Interview Questions and Answers

## **Question 1:**

1. Write a Python function that extracts and processes financial time series data from a Cassandra database. The function should handle missing values, normalize the data, and return a cleaned dataset. Ensure the function adheres to PEP 8 coding standards.

## **Answer:**

To create a Python function that extracts and processes financial time series data from a Cassandra database, handles missing values, normalizes the data, and returns a cleaned dataset, you can follow the steps below. This function will adhere to PEP 8 coding standards.

First, ensure you have the necessary libraries installed:

```bash

pip install cassandra-driver pandas

```

Here is the Python function:

```python

import pandas as pd

from cassandra.cluster import Cluster

from sklearn.preprocessing import MinMaxScaler

def extract\_and\_process\_data(keyspace, table, contact\_points, port=9042):

"""

Extracts and processes financial time series data from a Cassandra database.

Parameters:

keyspace (str): The keyspace in the Cassandra database.

table (str): The table in the Cassandra database.

contact\_points (list): List of contact points for the Cassandra cluster.

port (int): The port number for the Cassandra cluster.

Returns:

pd.DataFrame: Cleaned and normalized dataset.

"""

# Connect to the Cassandra cluster

cluster = Cluster(contact\_points=contact\_points, port=port)

session = cluster.connect(keyspace)

# Execute the query to fetch data

query = f"SELECT \* FROM {table}"

rows = session.execute(query)

# Convert the rows to a pandas DataFrame

data = pd.DataFrame(rows)

# Close the Cassandra session

session.shutdown()

cluster.shutdown()

# Handle missing values by filling them with the mean of the column

data.fillna(data.mean(), inplace=True)

# Normalize the data using MinMaxScaler

scaler = MinMaxScaler()

data\_normalized = pd.DataFrame(scaler.fit\_transform(data), columns=data.columns)

return data\_normalized

# Example usage

if \_\_name\_\_ == "\_\_main\_\_":

keyspace = 'finance\_data'

table = 'time\_series\_data'

contact\_points = ['127.0.0.1']

port = 9042

cleaned\_data = extract\_and\_process\_data(keyspace, table, contact\_points, port)

print(cleaned\_data.head())

```

**Explanation:**

1. **Connecting to Cassandra**: The function connects to the Cassandra cluster using the provided contact points and port.

2. **Fetching Data**: It executes a query to fetch all data from the specified table.

3. **DataFrame Conversion**: The fetched data is converted into a pandas DataFrame.

4. **Handling Missing Values**: Missing values are handled by filling them with the mean of the respective columns.

5. **Normalizing Data**: The data is normalized using `MinMaxScaler` from `sklearn.preprocessing`.

6. **Returning Cleaned Data**: The function returns the cleaned and normalized dataset.

**PEP 8 Compliance:**

- The function adheres to PEP 8 coding standards, including proper indentation, line length, and use of docstrings.

**Note:**

- Ensure that the Cassandra cluster is running and accessible from the environment where this script is executed.

- Adjust the `keyspace`, `table`, `contact\_points`, and `port` parameters as per your Cassandra database configuration.

--------------------------------------------------