**1.** Team members are Jingguo Liang (jliang35) and Ruixuan Zhai (rzhai2).

**2.** The target will be a database for Overwatch League (OWL), an esports league for the videogame Overwatch. We hope our interactive web page can provide useful information to each team about other teams and players based on past data. (i.e. which player has the specific skill they are looking for? What is the biggest threat coming from a specific player?) We can also provide fans general information about past games.

**3. Give a reasonably comprehensive and representative list of the kinds of English questions you would like your system to be able to answer (minimum 15).**

1. List all players who have served in at least two OWL teams, and have either defeated his new team in his original team, or defeated his original team in his new team.

2. List the player name, and role of all players who have played at least 3 different heroes in one match. Also list the number of heroes played, date, tournament title of the match. Order by the number of heroes in descending order.

3. Given a name of a player, list all the teams this player has defeated, together with the date of the match, the tournament title, and his team playing the match.

4. Given the tournament title, list the top 3 used heroes (in terms of total Time Played) and their total used time of each of the 3 roles (Tank, DPS, Support) used in all matches in this tournament, respectively.

5. Given the tournament title,list the top 10 players who have spent the greatest percentage of time respawning, meaning that they have the lowest Time Alive / Time Played ratio.

6. Given the tournament title and a hero,list the top 5 players who do the greatest amount of damage per 10min using this hero. Also list their damage/10min using this hero.

7. Fleta Deadlift is the scenario where a player accounts for 50% or more of  the final blows of the entire team in one map. The size of Fleta Deadlift is defined by the number of this player’s final blows / total number of final blows of his team members in the map. List all players who have performed a Fleta Deadlift, their team, the date of the match, their opponent in the match, the map, and the size of the Fleta Deadlift.

8. Given the name of a tournament, the name of the team, and the name of the map,list the number of wins, loses, and draws, and the win rate of the team on the map in all matches of that tournament.

9. Given the name of the tournament, list the name and date of all CONTROL type maps played, where all rounds finish with control percentage 100-99 or 99-100. Also list the two teams in the match.

10. List top 20 players of role Support, their team, and the date of the match, where the player heals the greatest percentage of the damage taken by the whole team in that match. One player can have multiple occurrences on the list.

11. List the top 5 players who play hero Mercy and have the highest final blow per 10min.

12. Given the name of the tournament, list all players and their role who have played a hero with the role different from the player’s role. Also list the name and role of the hero. List its damage/10min and its healing/10min.

Query 13-15 involves models to evaluate players and teams, and also some machine learning. Those queries are only in their early forms, and may change later.

13. Given two names of the players, return a statement on which one of the two is expected to behave better. (We will pretrain a model with the factors we pick and the percentage of winning as a label. With the input player name, we input the factors of that specific player into the model and provide the winning percentage predicted by the machine. We compare the winning percentage of the two and we believe the player with a higher winning percentage is “better”.

14. Given two teams, return a statement on which one of the two might be a better team. (Same as above, we will determine a model to decide this.)

15. Give a rank of all players based on the model we trained. Show in descending order.

**4. Design and show a relational data model that you plan to use for your system, with a preliminary implementation in standard SQL data-definition-language syntax. This specification should include appropriate primary key, foreign key, and domain specifications for each relation/attribute, as well as the not null constraint when appropriate. You may also find it useful, but not required, to create a few insert-into statements that populate your schema designs with representative values (both to document your choices and to exercise them. You are welcome to change and augment your design and its specification by Phase II, but any time investment now will reduce effort later.**

DROP TABLE Player;

CREATE TABLE Player (

    PlayerName VARCHAR(20),

    RealName VARCHAR(30),

    Birthday DATE,

    Country VARCHAR(15),

    Status VARCHAR(6),

    Role VARCHAR(7),

    PRIMARY KEY PlayerName

);

DROP TABLE Team;

CREATE TABLE Team (

    TeamName VARCHAR(30),

    City VARCHAR(15),

    Country VARCHAR(15),

    DateFounded DATE,

    PRIMARY KEY TeamName

);

DROP TABLE ServesIn;

CREATE TABLE ServesIn (

    PlayerName VARCHAR(20),

    TeamName VARCHAR(30),

    StartDate DATE,

    EndDate DATE,

    PRIMARY KEY (PlayerName, TeamName, StartDate)

);

DROP TABLE Hero;

CREATE TABLE Hero (

    HeroName VARCHAR(15),

    Role VARCHAR(7)

)

DROP TABLE Map;

CREATE TABLE Map (

    MapName VARCHAR(25),

    MapType VARCHAR(7),

    PRIMARY KEY MapName

);

DROP TABLE Match;

CREATE TABLE Match (

    MatchID INTEGER,

    TournamentTitle VARCHAR(50),

    StartDate DATE,

    StartTime TIME,

    MatchWinner VARCHAR(30),

    MatchLoser VARCHAR(30),

    PRIMARY KEY MatchID

);

DROP TABLE PlaysMatch;

CREATE TABLE PlaysMatch (

    MatchID INTEGER,

    TeamName VARCHAR(30),

    PRIMARY KEY (MatchID, TeamName)

);

DROP TABLE PlaysMap;

CREATE TABLE PlaysMap (

    MatchID INTEGER,

    MapName VARCHAR(25),

    MapNo INTEGER,

    MapWinner VARCHAR(30),

    MapLoser VARCHAR(30),

    PRIMARY KEY (MatchID, MapName)

);

DROP TABLE MapPlayerStat;

CREATE TABLE MapPlayerStat (

    MatchID INTEGER,

    MapName VARCHAR(25),

    PlayerName VARCHAR(20),

    HeroName VARCHAR(15),

    StatName VARCHAR(50),

    StatAmount DOUBLE,

    PRIMARY KEY (MatchID, MapName, PlayerName, HeroName, StatName)

);

DROP TABLE MapRoundStat;

CREATE TABLE MapRoundStat (

    MatchID INTEGER,

    MapName VARCHAR(25),

    RoundNo INTEGER,

    StartTime TIME,

    EndTime TIME,

    ControlRoundName VARCHAR(25),

    AttackerTeam VARCHAR(30),

    DefenderTeam VARCHAR(30),

    AttackerPayloadDistance DOUBLE,

    DefenderPayloadDistance DOUBLE,

    AttackerTimeBanked DOUBLE,

    DefenderTimeBanked DOUBLE,

    AttackerControlPercent DOUBLE,

    DefenderControlPercent DOUBLE,

    AttackerRoundEndScore INTEGER,

    DefenderRoundEndScore INTEGER,

    PRIMARY KEY (MatchID, MapName, RoundNo)

);

**5. Submit a set of SQL statements that will implement a representative sample of your target queries, including some of the more interesting or challenging cases. This is primarily to get you to think about your design and how it will be exercised as well as any limitations, so focus on queries that would be useful for doing so, rather than creating trivial or non-insightful queries just to fill space.**

/\* 1 \*/

DELIMITER |

DROP PROCEDURE IF EXISTS serve\_more\_than\_two;

CREATE PROCEDURE serve\_more\_than\_two( )

SELECT A.PlayerName

FROM

(SELECT PlayerName, count(TeamName) AS Num, TeamName

FROM ServesIn

WHERE EndDate IS NULL

GROUP BY PlayerName) AS A

WHERE A.Num >=

 |

/\* 2 \*/

**6.      Provide a plan for how you will load the database with values.**

Some of the data (i.e. playoffs for the past year or two, information about different maps) are provided by the website in .csv format. We need to write a script to separate the columns and insert them into the tables.

For the data that we get from web crawling, we will store them in an excel and repeat the steps above to load them into the database.

**7. Very briefly describe the form/type of output or result you plan to generate or any special user interface issues that you plan to implement.**

The output format will be similar to what we did in homework 3. We will have an opening page with instructions (what the user can input and what kind of result will they get) with places for user input and a button to indicate the showing of results.

By clicking the button, the user will be taken to another page in which there will be a form that has the information they expect and probably a sentence or two depends on which particular part the user chooses to explore. (i.e. If the user chooses to input two players and see who (will) perform(s) better, the user will not only see the score we give for each player but also a sentence stating who is better.)

**8. What are the specialized/advanced topics you plan to focus on in your database design?**

We would like to pick “complex data extraction issues from online data sources” as our major focus and “data mining”, “particularly advanced GUI form interface and/or report generation” as our minor focus.

For “complex data extraction issues for online data sources”, we believe we will do much web crawling using python since 50 percent of the data we need are scattered on different web pages. For example each player’s information (the nationality, the teams they have served, etc.) is under a webpage of his own and laid out in a table. We can’t download the table directly from the webpage, so we need to do web crawling to get those and do cross comparison with other sets of tables we download.

For “data mining”, we want to create a webpage that allows the users to type in two teams/players and return which one of the two has a greater chance of winning. This will include a construction of a model (i.e. what factors do we take into consideration when we try to judge which one of the two options perform better), and a basic machine learning code written in python(sklearn) and use PHP file to feed it the data we extract from the database using SQL.

For “particularly advanced GUI form interface and/or report generation”, we are thinking about making a user friendly UI using Bootstrap. It depends on how much time we have and how smooth is the progress, we might only furnish the opening page if time is limited.

Also, if things work well, we will do the two minors mentioned above, but if not, we will pick on out of the two. We would love to hear your suggestions as well.