**TDM:-**

Test data management (TDM), a sister function of Test Environment Management, is the construction of test data sets that reliably represents an organization’s actual data so that IT teams (developers and testers) can effectively exercise software testing. Tests include Unit Testing, System Testing, Integration Testing, User Acceptance Testing, Performance Testing, and Security Testing.

**Common Types of Test Data**

There are four main types of test data and developers must construct a set of strategies and tools that address all data types. The type of test data that teams encounter usually falls into one of the following categories.

* **Synthetic data.** Helpful for new feature tests, this type of data comes from manual tests. While it alleviates security concerns, it does fall victim to human error. It also requires more knowledge on behalf of the tester regarding the test environment, data relationships, and the data itself.
* **Production data.** For more complete test coverage, production data is the best option. However, it can result in breaches of sensitive information, higher storage costs, and reduced agility. And it will need to be protected from unintentional modifications during the testing process.
* **Masked production data.** Data masking requires a staging environment with sufficient storage to maintain referential integrity after any kind of data transformation. This lengthens environment provisioning but also enables development teams to leverage real data with no risk. Masking data comes from nulling, anagramming, encryption, or substitution.
* **Subsets of production data.** These segments allow for more agility, reduced hardware requirement, and lower costs. They do not provide as comprehensive test coverage in comparison to full copies and may still risk exposure of sensitive data.

**Best Test Data Management Tools**

Data testing requires the best test data management tools for the job. Finding the best one for your needs comes down to a few criteria such as:

* Usability
* Value proposition
* Intuitive UI
* Available integrations
* Data reusability
* Data masking
* Targeted testing and data sets sizes

**TEM:-**

**Test Environment Management is the function of understanding your Test Environments\* and applying that knowledge, with appropriate governance and services to ensure that each testing environment is available for IT Projects, the DevTest community, and the overall software development lifecycle.**

**Key areas to set up in Test Environment**

For the test environment, a key area to set up includes

* System and applications
* Test data
* Database server
* Front-end running environment
* Client operating system
* Browser
* Hardware includes Server Operating system
* Network
* Documentation required like reference documents/configuration guides/installation guides/ user manuals.

The following are some of the most frequent tools (or functions) you'd need in order to establish a Test Environment Management capability:

* Environment Knowledge Base (CMDB)
* Test Environment Booking System
* Environment (Infrastructure) Orchestration Tool
* Build & Deployment Tools / Continuous Integration Tools
* Infrastructure Configuration Tools
* Service Virtualization
* Environment Shakedown Tools
* Service Support Tools (Incident & Change)
* Environment Health Dashboards