In this assignment, you will explore the challenges of implementing a data warehouse and develop solutions to address them. You will start by creating a simple data warehouse backed by a single CSV file, then design a more advanced version that partitions data across multiple files with optimizations of your choice. Your implementations will help you understand key issues like data storage, query efficiency, and scalability.

Here are the steps:

1. Review the abstract class DataWarehouse and its four methods: `add_data`, `update_data`, `delete_data`, and `query_data`. Understand the expected behavior and input format (a map from column names to values).
2. Implement a baseline data warehouse in `csv_warehouse.py` that stores all data in a single CSV file. Each method should read from or write to this file as needed.
3. Test your baseline implementation using the provided `test_harness.py`. Run queries and measure performance to establish a baseline.
4. Design a second data warehouse implementation in `my_data_warehouse.py` that partitions data across multiple files. Collaborate with your group to brainstorm ideas. Consider:
    - How to split data (e.g., by row ranges, columns, or other criteria)
    - File formats (e.g., CSV, Parquet) and compression options
    - Strategies to avoid scanning all partitions for every query
    - Efficient handling of repeated or overlapping queries
5. Implement your design and test it with the test harness. Compare its performance and storage use to the baseline.
6. Write a clear docstring for your implementation explaining your design decisions and their motivations.

As a reminder, you may use AI tools to assist but must grasp the solutions you create. Please submit your files to Canvas when you are finished.