Ryerson University

Department of Electrical & Computer Engineering
Faculty of Engineering & Architectural Science

Course Title:	Database Systems I
Course Number:	CPS510
Semester/Year (e.g.F2016)	Fall 2021

Instructor:	Dr. Soheila Bashardoust Tajali

Assignment/Lab Title:	Final Report

Submission Date:	December 2, 2021
Due Date:	December 3, 2021

Student LAST Name	Student FIRST Name	Student Number	Signature
Logozzo	Salvatore	****56504	SL
Tran	Eddy	****82033	ET
Chiu Tran	Phuong Nam	****40717	PNCT

*By signing above you attest that you have contributed to this written lab report and confirm that all work you have contributed to this lab report is your own work. Any suspicion of copying or plagiarism in this work will result in an investigation of Academic Misconduct and may result in a "0" on the work, an "F" in the course, or possibly more severe penalties, as well as a Disciplinary Notice on your academic record under the Student Code of Academic Conduct, which can be found online at: http://www.ryerson.ca/senate/current/pol60.pdf

Table of Contents

Part 1: Assignment 10 - RA Queries

SQL	Relational Algebra
Select League_ID, Price_to_join, Team_ID From League Where price_to_join >= 25	$\Pi_{\text{League\_ID, Price\_to\_join, Team\_ID}}\ (\sigma_{\text{Price\_to\_join >= 25}}(\text{League}))$
SELECT Goalkeeper.Name, Goalkeeper.Team_ID, Goalkeeper.NumPenaltySaved, Lineup.Lineup_ID FROM Goalkeeper JOIN Lineup ON Lineup.Team_ID = Goalkeeper.Team_ID WHERE NumPenaltySaved >= 3;	$\Pi_{\text{Name,Team\_ID,NumPenaltySaved,Lineup\_ID}}(\sigma_{\text{NumPenaltySaved>=3}}(\text{Lineup} \bowtie \text{Goalkeeper}))$
SELECT team_name, team_id FROM team MINUS (SELECT t.team_name, s.team_id FROM team t, standings s WHERE s.team_id = t.team_id);	$\text{Sub} \leftarrow \Pi_{\text{Team\_Name,Team\_ID}}\ (\text{Team} \bowtie \text{Standings})$ $\text{Team} \leftarrow \Pi_{\text{Team\_name,Team\_ID}}(\text{Team})$ $\text{Result} \leftarrow \text{Team} - \text{Sub}$
SELECT * FROM Match Where Match_Day_Num = 1;	$\sigma_{\text{Match\_Day\_Num=1}}(\text{Match})$
SELECT t.team_name, s.team_id FROM standings s, team t WHERE s.league_points = 0 AND s.team_id = t.team_id UNION (SELECT team_name, team_id FROM team WHERE user_id is NULL);	$\text{Zero} \leftarrow \Pi_{\text{Team\_Name,Team\_ID}}(\sigma_{\text{league\_points=0}}(\text{Team} \bowtie \text{Standings}))$ $\text{Null} \leftarrow \Pi_{\text{Team\_Name,Team\_ID}}(\sigma_{\text{user\_id = null}}(\text{team}))$ $\text{Result} \leftarrow \text{Zero} \cup \text{Null}$
SELECT ROLE, COUNT(USER_ID) AS COUNT FROM usertb GROUP BY ROLE	$\Pi_{\text{Role,Count}} \leftarrow F_{\text{Count USER\_ID}}\ (\text{usertb})$ $\Pi_{\text{Role,Count}}(_{\text{Role}}F_{\text{P count(count(USER\_ID))}}(\text{usertb}))$ where P is rename and F is aggregation

Part 2: Final Report

Project Title: Fantasy Soccer League DBMS

Topic Description:

Sports has been around for many centuries and is a great way to spend quality time with friends and family, whether to watch or play it. However in recent years, sports viewership has been on a downward trend due to the pandemic, keeping fans away from stadiums/arenas and making them lose interest in the sport itself. Streaming services, such as Netflix, are shifting consumers' interests away from sports and towards tv shows and movies. To combat this, fantasy sports was created and has become very popular in the past few years to recapture those lost viewers and solidify the remaining viewer base. Fantasy sports consist of a group of people, from anywhere in the world, (that can be a group of friends or just a random generated league) that join a league with the intent of buying players that, with their statistics, will make points for the team manager. Points are updated after each game. At the end of the year, the top 3 team managers in the league will split the prize pool (Total $$ collected from all players joining the league.)

The team owners (we call them managers) can buy players from any team of the desired soccer league (i.e., Serie A, Premier League, etc.) and can set different lineups (i.e., 4-4-2, 4-3-3, 4-3-1-2, etc.). The league will then generate games between different managers for each matchday in the real league. For every matchday, managers from the league must set their lineups to play against another opponent, just like a soccer game. Every stat from their players is going to gain or lose game points for each manager. At the end of the matchday, the team with the highest game points will get 3 league points for the general standing (0 league points for loss and 1 league point for a draw).

A team manager/admin (USER) has various attributes like its name, email, password, managerID, leagueID, role in the league, and total remaining credits (the initial credits will be decided by the league admin before starting the auction for players). The league admin is responsible for setting up the rules and adding players to each manager's team at the end of the auction.

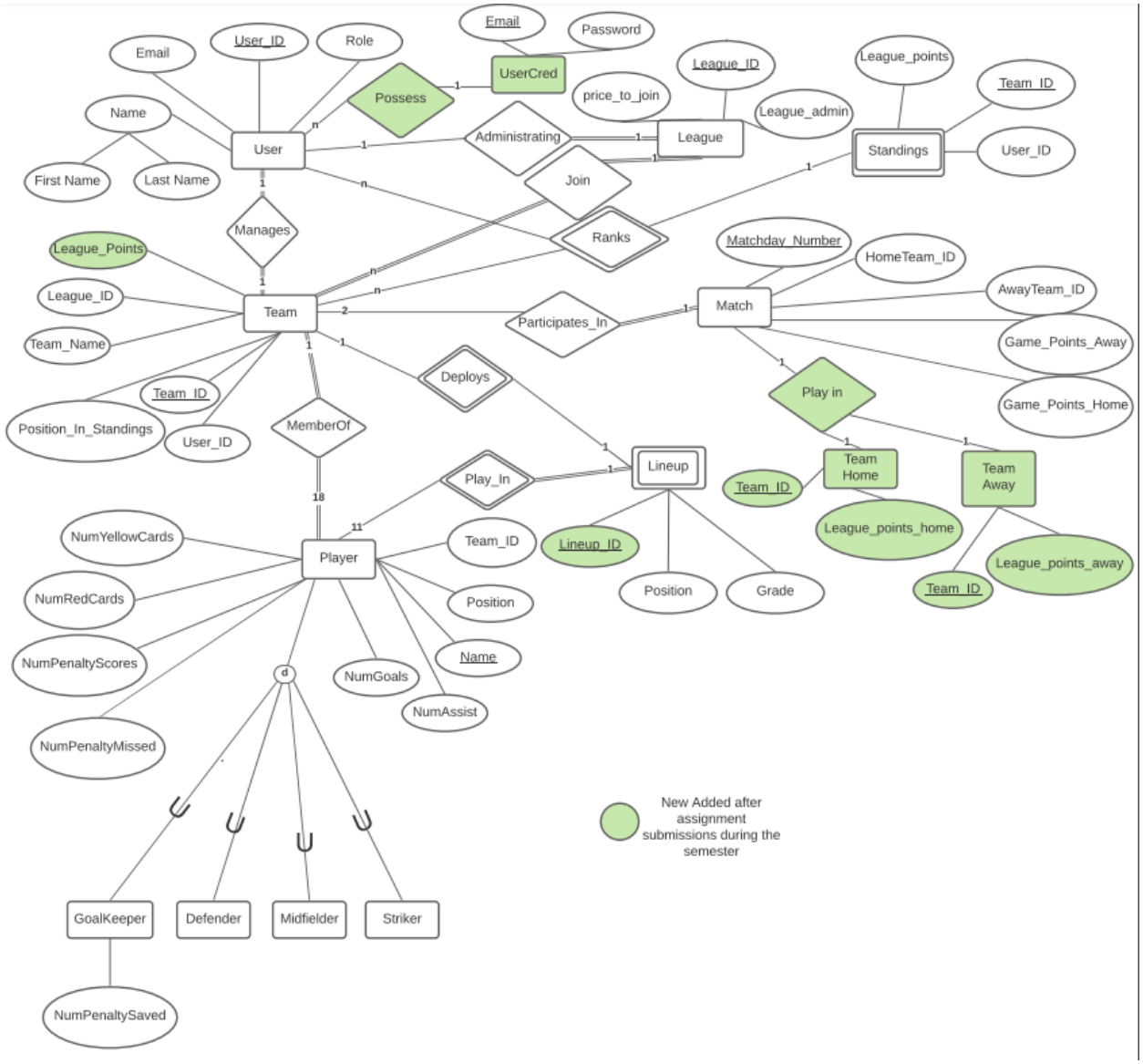
A league has a leagueID, a league admin and a price to join. This allows players to select which league they would like to participate in. The price to join is the amount required to participate in the league and will be added to the prize pool for the end of the league.

Each team has various attributes, like Team name (chosen by its manager), Team_ID, ManagerID, position in standings (rank highest to lowest league points) and League_ID. The last attribute creates a relationship between the league entity and team entity.

Players are to be drafted from managers, and have attributes which will give or take game points for managers that put them in their lineup for the gameday. Their attributes are: Player name, role, #Goals, #Assists, #Yellow cards, #Red cards, #Penalties scored, #Penalties missed, #Goal against (goalkeeper), #Penalties saved (goalkeeper), and finally the teamID of the team that drafted them.

Lastly, standings have a TeamID, ManagerID, and League points. This table will display the current position of each team relative to other teams in the same fantasy league.

ER Model


Email
Password
UserCred
Email
User_ID
Role
Possess
Name
User
First Name
Last Name
Administrating
League_ID
price_to_join
League
League_admin
League_points
Team_ID
Standings
User_ID
Join
Manages
Ranks
League_Points
League_ID
Team_Name
Team
Participates_In
Matchday_Number
HomeTeam_ID
AwayTeam_ID
Match
Game_Points_Away
Game_Points_Home
Deploys
Play in
Team_ID
Position_In_Standings
User_ID
MemberOf
Play_In
Lineup
Team_ID
Team Home
Team Away
League_points_home
League_points_away
Team_ID
NumYellowCards
NumRedCards
NumPenaltyScores
NumPenaltyMissed
Player
Team_ID
Lineup_ID
Position
Grade
Position
Name
NumGoals
NumAssist
d
GoalKeeper
Defender
Midfielder
Striker
NumPenaltySaved
New Added after assignment submissions during the semester

Relational Schema (UPDATED: Nov 29, 2021)

Usertb(User_ID, First_Name, Last_Name, Email, Role)

UserCred (Email, Password)

League(League_ID, League_Admin, price_to_join, Team_ID)

Team(Team_ID, Team_Name,Position_In_Standing, League_Points, User_ID)

Goalkeeper (Player_Name, Position, Team_ID, NumYellowCards, NumRedCards, NumPenaltyScored, NumPenaltyMissed, NumGoals, NumAssists)

Defender (Player_Name, Position, Team_ID, NumYellowCards, NumRedCards, NumPenaltyScored, NumPenaltyMissed, NumGoals, NumAssists)

Midfielder (Player_Name, Position, Team_ID, NumYellowCards, NumRedCards, NumPenaltyScored, NumPenaltyMissed, NumGoals, NumAssists)

Striker (Player_Name, Position, Team_ID, NumYellowCards, NumRedCards, NumPenaltyScored, NumPenaltyMissed, NumGoals, NumAssists)

Match (Match_Day_Num, Team_Home (ID), Team_Away (ID), Game_Points_Home, Game_Points_Away)

Lineup (Lineup_ID, Name, Position, Grade)

Standings (Team_ID, League_Points)

Simple Queries

1. Goalkeeper (Player_Name, Position, Team_ID, NumPenaltySaved,NumYellow_Cards, Num Red_cards, NumPenalties_scored, Num Penalties_missed, NumGoals, NumAssists)

```
SELECT Goalkeeper.Name, Goalkeeper.Team_ID, Goalkeeper.NumPenaltySaved
FROM Goalkeeper
WHERE NumPenaltySaved >= 3;
```

Selects Name, Team_ID, and NumPenaltySaved from Goalkeeper table if that tuple has more than or equal to 3 NumPenaltySaved

Results of Query

NAME	TEAM_ID	NUMPENALTYSAVED
Szczesny	(null)	3
Meret	(null)	4

2. Midfielder (Name, Position, Team_ID, NumYellowCards, NumRedcards, NumPenaltyScored, NumPenaltyMissed, NumGoals, NumAssists)

```
SELECT Name, Team_ID, NumAssists
From Midfielder
ORDER BY NumAssists DESC;
```

Selects Name, Team_ID, and NumAssists from table Midfielder, where the order is from highest to lowest of NumAssists.

Results of Query

NAME	TEAM_ID	NUMASSISTS
Bentancur	(null)	0
Zortea	(null)	0
Barella	(null)	0
Vidal	(null)	0
Lozano	(null)	0
Insigne	(null)	0
Kessie	(null)	0
Maldini	(null)	0
Cristante	(null)	0
Veretout	(null)	0
De Roon	(null)	0
Pessina	(null)	0
Luis Alberto	(null)	0
Leiva	(null)	0
Torreira	(null)	0
Duncan	(null)	0
Frattesi	(null)	0
Djuricic	(null)	0

3. Defender (Name, Position, Team_ID, NumYellowCards, NumRedcards, NumPenaltyScored, NumPenaltyMissed, NumGoals, NumAssists)

```
SELECT Name, Team_ID, NumRedCards
From Defender
ORDER BY NumRedCards DESC;
```

Selects Name, Team_ID, and NumRedCards from table Defender, where the order is from highest to lowest of NumRedCards.

Results of Query

NAME	TEAM_ID	NUMREDCARDS
Toloi	(null)	4
Hysaj	(null)	2
Skriniar	(null)	0
De Vrij	(null)	0
Koulibaly	(null)	0
Manolas	(null)	0
Kjaer	(null)	0
Theo	(null)	0
Mancini	(null)	0
Smalling	(null)	0
Gosens	(null)	0
Acerbi	(null)	0
Milenkovic	(null)	0
Biraghi	(null)	0
Chiriches	(null)	0
Ferrari	(null)	0
Ansaldi	(null)	0
Bremer	(null)	0

4. Striker (Name, Position, Team_ID, NumYellowCards, NumRedcards, NumPenaltyScored, NumPenaltyMissed, NumGoals, NumAssists)

```
SELECT Name, Team_ID, NumPenaltyScored
From Striker
ORDER BY NumPenaltyScored DESC;
```

Selects Name, Team_ID, and NumPenaltyScored from table Striker, where the order is from highest to lowest of NumPenaltyScored.

Results of Query

NAME	TEAM_ID	NUMPENALTYSCORED
Chiesa	(null)	0
Djuric	(null)	0
Lautaro	(null)	0
Dzeko	(null)	0
Politano	(null)	0
Oshimen	(null)	0
Leao	(null)	0
Giroud	(null)	0
Zaniolo	(null)	0
Abraham	(null)	0
Muriel	(null)	0
Ilicic	(null)	0
Immobile	(null)	0
Anderson	(null)	0
Callejon	(null)	0
Vlahovic	(null)	0
Berardi	(null)	0
Scamacca	(null)	0

5. Usertb(User_ID, First_Name, Last_Name, Email, Password, Role)

```
SELECT *
FROM Usertb
WHERE Role = 'user'
ORDER BY Last_Name
```

This query selects every column of the USERTB table that has the role ‘admin’. It orders the row alphabetically based on the items’ ‘Last_Name’.

Results of Query

USER_ID	FIRST_NAME	LAST_NAME	EMAIL	ROLE
717	joseph	ct	plchiu@ryerson.ca	user
1234	friendA	lastA	friendA@friend.ca	user
4321	friendB	lastB	friendB@friend.ca	user
1001	friendC	lastC	friendC@friend.ca	user
6504	salvatore	logozzo	salvatore.logozzo@ryerson.ca	user
2033	eddy	tran	eddy.tran@ryerson.ca	user

```
SELECT  ROLE, COUNT(USER_ID) AS  COUNT
FROM usertb
GROUP BY ROLE;
```

The query selects roles and the count of each role.

Results of Query

ROLE	COUNT
user	6
admin	1

6. Match (Match_Day_Num, Team_Home (ID), Team_Away (ID), Game_Points_Home, Game_Points_Away, League_Points_Home, League_Points_Away)

```
SELECT *
FROM Match
Where Match_Day_Num = 1;
```

Selects all columns of Match if the tuple has attribute Match_Day_Num equal to 1.

Results of Query

MATCH_DAY_NUM	TEAM_HOME	TEAM_AWAY	GAME_POINTS_HOME	GAME_POINTS_AWAY
1	1	2	0	0
1	3	5	0	0

7. Lineup (Lineup_ID, Position, Name, Team_ID, Grade)

```
SELECT Team_ID, Lineup_ID, COUNT (Lineup_ID) AS "Players in Lineup"
FROM Lineup
GROUP BY Team_ID, Lineup_ID
HAVING (Count(Lineup_ID) >= 2);
```

Select Team_ID, Lineup_ID, and the Count of Lineup_IDs (displayed as Players in Lineup) from table Lineup where the team_ids have a count of 2 or greater of the same Lineup_ID.

Results of Query

TEAM_ID	LINEUP_ID	Players in Lineup
(null)	1	4
(null)	2	2

8. Standings (Team_ID, League_Points)

```
SELECT *
FROM Standings
ORDER BY League_Points
```

Select all columns of Standings and order the rows by the League_Points in ascending order.

Results of Query

TEAM_ID	LEAGUE_POINTS
5	0
8	1
2	1
9	1
1	2
4	2

9. Team(Team_Name, Team_ID, Position_In_Standing, League_Points, User_ID)

```
SELECT *
 FROM team
ORDER BY Team_Name;
```

Selects all columns of the Team table and orders it alphabetically by Team_Name.

Results of Query

	TEAM_NAME	TEAM_ID	POSITION_IN_STANDING	LEAGUE_POINTS	USER_ID	LEAGUE_ID
1	Atalanta	5	(null)	0	1234	(null)
2	Cagliari	10	(null)	0	(null)	(null)
3	Fiorentina	7	(null)	0	(null)	(null)
4	Juventus	3	(null)	0	(null)	(null)
5	Lazio	6	(null)	0	(null)	(null)
6	Milan	1	(null)	0	6504	(null)
7	Roma	4	(null)	0	717	(null)
8	Sassuolo	8	(null)	0	4321	(null)
9	Torino	9	(null)	0	1001	(null)
10	team2	2	(null)	0	2033	(null)

10. League(League_ID, League_Admin, Price_to_Join, Team_ID)

```
SELECT League_ID, Price_to_join, Team_ID
FROM League
WHERE Price_to_Join >= 25
ORDER BY Price_to_Join DESC;
```

Selects League_ID, Price_to_join, and Team_ID from the League table where the price to join is $25+, ordered from highest to lowest in price.

Results of Query

	LEAGUE_ID	PRICE_TO_JOIN	TEAM_ID
1	11	45	3
2	5	25	7
3	9	25	4
4	12	25	10
5	13	25	1

Advanced Queries and Views

1. Standings (Team_ID, League_Points)

```
SELECT  team.team_name, usertb.first_name, standings.team_id, standings.league_points
FROM standings
JOIN team ON team.team_id = standings.team_id
JOIN usertb ON usertb.user_id = team.user_id
ORDER BY LEAGUE_POINTS DESC;
```

A query that displays the standings of the teams that have a user on it, with the team name, the name of the team's user, the team ID and the team's league points. This is done by using two join commands to join the team name and the team's user. Since this is supposed to display the standings, it will be ordered in descending order based on the team's league points.

Results of Query

TEAM_NAME	FIRST_NAME	TEAM_ID	LEAGUE_POINTS
Roma	joseph	4	2
Milan	salvatore	1	2
Torino	friendC	9	1
team2	eddy	2	1
Sassuolo	friendB	8	1
Atalanta	friendA	5	0

2. League(League_ID, League_Admin, Price_to_Join, Team_ID)

```
SELECT league.team_id, team.team_name, team.league_points
FROM League
JOIN team ON team.Team_ID = league.Team_ID
ORDER BY team.Team_ID
```

This query connects the LEAGUE and TEAM table using Team_ID, the foreign key. The program displays TEAM_ID from the LEAGUE table and TEAM_NAME and LEAGUE_POINTS from the TEAM table. The rows are ordered by Team_ID from table TEAM.

Results of Query

TEAM_ID	TEAM_NAME	LEAGUE_POINTS
1	Milan	0
2	team2	0
3	Juventus	0
4	Roma	0
5	Atalanta	0
6	Lazio	0
7	Fiorentina	0
8	Sassuolo	0
9	Torino	0
10	Cagliari	0

3. Team(Team_Name, Team_ID, Position_In_Standing, League_Points, User_ID)

```
CREATE VIEW No_User_Teams AS (SELECT team.team_name, team.team_id FROM team
WHERE USER_ID IS NULL);


SELECT *
FROM No_User_Teams
ORDER BY Team_Name;
```

This query creates a view for teams that have not yet been chosen by a user. These teams are displayed with their team name and team ID while the calling of the view has the team name ordered alphabetically.

Results of Query

TEAM_NAME	TEAM_ID
Cagliari	10
Fiorentina	7
Juventus	3
Lazio	6

```
CREATE VIEW team_user AS (
SELECT team.team_name, team.team_id, team.user_id, usertb.first_name,usertb.last_name
FROM team
JOIN usertb ON team.user_id = usertb.user_id);

SELECT *
 FROM team_user
ORDER BY Team_Name;
```

This query creates a view that joins the teams and users that match by the userID and displays the names of the user with their corresponding teams.

Results of Query

TEAM_NAME	TEAM_ID	USER_ID	FIRST_NAME	LAST_NAME
Atalanta	5	1234	friendA	lastA
Milan	1	6504	salvatore	logozzo
Roma	4	717	joseph	ct
Sassuolo	8	4321	friendB	lastB
Torino	9	1001	friendC	lastC
team2	2	2033	eddy	tran

4. Goalkeeper (Player_Name, Position, Team_ID, NumPenaltySaved,NumYellow_Cards, Num Red_cards, NumPenalties_scored, Num Penalties_missed, NumGoals, NumAssists)

```
SELECT Goalkeeper.Name, Goalkeeper.Team_ID, Goalkeeper.NumPenaltySaved,
Lineup.Lineup_ID
FROM Goalkeeper
JOIN Lineup ON Lineup.Team_ID = Goalkeeper.Team_ID
WHERE NumPenaltySaved >= 3;
```

Displays columns Name, Team_ID, and NumPenaltySaved of table GOALKEEPER and Lineup_ID of table LINEUP where NumPenaltySaved is greater than or equal to 3 where the goalkeeper is in a lineup with matching team_ids.

Results of Query

NAME	TEAM_ID	NUMPENALTYSAVED	LINEUP_ID
Szczesny	1	3	1

5. Match (Match_Day_Num, Team_Home (ID), Team_Away (ID), Game_Points_Home, Game_Points_Away, League_Points_Home, League_Points_Away)

```
CREATE VIEW MatchDay1 (Home, Away, Home_Points, Away_Points) AS
(SELECT T1.Team_Name, T2.Team_Name, Match.Game_Points_Home,
Match.Game_Points_Away
FROM Match
JOIN Team T1 ON T1.Team_ID = Match.Team_Home
JOIN Team T2 ON T2.Team_ID = Match.Team_Away);

SELECT *
FROM MatchDay1
ORDER BY Home;
```

Creates a view, MatchDay1, that has attributes Home, Away, Home_Points, Away_Points. This view joins tables Team and Match. Home and Away are the names of the teams participating in the match and Home_Points and Away_Points are the points respective to their team.

Results of Query

HOME	AWAY	HOME_POINTS	AWAY_POINTS
Milan	team2	0	0

/* **EXISTS**, selects unique admin users of a league and displays their user data*/
SELECT u.user_id "User Id", u.first_name "First Name", u.last_name "Last Name", u.email "Email"
FROM usertb u
WHERE EXISTS
(SELECT DISTINCT l.League_Admin FROM league l WHERE l.league_admin = u.user_id);

	User Id	First Name	Last Name	Email
1	6504	salvatore	logozzo	salvatore.logozzo@ryerson.ca

/* **UNION**, selects teams with zero points from standing and teams with no user, look for inactive teams*/
SELECT t.team_name, s.team_id FROM standings s, team t
WHERE s.league_points = 0 AND s.team_id = t.team_id
UNION
(SELECT team_name, team_id FROM team
WHERE user_id is NULL)
ORDER BY team_id;

	TEAM_NAME	TEAM_ID
1	Juventus	3
2	Atalanta	5
3	Lazio	6
4	Fiorentina	7
5	Cagliari	10

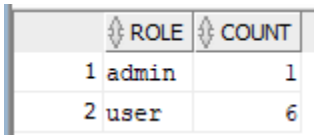
/* **MINUS**, missing teams not in the standings*/

SELECT team_name, team_id
FROM team
MINUS
(SELECT t.team_name, s.team_id
FROM team t, standings s
WHERE s.team_id = t.team_id)
ORDER BY team_id;

	TEAM_NAME	TEAM_ID
1	Juventus	3
2	Lazio	6
3	Fiorentina	7
4	Cagliari	10

/* **COUNT**, selects role and count the different roles of the user, in this case counts the amount of user and admin roles in the user table */

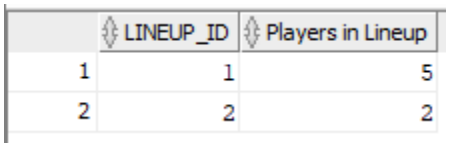
```
SELECT  ROLE, COUNT(USER_ID) AS  COUNT
FROM usertb
GROUP BY ROLE
ORDER BY ROLE;
```



	ROLE	COUNT
1	admin	1
2	user	6

/* **HAVING**, selects lineup ID from lineup and counts the number of players in a lineup and group by lineup ID*/

```
SELECT Lineup_ID, COUNT (Lineup_ID) AS "Players in Lineup"
FROM Lineup
GROUP BY Team_ID, Lineup_ID
HAVING (Count(Lineup_ID) >= 2);
```



	LINEUP_ID	Players in Lineup
1	1	5
2	2	2

Shell UI

$ CPS510GUI.sh

```
#!/bin/sh
MainMenu()
{

 while [ "$CHOICE" != "START" ]
 do
 clear
 echo
"============================================================="
 echo "| Oracle All Inclusive Tool
|"
 echo "| Main Menu - Select Desired Operation(s):
|"
 echo "| <CTRL-Z Anytime to Enter Interactive CMD Prompt>
|"
 echo "-------------------------------------------------------------
----"
 echo " $IS_SELECTEDM M) View Manual"
 echo " "
 echo " $IS_SELECTED1 1) Drop Tables"
 echo " $IS_SELECTED2 2) Create Tables"
 echo " $IS_SELECTED3 3) Populate Tables"
 echo " $IS_SELECTED4 4) Query Tables"
 echo " "
 echo " $IS_SELECTEDX X) Force/Stop/Kill Oracle DB"
 echo " "
 echo " $IS_SELECTEDE E) End/Exit"
 echo "Choose: "
 read CHOICE
 if [ "$CHOICE" == "0" ]
 then
 echo "Nothing Here"
 elif [ "$CHOICE" == "1" ]
 then
 bash drop_tables.sh
 Pause
 elif [ "$CHOICE" == "2" ]
 then
```

```
39      bash create_tables.sh
40      Pause
41      elif [ "$CHOICE" == "3" ]
42      then
43      bash populate_tables.sh
44      Pause
45      elif [ "$CHOICE" == "4" ]
46      then
47      bash queries.sh
48    Pause
49
50      elif [ "$CHOICE" == "E" ]
51      then
52      exit
53      fi
54      done
55    }
56    #--COMMENTS BLOCK--
57    # Main Program
58    #--COMMENTS BLOCK--
59    ProgramStart()
60    {
61      StartMessage
62      while [ 1 ]
63      do
64      MainMenu
65
66      done
67    }
68    ProgramStart
```

Figure 1: GUI Code.

```
$ drop_tables.sh
1   #!/bin/sh
2   export LD_LIBRARY_PATH=/usr/lib/oracle/12.1/client64/lib
3   sqlplus "[redacted]@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(Host=oracle.scs.ryerson.ca)(Port=1521))(CONNECT_DATA=(SID=orcl)))" <<EOF
4       DROP TABLE table_name;
5       exit;
6   EOF
7   echo "Table dropped"
```

Figure 2: Drop tables Code.

```
$ create_tables.sh
1   #!/bin/sh
2   export LD_LIBRARY_PATH=/usr/lib/oracle/12.1/client64/lib
3   sqlplus "[redacted]@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(Host=oracle.scs.ryerson.ca)(Port=1521))(CONNECT_DATA=(SID=orcl)))" <<EOF
4       CREATE TABLE table_name (
5           name VARCHAR2(12) PRIMARY KEY
6       );
7       exit;
8   EOF
9   echo "Table created"
```

Figure 3: Create Tables Code.

```
$ populate_tables.sh
#!/bin/sh
export LD_LIBRARY_PATH=/usr/lib/oracle/12.1/client64/lib
sqlplus "[redacted]@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(Host=oracle.scs.ryerson.ca)(Port=1521))(CONNECT_DATA=(SID=orcl)))" <<EOF
    INSERT INTO table_name VALUES ('name');
    exit;
EOF
echo "Table populated"
```

Figure 4: Populate tables Code.

```
$ queries.sh
#!/bin/sh
export LD_LIBRARY_PATH=/usr/lib/oracle/12.1/client64/lib
sqlplus "[redacted](DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(Host=oracle.scs.ryerson.ca)(Port=1521))(CONNECT_DATA=(SID=orcl)))" <<EOF
    SELECT ROLE, COUNT(USER_ID) AS COUNT
    FROM usertb
    GROUP BY ROLE;
    exit;
EOF
echo "Queries Performed"
```

Figure 5: Queries Code.

```
================================================================
|                  Oracle All Inclusive Tool                   |
|            Main Menu - Select Desired Operation(s):          |
|        <CTRL-Z Anytime to Enter Interactive CMD Prompt>       |
----------------------------------------------------------------
  M) View Manual

  1) Drop Tables
  2) Create Tables
  3) Populate Tables
  4) Query Tables

  X) Force/Stop/Kill Oracle DB

  E) End/Exit
Choose:
```

Figure 6: Main Menu GUI.

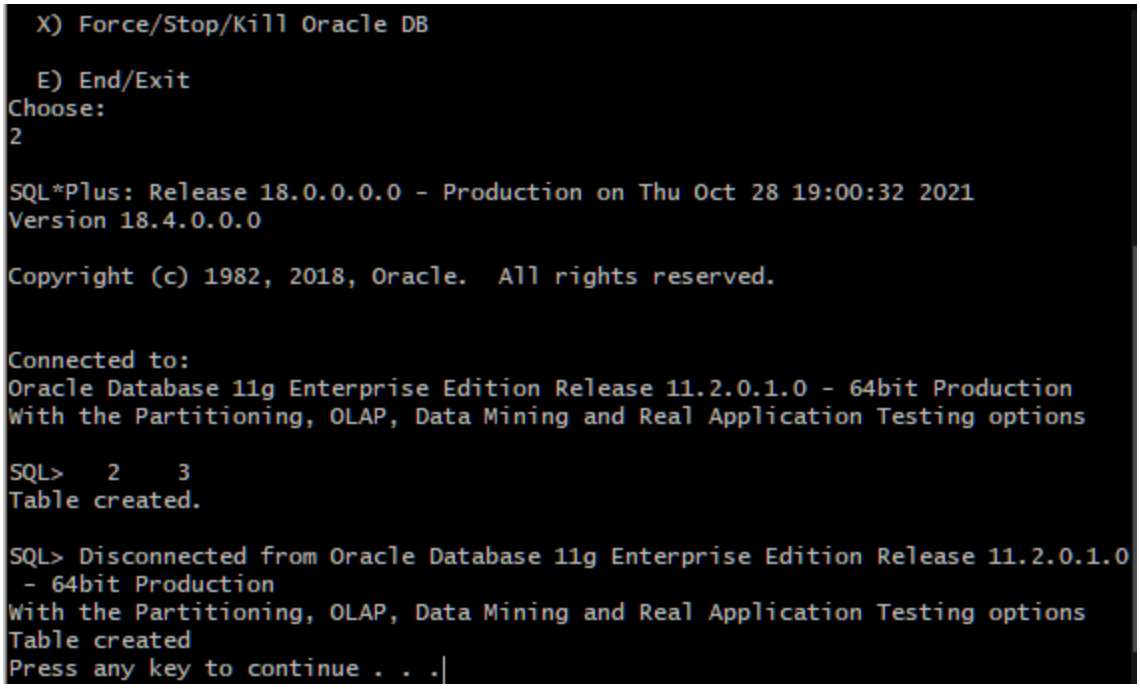


Figure 7: Results of create_tables.sh.

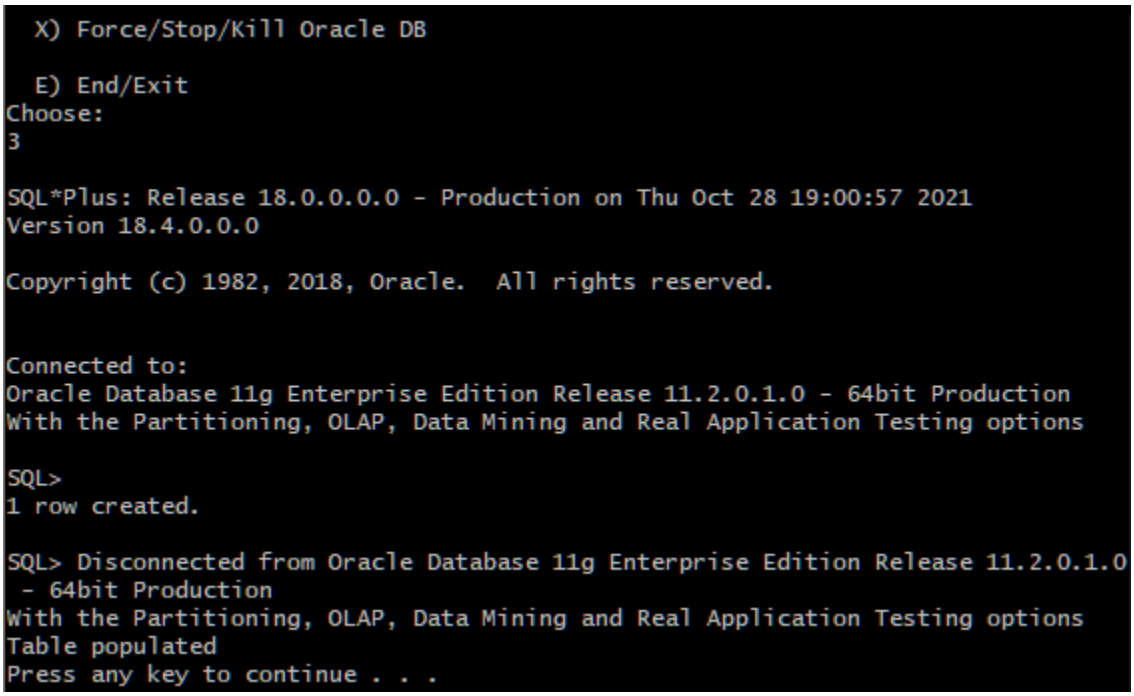


Figure 8: Results of populate_tables.sh.

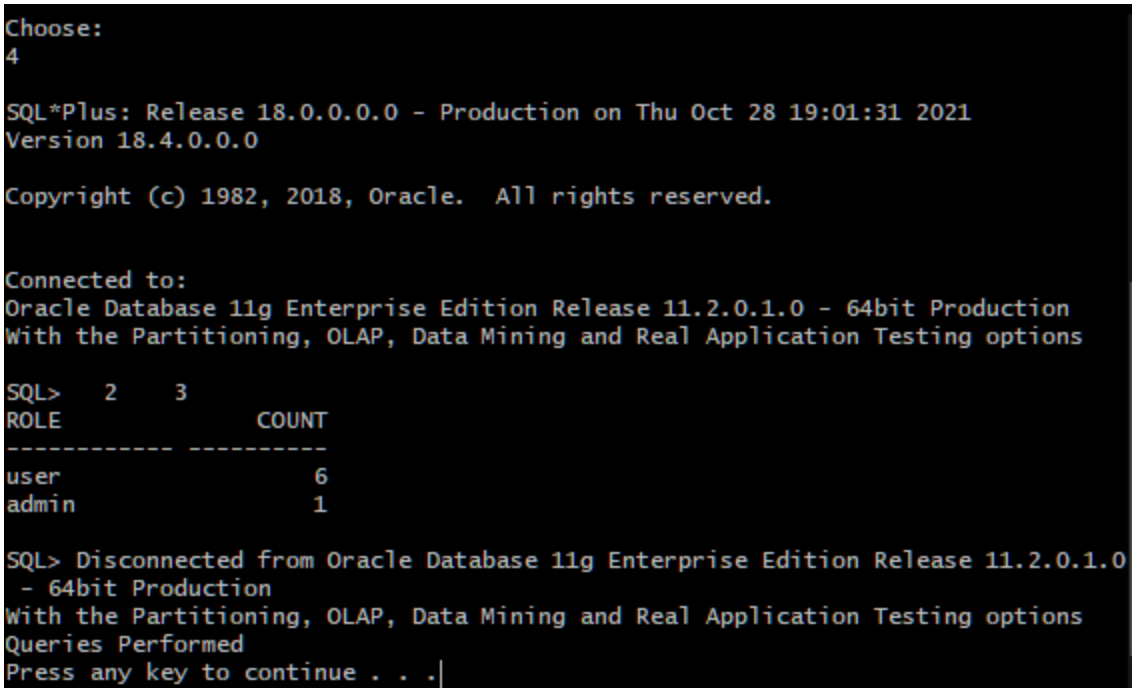


Figure 9: Results of queries.sh.

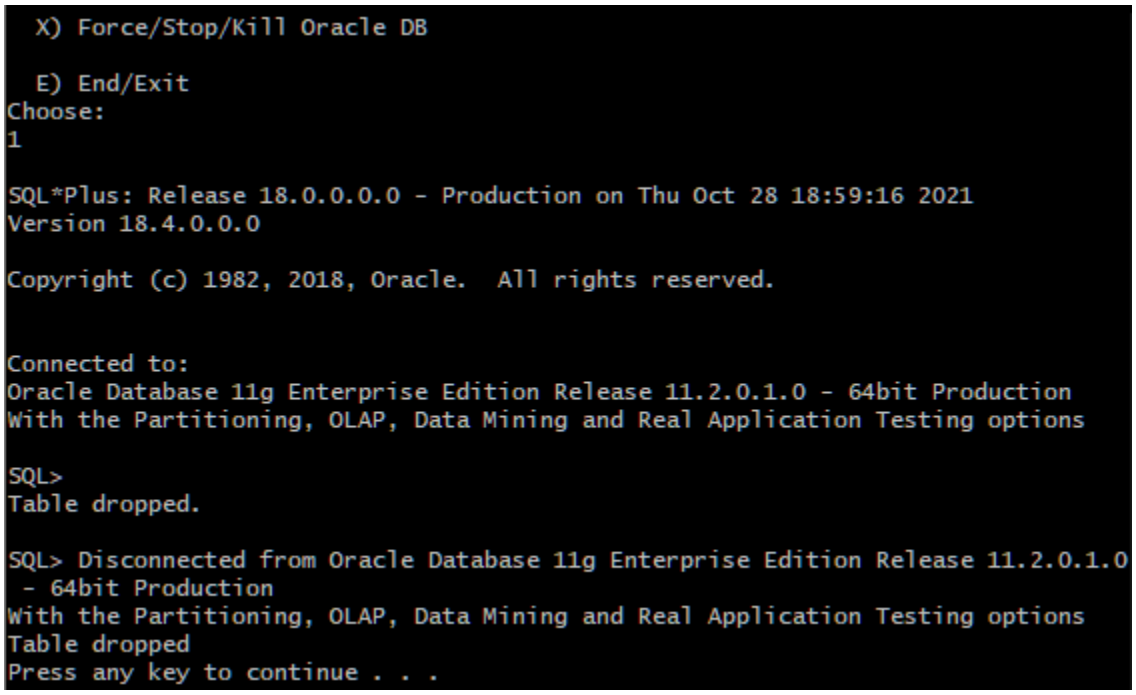


Figure 10: Results of drop_tables.sh.

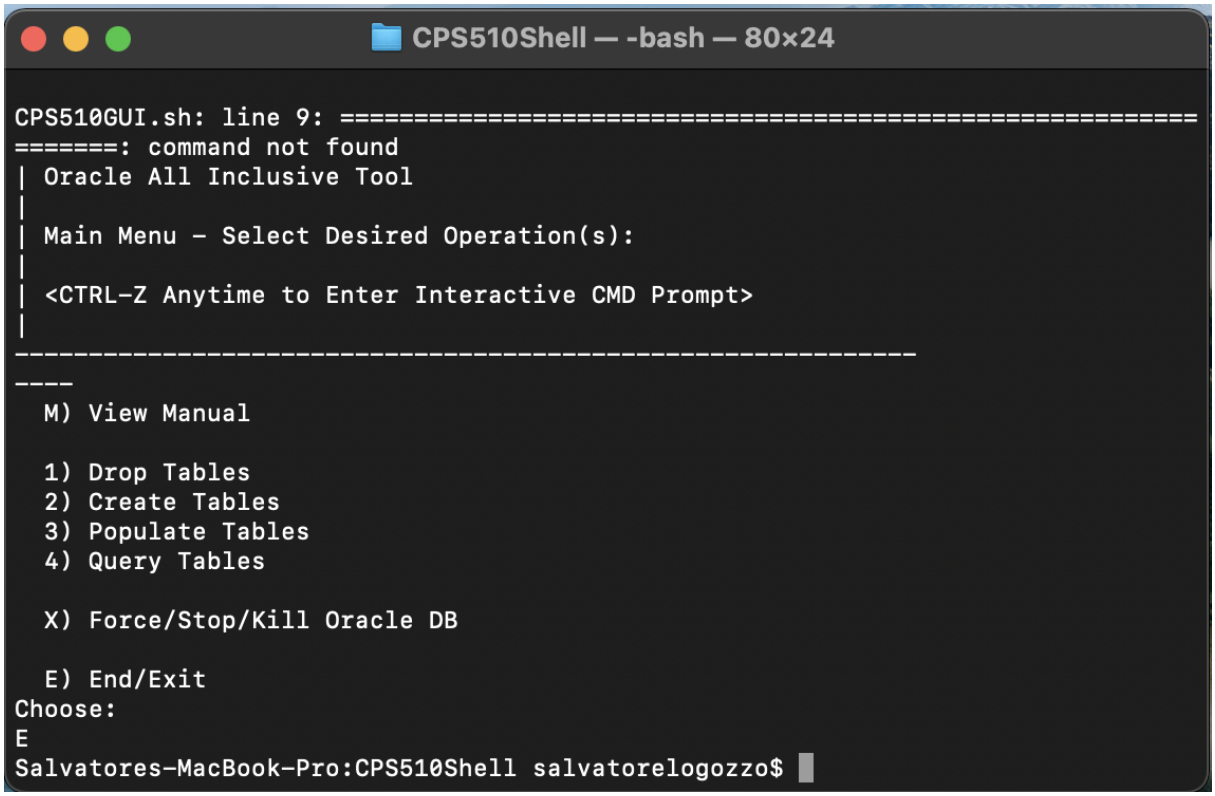


Figure 11: Results of E (Exit).

Functional Dependencies

Usertb (User_ID, First_Name, Last_Name, Email, Password, Role)

	USER_ID	FIRST_NAME	LAST_NAME	EMAIL	PASSWORD	ROLE
1	717	joseph	ct	plchiu@ryerson.ca	password2	user
2	6504	salvatore	logozzo	salvatore.logozzo@ryerson.ca	creativepassword	user
3	2033	eddy	tran	eddy.tran@ryerson.ca	abc123	user
4	0	admin	admin	(null)	abc123	admin
5	1234	friendA	lastA	friendA@friend.ca	friendpw	user
6	4321	friendB	lastB	friendA@friend.ca	friendpw	user
7	1001	friendC	lastC	friendA@friend.ca	friendpw	user

Functional Dependencies

User_ID → First_Name, Last_Name, Email, Password, Role

League (League_ID, League_Admin. Price_To_Join,Team_ID)

	LEAGUE_ID	LEAGUE_ADMIN	PRICE_TO_JOIN	TEAM_ID
1	1	6504	12	10
2	13	6504	25	1
3	12	6504	18	2
4	11	6504	45	3
5	9	6504	25	4
6	4	6504	12	5
7	2	6504	10	6
8	5	6504	25	7
9	10	6504	20	8
10	18	6504	15	9

Functional Dependencies

League_ID → Team_ID, League_Admin, Price_To_Join

Team (Team_Name, Team_ID, Position_In_Standing, League_Points, User_ID)

	TEAM_NAME	TEAM_ID	POSITION_IN_STANDING	LEAGUE_POINTS	USER_ID
1	Milan	1	(null)	0	6504
2	team2	2	(null)	0	2033
3	Roma	4	(null)	0	717
4	Atalanta	5	(null)	0	1234
5	Sassuolo	8	(null)	0	4321
6	Torino	9	(null)	0	1001
7	Juventus	3	(null)	0	(null)
8	Lazio	6	(null)	0	(null)
9	Fiorentina	7	(null)	0	(null)
10	Cagliari	10	(null)	0	(null)

Functional Dependencies

Team_ID → Team_Name, User_ID, Position_In_Standing, League_Points

Team_Name → Team_ID, User_ID, Position_In_Standing, League_Points

Lineup (Lineup_ID, Position, Name, Team_ID, Grade)

	LINEUP_ID	POSITION	NAME	TEAM_ID	GRADE
1	1	defender	De Ligt	(null)	0
2	2	defender	Koulibaly	(null)	0
3	3	defender	Handanovic	(null)	0
4	1	defender	Alex Sandrow	(null)	0
5	1	defender	Skriniar	(null)	0
6	1	defender	De Vrij	(null)	0
7	2	defender	Manolas	(null)	0
8	1	goalkeeper	Szczesny	1	0

Functional Dependencies

Lineup_ID, Name → Position, Team_ID, Grade

Goalkeeper (Name, Position, Team_ID, NumPenaltySaved, NumYellowCards, NumRedCards, NumPenaltyScored, NumPenaltyMissed, NumGoals, NumAssists)

	NAME	POSITION	TEAM_ID	NUMPENALTYSAVED	NUMYELLOWCARDS	NUMREDCARDS	NUMPENALTYSCORED	NUMPENALTYMISSED	NUMGOALS	NUMASSISTS
1	Szczesny	goalkeeper	1	3	0	0	0	0	0	0
2	Handanovic	goalkeeper	(null)	0	0	0	0	0	0	0
3	Meret	goalkeeper	(null)	4	0	0	0	0	0	0
4	Maignan	goalkeeper	(null)	0	0	0	0	0	0	0
5	Patricio	goalkeeper	(null)	0	0	0	0	0	0	0
6	Musso	goalkeeper	(null)	0	0	0	0	0	0	0
7	Strakosha	goalkeeper	(null)	0	0	0	0	0	0	0
8	Dragovski	goalkeeper	(null)	0	0	0	0	0	0	0
9	Consigli	goalkeeper	(null)	0	0	0	0	0	0	0
10	Vanja	goalkeeper	(null)	0	0	0	0	0	0	0
11	Cragno	goalkeeper	(null)	0	0	0	0	0	0	0
12	Skorupski	goalkeeper	(null)	0	0	0	0	0	0	0
13	Audero	goalkeeper	(null)	0	0	0	0	0	0	0
14	Montipo	goalkeeper	(null)	0	0	0	0	0	0	0
15	Silvestri	goalkeeper	(null)	0	0	0	0	0	0	0
16	Sirigu	goalkeeper	(null)	0	0	0	0	0	0	0
17	Vicario	goalkeeper	(null)	0	0	0	0	0	0	0
18	Zoet	goalkeeper	(null)	0	0	0	0	0	0	0

Functional Dependencies

Name → Position, Team_ID, NumPenaltySaved, NumYellowCards, NumRedCards, NumPenaltyScored, NumPenaltyMissed, NumGoals, NumAssists

Defender (Name, Position, Team_ID, NumYellowCards, NumRedCards, NumPenaltyScored, NumPenaltyMissed, NumGoals, NumAssists)

	NAME	POSITION	TEAM_ID	NUMYELLOWCARDS	NUMREDCARDS	NUMPENALTYSCORED	NUMPENALTYMISSED	NUMGOALS	NUMASSISTS
1	De Ligt	defender	(null)	0	0	0	0	0	0
2	Alex Sandro	defender	(null)	0	0	0	0	0	0
3	Skriniar	defender	(null)	0	0	0	0	0	0
4	De Vrij	defender	(null)	0	0	0	0	0	0
5	Koulibaly	defender	(null)	0	0	0	0	0	0
6	Manolas	defender	(null)	0	0	0	0	0	0
7	Kjaer	defender	(null)	0	0	0	0	0	0
8	Theo	defender	(null)	0	0	0	0	0	0
9	Mancini	defender	(null)	0	0	0	0	0	0
10	Smalling	defender	(null)	0	0	0	0	0	0
11	Toloi	defender	(null)	0	4	0	0	0	0
12	Gosens	defender	(null)	0	0	0	0	0	0
13	Acerbi	defender	(null)	0	0	0	0	0	0
14	Hysaj	defender	(null)	0	2	0	0	0	0
15	Milenkovic	defender	(null)	0	0	0	0	0	0
16	Biraghi	defender	(null)	0	0	0	0	0	0
17	Chiriches	defender	(null)	0	0	0	0	0	0
18	Ferrari	defender	(null)	0	0	0	0	0	0

Functional Dependencies

Name → Position, Team_ID, NumYellowCards, NumRedCards, NumPenaltyScored, NumPenaltyMissed, NumGoals, NumAssists

Midfielder (Name, Position, Team_ID, NumYellowCards, NumRedCards, NumPenaltyScored, NumPenaltyMissed, NumGoals, NumAssists)

	NAME	POSITION	TEAM_ID	NUMYELLOWCARDS	NUMREDCARDS	NUMPENALTYSCORED	NUMPENALTYMISSED	NUMGOALS	NUMASSISTS
1	Bentancur	midfielder	(null)	0	0	0	0	0	0
2	Locatelli	midfielder	(null)	0	0	0	0	0	0
3	Barella	midfielder	(null)	0	0	0	0	0	0
4	Vidal	midfielder	(null)	0	0	0	0	0	0
5	Lozano	midfielder	(null)	0	0	0	0	0	0
6	Insigne	midfielder	(null)	0	0	0	0	0	0
7	Kessie	midfielder	(null)	0	0	0	0	0	0
8	Maldini	midfielder	(null)	0	0	0	0	0	0
9	Cristante	midfielder	(null)	0	0	0	0	0	0
10	Veretout	midfielder	(null)	0	0	0	0	0	0
11	De Roon	midfielder	(null)	0	0	0	0	0	0
12	Pessina	midfielder	(null)	0	0	0	0	0	0
13	Luis Alberto	midfielder	(null)	0	0	0	0	0	0
14	Leiva	midfielder	(null)	0	0	0	0	0	0
15	Torreira	midfielder	(null)	0	0	0	0	0	0
16	Duncan	midfielder	(null)	0	0	0	0	0	0
17	Frattesi	midfielder	(null)	0	0	0	0	0	0
18	Djuricic	midfielder	(null)	0	0	0	0	0	0

Functional Dependencies

Name → Position, Team_ID, NumYellowCards, NumRedCards, NumPenaltyScored, NumPenaltyMissed, NumGoals, NumAssists

Striker (Name, Position, Team_ID, NumYellowCards, NumRedCards, NumPenaltyScored, NumPenaltyMissed, NumGoals, NumAssists)

	NAME	POSITION	TEAM_ID	NUMYELLOWCARDS	NUMREDCARDS	NUMPENALTYSCORED	NUMPENALTYMISSED	NUMGOALS	NUMASSISTS
1	Chiesa	striker	(null)	0	0	0	0	0	0
2	Dybala	striker	(null)	0	0	0	0	0	0
3	Lautaro	striker	(null)	0	0	0	0	0	0
4	Dzeko	striker	(null)	0	0	0	0	0	0
5	Politano	striker	(null)	0	0	0	0	0	0
6	Oshimen	striker	(null)	0	0	0	0	0	0
7	Leao	striker	(null)	0	0	0	0	0	0
8	Giroud	striker	(null)	0	0	0	0	0	0
9	Zaniolo	striker	(null)	0	0	0	0	0	0
10	Abraham	striker	(null)	0	0	0	0	0	0
11	Muriel	striker	(null)	0	0	0	0	0	0
12	Ilicic	striker	(null)	0	0	0	0	0	0
13	Immobile	striker	(null)	0	0	0	0	0	0
14	Anderson	striker	(null)	0	0	0	0	0	0
15	Callejon	striker	(null)	0	0	0	0	0	0
16	Vlahovic	striker	(null)	0	0	0	0	0	0
17	Berardi	striker	(null)	0	0	0	0	0	0
18	Scamacca	striker	(null)	0	0	0	0	0	0

Functional Dependencies

Name → Position, Team_ID, NumYellowCards, NumRedCards, NumPenaltyScored, NumPenaltyMissed, NumGoals, NumAssists

Match (Match_Day_Num, Team_Home, Team_Away, Game_Points_Home, Game_Points_Away, League_Points_Home, League_Points_Away)

	MATCH_DAY_NUM	TEAM_HOME	TEAM_AWAY	GAME_POINTS_HOME	GAME_POINTS_AWAY	LEAGUE_POINTS_HOME	LEAGUE_POINTS_AWAY
1	1	1	2	0	0	0	0
2	2	10	4	0	0	0	0
3	1	3	5	0	0	0	0
4	2	6	8	0	0	0	0

Functional Dependencies

Match_Day_Num, Team_Home, Team_Away → Game_Points_Home, Game_Points_Away, League_Points_Home, League_Points_Away

Standings (Team_ID, League_Points)

	TEAM_ID	LEAGUE_POINTS
1	1	2
2	2	1
3	4	2
4	5	0
5	8	1
6	9	1

Functional Dependencies

Team_ID → League_Points

Conversion to 3NF

User Table

User_ID → First_Name, Last_Name, Email, Password, Role

	USER_ID	FIRST_NAME	LAST_NAME	EMAIL	PASSWORD	ROLE
1	717	joseph	ct	plchiu@ryerson.ca	password2	user
2	6504	salvatore	logozzo	salvatore.logozzo@ryerson.ca	creativepassword	user
3	2033	eddy	tran	eddy.tran@ryerson.ca	abc123	user
4	0	admin	admin	(null)	abc123	admin
5	1234	friendA	lastA	friendA@friend.ca	friendpw	user
6	4321	friendB	lastB	friendA@friend.ca	friendpw	user
7	1001	friendC	lastC	friendA@friend.ca	friendpw	user

Decomposition(3NF): split transitivity

User_ID → First_Name, Last_Name, Email, Role

USER_ID	FIRST_NAME	LAST_NAME	EMAIL	ROLE
717	joseph	ct	plchiu@ryerson.ca	user
6504	salvatore	logozzo	salvatore.logozzo@ryerson.ca	user
2033	eddy	tran	eddy.tran@ryerson.ca	user
0	admin	admin	(null)	admin
1234	friendA	lastA	friendA@friend.ca	user
4321	friendB	lastB	friendA@friend.ca	user
1001	friendC	lastC	friendA@friend.ca	user

Email → Password

EMAIL	PASSWORD
plchiu@ryerson.ca	password2
salvatore.logozzo@ryerson.ca	creativepassword
eddy.tran@ryerson.ca	abc123
(null)	abc123
friendA@friend.ca	friendpw
friendA@friend.ca	friendpw
friendA@friend.ca	friendpw

League Table

League_ID → Team_ID, League_Admin, Price_To_Join

	LEAGUE_ID	LEAGUE_ADMIN	PRICE_TO_JOIN	TEAM_ID
1	1	6504	12	10
2	13	6504	25	1
3	12	6504	18	2
4	11	6504	45	3
5	9	6504	25	4
6	4	6504	12	5
7	2	6504	10	6
8	5	6504	25	7
9	10	6504	20	8
10	18	6504	15	9

This relation is already in 3NF.

Team Table

Team_ID → Team_Name, User_ID, Position_In_Standing, League_Points

	TEAM_NAME	TEAM_ID	POSITION_IN_STANDING	LEAGUE_POINTS	USER_ID
1	Milan	1	(null)	0	6504
2	team2	2	(null)	0	2033
3	Roma	4	(null)	0	717
4	Atalanta	5	(null)	0	1234
5	Sassuolo	8	(null)	0	4321
6	Torino	9	(null)	0	1001
7	Juventus	3	(null)	0	(null)
8	Lazio	6	(null)	0	(null)
9	Fiorentina	7	(null)	0	(null)
10	Cagliari	10	(null)	0	(null)

This relation is already in 3NF.

Lineup Table

Lineup_ID, Name → Position, Team_ID, Grade

	LINEUP_ID	POSITION	NAME	TEAM_ID	GRADE
1	1	defender	De Ligt	(null)	0
2	2	defender	Koulibaly	(null)	0
3	3	defender	Handanovic	(null)	0
4	1	defender	Alex Sandrow	(null)	0
5	1	defender	Skriniar	(null)	0
6	1	defender	De Vrij	(null)	0
7	2	defender	Manolas	(null)	0
8	1	goalkeeper	Szczesny	1	0

Decomposition(2NF): split partial dependency (becomes 3NF as well)
Lineup_ID, Name → Position, Grade

LINEUP_ID	NAME	POSITION	GRADE
1	De Ligt	defender	0
2	Koulibaly	defender	0
3	Handanovic	defender	0
1	Alex Sandrow	defender	0
1	Skriniar	defender	0
1	De Vrij	defender	0
2	Manolas	defender	0
1	Szczesny	goalkeeper	0

Lineup_ID → Team_ID

LINEUP_ID	TEAM_ID
1	(null)
2	(null)
3	(null)
1	(null)
1	(null)
1	(null)
2	(null)
1	1

Goalkeeper Table

Name → Position, Team_ID, NumPenaltySaved, NumYellowCards, NumRedCards, NumPenaltyScored, NumPenaltyMissed, NumGoals, NumAssists

NAME	POSITION	TEAM_ID	NUMPENALTYSAVED	NUMYELLOWCARDS	NUMREDCARDS	NUMPENALTYSCORED	NUMPENALTYMISSED	NUMGOALS	NUMASSISTS
Szczesny	goalkeeper	1	3	0	0	0	0	0	0
Handanovic	goalkeeper	(null)	0	0	0	0	0	0	0
Meret	goalkeeper	(null)	4	0	0	0	0	0	0
Maignan	goalkeeper	(null)	0	0	0	0	0	0	0
Patricio	goalkeeper	(null)	0	0	0	0	0	0	0
Musso	goalkeeper	(null)	0	0	0	0	0	0	0
Strakosha	goalkeeper	(null)	0	0	0	0	0	0	0
Dragovski	goalkeeper	(null)	0	0	0	0	0	0	0
Consigli	goalkeeper	(null)	0	0	0	0	0	0	0
Vanja	goalkeeper	(null)	0	0	0	0	0	0	0

This relation is already in 3NF.

Striker, Midfielder, Defender Tables

Name → Position, Team_ID, NumYellowCards, NumRedCards, NumPenaltyScored, NumPenaltyMissed, NumGoals, NumAssists

NAME	POSITION	TEAM_ID	NUMYELLOWCARDS	NUMREDCARDS	NUMPENALTYSCORED	NUMPENALTYMISSED	NUMGOALS	NUMASSISTS
De Ligt	defender	(null)	0	0	0	0	0	0
Alex Sandro	defender	(null)	0	0	0	0	0	0
Skriniar	defender	(null)	0	0	0	0	0	0
De Vrij	defender	(null)	0	0	0	0	0	0
Koulibaly	defender	(null)	0	0	0	0	0	0
Manolas	defender	(null)	0	0	0	0	0	0
Kjaer	defender	(null)	0	0	0	0	0	0
Theo	defender	(null)	0	0	0	0	0	0
Mancini	defender	(null)	0	0	0	0	0	0
Smalling	defender	(null)	0	0	0	0	0	0
Toloi	defender	(null)	0	4	0	0	0	0

Ex. Defender Table

This relation is already in 3NF.

Match Table

Match_Day_Num, Team_Home, Team_Away → Game_Points_Home, Game_Points_Away, League_Points_Home, League_Points_Away

	MATCH_DAY_NUM	TEAM_HOME	TEAM_AWAY	GAME_POINTS_HOME	GAME_POINTS_AWAY	LEAGUE_POINTS_HOME	LEAGUE_POINTS_AWAY
1	1	1	2	0	0	0	0
2	2	10	4	0	0	0	0
3	1	3	5	0	0	0	0
4	2	6	8	0	0	0	0

Decomposition(2NF): split partial dependency

Match_Day_Num, Team_Home, Team_Away → Game_Points_Home, Game_Points_Away

	MATCH_DAY_NUM	TEAM_HOME	TEAM_AWAY	GAME_POINTS_HOME	GAME_POINTS_AWAY
1	1	1	2	0	0
2	2	10	4	0	0
3	1	3	5	0	0
4	2	6	8	0	0

Team_Home → League_Points_Home

TEAM_HOME	LEAGUE_POINTS_HOME
1	0
10	0
3	0
6	0

Team_Away → League_Points_Away

TEAM_AWAY	LEAGUE_POINTS_AWAY
2	0
4	0
5	0
8	0

Standings Table

Team_ID → League_Points

TEAM_ID	LEAGUE_POINTS
1	2
2	1
4	2
5	0
8	1
9	1

This relation is already in 3NF.

Conversion to BCNF

User Table

Combined FDs with the same LHS and reduced to have one attribute on the LHS.
User_ID → First_Name, Last_Name, Email, Role

USER_ID	FIRST_NAME	LAST_NAME	EMAIL	ROLE
717	joseph	ct	plchiu@ryerson.ca	user
6504	salvatore	logozzo	salvatore.logozzo@ryerson.ca	user
2033	eddy	tran	eddy.tran@ryerson.ca	user
0	admin	admin	(null)	admin
1234	friendA	lastA	friendA@friend.ca	user
4321	friendB	lastB	friendA@friend.ca	user
1001	friendC	lastC	friendA@friend.ca	user

Password is dependent on Email and not User_ID.
Email → Password

EMAIL	PASSWORD
plchiu@ryerson.ca	password2
salvatore.logozzo@ryerson.ca	creativepassword
eddy.tran@ryerson.ca	abc123
(null)	abc123
friendA@friend.ca	friendpw
friendA@friend.ca	friendpw
friendA@friend.ca	friendpw

If joined back, there is a lossless join. User_ID is not dependent on any attribute. Therefore, this table is in BCNF
Compute User_ID+
= {User_ID}
= {User_ID, First_Name, Last_Name,Email, Role}
= {User_ID, First_Name, Last_Name,Email, Password, Role}

League Table

League_ID → Team_ID, League_Admin, Price_To_Join

	LEAGUE_ID	LEAGUE_ADMIN	PRICE_TO_JOIN	TEAM_ID
1	1	6504	12	10
2	13	6504	25	1
3	12	6504	18	2
4	11	6504	45	3
5	9	6504	25	4
6	4	6504	12	5
7	2	6504	10	6
8	5	6504	25	7
9	10	6504	20	8
10	18	6504	15	9

League_ID is independent of the other attributes. Multiple Team_IDs can have the same League_ID. Since the superkey (League_ID) is independent and all attributes depend on the super key, this relation is already in BCNF.

Team Table

Team_ID → Team_Name, User_ID, Position_In_Standing, League_Points

	TEAM_NAME	TEAM_ID	POSITION_IN_STANDING	LEAGUE_POINTS	USER_ID
1	Milan	1	(null)	0	6504
2	team2	2	(null)	0	2033
3	Roma	4	(null)	0	717
4	Atalanta	5	(null)	0	1234
5	Sassuolo	8	(null)	0	4321
6	Torino	9	(null)	0	1001
7	Juventus	3	(null)	0	(null)
8	Lazio	6	(null)	0	(null)
9	Fiorentina	7	(null)	0	(null)
10	Cagliari	10	(null)	0	(null)

Team_Name and Team_ID are both primary attributes since they can determine all attributes in the Team table. These two attributes determine the rest of the attributes with no transitivity. So this relation is already in BCNF.

Lineup Table

Lineup_ID, Name → Position, Grade

LINEUP_ID	NAME	POSITION	GRADE
1	De Ligt	defender	0
2	Koulibaly	defender	0
3	Handanovic	defender	0
1	Alex Sandrow	defender	0
1	Skriniar	defender	0
1	De Vrij	defender	0
2	Manolas	defender	0
1	Szczesny	goalkeeper	0

Lineup_ID → Team_ID

LINEUP_ID	TEAM_ID
1	(null)
2	(null)
3	(null)
1	(null)
1	(null)
1	(null)
2	(null)
1	1

This relation is already in BCNF.

Compute LineUp,Name+
= {LineUp, Name}
= {LineUp, Name, Team_ID}
= {LineUp, Name, Team_ID, Position, Grade}

Goalkeeper Table

Name → Position, Team_ID, NumPenaltySaved, NumYellowCards, NumRedCards, NumPenaltyScored, NumPenaltyMissed, NumGoals, NumAssists

NAME	POSITION	TEAM_ID	NUMPENALTYSAVED	NUMYELLOWCARDS	NUMREDCARDS	NUMPENALTYSCORED	NUMPENALTYMISSED	NUMGOALS	NUMASSISTS
Szczesny	goalkeeper	1	3	0	0	0	0	0	0
Handanovic	goalkeeper	(null)	0	0	0	0	0	0	0
Meret	goalkeeper	(null)	4	0	0	0	0	0	0
Maignan	goalkeeper	(null)	0	0	0	0	0	0	0
Patricio	goalkeeper	(null)	0	0	0	0	0	0	0
Musso	goalkeeper	(null)	0	0	0	0	0	0	0
Strakosha	goalkeeper	(null)	0	0	0	0	0	0	0
Dragovski	goalkeeper	(null)	0	0	0	0	0	0	0
Consigli	goalkeeper	(null)	0	0	0	0	0	0	0
Vanja	goalkeeper	(null)	0	0	0	0	0	0	0

Name is an independent attribute that determines all attributes of the table, so this relation is already in BCNF.

Striker, Midfielder, Defender Tables

Name → Position, Team_ID, NumYellowCards, NumRedCards, NumPenaltyScored, NumPenaltyMissed, NumGoals, NumAssists

NAME	POSITION	TEAM_ID	NUMYELLOWCARDS	NUMREDCARDS	NUMPENALTYSCORED	NUMPENALTYMISSED	NUMGOALS	NUMASSISTS
De Ligt	defender	(null)	0	0	0	0	0	0
Alex Sandro	defender	(null)	0	0	0	0	0	0
Skriniar	defender	(null)	0	0	0	0	0	0
De Vrij	defender	(null)	0	0	0	0	0	0
Koulibaly	defender	(null)	0	0	0	0	0	0
Manolas	defender	(null)	0	0	0	0	0	0
Kjaer	defender	(null)	0	0	0	0	0	0
Theo	defender	(null)	0	0	0	0	0	0
Mancini	defender	(null)	0	0	0	0	0	0
Smalling	defender	(null)	0	0	0	0	0	0
Toloi	defender	(null)	0	4	0	0	0	0

Ex. Defender Table

Name is an independent attribute that determines all attributes of the table, so this relation is already in BCNF.

Match Table

Match_Day_Num, Team_Home, Team_Away → Game_Points_Home, Game_Points_Away

	MATCH_DAY_NUM	TEAM_HOME	TEAM_AWAY	GAME_POINTS_HOME	GAME_POINTS_AWAY
1	1	1	2	0	0
2	2	10	4	0	0
3	1	3	5	0	0
4	2	6	8	0	0

Team_Home → League_Points_Home

TEAM_HOME	LEAGUE_POINTS_HOME
1	0
10	0
3	0
6	0

Team_Away → League_Points_Away

TEAM_AWAY	LEAGUE_POINTS_AWAY
2	0
4	0
5	0
8	0

Match_Day, Team_Home, and Team_Away are keys that determine the rest of the attributes. Team_Home and Team_Away have a decomposition table to determine the league_points of the respective teams. None of the other attributes have the keys dependent on them. So this relation is in BCNF.

Compute Match_Day_Num, Team_Home, Team_Away+
Match_Day_Num+ = {Match_Day_Num}
Team_Home+ = {Team_Home, League_Points_Home}
Team_Away+ = {Team_Away, League_Points_Away}
Match_Day_Num, Team_Home+ = {Match_Day_Num, Team_Home, Game_Points_Home}
Match_Day_Num, Team_Away+ = {Match_Day_Num, Team_Away, Game_Points_Away}

Match_Day_Num, Team_Home, Team_Away+ = {Game_Points_Home, League_Points_Home, Game_Points_Away, League_Points_Away}

Standings Table

Team_ID → League_Points

TEAM_ID	LEAGUE_POINTS
1	2
2	1
4	2
5	0
8	1
9	1

This relation is already in BCNF.

Python GUI

-- Fantasy Soccer DBMS
-- Created by:
-- Salvatore Logozzo - ****56504
-- Phuong Nam Chiu Tran - ****40717
-- Eddy Tran - ****82033
-- Section 8

This document is intended to show sample screenshots of the GUI created using Python and the two modules ‘tkinter’ and ‘cx_Oracle’.
Tkinter is the module used to create the GUI items, while cx_Oracle is the module used to connect the python program to our OracleSQL DBMS.

In the following screenshots, there will be a demo of the program’s functionalities, such as:

1. Logging into the DBMS using personal credentials
2. Creating a demo table
3. Showing the alter table screen (used to insert values)
4. Inserting dummy values into the demo table
5. Querying a table from our Fantasy Soccer DBMS (Query results are shown in the GUI itself.)
6. Dropping the demo table from the DBMS

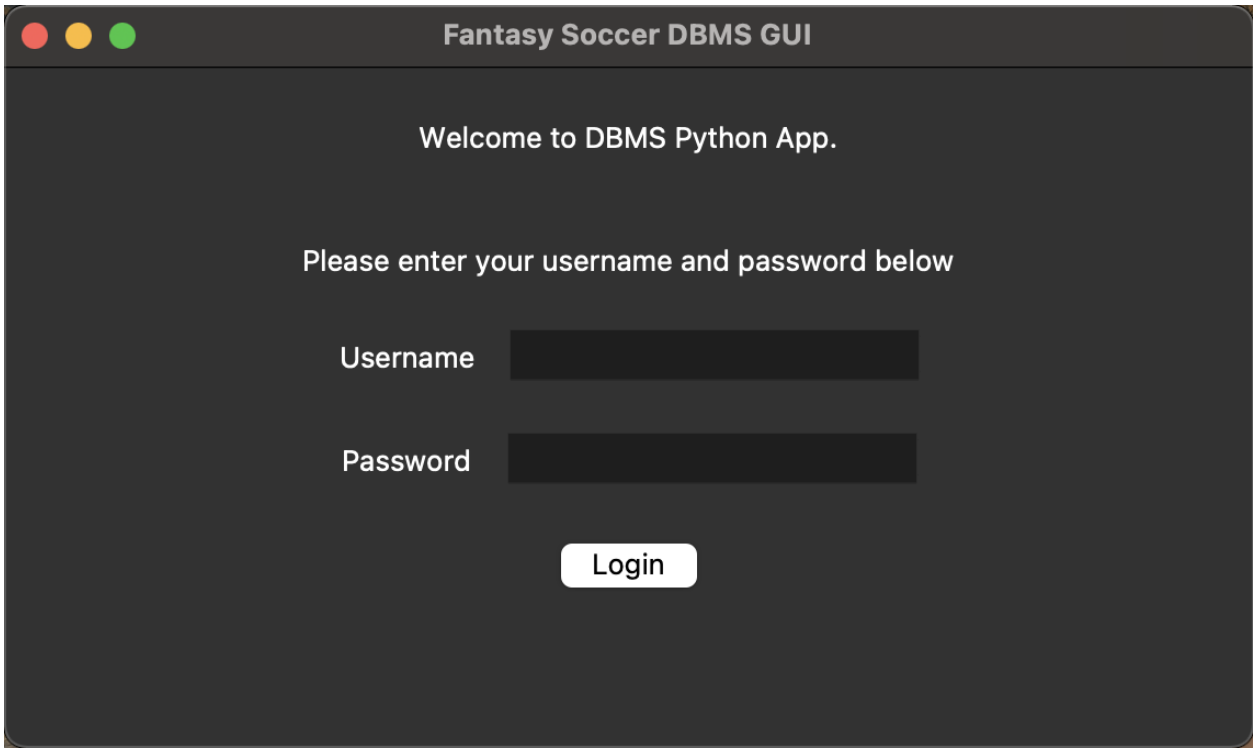


Image 1: Login Screen.

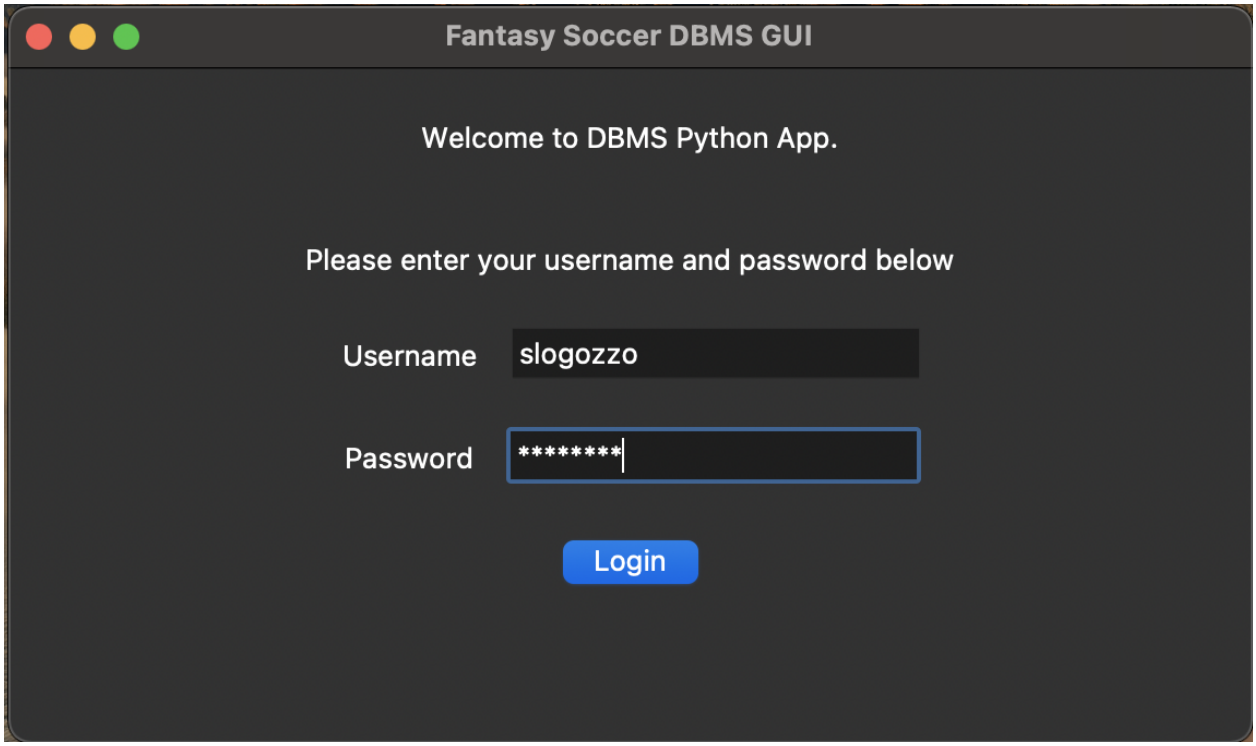


Image 2: Credentials typed and Login button pressed.

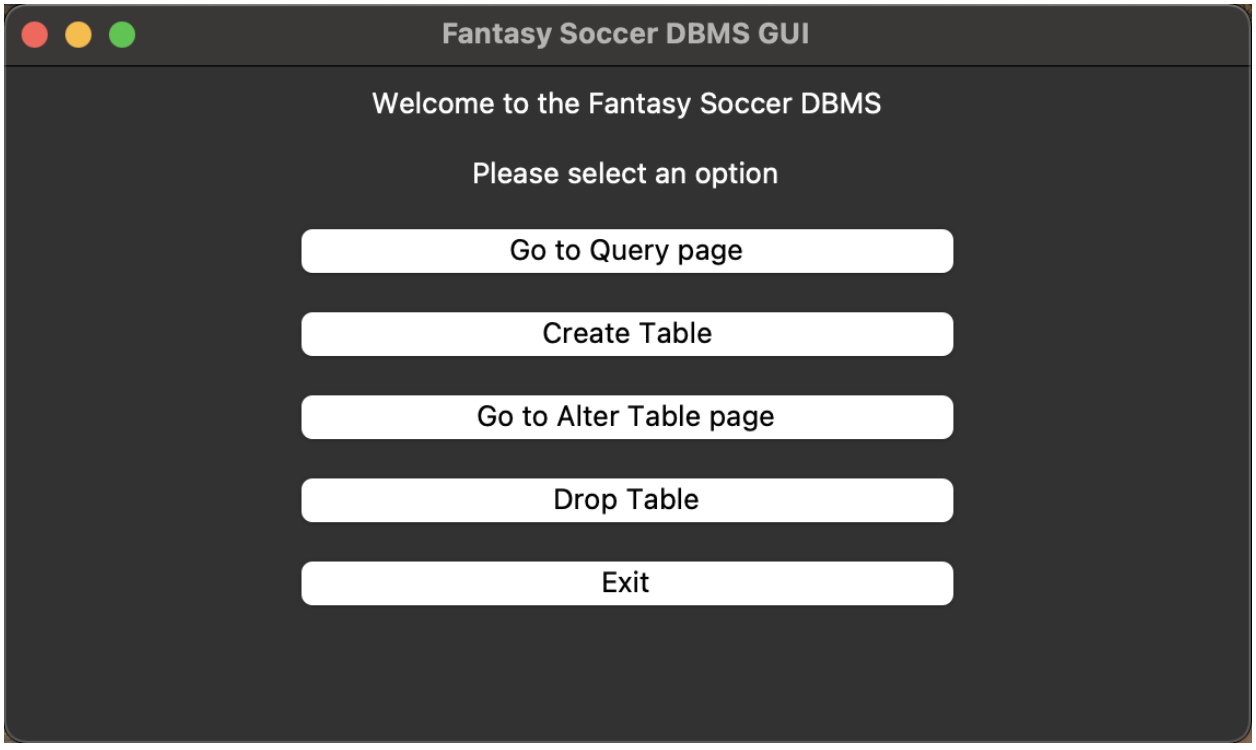


Image 3: Main window, containing five buttons: one that opens the query page, one that creates a demo table, one that opens the alter table page, one that drops the previously created demo table, and finally one that closes the connection and exits the program.

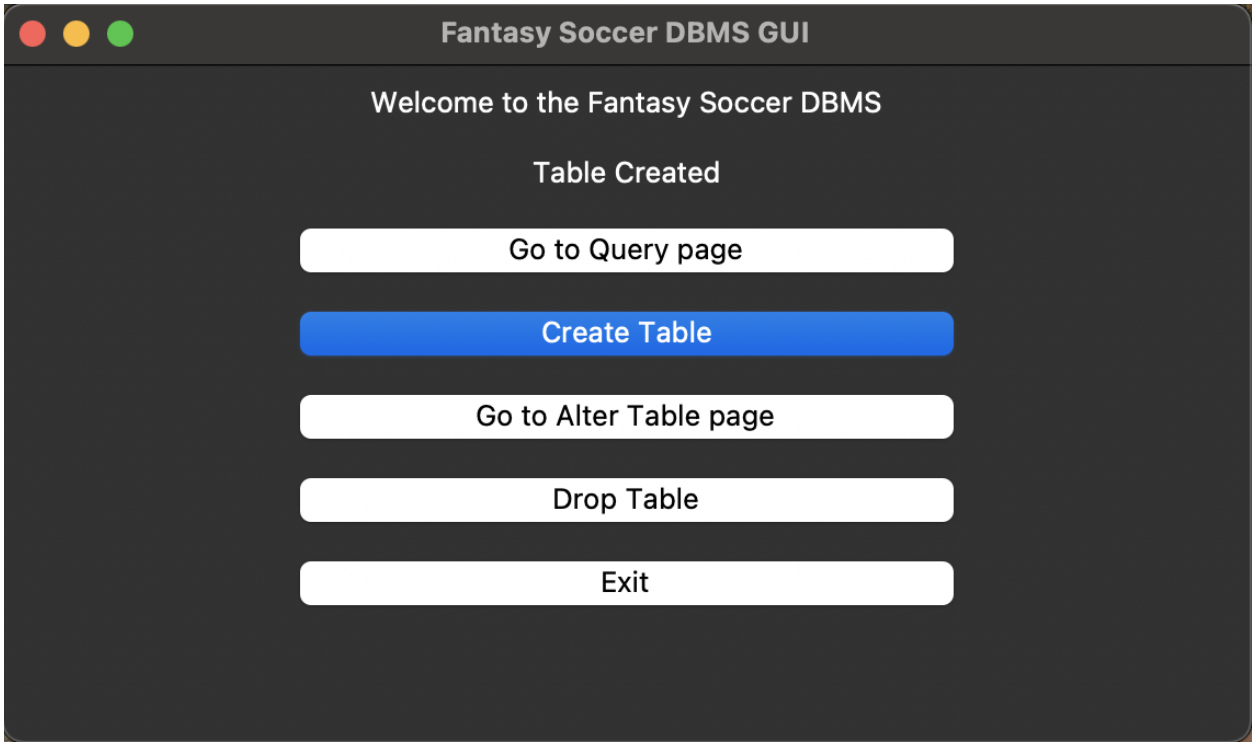


Image 4: Sample table created.

Image 5: Alter Table Page.

Image 6: Data inserted into sample table.

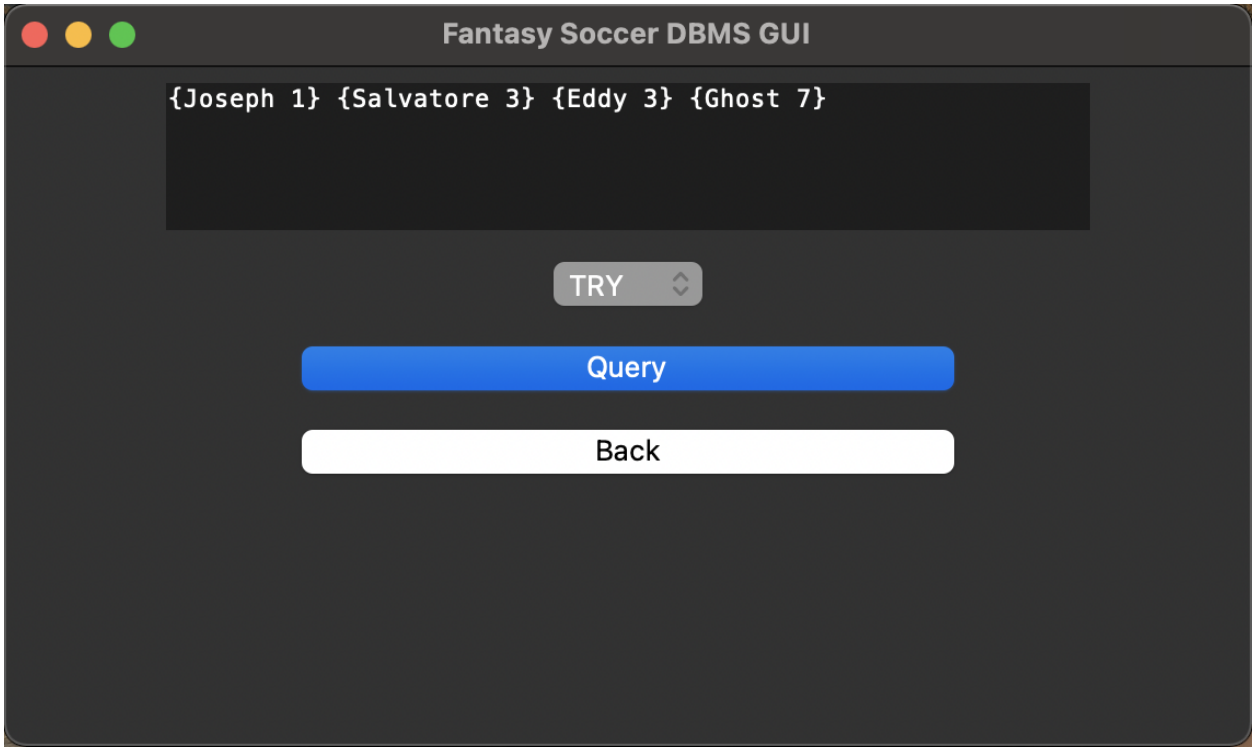


Image 7: Query performed on DBMS using the GUI.

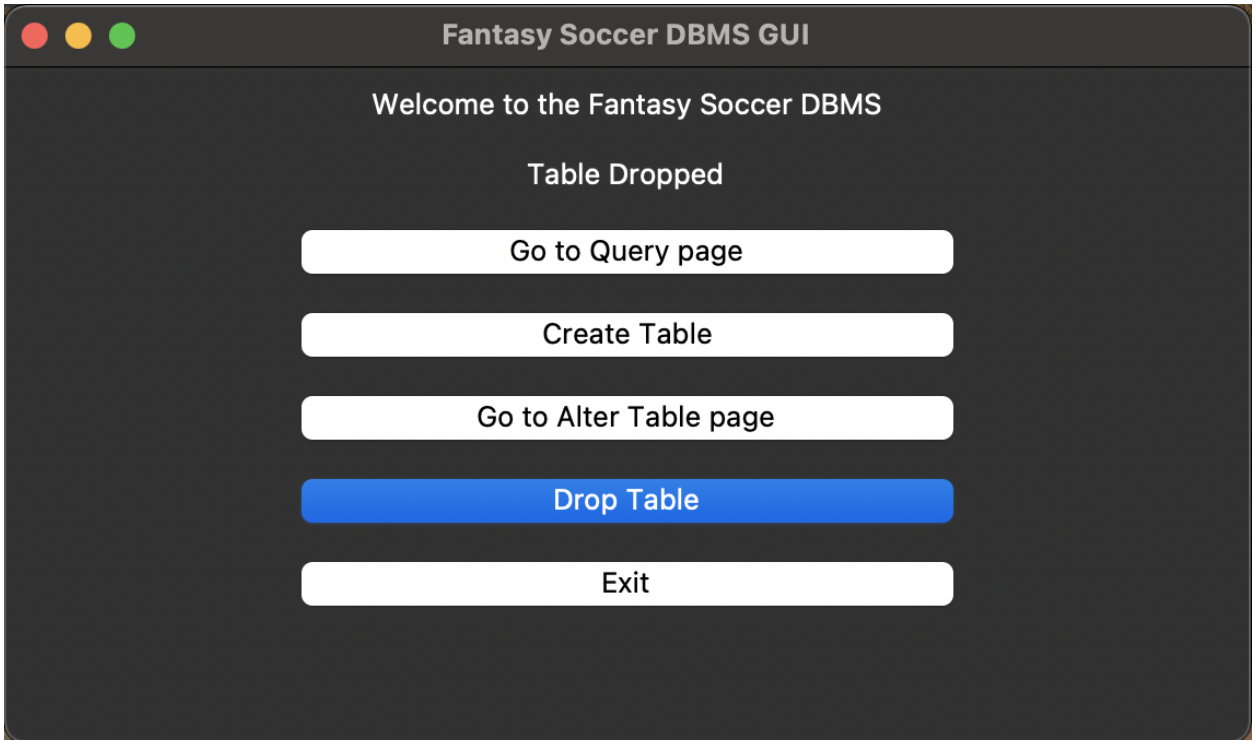


Image 8: Sample table dropped.

Final remarks:

- For the purpose of this demonstration, not everything was made dynamic due to time constraints. However, the key points were covered, such as:
    a. Options to query, create table, populate table, drop table, and exit are implemented and working as expected.
    b. UI is connected to the DBMS created during this course.
    c. The application can read or write to the DBMS based on the button the user clicks.
    d. The application was made with a GUI in python, and all the previously listed functionalities work.
    e. The one thing missing was the possibility to access and update records in each table like a complete DBMS application (More details below)
        ➔ For this demonstration, the create table button was static (it can only create one preset table hard-coded in python, but with a bit more time, it could be possible to create a separate screen to use textfields and buttons to choose a table name, variable name and variable types to make create table dynamic.
        ➔ Same goes for the drop table, we made it hard-coded to make the demo easier, but it is possible to implement a separate screen with a dropdown containing all the tables and a button to drop the one selected from the dropdown using the command “‘DROP TABLE ‘ + dropdown.get()”

- ➔ Another upgrade that could be done, similarly to create table, is alter table, where instead of just choosing the table to alter, we give the user a choice with textfields with the appropriate variables to add.
- ➔ Lastly, the query page is dynamic, but the user can only choose which table to query. One upgrade that could be done is to let the user also choose the columns to query or to use more advanced queries, like UNION or JOIN, etc.