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

Assessment Type	– Group assignment (up to three students), individual work is allowed but not encouraged. – 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	Due at 23:59, 28th September, 2025
Marks	25

Overview
In this assignment, you will use Apache Pig on the RMIT-provided AWS EMR cluster to process large datasets in HDFS. You'll practice loading/cleaning data, joining and enriching tables, aggregating metrics, and extending Pig with a simple UDF.

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: analyse 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.

Group Assessment
Groups must be formed by **up to 3 students** who are enrolled at the same level (either all Undergraduate COSC2633 or all Postgraduate COSC2637). All members are expected to contribute equally. Unless a contribution dispute is reported (and supported by the work log), all group members will receive the same grade. Individual work is allowed but not encouraged. Please note that no bonus marks will be awarded for completing the work individually.

Group Formation and Locking Policy
- Navigate to **People** in Canvas, then select the **Groups** tab. **Scroll past the A1 groups until you reach the A2 group section.**
- Choose an empty **A2 group** and join it along with your group members.
- **You are not permitted to create new groups.** This restriction is in place because student-created groups are not linked to the grading system, which means they cannot be identified by the system during marking.
- Groups are intended only for students working collaboratively. If you are completing the assignment individually, please do NOT join a group on Canvas.
- **Groups will be locked on 28 September at 11:59 PM — the same time as the assignment deadline.** Once groups are locked, students can no longer join, leave, or modify their groups. This ensures consistency in grading and prevents last-minute changes that could affect group accountability.

Group Leadership
The first student to join a group on Canvas is assigned as the group leader by default. However, this role can be changed later within the group settings if needed. **The group leader is responsible for ensuring that the group is correctly finalised before the deadline.** This includes:

- Confirming all required files are prepared and correctly named.
- Verifying the correct members are listed in the group.
- Submitting the final work before the deadline.
- Keeping and submitting the group work log as part of the submission.

Extension Policy for Group Assignments
Any extension requests for group assignments must be approved by the Course Coordinator. Group extensions will only be considered under exceptional circumstances (**if more than 50% of the group members are unable to contribute at the same time**) and require supporting documentation. Reasons such as being unable to cope with study load or taking on additional work shifts are not considered sufficient grounds for an extension.
If you wish to apply for an individual extension, you must withdraw from your group and complete the entire assignment individually. If your individual application is approved, alternative tasks will be given only to you — not to your former group members.

What to Do If a Team Member Isn't Contributing
- Try to resolve the issue first through communication within the group.
- Document communication attempts (e.g., emails, messages). Keep a simple log of who completed which parts of the work.
- If the situation doesn't improve, you may choose to leave the group, join other groups OR complete the assignment individually.

Please notify the tutor and the course coordinator early if you decide to do so. Adjustments due to non-contributing members will not be granted unless clear evidence is provided.

Assessment Details
All inputs are on HDFS, **tab-separated**, and all filenames are **case-sensitive**.

`/Input/Trips.txt`
`trip_id (PK), taxi_id (FK), company_id (FK), dropoff_lat, dropoff_lon, distance_km, fare`

```
47   100  5    37.7551   -122.48105 12.39    38.84
86   100  2    37.74983  -122.43255 11.08    27.23
69   100  3    37.72532  -122.4935  10.36    31.8
12   101  4    37.78995  -122.422   21.99    65.92
91   101  1    37.71841             18.09    45.48
…
```

`/Input/Taxis.txt`
`taxi_id (PK), license_plate, medallion_year, driver_rating`

```
100  RMIT100   2020 4.92
101  RMIT101   2021 4.48
102  RMIT102   2018 4.73
103  RMIT103   2018 4.05
…
```

`/Input/Companies.txt`
`company_id (PK), company_name`

```
1    City Cabs
2    Metro Taxis
3    QuickRide
4    Urban Transit
5    Star Taxis
```

Task 1 – Load and Clean (5 marks = 1mark + 1 mark + 2 marks + 1 mark)
1. Load the files using `PigStorage('\t')` with schemas.
2. Clean the trips data:
   - Include only those trips where every required field contains a value.
   - Remove trips with `distance_km <=0` (as wrong data) or `distance_km >20` or `fare <5` (as outliers).
3. `Store` the cleaned trips relation to `/Output/clean_trips` (use `PigStorage('\t')`).

Task 2 – Joins and Enrichment (5 marks = 1.5 marks + 1.5 marks + 1 mark + 1 mark)

1. First, inner join the cleaned trips to the taxis table on `taxi_id` and then join that intermediate results with companies on `company_id`.
2. Produce an enriched relation with `taxi_id, company_id, company_name, driver_rating, distance_km, fare, dropoff_lat, dropoff_lon.`
3. `Store` to `/Output/enriched_trips` (use `PigStorage('\t')`).

Task 3 – Aggregation (7 marks = 5 marks + 1 mark + 1 mark)

1. Compute per (`company_id, company_name`): `trip_count, total_distance_km (2 decimals), avg_distance_km (2 decimals)`, and `avg_fare (2 decimals)`
2. `Sort` **ascending** by `trip_count`; if two companies have the same `trip_count`, break the tie by sorting on `company_name`.
3. `Store` to /`Output/company_stats` (use `PigStorage('\t')`).

Combiner-friendly tip**:** Use GROUP and FOREACH ... GENERATE with SUM, COUNT, AVG.
A sample output is:

```
3     QuickRide       12    118.87     9.91       26.21
2     Metro Taxis     13    159.15     12.24      33.33
5     Star Taxis      14    142.68     10.19      26.36
4     Urban Transit   14    129.93     9.28       24.79
1     City Cabs       15    193.19     12.88      34.27
```

Task 4 – UDF for Fare Binning (8 marks)

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.
Write a UDF (in Python) `fare_band.py` that maps `fare` to: `LOW (<=15), MID (>15 and <=30)`, and `HIGH (>30).`

Undergraduate Requirement:

(2 marks for grouping + 3 marks for correct UDF + 3 marks for aggregation)

Group the data by `company_id`, `company_name` and `fare_band`, then compute the number of records in each group as count. A sample output is:

```
1     City Cabs       MID   6
1     City Cabs       HIGH  9
2     Metro Taxis     LOW   1
2     Metro Taxis     MID   7
2     Metro Taxis     HIGH  5
3     QuickRide       LOW   4
3     QuickRide       MID   2
3     QuickRide       HIGH  6
4     Urban Transit   LOW   5
4     Urban Transit   MID   2
4     Urban Transit   HIGH  7
5     Star Taxis      LOW   4
5     Star Taxis      MID   4
5     Star Taxis      HIGH  6
```

Postgraduate Requirement:

(2 marks for grouping + 2 marks for correct UDF + 4 marks for aggregation)

Aggregate by `company_id` and `company_name`, producing three fields — `low_count, mid_count`, and `high_count` — that tally the number of trips in each fare band. This is a more advanced task, it demands a deep understating of Pig UDF and aggregation mechanics. A sample output is:

```
1     City Cabs       0     6     9
2     Metro Taxis     1     7     5
3     QuickRide       4     2     6
4     Urban Transit   5     2     7
5     Star Taxis      4     4     6
```

Both undergraduate and postgraduate should `Store` to `/Output/fare_bands_by_company` (use `PigStorage('\t')`). Expect different outputs based on the respective requirements.

How We Will Run Your Work

You are provided a single Bash shell script named `run.sh` that runs the job (with all tasks in `a2.pig`) end-to-end without manual intervention. It should produce all required outputs in the specified HDFS locations. Marks will be awarded based on the accuracy of your outputs for each task. Example `run.sh`:

```
#!/bin/bash

hadoop fs -rm -r -f /Output/clean_trips || true
hadoop fs -rm -r -f /Output/enriched_trips || true
hadoop fs -rm -r -f /Output/company_stats || true
hadoop fs -rm -r -f /Output/fare_bands_by_company || true

pig -x mapreduce a2.pig

hadoop fs -cat /Output/clean_trips/part* > t1_output.txt
hadoop fs -cat /Output/enriched_trips/part* > t2_output.txt
hadoop fs -cat /Output/company_stats/part* > t3_output.txt
hadoop fs -cat /Output/fare_bands_by_company/part* > t4_output.txt
```

You must name your HDFS input directory, HDFS output directory, your Pig script, and your UDF file exactly as specified accordingly, failure to do so results in 0 marks.

Functional Requirements

Failure to follow the requirements incurs up to 5 marks penalty.

– The code must be well written using a good coding style.
– The codes and scripts must come with concise and clear comments to explain the logical flow of the program.

Submission Instructions

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

Failure to follow the format requirements incurs up to 10 marks penalty.

1. Prepare your zip file

- Place all required files into a single folder, including your pig file (`a2.pig`), your UDF file (`fare_band.py`) and the MS Word work log.
- No subfolders.
- Compress the folder into a zip file.

2. Prepare a pdf file for Turnitin plagiarism check

- Copy and paste the **pig code** and **UDF python code** into a plain text editor.
- Save this as a **pdf file**.
- **Submit the pdf file separately from the zip file**. This pdf must be uploaded as a separate file to ensure it is properly processed by Turnitin for plagiarism checking.

3. Name your zip file and pdf file using the following format:

`<GroupLeaderStudentID>_Group<GroupNumber>_<UG/PG>.zip`
`<GroupLeaderStudentID>_Group<GroupNumber>_<UG/PG>.pdf`

Replace:

- `GroupLeaderStudentID` with the student number of the group leader
- `GroupNumber` with your group number (e.g., 3)
- `PG` if you are in a postgraduate program; `UG` if you are in an undergraduate program

Example: `s1234567_Group3_UG.zip` and `s1234567_Group3_UG.pdf`

The group leader is responsible for submitting the group’s final zip file. Duplicate submissions from the same group will not be accepted.

If you are completing A1 individually, name your zip file using the following format:

```
<StudentID>_<UG/PG>.zip
<StudentID>_<UG/PG>.pdf
```

Replace:

`StudentID` with your actual student number

`PG` if you are in a postgraduate program; `UG` if you are in an undergraduate program

Example: `s1234567_UG.zip` and `s1234567_UG.pdf`

Important

- You must submit **both** the zip file and the pdf file.
- **Failure to submit the required PDF file separately from the zip file will result in a 10% penalty (2.5 marks).**
- Submissions **without the pdf file** will incur a **10% penalty (2.5 marks).**
- Submissions where the PDF **fails the Turnitin plagiarism check**, **will not be graded**.
- Please verify your submission carefully before the deadline.
- If you submit the **wrong file**, a **corrupted zip file**, or a file that **does not meet the required structure**, **you will not be given a second chance to resubmit after the deadline.** Such submissions may result in a **zero grade** for the affected components.
- Work must run on the **RMIT-provided EMR** cluster. Personal AWS resources are **not allowed**.
- Markers will **not modify your code**.

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 behaviours, 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

- **Marks will be awarded based on output correctness of each task.**
- 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