This document is a user manual guide that demonstrates how the cricket database management system can be implemented, operated, and maintained for future use or by another administrative user for analytic purposes.

User Guide for Cricket Database Management System

Final Assignment– Scenario A

Pramoda Gunarathne - 20531218

1. Requirement specification

The main goal of implementing a cricket management system is to help in the future for analytic purposes by administrative end or researchers and to reduce manual work for managing. This cricket database system stores all information of a particular match, series, team playing in matches, players in teams and other relative data.

In order for the database system to be used as an information processing unit to gather relevant data for their own purpose or for anyone who wishes to use the system, must first satisfy the following specifications mentioned below:

Minimum hardware specification

The hardware used for the development of the project were:

- Processor : Core™ i7 2.6 GHZ
- RAM : 16 GB
- Storage : 512 GB Solid State Drive

The hardware specifications do not have to be the same but using something like the above information would suffice.

1.1. Software requirement

- Operating System : Linux environment (Ubuntu – any version but this project was implemented in 20.04.1 version)
- Backend Software : MySQL (version 8.0 or higher)

Satisfying the above software requirements are adamant for the cricket database system to work prominently.

MySQL typically performs better on Linux, and you have more control over what exactly is going on in your system. In Linux environment most of the tools of MySQL server is already available therefore maintaining and compatibility is much better than in windows. The drawback is that you must learn how to control it. However, if the user wishes to use windows to implement the database, they can but they will have to download a lot of tools with the MySQL server. But using windows will be useful if in the future anyone wants to implement this backend with a front tier GUI.

1.2. MySQL Server

MySQL is a relational database which stores data in separate tables rather than in a single large storeroom. SQL is a programming language designed specifically for managing data stored in a relational database management system (RDBMS). SQL is made up of three languages: Data Definition Language (DDL), Data Manipulation Language (DML), and Data Control Language. SQL encompasses data insertion, query, update, and deletion, schema creation and modification, and data access control. Each cache data item is stored in its own field in SQL. Using MySQL as the

freeware is because it is much more reliable, fast, scalable, and easy to use. MySQL Database Software is a client/server system that includes a multithreaded SQL server that supports multiple back ends, several client programs and libraries, administrative tools, and a diverse set of application programming interfaces (APIs).

The database's software (MySQL) used for the development of this project are as shown below:

```
mysql>  show variables LIKE '%version%';
+-------------------------+-------------------------+
| Variable_name           | Value                   |
+-------------------------+-------------------------+
| admin_tls_version       | TLSv1.2,TLSv1.3         |
| immediate_server_version | 999999                  |
| innodb_version          | 8.0.31                  |
| original_server_version | 999999                  |
| protocol_version        | 10                      |
| replica_type_conversions |                        |
| slave_type_conversions  |                         |
| tls_version             | TLSv1.2,TLSv1.3         |
| version                 | 8.0.31-0ubuntu0.20.04.1 |
| version_comment         | (Ubuntu)                |
| version_compile_machine | x86_64                  |
| version_compile_os      | Linux                   |
| version_compile_zlib    | 1.2.12                  |
+-------------------------+-------------------------+
13 rows in set (0.01 sec)
```

The version I have used is 8.0.31. Any version higher than this can also be used.

As we can see on the left, the OS where the MySQL is used on is Linux and the ubuntu version is 20.04.1. Any Linux version can be used to run MySQL on it as this won't be an issue.

After configuring the MySQL server in Linux environment, we can check the username and password with the status information by entering the following command as shown below:

```
mysql> status;
--------------
mysql  Ver 8.0.31-0ubuntu0.20.04.1 for Linux on x86_64 ((Ubuntu))

Connection id:          9
Current database:
Current user:           root@localhost
SSL:                    Not in use
Current pager:          stdout
Using outfile:          ''
Using delimiter:        ;
Server version:         8.0.31-0ubuntu0.20.04.1 (Ubuntu)
Protocol version:       10
Connection:             Localhost via UNIX socket
Server characterset:    utf8mb4
Db     characterset:    utf8mb4
Client characterset:    utf8mb4
Conn.  characterset:    utf8mb4
UNIX socket:            /var/run/mysqld/mysqld.sock
Binary data as:         Hexadecimal
Uptime:                 6 min 22 sec

Threads: 2  Questions: 7  Slow queries: 0  Opens: 136  Flush tables: 3  Open tables: 55  Queries per second avg: 0.018
--------------
```

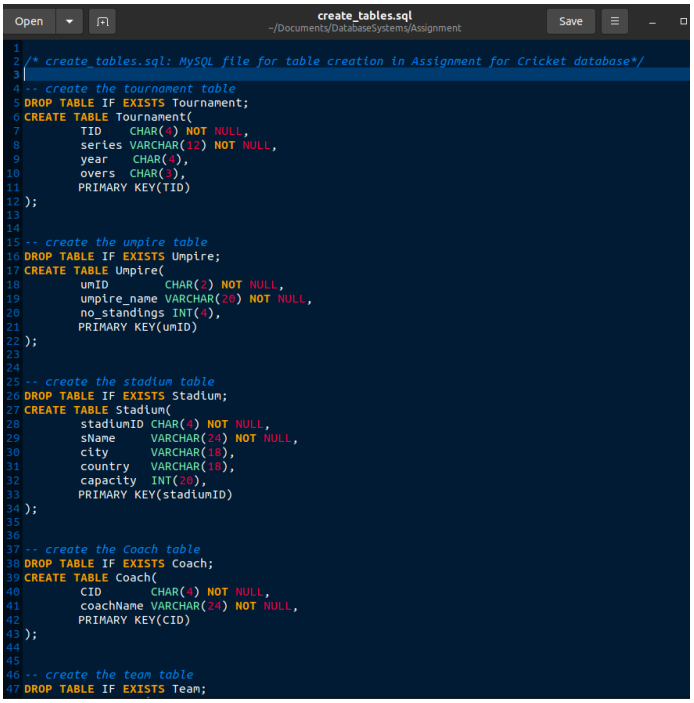
This query shows information about the current user, protocol version and more.

2. Database Implementation Details

The information system was implemented and tested for T20, T10, ODI and test match data. The statistical information produced by this system can be useful only when the database is updated with all the information using create table queries and insert queries. After the implementation of the MySQL Sever, we can proceed with the implementation of the cricket database.

The database tables created are organized into a single file that are optimized for performance and maintenance, the name of this file is create_tables.sql. If anyone wants to update existing

tables or add new tables to the list, they can simply add these tables to this file accordingly to its referential integrity constraints.




```sql
/* create_tables.sql: MySQL file for table creation in Assignment for Cricket database*/

-- create the tournament table
DROP TABLE IF EXISTS Tournament;
CREATE TABLE Tournament(
        TID     CHAR(4) NOT NULL,
        series VARCHAR(12) NOT NULL,
        year    CHAR(4),
        overs   CHAR(3),
        PRIMARY KEY(TID)
);


-- create the umpire table
DROP TABLE IF EXISTS Umpire;
CREATE TABLE Umpire(
        umID        CHAR(2) NOT NULL,
        umpire_name VARCHAR(20) NOT NULL,
        no_standings INT(4),
        PRIMARY KEY(umID)
);


-- create the stadium table
DROP TABLE IF EXISTS Stadium;
CREATE TABLE Stadium(
        stadiumID CHAR(4) NOT NULL,
        sName     VARCHAR(24) NOT NULL,
        city      VARCHAR(18),
        country   VARCHAR(18),
        capacity  INT(20),
        PRIMARY KEY(stadiumID)
);


-- create the Coach table
DROP TABLE IF EXISTS Coach;
CREATE TABLE Coach(
        CID       CHAR(4) NOT NULL,
        coachName VARCHAR(24) NOT NULL,
        PRIMARY KEY(CID)
);


-- create the team table
DROP TABLE IF EXISTS Team;
```

The 15 tables which were refined and created in the database is written in this file. To run this SQL file, we can SOURCE the file to the relevant database created, in my case the database is CricketDB_20531218 as shown above.

```
mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| CricketDB_20531218 |
| NewCompany         |
| company            |
| dswork             |
| dsworks            |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
9 rows in set (0.01 sec)
```

Before sourcing the tables to the database, we must select the database, we can do this by the command, USE CricketDB_20531218;

- To implement the tables, we can use the command → `mysql> SOURCE ./create_tables.sql`

To insert the data sets to each table, I have three insert SQL files as shown below:

insertOne.sql has Tournament, Umpire, Stadium, Coach and Team tables data insertion queries. insertTwo.sql has Player, WicketKeeper, Matchs, Plays, Scoreboard and Achievement table insertion queries and insertThree.sql has Batsmen, Bowler, Inning and RankIn insertion queries.

The order of insertion should be same as the numbering of the files, hence from insertOne.sql → insertTwo.sql → insertThree.sql.

- The command used to insert these files to the tables created → `mysql> SOURCE ./insertOne.sql`

The same command is used to source the other two files. Referential integrity constraints are considered when inserting data. Entering data by sourcing a SQL file is easier for maintenance and manipulation of the tables. If anyone in the future wants to insert data to a new table, they can write the insertion query to insertThree.sql file or create a new SQL file and write the relevant query.

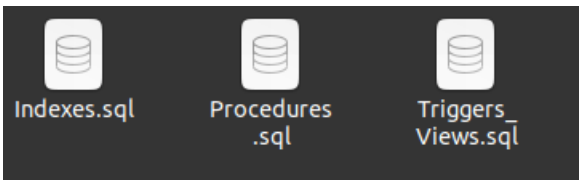
3. Data Retrieval Queries

After the database is successfully implemented with insertion of data, queries can be written to obtain results / retrieve data from the database. All the commands and query results are logged to a file called CricketDB_20531218.out so any user can review this file and compare with the query results they got.

- All the queries that I have written to be performed to the database is written in Queries.sql file. There are 25 queries written to retrieve data from the database in a reasonable manner. And to export these commands to the MySQL server, we can source the file by using the below command: `mysql> SOURCE ./Queries.sql`

The user can also copy paste individual commands to the MySQL Server and run the scripts. But since I have made it possible to source the entire Queries.sql file it saves time. If the user wishes they can write their own queries to this file and source the file altogether.

The advanced features queries are defined in three files as shown below:



where Procedures.sql file contains 3 stored procedures, Triggers_Views.sql file contains 2 of each and Indexes.sql file with 2 indexes.

- The commands for stored procedures that can be performed are written in Procedures.sql file where it contains 3 stored procedural functions. To view the results of the commands we can either source the entire file as follows, `mysql> SOURCE ./Procedures.sql` or copy paste each procedure individually and test the procedures by calling them. An example of copy pasting one procedure and testing is done below:

```
mysql> DROP PROCEDURE IF EXISTS get_schedules;
Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER //
mysql> CREATE PROCEDURE get_schedules()
    -> COMMENT 'To get the total schedules of the matches available in the database'
    -> BEGIN
    ->     SELECT Plays.date_time FROM Plays;
    ->     SELECT COUNT(scheduleID) AS Total_Schedules FROM Plays;
    -> END //
Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER ;
mysql> CALL get_schedules;
```

We can see that the procedure is added correctly and if we press enter and call the procedure the results will be outputted to the screen accordingly.

- The commands for triggers and views that can be performed are written in Triggers_Views.sql file where it contains 2 triggers and 2 views. Each trigger, view, and testing for them can be copy pasted inside the MySQL Server to view the results as sourcing the file would give errors because of the commenting style used.

```
mysql> DELIMITER //
mysql> CREATE TRIGGER updateHighScore
    ->
    -> AFTER UPDATE ON Scoreboard
    ->
    -> FOR EACH ROW
    -> BEGIN
    ->
Display all 762 possibilities? (y or n)
    -> IF NEW.highestScore < OLD.highestScore THEN
    ->
Display all 762 possibilities? (y or n)
    ->
Display all 762 possibilities? (y or n)
    -> UPDATE Scoreboard
    ->
Display all 762 possibilities? (y or n)
    ->
Display all 762 possibilities? (y or n)
    ->
Display all 762 possibilities? (y or n)
    -> SET highestScore = OLD.highestScore WHERE MID = OLD.MID;
    ->
Display all 762 possibilities? (y or n)
    -> END IF;
    ->
Display all 762 possibilities? (y or n)
    ->
    -> END
    -> //
Query OK, 0 rows affected (0.01 sec)
```

The picture to the left shows how the trigger is entered. This way we can input the triggers and views.

Using these queries by copy and pasting is much safer than sourcing the entire file.

- To execute the 2 indexes and its test queries, we can either source the file as follows, or copy `mysql> SOURCE ./Indexes.sql` paste them individually and run. But sourcing the file is more suitable as it contains the query to drop the indexes if it already exists in the database and create freshly.

If the user wishes to implement more indexes, they can write them in the Indexes.sql fie and source the entire file.

- The database connection with python3 code is added to the files as program.py. There is a README file added which shows how to run the program to retrieve query results. ‘

First check if python3 is installed by using the below command:

```
pramoda@pramoda-VirtualBox:~$ python3 -V
Python 3.8.10
```

If it is not available, install it.

Then check if mysql-connector-python is installed by using the below command:

```
pramoda@pramoda-VirtualBox:~$ pip install mysql-connector-python
Requirement already satisfied: mysql-connector-python in ./.local/lib/python3.8/site-packages (8.0.31)
Requirement already satisfied: protobuf<=3.20.1,>=3.11.0 in ./.local/lib/python3.8/site-packages (from my
sql-connector-python) (3.20.1)
```

If it is not, this command will automatically install it for you. After successfully installing these requirements, we can run the python program using the following command,

```
python3 program.py
```

Then, we can obtain the query results by successful connection made to the database.

If the user wishes they can add more to the program.py file and execute them using python3.

The user can either run each advanced queries individually by copy and pasting on the MySQL command line or just source the whole file as mentioned early to get the query results. Comments are added inside each SQL file to make it user friendlier for the users and more understandable for anyone trying to execute these commands.