

AASSIGNMENT

1. WHAT IS ERROR, DEFECT, BUG AND FAILURE?

1 Error (Human Mistake)

An error is a mistake made by a developer or tester while writing code, designing a document, or creating test cases.

Example: A developer uses the wrong formula in a program due to misunderstanding the requirement.

2. Defect (Problem in Code)

A defect is a problem found in the software code or document due to an error.

It is discovered before the product goes live.

Found during development or testing.

3. Bug (Industry Term for Defect)

A bug is another name for a defect.

Commonly used by testers and developers.

When a defect is found during testing, it is usually reported as a "bug".

4. Failure (When Software Misbehaves)

A failure happens when the software does not work as expected during execution.

This occurs when a defect or bug is triggered during testing or in live use.

2. EXPLAIN THE DIFFERENCE BETWEEN FUNCTIONAL TESTING AND NON FUNCTIONAL TESTING?

Functional Testing is a type of software testing that checks whether the software system performs and behaves according to the requirements (i.e., what it is supposed to do).

It focuses on testing each function or feature of the application by providing appropriate inputs and checking the outputs.

Unit testing, Integration Testing, System Testing, User Acceptance Testing (UAT)

AASSIGNMENT

Non-Functional Testing

checks how well the software system performs under various conditions.
It does not test specific features or functions, but rather the quality attributes

speed ,performance ,usability, reliability, security

3. WHAT IS RDBMS?

RDBMS stands for Relational Database Management System.

It is a type of database management system (DBMS) that stores data in tables (also and allows the data to be accessed, managed, modified, and organized using Structured Query Language (SQL)

RDBMS Software:

MySQL, oracle sql server, postgres sql, sqllite

4. WHAT IS SQL?

SQL stands for Structured Query Language.

It is a standard language used to communicate with relational databases.

SQL is used to: store retrieve update, delete

and Manage data in Relational Database Management Systems (RDBMS) like MySQL, Oracle, SQL Server, PostgreSQL, etc.

5. WRITE SQL COMMAMDS?

DDL - Data Definition Language Create , alter, drop

DML - Data Manipulation language - Insert, update, select, delete

DCL – Data control language Commit, roll back etc

TCL- Transactional control language

6. WHAT IS JOIN?

In SQL, a JOIN is used to combine rows from two or more tables based on a related column between them — usually a foreign key and a primary key.

7. WRITE TYPE OF JOINS?

Four types is a joins

Inner join, Right join, Left Join, Full join

AASSIGNMENT

8. HOW MANY CONSTRAINT AND DESCRIBES IT SELF?

There are 6 main types of constraints in SQL

- 1.**Not null** Ensures that a column cannot have a NULL value (must contain data).
- 2.**unique** Ensures that all values in a column are different (no duplicates).
- 3.**primary key** A combination of NOT NULL + UNIQUE. It uniquely identifies each **row** in the table
- 4.**foreign key** Enforces a relationship between two tables. It links one table to the primary key of another.
- 5.**check** Ensures that values in a column meet a specific condition.
- 6.**default** set a default value for a column if no value provided during insertion

9. DIFFERENCE BETWEEN RDBMS VS DBMS?

Definition:

- **DBMS:** Software to store, manage, and retrieve data.
- **RDBMS:** A type of DBMS that stores data in tables with relationships between them.

Data Structure:

- **DBMS:** Data stored as files, records, or objects.
- **RDBMS:** Data stored in tables (rows and columns).

Relationships:

- **DBMS:** Does not support relationships between data.
- **RDBMS:** Supports relationships using keys (primary key, foreign key).

Normalization:

- **DBMS:** Generally does not support normalization.
- **RDBMS:** Supports normalization to reduce data redundancy.

Data Integrity:

- **DBMS:** Limited support for data integrity and consistency.
- **RDBMS:** Strong support through constraints (like primary key, foreign key).

AASSIGNMENT

Examples:

- **DBMS:** File system, XML, NoSQL databases.
- **RDBMS:** MySQL, Oracle, SQL Server, PostgreSQL.

Query Language:

- **DBMS:** May not support SQL or has limited query language.
- **RDBMS:** Supports SQL (Structured Query Language).

Transaction Support:

- **DBMS:** Limited or no support for transactions.
- **RDBMS:** Supports ACID transactions (Atomicity, Consistency, Isolation, Durability).

Scalability:

- **DBMS:** Used for smaller or simpler applications.
- **RDBMS:** Suitable for large and complex applications.

Use Cases:

- **DBMS:** Simple data storage or small-scale apps.
- **RDBMS:** Enterprise-level apps with complex data relationships.

10. WRITE QUARY TO CREATE THE TABLE IN STRUCTURED QUERY LANGUAGE?

Here is the basic syntax and example of the CREATE TABLE command in SQL

```
CREATE TABLE table_name (  
    column1 datatype constraint,  
    column2 datatype constraint,  
    column3 datatype constraint,
```

```
);
```

11. WRITE QUARY TO INSERT DATA INTO TABLE?

Here is the basic syntax of the insert into TABLE command in SQL

```
INSERT INTO table_name (column1, column2, column3, ...)  
VALUES (value1, value2, value3, ...);
```

12. WRITE QUARY UPDATE DATA INTO TABLE WITH VALIDATIONS?

Here is the basic syntax of the update data into table command in SQL

AASSIGNMENT

```
UPDATE table_name  
SET column1 = value1, column2 = value2, ...  
WHERE condition;
```

13. WRITE QUARY DELETE DATA FROM TABLE WITH VALIDATIONS?

Here is the basic example of the delete data from table command in SQL

```
DELETE FROM Students  
WHERE Age < 20;
```

14. WRITE QUARY TO INSERT NEW COLUMN IN EXISTING TABLE?

Here is the basic syntax of the insert new column existing from table command in SQL

```
ALTER TABLE Students  
ADD Email VARCHAR(100);
```

15. WRITE QUARY TO DROP TABLE AND DATABASE?

Here is the syntax of the drop table
DROP TABLE table_name;

16. CREATE TWO TABLES NAMED SELLER AND PRODUCT APPLY FOREIGN KEY IN PRODUCT TABLE FETCH FROM BOTH TABLE USING DIFFERENTS JOINS?

Create table

```
CREATE TABLE SELLER (  
    seller_id INT PRIMARY KEY,  
    seller_name VARCHAR(100),  
    location VARCHAR(100)  
);
```

Create table product with foreign key

```
CREATE TABLE PRODUCT (  
    product_id INT PRIMARY KEY,  
    product_name VARCHAR(100),  
    price DECIMAL(10,2),  
    seller_id INT,  
    FOREIGN KEY (seller_id) REFERENCES SELLER(seller_id));
```

AASSIGNMENT

Insert data

```
INSERT INTO SELLER VALUES (1, 'John Traders', 'Mumbai');
INSERT INTO SELLER VALUES (2, 'Sara Distributors', 'Delhi');
INSERT INTO SELLER VALUES (3, 'Metro Mart', 'Chennai');

INSERT INTO PRODUCT VALUES (101, 'Laptop', 45000.00, 1);
INSERT INTO PRODUCT VALUES (102, 'Smartphone', 25000.00, 2);
INSERT INTO PRODUCT VALUES (103, 'Tablet', 18000.00, 1);
INSERT INTO PRODUCT VALUES (104, 'Keyboard', 1200.00, NULL);
```

Inner join

```
SELECT P.product_name, P.price, S.seller_name
FROM PRODUCT P
INNER JOIN SELLER S ON P.seller_id = S.seller_id;
```

left join

```
SELECT P.product_name, P.price, S.seller_name
FROM PRODUCT P
LEFT JOIN SELLER S ON P.seller_id = S.seller_id;
```

Right join

```
SELECT P.product_name, P.price, S.seller_name
FROM PRODUCT P
right JOIN SELLER S ON P.seller_id = S.seller_id;
```

Full join

```
SELECT P.product_name, P.price, S.seller_name
FROM PRODUCT P
FULL OUTER JOIN SELLER S ON P.seller_id = S.seller_id;
```

17. WHAT IS API TESTING?

API stands for Application Programming Interface.

It is a set of rules and protocols that allow different software applications to communicate with each other.

18. TYPES OF API TESTING?

API testing is a type of software testing that focuses on verifying that application programming interfaces (APIs) work as expected. Below are the main types of API testing

- 1.Functional Testing
- 2.Validation testing
- 3.Load testing
- 4.Performance testing

AASSIGNMENT

- 5.Security testing
- 6.Error/negative testing
- 7.unit testing
- 8.integration testing
- 9.Regression testing
- 10.Run time testing/Error detection

19. WHAT IS RESPONSIVE TESTING?

Responsive Testing is a type of testing done to check how a website or web application adjusts and behaves on different:

Screen sizes (mobile, tablet, desktop)

Devices (iPhone, Android, iPad, laptops, etc.)

Resolutions (720p, 1080p, 1440p, etc.)

AASSIGNMENT