



American International University-Bangladesh
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
Dhaka, Bangladesh

00892 ADVANCE DATABASE MANAGEMENT SYSTEM
PROJECT Report
[B][SUMMER 22-23]

E-Sports Management System

Group 5

Submitted by

| Names of Students | ID |
|----------------------|------------|
| AJLAN HOSSAIN | 19-39334-1 |
| SHARIF HADI MAHATAB | 20-43625-2 |
| MD. SARAFAT ALI ADIR | 20-41926-1 |
| MEDHA CHOWDHURY | 20-41930-1 |

Date of Submission

July 15, 2023

Submitted to

JUENA AHMED NOSHIN

Assistant Professor, Faculty

Department of Computer Science and Engineering
American International University-Bangladesh

Contribution

| | AJRAN HOSSAIN | SHARIF HADI MAHATAB | MD. SARAFAT ALI ADIR | MEDHA CHOWDHURY | Contribution (%) |
|--------------------|-------------------|---------------------|----------------------|-------------------|------------------|
| | <i>19-39334-1</i> | <i>20-43625-2</i> | <i>20-41926-1</i> | <i>20-41930-1</i> | |
| Diagram | 60% | 20% | 10% | 10% | 100(%) |
| UI Design | 100% | 0% | 0% | 0% | 100(%) |
| Normalization | 50% | 50% | 0% | 0% | 100(%) |
| SQL Query | 85% | 5% | 5% | 5% | 100(%) |
| Relational Algebra | 15% | 0% | 85% | 0% | 100(%) |
| Report Writing | 55% | 35% | 5% | 5% | 100(%) |

Contents

| | | |
|----------|--|-----------|
| 1 | Introduction | 1 |
| 1.1 | Project Proposal | 1 |
| 1.1.1 | Purposes | 2 |
| 1.1.2 | Methodology | 2 |
| 1.2 | Project Scenario | 3 |
| 2 | Diagrams | 4 |
| 2.1 | ER Diagram | 4 |
| 2.2 | Class Diagram | 5 |
| 2.3 | Use Case Diagram | 6 |
| 2.4 | Activity Diagram | 7 |
| 3 | User Interface | 8 |
| 3.1 | Technologies Used | 8 |
| 3.1.1 | Home Page | 8 |
| 3.1.2 | About Page | 8 |
| 3.1.3 | Login Page | 9 |
| 3.1.4 | Profile Page | 9 |
| 3.1.5 | Tournament Page | 9 |
| 3.1.6 | Team Page | 10 |
| 3.1.7 | Game Page | 10 |
| 4 | NF and Schema | 11 |
| 4.1 | Manage branch | 11 |
| 4.1.1 | (Admin \rightarrow Manager) | 11 |
| 4.1.2 | (Manager \rightarrow Finance) | 12 |
| 4.1.3 | (Manager \rightarrow Teams) | 13 |
| 4.1.4 | (SocialMedia \rightarrow ContentCreator) | 14 |
| 4.2 | Pay branch | 16 |
| 4.2.1 | (Finance \rightarrow SocialMedia) | 16 |
| 4.2.2 | (Finance \rightarrow organization) | 17 |
| 4.3 | Formed branch | 18 |
| 4.3.1 | (Teams \rightarrow Player) | 18 |
| 4.4 | Has branch | 20 |
| 4.4.1 | (Record \rightarrow Tournament) | 20 |
| 4.4.2 | (Tournament \rightarrow Game) | 21 |
| 4.5 | Participate branch | 22 |
| 4.5.1 | (Teams \rightarrow Game) | 22 |
| 4.6 | Host branch | 23 |
| 4.6.1 | (Organizer \rightarrow Tournament) | 23 |
| 4.7 | Sponsor branch | 24 |
| 4.7.1 | (Companies \rightarrow Organization) | 24 |
| 4.7.2 | (Companies \rightarrow Teams) | 25 |
| 4.8 | Temporary Tables | 26 |
| 4.9 | Final Tables | 28 |
| 4.10 | Schema Diagram | 30 |

| | | |
|----------|--------------------------------|-----------|
| 5 | SQL Queries | 31 |
| 5.1 | User Creation | 31 |
| 5.2 | Table Creation | 32 |
| 5.3 | Sequence Creation | 45 |
| 5.4 | Index for Table | 46 |
| 5.5 | Alter Table | 47 |
| 5.6 | Data Insertion | 48 |
| 5.7 | Single Row Functions | 59 |
| 5.8 | Group Functions | 60 |
| 5.9 | SubQuery | 61 |
| 5.10 | Join Queries | 63 |
| 5.11 | Creating View | 65 |
| 5.12 | Synonyms | 66 |
| 6 | Relational Algebra | 67 |
| 7 | Conclusion | 68 |

Introduction

Esports Management System is an innovative platform that will revolutionize the management and organization of esports teams, participants, tournaments, and sponsors. This system seeks to provide users with an efficient and user-friendly way to search for their preferred professional esports players.

A user-friendly interface is at the core of the Esports Management System, allowing users to seamlessly navigate and explore the realm of professional esports. With only a few clicks, users can search for potential professional athletes and teams, as well as access valuable information such as their winning records and accomplishments. This enables fans and enthusiasts to remain up-to-date on their preferred players and teams, nurturing a stronger connection within the esports community.

The Esports Management System's ability to facilitate sponsorships is a crucial feature. Numerous organizations and businesses can engage in sponsorship activities, whether for the purpose of supporting tournaments or individual athletes. The system serves as a centralized repository where the information and details of these sponsors can be efficiently stored and managed. This facilitates the sponsorship process and ensures that sponsors and the esports industry collaborate effectively.

There are specialized administrators within the Esports Management System who play crucial roles in managing and enhancing the overall experience. The social media manager is among these supervisors; he or she supervises the organization's online presence and engagement on various social media platforms. In addition, the content creator/VFX/GFX team assures the creation of visually stunning and captivating content that enhances the overall esports experience.

Dynamic features and functionalities make the Esports Management System an indispensable instrument for the esports industry. It makes it easier for fans to discover and connect with professional esports players, allowing them to remain informed and engaged. It enhances collaboration between organizations and the esports community by providing a centralized platform for sponsorship management. In addition, the system enables administrators to enhance the organization's online presence and develop visually appealing content, ensuring that all stakeholders have an engaging experience.

In the following sections, we will delve deeper into the features, functionalities, and innovative aspects of the Esports Management System, demonstrating its potential to revolutionize the management and celebration of esports teams, players, tournaments, and sponsors.

1.1 Project Proposal

This proposal for the development and implementation of an Esports Management System is presented with pleasure. This revolutionary platform seeks to transform the management and organization of esports teams, players, tournaments, and sponsors. The Esports Management System will improve the user experience, encourage community engagement, and expedite operations within the esports industry by leveraging advanced technology and comprehensive functionalities.

1.1.1 Purposes

- Create an intuitive web-based platform that serves as the central hub for esports administration, catering to the requirements of teams, players, tournament organizers, and sponsors.
- Implement a sophisticated matching algorithm to facilitate the search and discovery of favored professional esports players, thereby enhancing the fan experience and fostering esports community connections.
- Provide efficient sponsorship administration capabilities, enabling organizations and businesses to support tournaments or individual athletes through sponsorship activities.
- Enhance the organization's online presence by supervising social media platforms and having the content creator/VFX/GFX team produce visually spectacular and engaging content.

1.1.2 Methodology

System Development:

- Conduct exhaustive investigation on the necessary requirements and features of an effective Esports Management System.
- Utilize industry-standard programming languages and technologies to create a scalable and secure web-based platform.
- Implement a user-friendly interface with intuitive navigation in order to provide a seamless and enjoyable user experience.

Matching Algorithm

- Collaboration with data scientists and psychologists to create a matching algorithm based on personality traits, values, and beliefs.
- Integrate the matching algorithm into the system to recommend professional esports players compatible with the user's preferences.

Sponsorship Management

- Create an all-encompassing sponsorship management module to facilitate collaborations between organizations and the esports industry.
- Provide a centralized repository for sponsor information to facilitate communication and sponsorship efficiency.

Online Presence Enhancement

- Appoint a social media manager to supervise the organization's online presence and interact with the esports community.
- Appoint a social media manager to supervise the organization's online presence and interact with the esports community.

1.2 Project Scenario

Imagine an eSports organization called "eSports FTW" that manages various teams and tournaments in the gaming industry. The organization is led by an Admin who oversees the operations. The Admin entity contains attributes such as email, picture, password, name, and a unique ID.

Under the Admin, there are multiple Managers responsible for different departments. Each Manager has attributes including hire date, picture, salary, email, name, department ID, and a unique ID. One Manager specifically manages the Finance department, ensuring financial stability and handling the accounts for the organization. The Finance entity consists of attributes such as a unique ID, account number, and balance.

In addition to managing the finances, each Manager is in charge of a specific team. The Teams entity includes details such as the team's established date, country, name, team ID, team icon, winning numbers, and total prize money. Each team has a unique Manager assigned to it, ensuring proper coordination and organization. The Manager entity is linked to the Teams entity through the Manager ID attribute.

Within each team, there are multiple Players who represent the organization in various games. The Players entity contains attributes such as name, player ID, picture, salary, winning prize money, total hours played, phone number, and address (including country, city, zip code, and road number). Additionally, players have links to their social media profiles on platforms like Facebook, Instagram, Twitter, and YouTube.

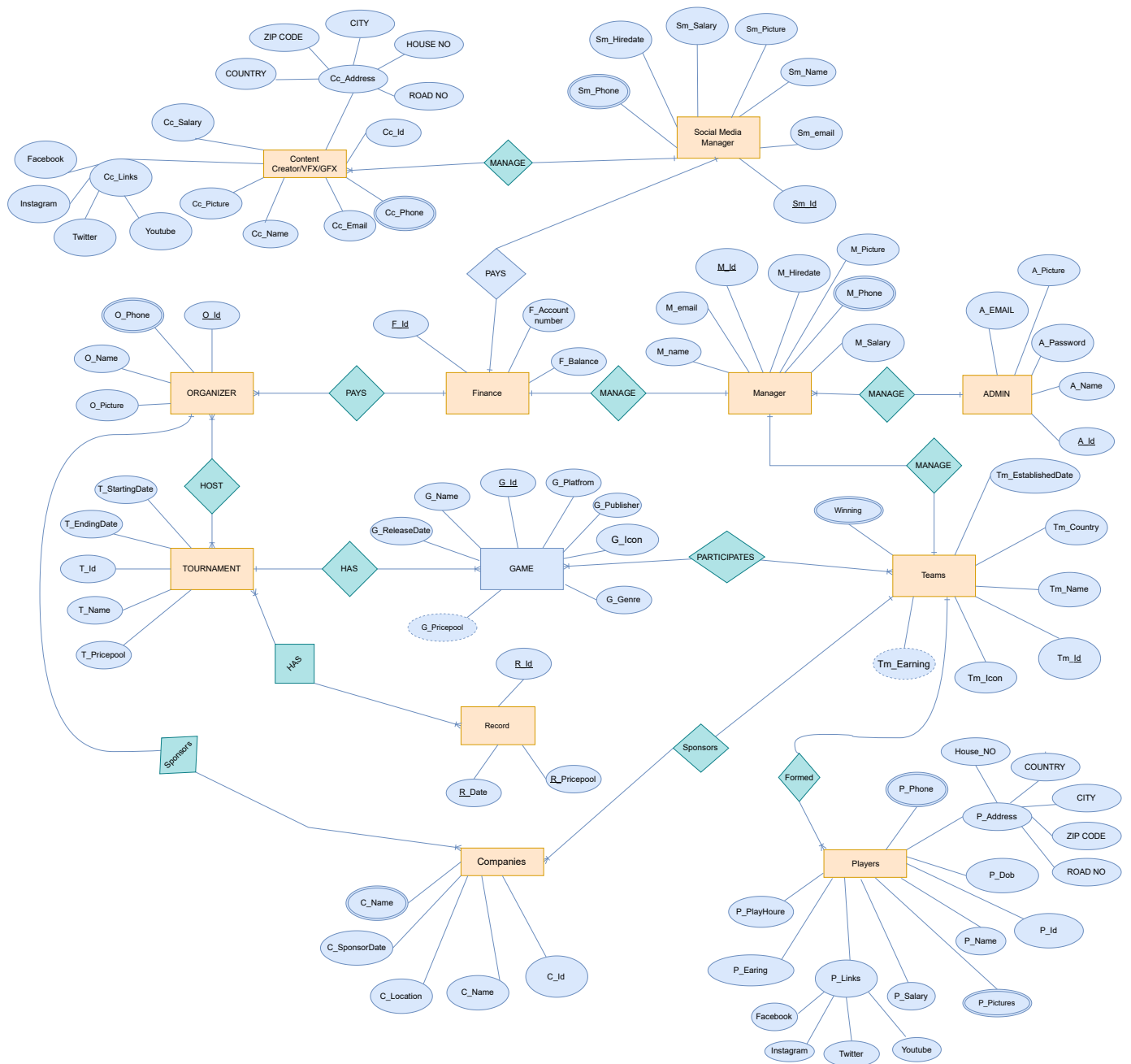
The organization hosts tournaments, bringing together teams from different games. The Tournament entity consists of attributes such as tournament ID, name, prize pool, starting date, and ending date. Each tournament features various games such as Valorant, Mobile Legends: Bang Bang (MLBB), and Rainbow Six Siege. The Game entity contains attributes like name, release date, genre, game picture, publisher, platform, game ID, and prize pool. Each tournament may have different games associated with it, creating diverse competitive environments.

To support the teams and tournaments financially, eSports FTW seeks sponsorships from different companies. The Companies entity includes attributes such as name, company ID, location, sponsor date, and phone number. Multiple companies can sponsor both teams and tournaments, forming a many-to-many relationship between the Companies, Teams, and Tournament entities.

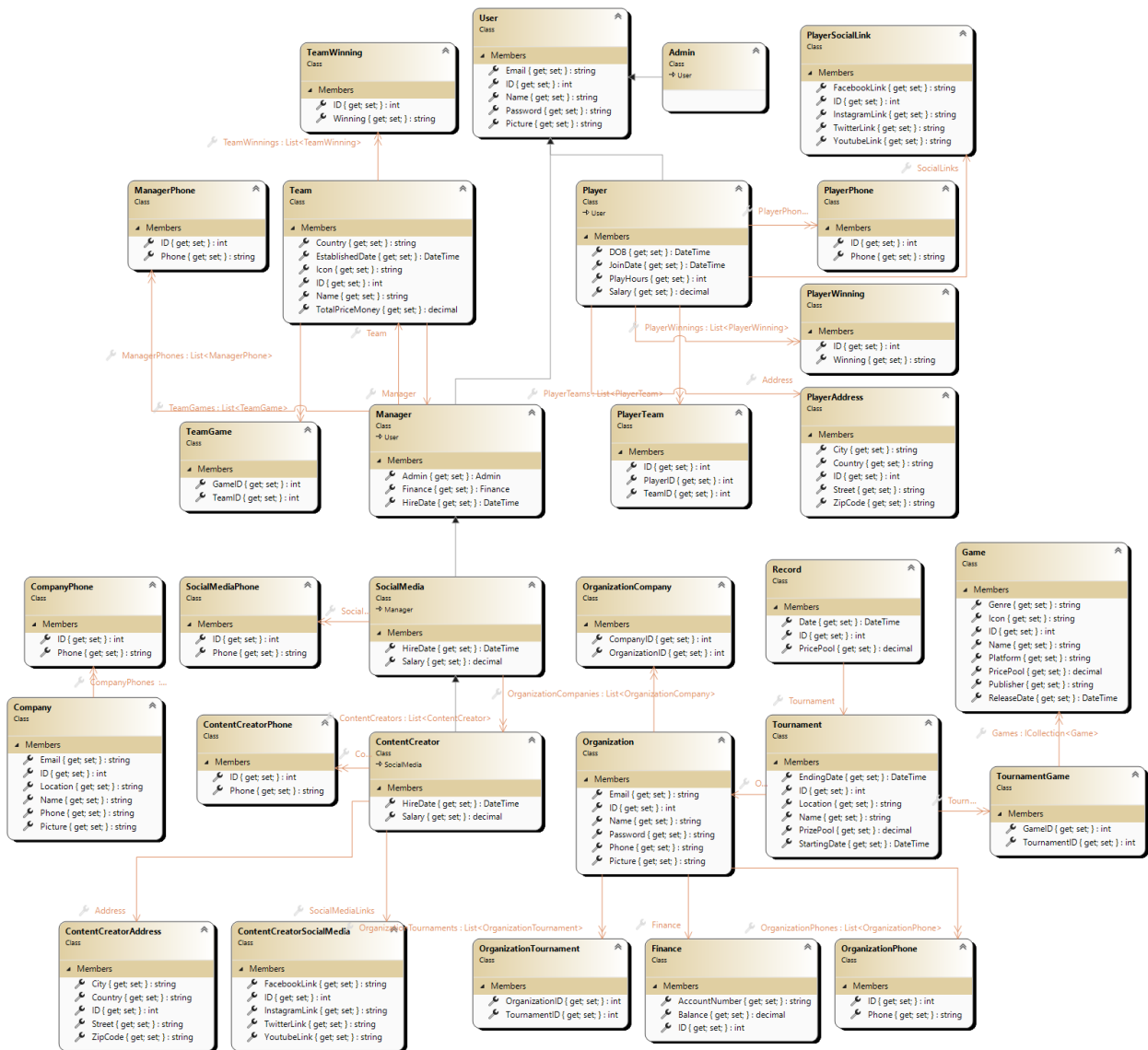
Additionally, eSports FTW employs a Social Media Manager responsible for managing the organization's online presence. The Social Media Manager entity contains attributes such as name, picture, email, manager ID, hire date, salary, phone number, and social media links (Facebook, Instagram, Twitter, YouTube). The Social Media Manager oversees the VFX/GFX and Content Creator teams, ensuring engaging content creation and visual effects. The VFX/GFX and Content Creator entities include attributes like name, picture, VFX/GFX ID, email, phone number, salary, and address (country, city, zip code, and road number).

Diagrams

2.1 ER Diagram

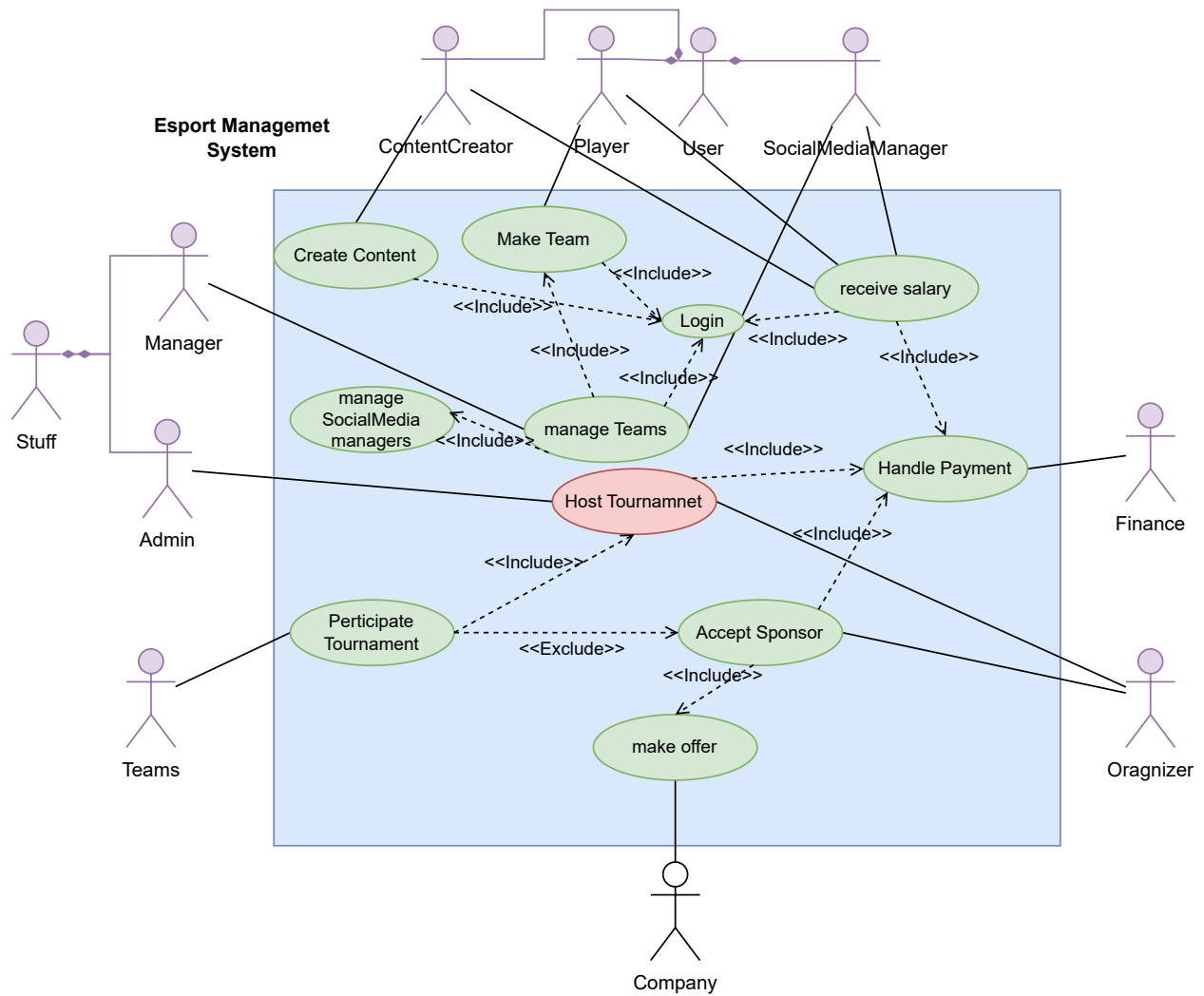


2.2 Class Diagram

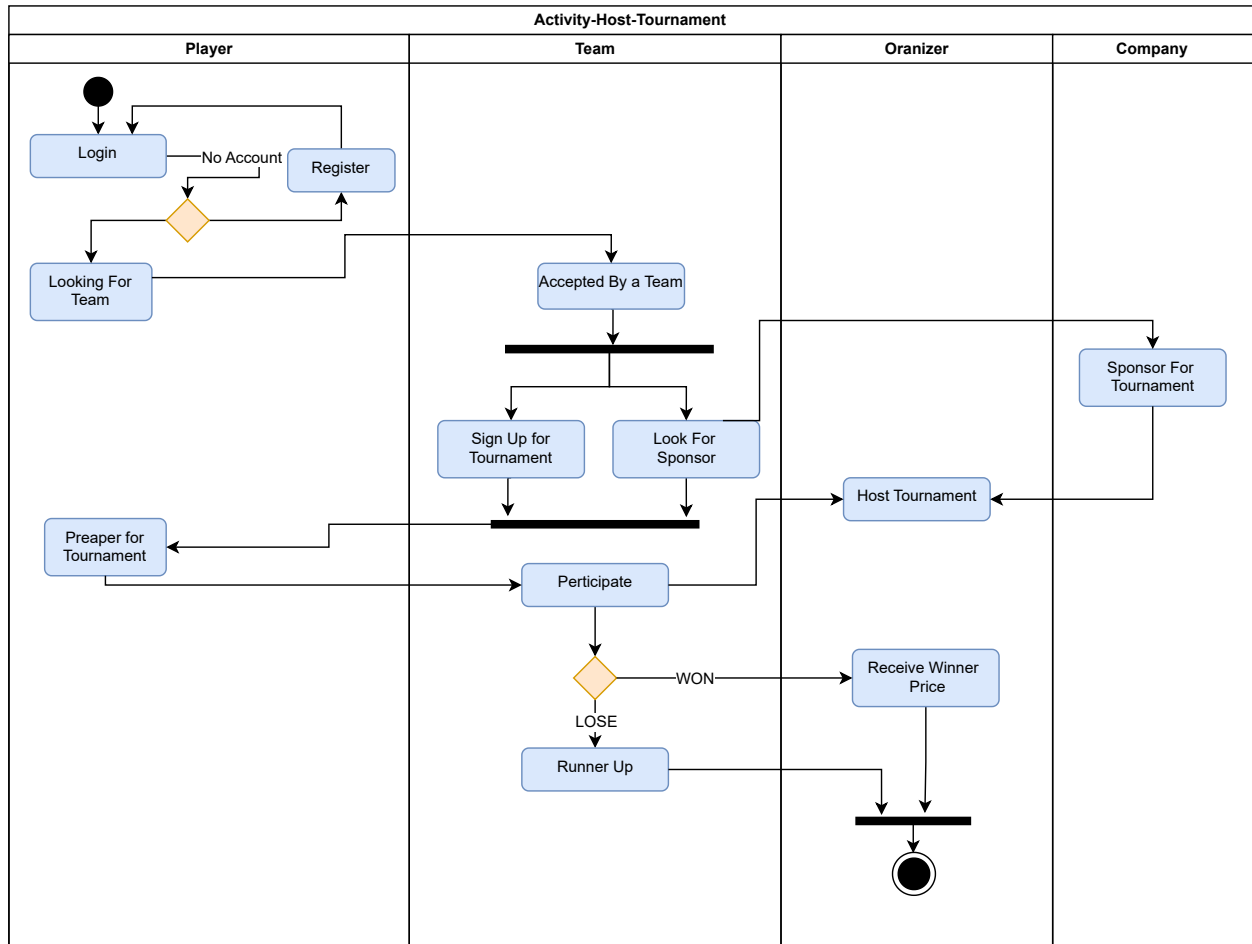


Class Diagram

2.3 Use Case Diagram



2.4 Activity Diagram



User Interface

3.1 Technologies Used

- SvelteKit
 - Tailwind CSS
-

3.1.1 Home Page



3.1.2 About Page



3.1.3 Login Page



The screenshot shows the login page of the E-Sports FTW website. The header includes the logo and navigation links for Tournament, Games, and Teams, along with a Login button. The main content area features a login form with fields for Email (containing 'yourname@email.com') and Password (containing '*****'). Below the fields are buttons for 'Sign In' and 'Forgot Password?'. The footer contains the copyright notice: '© 2023 E-SPORTS FTW. All rights reserved.'

3.1.4 Profile Page



The screenshot shows the profile page of the E-Sports FTW website. The header includes the logo and navigation links for Tournament, Games, Teams, and About, along with a Login button. The main content area features a profile card for 'Azran Hassan' with the email 'azran@gmail.com'. Below the card are sections for 'Change Name' and 'Change Password', each with an input field and a 'Save' button.

3.1.5 Tournament Page



The screenshot shows the tournament page of the E-Sports FTW website. The header includes the logo and navigation links for Tournament, Games, and Teams, along with a Login button. The main content area features a section titled 'Esports Events Statistics' with filters for Game, Game publisher, Year, and Teams. Below the filters is a search bar and a table of tournaments.

| Name | Game | Prize Pool | Platform | Event Date |
|--------------|-------------------|------------|----------|------------|
| Tournament 1 | Valorant | \$1000 | PC | 2022-01-01 |
| Tournament 2 | League of Legends | \$500 | PC | 2022-02-14 |
| Tournament 3 | Dota 2 | \$2000 | PC | 2022-03-17 |
| Tournament 4 | Overwatch | \$150 | PC | 2022-04-20 |
| Tournament 5 | Fortnite | \$1000 | PC | 2022-05-23 |

5 items

3.1.6 Team Page

ESPORTS FTW Tournament Games Teams Login

Esports Events Statistics

Game: All Games

Search tournament by name

| Name | Game | Prize Pool | Total Game | Location |
|--------|----------------------------------|------------|------------|----------|
| Team 1 | Valorant | \$1000 | 10 | NYC |
| Team 2 | Overwatch | \$2000 | 10 | VST |
| Team 3 | League of Legends | \$3000 | 10 | LA |
| Team 4 | Dota 2 | \$4000 | 10 | NYC |
| Team 5 | Counter-Strike: Global Offensive | \$5000 | 10 | NYC |

5 Items 1-5 of 5

© 2023 E-SPORTS FTW. All rights reserved.

3.1.7 Game Page

ESPORTS FTW Tournament Games Teams Login

Top esports games in 2023 by prize money

Search games by name

Period by: 2023 Sort by: Prize

| Name | Type | Prize Pool | Platform | Total Tournament |
|-------------------|---------------|------------|----------|------------------|
| Valorant | FPS | \$1000 | PC | 10 |
| League of Legends | MOBA | \$1000 | PC | 20 |
| Fortnite | Battle Royale | \$2000 | PC | 15 |
| Overwatch | FPS | \$3000 | PC | 12 |
| Dota 2 | MOBA | \$8000 | PC | 25 |

5 Items 1-5 of 5

© 2023 E-SPORTS FTW. All rights reserved.

Normalization and Schema Design

4.1 Manage branch

4.1.1 (Admin → Manager)

UNF

(Admin_ID, Admin_Name, Admin_Email, Admin_Password, Admin_Picture, Manager_ID, Manager_Name, Manager_Email, Manager_Password, Manager_Picture, Manager_Hiredate, Manager_Phone)

1NF

Phone is multi-valued attribute.

- (Admin_ID, Admin_Name, Admin_Email, Admin_Password, Admin_Picture, Manager_ID, Manager_Name, Manager_Email, Manager_Password, Manager_Picture, Manager_Hiredate, Manager_Phone)

2NF

- Admin_ID, Admin_Name, Admin_Email, Admin_Password, Admin_Picture
- Manager_ID, Manager_Name, Manager_Email, Manager_Password, Manager_Picture, Manager_Hiredate, Manager_Phone

3NF

No transitive dependencies found. Same as 2NF

Table after Normalization

- **Manager** (Manager_ID, Manager_Name, Manager_Email, Manager_Password, Manager_Picture, Manager_Hiredate, **Admin_ID**)
- **Admin** (Admin_ID, Admin_Name, Admin_Email, Admin_Password, Admin_Picture)
- **Manager_Phone** (Mp_ID, **Manager_ID**, Manager_Phone)

4.1.2 (Manager \rightarrow Finance)

UNF

(Manager_ID, Manager_Name, Manager_Email, Manager_Password, Manager_Picture, Manager_Hiredate, Manager_Phone, Finance_ID, Finance_Account_Number, Finance_Balance)

1NF

Phone is multi-valued attribute.

- (Manager_ID, Manager_Name, Manager_Email, Manager_Password, Manager_Picture, Manager_Hiredate, Manager_Phone, Finance_ID, Finance_Account_Number, Finance_Balance)

2NF

- **Manager**(Manager_ID, Manager_Name, Manager_Email, Manager_Password, Manager_Picture, Manager_Hiredate, Manager_Phone)
- **Finance**(Finance_ID, Finance_Account_Number, Finance_Balance)

3NF

No transitive dependency found. Same as 2NF.

Table after Normalization

- **Manager**(Manager_ID, Manager_Name, Manager_Email, Manager_Password, Manager_Picture, Manager_Hiredate)
- **Finance**(Finance_ID, Finance_Account_Number, Finance_Balance, **Manager_ID**)
- **Manager_Phone** (Mp_ID, **Manager_ID**, Manager_Phone)

4.1.3 (Manager \rightarrow Teams)

UNF

(Manager_ID, Manager_Name, Manager_Email, Manager_Password, Manager_Picture, Manager_Hiredate, Manager_Phone, Team_ID, Team_Name, Team_Icon, Team_established_date, Team_country, Total_Price_Money, Team_Winnig)

1NF

Winning & Phone are multi-valued attribute.

- (Manager_ID, Manager_Name, Manager_Email, Manager_Password, Manager_Picture, Manager_Hiredate, Manager_Phone, Team_ID, Team_Name, Team_Icon, Team_established_date, Team_country, Team_Total_Price_Money, Team_Winnig)

2NF

- **Manager**(Manager_ID, Manager_Name, Manager_Email, Manager_Password, Manager_Picture, Manager_Hiredate, Manager_Phone)
- **Team**(Team_ID, Team_Name, Team_Icon, Team_established_date, Total_Price_Money, Team_country)

3NF

No transitive dependency found. Same as 2NF.

Table after Normalization

- **Manager**(Manager_ID, Manager_Name, Manager_Email, Manager_Password, Manager_Picture, Manager_Hiredate)
- **Team**(Team_ID, Team_Name, Team_Icon, Team_established_date, Team_country, Total_Price_Money, **Manager_ID**)
- **Team_Winnig**(Tw_ID, **Team_ID**, Team_Winnig)
- **Manager_Phone**(Mp_ID, **Manager_ID**, Manager_Phone)

4.1.4 (SocialMedia → ContentCreator)

UNF

(SocialMedia_ID, SocialMedia_Name, SocialMedia_Email, SocialMedia_Password, SocialMedia_Picture, SocialMedia_Phone , SocialMedia_Hiredate , SocialMedia_Salary, ContentCreator_ID, ContentCreator_Name, ContentCreator_Email, ContentCreator_Password, ContentCreator_Picture, ContentCreator_Phone , ContentCreator_Hiredate , ContentCreator_Salary, ContentCreator_Facebook_Link, ContentCreator_Twitter_Link, ContentCreator_Instagram_Link, ContentCreator_Youtube_Link , ContentCreator_Country, ContentCreator_City, ContentCreator_Street, ContentCreator_Zip_Code)

1NF

Phone is multi-valued attribute.

- (SocialMedia_ID, SocialMedia_Name, SocialMedia_Email, SocialMedia_Password, SocialMedia_Picture, SocialMedia_Phone , SocialMedia_Hiredate , SocialMedia_Salary, ContentCreator_ID, ContentCreator_Name, ContentCreator_Email, ContentCreator_Password, ContentCreator_Picture, ContentCreator_Phone , ContentCreator_Hiredate , ContentCreator_Salary, ContentCreator_Facebook_Link, ContentCreator_Twitter_Link, ContentCreator_Instagram_Link, ContentCreator_Youtube_Link , ContentCreator_Country, ContentCreator_City, ContentCreator_Street, ContentCreator_Zip_Code)

2NF

- **SocialMedia** (SocialMedia_ID, SocialMedia_Name, SocialMedia_Email, SocialMedia_Password, SocialMedia_Picture, SocialMedia_Phone , SocialMedia_Hiredate , SocialMedia_Salary)
- **ContentCreator** (ContentCreator_ID, ContentCreator_Name, ContentCreator_Email, ContentCreator_Password, ContentCreator_Picture, ContentCreator_Phone , ContentCreator_Hiredate , ContentCreator_Salary, ContentCreator_Facebook_Link, ContentCreator_Twitter_Link, ContentCreator_Instagram_Link, ContentCreator_Youtube_Link , ContentCreator_Country, ContentCreator_City, ContentCreator_Street, ContentCreator_Zip_Code)

3NF

- **SocialMedia** SocialMedia_ID, SocialMedia_Name, SocialMedia_Email, SocialMedia_Password, SocialMedia_Picture , SocialMedia_Hiredate , SocialMedia_Salary)
- **ContentCreator** ContentCreator_ID, ContentCreator_Name, ContentCreator_Email, ContentCreator_Password, ContentCreator_Picture , ContentCreator_Hiredate , ContentCreator_Salary)
- **ContentCreator_SocialMedia** (**ContentCreator_ID**, **SocialMedia_ID** , ContentCreator_Facebook_Link, ContentCreator_Twitter_Link, ContentCreator_Instagram_Link, ContentCreator_Youtube_Link)
- **ContentCreator_Address** (**ContentCreator_ID**, ContentCreator_Country, ContentCreator_City, ContentCreator_Street, ContentCreator_Zip_Code)

Table after Normalization

- **SocialMedia** (SocialMedia_ID, SocialMedia_Name, SocialMedia_Email, SocialMedia_Password, SocialMedia_Picture , SocialMedia_Hiredate , SocialMedia_Salary)
- **ContentCreator** (ContentCreator_ID, ContentCreator_Name, ContentCreator_Email, ContentCreator_Password, ContentCreator_Picture , ContentCreator_Hiredate , ContentCreator_Salary)
- **ContentCreator_SocialMedia** (Ccs_ID, **ContentCreator_ID** , ContentCreator_Facebook_Link, ContentCreator_Twitter_Link, ContentCreator_Instagram_Link, ContentCreator_Youtube_Link)
- **ContentCreator_Address** (Cca_ID, **ContentCreator_ID**, ContentCreator_Country, ContentCreator_City, ContentCreator_Street, ContentCreator_Zip_Code)
- **ContentCreator_Phone** (Ccp_ID, **ContentCreator_ID**, ContentCreator_Phone)
- **SocialMedia_Phone** (Smp_ID, **SocialMedia_ID**, SocialMedia_Phone)
- **ContentCreator_SocialMedia_Phone** (Ccp_ID, **ContentCreator_ID**, ContentCreator_Phone)

4.2 Pay branch

4.2.1 (Finance \rightarrow SocialMedia)

UNF

(Finance_ID, Finance_Account_Number, Finance_Balance, SocialMedia_ID, SocialMedia_Name, SocialMedia_Email, SocialMedia_Password, SocialMedia_Picture, SocialMedia_Phone , SocialMedia_Hiredate , SocialMedia_Salary)

1NF

Phone is multi-valued attribute.

- (Finance_ID, Finance_Account_Number, Finance_Balance, SocialMedia_ID, SocialMedia_Name, SocialMedia_Email, SocialMedia_Password, SocialMedia_Picture, SocialMedia_Phone , SocialMedia_Hiredate , SocialMedia_Salary)

2NF

- **Finance**(Finance_ID, Finance_Account_Number, Finance_Balance)
- **SocialMedia**(SocialMedia_ID, SocialMedia_Name, SocialMedia_Email, SocialMedia_Password, SocialMedia_Picture, SocialMedia_Phone , SocialMedia_Hiredate , SocialMedia_Salary)

3NF

No transitive dependency found. Same as 2NF.

Table after Normalization

- **Finance**(Finance_ID, Finance_Account_Number, Finance_Balance)
- **SocialMedia**(SocialMedia_ID, SocialMedia_Name, SocialMedia_Email, SocialMedia_Password, SocialMedia_Picture , SocialMedia_Hiredate , SocialMedia_Salary, **Finance_ID**)
- **SocialMedia_Phone** (Smp_ID, **SocialMedia_ID**, SocialMedia_Phone)

4.2.2 (Finance \rightarrow organization)

UNF

(Finance_ID, Finance_Account_Number, Finance_Balance, Organization_ID, Organization_Name, Organization_Email, Organization_Password, Organization_Picture, Organization_Phone)

1NF

Phone is multi-valued attribute.

- (Finance_ID, Finance_Account_Number, Finance_Balance, Organization_ID, Organization_Name, Organization_Email, Organization_Password, Organization_Picture, Organization_Phone)

2NF

- **Finance**(Finance_ID, Finance_Account_Number, Finance_Balance)
- **Organization**(Organization_ID, Organization_Name, Organization_Email, Organization_Password, Organization_Picture, Organization_Phone)

3NF

No transitive dependency found. Same as 2NF.

Table after Normalization

- **Finance**(Finance_ID, Finance_Account_Number, Finance_Balance)
- **Organization**(Organization_ID, Organization_Name, Organization_Email, Organization_Password, Organization_Picture, Organization_Phone, **Finance_ID**)
- **Organization_Phone** (Op_ID, **Organization_ID**, Organization_Phone)

4.3 Formed branch

4.3.1 (Teams \rightarrow Player)

UNF

(Team_ID, Team_Name, Team_Icon, Team_established_date, Team_country, Total_Earning,

Team_Winnig, Player_ID, Player_Name, Player_Email, Player_Password, Player_Picture, Player_JoinDate, Player_Phone, Player_Salary, Player_Winnig_Money, Player_Play_Hours, Player_country, Player_City, Player_Street, Player_Zip_Code, Player_DOB, Player_Facebook_Link, Player_Instagram_Link, Player_Twitter_Link, Player_Youtube_Link)

1NF

Phone & Wining number are multi-valued attribute.

- (Team_ID, Team_Name, Team_Icon, Team_established_date, Team_country, Total_Earning, Team_Winnig, Player_ID, Player_Name, Player_Email, Player_Password, Player_Picture, Player_JoinDate, Player_Phone, Player_Salary, Player_Winnig_Money, Player_Play_Hours, Player_country, Player_City, Player_Street, Player_Zip_Code, Player_DOB, Player_Facebook_Link, Player_Instagram_Link, Player_Twitter_Link, Player_Youtube_Link)

2NF

- **Team**(Team_ID, Team_Name, Team_Icon, Team_established_date, Team_country, Total_Earning, Team_Winnig)
- **Player**(Player_ID, Player_Name, Player_Email, Player_Password, Player_Picture, Player_JoinDate, Player_Salary, Player_Winnig_Money, Player_Play_Hours, Player_country, Player_City, Player_Street, Player_Zip_Code, Player_DOB, Player_Facebook_Link, Player_Instagram_Link, Player_Twitter_Link, Player_Youtube_Link)

3NF

- **Team**(Team_ID, Team_Name, Team_Icon, Team_established_date, Team_country, Total_Earning, Team_Winnig)
- **Player**(Player_ID, Player_Name, Player_Email, Player_Password, Player_Picture, Player_JoinDate, Player_Salary, Player_Winnig_Money, Player_Play_Hours, Player_DOB)
- **Player_Address** (Pa_ID, **Player_ID**, Player_country, Player_City, Player_Street, Player_Zip_Code)
- **Player_Social_Link** (PsL_ID, **Player_ID**, Player_Facebook_Link, Player_Instagram_Link, Player_Twitter_Link, Player_Youtube_Link)

Table after Normalization

- **Team**(Team_ID, Team_Name, Team_Icon, Team_established_date, Team_country, Total_Earning, Team_Winnig)
- **Player**(Player_ID, Player_Name, Player_Email, Player_Password, Player_Picture, Player_JoinDate, Player_Salary, Player_Play_Hours, Player_DOB)

- **Player_Address** (Pa_ID, **Player_ID**, Player_country, Player_City, Player_Street, Player_Zip_Code)
- **Player_Social_Link** (Psl_ID, **Player_ID**, Player_Facebook_Link, Player_Instagram_Link, Player_Twitter_Link, Player_Youtube_Link)
- **Player_Phone** (Pp_ID, **Player_ID**, Player_Phone)
- **Player_Wining** (Pw_ID, **Player_ID**, Player_Winnig)
- **Player_Team** (Pt_ID, **Player_ID**, **Team_ID**)

4.4 Has branch

4.4.1 (Record \rightarrow Tournament)

UNF

(Record_ID, Record_Date, Record_Price_Pool, Tournament_ID, Tournament_Name, Tournament_StartingDate, Tournament_EndingDate, Tournament_Location, Tournament_Prize_Pool)

1NF

- (Record_ID, Record_Date, Record_Price_Pool, Tournament_ID, Tournament_Name, Tournament_StartingDate, Tournament_EndingDate, Tournament_Location, Tournament_Prize_Pool)

2NF

- **Record**(Record_ID, Record_Date, Record_Price_Pool, **Tournament_ID**)
- **Tournament**(Tournament_ID, Tournament_Name, Tournament_StartingDate, Tournament_EndingDate, Tournament_Location, Tournament_Prize_Pool)

3NF

No transitive dependency found. Same as 2NF.

Table after Normalization

- **Record**(Record_ID, Record_Date, Record_Price_Pool, **Tournament_ID**)
- **Tournament**(Tournament_ID, Tournament_Name, Tournament_StartingDate, Tournament_EndingDate, Tournament_Location, Tournament_Prize_Pool)

4.4.2 (Tournament → Game)

UNF

(Tournament_ID, Tournament_Name, Tournament_StartingDate, Tournament_EndingDate, Tournament_Location, Tournament_Prize_Pool, Game_ID, Game_Name, Game_Icon, Game_ReleaseDate, Game_Platform, Game_PricePool, Game_Genre, Game_Publisher)

1NF

- (Tournament_ID, Tournament_Name, Tournament_StartingDate, Tournament_EndingDate, Tournament_Location, Tournament_Prize_Pool, Game_ID, Game_Name, Game_Icon, Game_ReleaseDate, Game_Platform, Game_PricePool, Game_Genre, Game_Publisher)

2NF

- **Tournament**(Tournament_ID, Tournament_Name, Tournament_StartingDate, Tournament_EndingDate, Tournament_Location, Tournament_Prize_Pool)
- **Game**(Game_ID, Game_Name, Game_Icon, Game_ReleaseDate, Game_Platform, Game_PricePool, Game_Genre, Game_Publisher)

3NF

No transitive dependency found. Same as 2NF.

Table after Normalization

- **Tournament**(Tournament_ID, Tournament_Name, Tournament_StartingDate, Tournament_EndingDate, Tournament_Location, Tournament_Prize_Pool)
- **Game**(Game_ID, Game_Name, Game_Icon, Game_ReleaseDate, Game_Platform, Game_PricePool, Game_Genre, Game_Publisher)
- **Tournament_Game**(Tournament_ID, Game_ID)

4.5 Participate branch

4.5.1 (Teams \rightarrow Game)

UNF

(Team_ID, Team_Name, Team_Icon, Team_established_date, Team_country, Total_Earning, Team_Winnig, Game_ID, Game_Name, Game_Icon, Game_ReleaseDate, Game_Platform, Game_PricePool, Game_Genre, Game_Publisher)

1NF

- (Team_ID, Team_Name, Team_Icon, Team_established_date, Team_country, Total_Earning, Team_Winnig, Game_ID, Game_Name, Game_Icon, Game_ReleaseDate, Game_Platform, Game_PricePool, Game_Genre, Game_Publisher)

2NF

- **Team**(Team_ID, Team_Name, Team_Icon, Team_established_date, Team_country, Total_Earning, Team_Winnig)
- **Game**(Game_ID, Game_Name, Game_Icon, Game_ReleaseDate, Game_Platform, Game_PricePool, Game_Genre, Game_Publisher)

3NF

No transitive dependency found. Same as 2NF.

Table after Normalization

- **Team**(Team_ID, Team_Name, Team_Icon, Team_established_date, Team_country, Total_Earning, Team_Winnig)
- **Game**(Game_ID, Game_Name, Game_Icon, Game_ReleaseDate, Game_Platform, Game_PricePool, Game_Genre, Game_Publisher)
- **Team_Game**(Team_ID, Game_ID)

4.6 Host branch

4.6.1 (Organizer \rightarrow Tournament)

UNF

(Tournament_ID, Tournament_Name, Tournament_StartingDate, Tournament_EndingDate, Tournament_Location, Tournament_Prize_Pool, Organization_ID, Organization_Name, Organization_Email, Organization_Password, Organization_Picture, Organization_Phone)

1NF

Phone is a multi-valued attribute.

- (Tournament_ID, Tournament_Name, Tournament_StartingDate, Tournament_EndingDate, Tournament_Location, Tournament_Prize_Pool, Organization_ID, Organization_Name, Organization_Email, Organization_Password, Organization_Picture, Organization_Phone)

2NF

- **Tournament**(Tournament_ID, Tournament_Name, Tournament_StartingDate, Tournament_EndingDate, Tournament_Location, Tournament_Prize_Pool, Organization_ID)
- **Organization**(Organization_ID, Organization_Name, Organization_Email, Organization_Password, Organization_Picture, Organization_Phone)

3NF

No transitive dependency found. Same as 2NF.

Table after Normalization

- **Tournament**(Tournament_ID, Tournament_Name, Tournament_StartingDate, Tournament_EndingDate, Tournament_Location, Tournament_Prize_Pool, Organization_ID)
- **Organization**(Organization_ID, Organization_Name, Organization_Email, Organization_Password, Organization_Picture)
- **Organization_Phone**(Organization_ID, Organization_Phone)
- **Organization_Tournament**(Organization_ID, Tournament_ID)

4.7 Sponsor branch

4.7.1 (Companies \rightarrow Organization)

UNF

(Organization_ID, Organization_Name, Organization_Email, Organization_Picture, Organization_Phone, Company_ID, Company_Name, Company_Email, Company_Picture, Company_Phone, location)

1NF

phone is a multi-value attribute.

- (Organization_ID, Organization_Name, Organization_Email, Organization_Picture, Organization_Phone, Company_ID, Company_Name, Company_Email, Company_Picture, Company_Phone, location)

2NF

- **Organization**(Organization_ID, Organization_Name, Organization_Email, Organization_Picture, Organization_Phone, Company_ID)
- **Company**(Company_ID, Company_Name, Company_Email, Company_Picture, Company_Phone, location)

3NF

No transitive dependency found. Same as 2NF.

Table after Normalization

- **Organization**(Organization_ID, Organization_Name, Organization_Email, Organization_Picture, Company_ID)
- **Organization_Phone**(Organization_ID, Organization_Phone)
- **Company**(Company_ID, Company_Name, Company_Email, Company_Picture, Company_Phone, location)
- **Company_Phone**(Company_ID, Company_Phone)
- **Organization_Company**(Organization_ID, Company_ID)

4.7.2 (Companies \rightarrow Teams)

UNF

(Team_ID, Team_Name, Team_Icon, Team_established_date, Team_country, Total_Earning, Team_Winnig, Company_ID, Company_Name, Company_Email, Company_Password, Company_Picture, Company_Phone, Company_location)

1NF

Phone is multi-value attribute.

- (Team_ID, Team_Name, Team_Icon, Team_established_date, Team_country, Total_Earning, Team_Winnig, Company_ID, Company_Name, Company_Email, Company_Password, Company_Picture, Company_Phone, Company_location)

2NF

- **Team**(Team_ID, Team_Name, Team_Icon, Team_established_date, Team_country, Total_Earning, Team_Winnig)
- **Company**(Company_ID, Company_Name, Company_Email, Company_Password, Company_Picture, Company_Phone, Company_location)

3NF

No transitive dependency found. Same as 2NF.

Table after Normalization

- **Team**(Team_ID, Team_Name, Team_Icon, Team_established_date, Team_country, Total_Earning, Team_Winnig)
- **Company**(Company_ID, Company_Name, Company_Email, Company_Password, Company_Picture, Company_Phone, Company_location)
- **Team_Company**(Team_ID, Company_ID)
- **Company_Phone**(Company_ID, Company_Phone)

4.8 Temporary Tables

1. **Manager** (Manager_ID, Manager_Name, Manager_Email, Manager_Password, Manager_Picture, Manager_Hiredate, **Admin_ID**)
2. **Admin** (Admin_ID, Admin_Name, Admin_Email, Admin_Password, Admin_Picture)
3. **Manager_Phone** (Mp_ID, **Manager_ID**, Manager_Phone)
4. **Manager**(Manager_ID, Manager_Name, Manager_Email, Manager_Password, Manager_Picture, Manager_Hiredate)
5. **Finance**(Finance_ID, Finance_Account_Number, Finance_Balance, **Manager_ID**)
6. **Manager_Phone** (Mp_ID, **Manager_ID**, Manager_Phone)
7. **Manager**(Manager_ID, Manager_Name, Manager_Email, Manager_Password, Manager_Picture, Manager_Hiredate)
8. **Team**(Team_ID, Team_Name, Team_Icon, Team_established_date, Team_country, Total_Price_Money, **Manager_ID**)
9. **Team_Winnig**(Tw_ID, **Team_ID**, Team_Winnig)
10. **Manager_Phone** (Mp_ID, **Manager_ID**, Manager_Phone)
11. **SocialMedia** (SocialMedia_ID, SocialMedia_Name, SocialMedia_Email, SocialMedia_Password, SocialMedia_Picture , SocialMedia_Hiredate , SocialMedia_Salary)
12. **ContentCreator** (ContentCreator_ID, ContentCreator_Name, ContentCreator_Email, ContentCreator_Password, ContentCreator_Picture , ContentCreator_Hiredate , ContentCreator_Salary)
13. **ContentCreator_SocialMedia** (Ccs_ID, **ContentCreator_ID** , ContentCreator_Facebook_Link, ContentCreator_Twitter_Link, ContentCreator_Instagram_Link, ContentCreator_Youtube_Link)
14. **ContentCreator_Address** (Cca_ID, **ContentCreator_ID**, ContentCreator_Country, ContentCreator_City, ContentCreator_Street, ContentCreator_Zip_Code)
15. **ContentCreator_Phone** (Ccp_ID, **ContentCreator_ID**, ContentCreator_Phone)
16. **SocialMedia_Phone** (Smp_ID, **SocialMedia_ID**, SocialMedia_Phone)
17. **Finance**(Finance_ID, Finance_Account_Number, Finance_Balance)
18. **Organization**(Organization_ID, Organization_Name, Organization_Email, Organization_Password, Organization_Picture, Organization_Phone, **Finance_ID**)
19. **Organization_Phone** (Op_ID, **Organization_ID**, Organization_Phone)
20. **Finance**(Finance_ID, Finance_Account_Number, Finance_Balance)
21. **SocialMedia**(SocialMedia_ID, SocialMedia_Name, SocialMedia_Email, SocialMedia_Password, SocialMedia_Picture , SocialMedia_Hiredate , SocialMedia_Salary, **Finance_ID**)
22. **SocialMedia_Phone** (Smp_ID, **SocialMedia_ID**, SocialMedia_Phone)

23. ~~**Team**(Team_ID, Team_Name, Team_Icon, Team_established_date, Team_country, Total_Earning, Team_Winnig)~~
24. **Player**(Player_ID, Player_Name, Player_Email, Player_Password, Player_Picture, Player_JoinDate, Player_Salary, Player_Play_Hours, Player_DOB)
25. **Player_Address** (Pa_ID, **Player_ID**, Player_country, Player_City, Player_Street, Player_Zip_Code)
26. **Player_Social_Link** (Psl_ID, **Player_ID**, Player_Facebook_Link, Player_Instagram_Link, Player_Twitter_Link, Player_Youtube_Link)
27. **Player_Phone** (Pp_ID, **Player_ID**, Player_Phone)
28. **Player_Wining** (Pw_ID, **Player_ID**, Player_Winnig)
29. **Player_Team** (Pt_ID, **Player_ID**, **Team_ID**)
30. **Record**(Record_ID, Record_Date, Record_Price_Pool, **Tournament_ID**)
31. **Tournament**(Tournament_ID, Tournament_Name, Tournament_StartingDate, Tournament_EndingDate, Tournament_Location, Tournament_Prize_Pool)
32. **Tournament**(Tournament_ID, Tournament_Name, Tournament_StartingDate, Tournament_EndingDate, Tournament_Location, Tournament_Prize_Pool)
33. **Game**(Game_ID, Game_Name, Game_Icon, Game_ReleaseDate, Game_Platform, Game_PricePool, Game_Genre, Game_Publisher)
34. **Tournament_Game**(Tournament_ID, Game_ID)
35. **Tournament**(Tournament_ID, Tournament_Name, Tournament_StartingDate, Tournament_EndingDate, Tournament_Location, Tournament_Prize_Pool, Organization_ID)
36. ~~**Organization**(Organization_ID, Organization_Name, Organization_Email, Organization_Password, Organization_Picture)~~
37. ~~**Organization_Phone**(Organization_ID, Organization_Phone)~~
38. ~~**Organization_Tournament**(Organization_ID, Tournament_ID)~~
39. ~~**Team**(Team_ID, Team_Name, Team_Icon, Team_established_date, Team_country, Total_Earning, Team_Winnig)~~
40. ~~**Game**(Game_ID, Game_Name, Game_Icon, Game_ReleaseDate, Game_Platform, Game_PricePool, Game_Genre, Game_Publisher)~~
41. **Team_Game**(Team_ID, Game_ID)
42. ~~**Organization**(Organization_ID, Organization_Name, Organization_Email, Organization_Picture, Company_ID)~~
43. ~~**Organization_Phone**(Organization_ID, Organization_Phone)~~
44. **Company**(Company_ID, Company_Name, Company_Email, Company_Picture, Company_Phone, location)

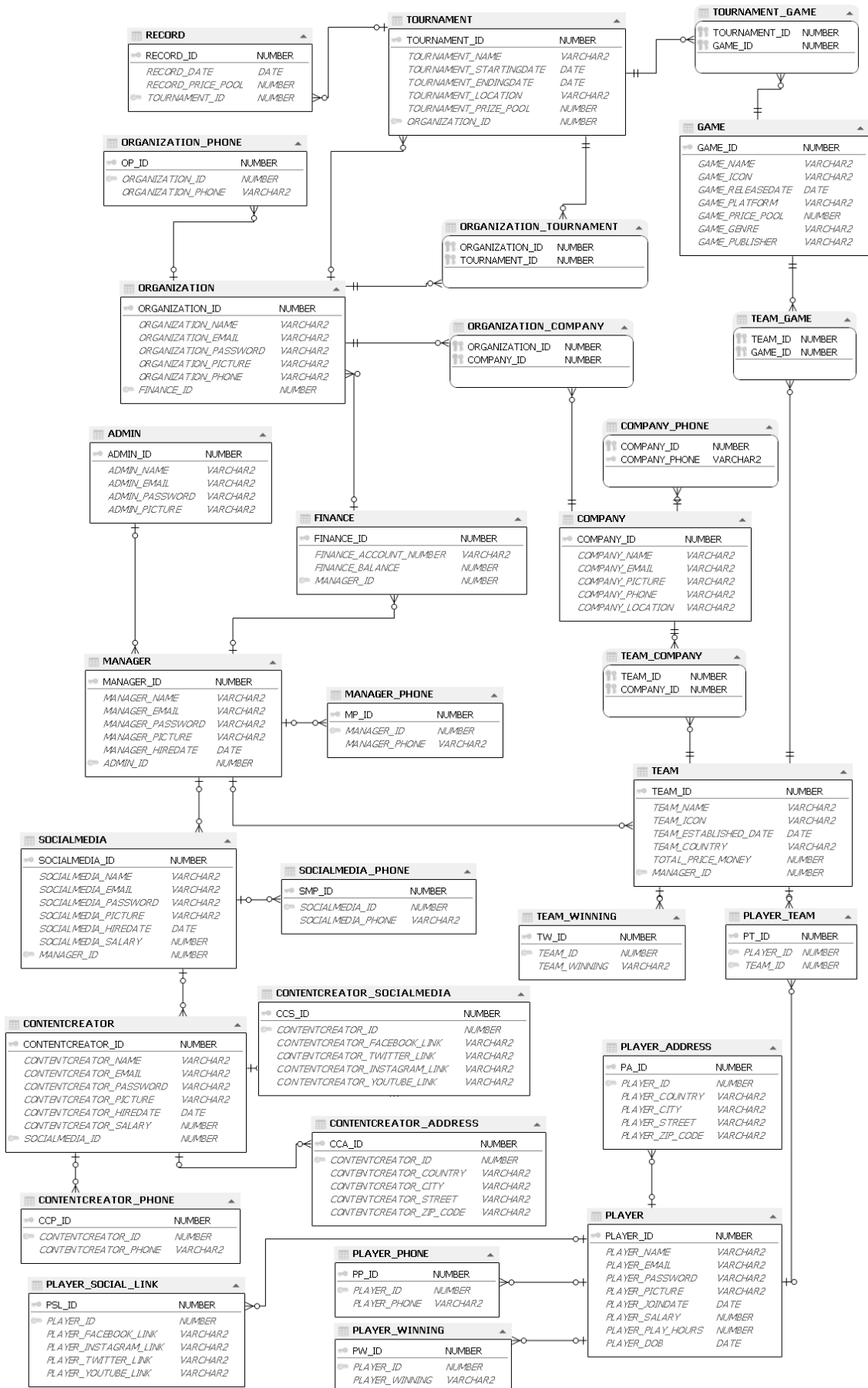
45. **Company_Phone**(Company_ID, Company_Phone)
46. **Organization_Company**(Organization_ID, Company_ID)
47. ~~**Team**(Team_ID, Team_Name, Team_Icon, Team_established_date, Team_country, Total_Earning, Team_Winnig)~~
48. ~~**Company**(Company_ID, Company_Name, Company_Email, Company_Password, Company_Picture, Company_Phone, Company_Location)~~
49. **Team_Company**(Team_ID, Company_ID)
50. ~~**Company_Phone**(Company_ID, Company_Phone)~~

4.9 Final Tables

1. **Manager** (Manager_ID, Manager_Name, Manager_Email, Manager_Password, Manager_Picture, Manager_Hiredate, **Admin_ID**)
2. **Admin** (Admin_ID, Admin_Name, Admin_Email, Admin_Password, Admin_Picture)
3. **Manager_Phone** (Mp_ID, **Manager_ID**, Manager_Phone)
4. **Finance**(Finance_ID, Finance_Account_Number, Finance_Balance, **Manager_ID**)
5. **Team**(Team_ID, Team_Name, Team_Icon, Team_established_date, Team_country, Total_Price_Money, **Manager_ID**)
6. **Team_Winnig**(Tw_ID, **Team_ID**, Team_Winnig)
7. **SocialMedia** (SocialMedia_ID, SocialMedia_Name, SocialMedia_Email, SocialMedia_Password, SocialMedia_Picture , SocialMedia_Hiredate , SocialMedia_Salary)
8. **ContentCreator** (ContentCreator_ID, ContentCreator_Name, ContentCreator_Email, ContentCreator_Password, ContentCreator_Picture , ContentCreator_Hiredate , ContentCreator_Salary)
9. **ContentCreator_SocialMedia** (Ccs_ID, **ContentCreator_ID** , ContentCreator_Facebook_Link, ContentCreator_Twitter_Link, ContentCreator_Instagram_Link, ContentCreator_Youtube_Link)
10. **ContentCreator_Address** (Cca_ID, **ContentCreator_ID**, ContentCreator_Country, ContentCreator_City, ContentCreator_Street, ContentCreator_Zip_Code)
11. **ContentCreator_Phone** (Ccp_ID, **ContentCreator_ID**, ContentCreator_Phone)
12. **SocialMedia_Phone** (Smp_ID, **SocialMedia_ID**, SocialMedia_Phone)

13. **Organization**(Organization_ID, Organization_Name, Organization_Email, Organization_Password, Organization_Picture, Organization_Phone, **Finance_ID**)
14. **Organization_Phone** (Op_ID, **Organization_ID**, Organization_Phone)
15. **Player**(Player_ID, Player_Name, Player_Email, Player_Password, Player_Picture, Player_JoinDate, Player_Salary, Player_Play_Hours, Player_DOB)
16. **Player_Address** (Pa_ID, **Player_ID**, Player_country, Player_City, Player_Street, Player_Zip_Code)
17. **Player_Social_Link** (Psl_ID, **Player_ID**, Player_Facebook_Link, Player_Instagram_Link, Player_Twitter_Link, Player_Youtube_Link)
18. **Player_Phone** (Pp_ID, **Player_ID**, Player_Phone)
19. **Player_Wining** (Pw_ID, **Player_ID**, Player_Winnig)
20. **Player_Team** (Pt_ID, **Player_ID**, **Team_ID**)
21. **Record**(Record_ID, Record_Date, Record_Price_Pool, **Tournament_ID**)
22. **Tournament**(Tournament_ID, Tournament_Name, Tournament_StartingDate, Tournament_EndingDate, Tournament_Location, Tournament_Prize_Pool)
23. **Game**(Game_ID, Game_Name, Game_Icon, Game_ReleaseDate, Game_Platform, Game_PricePool, Game_Genre, Game_Publisher)
24. **Tournament_Game**(Tournament_ID, Game_ID)
25. **Organization_Tournament**(Organization_ID, Tournament_ID)
26. **Team_Game**(Team_ID, Game_ID)
27. **Company**(Company_ID, Company_Name, Company_Email, Company_Picture, Company_Phone, location)
28. **Company_Phone**(Company_ID, Company_Phone)
29. **Organization_Company**(Organization_ID, Company_ID)
30. **Team_Company**(Team_ID, Company_ID)

4.10 Schema Diagram



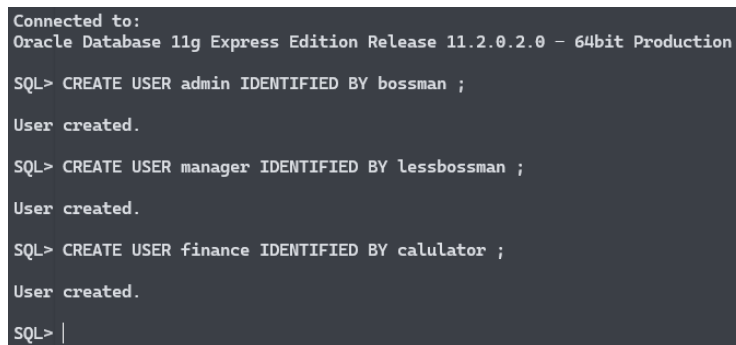
Schema Diagram

SQL Queries

5.1 User Creation

```
1 -- Create the users
2 CREATE USER admin IDENTIFIED BY bossman ;
3 CREATE USER manager IDENTIFIED BY lessbossman ;
4 CREATE USER finance IDENTIFIED BY calculator ;
```

User Creation



```
Connected to:
Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production

SQL> CREATE USER admin IDENTIFIED BY bossman ;

User created.

SQL> CREATE USER manager IDENTIFIED BY lessbossman ;

User created.

SQL> CREATE USER finance IDENTIFIED BY calculator ;

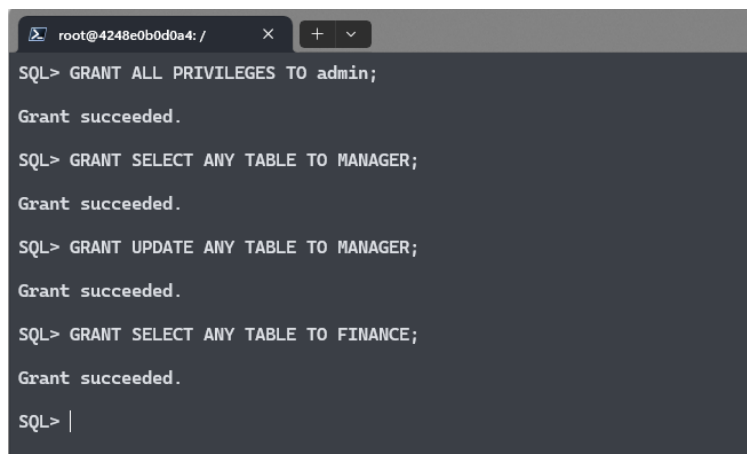
User created.

SQL> |
```

Screenshot of User Creation from SQL Plus

```
1 -- Grant All privileges to admin user
2 GRANT ALL PRIVILEGES TO admin;
3
4 -- Grant DML privileges to manager user
5 GRANT SELECT ANY TABLE TO MANAGER;
6 GRANT UPDATE ANY TABLE TO MANAGER;
7
8 -- Grant Read only privileges to finance user
9 GRANT SELECT ANY TABLE TO FINANCE;
```

User Creation



```
root@4248e0b0d0a4: /
SQL> GRANT ALL PRIVILEGES TO admin;

Grant succeeded.

SQL> GRANT SELECT ANY TABLE TO MANAGER;

Grant succeeded.

SQL> GRANT UPDATE ANY TABLE TO MANAGER;

Grant succeeded.

SQL> GRANT SELECT ANY TABLE TO FINANCE;

Grant succeeded.

SQL> |
```

Screenshot of User Privileges from SQL Plus

5.2 Table Creation

```

1 CREATE TABLE Admin (
2     Admin_ID INT PRIMARY KEY,
3     Admin_Name VARCHAR(100),
4     Admin_Email VARCHAR(100),
5     Admin_Password VARCHAR(100),
6     Admin_Picture VARCHAR(100)
7 );

```

Create Admin table

| Results | Explain | Describe | Saved SQL | History | | | | | |
|--|----------------|-----------|-----------|-----------|-------|-------------|----------|---------|---------|
| Object Type TABLE ? Object ADMIN ? | | | | | | | | | |
| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
| ADMIN | ADMIN_ID | NUMBER | 22 | - | 0 | 1 | - | - | - |
| | ADMIN_NAME | VARCHAR2 | 100 | - | - | - | ✓ | - | - |
| | ADMIN_EMAIL | VARCHAR2 | 100 | - | - | - | ✓ | - | - |
| | ADMIN_PASSWORD | VARCHAR2 | 100 | - | - | - | ✓ | - | - |
| | ADMIN_PICTURE | VARCHAR2 | 100 | - | - | - | ✓ | - | - |

Admin table description

```

1 CREATE TABLE Manager (
2     Manager_ID INT PRIMARY KEY,
3     Manager_Name VARCHAR(100),
4     Manager_Email VARCHAR(100),
5     Manager_Password VARCHAR(100),
6     Manager_Picture VARCHAR(100),
7     Manager_Hiredate DATE,
8     Admin_ID INT,
9     FOREIGN KEY (Admin_id) REFERENCES Manager (Admin_ID)
10 );

```

Create Manager table

| Object Type | | TABLE | | Object | | MANAGER | | | |
|-------------|------------------|-----------|--------|-----------|-------|-------------|----------|---------|---------|
| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
| MANAGER | MANAGER_ID | NUMBER | 22 | - | 0 | 1 | - | - | - |
| | MANAGER_NAME | VARCHAR2 | 100 | - | - | - | ✓ | - | - |
| | MANAGER_EMAIL | VARCHAR2 | 100 | - | - | - | ✓ | - | - |
| | MANAGER_PASSWORD | VARCHAR2 | 100 | - | - | - | ✓ | - | - |
| | MANAGER_PICTURE | VARCHAR2 | 100 | - | - | - | ✓ | - | - |
| | MANAGER_HIREDATE | DATE | 7 | - | - | - | ✓ | - | - |
| | ADMIN_ID | NUMBER | 22 | - | 0 | - | ✓ | - | - |

Manager table description

```

1 CREATE TABLE Manager_Phone (
2     Mp_ID INT PRIMARY KEY,
3     Manager_ID INT,
4     Manager_Phone VARCHAR(20),
5     FOREIGN KEY (Manager_ID) REFERENCES Manager (Manager_ID)
6 );

```

Create Manager Phone table

| Results | Explain | Describe | Saved SQL | History | | | | | |
|---------------|---------------|-----------|-----------|-----------|-----------------|-------------|----------|---------|---------|
| Object Type | | TABLE ? | Object | | MANAGER_PHONE ? | | | | |
| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
| MANAGER_PHONE | MP_ID | NUMBER | 22 | - | 0 | 1 | - | - | - |
| | MANAGER_ID | NUMBER | 22 | - | 0 | - | ✓ | - | - |
| | MANAGER_PHONE | VARCHAR2 | 20 | - | - | - | ✓ | - | - |

Manager Phone table description

```

1 CREATE TABLE Finance (
2     Finance_ID INT PRIMARY KEY,
3     Finance_Account_Number VARCHAR(100),
4     Finance_Balance DECIMAL(10, 2),
5     Manager_ID INT,
6     FOREIGN KEY (Manager_ID) REFERENCES Manager (Manager_ID)
7 );

```

Create Finance table

Results

Explain

Describe

Saved SQL

History

Object Type

TABLE ?

Object

FINANCE ?

| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
|---------|------------------------|-----------|--------|-----------|-------|-------------|----------|---------|---------|
| FINANCE | FINANCE_ID | NUMBER | 22 | - | 0 | 1 | - | - | - |
| | FINANCE_ACCOUNT_NUMBER | VARCHAR2 | 100 | - | - | - | ✓ | - | - |
| | FINANCE_BALANCE | NUMBER | - | 10 | 2 | - | ✓ | - | - |
| | MANAGER_ID | NUMBER | 22 | - | 0 | - | ✓ | - | - |

Finance table description

```

1 CREATE TABLE Team (
2     Team_ID INT PRIMARY KEY,
3     Team_Name VARCHAR(100),
4     Team_Icon VARCHAR(100),
5     Team_Established_Date DATE,
6     Team_Country VARCHAR(100),
7     Total_Price_Money DECIMAL(10, 2),
8     Manager_ID INT,
9     FOREIGN KEY (Manager_ID) REFERENCES Manager (Manager_ID)
10 );

```

Create Team table

| Results | Explain | Describe | Saved SQL | History | | | | | |
|-------------|-----------------------|-----------|-----------|-----------|--------|-------------|----------|---------|---------|
| Object Type | | TABLE ? | Object | | TEAM ? | | | | |
| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
| TEAM | TEAM_ID | NUMBER | 22 | - | 0 | 1 | - | - | - |
| | TEAM_NAME | VARCHAR2 | 100 | - | - | - | ✓ | - | - |
| | TEAM_ICON | VARCHAR2 | 100 | - | - | - | ✓ | - | - |
| | TEAM_ESTABLISHED_DATE | DATE | 7 | - | - | - | ✓ | - | - |
| | TEAM_COUNTRY | VARCHAR2 | 100 | - | - | - | ✓ | - | - |
| | TOTAL_PRICE_MONEY | NUMBER | - | 10 | 2 | - | ✓ | - | - |
| | MANAGER_ID | NUMBER | 22 | - | 0 | - | ✓ | - | - |

Team table description

```

1 CREATE TABLE Team_Winning (
2     Tw_ID INT PRIMARY KEY,
3     Team_ID INT,
4     Team_Winning VARCHAR(100),
5     FOREIGN KEY (Team_ID) REFERENCES Team (Team_ID)
6 );

```

Create Team Winning table

| Results | Explain | Describe | Saved SQL | | History | | | | |
|--------------|--------------|-----------|-----------|-----------|----------------|-------------|----------|---------|---------|
| Object Type | | TABLE ? | Object | | TEAM_WINNING ? | | | | |
| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
| TEAM_WINNING | TW_ID | NUMBER | 22 | - | 0 | 1 | - | - | - |
| | TEAM_ID | NUMBER | 22 | - | 0 | - | ✓ | - | - |
| | TEAM_WINNING | VARCHAR2 | 100 | - | - | - | ✓ | - | - |

Team Winning table description

```

1 CREATE TABLE SocialMedia (
2     SocialMedia_ID INT PRIMARY KEY,
3     SocialMedia_Name VARCHAR(100),
4     SocialMedia_Email VARCHAR(100),
5     SocialMedia_Password VARCHAR(100),
6     SocialMedia_Picture VARCHAR(100),
7     SocialMedia_Hiredate DATE,
8     SocialMedia_Salary DECIMAL(10, 2),
9     Manager_ID INT,
10    FOREIGN KEY (Manager_ID) REFERENCES Manager (Manager_ID)
11 );

```

Create SocialMedia table

| Results | Explain | Describe | Saved SQL | History | | | | | |
|-------------|----------------------|-----------|-----------|---------------|-------|-------------|----------|---------|---------|
| Object Type | | TABLE ? | Object | SOCIALMEDIA ? | | | | | |
| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
| SOCIALMEDIA | SOCIALMEDIA_ID | NUMBER | 22 | - | 0 | 1 | - | - | - |
| | SOCIALMEDIA_NAME | VARCHAR2 | 100 | - | - | - | ✓ | - | - |
| | SOCIALMEDIA_EMAIL | VARCHAR2 | 100 | - | - | - | ✓ | - | - |
| | SOCIALMEDIA_PASSWORD | VARCHAR2 | 100 | - | - | - | ✓ | - | - |
| | SOCIALMEDIA_PICTURE | VARCHAR2 | 100 | - | - | - | ✓ | - | - |
| | SOCIALMEDIA_HIREDATE | DATE | 7 | - | - | - | ✓ | - | - |
| | SOCIALMEDIA_SALARY | NUMBER | - | 10 | 2 | - | ✓ | - | - |
| | MANAGER_ID | NUMBER | 22 | - | 0 | - | ✓ | - | - |

SocialMedia table description

```

1 CREATE TABLE ContentCreator (
2     ContentCreator_ID INT PRIMARY KEY,
3     ContentCreator_Name VARCHAR(100),
4     ContentCreator_Email VARCHAR(100),
5     ContentCreator_Password VARCHAR(100),
6     ContentCreator_Picture VARCHAR(100),
7     ContentCreator_Hiredate DATE,
8     ContentCreator_Salary DECIMAL(10, 2),
9     SocialMedia_ID INT,
10    FOREIGN KEY (SocialMedia_ID) REFERENCES SocialMedia (SocialMedia_ID)
11 );

```

Create ContentCreator table

| Results | Explain | Describe | Saved SQL | History | | | | | |
|----------------|-------------------------|-----------|-----------|------------------|-------|-------------|----------|---------|---------|
| Object Type | | TABLE ? | Object | CONTENTCREATOR ? | | | | | |
| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
| CONTENTCREATOR | CONTENTCREATOR_ID | NUMBER | 22 | - | 0 | 1 | - | - | - |
| | CONTENTCREATOR_NAME | VARCHAR2 | 100 | - | - | - | ✓ | - | - |
| | CONTENTCREATOR_EMAIL | VARCHAR2 | 100 | - | - | - | ✓ | - | - |
| | CONTENTCREATOR_PASSWORD | VARCHAR2 | 100 | - | - | - | ✓ | - | - |
| | CONTENTCREATOR_PICTURE | VARCHAR2 | 100 | - | - | - | ✓ | - | - |
| | CONTENTCREATOR_HIREDATE | DATE | 7 | - | - | - | ✓ | - | - |
| | CONTENTCREATOR_SALARY | NUMBER | - | 10 | 2 | - | ✓ | - | - |
| | SOCIALMEDIA_ID | NUMBER | 22 | - | 0 | - | ✓ | - | - |

ContentCreator table description

```

1 CREATE TABLE ContentCreator_SocialMedia (
2     Ccs_ID INT PRIMARY KEY,
3     ContentCreator_ID INT,
4     ContentCreator_Facebook_Link VARCHAR(100),
5     ContentCreator_Twitter_Link VARCHAR(100),
6     ContentCreator_Instagram_Link VARCHAR(100),
7     ContentCreator_Youtube_Link VARCHAR(100),
8     FOREIGN KEY (ContentCreator_ID) REFERENCES ContentCreator (ContentCreator_ID)
9 );

```

Create ContentCreator SocialMedia table

| Results | Explain | Describe | Saved SQL History | | | | | | |
|----------------------------|-------------------------------|-----------|-----------------------------------|-----------|-------|-------------|----------|---------|---------|
| Object Type | | TABLE | Object CONTENTCREATOR_SOCIALMEDIA | | | | | | |
| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
| CONTENTCREATOR_SOCIALMEDIA | CCS_ID | NUMBER | 22 | - | 0 | 1 | - | - | - |
| | CONTENTCREATOR_ID | NUMBER | 22 | - | 0 | - | ✓ | - | - |
| | CONTENTCREATOR_FACEBOOK_LINK | VARCHAR2 | 100 | - | - | - | ✓ | - | - |
| | CONTENTCREATOR_TWITTER_LINK | VARCHAR2 | 100 | - | - | - | ✓ | - | - |
| | CONTENTCREATOR_INSTAGRAM_LINK | VARCHAR2 | 100 | - | - | - | ✓ | - | - |
| | CONTENTCREATOR_YOUTUBE_LINK | VARCHAR2 | 100 | - | - | - | ✓ | - | - |

ContentCreator SocialMedia table description

```

1 CREATE TABLE ContentCreator_Address (
2     Cca_ID INT PRIMARY KEY,
3     ContentCreator_ID INT,
4     ContentCreator_Country VARCHAR(100),
5     ContentCreator_City VARCHAR(100),
6     ContentCreator_Street VARCHAR(100),
7     ContentCreator_Zip_Code VARCHAR(20),
8     FOREIGN KEY (ContentCreator_ID) REFERENCES ContentCreator (ContentCreator_ID)
9 );

```

Create ContentCreator Address table

| Results | Explain | Describe | Saved SQL | History | | | | | |
|------------------------|-------------------------|--------------------------|-----------|-----------|-------|-------------|----------|---------|---------|
| Object Type | | TABLE ? | Object | | | | | | |
| | | CONTENTCREATOR_ADDRESS ? | | | | | | | |
| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
| CONTENTCREATOR_ADDRESS | CCA_ID | NUMBER | 22 | - | 0 | 1 | - | - | - |
| | CONTENTCREATOR_ID | NUMBER | 22 | - | 0 | - | ✓ | - | - |
| | CONTENTCREATOR_COUNTRY | VARCHAR2 | 100 | - | - | - | ✓ | - | - |
| | CONTENTCREATOR_CITY | VARCHAR2 | 100 | - | - | - | ✓ | - | - |
| | CONTENTCREATOR_STREET | VARCHAR2 | 100 | - | - | - | ✓ | - | - |
| | CONTENTCREATOR_ZIP_CODE | VARCHAR2 | 20 | - | - | - | ✓ | - | - |

ContentCreator Address table description

```

1  CREATE TABLE ContentCreator_Phone (
2      Ccp_ID INT PRIMARY KEY,
3      ContentCreator_ID INT,
4      ContentCreator_Phone VARCHAR(20),
5      FOREIGN KEY (ContentCreator_ID) REFERENCES ContentCreator (ContentCreator_ID)
6  );

```

Create ContentCreator Phone table


```

1 CREATE TABLE SocialMedia_Phone (
2     Smp_ID INT PRIMARY KEY,
3     SocialMedia_ID INT,
4     SocialMedia_Phone VARCHAR(20),
5     FOREIGN KEY (SocialMedia_ID) REFERENCES SocialMedia (SocialMedia_ID)
6 );

```

Create SocialMedia Phone table

Results

Explain

Describe

Saved SQL

History

Object Type

TABLE

Object

SOCIALMEDIA_PHONE

| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
|-------------------|-------------------|-----------|--------|-----------|-------|-------------|----------|---------|---------|
| SOCIALMEDIA_PHONE | SMP_ID | NUMBER | 22 | - | 0 | 1 | - | - | - |
| | SOCIALMEDIA_ID | NUMBER | 22 | - | 0 | - | ✓ | - | - |
| | SOCIALMEDIA_PHONE | VARCHAR2 | 20 | - | - | - | ✓ | - | - |

SocialMedia Phone table description

```

1 CREATE TABLE Organization (
2     Organization_ID INT PRIMARY KEY,
3     Organization_Name VARCHAR(100),
4     Organization_Email VARCHAR(100),
5     Organization_Password VARCHAR(100),
6     Organization_Picture VARCHAR(100),
7     Organization_Phone VARCHAR(20),
8     Finance_ID INT,
9     FOREIGN KEY (Finance_ID) REFERENCES Finance (Finance_ID)
10 );

```

Create Organization table

| Results | Explain | Describe | Saved SQL | History | | | | | |
|--------------|-----------------------|-----------|-----------|-----------|----------------|-------------|----------|---------|---------|
| Object Type | | TABLE ? | Object | | ORGANIZATION ? | | | | |
| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
| ORGANIZATION | ORGANIZATION_ID | NUMBER | 22 | - | 0 | 1 | - | - | - |
| | ORGANIZATION_NAME | VARCHAR2 | 100 | - | - | - | ✓ | - | - |
| | ORGANIZATION_EMAIL | VARCHAR2 | 100 | - | - | - | ✓ | - | - |
| | ORGANIZATION_PASSWORD | VARCHAR2 | 100 | - | - | - | ✓ | - | - |
| | ORGANIZATION_PICTURE | VARCHAR2 | 100 | - | - | - | ✓ | - | - |
| | ORGANIZATION_PHONE | VARCHAR2 | 20 | - | - | - | ✓ | - | - |
| | FINANCE_ID | NUMBER | 22 | - | 0 | - | ✓ | - | - |

Organization table description

```

1 CREATE TABLE Organization_Phone (
2     Op_ID INT PRIMARY KEY,
3     Organization_ID INT,
4     Organization_Phone VARCHAR(20),
5     FOREIGN KEY (Organization_ID) REFERENCES Organization (Organization_ID)
6 );

```

Create Organization Phone table

| Results | Explain | Describe | Saved SQL | History | | | | | |
|---|--------------------|-----------|-----------|-----------|-------|-------------|----------|---------|---------|
| Object Type TABLE ? Object ORGANIZATION_PHONE ? | | | | | | | | | |
| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
| ORGANIZATION_PHONE | OP_ID | NUMBER | 22 | - | 0 | 1 | - | - | - |
| | ORGANIZATION_ID | NUMBER | 22 | - | 0 | - | ✓ | - | - |
| | ORGANIZATION_PHONE | VARCHAR2 | 20 | - | - | - | ✓ | - | - |

Organization Phone table description

```

1 CREATE TABLE Player (
2     Player_ID INT PRIMARY KEY,
3     Player_Name VARCHAR(100),
4     Player_Email VARCHAR(100),
5     Player_Password VARCHAR(100),
6     Player_Picture VARCHAR(100),
7     Player_JoinDate DATE,
8     Player_Salary DECIMAL(10, 2),
9     Player_Play_Hours INT,
10    Player_DOB DATE
11 );

```

Create Player table

| Results | Explain | Describe | Saved SQL | History | | | | | | |
|-------------|-------------------|-----------|-----------|-----------|-------|-------------|----------|---------|---------|--|
| Object Type | | | TABLE | Object | | PLAYER | | | | |
| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment | |
| PLAYER | PLAYER_ID | NUMBER | 22 | - | 0 | 1 | - | - | - | |
| | PLAYER_NAME | VARCHAR2 | 100 | - | - | - | ✓ | - | - | |
| | PLAYER_EMAIL | VARCHAR2 | 100 | - | - | - | ✓ | - | - | |
| | PLAYER_PASSWORD | VARCHAR2 | 100 | - | - | - | ✓ | - | - | |
| | PLAYER_PICTURE | VARCHAR2 | 100 | - | - | - | ✓ | - | - | |
| | PLAYER_JOINDATE | DATE | 7 | - | - | - | ✓ | - | - | |
| | PLAYER_SALARY | NUMBER | - | 10 | 2 | - | ✓ | - | - | |
| | PLAYER_PLAY_HOURS | NUMBER | 22 | - | 0 | - | ✓ | - | - | |
| | PLAYER_DOB | DATE | 7 | - | - | - | ✓ | - | - | |

Player table description

```

1 CREATE TABLE Player_Address (
2     Pa_ID INT PRIMARY KEY,
3     Player_ID INT,
4     Player_Country VARCHAR(100),
5     Player_City VARCHAR(100),
6     Player_Street VARCHAR(100),
7     Player_Zip_Code VARCHAR(20),
8     FOREIGN KEY (Player_ID) REFERENCES Player (Player_ID)
9 );

```

Create Player Address table

| Results | Explain | Describe | Saved SQL | History | | | | | |
|----------------|-----------------|-----------|-----------|-----------|------------------|-------------|----------|---------|---------|
| Object Type | | TABLE ? | Object | | PLAYER_ADDRESS ? | | | | |
| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
| PLAYER_ADDRESS | PA_ID | NUMBER | 22 | - | 0 | 1 | - | - | - |
| | PLAYER_ID | NUMBER | 22 | - | 0 | - | ✓ | - | - |
| | PLAYER_COUNTRY | VARCHAR2 | 100 | - | - | - | ✓ | - | - |
| | PLAYER_CITY | VARCHAR2 | 100 | - | - | - | ✓ | - | - |
| | PLAYER_STREET | VARCHAR2 | 100 | - | - | - | ✓ | - | - |
| | PLAYER_ZIP_CODE | VARCHAR2 | 20 | - | - | - | ✓ | - | - |

Player Address table description

```

1 CREATE TABLE Player_Social_Link (
2     Psl_ID INT PRIMARY KEY,
3     Player_ID INT,
4     Player_Facebook_Link VARCHAR(100),
5     Player_Instagram_Link VARCHAR(100),
6     Player_Twitter_Link VARCHAR(100),
7     Player_Youtube_Link VARCHAR(100),
8     FOREIGN KEY (Player_ID) REFERENCES Player (Player_ID)
9 );

```

Create Player Social Link table

| Results | Explain | Describe | Saved SQL | History | | | | | |
|--------------------|-----------------------|-----------|-----------|-----------|----------------------|-------------|----------|---------|---------|
| Object Type | | TABLE ? | Object | | PLAYER_SOCIAL_LINK ? | | | | |
| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
| PLAYER_SOCIAL_LINK | PSL_ID | NUMBER | 22 | - | 0 | 1 | - | - | - |
| | PLAYER_ID | NUMBER | 22 | - | 0 | - | ✓ | - | - |
| | PLAYER_FACEBOOK_LINK | VARCHAR2 | 100 | - | - | - | ✓ | - | - |
| | PLAYER_INSTAGRAM_LINK | VARCHAR2 | 100 | - | - | - | ✓ | - | - |
| | PLAYER_TWITTER_LINK | VARCHAR2 | 100 | - | - | - | ✓ | - | - |
| | PLAYER_YOUTUBE_LINK | VARCHAR2 | 100 | - | - | - | ✓ | - | - |

Player Social Link table description

```

1 CREATE TABLE Player_Phone (
2     Pp_ID INT PRIMARY KEY,
3     Player_ID INT,
4     Player_Phone VARCHAR(20),
5     FOREIGN KEY (Player_ID) REFERENCES Player (Player_ID)
6 );

```

Create Player Phone table

| Results | Explain | Describe | Saved SQL | History | | | | | |
|--------------|--------------|-----------|----------------|-----------|-------|-------------|----------|---------|---------|
| Object Type | | TABLE ? | Object | | | | | | |
| | | | PLAYER_PHONE ? | | | | | | |
| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
| PLAYER_PHONE | PP_ID | NUMBER | 22 | - | 0 | 1 | - | - | - |
| | PLAYER_ID | NUMBER | 22 | - | 0 | - | ✓ | - | - |
| | PLAYER_PHONE | VARCHAR2 | 20 | - | - | - | ✓ | - | - |

Player Phone table description

```

1 CREATE TABLE Player_Winning (
2     Pw_ID INT PRIMARY KEY,
3     Player_ID INT,
4     Player_Winning VARCHAR(100),
5     FOREIGN KEY (Player_ID) REFERENCES Player (Player_ID)
6 );

```

Create Player Winning table

| Results | Explain | Describe | Saved SQL | History | | | | | |
|----------------|----------------|-----------|------------------|-----------|-------|-------------|----------|---------|---------|
| Object Type | | TABLE ? | Object | | | | | | |
| | | | PLAYER_WINNING ? | | | | | | |
| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
| PLAYER_WINNING | PW_ID | NUMBER | 22 | - | 0 | 1 | - | - | - |
| | PLAYER_ID | NUMBER | 22 | - | 0 | - | ✓ | - | - |
| | PLAYER_WINNING | VARCHAR2 | 100 | - | - | - | ✓ | - | - |

Player Winning table description

```

1 CREATE TABLE Player_Team (
2     Pt_ID INT PRIMARY KEY,
3     Player_ID INT,
4     Team_ID INT,
5     FOREIGN KEY (Player_ID) REFERENCES Player (Player_ID),
6     FOREIGN KEY (Team_ID) REFERENCES Team (Team_ID)
7 );

```

Create Player Team table

Results

Explain

Describe

Saved SQL

History

Object Type

TABLE

Object

PLAYER_TEAM

| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
|-------------|-----------|-----------|--------|-----------|-------|-------------|----------|---------|---------|
| PLAYER_TEAM | PT_ID | NUMBER | 22 | - | 0 | 1 | - | - | - |
| | PLAYER_ID | NUMBER | 22 | - | 0 | - | ✓ | - | - |
| | TEAM_ID | NUMBER | 22 | - | 0 | - | ✓ | - | - |

Player Team table description

```

1 CREATE TABLE Tournament (
2     Tournament_ID INT PRIMARY KEY,
3     Tournament_Name VARCHAR(100),
4     Tournament_StartingDate DATE,
5     Tournament_EndingDate DATE,
6     Tournament_Location VARCHAR(100),
7     Tournament_Prize_Pool DECIMAL(10, 2),
8     Organization_ID INT,
9     FOREIGN KEY (Organization_ID) REFERENCES Organization (Organization_ID)
10 );

```

Create Tournament table

| Results | Explain | Describe | Saved SQL | History | | | | | |
|-------------|-------------------------|-----------|-----------|--------------|-------|-------------|----------|---------|---------|
| Object Type | | TABLE ? | Object | TOURNAMENT ? | | | | | |
| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
| TOURNAMENT | TOURNAMENT_ID | NUMBER | 22 | - | 0 | 1 | - | - | - |
| | TOURNAMENT_NAME | VARCHAR2 | 100 | - | - | - | ✓ | - | - |
| | TOURNAMENT_STARTINGDATE | DATE | 7 | - | - | - | ✓ | - | - |
| | TOURNAMENT_ENDINGDATE | DATE | 7 | - | - | - | ✓ | - | - |
| | TOURNAMENT_LOCATION | VARCHAR2 | 100 | - | - | - | ✓ | - | - |
| | TOURNAMENT_PRIZE_POOL | NUMBER | - | 10 | 2 | - | ✓ | - | - |
| | ORGANIZATION_ID | NUMBER | 22 | - | 0 | - | ✓ | - | - |

Tournament table description

```

1 CREATE TABLE Record (
2     Record_ID INT PRIMARY KEY,
3     Record_Date DATE,
4     Record_Price_Pool DECIMAL(10, 2),
5     Tournament_ID INT,
6     FOREIGN KEY (Tournament_ID) REFERENCES Tournament (Tournament_ID)
7 );

```

Create Record table

| Results | Explain | Describe | Saved SQL | History | | | | | |
|-------------|-------------------|-----------|-----------|-----------|-------|-------------|----------|---------|---------|
| Object Type | | TABLE ? | Object | RECORD ? | | | | | |
| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
| RECORD | RECORD_ID | NUMBER | 22 | - | 0 | 1 | - | - | - |
| | RECORD_DATE | DATE | 7 | - | - | - | ✓ | - | - |
| | RECORD_PRICE_POOL | NUMBER | - | 10 | 2 | - | ✓ | - | - |
| | TOURNAMENT_ID | NUMBER | 22 | - | 0 | - | ✓ | - | - |

Record table description

```

1 CREATE TABLE Game (
2     Game_ID INT PRIMARY KEY,
3     Game_Name VARCHAR(100),
4     Game_Icon VARCHAR(100),
5     Game_ReleaseDate DATE,
6     Game_Platform VARCHAR(100),
7     Game_Price_Pool DECIMAL(10, 2),
8     Game_Genre VARCHAR(100),
9     Game_Publisher VARCHAR(100)
10 );

```

Create Game table

| Results | Explain | Describe | Saved SQL | History | | | | | |
|-------------|------------------|-----------|-----------|-----------|-------|-------------|----------|---------|---------|
| Object Type | | | TABLE ? | Object | | GAME ? | | | |
| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
| GAME | GAME_ID | NUMBER | 22 | - | 0 | 1 | - | - | - |
| | GAME_NAME | VARCHAR2 | 100 | - | - | - | ✓ | - | - |
| | GAME_ICON | VARCHAR2 | 100 | - | - | - | ✓ | - | - |
| | GAME_RELEASEDATE | DATE | 7 | - | - | - | ✓ | - | - |
| | GAME_PLATFORM | VARCHAR2 | 100 | - | - | - | ✓ | - | - |
| | GAME_PRICE_POOL | NUMBER | - | 10 | 2 | - | ✓ | - | - |
| | GAME_GENRE | VARCHAR2 | 100 | - | - | - | ✓ | - | - |
| | GAME_PUBLISHER | VARCHAR2 | 100 | - | - | - | ✓ | - | - |

Game table description

```

1 CREATE TABLE Tournament_Game (
2     Tournament_ID INT,
3     Game_ID INT,
4     PRIMARY KEY (Tournament_ID, Game_ID),
5     FOREIGN KEY (Tournament_ID) REFERENCES Tournament (Tournament_ID),
6     FOREIGN KEY (Game_ID) REFERENCES Game (Game_ID)
7 );

```

Create Tournament Game table

| Results | Explain | Describe | Saved SQL | History | | | | | |
|-----------------|---------------|-----------|-----------|-----------|-------------------|-------------|----------|---------|---------|
| Object Type | | TABLE ? | Object | | TOURNAMENT_GAME ? | | | | |
| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
| TOURNAMENT_GAME | TOURNAMENT_ID | NUMBER | 22 | - | 0 | 1 | - | - | - |
| | GAME_ID | NUMBER | 22 | - | 0 | 2 | - | - | - |

Tournament Game table description

```

1 CREATE TABLE Organization_Tournament (
2     Organization_ID INT,
3     Tournament_ID INT,
4     PRIMARY KEY (Organization_ID, Tournament_ID),
5     FOREIGN KEY (Organization_ID) REFERENCES Organization (Organization_ID),
6     FOREIGN KEY (Tournament_ID) REFERENCES Tournament (Tournament_ID)
7 );

```

Create Organization Tournament table

| Results | Explain | Describe | Saved SQL | History | | | | | |
|-------------------------|-----------------|-----------|----------------------------------|-----------|-------|-------------|----------|---------|---------|
| Object Type | | TABLE ? | Object ORGANIZATION_TOURNAMENT ? | | | | | | |
| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
| ORGANIZATION_TOURNAMENT | ORGANIZATION_ID | NUMBER | 22 | - | 0 | 1 | - | - | - |
| | TOURNAMENT_ID | NUMBER | 22 | - | 0 | 2 | - | - | - |

Organization Tournament table description

```

1 CREATE TABLE Team_Game (
2     Team_ID INT,
3     Game_ID INT,
4     PRIMARY KEY (Team_ID, Game_ID),
5     FOREIGN KEY (Team_ID) REFERENCES Team (Team_ID),
6     FOREIGN KEY (Game_ID) REFERENCES Game (Game_ID)
7 );

```

Create Team Game table

Results

Explain

Describe

Saved SQL

History

Object Type

TABLE

Object

TEAM_GAME

| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
|-----------|---------|-----------|--------|-----------|-------|-------------|----------|---------|---------|
| TEAM_GAME | TEAM_ID | NUMBER | 22 | - | 0 | 1 | - | - | - |
| | GAME_ID | NUMBER | 22 | - | 0 | 2 | - | - | - |

Team Game table description

```

1 CREATE TABLE Company (
2     Company_ID INT PRIMARY KEY,
3     Company_Name VARCHAR(100),
4     Company_Email VARCHAR(100),
5     Company_Picture VARCHAR(100),
6     Company_Phone VARCHAR(20),
7     Company_Location VARCHAR(100)
8 );

```

Create Company table

| Results | Explain | Describe | Saved SQL | History | | | | | |
|--|------------------|-----------|-----------|-----------|-------|-------------|----------|---------|---------|
| Object Type TABLE ? Object COMPANY ? | | | | | | | | | |
| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
| COMPANY | COMPANY_ID | NUMBER | 22 | - | 0 | 1 | - | - | - |
| | COMPANY_NAME | VARCHAR2 | 100 | - | - | - | ✓ | - | - |
| | COMPANY_EMAIL | VARCHAR2 | 100 | - | - | - | ✓ | - | - |
| | COMPANY_PICTURE | VARCHAR2 | 100 | - | - | - | ✓ | - | - |
| | COMPANY_PHONE | VARCHAR2 | 20 | - | - | - | ✓ | - | - |
| | COMPANY_LOCATION | VARCHAR2 | 100 | - | - | - | ✓ | - | - |

Company table description

```

1 CREATE TABLE Company_Phone (
2     Company_ID INT,
3     Company_Phone VARCHAR(20),
4     PRIMARY KEY (Company_ID, Company_Phone),
5     FOREIGN KEY (Company_ID) REFERENCES Company (Company_ID)
6 );

```

Create Company Phone table

Results

Explain

Describe

Saved SQL

History

Object Type

TABLE ?

Object

COMPANY_PHONE ?

| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
|---------------|---------------|-----------|--------|-----------|-------|-------------|----------|---------|---------|
| COMPANY_PHONE | COMPANY_ID | NUMBER | 22 | - | 0 | 1 | - | - | - |
| | COMPANY_PHONE | VARCHAR2 | 20 | - | - | 2 | - | - | - |

Company Phone table description

```

1 CREATE TABLE Organization_Company (
2   Organization_ID INT,
3   Company_ID INT,
4   PRIMARY KEY (Organization_ID, Company_ID),
5   FOREIGN KEY (Organization_ID) REFERENCES Organization (Organization_ID),
6   FOREIGN KEY (Company_ID) REFERENCES Company (Company_ID)
7 );

```

Create Organization Company table

| Results | Explain | Describe | Saved SQL | History | | | | | |
|---|-----------------|-----------|-----------|-----------|-------|-------------|----------|---------|---------|
| Object Type TABLE ? Object ORGANIZATION_COMPANY ? | | | | | | | | | |
| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
| ORGANIZATION_COMPANY | ORGANIZATION_ID | NUMBER | 22 | - | 0 | 1 | - | - | - |
| | COMPANY_ID | NUMBER | 22 | - | 0 | 2 | - | - | - |

Organization Company table description

```

1 CREATE TABLE Team_Company (
2   Team_ID INT,
3   Company_ID INT,
4   PRIMARY KEY (Team_ID, Company_ID),
5   FOREIGN KEY (Team_ID) REFERENCES Team (Team_ID),
6   FOREIGN KEY (Company_ID) REFERENCES Company (Company_ID)
7 );

```

Create Team Company table

| Results | Explain | Describe | Saved SQL | History | | | | | |
|--------------|------------|-----------|-----------|-----------|----------------|-------------|----------|---------|---------|
| Object Type | | TABLE ? | Object | | TEAM_COMPANY ? | | | | |
| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
| TEAM_COMPANY | TEAM_ID | NUMBER | 22 | - | 0 | 1 | - | - | - |
| | COMPANY_ID | NUMBER | 22 | - | 0 | 2 | - | - | - |

Team Company table description

5.3 Sequence Creation

```
1  -- Create sequence for Manager table
2  CREATE SEQUENCE seq_manager_id START WITH 1 INCREMENT BY 1;
3  -- Create sequence for Admin table
4  CREATE SEQUENCE seq_admin_id START WITH 1 INCREMENT BY 1;
5  -- Create sequence for Manager_Phone table
6  CREATE SEQUENCE seq_mp_id START WITH 1 INCREMENT BY 1;
7  -- Create sequence for Finance table
8  CREATE SEQUENCE seq_finance_id START WITH 1 INCREMENT BY 1;
9
10 -- Create sequence for Team table
11 CREATE SEQUENCE seq_team_id START WITH 1 INCREMENT BY 1;
12 -- Create sequence for Team_Winning table
13 CREATE SEQUENCE seq_tw_id START WITH 1 INCREMENT BY 1;
14 -- Create sequence for SocialMedia table
15 CREATE SEQUENCE seq_socialmedia_id START WITH 1 INCREMENT BY 1;
16 -- Create sequence for ContentCreator table
17 CREATE SEQUENCE seq_contentcreator_id START WITH 1 INCREMENT BY 1;
18 -- Create sequence for ContentCreator_SocialMedia table
19 CREATE SEQUENCE seq_ccs_id START WITH 1 INCREMENT BY 1;
20 -- Create sequence for ContentCreator_Address table
21 CREATE SEQUENCE seq_cca_id START WITH 1 INCREMENT BY 1;
22 -- Create sequence for ContentCreator_Phone table
23 CREATE SEQUENCE seq_ccp_id START WITH 1 INCREMENT BY 1;
24
25 -- Create sequence for SocialMedia_Phone table
26 CREATE SEQUENCE seq_smp_id START WITH 1 INCREMENT BY 1;
27 -- Create sequence for Organization table
28 CREATE SEQUENCE seq_organization_id START WITH 1 INCREMENT BY 1;
29 -- Create sequence for Organization_Phone table
30 CREATE SEQUENCE seq_op_id START WITH 1 INCREMENT BY 1;
31 -- Create sequence for Player table
32 CREATE SEQUENCE seq_player_id START WITH 1 INCREMENT BY 1;
33 -- Create sequence for Player_Address table
34 CREATE SEQUENCE seq_pa_id START WITH 1 INCREMENT BY 1;
35 -- Create sequence for Player_Social_Link table
36 CREATE SEQUENCE seq_psl_id START WITH 1 INCREMENT BY 1;
37
38 -- Create sequence for Player_Phone table
39 CREATE SEQUENCE seq_pp_id START WITH 1 INCREMENT BY 1;
40 -- Create sequence for Player_Winning table
41 CREATE SEQUENCE seq_pw_id START WITH 1 INCREMENT BY 1;
42 -- Create sequence for Record table
43 CREATE SEQUENCE seq_record_id START WITH 1 INCREMENT BY 1;
44 -- Create sequence for Tournament table
45 CREATE SEQUENCE seq_tournament_id START WITH 1 INCREMENT BY 1;
46 -- Create sequence for Game table
47 CREATE SEQUENCE seq_game_id START WITH 1 INCREMENT BY 1;
48 -- Create sequence for Company table
49 CREATE SEQUENCE seq_company_id START WITH 1 INCREMENT BY 1;
50 -- Create sequence for Company_Phone table
51 CREATE SEQUENCE seq_cp_id START WITH 1 INCREMENT BY 1;
52 -- Create sequence for Organization_Company table
53 CREATE SEQUENCE seq_oc_id START WITH 1 INCREMENT BY 1;
54 --
```

Sequence Creation

5.4 Index for Table

```
1  -- Create index for Manager table
2  CREATE INDEX idx_manager_email ON Manager (Manager_Email);
3  -- Create index for Admin table
4  CREATE INDEX idx_admin_email ON Admin (Admin_Email);
5  -- Create index for Manager_Phone table
6  CREATE INDEX idx_manager_phone_manager_id ON Manager_Phone (Manager_ID);
7  -- Create index for Finance table
8  CREATE INDEX idx_finance_manager_id ON Finance (Manager_ID);
9  -- Create index for Team table
10 CREATE INDEX idx_team_manager_id ON Team (Manager_ID);
11
12 -- Create index for Team_Winning table
13 CREATE INDEX idx_team_winning_team_id ON Team_Winning (Team_ID);
14 -- Create index for SocialMedia table
15 CREATE INDEX idx_socialmedia_name ON SocialMedia (SocialMedia_Name);
16 -- Create index for ContentCreator table
17 CREATE INDEX idx_contentcreator_name ON ContentCreator (ContentCreator_Name);
18 -- Create index for ContentCreator_SocialMedia table
19 CREATE INDEX idx_ccs_contentcreator_id ON ContentCreator_SocialMedia (ContentCreator_ID)
20 ;
21 -- Create index for ContentCreator_Address table
22 CREATE INDEX idx_cca_contentcreator_id ON ContentCreator_Address (ContentCreator_ID);
23 -- Create index for ContentCreator_Phone table
24 CREATE INDEX idx_ccp_contentcreator_id ON ContentCreator_Phone (ContentCreator_ID);
25 -- Create index for SocialMedia_Phone table
26 CREATE INDEX idx_smp_socialmedia_id ON SocialMedia_Phone (SocialMedia_ID);
27 -- Create index for Organization table
28 CREATE INDEX idx_organization_name ON Organization (Organization_Name);
29 -- Create index for Organization_Phone table
30 CREATE INDEX idx_op_organization_id ON Organization_Phone (Organization_ID);
31 -- Create index for Player table
32 CREATE INDEX idx_player_name ON Player (Player_Name);
33
34 -- Create index for Player_Address table
35 CREATE INDEX idx_pa_player_id ON Player_Address (Player_ID);
36 -- Create index for Player_Social_Link table
37 CREATE INDEX idx_psl_player_id ON Player_Social_Link (Player_ID);
38 -- Create index for Player_Phone table
39 CREATE INDEX idx_pp_player_id ON Player_Phone (Player_ID);
40 -- Create index for Player_Winning table
41 CREATE INDEX idx_pw_player_id ON Player_Winning (Player_ID);
42 -- Create index for Record table
43 CREATE INDEX idx_record_date ON Record (Record_Date);
44
45 -- Create index for Tournament table
46 CREATE INDEX idx_tournament_name ON Tournament (Tournament_Name);
47 -- Create index for Game table
48 CREATE INDEX idx_game_name ON Game (Game_Name);
49 -- Create index for Company table
50 CREATE INDEX idx_company_name ON Company (Company_Name);
51 -- Create index for Company_Phone table
52 CREATE INDEX idx_cp_company_id ON Company_Phone (Company_ID);
53 -- Create index for Organization_Company table
54 CREATE INDEX idx_oc_organization_id ON Organization_Company (Organization_ID);
55 --
```

Index for Table

5.5 Alter Table for effective indexing

```
1  -- Alter Manager table to add index
2  ALTER TABLE Manager ADD CONSTRAINT idx_manager_email UNIQUE (Manager_Email);
3  -- Alter Admin table to add index
4  ALTER TABLE Admin ADD CONSTRAINT idx_admin_email UNIQUE (Admin_Email);
5  -- Alter Manager_Phone table to add index
6  ALTER TABLE Manager_Phone ADD CONSTRAINT idx_manager_phone_manager_id UNIQUE (Manager_ID
7  );
8  -- Alter Finance table to add index
9  ALTER TABLE Finance ADD CONSTRAINT idx_finance_manager_id UNIQUE (Manager_ID);
10 -- Alter Team table to add index
11 ALTER TABLE Team ADD CONSTRAINT idx_team_manager_id UNIQUE (Manager_ID);
12 -- Alter Team_Winning table to add index
13 ALTER TABLE Team_Winning ADD CONSTRAINT idx_team_winning_team_id UNIQUE (Team_ID);
14
15 -- Alter SocialMedia table to add index
16 ALTER TABLE SocialMedia ADD CONSTRAINT idx_socialmedia_name UNIQUE (SocialMedia_Name);
17 -- Alter ContentCreator table to add index
18 ALTER TABLE ContentCreator ADD CONSTRAINT idx_contentcreator_name UNIQUE (
19   ContentCreator_Name);
20 -- Alter ContentCreator_SocialMedia table to add index
21 ALTER TABLE ContentCreator_SocialMedia ADD CONSTRAINT idx_ccs_contentcreator_id UNIQUE (
22   ContentCreator_ID);
23 -- Alter ContentCreator_Address table to add index
24 ALTER TABLE ContentCreator_Address ADD CONSTRAINT idx_cca_contentcreator_id UNIQUE (
25   ContentCreator_ID);
26 -- Alter ContentCreator_Phone table to add index
27 ALTER TABLE ContentCreator_Phone ADD CONSTRAINT idx_ccp_contentcreator_id UNIQUE (
28   ContentCreator_ID);
29 -- Alter SocialMedia_Phone table to add index
30 ALTER TABLE SocialMedia_Phone ADD CONSTRAINT idx_smp_socialmedia_id UNIQUE (
31   SocialMedia_ID);
32 -- Alter Organization table to add index
33 ALTER TABLE Organization ADD CONSTRAINT idx_organization_name UNIQUE (Organization_Name)
34 ;
35 -- Alter Organization_Phone table to add index
36 ALTER TABLE Organization_Phone ADD CONSTRAINT idx_op_organization_id UNIQUE (
37   Organization_ID);
38
39 -- Alter Player table to add index
40 ALTER TABLE Player ADD CONSTRAINT idx_player_name UNIQUE (Player_Name);
41 -- Alter Player_Address table to add index
42 ALTER TABLE Player_Address ADD CONSTRAINT idx_pa_player_id UNIQUE (Player_ID);
43 -- Alter Player_Social_Link table to add index
44 ALTER TABLE Player_Social_Link ADD CONSTRAINT idx_psl_player_id UNIQUE (Player_ID);
45 -- Alter Player_Phone table to add index
46 ALTER TABLE Player_Phone ADD CONSTRAINT idx_pp_player_id UNIQUE (Player_ID);
47 -- Alter Player_Winning table to add index
48 ALTER TABLE Player_Winning ADD CONSTRAINT idx_pw_player_id UNIQUE (Player_ID);
49 -- Alter Record table to add index
50 ALTER TABLE Record ADD CONSTRAINT idx_record_date UNIQUE (Record_Date);
51
52 -- Alter Tournament table to add index
53 ALTER TABLE Tournament ADD CONSTRAINT idx_tournament_name UNIQUE (Tournament_Name);
54 -- Alter Game table to add index
55 ALTER TABLE Game ADD CONSTRAINT idx_game_name UNIQUE (Game_Name);
56 -- Alter Company table to add index
57 ALTER TABLE Company ADD CONSTRAINT idx_company_name UNIQUE (Company_Name);
58 -- Alter Company_Phone table to add index
59 ALTER TABLE Company_Phone ADD CONSTRAINT idx_cp_company_id UNIQUE (Company_ID);
60 -- Alter Organization_Company table to add index
61 ALTER TABLE Organization_Company ADD CONSTRAINT idx_oc_organization_id UNIQUE (
62   Organization_ID);
63 --
```

Alter Table

5.6 Data Insertion

```
1 INSERT INTO Admin (Admin_ID, Admin_Name, Admin_Email, Admin_Password, Admin_Picture)
2 VALUES (seq_admin_id.NEXTVAL, 'Admin 1', 'admin1@example.com', 'adminpass1', 'admin1.jpg');
3
4 INSERT INTO Admin (Admin_ID, Admin_Name, Admin_Email, Admin_Password, Admin_Picture)
5 VALUES (seq_admin_id.NEXTVAL, 'Admin 2', 'admin2@example.com', 'adminpass2', 'admin2.jpg');
6
7 INSERT INTO Admin (Admin_ID, Admin_Name, Admin_Email, Admin_Password, Admin_Picture)
8 VALUES (seq_admin_id.NEXTVAL, 'Admin 3', 'admin3@example.com', 'adminpass3', 'admin3.jpg');
9
10 INSERT INTO Admin (Admin_ID, Admin_Name, Admin_Email, Admin_Password, Admin_Picture)
11 VALUES (seq_admin_id.NEXTVAL, 'Admin 4', 'admin4@example.com', 'adminpass4', 'admin4.jpg');
12
13 INSERT INTO Admin (Admin_ID, Admin_Name, Admin_Email, Admin_Password, Admin_Picture)
14 VALUES (seq_admin_id.NEXTVAL, 'Admin 5', 'admin5@example.com', 'adminpass5', 'admin5.jpg');
```

Inserting data into Admin table

| Results | Explain | Describe | Saved SQL | History |
|----------|------------|--------------------|----------------|---------------|
| ADMIN_ID | ADMIN_NAME | ADMIN_EMAIL | ADMIN_PASSWORD | ADMIN_PICTURE |
| 1 | Admin 1 | admin1@example.com | adminpass1 | admin1.jpg |
| 2 | Admin 2 | admin2@example.com | adminpass2 | admin2.jpg |
| 3 | Admin 3 | admin3@example.com | adminpass3 | admin3.jpg |
| 4 | Admin 4 | admin4@example.com | adminpass4 | admin4.jpg |
| 5 | Admin 5 | admin5@example.com | adminpass5 | admin5.jpg |

5 rows returned in 0.00 seconds

Inserted data of Admin table

```
1 INSERT INTO Manager (Manager_ID, Manager_Name, Manager_Email, Manager_Password,
2 Manager_Picture, Manager_Hiredate, Admin_ID)
3 VALUES (seq_manager_id.NEXTVAL, 'John Doe', 'john.doe@example.com', 'password123', 'profile.
4 jpg', TO_DATE('2022-01-01', 'YYYY-MM-DD'), 1);
5
6 INSERT INTO Manager (Manager_ID, Manager_Name, Manager_Email, Manager_Password,
7 Manager_Picture, Manager_Hiredate, Admin_ID)
8 VALUES (seq_manager_id.NEXTVAL, 'Jane Smith', 'jane.smith@example.com', 'password456', '
9 profile2.jpg', TO_DATE('2022-02-01', 'YYYY-MM-DD'), 2);
10
11 INSERT INTO Manager (Manager_ID, Manager_Name, Manager_Email, Manager_Password,
12 Manager_Picture, Manager_Hiredate, Admin_ID)
13 VALUES (seq_manager_id.NEXTVAL, 'Mike Johnson', 'mike.johnson@example.com', 'password789', '
14 profile3.jpg', TO_DATE('2022-03-01', 'YYYY-MM-DD'), 1);
15
16 INSERT INTO Manager (Manager_ID, Manager_Name, Manager_Email, Manager_Password,
17 Manager_Picture, Manager_Hiredate, Admin_ID)
18 VALUES (seq_manager_id.NEXTVAL, 'Sarah Williams', 'sarah.williams@example.com', 'password123
19 ', 'profile4.jpg', TO_DATE('2022-04-01', 'YYYY-MM-DD'), 2);
20
21 INSERT INTO Manager (Manager_ID, Manager_Name, Manager_Email, Manager_Password,
22 Manager_Picture, Manager_Hiredate, Admin_ID)
23 VALUES (seq_manager_id.NEXTVAL, 'Robert Davis', 'robert.davis@example.com', 'password456', '
24 profile5.jpg', TO_DATE('2022-05-01', 'YYYY-MM-DD'), 1);
```

Inserting data into manager tables

Results

Explain

Describe

Saved SQL

History

| MANAGER_ID | MANAGER_NAME | MANAGER_EMAIL | MANAGER_PASSWORD | MANAGER_PICTURE | MANAGER_HIREDATE | ADMIN_ID |
|------------|----------------|----------------------------|------------------|-----------------|------------------|----------|
| 1 | John Doe | john.doe@example.com | password123 | profile.jpg | 01-JAN-22 | 1 |
| 2 | Jane Smith | jane.smith@example.com | password456 | profile2.jpg | 01-FEB-22 | 2 |
| 3 | Mike Johnson | mike.johnson@example.com | password789 | profile3.jpg | 01-MAR-22 | 1 |
| 4 | Sarah Williams | sarah.williams@example.com | password123 | profile4.jpg | 01-APR-22 | 2 |
| 5 | Robert Davis | robert.davis@example.com | password456 | profile5.jpg | 01-MAY-22 | 1 |

5 rows returned in 0.00 seconds

Inserted data of manager table

```

1 INSERT INTO Manager_Phone (Mp_ID, Manager_ID, Manager_Phone)
2 VALUES (seq_mp_id.NEXTVAL, 1, '1234567890');
3
4 INSERT INTO Manager_Phone (Mp_ID, Manager_ID, Manager_Phone)
5 VALUES (seq_mp_id.NEXTVAL, 2, '0987654321');
6
7 INSERT INTO Manager_Phone (Mp_ID, Manager_ID, Manager_Phone)
8 VALUES (seq_mp_id.NEXTVAL, 3, '1112223333');
9
10 INSERT INTO Manager_Phone (Mp_ID, Manager_ID, Manager_Phone)
11 VALUES (seq_mp_id.NEXTVAL, 4, '4445556666');
12
13 INSERT INTO Manager_Phone (Mp_ID, Manager_ID, Manager_Phone)
14 VALUES (seq_mp_id.NEXTVAL, 5, '7778889999');

```

Inserting data into manager phone tables

Results

Explain

Describe

Saved SQL

History

| MP_ID | MANAGER_ID | MANAGER_PHONE |
|-------|------------|---------------|
| 1 | 1 | 1234567890 |
| 2 | 2 | 0987654321 |
| 3 | 3 | 1112223333 |
| 4 | 4 | 4445556666 |
| 5 | 5 | 7778889999 |

5 rows returned in 0.00 seconds

Inserted data of manager phone table

```

1 INSERT INTO Finance (Finance_ID, Finance_Account_Number, Finance_Balance, Manager_ID)
2 VALUES (seq_finance_id.NEXTVAL, 'ABC123456', 10000, 1);
3
4 INSERT INTO Finance (Finance_ID, Finance_Account_Number, Finance_Balance, Manager_ID)
5 VALUES (seq_finance_id.NEXTVAL, 'DEF789012', 20000, 2);
6
7 INSERT INTO Finance (Finance_ID, Finance_Account_Number, Finance_Balance, Manager_ID)
8 VALUES (seq_finance_id.NEXTVAL, 'GHI345678', 15000, 3);
9
10 INSERT INTO Finance (Finance_ID, Finance_Account_Number, Finance_Balance, Manager_ID)
11 VALUES (seq_finance_id.NEXTVAL, 'JKL901234', 18000, 4);
12
13 INSERT INTO Finance (Finance_ID, Finance_Account_Number, Finance_Balance, Manager_ID)
14 VALUES (seq_finance_id.NEXTVAL, 'MNO567890', 22000, 5);

```

Inserting data into Finance tables

| Results | Explain | Describe | Saved SQL | History |
|------------|------------------------|----------|-----------------|------------|
| FINANCE_ID | FINANCE_ACCOUNT_NUMBER | | FINANCE_BALANCE | MANAGER_ID |
| 1 | ABC123456 | | 10000 | 1 |
| 2 | DEF789012 | | 20000 | 2 |
| 3 | GHI345678 | | 15000 | 3 |
| 4 | JKL901234 | | 18000 | 4 |
| 5 | MNO567890 | | 22000 | 5 |

5 rows returned in 0.00 seconds

Inserted data of Finance table

```

1
2 INSERT INTO Team (Team_ID, Team_Name, Team_Icon, Team_Established_Date, Team_Country,
3   Total_Price_Money, Manager_ID)
4 VALUES (seq_team_id.NEXTVAL, 'Team A', 'teamA.png', TO_DATE('2022-01-01', 'YYYY-MM-DD'), '
5   USA', 50000, 1);
6
7 INSERT INTO Team (Team_ID, Team_Name, Team_Icon, Team_Established_Date, Team_Country,
8   Total_Price_Money, Manager_ID)
9 VALUES (seq_team_id.NEXTVAL, 'Team B', 'teamB.png', TO_DATE('2022-02-01', 'YYYY-MM-DD'), 'UK
10  ', 60000, 2);
11
12 INSERT INTO Team (Team_ID, Team_Name, Team_Icon, Team_Established_Date, Team_Country,
13   Total_Price_Money, Manager_ID)
14 VALUES (seq_team_id.NEXTVAL, 'Team C', 'teamC.png', TO_DATE('2022-03-01', 'YYYY-MM-DD'), '
15   Australia', 45000, 3);
16
17 INSERT INTO Team (Team_ID, Team_Name, Team_Icon, Team_Established_Date, Team_Country,
18   Total_Price_Money, Manager_ID)
19 VALUES (seq_team_id.NEXTVAL, 'Team D', 'teamD.png', TO_DATE('2022-04-01', 'YYYY-MM-DD'), '
20   Canada', 55000, 4);
21
22 INSERT INTO Team (Team_ID, Team_Name, Team_Icon, Team_Established_Date, Team_Country,
23   Total_Price_Money, Manager_ID)
24 VALUES (seq_team_id.NEXTVAL, 'Team E', 'teamE.png', TO_DATE('2022-05-01', 'YYYY-MM-DD'), '
25   Germany', 70000, 5);

```

Inserting data into Team tables

| Results | Explain | Describe | Saved SQL | History | | |
|---------|-----------|-----------|-----------------------|--------------|-------------------|------------|
| TEAM_ID | TEAM_NAME | TEAM_ICON | TEAM_ESTABLISHED_DATE | TEAM_COUNTRY | TOTAL_PRICE_MONEY | MANAGER_ID |
| 1 | Team A | teamA.png | 01-JAN-22 | USA | 50000 | 1 |
| 2 | Team B | teamB.png | 01-FEB-22 | UK | 60000 | 2 |
| 3 | Team C | teamC.png | 01-MAR-22 | Australia | 45000 | 3 |
| 4 | Team D | teamD.png | 01-APR-22 | Canada | 55000 | 4 |
| 5 | Team E | teamE.png | 01-MAY-22 | Germany | 70000 | 5 |

5 rows returned in 0.00 seconds

Inserted data of Team table

```

1
2 INSERT INTO Team_Winning (Tw_ID, Team_ID, Team_Winning)
3 VALUES (seq_tw_id.NEXTVAL, 1, 'Championship 2022');
4
5 INSERT INTO Team_Winning (Tw_ID, Team_ID, Team_Winning)
6 VALUES (seq_tw_id.NEXTVAL, 2, 'Tournament 2023');
7
8 INSERT INTO Team_Winning (Tw_ID, Team_ID, Team_Winning)
9 VALUES (seq_tw_id.NEXTVAL, 3, 'Cup 2022');
10
11 INSERT INTO Team_Winning (Tw_ID, Team_ID, Team_Winning)
12 VALUES (seq_tw_id.NEXTVAL, 4, 'League 2022');
13
14 INSERT INTO Team_Winning (Tw_ID, Team_ID, Team_Winning)
15 VALUES (seq_tw_id.NEXTVAL, 5, 'Championship 2023');

```

Inserting data into Team_Winning tables

| Results | Explain | Describe | Saved SQL | History |
|---------|---------|-------------------|-----------|---------|
| TW_ID | TEAM_ID | TEAM_WINNING | | |
| 1 | 1 | Championship 2022 | | |
| 2 | 2 | Tournament 2023 | | |
| 3 | 3 | Cup 2022 | | |
| 4 | 4 | League 2022 | | |
| 5 | 5 | Championship 2023 | | |

5 rows returned in 0.00 seconds

Inserted data of Team Winning table

```

1
2 INSERT INTO SocialMedia (SocialMedia_ID, SocialMedia_Name, SocialMedia_Email,
   SocialMedia_Password, SocialMedia_Picture, SocialMedia_Hiredate, SocialMedia_Salary,
   MANAGER_ID)
3 VALUES (seq_socialmedia_id.NEXTVAL, 'SocialMediaUser221', 'social.user1@example.com', '
   socialpass1', 'social1.jpg', TO_DATE('2022-01-01', 'YYYY-MM-DD'), 5000, 1);
4
5 INSERT INTO SocialMedia (SocialMedia_ID, SocialMedia_Name, SocialMedia_Email,
   SocialMedia_Password, SocialMedia_Picture, SocialMedia_Hiredate, SocialMedia_Salary,
   MANAGER_ID)
6 VALUES (seq_socialmedia_id.NEXTVAL, 'SocialMediaUser442', 'social.user2@example.com', '
   socialpass2', 'social2.jpg', TO_DATE('2022-02-01', 'YYYY-MM-DD'), 6000, 2);
7
8 INSERT INTO SocialMedia (SocialMedia_ID, SocialMedia_Name, SocialMedia_Email,
   SocialMedia_Password, SocialMedia_Picture, SocialMedia_Hiredate, SocialMedia_Salary,
   MANAGER_ID)
9 VALUES (seq_socialmedia_id.NEXTVAL, 'SocialMediaUser423', 'social.user3@example.com', '
   socialpass3', 'social3.jpg', TO_DATE('2022-03-01', 'YYYY-MM-DD'), 7000, 3);
10
11 INSERT INTO SocialMedia (SocialMedia_ID, SocialMedia_Name, SocialMedia_Email,
   SocialMedia_Password, SocialMedia_Picture, SocialMedia_Hiredate, SocialMedia_Salary,
   MANAGER_ID)
12 VALUES (seq_socialmedia_id.NEXTVAL, 'SocialMediaUser4234', 'social.user4@example.com', '
   socialpass4', 'social4.jpg', TO_DATE('2022-04-01', 'YYYY-MM-DD'), 8000, 4);
13
14 INSERT INTO SocialMedia (SocialMedia_ID, SocialMedia_Name, SocialMedia_Email,
   SocialMedia_Password, SocialMedia_Picture, SocialMedia_Hiredate, SocialMedia_Salary,
   MANAGER_ID)
15 VALUES (seq_socialmedia_id.NEXTVAL, 'SocialMediaUser5523', 'social.user5@example.com', '
   socialpass5', 'social5.jpg', TO_DATE('2022-05-01', 'YYYY-MM-DD'), 9000, 5);

```

Inserting data into SocialMedia tables

Results

Explain

Describe

Saved SQL

History

| SOCIALMEDIA_ID | SOCIALMEDIA_NAME | SOCIALMEDIA_EMAIL | SOCIALMEDIA_PASSWORD | SOCIALMEDIA_PICTURE | SOCIALMEDIA_HIREDATE | SOCIALMEDIA_SALARY | MANAGER_ID |
|----------------|---------------------|--------------------------|----------------------|---------------------|----------------------|--------------------|------------|
| 1 | SocialMediaUser221 | social.user1@example.com | socialpass1 | social1.jpg | 01-JAN-22 | 5000 | 1 |
| 2 | SocialMediaUser442 | social.user2@example.com | socialpass2 | social2.jpg | 01-FEB-22 | 6000 | 2 |
| 3 | SocialMediaUser423 | social.user3@example.com | socialpass3 | social3.jpg | 01-MAR-22 | 7000 | 3 |
| 4 | SocialMediaUser4234 | social.user4@example.com | socialpass4 | social4.jpg | 01-APR-22 | 8000 | 4 |
| 5 | SocialMediaUser5523 | social.user5@example.com | socialpass5 | social5.jpg | 01-MAY-22 | 9000 | 5 |

5 rows returned in 0.00 seconds

Inserted data of Social Media table

```

1 INSERT INTO ContentCreator (ContentCreator_ID, ContentCreator_Name, ContentCreator_Email,
  ContentCreator_Password, ContentCreator_Picture, ContentCreator_Hiredate,
  ContentCreator_Salary, SOCIALMEDIA_ID)
2 VALUES (seq_contentcreator_id.NEXTVAL, 'ContentCreator 1', 'cc1@example.com', 'ccpass1', '
  cc1.jpg', TO_DATE('2022-01-01', 'YYYY-MM-DD'), 3000, 1)
3
4 INSERT INTO ContentCreator (ContentCreator_ID, ContentCreator_Name, ContentCreator_Email,
  ContentCreator_Password, ContentCreator_Picture, ContentCreator_Hiredate,
  ContentCreator_Salary, SOCIALMEDIA_ID)
5 VALUES (seq_contentcreator_id.NEXTVAL, 'ContentCreator 2', 'cc2@example.com', 'ccpass2', '
  cc2.jpg', TO_DATE('2022-02-01', 'YYYY-MM-DD'), 3500, 2)
6
7 INSERT INTO ContentCreator (ContentCreator_ID, ContentCreator_Name, ContentCreator_Email,
  ContentCreator_Password, ContentCreator_Picture, ContentCreator_Hiredate,
  ContentCreator_Salary, SOCIALMEDIA_ID)
8 VALUES (seq_contentcreator_id.NEXTVAL, 'ContentCreator 3', 'cc3@example.com', 'ccpass3', '
  cc3.jpg', TO_DATE('2022-03-01', 'YYYY-MM-DD'), 4000, 3)
9
10 INSERT INTO ContentCreator (ContentCreator_ID, ContentCreator_Name, ContentCreator_Email,
  ContentCreator_Password, ContentCreator_Picture, ContentCreator_Hiredate,
  ContentCreator_Salary, SOCIALMEDIA_ID)
11 VALUES (seq_contentcreator_id.NEXTVAL, 'ContentCreator 4', 'cc4@example.com', 'ccpass4', '
  cc4.jpg', TO_DATE('2022-04-01', 'YYYY-MM-DD'), 4500, 4)
12
13 INSERT INTO ContentCreator (ContentCreator_ID, ContentCreator_Name, ContentCreator_Email,
  ContentCreator_Password, ContentCreator_Picture, ContentCreator_Hiredate,
  ContentCreator_Salary, SOCIALMEDIA_ID)
14 VALUES (seq_contentcreator_id.NEXTVAL, 'ContentCreator 5', 'cc5@example.com', 'ccpass5', '
  cc5.jpg', TO_DATE('2022-05-01', 'YYYY-MM-DD'), 5000, 5)

```

Inserting data into ContentCreator tables

Results

Explain

Describe

Saved SQL

History

| CONTENTCREATOR_ID | CONTENTCREATOR_NAME | CONTENTCREATOR_EMAIL | CONTENTCREATOR_PASSWORD | CONTENTCREATOR_PICTURE | CONTENTCREATOR_HIREDATE | CONTENTCREATOR_SALARY | SOCIALMEDIA_ID |
|-------------------|---------------------|----------------------|-------------------------|------------------------|-------------------------|-----------------------|----------------|
| 1 | ContentCreator 1 | cc1@example.com | ccpass1 | cc1.jpg | 01-JAN-22 | 3000 | 1 |
| 2 | ContentCreator 2 | cc2@example.com | ccpass2 | cc2.jpg | 01-FEB-22 | 3500 | 2 |
| 3 | ContentCreator 3 | cc3@example.com | ccpass3 | cc3.jpg | 01-MAR-22 | 4000 | 3 |
| 4 | ContentCreator 4 | cc4@example.com | ccpass4 | cc4.jpg | 01-APR-22 | 4500 | 4 |
| 5 | ContentCreator 5 | cc5@example.com | ccpass5 | cc5.jpg | 01-MAY-22 | 5000 | 5 |

5 rows returned in 0.00 seconds

Inserted data of Content Creator table

```

1 INSERT INTO ContentCreator_SocialMedia (Ccs_ID, ContentCreator_ID,
  ContentCreator_Facebook_Link, ContentCreator_Twitter_Link, ContentCreator_Instagram_Link
  , ContentCreator_Youtube_Link)
2 VALUES (seq_ccs_id.NEXTVAL, 1, 'https://facebook.com/cc1', 'https://twitter.com/cc1', 'https
  ://instagram.com/cc1', 'https://youtube.com/cc1');
3
4 INSERT INTO ContentCreator_SocialMedia (Ccs_ID, ContentCreator_ID,
  ContentCreator_Facebook_Link, ContentCreator_Twitter_Link, ContentCreator_Instagram_Link
  , ContentCreator_Youtube_Link)
5 VALUES (seq_ccs_id.NEXTVAL, 2, 'https://facebook.com/cc2', 'https://twitter.com/cc2', 'https
  ://instagram.com/cc2', 'https://youtube.com/cc2');
6
7 INSERT INTO ContentCreator_SocialMedia (Ccs_ID, ContentCreator_ID,
  ContentCreator_Facebook_Link, ContentCreator_Twitter_Link, ContentCreator_Instagram_Link
  , ContentCreator_Youtube_Link)
8 VALUES (seq_ccs_id.NEXTVAL, 3, 'https://facebook.com/cc3', 'https://twitter.com/cc3', 'https
  ://instagram.com/cc3', 'https://youtube.com/cc3');
9
10 INSERT INTO ContentCreator_SocialMedia (Ccs_ID, ContentCreator_ID,
  ContentCreator_Facebook_Link, ContentCreator_Twitter_Link, ContentCreator_Instagram_Link
  , ContentCreator_Youtube_Link)
11 VALUES (seq_ccs_id.NEXTVAL, 4, 'https://facebook.com/cc4', 'https://twitter.com/cc4', 'https
  ://instagram.com/cc4', 'https://youtube.com/cc4');
12
13 INSERT INTO ContentCreator_SocialMedia (Ccs_ID, ContentCreator_ID,
  ContentCreator_Facebook_Link, ContentCreator_Twitter_Link, ContentCreator_Instagram_Link
  , ContentCreator_Youtube_Link)
14 VALUES (seq_ccs_id.NEXTVAL, 5, 'https://facebook.com/cc5', 'https://twitter.com/cc5', 'https
  ://instagram.com/cc5', 'https://youtube.com/cc5');

```

Inserting data into ContentCreator_SocialMedia tables

Results

ExplainDescribeSaved SQLHistory

| CCS_ID | CONTENTCREATOR_ID | CONTENTCREATOR_FACEBOOK_LINK | CONTENTCREATOR_TWITTER_LINK | CONTENTCREATOR_INSTAGRAM_LINK | CONTENTCREATOR_YOUTUBE_LINK |
|--------|-------------------|------------------------------|-----------------------------|-------------------------------|-----------------------------|
| 1 | 1 | https://facebook.com/cc1 | https://twitter.com/cc1 | https://instagram.com/cc1 | https://youtube.com/cc1 |
| 2 | 2 | https://facebook.com/cc2 | https://twitter.com/cc2 | https://instagram.com/cc2 | https://youtube.com/cc2 |
| 3 | 3 | https://facebook.com/cc3 | https://twitter.com/cc3 | https://instagram.com/cc3 | https://youtube.com/cc3 |
| 4 | 4 | https://facebook.com/cc4 | https://twitter.com/cc4 | https://instagram.com/cc4 | https://youtube.com/cc4 |
| 5 | 5 | https://facebook.com/cc5 | https://twitter.com/cc5 | https://instagram.com/cc5 | https://youtube.com/cc5 |

5 rows returned in 0.00 seconds

Inserted data of Content Creator Social Media table

```

1 INSERT INTO ContentCreator_Address (Cca_ID, ContentCreator_ID, ContentCreator_Country,
2   ContentCreator_City, ContentCreator_Street, ContentCreator_Zip_Code)
3 VALUES (seq_cca_id.NEXTVAL, 1, 'USA', 'New York', '123 Street', '10001');
4 INSERT INTO ContentCreator_Address (Cca_ID, ContentCreator_ID, ContentCreator_Country,
5   ContentCreator_City, ContentCreator_Street, ContentCreator_Zip_Code)
6 VALUES (seq_cca_id.NEXTVAL, 2, 'USA', 'Los Angeles', '456 Avenue', '90001');
7 INSERT INTO ContentCreator_Address (Cca_ID, ContentCreator_ID, ContentCreator_Country,
8   ContentCreator_City, ContentCreator_Street, ContentCreator_Zip_Code)
9 VALUES (seq_cca_id.NEXTVAL, 3, 'UK', 'London', '789 Road', 'SW1A 1AA');
10 INSERT INTO ContentCreator_Address (Cca_ID, ContentCreator_ID, ContentCreator_Country,
11   ContentCreator_City, ContentCreator_Street, ContentCreator_Zip_Code)
12 VALUES (seq_cca_id.NEXTVAL, 4, 'Canada', 'Toronto', '321 Boulevard', 'M5V 2T3');
13 INSERT INTO ContentCreator_Address (Cca_ID, ContentCreator_ID, ContentCreator_Country,
14   ContentCreator_City, ContentCreator_Street, ContentCreator_Zip_Code)
15 VALUES (seq_cca_id.NEXTVAL, 5, 'Germany', 'Berlin', '987 Strasse', '12345');

```

Inserting data into ContentCreator_Address tables

Results

Explain

Describe

Saved SQL

History

| CCA_ID | CONTENTCREATOR_ID | CONTENTCREATOR_COUNTRY | CONTENTCREATOR_CITY | CONTENTCREATOR_STREET | CONTENTCREATOR_ZIP_CODE |
|--------|-------------------|------------------------|---------------------|-----------------------|-------------------------|
| 1 | 1 | USA | New York | 123 Street | 10001 |
| 2 | 2 | USA | Los Angeles | 456 Avenue | 90001 |
| 3 | 3 | UK | London | 789 Road | SW1A 1AA |
| 4 | 4 | Canada | Toronto | 321 Boulevard | M5V 2T3 |
| 5 | 5 | Germany | Berlin | 987 Strasse | 12345 |

5 rows returned in 0.00 seconds

Inserted data of Content Creator Address table

```

1 INSERT INTO ContentCreator_Phone (Ccp_ID, ContentCreator_ID, ContentCreator_Phone)
2 VALUES (seq_ccp_id.NEXTVAL, 1, '9876543210');
3
4 INSERT INTO ContentCreator_Phone (Ccp_ID, ContentCreator_ID, ContentCreator_Phone)
5 VALUES (seq_ccp_id.NEXTVAL, 2, '1234567890');
6
7 INSERT INTO ContentCreator_Phone (Ccp_ID, ContentCreator_ID, ContentCreator_Phone)
8 VALUES (seq_ccp_id.NEXTVAL, 3, '5551234567');
9
10 INSERT INTO ContentCreator_Phone (Ccp_ID, ContentCreator_ID, ContentCreator_Phone)
11 VALUES (seq_ccp_id.NEXTVAL, 4, '7775558888');
12
13 INSERT INTO ContentCreator_Phone (Ccp_ID, ContentCreator_ID, ContentCreator_Phone)
14 VALUES (seq_ccp_id.NEXTVAL, 5, '9990001111');

```

Inserting data into ContentCreator_Phone tables

| Results | Explain | Describe | Saved SQL | History |
|---------|-------------------|----------|----------------------|---------|
| CCP_ID | CONTENTCREATOR_ID | | CONTENTCREATOR_PHONE | |
| 1 | 1 | | 9876543210 | |
| 2 | 2 | | 1234567890 | |
| 3 | 3 | | 5551234567 | |
| 4 | 4 | | 7775558888 | |
| 5 | 5 | | 9990001111 | |

5 rows returned in 0.00 seconds

Inserted data of Content Creator Phone table

```

1 INSERT INTO SocialMedia_Phone (Smp_ID, SocialMedia_ID, SocialMedia_Phone)
2 VALUES (seq_smp_id.NEXTVAL, 1, '5551234567');
3
4 INSERT INTO SocialMedia_Phone (Smp_ID, SocialMedia_ID, SocialMedia_Phone)
5 VALUES (seq_smp_id.NEXTVAL, 2, '6669876543');
6
7 INSERT INTO SocialMedia_Phone (Smp_ID, SocialMedia_ID, SocialMedia_Phone)
8 VALUES (seq_smp_id.NEXTVAL, 3, '7774561230');
9
10 INSERT INTO SocialMedia_Phone (Smp_ID, SocialMedia_ID, SocialMedia_Phone)
11 VALUES (seq_smp_id.NEXTVAL, 4, '8887890123');
12
13 INSERT INTO SocialMedia_Phone (Smp_ID, SocialMedia_ID, SocialMedia_Phone)
14 VALUES (seq_smp_id.NEXTVAL, 5, '9996547890');

```

Inserting data into SocialMedia_Phone tables

| Results | Explain | Describe | Saved SQL | History |
|---------|----------------|----------|-------------------|---------|
| SMP_ID | SOCIALMEDIA_ID | | SOCIALMEDIA_PHONE | |
| 1 | 1 | | 5551234567 | |
| 2 | 2 | | 6669876543 | |
| 3 | 3 | | 7774561230 | |
| 4 | 4 | | 8887890123 | |
| 5 | 5 | | 9996547890 | |

5 rows returned in 0.00 seconds

Inserted data of Social Media Phone table

```

1 INSERT INTO Organization (Organization_ID, Organization_Name, Organization_Email,
2   Organization_Password, Organization_Picture, Organization_Phone, Finance_ID)
3 VALUES (seq_organization_id.NEXTVAL, 'Organization XYZ', 'info@xyz.com', 'orgpass', 'org.jpg',
4   '1234567890', 1);
5
6 INSERT INTO Organization (Organization_ID, Organization_Name, Organization_Email,
7   Organization_Password, Organization_Picture, Organization_Phone, Finance_ID)
8 VALUES (seq_organization_id.NEXTVAL, 'Organization ABC', 'info@abc.com', 'orgpass', 'org2.
9   jpg', '0987654321', 2);
10
11 INSERT INTO Organization (Organization_ID, Organization_Name, Organization_Email,
12   Organization_Password, Organization_Picture, Organization_Phone, Finance_ID)
13 VALUES (seq_organization_id.NEXTVAL, 'Organization DEF', 'info@def.com', 'orgpass', 'org3.
14   jpg', '1112223333', 3);
15
16 INSERT INTO Organization (Organization_ID, Organization_Name, Organization_Email,
17   Organization_Password, Organization_Picture, Organization_Phone, Finance_ID)
18 VALUES (seq_organization_id.NEXTVAL, 'Organization GHI', 'info@ghi.com', 'orgpass', 'org4.
19   jpg', '4445556666', 4);
20
21 INSERT INTO Organization (Organization_ID, Organization_Name, Organization_Email,
22   Organization_Password, Organization_Picture, Organization_Phone, Finance_ID)
23 VALUES (seq_organization_id.NEXTVAL, 'Organization JKL', 'info@jkl.com', 'orgpass', 'org5.
24   jpg', '7778889999', 5);

```

Inserting data into Organization tables

Results

Explain

Describe

Saved SQL

History

| ORGANIZATION_ID | ORGANIZATION_NAME | ORGANIZATION_EMAIL | ORGANIZATION_PASSWORD | ORGANIZATION_PICTURE | ORGANIZATION_PHONE | FINANCE_ID |
|-----------------|-------------------|--------------------|-----------------------|----------------------|--------------------|------------|
| 1 | Organization XYZ | info@xyz.com | orgpass | org.jpg | 1234567890 | 1 |
| 2 | Organization ABC | info@abc.com | orgpass | org2.jpg | 0987654321 | 2 |
| 3 | Organization DEF | info@def.com | orgpass | org3.jpg | 1112223333 | 3 |
| 4 | Organization GHI | info@ghi.com | orgpass | org4.jpg | 4445556666 | 4 |
| 5 | Organization JKL | info@jkl.com | orgpass | org5.jpg | 7778889999 | 5 |

5 rows returned in 0.00 seconds

5 rows returned in 0.00 seconds

Inserted data of Organization table

```

1 INSERT INTO Organization_Phone (Op_ID, Organization_ID, Organization_Phone)
2 VALUES (seq_op_id.NEXTVAL, 1, '9998887777');
3
4 INSERT INTO Organization_Phone (Op_ID, Organization_ID, Organization_Phone)
5 VALUES (seq_op_id.NEXTVAL, 2, '8887776666');
6
7 INSERT INTO Organization_Phone (Op_ID, Organization_ID, Organization_Phone)
8 VALUES (seq_op_id.NEXTVAL, 3, '7776665555');
9
10 INSERT INTO Organization_Phone (Op_ID, Organization_ID, Organization_Phone)
11 VALUES (seq_op_id.NEXTVAL, 4, '6665554444');
12
13 INSERT INTO Organization_Phone (Op_ID, Organization_ID, Organization_Phone)
14 VALUES (seq_op_id.NEXTVAL, 5, '5554443333');
```

Inserting data into Organization_Phone tables

Results

Explain

Describe

Saved SQL

History

| OP_ID | ORGANIZATION_ID | ORGANIZATION_PHONE |
|-------|-----------------|--------------------|
| 1 | 1 | 9998887777 |
| 2 | 2 | 8887776666 |
| 3 | 3 | 7776665555 |
| 4 | 4 | 6665554444 |
| 5 | 5 | 5554443333 |

5 rows returned in 0.00 seconds

Inserted data of Organization Phone table

```

1 INSERT INTO Player (Player_ID, Player_Name, Player_Email, Player_Password, Player_Picture,
2   Player_JoinDate, Player_Salary, Player_Play_Hours, Player_DOB)
3 VALUES (seq_player_id.NEXTVAL, 'Player 1', 'player1@example.com', 'playerpass', 'player1.jpg',
4   TO_DATE('2022-01-01', 'YYYY-MM-DD'), 5000, 100, TO_DATE('1990-01-01', 'YYYY-MM-DD'));
5
6 INSERT INTO Player (Player_ID, Player_Name, Player_Email, Player_Password, Player_Picture,
7   Player_JoinDate, Player_Salary, Player_Play_Hours, Player_DOB)
8 VALUES (seq_player_id.NEXTVAL, 'Player 2', 'player2@example.com', 'playerpass', 'player2.jpg',
9   TO_DATE('2022-02-01', 'YYYY-MM-DD'), 6000, 200, TO_DATE('1992-05-10', 'YYYY-MM-DD'));
10
11 INSERT INTO Player (Player_ID, Player_Name, Player_Email, Player_Password, Player_Picture,
12   Player_JoinDate, Player_Salary, Player_Play_Hours, Player_DOB)
13 VALUES (seq_player_id.NEXTVAL, 'Player 3', 'player3@example.com', 'playerpass', 'player3.jpg',
14   TO_DATE('2022-03-01', 'YYYY-MM-DD'), 7000, 150, TO_DATE('1994-09-20', 'YYYY-MM-DD'));
15
16 INSERT INTO Player (Player_ID, Player_Name, Player_Email, Player_Password, Player_Picture,
17   Player_JoinDate, Player_Salary, Player_Play_Hours, Player_DOB)
18 VALUES (seq_player_id.NEXTVAL, 'Player 4', 'player4@example.com', 'playerpass', 'player4.jpg',
19   TO_DATE('2022-04-01', 'YYYY-MM-DD'), 8000, 120, TO_DATE('1996-12-05', 'YYYY-MM-DD'));
20
21 INSERT INTO Player (Player_ID, Player_Name, Player_Email, Player_Password, Player_Picture,
22   Player_JoinDate, Player_Salary, Player_Play_Hours, Player_DOB)
23 VALUES (seq_player_id.NEXTVAL, 'Player 5', 'player5@example.com', 'playerpass', 'player5.jpg',
24   TO_DATE('2022-05-01', 'YYYY-MM-DD'), 9000, 180, TO_DATE('1998-03-15', 'YYYY-MM-DD'));
```

Inserting data into Player tables

Results

Explain

Describe

Saved SQL

History

| PLAYER_ID | PLAYER_NAME | PLAYER_EMAIL | PLAYER_PASSWORD | PLAYER_PICTURE | PLAYER_JOINDATE | PLAYER_SALARY | PLAYER_PLAY_HOURS | PLAYER_DOB |
|-----------|-------------|---------------------|-----------------|----------------|-----------------|---------------|-------------------|------------|
| 1 | Player 1 | player1@example.com | playerpass | player1.jpg | 01-JAN-22 | 5000 | 100 | 01-JAN-90 |
| 2 | Player 2 | player2@example.com | playerpass | player2.jpg | 01-FEB-22 | 6000 | 200 | 10-MAY-92 |
| 3 | Player 3 | player3@example.com | playerpass | player3.jpg | 01-MAR-22 | 7000 | 150 | 20-SEP-94 |
| 4 | Player 4 | player4@example.com | playerpass | player4.jpg | 01-APR-22 | 8000 | 120 | 05-DEC-96 |
| 5 | Player 5 | player5@example.com | playerpass | player5.jpg | 01-MAY-22 | 9000 | 180 | 15-MAR-98 |

5 rows returned in 0.00 seconds

Inserted data of Player table

```

1 INSERT INTO Player_Address (Pa_ID, Player_ID, Player_Country, Player_City, Player_Street,
  Player_Zip_Code)
2 VALUES (seq_pa_id.NEXTVAL, 1, 'USA', 'New York', '123 Street', '10001');
3 INSERT INTO Player_Address (Pa_ID, Player_ID, Player_Country, Player_City, Player_Street,
  Player_Zip_Code)
4 VALUES (seq_pa_id.NEXTVAL, 2, 'USA', 'Los Angeles', '456 Avenue', '90001');
5 INSERT INTO Player_Address (Pa_ID, Player_ID, Player_Country, Player_City, Player_Street,
  Player_Zip_Code)
6 VALUES (seq_pa_id.NEXTVAL, 3, 'UK', 'London', '789 Road', 'SW1A 1AA');
7 INSERT INTO Player_Address (Pa_ID, Player_ID, Player_Country, Player_City, Player_Street,
  Player_Zip_Code)
8 VALUES (seq_pa_id.NEXTVAL, 4, 'Canada', 'Toronto', '321 Boulevard', 'M5V 2T3');
9 INSERT INTO Player_Address (Pa_ID, Player_ID, Player_Country, Player_City, Player_Street,
  Player_Zip_Code)
10 VALUES (seq_pa_id.NEXTVAL, 5, 'Germany', 'Berlin', '987 Strasse', '12345');

```

Inserting data into Player_Address tables

Results

Explain

Describe

Saved SQL

History

| PA_ID | PLAYER_ID | PLAYER_COUNTRY | PLAYER_CITY | PLAYER_STREET | PLAYER_ZIP_CODE |
|-------|-----------|----------------|-------------|---------------|-----------------|
| 1 | 1 | USA | New York | 123 Street | 10001 |
| 2 | 2 | USA | Los Angeles | 456 Avenue | 90001 |
| 3 | 3 | UK | London | 789 Road | SW1A 1AA |
| 4 | 4 | Canada | Toronto | 321 Boulevard | M5V 2T3 |
| 5 | 5 | Germany | Berlin | 987 Strasse | 12345 |

5 rows returned in 0.00 seconds

Inserted data of Player Address table

```

1 INSERT INTO PLAYER_SOCIAL_LINK (Psl_ID, Player_ID, Player_Facebook_Link,
  Player_Instagram_Link, Player_Twitter_Link, Player_Youtube_Link)
2 VALUES (seq_psl_id.NEXTVAL, 1, 'https://facebook.com/player1', 'https://instagram.com/
  player1', 'https://twitter.com/player1', 'https://youtube.com/player1');
3
4 INSERT INTO PLAYER_SOCIAL_LINK (Psl_ID, Player_ID, Player_Facebook_Link,
  Player_Instagram_Link, Player_Twitter_Link, Player_Youtube_Link)
5 VALUES (seq_psl_id.NEXTVAL, 2, 'https://facebook.com/player2', 'https://instagram.com/
  player2', 'https://twitter.com/player2', 'https://youtube.com/player2');
6
7 INSERT INTO PLAYER_SOCIAL_LINK (Psl_ID, Player_ID, Player_Facebook_Link,
  Player_Instagram_Link, Player_Twitter_Link, Player_Youtube_Link)
8 VALUES (seq_psl_id.NEXTVAL, 3, 'https://facebook.com/player3', 'https://instagram.com/
  player3', 'https://twitter.com/player3', 'https://youtube.com/player3');
9
10 INSERT INTO PLAYER_SOCIAL_LINK (Psl_ID, Player_ID, Player_Facebook_Link,
  Player_Instagram_Link, Player_Twitter_Link, Player_Youtube_Link)
11 VALUES (seq_psl_id.NEXTVAL, 4, 'https://facebook.com/player4', 'https://instagram.com/
  player4', 'https://twitter.com/player4', 'https://youtube.com/player4');
12
13 INSERT INTO PLAYER_SOCIAL_LINK (Psl_ID, Player_ID, Player_Facebook_Link,
  Player_Instagram_Link, Player_Twitter_Link, Player_Youtube_Link)
14 VALUES (seq_psl_id.NEXTVAL, 5, 'https://facebook.com/player5', 'https://instagram.com/
  player5', 'https://twitter.com/player5', 'https://youtube.com/player5');

```

Inserting data into Player_SocialLink tables

Results

Explain

Describe

Saved SQL

History

| PSL_ID | PLAYER_ID | PLAYER_FACEBOOK_LINK | PLAYER_INSTAGRAM_LINK | PLAYER_TWITTER_LINK | PLAYER_YOUTUBE_LINK |
|--------|-----------|------------------------------|-------------------------------|-----------------------------|-----------------------------|
| 1 | 1 | https://facebook.com/player1 | https://instagram.com/player1 | https://twitter.com/player1 | https://youtube.com/player1 |
| 2 | 2 | https://facebook.com/player2 | https://instagram.com/player2 | https://twitter.com/player2 | https://youtube.com/player2 |
| 3 | 3 | https://facebook.com/player3 | https://instagram.com/player3 | https://twitter.com/player3 | https://youtube.com/player3 |
| 4 | 4 | https://facebook.com/player4 | https://instagram.com/player4 | https://twitter.com/player4 | https://youtube.com/player4 |
| 5 | 5 | https://facebook.com/player5 | https://instagram.com/player5 | https://twitter.com/player5 | https://youtube.com/player5 |

5 rows returned in 0.00 seconds

Inserted data of Player Social Link table

```

1 INSERT INTO Player_Phone (Pp_ID, Player_ID, Player_Phone)
2 VALUES (seq_pp_id.NEXTVAL, 1, '1112223333');
3
4 INSERT INTO Player_Phone (Pp_ID, Player_ID, Player_Phone)
5 VALUES (seq_pp_id.NEXTVAL, 2, '2223334444');
6
7 INSERT INTO Player_Phone (Pp_ID, Player_ID, Player_Phone)
8 VALUES (seq_pp_id.NEXTVAL, 3, '3334445555');
9
10 INSERT INTO Player_Phone (Pp_ID, Player_ID, Player_Phone)
11 VALUES (seq_pp_id.NEXTVAL, 4, '4445556666');
12
13 INSERT INTO Player_Phone (Pp_ID, Player_ID, Player_Phone)
14 VALUES (seq_pp_id.NEXTVAL, 5, '5556667777');

```

Inserting data into Player_Phone tables

Results

Explain

Describe

Saved SQL

History

| PP_ID | PLAYER_ID | PLAYER_PHONE |
|-------|-----------|--------------|
| 1 | 1 | 1112223333 |
| 2 | 2 | 2223334444 |
| 3 | 3 | 3334445555 |
| 4 | 4 | 4445556666 |
| 5 | 5 | 5556667777 |

5 rows returned in 0.00 seconds

Inserted data of Player Phone table

```

1 INSERT INTO Player_Winning (Pw_ID, Player_ID, Player_Winning)
2 VALUES (seq_pw_id.NEXTVAL, 1, 'Tournament 2022');
3
4 INSERT INTO Player_Winning (Pw_ID, Player_ID, Player_Winning)
5 VALUES (seq_pw_id.NEXTVAL, 2, 'Championship 2023');
6
7 INSERT INTO Player_Winning (Pw_ID, Player_ID, Player_Winning)
8 VALUES (seq_pw_id.NEXTVAL, 3, 'Cup 2022');
9
10 INSERT INTO Player_Winning (Pw_ID, Player_ID, Player_Winning)
11 VALUES (seq_pw_id.NEXTVAL, 4, 'Tournament 2023');
12
13 INSERT INTO Player_Winning (Pw_ID, Player_ID, Player_Winning)
14 VALUES (seq_pw_id.NEXTVAL, 5, 'League 2022');

```

Inserting data into Player_Winning tables

Results

Explain

Describe

Saved SQL

History

| PW_ID | PLAYER_ID | PLAYER_WINNING |
|-------|-----------|-------------------|
| 1 | 1 | Tournament 2022 |
| 2 | 2 | Championship 2023 |
| 3 | 3 | Cup 2022 |
| 4 | 4 | Tournament 2023 |
| 5 | 5 | League 2022 |

5 rows returned in 0.00 seconds

Inserted data of Player Winning table

```

1 INSERT INTO Player_Team (Pt_ID, Player_ID, Team_ID)
2 VALUES (seq_pt_id.NEXTVAL, 1, 1);
3
4 INSERT INTO Player_Team (Pt_ID, Player_ID, Team_ID)
5 VALUES (seq_pt_id.NEXTVAL, 2, 2);
6
7 INSERT INTO Player_Team (Pt_ID, Player_ID, Team_ID)
8 VALUES (seq_pt_id.NEXTVAL, 3, 3);
9
10 INSERT INTO Player_Team (Pt_ID, Player_ID, Team_ID)
11 VALUES (seq_pt_id.NEXTVAL, 4, 4);
12
13 INSERT INTO Player_Team (Pt_ID, Player_ID, Team_ID)
14 VALUES (seq_pt_id.NEXTVAL, 5, 5);

```

Inserting data into Player_Team tables

Results

Explain

Describe

Saved SQL

History

| PT_ID | PLAYER_ID | TEAM_ID |
|-------|-----------|---------|
| 1 | 1 | 1 |
| 2 | 2 | 2 |
| 3 | 3 | 3 |
| 4 | 4 | 4 |
| 5 | 5 | 5 |

5 rows returned in 0.00 seconds

Inserted data of Player Team table

5.7 Single Row Functions

- Retrieve the email domain for the admin with Admin_ID = 1

```
1 SELECT Admin_Email, SUBSTR(Admin_Email, INSTR(Admin_Email, '@') + 1) AS Email_Domain
2 FROM Admin
3 WHERE Admin_ID = 1;
```

Query 1

| Results | Explain | Describe | Saved SQL | History |
|---------------------------------|---------|--------------------------|-----------|---------|
| ADMIN_EMAIL | | EMAIL_DOMAIN | | |
| admin1@example.com | | example.com | | |
| 1 rows returned in 0.00 seconds | | Download | | |

Result of Query 1

- Get the hire date of the manager named 'John Doe' formatted in a specific way

```
1 SELECT Manager_Name, TO_CHAR(Manager_Hiredate, 'DD-Mon-YYYY') AS Hire_Date
2 FROM Manager
3 WHERE Manager_Name = 'John Doe';
```

Query 2

| Results | Explain | Describe | Saved SQL | History |
|--|---------|----------|-----------|---------|
| <div><div>MANAGER_NAME</div><div>HIRE_DATE</div><div>John Doe01-Jan-2022</div></div> | | | | |
| 1 rows returned in 0.00 seconds Download | | | | |

Result of Query 2

- Concatenate the first and last name of the content creator with ContentCreator_ID = 1

```
1 SELECT ContentCreator_Name, CONCAT(CONCAT(SUBSTR(ContentCreator_Name, 1, INSTR(
   ContentCreator_Name, ' ') - 1), ' '), SUBSTR(ContentCreator_Name, INSTR(
   ContentCreator_Name, ' ') + 1)) AS Full_Name
2 FROM ContentCreator
3 WHERE ContentCreator_ID = 1;
```

Query 3

Results

Explain

Describe

Saved SQL

History

| CONTENTCREATOR_NAME | FULL_NAME |
|---------------------|------------------|
| ContentCreator 1 | ContentCreator 1 |

1 rows returned in 0.00 seconds

[Download](#)

Result of Query 3

5.8 Group Functions

- Calculate the average balance for all finance records

```
1 SELECT AVG(Finance_Balance) AS Average_Balance
2 FROM Finance;
```

Query 1

| Results | Explain | Describe | Saved SQL | History |
|--|---------|----------|-----------|---------|
| AVERAGE_BALANCE | | | | |
| 17000 | | | | |
| 1 rows returned in 0.00 seconds Download | | | | |

Result of Query 1

- Count the number of teams established in each country

```
1 SELECT Team_Country, COUNT(*) AS Team_Count
2 FROM Team
3 GROUP BY Team_Country;
```

Query 2

Results

Explain

Describe

Saved SQL

History

| TEAM_COUNTRY | TEAM_COUNT |
|--------------|------------|
| USA | 1 |
| Germany | 1 |
| Australia | 1 |
| Canada | 1 |
| UK | 1 |

5 rows returned in 0.10 seconds [Download](#)

Result of Query 2

- Calculate the total salary expense for content creators

```
1 SELECT SUM(ContentCreator_Salary) AS Total_Salary_Expense
2 FROM ContentCreator;
```

Query 3

| Results | Explain | Describe | Saved SQL | History |
|--|---------|----------|-----------|---------|
| TOTAL_SALARY_EXPENSE | | | | |
| 20000 | | | | |
| 1 rows returned in 0.00 seconds Download | | | | |

Result of Query 3

5.9 SubQuery

- Retrieve the managers associated with teams established before a specific date

```
1  SELECT Manager_Name
2  FROM Manager
3  WHERE Manager_ID IN (
4      SELECT Manager_ID
5      FROM Team
6      WHERE Team_Established_Date < TO_DATE('2022-03-01', 'YYYY-MM-DD')
7  );
```

Query 1

| Results | Explain | Describe | Saved SQL | History |
|--|---------|----------|-----------|---------|
| MANAGER_NAME | | | | |
| John Doe | | | | |
| Jane Smith | | | | |
| 2 rows returned in 0.00 seconds Download | | | | |

Result of Query 1

- Get the content creators who have a higher salary than the average salary of all content creators

```
1  SELECT ContentCreator_Name
2  FROM ContentCreator
3  WHERE ContentCreator_Salary > (
4      SELECT AVG(ContentCreator_Salary)
5      FROM ContentCreator
6  );
```

Query 2

| Results | Explain | Describe | Saved SQL | History |
|--|---------|----------|-----------|---------|
| CONTENTCREATOR_NAME | | | | |
| ContentCreator 4 | | | | |
| ContentCreator 5 | | | | |
| 2 rows returned in 0.00 seconds Download | | | | |

Result of Query 2

- Retrieve the teams managed by managers who have won a championship

```
1  SELECT Team_Name
2  FROM Team
3  WHERE Manager_ID IN (
4  SELECT Manager_ID
5  FROM Manager
6  WHERE Manager_ID IN (
7      SELECT DISTINCT Manager_ID
8      FROM Team_Winning
9      WHERE Team_Winning LIKE '%Championship%'
10 )
11 );
```

Query 3

| Results | Explain | Describe | Saved SQL | History |
|-----------|---------|----------|-----------|---------|
| TEAM_NAME | | | | |
| Team A | | | | |
| Team B | | | | |
| Team D | | | | |
| Team E | | | | |
| Team C | | | | |

5 rows returned in 0.15 seconds [Download](#)

Result of Query 3

5.10 Join Queries

- Get the team name and manager name for each team

```
1 SELECT t.Team_Name , m.Manager_Name
2 FROM Team t
3 JOIN Manager m ON t.Manager_ID = m.Manager_ID;
```

Query 1

| Results | Explain | Describe | Saved SQL | History |
|-----------|---------|----------------|-----------|---------|
| TEAM_NAME | | MANAGER_NAME | | |
| Team A | | John Doe | | |
| Team B | | Jane Smith | | |
| Team C | | Mike Johnson | | |
| Team D | | Sarah Williams | | |
| Team E | | Robert Davis | | |

5 rows returned in 0.00 seconds [Download](#)

Result of Query 1

- Retrieve the player name, team name, and country for each player

```
1 SELECT p.Player_Name , t.Team_Name , t.Team_Country
2 FROM Player p
3 JOIN Player_Team pt ON p.Player_ID = pt.Player_ID
4 JOIN Team t ON pt.Team_ID = t.Team_ID;
```

Query 2

| Results | Explain | Describe | Saved SQL | History |
|-------------|-----------|--------------|-----------|---------|
| PLAYER_NAME | TEAM_NAME | TEAM_COUNTRY | | |
| Player 1 | Team A | USA | | |
| Player 2 | Team B | UK | | |
| Player 3 | Team C | Australia | | |
| Player 4 | Team D | Canada | | |
| Player 5 | Team E | Germany | | |

5 rows returned in 0.00 seconds [Download](#)

Result of Query 2

- Get the content creator name, social media name, and email for each content creator

```
1 SELECT cc.ContentCreator_Name, sm.SocialMedia_Name, sm.SocialMedia_Email
2 FROM ContentCreator cc
3 JOIN SocialMedia sm ON cc.SOCIALMEDIA_ID = sm.SocialMedia_ID;
```

Query 3

Results

Explain

Describe

Saved SQL

History

| CONTENTCREATOR_NAME | SOCIALMEDIA_NAME | SOCIALMEDIA_EMAIL |
|---------------------|---------------------|--------------------------|
| ContentCreator 1 | SocialMediaUser221 | social.user1@example.com |
| ContentCreator 2 | SocialMediaUser442 | social.user2@example.com |
| ContentCreator 3 | SocialMediaUser423 | social.user3@example.com |
| ContentCreator 4 | SocialMediaUser4234 | social.user4@example.com |
| ContentCreator 5 | SocialMediaUser5523 | social.user5@example.com |

5 rows returned in 0.01 seconds

[Download](#)

Result of Query 3

5.11 Creating View

- Create a view to display the details of managers and their associated teams

```
1 CREATE VIEW ManagerTeamView AS
2 SELECT m.Manager_Name, t.Team_Name, t.Team_Country
3 FROM Manager m
4 JOIN Team t ON m.Manager_ID = t.Manager_ID;
```

ManagerTeamView

| Results | Explain | Describe | Saved SQL | History |
|--------------|---------|----------|-----------|---------|
| MANAGER_NAME | | | | |
| Team A | | | | |
| Team B | | | | |
| Team C | | | | |
| Team D | | | | |
| Team E | | | | |
| USA | | | | |
| UK | | | | |
| Australia | | | | |
| Canada | | | | |
| Germany | | | | |

5 rows returned in 0.00 seconds [Download](#)

Result of ManagerTeamView

- Create a view to show the average salary of content creators

```
1 CREATE VIEW AverageSalaryView AS
2 SELECT AVG(ContentCreator_Salary) AS Average_Salary
3 FROM ContentCreator;
```

AvgSalaryView

| Results | Explain | Describe | Saved SQL | History |
|----------------|---------|----------|-----------|---------|
| AVERAGE_SALARY | | | | |
| 4000 | | | | |

1 rows returned in 0.00 seconds [Download](#)

Result of AverageSalaryView

- Create a view to list the players and their corresponding teams

```
1 CREATE VIEW PlayerTeamView AS
2 JOIN Team t ON pt.Team_ID = t.Team_ID;
3 SELECT p.Player_Name, t.Team_Name
4 FROM Player p
5 JOIN Player_Team pt ON p.Player_ID = pt.Player_ID
```

PlayerTeamView

| Results | Explain | Describe | Saved SQL | History |
|-------------|---------|-----------|-----------|---------|
| PLAYER_NAME | | TEAM_NAME | | |
| Player 1 | | Team A | | |
| Player 2 | | Team B | | |
| Player 3 | | Team C | | |
| Player 4 | | Team D | | |
| Player 5 | | Team E | | |

5 rows returned in 0.00 seconds [Download](#)

Result of PlayerTeamView

5.12 Synonyms

```

1 -- Create synonym for the ORGANIZATION_TOURNAMENT table
2 CREATE SYNONYM org_tour FOR Organization_Tournament;
3
4 -- Create synonym for the CONTENTCREATOR_SOCIALMEDIA table
5 CREATE SYNONYM cc_sm FOR ContentCreator_SocialMedia;
6
7 -- Create synonym for the CONTENTCREATOR_PHONE table
8 CREATE SYNONYM cc_ph FOR ContentCreator_Phone;

```

Synonyms

| Results | Explain | Describe | Saved SQL | History |
|-----------|--------------|-------------|----------------------------|---------|
| OWNER | SYNONYM_NAME | TABLE_OWNER | TABLE_NAME | DB_LINK |
| ESPORTFTW | CC_PH | ESPORTFTW | CONTENTCREATOR_PHONE | - |
| ESPORTFTW | CC_SM | ESPORTFTW | CONTENTCREATOR_SOCIALMEDIA | - |
| ESPORTFTW | ORG_TOUR | ESPORTFTW | ORGANIZATION_TOURNAMENT | - |

3 rows returned in 1.38 seconds [Download](#)

List of Synonyms

Relational Algebra

1. Find the name of the manager whose manager id is 5.

$$\Pi_{\text{Manager_Name}}(\sigma_{\text{Manager_ID}=5}(\text{Manager}))$$

2. Find the Salary of 'Player 1'.

$$\Pi_{\text{Player_Salary}}(\sigma_{\text{name}=\text{'Player 1'}}(\text{Player}))$$

3. Find Player id whose birthday is on 1992-05-10.

$$\Pi_{\text{Player_ID}}(\sigma_{\text{Player_DOB}=\text{'1992-05-10'}}(\text{Player}))$$

4. Find the Country, City, Street, and Zip code where Content Creator ID is equal to 4.

$$\Pi_{\text{ContentCreator_Country}, \text{ContentCreator_City}, \text{ContentCreator_Street}, \text{ContentCreator_Zip_Code}} \\ (\sigma_{\text{ContentCreator_ID}=4}(\text{ContentCreator_Address}))$$

5. Find the Team ID that won the Championship 2023.

$$\Pi_{\text{Team_ID}}(\sigma_{\text{Team_Winning}=\text{'Championship 2023'}}(\text{Team_Winning}))$$

Conclusion

In conclusion, the project for the development and implementation of an Esports Management System has outlined a revolutionary platform that aims to transform the management and organization of esports teams, players, tournaments, and sponsors. The proposed system seeks to improve user experience, encourage community engagement, and streamline operations within the esports industry by leveraging advanced technology and comprehensive functionalities.

For the final term, we plans to enhance the existing project by developing a web application using Dotnet and Sveltekit frameworks. This transition to a web app will provide greater accessibility and flexibility to users, allowing them to access the Esports Management System from any device with an internet connection.