Instruction Sheet-1	Learning Guide #1

This learning guide is developed to provide you the necessary information regarding the following content coverage and topics.

- Preparing Test environment
- Determining Software life cycle is based on work principles.
- Preparing Test Plane based on Client Requirement
- Scheduling Test
- Batch Testing
- Confirming Test Result Criteria based on system requirement Documentation

This guide will also assist the trainee to attain the learning outcome stated in the cover page. Specifically, upon completion of this Learning Guide, the trainee will be able to:

- Prepare Test environment
- Determine Software life cycle is based on work principles.
- Prepare Test Plane based on Client Requirement
- Schedule Test
- Batch Testing

Confirming Test Result Criteria based on system requirement Documentation

Learning Instructions:

1. Read the specific objectives of this Learning Guide.
2. Read the information written in the “Information Sheets 1”. Try to understand what are being discussed. Ask your teacher for assistance if you have hard time understanding them.
3. Accomplish the “Self-check 1”.
4. Ask from your teacher the key to correction (key answers) or you can request your teacher to correct your work. (You are to get the key answer only after you finished answering the Self-check 1).
5. If you earned a satisfactory evaluation proceed to “Information Sheet 2”. However, if your rating is not satisfactory, see your teacher for further instructions,
6. Submit your accomplished Self-check. This will form part of your training portfolio.

MODULE TITLE: **Perform Database System Test**

NOMINAL DURATION: **60** hrs.

MODULE CONTENTS:

LO1. Prepare for Test

1.1 Preparing Test environment inline with work guideline.

A **testing environment** is a setup of software and hardware on which the testing team is going to perform the testing of the newly built software/hardware product.

This setup consists of the physical setup which includes hardware, and logical setup that includes Server Operating system, client operating system, database server, front end running environment (interface) or any other software components required to run the new product.

1.2 Determining Software life cycle based on work principles.

The software life cycle is a general model of the software development process, including all the activities and work process required to develop a software system.

Software life cycle describe phases of the software cycle and the order in which those phases are executed.

Each phase produces deliverables required by the next phase in the life cycle.

There are six phases in every Software development life cycle model:

- Requirement gathering and analysis
- Design
- Implementation or coding
- Testing
- Deployment
- Maintenance

1) Requirement gathering and analysis: Business requirements are gathered in this phase.

The general questions that need answer during a requirements gathering phase include:

- Who is going to use the system?
- How will they use the system?
- What data should be input into the system?
- What data should be output by the system?

After requirements are gathered and analyzed for their validity, requirements Specification document is created which serves the purpose of guideline for the next phase of the model.

2) Design: In this phase, the system and software design is prepared from the requirement specifications documents which were studied in the first phase. System Design helps in specifying hardware and overall system architecture.

3) Implementation/Coding: On receiving system design documents, the work is divided in modules/units and actual coding is started. This is the longest phase of the software development life cycle.

4) Testing: After the code is developed, it is tested against the requirements to make sure that the product is actually solving the needs addressed and gathered during the requirements phase.
During this phase unit testing, integration testing, system testing, acceptance testing are done.

5) Deployment: After successful testing, the product is delivered/deployed to the customer for their use.

6) Maintenance: Once when the customers starts using the developed system, then the actual problems comes up and needs to be solved from time to time.

1.3 Defining Test plan and appropriate test tools

A test plan is a document detailing a systematic approach to testing a system such as a machine or software. A test plan can be defined as a document describing the scope, approach, resources, and schedule of intended testing activities. It identifies test items, the features to be tested, the testing tasks, who will do each task, and any risks requiring contingency planning.

A **test plan** outlines the strategy that will be used to test an application the resource that will be used and the test environment in which testing will be performed the limitation of testing and the schedule or testing activities. Typically the quality assurance team will be responsible for writing a test plan.

Test plan includes

- Introduction
- Assumption when testing the application
- List of test case
- List of features to be tested
- Approach
- The source used to perform testing
- Schedule

What is database testing?

A database has two main parts - the **data structures** (the schema) that store the data AND the **data** itself. Database testing involves finding out the answers to the following questions:

- Is the data organized well logically?
- Does the database perform well?
- Do the database objects like views, triggers, stored procedures, functions and jobs work correctly?
- Does the database implement constraints to allow only correct data to be stored in it?
- Is the data secured from unauthorized access?

Database testing means test engineer should test the data integrity, data accessing, query retrieving, modifications, updating and deletion etc.
Database testing basically include the following.

- Data validity testing - you should be good in SQL queries.
- Data Integrity testing - should know about referential integrity and different constraint.
  - Data integrity Refers to the validity of data (The accuracy and consistency of stored data).
- Performance related to database - you should have idea about the table structure and design.
- Testing of Procedure, triggers and functions.
- Checking the integrity of UI data with Database Data
- Checking execution of stored procedures with the input values taken from the database tables
- data accessing, query retrieving, modifications, updating and deletion etc

What are the major areas of DB testing?

1. Correctness of data entered in to the table

- ✓ In software system date travel back and forth from GUI to the back-end and vise-versa.
- ✓ So to check correctness of data the following aspect will be assessed
- ✓ To check whether the field in GUI correctly mapped with the corresponding DB table attributes.
- ✓ When even certain actions performed in the front end of an application a corresponding CRUD (create, retrieve, update, delete) action get invoked at the back end.
- ✓ A tester will have to check in the right action is invoked and the invoked action itself is success full or not.

2. Correctness and completeness of data migration

- ✓ Data migration is the process of converting a data from source to a new DBMS source.
- ✓ This data migration happened when an organization decided to upgrade or change its DB.
- ✓ For example it the original DB in to MY-SQ L DB is called data migration.

  Major check point during data migration
  - Table count
  - Data type matching
  - Perform check sum an column holding numerical data
  - Row count
  - Matching of total null values per column
  - Matching of not-null per column
  - Check min value per numeric column
  - Checking max value per numeric column
- ✓ Data redundancy
- ✓ Matching control tables verify the exact transfer of relationship among table.
- ✓ Functional testing un migrated data
- ✓ Random sampling picking and matching data from corresponding tables randomly

3. Functionality and performance DB testing

- ✓ Functionality testing is a type back box testing that is based on the specification of the SW that is to be tested.
- ✓ The application is tested by providing input and then results are examined that need to conform to the functionality of it was intended.

There are five steps that are involved when testing an application for functionality.

I. The determination of the functionality that the intended application is meant to perform.
II. The creation of test data based on the specification of the application.
III. The output based on the test data and the specification of the application.
IV. The writing of test scenarios and execution of test case.
V. The compression of actual & expected results based on executed test cases.

Conform test results based on system requirement documentation

- If the database functionalities that we are deleting match the contractual agreement of the customer & the developer then it means the taste results being confirmed based on system requirement documentation.

4. Data security testing

- ✓ Like any other in to system. DB systems require authentication and authorization controls.

  - ✓ Most commercial DB system contain stored procedure that simplifies the job of DB administrator unfortunately, if these procedure is not properly locked down it may discovery your system for hackers.
  - ✓ SW testing involves the testing of SW in order to identify any flows and gaps from security vulnerability.

- The followings are the main aspect which security testing should ensure
  - Confidentiality
  - Integrity
  - Authentication
  - Authorization (privilege)
  - Availability
  - Non repudiation

5. Data integrity testing

- This means following any of the CRUD operation the update and must resent values status of shared data should appear an all forms and screen.
- So device your DB test case in a way to include checking the data in all the places it appears to see if it is consistency the same.

❖ There are different types of data integrity testing

✓ Entity integrity- can be achieved through primary key and unique key constraint.

✓ Domain integrity- can be through not null default check should be tested whether it is taking default value even through it will not give checking the value in the column

✓ Referential integrity- can be achieved through foreign key should be tested checking whether "child" row are deleted or not when apparent row is deleted from parent table.

6. Data base schema testing

- Database is nothing but a formal definition of raw the data is going to be organized in to a DB.
- To test DB schema
- Identify the requirement based on which the database a grate sample requirements are
  - ✓ Primary key to be created before any other field are created
  - ✓ Foreign keys should be completely indexed for easy retrieval and searching
  - ✓ Fields name starting with character use one or the following according to relevance.
  - ✓ Regular expression for validating the name of the individual fields and their value
  - ✓ Tools like schema crawler

7. DB trigger testing

When a certain event takes places on a certain table a piece of code (a trigger) can be instructed to be executed.

Foreign a new student join a school and it the student is taking 2 classes match and science then the students add the student table. A trigger could be adding the student to the corresponding subject table an is added to the student table.

The common method to test a DB trigger is to execute SQL query embedded in the trigger independently first and record the result. Follow this op with executing the trigger as a work and compare the result.

❖ DB trigger are tested during both back box and white box testing.

White box testing –subs & drivers are to insert or update or delete data that would result in the trigger being invoked. The basic idea is to just test DB alone even before the integration with the front end (UI) is made.

❖ Black box testing

A. Since UI and DB integration is now available we can insert /delete/ update from the front end in a way that the trigger gets invoked following that select statements can based to retrieve the DB data. To see if the trigger was success full in performing the intended operation

B. Second to test this is to directly load that would invoke the trigger and see if it works as intended.

8. Stored procedure

- Stored procedures are more are less similar to user defined function.
- These can be invoked by a call or be procedure execute procedure statement and output usually in the form of result set. These are stored in RDMS and are available for application.

  There are also tested during

  - White box testing
  - Black box testing

Generally, Trigger and stored procedure

- ✧ Database triggers are database object used to implement business rule and data integrity.
- ✧ Triggers are special types of stored procedure. Examples logging deletion.
- ✧ Scenario we want to keep a log file contains data from row's that have been deleted from books table. Here is a trigger accomplice this

CREAT TRIGGER BOOK-DELETED ALTER DELETE ON Books REFERENGNG old ROWS As old1 FOR EACH ROWS INSERT in to Book-deleted-log VALUES (old title)

This trigger automatically executed when a book is deleted from books table. The trigger copied the tittle of every deleted book from books table in to book-deleted-log table

The difference between stored procedure and trigger

✧ **Stored procedure**	✧ **Trigger**
✓ executed during call	✓ Executed automatically
✓ Bind to any table	✓ Bind only for a single table
✓ Perform any activity	✓ Used to implement business rule & data integrity

9. **Field constraint**- default value unique value and foreign key.

- Perform a front end operation which over runs the DB object condition.
- Validate the result with SQL query
- Checking the default value for certain field is quite simple.
- It is part of business rule validation
- You can do it manually or you can use tools like …..to do so
- Manually you can perform an action is that will add a value other than the default value in to the filed from front end see if it results in an error.
- The generally step of DB testing is not very d/t from any other application testing.

The following are the step of DB testing

Step #1. Prepared the ……………….
Step #2. Prepared test case
Step #3. Run a test
Step #4. Validate a cording to expected result
Step #5. Report the result to the respective stake holder

Usually SQL queries are used to develop the test. The most commonly used command is a "select"

```
Select * from <TN>where <condition>
```

A part form select SQL has 3 important types of commands.

1. DDL
2. DML } DBMS
3. DCL

1. DDL- uses CREAT, ALTER, RENAME, DROP and TRUNCATE to handle table
2. DML- includes statement to add update and delete records.
3. DCL- deals with giving authorization to user for manipulation and access to the data.

Database testing can be performed by two mechanisms

a. Manual –open the database and using sql queries to check the database

There are different stages in manual testing or testing levels. Like

✓ **Unit testing**

Unit testing also known as component testing refers to tests that verify the functionality of a specific section of code .Usually at the function in an object oriented environment. This is usually at the level, and the minimal unit tests include the constructors and destructors.

These types of tests are usually written by developers as they work on code (white-box style) to ensure that the specific function is working as exected.one function might have multiple tests to catch corner cases or other branches in the code.

✓ **Integration testing**

Integration testing is any types of software testing that seeks to verify the interfaces between components against a software design. Software components may be integrated in an iterative way or all together ("big bang").Normally the former is considered a better practice since it allows interfaces issues to be located more quickly and fixed.

Integration testing works to expose defects in the interfaces and interaction between integrated components (modules).Progressively large groups of tested software components corresponding to the elements of the architectural design are integrated and tested until the software works as a system.

✓ **System testing**

System testing tests completely integrated system to verify that it meets requirements. In addition ,the software testing should ensure that the program, as well as working as expected, does not also destroy or partially corrupt its operating environment or cause other processes within the environment becomes inoperative(this includes not corrupting shared memory, not consuming or locking up excessive resources and leaving any parallel processes unharmed by its presence)

✓ **User acceptance testing**

User Acceptance Testing (UAT), also known as **beta** or **end-user** testing, is defined as testing the software by the user or client to determine whether it can be accepted or not. This is the final testing performed once the functional, system and regression testing are completed

b. **Automation-** means using any functional tools like win-runner or QTP and perform testing

Purposes of database testing

Database testing usually consists of a layered process, including the user interface (UI) layer, the business layer, the data access layer and the database itself.

Types of testing and processes

1. **Black box testing**
   - It involves testing interfaces and the integration of the database.
   - Black box testing (not knowledge) testing without having any knowledge of interior part of the SW.

Advantage

- ✓ Well suited and efficient for large code segment.
- ✓ Crealy separate were perspective and developer and developer perspective.

Disadvantage

- ✓ Limited coverage
- ✓ Inefficient testing

2. **White Box testing**
   - White box testing (also known as **clear box testing, glass box testing, transparent testing**) mainly deals with the internal structure of the database. The specification details are hidden from the user.
     - It involves the testing of database triggers and logical views.
     - It performs module testing of database functions, triggers, views, SQL queries etc.
     - It validates database tables, data models, database schema etc.
     - It checks rules of Referential integrity.
     - It selects default table values to check on database consistency.

While White box testing can be applied at the unit, integration and system levels of the software testing process. It is usually done at the unit level. It can test paths within a unit, paths between units during testing**.**

White box testing

- ✓ Tester has be internal code knowledge
- ✓ Testing is the detail investigation of internal

Logic and structure of the code

Advantage

- ✓ It is very easily to field out which type of data can help in testing
- ✓ It helps in permitting the code
- ✓ Extra lines of code can be removed which can bring on hidden defects.

Disadvantage

- ✓ Costly
- ✓ Difficult to maintain

The main advantage of white box testing in database testing is that coding error are detected, so internal bugs in the database can be eliminated.

Types of Software testing

1. Front-end testing
2. Back-end testing

- ❖ **Front-end testing (GUI) testing**

  Can be like
  - ➢ Log in functionality
  - ➢ Different types of validation
  - ➢ The password filed text presentation
  - ➢ Home pages
- ❖ **Back-end testing**
  - ➢ It is finding error on the DB

Why back-end testing? B/c a back-end may function of may cause system data corruption and bad performance.

Type of back-end testing

1. Structural- the relationship aspect
2. Functionality- expected function
3. Boundary value-the minimum and maximum value
4. Stress testing- heavy loads

Generally all testing types are named under software testing

Who does testing?

In it industries large company has testing team with responsibility to evaluate the developed SW.

More over the following personal can conduct testing.

- ✓ Developer
- ✓ Project manager
- ✓ End user

When to start testing?

- ✓ Cost
- ✓ Time to set

However in SDLC testing can be start from the requirement gathering phase and rest fill the SW

When to stop testing?

- ✓ Difficult to determine
- ✓ Never ending process

But testing can be ease at

- ✓ Testing dead line
- ✓ Competition of test case execution
- ✓ Management decision

Software testing documentation sample templates

I. **Introduction**

Under this title we have to explain

- What is testing?
- Why we test a SW?
- What is the purpose of preparing this document?

II. Relationship to other document

- What is the issue that link this document from other documents RAD, SDD and SRS

III. **Feature to be tested not to be tested** among the module of the SW which are included in the testing activity and are not?

IV. **Approach of testing**

- What is the appropriate testing strategy for this SW testing?

V. Criteria to be used for can firming the test result

- What is the remark of successful and unsuccessful testing?

VI. System over view

General description of the system

VII. Test cases specification

- Provide detail description of lest cases

1.4 Scheduling Test

Scheduling is the process of deciding how to commit resources between varieties of possible tasks.

Schedule Test means arrange or plan (an event) to take place at a particular time.

Scheduling through Task Scheduler allows you to automatically perform routine tasks on a chosen schedule. The Task Scheduler does this by monitoring whatever criteria you choose to initiate the tasks (daily, weekly etc and time also) and then execute the task when the criteria is met. With Scheduled Tasks, you can schedule any script, program, or document to run at a time that you specify when creating the task

1.5 Preparing Test scripts (online test) or test run (batch test)

A **test script** in software testing is a set of instructions that will be performed on the system under test to test that the system functions as expected, or

A test script is a short program written in a programming language used to test part of the functionality of a software system. A written set of steps that should be performed automatically can also be called a test script.

Batch testing is a laboratory testing procedure in which one test is done simultaneously on multiple specimens.

A test script is the executable form of a test. It defines the set of actions to carry out in order to conduct a test and it defines the expected outcomes and results that are used to identify any deviance in the actual behaviour of the program from the logical behavior in the script.

Testing uses a lot of terminology. In the test context, we will use the following definitions:

- **Test case**: a logical description of a test. It details the purpose of the test and the derivation audit trail.
- **Test Script**: the physical, executable, description of the test case.

  TSQL Scripts that executes a test case may include assertions like;
  Return values
  Output parameters
  Expected Errors

- **Automated test script**: a program that implements a test.

The development life cycle has a number of processes and tasks that the development community is involved in:

- Requirements
- Design
- Coding
- Testing

Testers are familiar with each of these stages in the context of system development and its relationship to the construction of tests.

1.6 Reviewing expected results against acceptance criteria and system requirements Documentation.

Expected results are reviewed against acceptance criteria (walkthrough) and system requirements Documentation.

Instruction Sheet-2	Learning Guide #2

This learning guide is developed to provide you the necessary information regarding the following content coverage and topics.

- ➢ Running test scripts and document results in line with test and acceptance processes.
- ➢ Performing Required quality benchmarks or comparisons.
- ➢ Adopting Organization/industry standards.
- ➢ Comparing Actual results to expected results.

This guide will also assist the trainee to attain the learning outcome stated in the cover page. Specifically, upon completion of this Learning Guide, the trainee will be able to:

- › Running test scripts and document results in line with test and acceptance processes.
- › Performing Required quality benchmarks or comparisons.
- › Adopting Organization/industry standards.
- › Comparing Actual results to expected results.

Learning Instructions:

7. Read the specific objectives of this Learning Guide.
8. Read the information written in the “Information Sheets 2”. Try to understand what are being discussed. Ask your teacher for assistance if you have hard time understanding them.
9. Accomplish the “Self-check 2”.
10. Ask from your teacher the key to correction (key answers) or you can request your teacher to correct your work. (You are to get the key answer only after you finished answering the Self-check 2).
11. If you earned a satisfactory evaluation proceed to “Information Sheet 3”. However, if your rating is not satisfactory, see your teacher for further instructions,
12. Submit your accomplished Self-check. This will form part of your training portfolio.

LO2. **Conduct Test**

- › Running test scripts and document results in line with test and acceptance processes.
- › Performing Required quality benchmarks or comparisons.
- › Adopting Organization/industry standards.
- › Comparing Actual results to expected results.

LO2. Conduct Test

Test script development involves the same processes and techniques used when constructing software programs, any experience.

A test script is the executable form of a test. It defines the set of actions to carry out in order to conduct a test and it defines the expected outcomes and results that are used to identify any deviance in the actual behaviour of the program from the logical behaviour in the script.

Testing uses a lot of terminology. In this text I will use the following definitions:

- **Test case**: a logical description of a test. It details the purpose of the test and the derivation audit trail.
- **Test Script**: the physical, executable, description of the test case.
- **Automated test script**: a program that implements a test.

The development life cycle has a number of processes and tasks that the development community is involved in:

- Requirements
- Design
- Coding
- Testing

Testers are familiar with each of these stages in the context of system development and its relationship to the construction of tests.

The development life Cycle

- **Requirements**

The test description itself should contain the requirements for the test script.

- **Design**

Test Script design involves the construction of an executable model which represents the usage of a system. It is an executable model because the model contains enough information to allow the tester to work through the model and at any point unambiguously knows what they can do next.

- **Executable Models**

Executable models use 3 main constructs:

- Sequence, one action after another.
- Selection, a choice between one or more actions
- Iteration, a repeated sequence or selection

Sequence:

- The model consists of three main stages done one after the other; initialise, Body, and Terminate.

Selection:

- The model consists of a selection between 'Action 1' or 'Action 2' or 'Action 3'

Iteration:

- The model will iterate while condition C1 is satisfied.

Coding:

The coding of a test script refers to the writing of a test script.

Each test script should follow the path identified from the design and as such should be fairly easy to construct if a design has been produced.

Test Scripts are typically represented by a series of steps, each step being given an id or sequence number, an action and a result.

Testing:

Testers are aware of the importance of testing software.

The process of constructing tests and executing them should give testers an appreciation of the difficulties of program construction.

Note that:

- Test scripts and document results should run in line with test and acceptance processes.
- Required quality benchmarks or comparisons must be performed in readiness for acceptance testing.
- Organization/industry standards are adopted, where appropriate.
- Actual results to expected results should be compared on completion of each system unit, and completed result sheets.