

1. Database use cases

1. Show all players of a team
 - a. Open the web browser
 - b. Open the webpage
 - c. Click on teams - to display all teams
 - d. Click on particular team of interest
 - e. Click on roster

```
SELECT Name FROM Player as p, Team as t WHERE t.club_name =  
"Barcelona" and p.team_ID = t.team_ID
```

2. Teams standing for the season
 - a. Open web browser
 - b. Open the webpage
 - c. Click on team standings

```
Select Club_Name,Team_Points from Team order by Team_Points asc
```

3. View Player details
 - a. Open the web browser
 - b. Open the webpage
 - c. Click on teams - to display all teams
 - d. Click on particular team of interest
 - e. Click on roster
 - f. Click on particular player

```
SELECT * from Player as p where p.Name = "Anurag Baral"
```

4. View Game details
 - a. Open the web browser
 - b. Open the webpage
 - c. Click on teams - to display all teams
 - d. Click on particular team of interest
 - e. Pick a game of interest

```
Select * from Game as g where team_1_id = "1" or team_2_id = "1"
```

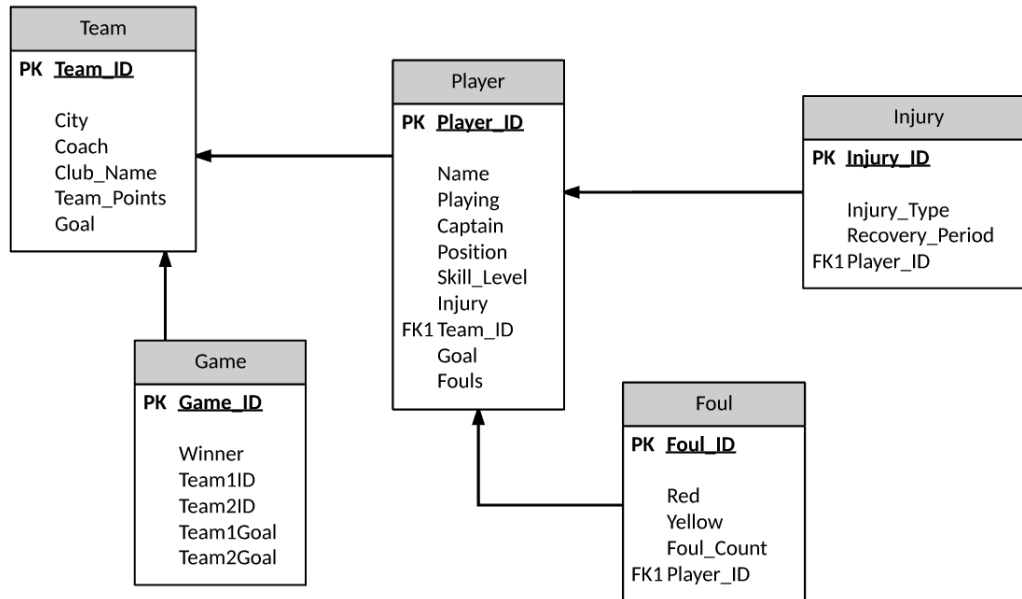
5. View all game schedules
 - a. Open the web browser
 - b. Open the webpage
 - c. Click on "schedule"

```
Select * from Game order by Time asc
```

6. Search for MVP of league
 - a. Open the web browser
 - b. Open the webpage
 - c. Click on Rankings

```
Select Name, Goal from Player order by Goal desc limit 1
```

2.1 Normalization: 1NF & 2NF



2.2 Dependencies and Normalization: 3NF

Table: Team

<u>Team_ID</u>	City	Coach	Club_Name	Team_Points	Goal
----------------	------	-------	-----------	-------------	------

Assumptions:

All attributes are dependent on Primary Key. No other attributes (except for primary key) will get you access to any other attributes in Team table. So in Team table, Club_Name cannot give you the value for City, Coach, Team_Points and Goal. All the attributes are the first level information of Team.

Logically it might seem possible that a Coach information can give you the information of the club_name. But the assumption is Coach and Club_name is both information given by the Team_ID. Similarly with any attributes you can get to all the attributes only by using Team_ID.

Table: Game


<u>Game_ID</u>	Winner	Team1ID	Team2ID	Team1Goal	Team2Goal
----------------	--------	---------	---------	-----------	-----------

Assumptions:

All attributes are dependent on Game_ID (the only primary key). For example, you cannot view Team1Goal just by accessing Winner or any other attributes. You have to relate the attribute(the one you want to view the data) with Game_ID in order to access any other attributes.

Also the Team1Goal or Team2Goal is not the same as Goal of the Team table. So every attributes on this table is also the first level information of that particular game.

Table: Player




<u>Player_ID</u>	Name	Playing	Captain	Position	Skill_Level	Injury	Team_ID	Goal	Fouls
------------------	------	---------	---------	----------	-------------	--------	---------	------	-------

Assumptions:

Any Player's information in a game can only be given via Player_ID, which is the only primary key in the Player Table.

Table: Foul




<u>Foul_ID</u>	Red	Yellow	Foul_count	Player_ID
----------------	-----	--------	------------	-----------

Assumptions:

Every other attribute is dependent on Foul_id(Primary Key). You cannot access Foul information of any player if you don't use Foul_ID attribute.

Table: Injury



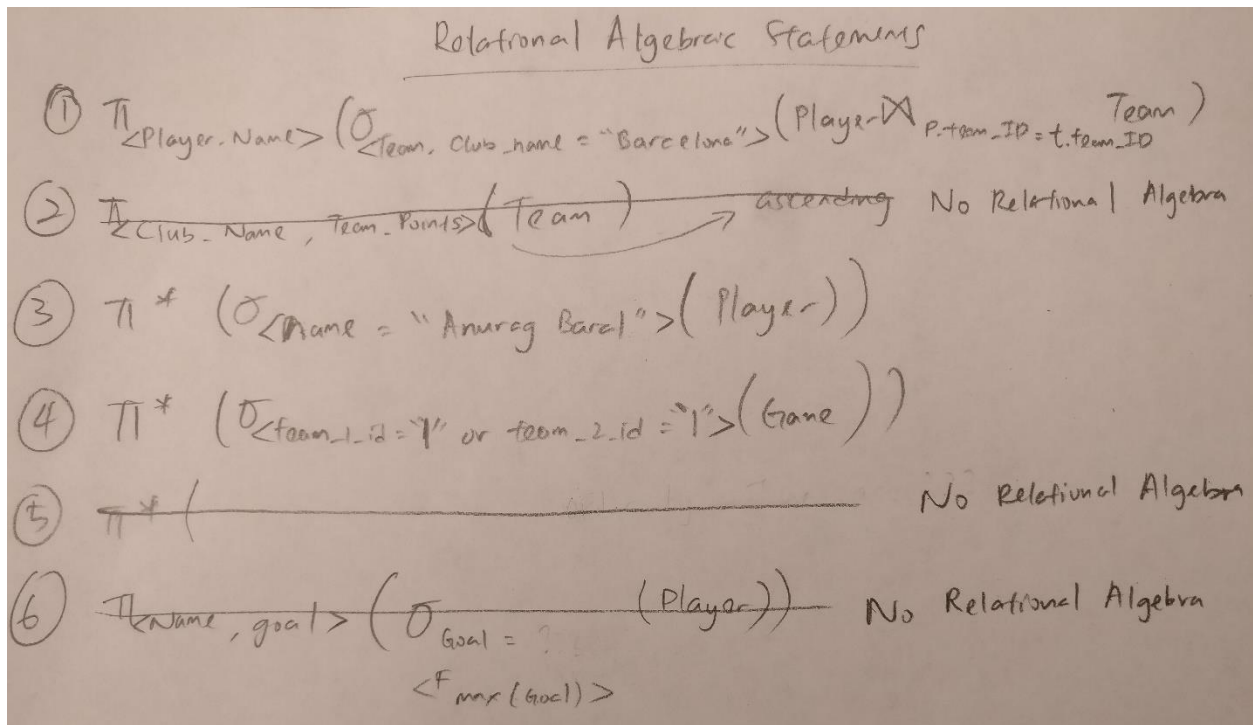
<u>Injury_ID</u>	Injury_Type	Recovery_Period	Player_ID
------------------	-------------	-----------------	-----------

Assumptions:

If you want to retrieve an Injury information of a player, you can only do so by using the Primary key which is Injury_Id. No other attributes can give you information of the other attribute.

2.3 Relational Algebra

Please view according to use case number sequence:



2.4 SQL statements

Please view according to use case number sequence:

1. SELECT Name FROM Player as p, Team as t WHERE t.club_name = "Barcelona" and p.team_ID = t.team_ID
2. Select Club_Name, Team_Points from Team order by Team_Points asc
3. SELECT * from Player as p where p.Name = "Anurag Baral"
4. Select * from Game as g where team_1_id = "1" or team_2_id = "1"
5. Select * from Game order by Time asc
6. Select Name, Goal from Player order by Goal desc limit 1

3.1 Description of MongoDB Design

We created the MongoDB schema design in a way such that all the use cases could be retrieved. The first level information is games which contains array of single games. Inside games is the first level attributes like winner, team1goal, and others. Nested inside games is also another collection of teams. Teams contain all the attributes inside a team relation like city, coach, and the others mentioned on the schema. Inside teams is also a nested collection of players. There are many players with each having information such as name, playing, skill_level. Inside players there are two nested collection injury and fouls which contains additional information.

The schema was created this way so that our use cases could be all facilitated! Look at the use cases for more information like attributes and relations!

3.2 MongoDB database design

Please refer to the file named “template.json” included in our team’s submission for better readability.

```
{
  "games": {
    "game1": {
      "winner": "Barcelona",
      "team1goal": 1,
      "team2goal": 3,
      "teams": {
        "team1": {
          "city": "Barcelona",
          "coach": "Anurag Baral",
          "club_name": "Barcelona",
          "team_points": 10,
          "goal": 20,
          "players": {
            "player1": {
              "name": "Lionel Messi",
              "playing": 1,
              "captain": 0,
              "position": "RW",
              "skill_level": 92,
              "injury": {
                "injury_id": {
                  "injury_type": "Shoulder Sprain",
                  "recovery_period": 30
                }
              },
              "goal": 2,
              "fouls": {
                "red": 0,

```

Sujan Aryal
Julian Fung
Harsha
Anurag Baral

```
        "yellow" : 0,
        "foul_count" : 1
    }
},
"player2" : {
    "name" : "Neymar" ,
    "playing" : 1,
    "captain" : 0,
    "position" : "LW",
    "skill_level" : 90,
    "injury" : {
        "injury_id" : {
            "injury_type" : "Leg Sprain",
            "recovery_period" : 3
        }
    },
    "goal" : 2,
    "fouls" : {
        "red" : 0,
        "yellow" : 0,
        "foul_count" : 1
    }
}
},
"team2" : {
    "city" : "London",
    "coach" : "Julian Fung",
    "club_name" : "Arsenal",
    "team_points" : 8,
    "goal" : 10,
    "players" : {
        "player1" : {
            "name" : "Mezut Ozil" ,
            "playing" : 1,
            "captain" : 0,
            "position" : "RW",
            "skill_level" : 87,
            "injury" : {
                "injury_id" : {
                    "injury_type" : "Shoulder Sprain",
                    "recovery_period" : 2
                }
            },
            "goal" : 3,
            "fouls" : {
                "red" : 0,
                "yellow" : 0,
                "foul_count" : 1
            }
        },
        "player2" : {
```

Sujan Aryal
Julian Fung
Harsha
Anurag Baral

```
"name" : "Theo Walcott" ,
"playing" : 1,
"captain" : 0,
"position" : "LW",
"skill_level" : 87,
"injury" : {
  "injury_id" : {
    "injury_type" : "Leg Sprain",
    "recovery_period" : 1
  }
},
"goal" : 6,
"fouls" : {
  "red" : 0,
  "yellow" : 0,
  "foul_count" : 1
}
}
}
}
}
}
}
```

4. Team member roles

1. Sujan
 - a. Brainstormed use cases
 - b. Wrote step by step procedures for each use cases
 - c. Discussed and participated in normalization of tables
 - d. Database dependencies, SQL queries
 - e. MongoDB design and schema
2. Harsha
 - a. Brainstormed use cases
 - b. Discussed and participated in normalization of tables
 - c. Database dependencies, SQL queries
 - d. Testing of SQL queries
 - e. MongoDB design and schema
3. Anurag
 - a. Brainstormed use cases
 - b. Wrote step by step procedures for each use case
 - c. Wrote step by step procedures for each use case
 - d. Discussed and participated in normalization of tables
 - e. Database dependencies, Relational algebraic statements in SQL
4. Julian
 - a. Brainstormed use cases

Sujan Aryal
Julian Fung
Harsha
Anurag Baral

- b. Wrote step by step procedures for each use case
- c. Visited Professor for approval of use cases
- d. Discussed and participated in normalization of tables
- e. Database dependencies, Relational algebraic statements in SQL
- f. Clarification of member roles