DATABASE SYSTEMS (ISYS2014) ASSIGNMENT

MySQL User Guide

Prarthanaa Ragulan (21323157)

Sri Lanka [Lab Slot - Tuesday 3:30 pm]

CONTENTS

DISCLAIMER

This documentation is based on a Linux operating system, specifically Ubuntu 22.04.2, and utilizes a MySQL environment of version 8.0.39-0. All MySQL statements and their corresponding results are derived from this specific operating system and MySQL environment.

CREATING THE DATABASE AND TABLES

The file *'create_tables.sql'* is designed to manage the structure of the *Olympics24_21323157* database. Its primary function is to ensure that the database is created without conflicts from pre-existing data. It achieves this by first checking if the database already exists. If it does, the database is dropped to remove any previously existing tables, relationships, or data that could cause conflicts.

The *'create_tables.sql'* file contains all the SQL statements required to create the various tables in the correct sequence, taking into account dependencies between tables to avoid errors related to foreign key constraints.

Hence, to initialize the structure of the database and its relevant tables, source the *'create_tables.sql'* file from the mysql prompt.

```
mysql> SOURCE create_tables.sql
```

POPULATING TABLES

ins_athlete. ins_coach. ins_ ins_host. ins_medal. ins_ ins_playin. ins_sport. ins_venue.
sql sql country.sql sql sql medalist. sql sql sql
sql

The files shown in the image above contain individual INSERT INTO statements to populate each of the nine tables in the database. However, when using this method, it's important to be cautious, as tables with foreign key constraints must not be populated before the tables they reference. To address this issue, the ins_values.sql file has been created (shown below).
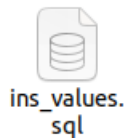

ins_values.
sql

The ins_values.sql file ensures simplicity and efficiency by sourcing each of the individual files in the correct order, ensuring that tables without foreign keys are populated first, followed by those that reference them. This guarantees smooth execution of all INSERT operations without foreign key conflicts.

```
-- Populates each table with values
SOURCE ins_country.sql;
SOURCE ins_sport.sql;
SOURCE ins_venue.sql;
SOURCE ins_athlete.sql;
SOURCE ins_medalist.sql;
SOURCE ins_coach.sql;
SOURCE ins_medal.sql;
SOURCE ins_playin.sql;
SOURCE ins_host.sql;
```

Hence, sourcing 'ins_values.sql' will populate all nine tables simultaneously.

```
mysql> SOURCE ins_values.sql;
```

EXECUTING QUERIES
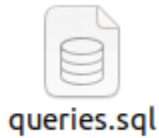
queries.sql

The *'queries.sql'* file contains all the eight SQL statements needed to query the database. These queries can be copied and executed individually by pasting them directly into the MySQL prompt. Or alternatively, for convenience, you can execute all eight queries at once by sourcing the *'queries.sql'* file.

```
mysql> SOURCE queries.sql
```

EXECUTING TRIGGERS

triggers.sql

The *'triggers.sql'* file contains the definitions for all three triggers used within the database. Each trigger can be executed individually by copying and pasting the corresponding trigger block statements into the MySQL command-line interface. For a more efficient approach though, the entire triggers.sql file can be sourced at once, which will execute all three triggers in sequence.

```
mysql> SOURCE triggers.sql
```

EXECUTING PROCEDURES
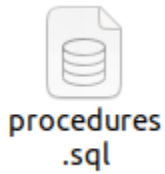
procedures
.sql

The '*procedures.sql*' file contains the definitions for the two procedures implemented within the database. Each procedure can be executed individually by copying and pasting the corresponding SQL statements into the MySQL command-line interface, or alternatively, for efficiency and convenience, the entire '*procedures.sql*' file can be sourced, which will execute both procedures simultaneously.

```
mysql> SOURCE procedures.sql
```