**Basic Concepts**

1. What is the difference between **verification** and **validation**?
2. What are the different levels of testing? (Unit, Integration, System, UAT)
3. What is the difference between functional and non-functional testing?
4. Explain the **SDLC** and **STLC**.
5. What is a test case? What are the key components of a test case?
6. What is the difference between a **test plan** and a **test strategy**?
7. What are severity and priority? Give examples.
8. What is regression testing, and when do you perform it?
9. What is exploratory testing? When would you use it?
10. What are positive and negative test cases? Give examples.

### ****Scenario-based****

1. If a defect is found in production, how will you handle it?
2. How will you test if a login page accepts only specific special characters?
3. You have very little time for testing — how do you prioritize?
4. If the developer says “it’s not a bug,” how will you proceed?
5. How would you test a search functionality?

ETL / Data Testing – Common Interview Questions

### ****Basic Concepts****

1. What is ETL testing? How is it different from database testing?
2. What are the key stages of ETL testing?
3. Explain the difference between **OLTP** and **OLAP**.
4. What are primary key, foreign key, and unique constraints?
5. What is data mapping in ETL?
6. What is the difference between **inner join**, **left join**, **right join**, and **full join**?
7. What is data validation vs data verification in ETL testing?
8. What are some common ETL tools you know? (e.g., Informatica, Talend, SSIS)
9. What is data transformation? Give an example.
10. What is a slowly changing dimension (SCD) in data warehousing?

### ****Scenario-based / SQL****

1. How do you verify that data is loaded correctly from source to target?
2. You find extra rows in the target — how will you troubleshoot?
3. Write a SQL query to find duplicate records in a table.
4. How do you compare row counts between source and target tables?
5. How do you validate if a column has only unique values?
6. How will you handle NULL values during ETL testing?
7. If data type mismatches occur during transformation, how will you fix it?
8. How do you check data integrity between multiple tables?

**As a software test engineer, my day-to-day activities include:**

* Understanding and reviewing data mapping documents shared by the development team
* Writing and reviewing test cases
* Uploading test cases into ALM after review completion
* Executing test cases and logging defects
* Performing regression testing and retesting
* Coordinating with developers and business analysts for issue resolution
* Updating test reports and status in JIRA
* For ETL testing – writing SQL queries to validate data from source to target
* Attending daily stand-up and team meetings

**Q 1. Difference between Verification and Validation:**

**Verification:**

Process of checking whether the product is being built correctly, according to requirements and design

|  |
| --- |
|  |

Are we building the product right?

**Type**: Static testing (no code execution)

**Example:**

Reviewing a data mapping document to ensure it matches the requirement.

**Validation:**

|  |
| --- |
|  |

Process of checking whether the right product has been built, and it meets the user's needs.

|  |
| --- |
|  |

Are we building the right product?

**Type**: Dynamic testing (code execution)

**Example**:   
Running the ETL job and checking if the data in the target matches the expected output

**Q 2. Different Levels of Testing**

**Unit Testing**

Testing individual components or modules of the application in isolation.

Done by developers to ensure each small part works as expected.

**Example**: Testing a single function that calculates interest in a banking app.

**Integration Testing**

Testing how different modules **work together** after integration.

Ensures data and control flow correctly between components.

**Example:** Testing how the “login module” connects with the “user profile” module.

**System Testing**

Testing the **entire application as a whole** against requirements.

Performed by **testers** to verify functional and non-functional aspects.

**Example:** Testing the full e-commerce site — login, search, add to cart, payment, order confirmation.

**User Acceptance Testing (UAT)**

Testing done by **end users or clients** to confirm the system meets business needs.

Usually the final phase before release.

**Example:** A business user checks if an ETL-generated sales report matches their expectations.

**Q 3. What is the difference between functional and non-functional testing?**

**Functional Testing**

Purpose: Verifies that the application’s features **work as expected** according to requirements.

Checks **what** the software should do.

Examples: Login works correctly, payment is processed.

Testing methods: Manual testing, automation of functional test cases, API testing.

**Non-Functional Testing**

Purpose: Checks **how well** the application works in terms of performance, usability, reliability, etc.

Checks **how well** the software works.

Examples: Application loads within 3 seconds, can handle 1,000 users, UI is user-friendly.

Testing methods: Load testing, stress testing, security testing, usability testing.

**Q 4. Explain the SDLC and STLC.**

**SDLC (Software Development Life Cycle):**

It’s the **process** used to plan, build, test, and deliver software in a structured way. or

SDLC is the step-by-step process for developing software from requirements gathering to maintenance.

**Phases of STLC:**

**Requirement Gathering (Requirement Analysis)**

* Understanding what the client or business needs.
* Example: Client wants an e-commerce website with payment integration.

**System** **Design**

* Creating architecture, database design, and UI mock-ups.
* Example: Designing how the website pages and database will look and work.

**Development**

* Writing code to build the application.
* Example: Developers code the product listing, cart, and payment modules.

**Testing**

* Verifying the application works as expected and is bug-free.
* Example: Testers check login, payment, and product search functions.

**Deployment**

* Releasing the application to users.
* Example: Making the website live for customers.

**Maintenance**

* Fixing issues and adding improvements after release.
* Example: Adding new features like coupons or offers.

### ****STLC (Software Testing Life Cycle)****

### STLC is the **step-by-step process followed during testing** to ensure software quality. Phases of STLC:

**Requirement Analysis**

* Understand testing requirements from the business documents (BRD, SRS, user stories).
* Identify what needs to be tested.

**Test Planning**

* Define the testing strategy, scope, tools, resources, timelines, and responsibilities.

**Test Case Design & Development**

* Write detailed test cases, test scripts, and prepare test data.

**Test Environment Setup**

* Prepare hardware, software, databases, and test environments needed to execute tests.

**Test Execution**

* Run test cases, compare actual vs expected results, and log defects in tools like JIRA or ALM.

**Defect Reporting & Tracking**

* Report bugs to developers and retest after fixes.

**Test Closure**

* Prepare final reports, summarize test coverage, defect details, and lessons learned.

**In one line for interviews:**  
"STLC is the process of planning, preparing, executing, and closing testing activities to ensure the software meets quality standards."

**Q 5. What is a test case? What are the key components of a test case?**

A **test case** is a set of conditions or steps written by a tester to check whether a specific feature or functionality of the software works as expected.

Example:  
To test the “Login” feature, a test case will describe steps like: open the login page, enter valid credentials, click login, and verify the home page is displayed.

**Key Components of a Test Case:**

**Test Case ID** – Unique identifier for the test case

**Test Case Description** – What is being tested

**Preconditions** – Conditions that must be met before executing the test

**Test Steps** – Detailed step-by-step instructions to perform the test

**Test Data** – Data required for the test

**Expected Result** – What should happen if the software works correctly

**Actual Result** – What actually happened during execution

**Status (Pass/Fail)** – Outcome of the test case

**Remarks/Comments** – Any additional notes or observations

**In one line:**  
"A test case is a written set of steps to check a specific function, including what to test, how to test it, and what result to expect."