**Sushmitha Yelamakonda**

**S555058**

**Example: Banking**

Input Data (Transaction Data):

|  |
| --- |
| **Customer Name Amount ($) Branch Location Timestamp** |
| Jeevan 1500 Dallas 2023-01-01 02:00 AM |
| Raghav 3000 Las Vegas 2023-01-01 02:30 AM |
| Vidya 2500 Charleston 2023-01-01 03:15 AM |
| Varun 650 St. Louis 2023-01-01 03:45 AM |
| Jeeva 700 Illinois 2023-01-01 04:05 AM |
| Jaya 300 Omaha 2023-01-01 04:30 AM |
| Rolex 200 Charleston 2023-01-01 05:15 AM |
| Surya 400 St. Joseph 2023-01-01 05:45 AM |
| Jaya 1000 Dallas 2023-01-01 06:10 AM |
| Roshan 2000 Las Vegas 2023-01-01 06:30 AM |
| Vidya 5000 Maryville 2023-01-01 07:00 AM |
| Varun 6050 St. Louis 2023-01-01 07:45 AM |
| Jeeva 7000 Illinois 2023-01-01 08:10 AM |
| Jaya 3500 Omaha 2023-01-01 08:30 AM |
| Rolex 200 Maryville 2023-01-01 09:15 AM |
| Surya 650 St. Joseph 2023-01-01 10:00 AM |

**Splitting:**

The input data is first divided into smaller chunks to be processed by individual mappers. Each transaction record represents a line of data, and we split the data based on the branch location.

Input is divided into four groups based on time intervals (every 2 hours).

**Mapping:**

In the mapping phase, each Mapper processes its assigned data chunk and emits key-value pairs. The key is the branch location, and the value is the transaction amount.

**Dallas:** 1500

**Las Vegas:** 3000

**Charleston:** 2500

**St. Louis:** 650

**Illinois:** 700

**Omaha:** 300

**Charleston:** 200

**St. Joseph:** 400

**Dallas:** 1000

**Las Vegas:** 2000

**Maryville:** 200

**St. Louis:** 6050

**Illinois:** 7000

**Omaha:** 3500

**Maryville:** 200

**St. Joseph:** 650

**Shuffling and Sorting:**

The Shuffling and SortingPhasewill group and sort the key-value pairs by the key branch location). All records with the same key are grouped together.

**Reducing:**

The Reducer will calculate the total transaction amounts for each branch location, providing a summary of the overall transaction activity at each branch.

Here MapReduction Function is **“Sum”.**

**Dallas:** 1500 + 1000.

**Las Vegas:** 3000 + 2000.

**Charleston:** 2500 + 200.

**St. Louis:** 650 + 6050.

**Illinois:** 700 + 7000.

**Omaha:** 300 + 3500.

**Maryville:** 200 + 200.

**St. Joseph:** 400 + 650**.**

**Output:**

The outcome consists of branch-specific reports that include the total transaction amounts for each branch.

**Dallas** 2500

**Las Vegas** 5000

**Charleston** 2700

**St. Louis** 6700

**Illinois** 7700

**Omaha** 3800

**Maryville** 400

**St. Joseph** 1050

This above example shows how MapReduce can be used to analyze banking data across different branches, helping financial institutions gain insights into branch-specific performance, customer behavior, and transaction trends.

**Diagram:**

**A screenshot of a computer

Description automatically generated**