1. What is Databricks?

Databricks is a Cloud-based industry-leading data engineering platform designed to process & transform huge volumes of data. Databricks is the latest big data tool that was recently added to Azure.

2. What is DBU?

Databricks Unified platform is a Databricks unit used to process the power, and it is also used to measure the pricing purposes.

3. Is Azure Databricks different from databricks?

Azure Databricks is an Artificial intelligence service developed by Microsoft and Databricks jointly to introduce innovation in data analytics, machine learning, and data engineering.

4. Which SQL version is used by databricks?

Spark implements ANSI 2003

Syntax: https://spark.apache.org/releases/spark-release-2-0-0.html

5. What are the different types of pricing tiers available in Databricks?

There are two types of pricing tiers available in Databricks they are:

1. Premium Tier
2. Standard Tier

6. What is the use of the databricks file system?

Databricks file system is a distributed file system used to ensure data reliability even after eliminating the cluster in Azure databricks.

7. What are the different ETL operations done on data in Azure Databricks?

The different ETL operations performed on data in Azure Databricks are:

1. The data is transformed from the databricks to the data warehouse.
2. Bold storage is used to load the data.
3. Bold storage acts as temporary storage of the data.

8. How to generate a personal access token in databricks?

We can generate a personal access token in seven steps they are:

1. In the upper right corner of Databricks workspace, click the icon named: “user profile.”
2. In the second step, you have to choose “User setting.”
3. navigate to the tab called “Access Tokens.”
4. Then you can find a “Generate New Token” button. Click it.

9. How to revoke a personal access token?

We have to follow five steps to revoke a personal access token they are:

1. In the upper right corner of Databricks workspace, click the icon named: “user profile.”
2. In the second step, you have to choose “User setting.”
3. navigate to the tab called “Access Tokens.”
4. In this step, you have to click x for the token you need to revoke.
5. Finally, click the button “Revoke Token” on the Revoke Token dialog.

10. What is the purpose of databricks runtime?

Databricks runtime is used to run the set of components on the databricks platform.

11. How to reuse the code in the azure notebook?

If we want to reuse the code in the azure notebook, then we must import that code into our notebook. We can import it in two ways–> 1) if the code is in a different workspace, we have to create a module/jar of the code and then import it into a module or jar. 2) if the code is in the same workspace, we can directly import and reuse it.

12. Write the syntax to connect the Azure storage account and databricks?

```
dbutils.fs.mount( source = “wasbs://@.blob.core.windows.net”, mount_point = “/mnt/”,
extra_configs = {“”:dbutils.secrets.get(scope = “”, key = “”)})
```

13. What is the use of ‘%sql’?

‘%sql’ is used to switch the scala/python notebook into a mere SQL notebook.

14. What is a databricks cluster?

A databricks cluster is a group of configurations and computation resources on which we can run data science, data analytics workloads, data engineering, like production ETL ad-hoc analytics, pipelines, machine learning, and streaming analytics.

15. List the different types of cluster modes in the azure databricks?

There are three different types of cluster modes in the azure databricks they are:

1. Single-node Cluster.

2. Standard Cluster.
3. High Concurrency Cluster.

16. What is the use of %run?

The %run command is used to parameterize a databricks notebook. %run is also used to modularize the code.

17. What is the use of widgets in databricks?

Widgets enable us to add parameters to our dashboards and notebooks. The API widget consists of calls to generate multiple input widgets, get the bound values and remove them.

18. What is a secret in databricks?

A secret is a key-value pair that stocks up the secret material; it consists of a unique key name inside a secret scope. The limit of each scope is up to 1000 secrets. The maximum size of the secret value is 128 KB.

19. Write a syntax to list secrets in a specific scope?

The syntax to list secrets in a specific scope is:

CLI: databricks secrets list –scope

DBUI:

```
dbutils.secrets.listAllScopes()

dbutils.secrets.list(“scopename”)
```

20. What is the use of Secrets utility?

Secrets utility is used to read the secrets in the job or notebooks.

21. How to delete a Secret?

We can use Azure Portal UI or Azure SetSecret Rest API to delete a Secret from any scope that is backed by an Azure key vault.

22. What are the two types of secret scopes?

There are two types of secret scopes they are:

1. Databricks-backed scopes.
2. Azure key Vault-backed scopes.

23. What are the things involved when pushing the data pipeline to a staging environment?

The four things involved when pushing the data pipeline to a staging environment are:

1. Notebooks
2. Libraries
3. Clusters and Jobs configuration
4. Results

24. **How to add new columns in data frames?**

new column to DataFrame using withColumn(), select(), sql(), Few ways include adding a constant column with a default value, derive based out of another column, add a column with NULL/None value, add multiple columns e.t.c

25. **How to delete duplicate data in data frames?**

There are many methods that you can use to identify and remove the duplicate records from the Spark SQL DataFrame. For example, you can use the functions such as distinct() or dropDuplicates() to remove duplicate while creating another dataframe.

You can use any of the following methods to identify and remove duplicate rows from Spark SQL DataFrame.

- Remove Duplicate using **distinct()** Function
- Remove Duplicate using **dropDuplicates()** Function
- Identify Spark DataFrame Duplicate records using **groupBy** method
- Identify Spark DataFrame Duplicate records using **row_number** window Function

26.**What is Mounting?**

Mounting is a process by which the operating system makes files and directories on a storage device

create Mount:

```
butils.fs.mount(

  source =
"wasbs://<container-name>@<storage-account-name>.blob.core.windows.net",
  mount_point = "/mnt/<mount-name>",
  extra_configs = {"<conf-key>":dbutils.secrets.get(scope = "<scope-name>", key = "<key-name>")})
```

UnMount:

```
dbutils.fs.unmount("/mnt/<mount-name>")
```

27.**Difference between RDD, Dataset and Dataframe ?**

	RDDs	**Dataframes**	**Datasets**
Data Representation	RDD is a distributed collection of data elements without any schema.	It is also the distributed collection organized into the named columns	It is an extension of Dataframes with more features like type-safety and object-oriented interface.
Optimization	No in-built optimization engine for RDDs. Developers need to write the optimized code themselves.	It uses a catalyst optimizer for optimization.	It also uses a catalyst optimizer for optimization purposes.
Projection of Schema	Here, we need to define the schema manually.	It will automatically find out the schema of the dataset.	It will also automatically find out the schema of the dataset by using the SQL Engine.
Aggregation Operation	RDD is slower than both Dataframes and Datasets to perform simple operations like grouping the data.	It provides an easy API to perform aggregation operations. It performs aggregation faster than both RDDs and Datasets.	Dataset is faster than RDDs but a bit slower than Dataframes.

28. Types of clusters?

Standard clusters

A Standard cluster is recommended for a single user. Standard clusters can run workloads developed in any language: Python, SQL, R, and Scala.

High Concurrency clusters

A High Concurrency cluster is a managed cloud resource. The key benefits of High Concurrency clusters are that they provide fine-grained sharing for maximum resource utilization and minimum query latencies.

High Concurrency clusters can run workloads developed in SQL, Python, and R. The performance and security of High Concurrency clusters is provided by running user code in separate processes, which is not possible in Scala.

Single Node clusters

A Single Node cluster has no workers and runs Spark jobs on the driver node.

In contrast, a Standard cluster requires *at least one* Spark worker node in addition to the driver node to execute Spark jobs.

To create a Single Node cluster, set **Cluster Mode** to **Single Node**.

29. How to call notebook from ADF?

Create linked services

In this section, you author a Databricks linked service. This linked service contains the connection information to the Databricks cluster:

Create an Azure Databricks linked service

1. On the home page, switch to the **Manage** tab in the left panel.

In the **New Linked Service** window, complete the following steps:

2. For **Name**, enter ***AzureDatabricks_LinkedService***

3. Select the appropriate **Databricks workspace** that you will run your notebook in

4. For **Select cluster**, select **New job cluster**

5. For **Domain/ Region**, info should auto-populate

6. For **Access Token**, generate it from Azure Databricks workplace. You can find the steps here.

7. For **Cluster version**, select **4.2** (with Apache Spark 2.3.1, Scala 2.11)

8. For **Cluster node type**, select **Standard_D3_v2** under **General Purpose (HDD)** category for this tutorial.

9. For **Workers**, enter **2**.

10. Select **Finish**

https://docs.microsoft.com/en-us/azure/data-factory/transform-data-using-databricks-notebook

30.Databricks architecture?

Spark architecture:

Driver Program:

Driver program will create **Spark Context,** which is the starting point of a program.

It connects with Cluster Manager and does the following thing,

o It acquires Executors on nodes in cluster

o It provides the details of transformations (DAG) to Worker Nodes, which need to perform.

o It send the Tasks to perform to executors.

o It collect the output from workers nodes if required.

Cluster Manager:

o Role of Cluster manager is to allocate the resources across the application.

o It consists of various types of cluster managers such as Hadoop YARN, Apache Mesos and Standalone Scheduler.

- Here, the Standalone Scheduler is a standalone spark cluster manager that facilitates to install Spark on an empty set of machines.

Worker Node:

- The worker node is a slave node
- Its role is to run the application code in the cluster and return the results back to Driver note if required.

Executor:

1. An executor is a process launched for an application on a worker node.
2. It runs tasks and keeps data in memory or disk storage across them.
3. It read and write data to the external sources.
4. Every application contains its executor.

Task:

- A unit of work that will be sent to one executor.

31. What is Azure Databricks?

Databricks is a managed platform for running Apache Spark. Azure Databricks is an Apache Spark-based analytics platform optimized for the Microsoft Azure cloud services platform. Azure Databricks is a fast, easy, and collaborative Apache Spark-based analytics service. For a big data pipeline.

32. What is Spark?

Spark is a tool for just that, managing and coordinating the execution of tasks on data across a cluster of computers.

1. It is faster in terms of execution because it is using in-memory computation.
2. It is a unified platform to replace many of the applications/Tools in bigdata platform

Python:

1. What is Python?

Python is a high-level, interpreted, general-purpose programming language. Being a general-purpose language, it can be used to build almost any type of application with the right tools/libraries. Additionally, python supports objects, modules, threads, exception-handling, and automatic memory management which help in modeling real-world problems and building applications to solve these problems.

2. What are the benefits of using Python?

- Python is a general-purpose programming language that has a simple, easy-to-learn syntax that emphasizes readability and therefore reduces the cost of program maintenance. Moreover, the language is capable of scripting, is completely open-source, and supports third-party packages encouraging modularity and code reuse.
- Its high-level data structures, combined with dynamic typing and dynamic binding, attract a huge community of developers for Rapid Application Development and deployment

3. What is an Interpreted language?

An Interpreted language executes its statements line by line. Languages such as Python, Javascript, R, PHP, and Ruby are prime examples of Interpreted languages. Programs written in an interpreted language runs directly from the source code, with no intermediary compilation step.

4. **What are lists and tuples? What is the key difference between the two?**

Lists and Tuples are both sequence data types that can store a collection of objects in Python. The objects stored in both sequences can have different data types. Lists are represented with square brackets ['sara', 6, 0.19], while tuples are represented with parantheses ('ansh', 5, 0.97).

But what is the real difference between the two? The key difference between the two is that while lists are mutable, tuples on the other hand are immutable objects. This means that lists can be modified, appended or sliced on the go but tuples remain constant and cannot be modified in any manner. You can run the following example on Python IDLE to confirm the difference:

```
my_tuple = ('sara', 6, 5, 0.97)

my_list = ['sara', 6, 5, 0.97]

print(my_tuple[0])     # output => 'sara'

print(my_list[0])     # output => 'sara'

my_tuple[0] = 'ansh'    # modifying tuple => throws an error

my_list[0] = 'ansh'    # modifying list => list modified

print(my_tuple[0])     # output => 'sara'

print(my_list[0])     # output => 'ansh'
```

5. What is slicing in Python?

- As the name suggests, ‘slicing’ is taking parts of.
- Syntax for slicing is [start : stop : step]
- start is the starting index from where to slice a list or tuple
- stop is the ending index or where to sop.
- step is the number of steps to jump.
- Default value for start is 0, stop is number of items, step is 1.
- Slicing can be done on strings, arrays, lists, and tuples.

6.What is lambda in Python? Why is it used?

Lambda is an anonymous function in Python, that can accept any number of arguments, but can only have a single expression. It is generally used in situations requiring an anonymous function for a short time period. Lambda functions can be used in either of the two ways:

- Assigning lambda functions to a variable:

```
mul = lambda a, b : a * b
print(mul(2, 5))    # output => 10
```

7. Explain split() and join() functions in Python?

- You can use split() function to split a string based on a delimiter to a list of strings.
- You can use join() function to join a list of strings based on a delimiter to give a single string.

```
string = "This is a string."
string_list = string.split(' ') #delimiter is ‘space’ character or ‘ ‘
print(string_list) #output: ['This', 'is', 'a', 'string.']
print(' '.join(string_list)) #output: This is a string.
```

8. What does *args and **kwargs mean?

*args

- *args is a special syntax used in the function definition to pass variable-length arguments.
- “*” means variable length and “args” is the name used by convention. You can use any other.

```
def multiply(a, b, *argv):
  mul = a * b
  for num in argv:
     mul *= num
  return mul
print(multiply(1, 2, 3, 4, 5)) #output: 120
```

**kwargs

- **kwargs is a special syntax used in the function definition to pass variable-length keyworded arguments.
- Here, also, “kwargs” is used just by convention. You can use any other name.
- Keyworded argument means a variable that has a name when passed to a function.
- It is actually a dictionary of the variable names and its value.

```
def tellArguments(**kwargs):
  for key, value in kwargs.items():
      print(key + ": " + value)
tellArguments(arg1 = "argument 1", arg2 = "argument 2", arg3 = "argument 3")
#output:
# arg1: argument 1
# arg2: argument 2
# arg3: argument 3
```

9. Define pandas dataframe.

A dataframe is a 2D mutable and tabular structure for representing data labelled with axes - rows and columns.

The syntax for creating dataframe:

```
import pandas as pd
dataframe = pd.DataFrame( data, index, columns, dtype)
```

10.What do you understand by NumPy?

NumPy is one of the most popular, easy-to-use, versatile, open-source, python-based, general-purpose package that is used for processing arrays. NumPy is short for NUMerical PYthon. This is very famous for its highly optimized tools that result in high performance and powerful N-Dimensional array processing feature that is designed explicitly to work on complex arrays.

11. What are the steps to create 1D, 2D and 3D arrays?

- 1D array creation:

```
import numpy as np
one_dimensional_list = [1,2,4]
one_dimensional_arr = np.array(one_dimensional_list)
print("1D array is : ",one_dimensional_arr)
```

- 2D array creation:

```
import numpy as np

two_dimensional_list=[[1,2,3],[4,5,6]]

two_dimensional_arr = np.array(two_dimensional_list)

print("2D array is : ",two_dimensional_arr)
```

- 3D array creation:

```
import numpy as np

three_dimensional_list=[[[1,2,3],[4,5,6],[7,8,9]]]

three_dimensional_arr = np.array(three_dimensional_list)

print("3D array is : ",three_dimensional_arr)
```

- ND array creation: This can be achieved by giving the ndmin attribute. The below example demonstrates the creation of a 6D array:

```
import numpy as np

ndArray = np.array([1, 2, 3, 4], ndmin=6)

print(ndArray)

print('Dimensions of array:', ndArray.ndim)
```

Python Programming Examples

1. Write python function which takes a variable number of arguments.

A function that takes variable arguments is called a function prototype. Syntax:

```
def function_name(*arg_list)
```

For example:

```
def func(*var):

  for i in var:
    print(i)
func(1)
func(20,1,6)
```

The * in the function argument represents variable arguments in the function.

2.Write a program for counting the number of every character of a given text file.

The idea is to use collections and pprint module as shown below:

```
import collections
import pprint
with open("sample_file.txt", 'r') as data:
 count_data = collections.Counter(data.read().upper())
 count_value = pprint.pformat(count_data)
print(count_value)
```

3.Write a program to check and return the pairs of a given array A whose sum value is equal to a target value N.

This can be done easily by using the phenomenon of hashing. We can use a hash map to check for the current value of the array, x. If the map has the value of (N-x), then there is our pair.

```
def print_pairs(arr, N):
  # hash set
  hash_set = set()

  for i in range(0, len(arr)):
     val = N-arr[i]
     if (val in hash_set):    #check if N-x is there in set, print the pair
        print("Pairs " + str(arr[i]) + ", " + str(val))
     hash_set.add(arr[i])
```

```
# driver code
arr = [1, 2, 40, 3, 9, 4]
N = 3
print_pairs(arr, N)
```

4.Write a Program to solve the given equation assuming that a,b,c,m,n,o are constants:

$ax + by = c$

$mx + ny = o$

By solving the equation, we get:

```
a, b, c, m, n, o = 5, 9, 4, 7, 9, 4
temp = a*n - b*m
if n != 0:
   x = (c*n - b*o) / temp
   y = (a*o - m*c) / temp
   print(str(x), str(y))
```

5.Write a Program to match a string that has the letter ‘a’ followed by 4 to 8 'b’s.

We can use the re module of python to perform regex pattern comparison here.

```
import re
def match_text(txt_data):
        pattern = 'ab{4,8}'
        if re.search(pattern,  txt_data):    #search for pattern in txt_data
            return 'Match found'
        else:
            return('Match not found')
```

```
print(match_text("abc"))         #prints Match not found

print(match_text("aabbbbbc"))    #prints Match found
```

6.Write a Program to convert date from yyyy-mm-dd format to dd-mm-yyyy format.

You can use the datetime module as shown below:

```
from datetime import datetime

new_date = datetime.strptime("2021-08-01", "%Y-%m-%d").strftime("%d:%m:%Y")

print(new_data)
```