Advanced MapReduce Techniques on Crime Data Analysis

Document overview:

1. Data Overview

2. Join By with MapReduce: Implement a MapReduce program to perform a join operation on Dataset 1 and Dataset 2 based on a common column.

3. Secondary Sort: Extending the join operation from above part. Sorting the joined results based on the chosen column in ascending or descending order

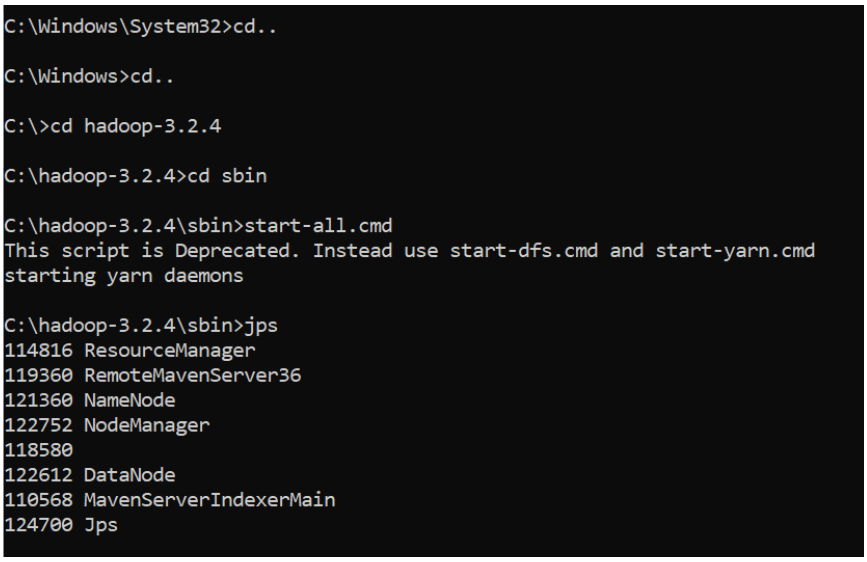
4. Testing with MRUnit: Implementing unit test for the MapReduce job from 2nd part using MRUnit and writing test cases to validate the correctness of the MapReduce job's output.

1. Data Overview:

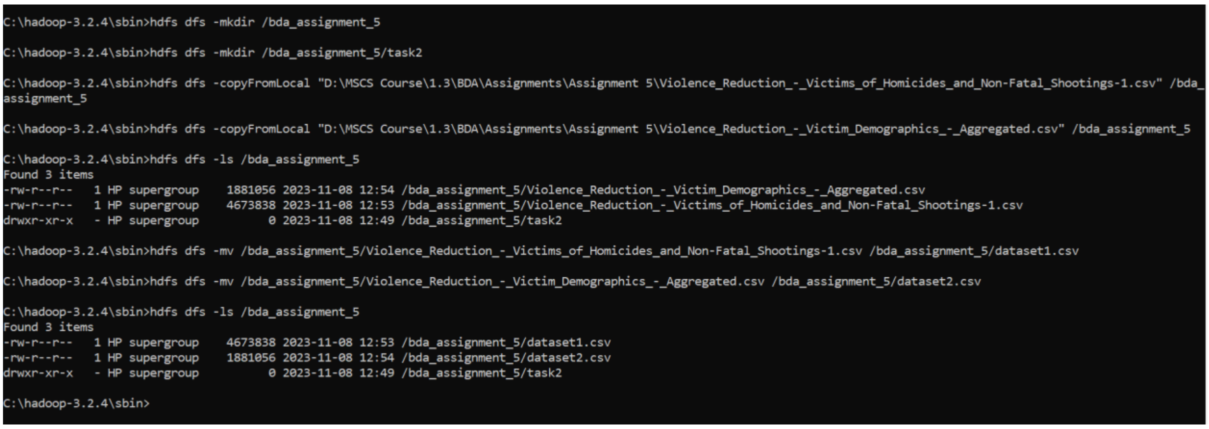
We were provided with 2 datasets(.csv) files.

- dataset1 - Violence_Reduction_-_Victim_Demographics_-_Aggregated.csv
- columns in dataset1 are :
    - TIME_PERIOD – It contains date in format of MM/DD/YYYY
    - VICTIMIZATION_PRIMARY, - type of victimization like HOMICIDE, BATTERY etc..
    - AGE, - It contains age range values like 30-39, 20-29
    - SEX – gender of victim
    - RACE – race of victim like BLK, WHI
    - NUMBER_OF_VICTIMS – number of victims. It’s a integer value
- dataset2 - Violence_Reduction_-_Victims_of_Homicides_and_Non-Fatal_Shootings-1.csv
- columns in dataset2 are :
    - DATE – Date and time when the incident happened
    - BLOCK – the block where it happened
    - LOCATION_DESCRIPTION – location where it happened like street names
    - COMMUNITY_AREA – the area where it happened
    - VICTIMIZATION_PRIMARY, - type of victimization like HOMICIDE, BATTERY etc..
    - AGE, - It contains age range values like 30-39, 20-29
    - SEX – gender of victim
    - RACE – race of victim like BLK, WHI
    - GUNSHOT_INJURY_I – It has YES or NO values to indicate if the incident has any gunshot injuries or not
- VICTIMIZATION_PRIMARY, AGE, SEX, and RACE are common columns in both the datasets.
- AGE values are range values and SEX and RACE are too generic so we can use VICTIMIZATION_PRIMARY to merge the two datasets based on common column.

Prerequisites for next step – Starting daemons.



```
C:\Windows\System32>cd..

C:\Windows>cd..

C:\>cd hadoop-3.2.4

C:\hadoop-3.2.4>cd sbin

C:\hadoop-3.2.4\sbin>start-all.cmd
This script is Deprecated. Instead use start-dfs.cmd and start-yarn.cmd
starting yarn daemons

C:\hadoop-3.2.4\sbin>jps
114816 ResourceManager
119360 RemoteMavenServer36
121360 NameNode
122752 NodeManager
118580
122612 DataNode
110568 MavenServerIndexerMain
124700 Jps
```



```
C:\hadoop-3.2.4\sbin>hdfs dfs -mkdir /bda_assignment_5

C:\hadoop-3.2.4\sbin>hdfs dfs -mkdir /bda_assignment_5/task2

C:\hadoop-3.2.4\sbin>hdfs dfs -copyFromLocal "D:\MSCS Course\1.3\BDA\Assignments\Assignment 5\Violence_Reduction_-_Victims_of_Homicides_and_Non-Fatal_Shootings-1.csv" /bda_
assignment_5

C:\hadoop-3.2.4\sbin>hdfs dfs -copyFromLocal "D:\MSCS Course\1.3\BDA\Assignments\Assignment 5\Violence_Reduction_-_Victim_Demographics_-_Aggregated.csv" /bda_assignment_5

C:\hadoop-3.2.4\sbin>hdfs dfs -ls /bda_assignment_5
Found 3 items
-rw-r--r--   1 HP supergroup    1881056 2023-11-08 12:54 /bda_assignment_5/Violence_Reduction_-_Victim_Demographics_-_Aggregated.csv
-rw-r--r--   1 HP supergroup    4673838 2023-11-08 12:53 /bda_assignment_5/Violence_Reduction_-_Victims_of_Homicides_and_Non-Fatal_Shootings-1.csv
drwxr-xr-x   - HP supergroup          0 2023-11-08 12:49 /bda_assignment_5/task2

C:\hadoop-3.2.4\sbin>hdfs dfs -mv /bda_assignment_5/Violence_Reduction_-_Victims_of_Homicides_and_Non-Fatal_Shootings-1.csv /bda_assignment_5/dataset1.csv

C:\hadoop-3.2.4\sbin>hdfs dfs -mv /bda_assignment_5/Violence_Reduction_-_Victim_Demographics_-_Aggregated.csv /bda_assignment_5/dataset2.csv

C:\hadoop-3.2.4\sbin>hdfs dfs -ls /bda_assignment_5
Found 3 items
-rw-r--r--   1 HP supergroup    4673838 2023-11-08 12:53 /bda_assignment_5/dataset1.csv
-rw-r--r--   1 HP supergroup    1881056 2023-11-08 12:54 /bda_assignment_5/dataset2.csv
drwxr-xr-x   - HP supergroup          0 2023-11-08 12:49 /bda_assignment_5/task2

C:\hadoop-3.2.4\sbin>
```

2. Joining Datasets

The bda_5_task_2 zip file which contains code for JoinMapper, JoinReducer and JoinRunner classes and a README file.

EXPLANATION:

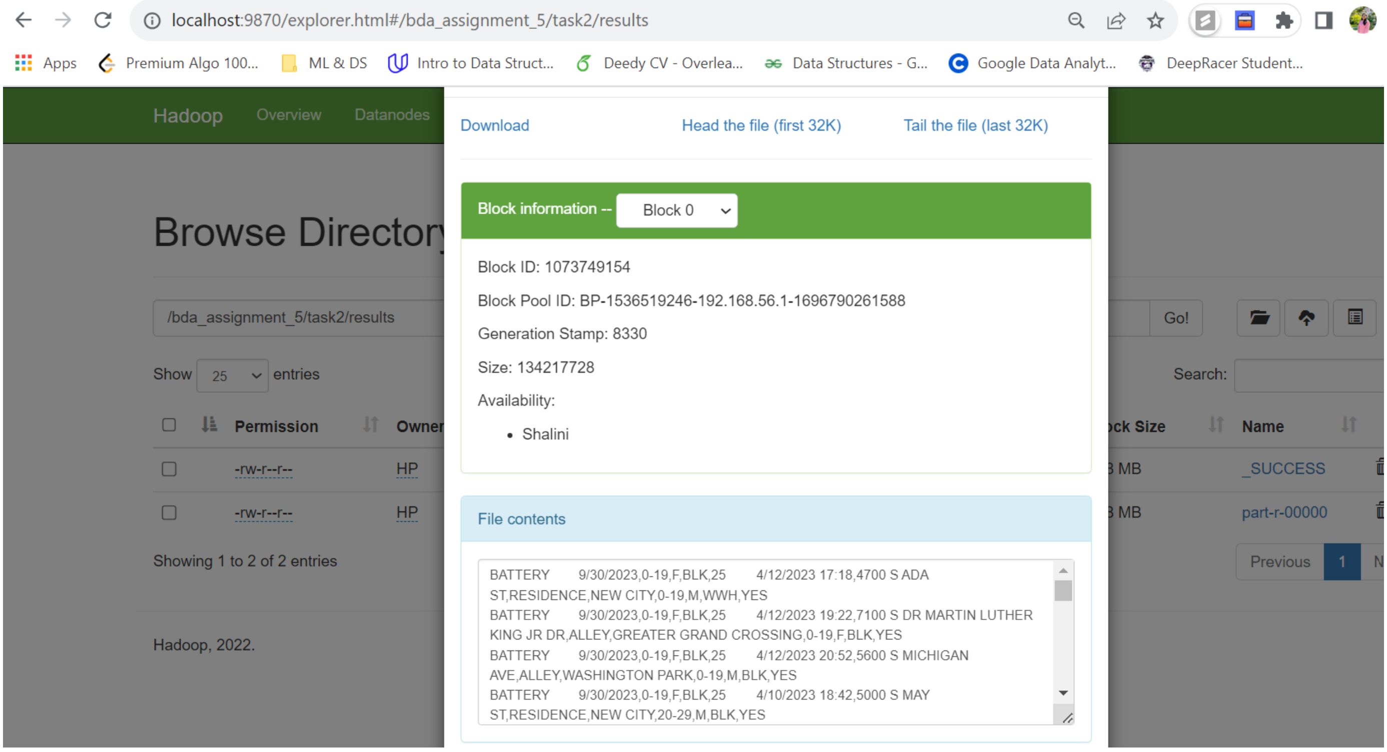
The MapReduce workflow for joining two datasets based on a common column involves 3 steps:

- JoinMapper: It takes 2 datasets rows as the input and distinguishes the rows of datasets based on numbers of columns for each row. If row has 6 columns, then it is dataset1, else it is dataset 2. Common column is located at 1st index in dataset1, and 4th index in dataset2. Then, it assigns a tag for each record and extracts the join key which is VICTIMIZATION_PRIMARY. The output of the JoinMapper script is a key-value pair, where the key is the join key(VICTIMIZATION_PRIMARY) and the value is the tagged record(rest of the columns data for that record)
- JoinReducer: It receives all values associated with the same key from the output of the JoinMapper class so that input will be key-value pairs from JoinMapper class. This class distinguishes each record based on their dataset tag assigned. It then combines records from both datasets that have the same column value for common column. The output from JoinReducer class is records with VICTIMIZATION_PRIMARY column, dataset1 columns except VICTIMIZATION_PRIMARY and dataset2 columns except VICTIMIZATION_PRIMARY
- JoinRunner: It is responsible for configuring and executing the MapReduce job. It specifies the job configuration, input and output formats, mapper, and reducer class. It submits the job to the Hadoop cluster and manages its execution.

- Hadoop automatically shuffles and sorts these key-value pairs, grouping them by key.

OUTPUT:

Block 0, head file



Block ID: 1073749154

Block Pool ID: BP-1536519246-192.168.56.1-1696790261588

Generation Stamp: 8330

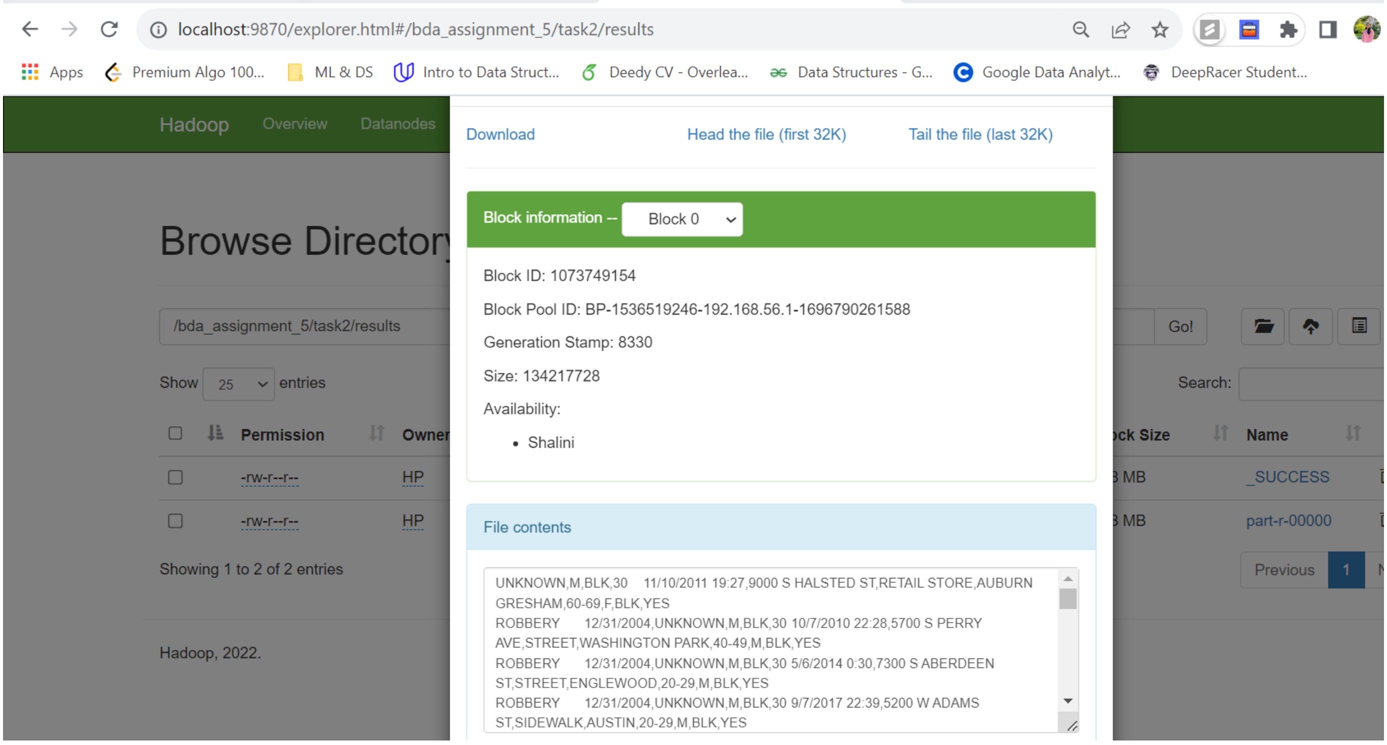
Size: 134217728

Availability:

- Shalini

```
BATTERY	9/30/2023,0-19,F,BLK,25	4/12/2023 17:18,4700 S ADA
ST,RESIDENCE,NEW CITY,0-19,M,WWH,YES
BATTERY	9/30/2023,0-19,F,BLK,25	4/12/2023 19:22,7100 S DR MARTIN LUTHER
KING JR DR,ALLEY,GREATER GRAND CROSSING,0-19,F,BLK,YES
BATTERY	9/30/2023,0-19,F,BLK,25	4/12/2023 20:52,5600 S MICHIGAN
AVE,ALLEY,WASHINGTON PARK,0-19,M,BLK,YES
BATTERY	9/30/2023,0-19,F,BLK,25	4/10/2023 18:42,5000 S MAY
ST,RESIDENCE,NEW CITY,20-29,M,BLK,YES
```

Block 0, tail file



Block ID: 1073749154

Block Pool ID: BP-1536519246-192.168.56.1-1696790261588

Generation Stamp: 8330

Size: 134217728

Availability:

- Shalini

```
UNKNOWN,M,BLK,30	11/10/2011 19:27,9000 S HALSTED ST,RETAIL STORE,AUBURN
GRESHAM,60-69,F,BLK,YES
ROBBERY	12/31/2004,UNKNOWN,M,BLK,30 10/7/2010 22:28,5700 S PERRY
AVE,STREET,WASHINGTON PARK,40-49,M,BLK,YES
ROBBERY	12/31/2004,UNKNOWN,M,BLK,30 5/6/2014 0:30,7300 S ABERDEEN
ST,STREET,ENGLEWOOD,20-29,M,BLK,YES
ROBBERY	12/31/2004,UNKNOWN,M,BLK,30 9/7/2017 22:39,5200 W ADAMS
ST,SIDEWALK,AUSTIN,20-29,M,BLK,YES
```

3. Secondary Sort

Extend the join operation from above part. Sorting the joined results based on the chosen column in ascending or descending order and outputing the sorted results in the format: <columns from Dataset 1> <columns from Dataset 2>.

The bda_5_task_3 zip file which contains code for all classes and a README file.

EXPLANATION:

I have the created or modified the following classes to implement task 3 secondary sort on given datasets. I chose age in addition to VICTIMIZATION_PRIMARY column to perform secondary sort.

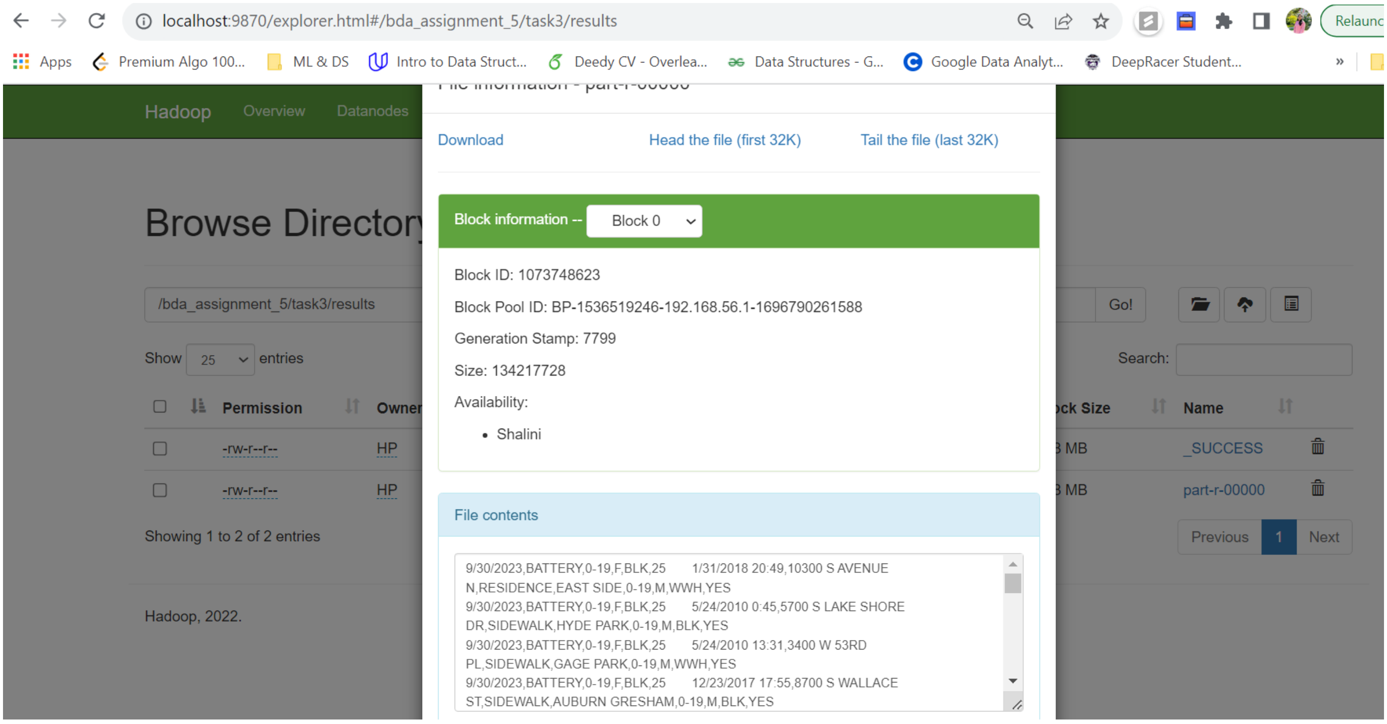
classes:

- CompositeKey: It combines the common column (VICTIMIZATION_PRIMARY) with an additional common column (age). This composite key is used for both sorting and grouping the mapper output.
- CustomPartitioner : It was created to ensure that all records with the same VICTIMIZATION_PRIMARY (common column) go to the same reducer, regardless of their age value. it is important for correct grouping in the reduce phase.
- GroupComparator: It was created to ensure that records with the same VICTIMIZATION_PRIMARY are grouped together in the reduce phase, regardless of the age value. It is important because the composite key now contains an additional sorting field (age).
- JoinMapper : The mapper class was updated to output CompositeKey as the key instead of just the one key used in Task 2. It ensures that each map output record has a composite key containing both the common column and the secondary sort column.
- JoinReducer : The reducer class now accepts CompositeKey as its input key type. However, the logic of joining the datasets based on the common column remains the same. here, the output is formatted to include only the joined columns from both datasets.
- JoinRunner : It was modified to use the new JoinMapper, JoinReducer, and the additional classes (CustomPartitioner, GroupComparator, KeyComparator). It also configures the job to use these new components.
- KeyComparator : It was created to ensure the order of records. It sorts the records based on both VICTIMIZATION_PRIMARY and age, allowing for secondary sorting within each group of common VICTIMIZATION_PRIMARY.

Doing these modifications to join operation scripts from task 2, I implemented secondary sort to perform join operation and then sort the results by an additional secondary column(age). However, output format remains the same which displays columns from dataset1 followed by dataset

OUTPUT

Block 0, head file



Block ID: 1073748623

Block Pool ID: BP-1536519246-192.168.56.1-1696790261588

Generation Stamp: 7799

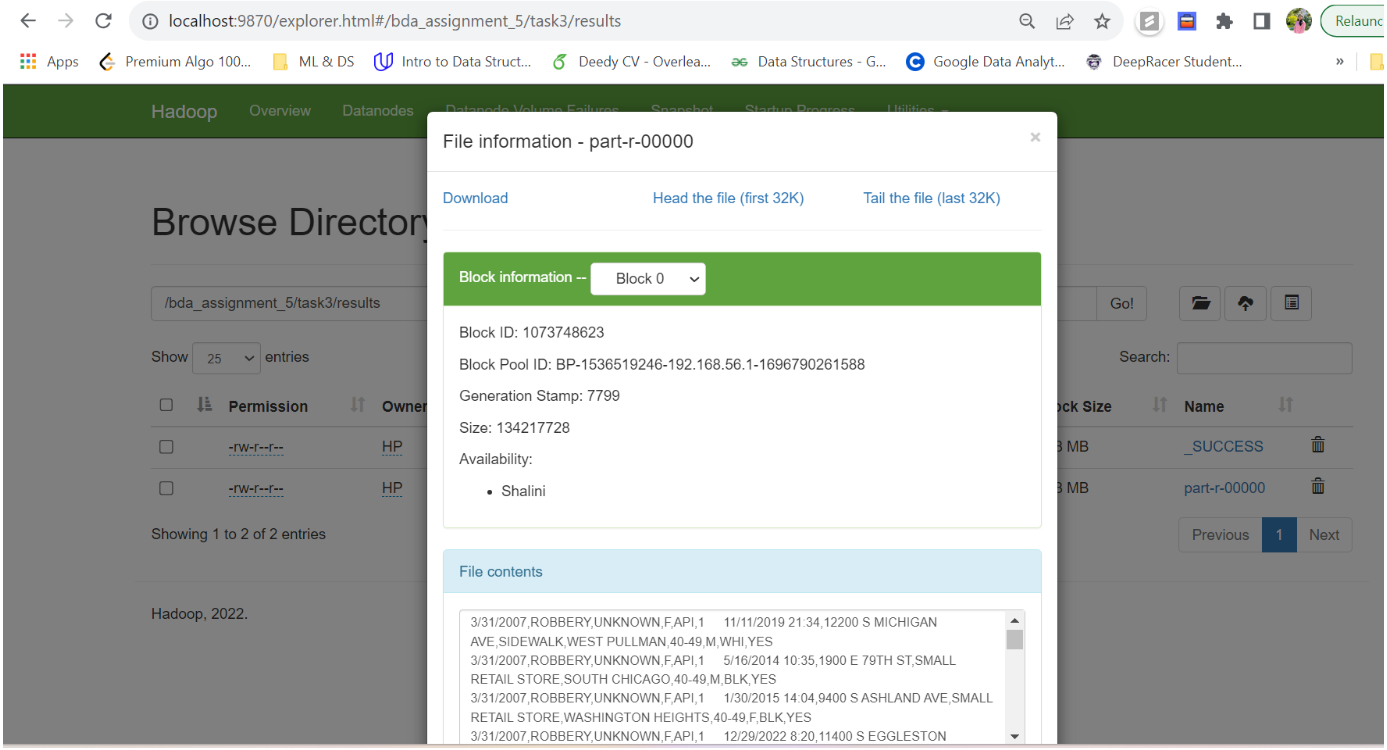
Size: 134217728

Availability:

- Shalini

```
9/30/2023,BATTERY,0-19,F,BLK,25      1/31/2018 20:49,10300 S AVENUE
N,RESIDENCE,EAST SIDE,0-19,M,WWH,YES
9/30/2023,BATTERY,0-19,F,BLK,25      5/24/2010 0:45,5700 S LAKE SHORE
DR,SIDEWALK,HYDE PARK,0-19,M,BLK,YES
9/30/2023,BATTERY,0-19,F,BLK,25      5/24/2010 13:31,3400 W 53RD
PL,SIDEWALK,GAGE PARK,0-19,M,WWH,YES
9/30/2023,BATTERY,0-19,F,BLK,25      12/23/2017 17:55,8700 S WALLACE
ST,SIDEWALK,AUBURN GRESHAM,0-19,M,BLK,YES
```

Block 0, tail file



Block ID: 1073748623

Block Pool ID: BP-1536519246-192.168.56.1-1696790261588

Generation Stamp: 7799

Size: 134217728

Availability:

- Shalini

```
3/31/2007,ROBBERY,UNKNOWN,F,API,1      11/11/2019 21:34,12200 S MICHIGAN
AVE,SIDEWALK,WEST PULLMAN,40-49,M,WHI,YES
3/31/2007,ROBBERY,UNKNOWN,F,API,1      5/16/2014 10:35,1900 E 79TH ST,SMALL
RETAIL STORE,SOUTH CHICAGO,40-49,M,BLK,YES
3/31/2007,ROBBERY,UNKNOWN,F,API,1      1/30/2015 14:04,9400 S ASHLAND AVE,SMALL
RETAIL STORE,WASHINGTON HEIGHTS,40-49,F,BLK,YES
3/31/2007,ROBBERY,UNKNOWN,F,API,1      12/29/2022 8:20,11400 S EGGLESTON
```

4. Implementing unit test for the MapReduce job from 2nd part using MRUnit and writing test cases to validate the correctness of the MapReduce job's output.

The bda_5_task_4 zip file which contains code for all classes and a README file.

I added JoinMapperReducerTest class to perform unit test the MapReduce program from Task 2 which has setup and test methods. I used setup to create objects for JoinMapper and JoinReducer classes and test method to provide input and expected output.

I used the following test case to test my mapreduce program

MapReduce program Input:

One record from D1(dataset1):

“3/31/2011,HOMICIDE,30-39,M,BLK,1”

One record from D2(dataset2):

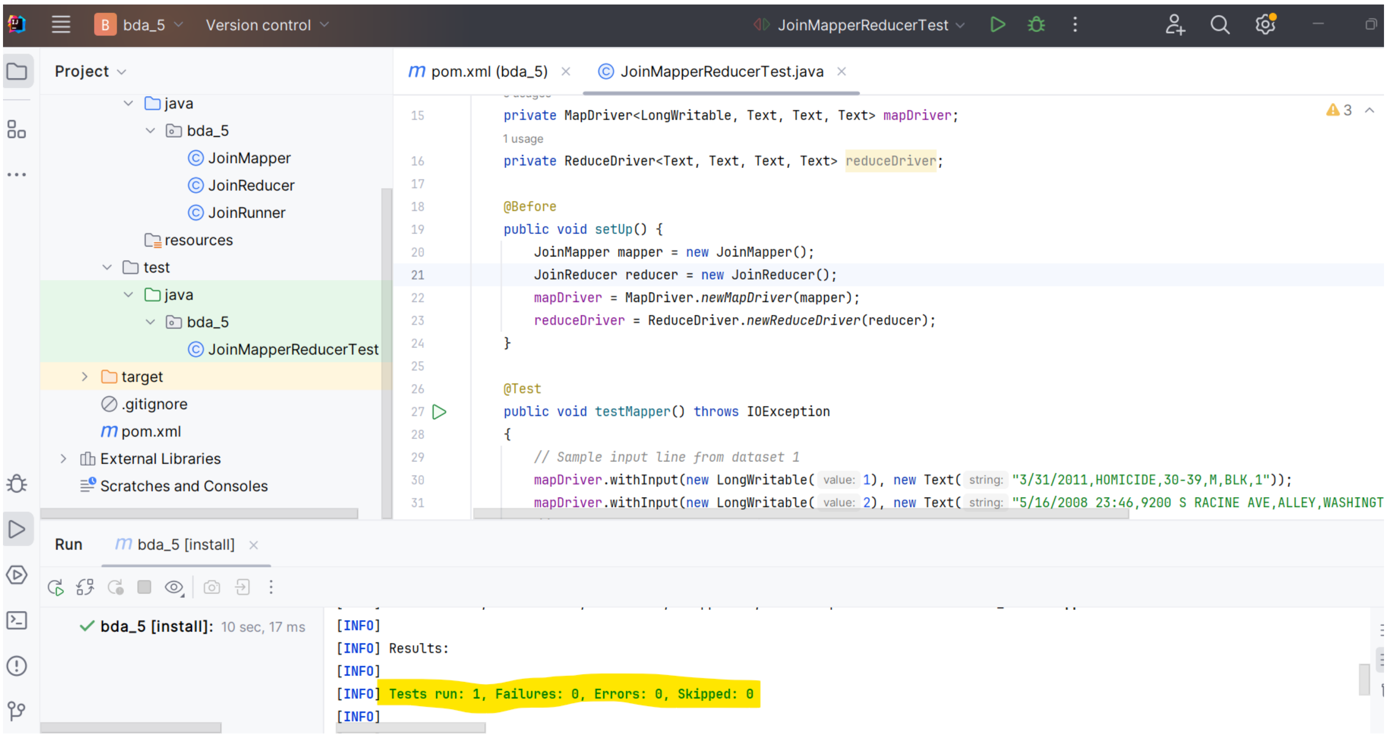
“5/16/2008 23:46,9200 S RACINE AVE,ALLEY,WASHINGTON HEIGHTS,HOMICIDE,40-49,M,BLK,YES”

MapReduce program output:

“HOMICIDE 3/31/2011,30-39,M,BLK,1 5/16/2008 23:46,9200 S RACINE AVE,ALLEY,WASHINGTON HEIGHTS,40-49,M,BLK,YES”

TASK 4 OUTPUT:

The given test case passed successfully



```java
    private MapDriver<LongWritable, Text, Text, Text> mapDriver;
    1 usage
    private ReduceDriver<Text, Text, Text, Text> reduceDriver;

    @Before
    public void setUp() {
        JoinMapper mapper = new JoinMapper();
        JoinReducer reducer = new JoinReducer();
        mapDriver = MapDriver.newMapDriver(mapper);
        reduceDriver = ReduceDriver.newReduceDriver(reducer);
    }

    @Test
    public void testMapper() throws IOException
    {
        // Sample input line from dataset 1
        mapDriver.withInput(new LongWritable( value: 1), new Text( string: "3/31/2011,HOMICIDE,30-39,M,BLK,1"));
        mapDriver.withInput(new LongWritable( value: 2), new Text( string: "5/16/2008 23:46,9200 S RACINE AVE,ALLEY,WASHINGT
```

✓ bda_5 [install]: 10 sec, 17 ms

```
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
```