Data Analytics 1
Assignment 2

Release: 26 August 2025
Deadline: 12 September 2025 (11:55 pm)

Part 1 [30 Marks]

This part is designed to provide you with hands-on experience in data warehousing fundamentals.

Data warehouse: (Choose any one)

- DuckDB
- ClickHouse

Task 1: Schema Design [12 Marks]

- Create a Star Schema for the dataset (by identifying the fact table(s) and dimension tables).
- **Submit:**
  - ER diagrams of Star schemas.
  - SQL scripts for creating the schema in your chosen data warehouse.

Task 2: Data Loading [5 Marks]

- Load data into your designed schemas (fact and dimension tables).
- Ensure keys and relationships are consistent.

Task 3: Queries & OLAP Analysis [8 Marks]

Write SQL queries to answer the following analytical questions. Use `ROLLUP`, `CUBE`, `GROUPING SETS`, drill-down, and roll-up where appropriate.

1. Compute total sales revenue per year, then drill down to quarter and month.
2. Identify the month with the highest sales in each year.
3. Which product categories generate the most revenue?
4. Drill down from category $\rightarrow$ product to find the best-selling products.
5. Use a `CUBE` operation to show total sales aggregated by (Brand, Category, Year).
6. List the top 5 customers by total purchase amount.

7. Which staff members generate the highest sales at their store?
8. Use a `CUBE` query to compute total sales across all combinations of:
    - Product Category
    - Store
    - Year

Task 4: Report [5 Marks]

2–3 pages summarising insights from your analysis and explaining the ETL procedure. Example: "Brand X dominates sales in the Mountain Bikes category," or "Store Y performs best in California."

Deliverables

You must submit the following:

- **Schema Design:**
  - ER diagrams for Star schema.
  - SQL table creation scripts.
- **ETL Process:** Steps taken to load the raw data into your schema.
- **SQL Queries:**
  - SQL scripts for all analytical questions.
  - Query outputs (screenshots or result tables).
- **Report**

Part 2 [70 Marks]

The objective of the part is to get exposure to extract interesting CUBEs from the given large dataset based on the BUC algorithm and gain an understanding of Attribute-Oriented Induction.

Assignment Tasks

Attribute-Oriented Induction (10 marks)

Extract characteristic rules using attribute-oriented induction. This task involves data generalization through attribute removal or attribute generalization.

BUC Algorithm Implementation (25 marks)

Implement the Bottom-Up Cube (BUC) algorithm. This task has two parts:

1. **In-Memory Implementation (10 marks):** Assume that all CUBEs can be supported by the main memory and write a straightforward implementation of BUC.
2. **Out-of-Memory Implementation (15 marks):** Assume that the program will run out of main memory and introduce paging in your implementation.

Performance Analysis (5 marks)

Plot the following graphs over multiple runs of the algorithm while varying some parameters and keeping others constant:

1. A plot of minsup vs. runtime, keeping allotted memory fixed.
2. A plot of allotted memory vs. runtime, keeping minsup fixed.

Provide a brief analysis of the trends observed in these plots.

Optimization Technique (15 marks)

Propose and implement one optimization technique for the BUC algorithm (e.g., Apriori pruning, iceberg cubing). Describe the optimization and its impact on performance.

Comparison of BUC and AOI (5 marks)

Compare and contrast the Bottom-Up Cube (BUC) algorithm and Attribute-Oriented Induction (AOI). Your comparison should include:

- The primary purposes and use cases of each technique.
- The types of insights or patterns each method is best suited to discover.
- The computational efficiency and scalability of each approach.
- The interpretability of the results produced by each method.
- Scenarios where one method might be preferable over the other.

Provide concrete examples from your implementations to support your comparison.

Grading Criteria (Total: 70 marks)

- Attribute-Oriented Induction implementation and explanation: 10 marks
- BUC Algorithm Implementation:
  - In-Memory Implementation: 10 marks
  - Out-of-Memory Implementation: 15 marks
- Performance Analysis and interpretation: 5 marks
- Optimization Technique implementation and analysis: 15 marks
- Comparison of BUC and AOI: 5 marks
- Code quality and documentation: 5 marks
- Report clarity and completeness: 5 marks

Submission Guidelines

- Submit your code along with a report detailing your implementation, analysis, and findings.
- Include all plots and their interpretations in the report.
- Clearly explain the optimization technique you chose and provide benchmarks comparing the optimized version with the original implementation.
- Ensure your code is well-commented and follows good programming practices.