Assignment 2

- You must work on the assignments in teams of two to three students (if possible, keep the same groups for further assignments). Individual submissions will not be accepted.
- Create your groups in Ilias until June 21st 2019. After you created the groups, you will receive the connection data of the remote VM in Ilias. Feel free to contact Manuel Fritz, if your ZIP file is not available within 24 hours.
- We provide a pre-built Hadoop environment on a VM with real world taxi trip data, fuel prices, and lottery data. Regarding the setup of the necessary infrastructure and connection, please refer to the slides on assignment 2.
- Username and password of the VM is always "cloudera".
- Submit a single ZIP file (name: AIM2019_A2_name1_name2_name3.zip) containing the PDF with your results (name: AIM2019_A2_name1_name2_name3_Results.pdf) **and** the source code (name: AIM2019_A2_name1_name2_name3_Source.zip). Make sure that the first page of your PDF lists the names of all team members.
- Export your Java Project in Eclipse via: Right-click on the project → "Export…" → "General" → "Archive File". Change the name of the archive file corresponding to the description above and click on "Finish". (Leave the options as they are, i.e. "Save in zip format" and "Create directory structure for files" selected as well as "Compress the contents of the file" checked)
- You have to submit your solution in Ilias until July 5th, 2019, 10:00 am.
- We will discuss selected solutions of this assignment on July 19th, 2019 at 11:30 am.
- If you have questions, please ask for help in the Ilias forum first. If required, send an email to manuel.fritz@ipvs.uni-stuttgart.de.

Task 1 – MapReduce Programming 1: Basics — 15 points

1.1 Count the number of occurrences for each individual drawn lottery number ("lotterynumbers"). Use the class LotteryCount in the package lottery. Copy the output of the file created by MapReduce into your result PDF. (5 points)

1.2 You want to play the lottery based on an evaluation of historical drawings. Therefore, you want to discover the 5 individual numbers ("lotterynumbers") that have been drawn most often. Limit the output on the exact top 5 occurrences of drawn numbers using the MapReduce paradigm. Use the class LotteryCountTop5 in the package lottery. Copy the table below into your result PDF and fill in the results. (10 points)

Drawn number	Occurrence

Task 2 – MapReduce Programming 2: Joins 25 points

2.1 Explain how a generic Reduce-side Join for an arbitrary relationship (1:1, 1:n as well as n:m) works. How is the data processed in each step? Make sure, that the output of the Map step is general enough to support all kinds of relationship types in the Reduce step. Copy the table below into your result PDF and insert the explanations to the corresponding step. (5 points)

Step	Explanation
Map	
Combine	
Partition	
Shuffle	
Sort	
Reduce	

2.2 Aggregate the distance and the earnings from the taxi trip data set per "dropoff_date". Use the class TaxiAggregate in the package taxi. Copy the table below into your result PDF and insert the sum of the distance as well as the earnings per "dropoff_date". (5 points)

dropoff_date	distance	Earnings
06-01-2015		
06-02-2015		
06-03-2015		
06-04-2015		
06-05-2015		
06-06-2015		
06-07-2015		
06-08-2015		

2.3 Based on the results of 2.2, implement a reduce-side join with the fuel prices. Join on the "dropoff_date" and "businessday" (fuel prices) columns. Assume that the averaged fuel efficiency is 21.3 miles per gallon (mpg). Calculate the theoretical gross margin per day if all drivers would have tanked up in Buffalo only. Use the class TaxiJoin in the package taxi. Copy the table below into your result PDF and insert the results. Note that not necessarily all cells will be needed. Explain briefly why! (15 points)

Date	Gross Margin (Buffalo)
06-01-2015	
06-02-2015	
06-03-2015	

06-04-2015	
06-05-2015	
06-06-2015	
06-07-2015	
06-08-2015	

Task 3 – Hive 10 points

3.1 What are the most 5 frequently drawn individual lottery numbers (see 1.2)? Solve this task using HiveQL. Copy your HiveQL query as well as the result of this query in the result document.
Hint: Have a look at the split() function provided by HiveQL. (3 points)

3.2 Using the taxi trip data set, aggregate the distance and earnings per "dropoff_date". Use the result and join on the "dropoff_date" and "businessday" (fuel prices) columns. Assume that the averaged fuel efficiency is 21.3 miles per gallon (mpg). Calculate the theoretical gross margin per day if all drivers would have tanked up in Buffalo only. See 2.2 & 2.3, but use HiveQL for this task. Copy your HiveQL query as well as the result of this query in the results. (7 points)