1) Why we need Azure Data Factory?

The amount of data generated these days are huge and these data comes from different sources.

When we move particular the data in cloud.there are few things need to be taken care of that

i.e we need to transform the data meaningful and delete the unnescessary parts.

Hence Data Factory helps to orchestrate this complete process into more manageable or organizable manner.

2) What is Azure Data Factory?

Cloud based integration service that allows creating the data-driven workflows in the cloud for orchestrating and automating data movement and data transformation.

Using the data factory, we can able to create and schedule the pipelines that can ingest from desperate data source.

It can process and transform the data by using compute services such as HDinsights hadoop, spark, azure data analytics and azure machine learning.

3)what is Integration Runtime?

The integration runtime is the compute infrastructure that azure data factory uses to provide the following data integration capabilities across various network environment:

Data movement: for data movement, the integration runtime moves data from source to destination data store.

Dispatch activities: for transformation, integration runtime provides capabilities to natively executes ssis packages.

Execute SSIS packages: The integration runtime natively executes SSIS packages in a managed azure compute enviroment.

Integration Runtime also supports dispatching and monitoring transformation activities running on variety compute services such as Azure HDinsight,Azure Machine learning,SQL database and SQL server

4)what is the limit on number of integration runtime?

There is no limit on the number of integration runtime instances you can have in data factory.

there is however a limit on the number on the vm cores that the integration runtime can use per subscription for ssis package execution

5)what is differnce between Azure Data lake and Azure Data Warehouse?

Azure data warehouse is traditional way of storing the data which is widely used. Data lake is complementary to data warehouse.

i.e if you have data at data lake that can be used stored in data warehouse as well but there are certain rules to be followed

-------------------------------------------------------------------------------

Data Lake || Data Warehouse

-------------------------------------------------------------------------------

1)Complementary to datawarehouse || 1)May be sourced to data lake

2)data is detailed or raw data.it can be ||2)data is filtered ,summarized and refined

any form.You need to take data and dump ||

it in data lake ||

3)schema on read (not structred,you can ||3)Schema on write(data is written in structred form

define your schema in n number of ways)||or in a particular schema)

4)one language to process data of any ||4)It uses SQL

format(Unified SQL)

6) ADLS gen1 vs ADLS gen2

|  |  |
| --- | --- |
| ***Azure Data Lake Gen1*** | ***Azure Data Lake Gen2*** |
| Azure Data Lake Gen 1 is file system storage in which data is distributed in blocks in a hierarchical file system. | Azure Data Lake Gen 2 contains both file system storage for performance & security and object storage for scalability. |
| Hot/Cold storage tier not supported | Supports Hot/Cold Storage tier |
| Redundant Storage not supported | Supports Redundant Storage |
| Azure data lake Analytics support available | Azure Data Lake analytics support is not available (till date 2nd July 2019). |

**7) Data profiling:**

Data profiling is a process of examining data from an existing source and summarizing information about that data. You profile data to determine the accuracy, completeness, and validity of your data. Data profiling can be done for many reasons, but it is most commonly part of helping to determine data quality as a component of a larger project. Commonly, data profiling is combined with an ETL (Extract, Transform, and Load) process to move data from one system to another. When done properly, ETL and data profiling can be combined to cleanse, enrich, and move quality data to a target location.

It is the step where we can understand about data and check quality of data. It is the first step for data pre-processing. In this step data is cleaned and enriching.

**8) Data encryption:**

**Data** at rest in **Azure** Blob storage and **Azure** file shares can be **encrypted** in both server-side and client-side scenarios. **Azure** Storage Service **Encryption** (SSE) can automatically **encrypt data** before it is stored, and it automatically decrypts the **data** when you retrieve it. The process is completely transparent to users

AES encryption-Advanced Encryption Standard

About **Azure** Storage **encryption**  
  
**Data** in **Azure** Storage is **encrypted** and decrypted transparently using 256-bit AES **encryption**, one of the strongest block ciphers available, and is FIPS 140-2 compliant.

OLTP:

**OLTP** and **OLAP**: The two terms look similar but refer to different kinds of systems. Online transaction processing (**OLTP**) captures, stores, and processes data from transactions in real time.

OLAP

Online analytical processing (**OLAP**) uses complex queries to analyze aggregated historical data from **OLTP** systems.

**9) Materialized view:**

It is database object that contains the result of a query that query can contains various joins of the tables for performing the transformation. For example it can contain local copy of data by this type of view

**10) Normalisation vs Denormalization**

| **r. No.** | **Key** | **Normalization** | **Denormalization** |
| --- | --- | --- | --- |
| 1 | Implementation | Normalization is used to remove redundant data from the database and to store non-redundant and consistent data into it. | Denormalization is used to combine multiple table data into one so that it can be queried quickly. |
| 2 | Focus | Normalization mainly focuses on clearing the database from unused data and to reduce the data redundancy and inconsistency. | Denormalization on the other hand focus on to achieve the faster execution of the queries through introducing redundancy. |
| 3 | Number of Tables | During Normalization as data is reduced so a number of tables are deleted from the database hence tables are lesser in number. | On another hand during Denormalization data is integrated into the same database and hence a number of tables to store that data increases in number. |
| 4 | Memory consumption | Normalization uses optimized memory and hence faster in performance. | On the other hand, Denormalization introduces some sort of wastage of memory. |
| 5 | Data integrity | Normalization maintains data integrity i.e. any addition or deletion of data from the table will not create any mismatch in the relationship of the tables. | Denormalization does not maintain any data integrity. |
| 6 | Where to use | Normalization is generally used where number of insert/update/delete operations are performed and joins of those tables are not expensive. | On the other hand Denormalization is used where joins are expensive and frequent query is executed on the tables. |

11) **ITIL:**

Information Technology Infrastructure Library.

It comprises a life cycle which contains

1. Service Strategy
2. Service Design
3. Service Transition (Changing)
4. Service Operation
5. Continual Service Improvement

**12 ) Elastic Search:**

Its is like search engine. Where the data is fetched from any source reliably and securely. It is used when we are working with tera bytes and peta bytes of data.

**13) Massive Parallel Processing:**

We placed the data in distributed manner. Hence, we need to faster means we need to split the process according to distributed data.

**14) Azure Devops:**

It is hosted service providing development and collaboration tool.

Agile means speedy finishing task or stories (Customer requirements)

Sprint- It is rapid release.

Scrum-it is process update for the task

We are setting the time for completing the stories

Phases available in Planning, retrospective and grooming

15)HDinsight vs ADLA

|  |  |
| --- | --- |
| HDinsight | Azure Data Lake Analytics |
| Its Platform as service | It is Software as service |
| If we want to process a dataset, we have to configure the cluster with predefined nodes | It is all about passing query, written for processing the data and Azure data lake analytics will create the necessary compute nodes as per the demand and propose of the dataset |
| Since we configured the cluster with HDInsight, we can create as we want and we can control as we want. All Hadoop subprojects such as spark, Kafka can be used without any limitations | In ADLA, in terms of cluster azure will take care of it. We don’t need to worry about the cluster creation. In addition, we can make use of USQL taking advantage of dotnet for processing the data |

16) what are top level concept of Azure Data Factory?

**Pipeline**: It acts as a carrier in which we have various processing taking place.

This individual process is an activity.

**Activities**: Activities represents the processing steps in a pipeline. A pipeline can have one or more activities. It can be anything i.e process like querying the data set or moving the the data set from one source to another.

**Dataset**: Source of Data. In simple words, It is a data structure that holds the data.

**Linked services**: this contains the information of connection string for connecting source and destination

17)How can I schedule a Pipeline?

Scheduler trigger or time window trigger to schedule a pipeline

The trigger uses a wall-clock calendar schedule, which can schedule pipelines periodically or in calendar based recurrent patterns.

18)How gracefully handle null values in an activity output?

We can use the @coalesce construct in the expression to handle the null values.

19) which data factory version do I use to create data flows?

Use the data factory v2 version to create data flows.

20) Types of integration Runtime?

There are three types Integration runtime

1. Azure integration runtime
2. Self-hosted integration runtime
3. Azure-SQL server integration service (SSIS)

21) what is wild card path

**wildcard** file filters to let Copy Activity pick up only files that have the defined naming pattern

22) Properties of copy behaviour:

1- **Preserve Hierarchy** (default): Preserves the file hierarchy in the target folder. The relative path of source file to source folder is identical to the relative path of target file to target folder.

2- **Flatten Hierarchy**: All files from the source folder are in the first level of the target folder. The target files have autogenerated names.

1. **Merge Files**: Merges all files from the source folder to one file. If the file name is specified, the merged file name is the specified name. Otherwise, it's an autogenerated file name

23) what and Why Widget are created in Databricks notebooks

Input widgets allow you to add parameters to your notebooks and dashboards. The widget API consists of calls to create various types of input widgets, remove them, and get bound values.

* Building a notebook or dashboard that is re-executed with different parameters
* Quickly exploring results of a single query with different parameters

24) Widgets Types

* **text**: Input a value in a text box.
* **dropdown:** Select a value from a list of provided values.
* **combobox:** Combination of text and dropdown. Select a value from a provided list or input one in the text box.
* **multiselect:** Select one or more values from a list of provided values.

25) Using Spark SQL, how will you merge the tables(<https://docs.databricks.com/spark/latest/spark-sql/language-manual/delta-merge-into.html>)

MERGE **INTO** [db\_name.]target\_table [**AS** target\_alias]

**USING** [db\_name.]source\_table [<time\_travel\_version>] [**AS** source\_alias]

**ON** <merge\_condition>

[ **WHEN** MATCHED [ **AND** <condition> ] **THEN** <matched\_action> ]

[ **WHEN** MATCHED [ **AND** <condition> ] **THEN** <matched\_action> ]

[ **WHEN** **NOT** MATCHED [ **AND** <condition> ] **THEN** <not\_matched\_action> ]

26) Difference between Python and scala?

| **PYTHON** | **SCALA** |
| --- | --- |
| Python is a dynamically typed language. | Scala is a statically typed language. |
| We don’t need to specify objects in Python because it is a dynamically typed Object Oriented Programming language. | We need to specify the type of variables and objects in Scala because Scala is statically typed Object Oriented Programming language. |
| Python is easy to learn and use. | Scala is less difficult to learn than Python. |
| An extra work is created for the interpreter at the runtime. | No extra work is created in Scala and thus it is 10 times faster than Python. |
| The data types are decided by it during runtime. | This is not the case in Scala that is why while dealing with large data process, Scala should be considered instead of Python |
| Python’s Community is huge compared to Scala. | Scala also has good community support. But still, it is lesser than Python. |
| Python supports heavyweight process forking and doesn’t support proper multithreading. | Scala has reactive cores and a list of asynchronous libraries and hence Scala is a better choice for implementing concurrency. |
| Its methodologies are much complex in Python as it is dynamic programming language. | Testing is much better in scala because it is a statically typed language. |
| It is popular because of its English-like syntax. | For scalable and concurrent systems, Scala play much bigger. |
| Python is easy for the developers to write code in it. | Scala is less difficult to learn than Python and it is difficult to write code in Scala. |
| There is an interface in Python to many OS system calls and libraries. It has many interpreters | It is basically a compiled language and all source codes are compiled before execution |
| Python language is highly prone to bugs whenever there is any change to the existing code. | No such problem is seen in Scala. |
| Python has libraries for Machine learning and proper data science tools and Natural Language Processing (NLP). | Where as Scala has no such tools. |
| Python can be used for small-scale projects. | Scala can be used for large-scale projects. |
| It doesn’t provide scalable feature support. | It provides scalable feature support. |