

Database Project



Table of Contents

Introduction.....	2
Business rules.....	2
ER Diagram	2
Data Dictionary.....	3
Normalization and Dependency Analysis.....	4
Implementation	7
References.....	16



Introduction

The aim of the report is to present information about the implementation of the table tennis matches database for the VU University. The document contains the details of the conceptual design, logical design, normalization and implementation details.

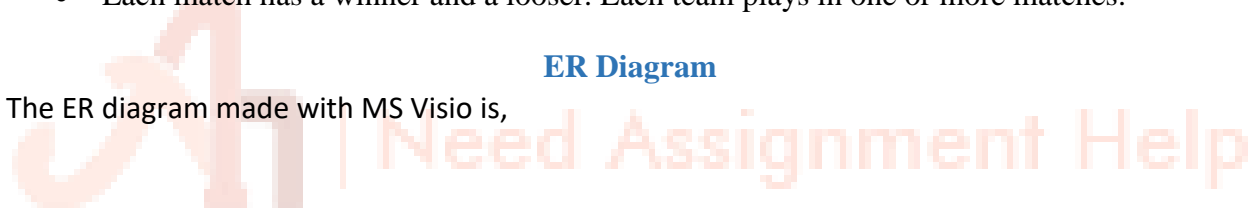
Business rules

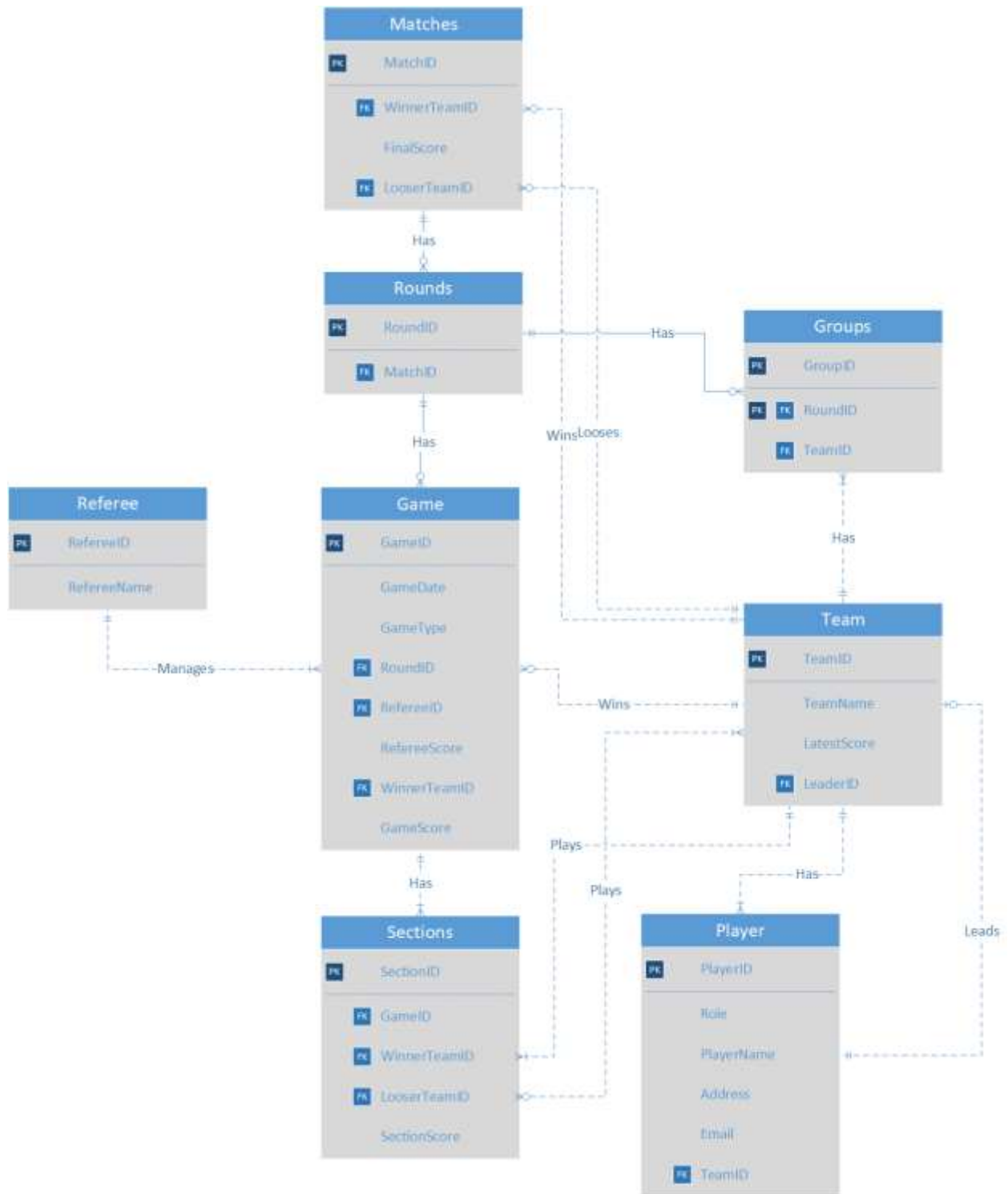
Following business rules have been identified from the given case study.

- A match has multiple rounds. Each round is in a match.
- A round has multiple games. Each game is played in a round.
- A game has sections. Each section belongs to a game.
- A referee manages one or more games. Each game is managed by a referee.
- A player plays in a team. Each team has one or more players.
- Two teams play in a section. Team plays in one or more sections.
- A player may lead no or one team. Each team is led by a player.
- Each group has two or more teams in a round. Each team plays in multiple groups in multiple rounds. Each round has one or more groups.
- Each match has a winner and a loser. Each team plays in one or more matches.

ER Diagram

The ER diagram made with MS Visio is,





Data Dictionary

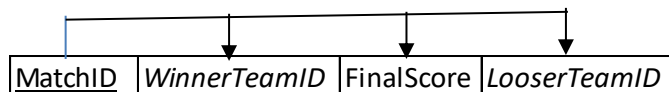
The data dictionary is,

Table	Attribute	Datatype	Key
Matches	MatchID	Integer	Primary key

	WinnerTeamID	Integer	foreign key
	FinalScore	Integer	
	LooserTeamID	Integer	foreign key
Rounds	RoundID	Integer	Primary key
	MatchID	Integer	foreign key
Game	GameID	Integer	Primary key
	GameDate	Date	
	GameType	Varchar	
	RoundID	Integer	foreign key
	RefereeID	Integer	foreign key
	RefereeScore	Integer	
	WinnerTeamID	Integer	foreign key
	GameScore	Integer	
Referee	RefereeID	Integer	Primary key
	RefereeName	Varchar	
Sections	SectionID	Integer	Primary key
	GameID	Integer	foreign key
	WinnerTeamID	Integer	foreign key
	LooserTeamID	Integer	foreign key
	SectionScore	Integer	
Player	PlayerID	Integer	Primary key
	Role	Varchar	
	PlayerName	Varchar	
	Address	Varchar	
	Email	Varchar	
	TeamID	Integer	foreign key
Team	TeamID	Integer	Primary key
	TeamName	Varchar	
	LatestScore	Integer	
	LeaderID	Integer	foreign key
Groups	GroupID	Integer	Primary key
	RoundID	Integer	Primary key, foreign key
	TeamID	Integer	foreign key

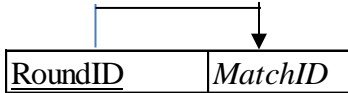
Normalization and Dependency Analysis

- Matches (MatchID, WinnerTeamID, FinalScore, LooserTeamID)



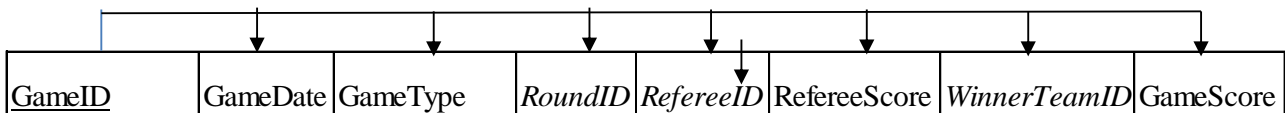
There is no multivalued attribute and there is the primary key. Hence, it is in 1NF. Then there is no partial dependency on the primary key. So, it is in 2NF. And there is no transitive dependency. So, it is in 3NF (Coronel & Morris, 2016).

- Rounds (RoundID, MatchID)



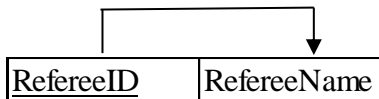
There is no multivalued attribute and there is the primary key. Hence, it is in 1NF. Then there is no partial dependency on the primary key. So, it is in 2NF. And there is no transitive dependency. So, it is in 3NF.

- Game (GameID, GameDate, GameType, RoundID, RefereeID, RefereeScore, WinnerTeamID, GameScore)



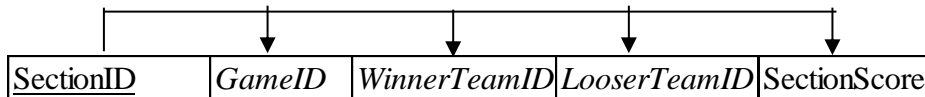
There is no multivalued attribute and there is the primary key. Hence, it is in 1NF. Then there is no partial dependency on the primary key. So, it is in 2NF. And there is no transitive dependency. So, it is in 3NF.

- Referee (RefereeID, RefereeName)



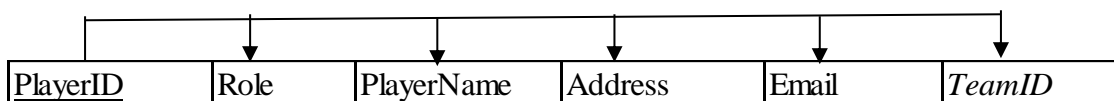
There is no multivalued attribute and there is the primary key. Hence, it is in 1NF. Then there is no partial dependency on the primary key. So, it is in 2NF. And there is no transitive dependency. So, it is in 3NF.

- Sections (SectionID, GameID, WinnerTeamID, LoserTeamID, SectionScore)



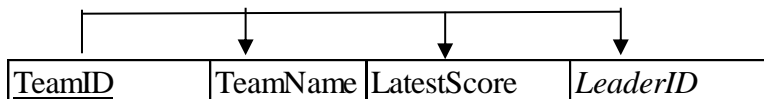
There is no multivalued attribute and there is the primary key. Hence, it is in 1NF. Then there is no partial dependency on the primary key. So, it is in 2NF. And there is no transitive dependency. So, it is in 3NF.

- Player (PlayerID, Role, PlayerName, Address, Email, TeamID)



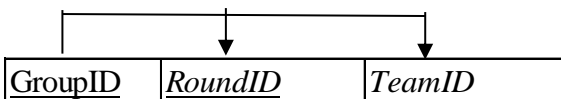
There is no multivalued attribute and there is the primary key. Hence, it is in 1NF. Then there is no partial dependency on the primary key. So, it is in 2NF. And there is no transitive dependency. So, it is in 3NF.

- Team (TeamID, TeamName, LatestScore, LeaderID)



There is no multivalued attribute and there is the primary key. Hence, it is in 1NF. Then there is no partial dependency on the primary key. So, it is in 2NF. And there is no transitive dependency. So, it is in 3NF.

- Groups (GroupID, RoundID, TeamID)



There is no multivalued attribute and there is the primary key. Hence, it is in 1NF. Then there is no partial dependency on the primary key. So, it is in 2NF. And there is no transitive dependency. So, it is in 3NF.

Implementation

SQL code to build the database,

```
create database ttl;
use ttl;
create table matches(
    matchID int not null,
    finalscore int not null,
    winnerteamID int not null,
    looserteamID int not null,
    primary key(matchID),
    foreign key(winnerteamID) references team(teamID),
    foreign key(looserteamID) references team(teamID)
);

insert into matches values (1, 100, 1, 10), (2, 110, 2, 9), (3,
100, 4, 5), (4, 105, 1, 9), (5, 90, 1, 10), (6, 80, 9, 8), (7, 103,
9, 1), (8, 101, 10, 5), (9, 120, 6, 3), (10, 50, 4, 1);

create table rounds(
    roundID int not null,
    matchID int not null,
    primary key(roundID),
    foreign key(matchID) references matches(matchID)
);

insert into rounds values
(1,1), (2,1), (3,1), (4,1), (5,2), (6,2), (7,2), (8,2), (9,3), (10,3);

create table game(
    gameID int not null,
    gamedate date not null,
    gametype varchar(10) not null,
    roundID int not null,
    refereeID int not null,
    refereescore int not null,
    winnerteamID int not null,
    gamescore int not null,
    primary key(gameID),
    foreign key (winnerteamID) references team(teamID),
    foreign key (roundID) references rounds(roundID)
);

insert into game values (1, '2018-01-06', 'Single', 1, 1, 9, 1,
30);
```



```

insert into game values (2, '2018-01-07', 'Pair', 2, 2, 10, 2,
40);
insert into game values (3, '2018-01-13', 'Single', 3, 3, 8, 3,
32);
insert into game values (4, '2018-01-14', 'Pair', 4, 4, 9, 4,
35);
insert into game values (5, '2018-01-20', 'Single', 5, 5, 7,
10, 12);
insert into game values (6, '2018-01-21', 'Pair', 6, 10, 10, 1,
25);
insert into game values (7, '2018-01-27', 'Single', 7, 8, 9, 3,
26);
insert into game values (8, '2018-01-28', 'Pair', 8, 6, 8, 4,
29);
insert into game values (9, '2018-01-03', 'Single', 9, 2, 10,
5, 36);
insert into game values (10, '2018-01-04', 'Pair', 10, 1, 8, 3,
35);

create table sections(
    sectionID int not null,
    gameID int not null,
    winnerteamID int not null,
    looserteamID int not null,
    sectionscore int not null,
    primary key (sectionID),
    foreign key (gameID) references game(gameID),
    foreign key (winnerteamID) references team(teamID),
    foreign key (looserteamID) references team(teamID)
);

insert into sections values (1, 1, 1, 10, 30), (2, 1, 1, 10,
24), (3, 1, 10, 1, 12), (4, 2, 2, 3, 21), (5, 2, 1, 4, 32), (6, 2,
5, 10, 15), (7, 3, 6, 5, 13), (8, 3, 8, 7, 30), (9, 3, 9, 8,
20), (10, 4, 7, 3, 16);

create table Player(
    playerID int not null,
    role varchar(20) not null,
    playername varchar(50) not null,
    address varchar(250) not null,
    email varchar(100) not null,
    teamID int not null,
    primary key (playerID),
    foreign key (teamID) references team(teamID)
);
insert into Player values (1, 'Johny Dep', 'Staff', '3268

```

```

Oakridge Farm Lane', 'jn@qq.com', 1);
insert into Player values (2, 'Samuel Neville', 'Student', '67
Dalgarno Street', 'sn@qq.com', 1);
insert into Player values (3, 'John Doe', 'Student', '32
Taltarni Road', 'jdoe@qq.com', 1);
insert into Player values (4, 'Koby Monds', 'Student', '93
Ghost Hill Road', 'mond@qq.com', 6);
insert into Player values (5, 'Samuel Mackinnon', 'Staff', '23
Creegans Road', 'sam@qq.com', 2);
insert into Player values (6, 'Leo Dane', 'Student', '48
Creedon Street', 'leo@qq.com', 2);
insert into Player values (7, 'Joshua Kidman', 'Staff', '9 Todd
Street', 'jk@qq.com', 3);
insert into Player values (8, 'Cameron Mollison', 'Student',
'16 Harris Street', 'cm@qq.com', 3);
insert into Player values (9, 'Rory Collie', 'Staff', '9 Rose
Street', 'rc@qq.com', 4);
insert into Player values (10, 'Nate Towns', 'Student', '19
Girvan Grove', 'nt@qq.com', 4);

create table team(
    teamID int not null,
    teamname varchar(10) not null,
    latestscore int not null,
    leaderID int not null,
    primary key(teamID),
    foreign key(leaderID) references player(playerID)
);

insert into team values (1, 'Team 1',100,1), (2, 'Team 2',111,
4), (3, 'Team 3',122,1), (4, 'Team 4',114,8), (5, 'Team
5',145,10), (6, 'Team 6',110,6), (7, 'Team 7',132,7), (8, 'Team
8',100,8), (9, 'Team 9',102,9), (10, 'Team 10',101,10);
create table groups(
    groupID int not null,
    roundID int not null,
    teamID int not null,
    primary key (groupID, roundID),
    foreign key (roundID) references rounds(roundID),
    foreign key (teamID) references rounds(teamID)
);
insert into groups values (1, 1, 1), (1,2,10), (1,3,9),
(2,1,10), (2,2,8), (2,3,6), (3,1,7), (3,2,5), (3,3,4), (4,1,3);
create table referee(
    refereeID int not null,
    refereeName varchar(20),
    primary key(refereeID)

```

```
);
insert into referee values (1, 'Hugo Lillie'), (2, 'Hugo
Lillie'), (3, 'Rolly De'), (4, 'koby Cleland'), (5, 'Jasper'),
(6, 'Riley R'), (7, 'Seth'), (8, 'Jackson'), (9, 'Henry'), (10,
'Caleb');
```

- The tables in the database,

```
mysql> show tables;
+-----+
| Tables_in_tt1 |
+-----+
| game          |
| groups        |
| matches       |
| player        |
| referee       |
| rounds        |
| sections      |
| team          |
+-----+
8 rows in set (0.00 sec)
```

- The game table

```
mysql> select * from game;
+----+-----+-----+-----+-----+-----+-----+-----+
| gameID | gamedate | gametype | roundID | refereeID | refereescore | winnerTeamID | gamescore |
+----+-----+-----+-----+-----+-----+-----+-----+
| 1      | 2018-01-06 | Single  | 1       | 1         | 9            | 1             | 30         |
| 2      | 2018-01-07 | Pair    | 2       | 2         | 10           | 2             | 48         |
| 3      | 2018-01-13 | Single  | 3       | 3         | 8            | 3             | 32         |
| 4      | 2018-01-14 | Pair    | 4       | 4         | 9            | 4             | 35         |
| 5      | 2018-01-20 | Single  | 5       | 5         | 7            | 10            | 12         |
| 6      | 2018-01-21 | Pair    | 6       | 10        | 10           | 1             | 25         |
| 7      | 2018-01-27 | Single  | 7       | 8         | 9            | 3             | 26         |
| 8      | 2018-01-28 | Pair    | 8       | 6         | 8            | 4             | 29         |
| 9      | 2018-01-03 | Single  | 9       | 2         | 10           | 5             | 36         |
| 10     | 2018-01-04 | Pair    | 10      | 1         | 8            | 3             | 35         |
+----+-----+-----+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

- The groups table

```
mysql> select * from groups;
+-----+-----+-----+
| groupID | roundID | teamID |
+-----+-----+-----+
| 1       | 1       | 1       |
| 1       | 2       | 10      |
| 1       | 3       | 9       |
| 2       | 1       | 10      |
| 2       | 2       | 8       |
| 2       | 3       | 6       |
| 3       | 1       | 7       |
| 3       | 2       | 5       |
| 3       | 3       | 4       |
| 4       | 1       | 3       |
+-----+-----+-----+
10 rows in set (0.00 sec)
```

- The matches table

```
mysql> select * from matches;
+-----+-----+-----+-----+
| matchID | finalscore | winnerteamID | looserteamID |
+-----+-----+-----+-----+
| 1       | 100       | 1             | 10            |
| 2       | 110       | 2             | 9             |
| 3       | 100       | 4             | 5             |
| 4       | 105       | 1             | 9             |
| 5       | 90        | 1             | 10            |
| 6       | 80        | 9             | 8             |
| 7       | 103       | 9             | 1             |
| 8       | 101       | 10            | 5             |
| 9       | 120       | 6             | 3             |
| 10      | 50        | 4             | 1             |
+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

- The player table

```
mysql> select * from player;
+-----+-----+-----+-----+-----+-----+
| playerID | role      | playerName | address          | email          | teamID |
+-----+-----+-----+-----+-----+-----+
| 1       | Johnny   | Dep         | 3268 Oakridge   | jn@qq.com     | 1      |
| 2       | Samuel   | Neville     | 67 Dalgarno     | sn@qq.com     | 1      |
| 3       | John     | Doe         | 32 Taltarni     | jdo@qq.com    | 1      |
| 4       | Koby     | Monds      | 93 Ghost Hill   | mon@qq.com    | 6      |
| 5       | Samuel   | Mackinnon  | 23 Creegans    | sam@qq.com    | 2      |
| 6       | Leo      | DANE       | 48 Creedon      | leo@qq.com    | 2      |
| 7       | Joshua   | Kidman     | 9 Todd Street   | jk@qq.com     | 3      |
| 8       | Cameron  | Mollison   | 16 Harris       | cm@qq.com     | 3      |
| 9       | Rory     | Collie     | 9 Rose Street   | rc@qq.com     | 4      |
| 10      | Nate     | Towns      | 19 Girvan Grove | nt@qq.com     | 4      |
+-----+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

- The referee table

```
mysql> select * from referee;
+-----+-----+
| refereeID | refereeName |
+-----+-----+
| 1       | Hugo Lillie |
| 2       | Hugo Lillie |
| 3       | Rolly De    |
| 4       | koby Cleland |
| 5       | Jasper     |
| 6       | Riley R.   |
| 7       | Seth       |
| 8       | Jackson    |
| 9       | Henry      |
| 10      | Caleb      |
+-----+-----+
10 rows in set (0.00 sec)
```

| Need Assignment Help

- The rounds table

```
mysql> select * from rounds;
+-----+-----+
| roundID | matchID |
+-----+-----+
| 1       | 1       |
| 2       | 1       |
| 3       | 1       |
| 4       | 1       |
| 5       | 2       |
| 6       | 2       |
| 7       | 2       |
| 8       | 2       |
| 9       | 3       |
| 10      | 3       |
+-----+-----+
10 rows in set (0.00 sec)
```

- The sections table

```
mysql> select * from sections;
```

sectionID	gameID	winnerteamID	looserteamID	sectionscore
1	1	1	10	30
2	1	1	10	24
3	1	10	1	12
4	2	2	3	21
5	2	1	4	32
6	2	5	10	15
7	3	3	6	13
8	3	8	7	30
9	3	9	8	20
10	4	7	3	16

```
10 rows in set (0.00 sec)
```

- The team table

```
mysql> select * from team;
```

teamID	teamname	initialscore	loserID
1	Team 1	100	1
2	Team 2	111	2
3	Team 3	122	3
4	Team 4	134	4
5	Team 5	145	5
6	Team 6	110	6
7	Team 7	112	7
8	Team 8	100	8
9	Team 9	102	9
10	Team 10	101	10

```
10 rows in set (0.00 sec)
```

The queries on the database are,

- The winner of each game in each round,

```
mysql> select game.roundID, gameID, gamedate, winnerteamID, teamname
-> from (game inner join rounds on game.roundid = rounds.roundid) inner join team on team.teamid = game.winnerteamid
-> group by game.roundid;
```

roundID	gameID	gamedate	winnerteamID	teamname
1	1	2018-01-06	1	Team 1
2	2	2018-01-07	2	Team 2
3	3	2018-01-13	3	Team 1
4	4	2018-01-14	4	Team 1
5	5	2018-01-20	10	Team 1
6	6	2018-01-21	1	Team 1
7	7	2018-01-27	3	Team 1
8	8	2018-01-28	4	Team 1
9	9	2018-01-03	5	Team 1
10	10	2018-01-04	3	Team 1

```
10 rows in set (0.03 sec)
```

- The teams with their scores,

```
mysql> select * from team order by latestscore ASC;
+-----+-----+-----+-----+
| teamID | teamname | latestscore | leaderID |
+-----+-----+-----+-----+
| 1 | Team 1 | 100 | 1 |
| 8 | Team 1 | 100 | 8 |
| 10 | Team 1 | 101 | 10 |
| 9 | Team 1 | 102 | 9 |
| 6 | Team 1 | 110 | 6 |
| 2 | Team 2 | 111 | 4 |
| 4 | Team 1 | 114 | 8 |
| 3 | Team 1 | 122 | 1 |
| 7 | Team 1 | 132 | 7 |
| 5 | Team 1 | 145 | 10 |
+-----+-----+-----+-----+
10 rows in set (0.01 sec)
```

- The referee with all games he served.

```
mysql> select game.refereeID, refereeName, gameID
-> from referee inner join game on referee.refereeID = game.refereeID
-> group by game.refereeID;
+-----+-----+-----+
| refereeID | refereeName | gameID |
+-----+-----+-----+
| 1 | Hugo Lillie | 1 |
| 2 | Hugo Lillie | 2 |
| 3 | Rolly De | 3 |
| 4 | koby Cleland | 4 |
| 5 | Jasper | 5 |
| 6 | Riley R | 8 |
| 8 | Jackson | 7 |
| 10 | Caleb | 6 |
+-----+-----+-----+
8 rows in set (0.00 sec)
```

- The list of looser teams

```
mysql> select distinct matches.looserteamID, teamname, latestscore
-> from matches inner join team on matches.looserteamID = team.teamID;
+-----+-----+-----+
| looserteamID | teamname | latestscore |
+-----+-----+-----+
| 1 | Team 1 | 100 |
| 3 | Team 3 | 122 |
| 5 | Team 5 | 145 |
| 8 | Team 8 | 100 |
| 9 | Team 9 | 102 |
| 10 | Team 10 | 101 |
+-----+-----+-----+
6 rows in set (0.00 sec)
```

- List of all players in all teams

```
mysql> select * from player;
```

playerID	role	playername	address	email	teamID
1	Johny Dep	Staff	3268 Oakridge Farm Lane	jn@qq.com	1
2	Samuel Neville	Student	67 Dalgarno Street	sn@qq.com	1
3	John Doe	Student	32 Taltarni Road	jdoe@qq.com	1
4	Koby Monds	Student	93 Ghost Hill Road	mond@qq.com	6
5	Samuel Mackinnon	Staff	23 Creegans Road	sam@qq.com	2
6	Leo Dane	Student	48 Creedon Street	leo@qq.com	2
7	Joshua Kidman	Staff	9 Todd Street	jk@qq.com	3
8	Cameron Mollison	Student	16 Harris Street	cm@qq.com	3
9	Rory Collie	Staff	9 Rose Street	rc@qq.com	4
10	Nate Towns	Student	19 Girvan Grove	nt@qq.com	4

```
10 rows in set (0.00 sec)
```

- Game score after each round, including section scores.

```
mysql> select game_gameID, gamescore, sectionID, sectionscore
-> from game inner join sections on game_gameID = sections_gameID;
```

gameID	gamescore	sectionID	sectionscore
1	30	1	30
1	30	2	24
1	30	3	12
2	40	4	21
2	40	5	32
2	40	6	15
3	32	7	13
3	32	8	30
3	32	9	20
4	35	16	16

```
10 rows in set (0.00 sec)
```

- The total number of games played by each team in a match

```
mysql> (select matches.matchid, team.teamid, count(game_gameID) as numGamesPlayed
-> from ((matches inner join rounds on matches.matchID = rounds.matchID) inner join game on rounds.roundID = game.roundID) inner join team on matches.winnerteamID =
team.teamID
-> group by matches.matchID, team.teamID)
-> UNION
(select matches.matchid, team.teamid, count(game_gameID) as numGamesPlayed
-> from ((matches inner join rounds on matches.matchID = rounds.matchID) inner join game on rounds.roundID = game.roundID) inner join team on matches.looserteamID =
team.teamid
-> group by matches.matchID, team.teamID);
```

matchid	teamid	numGamesPlayed
1	1	4
2	2	4
1	6	2
1	10	4
2	9	4
3	9	3

```
6 rows in set (0.00 sec)
```

- Total score of each team after a match

```
mysql> (Select winnerteamID, finalscore from matches)
-> UNION
-> (select looserteamID, finalscore from matches);
```

winnerteamID	finalscore
1	100
2	110
4	100
1	105
1	90
9	80
9	103
10	101
6	120
4	50
10	100
9	110
5	100
9	105
10	90
8	80
1	103
5	101
3	120
1	50

20 rows in set (0.01 sec)

- Winner list of all recorded matches

```
mysql> select matches.winnerteamID, teamname, finalscore
-> from matches inner join team on matches.winnerteamID = team.teamID
-> ;
```

winnerteamID	teamname	finalscore
1	Team 1	100
2	Team 2	110
4	Team 4	100
1	Team 1	105
1	Team 1	90
9	Team 9	80
9	Team 9	103
10	Team 10	101
6	Team 6	120
4	Team 4	50

10 rows in set (0.00 sec)

- Referees ranking

```
mysql> select matches.matchID, game.refereeID, refereescore
-> from (matches inner join rounds on matches.matchID = rounds.matchID) inner join game on rounds.roundID = game.roundID
-> group by matches.matchID
-> order by refereescore;
```

matchID	refereeID	refereescore
2	5	7
1	1	9
3	2	10

3 rows in set (0.00 sec)

References

Coronel, C., & Morris, S. (2016). *Database Systems: Design, Implementation, and Management*. Cengage Learning.

Ramez Elmasri, S. B. (2016). *Fundamentals of Database Systems*. Pearson.

