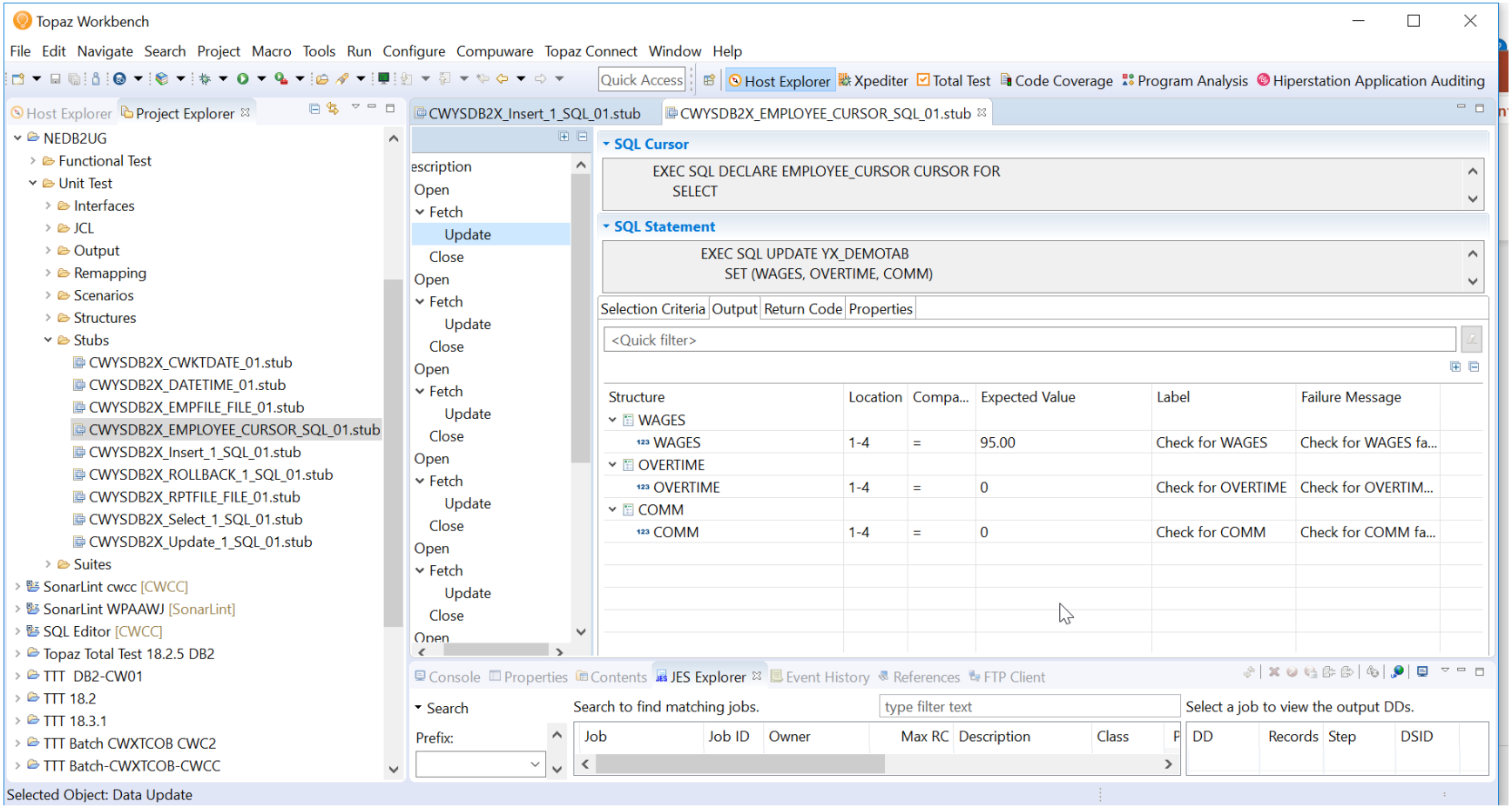
One that can handle Cursors



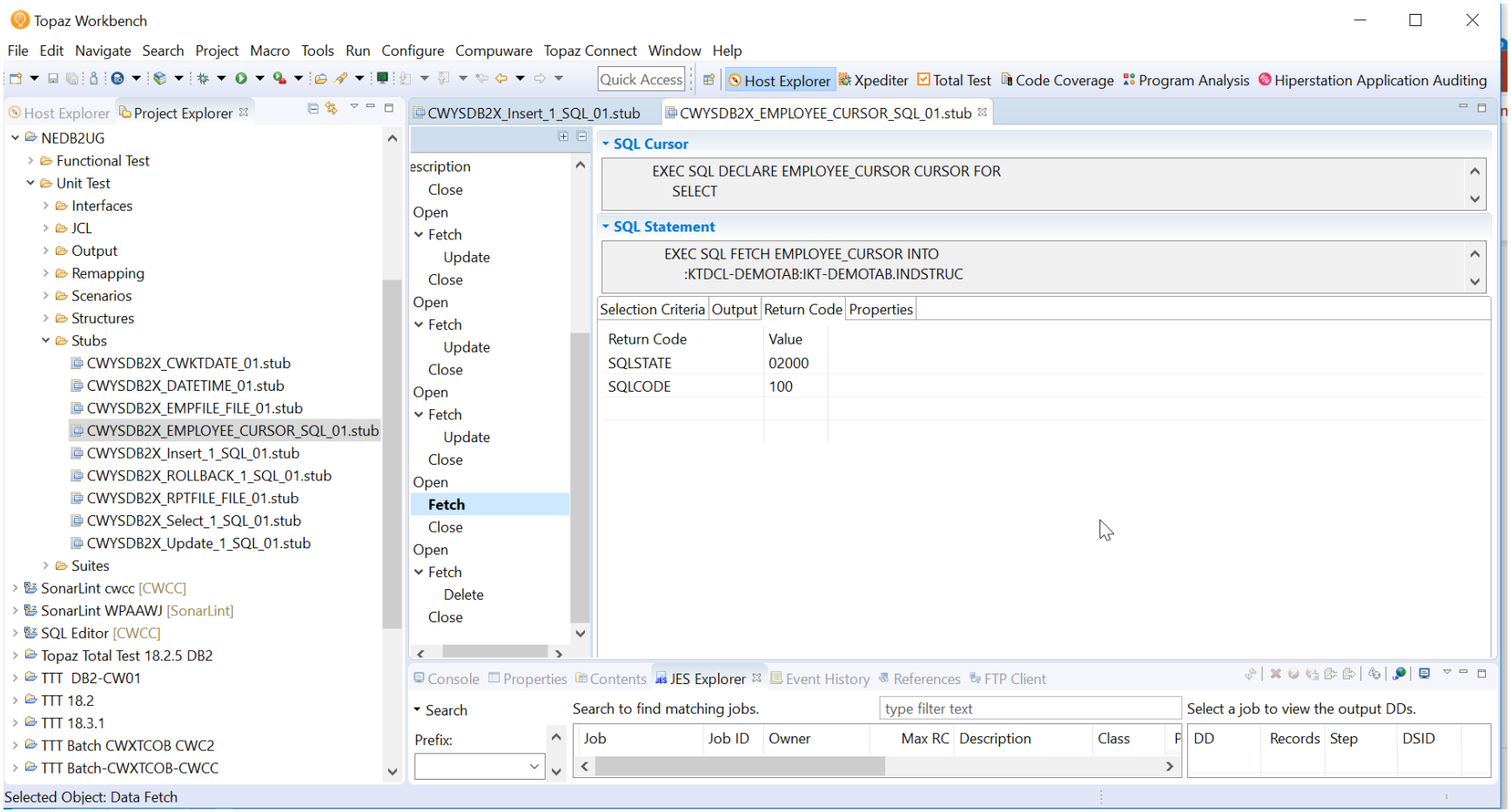
Inserts


Topaz Workbench
File Edit Navigate Search Project Macro Tools Run Configure Compuware Topaz Connect Window Help
Quick Access
Host Explorer
Xpediter
Total Test
Code Coverage
Program Analysis
Hiperstation Application Auditing
Host Explorer
Project Explorer
NEDB2UG
Functional Test
Unit Test
Interfaces
JCL
Output
Remapping
Scenarios
Structures
Stubs
CWYSDB2X_CWKTDATE_01.stub
CWYSDB2X_DATETIME_01.stub
CWYSDB2X_EMPFILE_FILE_01.stub
CWYSDB2X_EMPLOYEE_CURSOR_SQL_01.stub
CWYSDB2X_Insert_1_SQL_01.stub
CWYSDB2X_ROLLBACK_1_SQL_01.stub
CWYSDB2X_RPTFILE_FILE_01.stub
CWYSDB2X_Select_1_SQL_01.stub
CWYSDB2X_Update_1_SQL_01.stub
Suites
SonarLint cwcc [CWCC]
SonarLint WPAAWJ [SonarLint]
SQL Editor [CWCC]
Topaz Total Test 18.2.5 DB2
TTT DB2-CW01
TTT 18.2
TTT 18.3.1
TTT Batch CWXTCOB CWC2
TTT Batch-CWXTCOB-CWCC
CWYSDB2X_Insert_1_SQL_01.stub
Description
1 Insert
SQL Statement
EXEC SQL
INSERT INTO YX_DEMOTAB
(
Output
Return Code
Properties
<Quick filter>
Structure
Location
Compa...
Expected Value
Label
Failure Message
EMP_NUM
EMP_NUM 1-5 = 02175 Check for EMP_NUM Check for EMP_NU...
WA_EMP_TYPE
WA_EMP_TYPE 1-1 = H Check for WA_EMP_... Check for WA_EMP_...
REGION
REGION 1-2 = 2 Check for REGION Check for REGION f...
WA_EMP_NAME
WA_EMP_NAME 1-15 = KATIE WILLIAMS Check for WA_EMP_... Check for WA_EMP_...
WA_EMP_STREET
WA_EMP_STREET 1-15 = 123 MAIN ST Check for WA_EMP_... Check for WA_EMP_...
WA_EMP_CITY
WA_EMP_CITY 1-8 = COLUMBUS Check for WA_EMP_... Check for WA_EMP_...
WA_EMP_STATE
OH
Console
Properties
Contents
JES Explorer
Event History
References
FTP Client
Search
Prefix:
Search to find matching jobs.
type filter text
Job
Job ID
Owner
Max RC
Description
Class
Select a job to view the output DDs.
DD
Records
Step
DSID
Selected Object: Data Insert

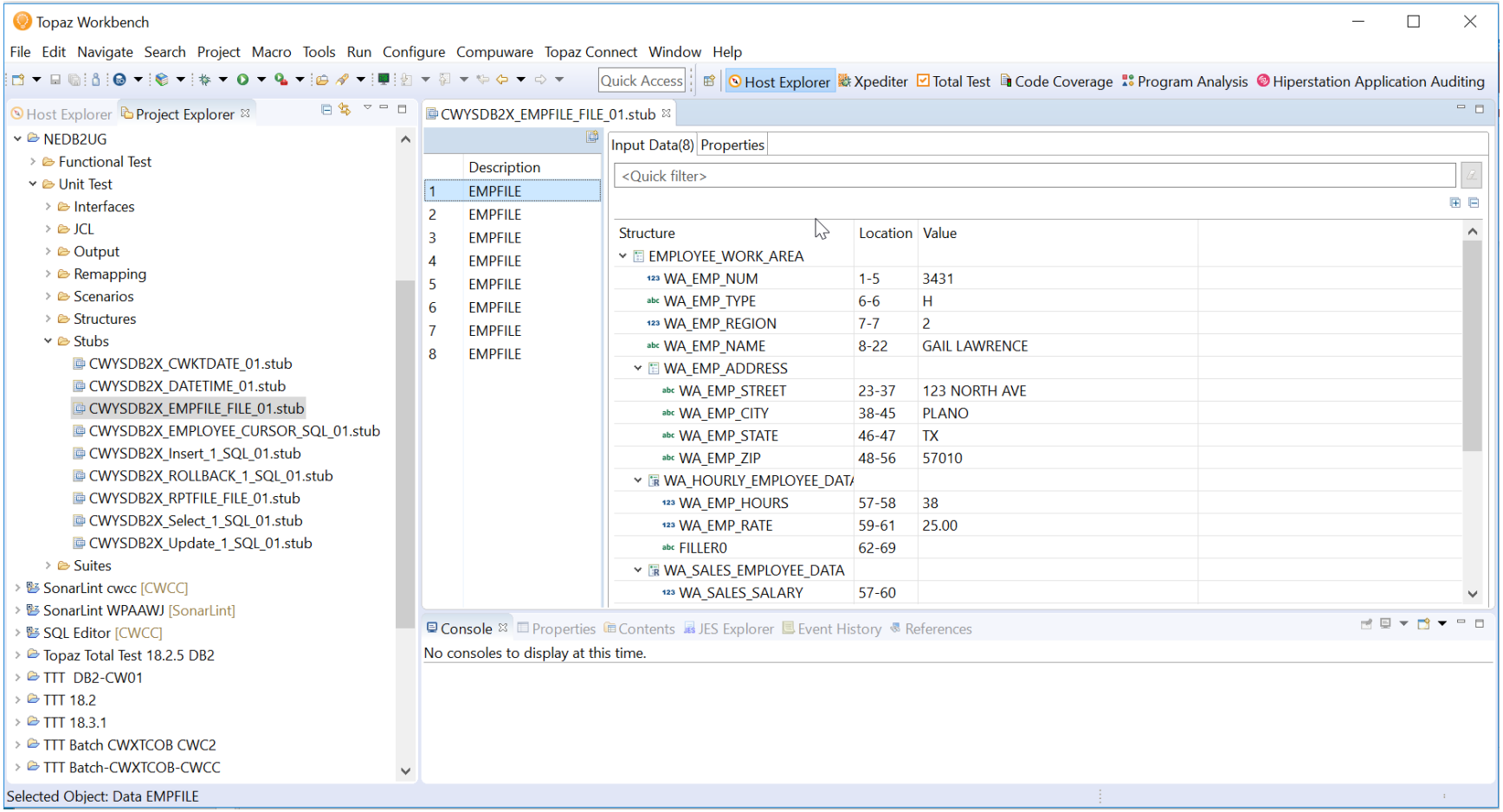
Updates


Topaz Workbench
File Edit Navigate Search Project Macro Tools Run Configure Compuware Topaz Connect Window Help
Quick Access
Host Explorer
Xpediter
Total Test
Code Coverage
Program Analysis
Hiperstation Application Auditing
Host Explorer
Project Explorer
NEDB2UG
Functional Test
Unit Test
Interfaces
JCL
Output
Remapping
Scenarios
Structures
Stubs
CWYSDB2X_CWKTDATE_01.stub
CWYSDB2X_DATETIME_01.stub
CWYSDB2X_EMPFILE_FILE_01.stub
CWYSDB2X_EMPLOYEE_CURSOR_SQL_01.stub
CWYSDB2X_Insert_1_SQL_01.stub
CWYSDB2X_ROLLBACK_1_SQL_01.stub
CWYSDB2X_RPTFILE_FILE_01.stub
CWYSDB2X_Select_1_SQL_01.stub
CWYSDB2X_Update_1_SQL_01.stub
Suites
SonarLint cwcc [CWCC]
SonarLint WPAAWJ [SonarLint]
SQL Editor [CWCC]
Topaz Total Test 18.2.5 DB2
TTT DB2-CW01
TTT 18.2
TTT 18.3.1
TTT Batch CWXTCOB CWC2
TTT Batch-CWXTCOB-CWCC
CWYSDB2X_Insert_1_SQL_01.stub
CWYSDB2X_EMPLOYEE_CURSOR_SQL_01.stub
escription
Open
Fetch
Update
Close
SQL Cursor
EXEC SQL DECLARE EMPLOYEE_CURSOR CURSOR FOR
SELECT
SQL Statement
EXEC SQL UPDATE YX_DEMOTAB
SET (WAGES, OVERTIME, COMM)
Selection Criteria
Output
Return Code
Properties
<Quick filter>
Structure
Location
Compa...
Expected Value
Label
Failure Message
WAGES
WAGES
1-4
=
95.00
Check for WAGES
Check for WAGES fa...
OVERTIME
OVERTIME
1-4
=
0
Check for OVERTIME
Check for OVERTIM...
COMM
COMM
1-4
=
0
Check for COMM
Check for COMM fa...
Console
Properties
Contents
JES Explorer
Event History
References
FTP Client
Search
Prefix:
Search to find matching jobs.
type filter text
Select a job to view the output DDs.
Job
Job ID
Owner
Max RC
Description
Class
DD
Records
Step
DSID
Selected Object: Data Update

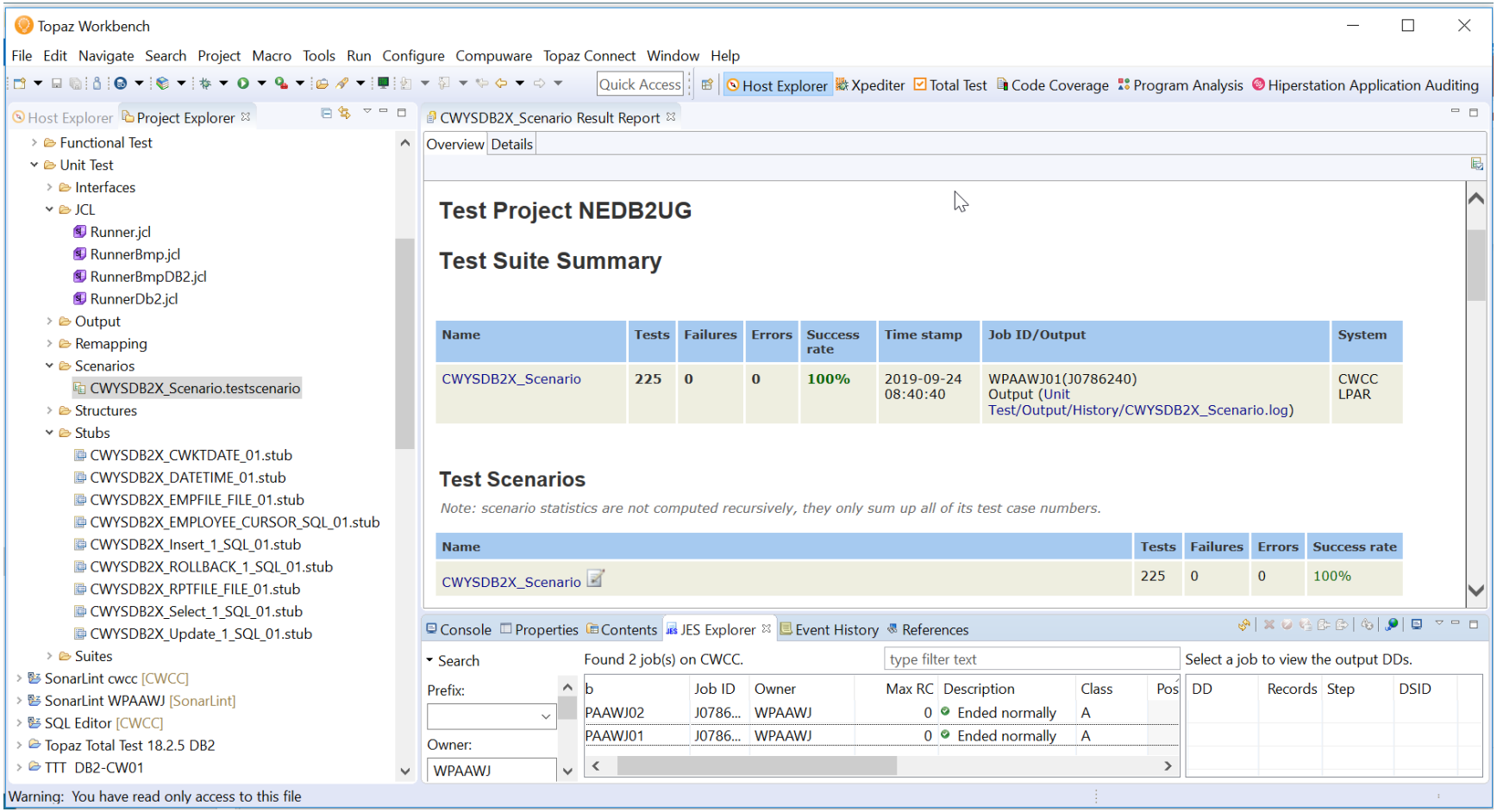
Non-zero SQL codes



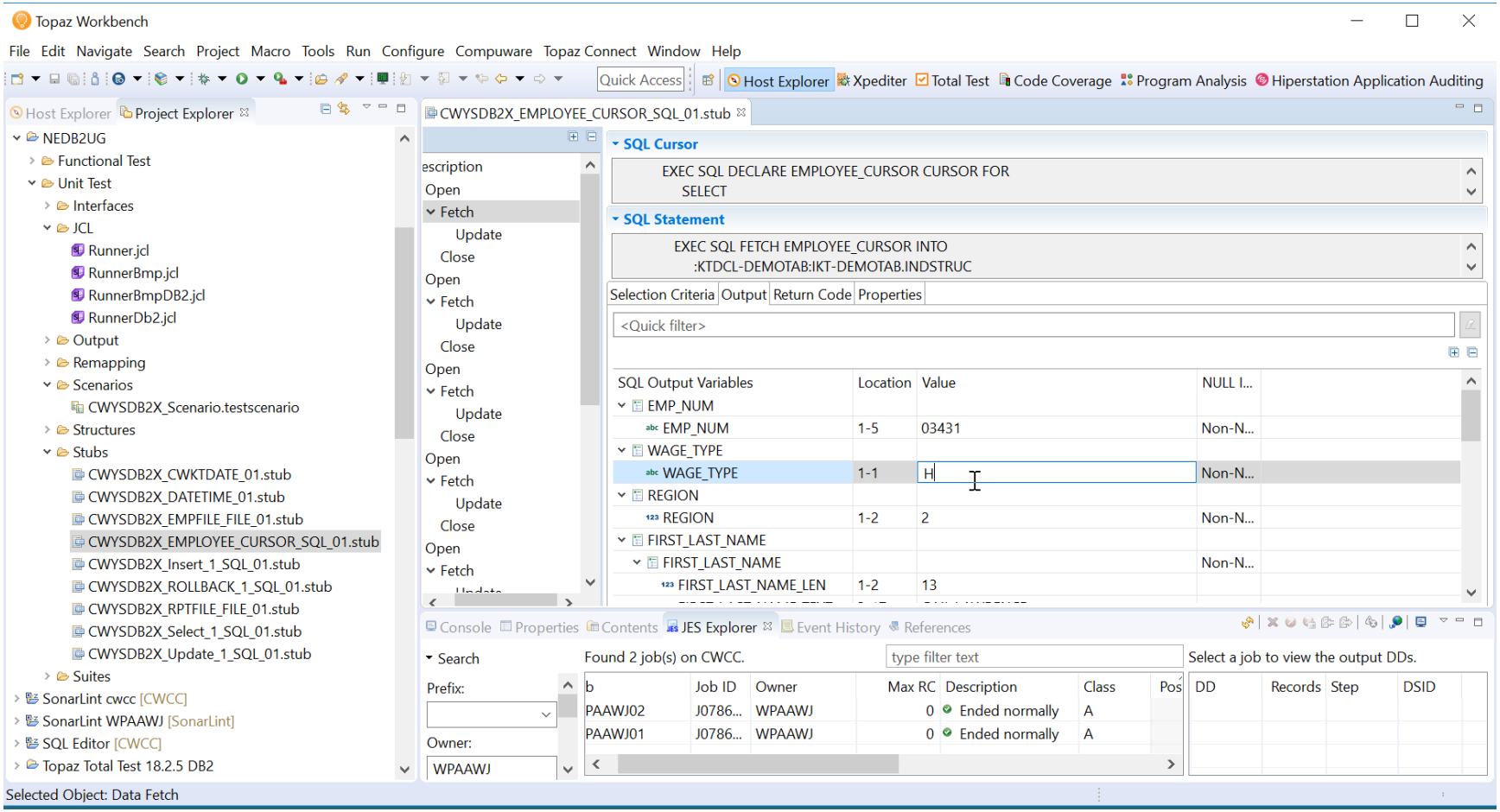
Handles other data and program calls



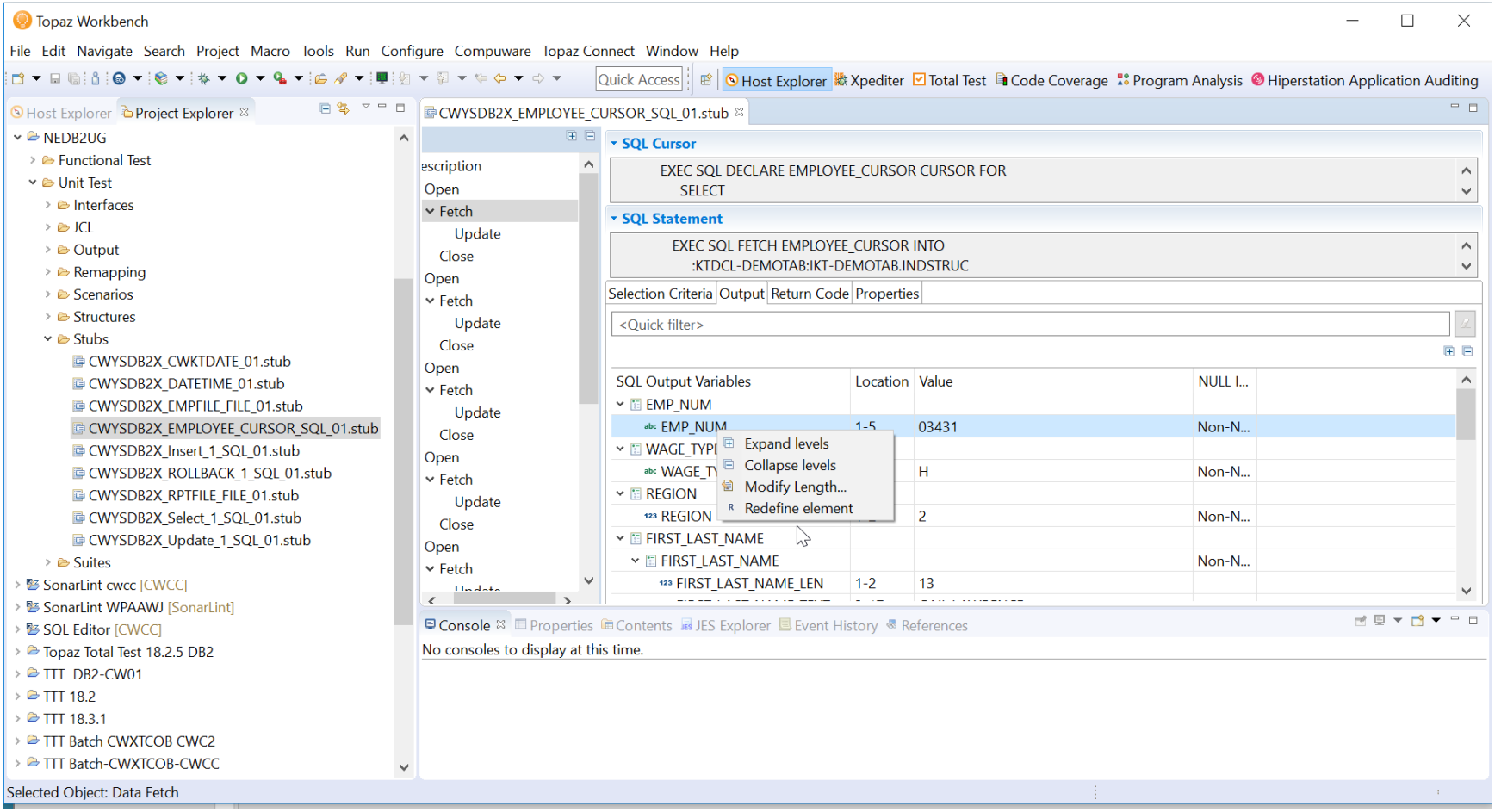
Quickly lets you see if the test passes or fails



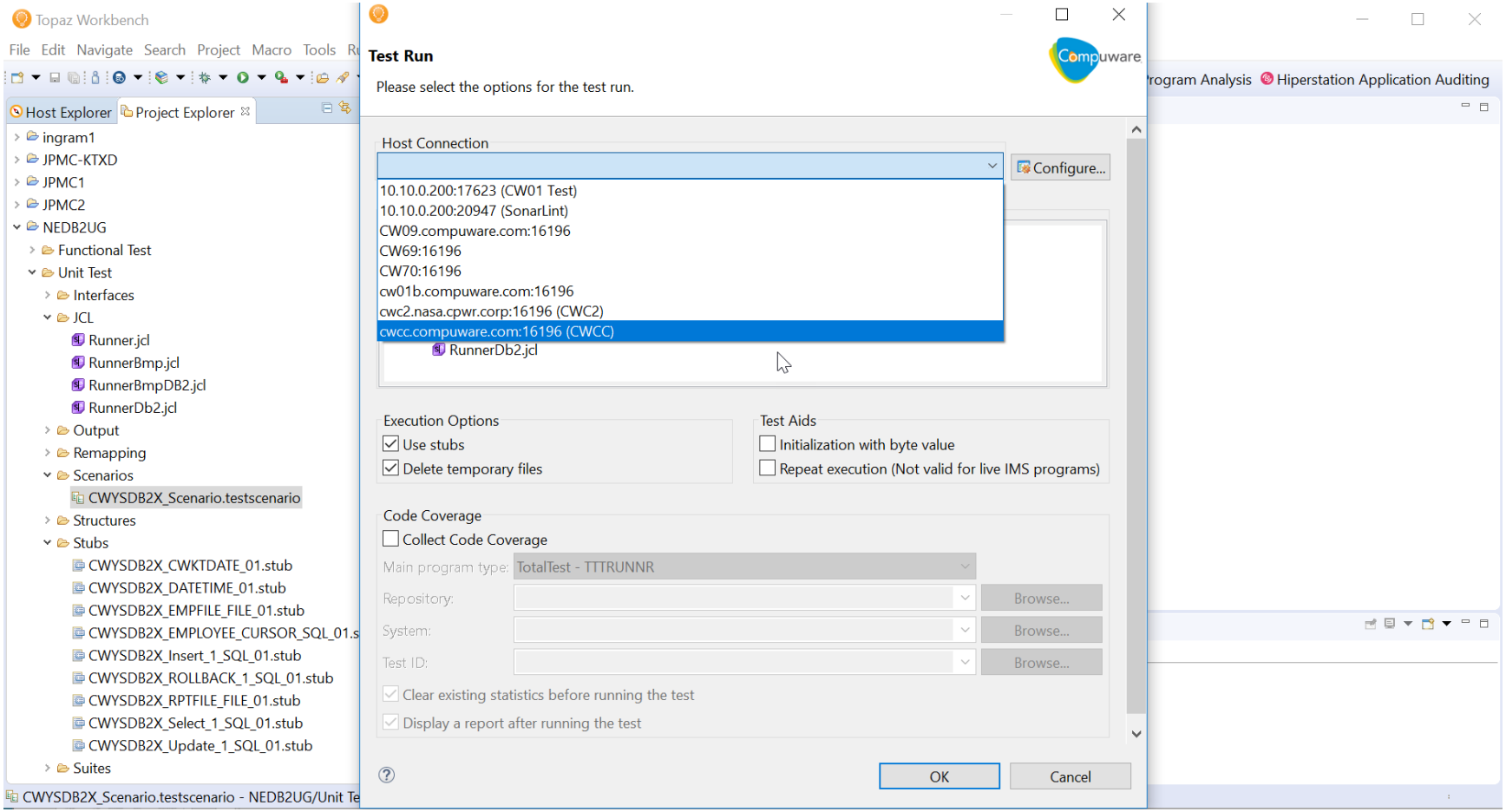
Lets Developers change test data easily



Lets Developers modify the data structure



This creates portable unit tests that can even be run on LPARs without DB2



Let DBAs and Sys Progs focus on their work, not get bogged down in maintaining testing environments.