Hive Questions:

1. What is Apache Hive?

* + - Apache Hive is an open-source data warehousing tool developed on top of Hadoop Distributed File System (HDFS) for performing distributed processing and data analysis.
    - Facilitates reading, writing, and managing large datasets residing in distributed storage using SQL

2. What kind of applications is supported by Apache Hive?

* + - Hive supports applications written in any language like Python, Java, C++, Ruby, etc. using JDBC, ODBC, and Thrift drivers, for performing queries on the Hive.

3. Is Hive suitable to be used for OLTP systems? Why?

* + - No Hive does not provide insert and update at row level. So it is not suitable for OLTP system.

4. Where does the data of a Hive table gets stored?

* + - hdfs://namenode\_server/user/hive/warehouse
    - One can change it by specifying the desired directory in hive.metastore.warehouse.dir configuration parameter present in the hive-site.xml.

5. What is a metastore in Hive?

* + - Metastore is a central repository that stores the metadata information about the structure of tables and partitions, including column and column type information.
    - It also stores information of serializer and deserializer, required for the read/write operation, and HDFS files where data is stored. This metastore is generally a relational database.
    - Metastore provides a Thrift interface for querying and manipulating Hive metadata.
    - We can configure metastore in any of the two modes:
      * Remote: In remote mode, metastore is a Thrift service and is useful for non-Java applications.
      * Embedded: In embedded mode, the client can directly interact with the metastore using JDBC.

6. Why does Hive not store metadata information in HDFS?

* + - Hive stores metadata information in the metastore using RDBMS instead of HDFS. The reason for choosing RDBMS is to achieve low latency as HDFS read/write operations are time consuming processes

7. What is the difference between local and remote metastore?

* + - Local Metastore:
      * In local metastore configuration, the metastore service runs in the same JVM in which the Hive service is running and connects to a database running in a separate JVM, either on the same machine or on a remote machine.
    - Remote Metastore:
      * In the remote metastore configuration, the metastore service runs on its own separate JVM and not in the Hive service JVM. Other processes communicate with the metastore server using Thrift Network APIs. You can have one or more metastore servers in this case to provide more availability.

8. What is the default database provided by Apache Hive for metastore?

* + - It offers an embedded Derby database instance backed by the local disk for the metastore, by default. It is what we call embedded metastore configuration.

9. What is the difference between the external table and managed table?

* + - In case of managed table, If one drops a managed table, the metadata information along with the table data is deleted from the Hive warehouse directory.
    - On the contrary, in case of an external table, Hive just deletes the metadata information regarding the table and leaves the table data present in HDFS untouched.

10. Is it possible to change the default location of a managed table?

* + - Yes, by using the LOCATION keyword, we can change the default location of Managed tables while creating the managed table in Hive. However, to do so, the user needs to specify the storage path of the managed table as the value to the LOCATION keyword, that will help to change the default location of a managed table.
    - LOCATION ‘<hdfs\_path>’

11. When should we use SORT BY instead of ORDER BY?

* + - Hive supports SORT BY which sorts the data per reducer. The difference between "order by" and "sort by" is that the former guarantees total order in the output while the latter only guarantees ordering of the rows within a reducer. If there are more than one reducer, "sort by" may give partially ordered final results.

12. What is a partition in Hive?

<https://data-flair.training/blogs/apache-hive-partitions/>

* + - Hive organizes tables into partitions. It is a way of dividing a table into related parts based on the values of partitioned columns such as date, city, and department. Using partition, it is easy to query a portion of the data.
    - Each table in the hive can have one or more partition keys to identify a particular partition.
    - Tables or partitions are sub-divided into buckets, to provide extra structure to the data that may be used for more efficient querying. Bucketing works based on the value of hash function of some column of a table.

13. Why do we perform partitioning in Hive?

* + - Using partition, it is easy to query a portion of the data. Thus this decreases the I/O time required by the query. Hence increases the performance speed.
    - Partitioning is the optimization technique in Hive which improves the performance significantly.

14. What is dynamic partitioning and when is it used?

* + - Single insert to partition table is known as a dynamic partition.
    - Usually, dynamic partition loads the data from the non-partitioned table.
    - Dynamic Partition takes more time in loading data compared to static partition.
    - When you have large data stored in a table then the Dynamic partition is suitable.
    - If you want to partition a number of columns but you don’t know how many columns then also dynamic partition is suitable.
    - Dynamic partition there is no required where clause to use limit.
    - we can’t perform alter on the Dynamic partition.
    - You can perform dynamic partition on hive external table and managed table.
    - If you want to use the Dynamic partition in the hive then the mode is in non-strict mode.

15. Why do we need buckets?

* + - To provide extra structure to the data that may be used for more efficient querying. Bucketing works based on the value of hash function of some column of a table.
    - Bucketing in hive is useful when dealing with large datasets that may need to be segregated into clusters for more efficient management and to be able to perform join queries with other large datasets. The primary use case is in joining two large datasets involving resource constraints like memory limits.

16. How Hive distributes the rows into buckets?

17. What is indexing and why do we need it?

* + - Indexes are a pointer or reference to a record in a table as in relational databases. Indexing is a relatively new feature in Hive. In Hive, the index table is different than the main table. Indexes facilitate in making query execution or search operation faster. However, storing indexes require disk space and creating an index involves cost. So, the use of indexes may not always be of any benefit. “EXPLAIN” query must be checked to evaluate the benefit through a query execution plan. Indexing in hive makes large dataset analysis relatively quicker by better query performance on operations.
    - With the petabytes of data that needs to be analyzed, querying Hive tables with millions of records and hundreds of columns becomes time-consuming. Indexing a table helps in performing any operation faster.
    - Bitmap Indexing: This is used with columns having a few distinct values. It is known to store both the indexed column’s value and the list of rows as a bitmap. From Hive V0.8.0 onwards, the bitmap index handler is built-in in Hive.
    - Compact Indexing: This type of indexing is known to store the column value and storage blockid.

18. What is the use of Hcatalog?

* + - HCatalog is a tool for accessing metadata that reside in Hive metastore.
    - HCatalog is a table and storage management layer for Hadoop that enables users with different data processing tools — Pig, MapReduce — to more easily read and write data.

19. Where is table data stored in Apache Hive by default?

* + - By default, table data is stored in default warehouse location: user/hive/warehouse. Use LOCATION to specify the directory where you want to reside your data in HDFS.

20. Are multi-line comments supported in Hive?

* + - No, Multi-line comments are not supported in Hive.

21. What is ObjectInspector functionality?

* + - Hive uses ObjectInspector to analyze the internal structure of the row object and also the structure of the individual columns. ObjectInspector provides a uniform way to access complex objects that can be stored in multiple formats in the memory, including: Instance of a Java class (Thrift or native Java)

22. Explain about the different types of join in Hive.

* + - HiveQL Select Joins: Hive inner join, hive left outer join, hive right outer join, and hive full outer join.

23. How can you configure remote metastore mode in Hive?

* + - To use this remote metastore, you should configure Hive service by setting hive.metastore.uris to the metastore server URI(s). Metastore server URIs are of the form thrift://host:port, where the port corresponds to the one set by METASTORE\_PORT when starting the metastore server.

25. How does data transfer happen from HDFS to Hive?

* + - If data is already present in HDFS then the user need not LOAD DATA that moves the files to the /user/hive/warehouse/. So the user just has to define the table using the keyword external that creates the table definition in the hive metastore.
    - LOAD DATA LOCAL INPATH 'hdfs://localhost:9000/test/bank-full.csv INTO TABLE BANK\_PART\_BUCKET partition (job 'management', education 'tertiary');

26. What are the different components of a Hive architecture?

* + - Hive Client
      * thrift clients, JDBC and ODBC clients.
    - Hive Services
      * Command-line interface (User Interface)
      * Hive Driver
      * Compiler
      * Execution Engine
      * Metastore
      * Optimizer
    - Processing and Resource Management
      * Hive internally uses a MapReduce framework as a defacto engine for executing the queries.
      * MapReduce is a software framework for writing those applications that process a massive amount of data in parallel on the large clusters of commodity hardware. MapReduce job works by splitting data into chunks, which are processed by map-reduce tasks.
    - Distributed Storage

27. Wherever (Different Directory) I run the hive query, it creates new metastore\_db, please explain the reason for it?

28. Is it possible to use the same metastore by multiple users, in case of the embedded hive?

* + - No, it is not possible to use metastore in sharing mode. It is recommended to use standalone "real" database like MySQL or PostGresSQL.

29. Usage of Hive.?

* + - Hive allows users to read, write, and manage petabytes of data using SQL. Hive is built on top of Apache Hadoop, which is an open-source framework used to efficiently store and process large datasets. As a result, Hive is closely integrated with Hadoop, and is designed to work quickly on petabytes of data.

30. Features and Limitations of Hive?

* + - Open-source: Apache Hive is an open-source tool. We can use it free of cost.
    - Query large datasets: Hive can query and manage huge datasets stored in Hadoop Distributed File System.
    - Multiple-users: Multiple users can query the data using Hive Query Language simultaneously.
    - Backward compatible: Apache Hive perfectly fits the low level interface requirement of Apache Hadoop.
    - Partitioning and Bucketing: Apache Hive supports partitioning and bucketing of data at the table level to improve performance.
    - File-formats: Hive provides support for various file formats such as textFile, ORC, Avro Files, SequenceFile, Parquet, RCFile, LZO Compression etc.
    - Hive Query Language: Hive uses Hive Query Language which is similar to SQL. We do not require any knowledge of programming languages to work with Hive. Only the knowledge of basic SQL query is enough to work with Hive.
    - Built-In function: Hive provides various Built-In functions.
    - User-Defined Functions: It also provides support for User-Defined Functions for the tasks like data cleansing and filtering. We can define UDFs according to our requirements
    - External Table: Apache Hive supports external tables. This allows us to process data without actually storing data in HDFS.
    - Fast: Hive is a fast, scalable, extensible tool and uses familiar concepts.
    - Warehouse: Apache Hive is a distributed data warehouse tool.
    - Table Structure: Table structure in Hive is similar to table structure in RDBMS.
    - ETL support: Hive supports ETL operations. Hive is an effective ETL tool.
    - Storage: Hive allows us to access files stored in HDFS and other similar data storage systems such as HBase.
    - OLAP: Hive is designed for OLAP (Online Analytical Processing).
    - Client application: Hive can support client applications written in PHP, Python, Java, C++ and Ruby.
    - Rule Based Optimizer: Hive has a rule based optimizer for optimizing logical plans.
    - Ad-hoc queries: Hive allows us to run Ad-hoc queries which are the loosely typed command or query whose value depends on some variable for the data analysis.
    - Data Visualization: Hive can be used for Data Visualization. Integrating Hive with Apache Tez will provide the real time processing capabilities.
  + Limitations of Apache Hive
    - Apache hive does not offer real-time queries and row level updates.
    - Latency of Apache Hive queries is generally very high.
    - Limited subquery support.
    - No support for materialized view.
    - update or delete operations are not supported in hive.
    - Not designed for OLTP(online transitional process).

<https://beyondcorner.com/learn-apache-hive/features-and-limitations-of-apache-hive/>

31. What are the components used in Hive Query Processor?

* + - Parse and Semantic Analysis (ql/parse)
    - Metadata Layer (ql/metadata)
    - Type Interfaces (ql/typeinfo)
    - Sessions (ql/session)
    - Map/Reduce Execution Engine (ql/exec)
    - Plan Components (ql/plan)
    - Hive Function Framework (ql/udf)
    - Tools (ql/tools)
    - Optimizer (ql/optimizer)

32. What is the precedence order of Hive configuration?

* + - SET Command in HIVE
    - The command line –hiveconf option
    - Hive-site.XML
    - Hive-default.xml
    - Hadoop-site.xml
    - Hadoop-default.xml

33. What are the uses of Hive Explode?

* + - explode(ARRAY<T> a)
      * Explodes an array to multiple rows. Returns a row-set with a single column (col), one row for each element from the array.
    - explode(MAP<Tkey,Tvalue> m)
      * Explodes a map to multiple rows. Returns a row-set with a two columns (key,value) , one row for each key-value pair from the input map.
    - posexplode(ARRAY<T> a)
      * Explodes an array to multiple rows with additional positional column of int type (position of items in the original array, starting with 0). Returns a row-set with two columns (pos,val), one row for each element from the array.
    - inline(ARRAY<STRUCT<f1:T1,...,fn:Tn>> a)
      * Explodes an array of structs to multiple rows. Returns a row-set with N columns (N = number of top level elements in the struct), one row per struct from the array.

34. Mention various data types supported by Hive?

* + - Overview
      * Numeric Types
      * Date/Time Types
      * String Types
      * Misc Types
      * Complex Types
    - Column Types
      * Integral Types (TINYINT, SMALLINT, INT/INTEGER, BIGINT)
      * Strings
      * Varchar
      * Char
      * Timestamps
        + Casting Dates
      * Intervals
      * Decimals
        + Decimal Literals
        + Decimal Type Incompatibilities between Hive 0.12.0 and 0.13.0

Upgrading Pre-Hive 0.13.0 Decimal Columns

* + - * Union Types
    - Literals
      * Floating Point Types
        + Decimal Types

Using Decimal Types

Mathematical UDFs

Casting Decimal Values

Testing Decimal Types

* + - Handling of NULL Values
    - Change Types
    - Allowed Implicit Conversions

36. Can a table be renamed in Hive?

* + - ALTER TABLE table\_name RENAME TO new\_table\_name; This statement lets you change the name of a table to a different name. As of version 0.6, a rename on a managed table moves its HDFS location as well. (Older Hive versions just renamed the table in the metastore without moving the HDFS location.)

37. What are the three different modes in which hive can be run?

* + - Standalone(Local) Mode
      * By default, Hadoop is configured to run in a no distributed mode. It runs as a single Java process. Instead of HDFS, this mode utilizes the local file system. This mode useful for debugging and there isn't any need to configure core-site.xml, hdfs-site.xml, mapred-site.xml, masters & slaves. Stand-alone mode is usually the fastest mode in Hadoop.
    - Pseudo-Distributed Mode(Single node Cluster) (Client Mode)
      * Hadoop can also run on a single node in a Pseudo Distributed mode. In this mode, each daemon runs on separate java process. In this mode custom configuration is required( core-site.xml, hdfs-site.xml, mapred-site.xml ). Here HDFS is utilized for input and output. This mode of deployment is useful for testing and debugging purposes.
      * All the daemons that are Namenode, Datanode, Secondary Name node, Resource Manager, Node Manager, etc. will be running as a separate process on separate JVM(Java Virtual Machine) or we can say run on different java processes that is why it is called a Pseudo-distributed.
    - Fully Distributed Mode(Multi node Cluster)(Cluster Mode)
      * This is the production mode of Hadoop. In this mode typically one machine in the cluster is designated as NameNode and another as Resource Manager exclusively. These are masters. All other nodes act as Data Node and Node Manager. These are the slaves. Configuration parameters and environment need to specified for Hadoop Daemons.
      * This mode offers fully distributed computing capability, reliability, fault tolerance and scalability.

For more questions refer: https://www.tutorialspoint.com/hive/hive\_interview\_questions.htm

Spark Questions:

1. How is Apache Spark different from MapReduce?

* + - MapReduce Spark
    - It is having a very slow speed as compared to Apache Spark. It is much faster than MapReduce.
    - It is unable to handle real-time processing. It can deal with real-time processing.
    - It supports more security projects. Its security is not as good as MapReduce and continuously working on its security issues.
    - For performing the task, It is unable to cache in memory. It can cache the memory data for processing its task.

2. What are the important components of the Spark ecosystem?

* + - 6 components in Apache Spark Ecosystem which empower to Apache Spark- Spark Core, Spark SQL, Spark Streaming, Spark MLlib, Spark GraphX, and SparkR

3. Explain how Spark runs applications with the help of its architecture?

4. What are the different cluster managers available in Apache Spark?

* + - Standalone cluster manager, Hadoop YARN and Apache Mesos.

5. What is the significance of Resilient Distributed Datasets in Spark?

* + - RDD is core abstraction of spark
    - Provides data lineage of spark partitions on data node
    - Fault tolerant - partition will be recreated in an other machine in case of data loss using lineage
    - RDD is immutable(it can’t be changed) but can be transformed
    - ####RDD was the primary user-facing API in Spark since its inception. At the core, an RDD is an immutable distributed collection of elements of your data, partitioned across nodes in your cluster that can be operated in parallel with a low-level API that offers transformations and actions.

6. What is a lazy evaluation in Spark?

* + - it evaluates something only when we require it. In accordance with a spark, it does not execute each operation right away, that means it does not start until we trigger any action. Once an action is called all the transformations will execute in one go.
    - Example of actions: count, take, show(truncate=True/False), collect(store data on driver node- Bad of processing)
    - Examples of transformations(one form of RDD converts to other form(big to small without changing anything)):map(), flatmap(), filter(), distinct(), join()
    - Advantages:
      * Reduces Complexities
      * Optimization
      * Develops Manageability
      * Saves computation and increases speed

7. What is a Parquet file and what are its advantages?

* + - Parquet is a file format designed to support fast data processing for complex data, with several notable characteristics:
      * Columnar: Unlike row-based formats such as CSV or Avro, Apache Parquet is column-oriented – meaning the values of each table column are stored next to each other, rather than those of each record
      * Open-source: Parquet is free to use and open source under the Apache Hadoop license, and is compatible with most Hadoop data processing frameworks.
      * Self-describing: In Parquet, metadata including schema and structure is embedded within each file, making it a self-describing file format.
    - Advantages
      * Compression
      * Parquet data can be compressed using these encoding methods:
        + Dictionary encoding
        + Bit packing
        + Run length encoding
      * Performance
      * Schema evolution(it will find schema of data)
      * Open-source support

8. What is shuffling in Spark? When does it occur?

* + - A shuffle occurs when data is rearranged between partitions. This is required when a transformation requires information from other partitions, such as summing all the values in a column. Spark will gather the required data from each partition and combine it into a new partition, likely on a different executor.
    - During a shuffle, data is written to disk and transferred across the network, halting Spark’s ability to do processing in-memory and causing a performance bottleneck. Consequently we want to try to reduce the number of shuffles being done or reduce the amount of data being shuffled.
    - The repartition method can be used to either increase or decrease the number of partitions in a DataFrame. Repartition is a full Shuffle operation, whole data is taken out from existing partitions and equally distributed into newly formed partitions.

9. What is the use of coalesce in Spark?

* + - The coalesce method reduces the number of partitions in a DataFrame. Coalesce avoids full shuffle, instead of creating new partitions, it shuffles the data using Hash Partitioner (Default), and adjusts into existing partitions, this means it can only decrease the number of partitions.

10. What are the various functionalities supported by Spark Core?

* + - It is in charge of essential I/O functionalities.
    - Significant in programming and observing the role of the Spark cluster.
    - Task dispatching.
    - Fault recovery.
    - It overcomes the snag of MapReduce by using in-memory computation.

11. Explain the types of operations supported by RDDs?

* + - Transformation: It is a function that produces new RDD from the existing RDDs.
    - Action: In Transformation, RDDs are created from each other. But when we want to work with the actual dataset, then, at that point we use Action.

12. What is the need for broadcast variables in Spark?

* + - Broadcast variables in Apache Spark is a mechanism for sharing variables across executors that are meant to be read-only. Without broadcast variables these variables would be shipped to each executor for every transformation and action, and this can cause network overhead. However, with broadcast variables, they are shipped once to all executors and are cached for future reference.
    - Broadcast Variables Use case
    - Imagine that while doing a transformation we need to lookup a large table of zip codes/pin codes. Here, it is neither feasible to send the large lookup table every time to the executors, nor can we query the database every time. The solution should be to convert this lookup table to a broadcast variables and Spark will cache it in every executor for future reference.

13. What are the different levels of persistence in Spark?

* + - Spark has various persistence levels to store the RDDs on disk or in memory or as a combination of both with different replication levels namely:
      * MEMORY\_ONLY
      * MEMORY\_ONLY\_SER
      * MEMORY\_AND\_DISK
      * MEMORY\_AND\_DISK\_SER, DISK\_ONLY
      * OFF\_HEAP

14. What is the difference between map and flatMap transformation in Spark Streaming?

* + - Spark map function expresses a one-to-one transformation. It transforms each element of a collection into one element of the resulting collection. While Spark flatMap function expresses a one-to-many transformation. It transforms each element to 0 or more elements.

15. What are the functions of Spark SQL?

16. How can you connect Hive to Spark SQL?

17. What is the role of Catalyst Optimizer in Spark SQL?

* + - The Spark SQL Catalyst Optimizer improves developer productivity and the performance of their written queries. Catalyst automatically transforms relational queries to execute them more efficiently using techniques such as filtering, indexes and ensuring that data source joins are performed in the most efficient order. In addition, its design allows the Spark community to implement and extend the optimizer with new features.
    - We use Catalyst’s general tree transformation framework in four phases, as shown below: (1) analyzing a logical plan to resolve references, (2) logical plan optimization, (3) physical planning, and (4) code generation to compile parts of the query to Java bytecode.
    - It is being used in Data frames,

18. Diff between RDD and DataFrame?

— Fill answer

18. What are the key features of Apache Spark that you like?

19. Which one will you choose for a project –Hadoop MapReduce or Apache Spark?

20. What is the difference between persist() and cache()? — Spark optimization

* + - The difference between cache() and persist() is that using cache() the default storage level is MEMORY\_ONLY while using persist() we can use various storage levels
    - Use persist() if you want to assign a storage level other than :
      * MEMORY\_ONLY to the RDD
      * or MEMORY\_AND\_DISK for Dataset
    - Both caching and persisting are used to save the Spark RDD, Dataframe and Dataset’s. But, the difference is, RDD cache() method default saves it to memory (MEMORY\_ONLY) whereas persist() method is used to store it to user-defined storage level.
    - When you persist a dataset, each node stores it’s partitioned data in memory and reuses them in other actions on that dataset. And Spark’s persisted data on nodes are fault-tolerant meaning if any partition of a Dataset is lost, it will automatically be recomputed using the original transformations that created it.
    - Below are the advantages of using Spark Cache and Persist methods.
      * Cost efficient – Spark computations are very expensive hence reusing the computations are used to save cost.
      * Time efficient – Reusing the repeated computations saves lots of time.
      * Execution time – Saves execution time of the job and we can perform more jobs on the same cluster.

21. Explain about the core components of a distributed Spark application.?

* + - Core components of a distributed application in Apache Spark are as follows:
      * I. Cluster Manager: This is the component responsible for launching executors and drivers on multiple nodes. We can use different types of cluster managers based on our requirements. Some of the common types are Standalone, YARN, Mesos etc.
      * II. Driver: This is the main program in Spark that runs the main() function of an application. A Driver program creates the SparkConetxt. Driver program listens and accepts incoming connections from its executors. Driver program can schedule tasks on the cluster. It runs closer to worker nodes.
      * III. Executor: This is a process on worker node. It is launched on the node to run an application. It can run tasks and use data in memory or disk storage to perform the task.

22. Explain about the common workflow of a Spark program?

<https://www.dezyre.com/article/apache-spark-architecture-explained-in-detail/338/>

<https://data-flair.training/blogs/how-apache-spark-works/>

23. Explain the difference between Spark SQL and Hive.

<https://data-flair.training/blogs/apache-hive-vs-spark-sql/>

|  |  |  |
| --- | --- | --- |
| **Basis of Comparison** | **Apache Hive** | **Apache Spark SQL** |
| **Structure** | An open source data warehousing system which is built on top of Hadoop | Mainly used for structured data processing where more information is retrieved by using structured query language. |
| **Processing** | Large datasets which are stored in hadoop files are analyzed and queried. Processing is mainly performed using SQL. | The processing of Apache [Spark SQL involves](https://www.educba.com/spark-sql-dataframe/) heavy computations performed due to which a right optimization technique is required. Interaction with Spark SQL is possible in different ways such as Dataset and DataFrame API. |
| **Latest Release** | November 2017 : release 2.3.2 | February 2018: 2.3.0 |
| **Implementation language** | Java language primarily can be used to implement apache Hive | Spark SQL can be implemented on Scala, Java, [R as well as Python](https://www.educba.com/r-vs-python/) |
| **Database model** | Primarily its database model is RDBMS | Though Spark SQL is capable of integrating with any NoSQL database but primarily its database model is RDBMS |
| **Additional Database Models** | Additional database model is a key-value store which can take data in the form of JSON | Key-value store is the additional database model |
| **Development** | Hive was originally developed by Facebook but later on donated to Apache Software foundation | It was originally developed by Apache Software Foundation itself |
| **Server Operating System** | It supports all operating system with a Java Virtual machine environment | It supports several operating systems such as Windows, X, Linux etc. |
| **Access Methods** | It supports ODBC, JDBC and Thrift | It only supports ODBC and JDBC |
| **Programming Language Support** | Several programming languages such as C++, PHP, Java, Python, etc. are supported | Several programming languages such as Java, R, Python, and Scala is supported |
| **Partitioning Methods** | Data sharding method is used to store data on various nodes | It makes [use of Apache Spark](https://www.educba.com/apache-spark-architecture/) Core for storing data on various nodes |

24. Spark optimization techniques?

<https://www.upgrad.com/blog/spark-optimization-techniques/>

<https://www.analyticsvidhya.com/blog/2020/11/8-must-know-spark-optimization-tips-for-data-engineering-beginners/>

25. what is data skew in spark? how to handle it?

<https://dataengi.com/2019/02/06/spark-data-skew-problem/>

<https://dzone.com/articles/improving-the-performance-of-your-spark-job-on-ske>

salting

Reference:

https://www.simplilearn.com/top-apache-spark-interview-questions-and-answers-article

https://www.dezyre.com/article/top-50-spark-interview-questions-and-answers-for-2021/208/

https://www.edureka.co/blog/interview-questions/top-apache-spark-interview-questions-2016/

Airflow Questions:

1.What is airflow? Architecture? components?

<https://www.mockquestions.com/company/Airflow+Systems%2C+Inc./>

MapReduce Architecture Questions:

1.What is the difference between HDFS block and InputSplit?

* + - Block – It is the physical representation of data. It contains a minimum amount of data that can be read or write.
    - InputSplit – It is the logical representation of data present in the block. It is used during data processing in MapReduce program or other processing techniques. InputSplit doesn’t contain actual data, but a reference to the data.

2.What is MapReduce?

* + - MapReduce is a processing technique and a program model for distributed computing based on java. The MapReduce algorithm contains two important tasks, namely Map and Reduce. Map takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs). Secondly, reduce task, which takes the output from a map as an input and combines those data tuples into a smaller set of tuples. As the sequence of the name MapReduce implies, the reduce task is always performed after the map job.

3.Illustrate a simple example of the working of MapReduce?

4. What is a “map” in Hadoop?

* + - Map stage − The map or mapper’s job is to process the input data. Generally the input data is in the form of file or directory and is stored in the Hadoop file system (HDFS). The input file is passed to the mapper function line by line. The mapper processes the data and creates several small chunks of data.

5. What is a “reducer” in Hadoop?

* + - Reduce stage − This stage is the combination of the Shuffle stage and the Reduce stage. The Reducer’s job is to process the data that comes from the mapper. After processing, it produces a new set of output, which will be stored in the HDFS.

6. Compare Spark and MapReduce?

* + - see above

7. Can you tell us how many daemon processes run on a Hadoop system?

* + - Hadoop is comprised of five separate daemons. Each of these daemons runs in its own JVM.
    - Following 3 Daemons run on Master
      * nodes.NameNode - This daemon stores and maintains the metadata for HDFS.
      * Secondary NameNode - Performs housekeeping functions for the NameNode. JobTracker - Manages MapReduce jobs, distributes individual tasks to machines running the Task Tracker. Following 2 Daemons run on each Slave nodes DataNode – Stores actual HDFS data blocks
      * TaskTracker – It is Responsible for instantiating and monitoring individual Map and Reduce task

CI/CD Questions:

Git:

1. What is Git?

* + - * modern version control system in the world today is Git.

2. What do you understand by the term ‘Version Control System’?

* + - * Version control systems are a category of software tools that helps in recording changes made to files by keeping a track of modifications done to the code.

3. What’s the difference between Git and GitHub?

* + - * git is a local VCS software that enables developers to save snapshots of their projects over time. It’s generally best for individual use.
      * GitHub is a web-based platform that incorporates git’s version control features so they can be used collaboratively. It also includes project and team management features, as well as opportunities for networking and social coding.

4. What is a Git repository?

* + - * Repositories in GIT contain a collection of files of various different versions of a Project. These files are imported from the repository into the local server of the user for further updations and modifications in the content of the file. A VCS or the Version Control System is used to create these versions and store them in a specific place termed as a repository. The process of copying the content from an existing Git Repository with the help of various Git Tools is termed as cloning. Once the cloning process is done, the user gets the complete repository on his local machine. Git by default assumes the work to be done on the repository is as a user, once the cloning is done.

5. Name a few Git commands with their function.

* + - * clone
      * push
      * pull
      * commit
      * add
      * revert
      * checkout
      * status
      * merge

6. What are the advantages of using Git?

* + - * One of the biggest advantages of Git is its branching capabilities. Unlike centralized version control systems, Git branches are cheap and easy to merge. This facilitates the feature branch workflow popular with many Git users. Feature branches provide an isolated environment for every change to your codebase.

7. Name some of the popular Git hosting repositories?

* + - * GitHub
      * gitlab
      * bit bucket
      * source forge

8. Explain the git push command?

* + - * After committing your changes, the next thing you want to do is send your changes to the remote server. Git push uploads your commits to the remote repository.
      * git push <remote> <branch-name>

9. Explain the git pull command?

* + - * The git pull command is used to get updates from the remote repo. This command is a combination of git fetch and git merge which means that, when we use git pull, it gets the updates from remote repository (git fetch) and immediately applies the latest changes in your local (git merge).

10. How do you resolve a merge conflict?

* + - * A merge conflict is an event that takes place when Git is unable to automatically resolve differences in code between two commits. Git can merge the changes automatically only if the commits are on different lines or branches.
      * There are a few steps that could reduce the steps needed to resolve merge conflicts in Git.
        + The easiest way to resolve a conflicted file is to open it and make any necessary changes
        + After editing the file, we can use the git add a command to stage the new merged content
        + The final step is to create a new commit with the help of the git commit command
        + Git will create a new merge commit to finalize the merge

11. What is the difference between fork, branch, and clone?

* + - * Branching and forking provide two ways of diverging from the main code line.
      * A repository code branch, like a branch of a tree, remains part of the original repository. The code that is branched (main trunk) and the branch know and rely on each other. Like a tree trunk's branch, a code branch knows about the trunk (original code base) it originated from.
      * Fork is another way of saying clone or copy. The term fork (in programming) derives from an Unix system call that creates a copy of an existing process. So, unlike a branch, a fork is independent from the original repository. If the original repository is deleted, the fork remains. If you fork a repository, you get that repository and all of its branches.
      * When you want to work on a project by updating its files or adding new files, you need to make a local clone of the remote GitHub/Bitbucket repository onto your machine or local network. You do this using the Clone button from the remote repository. If you forked a repository, you simply clone the fork. If you branched a repository, you clone the repository and checkout the branch.

12.How is ‘git remote’ different from ‘git clone’?

* + - * They are two completely different things. git remote is used to refer to a remote repository or your central repository. git clone is used to copy or clone a different repository.

ref:

https://www.edureka.co/blog/interview-questions/git-interview-questions/

https://www.simplilearn.com/tutorials/git-tutorial/git-interview-questions

Jenkins:

1. What is Jenkins?

* + - * Jenkins offers a simple way to set up a continuous integration and continuous delivery environment for almost any combination of languages and source code repositories using pipelines, as well as automating other routine development tasks. While Jenkins doesn’t eliminate the need to create scripts for individual steps, it does give you a faster and more robust way to integrate your entire chain of build, test, and deployment tools than you can easily build yourself.

2. Define the process of Jenkins?

* + - * To operate Jenkins, pipelines are created. A pipeline is a series of steps the Jenkins server will take to perform the required tasks of the CI/CD process. These are stored in a plain text Jenkinsfile. The Jenkinsfile uses a curly bracket syntax that looks similar to JSON. Steps in the pipeline are declared as commands with parameters and encapsulated in curly brackets. The Jenkins server then reads the Jenkinsfile and executes its commands, pushing the code down the pipeline from committed source code to production runtime. A Jenkinsfile can be created through a GUI or by writing code directly.

3. What are the benefits of using Jenkins?

* + - * It is an open-source tool with great community support.
      * It is easy to install.
      * It has 1000+ plugins to ease your work. ...
      * It is free of cost.
      * It is built with Java and hence, it is portable to all the major platforms.

4. What are the pre-requisites for using Jenkins?

* + - * Operating system--Linux versions of Ubuntu/Debian, Red Hat/Fedora/CentOS, openSUSE, FreeBSD, OpenBSD, Gentoo, Windows, macOS X.
      * JDK version.
      * Memory.
      * Disk space.
      * Java Containers--The Jenkins WAR file can run on any servlet-supported engine such as tomcat or Glassfish application servers.

5. Mention some of the useful plugins in Jenkins?

* + - * Kubernetes. The “Kubernetes” plugin is great for automating build agents on a Kubernetes cluster. ...
      * Swarm. This interesting plugin is useful if you plan on using Docker Swarm. ...
      * Amazon Elastic Container Service. …
      * GitHub
      * Azure Container Service.

6. What are the two components that you can integrate Jenkins with?

* + - * Jenkins is mainly integrated with two components. These include version systems and build tools. While Git and SVN are examples of the former, Apache Maven is an example of the latter.

7. What is Maven? What is the benefit of integrating Maven with Jenkins? not needed

* + - * Maven is a powerful project / build management tool, based on the concept of a POM (Project Object Model) that includes project information and configuration information for Maven such as construction directory, source directory, dependency, test source directory, Goals, plugins, etc.

8. How Can You Clone A Git Repository Via Jenkins? not needed

* + - * There are two ways to clone the project(repository) from the Github. Create a new Jenkins job called 'Clone-with-https', move to the “Source Control Management” setting and choose “Git” options if you cannot see the Git options that mean 'GitHub' plugin wasn't installed in the Jenkins machine.

9. Explain how you can set up Jenkins job?

10.What are the types of pipelines in Jenkins? not needed

* + - * A Jenkinsfile can be written using two types of syntax - Declarative and Scripted.
      * Declarative and Scripted Pipelines are constructed fundamentally differently. Declarative Pipeline is a more recent feature of Jenkins Pipeline which:
      * provides richer syntactical features over Scripted Pipeline syntax, and
      * is designed to make writing and reading Pipeline code easier.

11.Do you know any other Continuous Integration tools? How is Jenkins better than any of those?

circle CI

BMX

12.How do you achieve continuous integration using Jenkins?

13. How do you integrate Git with Jenkins?

jenkinfile - you write

series of command for jenkins server to be executed(code deploy instructions like where to pull it from and where to deploy)

web hook to git repo

git repo setup in jenkins

https://www.edureka.co/blog/interview-questions/jenkins-interview-questions/

https://www.guru99.com/jenkins-interview-questions.html

Docker/Kuberneties:

1.Do you know about cloud computing? How can Jenkins fit into a cloud computing environment? Explain with an example?

2.What is Kubernetes? How can you integrate Jenkins with Kubernetes?

3.Have you run automated tests on Jenkins? How is it done?

DWH Questions:

1.What is Datawarehousing?

* + - Data warehousing is the process of constructing and using a data warehouse. A data warehouse is constructed by integrating data from multiple heterogeneous sources that support analytical reporting, structured and/or ad hoc queries, and decision making. Data warehousing involves data cleaning, data integration, and data consolidations
    - A Data Warehousing (DW) is process for collecting and managing data from varied sources to provide meaningful business insights. A Data warehouse is typically used to connect and analyze business data from heterogeneous sources. The data warehouse is the core of the BI system which is built for data analysis and reporting.

2.What is Business Intelligence?

* + - Business intelligence (BI) leverages software and services to transform data into actionable insights that inform an organization’s strategic and tactical business decisions. BI tools access and analyze data sets and present analytical findings in reports, summaries, dashboards, graphs, charts and maps to provide users with detailed intelligence about the state of the business.
    - The term business intelligence often also refers to a range of tools that provide quick, easy-to-digest access to insights about an organization's current state, based on available data.
    - Business intelligence (BI) combines business analytics, data mining, data visualization, data tools and infrastructure, and best practices to help organizations to make more data-driven decisions.

3.What is Dimension Table?

* + - Dimension table is a table which contain attributes of measurements stored in fact tables
    - A dimension table contains dimensions of a fact.
    - They are joined to fact table via a foreign key.
    - Dimension tables are de-normalized tables.
    - The Dimension Attributes are the various columns in a dimension table
    - Dimensions offers descriptive characteristics of the facts with the help of their attributes
    - No set limit set for given for number of dimensions
    - The dimension can also contain one or more hierarchical relationships

4.What is Fact Table?

* + - A fact table is a primary table in a dimensional model.
    - A Fact Table contains
      * Measurements/facts
      * Foreign key to dimension table
    - Fact table contains the measurement of business processes, and it contains foreign keys for the dimension tables.
    - Example – If the business process is manufacturing of bricks
      * Average number of bricks produced by one person/machine – measure of the business process

5.What are the stages of Datawarehousing?

* + - Offline Operational Databases: This is the initial stage of data warehousing. In this stage the development of database of an operational system to an off-line server is done by simply copying the databases.
    - Offline Data warehouse: In this stage the data warehouses are updated on a regular time cycle from operational system and the data is persisted in a reporting-oriented data structure.
    - Real time Data Warehouse: Data warehouses are updated based on transaction or event basis in this stage. An operational system performs a transaction every time.
    - Integrated Data Warehouse: The activity or transactions generation which are passed back into the operational system is done in this stage. These transactions or generated transactions are used in the daily activity of the organization.

6.What is Data Mining?

* + - Data mining is a process used by companies to turn raw data into useful information. By using software to look for patterns in large batches of data, businesses can learn more about their customers to develop more effective marketing strategies, increase sales and decrease costs. Data mining depends on effective data collection, warehousing, and computer processing.

7.What is OLTP?

* + - OLTP (Online Transactional Processing) is a category of data processing that is focused on transaction-oriented tasks. OLTP typically involves inserting, updating, and/or deleting small amounts of data in a database. OLTP mainly deals with large numbers of transactions by a large number of users.

8.What is OLAP?

* + - OLAP is an acronym for Online Analytical Processing. OLAP performs multidimensional analysis of business data and provides the capability for complex calculations, trend analysis, and sophisticated data modeling.

9.What is the difference between OLTP and OLAP?

* + - Online Analytical Processing (OLAP) is a category of software tools that analyze data stored in a database whereas Online transaction processing (OLTP) supports transaction-oriented applications in a 3-tier architecture.
    - OLAP creates a single platform for all type of business analysis needs which includes planning, budgeting, forecasting, and analysis while OLTP is useful to administer day to day transactions of an organization.
    - OLAP is characterized by a large volume of data while OLTP is characterized by large numbers of short online transactions.
    - In OLAP, data warehouse is created uniquely so that it can integrate different data sources for building a consolidated database whereas OLTP uses traditional DBMS.

10.What is ODS?

* + - an ODS aggregates transactional data from multiple sources. Traditionally the ODS was designed and optimized for operational reporting, and was refreshed on a daily or even an hourly basis. Usually the ODS only stores a short time window worth of data.

11.What is the difference between View and Materialized View?

* + - The basic difference between View and Materialized View is that Views are not stored physically on the disk. On the other hands, Materialized Views are stored on the disc.
    - View can be defined as a virtual table created as a result of the query expression. However, Materialized View is a physical copy, picture or snapshot of the base table.
    - A view is always updated as the query creating View executes each time the View is used. On the other hands, Materialized View is updated manually or by applying triggers to it.
    - Materialized View responds faster than View as the Materialized View is precomputed.
    - Materialized View utilizes the memory space as it stored on the disk whereas, the View is just a display hence it do not require memory space.

12.What are Aggregate tables?

* + - Aggregated tables enable an organization to create an aggregated version of almost any table in the project. Sometimes it may not be necessary to see the transaction-level detail in financial or sales reports, but only data grouped by business unit, department, or sales team. This makes the aggregated tables feature very useful for reporting directly from the data warehouse as opposed to using OLAP cubes.

13.What are Non-additive facts?

* + - Additive:
      * Additive facts are facts that can be summed up through all of the dimensions in the fact table. A sales fact is a good example for additive fact.
    - Semi-Additive:
      * Semi-additive facts are facts that can be summed up for some of the dimensions in the fact table, but not the others.
      * Eg: Daily balances fact can be summed up through the customers dimension but not through the time dimension.
    - Non-Additive:
      * Non-additive facts are facts that cannot be summed up for any of the dimensions present in the fact table.
      * Eg: Facts which have percentages, ratios calculated.

14.What is factless fact tables? not needed

* + - Factless facts are those fact tables that have no measures associated with the transaction. Factless facts are a simple collection of dimensional keys which define the transactions or describing condition for the time period of the fact.
    - You may question the need for a factless fact but they are important dimensional data structures which capture important information which can be leveraged into rollup measures or as information presented to a user.
    - The most common example used for factless facts are student attendance in a class. As you can see from the dimensional diagram below the FACT\_ATTENDANCE is an amalgamation of the DATE\_KEY, the STUDENT\_KEY, and the CLASS\_KEY.

15.What is Datamart?

* + - A data mart is a subset of a data warehouse focused on a particular line of business, department, or subject area. Data marts make specific data available to a defined group of users, which allows those users to quickly access critical insights without wasting time searching through an entire data warehouse. For example, many companies may have a data mart that aligns with a specific department in the business, such as finance, sales, or marketing.

16.What is the difference between Datawarehouse and OLAP?

* + - A data warehouse serves as a repository to store historical data that can be used for analysis. OLAP is Online Analytical processing that can be used to analyze and evaluate data in a warehouse. The warehouse has data coming from varied sources. OLAP tool helps to organize data in the warehouse using multidimensional models.
    - A data mart is a subset of a data warehouse oriented to a specific business line. Data marts contain repositories of summarized data collected for analysis on a specific section or unit within an organization, for example, the sales department.
    - A data warehouse is a large centralized repository of data that contains information from many sources within an organization. The collated data is used to guide business decisions through analysis, reporting, and data mining tools.
    - Data Mart
      * Focus: A single subject or functional organization area
      * Data Sources: Relatively few sources linked to one line of business
      * Size: Less than 100 GB
      * Normalization: No preference between a normalized and denormalized structure
      * Decision Types: Tactical decisions pertaining to particular business lines and ways of doing things
      * Cost: Typically from $10,000 upwards
      * Setup Time: 3-6 months
      * Data Held: Typically summarized data
    - Data Warehouse
      * Focus: Enterprise-wide repository of disparate data sources
      * Data Sources: Many external and internal sources from different areas of an organization
      * Size: 100 GB minimum but often in the range of terabytes for large organizations
      * Normalization: Modern warehouses are mostly denormalized for quicker data querying and read performance
      * Decision Types: Strategic decisions that affect the entire enterprise
      * Cost: Varies but often greater than $100,000; for cloud solutions costs can be dramatically lower as organizations pay per use
      * Setup Time: At least a year for on-premise warehouses; cloud data warehouses are much quicker to set up
      * Data Held: Raw data, metadata, and summary data

17.What are the key columns in Fact and dimension tables?

18.What is SCD?

19.What are the types of SCD?

20.What is Star Schema?

21.What is Snowflake Schema?

22.What is called Dimensional Modelling?

23.What is surrogate key?

24.What are the different types of datawarehosuing?

25.What is the difference between metadata and data dictionary?

Ref:

https://career.guru99.com/top-50-datawarehousing-questions-answers/

https://www.janbasktraining.com/blog/data-warehouse-interview-questions/

https://mindmajix.com/data-warehouse-interview-questions

Python Questions:

https://www.edureka.co/blog/interview-questions/python-interview-questions/

https://codingcompiler.com/python-coding-interview-questions-answers/

Project related Questions:

1. explian design/architecture of project?

2. what are the data sources?

3. explain the ingestion process?

4. what is the business value of the project?

5. what is the size of biggest table?

6. how many rows of data?

7. how many tables you have for this project?

8. explain where you have used pyspark?

9. what are the development challenges you have seen while working on this process?

10. explain your role/responsibilities in this project?

ACID properties