Key Takeaways

Project Title: PySpark Foundations: Process, analyze, and summarize data

Task 1

- PySpark is an interface for Apache Spark in Python to manipulate and analyze data in a distributed processing environment.
- **SparkSession.builder** creates an entry point to using PySpark; **appName()** names the Spark application; **getOrCreate()** retrieves an existing Spark session or creates a new one.

Task 2

- **spark.read.csv:** a method in Spark that reads a CSV file into a data frame.
- **header=True:** specifies that the first row of the CSV file contains the header (column names)
- **inferSchema=True:** tells Spark to automatically infer the data types of each column based on the values in the CSV file.

Task 3

- Formatting allows for standardization and normalization of data. It aids in error detection and data cleaning, setting the stage for reliable data analytics.

Task 4

- Data exploration, one of the first steps in data preparation, is a way to get to know data before working with it.
- **toPandas()** converts the Spark data frame to a Pandas data frame.

Task 5

- The **groupBy** operation allows you to perform aggregate functions (such as sum, count, max, min, etc.) on grouped data, based on one or more columns.
- The **agg()** function is used to apply aggregate functions to the grouped data. You would typically pass functions inside **agg()** to specify what operation to perform on the grouped data.

Task 6

- Use left join when you need all records from the left table and the matching records from the right table. Unmatched records from the right table will have NULL values.