2. ER Diagram

The ER diagram to derive from will be the same as the one in Milestone 1.

- Person Entity:
  - Removed region - background information about managers and coaches are usually not as interesting compared the the Player for the API user
- Player Entity:
  - Added region, city and country - region was moved from Person to Player
    - City and country represents hometown of the player
- Tournament Entity:
  - Rename tournamentID attribute to ID because it is it's own primary key, does not need the 'tournament' prefix
- Match Entity
  - Rename matchID attribute to ID - same reasoning as above
- TeamPlaysMatch
  - Remove winner attribute - the columns winningTeamId and losingTeamId is sufficient
  - Add one more score attribute - call them winningTeamScore and losingTeamScore so score is not char(30) based but integer based with each in it's own column to make more sense
- TeamParticipatesTournament
  - Add furthestRound attribute - we want to see to what stage of the tournament each team made it to
- Item Entity
  - Add tier attribute - categorize items into tiers

3. Schemas

<u>Underline</u> = Primary Key

Bold = Foreign Key

Person(<u>id</u>: Integer, name: char(30))

- Constraints:
  - (name is Not Null)

Manager(**<u>id</u>**: Integer)

Coach(**<u>id</u>**: Integer)

Player(**<u>id</u>**: Integer, **teamId:** Integer, username: char(30), region: char(4), city: char(30), country: char(30))

- Constraints:

    - ((username, region) is unique and candidate key)

Team(id: Integer, name: char(30), region: char(4))

- Constraints:
    - (name is Not Null, Unique and candidate key)

CoachCoachesTeam(**teamId**: Integer, **coachId**: Integer)

ManagerManagesTeam(**teamId**: Integer, **managerId**: Integer)

Champion(name: char(30))

PlayerPlaysChampion(**playerId**: Integer, **championName**: char(30), KDARatio: float, winPercentage: float)

TopLaner(**name**: char(30))

MidLaner(**name**: char(30))

Jungler(**name**: char(30))

Support(**name**: char(30))

BottomLaner(**name**: char(30))

Item(name: Char(25), cost: Integer, tier: char(30))

ChampionUsesItems(**championName**: char(30), **itemName**: char(30))

Tournament(id: Integer, name: char(30), dateStart: Date, dateEnd: Date, season: Integer)

- Constraints
    - (name, season) unique and candidate key

Match(id: Integer, **tournamentId**: Integer, date: Date, gameLength: decimal)

TeamPlaysMatch(**winningTeamId**: Integer, **losingTeamId**: Integer, **tournamentId**: Integer, **matchId**: Integer, winningTeamScore: Integer, losingTeamScore: Integer)

TeamParticipatesTournament(**tournamentId**: Integer, **teamId**: Integer, ranking: Integer, furthestRound: CHAR(30))

- Constraints
    - (tournamentId, ranking) is unique and candidate key

Sponsor(companyName: char(20))

SponsorSponsorsTeam(**teamId**: Integer, **tournamentId**: Integer, **companyName**: char(20), amount: decimal)

4. Functional Dependencies

NOTE: Left out trivial FD's

Person:

- id -> name

Player:

- id -> teamId, username, region, city, country
- username, region -> teamId, id, region, city, country
- city, country -> region

Team:

- id -> name, region
- name -> region, id

PlayerPlaysChampion:

- playerId, championName -> KDARatio, winPercentage

Item:

- name -> cost, tier
- cost -> tier

Tournament:

- tournamentId -> name, dateStart, dateEnd, season
- name, season -> tournamentId, dateStart, dateEnd

Match:

- matchId, tournamentId -> date, gameLength

TeamPlaysMatch:

- winningTeamId, losingTeamId, tournamentId, matchId -> winningTeamScore, losingTeamScore

TeamParticipatesTournament:

- tournamentId, teamId -> ranking
- tournamentId, ranking -> teamId
- ranking -> furthestRound

SponsorSponsorsTeam:

- teamId, tournamentId, companyName -> amount

5. Normalization

Underline = Primary Key
Bold = Foreign Key

1. Item (2NF):

- name -> cost, tier
- cost -> tier

Item after normalization (BCNF):

- ItemNameCost(name, **cost**)
- ItemCostTier(cost, tier)

2. TeamParticipatesTournament (2NF):

- tournamentId, teamId -> ranking
- tournamentId, ranking -> teamId
- ranking -> furthestRound

TeamParticipatesTournament after normalization (BCNF):

- TeamParticipatesTournament(**tournamentId**, **teamId**, **ranking**)
    - Constraints:
        - (tournamentId, ranking) is unique and candidate key
- RankingFurthestRound(ranking, furthestRound)

6. SQL DDL

```
CREATE TABLE Person (
id INTEGER PRIMARY KEY,
name CHAR(30) NOT NULL,
)

CREATE TABLE Manager (
id INTEGER PRIMARY KEY
FOREIGN KEY id REFERENCES Person
        ON DELETE CASCADE
        ON UPDATE CASCADE
)
```

```
CREATE TABLE Coach (
id INTEGER PRIMARY KEY
FOREIGN KEY id
        REFERENCES Person
        ON DELETE CASCADE

)

CREATE TABLE Player (
id INTEGER PRIMARY KEY,
teamId INTEGER NOT NULL DEFAULT -1,
username CHAR(30),
region CHAR(4),
city CHAR(30),
country CHAR(30)
FOREIGN KEY id
        REFERENCES Person
        ON DELETE CASCADE
        ON UPDATE CASCADE,
FOREIGN KEY teamId
        REFERENCES Team
        ON DELETE SET DEFAULT
        ON UPDATE CASCADE,
UNIQUE(username, region)
)

CREATE TABLE Team (
id INTEGER PRIMARY KEY,
name CHAR(30) NOT NULL UNIQUE,
region CHAR(4)
)

CREATE TABLE CoachCoachesTeam (
teamId INTEGER,
coachId INTEGER,
PRIMARY KEY(teamId, coachId)
FOREIGN KEY teamId
        REFERENCES Team
        ON DELETE CASCADE
        ON UPDATE CASCADE,
FOREIGN KEY coachId
        REFERENCES Person
        ON DELETE CASCADE
```

```
          ON UPDATE CASCADE
)

CREATE TABLE ManagerManagesTeam (
teamId INTEGER,
managerId INTEGER,
PRIMARY KEY(teamId , managerId )
FOREIGN KEY teamId
          REFERENCES Team
          ON DELETE CASCADE
          ON UPDATE CASCADE,
FOREIGN KEY managerId
          REFERENCES Manager
          ON DELETE CASCADE
          ON UPDATE CASCADE
)

CREATE TABLE Champion (
name CHAR(15) PRIMARY KEY,
)

CREATE TABLE PlayerPlaysChampion (
playerId INTEGER,
championName CHAR(30),
KDARatio FLOAT,
winPercentage FLOAT,
PRIMARY KEY(PlayerID, ChampionName),
FOREIGN KEY playerId
          REFERENCES Player
          ON DELETE CASCADE
          ON UPDATE CASCADE,
FOREIGN KEY championName
          REFERENCES Champion
          ON DELETE CASCADE
          ON UPDATE CASCADE
)

CREATE TABLE TopLaner (
name CHAR(30) PRIMARY KEY,
FOREIGN KEY name
          REFERENCES Champion
          ON DELETE CASCADE
          ON UPDATE CASCADE
)
```

```
CREATE TABLE MidLaner (
name CHAR(30) PRIMARY KEY,
FOREIGN KEY name
        REFERENCES Champion
        ON DELETE CASCADE
        ON UPDATE CASCADE
)

CREATE TABLE Jungler (
name CHAR(30) PRIMARY KEY,
FOREIGN KEY name
        REFERENCES Champion
        ON DELETE CASCADE
        ON UPDATE CASCADE
)

CREATE TABLE Support (
name CHAR(30) PRIMARY KEY,
FOREIGN KEY name
        REFERENCES Champion
        ON DELETE CASCADE
        ON UPDATE CASCADE
)

CREATE TABLE BottomLaner (
name CHAR(30) PRIMARY KEY,
FOREIGN KEY name
        REFERENCES Champion
        ON DELETE CASCADE
        ON UPDATE CASCADE
)

CREATE TABLE ItemNameCost (
name CHAR(30) PRIMARY KEY,
cost INTEGER,
FOREIGN KEY cost
        REFERENCES ItemCostTier
        ON DELETE SET NULL
        ON UPDATE CASCADE
)

CREATE TABLE ItemCostTier (
cost INTEGER PRIMARY KEY,
tier CHAR(30),
)

CREATE TABLE ChampionUsesItems (
championName CHAR(30),
itemName CHAR (30)
PRIMARY KEY(itemName, championName)
FOREIGN KEY championName
        REFERENCES Champion
        ON DELETE CASCADE
        ON UPDATE CASCADE
FOREIGN KEY itemName
        REFERENCES Item
        ON DELETE CASCADE
        ON UPDATE CASCADE
)


CREATE TABLE Tournament (
tournamentId INTEGER PRIMARY KEY,
name CHAR(20),
dateStart DATE,
dateEnd DATE,
season INTEGER,
UNIQUE(name, season)
)

CREATE TABLE Match (
matchId INTEGER,
tournamentId INTEGER,
Date DATE,
gameLength DECIMAL,
PRIMARY KEY (MatchID, TournamentID)
FOREIGN KEY (TournamentID)
        REFERENCES Tournament
        ON DELETE CASCADE
        ON UPDATE CASCADE
)

CREATE TABLE TeamsPlaysMatch (
winninTeamId INTEGER NOT NULL,
losingTeamId INTEGER NOT NULL,
tournamentId INTEGER NOT NULL,
matchId INTEGER NOT NULL,
winningTeamId INTEGER,
losingTeamId INTEGER,
PRIMARY KEY (winningTeamId, losingTeamId, TournamentID, MatchID)
FOREIGN KEY (winningTeamId) REFERENCES Team,
FOREIGN KEY (losingTeamId) REFERENCES Team,
FOREIGN KEY (TournamentID) REFERENCES Tournament, ON DELETE CASCADE,
FOREIGN KEY (matchID) REFERENCES MatchID,
)

CREATE TABLE TeamParticipantsTournament (
tournamentId INTEGER,
teamId INTEGER,
ranking INTEGER,
PRIMARY KEY(tournamentId, teamId)
FOREIGN KEY teamId
        REFERENCES Team
        ON DELETE CASCADE
        ON UPDATE CASCADE,
FOREIGN KEY tournamentId
        REFERENCES Tournament
        ON DELETE CASCADE
        ON UPDATE CASCADE,
FOREIGN KEY ranking
        REFERENCES RankingFurthestRound
        ON DELETE CASCADE
        ON UPDATE CASCADE,
UNIQUE(tournamentId, ranking)
)

CREATE TABLE RankingFurthestRound (
ranking INTEGER PRIMARY KEY,
furthestRound CHAR(30),
)

CREATE TABLE Sponsor (
CompanyName CHAR(30) PRIMARY KEY
)

CREATE TABLE SponsorSponsorsTeam(
TeamID INTEGER,
TournamentID INTEGER
CompanyName CHAR(30),
Amount DECIMAL
PRIMARY KEY (TeamID, TournamentID, companyName)
```

```
FOREIGN KEY (TeamID) REFERENCES Team
FOREIGN KEY (TournamentID) REFERENCES Tournament
FOREIGN KEY (companyName) REFERENCES Sponsor
)
```

7. Data entries

Person

id	name
1	Soren Bjerg
2	William Li
3	Lee Sang Hyeok
4	Aileena Xu
5	Jian Zi Hao
6	Guy one
7	Guy two
8	Guy three
9	Guy four
10	Guy five
11	Girl one
12	Girl two
13	Girl three
14	Girl four
15	Girl five

Manager

id
4
6
7

8
9

Coach

id
2
10
11
12
13

Player

id	teamId	username	region	city	country
1	1	Bjergsen	NA	Vancouver	America
3	4	Faker	KR	Seoul	Korea
5	3	UZI	CN	Beijing	China
14	5	girl4	BR	Salvador	Brazil
15	5	girl5	BR	Salvador	Brazil

Team

ID	Name	Region
1	TSM	NA
2	G2 Esports	EU
3	RNG	CN
4	SKT	KR
5	Amigos	BR

CoachCoachesTeam

teamId	coachId

1	2
2	10
3	11
4	12
5	13

ManagerManagesTeam

teamId	managerId
1	4
2	6
3	7
4	8
5	9

Champion

name
Gnar
Udyr
Lux
Jinx
Thresh

PlayerPlaysChampion

playerId	championName	KDARatio	winPercentage
1	Gnar	1.3	0.99
3	Udyr	1.0	0.2
5	Lux	0.9	0.3
14	Jinx	5.0	0.4
15	Thresh	0.2	0.5

TopLaner

name
Gnar
Udyr
Lux
Jinx
Thresh

MidLaner

name
Gnar
Udyr
Lux
Jinx
Thresh

Jungler

name
Gnar
Udyr
Lux
Jinx
Thresh

Support

name
Gnar
Udyr
Lux

Jinx
Thresh

BottomLaner

name
Gnar
Udyr
Lux
Jinx
Thresh

ItemNameCost

name	cost
Doran’s ring	400
Doran’s shield	400
Doran’s blade	400
Sunfire cape	4000
Ardent Censer	2800

ItemCostTier

cost	tier
400	starter
4000	legendary
2800	basic

ChampionUsesItem

championName	itemName
Gnar	Doran’s shield
Udyr	Doran’s shield
Lux	Doran’s ring

Jinx	Doran’s blade
Thresh	Sunfire cape

Tournament

tournamentID	name	dateStart	dateEnd	season
1	Worlds	01-01-2008	01-30-2008	1
2	Qualifiers	01-01-2009	01-11-2009	2
3	Worlds	01-01-2010	01-31-2010	3
4	Worlds	01-01-2011	01-31-2011	4
5	Regionals	01-01-2012	01-31-2012	5

Match

matchId	tournamentId	date	gameLength
1	1	01-01-2008	20
2	1	01-02-2008	40
3	1	01-03-2008	60
4	1	01-04-2008	100
5	1	01-05-2008	90

TeamPlaysMatch

winningTeamId	losingTeamId	tournamentId	matchID	winningTeamScore	losingTeamScore
1	2	1	1	9	0
3	4	1	2	15	4
1	5	1	3	13	5
1	3	1	4	4	3
2	4	1	5	10	7

TeamParticipatesTournament

tournamentId	teamId	ranking

1	1	1
1	2	3
1	3	2
1	4	4
1	5	5

RankingFurthestRound

ranking	furthestRound
1	Finals
2	Finals
3	Semifinals
4	Semifinals
5	Quarterfinals

Sponsor

companyName
SK Telecom
Telus
Cisco
Microsoft
NordVPN

SponsorSponsorsTeam

teamId	tournamentId	companyName	amount
1	1	SK Telecom	10,000.00
2	1	Telus	500,000.00
3	1	Cisco	20.00
4	1	Microsoft	5.00
5	1	NordVPN	1,000,000.00

8. Example queries

Insertion: Add a player to the player list

Deletion: Delete a champion from the champion list

Update: Update the cost of an item in the item list

Selection: Select matches from the match list where game length was longer than 60 minutes

Projection: Select name, username, and region from the list of persons

Join: Find the players that play ‘Lux’ and the items they bought on her (Join PlayerPlaysChampion and ChampionUsesItem where ChampionName is Lux)

Division: Find items that are bought by *every* player