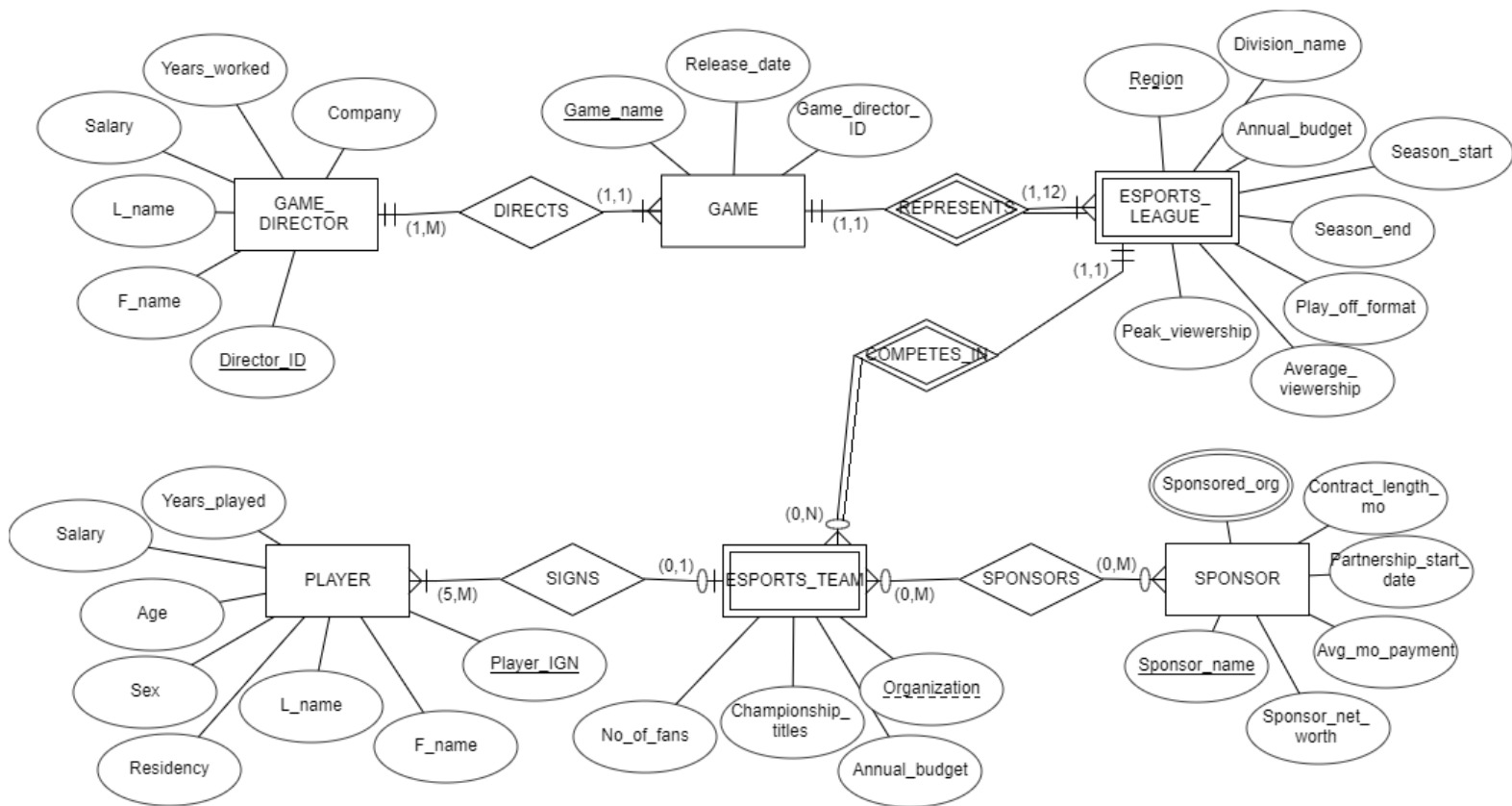
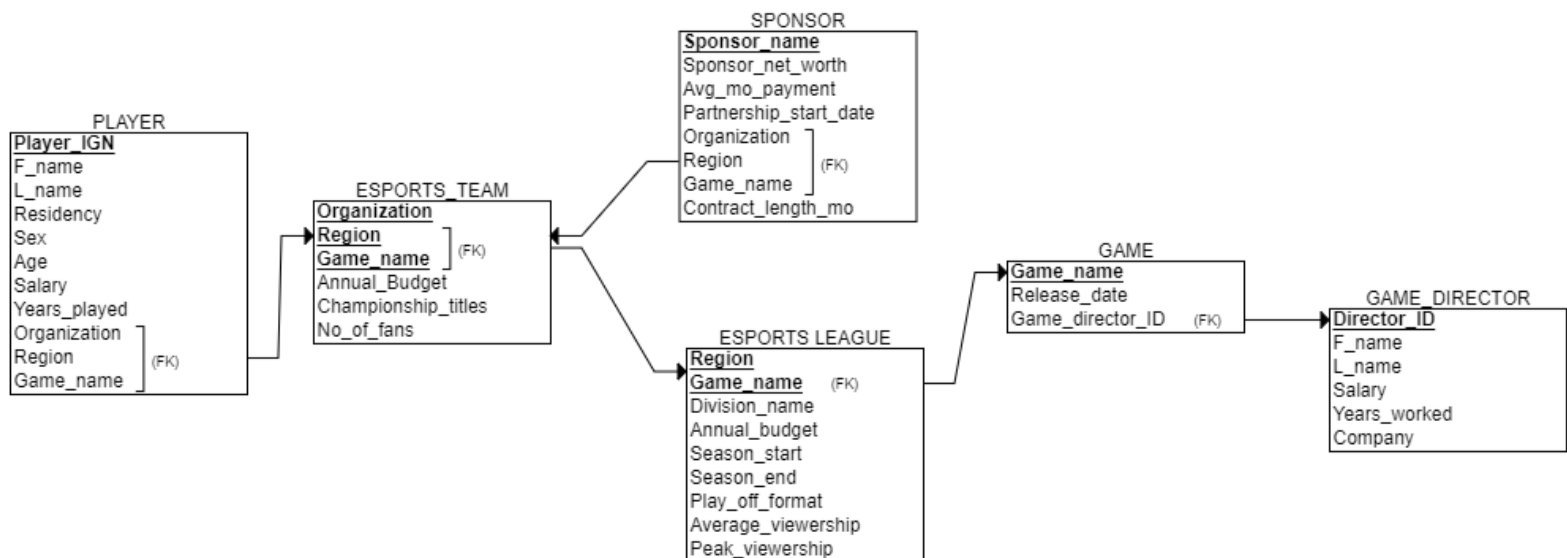


## Design Documentation



Assumptions: If a franchised league is newly formed, but hasn't announced their competing franchised teams yet, we would technically count that as an official ESPORTS LEAGUE having 0 teams.

## Relational Data Model



## SQL Statements

### SQL statements for tables

```
CREATE TABLE PLAYER
(
  Player_IGN VARCHAR(30) NOT NULL,
  F_name VARCHAR(15) NOT NULL,
  L_name VARCHAR(15) NOT NULL,
  Residency VARCHAR(30) NOT NULL,
  Sex CHAR(1) NOT NULL,
  Age INT NOT NULL,
  Salary INT NOT NULL,
  Years_played INT NOT NULL,
  Region VARCHAR(30) NOT NULL,
  Team_name VARCHAR(30) NOT NULL,
  PRIMARY KEY(Player_IGN),
  FOREIGN KEY(Region) REFERENCES ESPORTS_TEAM(Region),
  FOREIGN KEY(Team_name) REFERENCES ESPORTS_TEAM(Organization)
);
```

```
CREATE TABLE ESPORTS_TEAM
(
  Organization VARCHAR(30) NOT NULL,
  Region VARCHAR(30) NOT NULL,
  Annual_budget INT NOT NULL,
  Championship_titles INT NOT NULL,
  No_of_fans INT NOT NULL,
  Game_name VARCHAR(30) NOT NULL,
  PRIMARY KEY(Organization, Region, Game_name),
  FOREIGN KEY(Region) REFERENCES ESPORTS_LEAGUE(Region),
  FOREIGN KEY(Game_name) REFERENCES ESPORTS_LEAGUE(Game_name)
);
```

```
CREATE TABLE ESPORTS_LEAGUE
(
  Region VARCHAR(30) NOT NULL,
  Division_name VARCHAR(30) NOT NULL,
  Annual_budget INT NOT NULL,
  Season_start DATE,
  Season_end DATE,
  Play_off_format VARCHAR(15) NOT NULL,
  Average_viewership INT NOT NULL,
  Peak_viewership INT NOT NULL,
```

Anthony Vu, Dat Vo, Reagan Vu  
Final Project Report

```
Game_name VARCHAR(30) NOT NULL,  
PRIMARY KEY(Region, Game_name),  
FOREIGN KEY(Game_name) REFERENCES GAME(Game_name)  
);
```

```
CREATE TABLE SPONSOR  
(  
Sponsor_name VARCHAR(30) NOT NULL,  
Sponsor_net_worth INT NOT NULL,  
Avg_mo_payment INT NOT NULL,  
Partnership_start_date DATE,  
Contract_length_mo INT NOT NULL,  
Region VARCHAR(30) NOT NULL,  
Sponsored_org VARCHAR(30) NOT NULL,  
PRIMARY KEY(Sponsor_name),  
FOREIGN KEY(Sponsored_org) REFERENCES ESPORTS_TEAM(Organization)  
);
```

```
CREATE TABLE GAME  
(  
Game_name VARCHAR(30) NOT NULL,  
Release_date DATE,  
Game_director_ID INT NOT NULL,  
PRIMARY KEY(Game_name),  
FOREIGN KEY(Game_director_ID) REFERENCES GAME_DIRECTOR(Director_ID)  
);
```

```
CREATE TABLE GAME_DIRECTOR  
(  
Director_ID INT NOT NULL,  
F_name VARCHAR(15) NOT NULL,  
L_name VARCHAR(15) NOT NULL,  
Salary INT NOT NULL,  
Years_worked INT NOT NULL,  
Company VARCHAR(30) NOT NULL,  
PRIMARY KEY(Director_ID)  
);
```

### SQL statements for queries

SQL Statement	Purpose
SELECT F_name, L_name FROM PLAYER WHERE Organization = "100 Thieves";	Displays the first and last name of all players on a specific team.
SELECT Player_IGN FROM PLAYER WHERE F_name = "Woo-tae" AND L_name = "Park";	Displays the IGN of a specific player.
SELECT Organization, Game_name FROM ESPORTS_TEAM ORDER BY Game_name;	Displays the Organization name and the game(s) they are participating in.
SELECT avg(salary) FROM PLAYER WHERE Region = "NA";	Aggregate: Retrieves the average salary of all players in the NA region.
SELECT F_name, L_name FROM PLAYER WHERE Region = "KR";	Retrieves all the players that are playing in the korean region.
SELECT SUM(avg_mo_payment) FROM SPONSOR S JOIN ESPORTS_TEAM E ON E.Organization = S.Organization AND E.Region = S.Region AND E.Game_Name = S.Game_name WHERE E.Organization = "100 Thieves";	Grouping: Retrieves the total sum of monthly sponsored payment that is given to the organization 100 Thieves.
SELECT F_name, L_name FROM PLAYER P JOIN ESPORTS_TEAM E ON P.Organization = E.Organization AND P.Region = E.Region AND P.Game_name = E.Game_name, SPONSOR S ON S.Organization = E.Organization AND S.Region = E.Region AND S.Game_name = E.Game_name WHERE Sponsor_name = "Redbull";	Multiple Table: List players part of an organization sponsored by Redbull.

INSERT into PLAYER values ("Doublelift", "Yiliang", "Peng", "USA", "M", 29, 100000, 10, "NA", "100 Thieves", "League Of Legends");

INSERT into PLAYER values ("Closer", "Can", "Çelik", "TURKEY", "M", 24, 50000, 5, "NA", "100 Thieves", "League Of Legends");

INSERT into PLAYER values ("Bjergsen", "Bjerg", "Soren", "USA", "M", 26, 100000, 10, "NA", "100 Thieves", "League Of Legends");

INSERT into PLAYER values ("Busio", "Alan", "Cwalina", "USA", "M", 19, 55000, 2, "NA", "100 Thieves", "League Of Legends");

Anthony Vu, Dat Vo, Reagan Vu  
Final Project Report

INSERT into PLAYER values ("Tenacity", "Milan", "Oleksij", "USA", "M", 19, 35000, 3, "NA", "100 Thieves", "League Of Legends");

INSERT into ESPORTS\_TEAM values ("100 Thieves", "NA", "League Of Legends", 1000000, 01, 500000);

INSERT into PLAYER values ("Summit", "Woo-tae", "Park", "KOREA", "M", 24, 80000, 6, "NA", "Team Liquid", "League Of Legends");

INSERT into PLAYER values ("Pyosik", "Chang-hyeon", "Hong", "KOREA", "M", 22, 90000, 3, "NA", "Team Liquid", "League Of Legends");

INSERT into PLAYER values ("Haeri", "Harry", "Kang", "AUSTRALIA", "M", 21, 60000, 6, "NA", "Team Liquid", "League Of Legends");

INSERT into PLAYER values ("Yeon", "Sean", "Sung", "USA", "M", 20, 55000, 2, "NA", "Team Liquid", "League Of Legends");

INSERT into PLAYER values ("Corejj", "Yong-in", "Jo", "USA", "M", 28, 95000, 9, "NA", "Team Liquid", "League Of Legends");

INSERT into ESPORTS\_TEAM values ("Team Liquid", "NA", "League Of Legends", 9500000, 0, 450000);

INSERT into ESPORTS\_LEAGUE values ("NA", "League of Legends", "LCS", 6000000, "2022-02-05", "2023-04-24", "Single-Elimination", 200000, 255000);

INSERT into PLAYER values ("Canna", "Chang-dong", "Kim", "KOREA", "M", 22, 55000, 3, "KR", "DWG KIA", "League Of Legends");

INSERT into PLAYER values ("Canyon", "Geon-bu", "Kim", "KOREA", "M", 21, 100000, 4, "KR", "DWG KIA", "League Of Legends");

INSERT into PLAYER values ("ShowMaker", "Su", "Heo", "KOREA", "M", 22, 110000, 5, "KR", "DWG KIA", "League Of Legends");

INSERT into PLAYER values ("Deft", "Hyuk-kyu", "Kim", "KOREA", "M", 26, 120000, 9, "KR", "DWG KIA", "League Of Legends");

INSERT into ESPORTS\_TEAM values ("DWG KIA", "KR", "League of Legends", 9000000, 2, 900000);

INSERT into PLAYER values ("Zeus", "Woo-je", "Choi", "KOREA", "M", 18, 50000, 3, "KR", "T1", "League Of Legends");

INSERT into PLAYER values ("Oner", "Hyeon-joon", "Moon", "KOREA", "M", 19, 50000, 5, "KR", "T1", "League Of Legends");

INSERT into PLAYER values ("Faker", "Sang-hyeok", "Lee", "KOREA", "M", 26, 5000000, 12, "KR", "T1", "League Of Legends");

INSERT into PLAYER values ("Gumayusi", "Min-hyeong", "Lee", "KOREA", "M", 20, 60000, 2, "KR", "T1", "League Of Legends");

Anthony Vu, Dat Vo, Reagan Vu  
Final Project Report

INSERT into PLAYER values ("Keria", "Min-seok", "Ryu", "KOREA", "M", 20, 50000, 3, "KR", "T1", "League Of Legends");

INSERT into ESPORTS\_TEAM values ("T1", "KR", "League Of Legends", 7000000, 12, 500000);

INSERT into ESPORTS\_LEAGUE values ("KR", "League Of Legends", "LCK", 5000000, "2023-01-11", "2023-03-30", "Double-Elimination", 800000, 955000);

INSERT into PLAYER values ("Bang", "Sean", "Bezerra", "USA", "M", 18, 50000, 2, "NA", "100 Thieves", "Valorant");

INSERT into PLAYER values ("Stellar", "Brenden", "McGrath", "USA", "M", 20, 55000, 2, "NA", "100 Thieves", "Valorant");

INSERT into PLAYER values ("Cryocells", "Matthew", "Panganiban", "USA", "M", 19, 125000, 1, "NA", "100 Thieves", "Valorant");

INSERT into PLAYER values ("Derrek", "Derrek", "Ha", "USA", "M", 22, 80000, 2, "NA", "100 Thieves", "Valorant");

INSERT into PLAYER values ("Asuna", "Peter", "Mazuryk", "USA", "M", 20, 90000, 2, "NA", "100 Thieves", "Valorant");

INSERT into ESPORTS\_TEAM values ("100 Thieves", "NA", "Valorant", 1000000, 0, 50000);

INSERT into PLAYER values ("leaf", "Nathan", "Orf", "USA", "M", 19, 70000, 2, "NA", "Cloud 9", "Valorant");

INSERT into PLAYER values ("Xeppaa", "Erick", "Bach", "USA", "M", 22, 58000, 2, "NA", "Cloud 9", "Valorant");

INSERT into PLAYER values ("Vanity", "Anthony", "Malaspina", "USA", "M", 24, 65000, 2, "NA", "Cloud 9", "Valorant");

INSERT into PLAYER values ("Zellsis", "Jordan", "Montemurro", "USA", "M", 24, 85000, 2, "NA", "Cloud 9", "Valorant");

INSERT into PLAYER values ("yay", "Jacob", "Whiteaker", "USA", "M", 24, 150000, 2, "NA", "Cloud 9", "Valorant");

INSERT into ESPORTS\_TEAM values ("Cloud 9", "NA", "Valorant", 1250000, 0, 62000);

INSERT into ESPORTS\_LEAGUE values ("NA", "Valorant", "VCT-NA", 500000, "2023-01-11", "2023-03-30", "Double-Elimination", 100000, 125000);

INSERT into PLAYER values ("stax", "Gu-taek", "Kim", "KOREA", "M", 22, 70000, 2, "KR", "DRX", "Valorant");

INSERT into PLAYER values ("Rb", "Sang-Min", "Goo", "KOREA", "M", 21, 55000, 2, "KR", "DRX", "Valorant");

INSERT into PLAYER values ("Zest", "Gi-seok", "Kim", "KOREA", "M", 21, 41000, 1, "KR", "DRX", "Valorant");

Anthony Vu, Dat Vo, Reagan Vu  
Final Project Report

INSERT into PLAYER values ("BuZz", "Byung-chul", "Yu", "KOREA", "M", 19, 73000, 2, "KR", "DRX", "Valorant");

INSERT into PLAYER values ("MaKo", "Myeong-kwan", "Kim", "KOREA", "M", 20, 52000, 2, "KR", "DRX", "Valorant");

INSERT into ESPORTS\_TEAM values ("DRX", "KR", "Valorant", 300000, 0, 650000);

INSERT into ESPORTS\_LEAGUE values ("KR", "Valorant", "VCT-KR", 600000, "2023-02-11", "2023-04-30", "Double-Elimination", 60000, 85000);

INSERT into SPONSOR values ("Redbull", 2740000000, 55000, "2022-10-30", "100 Thieves", "NA", "Valorant", 15);

INSERT into SPONSOR values ("Tera", 122400000, 5000, "2022-06-30", "100 Thieves", "NA", "Valorant", 5);

INSERT into SPONSOR values ("Logitech", 9790000000, 44000, "2020-03-16", "T1", "KR", "League Of Legends", 20);

INSERT into GAME values ("League Of Legends", "2009-10-27", 01);

INSERT into GAME values ("Valorant", "2020-06-02", 02);

INSERT into GAME\_DIRECTOR values (01, "Andrei", "Roon", 3000000, 8, "Riot Games");

INSERT into GAME\_DIRECTOR values (02, "Joe", "Ziegler", 2000000, 2, "Riot Games");

## Normal Form

Player\_IGN -> F\_name, L\_name, Residency, Sex, Age, Salary, Years\_played, Organization, Region, Game\_Name

PLAYER:

1NF: All values in the PLAYER table holds only atomic values and there is a primary key that can derive all values in the table. PLAYER table is in 1NF.

2NF: The PLAYER table's primary-key is single-valued so it cannot violate 2NF.

Therefore PLAYER table satisfies 2NF.

3NF: No non-prime value in the PLAYER table can derive another non-prime value.

Therefore the PLAYER table satisfies 3NF.

BCNF: None of the non-prime keys can derive the prime-keys in the PLAYER table, therefore the PLAYER table satisfies BCNF.

Organization, Region, Game\_name -> Annual\_Budget, Championship\_titles, No\_of\_fans, Region

ESPORTS\_TEAM:

1NF: Only atomic values and a primary-key exists so we satisfy 1NF.

2NF: None of the partial primary keys can derive any other value in this table.

Therefore we satisfy 2NF for this table.

3NF: None of the non-prime attributes can derive another non-prime attribute within this table so we satisfy 3NF

BCNF: None of the non-prime attributes can derive any prime-attributes, therefore this table satisfies BCNF

Sponsor\_name -> Sponsor\_net\_worth, Avg\_mo\_payment, Partnership\_start\_date, Organization, Region, Game\_name, Contract\_length\_mo

SPONSOR:

1NF: Only atomic values and a primary key exists so satisfies 1NF.

2NF: Only one attribute in primary key so is in 2NF.

3NF: Cannot derive any non-prime from another non-prime so satisfies 3NF.

BCNF: Cannot derive prime attribute from any non-prime attributes so we satisfy BCNF.

Region, Game\_Name -> Division\_name, Annual\_budget, Season\_start, Season\_end, Play\_off\_format, Average\_viewership, Peak\_viewership

ESPORTS\_LEAGUE:

1NF: Only atomic values and has primary key, satisfies 1NF.

2NF: Neither Region nor Game\_name can derive any attribute/s within this table so we can say that we satisfy 2NF.

3NF: None of the non-prime can derive another non-prime so we satisfy 3NF.

BCNF: None of the non-prime can derive any prime attributes so we satisfy BCNF.

Game\_name -> Release\_date, Game\_director\_ID

GAME:



Anthony Vu, Dat Vo, Reagan Vu  
Final Project Report

1NF: Only atomic and has Primary key so satisfies 1NF.

2NF: Only 1 attribute for Primary Key so satisfies 2NF.

3NF: Neither of the non-primes can derive each other so satisfies 3NF.

BCNF: Neither of the non-primes can derive the prime attribute so satisfies BCNF.

Could argue that we can find the Game\_name from Director, but we assume one director can lead multiple games.

Director\_ID -> F\_name, L\_name, Salary, Years\_worked, Company GAME\_DIRECTOR:

1NF: Only atomic attributes and has Primary key, satisfies 1NF.

2NF: Single attribute Primary key satisfies 2NF.

3NF: None of the non-prime can derive another non-prime, satisfies 3NF.

BCNF: None of the non-prime can derive the prime attribute so satisfies BCNF.

From this, we can see that every table we have satisfies BCNF. This was intentional as we made sure we got rid of any violation of any of these normal forms while creating the tables.

## Project Evaluation

There was a considerable amount of effort put into this group project that went into learning the course material, planning sessions to work with each other, and changing/adapting to certain aspects of our project. Overall, our group would say that we're satisfied with the product we produced and are proud of the time, energy, and effort we put into it.

From our initial proposal, we've kept the same application concept, but we broadened it out more to be more inclusive and give a more holistic view of the application area's ecosystem. Due to this, we also took out some attributes, added some new ones in, and replaced one of our initial tables with two new ones. We adjusted our queries, PKs, and FKs accordingly.

In terms of things that went right, we all unanimously agreed to do a field that we were all familiar with and were really flexible in our willingness to work on certain requirements tasked by the project. Additionally, for the project itself, the initial design of our ER diagram and relational data model were pretty helpful in conceptualizing our project and we were very comfortable doing all the SQL work.

On the other hand, there were a number of things that "went wrong" or could be improved upon. Firstly, we realized that the more detailed we made our project, the less sense it made, which means there were plenty of times where we had to scrap certain attributes, or reassign other attributes to be a table's primary key. This would cause a chain reaction where we would also have to edit our relational model, SQL tables, and SQL statements/queries. This meant that even though we kept on track with our outlined timeline, we would still have to contribute a significant amount of time and energy to make the necessary changes and adjustments.

If we did this project again, the things we would change would be to more thoroughly plan out our ER diagram and relational data model before we make them. This would save us a lot of time and effort when producing our final product and would reduce redundancy in our work.

We would also like to briefly mention that our group lost a team member during the second half of the quarter which meant our now three person group had to adjust to the new circumstances that come with losing a team member.