**Assignment 20/12/2024**

1. **What is DevOps?**

DevOps can be defined in many ways such as:

* It is a methodology
* It is a set of tools
* It is used to automate things/tasks
* It is a process

***“DevOps can be defined as the process of delivering a product/project by ensuring automation in place and ensuring the quality with continuous monitoring and continuous testing”***

1. **Why DevOps?**

For any IT industry there are two teams namely the **development team** andthe **operations team**

* The task of the development team is to develop the product or project using the technologies or tools provided by the client and to push to the GitHub
* Similarly, the task of the operations team is to extract the code pushed and perform the release operation

Now, assume that the development team has built the code and pushed onto the GitHub server, the task of the operations team is to extract the code from the server and perform the release operation

Assume that the both teams have done their work properly but due to some reason it is not accessible for the user. This develops a misconception between both the teams. The development team says that the push operation is performed properly and similarly the operations team says that the release operation is performed. This develops a misunderstanding between the two teams. So, in order to bridge the gap between the two teams the DevOps is present.

* They act as a bridge between both the teams
* This is used to deliver the project on time
* This is used for automation and deliver the project without any errors
* This reduces the time for updating the versions of the product or project

1. **What is need of DevOps?**

Need for DevOps:

* Used for automating tasks
* Helps in reducing the time to update the versions
* Used to deliver the project/product on time
* Used to bridge the gap between the development and operations team
* Used for eliminating the errors

1. **What are the DevOps tools?**

The DevOps includes a various process such as:

* Planning/coding, building, testing, integration, deployment, operations and monitoring
* The above mentioned are the stages of the DevOps
* For performing the above operations there are wide variety of tools available.
* Some of them are as mentioned below:

1. **Planning:** Git or JIRA
2. **Building:** Maven, Gradle, Apache ANT
3. **Testing:** Selenium
4. **Integration:** Jenkins
5. **Deployment:** Docker, Kubernetes
6. **Operations:** Ansible
7. **Monitoring:** Terraform
8. **Difference b/w break continue and pass?**

Differences Between break, continue and Pass:

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Break** | **Continue** | **Pass** |
| **Definition** | Break can be defined as an inbuilt keyword that is used in the loops to exit a loop (for, while) when a certain condition is satisfied | Continue can be defined as the built-in keyword used in the loops to skip an iteration in the loops when a condition is matched | Pass statement as the name says it executes the next statement without considering about the condition |
| **Example** | for i in range(5):  if i = 3:  break  statement1  statement2 | for i in range(5):  if i = 3:  continue  statement1  statement2 | for i in range(5):  if i = 3:  pass  statement1  statement2 |
| **Output** | The above example exits the code if the value of i is 3 and doesn’t execute the next lines of code | The above example doesn’t exit the code instead it skips the next statement of code when the value of the i is 3 | For the above example the next statements after the pass statement are executed without considering the condition |

**6. d/w remove, delete, pop and write an example program in**

**python to demonstrate 3 of them.?**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Remove** | **Delete** | **Pop** |
| **Definition** | This method is used to remove an item in the list.  We can also remove the item by specifying the value | This method is used to delete the list completely | This method is used to remove the elements from the list.  By default, this method removes the last element from the list |
| **Example** | lst = [1,”hi”,3.3,9]  lst.remove(3.3)  print(lst) | lst = [1,”hi”,3.3,9]  del list  print(lst) | lst = [1,”hi”,3.3,9]  lst.pop()  print(lst) |
| **Output** | For the above example the element 3.3 is removed from the list and the list with remaining elements are printed. Here we can only specify the value | for the above example the list is completely deleted and while printing the list we get an error | for the above example the element at the last index is deleted by default. We can also specify the index value that we wanted to delete. Here we can only specify the index of the item |

**7. D/w append and extend?**

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Append** | **Extend** |
| **Definition** | This method is used to add the elements in the list. By default, it adds the elements at the last index of the list. This is used to add single element to the list | This is the method used to join two or more lists. And this is used for adding multiple items to a list |
| **Example** | fruits = [“apple”,”banana”,”grapes”]  a = fruits.append(“mango”) | fruits1 = [“apple”,”banana”]  fruits2 = [“mango”,”grapes”]  a = fruits1.extend(fruits2) |
| **Output** | For the above example the output is it prints the list with the all the elements including mango | For the above example the output is that the items in the fruits2 are added to the first list i.e., fruits1 |

**8. Write a python program to print the element in the array with**

**negative elements (ex : print the element which is present in -2 positions)?**

lst = [5,8,”apple”,9.5,4,”mango”,hyd,”grapes”]

sec\_pos = lst[-2]

print(sec\_pos)

**9. Explain about LAMDA function?**

**Lambda Function:** A lambda function in python is defined as small, anonymous function that is defined using the *lambda* keyword

* It takes any number of arguments and contains a single expression that needs to be evaluated
* The basic syntax of lambda function is :

lambda arguments : expression

* Example:

add\_ten = lambda x : x + 10

print(add\_ten(5))

**10.What is cloud? explain top 10 cloud providers?**

**Cloud:** cloud can be defined as a vast connection of servers accessible via internet that store, manage and process data.

**Top 10 cloud providers:**

* AWS (Amazon Web Services)
* Microsoft Azure
* GCP (Google Cloud Platform)
* Alibaba cloud
* IBM cloud
* Oracle
* Digital ocean
* Salesforce
* VMware
* ServiceNow

**11. what is cloud computing and explain types?**

Cloud computing can be defined as the delivery of services over the internet such as servers, storage, databases, networking and software without the use of physical infrastructure.

**The cloud computing is of two types:**

* Service mode
* Deployment mode

**The service mode is of four types namely:**

* IAAS (Infrastructure as a service)
* PAAS (Platform as a service)
* SAAS (Software as a service)
* FAAS (Function as a service)

**The deployment mode is of four types namely:**

* Public cloud
* Private cloud
* Hybrid cloud
* Community cloud

**12. what are the different levels of cloud storages?**

Cloud storage solutions are divided into different **tiers** or **levels**, based on performance, cost, and use case.

**1. Hot Storage**

* **Definition**: Hot storage is designed for frequently accessed data that requires low latency and high throughput (amount of data that can be transferred over a given period of time). It is ideal for real-time applications, active databases, and websites that require constant data access.
* **Example**: AWS S3 Standard, Microsoft Azure Blob Storage

**2. Cool Storage**

* **Definition**: Cool storage is for data that is infrequently accessed but needs to be stored for long-term retention. It is cheaper than hot storage but offers slightly higher latency for access.
* **Example**: AWS S3 Glacier Deep Archive, Microsoft Azure Blob Storage

**3. Cold Storage**

* **Definition**: Cold storage is designed for archival and infrequently accessed data. This is the most cost-effective tier, but access times are slower, and retrieval may take hours or days.
* **Example**: AWS S3 Glacier, Google Archive Storage, Microsoft Azure Blob Storage (Archive tier).

**4. Archive Storage**

* **Definition**: Archive storage is used for long-term data storage where retrieval is rare and typically not urgent. It is optimized for data that may be kept for regulatory or compliance reasons.
* **Example**: AWS S3 Glacier, Google Coldline, Microsoft Azure Archive Storage.

**13. Explain the architecture of service model with real time examples?**

The service mode provides services such as IAAS, PAAS, SAAS and FAAS

**IAAS (Infrastructure as a service):** Here, in simple terms the infrastructure is like the servers, storage, networking and virtualization. These are all provided/managed by the cloud provider (AWS). You only need to manage the OS, middleware, runtime and application data

**Ex: Amazon Web Services (AWS)**: AWS EC2 (Elastic Compute Cloud) is an IaaS offering where users can create and manage virtual machines, storage, and networking (VPC).

**PAAS (Platform as a service):**  Coming to the PAAS the cloud provider manages all the IAAS and OS, middleware, and runtime. The objective of the client is to manage the application

**Ex: Microsoft Azure App Service:** A fully managed platform for building and deploying applications using various programming languages like .NET, Node.js, Python, etc.

**SAAS (Software as a service):** This is the service that manages all the IAAS, PAAS and applications. The objective of the client is to manage or develop the data and configurations of the software

**Ex: Google Workspace (formerly G Suite):** Includes applications like Gmail, Google Docs, Google Drive, etc., which are fully managed by Google. Users just use the services through the web interface.

**FAAS (Function as a service):** This is a cloud computing service that allows developers to run small codes called functions. This is a part of serverless computing

**Ex:** Assume we are a travel agency. Your need is when a user books a ticket from your website you want to send an email to the user with the booking details. Instead of setting up a separate server for this task we can use Faas to simplify this process. You write a function and deploy it into a Faas platform such as AWS lambda. When a trigger occurs in the cloud this automatically sends the email with details to the users. You are billed for milliseconds or seconds that the function took to execute; this makes the cost effective

**14. Explain deployment model?**

The deploymentmodel in cloud computing refers to the way cloud services and infrastructure are made available and how they are managed and accessed by users. It defines the location, ownership, and the level of control over the cloud resources

**The deployment mode is of four types namely:**

* **Public** **cloud**: it is a type of cloud computing service in which the resources such as servers, storage and applications are shared among the multiple users.

**Ex:** Netflix uses AWS platform for storing and streaming the movies and web series. When a new movie is launched the AWS automatically scales up the resources for handling a large amount of load

* **Private** **cloud**: it is a cloud computing service in which the resources are shared with a single organization. This increases security, control and privacy for the organization.

**Ex:** this is used in healthcare, banking and more

* **Hybrid** **cloud**: it is cloud computing service and a combination of private and public cloud.

**Ex:** if we take an e-commerce application as an example it uses both the public and private clouds. For scalability it uses the public cloud and for payments and other sensitive information it uses private cloud

* **Community** **cloud**: as the name says it is a cloud computing service that is shared among a group or community. The infrastructure is shared among a group of people with similar goals and compliance

**Ex:1.** A group of hospitals share the cloud for accessing the patient’s data. It is only accessible to the authorized people

**2.** A group of universities form a community cloud for sharing resources for academic and research purposes

**15. Mention few differences between AWS, MICROSOFT AZURE,**

**and GCP?**

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature** | **AWS** | **Microsoft Azure** | **Google Cloud Platform (GCP)** |
| **Year launched** | 2006 | 2010 | 2008 |
| **Market Share** | Largest | Second-largest | Third-largest |
| **Compute Services** | EC2, Lambda, ECS | Virtual Machines, Functions | Compute Engine, Cloud Functions |
| **Key Strengths** | Scalability, variety of services, global presence | Hybrid cloud, enterprise IT integration | Big Data, machine learning, containers (Kubernetes) |
| **Pricing** | Complex but flexible | Flexible, discounts for Microsoft customers | Simple, sustained use discounts |
| **Hybrid Cloud** | AWS Outposts | Azure Stack | Anthos |
| **Global Reach** | 32 Regions, 100+ AZs | 60+ Regions | 34 Regions |
| **Compliance** | Strong, with broad certifications | Strong, with focus on enterprise | Focus on AI-driven security and privacy |
| **Developer Tools** | CloudFormation, CodePipeline | Visual Studio, Azure DevOps | Firebase, GKE, Cloud Build |

**16. Write a python program to print your name, designation,**

**technology 100 times?**

Name = str(input(“Enter your name: ”))

Designation = str(input(“Enter your designation: ”))

for i in range(100):

print(Name,” ”,Designation)

**17. Difference between agile and waterfall models?**

|  |  |  |
| --- | --- | --- |
| **Feature** | **Agile** | **Waterfall** |
| **Approach** | Iterative, incremental, flexible | Sequential, linear, rigid |
| **Flexibility** | Highly flexible with frequent changes | Inflexible once a phase is completed |
| **Customer Involvement** | Continuous with feedback after each cycle | Limited at the start and end |
| **Phases** | Repeated in each cycle (planning, design, development, testing, etc.) | Distinct, non-overlapping phases |
| **Risk Management** | Continuous risk management and early identification | Risks identified early, harder to manage if identified later |
| **Documentation** | Minimal | Extensive documentation at each phase |
| **Delivery** | Frequent delivery of working software | Final product delivered at the end |
| **Best For** | Dynamic projects with changing requirements | Projects with well-defined and fixed requirements |

**18. Explain about arithmetic and relational operators with**

**example?**

**Arithmetic Operators:**

Arithmetic operators are used to perform basic mathematical operations on numerical values. These operators work with integers and floating-point numbers.

* Addition
* Subtraction
* Multiplication
* Division
* Floor division
* Modulus division
* Exponentiation

**Ex:**

a = 5

b = 3

print(a + b)

print(a - b)

print(a \* b)

print(a / b)

print(a // b)

print(a % b)

print(a \*\* b)

**Relational Operators:**

Relational operators are used to compare two values and return a boolean value (True or False) based on the result of the comparison.

* Equal to
* Not equal to
* Greater than
* Less than
* Greater than or equal to
* Less than or equal to

**Ex:**

a = 4

b = 3

print(a == b)

print(a != b)

print(a >= b)

print(a <= b)

print(a > b)

print(a < b)

**19. compares b/w set, list, tuple and dictionary?**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Feature** | **List** | **Tuple** | **Set** | **Dictionary** |
| **Syntax** | [] (square brackets) | () (round brackets) | {} (curly braces) | {key: value} (curly braces with key-value pairs) |
| **Ordered** | Yes | Yes | No | Yes |
| **Mutable** | Yes | No | Yes | Yes |
| **Duplicates** | Yes | No | No | No |
| **Indexing** | Yes | Yes | No | Yes |
| **Methods/Operations** | .append(), .remove(), .pop(), .sort(), etc. | .count(), .index() | .add(), .remove(), .discard() | .get(), .keys(), .values(), .items(), etc. |
| **Use Case** | Best for ordered collections | Best for fixed unchanging collections | Best for unique collections and fast membership testing | Best for key-value pairs or mappings |
| **Memory Efficiency** | Less | More | More | Memory efficient when storing large amounts of data with unique keys |

**20. Explain the phases involved in software development life**

**cycle?**

* As the name explains it is a process used by the software development team in the IT industry used to design, develop, test and deploy the product/project
* This process contains a sequence of steps in completing a product/project
* This ensures a quality output
* This contains various phases such as:

1. **Requirement analysis and gathering**
2. **System design**
3. **Implementation**
4. **Testing**
5. **Deployment**
6. **Maintenance**

There are various types of methodologies such as:

Waterfall method, Agile method, V-model, spiral model

1. **Requirement Analysis and Gathering**: This phase involves understanding the needs and expectations of the stakeholders. It includes collecting detailed requirements, both functional and non-functional, to ensure the system will meet the users needs and business goals. Clear documentation of these requirements is essential for the success of the project.
2. **System Design**: Based on the gathered requirements, system architects and developers create a blueprint for the system's architecture. This phase defines the software's structure, including its components, user interfaces, databases. It also focuses on how the system will be implemented to meet the specified requirements.
3. **Implementation**: The design is translated into code during the implementation phase. Developers write the software, build components, and integrate them into the system. It involves choosing correct programming languages, tools, and technologies to make the design into a working product.
4. **Testing**: In this phase, the software is thoroughly tested to find and fix any bugs or issues. Various testing methods, such as unit testing, integration testing, and user acceptance testing, are used to ensure the software functions correctly, meets the requirements, and is free of defects.
5. **Deployment**: Once the software is tested and ready, it is deployed to the production environment. This phase includes installation, configuration, and making the system available for users. Deployment can be done in stages, especially for large systems, to ensure a smooth transition.
6. **Maintenance**: After deployment, the software enters the maintenance phase, where it is continuously updated and improved. This includes fixing bugs, addressing performance issues, and adding new features based on user feedback and evolving requirements. Maintenance ensures the system remains reliable and functional over time.

**21. What is database? what is DBMS and explain types of**

**DBMS?**

**Database:** a database can be defined as an application which stores the collection of data in a structured format

**DBMS (database management system):** this can be defined as a s/w used to store, manage, retrieve and manipulate data in a database and process the SQL queries

**Ex:** MySQL, PostgreSQL, Oracle Database, MongoDB, SQLite

**Flow of execution:**

SQL query (Workbench) **→** DBMS (MySQL) **→** Database (stored on disk) **→** DBMS (MySQL) **→** SQL query result (Workbench)**.**

**Relational DBMS: RDBMS (Relational Database Management System)** is a type of DBMS (Database Management System) that stores data in a relational model**,** which means that data is organized into tables (also called relations) that are linked together based on relationships between the data.

**Non-relational DBMS:** Non**-**RDBMS **(**Non-RelationalDatabaseManagementSystem) refers to databases that do not follow the traditional relational model used by RDBMS (Relational Database Management Systems) like MySQL, PostgreSQL, or SQL Server. In this DBMS the data is stored in an unstructured or unorganized format

**22. what are DDL and DML commands mention example of each**

**one?**

**DDL commands:** DDL commands do not manipulate the data itself, but instead handle the database schema.

* **CREATE**: Used to create a new table, view, index, or database.
* **ALTER**: Used to modify an existing database object (e.g., table structure).
* **DROP**: Used to delete a database object (e.g., table, view, or index).
* **TRUNCATE**: Used to remove all rows from a table, but the table structure remains.
* **RENAME**: Used to rename a database object (e.g., a table or column).

**DML commands:** DML commands allow you to retrieve, insert, update, or delete data in the database tables.

* **SELECT:** Retrieves data from a database.
* **INSERT**: Adds new records to a table.
* **UPDATE**: Modifies existing records in a table.
* **DELETE**: Removes records from a table.

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

In sql clauses help in filtering, sorting, and manipulating data retrieved from database tables.

**1. SELECT Clause**

The select clause is used to specify which columns to retrieve from a table.

select stu\_name, stu\_age from students;

**2. WHERE Clause**

The where clause filters records based on specified conditions.

select \* from students where stu\_age > 15;

**3. ORDER BY Clause**

The order by clause sorts the result set by one or more columns.

select \* from students order by stu\_fees desc;

**4. GROUP BY Clause**

The group by clause groups rows that have the same values in specified columns into summary rows.

select stu\_class, count(\*) from students group by stu\_class;

**5. HAVING Clause**

The having clause is used to filter records after aggregation, it is used with group by.

select stu\_class, avg(stu\_fees) from students group by stu\_class having avg(stu\_fees) > 2500;

**6. LIMIT Clause**

The limit clause restricts the number of rows returned by a query.

select \* from students limit 3;

**7. LIKE Clause**

The like clause is used for pattern matching in string comparisons.

select \* from students where stu\_name like 'm%';

**24. Explain the concept of joins with examples?**

**Joins:**

Joins in MySQL are essential for combining records from two or more tables based on related columns. This allows us for efficient data retrieval and analysis.

Joins are used to combine rows from two or more tables based on a related column between them.

This helps in retrieving the data from a single query rather than using multiple queries

**Types of joins:**

The joins are basically divided into the following types:

* Inner join
* Outer join (left outer or left join, right outer or right join, full join)
* Self join
* Cross join

**Inner join:** Returns only the rows with matching values in both tables.

**Syntax:**

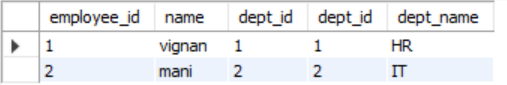
Assume two tables table\_1 and table\_2 for all the joins

Select columns from table\_1

Inner join table\_2 on table\_1.common\_col = table\_2.common\_col;

**Query:** select \* from emps

Inner join dept on emps.dept\_id = dept.dept\_id;



**Left outer join (or) left join:** Returns all rows from the left table and matched rows from the right table. If there is no match, NULLs are returned for columns from the right table.

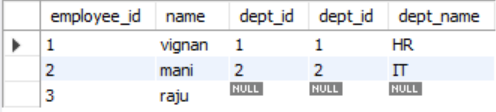
**Syntax:**

select columns from table1

left join table\_2 on table\_1.common\_col = table\_2.common\_col;

**Query:** select \* from emps

left join dept on emps.dept\_id = dept.dept\_id;



**Right outer join (or) right join:** Returns all rows from the right table and matched rows from the left table. If there is no match, NULLs are returned for columns from the left table.

**Syntax:**

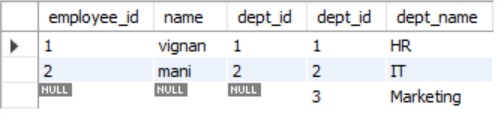
select columns from table1

right join table\_2 on table\_1.common\_col = table\_2.common\_col;

**Query:**

select \* from emps

right join dept on emps.dept\_id = dept.dept\_id;



**Cross join:** it is a join that returns cartesian product means every row from the first table is combined with every row from the second table.

Assume Table A has “m” rows and Table B has “N” rows then the total rows will be m x n rows

The Cartesian product of these two tables will contain m x n rows. Each row in the result is a combination of a row from Table A and a row from Table B.

**Syntax:**

select columns

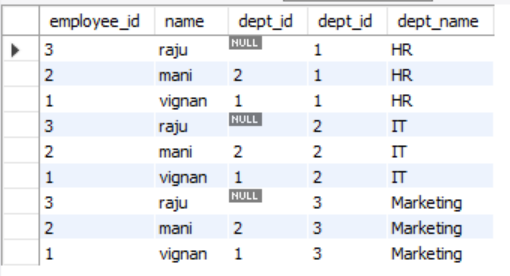
from table\_1

cross join table\_2;

**Query:**

select \* from emps

cross join dept;



**Self join**: A join where a table is joined with itself. It is often used to compare rows within the same table

**Syntax:**

select a.columns, b.columns

from table as a

join table as b on a.common\_column = b.common\_column;

**25. create a trigger and explain?**

A **trigger** is a special type of stored procedure that automatically executes when certain events occur in the database. Triggers are often used for:

* **Data validation**: Ensuring data integrity by enforcing rules.
* **Auditing**: Keeping track of changes made to data.
* **Automating tasks**: Performing automatic actions in response to data changes.

**Trigger creation:**

CREATE TRIGGER trg\_before\_insert\_emp

BEFORE INSERT ON employees

FOR EACH ROW

BEGIN

IF NEW.salary < 0 THEN

SET NEW.salary = 0;

END IF;

END;

**Explanation:**

* **Trigger Name**: trg\_before\_insert\_emp
* **Event**: This trigger fires BEFORE INSERT on the employees table.
* **Action**:
  + The trigger checks if the new salary (NEW.salary) being inserted is less than zero.
  + If it is, it sets the salary to zero before the insert operation is completed.