Day 4: Scenario-Based Questions for Azure Data Factory

1. Scenario: Your company needs to move data from an on-premises SQL Server database to an Azure SQL Database daily. How would you set up this data movement in Azure Data Factory?

- **Answer:** To set up this data movement:
    1. Create a Self-hosted Integration Runtime (IR) to securely connect to the on-premises SQL Server.
    2. Create linked services for both the on-premises SQL Server and Azure SQL Database.
    3. Create datasets for the source and destination tables.
    4. Create a pipeline with a Copy Data activity to move the data.
    5. Schedule the pipeline using a schedule trigger to run daily.

2. Scenario: You need to transform data from a CSV file in Azure Blob Storage and load it into an Azure SQL Database. Describe how you would accomplish this using Azure Data Factory.

- **Answer:** To accomplish this:
    1. Create linked services for Azure Blob Storage and Azure SQL Database.
    2. Create datasets for the source CSV file and the destination SQL table.
    3. Create a pipeline with a Data Flow activity.
    4. In the Data Flow, read the data from the CSV file, apply the required transformations, and write the transformed data to the SQL table.
    5. Trigger the pipeline as needed.

3. Scenario: Your data pipeline fails intermittently due to network issues. How would you handle this in Azure Data Factory?

- **Answer:** To handle intermittent pipeline failures:
    1. Configure retry policies for the affected activities, specifying the maximum retry count and the retry interval.
    2. Use the Set Variable activity to capture and log error details.
    3. Implement conditional activities like If Condition to retry or reroute the process based on error types.

4. Scenario: You need to copy data from multiple CSV files stored in an Azure Data Lake Storage Gen2 account to an Azure SQL Database. How would you configure this in Azure Data Factory?

- **Answer:** To configure this data movement:
    1. Create linked services for Azure Data Lake Storage Gen2 and Azure SQL Database.
    2. Create datasets for the source CSV files and the destination SQL table.
    3. Use a wildcard in the source dataset to specify multiple CSV files.
    4. Create a pipeline with a Copy Data activity to move the data from the CSV files to the SQL table.

5. Scenario: You have a pipeline that must run only after another pipeline completes successfully. How would you implement this in Azure Data Factory?

- **Answer:** To implement this dependency:
    1. Use Execute Pipeline activity to call the dependent pipeline.
    2. Set up an activity dependency to ensure that the subsequent pipeline runs only if the previous pipeline completes successfully.

6. Scenario: Your data transformation logic involves multiple steps, including filtering, aggregation, and joining data from two different sources. How would you implement this in Azure Data Factory?

- **Answer:** To implement complex data transformations:
    1. Create linked services for the data sources.
    2. Create datasets for the input and output data.
    3. Create a pipeline with a Mapping Data Flow activity.
    4. In the Data Flow, add transformations to filter, aggregate, and join the data from the two sources.
    5. Write the transformed data to the desired output destination.

7. Scenario: You need to incrementally load data from an on-premises SQL Server to an Azure SQL Database. Explain how you would achieve this in Azure Data Factory.

- **Answer:** To achieve incremental data loading:
    1. Identify a watermark column (e.g., last modified date) in the source table.
    2. Store the last processed value of the watermark column.
    3. Create a pipeline with a Copy Data activity.
    4. Use a dynamic query in the source dataset to filter data based on the stored watermark value.
    5. Update the watermark value after each successful load.

8. Scenario: You are tasked with integrating data from various formats (CSV, JSON, Parquet) stored in an Azure Data Lake Storage Gen2 into a single Azure SQL Database table. Describe your approach.

- **Answer:** To integrate data from various formats:
    1. Create linked services for Azure Data Lake Storage Gen2 and Azure SQL Database.
    2. Create datasets for each file format and the destination SQL table.
    3. Create a pipeline with multiple Copy Data activities, each handling a different file format.
    4. Use Data Flow activities to apply necessary transformations and merge the data into a single table.

9. Scenario: You need to implement a solution that dynamically chooses the source and destination based on input parameters. How would you configure this in Azure Data Factory?

- **Answer:** To configure dynamic source and destination selection:
    1. Create parameters in the pipeline for the source and destination.

2. Use parameterized linked services and datasets to reference the source and destination based on input parameters.
3. Pass the parameter values at runtime when triggering the pipeline.

10. Scenario: Your company requires a data pipeline to process and analyze streaming data in near real-time. Explain how you would implement this using Azure Data Factory.

- **Answer:** To implement near real-time data processing:
    1. Use Azure Event Hubs or Azure IoT Hub to ingest streaming data.
    2. Set up an Azure Stream Analytics job to process the streaming data and write the output to a data store like Azure Blob Storage or Azure SQL Database.
    3. Use Azure Data Factory to orchestrate the process, periodically running pipelines to load and transform the processed data for further analysis.