**Writing a Java Program to Access the Hadoop Database**

**Part 1: Ingesting Data into the HDFS**

For the first part of this assignment, the SalesData.csv file was successfully ingested into Hadoop data storage.

The testprogram.zip file was downloaded and extracted. The command line interface was opened, and the extracted folder was accessed.

A computer screen shot of a blue screen

Description automatically generated

The directory was changed to the parent of the testprogram, and the Docker copy command was executed, resulting in the testprogram folder being copied into the home directory of the Hadoop namenode.

A screenshot of a computer screen

Description automatically generated

The namenode CLI in Docker was accessed. Using the provided HDFS command, an input folder named inputMapReduce was established inside the home directory. Subsequently, the SalesData.csv file was transferred into the Hadoop data storage.

A screen shot of a computer

Description automatically generated

The newly transferred file was reviewed using the HDFS cat command, and its successful copying was confirmed through a provided screenshot.

A screen shot of a computer program

Description automatically generated

**Part 2: Performing MapReduce: Aggregation Sales by Country**

For the second segment of this assignment, the code files within the testprogram folder on the namenode were employed using the Docker CLI.

A review of the source code in the testprogram folder was conducted. Descriptions of each of the three Java files, which are pivotal for the MapReduce framework's application on the Hadoop database, were submitted.

**SalesCountryDriver.java**: This class acts as the driver for the MapReduce job, setting up and running it.

**SalesMapper.java**: Implements the mapping phase of the MapReduce job. It reads the input data, processes it, and emits key-value pairs to be processed by the reducer.

**SalesCountryReducer.java**: Implements the reducing phase of the MapReduce job. It receives the key-value pairs emitted by the mapper, aggregates them, and produces the final output.

Several commands were run in the namenode CLI to define an executable path in the system environment.

A screen shot of a computer screen

Description automatically generated

Subsequently, it was confirmed that the environment variables were correctly set, as evidenced by a provided screenshot.

A screenshot of a computer screen

Description automatically generated

The /home/testprogram folder in the namenode CLI was accessed, and a Java program was compiled. This resulted in the generation of a SalesCountry folder containing specific class files.

A blue screen with white text

Description automatically generated

A jar file was crafted from the Java code that had been compiled in the previous stage.

A screen shot of a computer

Description automatically generated

A MapReduce operation was conducted, and a screenshot was shared to confirm the successful execution and analysis distribution of the data.

A screenshot of a computer program

Description automatically generated

The output of the SalesCountry data in the part-00000 file was examined.

A screenshot of a computer

Description automatically generated