RMIT UNIVERSITY

COSC 2637/2633 Big Data Processing
Assignment 2 – Handling Big Data with Apache Pig

Assessment Type	– Individual assignment. – Submit online via Canvas → Assignment 2. – Marks awarded for meeting requirements as closely as possible. – Clarifications/updates may be made via announcements or relevant discussion forums.
Due Date	23:59, 24 Sep
Marks	25

Overview
Write Apache Pig scripts which give you a chance to develop a basic understanding of principles when handling queries on large data stored on HDFS.

Learning Outcomes
The key course learning outcomes are:
- CLO 1: model and implement efficient big data solutions for various application areas using appropriately selected algorithms and data structures.
- CLO 2: analyze methods and algorithms, to compare and evaluate them with respect to time and space requirements and make appropriate design choices when solving real-world problems.
- CLO 3: motivate and explain trade-offs in big data processing technique design and analysis in written and oral form.
- CLO 4: explain the Big Data Fundamentals, including the evolution of Big Data, the characteristics of Big Data and the challenges introduced.
- CLO 6: apply the novel architectures and platforms introduced for Big data, i.e., Hadoop, MapReduce and Spark.

Assessment Details
This assignment adopts a sample bookstore database from an online source. It has books, authors, customers, orders, and several other tables. The entity relationship diagram (ERD) can be found on the source site. The database is originally managed by a DBMS. If the database is too large, you need to store the data on a cluster of computers and manage it with a Hadoop platform. So, you need to develop Apache Pig scripts to process the data as SQL queries.

Task-1. From the tables cust_order and order_line of the database, the tuples have been extracted and stored in two files cust_order.csv and order_line.csv, respectively. Developing an Apache Pig script that outputs the same as the following SQL query. (**15 marks**)

```
SELECT
DATE_FORMAT(co.order_date, '%Y-%m-%d') AS order_day,
COUNT(DISTINCT co.order_id) AS num_orders,
COUNT(ol.book_id) AS num_books,
SUM(ol.price) AS total_price
FROM cust_order co
INNER JOIN order_line ol ON co.order_id = ol.order_id
GROUP BY
  DATE_FORMAT(co.order_date, '%Y-%m-%d')
ORDER BY total_price DESC;
```

PK: order_id

A sample of data in cust_order:

```
order_id,order_date,customer_id,shipping_method_id,dest_address_id
1,"2020-05-10 01:35:56",1,3,380
2,"2022-02-03 06:14:25",1,4,515
3,"2021-11-24 10:08:43",1,2,766
4,"2020-07-28 18:34:14",2,1,189
5,"2021-02-21 23:30:16",2,1,978
6,"2022-11-27 17:25:18",3,4,14
7,"2021-04-14 01:26:42",3,4,454
8,"2022-07-23 16:37:47",4,3,361
9,"2022-05-30 16:55:50",4,3,857
10,"2023-02-13 05:59:04",5,2,188
…
```

A sample of data in order_line.csv:

```
line_id,order_id,book_id,price
81267,1,10237,5.35
81268,2,6416,1.62
81269,3,7511,2.65
81270,4096,7081,15.69
81271,4097,164,14.39
81272,4098,6154,12.13
81273,6143,4165,1.02
81274,6144,1485,10.27
81275,6145,1857,5.88
…
```

The output should comply with the format requirement below. The columns are `order_date, num_books, num_orders, total_price` from left to right.

```
(2021,3,28)     36      15      366.9899970293045
(2022,2,22)     31      15      364.7399996519089
(2022,7,19)     29      12      340.4400019645691
(2020,9,27)     28      12      331.7300009727478
(2020,12,20)    32      13      317.3700006008148
(2021,11,17)    27      11      315.67000246047974
(2021,6,25)     33      16      314.3199996948242
…
```

Task-2. This is an advanced research task. You are asked to figure out how to develop Apache Pig User Defined Function (UDF) using Python. It is not instructed in detail in the learning materials but you can learn by studying https://pig.apache.org/docs/latest/udf.html#udfs. This UDF is designed to be used for Task-1. The purpose is to add a new column in the output. If the total_price >= 300, the value of the new column is "high value"; if 300>total_price >= 100, the value is "medium"; if 100>total_price, the value is "low value". (**10 marks**)

The output should comply with the format requirement as below. The columns are `order_date, num_books, num_orders, total_price, note` from left to right.

```
(2021,3,28)     36      15      366.9899970293045       high value
(2022,2,22)     31      15      364.7399996519089       high value
(2022,7,19)     29      12      340.4400019645691       high value
(2020,9,27)     28      12      331.7300009727478       high value
(2020,12,20)    32      13      317.3700006008148       high value
(2021,11,17)    27      11      315.67000246047974      high value
(2021,6,25)     33      16      314.3199996948242       high value
….
```

Submission

Your assignment should follow the requirements below and be submitted via Canvas > Assignment 2.
Assessment declaration: When you submit work electronically, you agree to the assessment declaration:

Format Requirements

Failure to follow the requirements incurs a 5-mark penalty for each.

- If your student ID is s1234567, then please create a zip file named s1234567_BDP_A2.zip.
  - You need to include a "README" file in the zip file. In the README, specify sufficient information on how to run your codes for each task on AWS EMR.
  - All files are in the same folder (i.e., no subfolders), and then zip the folder.
- On HDFS, the input files must be in /input/ and the output must be in /output/, as follows:
  /input/cust_order.csv
  /input/order_line.csv
  /output/task1
  /output/task2
- Besides the zip file, organize the codes of two tasks in a separate PDF file (copy & paste into a text editor and then save it as a PDF file). Submit the PDF file (so, there are two submissions, one is the zip file, and the other is the PDF file). The PDF file is for Turnitin plagiarism check.

Functional Requirements

Failure to follow the requirements incurs up to a 5-mark penalty.

- The code must include sufficient comments that can clearly explain the major logic flow of the program.

Academic integrity and plagiarism (standard warning)

Academic integrity is about the honest presentation of your academic work. It means acknowledging the work of others while developing your own insights, knowledge, and ideas. You should take extreme care that you have:

- Acknowledged words, data, diagrams, models, frameworks, and/or ideas of others you have quoted (i.e., directly copied), summarized, paraphrased, discussed, or mentioned in your assessment through the appropriate referencing methods.
- Provide a reference list of the publication details so your reader can locate the source if necessary. This includes material taken from Internet sites.

If you do not acknowledge the sources of your material, you may be accused of plagiarism because you have passed off the work and ideas of another person without appropriate referencing, as if they were your own.

RMIT University treats plagiarism as a very serious offense constituting misconduct. Plagiarism covers a variety of inappropriate behaviors, including:

- Failure to properly document a source.
- Copyright material from the internet or databases.
- Collusion between students.

For further information on our policies and procedures, please refer to
https://www.rmit.edu.au/students/student-essentials/rights-and-responsibilities/academic-integrity

Marking Guide

- Late submission results in a penalty of 10% marks for (up to) every 24 hours being late.
- If unexpected circumstances affect your ability to complete the assignment, you can apply for special consideration.
  - Requests for special consideration within 7*24 hours, please email the course coordinator directly with supporting evidence.
  - Request for special consideration of more than 7*24 hours must be via the University Special consideration: https://www.rmit.edu.au/students/student-essentials/assessment-and-exams/assessment/special-consideration.

Task 1	0 marks – script cannot run on Hadoop – no output – no field is correct according to SQL output on MySQL – completely incorrect script	1-6 marks – output has one field(s) incorrect according to SQL output on MySQL. – Output misses one field(s). – The tuples in the output are not ordered as required. – the script misses one or more necessary operators like group by, order by, join, and foreach. – Comments and readme are misleading and hard to follow.	7-9 marks – Output has all fields, and they are correct in general with obvious errors. – The script includes all necessary operators like group by, order by, join, and foreach but with obvious errors. – Comments and readme have various minor issues.	10-13 marks – Output has all fields, and they are correct in general but with minor errors. – The script includes all necessary operators like group by, order by, join, foreach but with minor errors. – Comments and readme have no obvious issues.	14-15 marks Correct output with clear, concise comments and readme.
Task 2	0 marks – script cannot run on Hadoop – no output – no field is correct according to SQL output on MySQL – No UDF defined completely incorrect script	1-3 marks – Output has one field(s) incorrect according to SQL output on MySQL. – output misses one field(s). – The tuples in the output are not ordered as required. – The script misses one or more necessary operators like group by, order by, join, and foreach. – UDF defined but used but with errors. – Comments and readme are misleading and hard to follow.	4-6 marks – Output has all fields, and they are correct in general with obvious errors. – The script includes all necessary operators like group by, order by, join, and foreach but with obvious errors. – UDF defined and used but with minor errors. – Comments and readme have various minor issues.	7-8 marks – Output has all fields, and they are correct in general but with minor errors. – The script includes all necessary operators like group by, order by, join, foreach but with minor errors. – UDF defined and used correctly. – Comments and readme have no obvious issues.	9-10 marks Correct output with clear, concise comments and readme.
Functional requirement	Failure penalty on functional requirements detailed in the specification				
Format requirement	Failure penalty on format requirements detailed in the specification				