Assignment 1: Database Creation & Loading & Query

CSE 511: Data Processing at Scale - Fall 2025

Available: 09/02/2025 **Due Date**: 09/15/2025 11:59 PM **Points: 100**

Introduction & Background

In this assignment, you'll learn to load large datasets into a database from scratch, and later, build applications on top of it. You'll also explore how to make various queries to these database tables. For this task, we will use a research-focused Reddit dataset.

Reddit, often called the "front page of the Internet," has been the subject of extensive scientific research. While Reddit is more open to data collection than other platforms like Facebook and Twitter, acquiring and analyzing its vast data—billions of comments, millions of subreddits, and hundreds of millions of users—remains a technical challenge. The Reddit Pushshift dataset addresses these challenges. This dataset, curated and updated in real-time by Pushshift, provides historical Reddit data from its inception. It is widely used in educational and research contexts.

Dataset Description

All the submissions and comments posted on Reddit between June 2005 and April 2019 are accessible through **Pushshift**. The original dataset comprises **2,611,778,198 submissions 5,601,331,385 comments, and 2,888,885 subreddits.**

Important Note:

1. In the scope of this assignment and future assignments, we will only use a portion of the Pushshift dataset which we have carefully curated for the students. The original dataset was stripped down, decreasing the number of columns in the dataset such that it is easier to manage. You can find the stripped-down files which you **MUST** use to complete the assignment here**.** Also, the sample entries of the multiple dataset files only highlight the available columns after the dataset is stripped and cleaned. This is NOT the original dataset. **Students MUST use only the link provided to the specific files of the dataset for the assignment and NOT the original dataset.**
2. The sample entries provided with each file are only for understanding of the columns in the dataset. Due to space limitations, some of the textual data may be cut down.

The stripped down Pushshift Reddit dataset is made up of multiple files and for ease of access we have created .csv for the same.

1. **submissions:**
    a. Below is a sample entry form “submission”, showing all the available columns and their sample values.
    b. Link to the submissions file. Total number of entries: `1,263,936`

```
"downs": 1,
"url": http://thinkprogress.org/economy/2011/06/29/some_url",
"id": "iemqy",
"edited": "False",
"num_reports": "",
"created_utc": 1309564792,
"name": "t3_iemqy",
"title": "A somewhat long title with some political discussion",
"author": "[deleted]",
"permalink": "/r/politics/comments/iemqy/another_string/",
"num_comments": 1,
"likes": "",
"subreddit_id": "t5_2cneq",
"ups": 3
```

2. **comments:**
    a. Below is a sample entry form “comments”, showing all the available columns and their sample values.
    b. Link to the comments file. Total number of entries: `10,557,466`

```
"distinguished": "",
"downs": 0,
"created_utc": 1309478400,
"controversiality": 0,
"edited": "False",
"gilded": 0,
"author_flair_css_class": "mordekaiser",
"id": "c22x4aq",
"author": "username",
"retrieved_on": 1427302516,
"score_hidden": "False",
"subreddit_id": "t5_2rfxx",
"score": 1,
"name": "t1_c22x4aq",
"author_flair_text": "[username] (NA)",
"link_id": "t3_id1nc",
"archived": "True",

"ups": 1,
"parent_id": "t3_id1nc",
"subreddit": "leagueoflegends",
"body": "Good lord. Yes."
```

3. **authors:**
   a. Below is a sample entry form “authors”, showing all the available columns and their sample values.
   b. Link to the authors file Total number of entries: `6,158,212`

```
"id": "t2_1rr1",
"retrieved_on": 1532086586,
"name": "duncan",

"created_utc": 1298437200,

"link_karma": 1029,
"comment_karma": 9943,

"profile_img": "https://www.redditstatic.com/avatars/avatar_default_02_25B79F.png",
"profile_color": "",
"profile_over_18": "False"
```

4. **subreddits**:
   a. Below is a sample entry form “subreddit”, showing all the available columns and their sample values.
   b. Link to the subreddits file Total number of entries: `914,067`

```
"banner_background_image": "",
"created_utc": 1137700161,
"description": "A very very long description of what the subreddit is about. SHown maybe

only to the members?", "display_name":
"John Doe",
"header_img": "https://b.thumbs.redditmedia.com/h5RmvyztneDL1.png",

"hide_ads": "False",
"id": "vf2",
"over_18": "True",

"public_description": "Still a description but this one shown to people not part of the
subreddit",
"retrieved_utc": 1591839904,

"name": "t5_vf2",

"subreddit_type": "public",

"subscribers": 1880887,

"title": "Another Random Subreddit",
"whitelist_status": "all"
```

Please find the corresponding diagram showing the relations across the tables for the stripped down Pushshift dataset.

Problem Statement

I. Considering the **four** tables: **submissions, comments, authors, and subreddits** Your task is to **create tables** and **load** the corresponding data to the table.

II. You need to figure out the primary keys, foreign keys, constraints, or other necessary settings by yourself. The key information in the requirement is not complete and attributes can be primary keys and foreign keys at the same time.

III. You need to implement the following 5 SQL queries:

1. `query1`: Write a SQL query to return the **total number of comments authored by the user `xymemez`.**
    a. Your column names MUST be: '`count of comments`'

2. `query2`: Write a SQL query to return the **total number of subreddits for each subreddit type**.
    a. Your column names MUST be: '`subreddit type`', '`subreddit count`'

3. `query3`: Write a SQL query to return the **top 10 subreddits arranged by the number of comments**. Calculate the average score for each of these subreddits and round it to 2 decimal places.
    a. Your column names MUST be: '`name`', '`comments count`', '`average score`'

4. `query4`: Write a SQL query to **print** `name`, `link_karma`, `comment_karma` **for users with >1,000,000 average karma in descending order**. Additionally, also have a column '`label`' which shows 1 if the `link_karma` >= `comment_karma`, else 0
    a. Your column names MUST be: '`name`', '`link karma`', '`comment karma`', '`label`'
    b. You can write this query with both having and where clauses (both will be considered correct and submit only one), however, try doing both just to see the speed difference. (if you do try it) let us know the results in the README file along with your theory for why!
    c. To fairly compare times between 2 queries, you need to clear the postgres cache! A helpful link: See and clear Postgres caches/buffers? - Stack Overflow

5. `query5`: Write a SQL query to give **count of comments in subreddit types where the user has commented**. Write this query for the user `[deleted_user]`
    a. Your column names MUST be: ‘sr type’, ‘comments num’

Above all, your script **MUST** generate **five tables**, namely, “query1”, “query2”, …, “query5” respectively for each query.

Note

- Your submitted code should include SQL commands for **creating tables, defining relationships, and writing queries**. However, it should **not** include commands for database creation, switching databases, or setting encoding.
- All table names and attribute names **must** be in lowercase letters and exactly the same as mentioned.
- You are free to create any other temp/permanent views, temp/permanent tables or functions to help your queries.
- You should use the following command to save your query result to a table.

  ```
  CREATE TABLE query0 AS YOUR SQL STATEMENT
  ```

  For instance, select the user from the users table which has userID = v1 and store it in query0 and rename the “username” column to “userfullname”.

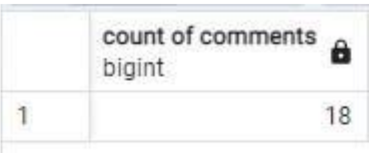
  CREATE TABLE query0 AS SELECT username AS userfullname FROM users WHERE users.userid = :v1

Grading

- The assignment will be graded using automated scripts.
- 100 points of the assignment is divided into 3 categories
    - **40%** for Database creation, normal insertion (i.e. without any optimization) and correctly assigning constraints.
    - **10%** for the optimized data insertion.
    - We have optimized the code using pg_bulkload, but you are free to also explore any other optimization techniques if interested. Please find some references below on how the data insertion can be optimized.
        - dbi Blog (dbi-services.com)

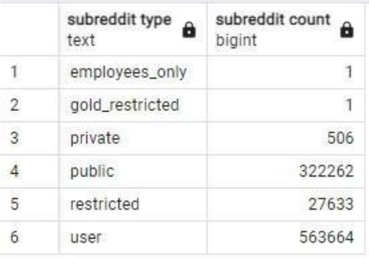
        - http://ossc-db.github.io/pg_bulkload/pg_bulkload.html
        - https://www.postgresql.org/docs/current/populate.html
        - sql - How to speed up insertion performance in PostgreSQL - Stack Overflow
        - 4.pg_bulkload Data Loading Use and Example - www.cqdba.cn - Blog Park (cnblogs.com)

        Note: The threshold to get full 10% grade points in the optimization test case is **~300 seconds** to insert all the entries. For example, Our optimized code performs the entire problem statement in ~**220 seconds**.
    - **50%** of the assignments are equally divided into **five** queries
    - To help you along the way, please find the sample output ONLY for query 1 and query 2
        - Query 1:



	count of comments bigint
1	18

        - Query 2



	subreddit type text	subreddit count bigint
1	employees_only	1
2	gold_restricted	1
3	private	506
4	public	322262
5	restricted	27633
6	user	563664

- You **MUST** provide **assignment1.sh** which will be responsible for the following:
    - We have optimized the code using pg_bulkload, but you are free to also explore any other optimization techniques if interested. In case you choose to use any additional program, your .sh script **MUST** install any third party and necessary libraries which you are using. The grading will be automated hence we won't install any other dependencies explicitly to grade, you MUST mention it as a part of your .sh script.
    - The assignment1.sh **MUST** call any required .sql file(s) which should **NOT** contain the commands to create a database, change database or set encoding. Please test your code, you will receive 0 grade points if your code crashes the grading environment!
    - The automated grading scripts will **only** run the .sh file from your submissions to test your code.
- You **Must** name your .sql file as **create_tables.sql, create_relations.sql, and queries.sql**

- You can test your code with a subset of the data but the grading scripts will use the entire mentioned dataset.

Submission Requirements & Guidelines

Assignment 1 is due on **09/15/2025 11:59 PM**. Submit the assignment following the below guidelines

1. This is an **individual** assignment which will be submitted as a zip file
2. Naming nomenclature:
   a. You **MUST** name your .sh file as **assignment1.sh**
   b. You **Must** name your .sql file as **create_tables.sql, create_relations.sql, and queries.sql**
   c. Name the zip file following the naming convention. **"Assignment-1.zip"** to submit on Canvas including **both .sh and .sql files**.
   d. README.md in case you want to mention anything to the grading team.

Submission Policies

1. Late submissions will ***absolutely not*** be graded (unless you have verifiable proof of emergency). It is much better to submit partial work on time and get partial credit for your work than to submit late for no credit.
2. Every student needs to ***work independently*** on this exercise. We encourage high-level discussions among students to help each other understand the concepts and principles. However, a code-level discussion is prohibited, and plagiarism will directly lead to failure of this course. We will use anti-plagiarism tools to detect violations of this policy.

Common Questions

1. Does the time constraint mentioned in the assignment (300 seconds) include the time required to install the libraries or just the time required for the optimized insertion (20%)?
   The 300 seconds include table insertion, data insertion and constraint creation only.
   a) To satisfy the 300s limit for data insertion, the speed would also depend on the system correct ? So what happens if it runs for me in my system under 300 seconds but exceeds that during evaluation ?
   As long as your script runs within 300 seconds on your system, you will be fine.

2. For the 4th query given in the assignment, based on what columns do we have to order it?

By average karma in descending order. If two users have the same average karma, sort them alphabetically by name.

3. The grading criteria say 40% for data insertion and 10% of optimized insertion. This doesn't mean we provide two separate insertion techniques, right? If we can get the optimized version, we need to have only that in the bash file, right?

You only need the optimized version in your bash script. A separate non-optimized insertion is not required.

4. Can you please shed some more light on create_relations.sql file.

Define how tables interact with each other in the database. Define primary keys, foreign keys, constraints, and indexing to establish relationships between tables.

Primary Keys: Ensure each table has a unique identifier by specifying a primary key.
Foreign Keys: Establish links between tables using FOREIGN KEY constraints.
Constraints: Constraints help enforce referential integrity and ensure data consistency like NOT NULL (ensures a column cannot have NULL values), UNIQUE (ensures column values are unique), ON DELETE CASCADE (deletes dependent records when a referenced record is deleted).

5. Are we supposed to create a new database? Or should we work on the postgres database which is created during installation?

You should work on the postgres database. Do not create a new database. Your .sh script should connect to the existing postgres database and execute your SQL scripts within it.

6. Since we can only submit three SQL files (creating_tables.sql, creating_relations.sql, and queries.sql) along with the .sh file, where should the data loading code go? It seems logical to include it in creating_tables.sql, but I want to confirm if this is the correct approach.

You are required to submit **four files**:

creating_tables.sql – Defines the table structures

creating_relations.sql – Establishes relationships between tables

queries.sql – Contains the queries

assignment1.sh – Executes the above SQL files and handles data loading

The **.sh file should include the data loading process** if you are using pg_bulkload. Do not place data loading in creating_tables.sql; instead, handle it in assignment1.sh.

7. I have created **four different control files** to optimize my data insertion process using pg_bulkload. These files are necessary for the data insertion to work, and they are referenced in my .sh script.

Would it be okay to upload these control files to GitHub? Do I need to make any changes to ensure compatibility?

That's fine since we will directly run the .sh file and check the results. No changes are required. Just ensure that all required files are included in the repository so the script runs without issues.

8. Could you please clarify whether the `over_18` column in `subreddits.csv` should be `over18` (without an underscore) or `over_18` (with an underscore)? The assignment document and Pushshift image use `over_18`, but the CSV file has `over18`, which is causing errors. Should we adjust our code to match the CSV file, or will the CSV be updated to align with the assignment document?

Match the CSV file. Adjust your code to use over18 instead of over_18.

9. Are commands for installing PostgreSQL/pg_bulkload to be written in the .sh file?

The grading machines will have the following tools already set up:

(1) psql
(2) pg_bulkload

Additionally, the scripts will be run after being logged into the postgres user. You do not need to write the switch user command. However, you will have to write the command to correctly access the correct database from the correct database user

10. What path to have for the csv files?

The same folder as your .sh script: ./filename.csv

11. What are the details of the database?

PostgreSQL v14 Configurations:

- username: postgres
- password: postgres
- Database Name: postgres
- Database IP: 127.0.0.1:5432

Additional tips:

- To fix **unterminated quoted fields** in pg_bulkload use the below syntax for the Quote QUOTE = "\""
- Regarding Query 3, where we need to calculate the average score for subreddits. However, the score data type is stored as TEXT, which prevents direct numerical calculations. - turn its type to numeric by sql command, like AVG(score)::numeric

*~ The crux of the assignment is to **gain the ability to ingest a (somewhat) large amount of data** and **to understand how that is performed**! Make sure that you **focus your time on that aspect** ~*