SVKM's NMIMS

School of Technology Management & Engineering, Navi Mumbai

A.Y. 2023 - 24

Course: Database Management Systems

Project Report

Program	Btech CE	
Semester	IV	
Name of the Project:	CHESS Database	
Details of Project Members		
Batch	Roll No.	Name
B2	A093	Antariksh Rajkonwar
B2	A112	Harshad More
B2	A096	Mayank Kumar
Date of Submission:		

Contribution of each project Members:

Roll No.	Name:	Contribution
A093	Antariksh Rajkonwar	Database, Queries
A112	Harshad More	Database, Report
A096	Mayank Kumar	Database, ER model

Project Report
ICC DATABASE

by

Antariksh Rajkonwar A093
Harshad More A112
Mayank Kumar A096

Course: DBMS

AY: 2023-24

Table of Contents

STORYLINE

In the digital age where every move is meticulously recorded, the world of chess is no exception. Enter Chess DBMS, a cutting-edge Database Management System (DBMS) designed specifically for chess enthusiasts and professionals. At its core, Chess DBMS serves as a comprehensive repository for storing, organizing, and analyzing vast volumes of chess data. From historic games played by grandmasters to the latest tournaments, Chess DBMS meticulously catalogs every move, opening strategy, and tactical exchange, providing users with an invaluable resource for honing their skills and deepening their understanding of the game. The journey begins with an amazing intuitive user interface, inviting users to explore the vast universe of chess knowledge at their fingertips. Whether a novice seeking to learn the fundamentals or a seasoned player aiming to refine their strategy, Chess DBMS offers a tailored experience for every skill level. Embark on a voyage through time as Chess DBMS transports users to iconic matches from chess history. From the legendary encounters between Kasparov and Karpov to the modern battles waged by Carlsen and Nakamura, every game is meticulously archived and ready for analysis. But Chess DBMSis more than just a repository of past games; it's a dynamic platform for innovation and collaboration. Engage with fellow enthusiasts in lively discussions, share insights and strategies, and collaborate on groundbreaking research projects. With Chess DBMS, the boundaries of possibility are limitless.

For the aspiring player, Chess DBMS offers a suite of powerful analytical tools to dissect games, uncover patterns, and refine techniques. Dive deep into the nuances of openings, explore tactical motifs, and unravel the mysteries of endgame theory with unparalleled precision and depth. But Chess DBMS isn't just for players—it's also a valuable resource for coaches, trainers, and educators. Create customized training programs, track progress over time, and identify areas for improvement with pinpoint accuracy. With Chess DBMS, mastery is within reach. As the chess landscape continues to evolve, so too does Chess DBMS. With regular updates and enhancements, the platform remains at the forefront of innovation, pushing the boundaries of what's possible in the world of chess analysis and exploration. In the ever-expanding universe of chess, knowledge is power. With Chess DBMS as your guide, unlock the secrets of the game, sharpen your skills, and embark on a journey of discovery unlike any other. Welcome to the future of chess analysis. Welcome to Chess DBMS.

COMPONENTS

Here is the list of all the Entities and its Attributes:

*Primary key mentioned in **BOLD***

Players: Player_**ID**, Name, Nationality, raitng,

Tournaments: **tournament_ID**, Name, Location, Start_date, End_date, organizer

Teams: **Team_ID**, name

Events**: Tournament_lD**, event_id, name, event_Date,location,description.

Commentaries: Commentary**_id**, game_id, Commentator_id, Commentary

Coaches : **Coach_lD,** name, contact_details

CoachPlayers : **Coach_lD, PlayerlD**

TeamPlayers : **Player_lD, PlayerlD**

Prizes**: Prize_lD**, Tournament_lD,amount, category

Games**: Game_lD**, match_lD, moves, timestamp

Matches: **tournament_ID**, **Match_lD**, Player1_id, player2_id, match_date, result

UserPreferences : **User_lD,** language, theme,notifications

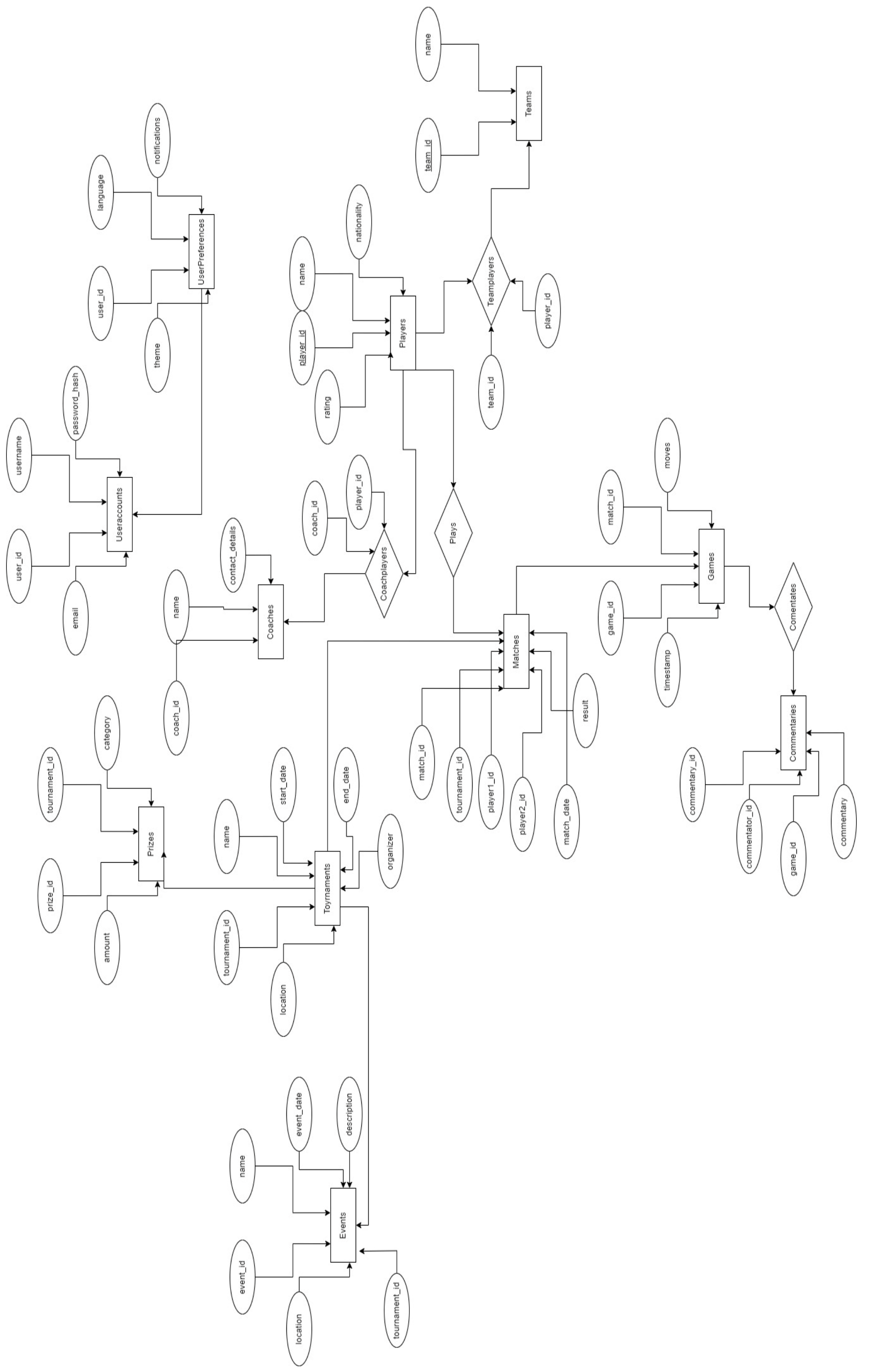
UserAccounts : **User_lD,** username ,email, password_hash

ChessPiceces : **piece_lD,** name

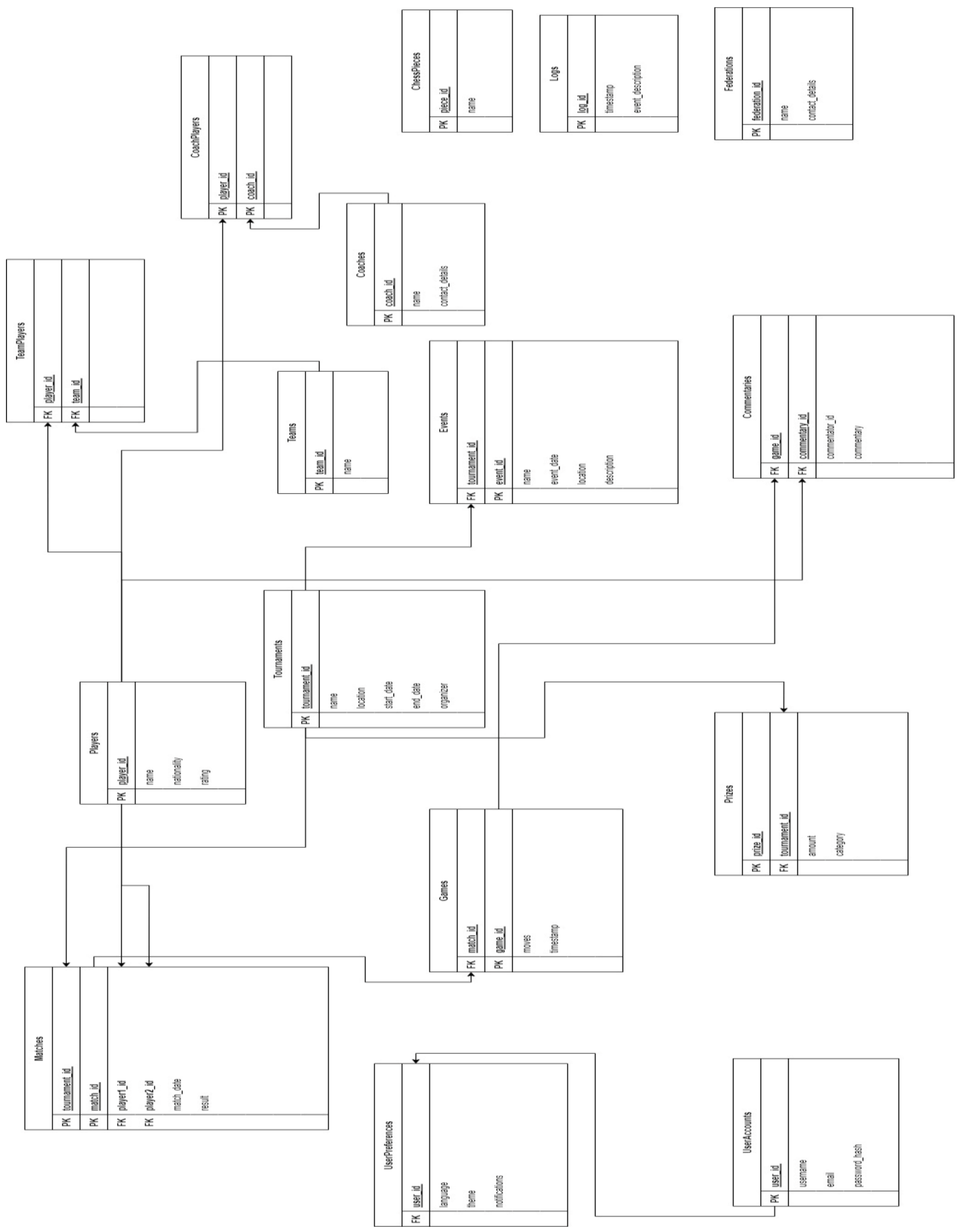
Logs : **log_lD,** timestamp, event_description

Federations : **federation_lD,** name ,contact_details

ENTITY-RELATIONSHIP DIAGRAM



RELATIONAL MODEL



NORMALIZATION

Players: Already in 1NF, 2NF, and 3NF.

Tournaments: Already in 1NF, 2NF, and 3NF..

Teams: Already in 1NF, 2NF, and 3NF.

Events: Already in 1NF, 2NF, and 3NF.

Commentaries: Already in 1NF, 2NF, and 3NF.

Coaches: Already in 1NF, 2NF, and 3NF.

CoachPlayers: Already in 1NF, 2NF, and 3NF.

TeamPlayers: Already in 1NF, 2NF, and 3NF.

Prizes: Already in 1NF, 2NF, and 3NF.

Games: Already in 1NF, 2NF, and 3NF.

Matches: Already in 1NF, 2NF, and 3NF.

UserPreference: Already in 1NF, 2NF, and 3NF.

UserAccounts: Already in 1NF, 2NF, and 3NF.

Logs: Already in 1NF, 2NF, and 3NF.

Federations: Already in 1NF, 2NF, and 3NF.

ChessPieces: Already in 1NF, 2NF, and 3NF.

All tables are already in 1NF, 2NF, and 3NF, so no further normalization is needed. Each table has a single theme, and all non-key attributes depend on the full primary key. Therefore, the database is already in BCNF.

SQL QUERIES

Below are the tables created for the entities and its attributes:

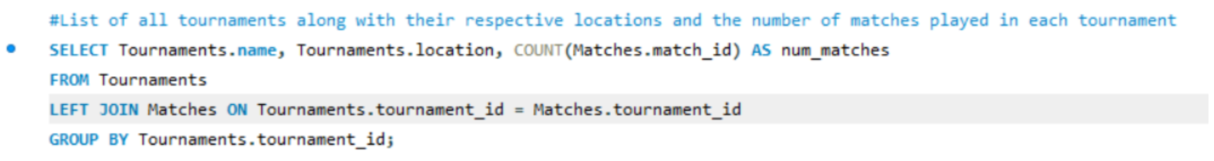
```
create database miniproj2;
use miniproj2;

-- Players Table
CREATE TABLE Players (
   player_id INT PRIMARY KEY AUTO_INCREMENT,
   name VARCHAR(255) NOT NULL,
   nationality VARCHAR(100),
   rating INT

);

-- Teams Table
CREATE TABLE Teams (
   team_id INT PRIMARY KEY AUTO_INCREMENT,
   name VARCHAR(255) NOT NULL

);

-- TeamPlayers Relationship Table
CREATE TABLE TeamPlayers (
   team_id INT,
   player_id INT,
   PRIMARY KEY (team_id, player_id),
   FOREIGN KEY (team_id) REFERENCES Teams(team_id),
   FOREIGN KEY (player_id) REFERENCES Players(player_id)
);

-- Matches Table
CREATE TABLE Matches (
   match_id INT PRIMARY KEY AUTO_INCREMENT,
   tournament_id INT,
   player1_id INT,
   player2_id INT,
   match_date DATE,
   result VARCHAR(50),
   FOREIGN KEY (tournament_id) REFERENCES Tournaments(tournament_id),
   FOREIGN KEY (player1_id) REFERENCES Players(player_id),
   FOREIGN KEY (player2_id) REFERENCES Players(player_id)

);
```

```
-- Tournaments Table
CREATE TABLE Tournaments (
   tournament_id INT PRIMARY KEY AUTO_INCREMENT,
   name VARCHAR(255) NOT NULL,
   location VARCHAR(255),
   start_date DATE,
   end_date DATE,
   organizer VARCHAR(255)

);

-- Events Table
CREATE TABLE Events (
   event_id INT PRIMARY KEY AUTO_INCREMENT,
   name VARCHAR(255) NOT NULL,
   event_date DATE,
   location VARCHAR(255),
   description TEXT,
   tournament_id INT,
   FOREIGN KEY (tournament_id) REFERENCES Tournaments(tournament_id)

);

-- Coaches Table
CREATE TABLE Coaches (
   coach_id INT PRIMARY KEY AUTO_INCREMENT,
   name VARCHAR(255) NOT NULL,
   contact_details VARCHAR(255)

);

-- CoachPlayers Relationship Table
CREATE TABLE CoachPlayers (
   coach_id INT,
   player_id INT,
   PRIMARY KEY (coach_id, player_id),
   FOREIGN KEY (coach_id) REFERENCES Coaches(coach_id),
   FOREIGN KEY (player_id) REFERENCES Players(player_id)
);

-- Chess Pieces Table
CREATE TABLE ChessPieces (
   piece_id INT PRIMARY KEY AUTO_INCREMENT,
   name VARCHAR(50) NOT NULL
```

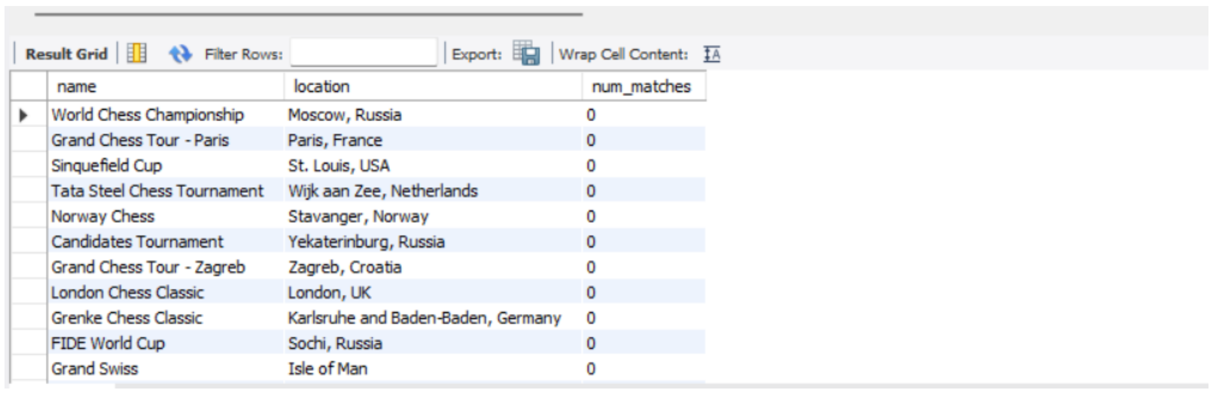
```
);

-- Prizes Table
CREATE TABLE Prizes (
    prize_id INT PRIMARY KEY AUTO_INCREMENT,
    tournament_id INT,
    amount DECIMAL(10, 2),
    category VARCHAR(100),
    FOREIGN KEY (tournament_id) REFERENCES Tournaments(tournament_id)

);

-- Federations Table
CREATE TABLE Federations (
    federation_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(255) NOT NULL,
    contact_details VARCHAR(255)

);

-- Games Table
CREATE TABLE Games (
    game_id INT PRIMARY KEY AUTO_INCREMENT,
    match_id INT,
    moves TEXT,
    timestamp DATETIME,
    FOREIGN KEY (match_id) REFERENCES Matches(match_id)

);


-- Commentaries Table
CREATE TABLE Commentaries (
    commentary_id INT PRIMARY KEY AUTO_INCREMENT,
    game_id INT,
    commentator_id INT,
    commentary TEXT,
    FOREIGN KEY (game_id) REFERENCES Games(game_id),
    FOREIGN KEY (commentator_id) REFERENCES Players(player_id)

);

-- User Accounts Table
CREATE TABLE UserAccounts (
    user_id INT PRIMARY KEY AUTO_INCREMENT,
    username VARCHAR(255) NOT NULL,
    email VARCHAR(255) NOT NULL,
    password_hash VARCHAR(255) NOT NULL

);

-- User Preferences Table
CREATE TABLE UserPreferences (
    user_id INT PRIMARY KEY,
    language VARCHAR(50),
    theme VARCHAR(50),
    notifications BOOLEAN,
    FOREIGN KEY (user_id) REFERENCES UserAccounts(user_id)

);

-- Logs Table
CREATE TABLE Logs (
    log_id INT PRIMARY KEY AUTO_INCREMENT,
    timestamp DATETIME,
    event_description TEXT

);
```

PROJECT DEMONSTRATION

1.

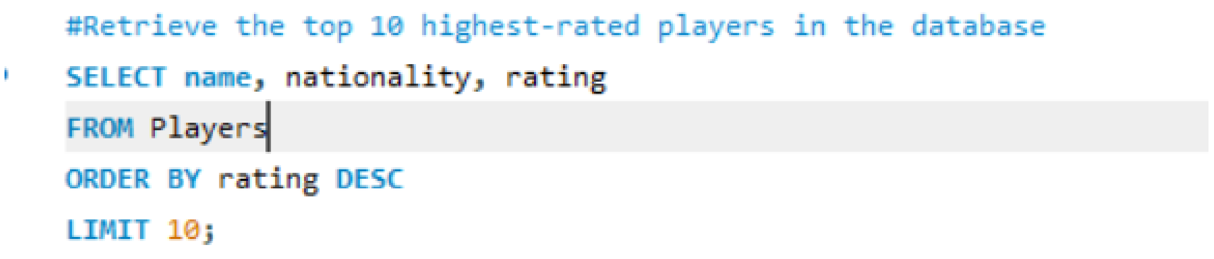


```
#List of all tournaments along with their respective locations and the number of matches played in each tournament
SELECT Tournaments.name, Tournaments.location, COUNT(Matches.match_id) AS num_matches
FROM Tournaments
LEFT JOIN Matches ON Tournaments.tournament_id = Matches.tournament_id
GROUP BY Tournaments.tournament_id;
```

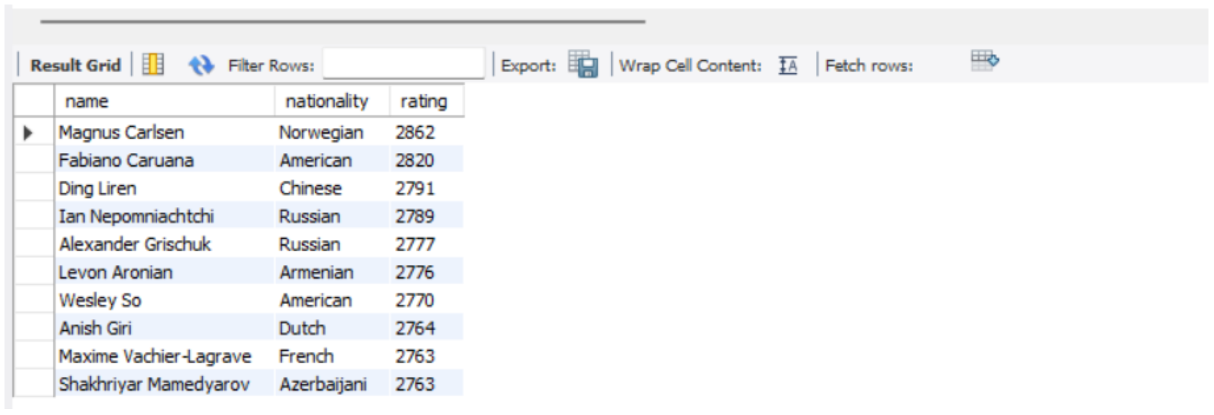


name	location	num_matches
World Chess Championship	Moscow, Russia	0
Grand Chess Tour - Paris	Paris, France	0
Sinquefield Cup	St. Louis, USA	0
Tata Steel Chess Tournament	Wijk aan Zee, Netherlands	0
Norway Chess	Stavanger, Norway	0
Candidates Tournament	Yekaterinburg, Russia	0
Grand Chess Tour - Zagreb	Zagreb, Croatia	0
London Chess Classic	London, UK	0
Grenke Chess Classic	Karlsruhe and Baden-Baden, Germany	0
FIDE World Cup	Sochi, Russia	0
Grand Swiss	Isle of Man	0

2.



```
#Retrieve the top 10 highest-rated players in the database
SELECT name, nationality, rating
FROM Players
ORDER BY rating DESC
LIMIT 10;
```



name	nationality	rating
Magnus Carlsen	Norwegian	2862
Fabiano Caruana	American	2820
Ding Liren	Chinese	2791
Ian Nepomniachtchi	Russian	2789
Alexander Grischuk	Russian	2777
Levon Aronian	Armenian	2776
Wesley So	American	2770
Anish Giri	Dutch	2764
Maxime Vachier-Lagrave	French	2763
Shakhriyar Mamedyarov	Azerbaijani	2763

SELF LEARNING BEYOND CLASSROOM

Through the process of creating a chess database management system (DBMS), several key learnings emerged Firstly, we gained a deeper appreciation for the complexity of representing the nuances of chess within a structured database schema. Understanding the intricacies of chess games, including moves, positions, players, and tournaments, was essential for designing an effective data model. Additionally, optimizing database performance became a crucial aspect of the project. Balancing the need for efficient query execution with data integrity and storage requirements required careful consideration of indexing strategies, query optimization techniques, and database normalization principles. Overall, the process of creating a chess database management system offered valuable lessons in collaboration, iteration, documentation, and user-centric design, which can be applied to future projects across various domains and industries. By embracing these learnings and best practices, teams can enhance their effectiveness, agility, and success in developing innovative and impactful software solutions.

LEARNING FROM PROJECT

This task supplied a number of useful discoverings in data source monitoring and also querying. First of all I found out just how to create as well as create a data source framework efficiently, arranging information right into tables and also developing partnerships in between them. Comprehending the value of main secrets as well as international secrets assisted make sure information stability together with uniformity. In addition, I obtained experience in placing, upgrading and also erasing information from the data source, which is necessary for preserving exact as well as current details. Matching the data source utilizing SQL showed me just how to recover details information based upon numerous requirements such as problems, arranging as well as gathering. I likewise discovered normalization concepts, which are vital for maximizing data source efficiency as well as decreasing information redundancy. Taking care of null worths and also applying restraints additionally improved my understanding of information administration ideal techniques. In general this job given a hands-on chance to use academic data source ideas in an useful setup, strengthening my understanding plus abilities in data source management as well as SQL querying.

CHALLENGES FACED

Creating a chess database management system in SQL poses challenges such as designing an efficient data model, optimizing query performance, ensuring data integrity, implementing concurrency control, and integrating the system with the user interface. Other challenges include scalability, user authentication, data migration, error handling, and regulatory compliance. Addressing these challenges is essential to develop a robust and reliable chess DBMS that meets the needs of users in managing and analyzing chess-related data effectively.

In addition to these challenges, developers must also consider the complexities of representing various chess entities such as games, moves, players, and tournaments within the database schema while minimizing redundancy. Optimizing database performance for complex queries, ensuring seamless integration with the user interface, and implementing robust security measures to protect user data are critical aspects. Furthermore, managing scalability as the volume of chess data grows, addressing data migration requirements, handling errors effectively.

CONCLUSION

In this task, we developed a data source for chess related info, consisting of information concerning nations, gamers, groups, suits, authorities, events as well as a lot more. We arranged the information right into tables and also developed connections in between them to stand for the complicated interconnections within the cricketing globe precisely. By composing SQL inquiries, we might draw out important understandings from the data source such as discovering top-ranked gamers assessing suit results together with recognizing corrective activities. Throughout the task we came across obstacles in making the data source schema composing complicated inquiries handling big datasets, together with analyzing question outcomes. Nevertheless by getting rid of these obstacles we obtained important experience in data source layout SQL querying plus information evaluation. In general, this task improved our understanding of relational data sources as well as their application in arranging as well as examining cricket-related info. It additionally offered useful direct exposure to real life data source monitoring circumstances strengthening vital ideas along with abilities in an useful way.