**Fractal Data Engineer Interview Guide – Experienced 3+**

**Round 1:**

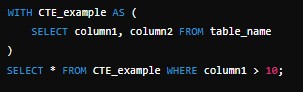
1. **CTE vs Temp Table – Differences and Use Cases: CTE (Common Table Expression):**

Temporary result set defined within a query using the WITH clause.

Exists only for the duration of the query.

Cannot be indexed, so it's ideal for complex, reusable queries.

Example:



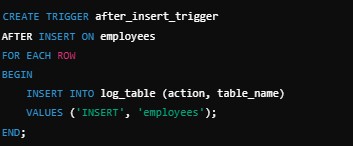
**Temp Table:**

Physically stored in the tempdb database.

Can be used across multiple queries and stored for a session.

Supports indexing, making it suitable for larger datasets.

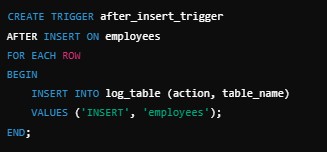
Example:



2. **Triggers in SQL – Examples and Scenarios for Use:**

**Triggers** automatically execute actions in response to specific events on a database table (INSERT, UPDATE, DELETE).

Example:



**Use Cases:**

Audit logging.

Automatically updating other tables when data is changed.

Preventing invalid data from being inserted into tables.

3. **Coalesce vs ISNULL – Differences in SQL:**

**Coalesce**: Returns the first non-null expression among the given list of values.

Example: SELECT COALESCE(column1, column2, 'Default') FROM table\_name;

**ISNULL**: Replaces NULL with a specified value.

Example: SELECT ISNULL(column1, 'Default') FROM table\_name;

4. **Optimization Techniques – Share Strategies for Query and ETL Optimization: Query Optimization**:

Use indexes for fast access to frequently queried columns.

Avoid SELECT \*; specify only needed columns.

Use JOIN instead of subqueries when possible.

Use EXPLAIN PLAN to analyze query performance.

**ETL Optimization**:

Minimize data transformations.

Use batch processing for large datasets.

Cache intermediate results to avoid recalculating.

Perform data filtering early in the process to reduce dataset size.

5. **Stack vs Unstack – Explain Their Use in Data Transformation: Stack**: Converts columns into rows.

Example:

SELECT stack(2, 'A', 1, 'B', 2) AS (letter, number);

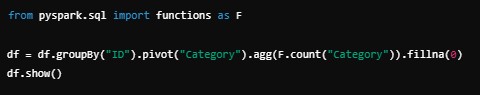
**Unstack**: Converts rows into columns.

Example (in Pandas/other tools):

df.pivot(index='row\_id', columns='category', values='value')

6. **Pivot in PySpark – Example Code and Its Purpose:**

**Purpose**: Transforms unique values from one column into multiple columns.



7. **Data Lake vs Delta Lake – Highlight Differences:**

**Data Lake**: Stores raw, unstructured data, generally in formats like CSV, JSON, or Parquet.

**Delta Lake**: Built on top of Data Lake with ACID transactions, schema enforcement, and time travel for handling data integrity.

8. **How to Help Stakeholders Query Delta Lake Tables – Tools and Approaches: Tools**:

Use Databricks for interactive querying and dashboarding.

Delta Lake enables easy integration with SQL engines, Apache Spark, and BI tools.

**Approaches**:

Provide access via Databricks notebooks or SQL endpoints.

Create views and optimize queries using Z-ordering for faster performance.

9. **How to Get New Records from a Table/File Without a Modified Column – Discuss Approaches Like Hashing or Row Comparison:**

**Hashing**: Generate a hash for each row and store it in a temporary location. Compare the hashes of current and previous data to detect new records.

**Row Comparison**: Compare the current data with previous data using a unique key (e.g., ID) to identify new rows.

10. **Microsoft Fabric – Explain Its Use in Data Integration:**

 **Microsoft Fabric** is an integrated analytics platform that unifies big data and AI workloads, enabling seamless integration of data across various sources (Data Lake, SQL Data Warehouse) and real-time analytics.

**Round 2:**

**1. What is Azure Data Lake Storage (ADLS) Gen2, and how does it differ from Blob**

**Storage?**

Azure Data Lake Storage (ADLS) Gen2 combines the capabilities of a hierarchical file system with Blob Storage, designed for big data analytics.

Differences:

 ADLS Gen2: Offers POSIX-compliant file system with directory and file-based access controls. Ideal for big data workloads and analytics.

 Blob Storage: A flat namespace storage designed for object storage. It lacks the hierarchical structure that ADLS Gen2 provides.

 Performance: ADLS Gen2 is optimized for high-throughput workloads with parallel processing.

Use Case: ADLS Gen2 is used for data lakes, while Blob Storage is more suitable for object storage like backups or static files for web apps.

**2. Explain the purpose and architecture of Azure Synapse Analytics.**

Azure Synapse Analytics is a limitless analytics service combining enterprise data warehousing with big data analytics. It integrates T-SQL-based queries for structured data and Spark for unstructured data.

Key Components:

 Synapse Pipelines: Data integration.

 Synapse SQL Pools: Dedicated and serverless options for querying data.

 Synapse Studio: Unified interface for data professionals.

 Integration with Power BI and Azure ML for reporting and machine learning.

Architecture: Synapse allows seamless querying across data lake files, databases, and other data sources, leveraging distributed computing for parallel processing.

**3. What are Managed Identities in Azure, and how are they used in securing resources?**

Managed Identities simplify Azure service-to-service authentication without the need to manage credentials. They can be used to authenticate to any Azure service that supports Azure AD authentication.

Types:

 System-Assigned: Tied to a single resource; deleted when the resource is deleted.

 User-Assigned: A standalone identity that can be shared across multiple resources. Use Case: For securing a VM accessing Azure Key Vault, the VM can use its managed

identity to fetch secrets without storing passwords in the code.

**4. Explain the difference between Azure Event Hub and Azure Service Bus.**

 Azure Event Hub: Designed for streaming large volumes of data. It’s a data ingestion service used for real-time analytics and event streaming.

 Azure Service Bus: Used for message-based communication between applications. It supports FIFO and dead-letter queues.

Key Difference:

 Event Hub is optimized for telemetry and event stream processing.

 Service Bus focuses on reliable message delivery with features like sessions and transactions.

**5. What are Azure Blueprints, and how are they different from Azure Policies?**

 Azure Blueprints: Allow deploying a repeatable set of Azure resources (like ARM

templates, role assignments, and policies) for environment setup.

 Azure Policies: Enforce rules to control resource configurations (e.g., restrict resource sizes).

Key Difference:

 Blueprints create environments from templates.

 Policies ensure that resources remain compliant with organizational standards.

**6. Explain Azure Databricks architecture and its integration with other Azure services.**

 Azure Databricks: A data analytics platform optimized for Apache Spark with Azure integration.

 Components:

Driver and worker nodes. Distributed Spark environment.

 Integration:

Azure Data Lake Storage (ADLS) and Blob Storage for data storage. Azure Synapse Analytics for data warehousing.

Power BI for visualization.

 Security: Uses Azure AD for identity management and Role-Based Access Control

(RBAC).

**7. Describe the process and use cases of implementing Azure Data Factory pipelines.**

Azure Data Factory (ADF) orchestrates and automates data movement and data transformation using pipelines.

Steps:

1. Create a pipeline: Define activities for data extraction, transformation, and loading

(ETL).

2. Add linked services: Connect to data sources and sinks (e.g., Blob Storage, SQL).

3. Set triggers: Schedule pipeline executions.

Use Case: Automating data ingestion from on-premises databases into Azure Synapse for analysis.

**8. How does Azure Kubernetes Service (AKS) manage scaling and updates for containerized applications?**

Azure Kubernetes Service (AKS) offers managed Kubernetes for deploying containerized apps.

Scaling:

 Horizontal Pod Autoscaler (HPA): Automatically scales pods based on CPU/memory usage.

 Cluster Autoscaler: Adjusts node count in the cluster based on demand.

Rolling Updates: Deploy new versions of containers without downtime. It updates pods incrementally.

Use Case: Deploying a microservices-based application with automatic scaling and zero- downtime updates.

**9. What are Azure Functions Durable Functions, and how do they differ from regular**

**Azure Functions?**

 Azure Functions: Serverless compute service that executes code in response to triggers (e.g., HTTP requests, messages).

 Durable Functions: Extend Azure Functions to support stateful workflows with orchestration patterns.

Differences:

 Regular functions are stateless, while Durable Functions maintain state between executions.

 Durable Functions are used for long-running workflows (e.g., chaining multiple function calls).

Use Case: Orchestrating approval processes where multiple steps depend on external events.

**10. Explain the differences between Azure SQL Database, Azure SQL Managed**

**Instance, and Azure Synapse.**

 Azure SQL Database: A fully managed relational database as a service (DBaaS).

 Azure SQL Managed Instance: Offers full SQL Server features with compatibility for on-premises migration.

 Azure Synapse: An analytics platform combining data warehousing and big data processing.

Differences:

 Azure SQL Database: For cloud-first applications with automatic backups and scaling.

 SQL Managed Instance: Best for migrating legacy SQL Server applications.

 Azure Synapse: Used for massive parallel processing (MPP) and integrating with data lakes.

Use Case Example:

 Use SQL Database for an OLTP system, Managed Instance for an on-premises-to- cloud migration, and Synapse for complex analytics over large datasets.