Mandapallihema

ASSESSMENT-19

ETL WITH PYSPARK:

Efficient Transformations:

1. Use built-in PySpark functions whenever possible, as they are optimized for distributed computing.

Partitioning and Clustering:

1. Choose appropriate column(s) for partitioning to optimize data distribution across nodes.

Caching:

1. Cache intermediate DataFrames if they are reused multiple times to avoid recomputation.

Error Handling:

1. Implement robust error handling to manage potential issues during the ETL process.

Testing:

1. Develop unit tests for your ETL pipeline to ensure correctness.
2. Utilize PySpark’s testing capabilities for DataFrame testing.

The PySpark ETL Workflow:

Extract: Retrieve data from various sources like databases, files, or APIs.

Transform: Clean, aggregate, and manipulate data to fit your analysis needs.

Load: Store the transformed data into a database or data warehouse for analysis.

Installed jupyter using the command “py -m pip install jupyter”.

Launched by using command “jupyter lab”.

