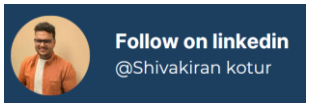
1. Reading and Writing Data

- How would you read a large CSV file from Azure Blob Storage into a Spark DataFrame?
- Describe the process for reading a JSON file with deeply nested structures and flattening it?
- How can you handle different delimiters when reading a text file into Spark?
- Write code to read a Parquet file and save it as a Delta Lake table?
- Explain how to read data from a Kafka stream into a Spark DataFrame?

---

2. Data Transformation

- How would you split a column with concatenated values into multiple columns?
- Describe the process of exploding a column that contains arrays into separate rows?
- Write a PySpark code snippet to remove special characters from a string column?
- How do you normalize numerical data in a DataFrame?
- Explain how to convert a column of timestamps into a different time zone?



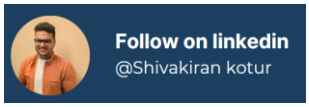
3. Aggregations and Metrics

- Write a Spark SQL query to calculate the total sales and average sales amount per product category?
- How would you find the top 10 customers by total spending?
- Describe how to compute the running total of a column over a specified window?
- How do you aggregate data by multiple columns and calculate statistics such as count, sum, and average?
- Write code to compute the median value of a column in a DataFrame?

---

4. Window Functions

- How do you use window functions to rank items within each partition of a DataFrame?
- Write a PySpark example to calculate the moving average of sales over a 30-day window?
- Explain how to use window functions to compute cumulative sums or averages?
- How would you partition data by date and compute the last value in each partition using window functions?



- Describe the process to calculate the lag and lead values of a column in Spark?

---

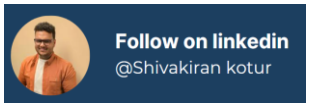
5. Performance Optimization

- What strategies would you use to optimize the performance of a Spark job?
- How do you handle data skew in a join operation to improve performance?
- Write code to repartition a DataFrame to optimize parallel processing?
- Describe how to cache intermediate results to speed up iterative computations?
- Explain the importance of optimizing shuffle operations and how to achieve it?

---

6. Handling Large Datasets

- How would you partition a large DataFrame to improve query performance?
- Describe the process of handling and processing a dataset that exceeds available memory?



- Explain how to use Delta Lake's data versioning to manage large datasets effectively?
- Write code to perform an incremental update on a large DataFrame?
- How would you manage large files and ensure efficient processing in Spark?

---

7. Data Quality and Validation

- How do you handle missing values in a DataFrame?
- Describe how to validate and clean data based on specific quality rules?
- Write a PySpark function to identify and remove duplicate records?
- How would you convert inconsistent date formats in a DataFrame to a standard format?
- Explain how to check for and handle data inconsistencies in a dataset?

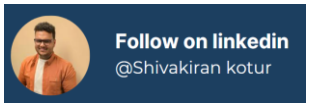
8. Data Integration and ETL

- How would you design an ETL pipeline to extract data from multiple sources and load it into a data lake?
- Describe the process of integrating data from relational databases into Spark?
- Write code to perform a merge operation between two DataFrames, handling conflicts and updates?
- Explain how to use PySpark to create a data pipeline that includes transformation and loading steps?
- How would you set up a scheduled ETL job in Databricks?

---

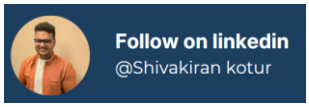
9. Joins and Unions

- Write code to perform an inner join between two DataFrames on a common key.
- How do you handle joins between DataFrames with different schemas?
- Explain the use of broadcast joins and when they are beneficial.
- Write a PySpark example to perform a union operation on two DataFrames with different column names.
- Describe how to handle null values during a join operation.



100+ different scenario based SPARK / DATABRICKS Questions

- Write code to perform a left outer join between two DataFrames and filter out rows with null values in the right DataFrame.
- How would you perform a cross join between two DataFrames and filter the results based on a condition?
- Explain how to perform a self-join on a DataFrame to find hierarchical relationships, such as employees reporting to managers.
- Write code to perform an anti-join to find records in one DataFrame that do not have matching records in another DataFrame.
- Describe how to use a full outer join to combine two DataFrames and include all records from both DataFrames.
- How would you handle duplicate records resulting from a union operation?
- Write code to perform a union of two DataFrames with different column types and ensure compatibility by casting columns appropriately.



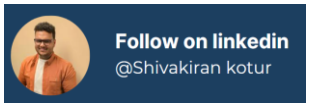
10. Data Storage and Formats

- Compare reading and writing data in Parquet format versus Avro format. Which is more efficient for large datasets?
- Write code to save a DataFrame as a Delta Lake table with partitioning?
- Explain how to read data from an external SQL database and load it into a Spark DataFrame?
- How would you convert a DataFrame into a JSON format and save it to cloud storage?
- Describe the advantages of using Delta Lake for data storage and management?

---

11. Streaming Data

- How do you set up a Spark Streaming job to process data from Kafka?
- Write code to write streaming data to a Delta Lake table with real-time updates?
- Explain how to handle late-arriving data in a streaming job?
- Describe how to perform aggregations on streaming data using Spark?

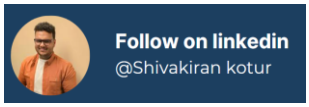


- How would you implement windowed aggregations on streaming data?
- Write code to process streaming data with schema evolution in mind?
- Explain how to manage stateful transformations in Spark Streaming?
- How would you handle data schema changes in real-time streaming pipelines?
- Write code to implement exactly-once processing semantics in a streaming job?
- Describe how to monitor and manage the performance of Spark Streaming jobs?

---

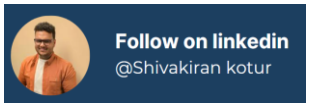
12. Advanced Transformations

- Write a PySpark UDF to perform sentiment analysis on text data.
- How would you apply a function to compute custom metrics for each record in a DataFrame? Provide a code example.
- Describe how to use PySpark's `explode` function to normalize a DataFrame with nested array columns.
- Explain how to create a derived column based on multiple existing columns using PySpark SQL functions.



100+ different scenario based SPARK / DATABRICKS Questions

- Write a PySpark code snippet to pivot a DataFrame based on a dynamic list of column values.
- How would you apply hierarchical clustering to a DataFrame? Describe the approach and provide code.
- Explain how to use PySpark's `flatMap` to handle complex nested data structures in a DataFrame.
- Describe the process of using PySpark's `withColumn` to perform conditional transformations based on complex business rules.
- Write a PySpark example to transform a DataFrame by merging multiple columns into a single JSON column.
- How would you perform a series of transformations on a DataFrame using PySpark's `transform` function? Provide an example.
- Explain how to use `groupBy` and aggregation functions to perform custom rolling calculations across partitions.
- Write a PySpark code snippet to handle and process complex XML data by extracting specific elements into a DataFrame.



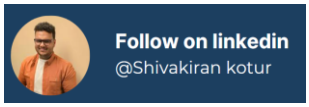
13. Data Security and Compliance

- How do you implement row-level security in Spark?
- Write code to encrypt data before saving it to a data lake?
- Describe how to ensure data compliance with privacy regulations in a Spark environment?
- Explain how to audit data access and modifications in a Spark job?
- How would you implement data anonymization techniques in a DataFrame?

---

14. Custom Functions and UDFs

- How do you register and use a user-defined function (UDF) in Spark?
- Write a PySpark UDF to calculate the Levenshtein distance between two strings?
- Describe how to optimize the performance of UDFs in Spark?
- How would you implement a custom aggregation function in Spark?
- Explain how to handle exceptions within UDFs and ensure data consistency?

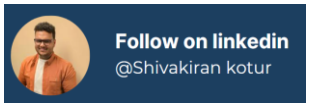


15.Incremental Loading

- How would you design an incremental loading strategy to process new and updated records in a DataFrame?
- Write a PySpark code snippet to implement incremental loading by comparing timestamp columns to identify new or modified records.
- Describe how to use Delta Lake for incremental updates and how to handle schema evolution in this context.
- Explain the process of maintaining and managing a watermark for handling late-arriving data in a streaming incremental load.
- How would you handle data deduplication in an incremental loading process to ensure that only unique records are loaded?

16. Data Versioning and Time Travel

- How do you use Delta Lake's time travel feature to query historical data?
- Describe the process of implementing data versioning in a Delta Lake table?
- Write code to roll back to a previous version of a Delta Lake table?
- Explain how to manage schema changes and maintain historical versions in a data lake?



- How would you use time travel to compare data snapshots in Delta Lake?

---

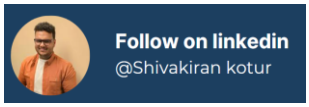
17. Error Handling and Logging

- How do you handle and log errors in a Spark job?
- Write code to implement custom error handling strategies in PySpark?
- Describe how to use Spark's logging features to monitor job execution?
- Explain how to capture and debug exceptions that occur during data processing?
- How would you set up alerts for job failures or performance issues?

---

18. Scalability and Resource Management

- How do you configure Spark to scale from a small cluster to a large cluster?



- Describe how to manage Spark resources and optimize cluster utilization?
- Write code to dynamically adjust the number of partitions based on data size?
- Explain how to use Spark's resource management features to handle varying workloads?
- How would you monitor and optimize Spark job performance across a distributed environment?

---

19. Data Engineering Best Practices

- How do you ensure data consistency and reliability in an ETL pipeline?
- Describe best practices for designing scalable data pipelines in Spark?
- Write code to implement data partitioning strategies for optimal performance?
- Explain how to manage and deploy Spark jobs in a production environment?
- How would you document and maintain data engineering workflows and processes?