# Activity 4

Let us go back to the Wordcount application example. This time, we will try to execute the same MapReduce job in **Hive queries**.

To perform the same Job in Hive, the following queries can be executed:

# Create a table to store results

$ hive> CREATE TABLE files (line STRING);

# Load data into the database using a file on your local system (NOT HDFS)

$ hive> LOAD DATA LOCAL INPATH '/file01.txt' INTO TABLE files;

# Create a new table using data from the files table

$ hive> CREATE TABLE word\_counts AS

> SELECT word, count(1) AS count FROM

> (SELECT explode(split(line, ' ')) AS word FROM files) w

> GROUP BY word

> ORDER BY word;

To see the result, you can run a SELECT query on the word\_count table:

$ SELECT \* FROM word\_count;

Bye,1

Goodbye,1

Hadoop,2

Hello,2

World,2

Solution

|  |
| --- |
| # Create a table to store results |
|  |

|  |
| --- |
| $ hive> CREATE TABLE files (line STRING); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| # Load data into the database using a file on your local system (NOT HDFS) |
|  |

|  |
| --- |
| $ hive> LOAD DATA LOCAL INPATH '/file01.txt' INTO TABLE files; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| # Create a new table using data from the files table |
|  |

|  |
| --- |
| $ hive> CREATE TABLE word\_counts AS |
|  |

|  |
| --- |
| > SELECT word, count(1) AS count FROM |
|  |

|  |
| --- |
| > (SELECT explode(split(line, ' ')) AS word FROM files) w |
|  |

|  |
| --- |
| > GROUP BY word |
|  |

> ORDER BY word;