|  |  |
| --- | --- |
|  |  |
| 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

1.2 Determine Software life cycle is based on work principles.

1.3 Prepare Test Plane based on Client Requirement

1.4 Schedule Test

1.5 Batch Testing

1.6 Confirm Test Result Criteria based on system requirement Documentation

**LO1. Prepare for Test**

* 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.

**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?
  1. **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**](http://istqbexamcertification.com/what-is-a-software-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. **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.

**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](http://www.webopedia.com/TERM/D/data.html) (The [accuracy](http://www.businessdictionary.com/definition/accuracy.html) and [consistency](http://www.businessdictionary.com/definition/consistency-principle.html) of stored [data](http://www.businessdictionary.com/definition/data.html)).

* 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

## Purposes

Database testing usually consists of a layered process, including the [user interface](http://en.wikipedia.org/wiki/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.

1. **White Box testing**

* White box 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](http://en.wikipedia.org/wiki/SQL) queries etc.
* It validates database tables, data models, database schema etc.
* It checks rules of [Referential integrity](http://en.wikipedia.org/wiki/Referential_integrity).
* It selects default table values to check on database consistency.

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.

* 1. **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. **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 behaviour 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.
* **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. **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:**   1. Read the specific objectives of this Learning Guide. 2. 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. 3. Accomplish the “Self-check 2”. 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 2). 5. If you earned a satisfactory evaluation proceed to “Information Sheet 3”. 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. | |

**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.