# Hadoop – Assignment 1 (Steam on Hive)

Name: Mohammed Al-selmi

Student number: 669668

Github URL for the assignment: https://github.com/mm55355/Hadoop-Assignments

## Steps:

1. Before starting with assignment, I did some research on Hive in order to get an idea of Hive and how can I work with it.
2. After that, I downloaded the Steam dataset from moodle, and uploaded the dataset as a table in Hive, and of course I opened the CSV file before uploading it just to get a rough idea about the data in type of columns were there.
3. Before creating the table, I made sure that I selected the proper field delimiter, escape character, quote character, and removing the first row since it was a header. Additionally, I checked each column data type to make sure that every column has the correct data type.
4. After creating the table, I used the SQL window in Hive view to write the SQL queries for the assignment.

## Explanation of source code:

### Part 1:

Code:

A picture containing text, screenshot, font, line

Description automatically generated

The above SQL query selects the 'ResponseName' and 'RecommendationCount' columns from the table and returns them as 'name' and 'count', respectively. It then sorts the results by 'count' in descending order, so that the games with the highest recommendation count are displayed first. The result is a list of games and their corresponding recommendation counts, ordered from highest to lowest. The query resulted in the following:

A screenshot of a game

Description automatically generated with medium confidence

### Part 2:

Code:

A picture containing text, screenshot, font, line

Description automatically generated

The above SQL query selects the 'QueryName' and 'ScreenshotCount' columns from the 'steam\_games\_features' table and returns them as 'name' and 'screenshot', respectively. It then filters the results to only include games that are marked as free 'isFree=true' and have more than 10 screenshots 'screenshotcount>10'. The filtered results are then sorted by 'ScreenshotCount' in descending order using 'ORDER BY screenshot DESC'. The final output is a list of game names ('name') and their corresponding screenshot counts 'screenshot', ordered by the number of screenshots in descending order. The query resulted in the following:

A screenshot of a computer

Description automatically generated with low confidence

### Part 3:

Code:

A picture containing text, screenshot, font, number

Description automatically generated

The above SQL query retrieves information from the table. It calculates the count of each game genre based on the platform it is available on. It groups the results by the platform name ('Windows', 'Linux', 'Mac', and 'Other'), and counts the number of games that belong to each genre for that platform. The results are returned as a table with columns for platform name and the count of games in each genre for that platform. The query uses a CASE statement to determine the platform name based on the boolean values for the 'PlatformWindows', 'PlatformLinux', and 'PlatformMac' columns. It then uses COUNT and CASE statements to calculate the number of games in each genre for each platform. Finally, the GROUP BY clause is used to group the results by platform name. The query resulted in the following:

A screenshot of a computer

Description automatically generated with low confidence