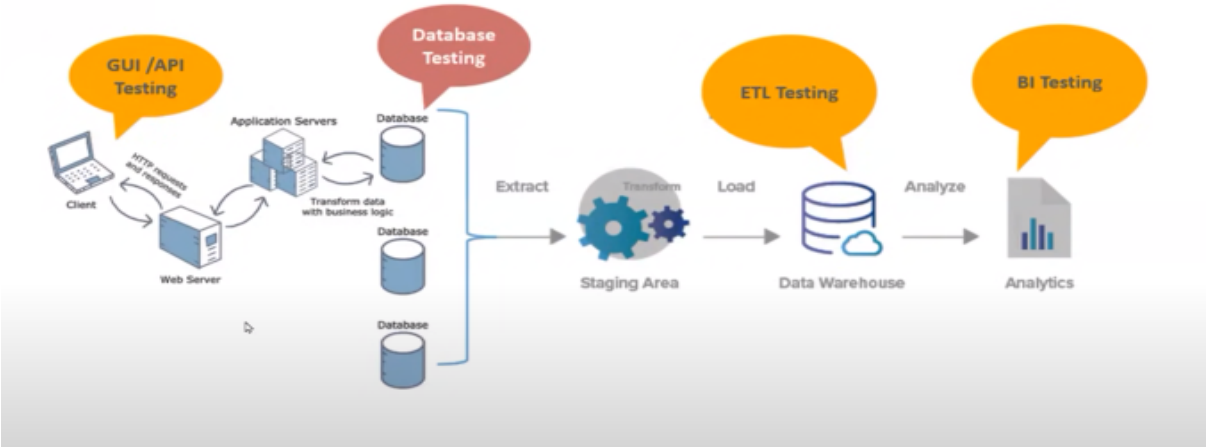
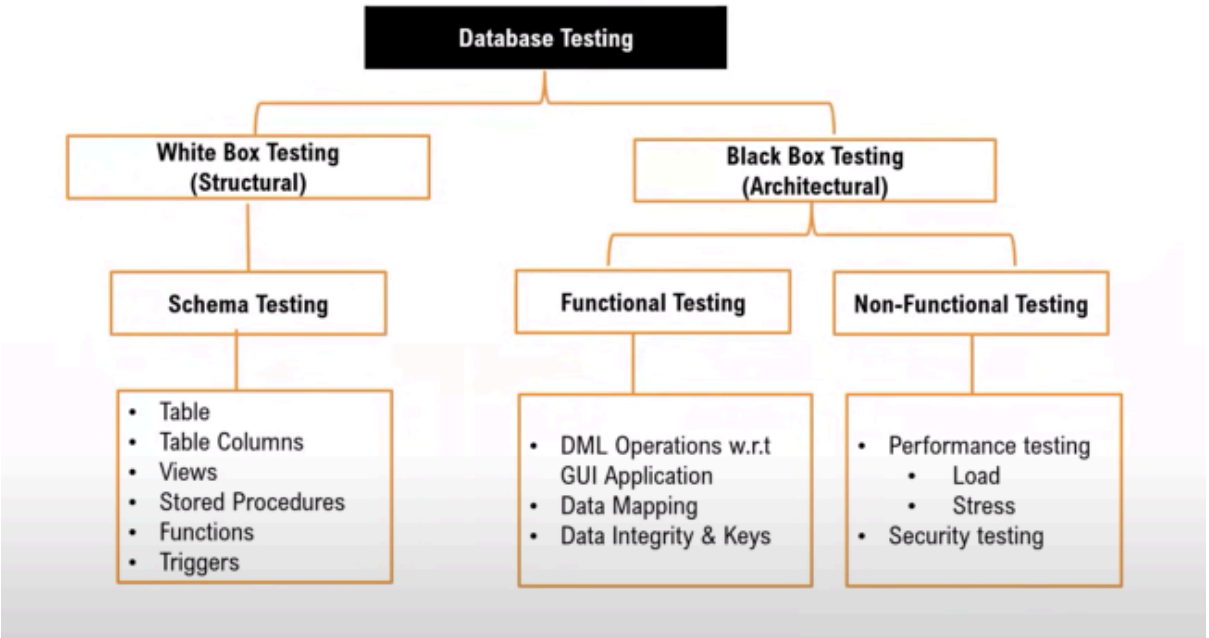


Database Testing :

- APIs interact with the database to perform CRUD operations (Create, Read, Update, Delete).
- Frontend operations are typically tested through the API.



Database Testing

- Focuses on backend testing, often referred to as the "heart of the application."

- Types of database testing:

    - **Objects**: Tables, Views, Indexes, Procedures.
    - **Data Testing**: Ensuring data validity and integrity.
- **White-Box Testing**: Requires understanding the database schema and structure.

Environment Setup

- Install and configure databases like MySQL, PostgreSQL, etc.

Schema Testing

- Verifies:
    - Data types and sizes.
    - NULL constraints.
    - Primary and foreign keys.

Example Test Cases:

- Check table presence in the database schema.
- Verify table name conventions.
- Validate the number of columns in a table.
- Ensure correct column names in a table.
- Check data types and sizes of columns.

Stored Procedure Testing

- Stored procedures contain multiple SQL statements executed together.
- Advantages:

- Simplifies transactions by grouping related SQL commands.

Example Stored Procedure Test Case:

- Verify if stored procedures exist in the database.
- Validate the output of stored procedures against equivalent SQL queries.
  - Example: A stored procedure retrieves 3 customer records for a specific city.
  - Run an equivalent SQL query manually to ensure the same output is returned.

Stored Procedure Example:

```
DELIMITER //
CREATE PROCEDURE GetCustomersByCity()
BEGIN
  SELECT * FROM Customers WHERE City = 'Berlin';
END //
DELIMITER ;
```

Types of Stored Procedures:

- Data Manipulation.
- Data Retrieval.
- Transaction Handling.

ETL Testing (Extract, Transform, Load)

- Ensures accurate data extraction, transformation, and loading into a data warehouse.
- **OLTP System**:
  - Customer data is first stored in a transactional database.
- Data from the OLTP system is aggregated into a single server for analysis.

BI (Business Intelligence) Testing

- Focuses on generating reports and analyzing data.
- Ensures data correctness and usability in BI tools.

Data Mapping

- Ensures data sent from the frontend is stored correctly in the backend.
- **Data Integrity and Keys**:
    - Validates how tables communicate using integrity constraints like primary and foreign keys.

Non-Functional Testing

- **Performance Testing**: Tests speed and scalability.
- **Load and Stress Testing**: Evaluates system behavior under heavy load.
- **Security and Penetration Testing**: Ensures data safety and protects against vulnerabilities.
- **Smoke Testing**: Verifies basic functionality.

Diagrams

1. Database Testing Overview

[Frontend] --(API)--> [Database Server] --> [Tables/Views/Procedures]

2. ETL Process

[Transactional Database] --> [ETL Process] --> [Data Warehouse] --> [BI Reports]

Stored Functions

- Stored function takes only one input and stored procedure takes input and as well as output
- You can call stored function through stored procedure but not vice versa

```
CREATE OR REPLACE FUNCTION add_numbers(a INTEGER, b INTEGER)
RETURNS INTEGER AS $$
BEGIN
   RETURN a + b;
END;
$$ LANGUAGE plpgsql;
```
