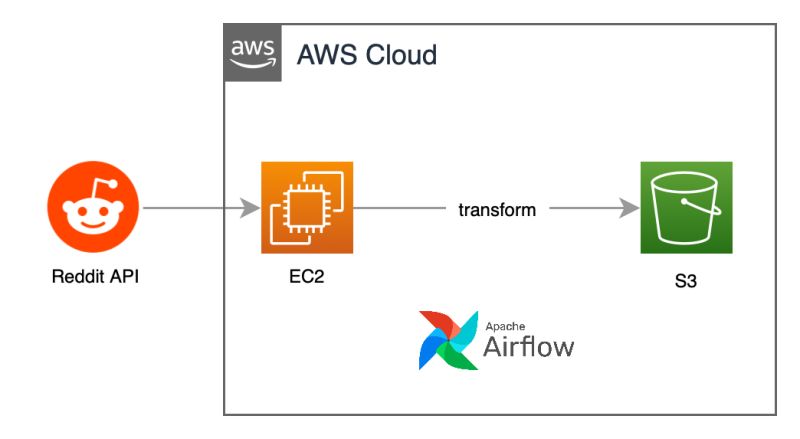
Recently, I made a Simple ETL Pipeline using Airflow and Reddit API.  
  
Here's the workflow of the Pipeline:  
1. Data is extracted from the Reddit API using Python Script  
2. The data is then transformed accordingly using Pandas  
3. Transformed data is then saved and loaded into an S3 Bucket as a csv file.  
  
An IAM role is assigned to the EC2 instance with a policy to Allow Full S3 Access. This process is orchestrated using Airflow  
  
Project Link -> <https://lnkd.in/dueeS7YK>

Project Credits : Thanks to [Darshil Parmar](https://www.linkedin.com/in/ACoAAB5QzkIB7H3vliuE1puA21D60J9uTPkTMpw) for this amazing project, learned a lot!  
  
Since the Twitter API is not free anymore, I decided to use Reddit API.  
  
[#etl](https://www.linkedin.com/feed/hashtag/?keywords=etl&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A7052961251852787712) [#airflow](https://www.linkedin.com/feed/hashtag/?keywords=airflow&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A7052961251852787712) [#aws](https://www.linkedin.com/feed/hashtag/?keywords=aws&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A7052961251852787712) [#s3](https://www.linkedin.com/feed/hashtag/?keywords=s3&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A7052961251852787712) [#ec2](https://www.linkedin.com/feed/hashtag/?keywords=ec2&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A7052961251852787712) [#pipeline](https://www.linkedin.com/feed/hashtag/?keywords=pipeline&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A7052961251852787712) [#pandas](https://www.linkedin.com/feed/hashtag/?keywords=pandas&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A7052961251852787712) [#dataengineer](https://www.linkedin.com/feed/hashtag/?keywords=dataengineer&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A7052961251852787712) [#dataengineering](https://www.linkedin.com/feed/hashtag/?keywords=dataengineering&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A7052961251852787712)

Activate to view larger image,



Hi everyone - I am looking for a new role and would appreciate your support. Thank you in advance for any connections, advice, or opportunities you can offer. #OpenToWork

Apache Spark Best Practices in Data Engineering.  
  
1. 𝗗𝗮𝘁𝗮 𝗨𝗻𝗱𝗲𝗿𝘀𝘁𝗮𝗻𝗱𝗶𝗻𝗴 𝗮𝗻𝗱 𝗣𝗿𝗲𝗽𝗮𝗿𝗮𝘁𝗶𝗼𝗻:  
- Understand your data and its characteristics thoroughly before starting.  
- Preprocess and clean your data to remove inconsistencies and errors.  
  
2. 𝗖𝗹𝘂𝘀𝘁𝗲𝗿 𝗖𝗼𝗻𝗳𝗶𝗴𝘂𝗿𝗮𝘁𝗶𝗼𝗻 𝗮𝗻𝗱 𝗥𝗲𝘀𝗼𝘂𝗿𝗰𝗲 𝗠𝗮𝗻𝗮𝗴𝗲𝗺𝗲𝗻𝘁:  
- Configure Spark cluster settings according to the hardware and resources available.  
- Optimize memory, CPU cores, and executor instances based on the workload.  
  
3. 𝗖𝗵𝗼𝗼𝘀𝗶𝗻𝗴 𝘁𝗵𝗲 𝗥𝗶𝗴𝗵𝘁 𝗧𝗿𝗮𝗻𝘀𝗳𝗼𝗿𝗺𝗮𝘁𝗶𝗼𝗻𝘀 𝗮𝗻𝗱 𝗔𝗰𝘁𝗶𝗼𝗻𝘀:  
- Use transformations like `map`, `filter`, `join`, and `groupBy` wisely to minimize data shuffling and reduce unnecessary overhead.  
  
4. 𝗖𝗮𝗰𝗵𝗶𝗻𝗴 𝗮𝗻𝗱 𝗣𝗲𝗿𝘀𝗶𝘀𝘁𝗲𝗻𝗰𝗲:  
- Cache intermediate data using `cache()` or `persist()` when reused multiple times to avoid recomputation.  
- However, be cautious not to cache too much data, as it might lead to memory issues.  
  
5. 𝗣𝗮𝗿𝘁𝗶𝘁𝗶𝗼𝗻𝗶𝗻𝗴 𝗮𝗻𝗱 𝗗𝗮𝘁𝗮 𝗟𝗮𝘆𝗼𝘂𝘁:  
- Choose appropriate partitioning strategies to balance data distribution across nodes.  
- Use columnar storage formats like Parquet for efficient data storage and compression.  
  
6. 𝗦𝗵𝘂𝗳𝗳𝗹𝗶𝗻𝗴 𝗢𝗽𝘁𝗶𝗺𝗶𝘇𝗮𝘁𝗶𝗼𝗻:  
- Minimize shuffling by using transformations like `reduceByKey` instead of `groupByKey`.  
- Consider using operations like `mapPartitions` to process data partition-wise.  
  
7. 𝗕𝗿𝗼𝗮𝗱𝗰𝗮𝘀𝘁𝗶𝗻𝗴:  
- Broadcast small datasets to all nodes when performing joins, reducing the need for data shuffling.  
  
8. 𝗣𝗮𝗿𝗮𝗹𝗹𝗲𝗹𝗶𝘀𝗺 𝗮𝗻𝗱 𝗗𝗮𝘁𝗮 𝗟𝗼𝗰𝗮𝗹𝗶𝘁𝘆:  
- Ensure tasks are distributed evenly across nodes by setting appropriate parallelism levels.  
- Leverage data locality by co-locating computation with data using `preferredLocations`.  
  
9. 𝗠𝗼𝗻𝗶𝘁𝗼𝗿𝗶𝗻𝗴 𝗮𝗻𝗱 𝗣𝗿𝗼𝗳𝗶𝗹𝗶𝗻𝗴:  
- Monitor cluster performance using Spark's built-in web UI and logging.  
- Profile your application to identify bottlenecks and resource usage.  
  
10. 𝗧𝗲𝘀𝘁𝗶𝗻𝗴 𝗮𝗻𝗱 𝗜𝘁𝗲𝗿𝗮𝘁𝗶𝘃𝗲 𝗗𝗲𝘃𝗲𝗹𝗼𝗽𝗺𝗲𝗻𝘁:  
- Develop and test your pipelines incrementally on small samples of data before running on the full dataset.  
- Use Spark's interactive shell or notebooks for rapid prototyping.  
  
11. 𝗘𝗿𝗿𝗼𝗿 𝗛𝗮𝗻𝗱𝗹𝗶𝗻𝗴 𝗮𝗻𝗱 𝗙𝗮𝘂𝗹𝘁 𝗧𝗼𝗹𝗲𝗿𝗮𝗻𝗰𝗲:  
- Implement appropriate error handling and fault tolerance mechanisms in your pipelines.  
- Use checkpoints to recover from failures more efficiently.  
12. 𝗩𝗲𝗿𝘀𝗶𝗼𝗻 𝗖𝗼𝗻𝘁𝗿𝗼𝗹 𝗮𝗻𝗱 𝗗𝗼𝗰𝘂𝗺𝗲𝗻𝘁𝗮𝘁𝗶𝗼𝗻:  
- Keep your codebase under version control to track changes.  
- Document your pipeline steps and logic  
13. 𝗖𝗼𝗱𝗲 𝗢𝗽𝘁𝗶𝗺𝗶𝘇𝗮𝘁𝗶𝗼𝗻:  
- Write efficient code by avoiding unnecessary transformations and actions.  
- Leverage the Catalyst optimizer by using DataFrame operations.  
14. 𝗥𝗲𝘀𝗼𝘂𝗿𝗰𝗲 𝗖𝗹𝗲𝗮𝗻𝘂𝗽:  
15. 𝗖𝗼𝗻𝘁𝗶𝗻𝘂𝗼𝘂𝘀 𝗜𝗺𝗽𝗿𝗼𝘃𝗲𝗺𝗲𝗻𝘁

Interview Question

**1. What is Data Engineering?**

Data Engineering is a term one uses when working with data. The main process of converting the raw entity of data into useful information that can be used for various purposes is called Data Engineering. This involves the [**Data Engineer**](https://intellipaat.com/community/42703/how-to-become-a-big-data-engineer) working with the data by performing data collection and research on the same.

**2. Define Data Modeling.**

Data modeling is the simplification of complex software designs by breaking them up into simple diagrams that are easy to understand, and it does not require any prerequisites for the same. This provides numerous advantages as there is a simple visual representation between the data objects involved and the rules associated with them.

**3. What are some of the design schemas used when performing Data Modeling?**

There are two schemas when one works with data modeling. They are:

* Star schema
* Snowflake schema

**4. What are the differences between structured and unstructured data?**

|  |  |  |
| --- | --- | --- |
| **Parameters** | **Structured Data** | **Unstructured Data** |
| Storage Method | DBMS | Most of it is unmanaged |
| Protocol Standards | ODBC, SQL, and ADO.NET | XML, CSV, SMSM, and SMTP |
| Scaling | Schema scaling is difficult | Schema scaling is very easy |
| Example | An ordered text dataset file | Images, videos, etc. |

**5. What is Hadoop? Explain briefly.**

[**Hadoop**](https://intellipaat.com/blog/tutorial/hadoop-tutorial/) is an open-source framework, which is used for data manipulation and data storage, as well as for running applications on units called clusters. Hadoop has been the gold standard of the day when it comes to working with and handling Big Data.

The main advantage is the easy provision of the huge amounts of space needed for data storage and a vast amount of processing power to handle limitless jobs and tasks concurrently.

**6. What are some of the important components of Hadoop?**

There are many components involved when working with Hadoop, and some of them are as follows:

* **Hadoop Common**: This consists of all libraries and utilities that are commonly used by the Hadoop application.
* **HDFS**: The Hadoop File System is where all the data is stored when working with Hadoop. It provides a distributed file system with very high bandwidth.
* **Hadoop YARN**: Yet Another Resource Negotiator is used for managing resources in the Hadoop system. Task scheduling can also be performed using YARN.
* **Hadoop MapReduce**: It is based on techniques that provide user access to large-scale data processing.

**7. What is a NameNode in HDFS?**

NameNode is one of the vital parts of HDFS. It is used as a way to store all the HDFS data and, at the same time, keep track of the files in all clusters as well.

However, you must know that the data is actually stored in the DataNodes and not in the NameNodes.

***Check all the requirements a company is looking for in Data Engineer in the***[***Data Engineer Job Description***](https://intellipaat.com/blog/data-engineer-job-description/)***blog.***

**8. What is Hadoop Streaming?**

[**Hadoop streaming**](https://intellipaat.com/blog/tutorial/hadoop-tutorial/hadoop-streaming/) is one of the widely used utilities provided by Hadoop for users to easily create maps and perform reduction operations. Later, this can be submitted into a specific cluster for usage.

**9. What are some of the important features of Hadoop?**

* Hadoop is an open-source framework.
* Hadoop works on the basis of distributed computing.
* It provides faster data processing due to parallel computing.
* Data is stored in separate clusters away from the operations.
* Data redundancy is given priority to ensure no data loss.

**Get 100% Hike!**

Master Most in Demand Skills Now !

Top of Form





 By providing your contact details, you agree to our [Terms of Use](https://intellipaat.com/terms-of-use/) & [Privacy Policy](https://intellipaat.com/terms-of-use-and-privacy-policy/)

Bottom of Form

**10. What are the four Vs of Big Data?**

The following forms to be the vital foundation of [**Big Data**](https://intellipaat.com/blog/big-data-tutorial-for-beginners/):

* Volume
* Variety
* Velocity
* Veracity

**Ready to ace your Desktop Support Engineer interview? Check out our blog for a comprehensive list of**[**Desktop Support Engineer Interview Questions**](https://intellipaat.com/blog/interview-question/desktop-support-engineer-interview-questions/)**and expert tips!**

**11. What is Block and Block Scanner in HDFS?**

Block is considered as a singular entity of data, which is the smallest factor. When Hadoop encounters a large file, it automatically slices the file into smaller chunks called blocks.

A block scanner is put into place to verify whether the loss-of-blocks created by Hadoop is put on the DataNode successfully or not.

**12. How does a Block Scanner handle corrupted files?**

* When the block scanner comes across a file that is corrupted, the DataNode reports this particular file to the NameNode.
* The NameNode then processes the file by creating replicas of the same using the original (corrupted) file.
* If there is a match between the replicas created and the replication block, then the corrupted data block is not removed.

**13. How does the NameNode communicate with the DataNode?**

The NameNode and the DataNode communicate via messages. There are two messages that are sent across the channel:

* Block reports
* Heartbeats

**14. What is meant by COSHH?**

COSHH is the abbreviation for Classification and Optimization-based Scheduling for Heterogeneous Hadoop systems. As the name suggests, it provides scheduling at both the cluster and the application levels to directly have a positive impact on the completion time for jobs.

***Learn about the difference between Data Engineer and Data Scientist in our blog on***[***Data Engineer vs Data Scientist***](https://intellipaat.com/blog/data-engineer-vs-data-scientist-difference/)***!***

**15. What is Star Schema, in brief?**

Star schema is also called the star join schema, which is one of the simple schemas in the concept of Data Warehousing. Its structure resembles a star that consists of fact tables and associated dimension tables. The star schema is widely used when working with large amounts of data.

**16. Explain Snowflake in brief.**

The snowflake schema is a primary extension of the star schema with the presence of more dimensions. It is spanned across as the structure of a snowflake, hence, the name. Data is structured here and split into more tables after normalization.

**17. State the differences between Star Schema and Snowflake Schema.**

|  |  |
| --- | --- |
| **Star Schema** | **Snowflake Schema** |
| The dimension hierarchy is stored in dimension tables. | Each hierarchy gets stored in individual tables. |
| High data redundancy | Low data redundancy |
| Simple database designs | Complex data-handling storage space |
| Fast cube processing | Slower cube processing (complex joins) |

**18. Name the XML configuration files present in Hadoop.**

Following are the XML configuration files available in **[Hadoop](https://intellipaat.com/blog/tutorial/hadoop-tutorial/)**:

* Core-site
* Mapred-site
* HDFS-site
* YARN-site

**19. What is the meaning of FSCK?**

FSCK is also known as the File System Check, which is one of the important commands used in HDFS. It is primarily put to use when you have to check for problems and discrepancies in files.

**20. What is ETL and why is it important in data engineering?**

ETL stands for Extract, Transform, Load. It is a process used in data engineering to extract data from source systems, transform it into a suitable format, and load it into a target system, typically a data warehouse or a data lake. ETL is crucial in data engineering because it allows organizations to collect, clean, and transform data from various sources into a structured and usable format for analysis. Without ETL, data would remain in its raw, often unstructured state, making it difficult to analyze and gain insights from.

**21. Explain the difference between a data warehouse and a data lake.**

A data warehouse is a structured and highly organized repository of data that is designed for querying and reporting. It typically stores structured data and enforces schema consistency. In contrast, a data lake is a more flexible storage system that can handle structured, semi-structured, and unstructured data. It allows for data to be ingested without a predefined schema and is suited for big data and data exploration. Data warehouses are optimized for analytics, while data lakes are more suitable for data storage and exploration.

**22. What is a primary key, foreign key, and how are they used in database design?**

A primary key is a unique identifier for each row in a database table. It ensures that each row is unique and provides a way to access and reference individual records. A foreign key, on the other hand, is a field in a database table that is used to establish a link between two tables. It creates a relationship between the tables, enabling referential integrity. Foreign keys help maintain data consistency and enforce relationships between related data.

**23. What is the CAP theorem, and how does it relate to distributed systems in data engineering?**

The CAP theorem, also known as Brewer’s theorem, states that in a distributed system, it’s impossible to achieve all three of the following simultaneously: Consistency, Availability, and Partition tolerance. You can have two of these qualities at the expense of the third. This theorem is critical in distributed systems because it helps in making design trade-offs. For example, in the face of network partitions (P), you might have to choose between ensuring strong data consistency (C) or high availability (A).

**24. What is the purpose of partitioning in distributed data processing frameworks like Hadoop or Spark?**

Partitioning divides a large dataset into smaller, manageable subsets called partitions. It helps in parallelizing data processing tasks across multiple nodes in a cluster. By breaking data into partitions, distributed systems like Hadoop and Spark can process data more efficiently, as each node can work on its partition concurrently. Partitioning also reduces data movement and improves data locality, which is crucial for optimizing performance in distributed systems.

**25. Explain the concept of data serialization and why it is important in data engineering.**

Data serialization is the process of converting complex data structures or objects into a format that can be easily stored, transmitted, or reconstructed. It’s essential in data engineering because it allows data to be stored in a compact format that can be easily read and processed. Common serialization formats include JSON, Avro, and Parquet. Serialization is important for data interchange between different systems, such as between a producer and a consumer in a data pipeline, as it ensures data consistency and compatibility.

**26. How do you ensure data quality in a data pipeline, and what are some common data quality issues to watch out for?**

Data quality in a data pipeline can be ensured through various techniques, such as data validation, data cleansing, and monitoring. Common data quality issues include missing values, duplicate records, inconsistent formatting, and inaccurate data. Data validation rules and data profiling can help identify and address these issues, and data quality monitoring can provide ongoing assurance that data remains accurate and reliable throughout the pipeline.

**27. What is data skew in the context of distributed data processing, and how can it be mitigated?**

Data skew refers to an imbalance in the distribution of data across partitions or nodes in a distributed system. It can result in some nodes taking significantly longer to process their data, leading to performance issues. Data skew can be mitigated by employing techniques like data shuffling, using custom partitioning strategies, or applying dynamic load balancing. Additionally, using appropriate data structures and algorithms can help spread the workload more evenly.

**28. Describe the differences between batch processing and stream processing, and provide use cases for each.**

Batch processing involves processing data in large, discrete chunks, whereas stream processing deals with data in real-time, one record at a time. Batch processing is suitable for use cases where you can afford a delay in processing, like generating daily reports or historical data analysis. Stream processing is used for real-time analytics, fraud detection, monitoring, and any application that requires immediate insights from data as it arrives.

**29. Can you explain the concept of data lineage and why it is crucial in data engineering and compliance?**

Data lineage is the tracking of data as it moves through various stages of a data pipeline or system. It’s crucial in data engineering because it helps in understanding where data originates, how it’s transformed, and where it’s consumed. Data lineage is essential for compliance, as it provides a clear audit trail for data, ensuring data governance and regulatory requirements are met. It also aids in debugging, troubleshooting, and optimizing data pipelines.

**Career Transition**







**Intermediate Interview Questions**

**30. What are some of the methods of Reducer?**

Following are the three main methods involved with reducer:

* **setup()**: This is primarily used to configure input data parameters and cache protocols.
* **cleanup()**: This method is used to remove the temporary files stored.
* **reduce()**: The method is called one time for every key, and it happens to be the single most important aspect of the reducer on the whole.

**31. What are the different usage modes of Hadoop?**

Hadoop can be used in three different modes. They are:

* Standalone mode
* Pseudo distributed mode
* Fully distributed mode

**32. How is data security ensured in Hadoop?**

Following are some of the steps involved in securing data in Hadoop:

* You need to begin by securing the authentic channel that connects clients to the server.
* Second, the clients make use of the stamp that is received to request a service ticket.
* Lastly, the clients use the service ticket as a tool for authentically connecting to the corresponding server.

***Get a deep understanding of Top***[***Data Engineering Projects***](https://intellipaat.com/blog/data-engineering-projects/)***for Beginners through this blog!***

**33. Which are the default port numbers for Port Tracker, Task Tracker, and NameNode in Hadoop?**

* Job Tracker has the default port: 50030
* Task Tracker has the default port: 50060
* NameNode has the default port: 50070

**34. How does Big Data Analytics help increase the revenue of a company?**

Data Analytics helps the companies of today’s world in numerous ways. Following are the foundational concepts in which it helps:

* Effective use of data to relate to structured growth
* Effective customer value increase and retention analysis
* Manpower forecasting and improved staffing methods
* Bringing down the production cost majorly

***Learn more about the salary structure of this professional in our blog on the***[***Big Data Engineer Salary in India***](https://intellipaat.com/blog/big-data-engineer-salary-in-india/)***!***

**35. In your opinion, what does a Data Engineer majorly do?**

A Data Engineer is responsible for a wide array of things. Following are some of the important ones:

* Handling data inflow and processing pipelines
* Maintaining data staging areas
* Responsible for ETL data transformation activities
* Performing data cleaning and the removal of redundancies
* Creating ad-hoc query building operations and native data extraction methods

If you are considering becoming proficient in Data Analytics and earn a certification while doing the same, make sure to check out Intellipaat’s [**online Data** **Analyst Training**](https://intellipaat.com/data-analytics-master-training-course/) Program.

**36. What are some of the technologies and skills that a Data Engineer should possess?**

Following are the important technologies that a Data Engineer must be proficient in:

* Mathematics (probability and linear algebra)
* Summary statistics
* Machine Learning
* R and SAS programming languages
* Python
* SQL and HiveQL

Followed by this, a Data Engineer must also have good problem-solving skills and analytical thinking ability.

**37. What is the difference between a Data Architect and a Data Engineer?**

A Data Architect is a person who is responsible for managing the data that comes into the organization from a variety of sources. Data handling skills such as database technologies are a must-have skill of a Data Architect. The Data Architect is also concerned with how changes in the data will lead to major conflicts in the organization model.

Now, a Data Engineer is the person who is primarily responsible for helping the Data Architect with setting up and establishing the Data Warehousing pipeline and the architecture of enterprise data hubs.

**Courses you may like**

[](https://intellipaat.com/advanced-certification-data-science-artificial-intelligence-iit-madras/)

[](https://intellipaat.com/data-science-ai-mnit/)

[](https://intellipaat.com/data-science-architect-masters-program-training/)

**38. How is the distance between nodes defined when using Hadoop?**

The distance between nodes is the simple sum of the distances to the closest corresponding nodes. The getDistance() method is used to calculate these distances.

**39. What is the data stored in the NameNode?**

NameNode primarily consists of all of the metadata information for HDFS such as the namespace details and the individual block information.

Here is one of the very important Facebook Data Engineer interview questions that is quite commonly asked.

**40. What is meant by Rack Awareness?**

Rack awareness is a concept in which the NameNode makes use of the DataNode to increase the incoming network traffic while concurrently performing reading or writing operations on the file, which is the closest to the rack from which the request was called for.

***Increase your interest in Data engineering after knowing the salary in India through***[***Data Engineer Salary in India***](https://intellipaat.com/blog/data-engineer-salary-in-india/)***blog.***

**41. What is a Heartbeat message?**

Heartbeat is one of the two ways the DataNode communicates with the NameNode. It is an important signal which is sent by the DataNode to the NameNode in a structured interval to show that it is still operational.

**42. What is the use of a Context Object in Hadoop?**

A context object is used in Hadoop, along with the mapper class, as a means of communication with the other parts of the system. System configuration details and jobs present in the constructor are obtained easily using the context object.

It is also used to send information to methods such as setup(), cleanup(), and map().

**43. What is the use of Hive in the Hadoop ecosystem?**

Hive is used to provide the user interface to manage all the stored data in Hadoop. The data is mapped with HBase tables and worked upon, as and when needed. Hive queries (similar to SQL queries) are executed to be converted into MapReduce jobs. This is done to keep the complexity under check when executing multiple jobs at once.

**44. What is the use of Metastore in Hive?**

Metastore is used as a storage location for the schema and Hive tables. Data such as definitions, mappings, and other metadata can be stored in the metastore. This is later stored in an RDMS when required.

Next up on this compilation of top Data Engineer interview questions, let us check out the advanced set of questions.

**45. Explain the concept of Data Sharding and how it affects database scalability.**

Data Sharding involves splitting a large database into smaller, more manageable pieces or ‘shards’, which are distributed across multiple servers. This enhances scalability as it allows the database to handle more requests by spreading the load.

**46. How would you design a system to deduplicate streaming data in real-time?**

Designing a system to deduplicate streaming data involves using techniques like Bloom Filters or Cuckoo Filters to check for duplicates efficiently, along with windowing and time-based checks to ensure data consistency.

**47. Describe the use of Directed Acyclic Graphs (DAGs) in data processing frameworks like Apache Spark.**

In frameworks like Apache Spark, DAGs represent a sequence of computations performed on data. Each node represents an operation, and the edges represent the data flow. DAGs allow for fault tolerance and optimization as they clearly define stages of computation.

**48. How can eventual consistency be handled in a distributed database system?**

Eventual Consistency can be handled by implementing mechanisms like Conflict Resolution Strategies (e.g., Last Write Wins), Version Vectors, or Quorum-based Replication to ensure that, over time, all replicas converge to the same state.

**49. Explain how a Bloom Filter works and where it might be used in a data engineering pipeline.**

A Bloom Filter is a probabilistic data structure used to test whether an element is a member of a set. It can introduce false positives but not false negatives. It is used to reduce unnecessary disk I/O or network calls, like checking if a key exists in a database.

**50. How would you implement data retention policies in a data warehouse?**

Implementing data retention involves setting up Time-To-Live (TTL) policies, archiving strategies, and partitioning data based on time, allowing for efficient deletion or archiving of older data.

**51. Discuss the CAP theorem and its implications for distributed systems.**

The CAP theorem posits that a distributed system can only achieve two out of three properties: Consistency, Availability, and Partition tolerance. The theorem guides the design and trade-offs in distributed systems.

**52. How can skewness be handled during a join operation in a distributed data processing environment?**

Skew can be mitigated by techniques such as salting keys (adding random prefixes/suffixes), broadcasting smaller tables, or repartitioning the data to ensure even distribution among processing nodes.

**53. Explain how a Time-series Database is different from a traditional Relational Database and provide examples.**

Time-series Databases (e.g., InfluxDB, TimescaleDB) are optimized for handling time-stamped data and are efficient for write-heavy workloads. Relational Databases (e.g., MySQL, PostgreSQL) are general-purpose and may not perform as efficiently with time-series data.

**54. How would you ensure data quality and integrity while ingesting data from multiple heterogeneous sources?**

Ensuring data quality involves implementing data validation checks, schema validation, de-duplication strategies, and data profiling. Anomalies and inconsistencies can be logged and corrected using predefined rules or manual intervention.

**Advanced Interview Questions**

**55. What are the components that are available in the Hive data model?**

Following are some of the components in Hive:

* Buckets
* Tables
* Partitions

**56. Can you create more than a single table for an individual data file?**

Yes, it is possible to create more than one table for a data file. In Hive, schemas are stored in the metastore. Therefore, it is very easy to obtain the result for the corresponding data.

**57. What is the meaning of Skewed tables in Hive?**

Skewed tables are the tables in which values appear in a repeated manner. The more they repeat, the more the skewness.

Using Hive, a table can be classified as SKEWED while creating it. By doing this, the values will be written to different files first, and later, the remaining values will go to a separate file.

***Also, check out the blog on***[***Data Engineer Salary***](https://intellipaat.com/blog/data-engineer-salary/)***!***

**58. What are the collections that are present in Hive?**

Hive has the following collections/data types:

* Array
* Map
* Struct
* Union

[](https://intellipaat.com/post-graduate-certification-big-data-analytics/)

**59. What is SerDe in Hive?**

SerDe stands for Serialization and Deserialization in Hive. It is the operation that is involved when passing records through Hive tables.

The Deserializer takes a record and converts it into a Java object, that is understood by Hive.

Now, the Serializer takes this Java object and converts it into a format that is processable by HDFS. Later, HDFS takes over for the storage function.

**60. What are the table creation functions present in Hive?**

Following are some of the table creation functions in Hive:

* Explode(array)
* Explode(map)
* JSON\_tuple()
* Stack()

**61. What is the role of the .hiverc file in Hive?**

The role of the .hiverc file is initialization. Whenever you want to write code for Hive, you open up the CLI (command-line interface), and whenever the CLI is opened, this file is the first one to load. It contains the parameters that you initially set.

**62. What are \*args and \*\*kwargs used for?**

The \*args function lets users define an ordered function for usage in the command line, and the \*\*kwargs function is used to denote a set of arguments that are unordered and in line to be input to a function.

**63. How can you see the structure of a database using MySQL?**

To see the structure of a database, the describe command can be used. The syntax is simple:

describe tablename;

**64. Can you search for a specific string in a column present in a MySQL table?**

Yes, specific strings and corresponding substring operations can be performed in MySQL. The regex operator is used for this purpose.

**65. In brief, what is the difference between a Data Warehouse and a Database?**

When working with Data Warehousing, the primary focus goes on using aggregation functions, performing calculations, and selecting subsets in data for processing. With databases, the main use is related to data manipulation, deletion operations, and more. Speed and efficiency play a big role when working with either of these.

**66. Have you earned any sort of certification to boost your opportunities as a Data Engineer?**

Interviewers look for candidates who are serious about advancing their career options by making use of additional tools like certifications. Certificates are strong proof that you have put in all efforts to learn new skills, master them, and put them into use at the best of your capacity. List the certifications, if you have any, and do talk about them in brief, explaining what all you learned from the program and how it’s been helpful to you so far.

**67. Do you have any experience working in the same industry as ours before?**

This question is a frequent one. It is asked to understand if you have had any previous exposure to the environment and work in the same. Make sure to elaborate the experience you have, with the tools you’ve used and the techniques you’ve implemented. This ensures to provide a complete picture to the interviewer.

[](https://intellipaat.com/data-science-architect-masters-program-training/)

**68. Why are you applying for the Data Engineer role in our company?**

Here, the interviewer is trying to see how well you can convince them regarding your proficiency in the subject, handling all the concepts needed to bring in large amounts of data, work with it, and help build a pipeline. It is always an added advantage to know the job description in detail, along with the compensation and the details of the company, thereby, obtaining a complete understanding of what tools, software packages, and technologies are required to work in the role.

**69. What is your plan after joining this Data Engineer role?**

While answering this question, make sure to keep your explanation concise on how you would bring about a plan that works with the company set up and how you would implement the plan, ensuring that it works by first understanding the data infrastructure setup of the company, and you would also talk about how it can be made better or further improvised in the coming days with further iterations.

**70. Do you have prior experience working with Data Modeling?**

If you are interviewed for an intermediate-level role, this is a question that you will always be asked. Begin your answer with a simple yes or no. It is alright if you have not worked with data modeling before, but make sure to explain whatever you know about data modeling in a concise and structured manner. It would be advantageous if you have used tools like Pentaho or Informatica for this purpose.

If you are looking forward to learning and mastering all of the Data Analytics and Data Science concepts and earning a certification in the same, do take a look at Intellipaat’s latest [**Data Science with R Certification**](https://intellipaat.com/r-programming-certification-training/) offerings.

**71. Discuss the implications of the General Data Protection Regulation (GDPR) on data engineering pipelines and how to ensure compliance.**

GDPR affects data pipelines by imposing strict rules on data collection, processing, and storage, particularly personal data of EU citizens. To comply, engineers must ensure data anonymization through techniques like pseudonymization and encryption, establish clear consent mechanisms, and provide easy data access and deletion functionalities. Additionally, maintaining thorough documentation, performing Data Protection Impact Assessments (DPIAs), and appointing Data Protection Officers (DPOs) are essential steps.

**72. How would you design a globally distributed and highly available data pipeline ensuring data consistency?**

To ensure data consistency in a distributed system, employ data replication across regions. Use consistent hashing to distribute data evenly across servers, and choose a suitable consistency model (e.g., strong consistency with a quorum-based algorithm like Paxos or Raft, or eventual consistency for higher availability). Conflict-free replicated data types (CRDTs) or multi-version concurrency control (MVCC) can help manage data version conflicts.

**73. Explain the considerations and strategies for optimizing query performance in a columnar data store.**

To boost query performance in a columnar store, leverage the intrinsic benefits of columnar storage by minimizing I/O operations through column pruning and partition pruning. Employ efficient compression algorithms to reduce storage and speed up query processing. Use data indexing for faster lookups and consider the cost-based optimizer to dynamically choose the best query execution plan based on data statistics.

**74. Discuss the challenges and solutions for real-time anomaly detection in high-velocity data streams.**

For anomaly detection in high-velocity streams, one must handle the volume and velocity of data while maintaining accuracy. Employing scalable machine learning models like Isolation Forests can help. Windowing techniques in stream processing platforms allow for handling out-of-order events and late arrivals. Tools like Apache Flink provide advanced state management and event-time processing capabilities for complex event processing.

**75. How would you approach designing a Data as a Service (DaaS) platform?**

When designing a DaaS platform, consider the full lifecycle management of data services. Implement RESTful APIs for data access, apply robust security measures including authentication, authorization, and encryption, and incorporate data governance and quality assurance measures. Support various data delivery models, including real-time streams and batch downloads, while ensuring the platform’s ability to scale out and manage varying load patterns.

**76. Explain the complications of cross-cloud data migration and strategies to minimize downtime and data loss.**

In cross-cloud migrations, transfer costs, and potential data inconsistencies due to network issues are primary concerns. To mitigate downtime, use database replication techniques, ensure data integrity checks, and apply change data capture for continuous synchronization. For minimizing data transfer costs, consider data compression, transfer scheduling, and possibly utilizing dedicated data transfer networks or appliances offered by cloud providers.

**77. Discuss how Quantum Computing might affect data encryption and how to prepare for these changes.**

Quantum computing threatens current cryptographic algorithms. To future-proof encryption, invest in researching post-quantum cryptography, focusing on algorithms that are considered resistant to quantum attacks, such as lattice-based, hash-based, code-based, or multivariate quadratic equations cryptography. Keep abreast of NIST’s post-quantum cryptographic standardization process.

**78. How would you implement a scalable and efficient data versioning system for a large dataset?**

Implementing an efficient data versioning system for large datasets can be achieved by leveraging structures like LSM-trees which are write-optimized and handle large-scale versioning well. Delta Lake, on top of a data lake, provides ACID transactions, scalable metadata handling, and unifies streaming and batch data processing.

**79. Discuss the concept of Federated Learning and how it can be used to build privacy-preserving machine learning models.**

Federated Learning’s key benefit is the ability to train models on decentralized data, ensuring privacy by design. It requires managing model updates across distributed nodes, aggregating them centrally without transferring the underlying data. This is critical for sensitive information and complies with privacy regulations like GDPR.

**80. Explain how to design a system to guarantee data integrity and accuracy in a Microservices architecture.**

Maintaining data integrity in a microservices environment requires decentralized data management. Implement distributed transactions using the Saga pattern, where each service performs its transaction and publishes events, while other services react to these events and execute local transactions. Event sourcing ensures all changes to application state are stored as a sequence of events, which can be replayed to restore the state of a system.

**81. Describe how you would set up a data pipeline to handle both batch and stream processing workloads. What technologies would you use, and how would you ensure minimal latency for the streaming data while managing the efficiency of the batch processing tasks?**

To handle both batch and stream processing workloads, I would design the data pipeline using a unified processing engine like Apache Flink or Apache Spark, which supports both processing methods. For stream processing, I would ensure minimal latency by leveraging in-memory processing and carefully tuning checkpoint intervals and state backends. For batch jobs, I’d focus on optimizing resource allocation and job scheduling to run during low-traffic periods to maintain efficiency. Kafka could be used as the messaging system to buffer the stream of data, ensuring durability and fault tolerance.

**82. How would you design a schema evolution strategy for a data lake that receives heterogeneous data sources and formats? What would be your approach to handling breaking schema changes in a production environment without causing downtime or data loss?**

Schema evolution in a data lake is challenging due to diverse data sources and formats. My strategy would include implementing a schema registry that supports schema versioning and validation like Confluent Schema Registry. For managing schema changes, I would use a format that supports schema evolution such as Avro or Parquet. Additionally, I’d ensure that data ingestion pipelines are robust enough to handle schema changes by using dynamic schema discovery and validation. Breaking changes would be managed by versioning datasets and using backward and forward compatibility checks to prevent data loss or downtime.

**83. Discuss the trade-offs between different data serialization formats such as Avro, Parquet, and JSON in the context of real-time analytics. How would you choose the appropriate format for a given use case, considering factors such as schema evolution, compression, and processing speed?**

The trade-offs between data serialization formats like Avro, Parquet, and JSON significantly depend on the specific use case. Avro is great for schema evolution and is compact, making it suitable for Kafka messages. Parquet is a columnar format that provides efficient compression and speedy query performance, ideal for OLAP workloads. JSON, while human-readable and flexible, is less efficient in both space and processing. For real-time analytics, where processing speed is important, I would lean towards Avro for its balance of performance and schema evolution. If analytical queries are also a requirement, using Avro for streaming into a system and then transforming it to Parquet for long-term storage could be optimal.

### ****1. What is Apache Spark?****

Spark is a fast, easy-to-use, and flexible data processing framework. It is an open-source analytics engine that was developed by using [Scala](https://intellipaat.com/blog/what-is-scala/" \t "_blank), [Python](https://intellipaat.com/blog/tutorial/python-tutorial/what-is-python/), [Java](https://intellipaat.com/blog/tutorial/java-tutorial/), and [R](https://intellipaat.com/blog/tutorial/r-programming/introduction/). It has an advanced execution engine supporting acyclic data flow and in-memory computing. It uses in-memory caching and optimized execution of queries for faster query analytics of data of any size. [Apache Spark](https://intellipaat.com/blog/what-is-apache-spark/) can run standalone, on Hadoop, or in the cloud and is capable of accessing diverse data sources including HDFS, HBase, and Cassandra, among others.

### ****2. Explain the key features of Spark.****

* Apache Spark allows integrating with [Hadoop](https://intellipaat.com/blog/tutorial/hadoop-tutorial/" \t "_blank).
* It has an interactive language shell, Scala (the language in which Spark is written).
* Spark consists of RDDs (Resilient Distributed Datasets), which can be cached across the computing nodes in a cluster.
* Apache Spark supports multiple analytic tools that are used for interactive query analysis, real-time analysis, and graph processing
* Apache Spark supports stream processing in real-time.
* Spark helps in achieving a very high processing speed of data, which it achieves by reducing the read or write operations to disk.
* Apache Spark codes can be reused for data streaming, running ad-hoc queries, batch processing, etc.
* Spark is considered a better cost-efficient solution when compared to Hadoop.

***Learn more key features of Apache Spark in this***[***Apache Spark Tutorial***](https://intellipaat.com/blog/tutorial/spark-tutorial/)***!***

### ****3. What is MapReduce?****

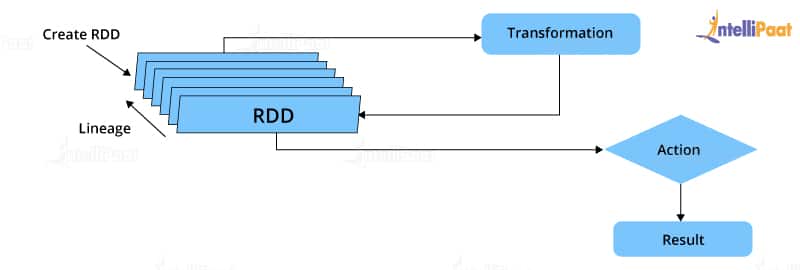
It is a software framework and programming model which is used for processing huge datasets. [MapReduce](https://intellipaat.com/blog/tutorial/mapreduce-tutorial/" \t "_blank) is basically split into two parts, Map and Reduce. Map handles data splitting and data mapping, meanwhile, Reduce handles shuffle and reduction in data.

### ****4. Compare MapReduce with Spark.****

|  |  |  |
| --- | --- | --- |
| **Criteria** | **MapReduce** | **Spark** |
| Processing speed | Good | Excellent (up to 100 times faster) |
| Data caching | Hard disk | In-memory |
| Performing iterative jobs | Average | Excellent |
| Dependency on Hadoop | Yes | No |
| Machine Learning applications | Average | Excellent |

### ****5. Define RDD.****

RDD is the acronym for [Resilient Distribution Datasets](https://intellipaat.com/blog/tutorial/spark-tutorial/programming-with-rdds/)—a fault-tolerant collection of operational elements that run in parallel. The partitioned data in an RDD is immutable and distributed. There are primarily two types of RDDs:

RDD in Spark

* Parallelized collections: The existing RDDs running in parallel with one another
* Hadoop datasets: Those performing a function on each file record in HDFS or any other storage system

### ****6. What does a Spark Engine do?****

A Spark engine is responsible for scheduling, distributing, and monitoring the data application across the cluster. Spark Engine is used to run mappings in Hadoop clusters. It is suitable for wide-ranging circumstances. It includes SQL batch and ETL jobs in Spark, streaming data from sensors, IoT, ML, etc.

***Read on Spark Engine and more in this***[***Apache Spark Community***](https://intellipaat.com/community/big-data-hadoop-spark)***!***

**Get 100% Hike!**

Master Most in Demand Skills Now !

Top of Form



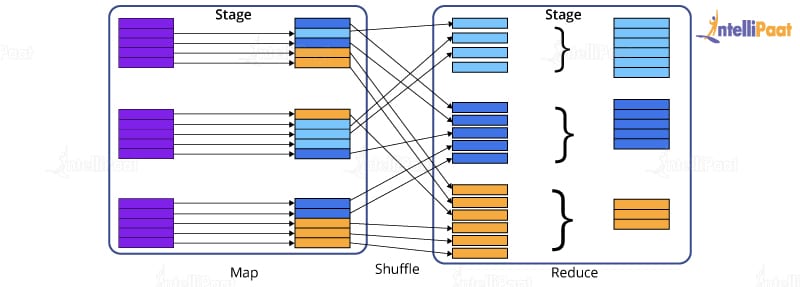


 By providing your contact details, you agree to our [Terms of Use](https://intellipaat.com/terms-of-use/) & [Privacy Policy](https://intellipaat.com/terms-of-use-and-privacy-policy/)

Bottom of Form

### ****7. Define Partitions.****

As the name suggests, a partition is a smaller and logical division of data similar to a ‘split’ in MapReduce. Partitioning is the process of deriving logical units of data to speed up data processing. Everything in Spark is a partitioned RDD.



### ****8. What operations does an RDD support?****

* **Transformations:**Transformations produce a new RDD from an existing RDD, every time we apply a transformation to the RDD. Always it takes an RDD as input and ejects one or more RDD as output.
* **Actions:**Actions are used when we wish to use the actual RDD instead of working with a new RDD after we apply transformations. Actions eject out non-RDD values unlike transformations, which only eject RDD values.

### ****9. What do you understand about Transformations in Spark?****

Transformations are functions applied to RDDs, resulting in another RDD. It does not execute until an action occurs. Functions such as map() and filer() are examples of transformations, where the map() function iterates over every line in the RDD and splits into a new RDD. The filter() function creates a new RDD by selecting elements from the current RDD that passes the function argument.

### ****10. Define Actions in Spark.****

Actions are operations in Spark; they help in working with the actual data set. They help in transferring data from executor to driver. In Spark, an action helps in bringing back data from an RDD to the local machine. They are RDD operations giving non-RDD values, which is unlike transformations operations, which only eject RDD as output. The reduce() function is an action that is implemented again and again until only one value is left. The take() action takes all the values from an RDD to the local node.

### ****Check out this insightful video on Spark Tutorial for Beginners:****

### ****11. Define the functions of Spark Core.****

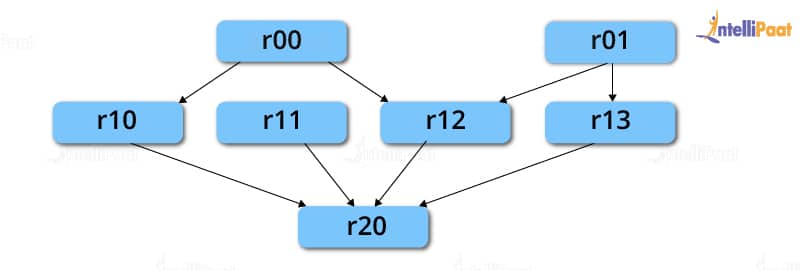
Serving as the base engine, Spark Core performs various important functions like memory management, basic I/O functionalities, monitoring jobs, providing fault-tolerance, job scheduling, interaction with storage systems, distributed task dispatching, and many more. Spark Core is the base of all projects. The above-mentioned functions are Spark Core’s primary functions.

***Learn more about Spark from this***[***Spark Training in New York***](https://intellipaat.com/apache-spark-scala-training-new-york/)***to get ahead in your career!***

## Intermediate Spark Interview Questions for experienced

### ****12. What is RDD Lineage?****

Spark does not support data replication in memory and thus, if any data is lost, it is rebuilt using RDD lineage.



RDD lineage is a process that reconstructs lost data partitions. The best thing about this is that RDDs always remember how to build from other datasets.

**Career Transition**







### ****13. What is Spark Driver?****

Spark Driver is the program that runs on the master node of a machine and declares transformations and actions on data RDDs. In simple terms, a driver in Spark creates SparkContext, connected to a given Spark Master. It also delivers RDD graphs to Master, where the standalone Cluster Manager runs.

### ****14. What is Hive on Spark?****

[Hive](https://intellipaat.com/blog/tutorial/hadoop-tutorial/apache-hive/) contains significant support for Apache Spark, wherein Hive execution is configured to Spark:

hive> set spark.home=/location/to/sparkHome;

hive> set hive.execution.engine=spark;

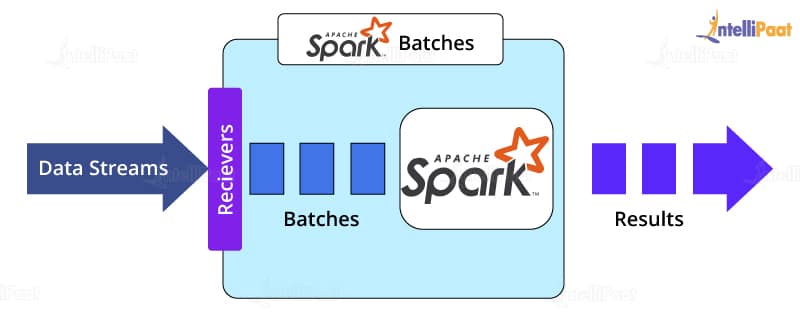
Hive supports Spark on YARN mode by default.

### ****15. Name the commonly used Spark Ecosystems.****

* [Spark SQL](https://intellipaat.com/blog/what-is-spark-sql/) (Shark) for developers
* Spark Streaming for processing live data streams
* GraphX for generating and computing graphs
* MLlib ([Machine Learning Algorithms](https://intellipaat.com/blog/tutorial/machine-learning-tutorial/machine-learning-algorithms/))
* SparkR to promote R Programming in the Spark engine

### ****16. Define Spark Streaming.****

Spark supports stream processing—an extension to the Spark API allowing stream processing of live data streams.



Data from different sources like Kafka, Flume, Kinesis is processed and then pushed to file systems, live dashboards, and databases. It is similar to batch processing in terms of the input data which is here divided into streams like batches in batch processing.

***Learn in detail about the***[***Top Four Apache Spark Use Cases***](https://intellipaat.com/blog/top-4-apache-spark-use-cases/)***including Spark Streaming!***

### ****17. What is GraphX?****

Spark uses GraphX for graph processing to build and transform interactive graphs. The GraphX component enables programmers to reason about structured data at scale.

### ****18. What does MLlib do?****

MLlib is a scalable Machine Learning library provided by Spark. It aims at making [Machine Learning](https://intellipaat.com/blog/what-is-machine-learning/) easy and scalable with common learning algorithms and use cases like clustering, regression filtering, dimensional reduction, and the like.

### ****19. What is Spark SQL?****

Spark SQL, better known as Shark, is a novel module introduced in Spark to perform structured data processing. Through this module, Spark executes relational SQL queries on data. The core of this component supports an altogether different RDD called SchemaRDD, composed of row objects and schema objects defining the data type of each column in a row. It is similar to a table in relational databases.

### ****20. What is a Parquet file?****

Parquet is a columnar format file supported by many other data processing systems. Spark SQL performs both read and write operations with the Parquet file and considers it to be one of the best [Big Data Analytics](https://intellipaat.com/blog/big-data-analytics/) formats so far.

**Courses you may like**

[](https://intellipaat.com/advanced-certification-data-science-artificial-intelligence-iit-madras/)

[](https://intellipaat.com/data-science-ai-mnit/)

[](https://intellipaat.com/data-science-architect-masters-program-training/)

### ****21. What file systems does Apache Spark support?****

Apache Spark is a powerful distributed data processing engine that processes data coming from multiple data sources. The file systems that Apache Spark supports are:

* [Hadoop Distributed File System (HDFS)](https://intellipaat.com/blog/tutorial/hadoop-tutorial/hdfs-overview/)
* Local file system
* [Amazon S3](https://intellipaat.com/blog/what-is-amazon-s3/)
* [HBase](https://intellipaat.com/blog/what-is-apache-hbase/)
* [Cassandra](https://intellipaat.com/blog/what-is-apache-cassandra/), etc.

### ****22. What is Directed Acyclic Graph in Spark?****

Directed Acyclic Graph or DAG is an arrangement of edges and vertices. As the name implies the graph is not cyclic. In this graph, the vertices represent RDDs, and the edges represent the operations applied to RDDs. This graph is unidirectional, which means it has only one flow. DAG is a scheduling layer that implements stage-oriented scheduling and converts a plan for logical execution to a physical execution plan.

### ****23.What are deploy modes in Apache Spark?****

There are only two deploy modes in Apache Spark, client mode and cluster mode. The behavior of Apache Spark jobs depends on the driver component. If the driver component of Apache Spark will run on the machine from which the job is submitted, then it is the client mode. If the driver component of Apache Spark will run on Spark clusters and not on the local machine from which the job is submitted, then it is the cluster mode.

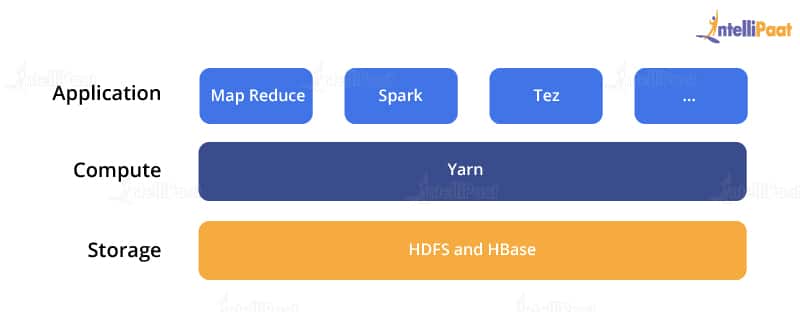
### ****24. Roles of receivers in Apache Spark Streaming?****

Within [Apache Spark Streaming](https://intellipaat.com/blog/a-guide-to-apache-spark-streaming-tutorial/) Receivers are special objects whose only goal is to consume data from different data sources and then move it to Spark. You can create receiver objects by streaming contexts as long-running tasks on various executors. There are two types of receivers. They are **Reliable receivers:**This receiver acknowledges data sources when data is received and replicated successfully in Apache Spark Storage. **Unreliable receiver:**These receivers do not acknowledge data sources even when they receive or replicate in Apache Spark Storage.

## Advanced Spark Interview Questions

### ****25. What is YARN?****

Similar to Hadoop, [YARN](https://intellipaat.com/blog/tutorial/hadoop-tutorial/what-is-yarn/) is one of the key features in Spark, providing a central and resource management platform to deliver scalable operations across the cluster. Running Spark on YARN needs a binary distribution of Spark that is built on YARN support.



**Enroll in Intellipaat’s**[**Spark Course in London**](https://intellipaat.com/apache-spark-scala-training-london/)**today to get a clear understanding of Spark!**

### ****26. List the functions of Spark SQL.****

Spark SQL is capable of:

* Loading data from a variety of structured sources
* Querying data using SQL statements, both inside a Spark program and from external tools that connect to Spark SQL through standard database connectors (JDBC/ODBC), e.g., using Business Intelligence tools like Tableau
* Providing rich integration between SQL and the regular Python/Java/Scala code, including the ability to join RDDs and SQL tables, expose custom functions in SQL, and more.

### ****27. What are the benefits of Spark over MapReduce?****

* Due to the availability of in-memory processing, Spark implements data processing 10–100x faster than Hadoop MapReduce. MapReduce, on the other hand, makes use of persistence storage for any of the data processing tasks.
* Unlike Hadoop, Spark provides in-built libraries to perform multiple tasks using batch processing, steaming, Machine Learning, and interactive SQL queries. However, Hadoop only supports batch processing.
* Hadoop is highly disk-dependent, whereas Spark promotes caching and in-memory data storage.
* Spark is capable of performing computations multiple times on the same dataset, which is called iterative computation. Whereas, there is no iterative computing implemented by Hadoop.

***For more insights, read on***[***Spark vs MapReduce***](https://intellipaat.com/blog/spark-vs-map-reduce/)***!***

### ****28. Is there any benefit of learning MapReduce?****

Yes, MapReduce is a paradigm used by many Big Data tools, including Apache Spark. It becomes extremely relevant to use MapReduce when data grows bigger and bigger. Most tools like Pig and Hive convert their queries into MapReduce phases to optimize them better.

### ****29. What is a Spark Executor?****

When SparkContext connects to Cluster Manager, it acquires an executor on the nodes in the cluster. Executors are Spark processes that run computations and store data on worker nodes. The final tasks by SparkContext are transferred to executors for their execution.

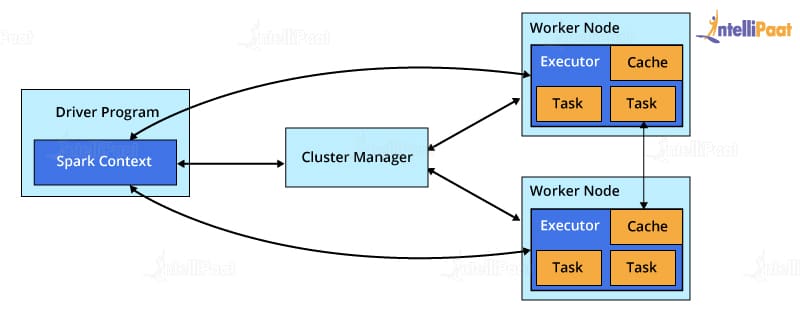
### ****30. Name the types of Cluster Managers in Spark.****

The Spark framework supports three major types of Cluster Managers.

* **Standalone:** A basic Cluster Manager to set up a cluster
* **Apache Mesos:** A generalized/commonly-used Cluster Manager, running Hadoop MapReduce and other applications
* **YARN:** A Cluster Manager responsible for resource management in Hadoop

### ****31. What do you understand by a Worker node?****

A worker node refers to any node that can run the application code in a cluster.



### ****32. What is PageRank?****

A unique feature and algorithm in GraphX, PageRank is the measure of each vertex in a graph. For instance, an edge from u to v represents an endorsement of v‘s importance w.r.t. u. In simple terms, if a user on Instagram is followed massively, he/she will be ranked high on that platform.

### ****33. Do you need to install Spark on all the nodes of the YARN cluster while running Spark on YARN?****

No, because Spark runs on top of YARN.

### ****34. Illustrate some demerits of using Spark.****

Since Spark utilizes more storage space when compared to Hadoop and MapReduce, there might arise certain problems. Developers need to be careful while running their [applications of Spark](https://intellipaat.com/blog/tutorial/spark-tutorial/apache-spark-applications/). To resolve the issue, they can think of distributing the workload over multiple clusters, instead of running everything on a single node.

### ****35. How to create an RDD?****

Spark provides two methods to create an RDD:

* By parallelizing a collection in the driver program. This makes use of SparkContext’s ‘parallelize’ method **val**

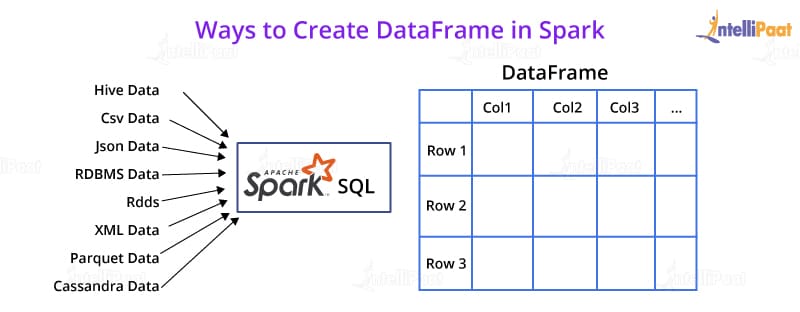
IntellipaatData = Array(2,4,6,8,10)

val distIntellipaatData = sc.parallelize(IntellipaatData)

By loading an external dataset from external storage like HDFS, the shared file system

### ****36. What are Spark DataFrames?****

When a dataset is organized into SQL-like columns, it is known as a DataFrame.



This is, in concept, equivalent to a data table in a relational database or a literal ‘DataFrame’ in R or Python. The only difference is the fact that [Spark DataFrames](https://intellipaat.com/blog/tutorial/spark-tutorial/spark-dataframe/) are optimized for Big Data.

### ****37. What are Spark Datasets?****

Datasets are data structures in Spark (added since Spark 1.6) that provide the JVM object benefits of RDDs (the ability to manipulate data with lambda functions), alongside a Spark SQL-optimized execution engine.

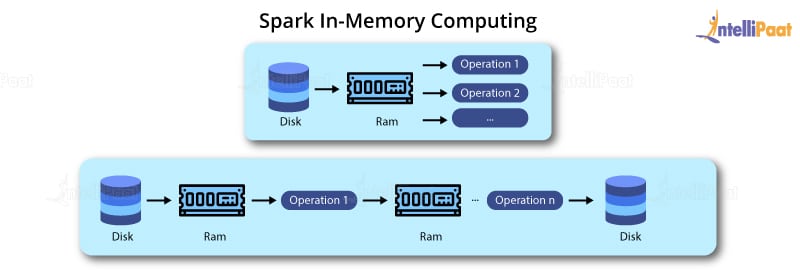
### ****38. Which languages can Spark be integrated with?****

Spark can be integrated with the following languages:

* Python, using the Spark Python API
* R, using the R on Spark API
* Java, using the Spark Java API
* Scala, using the Spark Scala API

### ****39. What do you mean by in-memory processing?****

In-memory processing refers to the instant access of data from physical memory whenever the operation is called for.



This methodology significantly reduces the delay caused by the transfer of data. Spark uses this method to access large chunks of data for querying or processing.

### ****40. What is lazy evaluation?****

Spark implements a functionality, wherein if you create an RDD out of an existing RDD or a data source, the materialization of the RDD will not occur until the RDD needs to be interacted with. This is to ensure the avoidance of unnecessary memory and CPU usage that occurs due to certain mistakes, especially in the case of Big Data Analytics.

***Interested in learning Spark? Take up our***[***Spark Training in Sydney***](https://intellipaat.com/apache-spark-scala-training-sydney/)***now!***

## General data engineer interview questions

Interviewers want to know about you and why you’re interested in becoming a data engineer. Data engineering is a technical role, so while you’re less likely to be asked behavioral questions, these higher-level questions might show up early in your interview.

### 1. Tell me about yourself.

**What they’re really asking:**What makes you a good fit for this job?

This question is asked so often in interviews that it can seem generic and open-ended, but it’s really about your relationship with data engineering. Keep your answer focused on your path to becoming a data engineer. What attracted you to this career or industry? How did you develop your technical skills?

**The interviewer might also ask:**

* Why did you choose to pursue a career in data engineering?
* Describe your path to becoming a data engineer.

### 2. What is a data engineer’s role within a team or company?

**What they’re really asking:**What is a data engineer responsible for?

For this question, recruiters want to know that you’re aware of the duties of a data engineer. What do they do? What role do they play within a team? You should be able to describe the typical responsibilities, as well as who a data engineer works with on a team. If you have experience as a data scientist or analyst, you may want to describe how you’ve worked with data engineers in the past.

**The interviewer might also ask:**

* What do data engineers do?
* How do data engineers work within a team?
* What impact does a data engineer have?

### 3. When did you face a challenge in dealing with unstructured data and how did you solve it?

**What they’re really asking:**How do you deal with problems? What are your strengths and weaknesses?

Essentially, a [data engineer’s main responsibility](https://www.coursera.org/articles/what-does-a-data-engineer-do-and-how-do-i-become-one) is to build the systems that collect, manage, and convert raw data into usable information for data scientists and business analysts to interpret. This question aims to ask about any obstacles you may have faced when dealing with a problem, and how you solved it.

This is your time to shine, where you can describe how you make data more accessible through coding and algorithms. Rather than explaining the technicalities at this point, remember the specific responsibilities listed in the job description and see if you can incorporate them into your answer.

**The interviewer might also ask:**

* How do you solve a business problem?
* What is your process for dealing with and solving problems during a project?
* Can you describe a time when you encountered a problem and solved it in an innovative manner?

## Data engineer process questions

Most often, data engineer job candidates will be asked about their projects. If you’ve never been a data engineer previously, you can describe projects that you either worked on for a class or one you posted on GitHub, a code hosting platform that promotes collaboration among developers.

### 4. Walk me through a project you worked on from start to finish.

**What they’re really asking:**How do you think through the process of acquiring, cleaning, and presenting data?

You’ll definitely be asked a question about your thought process and methodology for completing a project. Hiring managers want to know how you transformed the unstructured data into a complete product. You’ll want to practice explaining your logic for choosing certain algorithms in an easy-to-understand manner, to demonstrate you *really* know what you’re talking about. Afterward, you’ll be asked follow-up questions based on this project.

**The interviewer might also ask:**

* What was the most challenging project you’ve worked on, and how did you complete it?
* What is your process when you start a new project?

### 5. What algorithm(s) did you use on the project?

**What they’re really asking:**Why did you choose this algorithm, and can you compare it with other similar algorithms?

They want to know how you think through choosing one algorithm over another. It might be easiest to focus on a project that you worked on and link any follow-up questions to that project. If you have an example of a project and algorithm that relates to the company’s work, then choose that one to impress the interviewer. List the models you worked with, and then explain the analysis, results, and impact.

**The interviewer might also ask:**

* What is the scalability of this algorithm?
* If you were to do the project again, what would you do differently?

### 6. What tools did you use on the project?

**What they’re really asking:**How did you arrive at your decision to use certain tools?

Data engineers must manage huge swaths of data, so they need to use the right tools and technologies to gather and prepare it all. If you have experience using different tools such as Hadoop, MongoDB, and Kafka, you’ll want to explain which one you used for that particular project.

You can go into detail about the ETL (extract, transform, and load) systems you used to move data from databases into a data warehouse, such as Stitch, Alooma, Xplenty, and Talend. Some tools work better for back-end, so if you can communicate strong decision-making abilities, then you’ll shine as a candidate who’s confident in their skills.

**The interviewer might also ask:**

* What are your favorite tools to use, and why?
* Compare and contrast two or three tools that you used on a recent project.

## Data engineer technical questions

Some interviewers might follow up with more technical questions, for which you may want to refresh your memory prior to the interview. Familiarize yourself with the concepts listed in the job description and practice talking through them.

### 7. What is data modeling?

Data modeling is the initial step toward designing the database and analyzing data. You’ll want to explain that you’re capable of showing the relationship between structures, first with the conceptual model, then the logical model, and followed by the physical model.

### 8. Explain the difference between structured data and unstructured data.

Data engineers must turn unstructured data into structured data for data analysis using different methods for transformation. First, you can explain the difference between the two.

Structured data is made up of well-defined data types with patterns (using algorithms and coding) that make them easily searchable, whereas unstructured data is a bundle of files in various formats, such as videos, photos, texts, audio, and more.

Unstructured data exists in unmanaged file structures, so engineers collect, manage, and store it in database management systems (DBMS) turning it into structured data that is searchable. Unstructured data might be inputted through manual entry or batch processing with coding, so ELT is the tool used to transform and integrate data into a cloud-based data warehouse.

Second, you can share a situation in which you transformed data into a structured format, drawing from learning projects if you’re lacking professional experience.

### 9. What are the design schemas of data modeling?

Design schemas are fundamental to data engineering, so try to be accurate while explaining the concepts in everyday language. There are two schemas: star schema and snowflake schema.

**Star schema** has a fact table that has several associated dimension tables, so it looks like a star and is the simplest type of data warehouse schema. **Snowflake schema** is an extension of a star schema and adds additional dimension tables that split the data up, flowing out like a snowflake’s spokes.

### 10. What are big data’s four Vs?

The four Vs are volume, velocity, variety, and veracity. Chances are, the interviewer will ask you not just what they are, but why they matter. You might explain that big data is about compiling, storing, and exploiting huge amounts of data to be useful for businesses. The four Vs must create a fifth V, which is *value.*

* **Volume:**Refers to the size of the data sets (terabytes or petabytes) that need to be processed—for example, all of the credit card transactions that occur in a day in Latin America.
* **Velocity:** Refers to the speed at which the data is generated. Instagram posts have high velocity.
* **Variety:** Refers to the many sources and files types of structured and unstructured data.
* **Veracity:** Refers to the quality of the data being analyzed. Data engineers need to understand different tools, algorithms, and analytics in order to cultivate meaningful information.

### 11. Tell me some of the important features of Hadoop.

Hadoop is an open-source software framework for storing data and running applications that provides mass amounts of storage and processing power. Your interviewer is testing whether you understand its significance in data engineering, so you’ll want to explain that it is compatible with multiple types of hardware that make it easy to access.

Hadoop supports rapid processing of data, storing it in the cluster which is independent of the rest of its operations. It allows you to create three replicas for each block with different nodes (collections of computers networked together to compute multiple data sets at the same time).

### 12. Which ETL tools have you worked with? What is your favorite, and why?

The interviewer is assessing your understanding of and experience with ETL tools. You’ll want to list the tools that you’ve mastered, explain your process for choosing certain tools for a particular project, and choose one. Explain the properties that you like about the tool to validate your decision.

### 13. What is the difference between a data warehouse and an operational database?

For this question, you can answer by explaining that databases using Delete SQL statements, Insert, and Update focus on speed and efficiency, so analyzing data can be more challenging. With data warehouses, the primary focus is on calculations, aggregations, and select statements that make it ideal for data analysis.

### The final question: Do you have any questions for us?

Most interviews end with this question, in one form or another. Consider this your chance to end on a high note, because *not* asking questions reflects poorly—it could demonstrate that you are not interested in the company, the role, or learning more about how you could fit in. Prepare a few questions, and select at least two or three to ask during the interview. Common [questions](https://www.coursera.org/articles/questions-to-ask-at-the-end-of-an-interview) include:

* What is the company culture?
* What does a typical day look like in this job?
* What are the expectations for the first three months in the role, and what are the benchmarks for evaluating success?
* Who will I be working with?
* Is there any other information I can offer to clear up any doubts about my qualifications?

## Prepare for your data engineer interview

To prepare for your interview, you may find confidence in reviewing everything you’ve learned from previous roles and courses you’ve taken. Imagine yourself in the interview, whether it is in person or over Zoom, with the hiring manager asking you technical questions.

* **Study and master SQL.** Review data pipeline systems and emerging technologies in the Hadoop ecosystem.
* **Design a sample data pipeline.** Make sure you understand the objective, and how you factor in data lineage, data duplication, loading data, scaling, testing, and end-user access patterns.
* **Learn and review languages.**Look at the job description to understand what the role entails. For backend-oriented systems, you’ll want to know Scala, and for analytics and data science-oriented systems, you’ll want to be well-versed in Python.
* **Research potential interview questions.** Besides those listed above, you may be able to find interview questions for the company on Glassdoor. It’s worth peeking there as part of your prep, in case someone has kindly made that advice available to the public.
* **Talk through your process.**This is perhaps the most important tip of all. Knowing how to write code and assemble data is not enough, you must be able to communicate your process and decision-making to the interviewers. Practice by talking through a recent project to a friend who is unfamiliar with big data.