**Friday Test**

**1. What is DevOps?**

DevOps brings the development and operations teams together to make software delivery faster and better. It uses tools and automation to simplify processes like testing, monitoring, and deployment, ensuring the application is of good quality and delivered quickly. It covers everything from building the app to running and maintaining it.

**2. Why DevOps?**

DevOps improves collaboration between development and operations teams, automates repetitive tasks, and ensures faster and more efficient delivery of software updates.

**3. What is the need for DevOps?**

* Faster software delivery.
* Improved communication.
* Reliable releases with fewer bugs.
* Automation of manual processes.

**4. What are DevOps tools?**

* CI/CD Tools: Jenkins, GitHub Actions.
* Configuration Management: Ansible, Puppet.
* Containerization: Docker.
* Monitoring: Prometheus, Grafana.
* Cloud: AWS, Azure, GCP.
* Version Control: Git.

**5. Difference between break, continue, and pass?**

* break: Stops the loop entirely.
* continue: Skips the current step in the loop and goes to the next one.
* pass: Does nothing; it's just a placeholder in the code.

**6. Difference between remove, delete, and pop in Python**

* remove: Removes a specific value from the list.
* del: Deletes an element by index or the entire list.
* pop: Removes and returns an element by index.

**Example Program**:

list = [1, 2, 3, 4]  
list.remove(2)  
del list[1]  
popped = list.pop(0)  
print(list)

**7. Difference between append and extend in Python**

* append: Adds a single element to the end of the list.
* extend: Adds all elements of an iterable to the list.

**Example**:

list = [1, 2]

list.append([3, 4]) # Adds a list as a single element

list.extend([5, 6]) # Adds elements individually

print(list) # [1, 2, [3, 4], 5, 6]

**8. Print the element in the array with negative index**

arr = [10, 20, 30, 40, 50]

print(arr[-2]) # Prints the second last element: 40

**9. What is a Lambda function?**

A lambda function is an anonymous (nameless) function in Python. It's used for short, one-line functions.  
**Example**:

square = lambda x: x \*\* 2

print(square(5)) # Output: 25

**10. What is Cloud? Top 10 Cloud Providers**

The cloud refers to servers accessed over the internet and the software/services they provide.  
Top 10 Providers: Top 10 Cloud Providers

1. Amazon web service(36%)
2. Microsoft Azure(25%)
3. Google cloud platform(11%)
4. Alibaba Cloud(4%)
5. Salesforce(3%)
6. IBM(2%)
7. Oracle(2%)
8. Tencent cloud(2%)
9. Digital ocean(1%)

AWS, Microsoft Azure, Google Cloud, IBM Cloud, Oracle Cloud, Salesforce, Alibaba Cloud, VMware, SAP, Tencent Cloud.

**10.what is Cloud Computing? Types**

It is the distributed Collection of servers that are hosted on the internet, instead of local servers.

**Two Types of Cloud Computing Models:**

1.Service Model: This defines what services are provided by the cloud, like computing power, storage, or software. There are different models based on what the cloud offers (like IaaS, PaaS, SaaS).

2.Deployment Model: This defines how cloud resources are made available to users. There are four types:

Private Cloud: Resources are used by only one organization.

*Example*: A college website for checking attendance.

Public Cloud: Resources are shared with everyone by the cloud provider.

*Example*: Social media platforms like Facebook.

Hybrid Cloud: Combines both private and public clouds, allowing data and apps to move between them.

*Example*: Netflix, using both public and private resources. Community Cloud: Shared by multiple organizations with similar goals.

*Example*: Two research organizations working together on a project

**12. Different levels of cloud storage**

**13. Architecture of Service Model with Examples**

IaaS: Rent computing resources (virtual servers and storage) and manage them yourself. Example: AWS EC2.

PaaS: Build and run apps on a platform without managing servers. Example: Google App Engine.

SaaS: Use software directly from the cloud with no installation. Example: Google Workspace.

**14. Deployment Models**

Public Cloud: Shared, external resources that are available to anyone (e.g., AWS, Azure).

Private Cloud: Exclusive resources used by one organization, either on-premises or hosted externally.

Hybrid Cloud: A mix of both public and private clouds, offering flexibility.

Community Cloud: Shared resources for a group of organizations with similar needs or interests.

**15. Differences between AWS, Azure, and GCP**

1. AWS (Amazon Web Services):AWS is the largest cloud provider and has the most services.It’s great for businesses that need a lot of flexibility and different tools for everything, from computing to storage and databases.AWS is used by many startups, big companies, and tech businesses because of its huge range of options and global reach.
2. Azure (Microsoft Azure):Azure is popular with companies that already use Microsoft products, like Windows or Office.It works well for hybrid setups, where you mix your on-site servers with the cloud.Azure is often chosen by businesses that need strong integration with existing Microsoft tools and software, making it a good fit for enterprise environments.
3. GCP (Google Cloud Platform):GCP focuses heavily on data and machine learning, making it a good choice for companies that work with a lot of data or want to build advanced AI models.It’s known for its excellent tools for analytics and big data processing, like BigQuery and TensorFlow.GCP is often favored by tech companies, data scientists, and businesses focusing on big data and AI.

* AWS offers the most variety and is used by many businesses.
* Azure is best for companies that already use Microsoft products.
* GCP shines when it comes to data-heavy projects and AI work.

**16. Python Program to Print Your Details 100 Times**

for \_ in range(100):

print("Name: AMULYA: Designation: Trainee, Technology: DevOps")

**17. Agile vs Waterfall Models**

**Agile**: Iterative, flexible, customer-focused.

**What**: Agile is a flexible and iterative approach to project management, where work is done in small, repeated cycles (called sprints).

**How** : The project is broken into smaller parts, and each part is completed step-by-step. After each step, the team gets feedback and makes changes based on that.

**When:** Agile is great for projects that are likely to change over time, where you need to adapt and improve quickly based on customer feedback.

**Example**: Building a website where features are added and updated regularly based on user feedback.

**Waterfall:**

**What** : Waterfall is a more rigid, step-by-step approach to project management, where each phase must be completed before moving to the next one.

**How**: The project follows a clear and fixed sequence of steps. Once a phase (like planning or design) is finished, you can't go back to change anything.

**When**: Waterfall is best for projects that have a clear and fixed plan with little need for changes, like building a bridge or developing a software with a defined set of features.

**Example**: Developing a simple software application where all the requirements are known in advance.

**Waterfall**: Sequential, rigid, suitable for well-defined projects.

**18. Arithmetic and Relational Operators with Examples**

**Arithmetic**: +, -, \*, /.

Addition: 5 + 2 = 7

Subtraction: 5 - 2 = 3

Multiplication: 5 \* 2 = 10

Division: 5 / 2 = 2.5

**Relational Operators**

Equal (a == b): False (5 is not equal to 2)

Not Equal (a != b): True (5 is not equal to 2)

Greater Than (a > b): True (5 is greater than 2)

Less Than (a < b): False (5 is not less than 2)

Greater or Equal (a >= b): True (5 is greater than or equal to 2)

Less or Equal (a <= b): False (5 is not less than or equal to 2)

**19. Comparison: Set, List, Tuple, Dictionary**

List: Ordered, changeable, allows duplicates. Example: [1, 2, 3].

Tuple: Ordered, unchangeable, allows duplicates. Example: (1, 2, 3).

Set: Unordered, changeable, no duplicates allowed. Example: {1, 2, 3}.

Dictionary: Ordered, changeable, stores key-value pairs. Example: {'a': 1, 'b': 2}

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Feature** | **List** | **Tuple** | **Set** | **Dictionary** |
| **Order** | Maintains order | Maintains order | Does not maintain order | Maintains order (Python 3.7+) |
| **Mutability** | Mutable (can be changed) | Immutable (cannot be changed) | Mutable (can add/remove items) | Mutable (can add/remove key-value pairs) |
| **Duplicates** | Allows duplicates | Allows duplicates | Does not allow duplicates | Keys are unique; values can be duplicated |
| **Access** | By index (e.g., my\_list[0]) | By index (e.g., my\_tuple[0]) | No index; check existence  (e.g., item in my\_set) | By key (e.g., my\_dict['key']) |
| **Use Case** | For ordered, modifiable items | For fixed, ordered data | For unique, unordered data | For mapping keys to values |
| **Example** | [1, 2, 3, 3] | (1, 2, 3, 3) | {1, 2, 3} | {'a': 1, 'b': 2} |

**20. Phases of Software Development Life Cycle (SDLC)**

There are 5 steps involved in the SDLC process :

1. Requirement Analysis
2. Design
3. Development
4. Testing
5. Maintainence

The Software Development Life Cycle (SDLC) is a structured process used for designing, developing, testing, and deploying software. It ensures the software meets quality standards and user requirements while being delivered efficiently. The SDLC consists of several stages, each with specific tasks and goals, providing a clear roadmap for the software development process.

Key Stages of SDLC:

1. Requirement Analysis: Understanding and documenting what the users need and expect from the software.

3. Design: Developing system architecture, interfaces, and data flow diagrams to plan the solution.

4. Development: Writing and coding the software based on design specifications.

5. Testing: Verifying that the software works as intended, identifying and fixing any bugs.

6. Deployment: Releasing the software for use in the production environment.

7. Maintenance: Providing ongoing support, updates, and fixes to keep the software functional and relevant.

**21. Database and DBMS**

Database: It is an application which stores the collection of data.

DBMS: Database Management system is a software that provides an interface to interact with databases. It allows users to create, manage, and manipulate data efficiently. A DBMS ensures the data is stored securely, can be accessed quickly, and maintains data integrity and consistency.

We have 2 types of DBMS they are

1.RDBMS-Relational DBMS

2.Non RDBMS-Non Relational DBMS

**22. DDL and DML Commands**

Types of Commands :

1. DDL—DATA DEFINATION LANGUAGE
2. DML—DATA MANIPULATION LANGUAGE

DDL : DDL commands are used to define or modify the structure of a database, table, or other database objects.

1. CREATE—to create a d/b, tables
2. ALTER—update,add a row to existing table/column
3. DROP—delete the records from the database
4. TRUNCATE—remove the records from the table
5. RENAME—rename the table or records in the table

DML : DML commands are used to manipulate data within tables.

1. INSERT—inserts datainto table
2. UPDATE—update the existing data with in the table
3. DELETE—delete the records from the database of a table
4. RETRIVE,CALL,EXPlAIN

**23. what are clauses and explain with example..?**

1. SELECT

* Specifies which columns of data to retrieve from the database.

Example:

* SELECT column1, column2 FROM table\_name;

2. FROM

Specifies the table(s) from which to retrieve the data.

Example:

SELECT column1 FROM table\_name;

3. WHERE

Filters records based on a given condition.

Example:

SELECT column1 FROM table\_name WHERE column2 = 'value';

4. ORDER BY

Sorts the result set by one or more columns, either in ascending (ASC) or descending (DESC) order.

Example:

SELECT column1 FROM table\_name ORDER BY column2 DESC;

5. GROUP BY

Groups rows that have the same values into summary rows, often used with aggregate functions like COUNT(), AVG(), SUM(), etc.

Example:

SELECT column1, COUNT(\*) FROM table\_name GROUP BY column1;

6. HAVING

Filters records that are grouped by the GROUP BY clause. It’s like WHERE, but used for aggregated data.

Example:

SELECT column1, COUNT(\*) FROM table\_name GROUP BY column1 HAVING COUNT(\*) > 5;

7. JOIN

Purpose: Combines rows from two or more tables based on a related column between them. There are different types of joins: INNER JOIN, LEFT JOIN, RIGHT JOIN, etc.

Example (INNER JOIN):

SELECT column1, column2 FROM table1 INNER JOIN table2 ON table1.id = table2.id;

8. INSERT INTO

Adds new records to a table.

Example:

INSERT INTO table\_name (column1, column2) VALUES (value1, value2);

9. UPDATE

Modifies existing records in a table.

Example:

UPDATE table\_name SET column1 = value1 WHERE column2 = 'value';

10. DELETE

Deletes records from a table based on a condition.

Example:

DELETE FROM table\_name WHERE column1 = 'value';

11.LIMIT

Limits the number of records returned by a query.

Example:

SELECT column1 FROM table\_name LIMIT 10;

12. DISTINCT

Removes duplicate records from the result set.

Example:

SELECT DISTINCT column1 FROM table\_name;

13. IN

Specifies a set of values to match against a column in the WHERE clause.

Example:

SELECT column1 FROM table\_name WHERE column2 IN ('value1', 'value2');

14. BETWEEN

Filters results that fall within a specified range.

Example:

SELECT column1 FROM table\_name WHERE column2 BETWEEN 10 AND 20;

15. LIKE

Filters results based on a pattern match in a column.

Example:

SELECT column1 FROM table\_name WHERE column2 LIKE 'A%';

**24. Joins in SQL with Examples**

Joins are used with selected statement , used to retrive the data from multiple table from same database. Fetching the records from different tables will be very easy.

There are 3 types of MySQL Joins:

1.INNER JOIN

2.LEFT JOIN

3.RIGHT JOIN

INNER JOIN

In order to written all the rows from multiple tables where the join condition is satisfied

This is the most commonly used joins in MySQL.

**LEFT JOIN**

Return all row from the left hand side table and all the rows from right hand side by satisfying the join condition.

**SELF JOIN**

The data/rows in the table are connected/joined with the same data/rows in the same table

**RIGHT JOIN**

Return all rows feom the right hand table with right hand table rows on to the right table by satisfying the Join condition.

**CROSS JOIN**

Return all the records from both the tables

**25. Create a Trigger**

DELIMITER //

CREATE TRIGGER age\_verify

BEFORE INSERT ON newjoinerss

FOR EACH ROW

BEGIN

IF new.emp\_age < 0 THEN

SET new.emp\_age = 0;

END IF;

END //

DELIMITER ;