DAS 839 NoSQL Systems

Assignment-1

Due date: 14/Feb/2024

Instructions for Submission:

- Prepare a concise and well-structured document explaining the solution. Clarity and readability are essential.
- Submit the required code in a separate folder.
- Include clear instructions on how to execute the code.
- Ensure timely submission. Details regarding late submission penalties are available on LMS.
- Assignments found to be plagiarized or inauthentic will receive no marks.
- Submit a zip folder containing the source code (excluding the dataset) and the report. The report should include relevant screenshots and a brief explanation of the logic behind the SQL query.

Section A (2 Points each)

The problems in this section are designed to test your understanding and help you refine your SQL skills.

Problem 1

The latest scores from the Japanese Baseball League are in the table with schema:

$$\text{Scores}(\text{Team}, \text{Opponent}, \text{RunsFor}, \text{RunsAgainst})$$

The data in this table is as follows:

Team	Opponent	RunsFor	RunsAgainst
Dragons	Tigers	5	3
Carp	Swallows	4	6
Bay Stars	Giants	2	1
Marines	Hawks	5	3
Ham Fighters	Buffaloes	1	6
Lions	Golden Eagles	8	12
Tigers	Dragons	3	5
Swallows	Carp	6	4
Giants	Bay Stars	1	2
Hawks	Marines	3	5
Buffaloes	Ham Fighters	6	1
Golden Eagles	Lions	12	8

What is the result of executing on this data the query:

```
SELECT Team
FROM Scores
WHERE RunsFor > RunsAgainst AND
      RunsFor <= RunsAgainst + 2
```

Identify in the list below the team that appears in the output:

a) Marines

b) Golden Eagles

c) Ham Fighters

d) Giants

Problem 2

The table:

Scores(Team, Day, Opponent, Runs)

Gives the scores in the Japanese Baseball League for two consecutive days. The **Opponent** is `NULL` if the **Team** did not play on that day. The number of **Runs** is given as `NULL` if either the team did not play or will play on that day but the game is not yet concluded. The data in this table is as follows:

Team	Day	Opponent	Runs
Dragons	Sunday	Swallows	4
Tigers	Sunday	Bay Stars	9
Carp	Sunday	NULL	NULL
Swallows	Sunday	Dragons	7
Bay Stars	Sunday	Tigers	2
Giants	Sunday	NULL	NULL
Dragons	Monday	Carp	NULL
Tigers	Monday	NULL	NULL
Carp	Monday	Dragons	NULL
Swallows	Monday	Giants	0
Bay Stars	Monday	NULL	NULL
Giants	Monday	Swallows	5

What is the result of executing on this data the query:

```
SELECT Team, Day
FROM Scores
WHERE Opponent IS NULL OR
      NOT (Runs >= 0)
```

Identify, in the list below, a row of the result:

a) Giants, Monday

b) Swallows, Sunday

c) Dragons, Monday

d) Bay Stars, Monday

Problem 3

The table:

Scores(Team, Day, Opponent, Runs)

Gives the scores in the Japanese Baseball League for two consecutive days. The **Opponent** is `NULL` if the **Team** did not play on that day. The number of **Runs** is given as `NULL` if either the team did not play or the game is not yet concluded. The data in this table is as follows:

Team	Day	Opponent	Runs
Dragons	Sunday	Swallows	4
Tigers	Sunday	Bay Stars	9
Carp	Sunday	NULL	NULL
Swallows	Sunday	Dragons	7
Bay Stars	Sunday	Tigers	2
Giants	Sunday	NULL	NULL
Dragons	Monday	Carp	NULL
Tigers	Monday	NULL	NULL
Carp	Monday	Dragons	NULL
Swallows	Monday	Giants	0
Bay Stars	Monday	NULL	NULL
Giants	Monday	Swallows	5

What is the result of the following query?

```
SELECT Team, Min(Opponent), Max(Runs)
FROM Scores
GROUP BY Team
```

Note: When a column has a string type, `Min` and `Max` refer to lexicographic (alphabetical) order of strings. Identify in the list below one of the tuples in the result:

a) Bay Stars, Tigers, 2

b) Giants, `NULL`, 5

c) Swallows, Dragons, 0

d) Dragons, Swallows, 4

Problem 4

The table contains the following 12 rows:

Team	**Day**	**Opponent**	**Runs**
Dragons	Sunday	Swallows	4
Tigers	Sunday	Bay Stars	9
Carp	Sunday	`NULL`	`NULL`
Swallows	Sunday	Dragons	7
Bay Stars	Sunday	Tigers	2
Giants	Sunday	`NULL`	`NULL`
Dragons	Monday	Carp	6
Tigers	Monday	`NULL`	5
Carp	Monday	Dragons	`NULL`
Swallows	Monday	Giants	0
Bay Stars	Monday	`NULL`	7
Giants	Monday	Swallows	5

What is the result of the query:

```
SELECT *
FROM Scores
ORDER BY Runs DESC, Team ASC
```

Identify, in the list below, a tuple and the correct order in which it appears:

a) Giants, Sunday, `NULL`, `NULL` could appear seventh through twelfth.

b) Giants, Sunday, `NULL`, `NULL` must appear second.

c) Giants, Sunday, `NULL`, `NULL` must appear fifth.

d) Carp, Monday, Dragons, `NULL` could appear eighth or ninth.

Problem 5

The table contains the following 12 rows:

Team	Day	Opponent	Runs
Dragons	Sunday	Swallows	4
Tigers	Sunday	Bay Stars	9
Carp	Sunday	Giants	2
Swallows	Sunday	Dragons	7
Bay Stars	Sunday	Tigers	2
Giants	Sunday	Carp	4
Dragons	Monday	Carp	6
Tigers	Monday	Bay Stars	5
Carp	Monday	Dragons	3
Swallows	Monday	Giants	0
Bay Stars	Monday	Tigers	7
Giants	Monday	Swallows	5

What is the result of the query:

```
SELECT S1.Team, S2.Team
FROM Scores S1, Scores S2
WHERE S1.Opponent = S2.Opponent
    AND S1.Team <> S2.Team
```

Identify, in the list below, a tuple of the result:

a) Tigers, Tigers

b) Giants, Giants

c) Dragons, Giants

d) Bay Stars, Tigers

Section B

For this section, we will use the PostgreSQL open-source database system for loading and querying the prepared dumps of the Wikipedia articles provided on LMS.

- Download and install the **PostgreSQL** (version 14.6 or above) database server by choosing the respective installer binaries for your operating system from: `https://www.postgresql.org/download/`
- PostgreSQL provides a graphical user interface (GUI) for its installation under all operating systems. Simply follow the installation steps provided by this interface and make sure that you remember your admin password for the provided example database instance. This example database instance is called `postgres` by default and can be accessed by using your password from either the admin GUI (`pgAdmin 4`) or the command-line shell (`psql`).
- See also Chapter 2 of the "Seven Databases in Seven Weeks" book for useful hints about installing and administrating PostgreSQL (if needed).

The goal of this exercise is to develop an understanding for what the technical challenges in processing queries over a large collection of text documents are. We will practice this by designing a simple search engine for Wikipedia. For this purpose, the two files `Wikipedia-EN-20120601_KEYWORDS.TSV.gz` and `Wikipedia-EN-20120601_REVISION_URIS.TSV.gz` are provided for download on the course homepage on `https://learn.iiitb.net/`.

The file `Wikipedia-EN-20120601_KEYWORDS.TSV` is a tab-separated file that contains a dump of 10,000 Wikipedia articles, which have been preprocessed and parsed into the following format:

```
ID<tab>TERM<tab>SCORE
```

- `ID` represents a unique numerical identifier for each Wikipedia article in the dump file.

- `TERM` denotes the distinct keywords that are contained in the corresponding Wikipedia articles in UTF-8 format. All keywords have been stemmed according to the Porter stemming algorithm[1] and stopwords, such as `the`, `and`, `have` and so on, have been removed.
- `SCORE` represents a decimal numeral value that captures the relevance of the keyword in the given article. The higher the score, the more relevant the article should be for this keyword.

Additionally, the file `Wikipedia-EN-20120601_REVISION_URIS.TSV` contains mappings of the article identifiers to the archived revisions of these Wikipedia articles. These URLs still are mostly intact, such that you can verify your results by opening the Wikipedia articles in a browser.

Problem 1 - Bulkloading Data into a PostgreSQL Database (6 Points)

1. Create a new database instance in PostgreSQL and create two relation schemas (i.e., tables) in order to store the content of the two files `Wikipedia-EN-20120601_KEYWORDS.TSV` and `Wikipedia-EN-20120601_REVISION_URIS.TSV` in your database. Think of appropriate primary- and foreign-key constraints for your two relation schema. **(2 Points)**
2. Bulkload the two TSV files into your relation schema using the PostgreSQL command `COPY`[2]. Make sure that the original UTF-8 encoding of the TSV files is preserved in your database. **(2 Points)**
3. Repeat the above bulkloading step once by including all your primary- and foreign-key constraints into the relation schema, and once by omitting all the primary- and foreign-key constraints from your relation schema. Compare the runtimes of the two options. **(2 Points)**

Problem 2 - Running Keyword Queries over Wikipedia (16 Points)

In the next step, we will compare different retrieval modes for evaluating keyword queries against the Wikipedia database you created in the previous problem. Formulate appropriate SQL queries to solve the following tasks.

Note: All of the following query tasks can be solved by using `GROUP BY` queries in SQL over the table containing the contents of the `Wikipedia-EN-20120601_KEYWORDS.TSV` file. A final join operation is only required between the table with the contents of `Wikipedia-EN-20120601_KEYWORDS.TSV` and the table with the contents of `Wikipedia-EN20120601_REVISION_URIS.TSV`.

1. *Boolean Retrieval 1*: Find URLs of Wikipedia articles that contain *all* of the stemmed keywords `infantri`, `reinforc`, `brigad`, and `fire`. **(2 Points)**
2. *Boolean Retrieval 2*: Find URLs of Wikipedia articles that contain *exactly one* of the stemmed keywords `infantri`, `reinforc`, `brigad`, or `fire`. **(2 Points)**
3. *Boolean Retrieval 3*: Find URLs of Wikipedia articles that contain the stemmed keyword `reinforc` but *not* any of the stemmed keywords `infantri`, `brigad`, or `fire`. **(2 Points)**
4. *Ranked Retrieval 1*: Find URLs of Wikipedia articles that contain *all* of the stemmed keywords `infantri`, `reinforc`, `brigad`, and `fire`, ordered by the sum of the scores of these keywords in the articles. **(3 Points)**
5. *Ranked Retrieval 2*: Find URLs of Wikipedia articles that contain *any* of the stemmed keywords `infantri`, `reinforc`, `brigad`, and `fire`, ordered by the sum of the scores of these keywords in the articles. **(3 Points)**

   Note: An article may be returned here if it contains any subset of these keywords, but the higher the sum of the scores of the keywords in an article, the higher this article should be ranked.
6. *Ranked Retrieval 3*: Find URLs of Wikipedia articles that contain the stemmed keyword `reinforc` but *not all* of the stemmed keywords `infantri`, `brigad`, or `fire`. Consider the following note while assigning the ranks. **(4 Points)**

[1] `http://tartarus.org/martin/PorterStemmer/`
[2] `http://www.postgresql.org/docs/current/interactive/populate.html`

Note: An article shall be returned even if it contains any proper subset of the keywords `infantri`, `brigad`, and `fire` – it should of course contain the keyword `reinforc`. However, if the score for `reinforc` is higher than the sum of the scores for the terms `infantri`, `brigad`, and `fire` for an article, then the article should be ranked higher.

Note: Please provide simple and easily understandable queries for the above problem. Complex or highly optimized queries, like those generated by AI tools, will not be evaluated.