Oracle SQL/PLSQL TDD

Trainer guide

A black background with yellow circles

Description automatically generated

Contents

[Overview 2](#_Toc182296079)

[Course objectives 2](#_Toc182296080)

[Course agenda 2](#_Toc182296081)

[Target audience 2](#_Toc182296082)

[Course author 3](#_Toc182296083)

[Additional resources 3](#_Toc182296084)

[Agenda and timings 4](#_Toc182296085)

[Training schedule 5](#_Toc182296086)

# Overview

#### Course objectives

* Understand the principles and mindset of Test-Driven Development (TDD).
* Write effective, isolated unit tests using **utPLSQL**
* Apple the Red-Green-Refactor cycle to incrementally build functionality.
* Understand the structure of the Test Package using utPLSQL
* Write the first TDD Test package.
* Write the Test to test the *SQL queries, Stored Procedures and Database Triggers*
* Identify and refactor untestable code.

#### Course agenda

Day 1:

1. Introductions, setup and goals.
2. Why TDD? Exploring test-first versus test-after development.
3. The TDD cycle: Red-Green-Refactor and test design basics.
4. Understand the structure of the Test Package using utPLSQL
5. Writing unit tests: using utPLSQL
6. Write the first TDD Test package.
7. Write the Test to test the *Select*, *Insert*, *Update*, *Delete* queries.
8. Identify and refactor untestable code.

Day 2:

1. Mini project to TEST package to verify the insert, delete and update the queries on the tables employees.
2. Writing test
3. Write Test for verify and validate view, joins and complex joins.
4. Write a code to test view with aggregates.
5. Write a code test stored procedure like Functions.
6. Testing the Database Triggers
7. Code coverage using utPLSQL

#### Target audience

This course is designed for:

* Junior to mid-level frontend developers who use JavaScript but may not yet use a modern framework.
* Leaners from software, testing, or QA backgrounds who want to build confidence in JavaScript based testing.
* Professionals who may need to train or support colleagues in test practices.
* Individuals preparing to learn frameworks like React or Angular and looking to establish strong testing habits early.

# Course author

|  |  |  |
| --- | --- | --- |
| Name | Courseware title | Date of ownership |
| Muhammad Shafeeq | - | 15/07/2025 |

# Additional resources

N/A

# Agenda and timings

|  |  |  |
| --- | --- | --- |
| Timings | Length | Topic |
| 9:30am | 30 mins | Introductions and setup |
| 10:00am | 45 mins | Why TDD? What’s the Problem? |
| 10:45am | 15 mins | Break |
| 11:00am | 30 mins | TDD Lifecycle and Writing Frist Test |
| 11:30am | 60 mins | **Lab 1- First Test- Its Structure and working** |
| 12:30pm | 60 mins | Lunch |
| 13:30pm | 45 mins | Writing Good Unit Tests |
| 14:15pm | 30 mins | **Lab 2-TDD SQL – Select Query** |
| 14:45 | 30 mins | **Lab 3-TDD SQL – Delete Query** |
| 15:15pm | 15 mins | Break |
| 15:30pm | 30 mins | **Lab 4-TDD Insert - Query** |
| 16:00pm | 60 mins | **Lab 5- TDD SQL - Update-Query** |
| 17:00pm | Finish | End of Day |
|  |  |  |
| 9:30am | 30 mins | Recap |
| 10:00am | 60 mins | **Mini project (Pair Programming)**  *Build building SQL scripts that:*   * *Insert new employees* * *Update salaries* * *Delete employees*   *Employees table structure(emp\_id NUMBER PRIMARY KEY, emp\_name VARCHAR2(100), department VARCHAR2(50), salary NUMBER)*  You want to write tests first, make them fail, then write the SQL, and finally pass the tests — the TDD way.  **Task 6- TDD - Insert, update, delete Query**  **Create one Package which will contain 3 tests.**   1. **For test\_insert\_Employee** 2. **For test\_update\_Salary** 3. **For Delete\_by\_Department** |
| 11:00 am | 15 mins | Break |
| 11:15am | 30 mins | Project Continues |
| 11:45am | 45 mins | **Lab 7- TDD - SQL Views and Complex Joins** |
| 12:30 pm | 45 mins | **Lab 8- TDD - views with aggregations** |
| 13:15 pm | 60 mins | Lunch |
| 14:15 pm | 45 mins | **Lab 9- TDD - Stored Function** |
| 15:00 pm | 45 mins | **Lab 10 - TDD Triggers** |
| 15:45 PM | 15 mins | Break |
| 16:00 pm | 30 mins | **Lab 11- TDD - Code Coverage in utPLSQL** |
| 16:30 PM | 15 mins | Wrap-Up and Feedback |
| 17:00pm | Finish | End of Day |

