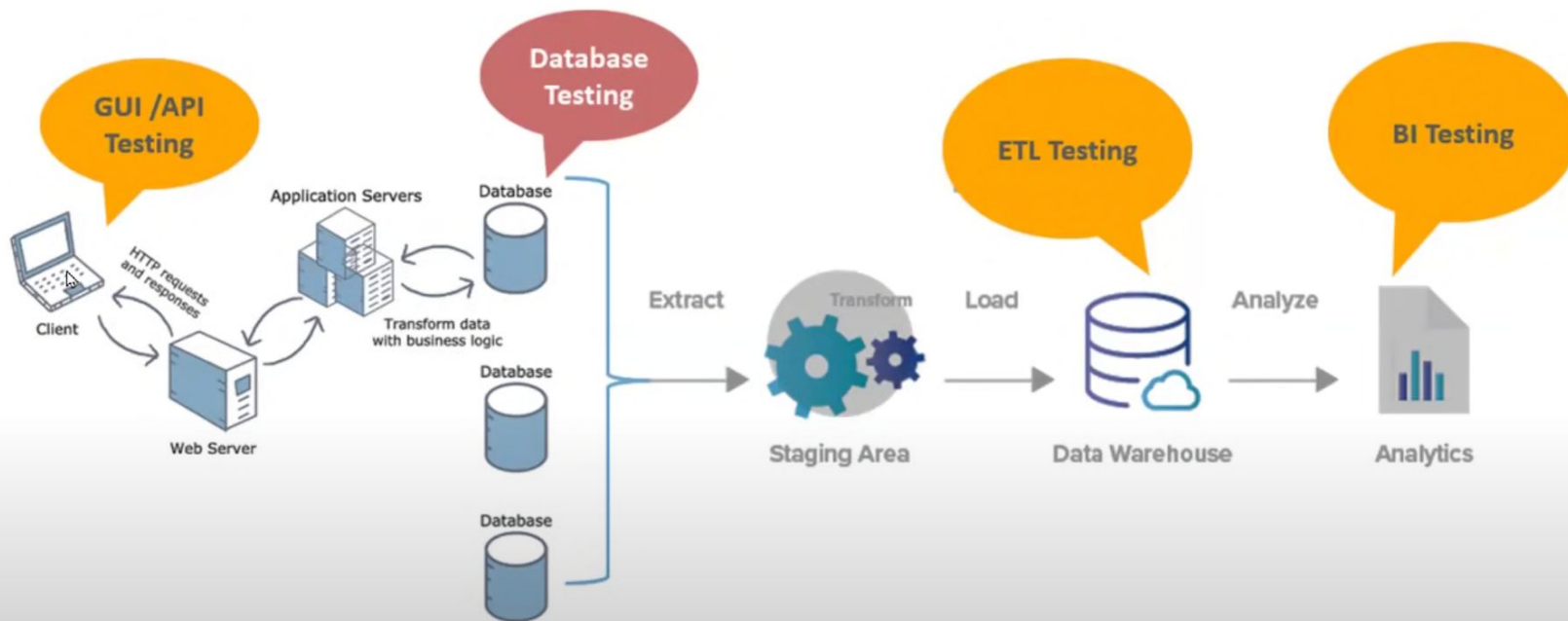


Database Testing



Database Testing

```
graph TD;
    A[Database Testing] --> B[White Box Testing<br/>(Structural)];
    A --> C[Black Box Testing<br/>(Architectural)];
    B --> D[Schema Testing];
    D --> E["• Table<br/>• Table Columns<br/>• Views<br/>• Stored Procedures<br/>• Functions<br/>• Triggers"];
    C --> F[Functional Testing];
    C --> G[Non-Functional Testing];
    F --> H["• DML Operations w.r.t<br/>GUI Application<br/>• Data Mapping<br/>• Data Integrity & Keys"];
    G --> I["• Performance testing<br/>• Load<br/>• Stress<br/>• Security testing"];
```

White Box Testing (Structural)

Schema Testing

- Table
- Table Columns
- Views
- Stored Procedures
- Functions
- Triggers

Black Box Testing (Architectural)

Functional Testing

- DML Operations w.r.t
GUI Application
- Data Mapping
- Data Integrity & Keys

Non-Functional Testing

- Performance testing
 - Load
 - Stress
- Security testing

Schema Testing - Table

Test Cases

- Check table presence in database schema
- Check table name conventions
- Check number of columns in a table
- Check column names in a table
- Check data type of columns in table
- Check size of the columns in a table
- Check nulls fields in a table
- Check column keys in a table

Stored Procedure Testing

Advantages

- **Reduce network traffic**

- Stored procedures help *reduce the network traffic between applications and MySQL Server*. Because instead of sending multiple lengthy SQL statements, applications have to send only the name and parameters of stored procedures.

- **Centralize business logic in the database**

- We can use the stored procedures to *implement business logic* that is reusable by multiple applications. The stored procedures help reduce the efforts of duplicating the same logic in many applications and make your database more consistent.

- **Make database more secure**

- The database administrator can grant appropriate privileges to applications that only access specific stored procedures without giving any privileges on the underlying tables.

Stored Functions Testing

What is Stored Function in SQL? Stored Procedure Vs Stored Function

- A stored function is a special kind stored program that returns a single value.
- **Stored Procedure Vs Stored Function**
 - The stored function must return a value but in Stored Procedure it is optional.
 - Even a procedure can return zero or n values.
 - Functions can have only input parameters for it whereas Procedures can have input or output parameters.
 - Functions can be called from Procedure whereas Procedures cannot be called from a Function.

Common Test Scenarios for Stored Function

- 1) Check Stored Function exist in database
- 2) Check Stored Function with valid input data
- 3) Check Stored Function handle exceptions when you pass invalid input data.
- 4) Check Stored Function returns results as expected.
- 5) Check Stored Function not used insert/update/delete operations.
- 6) Check Stored Function used only select statements
- 7) Check calling Stored Function from stored procedure.

Trigger Testing

What is Trigger?

- A trigger is a set of SQL statements that reside in a system catalog.
- It is a special type of stored procedure that is invoked automatically in response to an event.
- Each trigger is associated with a table, which is activated on any DML statement such as INSERT, UPDATE, or DELETE.
- A trigger is called a special procedure because it cannot be called directly like a stored procedure.
- The main difference between the trigger and procedure is that **a trigger is called automatically when a data modification event** is made against a table.
- A stored procedure **must be called explicitly**.

Types of Triggers in MySQL?

- We can define the maximum six types of actions or events in the form of triggers:
- **Before Insert:** It is activated before the insertion of data into the table.
- **After Insert:** It is activated after the insertion of data into the table.
- **Before Update:** It is activated before the update of data in the table.
- **After Update:** It is activated after the update of the data in the table.
- **Before Delete:** It is activated before the data is removed from the table.
- **After Delete:** It is activated after the deletion of data from the table.

Data Mapping Testing (CRUD)

- Testing database w.r.t frontend operations.
- Focused on:
 - Data Existence
 - Data Correctness
 - Data Completeness

Front-End UI

Register Account

If you already have an account with us, please login at the [login page](#).

Your Personal Details

* First Name

* Last Name

* E-Mail

* Telephone

Your Password

* Password

* Password Confirm

Newsletter

Subscribe ☐ Yes ☒ No

I have read and agree to the [Privacy Policy](#) ☐

Database Tables



Data Integrity Testing

Integrity Constraints

- SQL constraints are used to specify rules for data in a table.
- Constraints can be specified when the table is created with the CREATE TABLE statement, or after the table is created with the ALTER TABLE statement.
- **SQL Constraints**
 - **NOT NULL** - Ensures that a column cannot have a NULL value
 - **UNIQUE** - Ensures that all values in a column are different
 - **PRIMARY KEY** - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table
 - **FOREIGN KEY** - Uniquely identifies a row/record in another table
 - **CHECK** - Ensures that all values in a column satisfies a specific condition
 - **DEFAULT** - Sets a default value for a column when no value is specified

Courses Table

Column Name	Data Type	Constraint
COURSE ID	integer(2)	Primary Key
COURSE NAME	varchar(20)	Unique
DURATION	integer(2)	
FEE	integer(3)	Between 100 – 500 Dollars

Students Table



Column Name	Data Type	Constraint
SID	integer(5)	Primary Key
SNAME	varchar(20)	Not Null
AGE	integer(2)	Between 15-30 Years
DOJ (Date of Joined)	datetime	Default
DOC (Date of Completion)	datetime	
COURSE ID	integer(2)	References courses(COURSEID)