Assignment 1 Grading Rubric

Overview

This rubric ensures consistency between students and graders. Assignment 1 focuses on creating tables, loading the Reddit dataset, and correctly applying constraints. Grading is fully automated based on your assignment1.sh and .sql scripts.

Grading Breakdown (100 points total)

1. Database Creation & Normal Insertion – 80 points

- Table creation (40 pts):

All required tables must be created with correct names, schema, lowercase attribute names, primary keys, foreign keys, and necessary constraints.

- Data loading (20 pts):

All CSV files (authors, subreddits, submissions, comments) must load successfully into their respective tables.

  - Row counts should match the raw datasets.
- Integrity checks (20 pts):
  - Primary key and foreign key constraints must be valid.
  - Only the five relationships in the provided figure will be graded.

2. Optimized Data Insertion – 20 points

  - Efficient insertion of all entries using **COPY** or **pg_bulkload** (only one method is required).
  - To earn full credit, your loading process must complete within ~300 seconds in the grading environment.
  - Reference performance: ~100 seconds using pg_bulkload.

Testing Guidelines

Students should verify correctness by checking:
• All four required tables are created.
• Row counts for authors, subreddits, comments, and submissions match the assignment description.
• Primary key and foreign key constraints are present and enforced.
• assignment1.sh successfully executes without manual intervention.
• assignment1.sh calls the required .sql files but does not contain database creation.

Submission Requirements

• Submit a zip file named Assignment-1.zip containing:
 - **assignment1.sh** (entry point script)
 - **.sql** file(s)
 - **README.md** (optional notes to graders)
• assignment1.sh must:
 - Call the .sql file to create the table(s)
 - Load the data using either:

- COPY (from an additional .sql file), **or**
- pg_bulkload

• Assume CSV files are already in the same folder (./filename.csv)

Policies & Notes

• No late submissions unless under documented emergency.
• Plagiarism will result in course failure; anti-plagiarism tools will be used.
• The grading environment uses **PostgreSQL 14** with **pg_bulkload pre-installed**.
• Scripts are executed under the postgres user with database name 'postgres'.
• For reference, we have attached a screenshot of the autograde output.
• After discussion with the professor, the autograding scripts themselves will not be released.

```
chang@chang-pc:/data/cse511/assignment1$ ./autograde.sh
=== Assignment 1 Autograding Script ===
Total Points: 100
Time Limit for Optimization: 300s

=== Pre-execution Cleanup ===
Cleaned up existing tables
=== Executing Student Script ===
Student script executed in 82s
Exit code: 0

=== Table Creation Tests (40 points) ===
✓ Table 'authors' exists: PASS (10/10 points)
✓ Table 'subreddits' exists: PASS (10/10 points)
✓ Table 'submissions' exists: PASS (10/10 points)
✓ Table 'comments' exists: PASS (10/10 points)

=== Data Loading Tests (20 points) ===
✓ Authors row count (6158212, expected 6158212±1): PASS (5/5 points)
✓ Subreddits row count (914067, expected 914067±1): PASS (5/5 points)
✓ Submissions row count (1263937, expected 1263936±1): PASS (5/5 points)
✓ Comments row count (10557466, expected 10557466±1): PASS (5/5 points)

=== Integrity Tests (20 points) ===
✓ submissions.author → authors.name: PASS (4/4 points)
✓ submissions.subreddit_id → subreddits.name: PASS (4/4 points)
✓ comments.author → authors.name: PASS (4/4 points)
✓ comments.subreddit_id → subreddits.name: PASS (4/4 points)
✓ comments.subreddit → subreddits.display_name: PASS (4/4 points)

=== Optimization Tests (20 points) ===
✓ Execution time (82s ≤ 300s): PASS (20/20 points)

=== Final Grade Summary ===
Table Creation:     40/40
Data Loading:       20/20
Integrity Checks:   20/20
Optimization:       20/20
---------------------------------------
Total Score:        100/100
Percentage:         100%
Grade: A

=== Additional Information ===
Execution time: 82s
Script output saved to: execution_output.log
chang@chang-pc:/data/cse511/assignment1$ vim ./autograde.sh
chang@chang-pc:/data/cse511/assignment1$
```