Azure Databricks

Top 100+ Company–Specific Interview Questions and Answers Guide

Crack Interviews with Confidence

- Sourced from genuine candidate experiences across 20+ companies
- Already trusted by 100+ learners who've leveled up their interview prep
- Instant Download + Lifetime Access

Topmate

Take the next step towards your Azure Data Engineering dream job – today

Follow Me to Get Such More Updates

Praveen Patel
Mentor & Azure Data Engineer

About Creator

Hi, I’m Praveen Patel, a dedicated Azure Data Engineer & Top 0.1% Mentor at Topmate.io, and a Top Azure Data Engineering Content Creator on LinkedIn with a fast-growing community of professionals and job seekers.

Over the months, I’ve mentored many learners, helping them crack real interviews, build project-based confidence, and land roles in top MNCs. I specialize in breaking down complex Azure concepts into simple, real-time use cases that learners can apply instantly.

Through this course, I’ve combined everything I’ve learned as a mentor and content creator — including

100+ interview questions (with answers) directly collected from candidates placed at TCS, Infosys, EY, Accenture, Cognizant, and more.

Whether you're preparing for your first job or switching to a better one, I’m here to guide you with practical skills and real interview readiness.

Let’s build your Azure Data Engineering career — together.

Previously, These top 100+ interview questions were asked in the following companies for Azure Data Engineer Role Experience Range 3 to 10 years, Package Range between 15 and 35 LPA

- EY
- TCS
- Accenture
- PWC
- KPMG
- Infosys (Created By Praveen Patel)
- Tiger Analytics
- Tech Mahindra
- Fractal Analytics
- Deloitte
- Zensar
- Capgemini
- Databricks
- Walmart
- Publicis Sapient
- Tredence
- Mastercard
- Cognizant
- LTI Mindtree
- Hexaware
- NTT Data

Q.1). Describe the concept of Delta Lake and its advantages ?

Asked in Tiger Analytics, TCS, Deloitte (Created By Praveen Patel)

Delta Lake is an open-source storage layer that brings ACID transactions to Apache Spark and big data workloads. It stores data in the Parquet format and adds transaction logs for consistency. Its key advantages include versioned data (time travel), support for scalable metadata handling, schema enforcement, and easy handling of batch and streaming data in one pipeline.

Created By – Praveen Patel

Q.2). Explain the significance of Z-ordering in Delta tables ?

Asked in Tiger Analytics (Created By Praveen Patel)

Z-ordering is a technique used in Delta Lake to optimize the storage layout of data. It organizes data files based on one or more columns to improve query performance. For example, if you're frequently filtering by "customer_id", Z-ordering on that column groups related records together, reducing the amount of data scanned and speeding up query performance, especially on large datasets.

Q.3). Define Delta logs, and how to track data versioning in Delta tables? Asked in Tredence, Capgemini (Created By Praveen Patel)

Delta logs are transaction logs stored in the **_delta_log** directory of a Delta table. These logs record every operation like inserts, updates, and deletes. Delta Lake uses these logs to support features like data versioning. By using **DESCRIBE HISTORY** in Databricks, you can track changes and access previous versions of your table, enabling rollback, audit, and time-travel operations.

Q.4). What is the use of Delta Lake, and how does it support ACID transactions? Asked in Tredence (Created By Praveen Patel)

Delta Lake is used to build reliable data pipelines by adding ACID (Atomicity, Consistency, Isolation, Durability) transaction support to data lakes. It ensures that read and write operations are consistent, even during concurrent operations. The transaction log maintains the state of the data, ensuring no partial or corrupted writes, making it ideal for big data systems requiring strong data consistency.

Q.5). How do you handle schema evolution in Delta Lake?

Asked in Tredence, NTTData (Created By Praveen Patel)

Delta Lake supports automatic schema evolution, which means it can adapt to changes in the data schema over time. When enabled, you can add new columns or modify the schema during write operations using options like **mergeSchema** in Databricks. This makes it easier to handle changes in source systems without breaking your pipelines or manually updating the table structure.

Created By – Praveen Patel

Q.6). What are the best practices for managing large datasets in Databricks? Asked in Tredence, Capgemini (Created By Praveen Patel)

To manage large datasets in Databricks effectively, use Delta Lake for reliability, apply Z-ordering for faster queries, and partition tables based on frequently filtered columns. Also, clean up unused files using **VACUUM**, enable auto-optimize and auto-compaction, and monitor workloads with cluster autoscaling. Regularly updating statistics and avoiding small files are also important for maintaining performance.

Q.7). Explain the use of Delta Lake for data versioning ?

Asked in Deloitte (Created By Praveen Patel)

Delta Lake enables data versioning by storing metadata about changes made to a dataset. Every write operation creates a new version using a transaction log. This allows users to access and query older versions of data using the “time travel” feature. It’s useful for auditing, rollback, debugging, or reproducing past reports with exact historical data.

Q.8). What are the key features of Databricks notebooks?

Asked in Deloitte (Created By Praveen Patel)

Databricks notebooks offer collaborative features like real-time code sharing and comments. They support multiple languages (like Python, SQL, Scala) in the same notebook. Key features include interactive visualizations, version control, rich text formatting using Markdown, and easy integration with data sources. Notebooks also support scheduled jobs, allowing users to automate and monitor data workflows efficiently.

Created By – Praveen Patel

Q.9). Explain the concept of Data Lakehouse ?

Asked in Deloitte (Created By Praveen Patel)

A Data Lakehouse combines the low-cost storage of a data lake with the structure and performance of a data warehouse. It supports both structured and unstructured data while enabling ACID transactions, schema enforcement, and data governance. This architecture simplifies data pipelines, reduces duplication, and allows advanced analytics and machine learning on a unified platform.

Q.10).How do autoscaling clusters work in Databricks ?

Asked in Fractal (Created By Praveen Patel)

Autoscaling clusters in Databricks automatically increase or decrease the number of worker nodes based on workload demand. When jobs require more resources, the cluster scales up; when activity reduces, it scales down to save costs. This dynamic scaling ensures efficient resource usage, minimizes manual intervention, and maintains performance for both batch and streaming workloads without overprovisioning compute.

Created By – Praveen Patel

Q.11). How do you implement streaming pipelines in Databricks ?

Asked in Fractal (Created By Praveen Patel)

Streaming pipelines in Databricks are implemented using Structured Streaming APIs with Delta Lake. We read data from streaming sources like Kafka or Event Hubs, process it in micro-batches or continuous mode, and write to a Delta table or sink. The pipeline ensures fault tolerance using checkpoints and watermarking to handle late data, enabling reliable, real-time analytics.

Q.12). Explain the purpose of Delta Lake checkpoints ?

Asked in Fractal (Created By Praveen Patel)

Delta Lake checkpoints store the state of a Delta table at a specific point in time, enabling faster query performance. Instead of scanning the full transaction log, Databricks reads from the latest checkpoint and applies only newer changes. Checkpoints improve read efficiency, reduce latency, and are created automatically every 10 commits by default for optimized processing.

Q.13). What is the difference between a job cluster and an interactive cluster in Databricks? Asked in Infosys (Created By Praveen Patel)

A job cluster is created automatically for a specific job and terminates after the job completes, making it cost-effective for scheduled or automated workflows. An interactive cluster is manually created and remains active for tasks like development, debugging, or ad-hoc analysis. Job clusters ensure resource isolation per job, while interactive clusters support collaboration and iterative notebook execution.

Q.14). Explain the concept of Delta Lake compaction ?

Asked in Infosys (Created By Praveen Patel)

Delta Lake compaction, also known as file compaction or optimize, is the process of merging many small files into fewer large files using the OPTIMIZE command. This is essential because frequent small writes can degrade query performance. Compaction reduces file overhead, improves read efficiency, and speeds up downstream queries, especially in partitioned Delta tables. It's commonly used in streaming or microbatch pipelines.

Created By – Praveen Patel

Q.15). How do you implement incremental load in Databricks ?

Asked in Infosys (Created By Praveen Patel)

Incremental load in Databricks is implemented using techniques like watermarking in Structured Streaming or comparing audit columns (e.g., last_updated_date) in batch processing. You load only new or changed data by filtering based on timestamps or versioning, then merge into Delta tables using MERGE INTO. This approach optimizes performance and reduces processing overhead.

Q.16). What is AQE (Adaptive Query Execution) in Databricks ?

Asked in Infosys (Created By Praveen Patel)

Adaptive Query Execution (AQE) in Databricks optimizes queries at runtime based on actual data statistics. It adjusts joins, partitions, and query plans dynamically during execution. For example, it can switch from sort-merge to broadcast join if one side is small. AQE enhances performance without manual tuning, especially helpful with skewed or unknown data distributions.

Q.17).How do you manage and automate ETL workflows using Databricks Workflows ?

Asked in Infosys (Created By Praveen Patel)

Databricks Workflows allows you to orchestrate ETL pipelines by chaining notebooks, Python scripts, JARs, or Delta Live Tables into a scheduled job. You can set dependencies between tasks, manage retries, set timeouts, and pass parameters across tasks. This ensures modular, reusable, and maintainable ETL flows. Scheduling and triggering workflows through UI, API, or REST endpoints allows full automation of your data pipelines without relying on external schedulers.

Created By – Praveen Patel

Q.18). What is the role of Delta Lake in modern data architectures ?

Asked in Infosys (Created By Praveen Patel)

Delta Lake plays a vital role in modern data architectures by bringing ACID transactions, schema enforcement, and time travel to data lakes. It enables reliable and consistent data pipelines on cloud storage like ADLS or S3. This unifies batch and streaming workloads, ensuring high data quality, auditability, and simplified ETL processes in lakehouse designs.

Q.19). How do you implement real-time data processing in Databricks? Asked in Infosys (Created By Praveen Patel)

Real time data processing in Databricks is achieved using Structured Streaming with Delta Lake. Data is ingested from streaming sources like Kafka, Event Hubs, or Auto Loader. We define a streaming DataFrame, apply transformations, and write results to Delta tables or other sinks. Databricks manages state, checkpointing, and fault tolerance automatically, enabling scalable, low-latency processing. This approach supports continuous ETL, real-time analytics, and dashboards with fresh data always available.

Created By – Praveen Patel

Q.21). How do you implement data governance in a data lake environment ? Asked in TCS (Created By Praveen Patel)

In data lake environment, I implement data governance by combining access control, data security, and metadata management. I use Azure ACLs or Unity Catalog for role-based access, apply data masking for sensitive columns, and manage metadata with a centralized catalog for discoverability. Auditing and data lineage tools help track usage and changes. This ensures compliance, improves trust in data, and enables secure, organized access across teams and workloads.

Q.22). How to create and deploy notebooks in ADB ? (Asked in KPMG)

To create a notebook in Databricks, navigate to your workspace, click “New” → “Notebook”, choose a name, language (Python, SQL, etc.), and cluster. You can write code in cells, run them interactively, and visualize results. To deploy, schedule it as a job using the Jobs UI, configure the cluster, parameters, and triggers. This automates execution and enables monitoring, making it production-ready for batch or streaming tasks.

Q.23). How do you connect ADLS to Databricks ? Asked in KPMG

To connect ADLS Gen2 to Databricks, you typically use Azure service principal authentication with OAuth 2.0. Register an app in Azure AD, assign it the necessary permissions (Storage Blob Data Contributor), and generate a client ID and secret. Then, configure a Spark mount point in Databricks using the dbutils.fs.mount() command with your credentials and storage URL. This enables secure and scalable access to ADLS from Databricks notebooks and jobs.

Q.24).What is alternative to medallion architecture ?

Asked in KPMG (Created By Praveen Patel)

An alternative is the Lambda Architecture, which combines batch and streaming data processing through three layers - batch (historical data), speed (real-time data), and serving (merged views). It enables real-time analytics while retaining historical accuracy. Unlike the Medallion (Bronze-Silver-Gold) model used in Delta Lake, Lambda involves maintaining two code paths, making it more complex. It suits systems needing low-latency insights with high data volume, though Medallion is simpler for unified ETL pipelines.

Created By – Praveen Patel

Q.25).What is medallion Architecture ? Asked in KPMG

The Medallion Architecture in Databricks is a data design pattern that organizes data into Bronze, Silver, and Gold layers. The Bronze layer stores raw, ingested data, Silver cleanses and transforms it, Gold holds curated, business-ready data for analytics. This layered approach improves data quality, scalability, and governance while supporting realtime and batch processing pipelines efficiently in lakehouse environments.

Q.26). Difference Between Spark, Pyspark and Databricks ?

Asked in KPMG and EY (Created By Praveen Patel)

Apache Spark is a fast, distributed data processing engine for big data analytics. PySpark is the Python API for Spark, allowing users to write Spark applications using Python. Databricks is a cloud-based platform built on top of Spark, offering a collaborative environment, notebooks, cluster management, and optimized performance features like Delta Lake and MLflow for simplified big data and AI workflows

Created By – Praveen Patel

Q.27). Difference Between Control Plane and Data Plane ?

Asked in Big Four (KPMG, PWC, EY, Deloitte) Created By – Praveen Patel

The control plane manages and orchestrates Databricks resources like cluster creation, job scheduling, and user management hosted in the Databricks cloud. The data plane is where actual data processing happens, typically within your cloud account (e.g., Azure or AWS). This separation ensures security, as your data never leaves your environment, while Databricks handles operations and monitoring securely from its control plane.

Q.28). What are different types of cluster in azure databricks ?

Asked in Big Four (KPMG, PWC, EY, Deloitte)

Azure Databricks supports two main types of clusters Interactive clusters (used for development, notebooks, and ad-hoc analysis) and Job clusters (created automatically to run scheduled jobs or production workflows). Interactive clusters are manually managed, whereas job clusters are ephemeral and cost-efficient.

Q.29).What is the difference between data lake and delta lake ?

Asked in Big Four (KPMG, PWC, EY, Deloitte)

A Data Lake stores raw data in various formats but lacks ACID transaction support and data versioning. Delta Lake is built on top of Data Lake and adds ACID transactions, schema enforcement, time travel, and better performance using a transaction log. While Data Lake is ideal for storing big data, Delta Lake makes it reliable and suitable for analytical workloads.

Q.30).Describe your approach to designing a data lakehouse architecture using Azure and Databricks ? Asked in HCL

To design a data lakehouse, I use Azure Data Lake Gen2 for raw data storage and Delta Lake on Databricks for processing. I structure layers as Bronze (raw), Silver (cleaned), and Gold (aggregated). Azure Data Factory handles orchestration, and Unity Catalog manages governance. This architecture enables scalable, reliable, and cost-effective analytics with strong data quality and security controls

Created By – Praveen Patel

Q.31). What is your strategy for managing schema evolution in batch pipelines using Delta Lake?

Asked in TCS (Created By Praveen Patel)

My strategy uses mergeSchema in write operations to handle new columns automatically. I enable schema enforcement in production to prevent breaking changes and manage schema versions via Git or notebooks. All schema changes are tested in lower environments before deployment. This approach ensures safe, backward-compatible evolution in batch pipelines using Delta Lake

Q.32). What is Auto Optimize in Databricks?

Asked in Tiger Analytics, Deloitte, EY (Created By Praveen Patel)

Auto Optimize in Databricks automatically optimizes file sizes during write operations to Delta Lake tables. It reduces the creation of small files by compacting them in real-time or asynchronously. This feature improves read performance and reduces storage overhead without manual OPTIMIZE commands. Auto Optimize works well for frequent writes and streaming workloads, ensuring efficient data layout for faster query execution.

Created By Praveen Patel

Q.33). Explain the architecture of Databricks?

Asked in Tiger Analytics, EY, TCS (Created By Praveen Patel)

Databricks architecture is built on a Lakehouse model combining data lake and data warehouse features. It runs on cloud platforms like Azure and uses separate compute and storage layers. Data is stored in ADLS or S3, while compute is managed through clusters. The control plane manages jobs, notebooks, and user access, and the data plane executes code close to the data.

Q.34). What is the Parquet file format, and how does it differ from Delta ? Asked in Tech Mahindra, Tiger Analytics, Infosys

Parquet is a columnar storage format optimized for read-heavy analytics and efficient data compression. Delta Lake builds on Parquet by adding ACID transactions, versioning, schema enforcement, and time travel. While Parquet is static, Delta enables reliable data updates, deletes, and merges making it ideal for both batch and streaming data pipelines.

Q.35). How do you perform time travel in a Delta table?

Asked in Tiger Analytics (Created By Praveen Patel)

Delta Lake supports time travel using the VERSION AS OF or TIMESTAMP AS OF options in SQL or Python. You can query or restore previous states of a Delta table by specifying a version number or timestamp. This is useful for auditing, debugging, or restoring accidentally deleted or changed data.

Q.36). What is Incremental Loading in Delta table, how would you implement it ?

Asked in Wipro, EY, Deloitte, KPMG, Tiger Analytics

Incremental loading in a Delta table means loading only new or changed data instead of full data every time. It's implemented using filters like last_updated_date or using change data capture (CDC) mechanisms. In Databricks, we use MERGE INTO to update the Delta table efficiently, ensuring minimal data movement and optimized performance for large datasets.

Created By Praveen Patel

Q.37). What is the role of Data Vaccuming in Delta Lake ?

Asked in Tiger Analytics (Created By – Praveen Patel)

Data vacuuming in Delta Lake permanently deletes obsolete data files that are no longer referenced by the Delta transaction log. This helps reduce storage costs and keeps the storage layer clean. It's triggered using the VACUUM command and works only after the retention period, ensuring safe and efficient data cleanup.

Q.38).What is Delta Table ?

Asked in EY, PWC (Created By Praveen Patel)

A Delta Table is a transactional storage layer built on top of data lakes using Delta Lake. It supports ACID transactions, scalable metadata handling, and unified batch and streaming data processing. Delta Tables ensure data reliability, allow time travel, and enable efficient updates, merges, and deletes directly within the data lake.

Created By Praveen Patel

Q.39).What is the difference between managed table and external table ?

Asked in EY (Created By Praveen Patel)

In a managed table, Databricks controls both the data and metadata; if the table is dropped, the data is deleted. In an external table, only the metadata is managed, while the data stays in the external location (e.g., ADLS). Dropping an external table keeps the data intact, offering more control over data storage.

Q.40). What is Unity Catalog in Databricks? How have you used it for data governance or access control ?

Asked in Wipro (Created By Praveen Patel)

Unity Catalog is a unified governance solution in Databricks that manages permissions across workspaces using a centralized metastore. I used it to define fine-grained access at the table, column, and row level, ensuring role-based control and auditability. It simplified compliance and improved data security by centralizing policies across all data assets in the lakehouse.

Q.41).What are the differences between Data Lake and Delta Lake? Why was Delta Lake introduced?

Asked in Wipro (Created By – Praveen Patel)

A Data Lake stores raw data but lacks ACID compliance and schema enforcement, leading to data quality issues. Delta Lake adds ACID transactions, schema evolution, versioning, and time travel to Data Lake storage. It was introduced to bring reliability, consistency, and performance to data lakes, enabling better handling of big data pipelines and analytics.

Created By – Praveen Patel

Q.42).What is Autoloader in Azure Databricks

Asked in Wipro (Created By – Praveen Patel)

Autoloader in Azure Databricks is a highly scalable and efficient tool to incrementally load new files from cloud storage like ADLS Gen2. It uses the **cloudFiles** syntax with Structured Streaming, supports schema evolution, and eliminates file listing bottlenecks. Autoloader is ideal for building production-grade ingestion pipelines that automatically handle new data with minimal code and high reliability.

Q.43).What is delta live tables ?

Asked in Accenture (Created By – Praveen Patel)

Delta Live Tables (DLT) is a Databricks framework for building reliable ETL pipelines using SQL or Python. It automates data processing, supports streaming and batch workloads, manages dependencies, and enforces data quality. DLT simplifies pipeline creation, tracks data lineage, and ensures fresh, accurate data with minimal manual effort.

Q.44).What is the difference between Data Warehouse, Data Lake and Delta Lake ?

Asked in Infosys, KPMG, EY, Tiger Analytics (Created By – Praveen Patel)

A Data Warehouse is optimized for structured, historical data with strict schema and fast SQL queries. A Data Lake stores raw, unstructured to structured data at scale but lacks data management features. Delta Lake enhances Data Lakes with ACID transactions, time travel, and schema enforcement, offering a unified platform that blends warehouse reliability with lake flexibility for modern analytics.

Created By Praveen Patel

Q.45).How do you create a delta table in databricks, and how would you manage ACID within same ? Asked in Publicis Sapient

In Databricks, a Delta table is created using SQL **(CREATE TABLE USING DELTA)** or via DataFrame writes with .**write.format("delta")**. Delta Lake enforces ACID properties through a transaction log that records every operation. This ensures atomic commits, isolation during concurrent updates, consistent reads, and durability—making data pipelines fault-tolerant, reliable, and perfectly suited for enterprise-grade workloads.

Q.46).How do you create view in Databricks ?

Asked in Publicis Sapient (Created By – Praveen Patel)

In Databricks, a view is created using CREATE OR REPLACE VIEW to simplify complex queries and promote reusability. For example -

```
CREATE OR REPLACE VIEW active_users AS
SELECT id, name FROM users WHERE status = 'active';
```

Views act as logical layers, helping teams standardize business logic, secure data access, and streamline analytics without duplicating storage.

Q.47).What is mount point in azure databricks ? How do you mount ADLS Gen2 to Databricks ? Asked in Accenture, Deloitte, Capgemini

A mount point in Azure Databricks is a path within the Databricks File System (DBFS) that links to external storage like ADLS Gen2. It allows users to access data using standard file paths. You mount ADLS Gen2 using **dbutils.fs.mount()** by providing the storage URL, mount path, and credentials (like service principal or OAuth), enabling easy, secure data access.

Q.48).How to integrate Databricks with Delta Lake ?

Asked in Kaseya, Tiger Analytics, Infosys (Created By Praveen Patel)

To integrate Databricks with Delta Lake, create a Delta table using Apache Spark APIs within Databricks. Write data using **write.format("delta")** and read using **read.format("delta")**. Delta Lake provides ACID transactions, schema enforcement, and time travel. Ensure the Databricks runtime supports Delta Lake, and use notebooks or jobs to process and manage structured data efficiently for scalable, fault-tolerant pipelines.

Created By Praveen Patel

Q.49). Explain the key features of Azure Databricks ?

Asked in Kaseya, Infosys, Wipro (Created By Praveen Patel)

Azure Databricks is a unified analytics platform built on Apache Spark, offering high-performance data processing, machine learning, and real-time analytics. Key features include collaborative notebooks, seamless integration with Azure services, autoscaling clusters, Delta Lake for reliable data lakes, support for multiple languages (SQL, Python, Scala), and robust job scheduling. It simplifies big data workflows with enterprise-grade security, scalability, and interactive development capabilities.

Q.50).How do you configure resources in databricks ?

Asked in Kaseya, Infosys, Tech Mahindra (Created By - Praveen Patel)

You configure resources in Databricks by selecting cluster size, node types (driver and workers), autoscaling, and runtime version while creating a cluster. You can also attach libraries, set environment variables, and define instance pools for cost efficiency. Proper configuration ensures the cluster meets performance needs while optimizing resource usage and cost based on the workload.

Q.51).What is the difference between databricks catalog table and a delta table ?

Asked in Deloitte (Created By – Praveen Patel)

A Databricks catalog table is a metadata object registered in Unity Catalog or Hive Metastore, which can point to data in any format (CSV, Parquet, Delta). A Delta table specifically uses Delta Lake format with ACID transactions, versioning, and schema enforcement. So, all Delta tables can be catalog tables, but not all catalog tables are Delta tables.

Created By – Praveen Patel

Q.52).Explain the difference between silver and gold layer in medallion architecture ?

Asked in Deloitte (Created By – Praveen Patel)

In medallion architecture, the silver layer contains cleaned and enriched data, often joined from multiple bronze sources. It serves as a refined, query-ready layer. The gold layer contains aggregated, business-level data used for reporting and analytics. Silver focuses on data quality and transformation; gold focuses on insights, KPIs, and consumption by BI tools or dashboards.

Q.53).What are the different ways to create a secret scope in azure databricks ?

Asked in Tech Mahindra (Created By – Praveen Patel)

You can create a secret scope in Azure Databricks using two main methods - the Databricks CLI and the Databricks UI. The CLI is preferred for automation by running databricks secrets create-scope. The UI method involves navigating to the "User Settings" → "Access Tokens" → "Create Secret Scope." Both methods securely store credentials like keys and passwords for pipelines.

Q.54).What are the different ways to install libraries in an azure databricks notebook ?

Asked in Tech Mahindra (Created By – Praveen Patel)

You can install libraries in Databricks using **%pip** or **%conda** magic commands directly in notebooks, via the Libraries tab in a cluster, or through init scripts. You can also use workspace or cluster-scoped libraries from PyPI, Maven, or upload **.whl/.jar files**. %pip is the most flexible and preferred method.

Created By – Praveen Patel

Q.55).How to mount storage location in databricks ?

Asked in Tech Mahindra (Created By – Praveen Patel)

To mount a storage location in Databricks, use the **dbutils.fs.mount()** function with the source path (e.g., ADLS Gen2), mount point, and credentials. This creates a reusable path like **/mnt/mydata**. Mounting allows simplified access to files using standard paths. Example -
dbutils.fs.mount("source"**,** "/mnt/mydata"**, extra_configs={...})**
Used typically for frequent file access.

Q.56).You have Data in ADLS Gen2 How do you access it in Databricks?

Asked in Infosys (Created By – Praveen Patel)

To access ADLS Gen2 data in Databricks, configure the storage account with OAuth or SAS authentication. Mount the ADLS Gen2 path using dbutils.fs.mount, or access it directly using the abfss:// URI. Set Spark configs with service principal credentials to enable secure access. Once configured, read data using Spark APIs like spark.read.format().load() for processing.

Created By – Praveen Patel

Q.57). How would you handle slow query performance for a single-user SQL endpoint in Databricks, where all sequentially run queries are affected?

Asked in Databricks (Created By – Praveen Patel)

I'd start by reviewing the Query Profile to identify bottlenecks like skewed joins or long shuffle operations. Next, I'd check endpoint size and enable Photon for acceleration. If multiple queries are slow, I'd restart the endpoint to reset driver state and memory. I'd also analyze recent query history for regressions and ensure Delta tables are optimized and vacuumed for performance.

Q.58). When should you use Delta Live Tables over standard data pipelines built on Spark and Delta Lake? Asked in Databricks

Use Delta Live Tables (DLT) when building reliable, declarative ETL pipelines that require automatic orchestration, monitoring, and data quality enforcement. Unlike standard Spark-Delta pipelines, DLT manages schema changes, retries, dependencies, and lineage out of the box. It's ideal for real-time and batch pipelines in production where minimal DevOps effort and high reliability are needed—especially in regulated or high-availability data environments.

Q.59). When should you use a job cluster instead of an all-purpose cluster?

Asked in Databricks (Created By – Praveen Patel)

Use a job cluster when running scheduled or automated workloads, like production ETL pipelines, where cost and performance are key. It spins up for the job and terminates after completion, reducing resource usage. In contrast, all-purpose clusters are ideal for interactive development or ad hoc analysis. Job clusters offer better resource isolation, security, and cost efficiency for non-interactive workloads.

Created By – Praveen Patel

Q.60).What is the difference between data lakehouse and data warehouse ?

Asked in Databricks, IBM (Created By – Praveen Patel)

A data warehouse is optimized for structured data and analytics, offering high performance and consistency. A data lakehouse combines features of both data lakes and warehouses—it handles structured, semi-structured, and unstructured data with support for ACID transactions, enabling advanced analytics and machine learning directly on raw data. Lakehouse offers flexibility with unified storage and compute.

Q.61).What is the relationship between delta table and parquet format ? Asked in TCS, Infosys (Created By – Praveen Patel)

Delta tables are built on top of the Parquet format, adding ACID transaction support, schema enforcement, and time travel capabilities. While Parquet stores data in a columnar format efficiently, Delta Lake enhances it with metadata and transaction logs, enabling reliable, scalable, and consistent data processing workflows in big data and streaming environments like Databricks.

Q.62). How do you import data into delta lake ?

Asked in TCS, Infosys, KPMG (Created By – Praveen Patel)

You can import data into Delta Lake by using Spark APIs like spark.read to load data from various sources (CSV, Parquet, JSON, etc.), then writing it using write.format("delta"). For Example - **spark.read.csv("path").write.format("delta").save("delta_table_path").** This enables schema enforcement, ACID transactions, and scalable data ingestion for analytics.

Created By – Praveen Patel

Q.63).Does Delta Lake offer access controls for security and governance ?

Asked in TCS, Cognizant (Created By – Praveen Patel)

Yes, Delta Lake supports access controls through integration with Unity Catalog in Databricks. Unity Catalog enables fine-grained access control at the table, column, and row levels using role-based permissions. This ensures secure data governance, centralized auditing, and compliance across workspaces, making Delta Lake suitable for enterprise-grade data protection and regulatory requirements.

Q.64).What does an index mean in the context of delta lake ?

Asked in TCS, Tech Mahindra, EY (Created By – Praveen Patel)

In Delta Lake, an index refers to data structures like Delta Lake's file-level statistics and data skipping indexes, which help speed up queries by skipping unnecessary files. Although Delta Lake doesn’t support traditional indexing, these optimizations enable faster filtering and scanning by leveraging min/max values and partitioning.

Q.65). How do you implement table level, column level, row level security in databricks ? Asked in TCS, Wipro

In Databricks, table and column-level security is implemented using Unity Catalog with GRANT statements to control access. Row-level security is achieved via dynamic views, where WHERE clauses filter data based on user identity using current_user(). Combine these features to enforce fine-grained access control across different user roles securely and efficiently within your data lakehouse.

Created By – Praveen Patel

Q.66).What are the limitations of delta lake merges and how do you mitigate them ?

Asked in TCS, Accenture, Zensar (Created By – Praveen Patel)

Delta Lake merges can face limitations like high compute cost, performance degradation with large datasets, and slow processing due to frequent updates or skewed data. To mitigate these, use optimized file sizes, partition pruning, Z-ordering for efficient filtering, and ensure proper clustering. Also, avoid excessive small files and leverage OPTIMIZE and VACUUM commands regularly.

Q.67).What is mounting in Databricks and why it is useful ?

Asked in TCS, Wipro, IBM (Created By – Praveen Patel)

Mounting in Databricks allows you to link external storage (like Azure Data Lake or S3) to DBFS (Databricks File System), enabling easy access using standard file paths. It's useful because it simplifies data access, avoids repeated authentication, and provides a consistent way to read/write data across notebooks without dealing with complex storage APIs.

Q.68). What is cluster ? Explain it's types and use case ?

Asked in Infosys, Tech Mahindra, TCS (Created By Praveen Patel)

A cluster is a set of compute resources (nodes) used to run big data workloads in parallel. In Databricks, clusters are of two types - All-purpose clusters (for interactive data analysis) and Job clusters (for automated tasks). All-purpose clusters support notebooks and collaboration, while Job clusters are ephemeral and optimized for jobs. Use case: Use Job clusters for scheduled ETL pipelines and All-purpose clusters for ad-hoc data exploration.

Created By – Praveen Patel

Q.69).What are the different utilities in Azure Databricks ?

Asked in Fractal, Capgemini (Created By – Praveen Patel)

Azure Databricks provides key utilities via dbutils, including fs (file system access), notebook (run notebooks), widgets (parameterize notebooks), library (manage libraries), secrets (secure credentials), and data (access tables). These utilities help streamline development, enable dynamic workflows, securely handle data, and simplify interactions with notebooks and external storage within Databricks environments.

Q.70). How does Databricks handle cluster management and resource allocation for optimized performance?

Asked in Accenture, KPMG, Wipro (Created By – Praveen Patel)

Databricks handles cluster management using dynamic allocation and autoscaling, which automatically adjusts the number of executors based on workload. It supports job, all-purpose, and interactive clusters with optimized configurations. Resources are managed through Apache Spark's scheduler and Databricks' own intelligent workload management to ensure high performance, reduced idle time, and cost-efficient execution of tasks

Q.71). What are the main performance bottlenecks in Databricks Spark jobs, and how do you troubleshoot them?

Asked in Databricks, Fractal (Created By Praveen Patel)

Common performance bottlenecks in Databricks Spark jobs include skewed data, excessive shuffling, small files, and inefficient partitioning. Troubleshooting involves analyzing Spark UI stages, optimizing joins, tuning partitions, and caching intermediate data. Use broadcast joins, ZORDER in Delta Lake, and adaptive query execution to improve performance. Monitoring job metrics and profiling data distributions are also key to effective debugging.

Q.72). What are Delta Lakes in Databricks, and how do they improve data reliability compared to Parquet or ORC ?

Asked in Databricks (Created By Praveen Patel)

Delta Lake in Databricks is an open-source storage layer that brings ACID transactions to Apache Spark and big data workloads. Unlike Parquet or ORC, it ensures data reliability with features like schema enforcement, time travel, and atomic operations. This prevents data corruption, enables consistent reads/writes, and supports reliable batch and streaming data processing at scale.

Q.73). Explain how ACID transactions work in Delta Lake and their benefits in a Data Engineering workflow ?

Asked in Accenture (Created By Praveen Patel)

ACID transactions in Delta Lake ensure Atomicity, Consistency, Isolation, and Durability, enabling reliable data operations. They allow concurrent reads/writes without corrupting data, maintain schema integrity, and support rollback on failures. This is crucial in data engineering for maintaining data accuracy, preventing partial updates, and ensuring trustworthy analytics pipelines across batch and streaming workflows.

Q.74).What are the types of cluster in Azure Databricks ?

Asked in IBM, Fractal (Created By Praveen Patel)

Azure Databricks offers three main cluster types - Interactive Clusters for data exploration and development, Job Clusters for running automated jobs and pipelines, and Shared Clusters used across multiple users or jobs. Each type supports autoscaling and custom configurations, allowing teams to balance cost, performance, and resource isolation based on workload requirements.

Q.75). What is Databricks Unit ? and Explain caching in databricks ?

Asked in Cognizant (Created By – Praveen Patel)

A Databricks Unit (DBU) is a normalized unit of processing power billed per second, based on the instance type and workload. It helps measure resource usage. Caching in Databricks stores frequently accessed data in memory (using cache() or persist()), reducing disk I/O and improving performance, especially for iterative operations in Spark workflows.

Created By – Praveen Patel

Q.76).What is Databricks Runtime explain it with its key features ?

Asked in Cognizant, Tiger Analytics, EY (Created By – Praveen Patel)

Databricks Runtime is an optimized Apache Spark environment on the Databricks platform. It includes performance-boosted Spark, ML libraries, Delta Lake support, and built-in connectors. Key features are faster job execution, auto-optimization, GPU acceleration (ML Runtime), native Delta support, and seamless integration with cloud storage and data services, enhancing both batch and streaming workloads.

Q.77).How do you optimize performance in databricks ?

Asked in Cognizant, Accenture, TCS (Created By – Praveen Patel)

To optimize performance in Databricks, use Delta Lake for faster reads/writes, enable Auto Optimize and Z-Ordering, cache intermediate data when reused, choose appropriate cluster size and autoscaling, minimize data shuffling, and use broadcast joins wisely. Also, monitor with Spark UI and use adaptive query execution (AQE) to dynamically optimize execution plans during runtime.

Created By – Praveen Patel

Q.78).How does Z ordering improve query performance in Delta Lake Tables ?

Asked in TCS, Fractal (Created By – Praveen Patel)

Z-ordering in Delta Lake improves query performance by colocating related data in the same set of files. It organizes data based on specified columns using a multi-dimensional clustering technique. This reduces the amount of data scanned during queries by pruning files that don't match filter conditions. As a result, it speeds up selective queries, especially on large tables with frequently filtered columns

Q.79).How do you implement data governance in a data lake environment?

Asked in TCS (Created By – Praveen Patel)

Data governance in a data lake is implemented using access control, data classification, lineage tracking, and audit logging. Tools like Unity Catalog or Purview help enforce policies, monitor data usage, and ensure compliance. Organizing data with zones (raw, curated, trusted) and tagging sensitive information ensures secure, traceable, and well-managed data across the entire data lifecycle.

Q.80).What is the difference between Delta Table and Delta Live Tables ?

Asked in Capgemini (Created By – Praveen Patel)

Delta Table is a storage format built on Parquet that supports ACID transactions, time travel, and schema enforcement. It is used for batch and streaming data operations. Delta Live Tables (DLT) is a framework for building reliable ETL pipelines on Delta Tables with built-in monitoring, automated error handling, and data quality checks, simplifying pipeline development and management.

Created By – Praveen Patel

Q.81).What are the cluster modes and cluster types in Databricks?

Asked in Capgemini (Created By Praveen Patel)

Databricks supports three cluster modes - Standard, High Concurrency, and Single Node. Standard is ideal for engineering workloads, High Concurrency supports multiple users with SQL and Python, and Single Node is for lightweight tasks. Cluster types include All-purpose clusters (for interactive development) and Job clusters (ephemeral clusters launched for scheduled jobs or workflows), ensuring efficient resource utilization and workload-specific scalability.

Q.82).How databricks is related to pyspark and dataframe ?

Asked in Capgemini (Created By Praveen Patel)

Databricks is built on Apache Spark, and PySpark is its Python API. Within Databricks, PySpark allows developers to write Spark applications using Python. DataFrames are a key abstraction in PySpark for handling structured data efficiently. So, Databricks leverages PySpark and DataFrames to enable scalable, distributed data processing and analytics using familiar Python syntax.

Q.83).How do you handle delta load in databricks ?

Asked in Capgemini, Tech Mahindra (Created By Praveen Patel)

To handle delta load in Databricks, I use Merge (MERGE INTO) with Delta Lake. It allows comparing source and target data using keys and performing insert, update, or delete based on matched conditions. I ensure data consistency using watermarking or audit columns like last_updated. This efficiently processes only changed records without impacting existing data.

Created By – Praveen Patel

Q.84).How to Choose right Cluster Configuration for 100 GB + Workload ? Asked in NTT Data, Capgemini (Created By Praveen Patel)

To choose the right cluster configuration for a 100+ GB workload in Databricks, consider a Standard or Autoscaling Cluster with memory-optimized instances (e.g., E-series on Azure). Use 8–16 worker nodes with at least 32–64 GB RAM per node. Enable photon acceleration for performance boost. Adjust based on workload type—ETL, streaming, or ML—and monitor Spark UI for tuning CPU, memory, and shuffle performance bottlenecks.

Q.85).How to create and deploy notebooks in Databricks ?

Asked in KPMG, Cognizant (Created By Praveen Patel)

To create and deploy notebooks in Databricks, navigate to the Workspace, click “Create,” then select “Notebook.” Choose a language (Python, SQL, Scala, R), name the notebook, and attach it to a cluster. After developing the logic, you can deploy it by scheduling via Jobs or triggering with APIs. For version control, link notebooks to a Git provider like GitHub for CI/CD integration and collaborative development.

Q.86).How do you connect ADLS (Azure Data Lake Storage) to Databricks? Asked in KPMG, Cognizant, NTT Data (Created By – Praveen Patel)

To connect ADLS to Databricks, you configure access using either OAuth 2.0 with service principal or Azure managed identity. Set the Spark configurations with storage account credentials using spark.conf.set() or mount the ADLS path using dbutils.fs.mount(). This enables seamless read/write access to files stored in ADLS from your Databricks notebooks or jobs.

Created By - Praveen Patel

Q.87). How does Delta Lake support schema enforcement vs schema evolution ?

Asked in Publicis Sapient, Tech Mahindra (Created By – Praveen Patel)

Delta Lake supports schema enforcement by preventing writes that don't match the table's existing schema, ensuring data consistency. Schema evolution, on the other hand, allows automatic updates to the table schema when new columns are added. Together, they offer flexibility with control - enforcing structure while adapting to changing data needs during append or merge operations.

Q.88). How do you implement time travel queries in Delta?

Asked in Hexaware, Publicis Sapient (Created By – Praveen Patel)

Time travel queries in Delta can be implemented by using the VERSION AS OF or TIMESTAMP AS OF options in SQL. This allows you to query a previous version of the data for auditing, debugging.

```
SELECT * FROM table_name VERSION AS OF 5
SELECT * FROM table_name TIMESTAMP AS OF '2025-08-01T00:00:00'
```

Delta automatically retains historical versions based on the configured retention period.

Q.89). What causes OPTIMIZE + ZORDER to fail or produce no stats file? Asked in Publicis Sapient (Created By – Praveen Patel)

OPTIMIZE + ZORDER can fail to produce a stats file when the target Delta table is too small to require optimization, or when data is already optimally organized. It may also occur if ZORDER columns have low cardinality or missing values, reducing its effectiveness. Additionally, if no new data files were rewritten during OPTIMIZE, Databricks skips stats generation as there's no performance benefit to capture.

Q.90).How do you use API in Azure Databricks to bring Data ?

Asked in EY, Tech Mahindra, and Capgemini (Created By Praveen Patel)

To bring data via an API in Azure Databricks, use Python's requests library to make REST API calls from a notebook. Parse the JSON/XML response and load it into a Spark DataFrame using spark.read.json() or spark.createDataFrame(). This method is commonly used to integrate third-party services or real-time data into Databricks workflows for processing and analysis.

Created By – Praveen Patel

Q.91). How do you mask credentials in Azure Databricks ?

Asked in Infosys, Tech Mahindra, Deloitte (Created By Praveen Patel)

To mask credentials in Azure Databricks, use Databricks secrets stored in a Databricks-backed or Azure Key Vault-backed scope. Create secrets using the Databricks CLI or UI and access them securely in notebooks using dbutils.secrets.get(scope, key). This approach keeps sensitive information like passwords and tokens hidden from logs, notebooks, and version control, ensuring secure handling of credentials.

Q.92). What is the main advantage of Delta Lake ?

Asked in Hexaware (Created By Praveen Patel)

The main advantage of Delta Lake is its ability to bring ACID transactions to Apache Spark and big data workloads. It ensures data reliability, schema enforcement, and time travel for historical data versioning. This enables consistent reads and writes, making it ideal for building robust, production-grade data pipelines on data lakes without compromising on performance, scalability, or data quality.

Created By – Praveen Patel

Q.93). What is the purpose of delta lake, and how does it enhance data reliability in databricks ?

Asked in Jio, Hexaware (Created By Praveen Patel)

Delta Lake is an open-source storage layer that brings ACID transactions to Apache Spark and big data workloads. In Databricks, it enhances data reliability by ensuring consistency through atomic writes, scalable metadata handling, and data versioning via time travel. This prevents data corruption and enables rollback capabilities, making pipelines more robust and simplifying debugging, auditing, and recovery in production environments.

Q.94).What are the different data sources supported in Databricks ?

Asked in HCL, Hexaware, EY (Created By Praveen Patel)

Databricks supports a wide range of data sources, including cloud storage like Azure Data Lake Gen2, Amazon S3, and Google Cloud Storage. It integrates with Delta Lake, Apache Hive, JDBC, Kafka, MongoDB, and REST APIs. Databricks also connects to relational databases like SQL Server, PostgreSQL, MySQL, and supports external BI tools, enabling seamless data ingestion, transformation, and advanced analytics at scale.

Q.95). Explain Databricks lakehouse architecture and it's benefit ?

Asked in HCL, TCS, EY, PWC (Created By Praveen Patel)

Databricks Lakehouse architecture unifies data warehousing and AI workloads on a single platform using Delta Lake. It combines the reliability and performance of data warehouses with the scalability and cost-efficiency of data lakes. Key benefits include simplified data management, support for BI and ML on the same data, reduced data silos, and faster insights with open standards and governance.

Created By – Praveen Patel

Q.96).How does medallion architecture work ?

Asked in KPMG, Deloitte, Wipro (Created By Praveen Patel)

The Medallion Architecture in Databricks organizes data into three layers - Bronze, Silver, and Gold. Bronze stores raw data, Silver contains cleaned and enriched data, and Gold provides aggregated, business-ready insights. This layered design enables incremental processing, ensures data quality, and simplifies governance. It supports scalable, reliable pipelines and improves performance for analytics and machine learning across structured and semi-structured datasets.

Q.97). What is the difference between OPTIMIZE, Z-Order By, and VACCUM in Delta Lake ?

Asked in Accenture, PWC, Capgemini (Created By Praveen Patel)

OPTIMIZE compacts small files into larger ones to improve query performance. Z-ORDER BY organizes data within files based on specified columns to speed up filtering. VACUUM removes obsolete files from storage to free space. Use OPTIMIZE and Z-ORDER for performance tuning; VACUUM for storage cleanup and managing Delta Lake's retention.

Q.98).What are the benefits of using unity catalog in Databricks ?

Asked in KPMG, EY, IBM (Created By Praveen Patel)

Unity Catalog in Databricks provides centralized governance for all data and AI assets, enabling fine-grained access control, data lineage, and auditing across workspaces. It simplifies data management with unified security, making it easier to enforce compliance. By supporting multi-cloud environments and enabling discoverability through metadata, Unity Catalog enhances collaboration, data reliability, and operational efficiency across teams and projects.

Created By – Praveen Patel

Q.99). What is delta sharing ?

Asked in IBM, HCL (Created By Praveen Patel)

Delta Sharing is an open protocol for secure data sharing across organizations, regardless of platform or cloud. It enables users to share live, real-time data from Delta Lake tables with other users or systems without duplication. Consumers can access shared data using tools like Pandas, Apache Spark, or BI tools, ensuring consistency and governance.

Q.100). Difference between merge, update, and overwrite modes in delta lake ?

Asked in Infosys, Tiger Analytics, IBM, Tech Mahindra, Databricks

In Delta Lake, merge updates, inserts, or deletes records based on a condition. Update modifies existing rows that meet a condition without inserting new ones. Overwrite replaces the entire target table or partition with new data. Use merge for upserts, update for conditional changes, and overwrite to fully refresh data.

Q.101).How do you call one notebook from another notebook in databricks ?

Asked in Walmart (Created By – Praveen Patel)

You can call one notebook from another in Databricks using the dbutils.notebook.run() command. It allows you to execute a child notebook by specifying its path and parameters if needed.

Example-dbutils.notebook.run("/path/to/notebook",timeout_seconds, {"param":"value"}). This is useful for modularizing code and building reusable, maintainable pipelines across multiple notebooks.

Created By – Praveen Patel

Q.102). What are the different types of Databricks Runtime ?

Asked in Walmart (Created By – Praveen Patel)

Databricks Runtime has several types, including the standard Databricks Runtime, Databricks Runtime for Machine Learning (includes ML libraries), Databricks Runtime for Genomics, and Databricks Runtime with Photon (for faster performance). Each runtime is optimized for specific workloads like data engineering, AI/ML, or high-speed analytics, allowing users to choose the best fit for their project needs.

Q.103).How does Delta Lake handle ACID transactions in a data lake ?

Asked in Mastercard (Created By – Praveen Patel)

Delta Lake ensures ACID transactions in a data lake using a transaction log called Delta Log. It tracks all changes to data as atomic commits. Each transaction creates a new version of the table, ensuring atomicity, consistency, isolation, and durability. This allows concurrent reads/writes while preventing data corruption and maintaining reliable data versioning and recovery.

Q.104). How would you use Databricks to perform incremental loading from bronze to silver layer ?

Asked in Mastercard (Created By – Praveen Patel)

To perform incremental loading from bronze to silver in Databricks, use merge (upsert) logic with Delta Lake's MERGE INTO command. Track the latest watermark column like last_updated to filter only new or changed records. This ensures only delta data is loaded. Optionally, maintain a checkpoint or metadata table to manage state and handle late-arriving data efficiently for robust and scalable processing.

Q.105). How would you handle schema evolution in Azure Databricks?

Asked in Mastercard (Created By – Praveen Patel)

In Azure Databricks, I handle schema evolution using Auto Merge with Delta Lake. When using merge or write operations, I enable mergeSchema or overwriteSchema options. This allows new columns to be automatically added to the target schema without manual intervention. It ensures data compatibility across evolving datasets while maintaining ACID transactions and avoids job failures due to schema mismatch.

Q.106). Explain different types of secret scope in Azure Databricks ?

Asked in Hexaware, TCS (Created By – Praveen Patel)

In Azure Databricks, secret scopes securely store credentials like keys or passwords. There are two types - **Databricks-backed scopes**, where secrets are managed directly within Databricks and encrypted at rest, and **Azure Key Vault–backed scopes**, which integrate with Azure Key Vault to centrally manage secrets. Key Vault–backed scopes are recommended for enterprise security, Databricks-backed scopes are useful for notebooks or jobs requiring quick access to secrets without external dependencies

Q.107).Explain Different Transformation used in Azure Databricks ?

Asked in Hexaware, Cognizant (Created By – Praveen Patel)

Transformations in Azure Databricks are operations on DataFrames or RDDs that produce new datasets without modifying the original data. Common transformations include map (applies a function to each element), filter (selects rows based on a condition), join (combines datasets), groupBy/agg (aggregates data), withColumn (adds or modifies columns), and distinct (removes duplicates).

Q.108).What is partition in Azure Databricks ?

Asked in Hexaware, Wipro (Created By – Praveen Patel)

A partition in Azure Databricks is a logical division of data in a DataFrame or Delta table, stored across multiple files or nodes. It helps organize and process large datasets efficiently by distributing computation. Partitioning improves query performance, enables parallel processing, and reduces read/write times by allowing operations to target only relevant data segments instead of scanning the entire dataset.

Q.109). How to store data in medallion architecture different layer (Bronze, Silver, Gold) ?

Asked in TCS, Hexaware, Infosys (Created By – Praveen Patel)

In Databricks, data is organized into Bronze, Silver, and Gold layers using Delta Lake. Bronze stores raw ingested data with minimal processing. Silver contains cleaned, transformed, and deduplicated data for analytics. Gold holds aggregated, business-ready data for reporting or ML. This layered approach ensures data quality, scalability, and reusability while supporting incremental processing and governance.

Q.110). What are the common source and destination format used in azure databricks ?

Asked in Hexaware (Created By – Praveen Patel)

In Azure Databricks, common source formats include CSV, JSON, Avro, Parquet, ORC, Delta, and external sources like Azure Data Lake Gen2, Blob Storage, and SQL databases. Destination formats typically are Parquet, Delta Lake, and ORC due to their efficiency and ACID support.

Q.111). What are the different components of Azure Databricks ?

Asked in Hexaware (Created By – Praveen Patel)

Azure Databricks has four main components – workspace (environment for notebooks and projects), Clusters (compute resources for running jobs), Libraries (packages to extend functionality), and Jobs (to schedule and automate workflows). Additionally, Delta Lake ensures reliable data storage and management. Together, these components enable scalable data engineering, machine learning, and analytics on the cloud.

Q.112). How to improve query performance in slow running spark jobs ?

Asked in Hexaware (Created By – Praveen Patel)

To improve slow Spark jobs, start by optimizing data formats using Delta Lake or Parquet with proper partitioning and caching frequently used data. Use predicate pushdown to minimize scanned data, and tune shuffle partitions to avoid skew. Enable Adaptive Query Execution (AQE) for dynamic optimization. Avoid wide transformations when possible, leverage broadcast joins for smaller tables, and monitor with the Spark UI to identify bottlenecks. This ensures faster, efficient query performance.

Q.113). What is autoscaling in azure databricks ?

Asked in Hexaware (Created By – Praveen Patel)

Autoscaling in Azure Databricks is the feature that automatically adjusts the number of worker nodes in a cluster based on workload demands. When workloads increase, more nodes are added to maintain performance, and when demand drops, nodes are removed to reduce costs. This ensures clusters run efficiently without manual intervention, offering a balance between performance, resource utilization, and cost optimization for both batch and streaming workloads.

Q.114). What are the transformations you are performing on top of the bronze data for data cleaning?

Asked in Fractal Analytics (Created By – Praveen Patel)

On top of bronze data, I perform key cleaning transformations like removing duplicates, handling missing or null values, standardizing column formats (date, time, string casing), filtering out corrupted or irrelevant records, and applying schema validation. I also enrich data by trimming spaces, normalizing values, and casting datatypes properly. These steps make the data reliable and ready to move into the silver layer for analytics.

Q.115. Write the code to get data from bronze and give sample code for data cleaning ?

Asked in Fractal Analytics (Created By – Praveen Patel)

```
# Load Bronze Data
bronze_df = spark.read.format("delta").load("/mnt/data/bronze")

# Sample Cleaning Transformations
clean_df = (bronze_df.dropDuplicates().filter("columnX IS NOT NULL")
```

```
.withColumn("date_col", to_date(col("date_col"), "yyyy-MM-dd"))
        .withColumn("trimmed_name", trim(col("name"))))
clean_df.write.format("delta").mode("overwrite").save("/mnt/data/silver")
```

This code loads bronze data, removes duplicates, handles nulls, formats dates, and saves it into silver.

Q.116). What are the challenges you are facing while performing optimization?

Asked in Fractal Analytics (Created By – Praveen Patel)

Some challenges include dealing with skewed data, which creates performance bottlenecks during joins, managing large shuffle operations, and tuning Spark configurations like spark.sql.shuffle.partitions. Sometimes jobs fail due to memory limits or inefficient queries. Another challenge is deciding the right partitioning strategy for Delta tables, as over-partitioning increases metadata load while under-partitioning slows queries. Balancing these trade-offs is a common challenge in real projects.

Q.117).What are the issues you face in Databricks?

Asked in Fractal Analytics (Created By – Praveen Patel)

In Databricks, I've faced issues like cluster startup delays, job failures due to library conflicts, unexpected costs from inefficient clusters, and version compatibility problems with certain Spark or Python libraries. Sometimes UI lags occur when handling very large data. Also, managing concurrent notebook execution and troubleshooting job failures due to data quality or schema drift are common issues encountered in production environments.

Q.118). What is the use of workflow in Databricks and have you ever used it ? Asked in Fractal Analytics (Created By – Praveen Patel)

Databricks Workflows are used for orchestrating jobs, chaining multiple notebooks or tasks, and scheduling them with dependencies. They help automate ETL pipelines, integrate with data quality checks, and manage retries or alerts in case of failures. Yes, I've used workflows to schedule batch pipelines, trigger downstream tasks, and integrate with monitoring. It helps replace external schedulers when orchestration needs are fully within Databricks.

Q.119). Why do we still use ADF to call Databricks notebooks although we can orchestrate it in Databricks itself Asked in Fractal Analytics

We still use ADF to call Databricks notebooks because ADF provides enterprise-level orchestration, monitoring, and integration with various Azure services. Many organizations have ADF as a central data orchestration tool, connecting Databricks with pipelines involving SQL Database, Data Lake, Synapse, etc. ADF also offers better lineage tracking, trigger-based executions, and centralized control, making it more suitable for hybrid data platforms compared to Databricks-only workflows.

Q.120). How is table stored in backened in delta lake ?

Asked in Impetus, IBM, Accenture (Created By – Praveen Patel)

In Delta Lake, a table is stored in the backend as Parquet files on a data lake, along with a transaction log (_delta_log folder). The Parquet files hold the actual data, while the transaction log keeps track of every change like inserts, updates, and deletes. This log ensures ACID transactions, versioning, and time travel, making Delta tables reliable, consistent, and easy to query for analytics or machine learning.

Q.121). How does delta table handle ACID Properties ?

Asked in Impetus, Wipro (Created By – Praveen Patel)

Delta tables handle ACID properties (Atomicity, Consistency, Isolation, Durability) by using a transaction log called Delta Log. Every change is recorded as a version, ensuring rollback if needed (atomicity). Schema enforcement and constraints maintain consistency. Optimistic concurrency control avoids conflicts, supporting isolation. Finally, changes are stored reliably in storage like ADLS or S3, ensuring durability. This makes Delta highly reliable for enterprise-grade data pipelines.

Q.122).How do you manage spark dataframe in databricks ?

Asked in Impetus, Wipro (Created By – Praveen Patel)

In Databricks, Spark DataFrames are managed by leveraging PySpark APIs to load, transform, and save data efficiently. You can read data from multiple sources like ADLS, Delta, or SQL tables, then apply transformations such as filtering, joins, and aggregations. Databricks notebooks support dynamic exploration, while Delta Lake ensures schema enforcement and ACID transactions. Finally, DataFrames can be cached, optimized, and written back for reliable and scalable data engineering workflows.

Q.123). Write a code to read CSV file in databricks using Pyspark ?

Asked in EY, Impetus, EPAM (Created By – Praveen Patel)

To read a CSV file in Databricks using PySpark, you can use the spark.read.csv() method. You need to specify the file path and options like header=True to read column names and inferSchema=True to automatically detect data types. Example -

```
df = spark.read.csv("/mnt/data/filename.csv", header=True, inferSchema=True)
df.show()
```

Q.124). Why do we need Delta live tables when we already have ETL pipeline ? Asked in LTIMindtree, Tech Mahindra

Delta Live Tables (DLT) is needed even when we already have ETL pipelines because it simplifies building and managing reliable data pipelines. Traditional ETL requires manual coding, monitoring, and handling failures. DLT automates pipeline orchestration, ensures data quality with built-in validation rules, supports incremental processing, and maintains data lineage. This makes pipelines more efficient, scalable, and easier to manage compared to manually created ETL pipelines.

Q.125).How do you force autoloader to reprocess a file that is already recorded in checkpoint ? Asked in LTIMindtree, Deloitte

You can force Auto Loader to reprocess a file that is already recorded in a checkpoint by using the option cloudFiles.includeExistingFiles set to true or by changing the input path or schema location so Auto Loader treats it as a new stream. Another way is to delete or reset the checkpoint directory, which makes Auto Loader forget processed files and re-ingest them as fresh inputs.

Q.126). If a 1 TB file is ingested by Databricks Autoloader and the load fails halfway (at 500 GB), how does Autoloader handle recovery and ensure data consistency ?

Asked in LTIMindtree (Created By – Praveen Patel)

If a 1 TB file ingestion fails halfway using Databricks Autoloader, it does not restart the whole process. Autoloader uses checkpointing and file notification services to track progress. When the pipeline restarts, it resumes from the point of failure instead of reloading already processed data. This ensures consistency, avoids duplicate records, and saves time, making ingestion of large files reliable and fault-tolerant in production environments.

Q.127). What are the different optimization techniques available are there in delta tables ?

Asked in KPMG, EY, Tiger Analytics (Created By – Praveen Patel)

Delta tables offer several optimization techniques to improve query performance and data management. Common methods include OPTIMIZE command to compact small files into larger ones, Z-ORDER clustering for faster filtering and range queries, and VACUUM to clean up old snapshots and unused files. Additionally, techniques like data skipping, partitioning, and caching help speed up queries. These optimizations ensure reduced latency, efficient storage, and better scalability in large-scale data workloads.

Q.128). How to pass parameters from Azure Data Factory to Databricks Notebook ?

Asked in Infosys , Inxite out (Created By Praveen Patel)

You can pass parameters from Azure Data Factory (ADF) to a Databricks notebook by defining base parameters in the Databricks activity within the ADF pipeline. In ADF, open the Databricks Notebook activity → Settings tab → Base parameters, and specify key-value pairs (e.g., param1: @pipeline().parameters.param1). Inside the Databricks notebook, use dbutils.widgets.get("param1") to retrieve the value. This enables dynamic, parameterized notebook executions directly from ADF for reusable and scalable pipeline designs.

Q. 129). Explain delta lake time travel feature with an example ?

Asked in Genpect , Capgemini (Created By Praveen Patel)

Delta Lake's Time Travel feature allows you to query or restore data from previous versions of a Delta table. It maintains a transaction log

(_delta_log) that records all changes, enabling you to access historical data for auditing, debugging, or reproducing experiments.

Example -

Suppose you updated a Delta table yesterday but now want to view the old data. You can query it using:

SELECT * **FROM** delta.`/mnt/data/sales` VERSION **AS OF** 5;

SELECT * FROM delta.`/mnt/data/sales` TIMESTAMP AS OF '2025-11-10';

Q.130). Explain the difference between delta table and parquet table ?

Asked in Genpect , TCS (Created By Praveen Patel)

A Parquet table is a storage format that saves data in a columnar, compressed, and read-optimized structure. It doesn't store transaction history or support ACID operations. In contrast, a Delta table is built on top of Parquet but adds ACID transactions, versioning, time travel, and schema evolution. Delta tables maintain a transaction log (_delta_log) that tracks every change, making data reliable, consistent, and easy to update or delete. In short, Parquet is great for static analytics, while Delta is ideal for real-time, incremental, and production-grade data pipelines.

Q.131). What is the syntax to write delta tables ?

Asked in Genpect, Tiger Analytics (Created By Praveen Patel)

To write data into a Delta table, you can use the write method with the format("delta") option in PySpark

df.write.format("delta").mode("overwrite").save("/mnt/delta/sales")

Or to write into a managed table:

df.write.format("delta").mode("append").saveAsTable("sales_delta")

Here, mode defines how data is written —

- overwrite: replaces existing data
- append: adds new records
- ignore: skips if table exists
- errorifexists: throws an error if table already exists

This syntax ensures transactional consistency and schema enforcement in Delta Lake.

Q.132). Difference between Data Lake, DataLakehouse and Data Warehouse ?

Asked in Genpect, Tiger Analytics (Created By Praveen Patel)

A Data Lake stores raw, unprocessed data from multiple sources in its native format, suitable for big data and machine learning workloads. A Data Warehouse stores structured, processed, and curated data optimized for analytics and reporting using predefined schemas. A Data Lakehouse combines both — it supports raw data storage like a Data Lake but also enforces schema, ACID transactions, and performance optimizations like a Data Warehouse, enabling unified storage for both analytical (BI) and data science workloads efficiently in one platform.

Q.133). Your spark job on Azure Databricks runs slow what steps will you take to optimize it ?

Asked in Capgemini, Deloitte (Created By Praveen Patel)

To optimize a slow Spark job in Azure Databricks, I first review the physical plan using Spark UI to identify skew, shuffle issues, or expensive stages. I enable Adaptive Query Execution (AQE) to optimize joins and handle skew automatically. I also adjust cluster sizing, ensuring proper worker memory and using Autoscaling. I optimize input data by converting to Delta format, using Z-Ordering, and pruning unnecessary columns. If data skew exists, I apply salting or broadcast joins where appropriate. Finally, I cache reused DataFrames and tune shuffle partitions for balanced parallelism

Q.134). Write a pyspark code for reading text for position based text file storing it in delta format ?

Asked in Tredence, IBM, Accenture, Capgemini (Created By Praveen Patel)

To read a position-based text file in PySpark, you extract fixed-width columns using substring, then write the processed DataFrame to Delta. Position-based files don't have delimiters, so each field is extracted by character index. After defining a schema mapping start/end positions, you convert the raw text into structured columns and store it in Delta Lake for ACID reliability and fast reads.

```
df_raw = spark.read.text("/mnt/raw/fixedwidth.txt")

df = df_raw.select(
    F.substring("value", 1, 5).alias("id"),
    F.substring("value", 6, 20).alias("name"),
    F.substring("value", 26, 10).alias("amount")
)

df.write.format("delta").mode("overwrite").save("/mnt/delta/fixedwidth")
```

Q.135). How does delta lake provide ACID transaction in spark and why it is essential for certain cases ?

Asked in Tredence (Created By Praveen Patel)

Delta Lake provides ACID transactions in Spark through its transaction log (**delta_log**), which records every write as an atomic commit with versioning. Each commit includes actions like AddFile, RemoveFile, and schema updates, ensuring atomicity, consistency, isolation, and durability on cloud storage. Spark writers use optimistic concurrency control (OCC) to detect conflicting writes and retry safely. This is essential for pipelines requiring exact data, such as financial reporting, GDPR corrections, or incremental upserts.

For Example

```
from delta.tables import DeltaTable

deltaTable = DeltaTable.forPath(spark, path)

deltaTable.update("id = 10", {"status": "'active'"})
```

Delta guarantees that this update is isolated, consistent, and fully durable.

Q.136). How do you create a pipeline in databricks ?

Asked in TCS, Tredence, EXL (Created By Praveen Patel)

To create a pipeline in Databricks, you use Delta Live Tables (DLT). First, open the Databricks workspace → go to Workflows → select Delta Live Tables → click Create Pipeline. Then choose your notebook, SQL/Python language, and set the storage location, target schema, and cluster mode (continuous or triggered). Inside the notebook, define tables using LIVE and STREAMING LIVE with expectations for data quality. After configuration, click Start to deploy the pipeline.

Databricks automatically handles orchestration, dependency management, schema evolution, and monitoring through the DLT event logs and pipeline UI.

Q.137). Suppose your spark job is taking longer than expected What steps would you take to debug and optimize it ?

Asked in Accenture, Wipro, HCLTech (Created By Praveen Patel)

To debug and optimize a slow Spark job, I first review the Spark UI to identify the slowest stages, long-running tasks, or skewed partitions. Then I check for data skew by analyzing shuffle read/write sizes. If skew exists, I apply techniques like salting, repartitioning, or using broadcast joins for smaller tables. I also validate cluster sizing—often upgrading to autoscaling or using a more suitable node type improves performance. Next, I optimize transformations by avoiding unnecessary shuffles, caching only when reused, and using Delta optimizations like Z-Ordering or Optimize. Finally, I review code logic to eliminate wide transformations and improve parallelism.

Q.138). How do you optimize data pipeline in databricks ?

Asked in EXL, Mastercard, HCLTech, Oracle (Created By Praveen Patel)

To optimize data pipelines in Databricks, I start by designing Delta Lake–based pipelines to leverage efficient storage, ACID transactions, and file compaction. I use Auto Loader for scalable ingestion and enable schema evolution to avoid failures. For transformations, I optimize Spark jobs by using partition pruning, broadcast joins, caching, and adaptive query execution (AQE) to reduce shuffle. I also keep file sizes optimized using OPTIMIZE with Z-ORDER for faster reads. Cluster optimization includes using the right cluster size, autoscaling, and Photon for performance. Finally, I monitor pipeline performance with Ganglia, Spark UI, and job run metrics to continuously tune bottlenecks.

Q.139). What is lakehouse federation , and how does it work ?

Asked in Cognizant, EXL, HCLTech (Created By Praveen Patel)

Lakehouse Federation in Databricks allows you to **query and manage data across multiple external systems**—such as SQL databases, NoSQL stores, cloud warehouses, and data lakes—**without copying or moving the data**. It works by creating **federated connections** to external sources and exposing them as **Unity Catalog tables**. Databricks then uses its **query engine** to push down filters, aggregations, and joins to the underlying system for optimal performance. This lets teams achieve **a single governance layer, unified metadata, and end-to-end analytics** while keeping datasets in their original location.

Q.140). How do you improve query performance in pyspark ?

Asked in TCS, Wipro, HCLTech (Created By Praveen Patel)

To improve query performance in PySpark, I focus on optimizing both the execution plan and data handling. I use DataFrame APIs instead of RDDs, enable predicate pushdown, and ensure column pruning by selecting only required columns. I reduce data shuffling using repartition() carefully or prefer coalesce() when decreasing partitions. I cache reusable DataFrames with cache() or persist(), and optimize joins using broadcast joins for small tables. I also write data in efficient formats like Delta or Parquet and review the Spark UI to identify skew, shuffle hotspots, and optimize accordingly.

Q.141). How do you calculate running total in pyspark and SQL ?

Asked in IBM, EXL, HCLTech (Created By Praveen Patel)

In PySpark, a running total can be calculated using the Window function along with sum() over an ordered partition. Example:

```
from pyspark.sql.window import Window

from pyspark.sql.functions import sum

window_spec =

Window.orderBy("date_column").rowsBetween(Window.unbounde
dPreceding, 0)

df = df.withColumn("running_total",
sum("amount").over(window_spec))
```

In SQL, you can use the SUM() function as a window function -

```
SELECT date_column, amount,

SUM(amount) OVER (ORDER BY date_column ROWS BETWEEN
UNBOUNDED PRECEDING AND CURRENT ROW) AS running_total

FROM table_name;
```

Q.142). How would you handle large scale joins efficiently in Pyspark ?

Asked in HCLTech, Deloitte, EY (Created By Praveen Patel)

To handle large-scale joins efficiently in PySpark, I would first analyze the size of datasets. For joining a large dataset with a small one, I would use broadcast joins to avoid full shuffle. For two large datasets, I would repartition or bucket data on join keys to minimize shuffling. Caching intermediate results can improve performance if reused multiple times. I would also filter unnecessary columns and rows before the join, and use sort-merge join or broadcast hint wisely. Monitoring Spark UI helps identify skewed partitions and optimize join strategies.

Q.143). What are the best practices for partitioning and bucketing in Pyspark ?

Asked in HCLTech, Wipro (Created By Praveen Patel)

In PySpark, partitioning and bucketing optimize performance and manage large datasets efficiently. Best practices for partitioning include choosing high-cardinality columns to avoid skew, keeping partition numbers balanced (not too small or too large), and filtering frequently accessed columns. For bucketing, select columns with high cardinality and consistent distribution, and use a reasonable number of buckets to optimize join and aggregation performance. Combine partitioning and bucketing when necessary for large datasets. Always monitor query performance using Spark UI and adjust partition/bucket strategy based on data size and access patterns to reduce shuffle and improve execution speed.

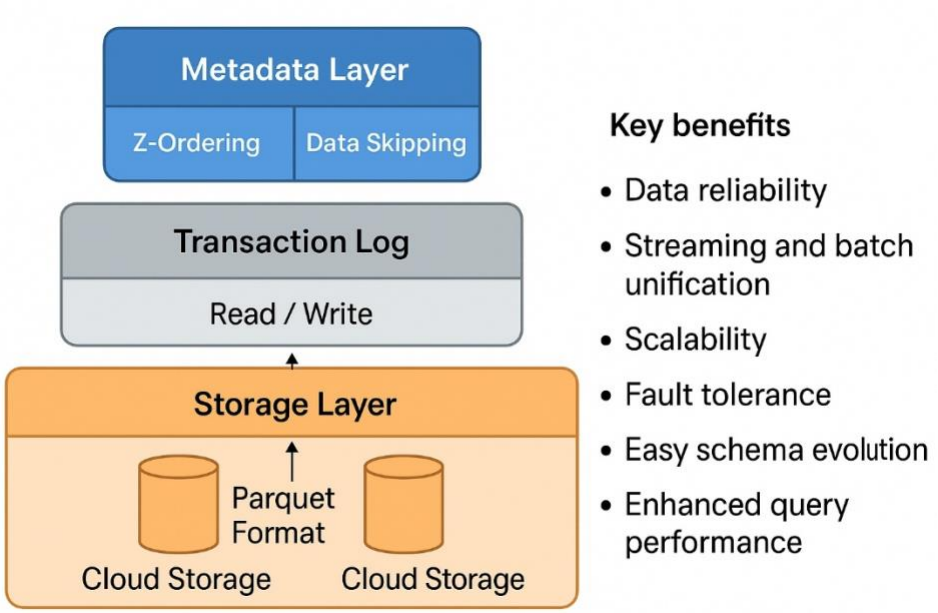
Q.144). Describe challenging data pipeline issue you have encountered and how you resolved it ?

Asked in Accenture (Created By – Praveen Patel)

During a project at Accenture, I faced a challenging data pipeline issue where incremental loads from multiple source systems were causing duplicate records in the Delta Lake table. After analyzing the pipeline in Azure Databricks, I identified that the issue was due to late-arriving data and improper merge logic in the upsert operation. I resolved it by implementing Delta Lake's **MERGE INTO** with proper **update** and **insert** conditions, combined with watermarking to handle late data. This ensured deduplication, maintained data consistency, and optimized pipeline performance without reprocessing the full dataset.

Q.145). Explain the architecture of delta lake in databricks and its Key benefits ? Asked in TCS (Created By – Praveen Patel)

Delta Lake in Databricks is built on a **layered architecture** that combines **storage, transaction log, and metadata management**. The **storage layer** resides on cloud storage (like ADLS Gen2 or S3) and holds data in **Parquet format**. The **transaction log layer** (Delta Log) maintains **ACID transactions**, enabling reliable **read/write operations**, **time travel**, and **schema enforcement**. Delta Lake also provides **optimized data access** through **Z-ordering and data skipping**. Key benefits include **data reliability**, **streaming and batch unification**, **scalability**, **fault tolerance**, **easy schema evolution**, and **enhanced query performance**, making it ideal for modern data lakes.

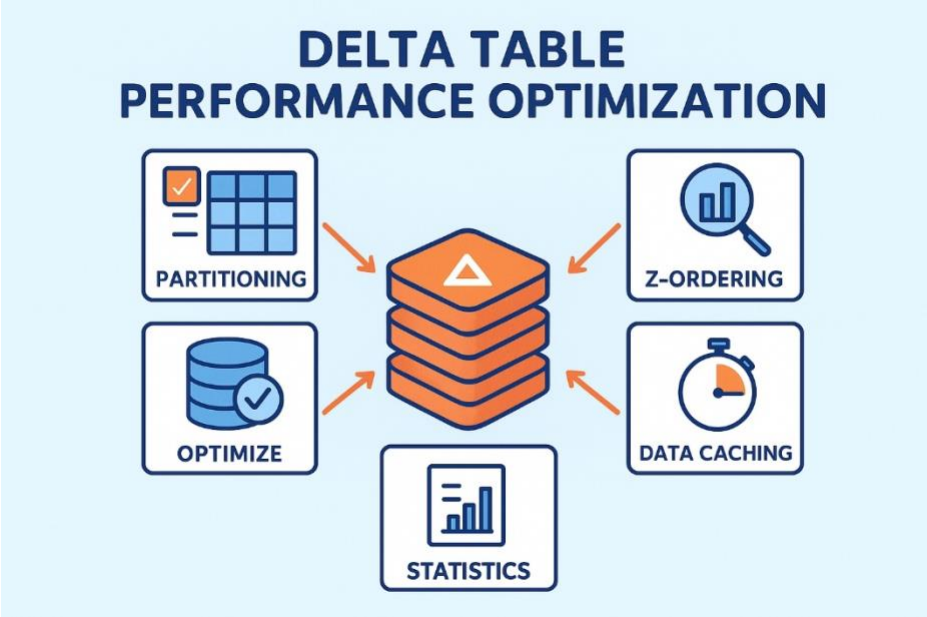


Q.146). How do you optimize databricks delta table performance ?

Asked in Capgemini (Created By – Praveen Patel)

To optimize Databricks Delta table performance, start by **properly partitioning** the table based on query patterns to reduce scan times. Use **Z-Ordering** on frequently filtered columns to colocate related data and speed up queries. Regularly **OPTIMIZE** the table to compact small

files into larger ones, reducing file I/O overhead. Enable **data caching** for hot tables to improve read performance. Consider **using Delta caching** on Databricks clusters and leverage **auto-compaction** for continuous optimization. Additionally, maintain **statistics** for tables to help the query optimizer make efficient execution plans.



Q.147).Describe the process of implementing auto scaling in databricks Cluster ?

Asked in Tiger Analytics (Created By – Praveen Patel)

Auto-scaling in Databricks clusters allows the cluster to dynamically adjust the number of worker nodes based on workload demand. To implement it, first, create or edit a cluster and enable “Enable Autoscaling”. Specify the minimum and maximum number of worker nodes. Databricks will automatically add nodes when job demand increases and remove idle nodes when demand decreases, optimizing cost and performance. Auto-scaling works with both Standard and High

Concurrency clusters and ensures efficient resource utilization without manual intervention, making workloads like ETL, streaming, and ML training scalable and cost-effective.

Q.148). What is the difference between Spark SQL and Pyspark ?

Asked in Tiger Celebal (Created By – Praveen Patel)

Spark SQL and PySpark serve different purposes in the Apache Spark ecosystem. Spark SQL is a Spark module used to perform structured data processing using SQL queries or the DataFrame API, allowing users to query data with standard SQL syntax. PySpark, on the other hand, is the Python API for Apache Spark, enabling developers to write Spark applications using Python. While Spark SQL focuses on querying structured data efficiently, PySpark provides broader capabilities, including transformations, actions, machine learning, and graph processing. In practice, Spark SQL can be called within PySpark using DataFrames for optimized execution.

How this guide will help you to crack Databricks, and Azure Data Engineering interview

- ✔ Covers theory and practical scenarios asked by top MNCs.
- ✔ Prepares you for frequently repeated & high-weight topics.
- ✔ Saves hours of preparation time with a structured, focused guide.
- ✔ Boosts readiness for technical rounds, especially in product and consulting firms.
- ✔ Helps professionals with 3 – 10 years of experience prepare confidently for mid-to-senior level roles

Creation Journey of This Kit

This kit represents weeks of dedication and effort. To ensure its value:

- I explored multiple trusted sources to gather authentic interview information.
- I had one-on-one discussions with professionals who had recently moved into new roles, capturing their first-hand experiences.
- In certain cases, I even offered incentives to individuals for sharing their interview journeys, ensuring accuracy and reliability.

Important Usage Terms

- This document is strictly meant for **individual learning**.
- Sharing it on public platforms such as LinkedIn or elsewhere is not allowed.
- It must not be repurposed, resold, or distributed under someone else's name on Topmate or any other site.

- Every page of this kit is the result of personal effort and should not be duplicated without prior consent.

Consequences of Misuse

Unauthorized selling or redistribution will lead to strict actions, such as:

- Posting about the violation on LinkedIn.
- Highlighting the individual involved, including their profile and workplace.
- Attaching clear proof of the misuse.

This can cause serious harm to one’s professional reputation in the data engineering community.