Project Title: GPU-Accelerated Joins on CSV Files Using CUDA

Problem and Importance:

Joining CSV tables using joins can be slow using a CPU, especially regarding large datasets that require numerous operations. With this project, I aim to leverage GPU parallelism using CUDA to speed up the process of performing "SQL-like" joins on CSV files, focusing on efficiency for large-scale data. Naturally, fast joins are critical for various data processing tasks in analytics, reporting, and even ETL (extract, transform, load) pipelines. GPUs are essential to accelerate this process and handle much larger datasets than a CPU can handle in a reasonable time.

Ultimately, showing success at this CSV level, we can later abstract to the SQL level itself in a MySQL or PostgreSQL database.

Solution:

I will create a C++ program that inputs two CSV tables, each table containing a column that will act as the join key (similar to a SQL join). The program will then

1. Load the CSV data into memory,
2. Use CUDA operations to parallel join the data, and
3. Return/output a new CSV file that contains the joined rows based on the combined ID column.

Goals:

75% Goal: A basic INNER JOIN will be implemented using CUDA to join two CSV tables on a common key column.

100% Goal: This implementation will be extended to support both LEFT and RIGHT JOINS using CUDA, where the LEFT JOIN will include all rows from the left CSV file, with NULLs where there's no match in the right file (and vice versa for the RIGHT JOIN).

125% Goal: This implementation will support full OUTER JOIN operations or complex multi-join operations that support joining 3+ CSV tables at once.

Validation:

We can verify speed and correctness by comparing the CUDA-based join vs a standard C++ CPU-based join, measuring the speedup achieved by using the CUDA-based join. We can also verify that the output is the same for both CUDA and CPU, matching the expected output (we will have to account for edge cases like missing data).

We can use synthetic datasets and publicly available datasets to test these joins.

Resources:

C++ with Cuda
Parsing Library (like pandas) or basic C++
GPU access (can use PACE cluster for this: L40S or H100)
Resources from CS 6422 and CS 8803 (GPU), both of which I am taking
Public Datasets for testing (for example CSVs from Kaggle)