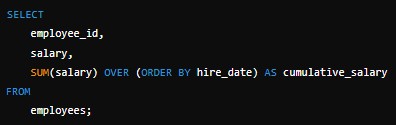
**Hexaware Technologies Data Engineer Interview Guide – Experienced 3+**

**Technical round 1 and 2 combined**

**1. Cumulative Sum in SQL – Explain how to implement.**

A cumulative sum (running total) adds each row's value to the sum of the preceding rows. This can be implemented using the SUM() window function.



Explanation:

 OVER (ORDER BY hire\_date) defines the order in which rows are processed.

 No PARTITION BY is used for a global running total; use it to reset per group.

**2. Delta Logs File Format – Discuss the format and its significance.**

Delta logs are stored in JSON and Parquet format in the \_delta\_log/ directory. These files track transaction history and data versions, enabling ACID transactions and time travel.

 **JSON** files store metadata for operations (add/remove files).

 **Parquet** files contain checkpoints for faster access.

**3. How to Access Delta Logs – Explain the process.**

To access Delta logs in Databricks:

1. Navigate to the \_delta\_log/ directory in the Delta Lake path.

2. Use %fs ls to list files:

%fs ls dbfs:/mnt/delta\_table/\_delta\_log/

3. Read JSON logs for operation history or Parquet for checkpoints.

**4. How to See Files Before Update (History Records/Versioning).**

Delta Lake provides a DESCRIBE HISTORY command to view version history. DESCRIBE HISTORY delta.`/mnt/delta\_table/`;

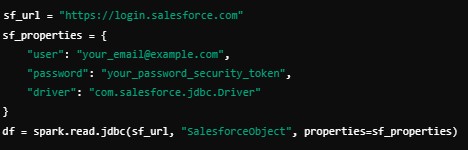
 Use VERSION AS OF or TIMESTAMP AS OF to query old data versions: SELECT \* FROM delta.`/mnt/delta\_table/` VERSION AS OF 3;

**5. How to Connect to Salesforce – Steps for Integration.**

1. Use a connector like **Databricks Partner Connect** or **JDBC/ODBC drivers**.

2. Generate Salesforce credentials (username, password, security token).

3. Example code using Spark:



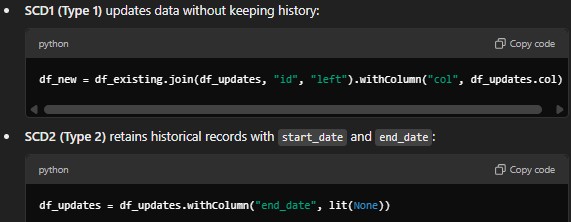
**6. How to Connect to Blob Storage in Databricks.** storage\_account\_name = "your\_account\_name" storage\_account\_key = "your\_account\_key" container\_name = "your\_container"

spark.conf.set(f"fs.azure.account.key.{storage\_account\_name}.blob.core.windows.net", storage\_account\_key)

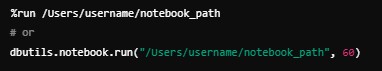
df = spark.read.csv(f"wasbs://{container\_name}@{storage\_account\_name}.blob.core.windows.ne t/data.csv")

**7. SCD1 and SCD2 in Databricks PySpark – Explain with examples.**

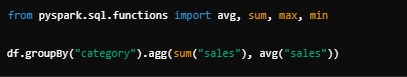
 **SCD1 (Type 1)** updates data without keeping history:



**8. How to Run One Notebook in Another Notebook – Use %run.**



**9. Aggregation Functions in PySpark – Examples and Use Cases.**



**10. SparkContext and SparkSession – Explain Their Purpose.**

 **SparkContext**: Core entry point for low-level Spark functionality.

 **SparkSession**: Unified API to create DataFrames, supports modern applications. spark = SparkSession.builder.appName("example").getOrCreate()

**11. Broadcast Join in PySpark – When and How to Use.**

Use for small lookup tables to avoid shuffle operations:

from pyspark.sql.functions import broadcast df\_result = df\_large.join(broadcast(df\_small), "key")

**12. How to Increase Job Performance – Techniques and Optimizations.**

 Use broadcast joins for small tables.

 Optimize partitions using repartition() or coalesce().

 Enable caching for iterative processes.

 Avoid wide transformations when possible.

**13. Cluster Types for Work – Explain Job vs. Interactive Clusters.**

 **Job Clusters**: Temporary, used for scheduled tasks.

 **Interactive Clusters**: Long-running, used for development and testing.

**14. How to Copy All Files from One Source Path to Target in ADF.**

 Use a **Copy Data activity** with source and sink configuration.

 Enable recursive file copying for nested directories.