Cognizant - DN 4.0 I Deep Skilling

Week-2

Control Structures:-

Code:-

DECLARE

num NUMBER := 10;

BEGIN

IF MOD(num, 2) = 0 THEN

DBMS\_OUTPUT.PUT\_LINE('The number is even.');

ELSE

DBMS\_OUTPUT.PUT\_LINE('The number is odd.');

END IF;

END;

/

Stored Procedures

Code:-

CREATE OR REPLACE PROCEDURE check\_even\_odd(p\_num IN NUMBER) IS

BEGIN

IF MOD(p\_num, 2) = 0 THEN

DBMS\_OUTPUT.PUT\_LINE('Even number.');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Odd number.');

END IF;

END;

/

Setting Up Junit

Code:-

XML

<dependencies>

<!-- JUnit -->

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.13.2</version>

<scope>test</scope>

</dependency>

<!-- H2 In-Memory Database -->

<dependency>

<groupId>com.h2database</groupId>

<artifactId>h2</artifactId>

<scope>test</scope>

</dependency>

</dependencies>

SQL

CREATE TABLE users (

id INT PRIMARY KEY,

name VARCHAR(50)

);

Assertions in Junit

Code:-

import org.junit.\*;

import java.sql.\*;

import static org.junit.Assert.\*;

public class UserDaoTest {

private static Connection conn;

@BeforeClass

public static void setup() throws Exception {

conn = DriverManager.getConnection("jdbc:h2:mem:testdb", "sa", "");

Statement stmt = conn.createStatement();

stmt.execute("CREATE TABLE users (id INT PRIMARY KEY, name VARCHAR(50))");

stmt.execute("INSERT INTO users VALUES (1, 'Keerthana')");

}

@Test

public void testUserExists() throws Exception {

PreparedStatement ps = conn.prepareStatement("SELECT name FROM users WHERE id = ?");

ps.setInt(1, 1);

ResultSet rs = ps.executeQuery();

assertTrue("User should exist", rs.next());

assertEquals("Keerthana", rs.getString("name"));

}

@AfterClass

public static void teardown() throws Exception {

conn.close();

}

}

Arrange Code:-

@BeforeClass

public static void setupDatabase() throws Exception {

conn = DriverManager.getConnection("jdbc:h2:mem:testdb", "sa", "");

Statement stmt = conn.createStatement();

stmt.execute("CREATE TABLE users (id INT PRIMARY KEY, name VARCHAR(50))");

stmt.execute("INSERT INTO users VALUES (1, 'Keerthana')");

}

Act Code:-

@Test

public void testUserRetrieval() throws Exception {

// Arrange

PreparedStatement ps = conn.prepareStatement("SELECT name FROM users WHERE id = ?");

ps.setInt(1, 1);

// Act

ResultSet rs = ps.executeQuery();

// Assert

assertTrue("User should exist", rs.next());

assertEquals("Keerthana", rs.getString("name"));

}

Teardown Code:-

@AfterClass

public static void teardownDatabase() throws Exception {

conn.close();

}

Mocking and Stubbing:-

import static org.mockito.Mockito.\*;

import org.junit.\*;

import java.sql.\*;

public class UserDaoTest {

@Test

public void testGetUserById() throws Exception {

// Arrange

Connection mockConn = mock(Connection.class);

PreparedStatement mockStmt = mock(PreparedStatement.class);

ResultSet mockRs = mock(ResultSet.class);

when(mockConn.prepareStatement("SELECT name FROM users WHERE id = ?")).thenReturn(mockStmt);

when(mockStmt.executeQuery()).thenReturn(mockRs);

when(mockRs.next()).thenReturn(true);

when(mockRs.getString("name")).thenReturn("Keerthana");

// Act

mockStmt.setInt(1, 1);

ResultSet rs = mockStmt.executeQuery();

// Assert

Assert.assertTrue(rs.next());

Assert.assertEquals("Keerthana", rs.getString("name"));

}

}

Verifying Interactions:-

Code:-

-- Users table

CREATE TABLE users (

user\_id INT PRIMARY KEY,

name VARCHAR(50)

);

-- Interactions table

CREATE TABLE interactions (

interaction\_id INT PRIMARY KEY,

user1\_id INT,

user2\_id INT,

interaction\_date DATE,

FOREIGN KEY (user1\_id) REFERENCES users(user\_id),

FOREIGN KEY (user2\_id) REFERENCES users(user\_id)

);

Logging Error Messages and Warning Levels

Code:-

CREATE TABLE ErrorLog (

ErrorID INT IDENTITY(1,1) PRIMARY KEY,

ErrorMessage NVARCHAR(4000),

ErrorSeverity INT,

ErrorState INT,

ErrorTime DATETIME DEFAULT GETDATE()

);

BEGIN TRY

-- Simulate an error (e.g., divide by zero)

DECLARE @x INT = 1 / 0;

END TRY

BEGIN CATCH

INSERT INTO ErrorLog (ErrorMessage, ErrorSeverity, ErrorState)

VALUES (

ERROR\_MESSAGE(),

ERROR\_SEVERITY(),

ERROR\_STATE()

);

END CATCH;