2. Documentation

2.1. Design Choices

- **Schemas:** Separated into Dimension for dimensional data and Fact for transactional data.
- **Dimension Tables:** Date and Customer store descriptive information used for analysis.
- **Fact Table:** Orders captures transactional data and links to dimension tables via foreign keys.
- **Indexed View:** OrdersByDate aggregates order data by year and month to enhance query performance.

2.2. Relationships Between Tables

- Fact.Orders is related to Dimension.Date and Dimension.Customer via DateKey and CustomerKey, respectively.
- Foreign keys ensure referential integrity between fact and dimension tables.

2.3. Purpose of the Indexed View

- **Efficiency:** OrdersByDate precomputes and stores aggregated data (order counts and totals) to reduce query computation time.
- **Optimization:** Indexed view improves performance for queries that aggregate orders by year and month.

4. Execution Time Comparison

To demonstrate performance benefits:

1. Run the "Without Indexed View" query and note the execution time and resource usage.

2. Run the "With Indexed View" query and compare the execution time and resource usage.

3. Analyze Execution Plans: Use SQL Server Management Studio to view and compare the execution plans for both queries to observe how the indexed view impacts query performance.

By running these tests and analyzing the results, you can clearly demonstrate the performance improvements gained from using the indexed view.