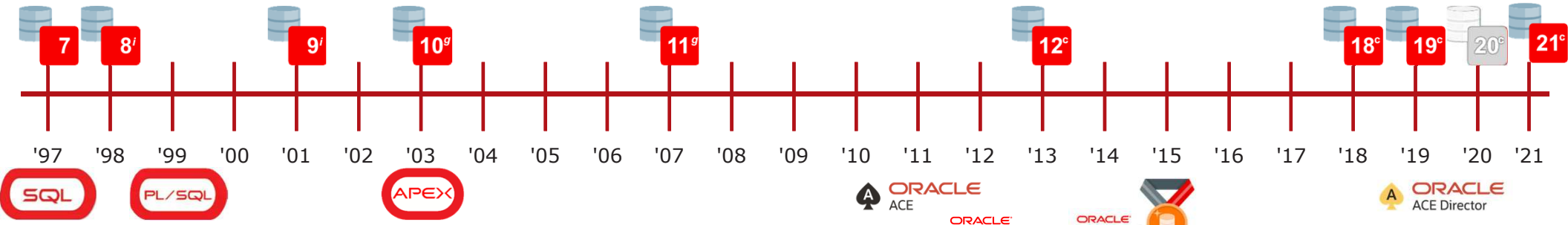
QUALOGY

PBarel@Qualogy.com
http://blog.bar-solutions.com

QUALOGY

About me…



ORACLE ACE

ORACLE ACE Director

ORACLE Certified Associate PL/SQL Developer

ORACLE Certified Professional Advanced PL/SQL Developer

bar-solutions.com blog http://blog.bar-solutions.com

All things ORACLE http://allthingsoracle.com

OTECH MAGAZINE http://www.otechmag.com

www.red-gate.com/simple-talk/author/patrick-barel/

bar-solutions.com/otechmagazine.php

Plugins for PL/SQL Developer
http://plugins.bar-solutions.com

Contact me…

@patch72

PBarel@Qualogy.com

Patrick.Barel@GMail.com

patrick@bar-solutions.com

Patrick.Barel@GMail.com

3029156

40338721

Patrick Barel

ORACLE
ACE Program

500+ technical experts helping peers globally

The Oracle ACE Program recognizes and rewards community members for their technical contributions in the Oracle community

3 membership tiers

ORACLE ACE Director | ORACLE ACE | ORACLE ACE Associate

For more details on Oracle ACE Program:
bit.ly/OracleACEProgram

Oracle Groundbreakers

Nominate

yourself or someone you know:

acenomination.oracle.com

Connect: **oracle-ace_ww@oracle.com** Facebook.com/oracleaces @oracleace

Oracle Cloud Infrastructure

New Free Tier

oracle.com/cloud/free

Always Free

Services you can use for unlimited time

+

30-Day Free Trial

Free credits you can use for more services

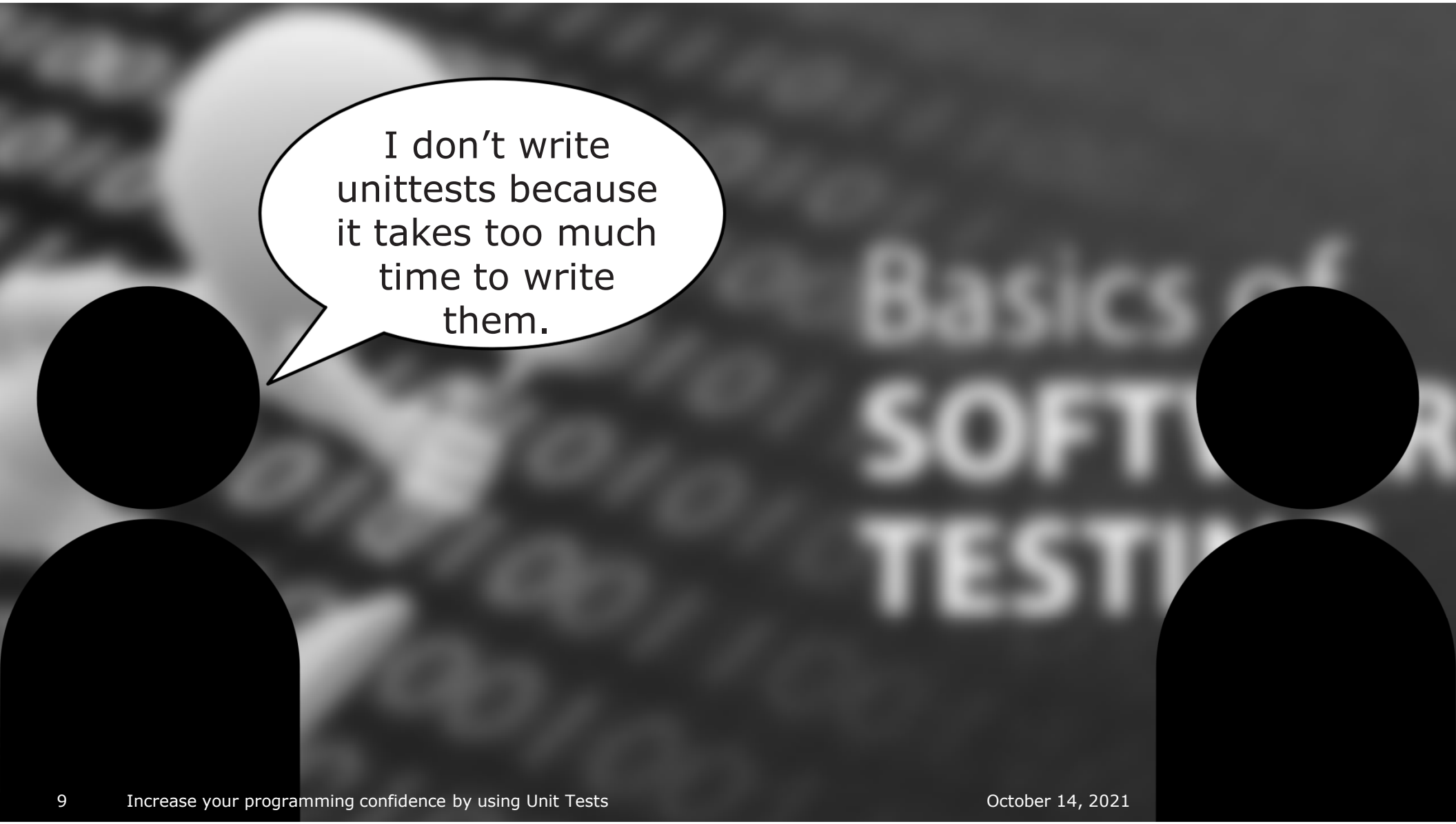
Increase your programming confidence by using Unit Tests

Patrick Barel, Qualogy October 14, 2021

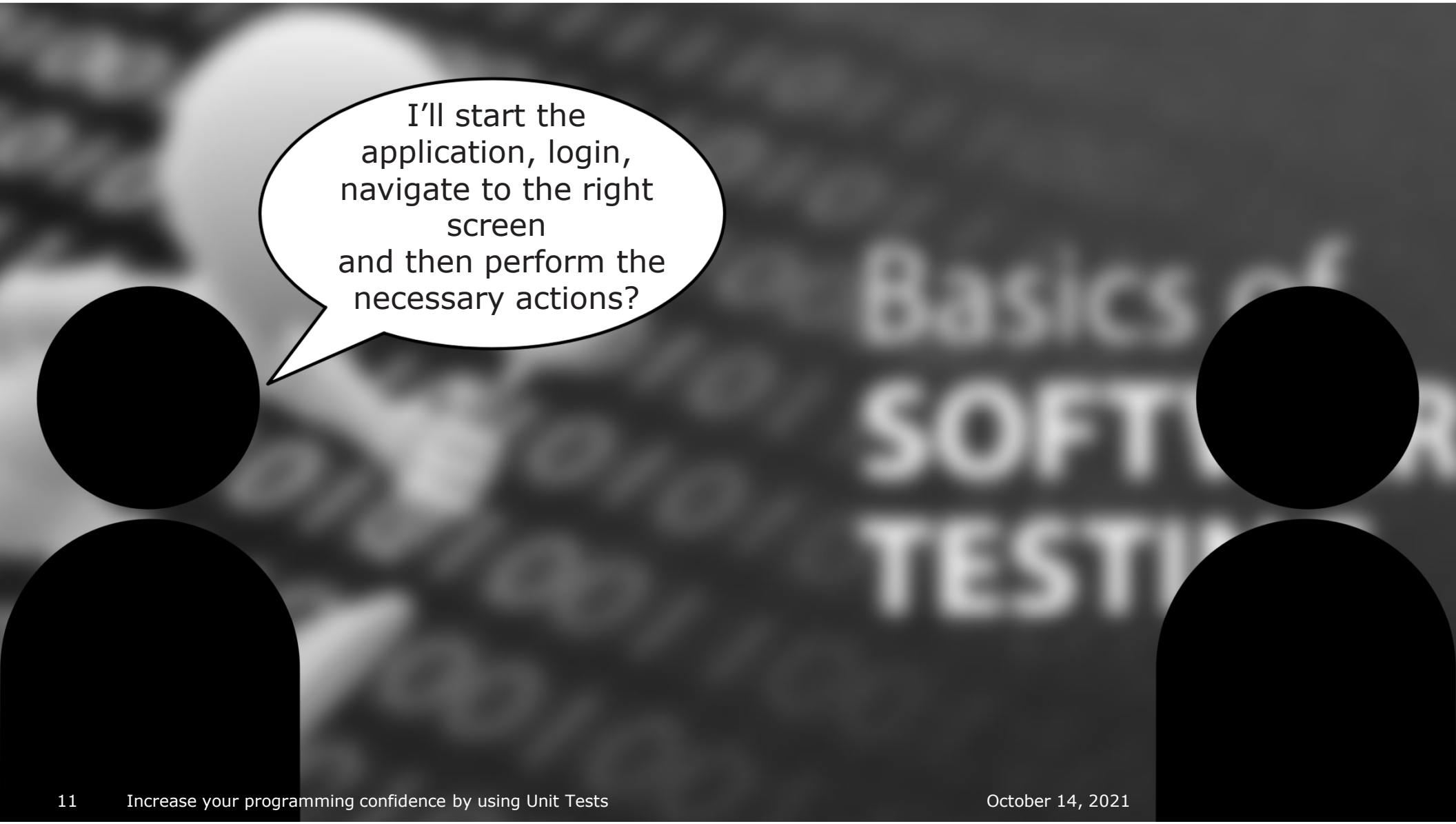
QUALOGY

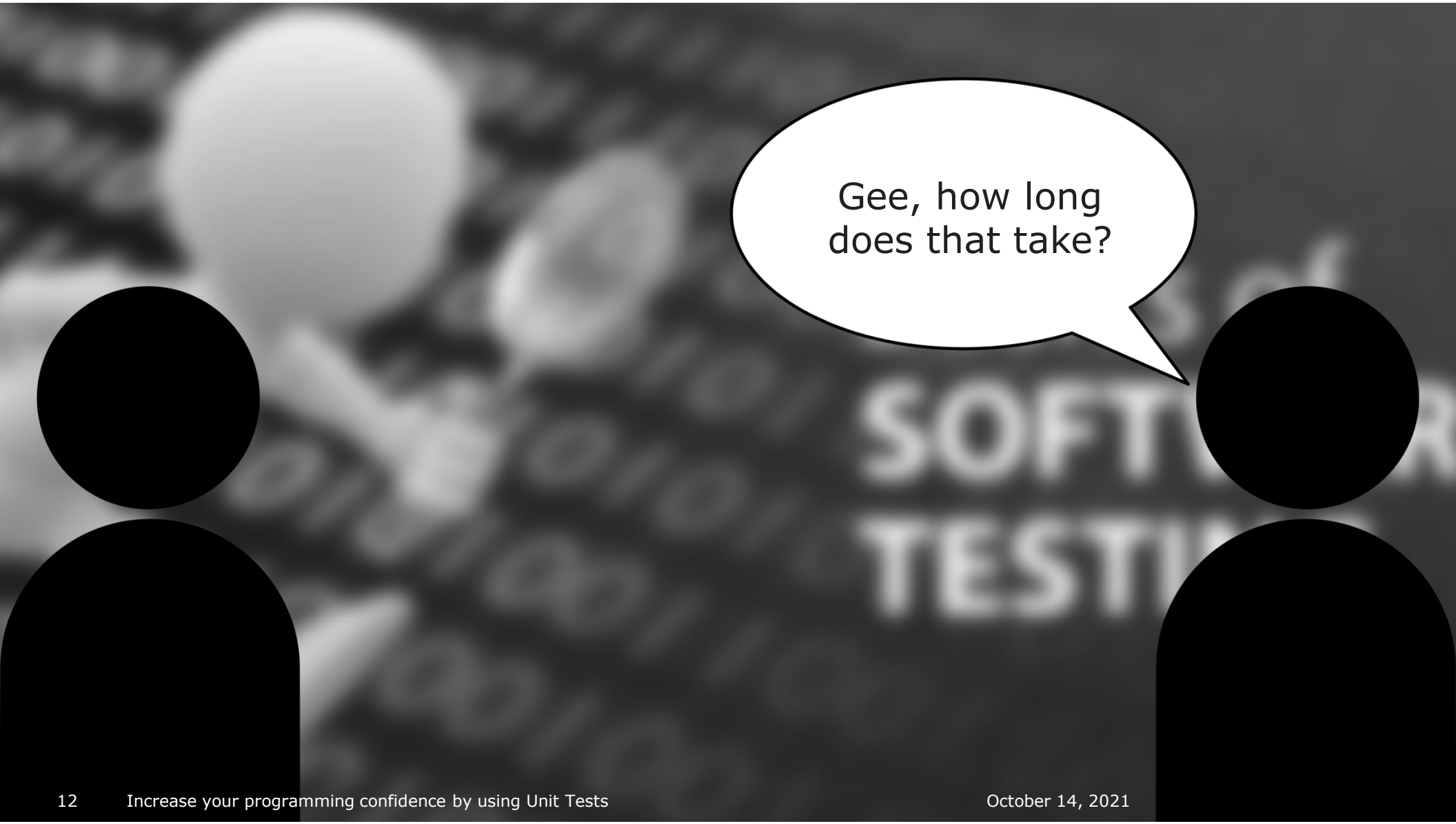

Basics of
SOFTWARE
TESTING


Basics of
SOFTWARE
TESTING


I don’t write unittests because it takes too much time to write them.


Then, how do you test your code?


I'll start the application, login, navigate to the right screen and then perform the necessary actions?


Gee, how long
does that take?

As a developer
I need a function
That returns the characters from a given string between a given starting character and an ending character.
Example: Send in abcdefgh and the numbers 3 and 5 and receive cde

As a developer
I need a function
That returns the characters from a given string between a given starting character and an ending character.
Example: Send in abcdefgh and the numbers 3 and 5 and receive cde

As a developer
I need a function
That returns the characters from a given string between a given starting character and an ending character.
Example: Send in abcdefgh and the numbers 3 and 5 and receive cde

```
function
```

As a developer
I need a function
That returns the characters from a given string between a given starting character and an ending character.
Example: Send in abcdefgh and the numbers 3 and 5 and receive cde

```
function betwnstr
```

As a developer
I need a function
That returns the characters from a given string between a given starting character and an ending character.
Example: Send in abcdefgh and the numbers 3 and 5 and receive cde

```
function betwnstr(

                                    ) return varchar2
```

As a developer
I need a function
That returns the characters from a given string between a given starting character and an ending character.
Example: Send in abcdefgh and the numbers 3 and 5 and receive cde

```
function betwnstr(string_in in varchar2
                 ,start_in  in integer
                 ,end_in    in integer) return varchar2
```

As a developer
I need a function
That returns the characters from a given string between a given starting character and an ending character.
Example: Send in abcdefgh and the numbers 3 and 5 and receive cde

```
SQL> create or replace function betwnstr(string_in in varchar2
  2                                     ,start_in  in integer
  3                                     ,end_in    in integer) return varchar2 is
  4  begin
  5    return substr(                                          );
  6  end betwnstr;
  7  /
```

```
SQL> create or replace function betwnstr(string_in in varchar2
  2                                      ,start_in  in integer
  3                                      ,end_in    in integer) return varchar2 is
  4  begin
  5    return substr(                                          );
  6  end betwnstr;
  7  /
```

```
SQL> create or replace function betwnstr(string_in in varchar2
  2                                     ,start_in  in integer
  3                                     ,end_in    in integer) return varchar2 is
  4  begin
  5    return substr(string_in, start_in, end_in              );
  6  end betwnstr;
  7  /

SQL> exec dbms_output.put_line(betwnstr('abcdefgh', 3, 5))                    cdefg
```

```
SQL> create or replace function betwnstr(string_in in varchar2
  2                                     ,start_in  in integer
  3                                     ,end_in    in integer) return varchar2 is
  4  begin
  5    return substr(string_in, start_in, end_in - start_in    );
  6  end betwnstr;
  7  /

SQL> exec dbms_output.put_line(betwnstr('abcdefgh', 3, 5))                      cd
```

```
SQL> create or replace function betwnstr(string_in in varchar2
  2                                     ,start_in  in integer
  3                                     ,end_in    in integer) return varchar2 is
  4  begin
  5    return substr(string_in, start_in, end_in - start_in - 1);
  6  end betwnstr;
  7  /

SQL> exec dbms_output.put_line(betwnstr('abcdefgh', 3, 5))                      c
```

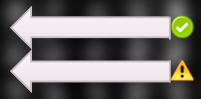
```
SQL> create or replace function betwnstr(string_in in varchar2
  2                                     ,start_in  in integer
  3                                     ,end_in    in integer) return varchar2 is
  4  begin
  5    return substr(string_in, start_in, end_in - start_in + 1);
  6  end betwnstr;
  7  /

SQL> exec dbms_output.put_line(betwnstr('abcdefgh', 3, 5))
cde

SQL> exec dbms_output.put_line(betwnstr('abcdefgh', 0, 2))
abc

SQL> exec dbms_output.put_line(betwnstr('abcdefgh', 3, 100))
cdefgh
```

```
SQL> begin
  2     dbms_output.put_line(betwnstr('abcdefgh', 3, 5));
  3     dbms_output.put_line(betwnstr('abcdefgh',0,2));
  4     dbms_output.put_line(betwnstr('abcdefgh',null,5));
  5     dbms_output.put_line(betwnstr('abcdefgh',3,null));
  6     dbms_output.put_line(betwnstr('abcdefgh',3,100));
  7     dbms_output.put_line(betwnstr('abcdefgh',-3,-5));
  8     dbms_output.put_line(betwnstr('abcdefgh',-3,0));
  9  end;
 10  /
cde
abc

cdefgh

fgh


PL/SQL procedure successfully completed.
```

```
SQL> create or replace function betwnstr(string_in in varchar2
  2                                      ,start_in  in integer
  3                                      ,end_in    in integer) return varchar2 is
  4    l_start integer := start_in;
  5  begin
  6    -- 0 should be start of string so change it to 1
  7    if l_start = 0 then
  8      l_start := 1;
  9    end if;
 10
 11    return substr(string_in, l_start, end_in - l_start + 1);
 12  end betwnstr;
 13  /
```

```
SQL> begin
  2    dbms_output.put_line(betwnstr('abcdefgh', 3, 5));
  3    dbms_output.put_line(betwnstr('abcdefgh',0,2));
  4    dbms_output.put_line(betwnstr('abcdefgh',null,5));
  5    dbms_output.put_line(betwnstr('abcdefgh',3,null));
  6    dbms_output.put_line(betwnstr('abcdefgh',3,100));
  7    dbms_output.put_line(betwnstr('abcdefgh',-3,-5));
  8    dbms_output.put_line(betwnstr('abcdefgh',-3,0));
  9  end;
 10  /
cde
ab

cdefgh

fgh


PL/SQL procedure successfully completed.
```

```
SQL> create or replace function betwnstr(string_in in varchar2
  2                                      ,start_in  in integer
  3                                      ,end_in    in integer) return varchar2 is
  4    l_start integer := start_in;
  5  begin
  6    -- 0 should be start of string so change it to 1
  7    if l_start = 0 then
  8      l_start := 1;
  9    end if;
 10    if l_start is null then
 11      l_start := 1;
 12    end if;
 13    return substr(string_in, l_start, end_in - l_start + 1);
 14  end betwnstr;
 15  /
```

```
SQL> begin
  2    dbms_output.put_line(betwnstr('abcdefgh', 3, 5));
  3    dbms_output.put_line(betwnstr('abcdefgh',0,2));
  4    dbms_output.put_line(betwnstr('abcdefgh',null,5));
  5    dbms_output.put_line(betwnstr('abcdefgh',3,null));
  6    dbms_output.put_line(betwnstr('abcdefgh',3,100));
  7    dbms_output.put_line(betwnstr('abcdefgh',-3,-5));
  8    dbms_output.put_line(betwnstr('abcdefgh',-3,0));
  9  end;
 10  /
cde
ab
abcde


cdefgh

fgh


PL/SQL procedure successfully completed.
```


Test Automation

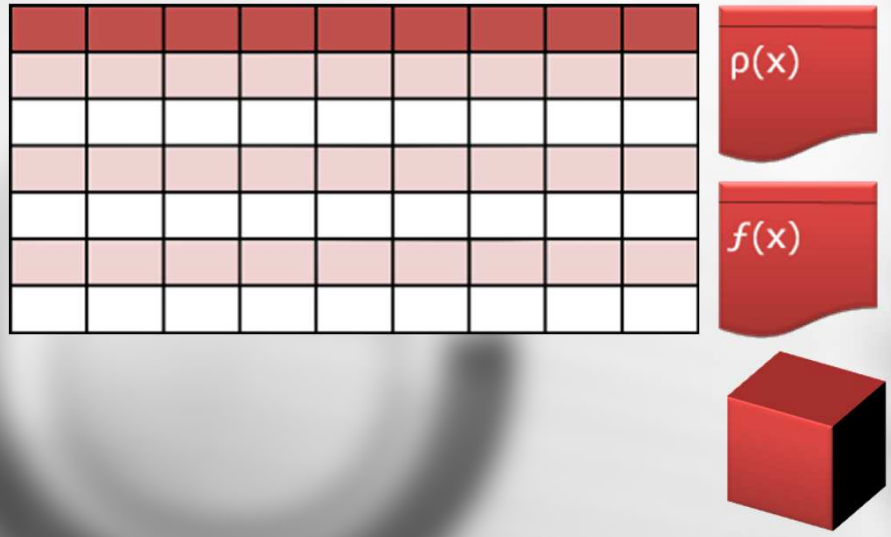

Test Automation

Unit Testing

Setup 

Unit Testing

- Build tables
- Enter records
- Run Code




```
SQL> declare
  2    test_this varchar2(100);
  3    against_this varchar2(100);
  4  begin
  5    against_this := 'cde';
```

Setup

```
 12  end;
 13  /
```

Unit Testing

```
SQL> declare
  2    test_this varchar2(100);
  3    against_this varchar2(100);
  4  begin
  5    against_this := 'cde';
```

Run

Setup

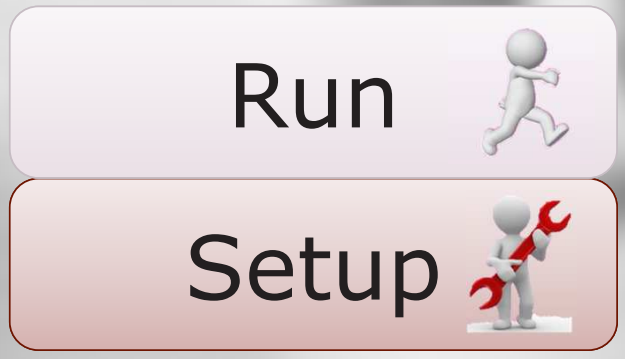
```
 12  end;
 13  /
```

Unit Testing

- Run the actual code
- Record the outcome

```
SQL> declare
  2    test_this varchar2(100);
  3    against_this varchar2(100);
  4  begin
  5    against_this := 'cde';
  6    test_this := betwnstr('abcdefgh',3,5);
```

```
 12  end;
 13  /
```



Unit Testing

```
SQL> declare
  2    test_this varchar2(100);
  3    against_this varchar2(100);
  4  begin
  5    against_this := 'cde';
  6    test_this := betwnstr('abcdefgh',3,5);
```

```
 12  end;
 13  /
```

Validate 

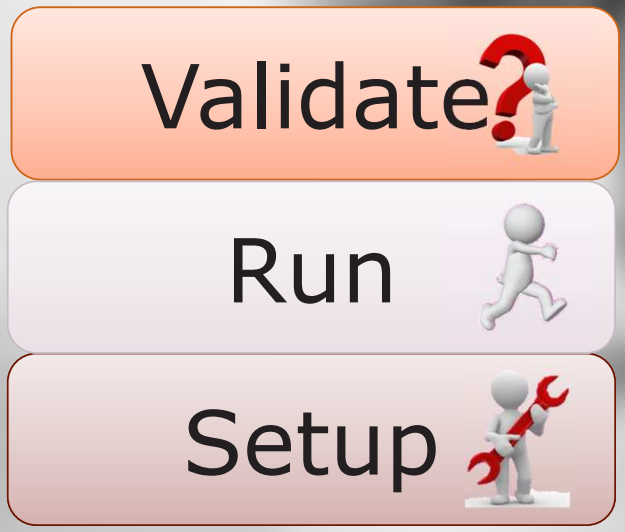
Run 

Setup 

Unit Testing

- Do the validation of the results

```
SQL> declare
  2    test_this varchar2(100);
  3    against_this varchar2(100);
  4  begin
  5    against_this := 'cde';
  6    test_this := betwnstr('abcdefgh',3,5);
  7    if test_this = against_this then
  8      dbms_output.put_line('OK');
  9    else
 10      dbms_output.put_line('NOT OK');
 11    end if;
 12  end;
 13  /
```



Unit Testing

```
SQL> declare
  2    test_this varchar2(100);
  3    against_this varchar2(100);
  4  begin
  5    against_this := 'cde';
  6    test_this := betwnstr('abcdefgh',3,5);
  7    if test_this = against_this then
  8      dbms_output.put_line('OK');
  9    else
 10      dbms_output.put_line('NOT OK');
 11    end if;
 12  end;
 13  /
```

Teardown

Validate

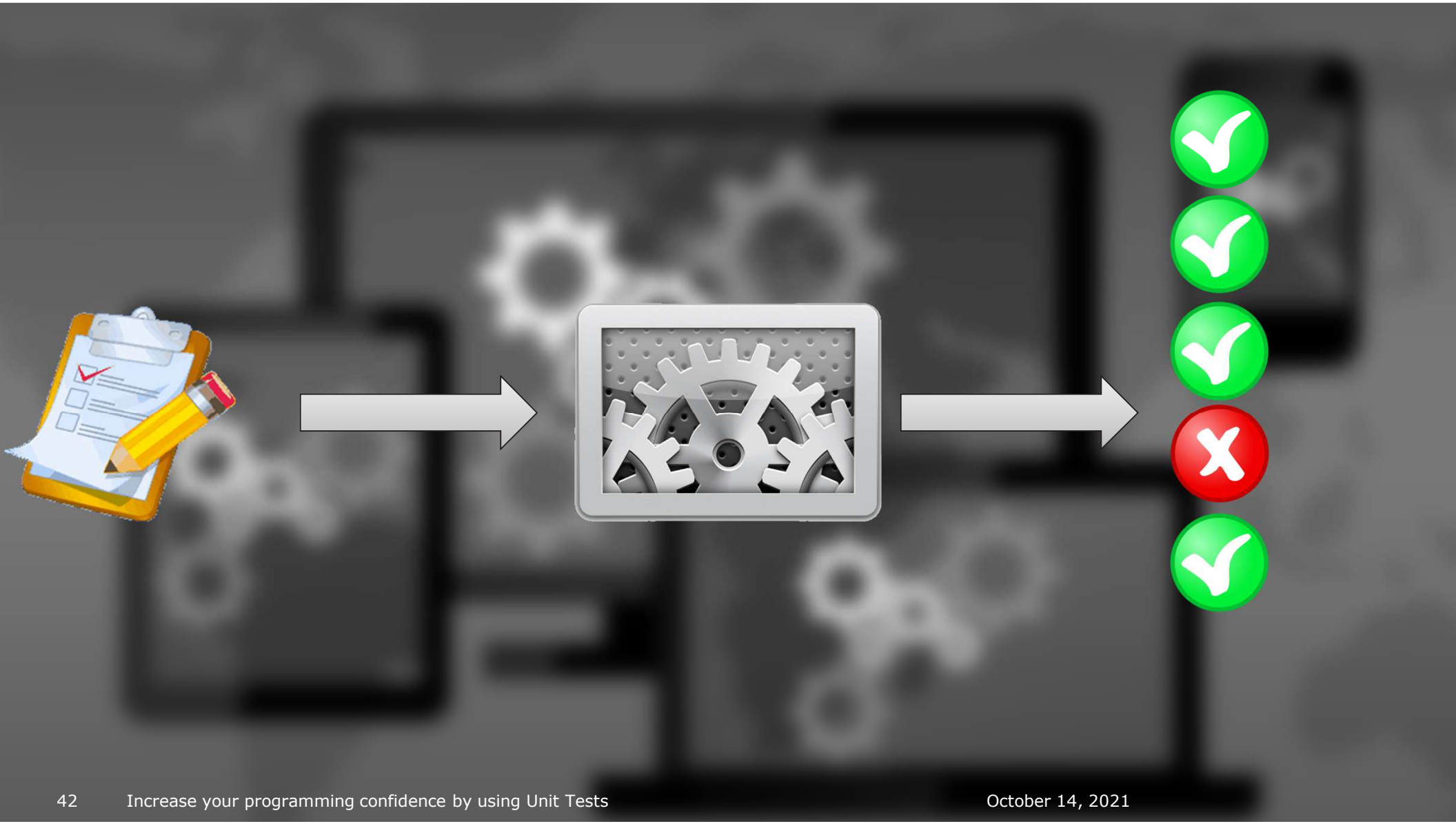
Run

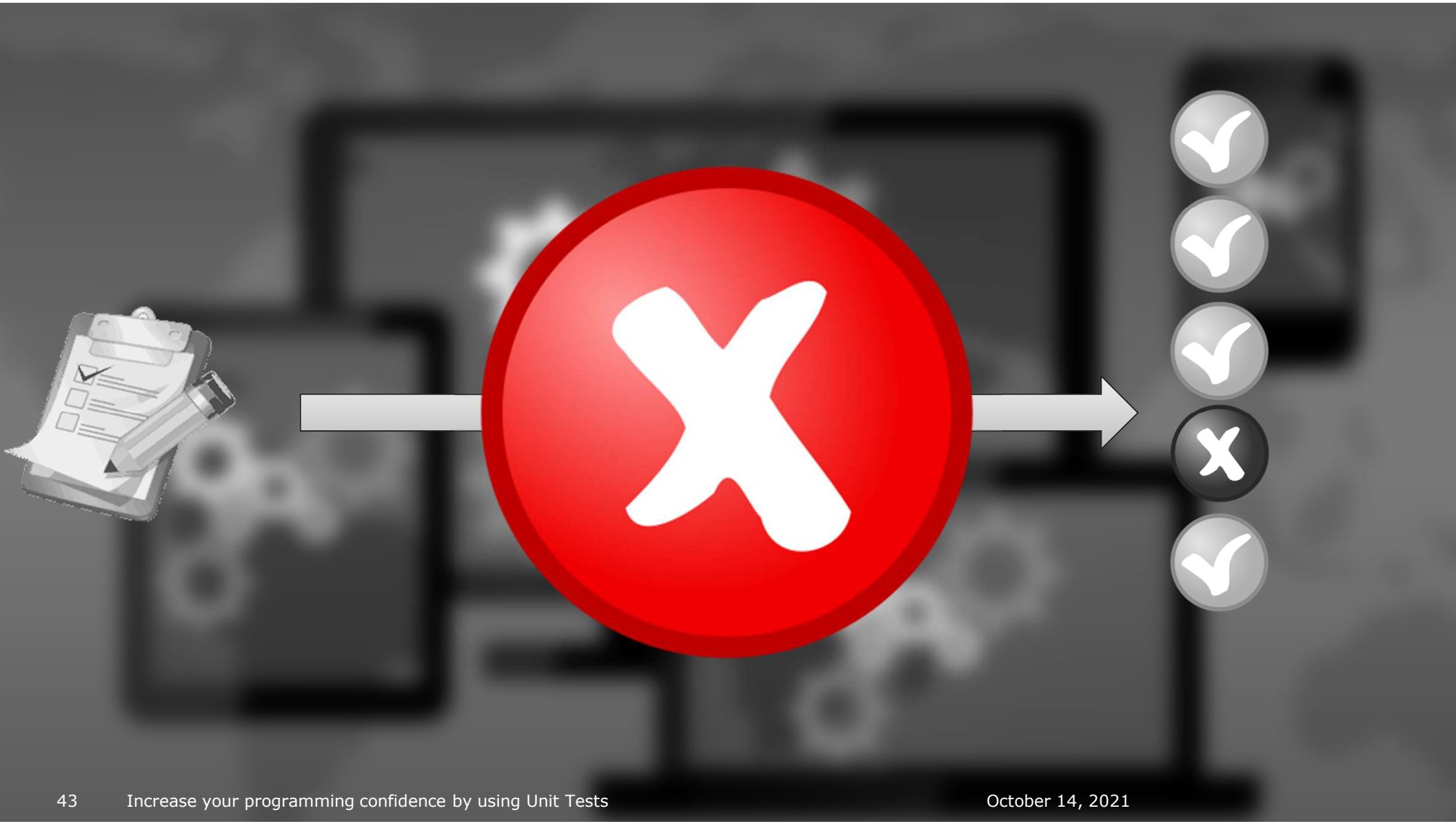
Setup

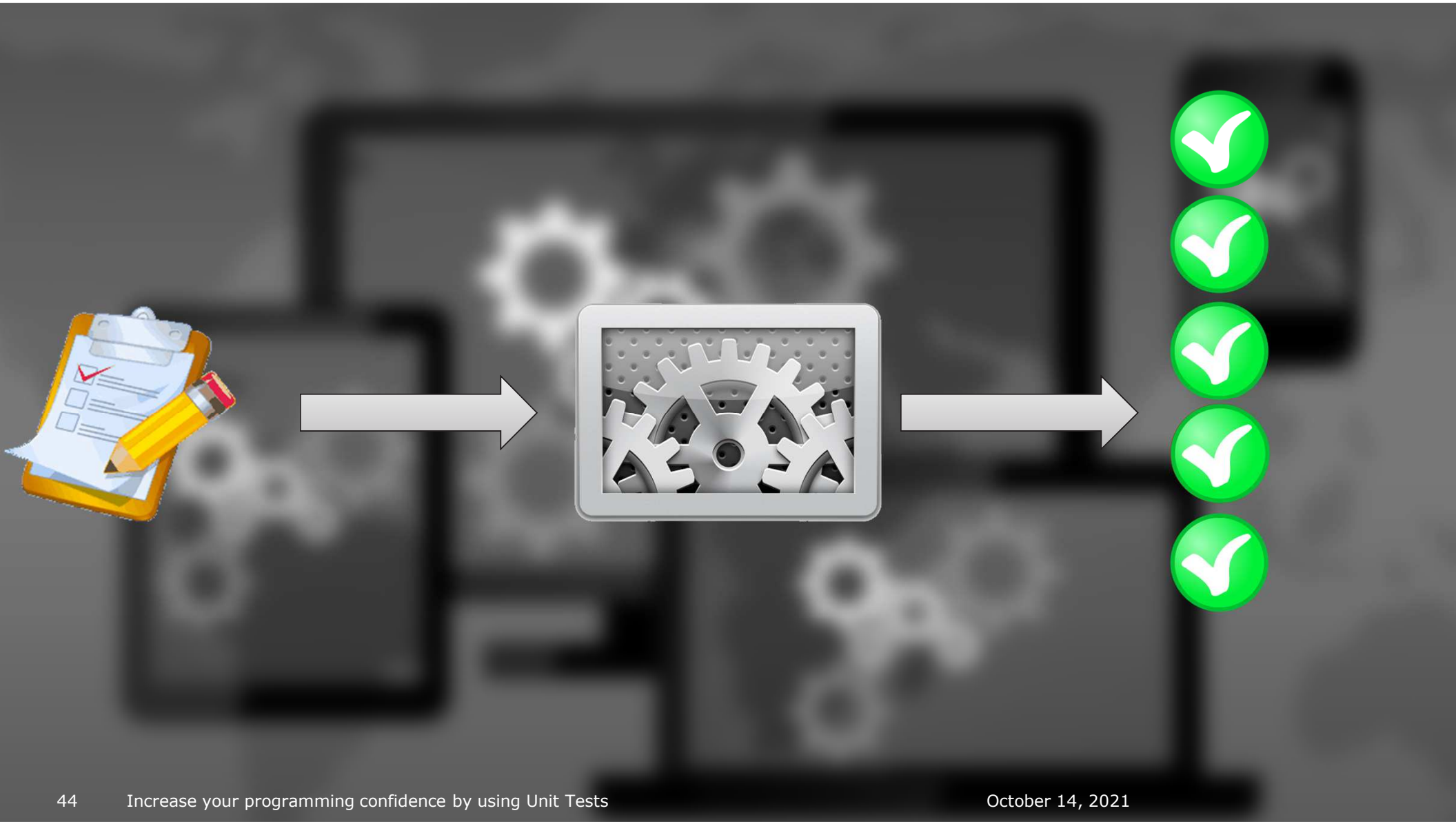
Unit Testing

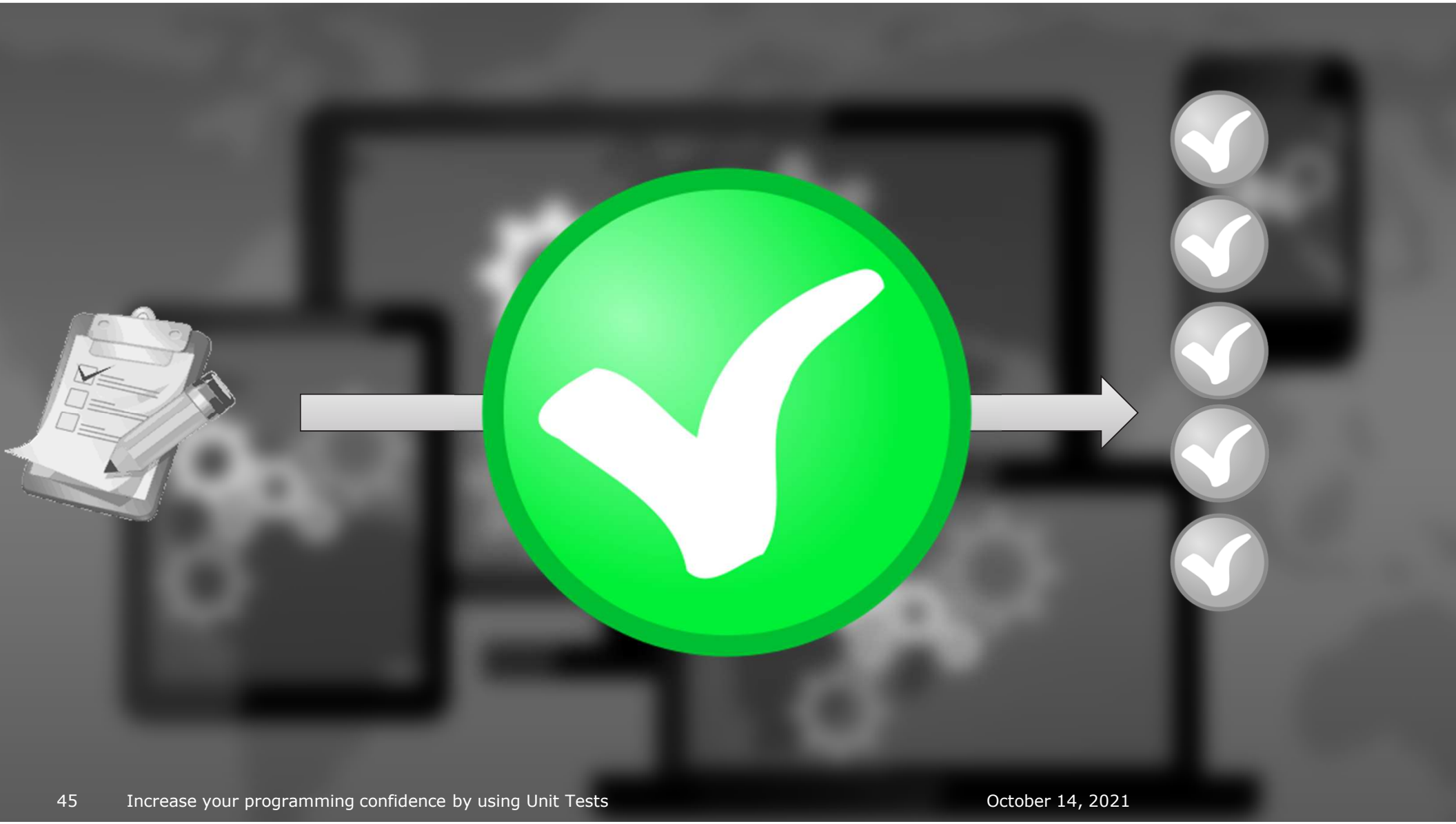
- Teardown everything you setup

```
SQL> declare
  2    test_this varchar2(100);
  3    against_this varchar2(100);
  4  begin
  5    against_this := 'cde';
  6    test_this := betwnstr('abcdefgh',3,5);
  7    if test_this = against_this then
  8      dbms_output.put_line('OK');
  9    else
 10      dbms_output.put_line('NOT OK');
 11    end if;
 12  end;
 13  /
```

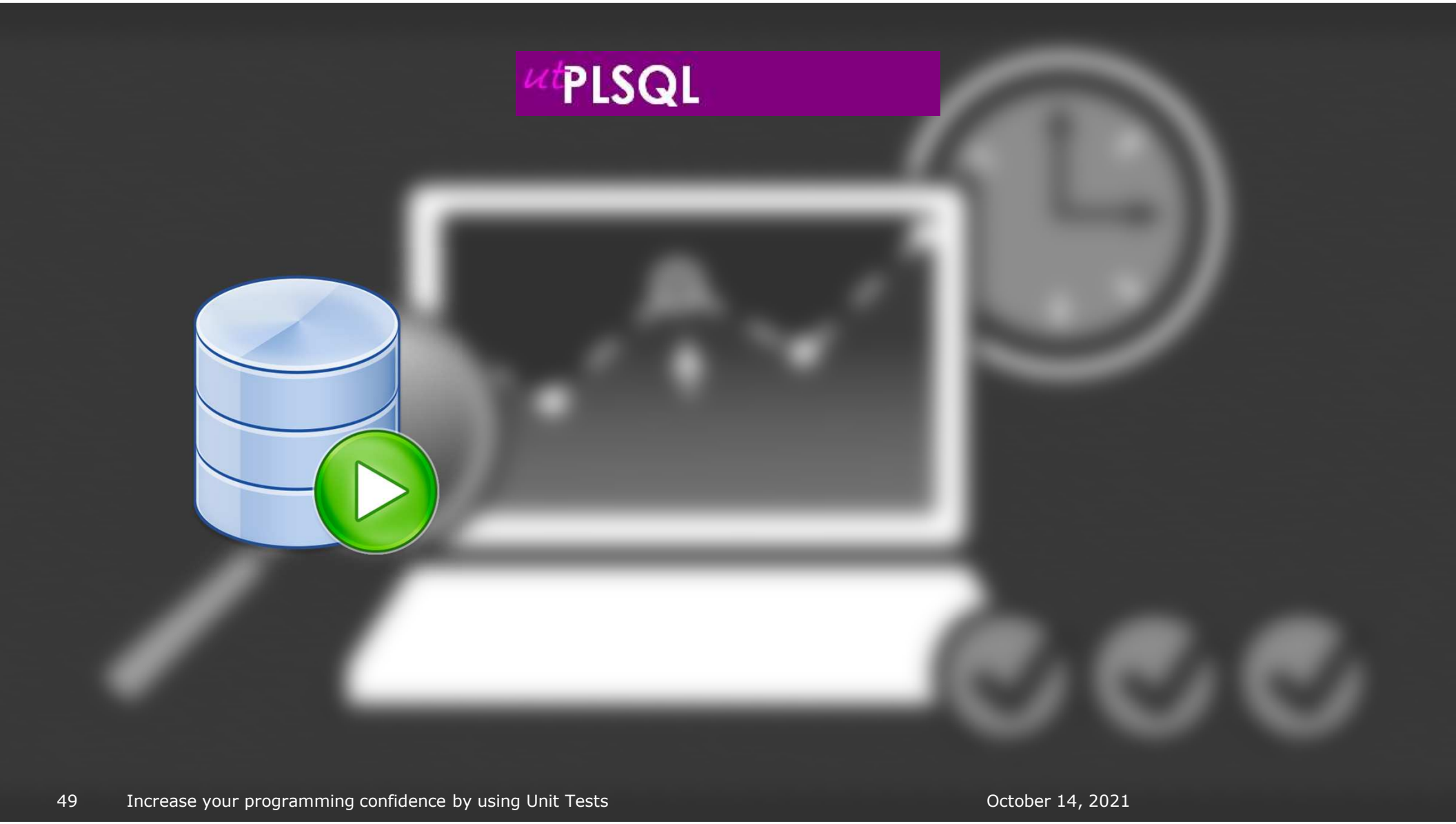




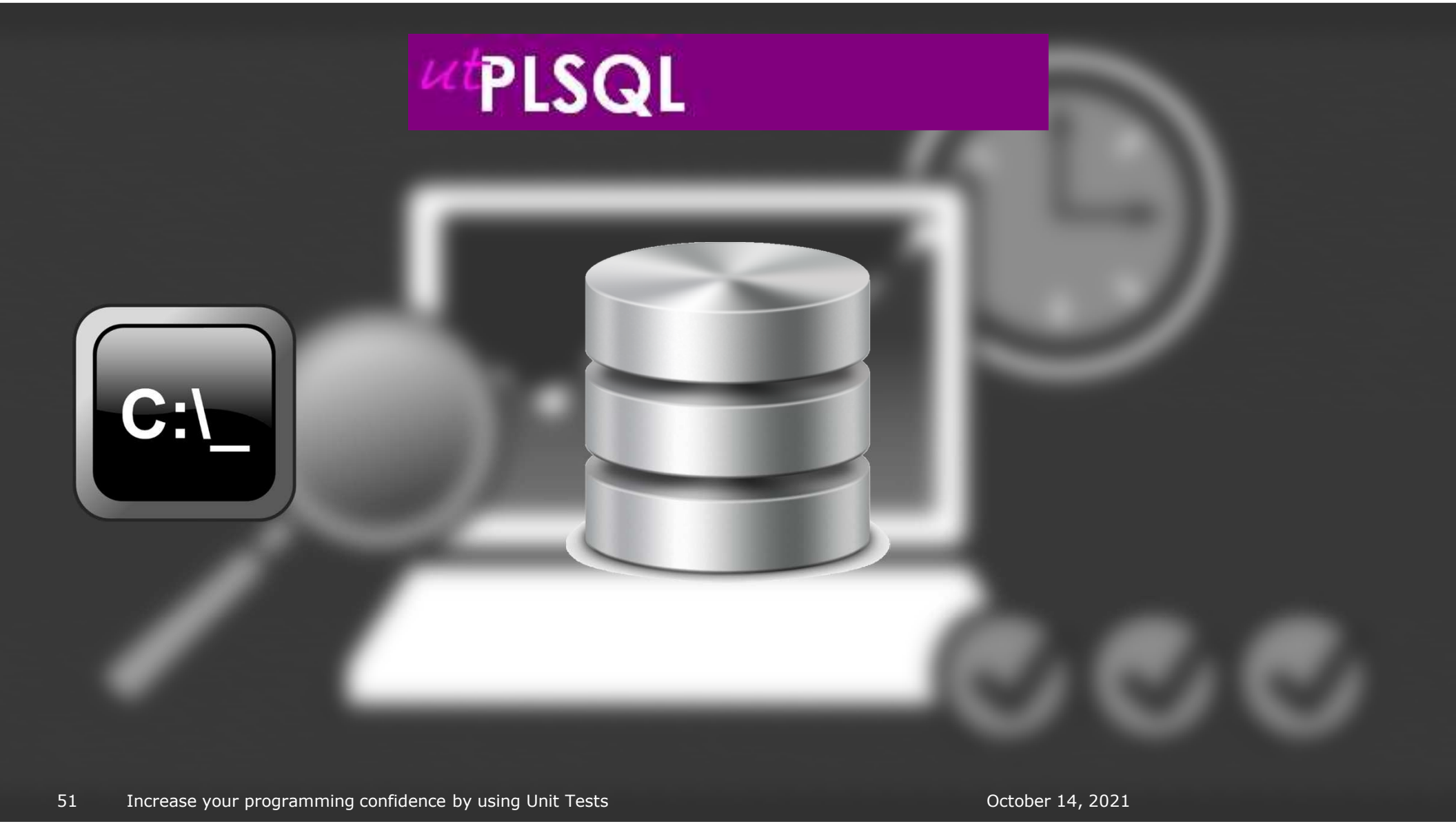


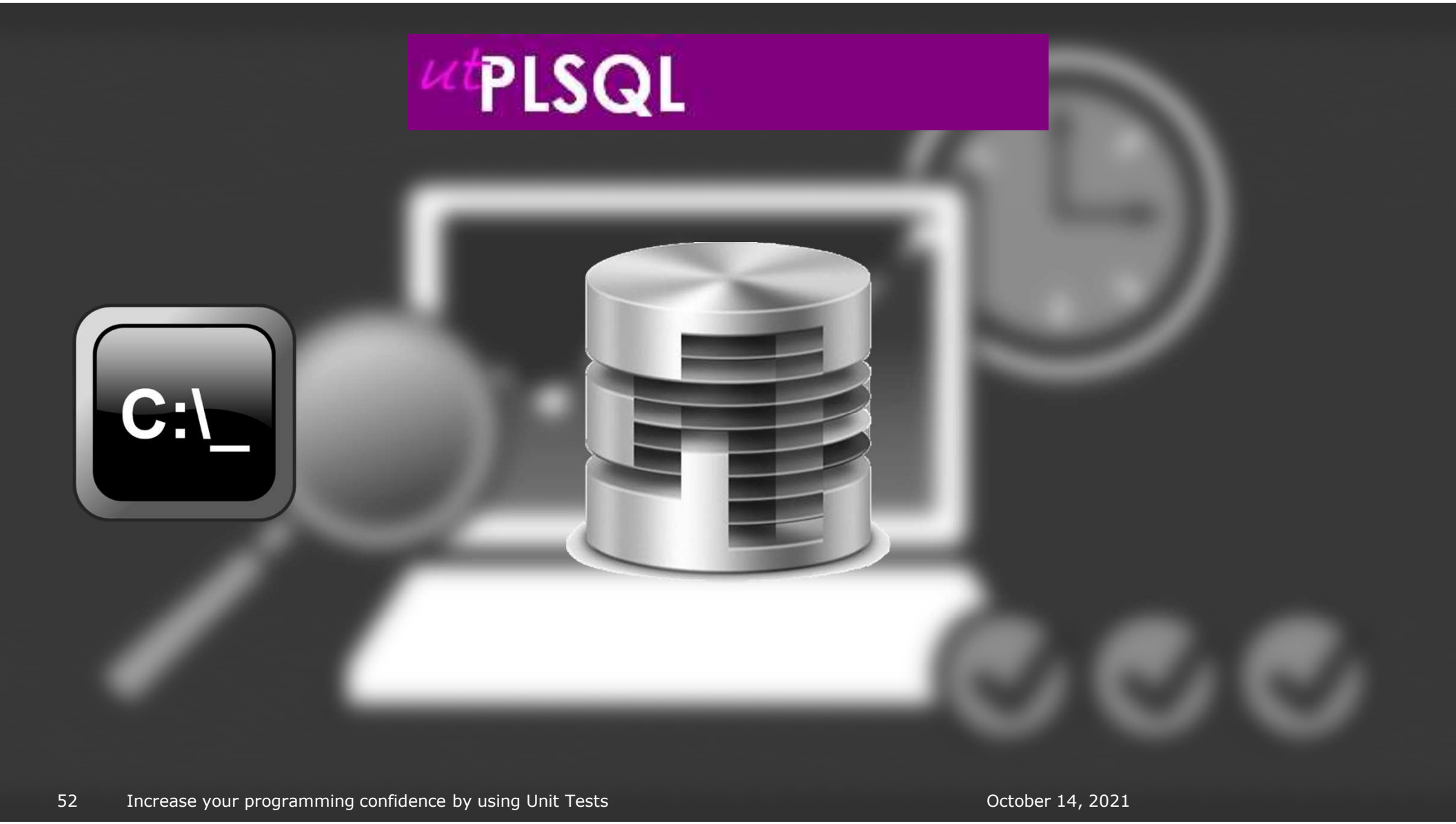


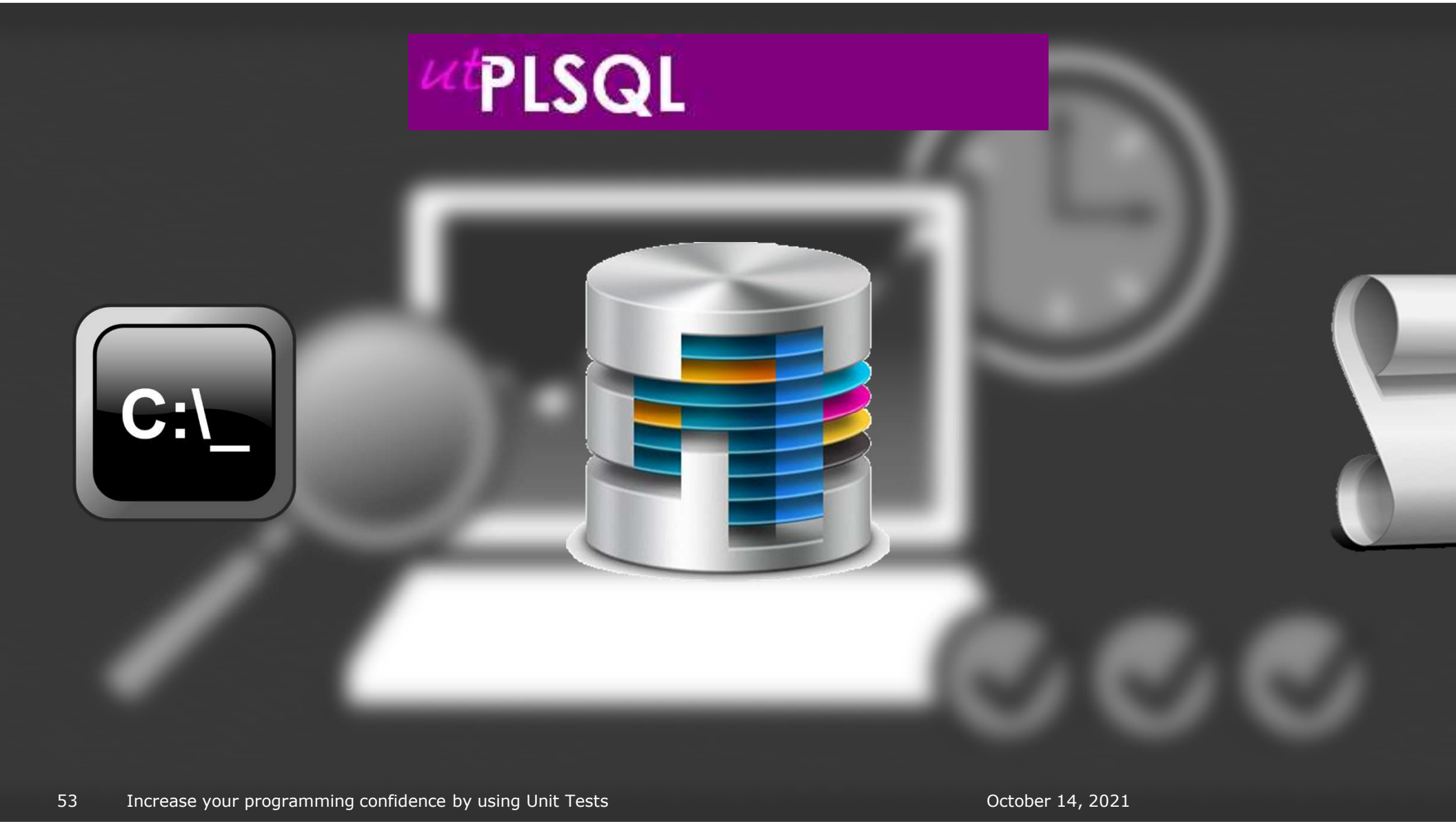

PLUTO
utPLSQL
PL/Unit
Quest Code Tester for Oracle
"Stop guessing and start testing with Quest Code Tester!" — Steven Feuerstein

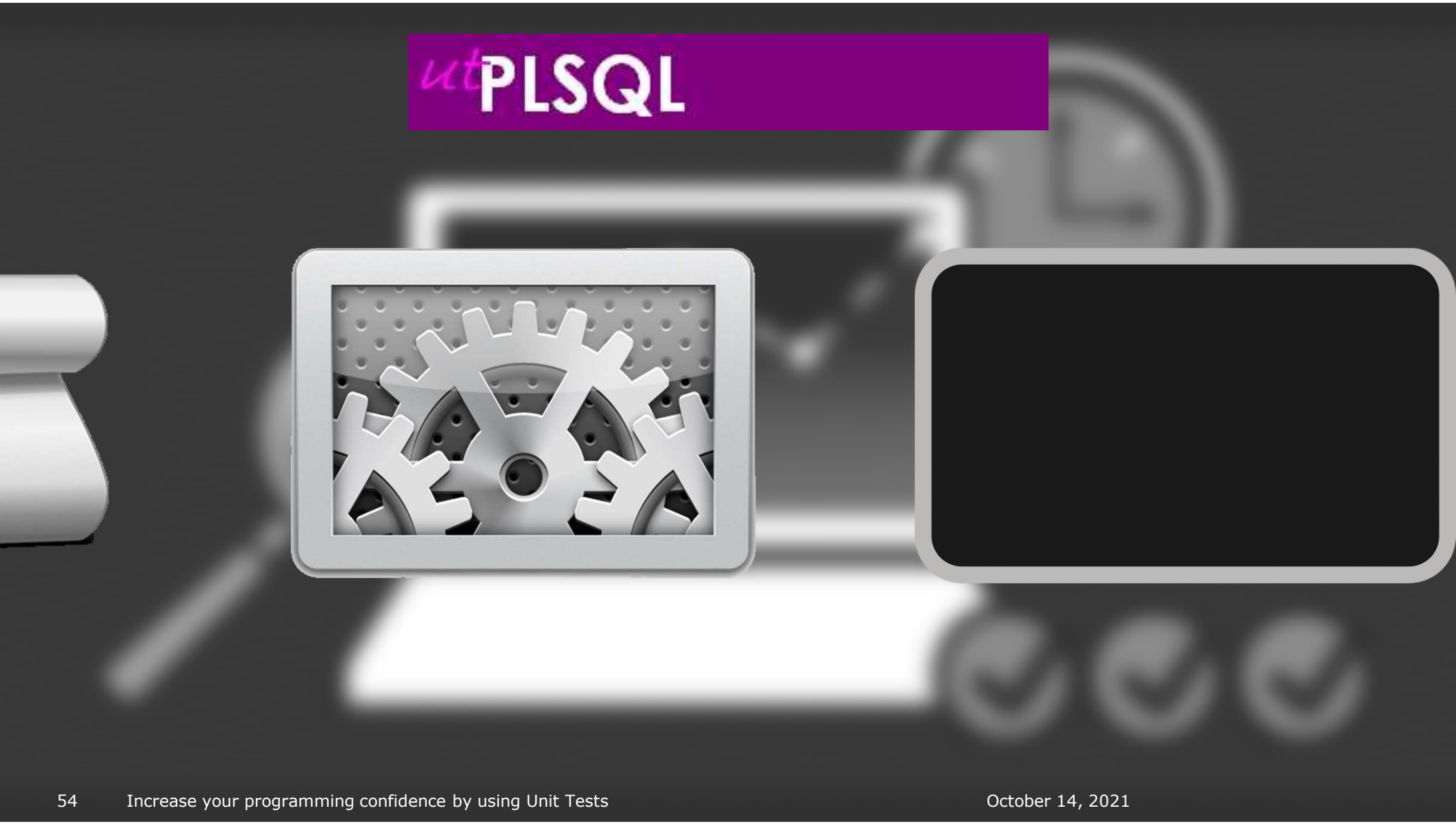

utPLSQL


utPLSQL


utPLSQL
C:_


utPLSQL
C:_


utPLSQL
C:_


utPLSQL

utPLSQL

```
.
>  FFFFFFF   AA     III  L      U     U RRRRR   EEEEEEE
>  F        A A     I   L      U     U R    R  E
>  F       A   A    I   L      U     U R     R E
>  F      A     A   I   L      U     U R     R E
>  FFFF   A     A   I   L      U     U RRRRRR  EEEE
>  F      AAAAAAA   I   L      U     U R   R   E
>  F      A     A   I   L      U     U R    R  E
>  F      A     A   I   L       U   U  R     R E
>  F      A     A  III  LLLLLLL  UUU   R     R EEEEEEE
.
FAILURE: ".ut_betwnstr"
.
> Individual Test Case Results:
>
FAILURE - ut_betwnstr.UT_BETWNSTR: EQ "Typical valid usage"
Expected "cd" and
got "cde"
>
SUCCESS - ut_betwnstr.UT_BETWNSTR: EQ "Zero start" Expected
"ab" and got "ab"
```

utPLSQL

```
.
>  FFFFFFF   AA     III  L      U     U RRRRR   EEEEEEE
>  F        A A     I   L      U     U R    R  E
>  F       A   A    I   L      U     U R     R E
>  F      A     A   I   L      U     U R     R E
>  FFFF   A     A   I   L      U     U RRRRRR  EEEE
>  F      AAAAAAA   I   L      U     U R   R   E
>  F      A     A   I   L      U     U R    R  E
>  F      A     A   I   L       U   U  R     R E
>  F      A     A  III  LLLLLLL  UUU   R     R EEEEEEE
.
FAILURE: ".ut_betwnstr"
.
> Individual Test Case Results:
>
FAILURE - ut_betwnstr.UT_BETWNSTR: EQ "Typical valid usage"
Expected "cd" and
got "cde"
>
SUCCESS - ut_betwnstr.UT_BETWNSTR: EQ "Zero start" Expected
"ab" and got "ab"
```

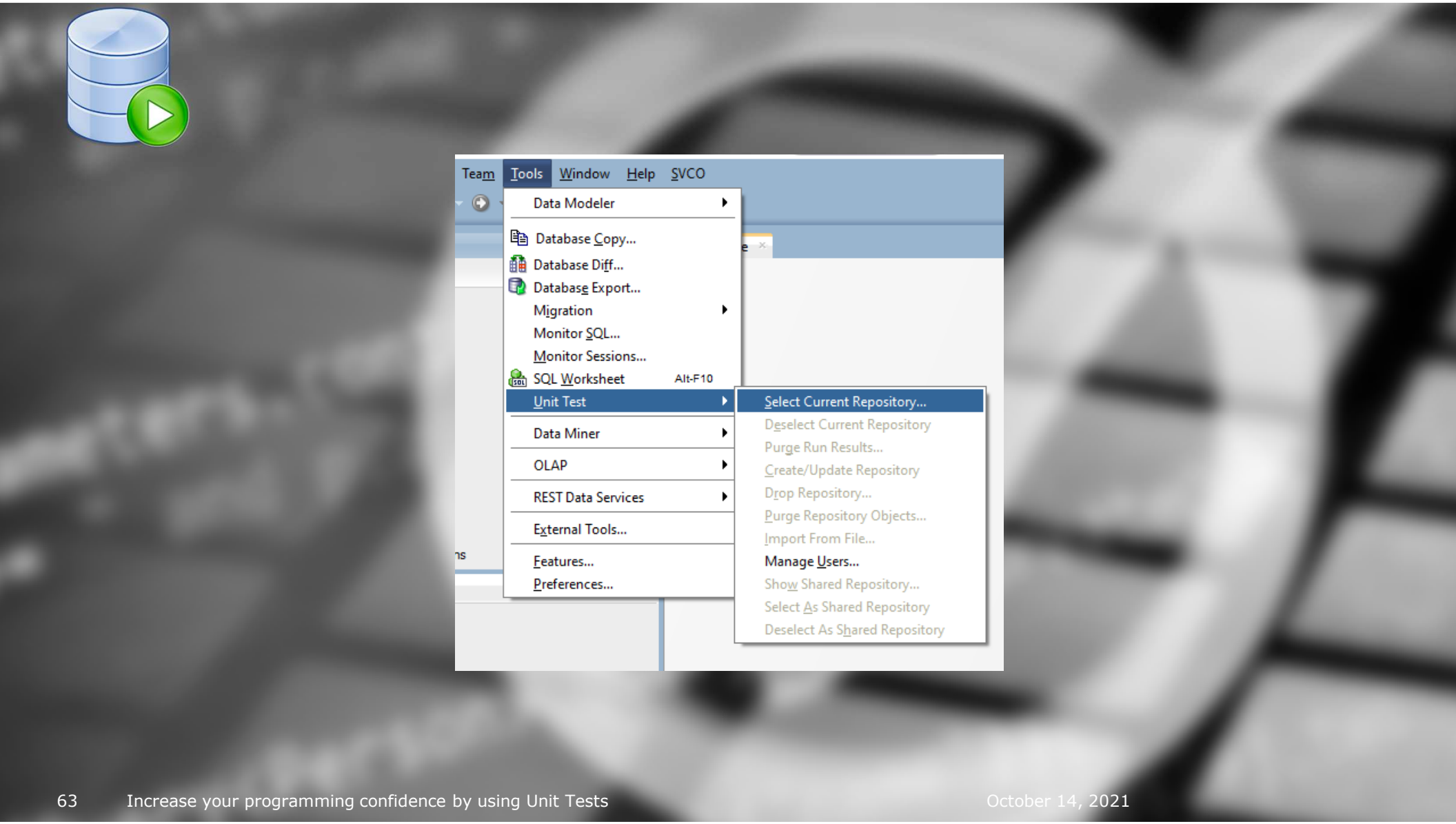
utPLSQL

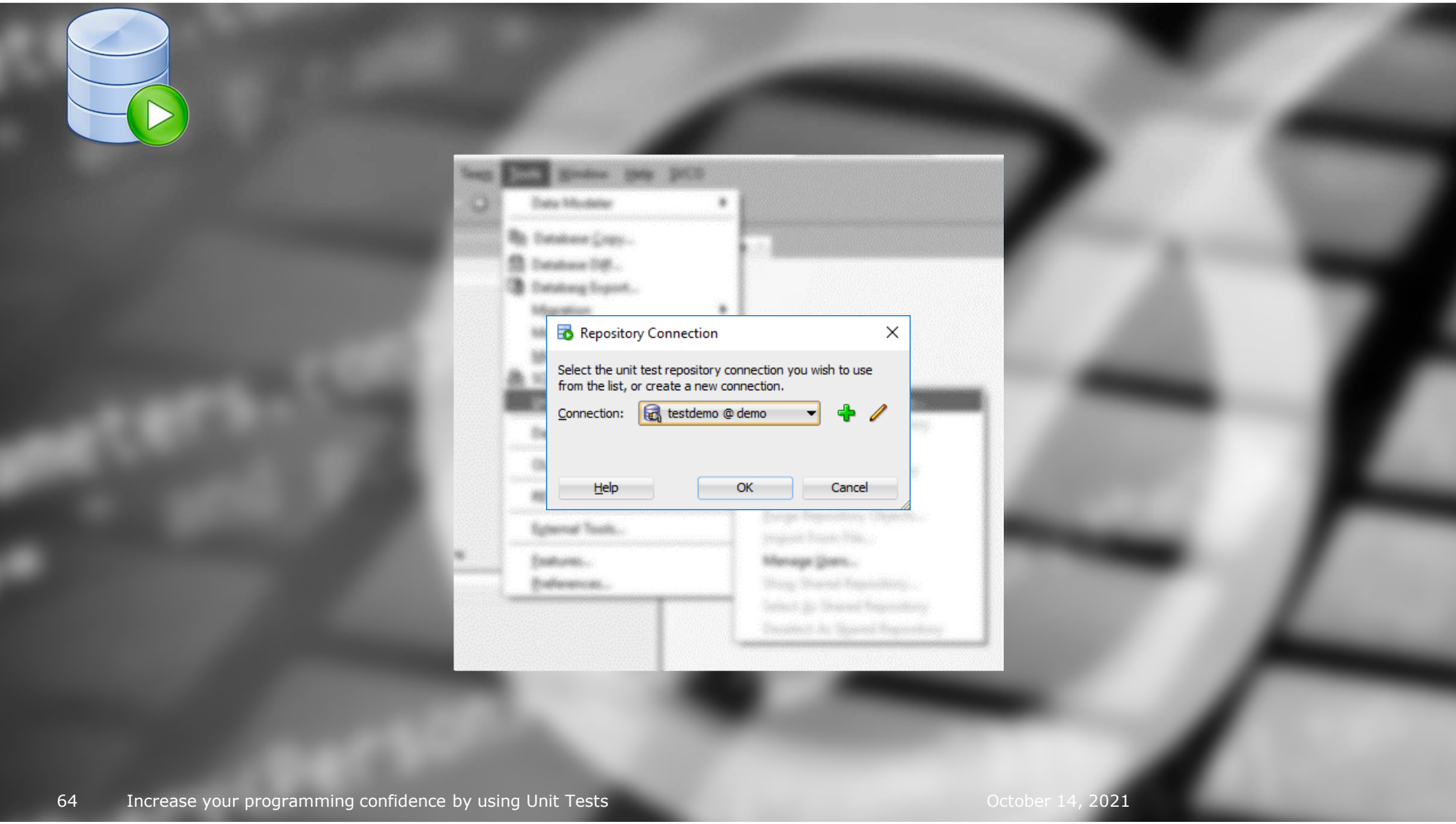
```
.
>    SSSS   U     U   CCC     CCC   EEEEEEE   SSSS     SSSS
>   S    S  U     U  C   C   C   C  E        S    S   S    S
>  S       U     U C     C C     C E       S       S
>   S      U     U C       C       E       S        S
>    SSSS  U     U C       C       EEEE     SSSS     SSSS
>        S U     U C       C       E           S        S
>         S U     U C     C C     C E           S        S
>   S    S  U   U   C   C   C   C  E        S    S   S    S
>    SSSS     UUU     CCC     CCC   EEEEEEE   SSSS     SSSS
.
SUCCESS: ".ut_betwnstr"
.
> Individual Test Case Results:
>
SUCCESS - ut_betwnstr.UT_BETWNSTR: EQ "Typical valid usage"
Expected "cde" and
got "cde"
>
SUCCESS - ut_betwnstr.UT_BETWNSTR: EQ "Zero start" Expected
"ab" and got "ab"
```

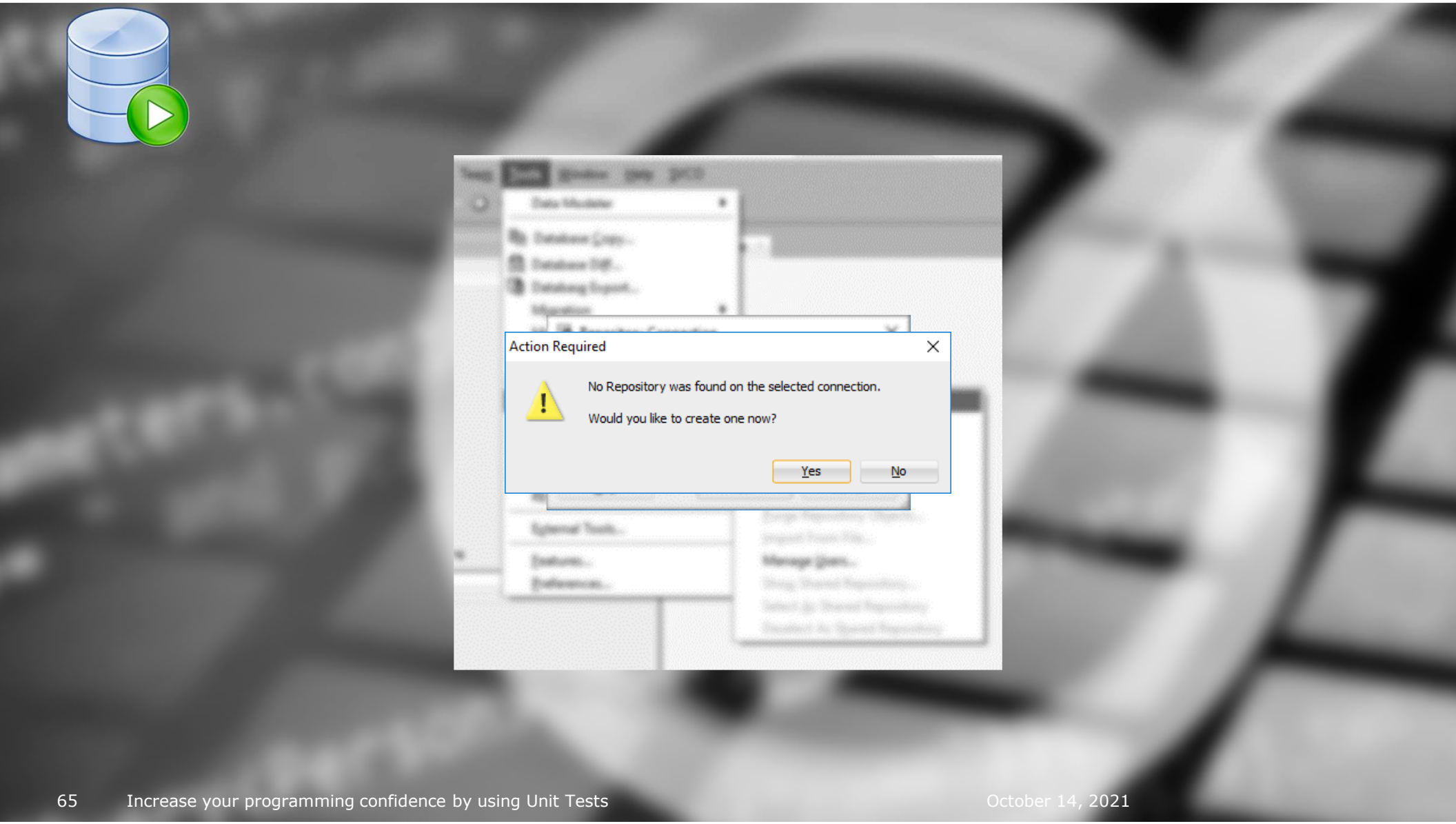
utPLSQL

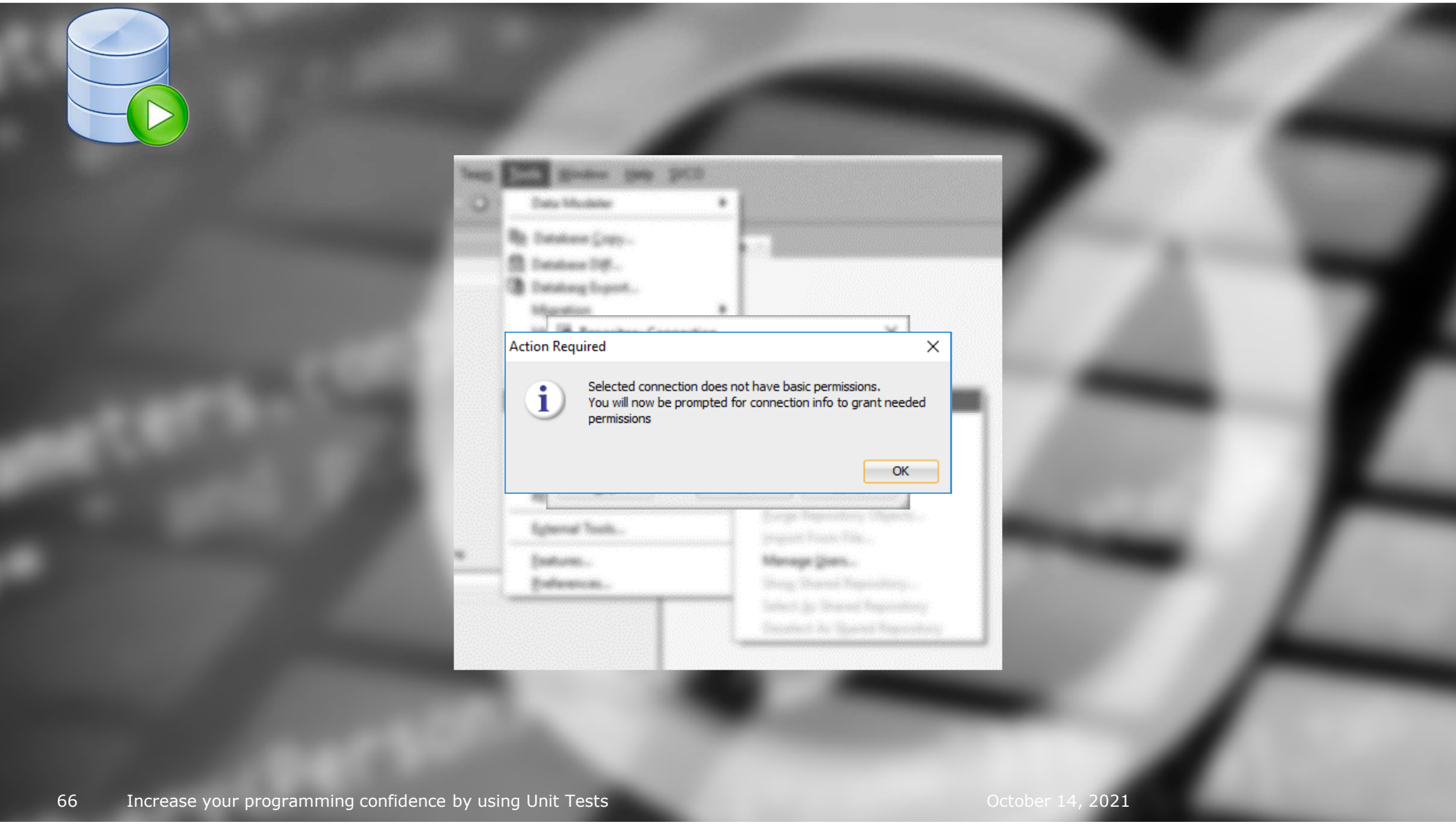

if(parameters.contains(
hql +=
and p.name =
}
if(parameters.contains(
hql +=
and p.
}
TypedQuery<Person>

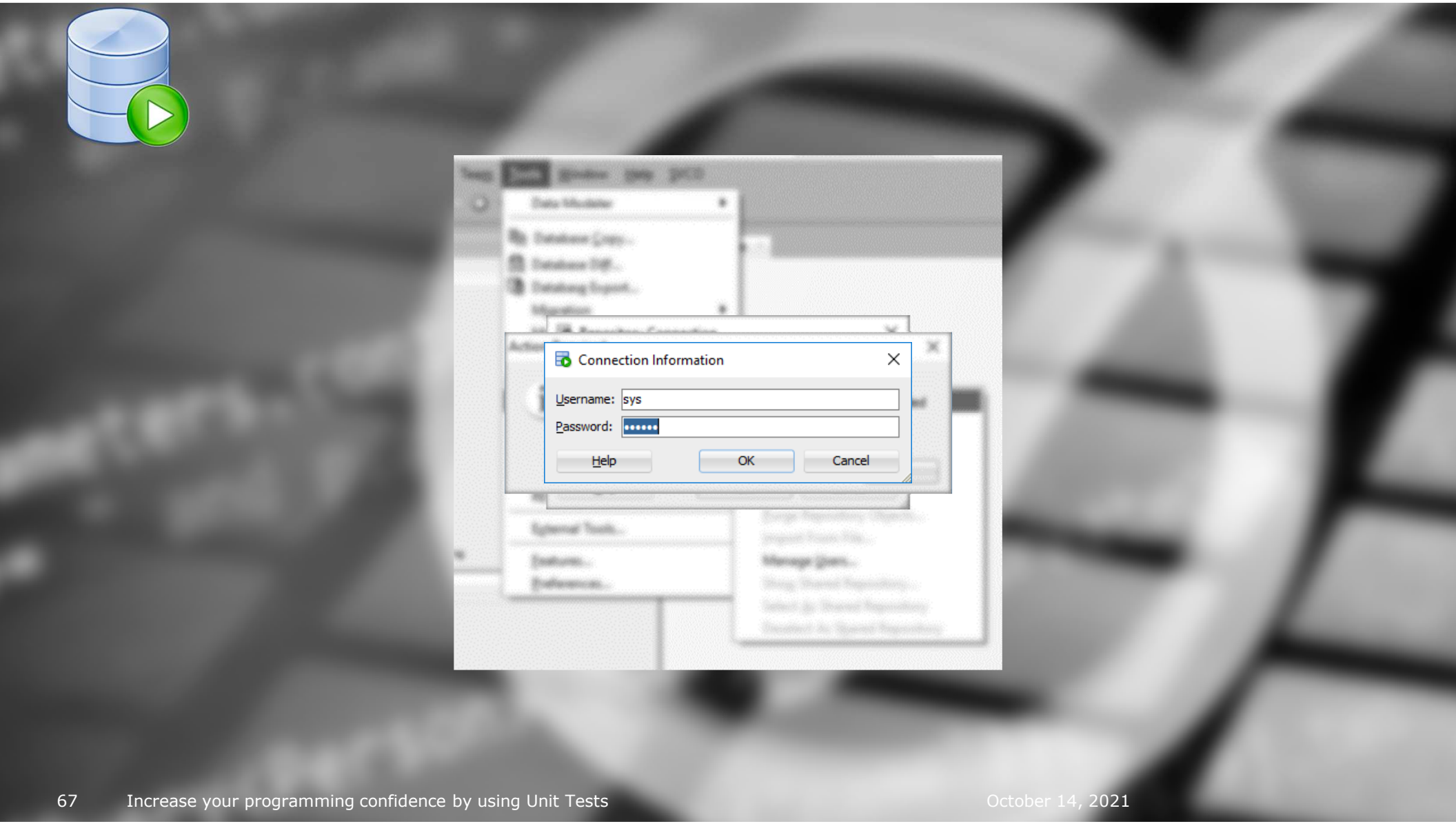

if(parameters.contains(
hql += " and p.name = :name
}
if(parameters.contains(
hql += " and p.
}
TypedQuery<Person>

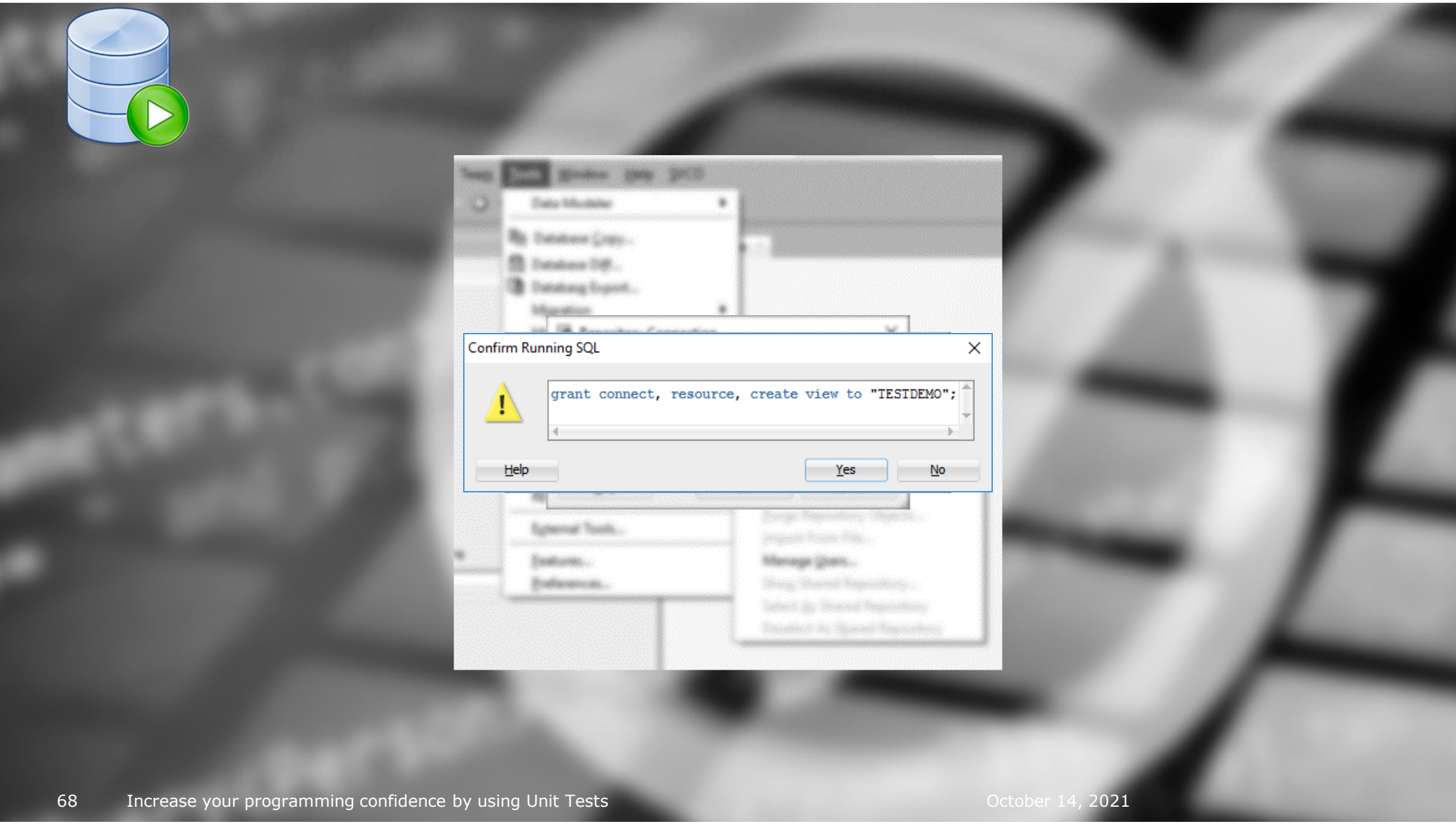

Team
Tools
Window
Help
SVCO
Data Modeler
Database Copy...
Database Diff...
Database Export...
Migration
Monitor SQL...
Monitor Sessions...
SQL Worksheet
Alt-F10
Unit Test
Data Miner
OLAP
REST Data Services
External Tools...
Features...
Preferences...
Select Current Repository...
Deselect Current Repository
Purge Run Results...
Create/Update Repository
Drop Repository...
Purge Repository Objects...
Import From File...
Manage Users...
Show Shared Repository...
Select As Shared Repository
Deselect As Shared Repository

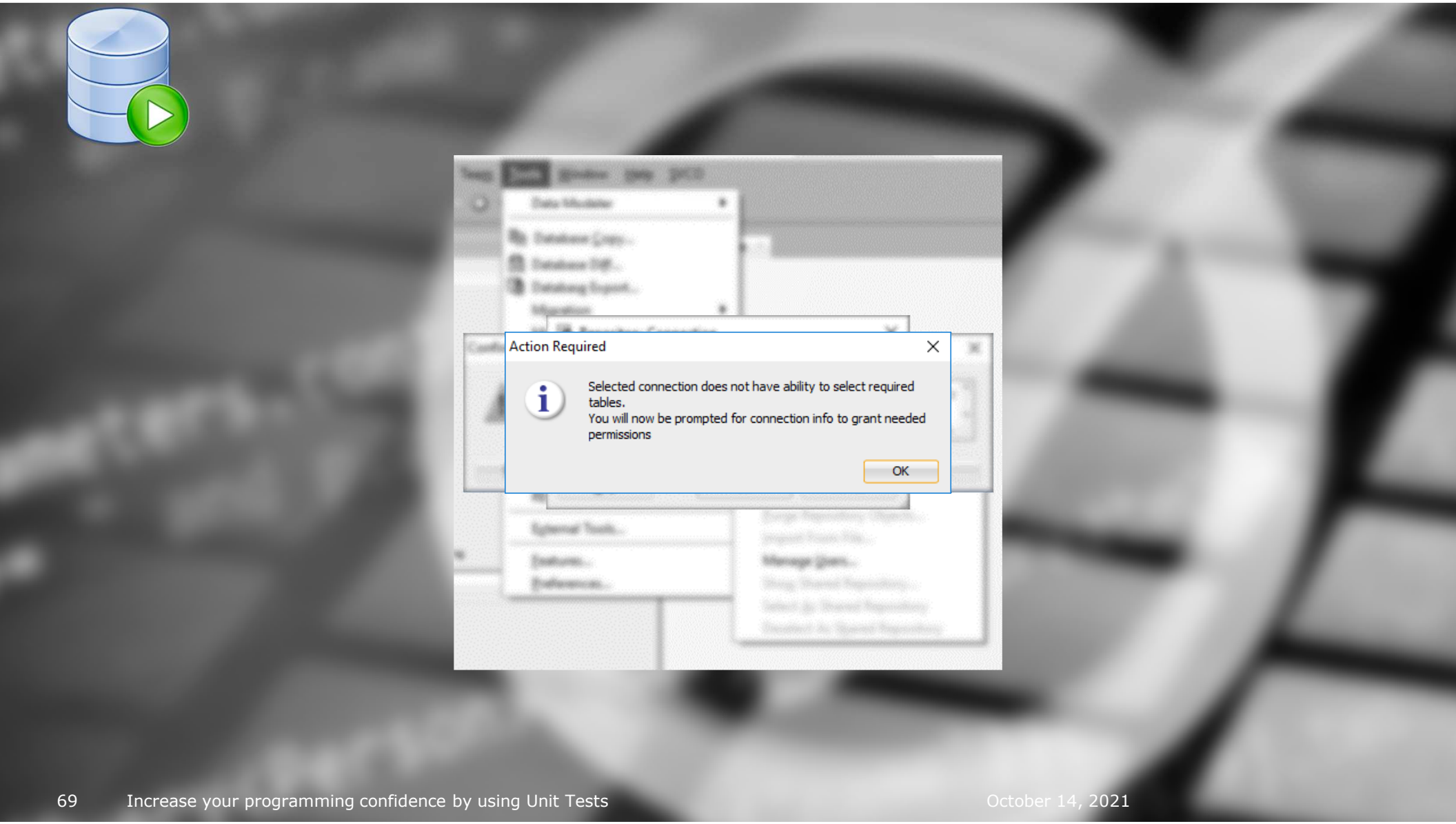

Repository Connection
Select the unit test repository connection you wish to use from the list, or create a new connection.
Connection:
testdemo @ demo
Help
OK
Cancel

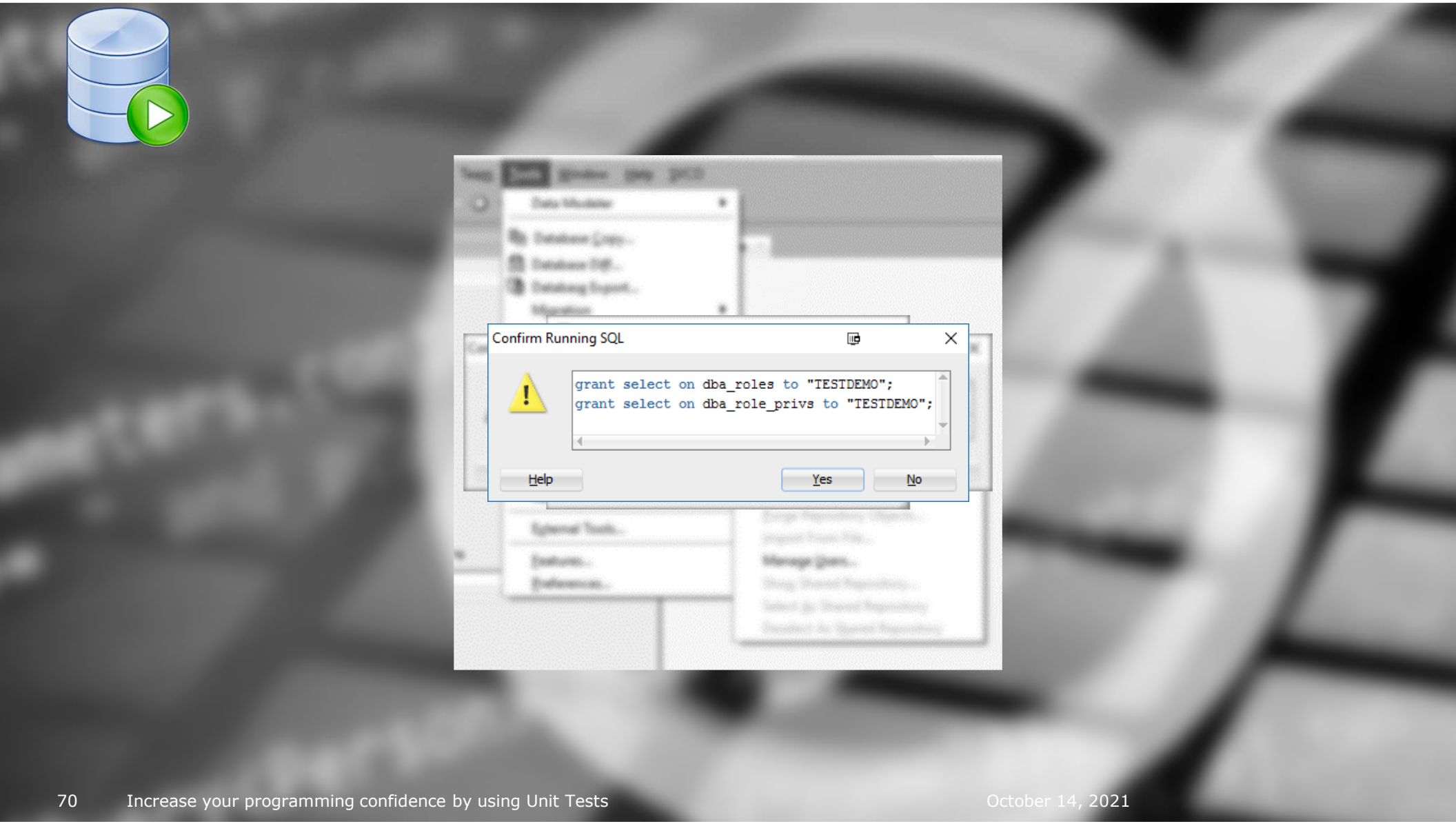

Action Required
No Repository was found on the selected connection.
Would you like to create one now?
Yes
No

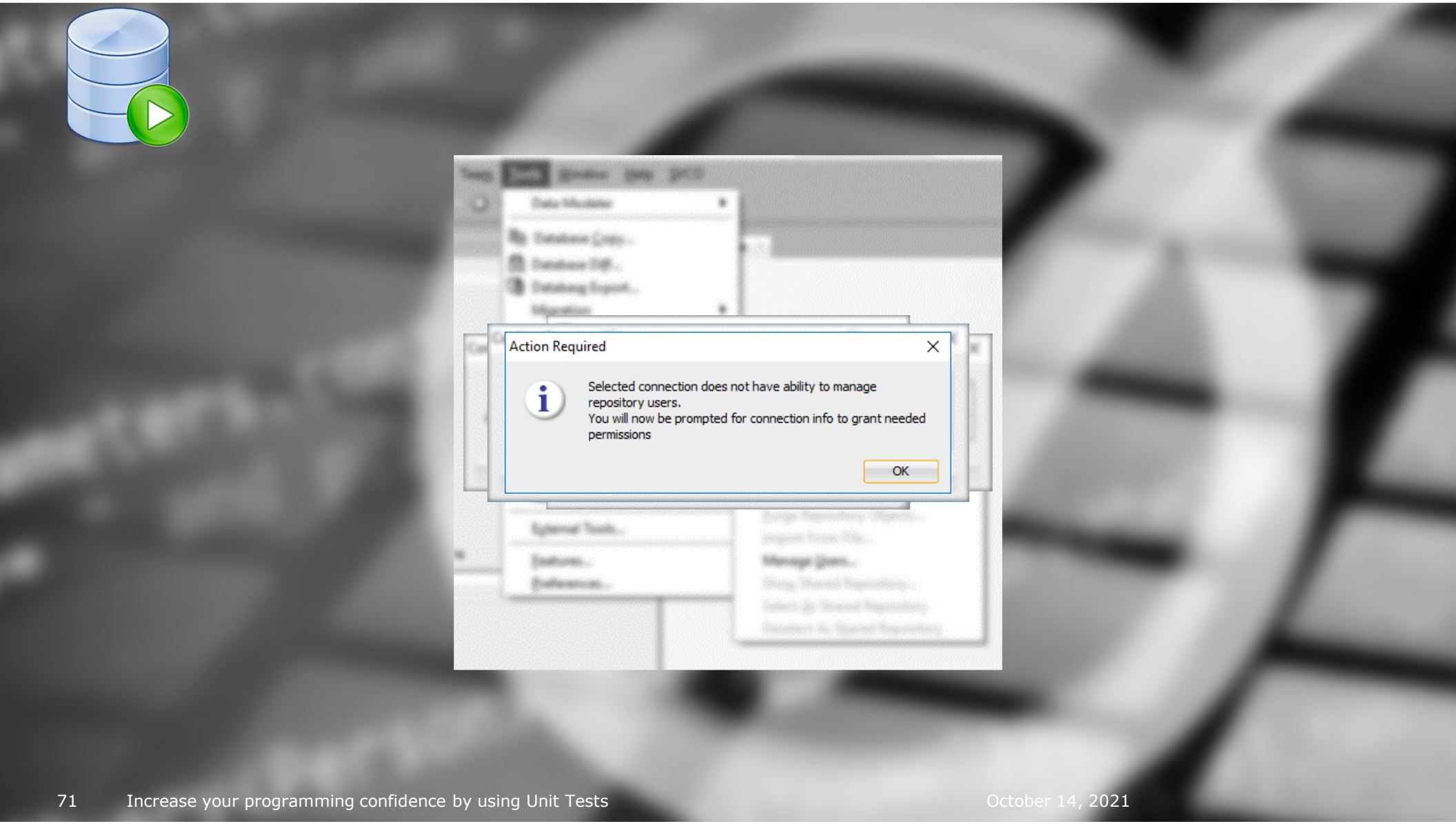

Action Required
Selected connection does not have basic permissions.
You will now be prompted for connection info to grant needed permissions
OK

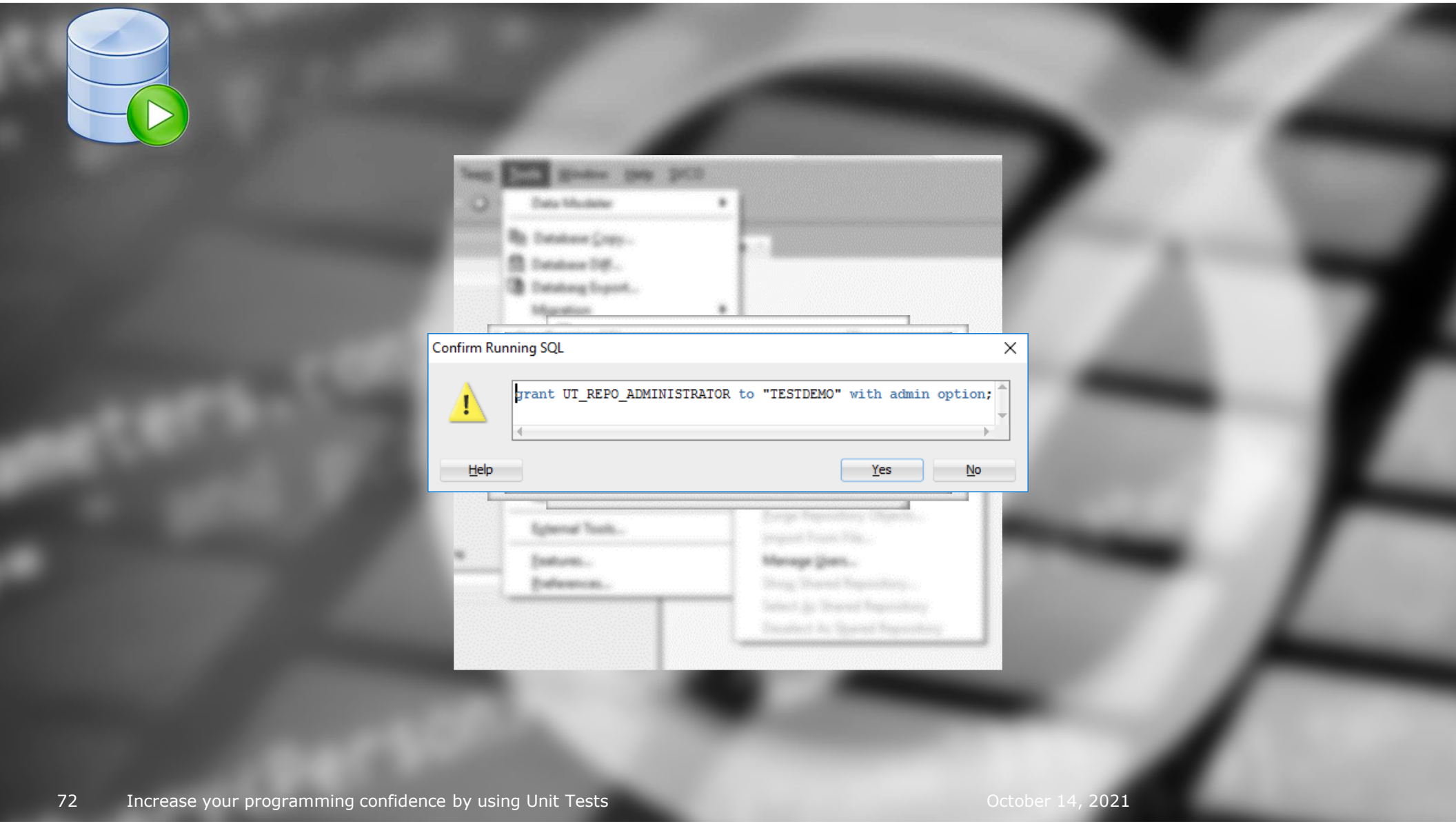

Connection Information
Username: sys
Password:
Help
OK
Cancel

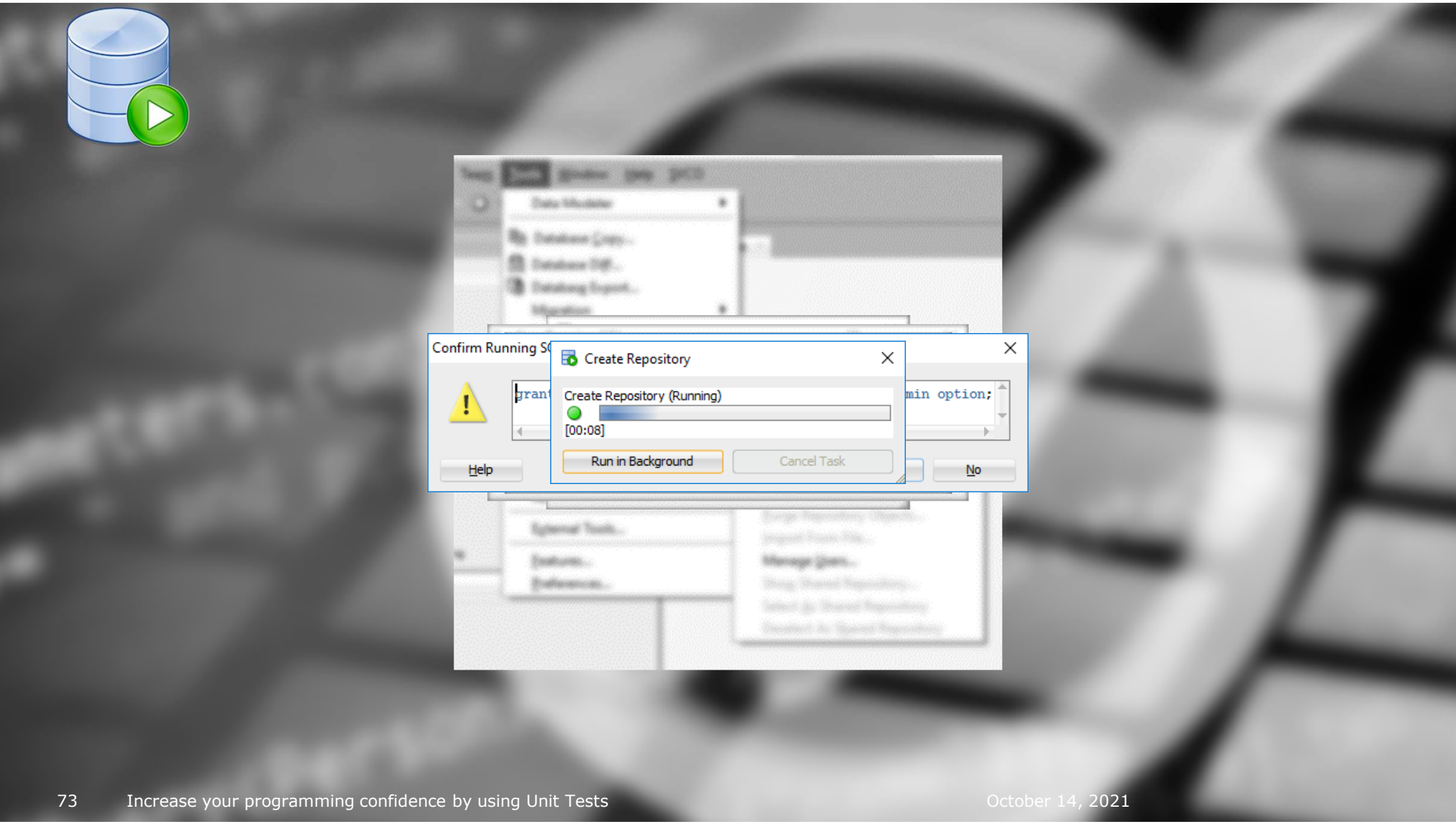

Confirm Running SQL
grant connect, resource, create view to "TESTDEMO";
Help
Yes
No

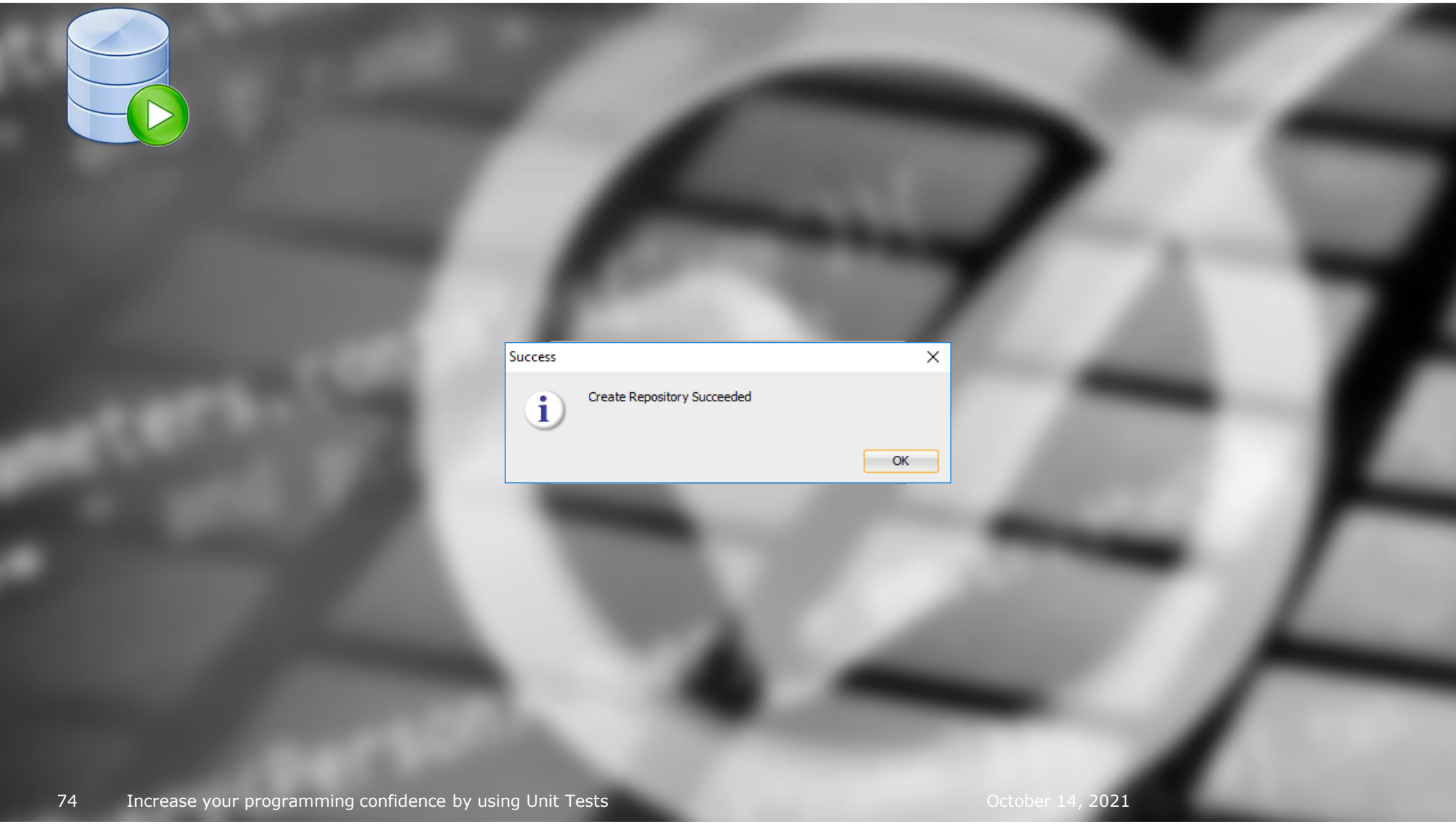

Action Required
Selected connection does not have ability to select required tables.
You will now be prompted for connection info to grant needed permissions
OK

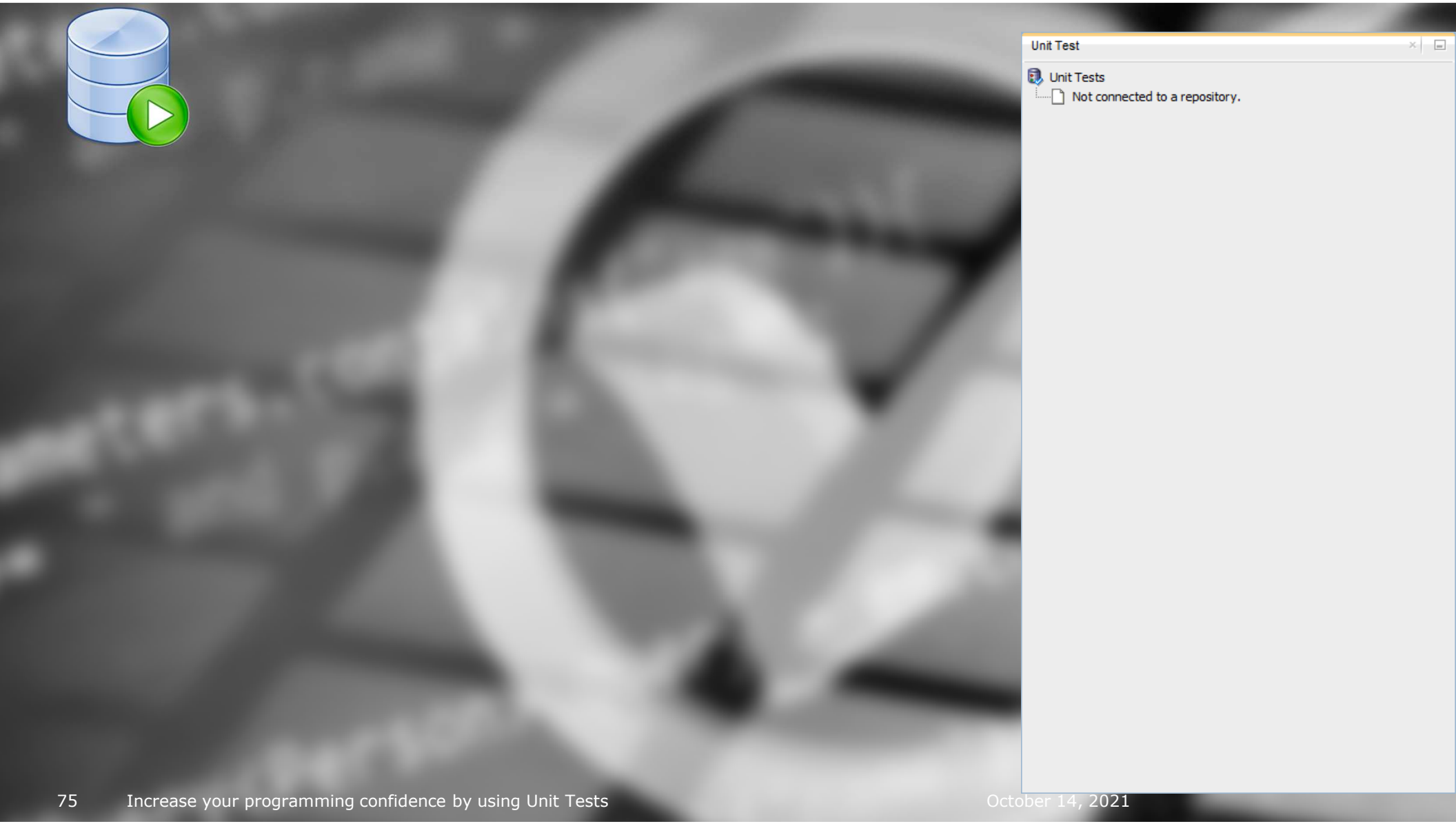

Confirm Running SQL
grant select on dba_roles to "TESTDEMO";
grant select on dba_role_privs to "TESTDEMO";
Help
Yes
No

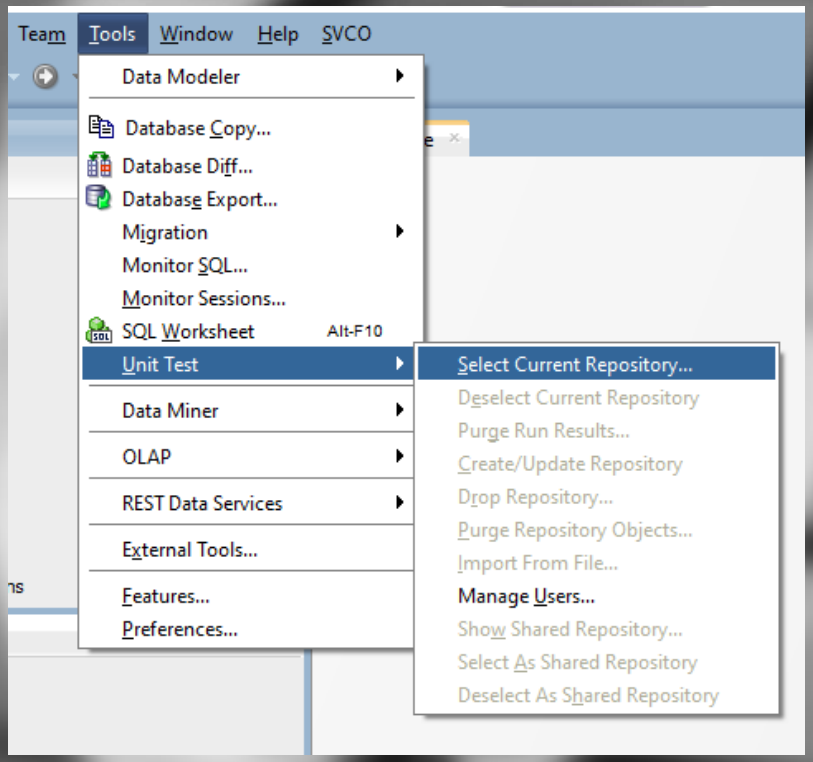

Action Required
Selected connection does not have ability to manage repository users.
You will now be prompted for connection info to grant needed permissions
OK

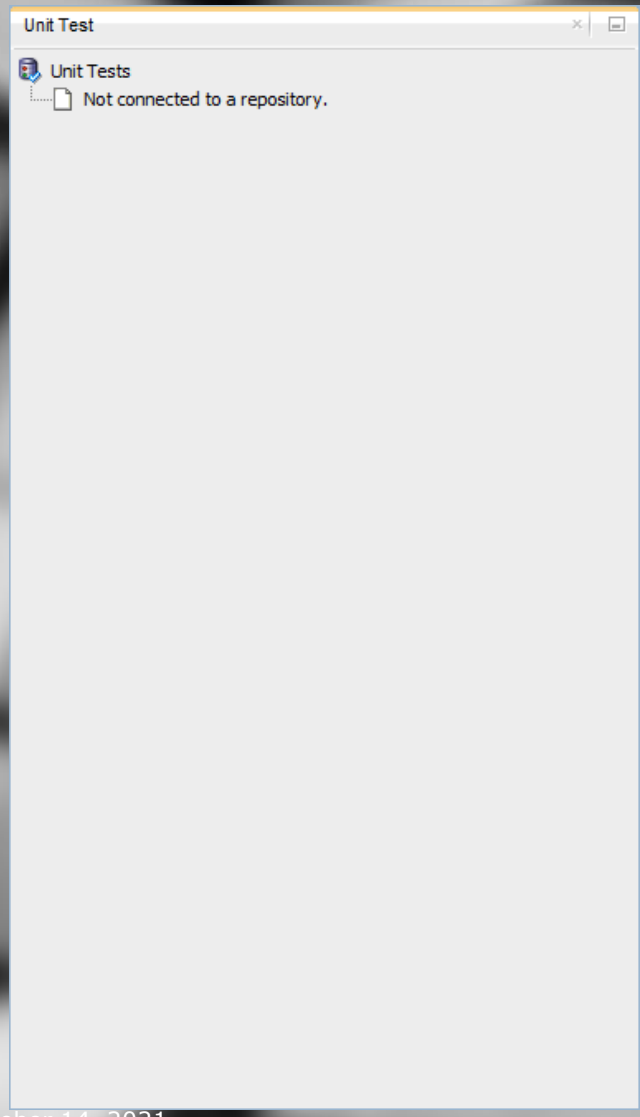

Confirm Running SQL
grant UT_REPO_ADMINISTRATOR to "TESTDEMO" with admin option;
Help
Yes
No

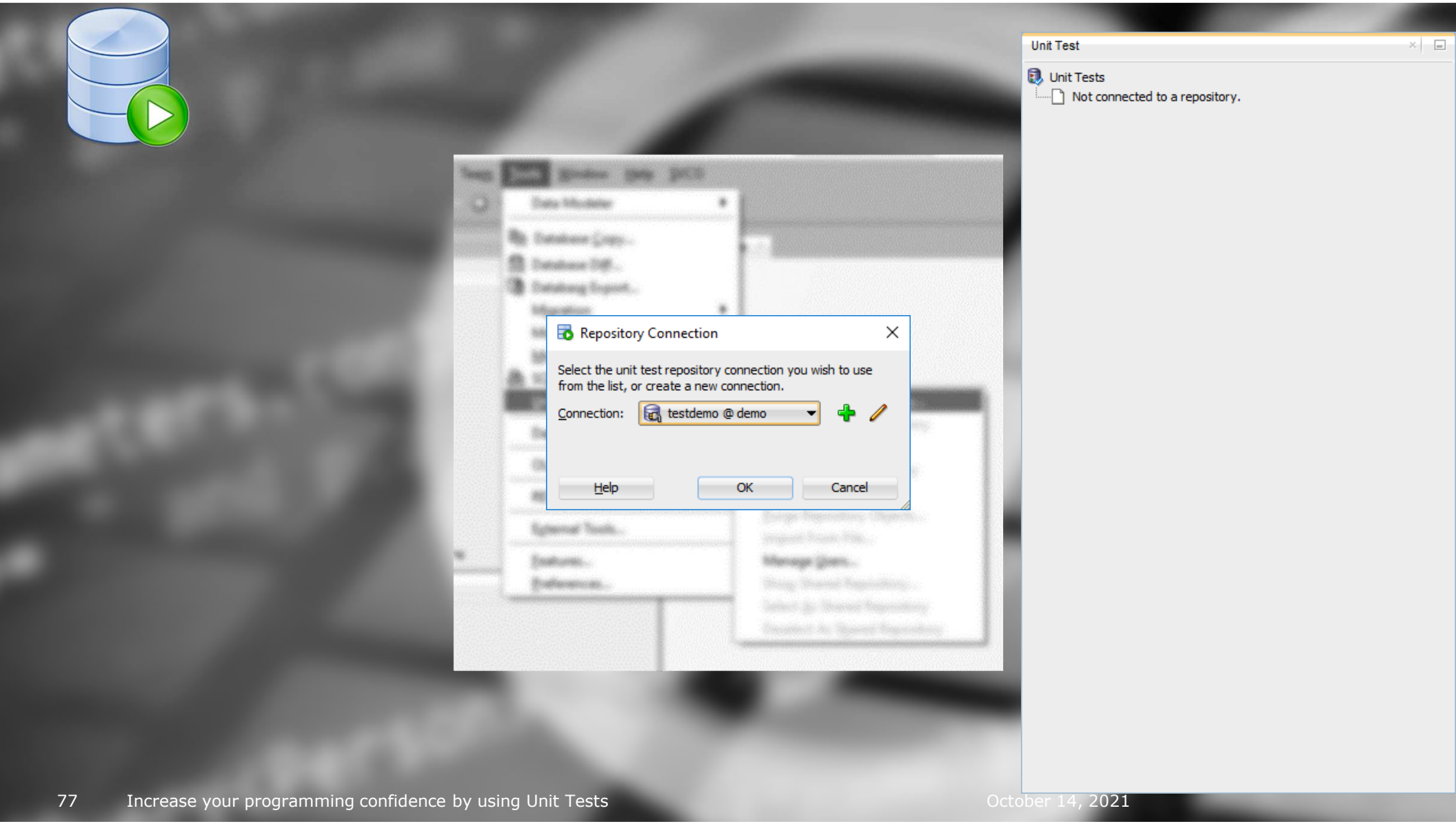

Create Repository
Create Repository (Running)
[00:08]
Run in Background
Cancel Task
Help
No

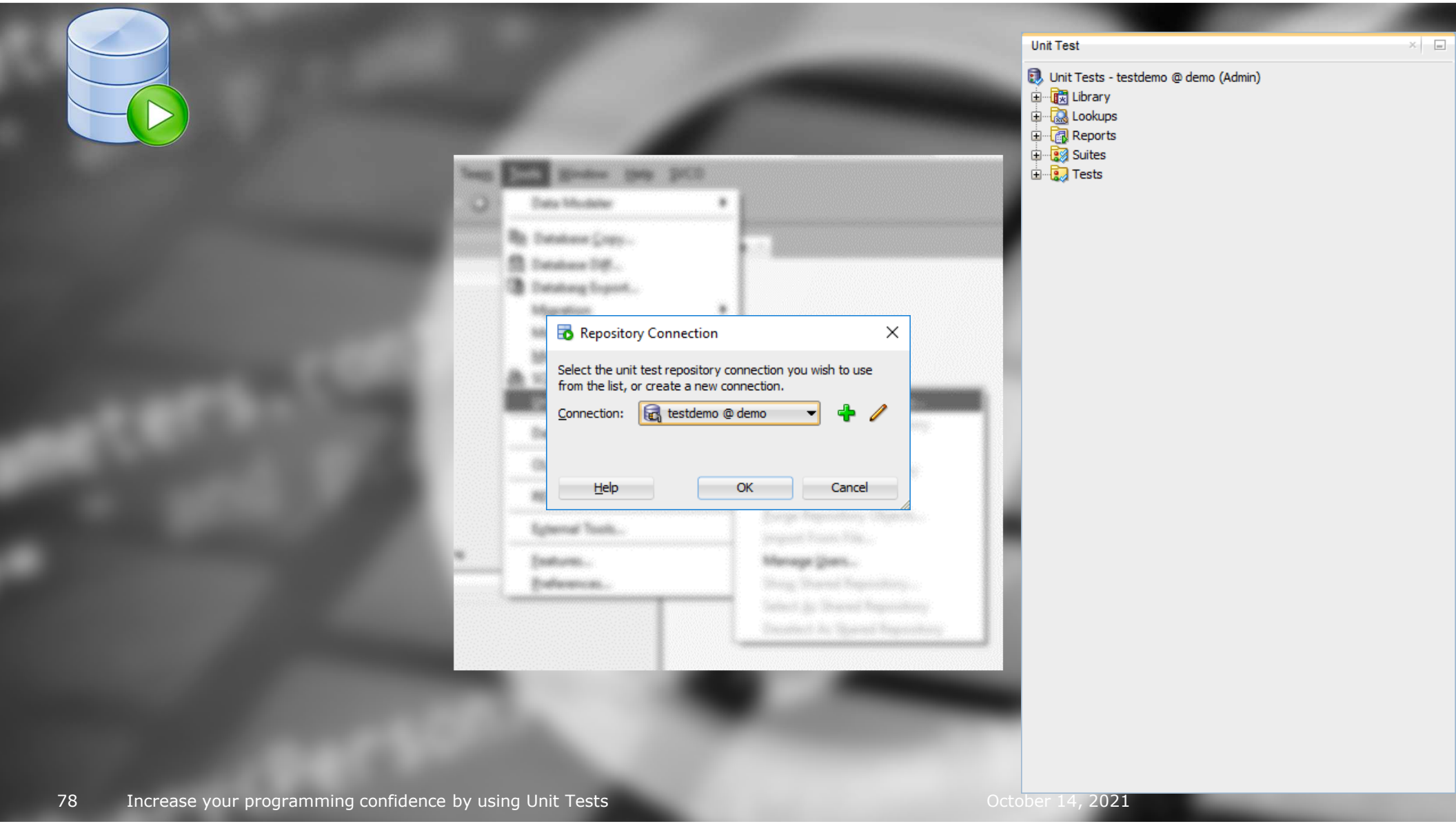

Success
Create Repository Succeeded
OK

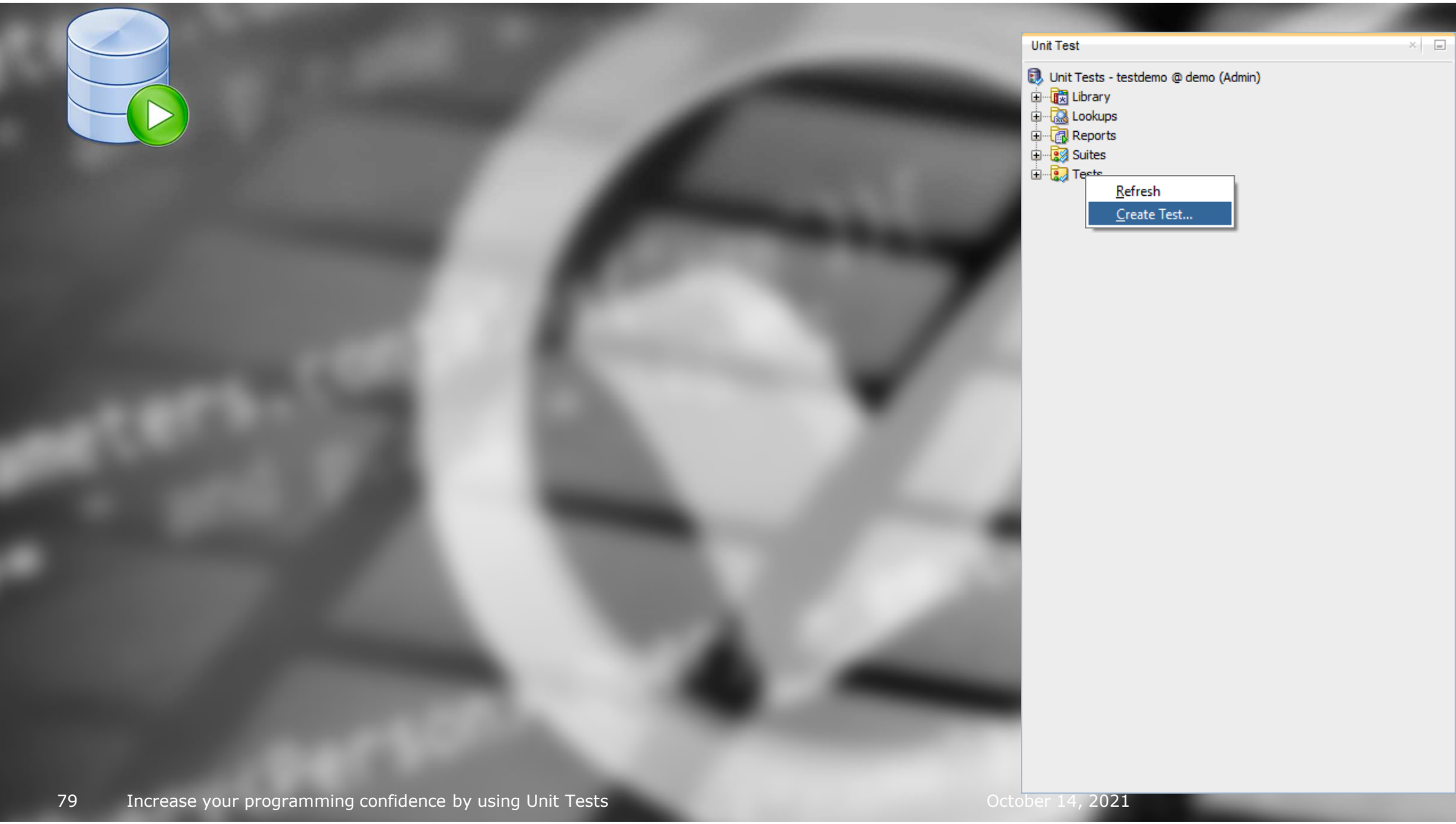

Unit Test
Unit Tests
Not connected to a repository.

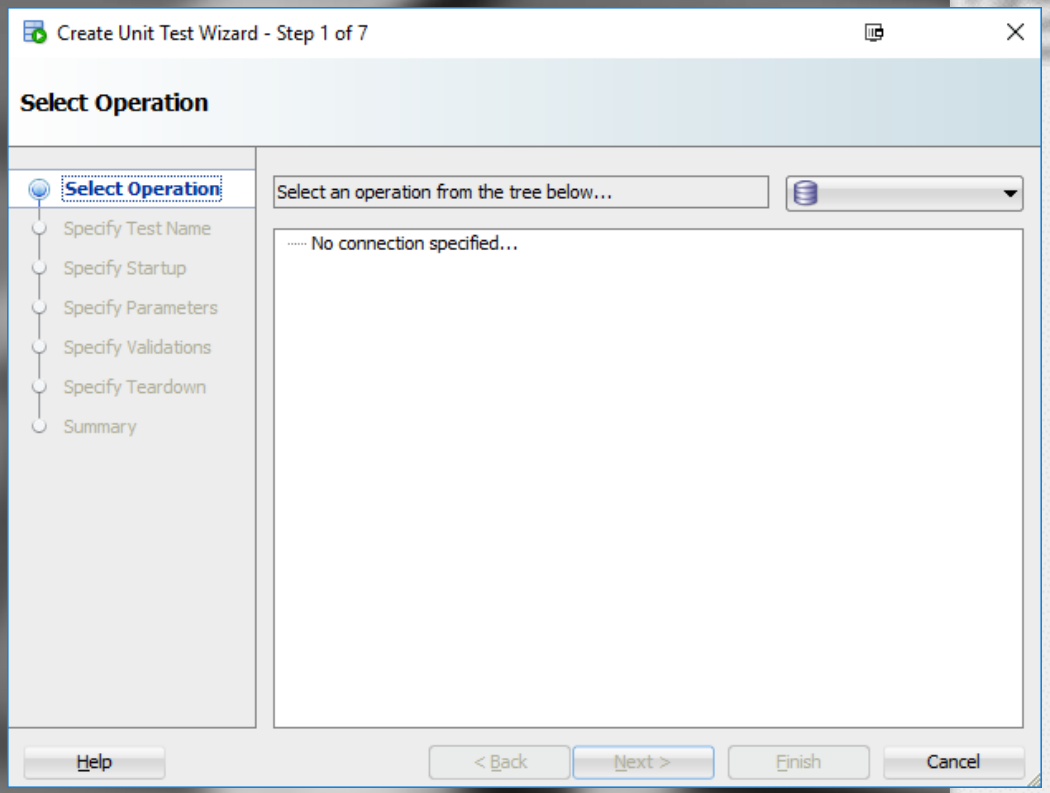

Team
Tools
Window
Help
SVCO
Data Modeler
Database Copy...
Database Diff...
Database Export...
Migration
Monitor SQL...
Monitor Sessions...
SQL Worksheet
Alt-F10
Unit Test
Data Miner
OLAP
REST Data Services
External Tools...
Features...
Preferences...
Select Current Repository...
Deselect Current Repository
Purge Run Results...
Create/Update Repository
Drop Repository...
Purge Repository Objects...
Import From File...
Manage Users...
Show Shared Repository...
Select As Shared Repository
Deselect As Shared Repository

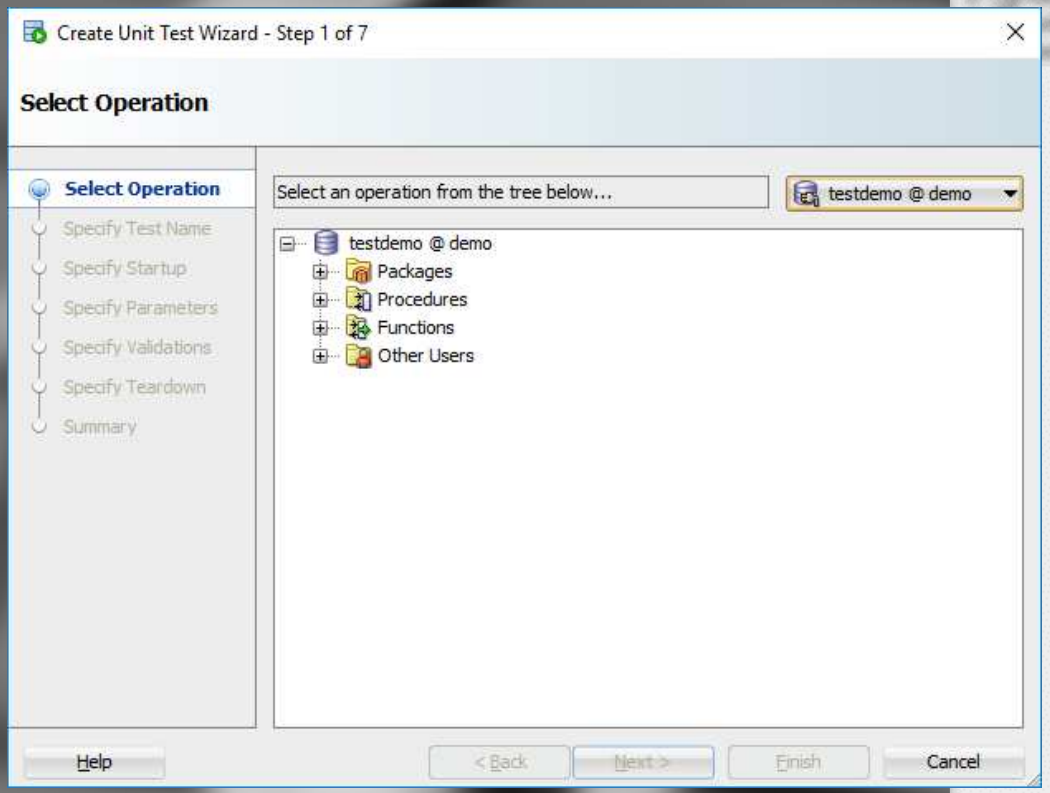

Unit Test
Unit Tests
Not connected to a repository.

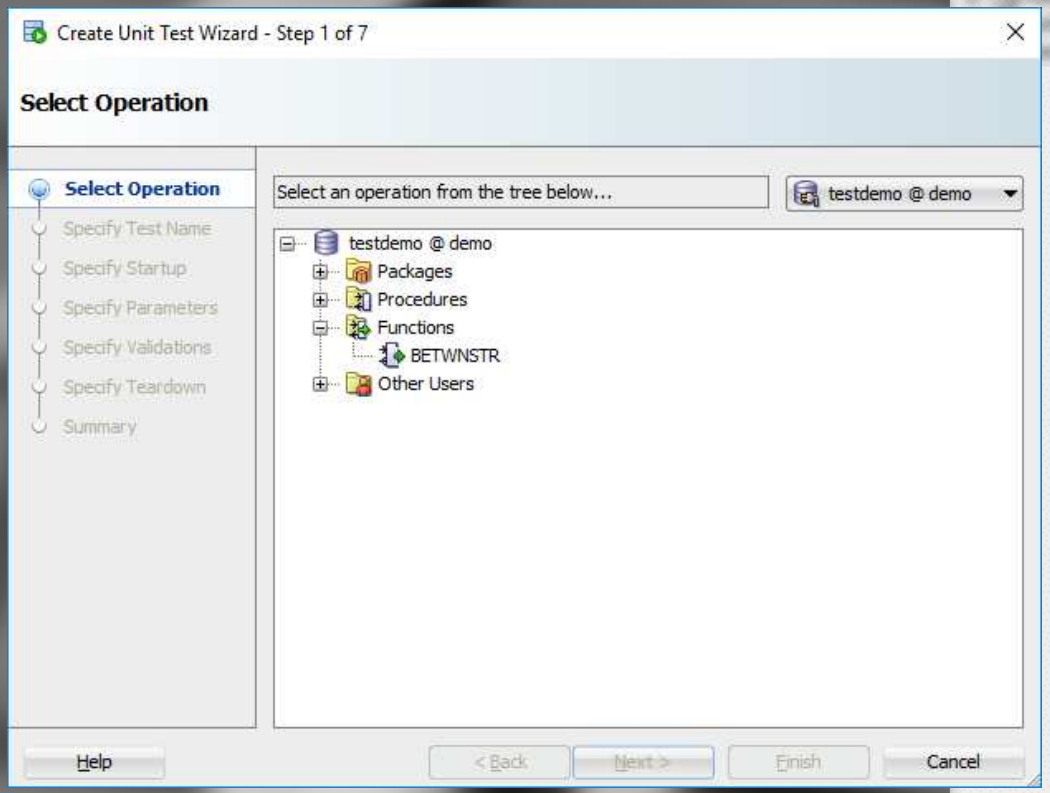

Repository Connection
Select the unit test repository connection you wish to use from the list, or create a new connection.
Connection:
testdemo @ demo
Help
OK
Cancel
Unit Test
Unit Tests
Not connected to a repository.

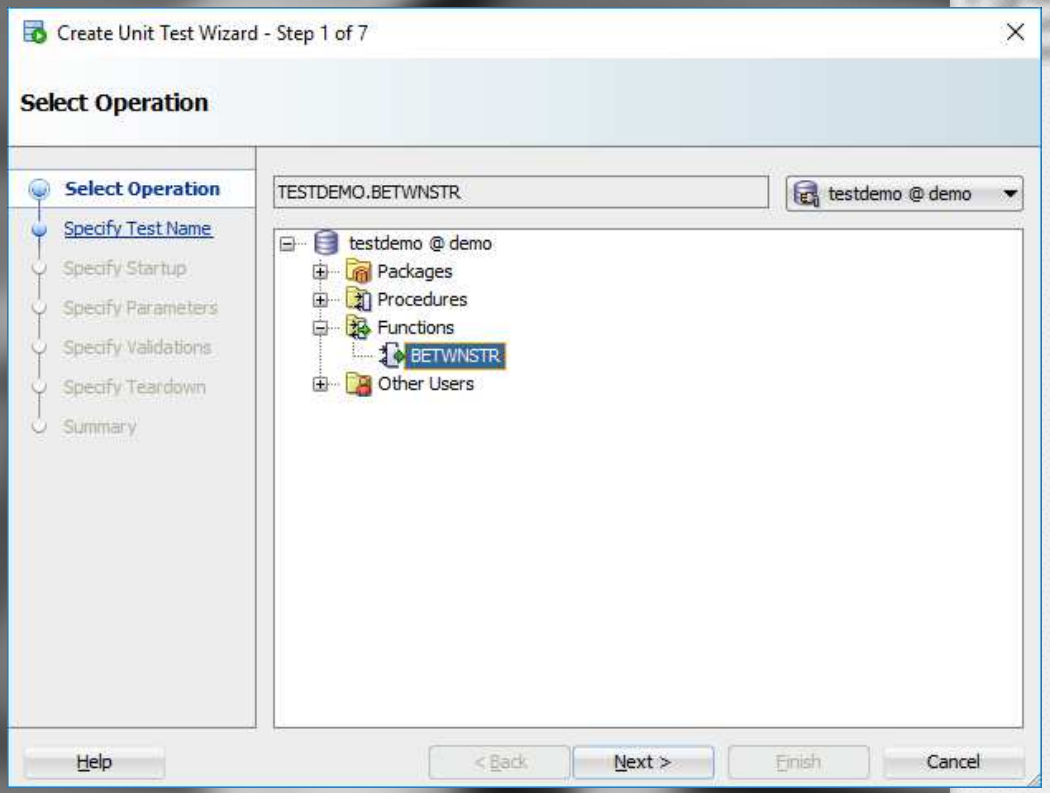

Repository Connection
Select the unit test repository connection you wish to use from the list, or create a new connection.
Connection:
testdemo @ demo
Help
OK
Cancel
Unit Test
Unit Tests - testdemo @ demo (Admin)
Library
Lookups
Reports
Suites
Tests

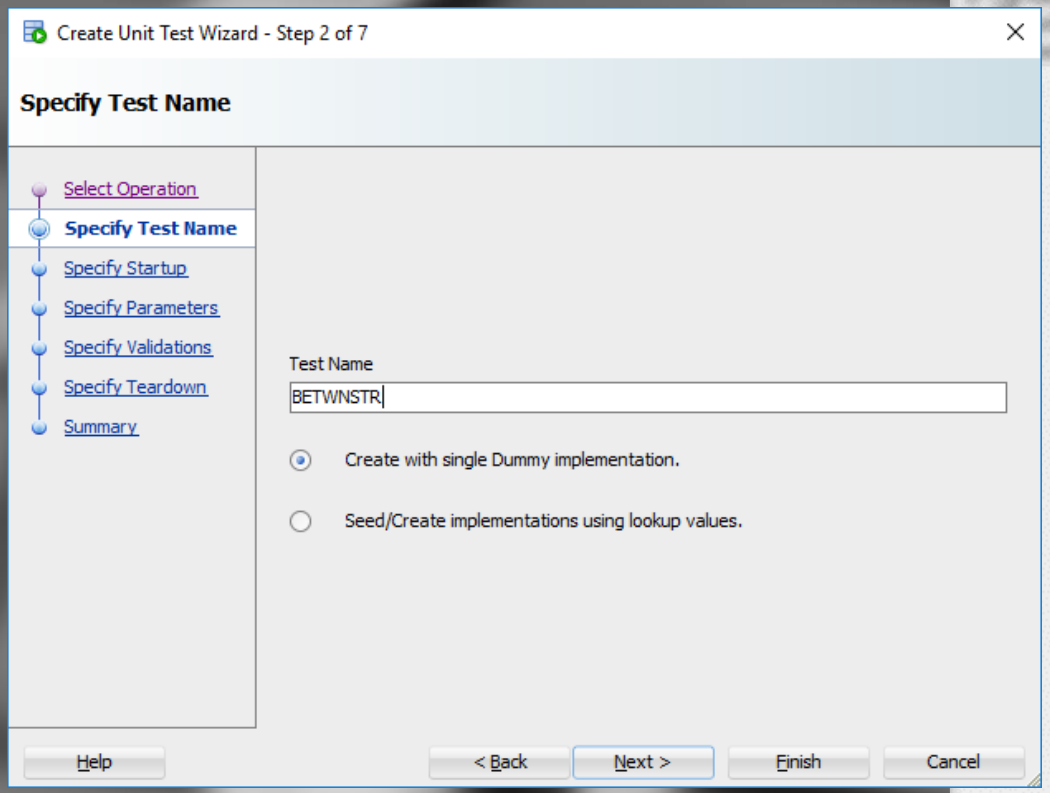

Unit Test
Unit Tests - testdemo @ demo (Admin)
Library
Lookups
Reports
Suites
Tests
Refresh
Create Test...

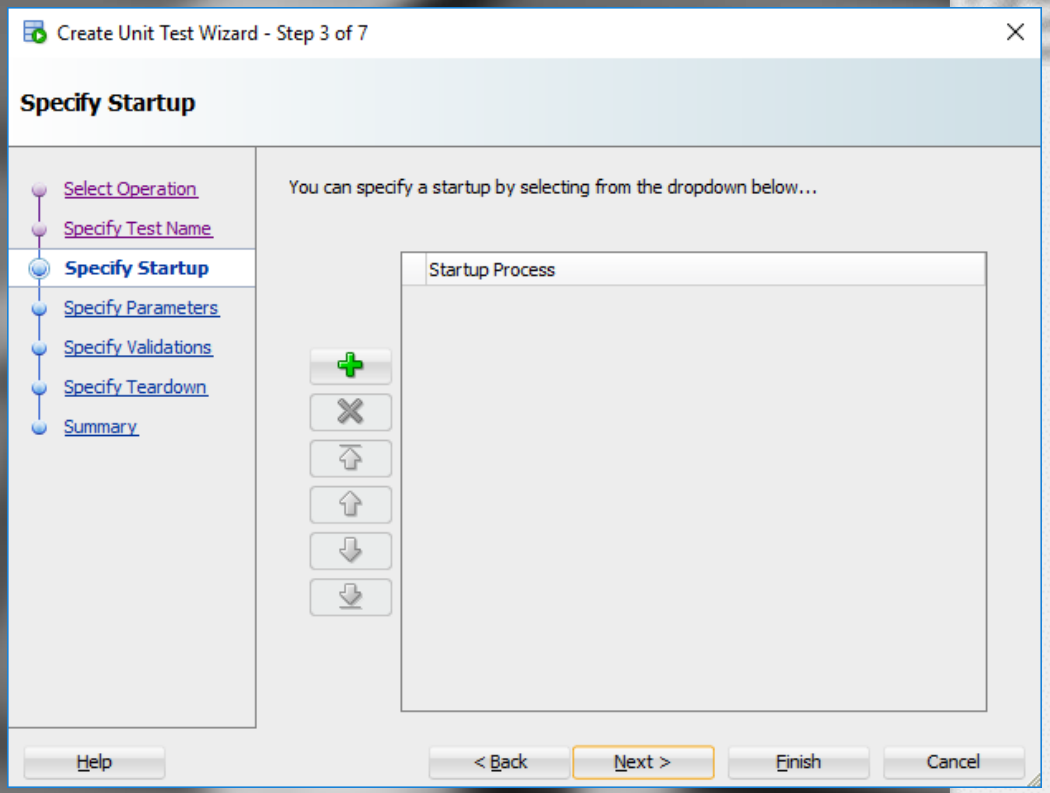

Create Unit Test Wizard - Step 1 of 7
Select Operation
Select Operation
Specify Test Name
Specify Startup
Specify Parameters
Specify Validations
Specify Teardown
Summary
Select an operation from the tree below...
No connection specified...
Help
< Back
Next >
Finish
Cancel

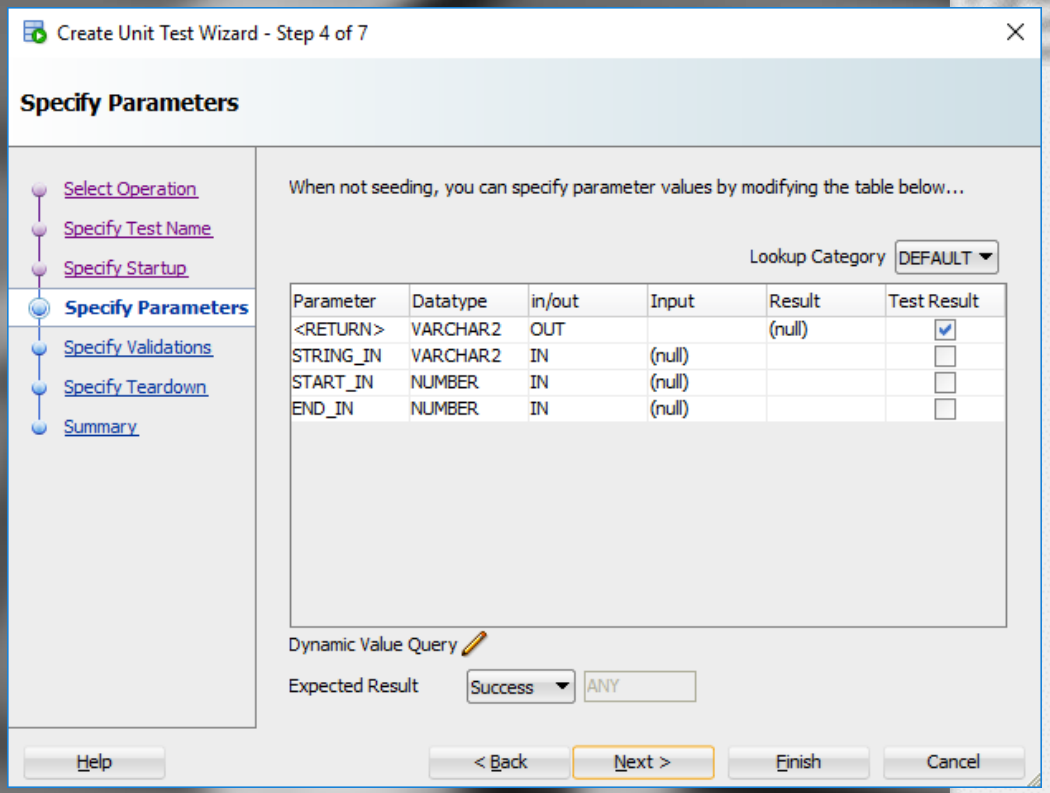

Create Unit Test Wizard - Step 1 of 7
Select Operation
Select Operation
Specify Test Name
Specify Startup
Specify Parameters
Specify Validations
Specify Teardown
Summary
Select an operation from the tree below...
testdemo @ demo
testdemo @ demo
Packages
Procedures
Functions
Other Users
Help
< Back
Next >
Finish
Cancel

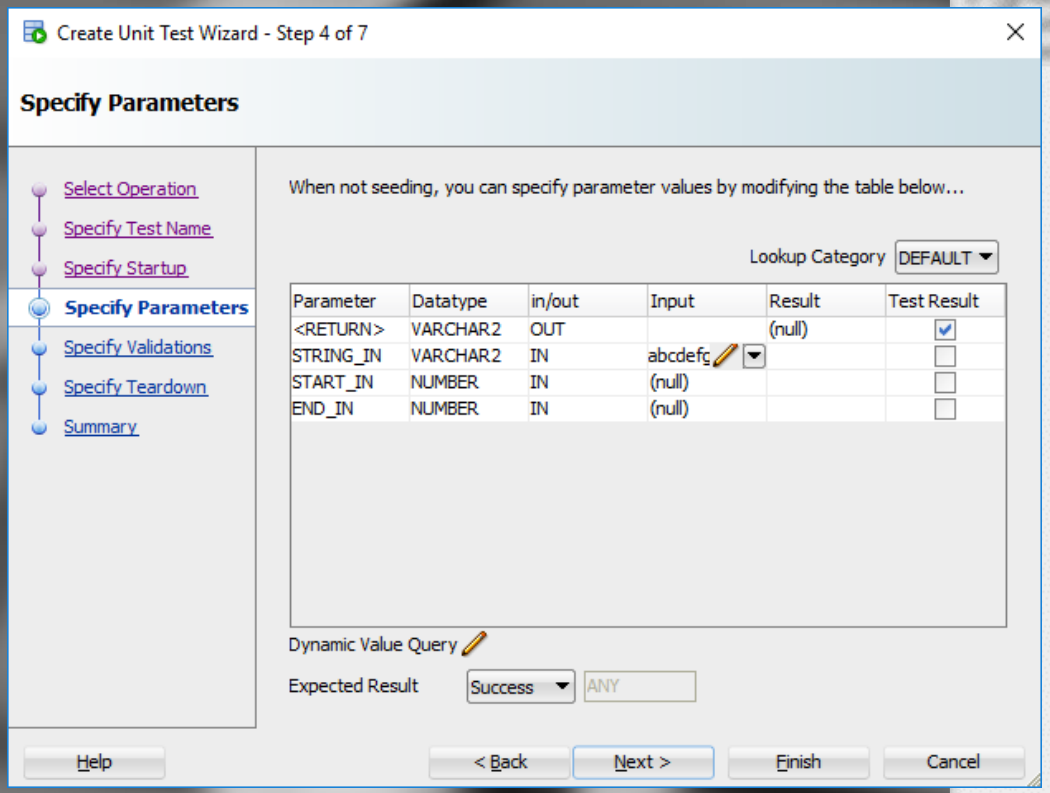

Create Unit Test Wizard - Step 1 of 7
Select Operation
Select Operation
Specify Test Name
Specify Startup
Specify Parameters
Specify Validations
Specify Teardown
Summary
Select an operation from the tree below...
testdemo @ demo
testdemo @ demo
Packages
Procedures
Functions
BETWNSTR
Other Users
Help
< Back
Next >
Finish
Cancel


Create Unit Test Wizard - Step 1 of 7
Select Operation
Select Operation
Specify Test Name
Specify Startup
Specify Parameters
Specify Validations
Specify Teardown
Summary
TESTDEMO.BETWNSTR
testdemo @ demo
testdemo @ demo
Packages
Procedures
Functions
BETWNSTR
Other Users
Help
< Back
Next >
Finish
Cancel


Create Unit Test Wizard - Step 2 of 7
Specify Test Name
Select Operation
Specify Test Name
Specify Startup
Specify Parameters
Specify Validations
Specify Teardown
Summary
Test Name
BETWNSTR
Create with single Dummy implementation.
Seed/Create implementations using lookup values.
Help
< Back
Next >
Finish
Cancel


Create Unit Test Wizard - Step 3 of 7
Specify Startup
Select Operation
Specify Test Name
Specify Startup
Specify Parameters
Specify Validations
Specify Teardown
Summary
You can specify a startup by selecting from the dropdown below...
Startup Process
Help
< Back
Next >
Finish
Cancel


Create Unit Test Wizard - Step 4 of 7
Specify Parameters
Select Operation
Specify Test Name
Specify Startup
Specify Parameters
Specify Validations
Specify Teardown
Summary
When not seeding, you can specify parameter values by modifying the table below...
Lookup Category
DEFAULT
Parameter
Datatype
in/out
Input
Result
Test Result
<RETURN>
VARCHAR2
OUT
(null)
STRING_IN
VARCHAR2
IN
(null)
START_IN
NUMBER
IN
(null)
END_IN
NUMBER
IN
(null)
Dynamic Value Query
Expected Result
Success
ANY
Help
< Back
Next >
Finish
Cancel


Create Unit Test Wizard - Step 4 of 7
Specify Parameters
Select Operation
Specify Test Name
Specify Startup
Specify Parameters
Specify Validations
Specify Teardown
Summary
When not seeding, you can specify parameter values by modifying the table below...
Lookup Category
DEFAULT
Parameter
Datatype
in/out
Input
Result
Test Result
<RETURN>
VARCHAR2
OUT
(null)
STRING_IN
VARCHAR2
IN
abcdefg
START_IN
NUMBER
IN
(null)
END_IN
NUMBER
IN
(null)
Dynamic Value Query
Expected Result
Success
ANY
Help
< Back
Next >
Finish
Cancel

Create Unit Test Wizard - Step 4 of 7

Specify Parameters

- Select Operation
- Specify Test Name
- Specify Startup
- **Specify Parameters**
- Specify Validations
- Specify Teardown
- Summary

When not seeding, you can specify parameter values by modifying the table below...

Lookup Category: DEFAULT

Parameter	Datatype	in/out	Input	Result	Test Result
<RETURN>	VARCHAR2	OUT		(null)	☑
STRING_IN	VARCHAR2	IN	abcdefgh		☐
START_IN	NUMBER	IN	3		☐
END_IN	NUMBER	IN	(null)		☐

Dynamic Value Query

Expected Result: Success | ANY

Help | < Back | Next > | Finish | Cancel

Create Unit Test Wizard - Step 4 of 7

Specify Parameters

- Select Operation
- Specify Test Name
- Specify Startup
- **Specify Parameters**
- Specify Validations
- Specify Teardown
- Summary

When not seeding, you can specify parameter values by modifying the table below...

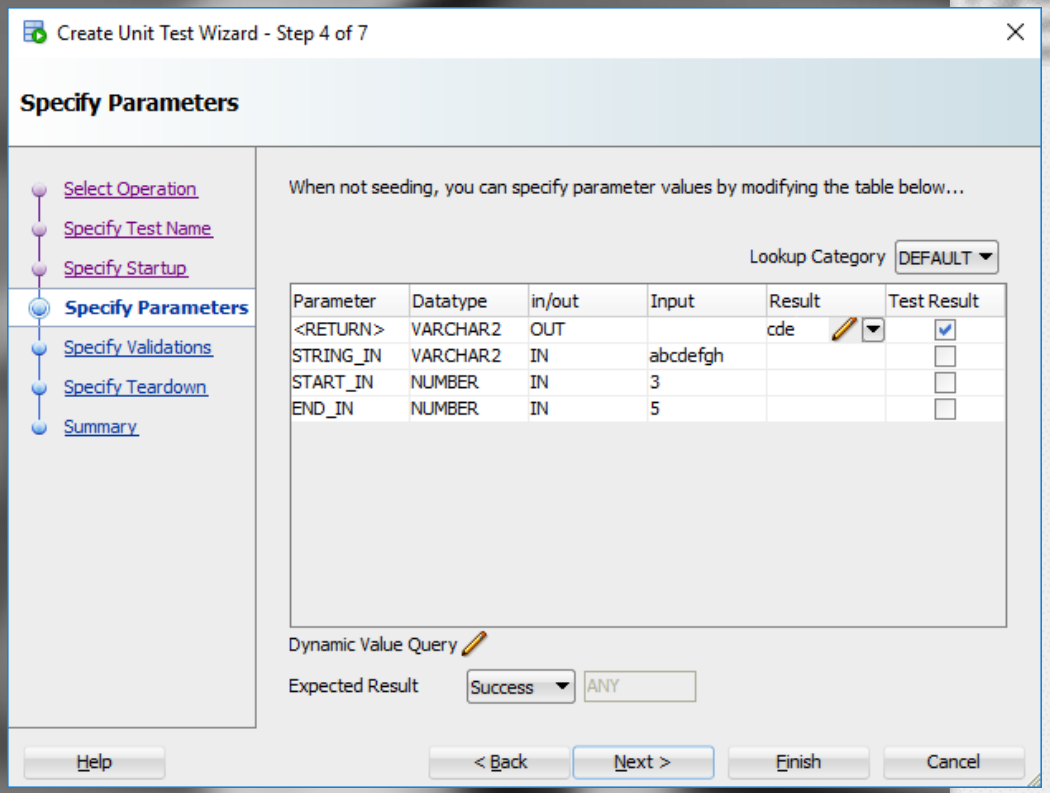
Lookup Category: DEFAULT

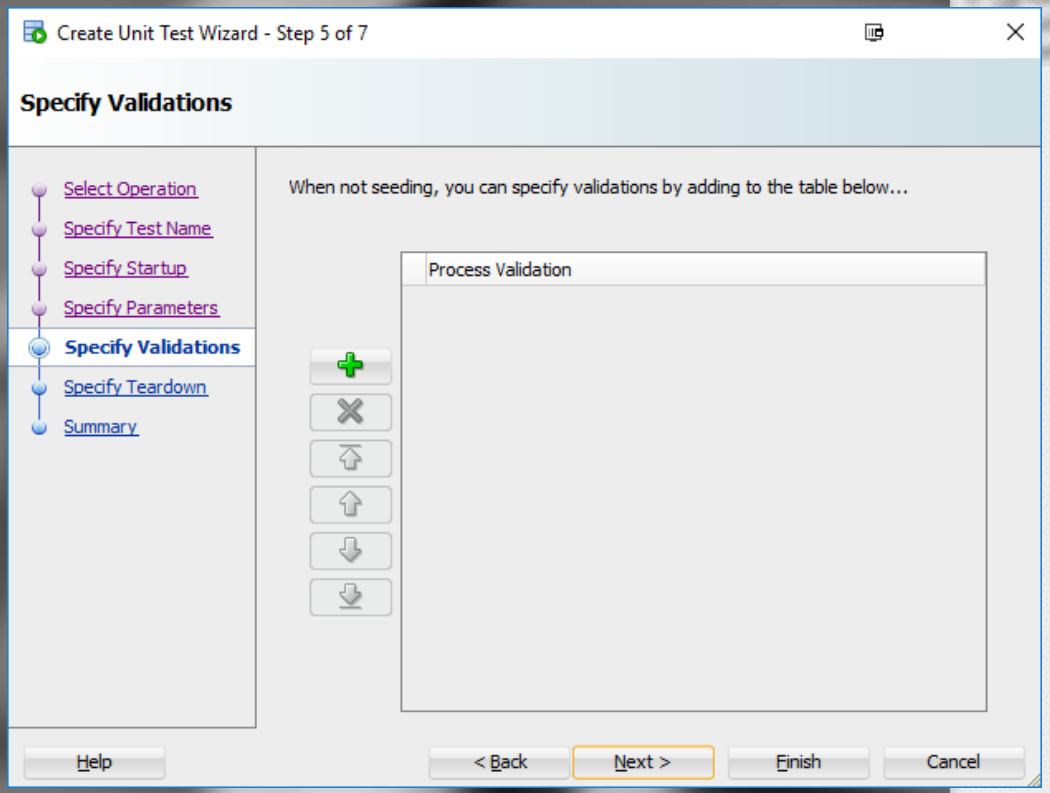
Parameter	Datatype	in/out	Input	Result	Test Result
<RETURN>	VARCHAR2	OUT		(null)	☑
STRING_IN	VARCHAR2	IN	abcdefgh		☐
START_IN	NUMBER	IN	3		☐
END_IN	NUMBER	IN	5		☐

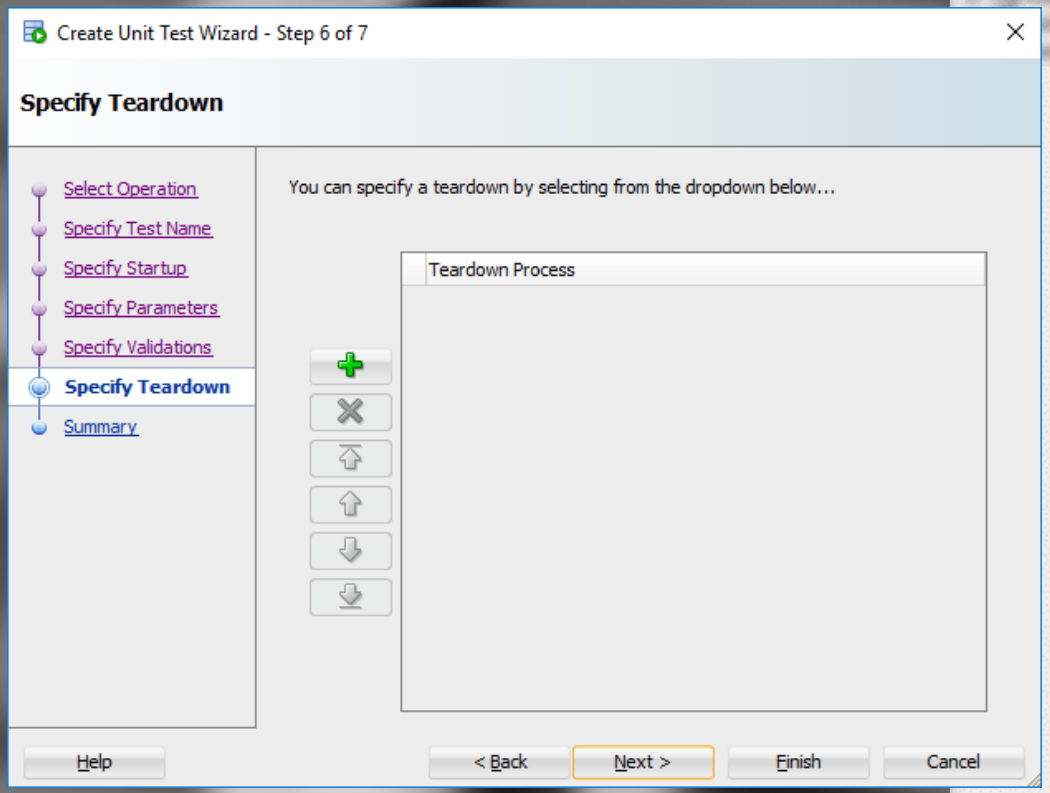
Dynamic Value Query

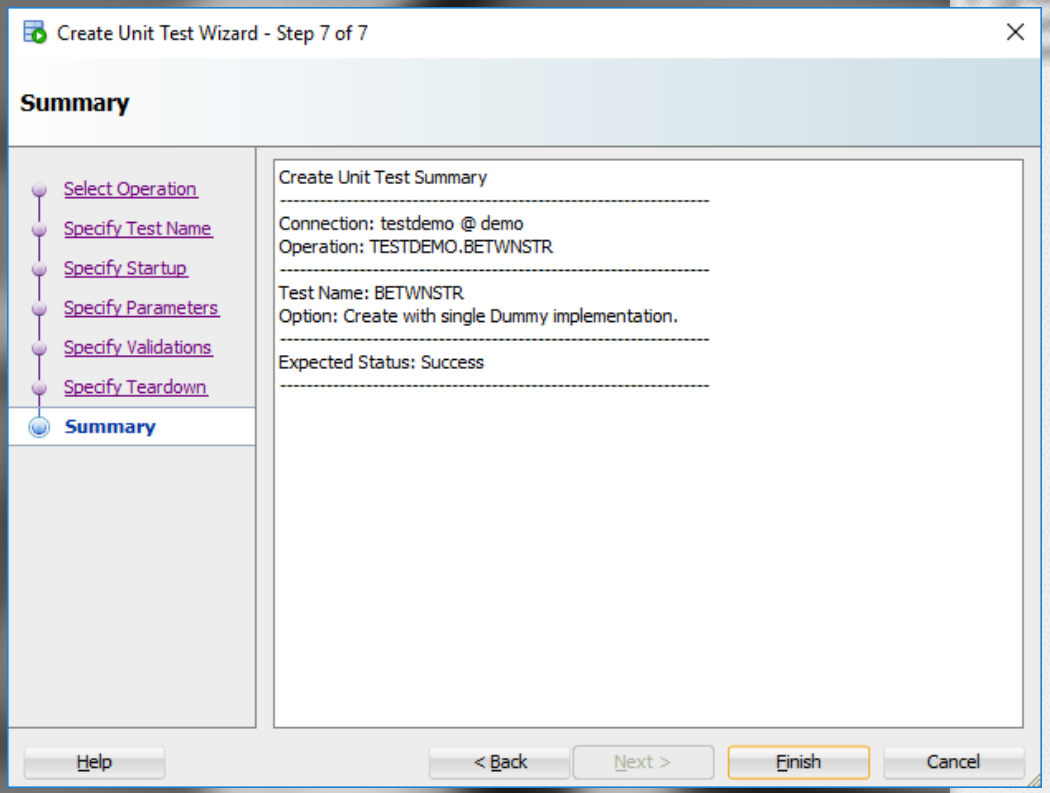
Expected Result: Success | ANY

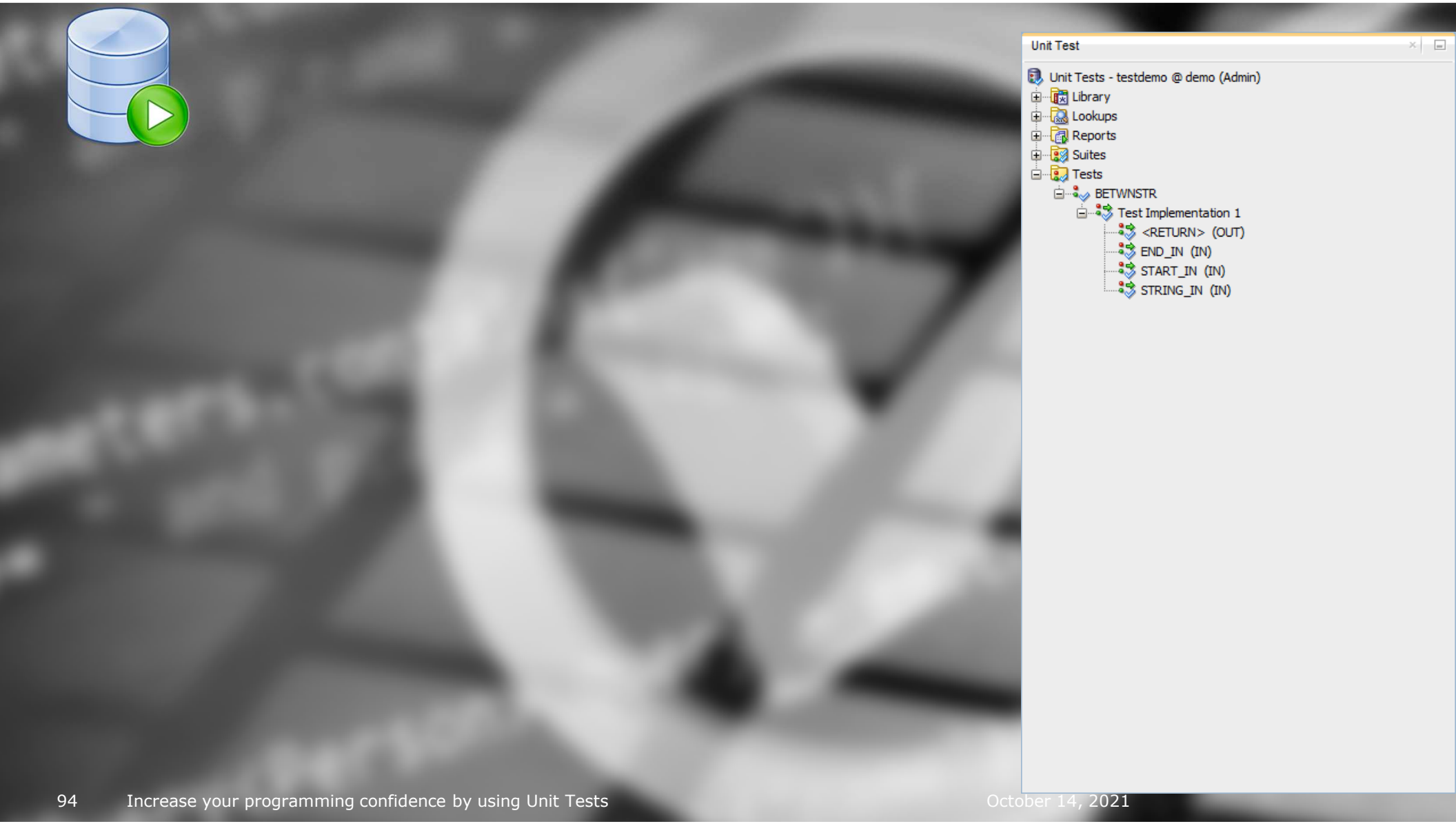
Help | < Back | Next > | Finish | Cancel


Create Unit Test Wizard - Step 4 of 7
Specify Parameters
Select Operation
Specify Test Name
Specify Startup
Specify Parameters
Specify Validations
Specify Teardown
Summary
When not seeding, you can specify parameter values by modifying the table below...
Lookup Category
DEFAULT
Parameter
Datatype
in/out
Input
Result
Test Result
<RETURN>
VARCHAR2
OUT
cde
STRING_IN
VARCHAR2
IN
abcdefgh
START_IN
NUMBER
IN
3
END_IN
NUMBER
IN
5
Dynamic Value Query
Expected Result
Success
ANY
Help
< Back
Next >
Finish
Cancel


Create Unit Test Wizard - Step 5 of 7
Specify Validations
Select Operation
Specify Test Name
Specify Startup
Specify Parameters
Specify Validations
Specify Teardown
Summary
When not seeding, you can specify validations by adding to the table below...
Process Validation
Help
< Back
Next >
Finish
Cancel


Create Unit Test Wizard - Step 6 of 7
Specify Teardown
Select Operation
Specify Test Name
Specify Startup
Specify Parameters
Specify Validations
Specify Teardown
Summary
You can specify a teardown by selecting from the dropdown below...
Teardown Process
Help
< Back
Next >
Finish
Cancel


Create Unit Test Wizard - Step 7 of 7
Summary
Select Operation
Specify Test Name
Specify Startup
Specify Parameters
Specify Validations
Specify Teardown
Summary
Create Unit Test Summary
Connection: testdemo @ demo
Operation: TESTDEMO.BETWNSTR
Test Name: BETWNSTR
Option: Create with single Dummy implementation.
Expected Status: Success
Help
< Back
Next >
Finish
Cancel


Unit Test
Unit Tests - testdemo @ demo (Admin)
Library
Lookups
Reports
Suites
Tests
BETWNSTR
Test Implementation 1
<RETURN> (OUT)
END_IN (IN)
START_IN (IN)
STRING_IN (IN)

Start Page | BETWNSTR

Details | Results

testdemo @ demo

FUNCTION TESTDEMO.BETWNSTR(STRING_IN IN VARCHAR2,START_IN IN NUMBER,END_IN IN NUMBER) RETURNS VARCHAR2

Startup Process

Teardown Process

Gather Code Coverage Statistics

Test Implementation 1

Lookup Category DEFAULT

Parameter	Datatype	in/out	Input	Result	Test Result
<RETURN>	VARCHAR2	OUT		cde	☑
STRING_IN	VARCHAR2	IN	abcdefgh		☐
START_IN	NUMBER	IN	3		☐
END_IN	NUMBER	IN	5		☐

Dynamic Value Query

Expected Result Success ANY Enter expected error number or "ANY".

Unit Test

- Unit Tests - testdemo @ demo (Admin)
  - Library
  - Lookups
  - Reports
  - Suites
  - Tests
    - BETWNSTR
      - Test Implementation 1
        - <RETURN> (OUT)
        - END_IN (IN)
        - START_IN (IN)
        - STRING_IN (IN)

Start Page | BETWNSTR

Details | Results

testdemo @ demo

FUNCTION TESTDEMO.BETWNSTR(STRING_IN IN VARCHAR2,START_IN IN NUMBER,END_IN IN NUMBER) RETURNS VARCHAR2

Startup Process

Teardown Process

Gather Code Coverage Statistics

Test Implementation 1

Lookup Category DEFAULT

Parameter	Datatype	in/out	Input	Result	Test Result
<RETURN>	VARCHAR2	OUT		cde	☑
STRING_IN	VARCHAR2	IN	abcdefgh		☐
START_IN	NUMBER	IN	3		☐
END_IN	NUMBER	IN	5		☐

Dynamic Value Query

Expected Result Success ANY Enter expected error number or "ANY".

Unit Test

Unit Tests - testdemo @ demo (Admin)

- Library
- Lookups
- Reports
- Suites
- Tests
  - BETWNSTR
    - Te

Open
Delete Test...
Rename Test...
Copy Test...
Add Implementation...
Run Test
Purge Test Results...
Synchronize Test...
Export To File...

Start Page | BETWNSTR

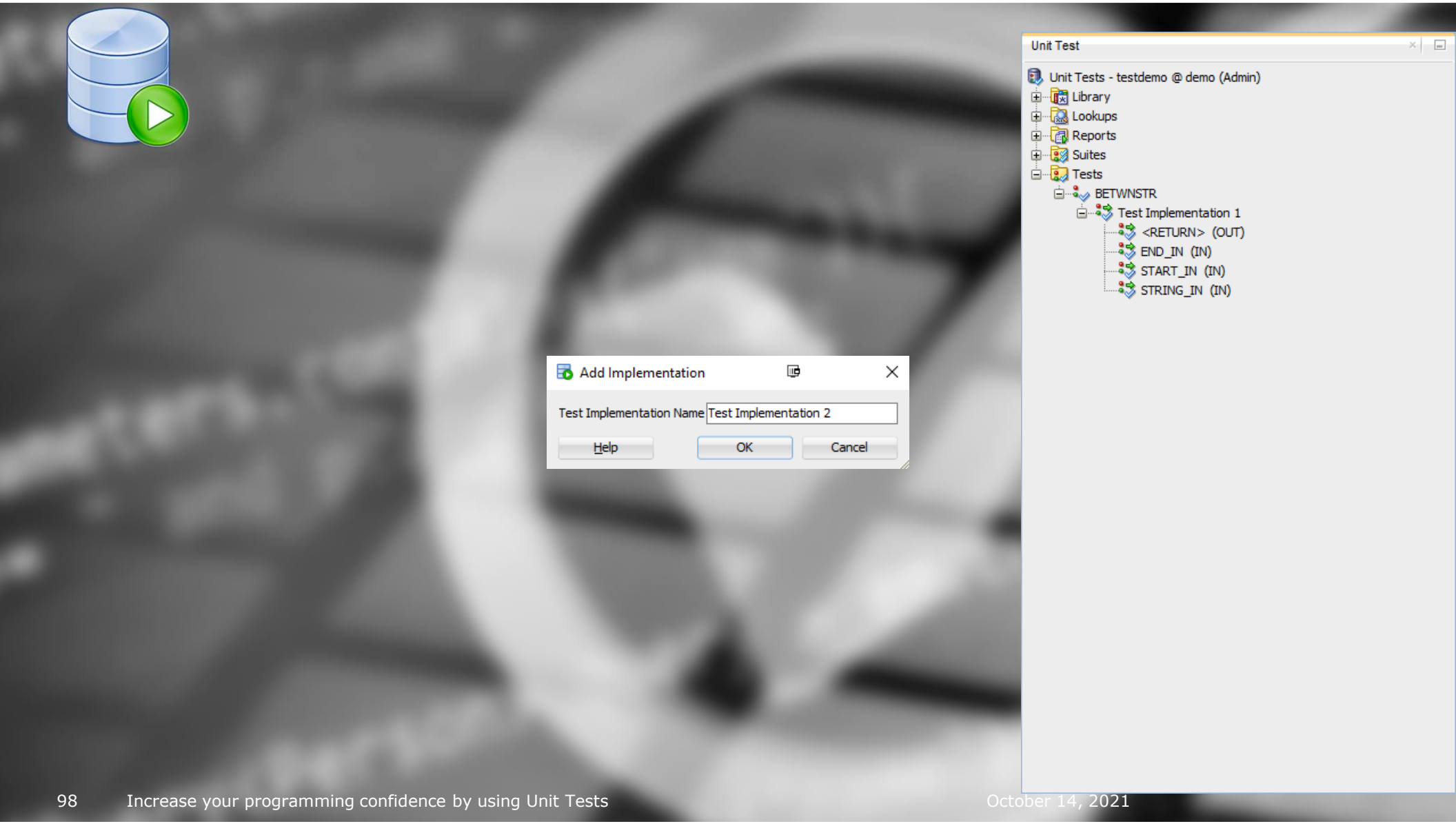
Details | Results

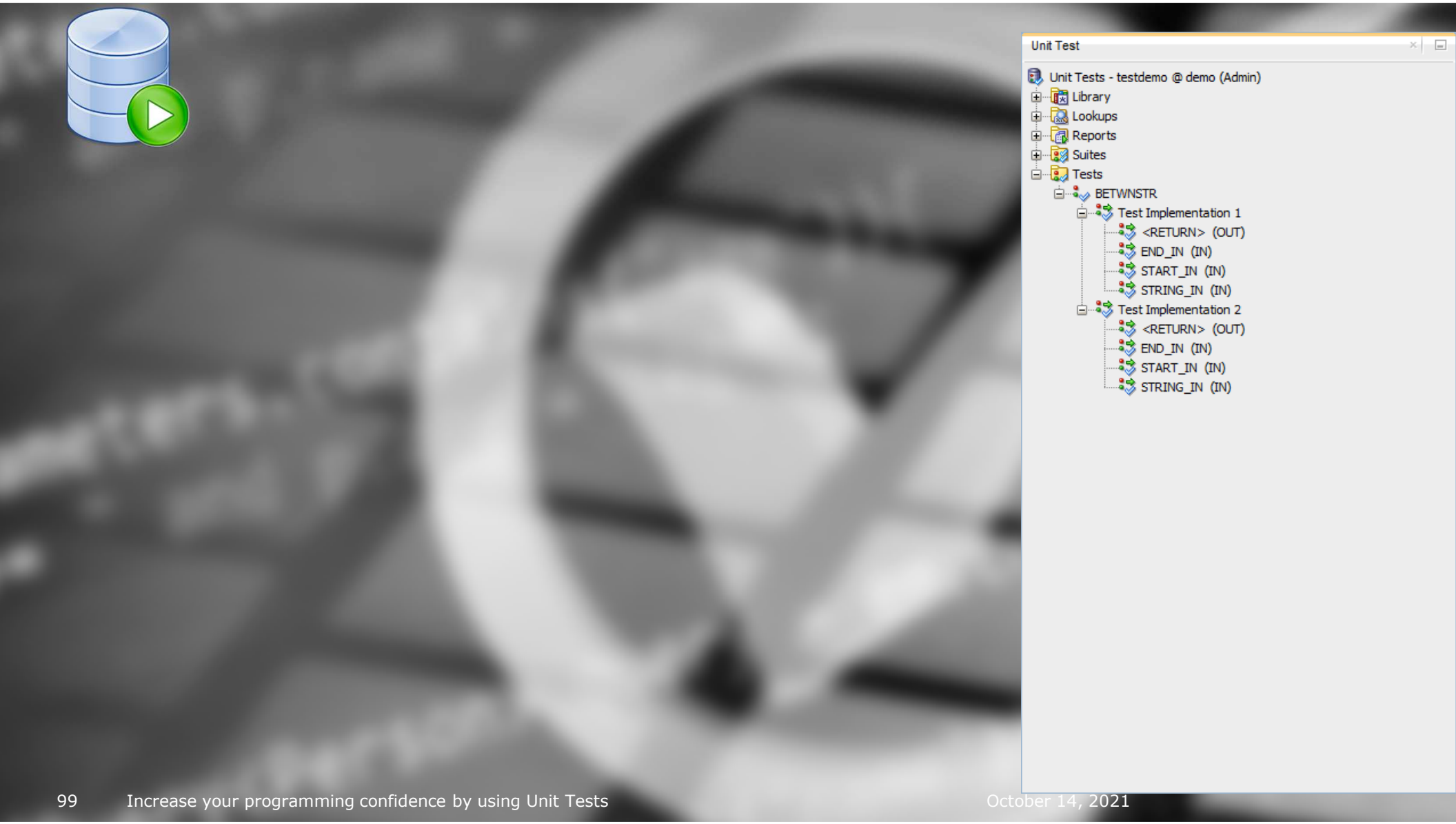
testdemo @ demo

Test Run	Status	Duration	Message
BETWNSTR			
TESTDEMO: Run On - 2017-01-08 07:55:48.544753	SUCCESS	17	
Implementation - Test Implementation 1	SUCCESS	17	
Operation Call	SUCCESS	17	
<RETURN>	SUCCESS		Expected: [cde], Received: [cde]
IN Parameter #1 - STRING_IN			Value: [abcdefgh]
IN Parameter #2 - START_IN			Value: [3]
IN Parameter #3 - END_IN			Value: [5]

Unit Test

- Unit Tests - testdemo @ demo (Admin)
  - Library
  - Lookups
  - Reports
  - Suites
  - Tests
    - BETWNSTR
      - Test Implementation 1
        - <RETURN> (OUT)
        - END_IN (IN)
        - START_IN (IN)
        - STRING_IN (IN)


Add Implementation
Test Implementation Name
Test Implementation 2
Help
OK
Cancel
Unit Test
Unit Tests - testdemo @ demo (Admin)
Library
Lookups
Reports
Suites
Tests
BETWNSTR
Test Implementation 1
<RETURN> (OUT)
END_IN (IN)
START_IN (IN)
STRING_IN (IN)


Unit Test
Unit Tests - testdemo @ demo (Admin)
Library
Lookups
Reports
Suites
Tests
BETWNSTR
Test Implementation 1
<RETURN> (OUT)
END_IN (IN)
START_IN (IN)
STRING_IN (IN)
Test Implementation 2
<RETURN> (OUT)
END_IN (IN)
START_IN (IN)
STRING_IN (IN)

Test Implementation 2

Parameter	Datatype	in/out	Input	Result	Test Result
<RETURN>	VARCHAR2	OUT		(null)	
STRING_IN	VARCHAR2	IN	(null)		
START_IN	NUMBER	IN	(null)		
END_IN	NUMBER	IN	(null)		

Dynamic Value Query

Expected Result Success ANY Enter expected error number or "ANY".

Unit Test

- Unit Tests - testdemo @ demo (Admin)
  - Library
  - Lookups
  - Reports
  - Suites
  - Tests
    - BETWNSTR
      - Test Implementation 1
        - <RETURN> (OUT)
        - END_IN (IN)
        - START_IN (IN)
        - STRING_IN (IN)
      - Test Implementation 2
        - <RETURN> (OUT)
        - END_IN (IN)
        - START_IN (IN)
        - STRING_IN (IN)

Test Implementation 2

Parameter	Datatype	in/out	Input	Result	Test Result
<RETURN>	VARCHAR2	OUT		(null)	
STRING_IN	VARCHAR2	IN	abcdefgh		
START_IN	NUMBER	IN	(null)		
END_IN	NUMBER	IN	(null)		

Dynamic Value Query

Expected Result Success ANY Enter expected error number or "ANY".

Unit Test

- Unit Tests - testdemo @ demo (Admin)
  - Library
  - Lookups
  - Reports
  - Suites
  - Tests
    - BETWNSTR
      - Test Implementation 1
        - <RETURN> (OUT)
        - END_IN (IN)
        - START_IN (IN)
        - STRING_IN (IN)
      - Test Implementation 2
        - <RETURN> (OUT)
        - END_IN (IN)
        - START_IN (IN)
        - STRING_IN (IN)

Test Implementation 2

Parameter	Datatype	in/out	Input	Result	Test Result
<RETURN>	VARCHAR2	OUT		(null)	
STRING_IN	VARCHAR2	IN	abcdefgh		
START_IN	NUMBER	IN	0		
END_IN	NUMBER	IN	(null)		

Dynamic Value Query

Expected Result Success ANY Enter expected error number or "ANY".

Unit Test

- Unit Tests - testdemo @ demo (Admin)
  - Library
  - Lookups
  - Reports
  - Suites
  - Tests
    - BETWNSTR
      - Test Implementation 1
        - <RETURN> (OUT)
        - END_IN (IN)
        - START_IN (IN)
        - STRING_IN (IN)
      - Test Implementation 2
        - <RETURN> (OUT)
        - END_IN (IN)
        - START_IN (IN)
        - STRING_IN (IN)

Test Implementation 2

Parameter	Datatype	in/out	Input	Result	Test Result
<RETURN>	VARCHAR2	OUT		(null)	
STRING_IN	VARCHAR2	IN	abcdefgh		
START_IN	NUMBER	IN	0		
END_IN	NUMBER	IN	2		

Dynamic Value Query

Expected Result: Success | ANY | Enter expected error number or "ANY".

Unit Test

- Unit Tests - testdemo @ demo (Admin)
  - Library
  - Lookups
  - Reports
  - Suites
  - Tests
    - BETWNSTR
      - Test Implementation 1
        - <RETURN> (OUT)
        - END_IN (IN)
        - START_IN (IN)
        - STRING_IN (IN)
      - Test Implementation 2
        - <RETURN> (OUT)
        - END_IN (IN)
        - START_IN (IN)
        - STRING_IN (IN)

Test Implementation 2

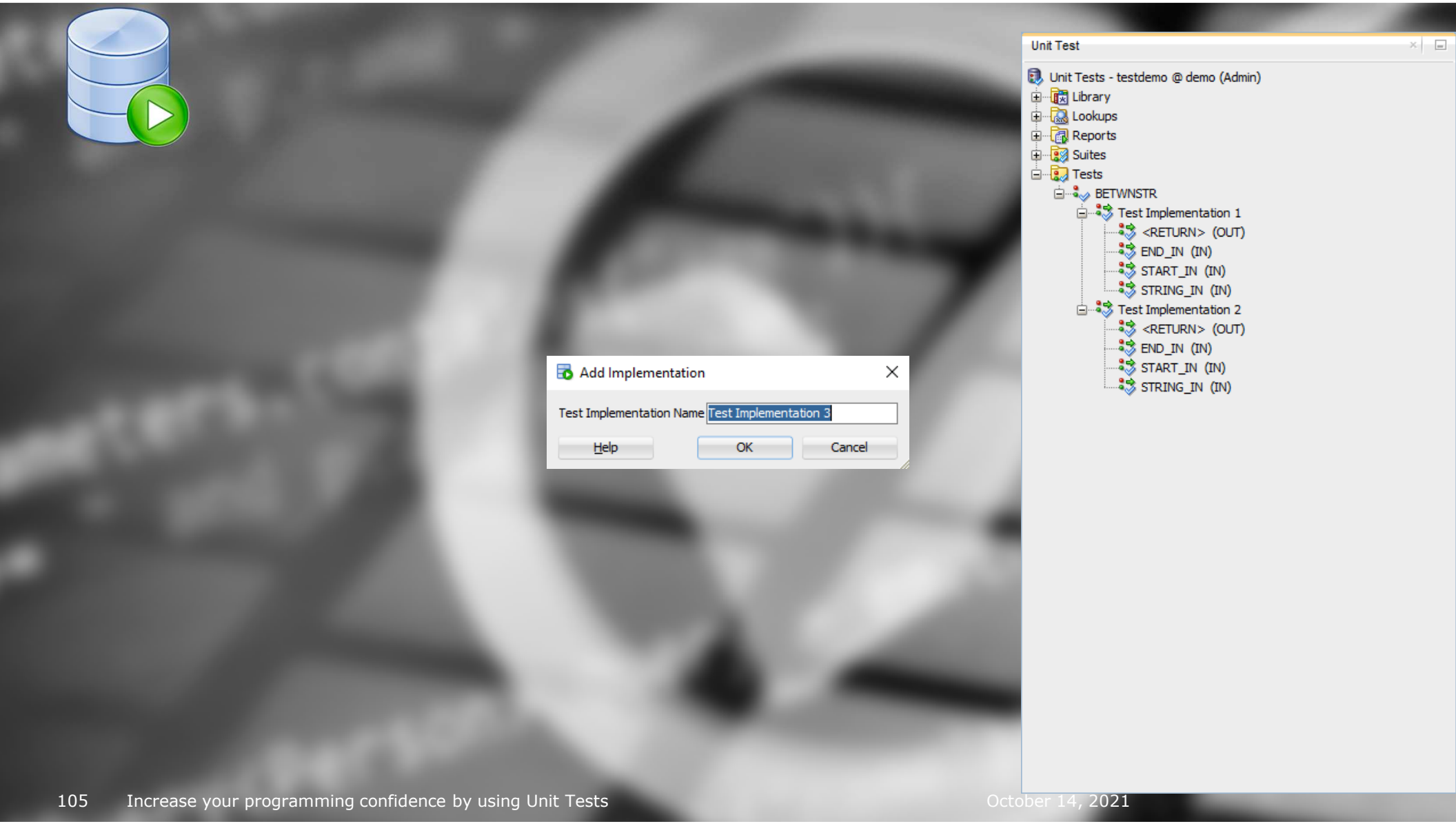
Parameter	Datatype	in/out	Input	Result	Test Result
<RETURN>	VARCHAR2	OUT		ab	
STRING_IN	VARCHAR2	IN	abcdefgh		
START_IN	NUMBER	IN	0		
END_IN	NUMBER	IN	2		

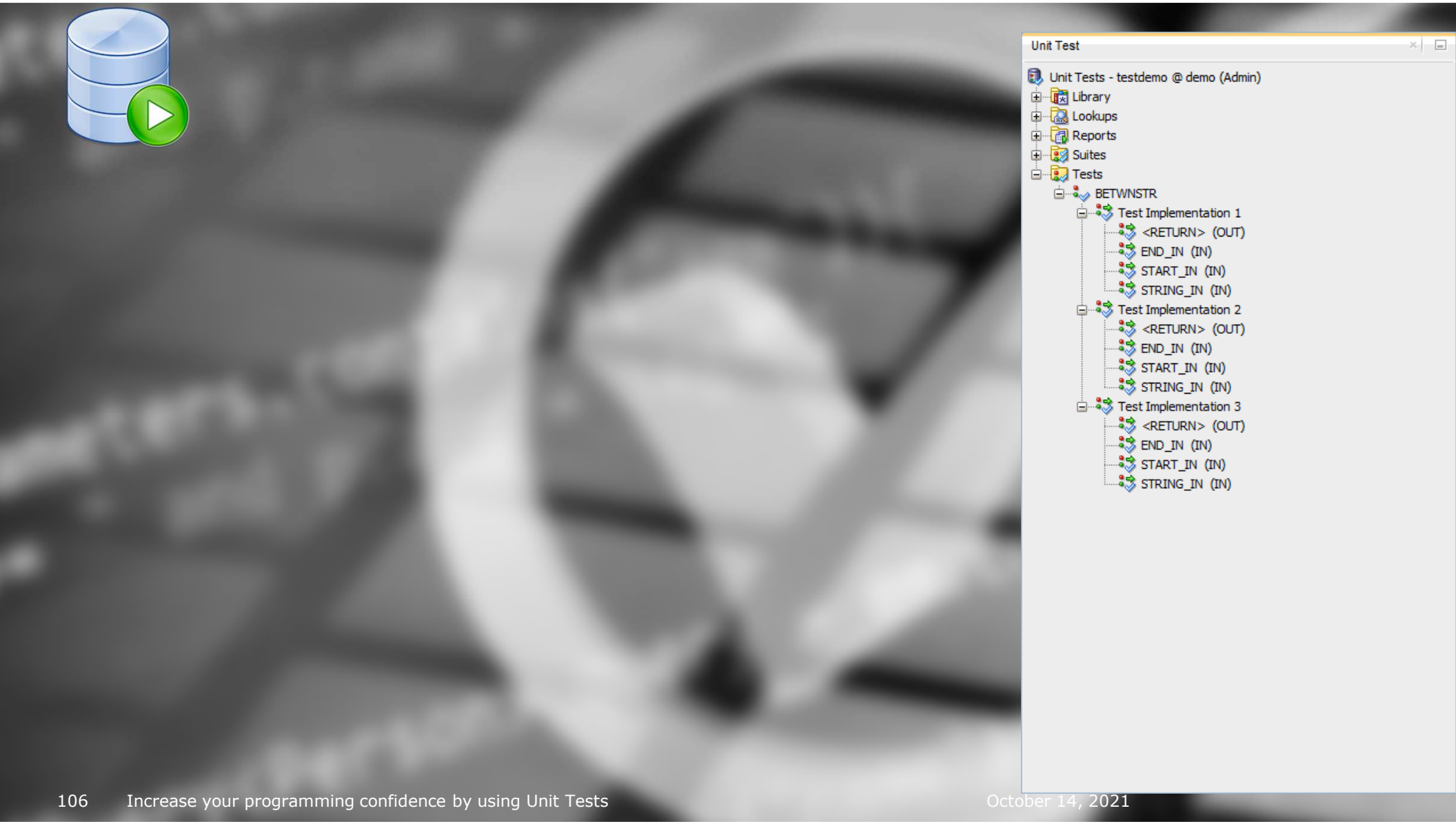
Dynamic Value Query

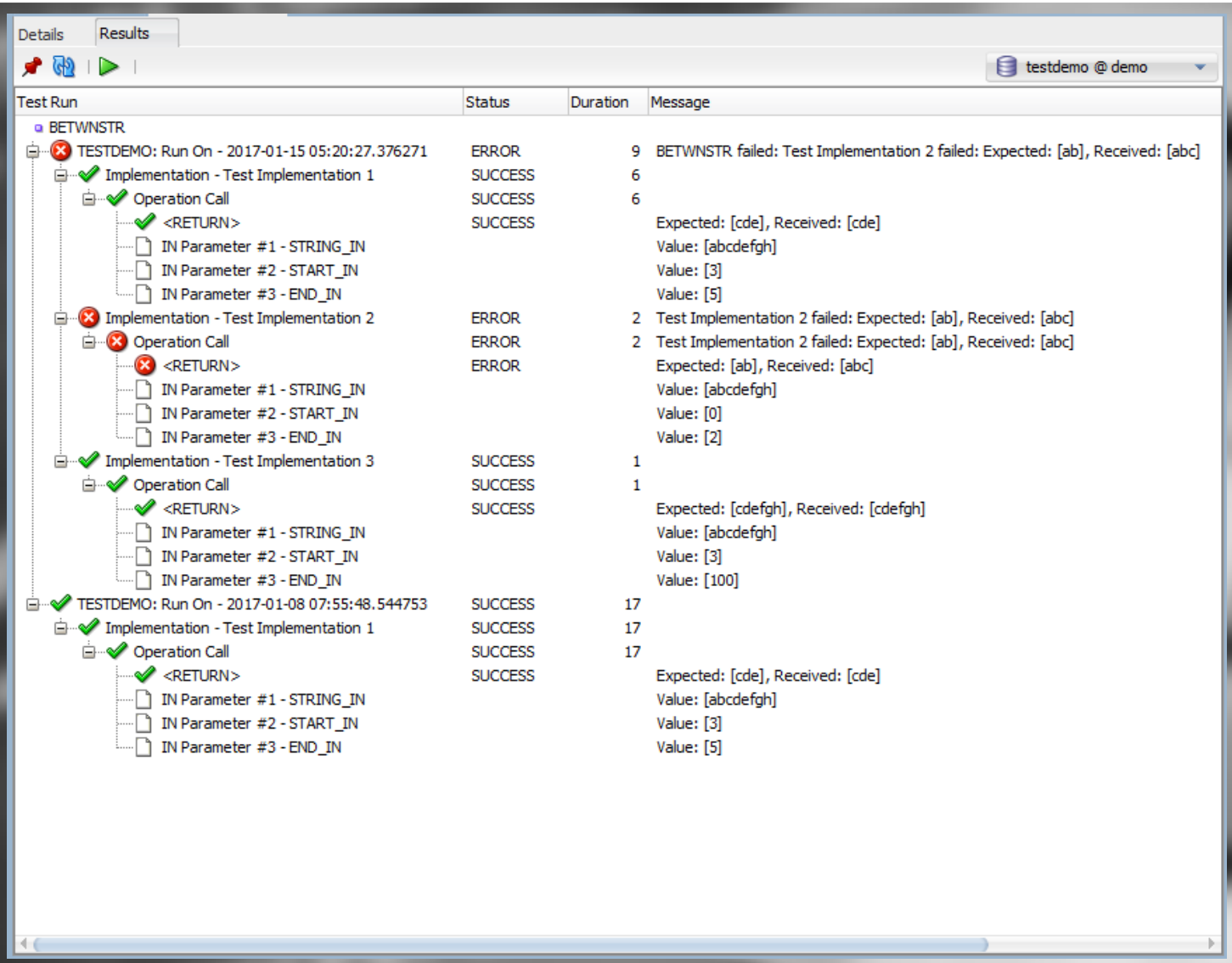
Expected Result Success ANY Enter expected error number or "ANY".

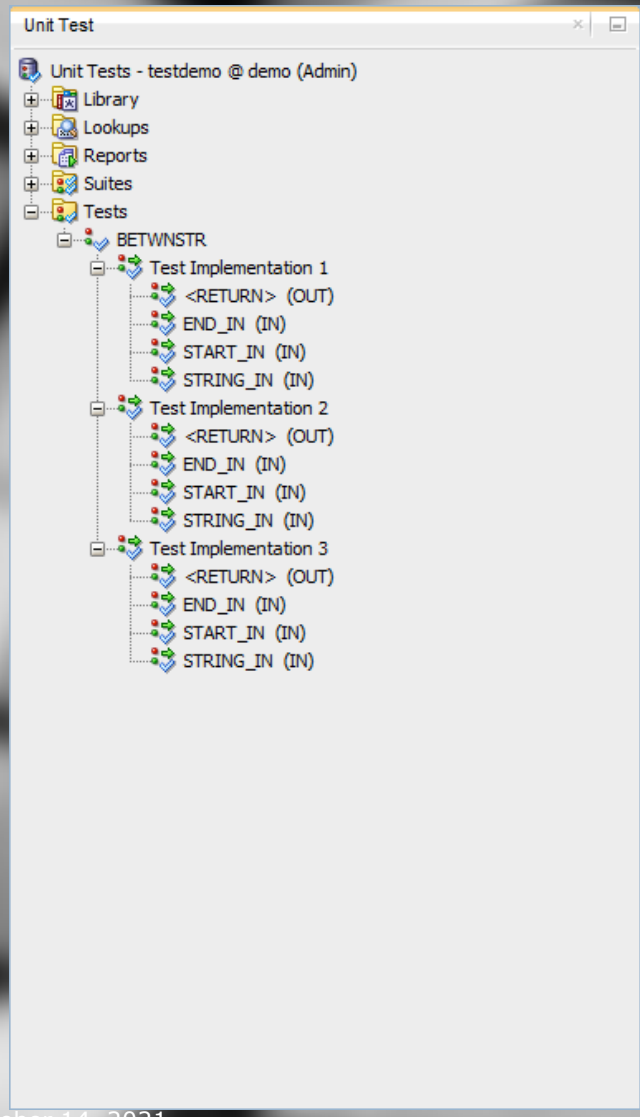
Unit Test

- Unit Tests - testdemo @ demo (Admin)
  - Library
  - Lookups
  - Reports
  - Suites
  - Tests
    - BETWNSTR
      - Test Implementation 1
        - <RETURN> (OUT)
        - END_IN (IN)
        - START_IN (IN)
        - STRING_IN (IN)
      - Test Implementation 2
        - <RETURN> (OUT)
        - END_IN (IN)
        - START_IN (IN)
        - STRING_IN (IN)


Add Implementation
Test Implementation Name
Test Implementation 3
Help
OK
Cancel
Unit Test
Unit Tests - testdemo @ demo (Admin)
Library
Lookups
Reports
Suites
Tests
BETWNSTR
Test Implementation 1
<RETURN> (OUT)
END_IN (IN)
START_IN (IN)
STRING_IN (IN)
Test Implementation 2
<RETURN> (OUT)
END_IN (IN)
START_IN (IN)
STRING_IN (IN)

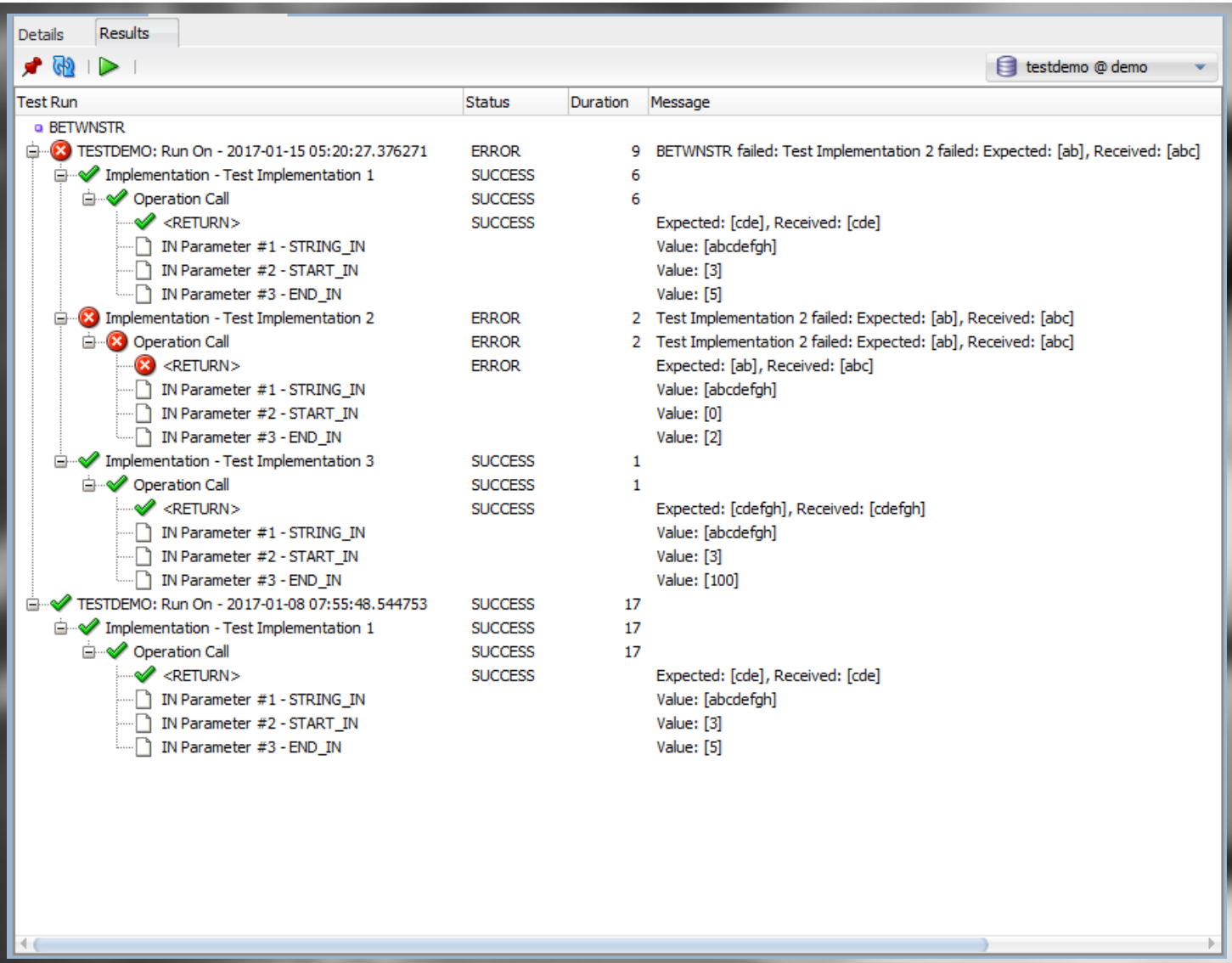

Unit Test
Unit Tests - testdemo @ demo (Admin)
Library
Lookups
Reports
Suites
Tests
BETWNSTR
Test Implementation 1
<RETURN> (OUT)
END_IN (IN)
START_IN (IN)
STRING_IN (IN)
Test Implementation 2
<RETURN> (OUT)
END_IN (IN)
START_IN (IN)
STRING_IN (IN)
Test Implementation 3
<RETURN> (OUT)
END_IN (IN)
START_IN (IN)
STRING_IN (IN)

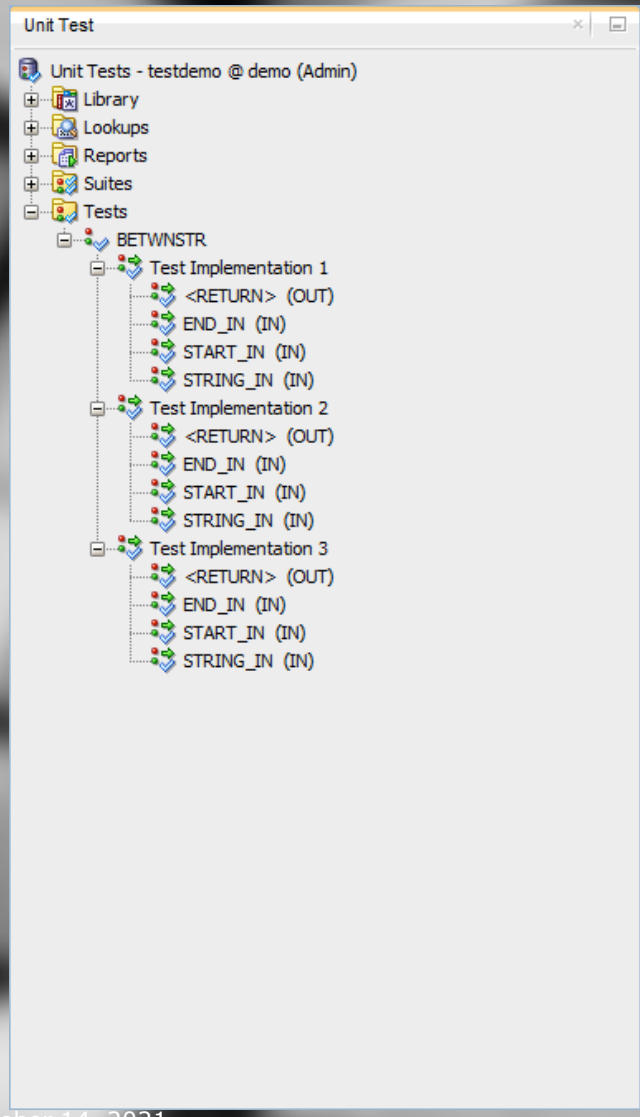

Details
Results
testdemo @ demo
Test Run
Status
Duration
Message
BETWNSTR
TESTDEMO: Run On - 2017-01-15 05:20:27.376271 ERROR 9 BETWNSTR failed: Test Implementation 2 failed: Expected: [ab], Received: [abc]
Implementation - Test Implementation 1 SUCCESS 6
Operation Call SUCCESS 6
<RETURN> SUCCESS Expected: [cde], Received: [cde]
IN Parameter #1 - STRING_IN Value: [abcdefgh]
IN Parameter #2 - START_IN Value: [3]
IN Parameter #3 - END_IN Value: [5]
Implementation - Test Implementation 2 ERROR 2 Test Implementation 2 failed: Expected: [ab], Received: [abc]
Operation Call ERROR 2 Test Implementation 2 failed: Expected: [ab], Received: [abc]
<RETURN> ERROR Expected: [ab], Received: [abc]
IN Parameter #1 - STRING_IN Value: [abcdefgh]
IN Parameter #2 - START_IN Value: [0]
IN Parameter #3 - END_IN Value: [2]
Implementation - Test Implementation 3 SUCCESS 1
Operation Call SUCCESS 1
<RETURN> SUCCESS Expected: [cdefgh], Received: [cdefgh]
IN Parameter #1 - STRING_IN Value: [abcdefgh]
IN Parameter #2 - START_IN Value: [3]
IN Parameter #3 - END_IN Value: [100]
TESTDEMO: Run On - 2017-01-08 07:55:48.544753 SUCCESS 17
Implementation - Test Implementation 1 SUCCESS 17
Operation Call SUCCESS 17
<RETURN> SUCCESS Expected: [cde], Received: [cde]
IN Parameter #1 - STRING_IN Value: [abcdefgh]
IN Parameter #2 - START_IN Value: [3]
IN Parameter #3 - END_IN Value: [5]

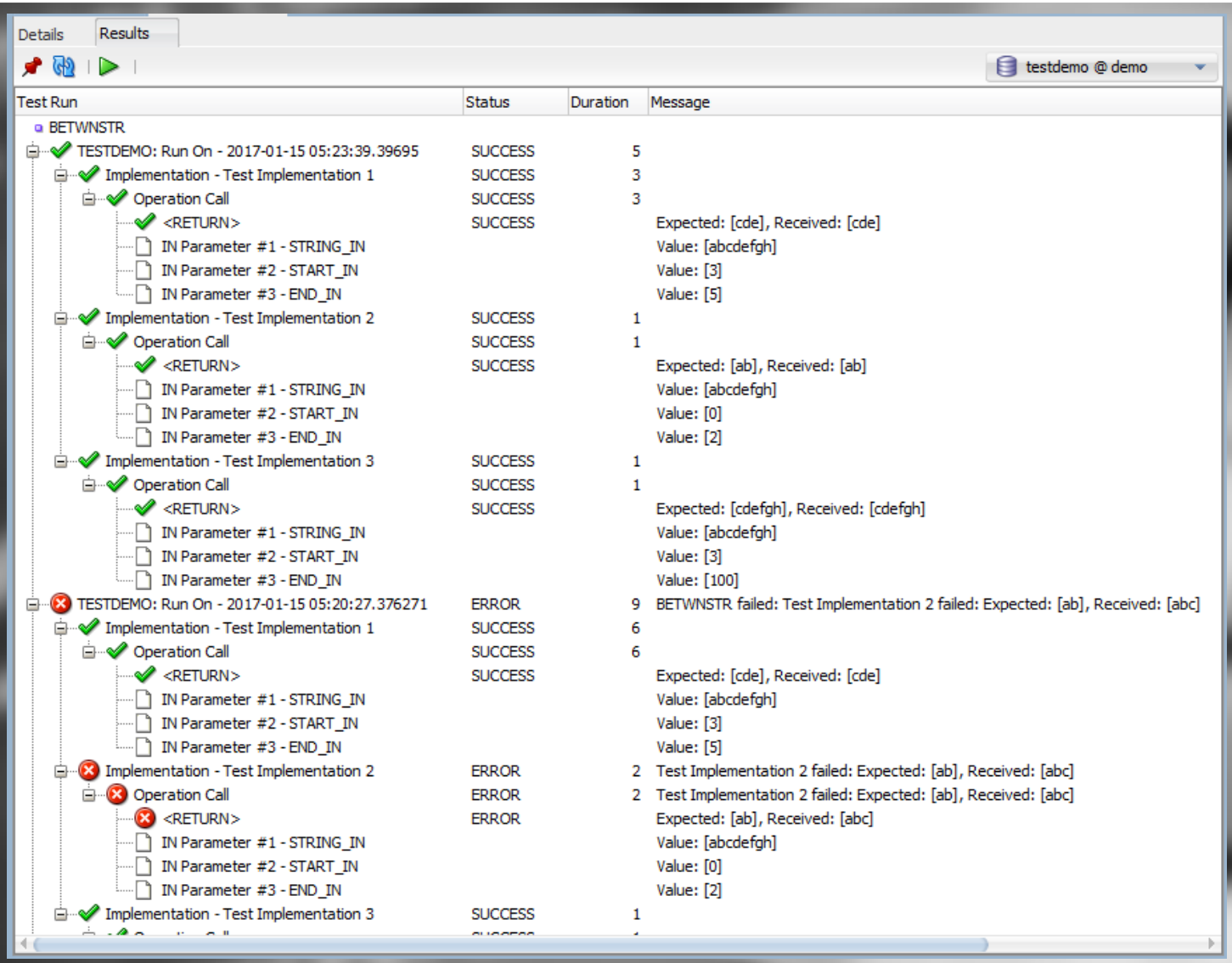

Unit Test
Unit Tests - testdemo @ demo (Admin)
Library
Lookups
Reports
Suites
Tests
BETWNSTR
Test Implementation 1
<RETURN> (OUT)
END_IN (IN)
START_IN (IN)
STRING_IN (IN)
Test Implementation 2
<RETURN> (OUT)
END_IN (IN)
START_IN (IN)
STRING_IN (IN)
Test Implementation 3
<RETURN> (OUT)
END_IN (IN)
START_IN (IN)
STRING_IN (IN)

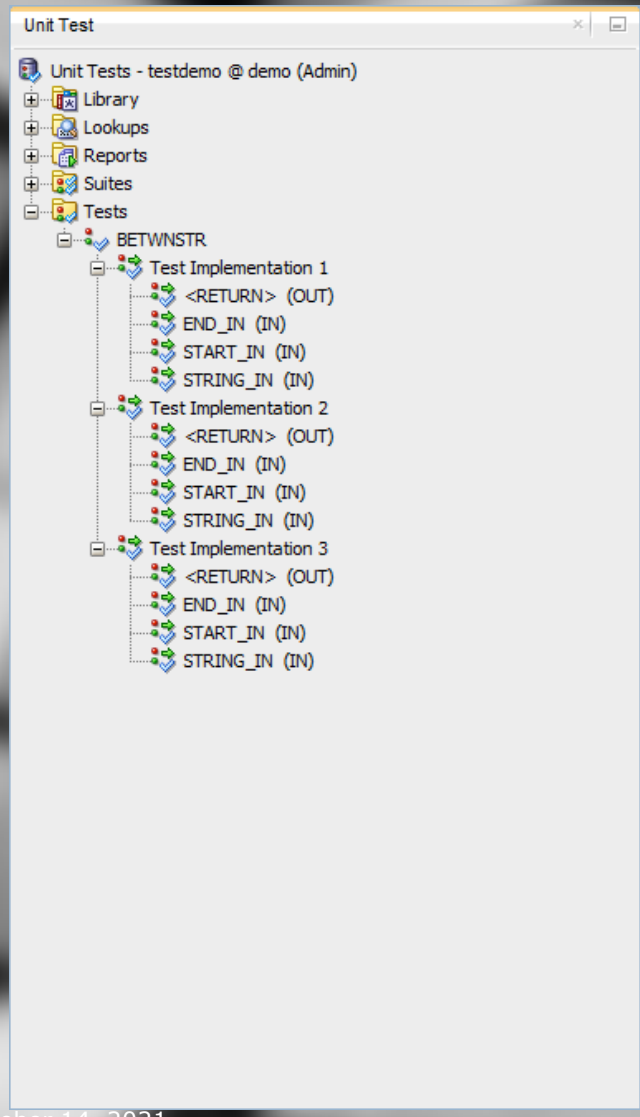
```
SQL> create or replace function betwnstr(string_in in varchar2
  2                                      ,start_in  in integer
  3                                      ,end_in    in integer) return varchar2 is
  4  begin
  5    return substr(string_in, start_in, end_in - start_in + 1);
  6  end betwnstr;
  7  /
```

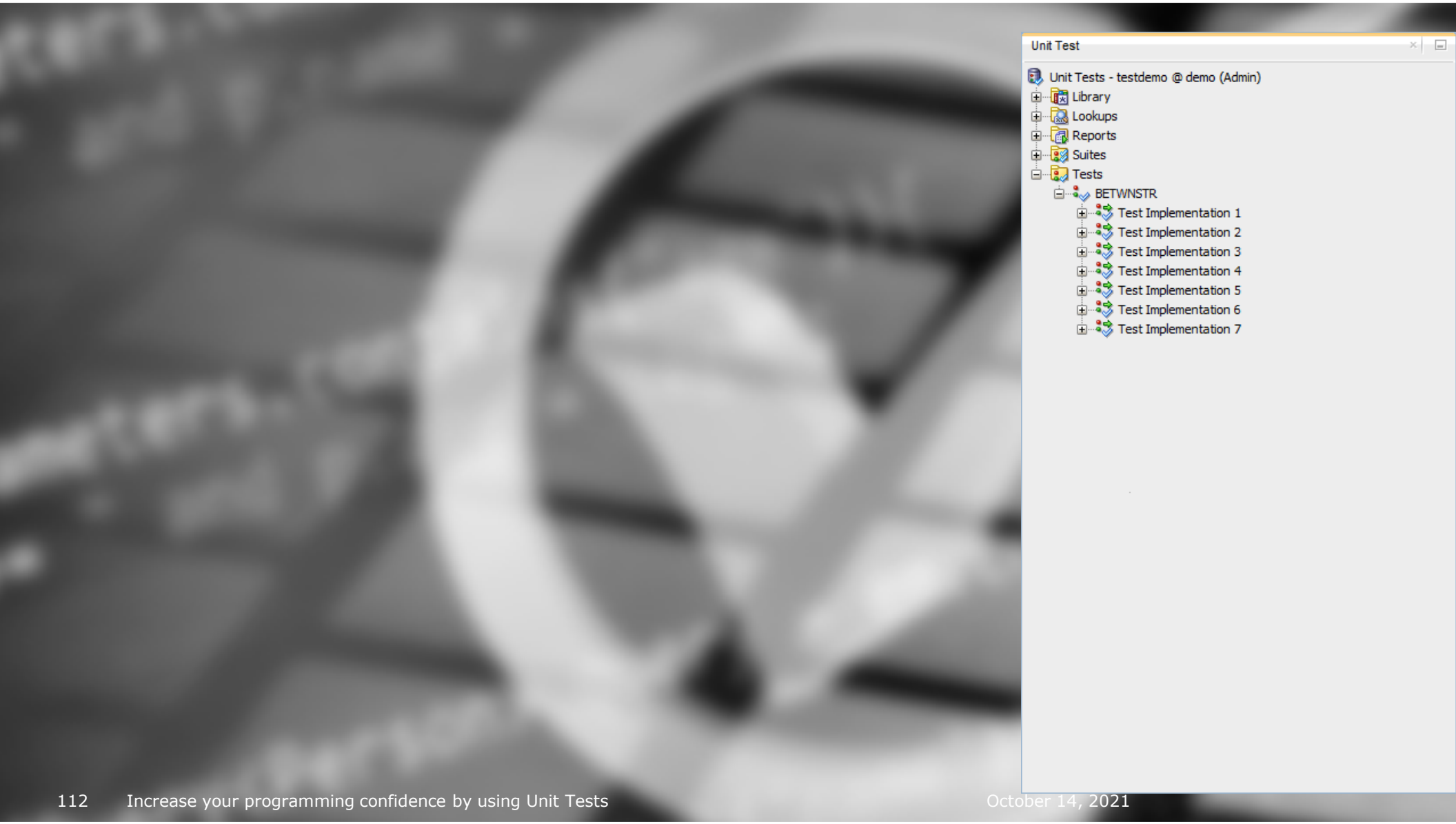
```
SQL> create or replace function betwnstr(string_in in varchar2
  2                                      ,start_in  in integer
  3                                      ,end_in    in integer) return varchar2 is
  4    l_start integer := start_in;
  5  begin
  6    -- 0 should be start of string so change it to 1
  7    if l_start = 0 then
  8      l_start := 1;
  9    end if;
 10
 11    return substr(string_in, l_start, end_in - l_start + 1);
 12  end betwnstr;
 13  /
```

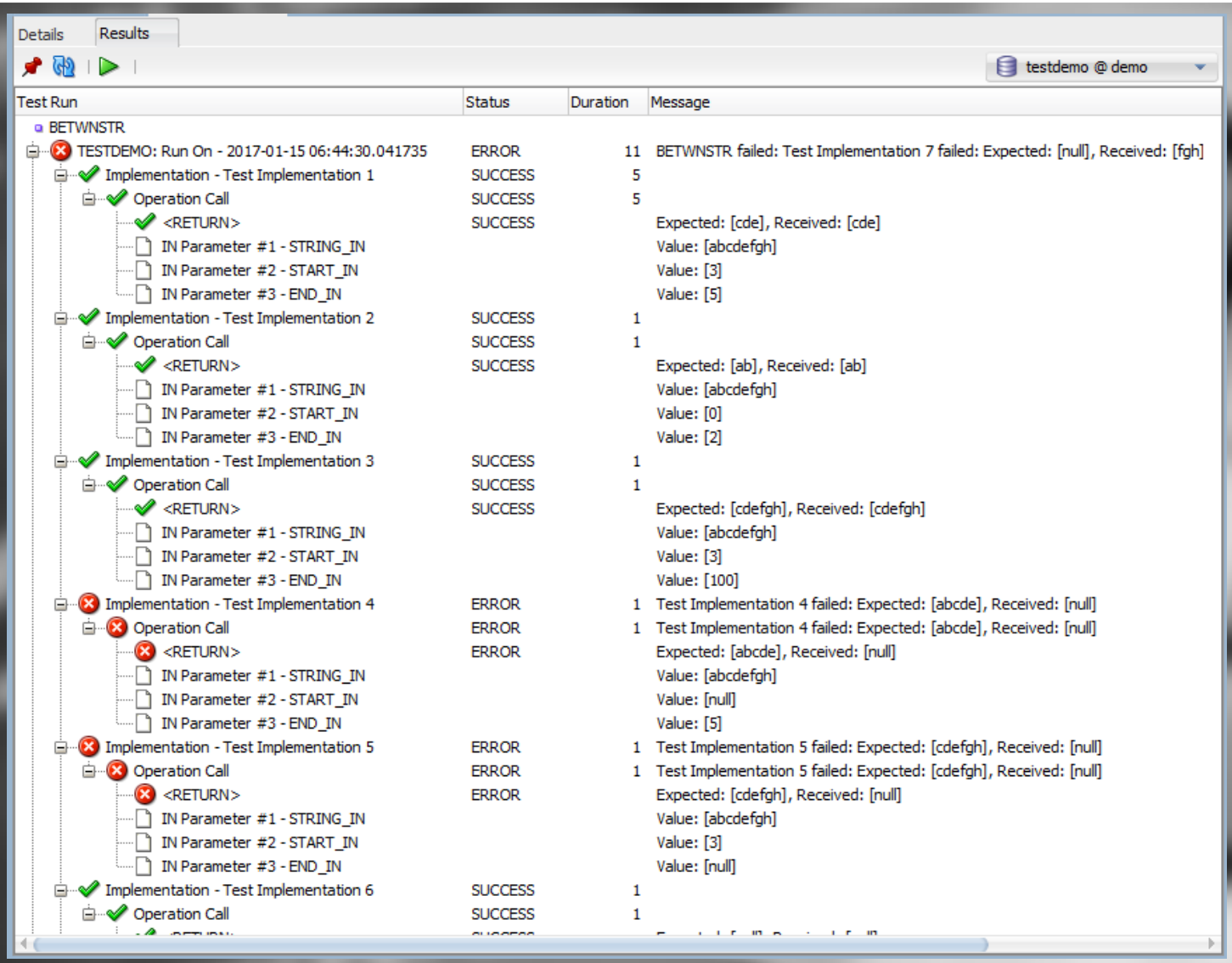

Details
Results
testdemo @ demo
Test Run
Status
Duration
Message
BETWNSTR
TESTDEMO: Run On - 2017-01-15 05:20:27.376271 ERROR 9 BETWNSTR failed: Test Implementation 2 failed: Expected: [ab], Received: [abc]
Implementation - Test Implementation 1 SUCCESS 6
Operation Call SUCCESS 6
<RETURN> SUCCESS Expected: [cde], Received: [cde]
IN Parameter #1 - STRING_IN Value: [abcdefgh]
IN Parameter #2 - START_IN Value: [3]
IN Parameter #3 - END_IN Value: [5]
Implementation - Test Implementation 2 ERROR 2 Test Implementation 2 failed: Expected: [ab], Received: [abc]
Operation Call ERROR 2 Test Implementation 2 failed: Expected: [ab], Received: [abc]
<RETURN> ERROR Expected: [ab], Received: [abc]
IN Parameter #1 - STRING_IN Value: [abcdefgh]
IN Parameter #2 - START_IN Value: [0]
IN Parameter #3 - END_IN Value: [2]
Implementation - Test Implementation 3 SUCCESS 1
Operation Call SUCCESS 1
<RETURN> SUCCESS Expected: [cdefgh], Received: [cdefgh]
IN Parameter #1 - STRING_IN Value: [abcdefgh]
IN Parameter #2 - START_IN Value: [3]
IN Parameter #3 - END_IN Value: [100]
TESTDEMO: Run On - 2017-01-08 07:55:48.544753 SUCCESS 17
Implementation - Test Implementation 1 SUCCESS 17
Operation Call SUCCESS 17
<RETURN> SUCCESS Expected: [cde], Received: [cde]
IN Parameter #1 - STRING_IN Value: [abcdefgh]
IN Parameter #2 - START_IN Value: [3]
IN Parameter #3 - END_IN Value: [5]

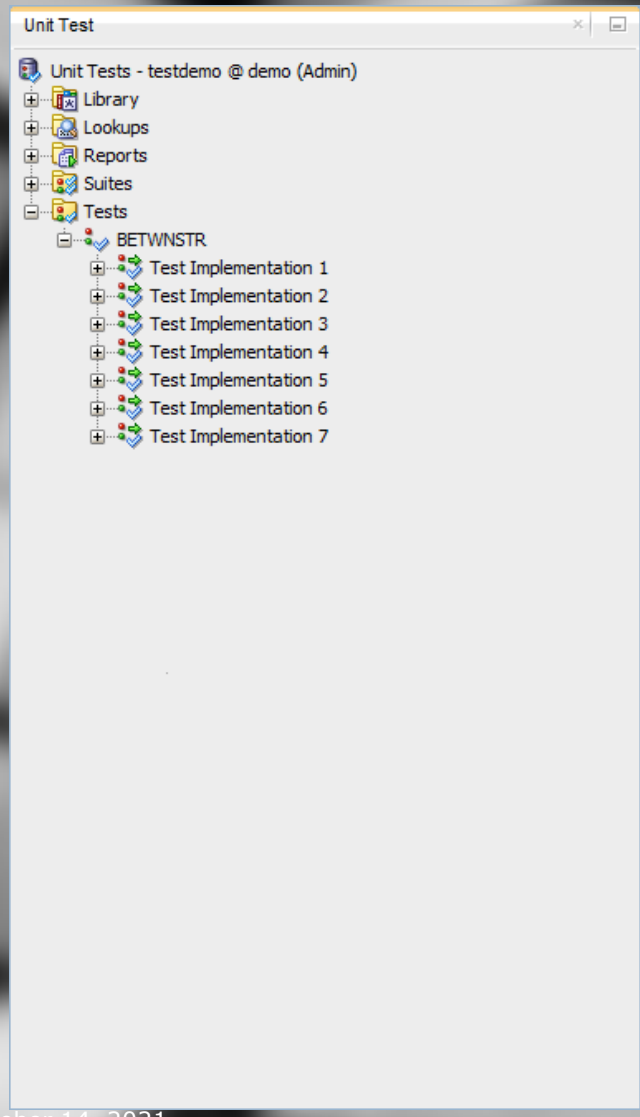

Unit Test
Unit Tests - testdemo @ demo (Admin)
Library
Lookups
Reports
Suites
Tests
BETWNSTR
Test Implementation 1
<RETURN> (OUT)
END_IN (IN)
START_IN (IN)
STRING_IN (IN)
Test Implementation 2
<RETURN> (OUT)
END_IN (IN)
START_IN (IN)
STRING_IN (IN)
Test Implementation 3
<RETURN> (OUT)
END_IN (IN)
START_IN (IN)
STRING_IN (IN)


Details
Results
testdemo @ demo
Test Run
Status
Duration
Message
BETWNSTR
TESTDEMO: Run On - 2017-01-15 05:23:39.39695 SUCCESS 5
Implementation - Test Implementation 1 SUCCESS 3
Operation Call SUCCESS 3
<RETURN> SUCCESS Expected: [cde], Received: [cde]
IN Parameter #1 - STRING_IN Value: [abcdefgh]
IN Parameter #2 - START_IN Value: [3]
IN Parameter #3 - END_IN Value: [5]
Implementation - Test Implementation 2 SUCCESS 1
Operation Call SUCCESS 1
<RETURN> SUCCESS Expected: [ab], Received: [ab]
IN Parameter #1 - STRING_IN Value: [abcdefgh]
IN Parameter #2 - START_IN Value: [0]
IN Parameter #3 - END_IN Value: [2]
Implementation - Test Implementation 3 SUCCESS 1
Operation Call SUCCESS 1
<RETURN> SUCCESS Expected: [cdefgh], Received: [cdefgh]
IN Parameter #1 - STRING_IN Value: [abcdefgh]
IN Parameter #2 - START_IN Value: [3]
IN Parameter #3 - END_IN Value: [100]
TESTDEMO: Run On - 2017-01-15 05:20:27.376271 ERROR 9 BETWNSTR failed: Test Implementation 2 failed: Expected: [ab], Received: [abc]
Implementation - Test Implementation 1 SUCCESS 6
Operation Call SUCCESS 6
<RETURN> SUCCESS Expected: [cde], Received: [cde]
IN Parameter #1 - STRING_IN Value: [abcdefgh]
IN Parameter #2 - START_IN Value: [3]
IN Parameter #3 - END_IN Value: [5]
Implementation - Test Implementation 2 ERROR 2 Test Implementation 2 failed: Expected: [ab], Received: [abc]
Operation Call ERROR 2 Test Implementation 2 failed: Expected: [ab], Received: [abc]
<RETURN> ERROR Expected: [ab], Received: [abc]
IN Parameter #1 - STRING_IN Value: [abcdefgh]
IN Parameter #2 - START_IN Value: [0]
IN Parameter #3 - END_IN Value: [2]
Implementation - Test Implementation 3 SUCCESS 1


Unit Test
Unit Tests - testdemo @ demo (Admin)
Library
Lookups
Reports
Suites
Tests
BETWNSTR
Test Implementation 1
<RETURN> (OUT)
END_IN (IN)
START_IN (IN)
STRING_IN (IN)
Test Implementation 2
<RETURN> (OUT)
END_IN (IN)
START_IN (IN)
STRING_IN (IN)
Test Implementation 3
<RETURN> (OUT)
END_IN (IN)
START_IN (IN)
STRING_IN (IN)

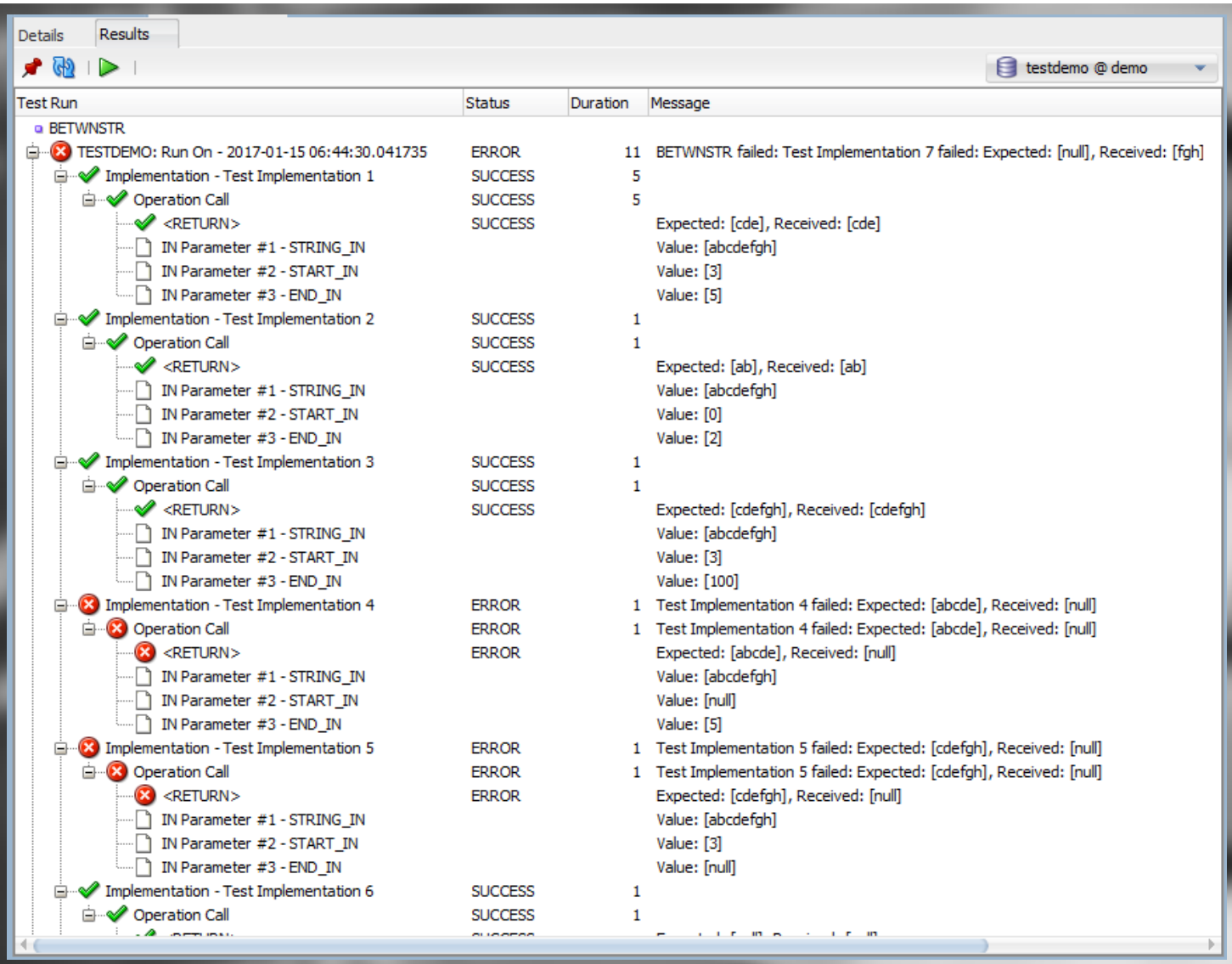

Unit Test
Unit Tests - testdemo @ demo (Admin)
Library
Lookups
Reports
Suites
Tests
BETWNSTR
Test Implementation 1
Test Implementation 2
Test Implementation 3
Test Implementation 4
Test Implementation 5
Test Implementation 6
Test Implementation 7

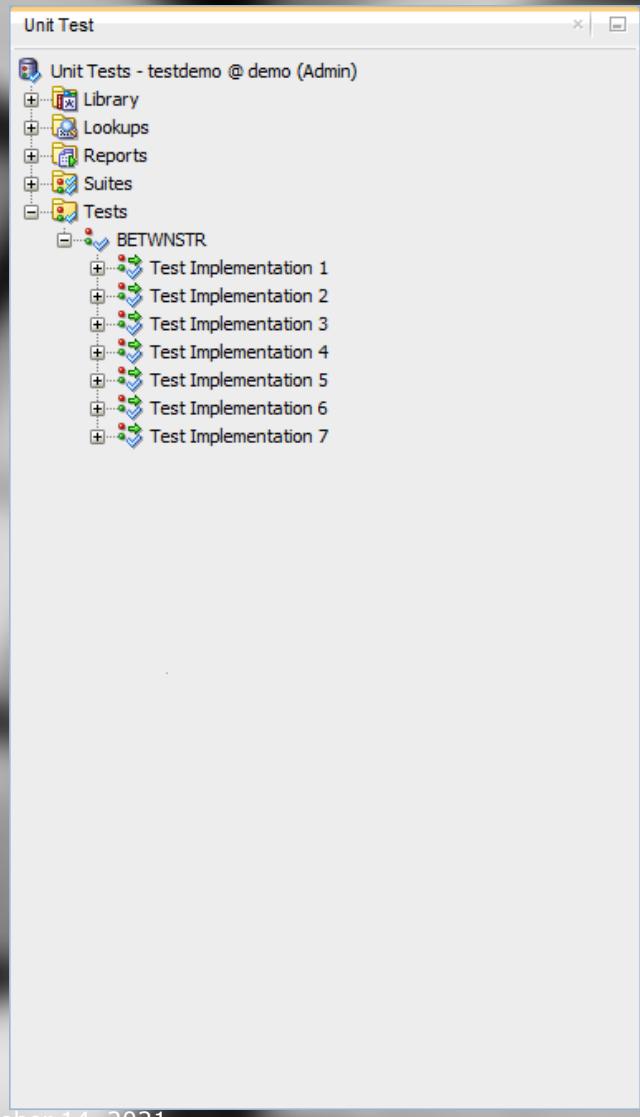

Details
Results
testdemo @ demo
Test Run
Status
Duration
Message
BETWNSTR
TESTDEMO: Run On - 2017-01-15 06:44:30.041735 ERROR 11 BETWNSTR failed: Test Implementation 7 failed: Expected: [null], Received: [fgh]
Implementation - Test Implementation 1 SUCCESS 5
Operation Call SUCCESS 5
<RETURN> SUCCESS Expected: [cde], Received: [cde]
IN Parameter #1 - STRING_IN Value: [abcdefgh]
IN Parameter #2 - START_IN Value: [3]
IN Parameter #3 - END_IN Value: [5]
Implementation - Test Implementation 2 SUCCESS 1
Operation Call SUCCESS 1
<RETURN> SUCCESS Expected: [ab], Received: [ab]
IN Parameter #1 - STRING_IN Value: [abcdefgh]
IN Parameter #2 - START_IN Value: [0]
IN Parameter #3 - END_IN Value: [2]
Implementation - Test Implementation 3 SUCCESS 1
Operation Call SUCCESS 1
<RETURN> SUCCESS Expected: [cdefgh], Received: [cdefgh]
IN Parameter #1 - STRING_IN Value: [abcdefgh]
IN Parameter #2 - START_IN Value: [3]
IN Parameter #3 - END_IN Value: [100]
Implementation - Test Implementation 4 ERROR 1 Test Implementation 4 failed: Expected: [abcde], Received: [null]
Operation Call ERROR 1 Test Implementation 4 failed: Expected: [abcde], Received: [null]
<RETURN> ERROR Expected: [abcde], Received: [null]
IN Parameter #1 - STRING_IN Value: [abcdefgh]
IN Parameter #2 - START_IN Value: [null]
IN Parameter #3 - END_IN Value: [5]
Implementation - Test Implementation 5 ERROR 1 Test Implementation 5 failed: Expected: [cdefgh], Received: [null]
Operation Call ERROR 1 Test Implementation 5 failed: Expected: [cdefgh], Received: [null]
<RETURN> ERROR Expected: [cdefgh], Received: [null]
IN Parameter #1 - STRING_IN Value: [abcdefgh]
IN Parameter #2 - START_IN Value: [3]
IN Parameter #3 - END_IN Value: [null]
Implementation - Test Implementation 6 SUCCESS 1
Operation Call SUCCESS 1

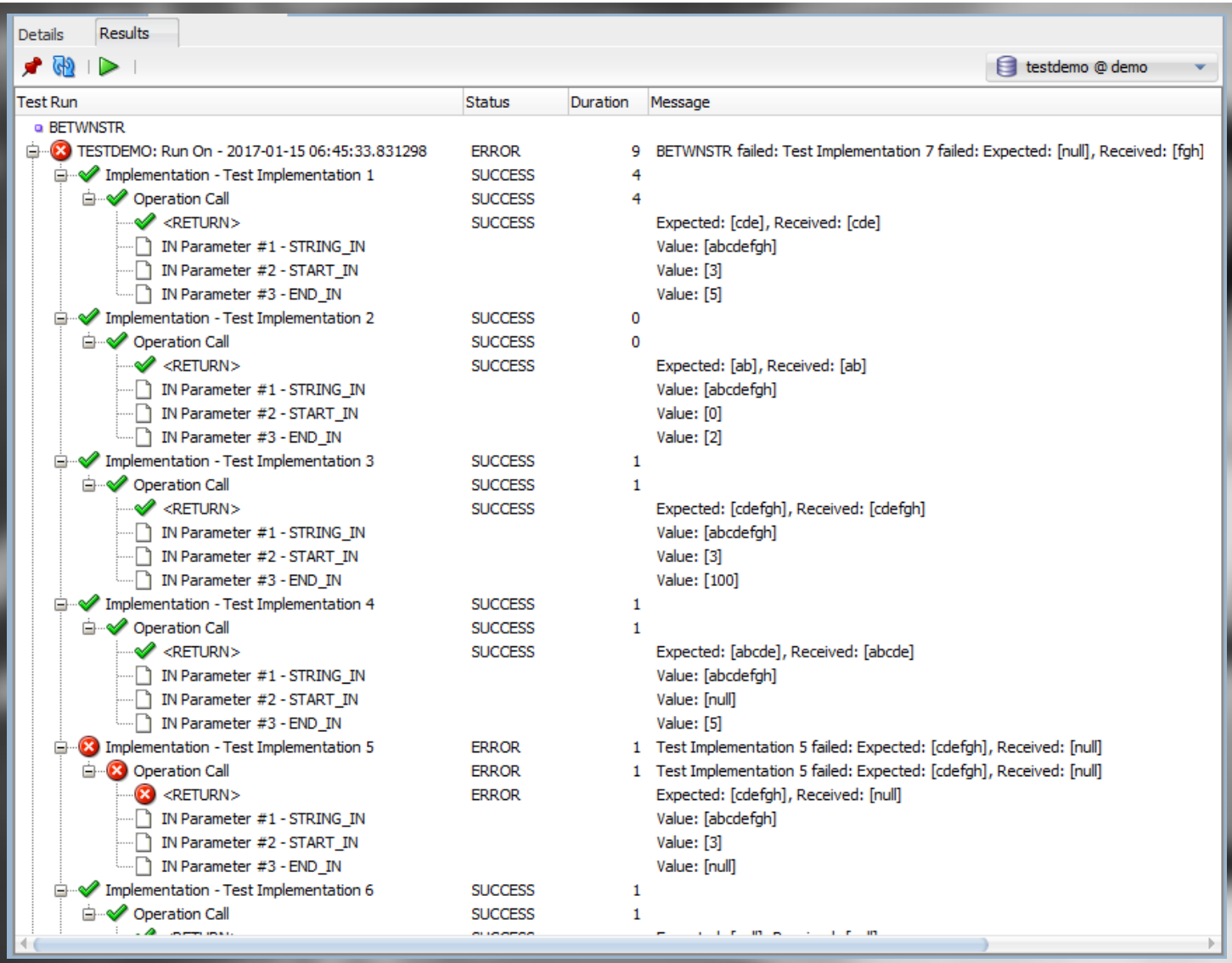

Unit Test
Unit Tests - testdemo @ demo (Admin)
Library
Lookups
Reports
Suites
Tests
BETWNSTR
Test Implementation 1
Test Implementation 2
Test Implementation 3
Test Implementation 4
Test Implementation 5
Test Implementation 6
Test Implementation 7

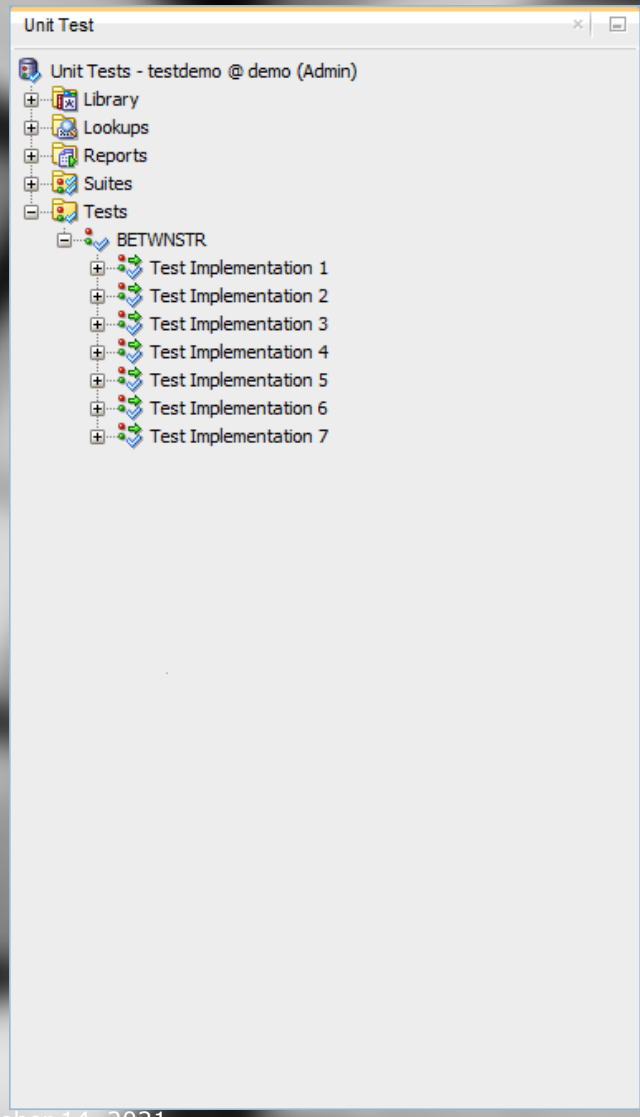
```
SQL> create or replace function betwnstr(string_in in varchar2
  2                                      ,start_in  in integer
  3                                      ,end_in    in integer) return varchar2 is
  4    l_start integer := start_in;
  5  begin
  6    -- 0 should be start of string so change it to 1
  7    if l_start = 0 then
  8      l_start := 1;
  9    end if;
 10
 11    return substr(string_in, l_start, end_in - l_start + 1);
 12  end betwnstr;
 13  /
```

```
SQL> create or replace function betwnstr(string_in in varchar2
  2                                      ,start_in  in integer
  3                                      ,end_in    in integer) return varchar2 is
  4    l_start integer := start_in;
  5  begin
  6    -- 0 should be start of string so change it to 1
  7    if l_start = 0 then
  8      l_start := 1;
  9    end if;
 10    if l_start is null then
 11      l_start := 1;
 12    end if;
 13    return substr(string_in, l_start, end_in - l_start + 1);
 14  end betwnstr;
 15  /
```


Details
Results
testdemo @ demo
Test Run
Status
Duration
Message
BETWNSTR
TESTDEMO: Run On - 2017-01-15 06:44:30.041735 ERROR 11 BETWNSTR failed: Test Implementation 7 failed: Expected: [null], Received: [fgh]
Implementation - Test Implementation 1 SUCCESS 5
Operation Call SUCCESS 5
<RETURN> SUCCESS Expected: [cde], Received: [cde]
IN Parameter #1 - STRING_IN Value: [abcdefgh]
IN Parameter #2 - START_IN Value: [3]
IN Parameter #3 - END_IN Value: [5]
Implementation - Test Implementation 2 SUCCESS 1
Operation Call SUCCESS 1
<RETURN> SUCCESS Expected: [ab], Received: [ab]
IN Parameter #1 - STRING_IN Value: [abcdefgh]
IN Parameter #2 - START_IN Value: [0]
IN Parameter #3 - END_IN Value: [2]
Implementation - Test Implementation 3 SUCCESS 1
Operation Call SUCCESS 1
<RETURN> SUCCESS Expected: [cdefgh], Received: [cdefgh]
IN Parameter #1 - STRING_IN Value: [abcdefgh]
IN Parameter #2 - START_IN Value: [3]
IN Parameter #3 - END_IN Value: [100]
Implementation - Test Implementation 4 ERROR 1 Test Implementation 4 failed: Expected: [abcde], Received: [null]
Operation Call ERROR 1 Test Implementation 4 failed: Expected: [abcde], Received: [null]
<RETURN> ERROR Expected: [abcde], Received: [null]
IN Parameter #1 - STRING_IN Value: [abcdefgh]
IN Parameter #2 - START_IN Value: [null]
IN Parameter #3 - END_IN Value: [5]
Implementation - Test Implementation 5 ERROR 1 Test Implementation 5 failed: Expected: [cdefgh], Received: [null]
Operation Call ERROR 1 Test Implementation 5 failed: Expected: [cdefgh], Received: [null]
<RETURN> ERROR Expected: [cdefgh], Received: [null]
IN Parameter #1 - STRING_IN Value: [abcdefgh]
IN Parameter #2 - START_IN Value: [3]
IN Parameter #3 - END_IN Value: [null]
Implementation - Test Implementation 6 SUCCESS 1
Operation Call SUCCESS 1

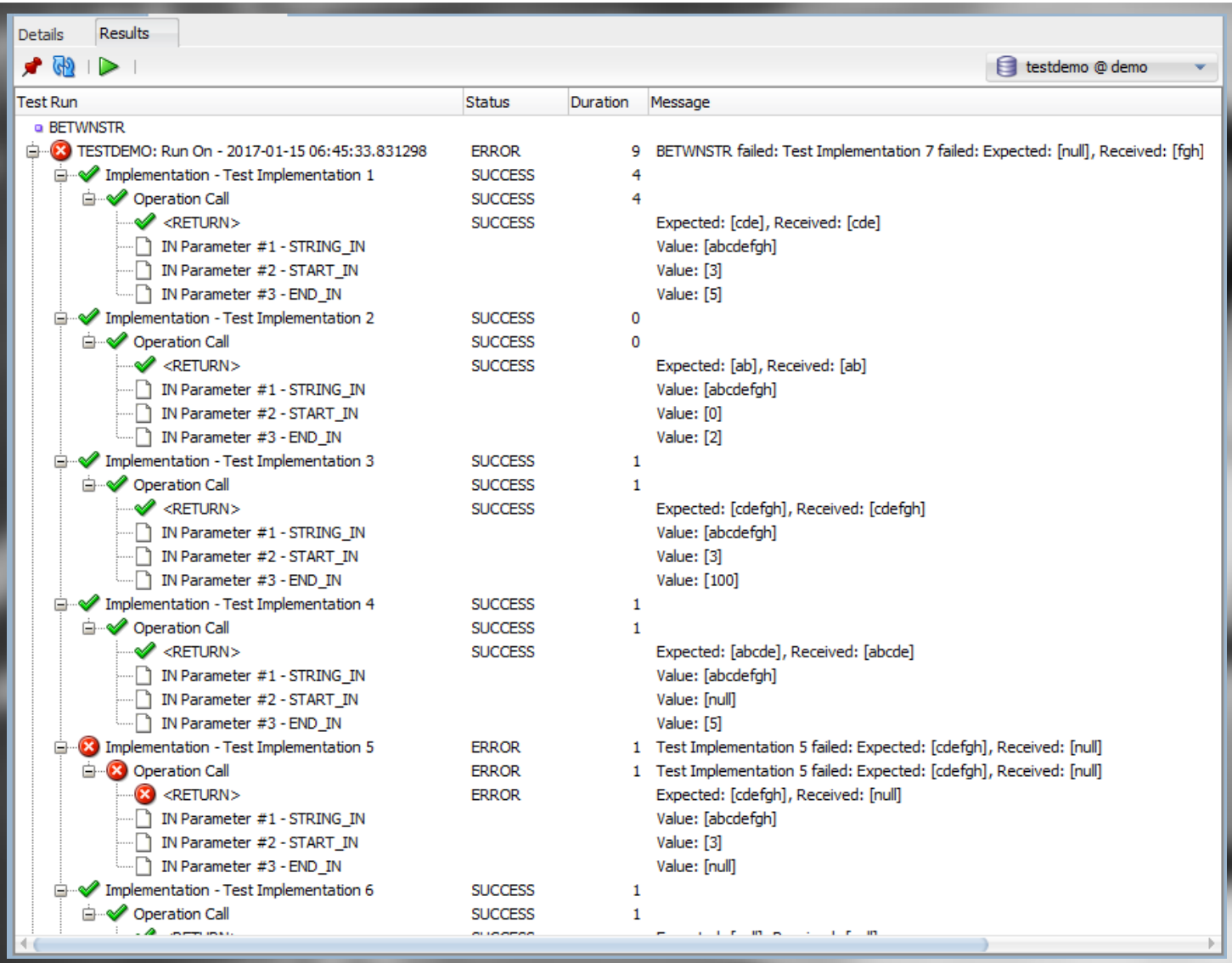

Unit Test
Unit Tests - testdemo @ demo (Admin)
Library
Lookups
Reports
Suites
Tests
BETWNSTR
Test Implementation 1
Test Implementation 2
Test Implementation 3
Test Implementation 4
Test Implementation 5
Test Implementation 6
Test Implementation 7

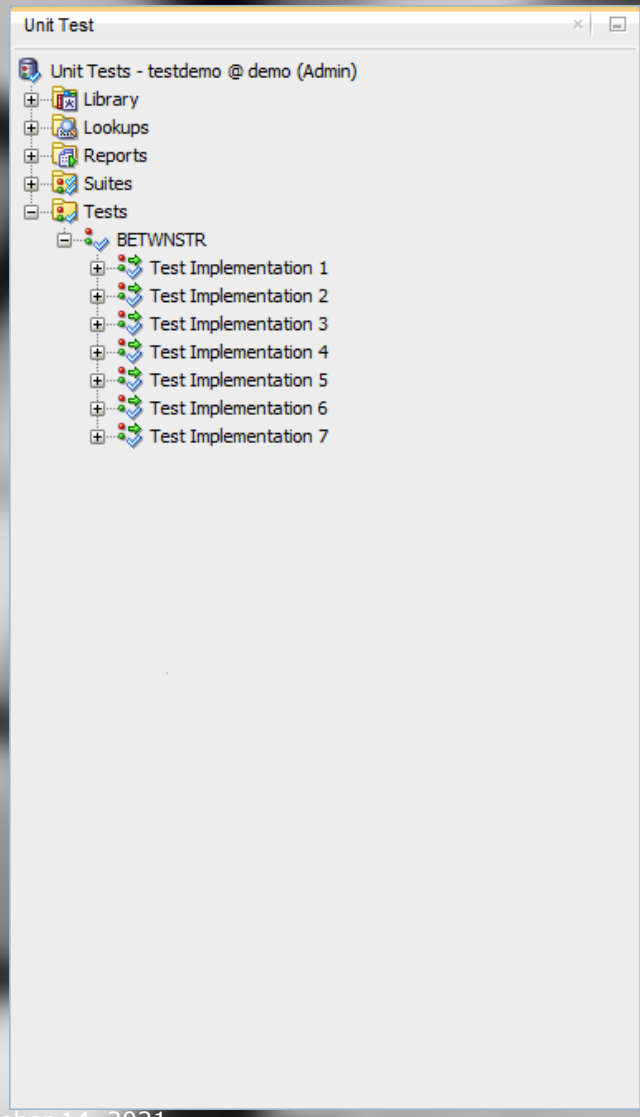

Details
Results
testdemo @ demo
Test Run
Status
Duration
Message
BETWNSTR
TESTDEMO: Run On - 2017-01-15 06:45:33.831298 ERROR 9 BETWNSTR failed: Test Implementation 7 failed: Expected: [null], Received: [fgh]
Implementation - Test Implementation 1 SUCCESS 4
Operation Call SUCCESS 4
<RETURN> SUCCESS Expected: [cde], Received: [cde]
IN Parameter #1 - STRING_IN Value: [abcdefgh]
IN Parameter #2 - START_IN Value: [3]
IN Parameter #3 - END_IN Value: [5]
Implementation - Test Implementation 2 SUCCESS 0
Operation Call SUCCESS 0
<RETURN> SUCCESS Expected: [ab], Received: [ab]
IN Parameter #1 - STRING_IN Value: [abcdefgh]
IN Parameter #2 - START_IN Value: [0]
IN Parameter #3 - END_IN Value: [2]
Implementation - Test Implementation 3 SUCCESS 1
Operation Call SUCCESS 1
<RETURN> SUCCESS Expected: [cdefgh], Received: [cdefgh]
IN Parameter #1 - STRING_IN Value: [abcdefgh]
IN Parameter #2 - START_IN Value: [3]
IN Parameter #3 - END_IN Value: [100]
Implementation - Test Implementation 4 SUCCESS 1
Operation Call SUCCESS 1
<RETURN> SUCCESS Expected: [abcde], Received: [abcde]
IN Parameter #1 - STRING_IN Value: [abcdefgh]
IN Parameter #2 - START_IN Value: [null]
IN Parameter #3 - END_IN Value: [5]
Implementation - Test Implementation 5 ERROR 1 Test Implementation 5 failed: Expected: [cdefgh], Received: [null]
Operation Call ERROR 1 Test Implementation 5 failed: Expected: [cdefgh], Received: [null]
<RETURN> ERROR Expected: [cdefgh], Received: [null]
IN Parameter #1 - STRING_IN Value: [abcdefgh]
IN Parameter #2 - START_IN Value: [3]
IN Parameter #3 - END_IN Value: [null]
Implementation - Test Implementation 6 SUCCESS 1
Operation Call SUCCESS 1

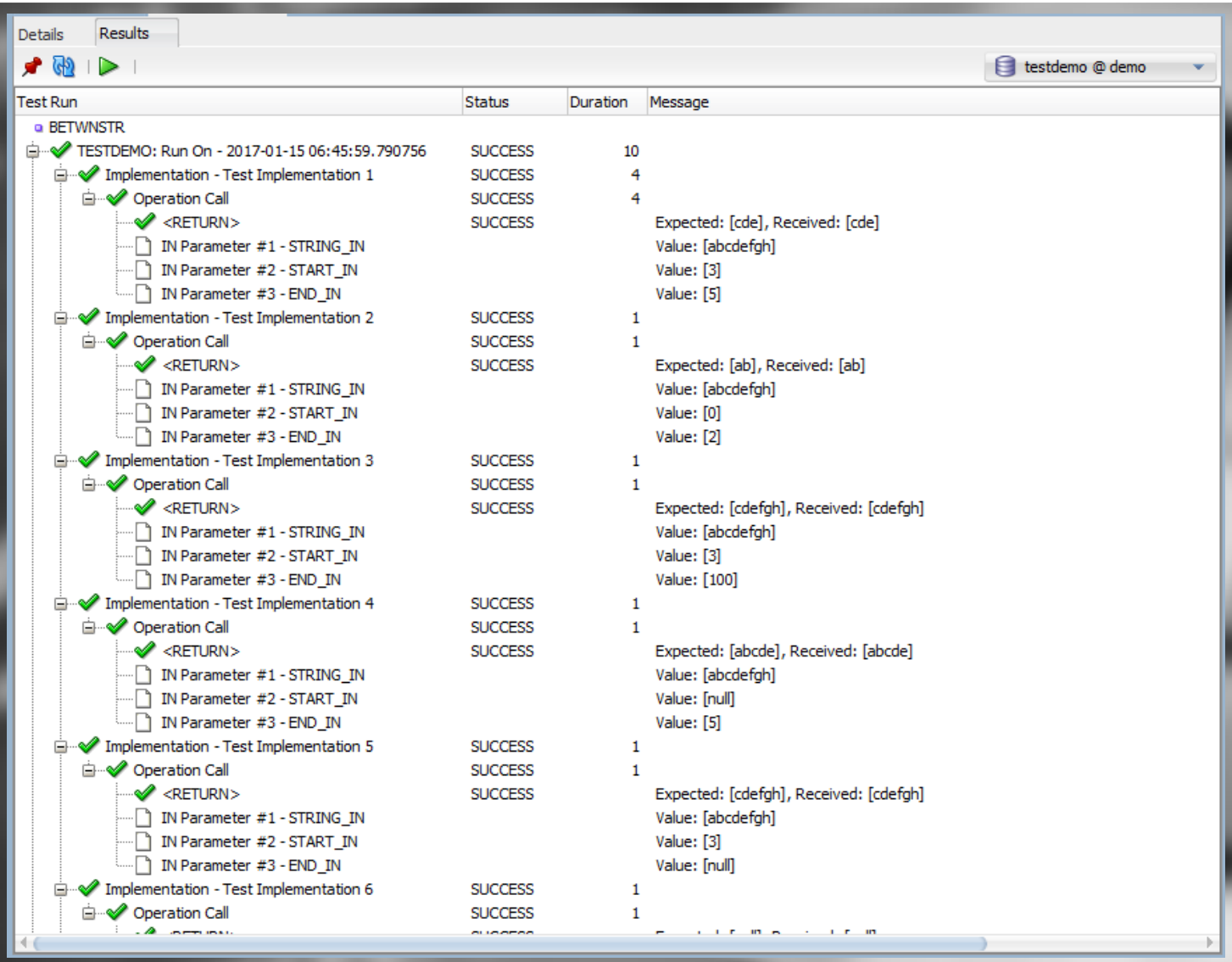

Unit Test
Unit Tests - testdemo @ demo (Admin)
Library
Lookups
Reports
Suites
Tests
BETWNSTR
Test Implementation 1
Test Implementation 2
Test Implementation 3
Test Implementation 4
Test Implementation 5
Test Implementation 6
Test Implementation 7

```
SQL> create or replace function betwnstr(string_in in varchar2
  2                                      ,start_in  in integer
  3                                      ,end_in    in integer) return varchar2 is
  4    l_start integer := start_in;
  5  begin
  6    -- 0 should be start of string so change it to 1
  7    if l_start = 0 then
  8      l_start := 1;
  9    end if;
 10    if l_start is null then
 11      l_start := 1;
 12    end if;
 13    return substr(string_in, l_start, end_in - l_start + 1);
 14  end betwnstr;
 15  /
```

```
SQL> create or replace function betwnstr(string_in in varchar2
  2                                      ,start_in  in integer
  3                                      ,end_in    in integer) return varchar2 is
  4    l_string varchar2(32767) := string_in;
  5    l_start       integer := start_in;
  6    l_end         integer := end_in;
  7    l_returnvalue varchar2(32767);
  8  begin
  9    if (l_start < 1)
 10       or (l_start is null) then
 11      l_start := 1;
 12    end if;
 13
 14    if (l_end is null) then
 15      l_end := length(l_string);
 16    end if;
 17
 18    l_returnvalue := substr(l_string, l_start, (l_end - l_start) + 1);
 19
 20    return l_returnvalue;
 21  end betwnstr;
 22  /
```

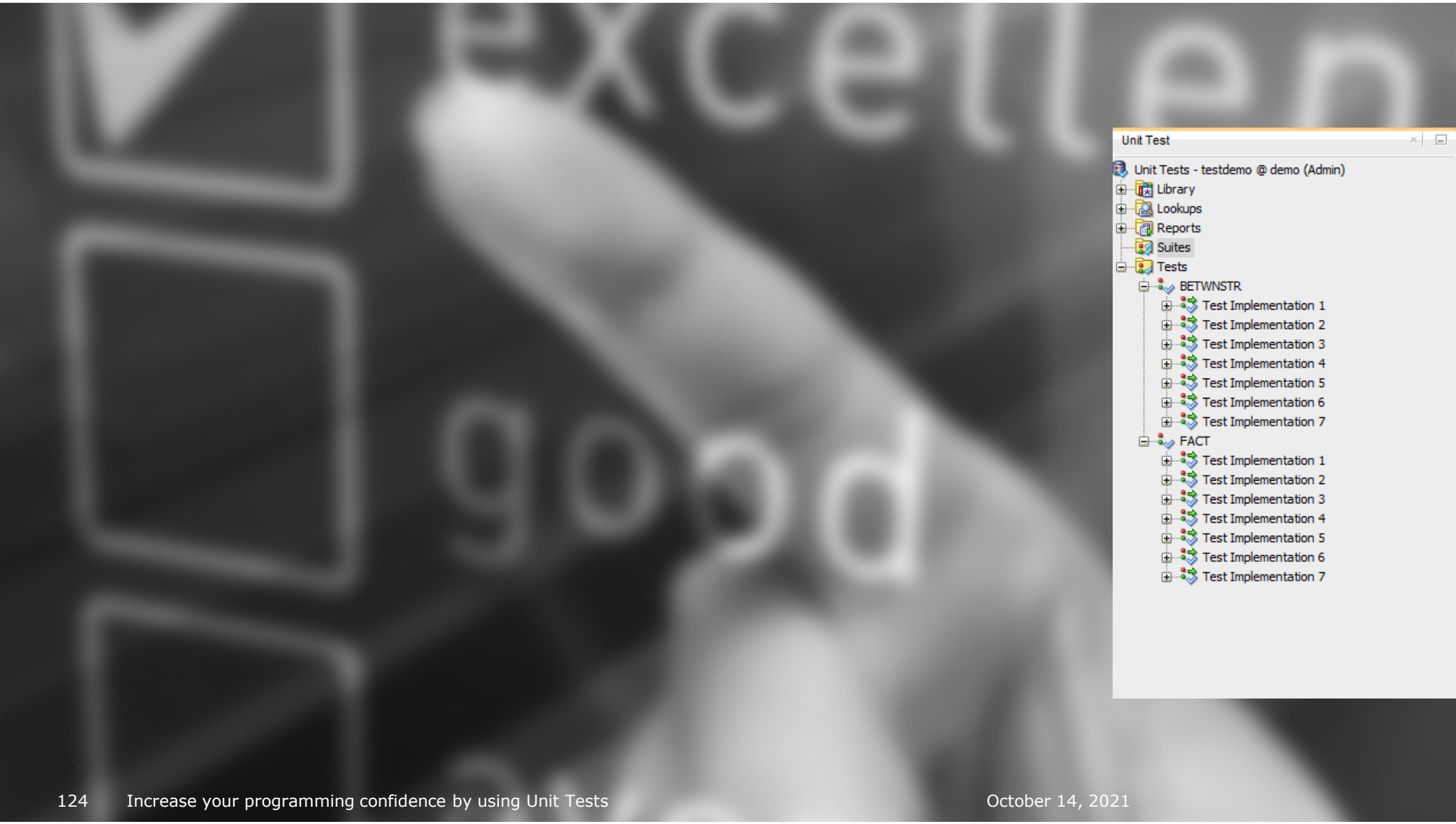

Details
Results
testdemo @ demo
Test Run
Status
Duration
Message
BETWNSTR
TESTDEMO: Run On - 2017-01-15 06:45:33.831298 ERROR 9 BETWNSTR failed: Test Implementation 7 failed: Expected: [null], Received: [fgh]
Implementation - Test Implementation 1 SUCCESS 4
Operation Call SUCCESS 4
<RETURN> SUCCESS Expected: [cde], Received: [cde]
IN Parameter #1 - STRING_IN Value: [abcdefgh]
IN Parameter #2 - START_IN Value: [3]
IN Parameter #3 - END_IN Value: [5]
Implementation - Test Implementation 2 SUCCESS 0
Operation Call SUCCESS 0
<RETURN> SUCCESS Expected: [ab], Received: [ab]
IN Parameter #1 - STRING_IN Value: [abcdefgh]
IN Parameter #2 - START_IN Value: [0]
IN Parameter #3 - END_IN Value: [2]
Implementation - Test Implementation 3 SUCCESS 1
Operation Call SUCCESS 1
<RETURN> SUCCESS Expected: [cdefgh], Received: [cdefgh]
IN Parameter #1 - STRING_IN Value: [abcdefgh]
IN Parameter #2 - START_IN Value: [3]
IN Parameter #3 - END_IN Value: [100]
Implementation - Test Implementation 4 SUCCESS 1
Operation Call SUCCESS 1
<RETURN> SUCCESS Expected: [abcde], Received: [abcde]
IN Parameter #1 - STRING_IN Value: [abcdefgh]
IN Parameter #2 - START_IN Value: [null]
IN Parameter #3 - END_IN Value: [5]
Implementation - Test Implementation 5 ERROR 1 Test Implementation 5 failed: Expected: [cdefgh], Received: [null]
Operation Call ERROR 1 Test Implementation 5 failed: Expected: [cdefgh], Received: [null]
<RETURN> ERROR Expected: [cdefgh], Received: [null]
IN Parameter #1 - STRING_IN Value: [abcdefgh]
IN Parameter #2 - START_IN Value: [3]
IN Parameter #3 - END_IN Value: [null]
Implementation - Test Implementation 6 SUCCESS 1
Operation Call SUCCESS 1

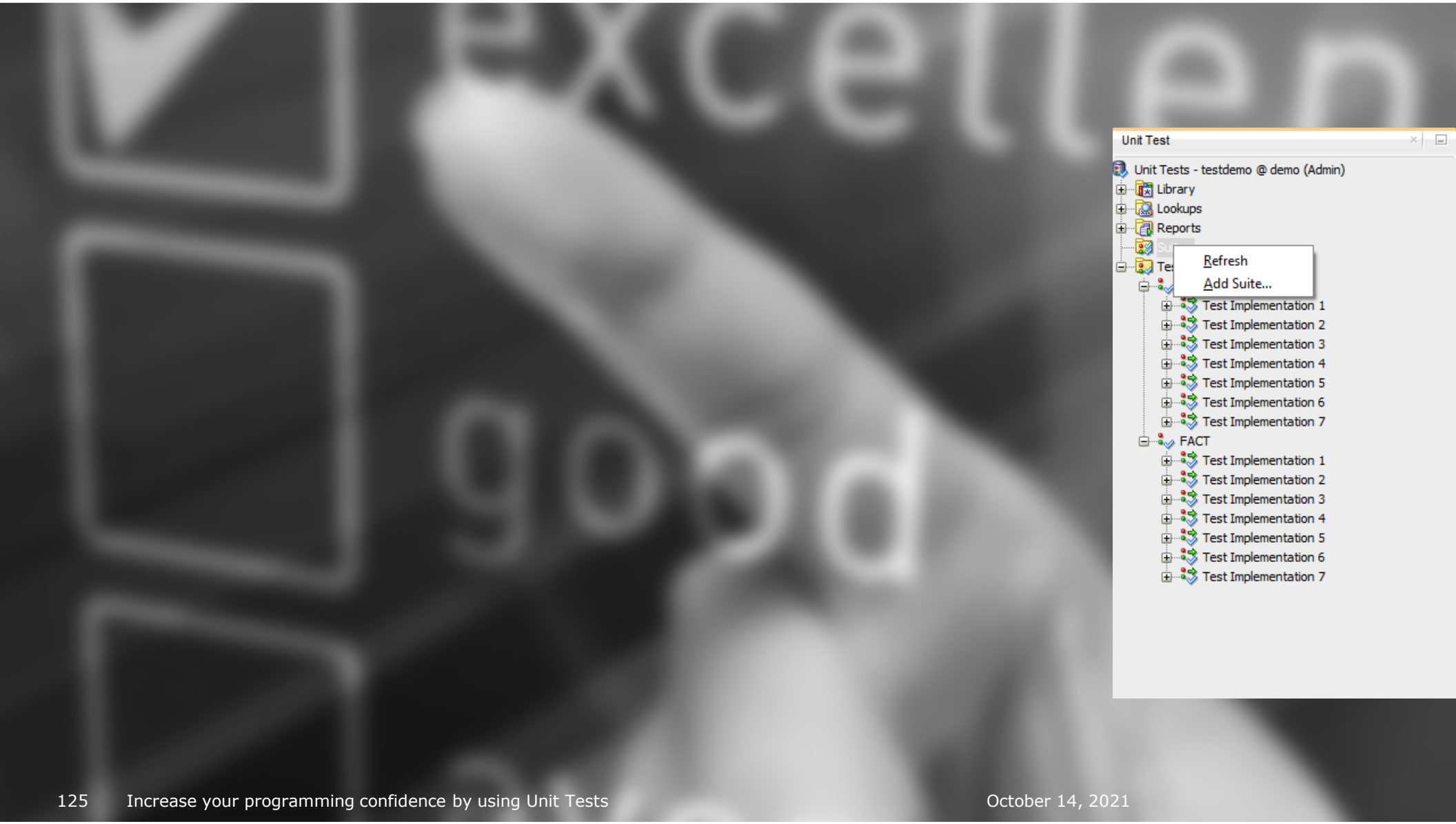

Unit Test
Unit Tests - testdemo @ demo (Admin)
Library
Lookups
Reports
Suites
Tests
BETWNSTR
Test Implementation 1
Test Implementation 2
Test Implementation 3
Test Implementation 4
Test Implementation 5
Test Implementation 6
Test Implementation 7

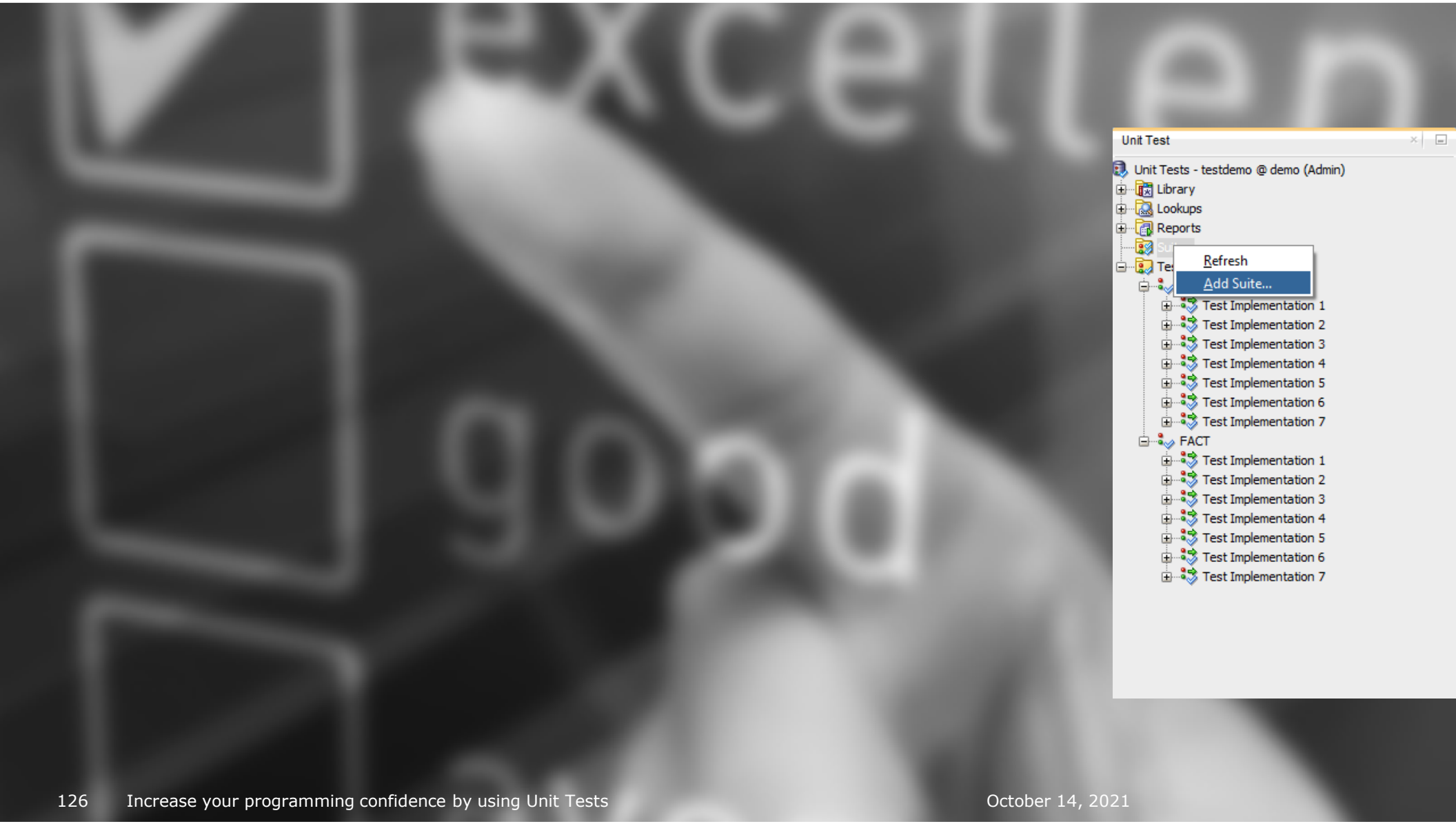

Details
Results
testdemo @ demo
Test Run
Status
Duration
Message
BETWNSTR
TESTDEMO: Run On - 2017-01-15 06:45:59.790756 SUCCESS 10
Implementation - Test Implementation 1 SUCCESS 4
Operation Call SUCCESS 4
<RETURN> SUCCESS Expected: [cde], Received: [cde]
IN Parameter #1 - STRING_IN Value: [abcdefgh]
IN Parameter #2 - START_IN Value: [3]
IN Parameter #3 - END_IN Value: [5]
Implementation - Test Implementation 2 SUCCESS 1
Operation Call SUCCESS 1
<RETURN> SUCCESS Expected: [ab], Received: [ab]
IN Parameter #1 - STRING_IN Value: [abcdefgh]
IN Parameter #2 - START_IN Value: [0]
IN Parameter #3 - END_IN Value: [2]
Implementation - Test Implementation 3 SUCCESS 1
Operation Call SUCCESS 1
<RETURN> SUCCESS Expected: [cdefgh], Received: [cdefgh]
IN Parameter #1 - STRING_IN Value: [abcdefgh]
IN Parameter #2 - START_IN Value: [3]
IN Parameter #3 - END_IN Value: [100]
Implementation - Test Implementation 4 SUCCESS 1
Operation Call SUCCESS 1
<RETURN> SUCCESS Expected: [abcde], Received: [abcde]
IN Parameter #1 - STRING_IN Value: [abcdefgh]
IN Parameter #2 - START_IN Value: [null]
IN Parameter #3 - END_IN Value: [5]
Implementation - Test Implementation 5 SUCCESS 1
Operation Call SUCCESS 1
<RETURN> SUCCESS Expected: [cdefgh], Received: [cdefgh]
IN Parameter #1 - STRING_IN Value: [abcdefgh]
IN Parameter #2 - START_IN Value: [3]
IN Parameter #3 - END_IN Value: [null]
Implementation - Test Implementation 6 SUCCESS 1
Operation Call SUCCESS 1
Unit Test
Unit Tests - testdemo @ demo (Admin)
Library
Lookups
Reports
Suites
Tests
BETWNSTR
Test Implementation 1
Test Implementation 2
Test Implementation 3
Test Implementation 4
Test Implementation 5
Test Implementation 6
Test Implementation 7

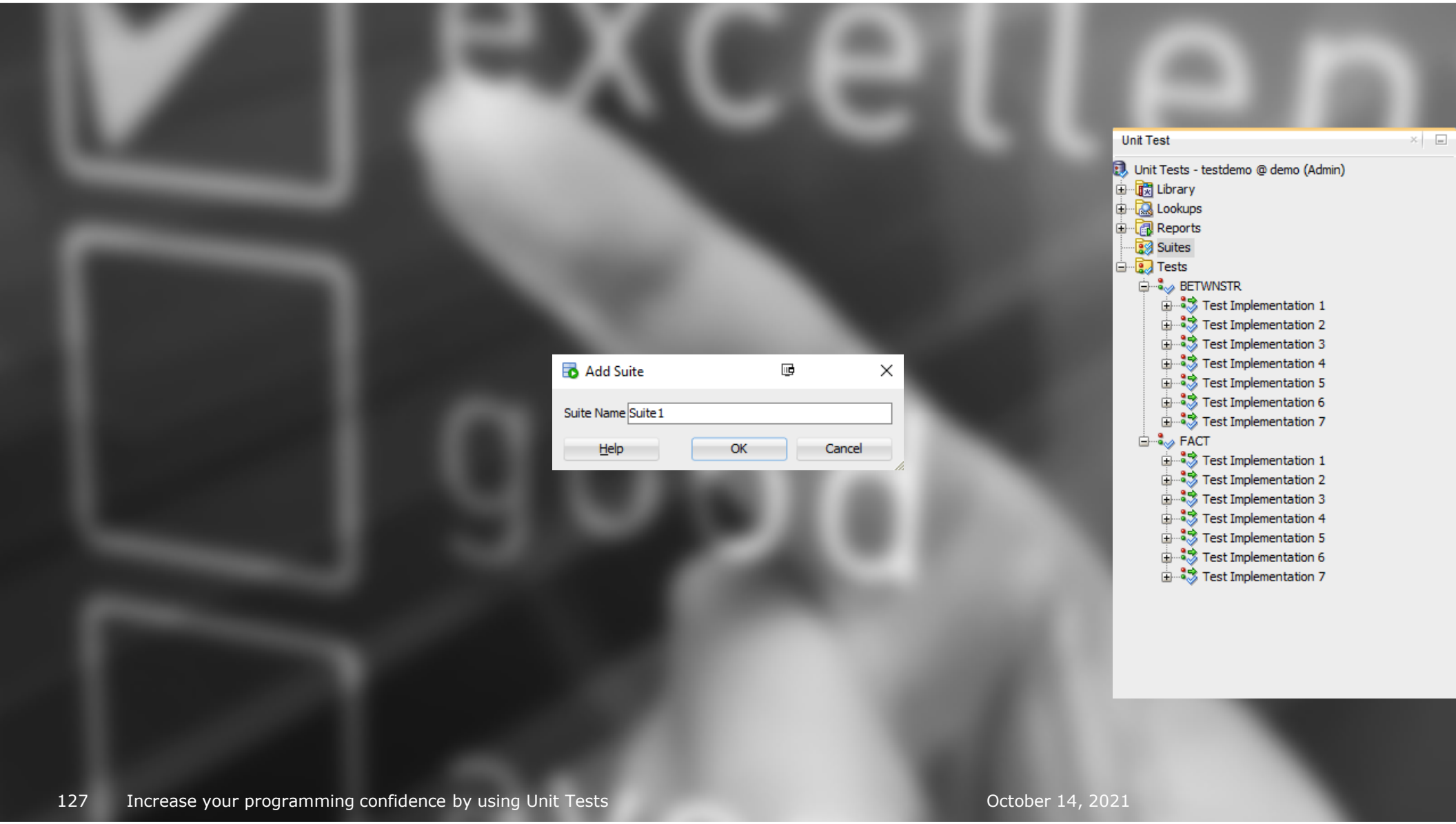

excellent
good
average

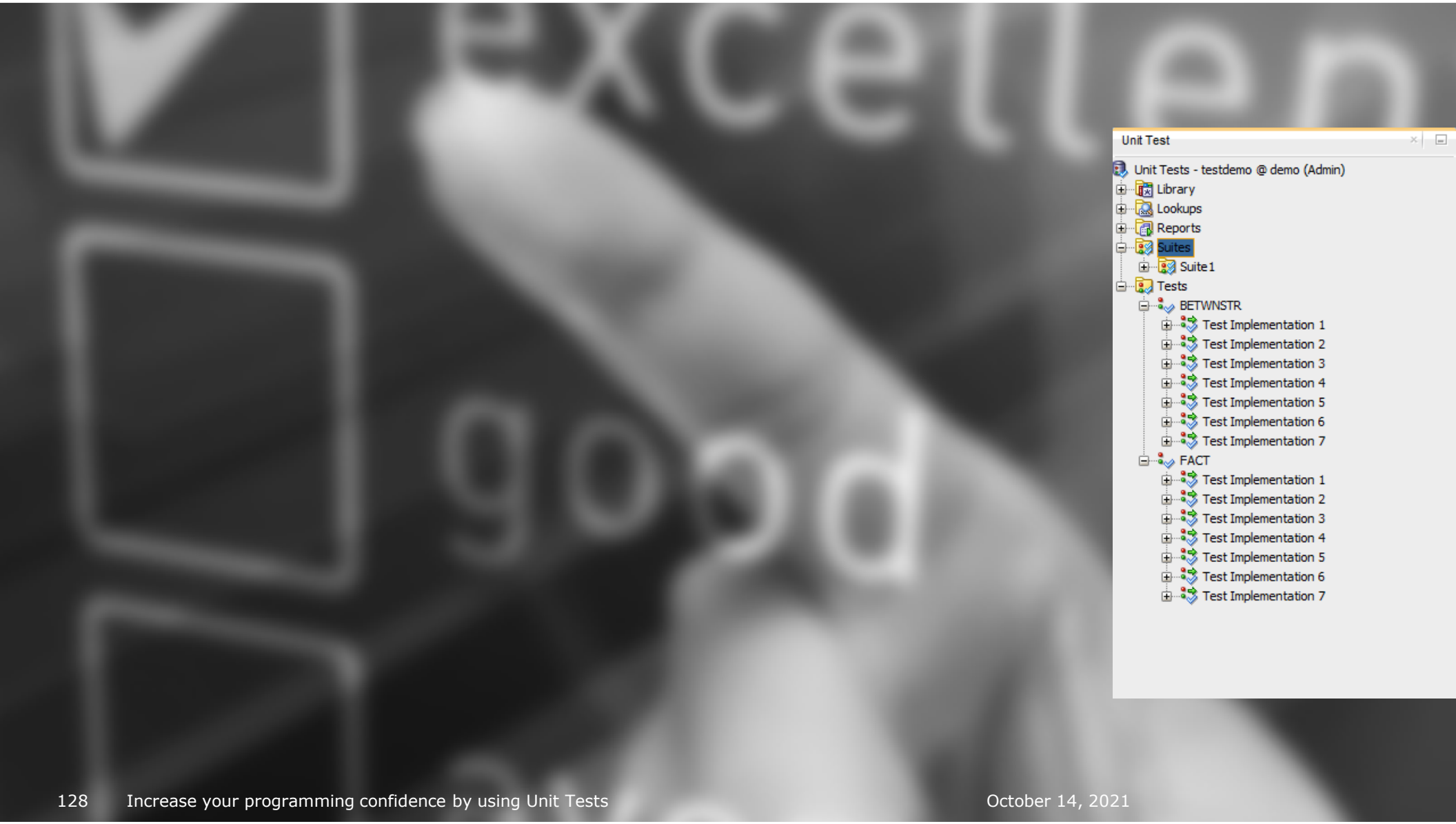

excellent
good
average

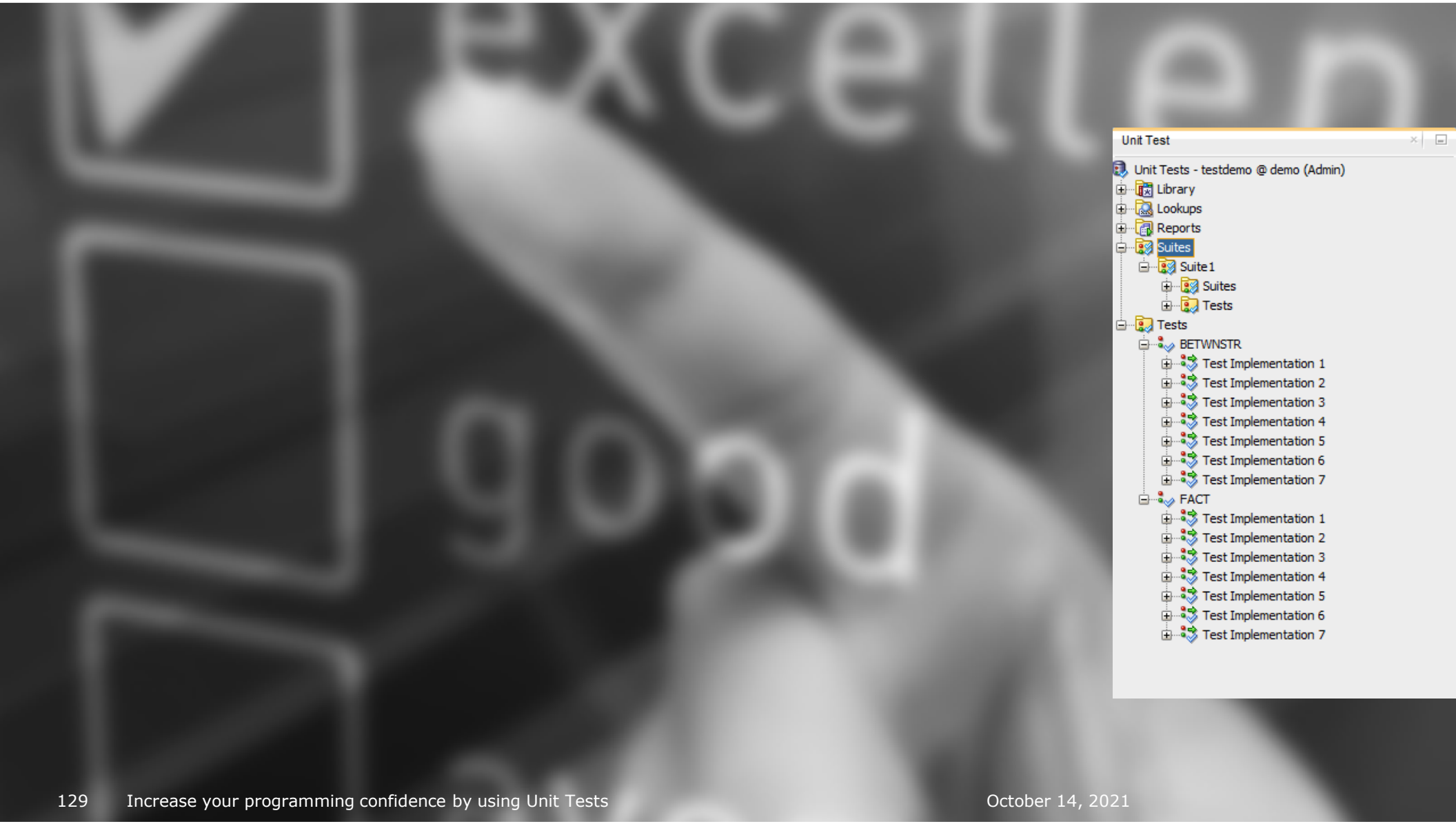

Unit Test
Unit Tests - testdemo @ demo (Admin)
Library
Lookups
Reports
Suites
Tests
BETWNSTR
Test Implementation 1
Test Implementation 2
Test Implementation 3
Test Implementation 4
Test Implementation 5
Test Implementation 6
Test Implementation 7
FACT
Test Implementation 1
Test Implementation 2
Test Implementation 3
Test Implementation 4
Test Implementation 5
Test Implementation 6
Test Implementation 7

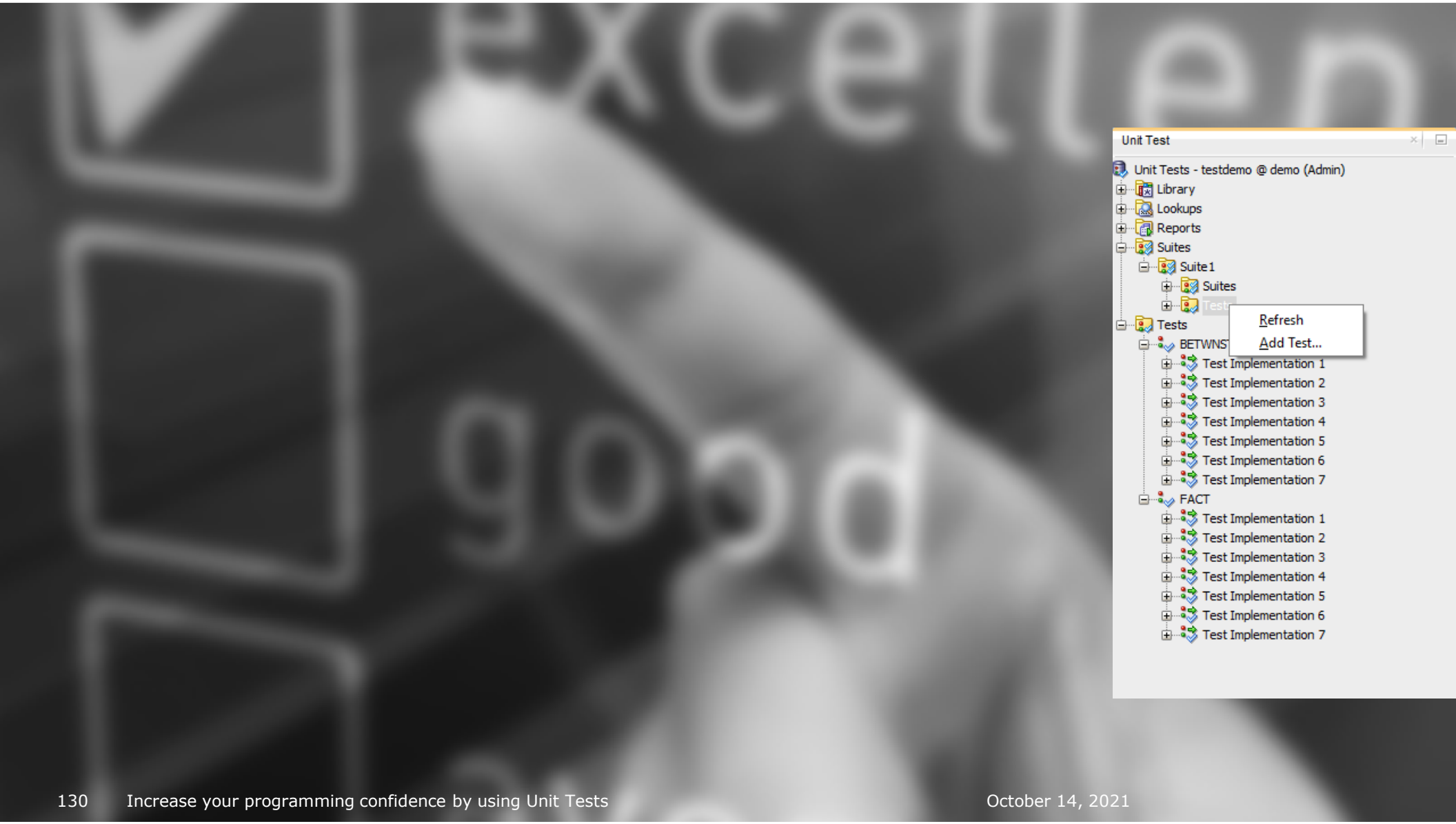

Unit Test
Unit Tests - testdemo @ demo (Admin)
Library
Lookups
Reports
Refresh
Add Suite...
Test Implementation 1
Test Implementation 2
Test Implementation 3
Test Implementation 4
Test Implementation 5
Test Implementation 6
Test Implementation 7
FACT
Test Implementation 1
Test Implementation 2
Test Implementation 3
Test Implementation 4
Test Implementation 5
Test Implementation 6
Test Implementation 7

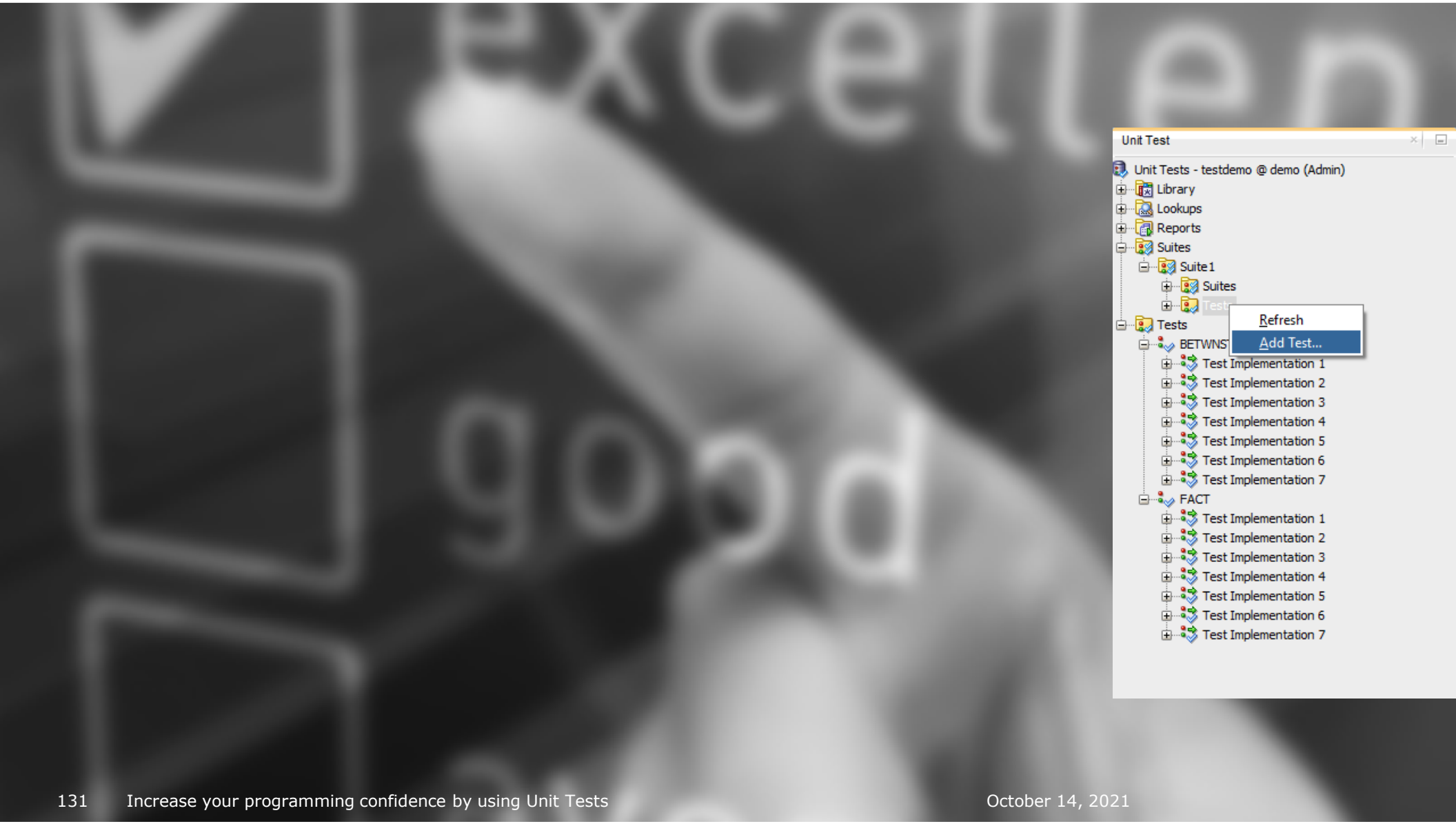

Unit Test
Unit Tests - testdemo @ demo (Admin)
Library
Lookups
Reports
Refresh
Add Suite...
Test Implementation 1
Test Implementation 2
Test Implementation 3
Test Implementation 4
Test Implementation 5
Test Implementation 6
Test Implementation 7
FACT
Test Implementation 1
Test Implementation 2
Test Implementation 3
Test Implementation 4
Test Implementation 5
Test Implementation 6
Test Implementation 7

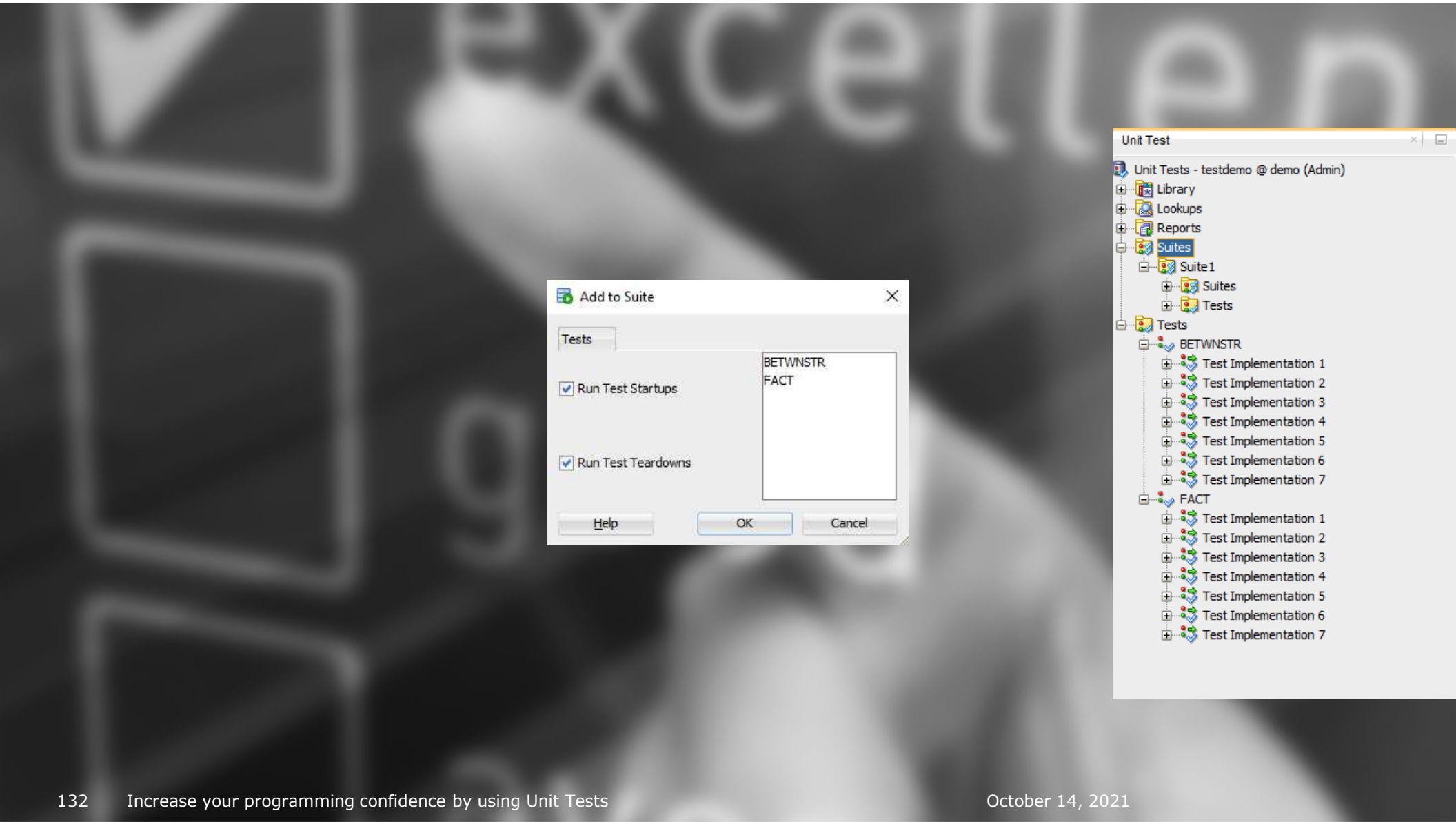

Add Suite
Suite Name Suite 1
Help
OK
Cancel
Unit Test
Unit Tests - testdemo @ demo (Admin)
Library
Lookups
Reports
Suites
Tests
BETWNSTR
Test Implementation 1
Test Implementation 2
Test Implementation 3
Test Implementation 4
Test Implementation 5
Test Implementation 6
Test Implementation 7
FACT
Test Implementation 1
Test Implementation 2
Test Implementation 3
Test Implementation 4
Test Implementation 5
Test Implementation 6
Test Implementation 7

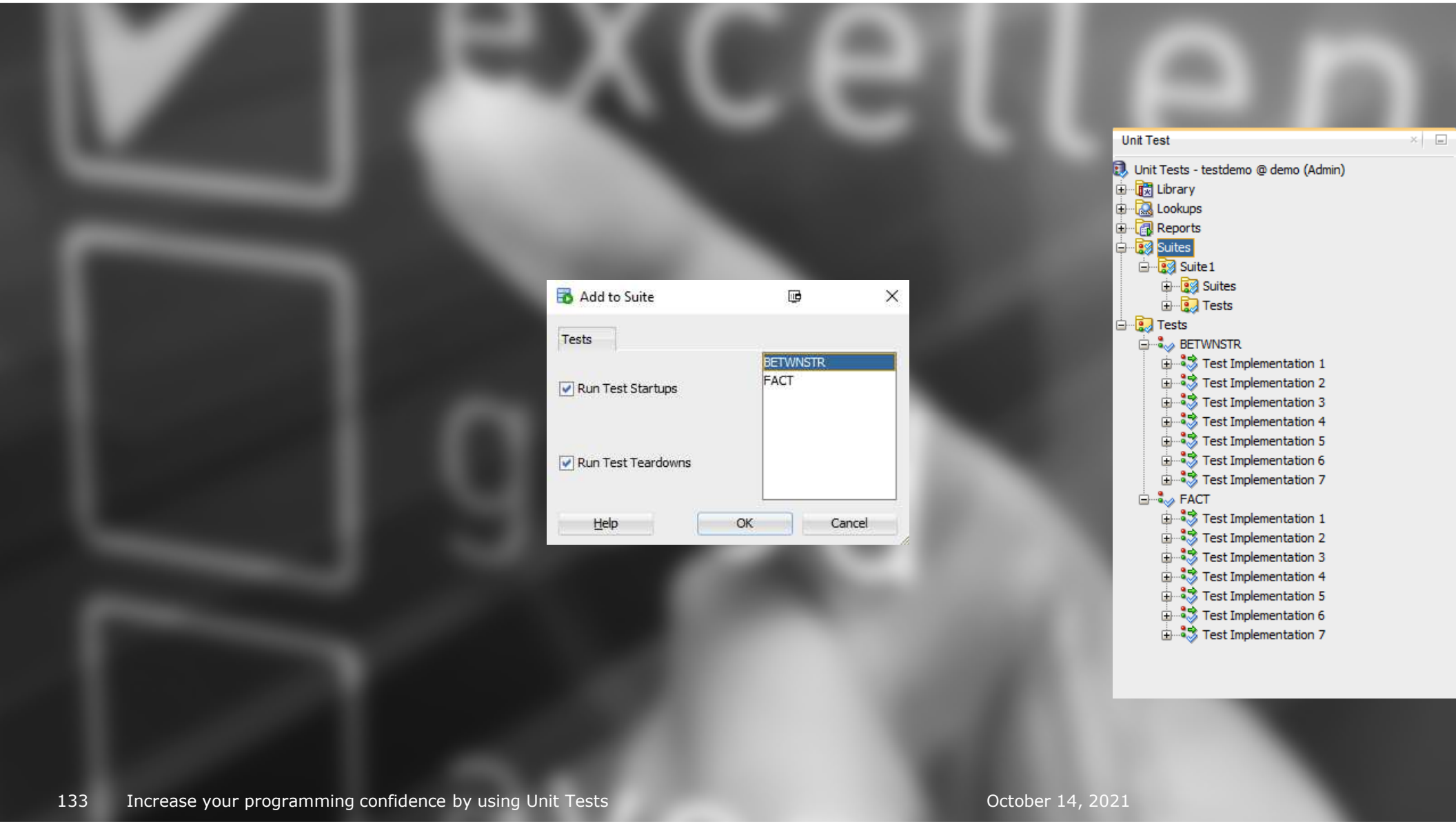

Unit Test
Unit Tests - testdemo @ demo (Admin)
Library
Lookups
Reports
Suites
Suite1
Tests
BETWNSTR
Test Implementation 1
Test Implementation 2
Test Implementation 3
Test Implementation 4
Test Implementation 5
Test Implementation 6
Test Implementation 7
FACT
Test Implementation 1
Test Implementation 2
Test Implementation 3
Test Implementation 4
Test Implementation 5
Test Implementation 6
Test Implementation 7

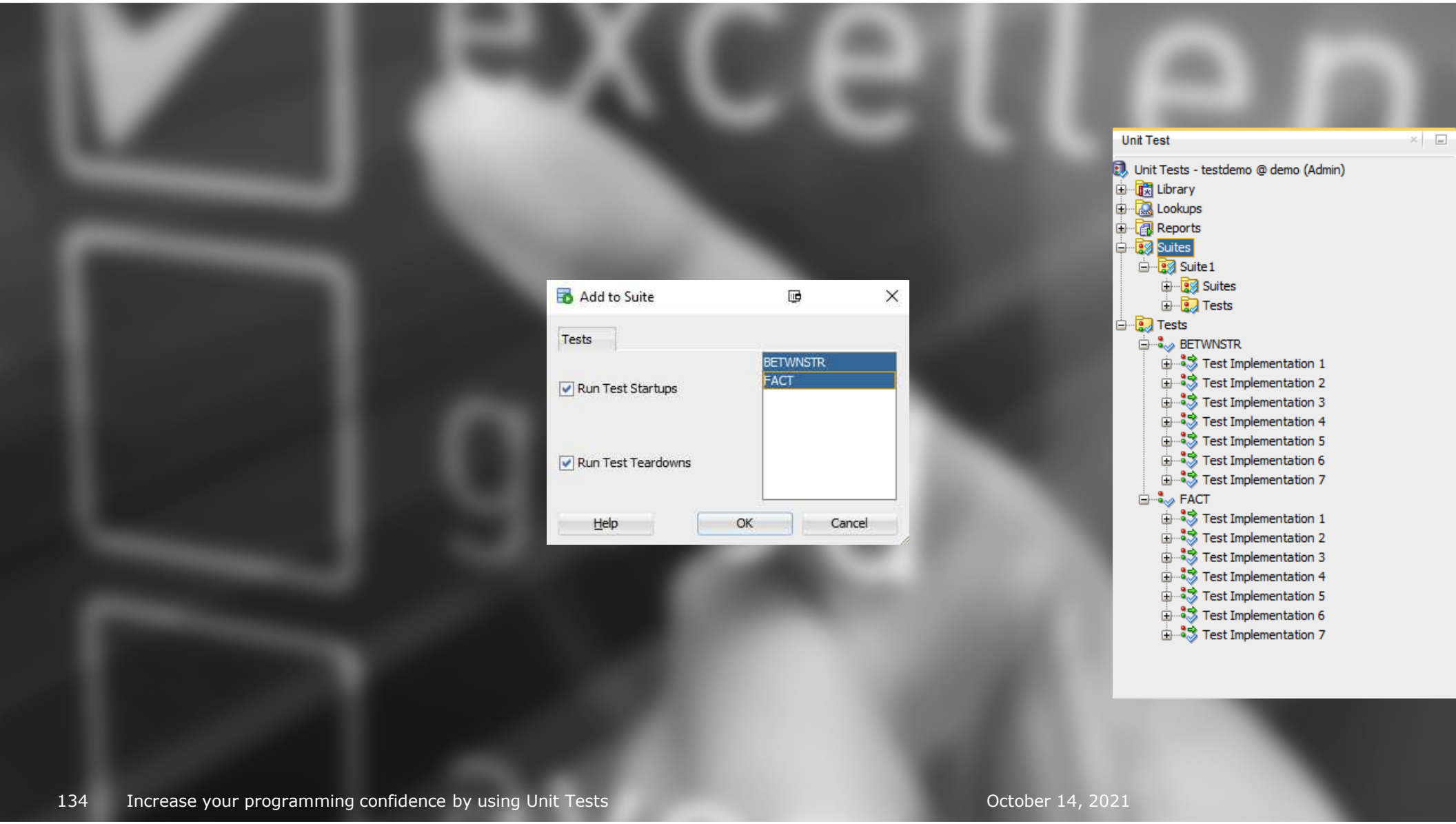

Unit Test
Unit Tests - testdemo @ demo (Admin)
Library
Lookups
Reports
Suites
Suite1
Suites
Tests
Tests
BETWNSTR
Test Implementation 1
Test Implementation 2
Test Implementation 3
Test Implementation 4
Test Implementation 5
Test Implementation 6
Test Implementation 7
FACT
Test Implementation 1
Test Implementation 2
Test Implementation 3
Test Implementation 4
Test Implementation 5
Test Implementation 6
Test Implementation 7

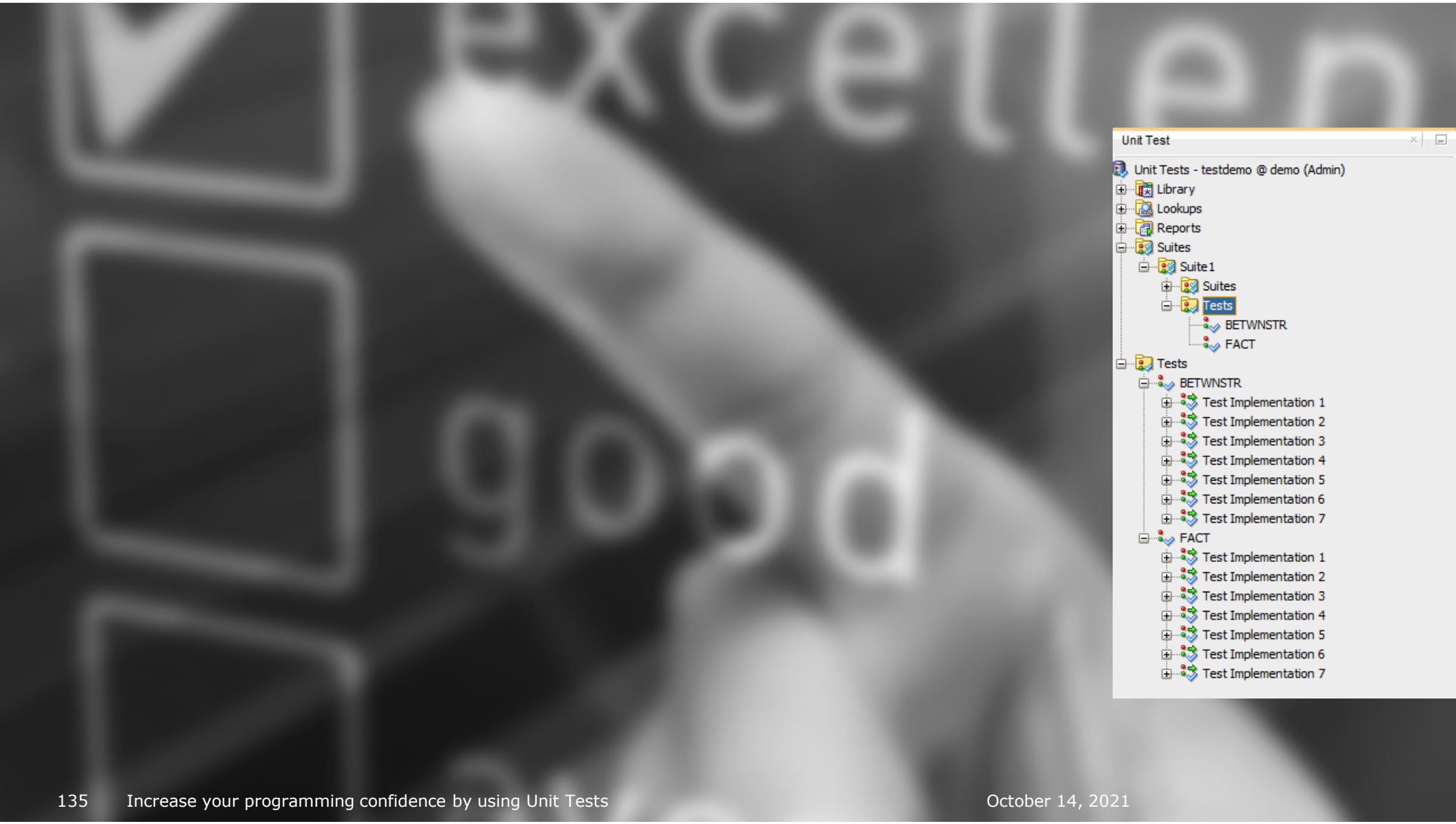

Unit Test
Unit Tests - testdemo @ demo (Admin)
Library
Lookups
Reports
Suites
Suite1
Suites
Tests
Refresh
Add Test...
BETWNST
Test Implementation 1
Test Implementation 2
Test Implementation 3
Test Implementation 4
Test Implementation 5
Test Implementation 6
Test Implementation 7
FACT
Test Implementation 1
Test Implementation 2
Test Implementation 3
Test Implementation 4
Test Implementation 5
Test Implementation 6
Test Implementation 7

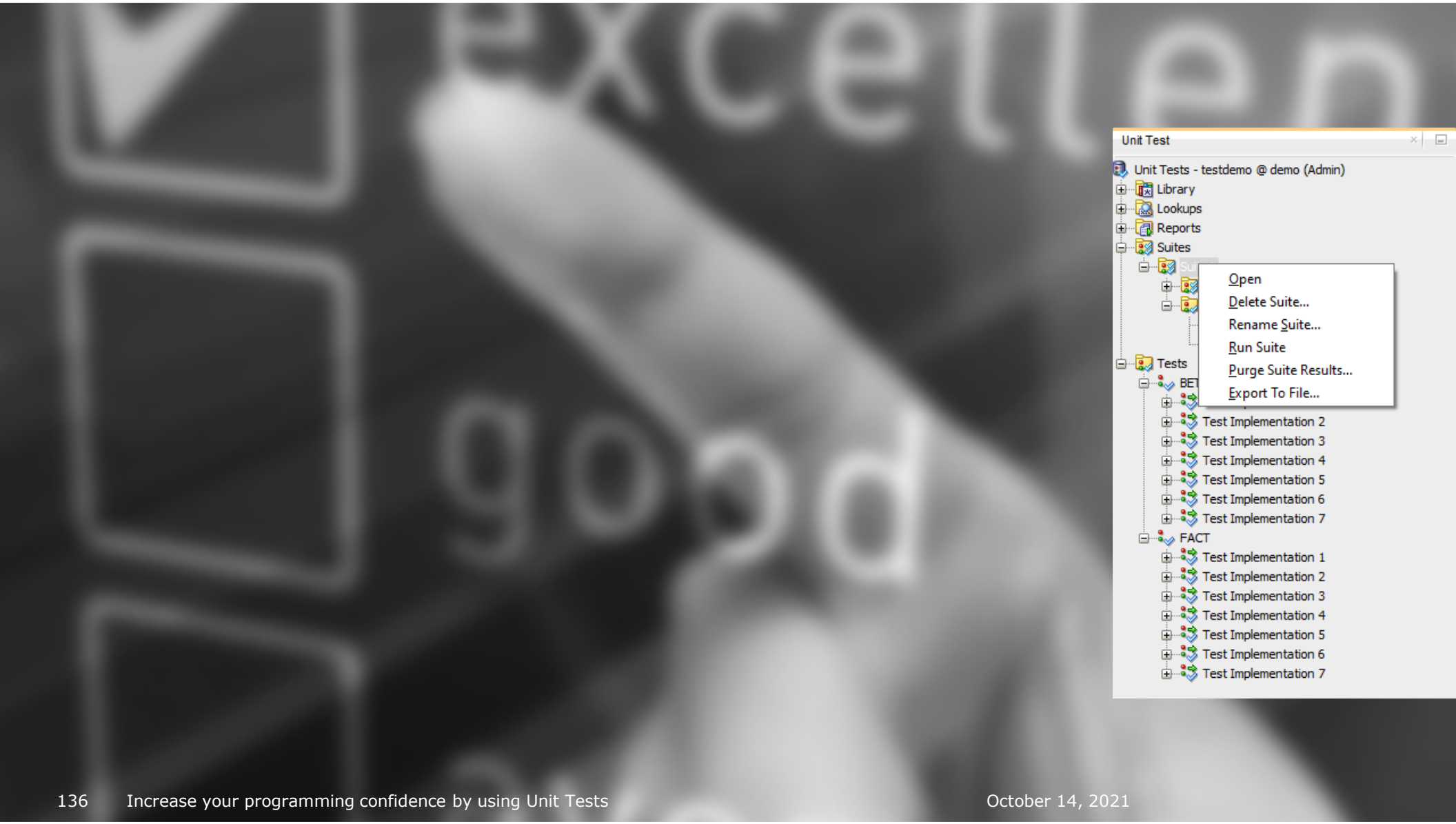

Unit Test
Unit Tests - testdemo @ demo (Admin)
Library
Lookups
Reports
Suites
Suite1
Suites
Tests
Refresh
Add Test...
BETWNST
Test Implementation 1
Test Implementation 2
Test Implementation 3
Test Implementation 4
Test Implementation 5
Test Implementation 6
Test Implementation 7
FACT
Test Implementation 1
Test Implementation 2
Test Implementation 3
Test Implementation 4
Test Implementation 5
Test Implementation 6
Test Implementation 7

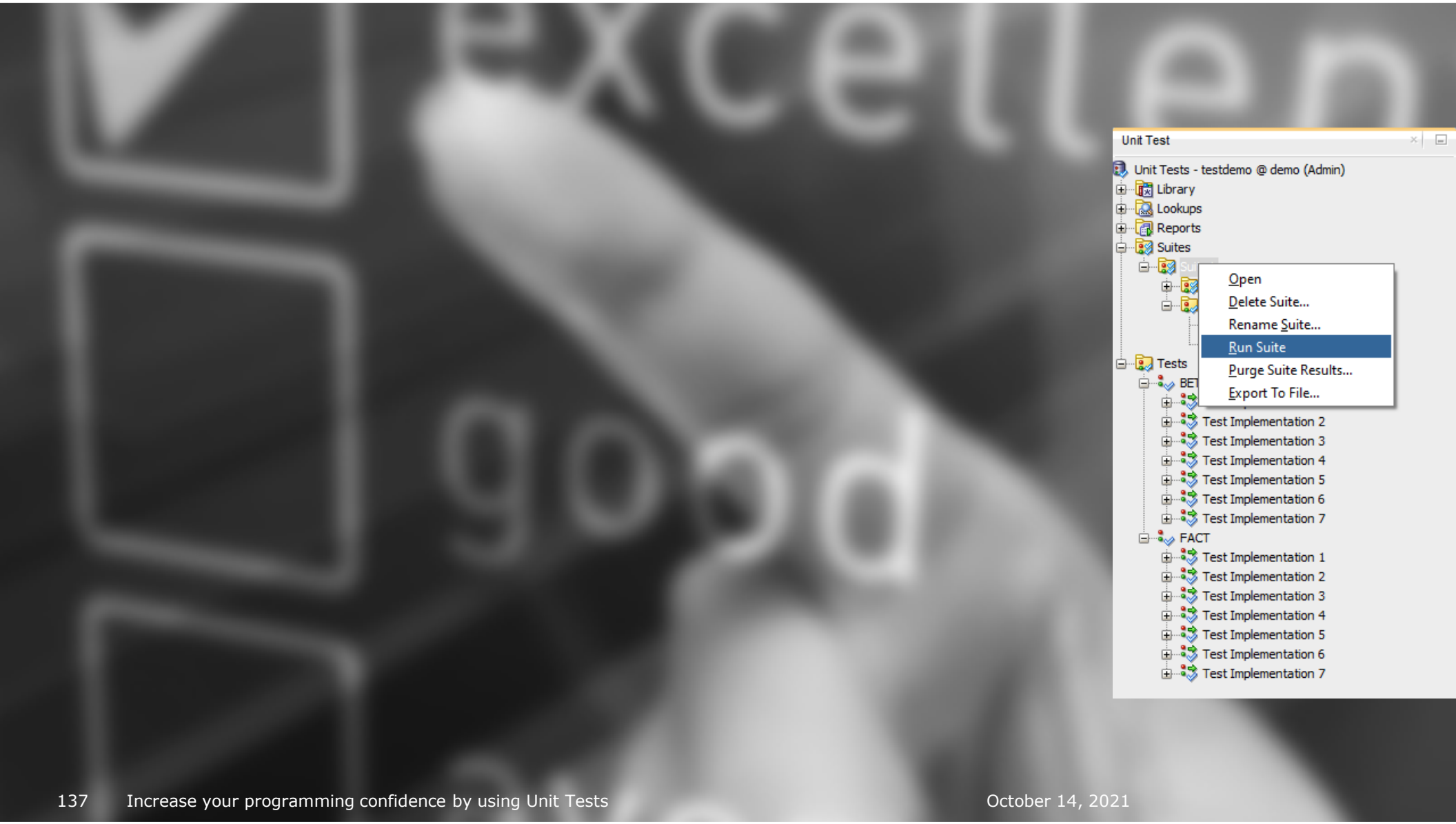

Add to Suite
Tests
Run Test Startups
Run Test Teardowns
BETWNSTR
FACT
Help
OK
Cancel
Unit Test
Unit Tests - testdemo @ demo (Admin)
Library
Lookups
Reports
Suites
Suite1
Suites
Tests
Tests
BETWNSTR
Test Implementation 1
Test Implementation 2
Test Implementation 3
Test Implementation 4
Test Implementation 5
Test Implementation 6
Test Implementation 7
FACT
Test Implementation 1
Test Implementation 2
Test Implementation 3
Test Implementation 4
Test Implementation 5
Test Implementation 6
Test Implementation 7

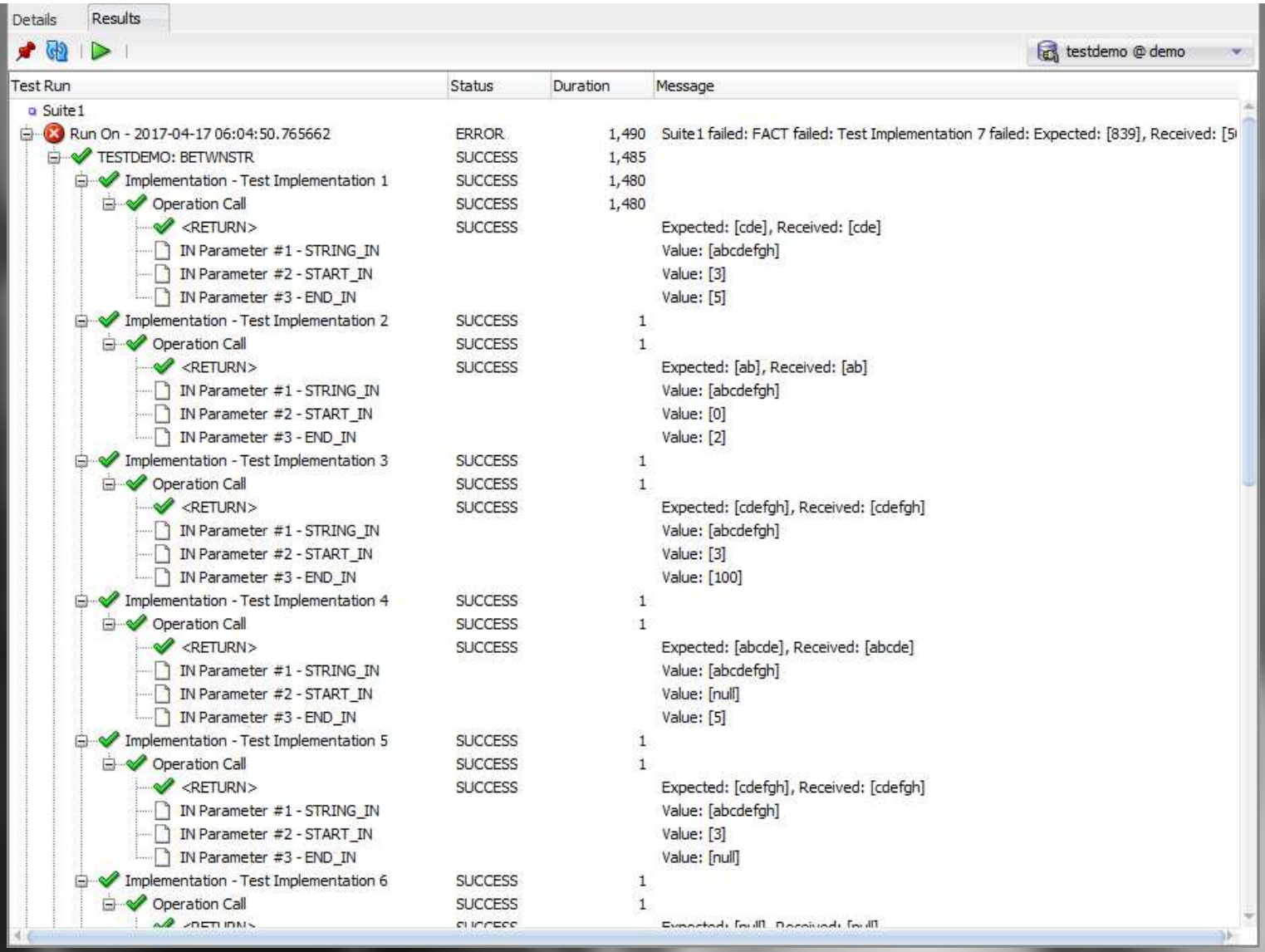

Add to Suite
Tests
Run Test Startups
Run Test Teardowns
BETWNSTR
FACT
Help
OK
Cancel
Unit Test
Unit Tests - testdemo @ demo (Admin)
Library
Lookups
Reports
Suites
Suite1
Suites
Tests
Tests
BETWNSTR
Test Implementation 1
Test Implementation 2
Test Implementation 3
Test Implementation 4
Test Implementation 5
Test Implementation 6
Test Implementation 7
FACT
Test Implementation 1
Test Implementation 2
Test Implementation 3
Test Implementation 4
Test Implementation 5
Test Implementation 6
Test Implementation 7

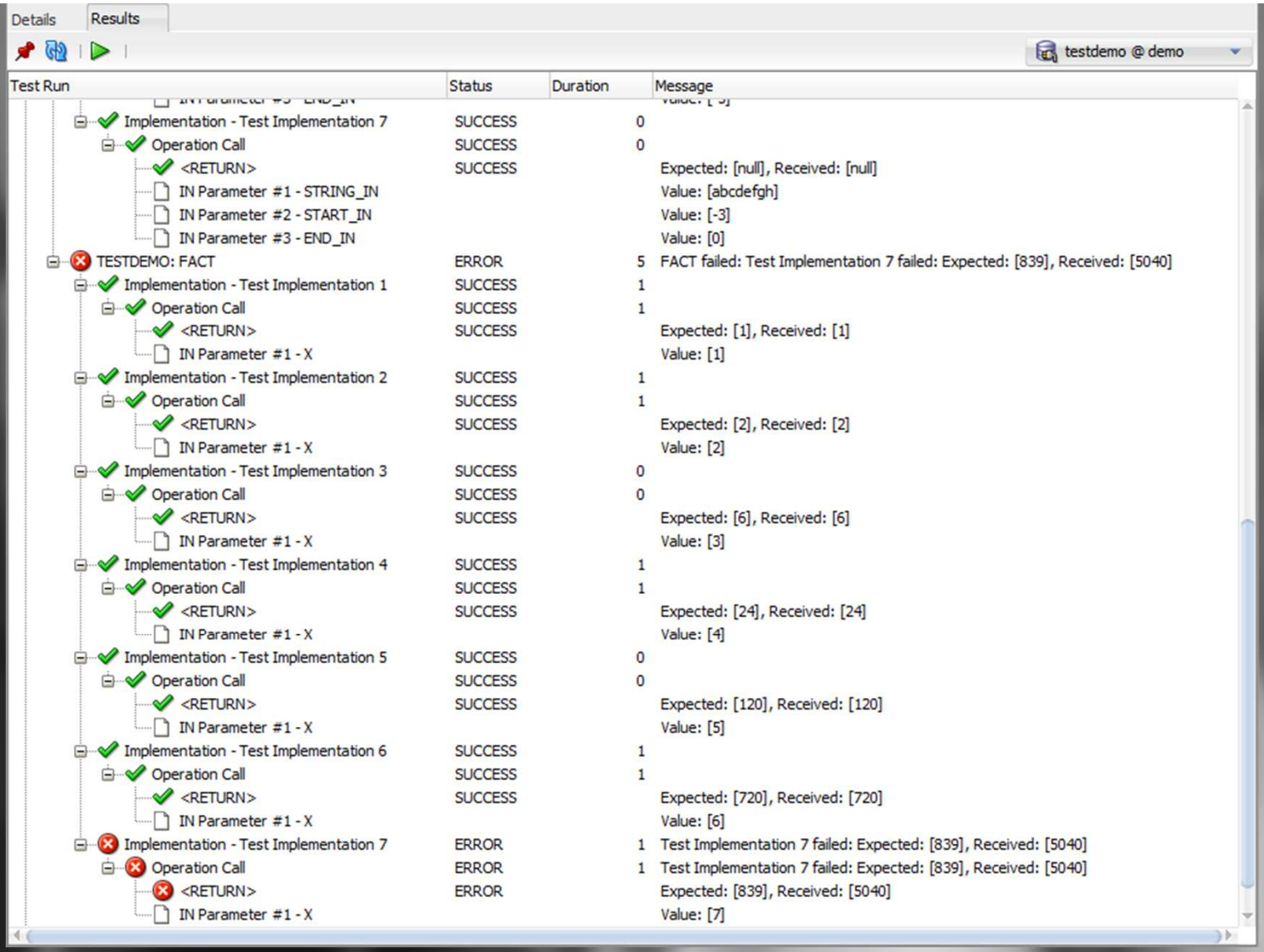

Add to Suite
Tests
Run Test Startups
Run Test Teardowns
BETWNSTR
FACT
Help
OK
Cancel
Unit Test
Unit Tests - testdemo @ demo (Admin)
Library
Lookups
Reports
Suites
Suite1
Suites
Tests
Tests
BETWNSTR
Test Implementation 1
Test Implementation 2
Test Implementation 3
Test Implementation 4
Test Implementation 5
Test Implementation 6
Test Implementation 7
FACT
Test Implementation 1
Test Implementation 2
Test Implementation 3
Test Implementation 4
Test Implementation 5
Test Implementation 6
Test Implementation 7

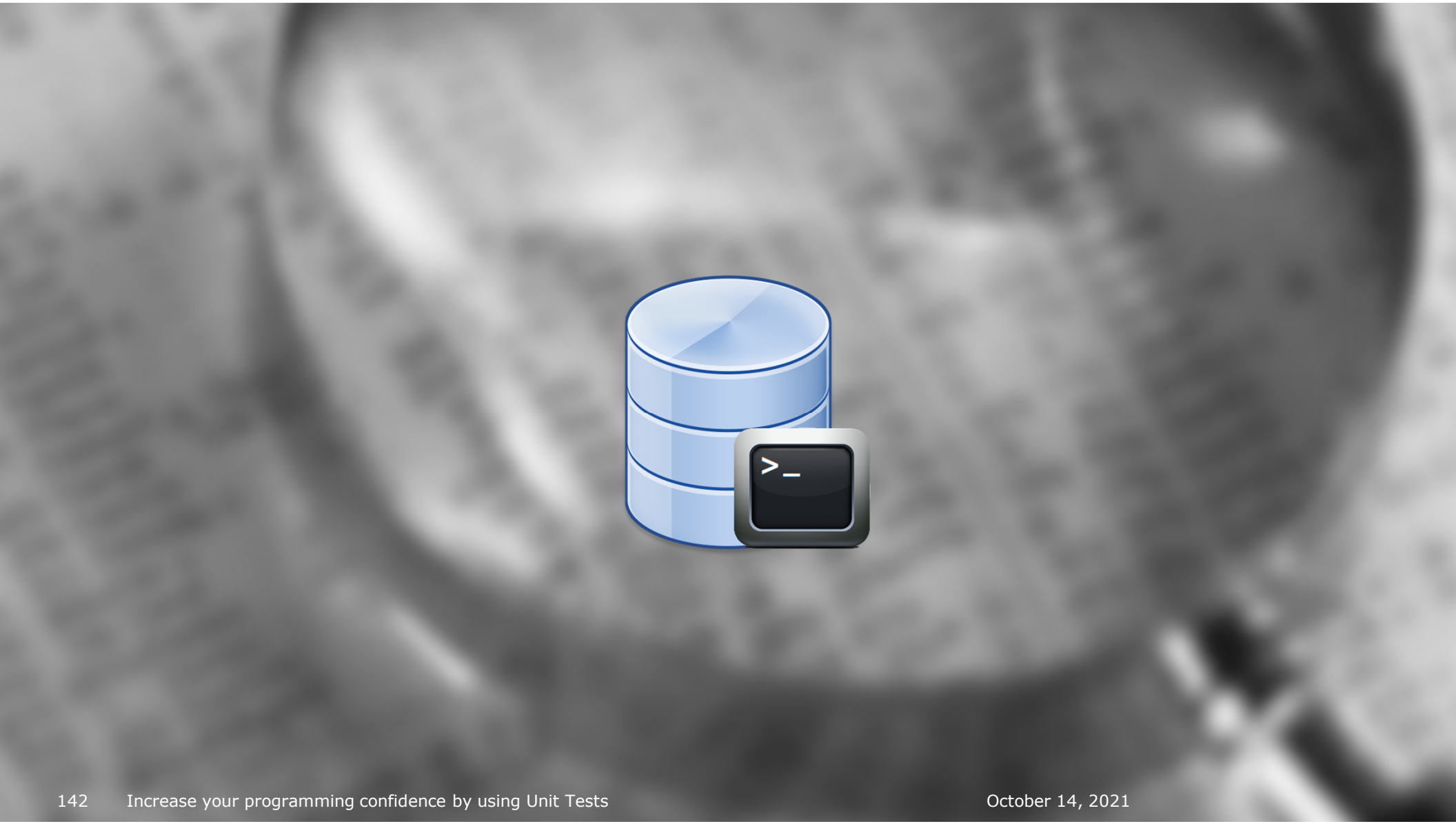

Unit Test
Unit Tests - testdemo @ demo (Admin)
Library
Lookups
Reports
Suites
Suite1
Suites
Tests
BETWNSTR
FACT
Tests
BETWNSTR
Test Implementation 1
Test Implementation 2
Test Implementation 3
Test Implementation 4
Test Implementation 5
Test Implementation 6
Test Implementation 7
FACT
Test Implementation 1
Test Implementation 2
Test Implementation 3
Test Implementation 4
Test Implementation 5
Test Implementation 6
Test Implementation 7


Unit Test
Unit Tests - testdemo @ demo (Admin)
Library
Lookups
Reports
Suites
Open
Delete Suite...
Rename Suite...
Run Suite
Purge Suite Results...
Export To File...
Tests
Test Implementation 2
Test Implementation 3
Test Implementation 4
Test Implementation 5
Test Implementation 6
Test Implementation 7
FACT
Test Implementation 1
Test Implementation 2
Test Implementation 3
Test Implementation 4
Test Implementation 5
Test Implementation 6
Test Implementation 7


Unit Test
Unit Tests - testdemo @ demo (Admin)
Library
Lookups
Reports
Suites
Open
Delete Suite...
Rename Suite...
Run Suite
Purge Suite Results...
Export To File...
Tests
Test Implementation 2
Test Implementation 3
Test Implementation 4
Test Implementation 5
Test Implementation 6
Test Implementation 7
FACT
Test Implementation 1
Test Implementation 2
Test Implementation 3
Test Implementation 4
Test Implementation 5
Test Implementation 6
Test Implementation 7


Details
Results
testdemo @ demo
Test Run
Status
Duration
Message
Suite1
Run On - 2017-04-17 06:04:50.765662 ERROR 1,490 Suite1 failed: FACT failed: Test Implementation 7 failed: Expected: [839], Received: [5
TESTDEMO: BETWNSTR SUCCESS 1,485
Implementation - Test Implementation 1 SUCCESS 1,480
Operation Call SUCCESS 1,480
<RETURN> SUCCESS Expected: [cde], Received: [cde]
IN Parameter #1 - STRING_IN Value: [abcdefgh]
IN Parameter #2 - START_IN Value: [3]
IN Parameter #3 - END_IN Value: [5]
Implementation - Test Implementation 2 SUCCESS 1
Operation Call SUCCESS 1
<RETURN> SUCCESS Expected: [ab], Received: [ab]
IN Parameter #1 - STRING_IN Value: [abcdefgh]
IN Parameter #2 - START_IN Value: [0]
IN Parameter #3 - END_IN Value: [2]
Implementation - Test Implementation 3 SUCCESS 1
Operation Call SUCCESS 1
<RETURN> SUCCESS Expected: [cdefgh], Received: [cdefgh]
IN Parameter #1 - STRING_IN Value: [abcdefgh]
IN Parameter #2 - START_IN Value: [3]
IN Parameter #3 - END_IN Value: [100]
Implementation - Test Implementation 4 SUCCESS 1
Operation Call SUCCESS 1
<RETURN> SUCCESS Expected: [abcde], Received: [abcde]
IN Parameter #1 - STRING_IN Value: [abcdefgh]
IN Parameter #2 - START_IN Value: [null]
IN Parameter #3 - END_IN Value: [5]
Implementation - Test Implementation 5 SUCCESS 1
Operation Call SUCCESS 1
<RETURN> SUCCESS Expected: [cdefgh], Received: [cdefgh]
IN Parameter #1 - STRING_IN Value: [abcdefgh]
IN Parameter #2 - START_IN Value: [3]
IN Parameter #3 - END_IN Value: [null]
Implementation - Test Implementation 6 SUCCESS 1
Operation Call SUCCESS 1
Unit Test
Unit Tests - testdemo @ demo (Admin)
Library
Lookups
Reports
Suites
Suite1
Suites
Tests
BETWNSTR
FACT
Tests
BETWNSTR
Test Implementation 1
Test Implementation 2
Test Implementation 3
Test Implementation 4
Test Implementation 5
Test Implementation 6
Test Implementation 7
FACT
Test Implementation 1
Test Implementation 2
Test Implementation 3
Test Implementation 4
Test Implementation 5
Test Implementation 6
Test Implementation 7


Details
Results
testdemo @ demo
Test Run
Status
Duration
Message
Implementation - Test Implementation 7 SUCCESS 0
Operation Call SUCCESS 0
<RETURN> SUCCESS Expected: [null], Received: [null]
IN Parameter #1 - STRING_IN Value: [abcdefgh]
IN Parameter #2 - START_IN Value: [-3]
IN Parameter #3 - END_IN Value: [0]
TESTDEMO: FACT ERROR 5 FACT failed: Test Implementation 7 failed: Expected: [839], Received: [5040]
Implementation - Test Implementation 1 SUCCESS 1
Operation Call SUCCESS 1
<RETURN> SUCCESS Expected: [1], Received: [1]
IN Parameter #1 - X Value: [1]
Implementation - Test Implementation 2 SUCCESS 1
Operation Call SUCCESS 1
<RETURN> SUCCESS Expected: [2], Received: [2]
IN Parameter #1 - X Value: [2]
Implementation - Test Implementation 3 SUCCESS 0
Operation Call SUCCESS 0
<RETURN> SUCCESS Expected: [6], Received: [6]
IN Parameter #1 - X Value: [3]
Implementation - Test Implementation 4 SUCCESS 1
Operation Call SUCCESS 1
<RETURN> SUCCESS Expected: [24], Received: [24]
IN Parameter #1 - X Value: [4]
Implementation - Test Implementation 5 SUCCESS 0
Operation Call SUCCESS 0
<RETURN> SUCCESS Expected: [120], Received: [120]
IN Parameter #1 - X Value: [5]
Implementation - Test Implementation 6 SUCCESS 1
Operation Call SUCCESS 1
<RETURN> SUCCESS Expected: [720], Received: [720]
IN Parameter #1 - X Value: [6]
Implementation - Test Implementation 7 ERROR 1 Test Implementation 7 failed: Expected: [839], Received: [5040]
Operation Call ERROR 1 Test Implementation 7 failed: Expected: [839], Received: [5040]
<RETURN> ERROR Expected: [839], Received: [5040]
IN Parameter #1 - X Value: [7]
Unit Test
Unit Tests - testdemo @ demo (Admin)
Library
Lookups
Reports
Suites
Suite1
Suites
Tests
BETWNSTR
FACT
Tests
BETWNSTR
Test Implementation 1
Test Implementation 2
Test Implementation 3
Test Implementation 4
Test Implementation 5
Test Implementation 6
Test Implementation 7
FACT
Test Implementation 1
Test Implementation 2
Test Implementation 3
Test Implementation 4
Test Implementation 5
Test Implementation 6
Test Implementation 7



```
> C:\oracle\sqldeveloper\sqldeveloper\bin\sdcli

Available features:
cart: Database Cart Batch Tasks
dba: Basic Batch DBA Tasks
format: Utility Import Task
migration: Database Migration Tasks
reports: Basic Batch Reporting Tasks
unittest: Unit Testing Batch Tasks
utility: Utility Import Task
```

```
> C:\oracle\sqldeveloper\sqldeveloper\bin\sdcli unittest

unittest ?
unittest -run ?
unittest -exp ?
unittest -imp ?

Command Completed.
```

```
> C:\oracle\sqldeveloper\sqldeveloper\bin\sdcli unittest -run ?

unittest -run -test (-id <id>|-name <name>} -repo <connection name>
         -db <connection name> {-return <return id>} {-log <0,1,2,3>}
unittest -run -suite (-id <id>|-name <name>} -repo <connection name>
         -db <connection name> {-return <return id>} {-log <0,1,2,3>}
```

```
> C:\oracle\sqldeveloper\sqldeveloper\bin\sdcli unittest -run -test -name "BETWNSTR"
 -repo "testdemo @ demo" -db "testdemo @ demo"
Command Completed.
```

```
> C:\oracle\sqldeveloper\sqldeveloper\bin\sdcli unittest -run -test -name "BETWNSTR"
 -repo "testdemo @ demo" -db "testdemo @ demo" -log 3
a23124be-4f86-4d67-9219-edaabc8cc3b3
UT_SUCCESS
null
Command Completed.
```

```
> C:\oracle\sqldeveloper\sqldeveloper\bin\sdcli unittest -run -test -name "BETWNSTR"
 -repo "testdemo @ demo" -db "testdemo @ demo" -log 3
37212a19-2962-43bc-92fd-103421edf979
UT_ERROR
BETWNSTR failed: Test Implementation 7 failed: Expected: [null], Received: [fgh]
Command Completed.
```

```
> C:\oracle\sqldeveloper\sqldeveloper\bin\sdcli unittest -run -test -name "BETWNSTR"
 -repo "testdemo @ demo" -db "testdemo @ demo" -log 3 -return 1
1
UT_SUCCESS
null
Command Completed.
```

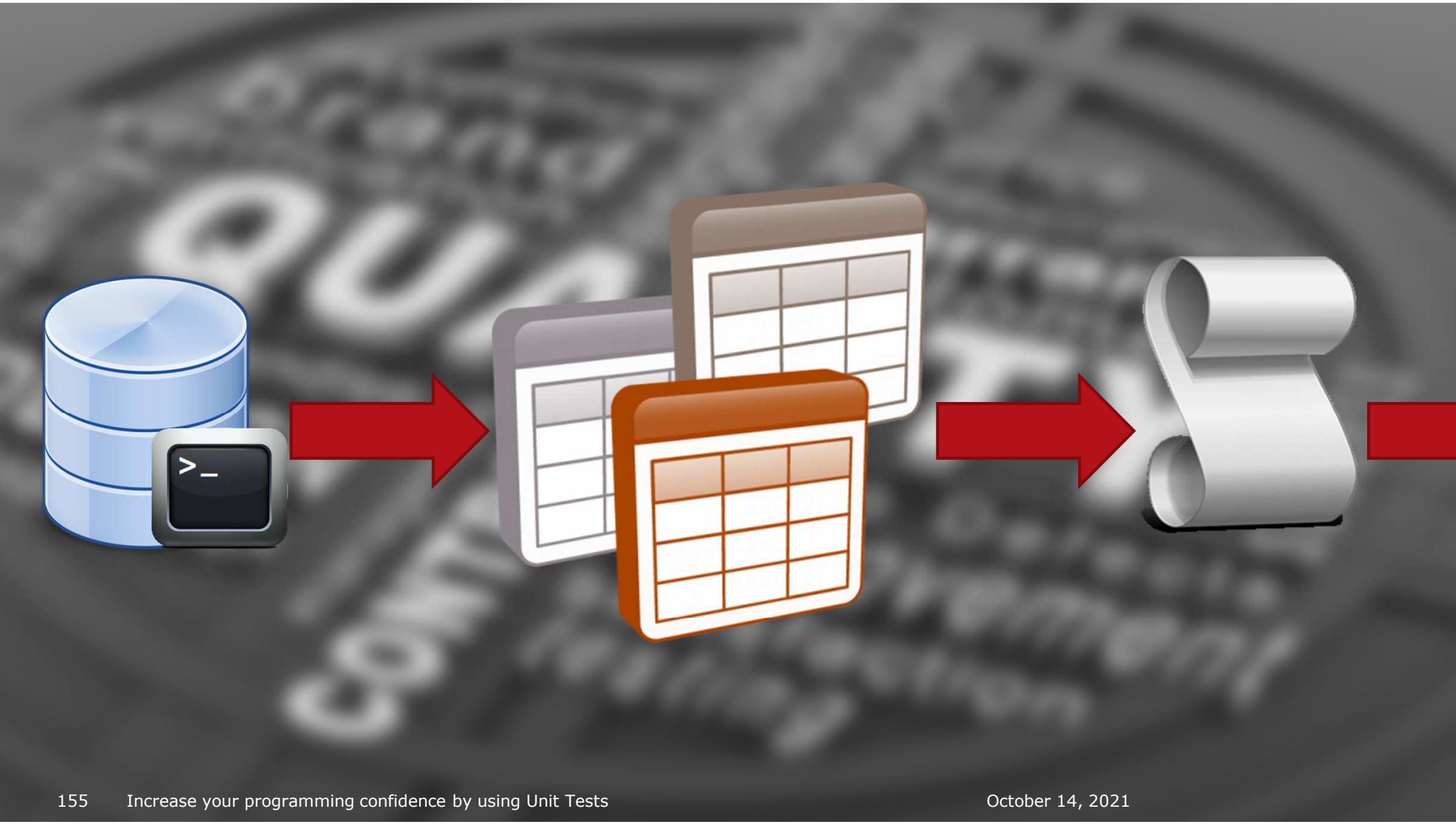
```
> sqlplus testdemo/testdemo@demo

SQL*Plus: Release 12.1.0.1.0 Production on Sun Jan 22 07:56:33 2017

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

Last Successful login time: Sun Jan 22 2017 07:43:31 +01:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options

TESTDEMO@demo> create sequence unittest_seq start with 2 nocache
  2  /

Sequence created.

TESTDEMO@demo>
```

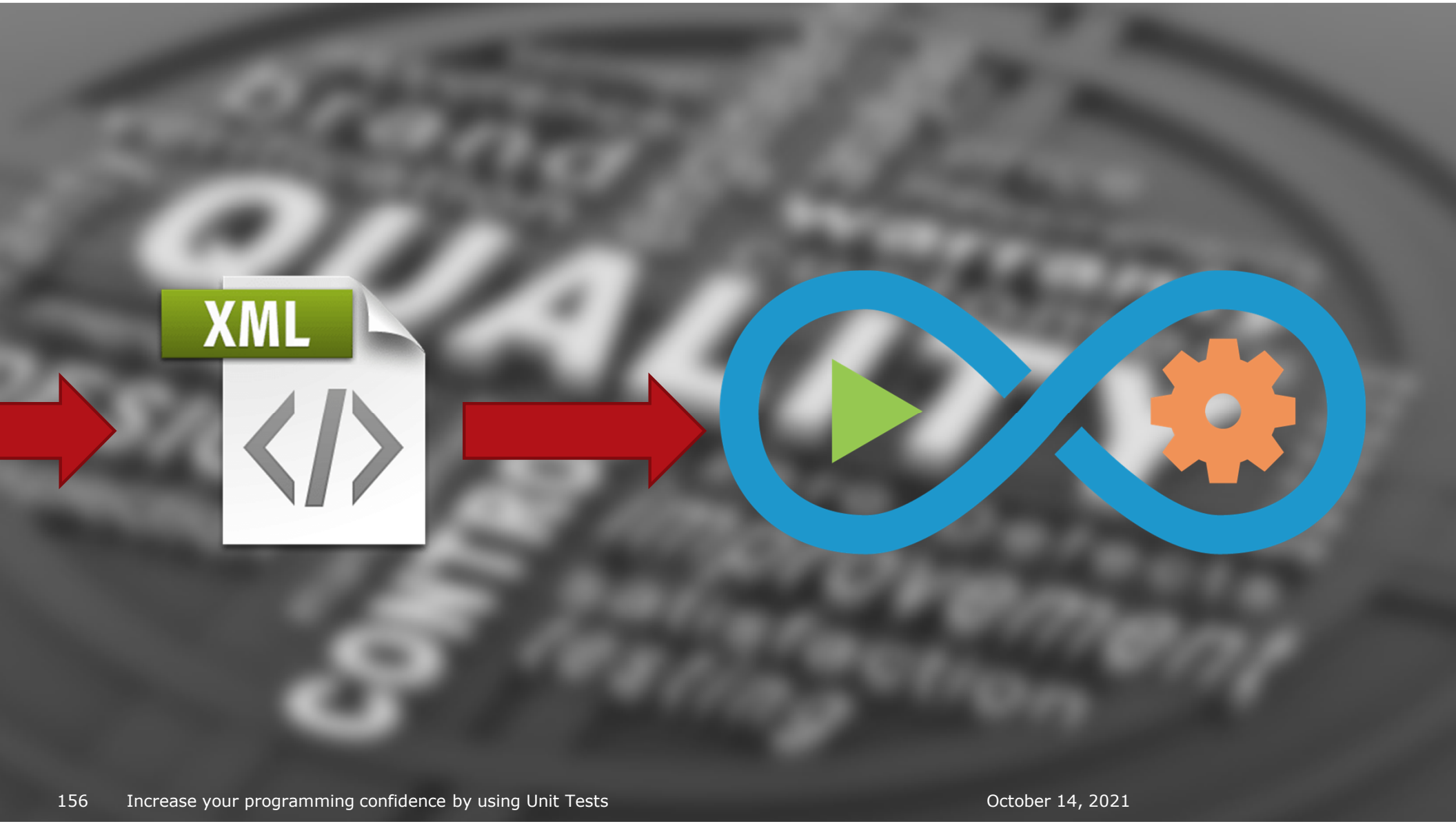
```
@echo off
ECHO Retrieve the next UnitTestID
for /f %%i in ('sqlplus -s testdemo/testdemo@demo @getSequenceNextVal.sql') do @set
UnitTestID=%%i
ECHO Run the unittest using this ID
C:\oracle\sqldeveloper\sqldeveloper\bin\sdcli unittest -run -test -name "BETWNSTR" -
repo "testdemo @ demo" -db "testdemo @ demo" -log 3 -return %UnitTestID%
```

```
@echo off
ECHO Retrieve the next UnitTestID
for /f %%i in ('sqlplus -s testdemo/testdemo@demo @getSequenceNextVal.sql') do @set
UnitTestID=%%i
ECHO Run the unittest using this ID
C:\oracle\sqldeveloper\sqldeveloper\bin\sdcli unittest -run -test -name "BETWNSTR" -
repo "testdemo @ demo" -db "testdemo @ demo" -log 3 -return %UnitTestID%

> test

Retrieve the next UnitTestID
Run the unittest using this ID
2
UT_SUCCESS
null
Command Completed.
```


QUALITY
brand
assurance
optimal
verification
engineering
analysis
price
Requirements
warranty
customers
guaranty
merchandise
DESIGN
inspection
management
CONTROL
zero Defects
improvement
success
satisfaction
testing


QUALITY
DESIGN
CONTROL
brand
assurance
optimal
verification
guaranty
engineering
analysis
price
Requirements
warranty
customers
merchandise
inspection
management
zero Defects
improvement
satisfaction
testing
success




XML

```
> C:\oracle\sqldeveloper\sqldeveloper\bin\sdcli unittest -run ?

unittest -run -test (-id <id>|-name <name>} -repo <connection name>
         -db <connection name> {-return <return id>} {-log <0,1,2,3>}
unittest -run -suite (-id <id>|-name <name>} -repo <connection name>
         -db <connection name> {-return <return id>} {-log <0,1,2,3>}
```

```
@echo off
ECHO Retrieve the next UnitTestID
for /f %%i in ('sqlplus -s testdemo/testdemo@demo @getSequenceNextVal.sql') do @set
UnitTestID=%%i
ECHO Run the testsuite using this ID
C:\oracle\sqldeveloper\sqldeveloper\bin\sdcli unittest -run -suite -name "Suite1" -repo
"testdemo @ demo" -db "testdemo @ demo" -log 3 -return %UnitTestID%
ECHO Create the XML file with the results
sqlplus -s -l testdemo/testdemo@demo @ut##dumpxml.sql %UnitTestID%
```

```
@echo off
ECHO Retrieve the next UnitTestID
for /f %%i in ('sqlplus -s testdemo/testdemo@demo @getSequenceNextVal.sql') do @set
UnitTestID=%%i
ECHO Run the testsuite using this ID
C:\oracle\sqldeveloper\sqldeveloper\bin\sdcli unittest -run -suite -name "Suite1" -repo
"testdemo @ demo" -db "testdemo @ demo" -log 3 -return %UnitTestID%
ECHO Create the XML file with the results
sqlplus -s -l testdemo/testdemo@demo @ut##dumpxml.sql %UnitTestID%

> suite

Retrieve the next UnitTestID
Run the testsuite using this ID
7
UT_ERROR
Suite1 failed: FACT failed: Test Implementation 7 failed: Expected: [839], Received:
[5040]
Command Completed.
Create the XML file with the results
```


QA and Testing

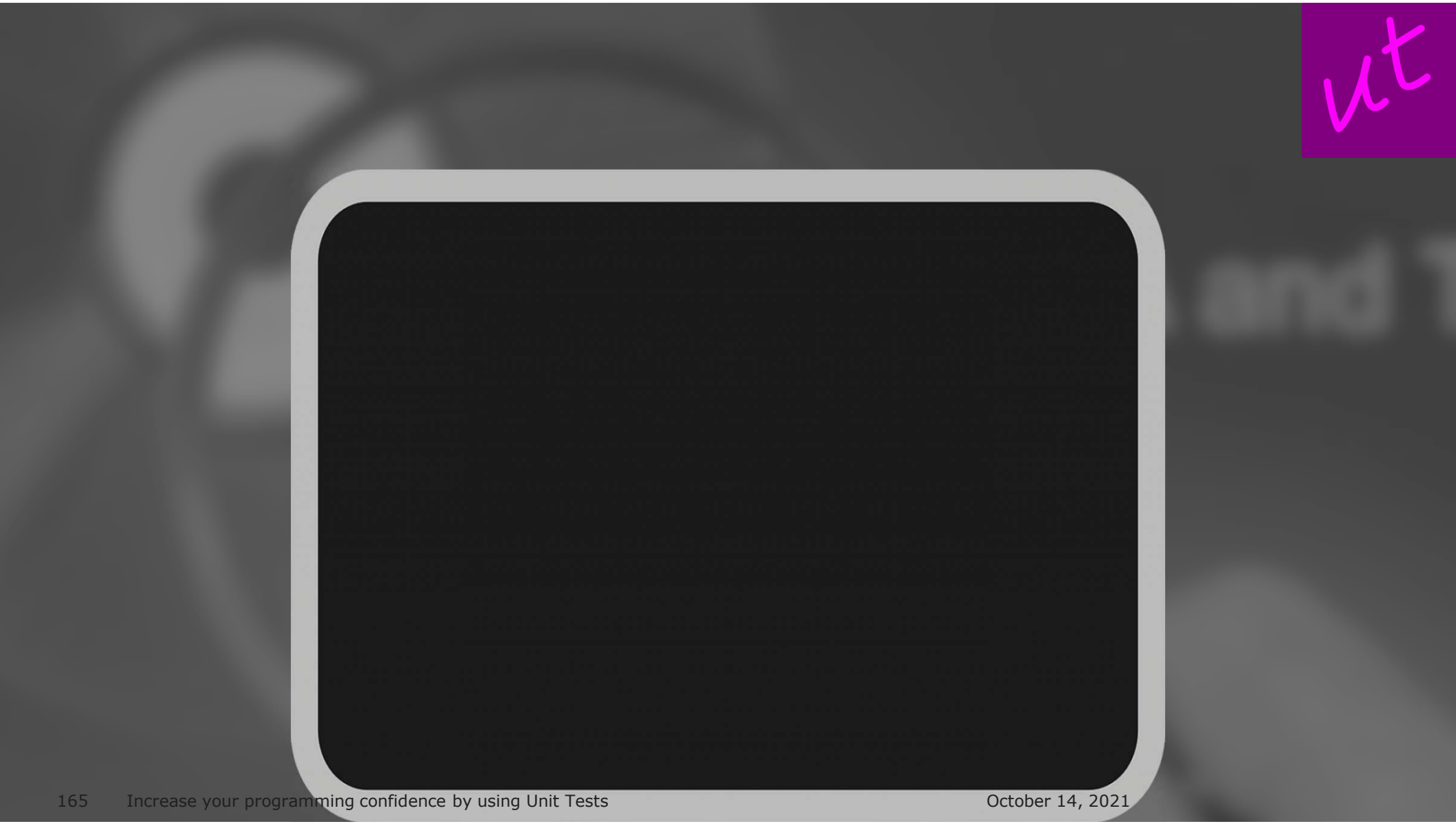

QA and Testing


ut


ut

utPL/SQL v3

- Completely new version
- Implementation is different from utPL/SQL
  - Test converter available
- Uses annotation
  - Still have to write your own code, but less


ut

```
create or replace package test_betwnstr is
  -- %suite (betwnstr)
  -- Purpose : Unittesting betwnstr with utPL/SQL 3

  -- %test(BETWNSTR Tests)
  procedure test_betwnstr;
end test_betwnstr;
/
create or replace package body test_betwnstr is
  c_teststring constant varchar2(8) := 'abcdefgh';
  procedure test_betwnstr is
  begin
    ut.expect(betwnstr(string_in => c_teststring, start_in => 3, end_in => 5))
       .to_equal('cde');
    ut.expect(betwnstr(string_in => c_teststring, start_in => 0, end_in => 2))
       .to_equal('ab');
    ut.expect(betwnstr(string_in => c_teststring, start_in => 3, end_in => 9999))
       .to_equal('cdefgh');
    ut.expect(betwnstr(string_in => c_teststring, start_in => null, end_in => 5))
       .to_equal('abcde');
    ut.expect(betwnstr(string_in => c_teststring, start_in => 3, end_in => null))
       .to_equal('cdefgh');
    ut.expect(betwnstr(string_in => c_teststring, start_in => -3, end_in => -5))
       .to_be_null;
    ut.expect(betwnstr(string_in => c_teststring, start_in => -3, end_in => 0))
       .to_be_null;
  end;
end test_betwnstr;
/
```

Expectation

- More natural language

```
utAssert.eq ('Typical valid usage'
            ,betwnstr(string_in => 'abcdefgh'
                     ,start_in => 3
                     ,end_in => 5)
            ,'cde');

ut.expect(betwnstr(string_in => 'abcdefgh'
                  ,start_in => 3
                  ,end_in => 5)
         ).to_equal('cde');
```

Expectation

to_be_null
to_be_not_null
to_be_true
to_be_false
to_equal
to_be_like
to_match
to_be_between
to_be_greater_or_equal
to_be_greater_than
to_be_less_or_equal
to_be_less_than

not_to_be_null
not_to_be_not_null
not_to_be_true
not_to_be_false
not_to_equal
not_to_be_like
not_to_match
not_to_be_between
not_to_be_greater_or_equal
not_to_be_greater_than
not_to_be_less_or_equal
not_to_be_less_than

Expectation

Matcher	blob	boolean	clob	date	number	timestamp	timestamp with timezone	timestamp with local timezone	varchar2	interval year to month	interval day to second	cursor	nested table/ varray	object
be_not_null	X	X	X	X	X	X	X	X	X	X	X	X	X	X
be_null	X	X	X	X	X	X	X	X	X	X	X	X	X	X
be_false		X												
be_true		X												
be_greater_than				X	X	X	X	X		X	X			
be_greater_or_equal				X	X	X	X	X		X	X			
be_less_or_equal				X	X	X	X	X		X	X			
be_less_than				X	X	X	X	X		X	X			
be_between				X	X	X	X	X	X	X	X			
equal	X	X	X	X	X	X	X	X	X	X	X	X	X	X
match			X						X					
be_like			X						X					
be_empty	X		X									X	X	
have_count												X	X	

Annotation

```
create or replace package test_betwnstr is


end test_betwnstr;
```

Annotation

```
create or replace package test_betwnstr is
  -- %suite (betwnstr)



end test_betwnstr;
```

Annotation

```
create or replace package test_betwnstr is
  -- %suite (betwnstr)


  procedure test_betwnstr;
end test_betwnstr;
```

Annotation

```
create or replace package test_betwnstr is
  -- %suite (betwnstr)
  -- Purpose : Unittesting betwnstr with utPL/SQL 3

  -- %test(BETWNSTR Tests)
  procedure test_betwnstr;
end test_betwnstr;
```

```
create or replace package body test_betwnstr is
  c_teststring constant varchar2(8) := 'abcdefgh';
  procedure test_betwnstr is
  begin
    ut.expect(betwnstr(string_in => c_teststring, start_in => 3, end_in => 5))
      .to_equal('cde');
    ut.expect(betwnstr(string_in => c_teststring, start_in => 0, end_in => 2))
      .to_equal('ab');
    ut.expect(betwnstr(string_in => c_teststring, start_in => 3, end_in => 9999))
      .to_equal('cdefgh');
    ut.expect(betwnstr(string_in => c_teststring, start_in => null, end_in => 5))
      .to_equal('abcde');
    ut.expect(betwnstr(string_in => c_teststring, start_in => 3, end_in => null))
      .to_equal('cdefgh');
    ut.expect(betwnstr(string_in => c_teststring, start_in => -3, end_in => -5))
      .to_be_null;
    ut.expect(betwnstr(string_in => c_teststring, start_in => -3, end_in => 0))
      .to_be_null;
  end;
end test_betwnstr;
```


ut
DEMO@demo>


DEMO@demo> set serveroutput on size unlimited
DEMO@demo>


```
DEMO@demo> set serveroutput on size unlimited
DEMO@demo> exec ut.run
betwnstr
BETWNSTR Tests [.002 sec]
Finished in .005925 seconds
1 tests, 0 failed, 0 errored, 0 disabled, 0 warning(s)

PL/SQL procedure successfully completed.

DEMO@demo>
```


```
DEMO@demo> set serveroutput on size unlimited
DEMO@demo> exec ut.run
betwnstr
BETWNSTR Tests [.002 sec]
Finished in .005925 seconds
1 tests, 0 failed, 0 errored, 0 disabled, 0 warning(s)

PL/SQL procedure successfully completed.

DEMO@demo> exec ut.run('demo.test_betwnstr')
betwnstr
BETWNSTR Tests [.003 sec]
Finished in .006984 seconds
1 tests, 0 failed, 0 errored, 0 disabled, 0 warning(s)

PL/SQL procedure successfully completed.

DEMO@demo>
```

Annotation

Annotation	Level	Description
--%beforeall	Procedure	Denotes that the annotated procedure should be executed once before all elements of the suite.
--%beforeall([[<owner>.]<package>.]<procedure>[,...])	Package	Denotes that the mentioned procedure(s) should be executed once before all elements of the suite.
--%afterall	Procedure	Denotes that the annotated procedure should be executed once after all elements of the suite.
--%afterall([[<owner>.]<package>.]<procedure>[,...])	Package	Denotes that the mentioned procedure(s) should be executed once after all elements of the suite.

Annotation

Annotation	Level	Description
--%beforeall	Procedure	Denotes that the annotated procedure should be executed once before all elements of the suite.
--%beforeall([[<owner>.]<package>.]<procedure>[,...])	Package	Denotes that the mentioned procedure(s) should be executed once before all elements of the suite.
--%afterall	Procedure	Denotes that the annotated procedure should be executed once after all elements of the suite.
--%afterall([[<owner>.]<package>.]<procedure>[,...])	Package	Denotes that the mentioned procedure(s) should be executed once after all elements of the suite.
--%beforeeach	Procedure	Denotes that the annotated procedure should be executed before each %test procedure in the suite.
--%beforeeach([[<owner>.]<package>.]<procedure>[,...])	Package	Denotes that the mentioned procedure(s) should be executed before each %test procedure in the suite.
--%aftereach	Procedure	Denotes that the annotated procedure should be executed after each %test procedure in the suite.
--%aftereach([[<owner>.]<package>.]<procedure>[,...])	Package	Denotes that the mentioned procedure(s) should be executed after each %test procedure in the suite.

Annotation

Annotation	Level	Description
--%beforeall	Procedure	Denotes that the annotated procedure should be executed once before all elements of the suite.
--%beforeall([[<owner>.]<package>.]<procedure>[,...])	Package	Denotes that the mentioned procedure(s) should be executed once before all elements of the suite.
--%afterall	Procedure	Denotes that the annotated procedure should be executed once after all elements of the suite.
--%afterall([[<owner>.]<package>.]<procedure>[,...])	Package	Denotes that the mentioned procedure(s) should be executed once after all elements of the suite.
--%beforeeach	Procedure	Denotes that the annotated procedure should be executed before each %test procedure in the suite.
--%beforeeach([[<owner>.]<package>.]<procedure>[,...])	Package	Denotes that the mentioned procedure(s) should be executed before each %test procedure in the suite.
--%aftereach	Procedure	Denotes that the annotated procedure should be executed after each %test procedure in the suite.
--%aftereach([[<owner>.]<package>.]<procedure>[,...])	Package	Denotes that the mentioned procedure(s) should be executed after each %test procedure in the suite.
--%beforetest([[<owner>.]<package>.]<procedure>[,...])	Procedure	Denotes that mentioned procedure(s) should be executed before the annotated %testprocedure.
--%aftertest([[<owner>.]<package>.]<procedure>[,...])	Procedure	Denotes that mentioned procedure(s) should be executed after the annotated %test procedure.

Some Key Features

- native comparison of complex types (objects/collections/cursors)
- tests identified and configured by annotations
- Build-in coverage reporting
- Integration with SonarQube, Coveralls, Jenkins and Teamcity with reporters
- plugin architecture for reporters and matchers
- multi-reporting from test-run from command line

Code coverage

All files (66.04%)

Generated about 5 hours ago

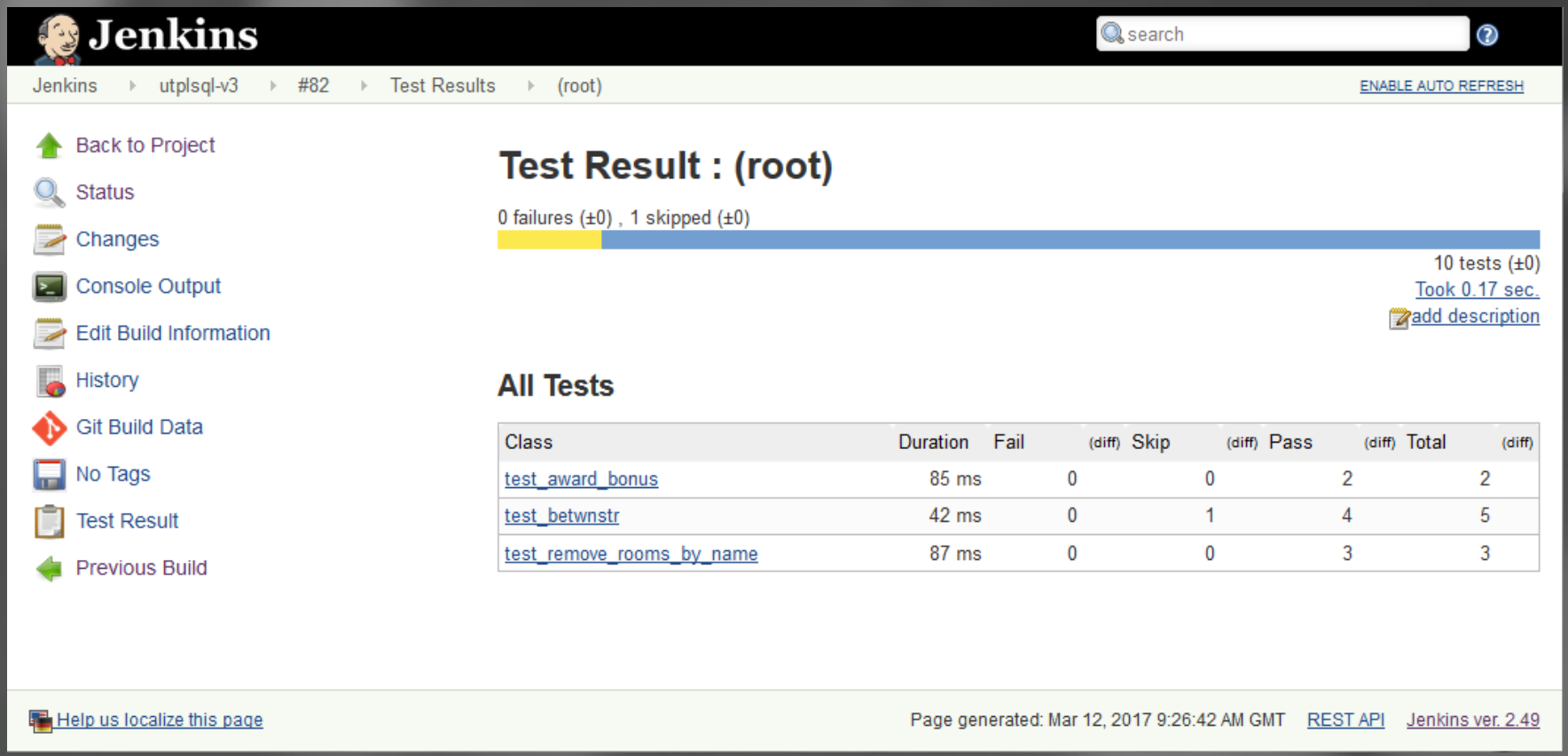
All files (66.04% covered at 71 hits/line)

69 files in total. 2285 relevant lines. 1509 lines covered and 776 lines missed

Search:

File	% covered	Lines	Relevant Lines	Lines covered	Lines missed	Avg. Hits / Line
source/core/ut_suite_manager.pkb	88.51 %	526	174	154	20	51
source/core/ut_annotations.pkb	97.01 %	363	67	65	2	268
source/core/coverage/ut_coverage.pkb	1.19 %	324	84	1	83	0
source/core/ut_utils.pkb	88.12 %	295	101	89	12	171
source/reporters/ut_coverage_report_html_helper.pkb	0 %	255	255	0	255	0
source/expectations/matchers/ut_equal.tpb	98.97 %	231	97	96	1	83
source/reporters/ut_documentation_reporter.tpb	97.22 %	178	72	70	2	35
source/api/ut.pkb	81.25 %	162	48	39	9	24
source/expectations/ut_expectation.tpb	62.5 %	157	48	30	18	136
source/reporters/ut_teamcity_reporter_helper.pkb	48.21 %	153	56	27	29	17
source/expectations/matchers/ut_be_between.tpb	98.39 %	151	62	61	1	14
source/core/ut_metadata.pkb	94.44 %	151	36	34	2	861

Integration with Jenkins with reporters




QUALITY


QUALITY

Now what?

Resources

- SQL Developer

  https://www.oracle.com/database/technologies/appdev/sql-developer.html

  https://www.oracle.com/technetwork/developer-tools/sql-developer/downloads/index.html
- utPLSQL

  http://utplsql.org/
- utPLSQL v3 Cheat Sheet

  https://www.cheatography.com/jgebal/cheat-sheets/utplsql-v3/


Q&A

Oracle Cloud Infrastructure

New Free Tier

oracle.com/gbtour

Always Free

Services you can use for unlimited time

+

30-Day Free Trial

Free credits you can use for more services