University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 2

Date: October 25th, 2021

Group Number: 62

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Adam Chesshire	21145636	f4z2b	adam_chesshire@hotmail.com
Abigail Moraes	51407138	l8k1b	amoraes@student.ubc.ca
Allen Jiang	90858895	o2z2b	allenjiang01@outlook.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

ER Diagram Changes

Several changes were made to the ER diagram from milestone 1 and are illustrated below in Figure 1. The driverID and PhoneNumber attributes were added to the Spectators entity set to allow for identification of attendees at esports events. Security will be able to match the ID to the ticket to prevent the use of stolen or lost tickets. The Spectator entity set now attends the tournament rather than an aggregation with the attends relation. The spectators buy tickets to one or more tournaments.

The aggregation that originally included the Organization, Tournament, Sponsor, Game entities was reduced to include tournament and game. Teams participate in and broadcastServices broadcast one or more tournaments involving certain games while spectators attend one or more tournaments specifically.

The Employees entity was converted to a weak entity since employeeID is more of a partial key. Different organizations may use the same employee IDs. The orgName together with the employee ID would provide a more effective Primary Key. The employeeID attribute was converted to a partial key.

The Broadcast Service has two more important attributes, Language and Country to indicate the languages and countries in which the service is available. The sponsor entity set has additional attributes city, Location and TimeZone which would allow the organization to organize meetings with international sponsors.

The Game entity was modified to facilitate a more meaningful ISA relationship. The attribute numParticipants was removed from each of the subclasses and into the superclass, Game. In addition, Game has the attributes gameName, numParticipants and the boolean crossPlatFormPlay which indicates whether individuals can play the game on PC and one or more consoles. Attributes unique to each subclass were added. The numMaps attribute indicates the number of maps available for an FPS game, the number of characters available in the character select screen are represented by numCharacters attribute in fightingGame, the DLCincuded attribute indicates whether a strategyGame will have additional content and MOBA indicates the number of champions available with the numChampions attribute.
In addition, the teamColours attribute was removed from Team since it is not as essential to esports organization as it is in other sports.

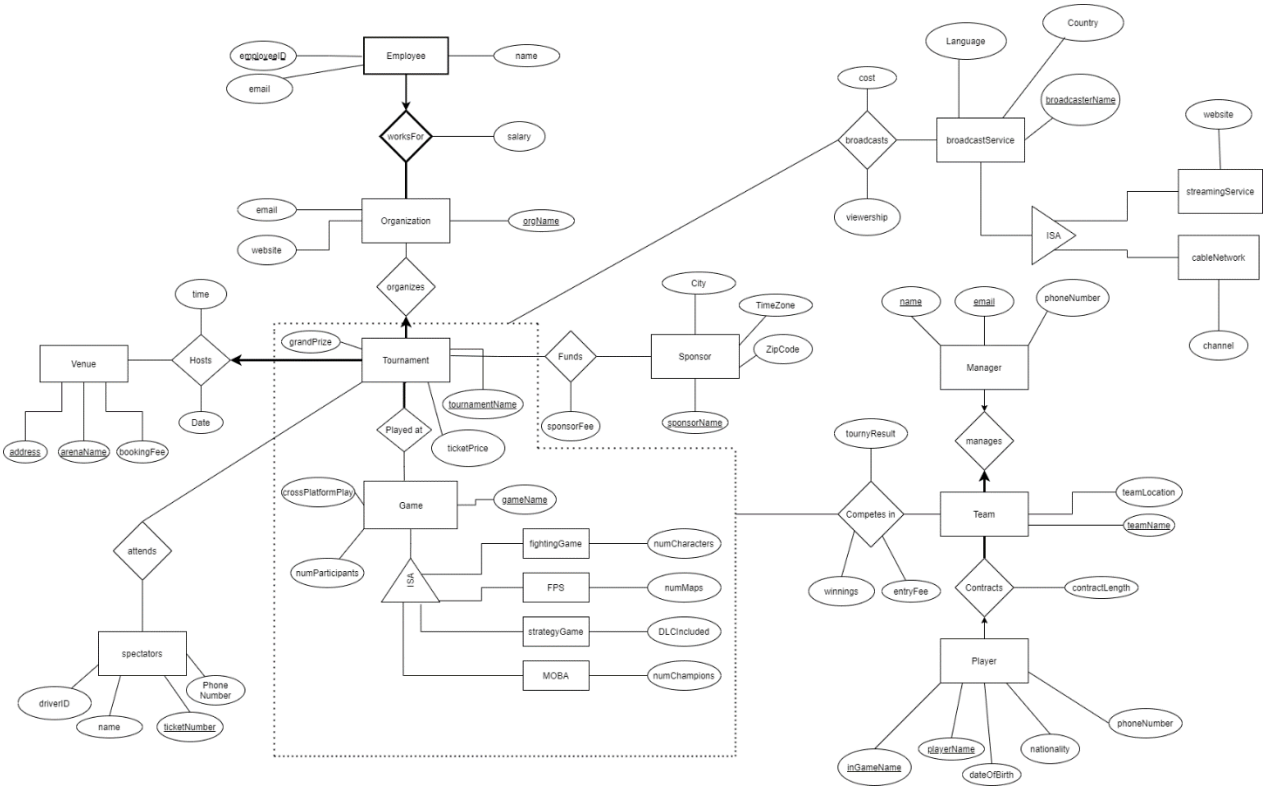

Fig 1. **Updated Esports Organization ER Diagram.** The aggregation was reduced to include tournament and game entity sets. Employee is now a weak entity that depends on the owner entity Organization. Attributes driverID and phoneNumber were added to the spectators entity set. The attends relationship now permits a spectator to attend one or more tournaments rather than the aggregation as a whole. Language and Country attributes were added to broadcast service. The isA relationship for the Game entity was modified. Game has additional attributes crossPlatformPlay and numParticipants. Subclasses have subclass specific attributes like numCharacters for fightGame, numMaps for FPS, etc. The sponsor entity has additional attributes City, TimeZone and ZipCode.

Relational Schema

Details about the relational schema derived from the ER diagram including the table definitions, primary keys, candidate keys and other constraints are listed below in Table 1. Table 1 indicates the relational schema before normalization.

Table 1. Relational Schema

Table Definition	Primary Key	Candidate Keys	Foreign Keys	Constraints
Spectators (name: Char(40), ticketNumber: Integer, driverID: Integer, phoneNumber: Char(20))	ticketNumber	ticketNumber	None	None
Attends (**ticketNumber**: Integer, **tournamentName**: Char(40), name: Char(40))	**(ticketNumber, tournamentName)**	**(ticketNumber, tournamentName)**	**ticketNumber REFERENCES Spectators (ticketNumber), tournamentName REFERENCES Tournament (tournamentName)**	None
Player(inGameName: Char(40), playerName: Char(40), dateOfBirth: Char(40), nationality: Char(20), **teamName**: Char(30), contractLength: Integer, phoneNumber: Char(20))	(inGameName, playerName)	(inGameName, playerName)	**teamName REFERENCES Team(teamName)**	None
Team(teamLocation: char(30), teamName: char(30), **managerName**: char(40), **managerEmail**: char(40))	teamName	teamName	**(managerName, managerEmail) REFERENCES Manager (managerName, managerEmail)**	managerName is NOT NULL managerEmail is Not NULL
Manager(name: char(40), email: char(40), phoneNumber: char(15))	(name, email)	(name, email)	None	None
CompetesIn(**gameName** : char(50), **tournamentName** : char(40), **teamName**: char(30), entryFee: Integer, winnings: Integer, tournyResult: Integer)	**(gameName, tournamentName, teamName)**	**(gameName, tournamentName, teamName)**	**gameName REFERENCES Game(gameName) tournamentName REFERENCES Tournament (tournamentName) teamName REFERENCES Team(teamName)**	None
Tournament(tournamentName: Char(40), ticketPrice: Integer, grandPrize: Integer, **orgName**: Char(40), **address**: Char(40), **arenaName**: Char(40), date: Char(20), time: Char(5))	tournamentName	tournamentName	**orgName REFERENCES Organization (orgName), (arenaName, address) REFERENCES Venue (address, arenaName)**	**orgName is NOT NULL address is NOT NULL arenaName is NOT NULL**
Game(gameName: Char(50), numParticipants: Integer, crossPlatformPlay: Boolean)	gameName	gameName	**None**	**None**
MOBA(**gameName**: Char(50), numChampions: Integer)	**gameName**	**gameName**	**gameName REFERENCES Game(gameName)**	**None**

fightingGame (**gameName**: Char(50), numCharacters : Integer)	**gameName**	**gameName**	**gameName REFERENCES Game(gameName)**	**None**
FPS(**gameName**: Char(50), numMaps: Integer)	**gameName**	**gameName**	**gameName REFERENCES Game(gameName)**	**None**
strategyGame(**gameName**: Char(50), DLCIncluded: Boolean)	**gameName**	**gameName**	**gameName REFERENCES Game(gameName)**	**None**
PlayedAt (**tournamentName**: Char(40), **gameName**: Char(50), bookingFee: Integer)	**(tournamentName, gameName)**	**(tournamentName, gameName)**	**gameName REFERENCES Game(gameName) tournamentName REFERENCES Tournament (tournamentName)**	**None**
Venue(address: Char(40), arenaName: Char(40), bookingFee: Integer)	(address, arenaName)	(address, arenaName)	None	None
Sponsor(sponsorName: Char(50), zipCode: Char(20), city: char(30), timezone: Char(30))	sponsorName	sponsorName	None	None
Funds(**tournamentName**: Char(40), **sponsorName**: Char(50), sponsorFee: Integer)	**(tournamentName, sponsorName)**	**(tournamentName, sponsorName)**	**tournamentName REFERENCES Tournament (tournamentName) sponsorName REFERENCES Sponsor (sponsorName)**	None
Organization(orgName: Char(40), email: Char(40), website: Char(50))	orgName	orgName	None	None
Employee(employeeID: Integer, email: Char(40), name: char(40), **orgName**: Char(40))	(employeeID, **orgName**)	(employeeID, **orgName**)	**orgName REFERENCES Organization (orgName)**	orgName ON DELETE CASCADE
CableNetwork(channel : Char(40), **broadcasterName**: char(40))	(channel, **broadcasterName**)	(channel, **broadcasterName**)	**broadcasterName REFERENCES BroadcastService (broadcasterName)**	None
StreamingService(website: Char(40), **broadcasterName**: char(40))	(website, **broadcasterName)**	(website, **broadcasterName)**	**broadcasterName REFERENCES BroadcastService (broadcasterName)**	None
BroadcastService(broadcasterName: Char(40), country: Char(30), language: Char(20))	broadcasterName	broadcasterName	None	None
Broadcasts(**broadcasterName**: Char(40), **tournamentName**: Char(40), **gameName**: Char(50), cost: Integer, viewership: Integer)	**(broadcasterName, tournamentName, gameName)**	**(broadcasterName, tournamentName, gameName)**	**broadcasterName REFERENCES BroadcastService (broadcasterName) tournamentName REFERENCES Tournament (tournamentName) gameName REFERENCES Game(gameName)**	None

Foreign Keys : Bolded, Primary Keys or components of primary keys: Underlined

Functional Dependencies and Decomposition with Populated Tables

The two decompositions were performed on Spectators and Sponsor are listed first. The other relations were left unaltered and are listed with their respective functional dependencies after the

Decomposition on Spectators

Before Decomposition

Spectators			
Name	TicketNumber	driverID	phoneNumber
Peter Parker	8930	1239076	778-123-3214
Natasha Romanov	9730	1257784	434-436-7654
Steve Rogers	2325	4367654	982-543-9685
Wanda Maximoff	2313	3237894	234-654-7655
Peggy Carter	4366	5790753	234-859-4534

Functional Dependencies:
1. TicketNumber → ALL
2. DriverID → Name, phoneNumber

Functional dependency 2 violates BCNF since DriverID is not a superkey.

Decomposition Step 1: decompose on DriverID → Name, phoneNumber

After Decomposition

spectatorsInfo		
Name	driverID	phoneNumber
Peter Parker	1239076	778-123-3214
Natasha Romanov	1257784	434-436-7654
Steve Rogers	4367654	982-543-9685
Wanda Maximoff	3237894	234-654-7655
Peggy Carter	5790753	234-859-4534

Functional Dependencies
DriverID → phoneNumber, Name

spectatorsTicket	
TicketNumber	**driverID (candidate key)**
8930	1239076
9730	1257784
2325	4367654
2313	3237894
4366	5790753

Functional Dependencies:
TicketNumber → DriverID

Decomposition on Sponsors

Before Decomposition

Sponsor			
sponsorName	zipCode	city	timezone
Justice League	s3e 2sg	Vancouver	Pacific Daylight Time
The Avengers	h5r 1g6	Seattle	Pacific Daylight Time
The Martell Foundation	y7i 5f9	Calgary	Mountain Daylight Time
RNG	v2q 7u8	Vancouver	Pacific Daylight Time
Edifier	v5e 2z4	Vancouver	Pacific Daylight Time

Functional Dependencies:
1. sponsorName → ALL
2. zipCode, city→ timezone
3. zipCode → city

Functional dependency 2 violates BCNF since zipCode,city is not a superkey.

Decomposition step 1: decompose on zipCode, city → timezone
After Decomposition

sponsorLocation		
zipCode	city	timezone
s3e 2sg	Vancouver	Pacific Daylight Time
h5r 1g6	Seattle	Pacific Daylight Time
y7i 5f9	Calgary	Mountain Daylight Time
v2q 7u8	Vancouver	Pacific Daylight Time
v5e 2z4	Vancouver	Pacific Daylight Time

Functional Dependencies
zipCode, city → timezone

sponsorInfo		
sponsorName	**zipCode**	**city**
Justice League	s3e 2sg	Vancouver
The Avengers	h5r 1g6	Seattle
The Martell Foundation	y7i 5f9	Calgary
RNG	v2q 7u8	Vancouver
Edifier	v5e 2z4	Vancouver

Functional Dependencies
sponsorName → zipCode, city
zipCode → city

Other relations and Functional dependencies

Organization		
orgName	email	website
Riot.inc	league@riot.com	https://www.riotgames.com/en
Valve	valve@steam.com	https://www.valvesoftware.com/en/
Electronic Arts	ea@outlook.com	https://www.ea.com/en-ca
Tencent	tencent@qq.com	https://www.tencent.com/
NetEase	wangyi@163.com	https://www.neteasegames.com/

Functional Dependencies
orgName → email, website

Employee			
employeeID	email	name	**OrgName**
12321	js@westeros.com	Jon Snow	Riot.inc
43543	tl@westeros.com	Tywin Lannister	Riot.inc
42323	ss@westeros.com	Sansa Stark	Valve
12341	om@westeros.com	Oberyn Martell	Tencent
88726	dt@westeros.com	Daenerys Targaryen	NetEase

Functional Dependencies
employeeID, OrgName → email, name

Venue		
address	arenaName	bookingFee
Mercedes-Platz 2	Verti Music Hall	60,000
1 National Stadium S Rd	Beijing National Stadium	100,000
101 Denmark way	Odense Congress Center	30,000
10-1 Kasumigaokamachi	Japan National Stadium	80,000
2576 Korea Rd	LCK Arena	10,000

Functional Dependencies
address, arenaName → bookingFee
Address → arenaName

Tournament

TournamentName	ticketPrice	grandPrize	**orgName**	**address**	**arenaName**	date	time
Berlin Masters	200	600,000	Riot.inc	Mercedes-Platz 2	Verti Music Hall	01-11-2021	13:00
Worlds 2021	300	2,000,000	Riot.inc	1 National Stadium S Rd	Beijing National Stadium	05-10-2021	18:00
ESL Invitational 2021	200	1,000,000	Valve	101 Denmark way	Odense Congress Center	20-02-2021	9:00
Tournament of Power	100	100,000	Electronic Arts	10-1 Kasumigaokamachi	Japan National Stadium	20-05-2021	16:00
Faker Cup	100	100,000	Riot.inc	2576 Korea Rd	LCK Arena	13-01-2021	18:00

Functional Dependencies
TournamentName → ALL
orgName, arenaName, date, time, address → tournamentName

MOBA

gameName	numChampions
League of legends	8
DOTA	5
Smite	5
Heroes of the Storm	3
Vainglory	8

Functional Dependencies
gameName → numChampions

fightingGame

gameName	numCharacters
Street Fighters 10	21
League of Legos	101
Dota Fighters	78
Tekken 8	17
Goru Goru	18

Functional Dependencies
gameName → numCharacters

FPS	
gameName	numMaps
CS: Return	6
Valorant 2	4
Overwatch 3	6
Call of Duty: Duty Calls	12
Battlefield 2022	10

Functional Dependencies
gameName → numMaps

strategyGames	
gameName	DLCIncluded
Age of Empires: Conquest	True
StarCraft 3	True
Warhammer 3000	False
Clash of Clans 2	False
Warcraft 2020	False

Functional Dependencies
gameName → DLCIncluded

Game		
gameName	numParticipants	crossPlatformPlay
Age of Empires: Conquest	8	TRUE
StarCraft 3	32	TRUE
Warhammer 3000	8	FALSE
Clash of Clans 2	32	FALSE
Warcraft 2020	32	FALSE
CS: Return	12	TRUE
Valorant 2	12	TRUE
Overwatch 3	12	FALSE
Call of Duty: Duty Calls	18	TRUE
Battlefield 2022	18	FALSE
Street Fighters 10	81	TRUE
League of Legos	64	FALSE
Dota Fighters	32	TRUE
Tekken 8	32	TRUE
Goru Goru	32	FALSE
League of legends	50	TRUE
DOTA	20	FALSE
Smite	100	TRUE
Heroes of the Storm	35	TRUE
Vainglory	15	TRUE

Functional Dependencies
gameName → numParticipants, crossPlatformPlay

Broadcasts				
broadcasterName	**tournamentName**	**gameName**	cost	viewership
ESL	ESL Invitational 2021	Age of Empires: Conquest	8000	1,320
Bilibili	Berlin Masters	Valorant 2	60,000	1,000,000
Nerd Street Gamers	Worlds 2021	League of Legos	100,000	2,000,000
ESL	ESL Invitational 2021	Warcraft 2020	10,000	1,143,002
ESPN	Tournament of Power	Clash of Clans 2	8000	897,975

Functional Dependencies
broadcasterName, tournamentName, gameName → cost, viewership

Funds

tournamentName	**sponsorName**	sponsorFee
Berlin Masters	Justice League	2,000
Worlds 2021	Justice League	5,000
Berlin Masters	The Avengers	2,000
Berlin Masters	RNG	2,000
Faker Cup	Edifier	1,000

Functional Dependencies
tournamentName, sponsorName → sponsorFee

Played at

tournamentName	**gameName**	tier
Berlin Masters	CS: Return	A
Worlds 2021	League of Legos	S
Faker Cup	League of Legos	B
Berlin Masters	Valorant 2	A
Tournament of Power	Street Fighters 10	A

Functional Dependencies
tournamentName, gameName → tier

Competes in					
gameName	**tournamentName**	**teamName**	entryFee	winnings	tournyResult
CS: Return	ESL Invitational 2021	Cloud 9	1,000	0	5
League of Legos	Worlds 2021	T1	8,000	80,000	1
League of Legos	Worlds 2021	Atlanta FaZe	8,000	0	6
League of Legos	Worlds 2021	Scarlett	8,000	20,000	3
Tekken 8	Tournament of Power	Punk	200	0	4

Functional Dependencies
teamName, gameName, tournamentName → ALL

attends		
ticketNumber	**tournamentName**	name
12211	Worlds 2021	Jim Jones
43534	Worlds 2021	Kim Kardashian
13451	Worlds 2021	Khloe Kardashian
65422	Worlds 2021	Frank Lim
15533	Berlin Masters	Poppy Green

Functional Dependencies
ticketNumber, tournamentName → name

BroadcastService		
broadcasterName	country	language
Valve TV	United States	English
Ginx TV	England	English
Riot Games	Japan	Japanese
ESPN	United States	English
HBO	United States	English
Nerd Street Gamers	Canada	French
ESL	Canada	English
AWL	South Korea	Korean
ESLT	China	Cantonese
Bilibili	China	Mandarin

Functional Dependencies
broadcasterName → country, language

Player						
inGameName	playerName	dateOfBirth	nationality	phoneNumber	**teamName**	contractLength(Years)
Punk	VictorWoodley	29-Jul-1998	United States	234-398-2898	Punk	2
Faker	Sang-hyeok	7-May-1996	South Korea	984-590-3090	T1	6
Gumayusi	Lee Min-hyeong	6-Feb-2002	South Korea	878-983-2900	T1	3
Scarlett	Sasha Hostyn	14-Dec-1993	Canada	234-589-4903	Scarlett	2
Simp	Chris Lehr	6-Feb-2001	United States	234-456-1290	Atlanta FaZe	5

Functional Dependencies:
inGameName, playerName → ALL

Manager		
name	email	phoneNumber
Bob Ross	bob.ross@gmail.com	123-445-6546
Amber Howard	ahoward@a3a.com	435-342-4309
Nick Garrett	ngarett@amg.com	455-654-2342
Peter Letz	pletz@caa.com	343-564-4656
Andrew Tomlinson	atomlin@csa.com	545-234-1231

Functional Dependencies
name, email → phoneNumber

cableNetwork	
channel	**broadcasterName**
CCTV 1	Valve TV
CCTV 2	Ginx TV
CCTV 3	Riot Games
ESPN	ESPN
HBO Live	HBO

Functional Dependencies:
broadcasterName → channel

streamingService	
website	broadcasterName
Twitch.com	Nerd Street Gamers
Twitch.com	ESL
Twitch.com	AWL
Youtube.com	ESLT
Bilibili.com	Bilibili

Functional Dependencies:
BroadcasterName → website

Team			
teamLocation	teamName	**managerName**	**managerEmail**
California, USA	Punk	Bob Ross	bob.ross@gmail.com
Seoul, Korea	T1	Amber Howard	ahoward@a3a.com
Shanghai, China	Atlanta FaZe	Nick Garrett	ngarett@amg.com
Tokyo, Japan	Scarlett	Peter Letz	pletz@caa.com
Melbourne, Australia	Cloud 9	Andrew Tomlinson	atomlin@csa.com

Functional Dependencies
teamName → ALL
managerName, managerEmail → teamName

Table 2. Relational Schema - After Decomposition

Table Definition	Primary Key	Candidate Keys	Foreign Keys	Constraints
Spectators(name:Char(40), ticketNumber: Integer, driverID: Integer, phoneNumber: Char(20))	ticketNumber	ticketNumber	None	None
SpectatorInfo(driverID: Integer, name: Char(40), phoneNumber: Char(20)	driverID	driverID	None	None
spectatorTicket(ticketNumber: Integer, **driverID**: Integer)	ticketNumber	ticketNumber **driverID**	**driverID**	None
Attends(**ticketNumber**: Integer, **tournamentName**: Char(40), name: Char(40))	**(ticketNumber,** **tournamentName)**	**(ticketNumber,** **tournamentName)**	**ticketNumber REFERENCES Spectators(ticketNumber),** **tournamentName REFERENCES Tournament(tournamentName)**	None
Player(inGameName:Char(40), playerName: Char(40), dateOfBirth: Char(40), nationality: Char(20), **teamName**: Char(30), contractLength: Integer, phoneNumber: Char(20))	(inGameName, playerName)	(inGameName, playerName)	**teamName** **REFERENCES** **Team(teamName)**	None
Team(teamLocation: char(30), teamName: char(30), **managerName**: char(40), managerEmail: char(40))	teamName	teamName	**managerName**	managerName is NOT NULL managerEmail is Not NULL

Manager(name: char(40), email: char(40), phoneNumber: char(15))	(name,email)	(name,email)	None	None
CompetesIn(**gameName** : char(50), **tournamentName** : char(40), **teamName**: char(30), entryFee: Integer, winnings: Integer, tournyResult: Integer)	(**gameName,tournamentName,teamName**)	(**gameName,tournamentName,teamName**)	**gameName REFERENCES Game(gameName)** **tournamentName REFERENCES Tournament(tournamentName)** **teamName REFERENCES Team(teamName)**	None
Tournament(tournamentName: Char(40), ticketPrice: Integer, grandPrize: Integer, **orgName**: Char(40), **address**: Char(40), **arenaName**: Char(40), date: Char(20), time: Char(5))	tournamentName	tournamentName	**orgName REFERENCES** **Organization(orgName),** **(arenaName, address) REFERENCES Venue(address, arenaName)**	**None**
Game(gameName: Char(50), numParticipants: Integer, crossPlatformPlay: Boolean)	gameName	gameName	None	**None**
MOBA(**gameName**:Char(50), numChampions: Integer)	**gameName**	**gameName**	**gameName REFERENCES Game(gameName)**	**None**
fightingGame(**gameName**: Char(50), numCharacters : Integer)	**gameName**	**gameName**	**gameName REFERENCES Game(gameName)**	**None**
FPS(**gameName**: Char(50), numMaps: Integer)	**gameName**	**gameName**	**gameName REFERENCES Game(gameName)**	**None**

strategyGame(**gameName**: Char(50), DLCIncluded: Boolean)	**gameName**	**gameName**	**gameName REFERENCES Game(gameName)**	**None**
PlayedAt(**tournamentName**: Char(40), **gameName**: Char(50), bookingFee: Integer)	**(tournamentName,gameName)**	**(tournamentName, gameName)**	**gameName REFERENCES Game(gameName)** **tournamentName** **REFERENCES** **Tournament(tournamentName)**	**None**
Venue(address: Char(40), arenaName: Char(40), bookingFee: Integer)	(address, arenaName)	(address, arenaName)	None	None
Sponsor(sponsorName: Char(50), zipCode: Char(20), city: char(30), timezone: Char(30))	sponsorName	sponsorName	None	None
SponsorLocation(zipCode: Char(20), city: Char(30), timezone: Char(30))	(zipCode, city)	(zipCode, city)	None	None
SponsorInfo(sponsorName: Char(50), **zipCode**: Char(20), **city**: Char(30))	sponsorName	sponsorName	**zipCode, city REFERENCES SponsorLocation(zipCode, city)**	None
Funds(**tournamentName**: Char(40), **sponsorName**: Char(50), sponsorFee: Integer)	**(tournamentName, sponsorName)**	**(tournamentName, sponsorName)**	**tournamentName** **REFERENCES** **Tournament(tournamentName)** **sponsorName REFERENCES** **Sponsor(sponsorName)**	None
Organization(orgName: Char(40), email: Char(40), website: Char(50))	orgName	orgName	None	None

Employee(employeeID : Integer, email: Char(40), name: char(40), **orgName**: Char(40))	(employeeID,**orgName**)	(employeeID,**orgName**)	**orgName REFERENCES** **Organization(orgName)**	None
CableNetwork(channel : Char(40), **broadcasterName**: char(40))	(channel, **broadcasterName**)	(channel, **broadcasterName**)	**broadcasterName REFERENCES** **BroadcastService(broadcasterName)**	None
StreamingService(website: Char(40), **broadcasterName**: char(40))	(website,**broadcasterName**)	(website,**broadcasterName**)	**broadcasterName REFERENCES** **BroadcastService(broadcasterName)**	None
BroadcastService(broadcasterName: Char(40), country: Char(30), language: Char(20))	broadcasterName	broadcasterName	None	None
Broadcasts(**broadcasterName**: Char(40), **tournamentName**: Char(40), **gameName**: Char(50), cost: Integer, viewership: Integer)	**(broadcasterName, tournamentName, gameName)**	**(broadcasterName, tournamentName, gameName)**	**broadcasterName REFERENCES** **BroadcastService(broadcasterName)** **tournamentName** **REFERENCES** **Tournament(tournamentName)** **gameName REFERENCES Game(gameName)**	None

Foreign Keys : Bolded, Primary Keys or components of primary keys: Underlined

SQL DDL

Bolded Tables are decomposed into the two tables immediately below them

```
CREATE TABLE Spectators {
   name char[40],
   ticketNumber int,
   driverID int,
   phoneNumber char[20],
   PRIMARY KEY(ticketNumber)
   }

CREATE TABLE SpectatorInfo{
  driverID int,
  name char[40],
  phoneNumber char[20],
  PRIMARY KEY(driverID)

}

CREATE TABLE SpectatorTicket{
 ticketNumber int,
 driverID int,
 PRIMARY KEY(ticketNumber),
 FOREIGN KEY(driverID) REFERENCES SpectatorInfo
}


CREATE TABLE Attends {
   ticketNumber int,
```

```
   tournamentName char[40],
   name char[40],
   PRIMARY KEY(ticketNumber, tournamentName),
   FOREIGN KEY(ticketNumber) REFERENCES Spectators(ticketNumber),
   FOREIGN KEY(tournamentName) REFERENCES Tournament(tournamentName)
   }

CREATE TABLE Player {
   inGameName char[30],
   playerName char[40],
   dateOfBirth char[20],
   nationality char[20],
   teamName char[30],
   contractLength int,
   phoneNumber char[20],
   PRIMARY KEY(inGameName, playerName),
   FOREIGN KEY(teamName) REFERENCES Team(teamName)
   }

CREATE TABLE Team {
   teamLocation char[30],
   teamName char[30],
   managerName char[40] NOT NULL,
   managerEmail char[40] NOT NULL,
   PRIMARY KEY(teamName),
   FOREIGN KEY(managerName, managerEmail) REFERENCES Manager(managerName, managerEmail) ON DELETE
NO ACTION
   }

CREATE TABLE Manager {
   name char[40],
   email char[40],
   phoneNumber char[15],
   PRIMARY KEY(name, email)
   }


CREATE TABLE CompetesIn {
   gameName char[50],
   tournamentName char[40],
   teamName char[30],
   entryFee int,
   winnings int,
   tournyResult int,
   PRIMARY KEY(gameName, tournamentName, teamName),
   FOREIGN KEY(teamName) REFERENCES Team(Teamname),
   FOREIGN KEY(tournamentName) REFERENCES Tournament(tournamentName),
   FOREIGN KEY(gameName) REFERENCES Game(gameName)
   }

CREATE TABLE Tournament {
   tournamentName char[40],
   ticketPrice int,
```

```
  grandPrize int,
  orgName char[40] NOT NULL,
  address char[40] NOT NULL,
  arenaName char[40] NOT NULL,
  date char[20],
  time char[5],
  PRIMARY KEY(tournamentName),
  FOREIGN KEY(orgName) REFERENCES Organization(orgName)
    ON DELETE NO ACTION,
  FOREIGN KEY(address, arenaName) REFERENCES Venue(address, arenaName)
    ON DELETE NO ACTION
  }

CREATE TABLE Game {
  gameName char[50],
  numParticipants int,
  crossPlatformPlay boolean,
  PRIMARY KEY(gameName)
  }

CREATE TABLE MOBA {
  gameName char[50],
  numChampions int,
  PRIMARY KEY(gameName),
  FOREIGN KEY(gameName) REFERENCES Game(gameName)
  }

CREATE TABLE fightingGame {
  gameName char[50],
  numCharacters int,
  PRIMARY KEY(gameName),
  FOREIGN KEY(gameName) REFERENCES Game(gameName)
  }

CREATE TABLE FPS {
  gameName char[50],
  numMaps int,
  PRIMARY KEY(gameName),
  FOREIGN KEY(gameName) REFERENCES Game(gameName)
  }

CREATE TABLE strategyGame {
  gameName char[50],
  DLCIncluded boolean,
  PRIMARY KEY(gameName),
  FOREIGN KEY(gameName) REFERENCES Game(gameName)
  }

CREATE TABLE PlayedAt {
  tournamentName char[40],
  gameName char[50],
  tier char[1],
  PRIMARY KEY(tournamentName, gameName),
```

```
   FOREIGN KEY(tournamentName) REFERENCES Tournament(tournamentName),
   FOREIGN KEY(gameName) REFERENCES Game(gameName)
   }

CREATE TABLE Venue {
   address char[40],
   arenaName char[40],
   bookingFee int,
   PRIMARY KEY(address, arenaName)
   }

CREATE TABLE Sponsor {
   sponsorName char[50],
   zipCode char[20],
   city char[30],
   timezone char[30],
   PRIMARY KEY(sponsorName)
   }



CREATE TABLE SponsorLocation{
  zipCode char[20],
  city  char[30],
  timezone char[30],
  PRIMARY KEY(zipCode, city)
}

CREATE TABLE SponsorInfo{
      sponsorName char[50],
      zipCode char[20],
      city  char[30],
 PRIMARY KEY(sponsorName),
 FOREIGN KEY(zipCode,city) REFERENCES SponsorLocation(zipCode,city)
}


CREATE TABLE Funds {
   tournamentName char[40],
   sponsorName char[50],
   sponsorFee int,
   PRIMARY KEY(tournamentName, sponsorName),
   FOREIGN KEY(tournamentName) REFERENCES Tournament(tournamentName),
   FOREIGN KEY(sponsorName) REFERENCES Sponsor(sponsorName)
   }

CREATE TABLE Organization {
   orgName char[40],
   email char[40],
   website char[50],
   PRIMARY KEY(orgName)
```

```
    }

CREATE TABLE Employee {
   employeeID int,
   email char[40],
   name char[40],
   orgName char[40],
   PRIMARY KEY(employeeID, orgName),
   FOREIGN KEY(orgName) REFERENCES Organization(orgName) ON DELETE CASCADE
   }

CREATE TABLE CableNetwork {
   channel char[40],
   broadcasterName char[40],
   PRIMARY KEY(broadcasterName),
   FOREIGN KEY(broadcasterName) REFERENCES BroadcastService(broadcasterName)
   }




CREATE TABLE StreamingService {
   website char[40],
   broadcasterName char[40],
   PRIMARY KEY(broadcasterName),
   FOREIGN KEY(broadcasterName) REFERENCES BroadcastService(broadcasterName)
   }

CREATE TABLE BroadcastService {
   broadcasterName char[40],
   country char[30],
   language char[20],
   PRIMARY KEY(broadcasterName)
   }

CREATE TABLE Broadcasts {
   broadcasterName char[40],
   tournamentName char[40],
   gameName char[40],
   cost int,
   viewership int,
   PRIMARY KEY(broadcasterName, tournamentName, gameName)
   FOREIGN KEY(broadcasterName) REFERENCES BroadcastService(broadcasterName),
   FOREIGN KEY(tournamentName) REFERENCES Tournament(tournamentName),
   FOREIGN KEY(gameName) REFERENCES Game(gameName)
   }
```

Queries

“Insertion: Add a player to the team”
SQL DDL:
INSERT INTO Player VALUES (**“**Ninja”, “Tyler Blevins”, “5-June-1991”, “American”, “324-556-6578”, “Cloud9”, 5);

“Insertion: Add an employer to an organization”
SQL DDL:
INSERT INTO Employee VALUES (0839, “aTarik@netease.com”, “Avid Tarik”, “NetEase”);

“Deletion: Remove a player from a team”
SQL DDL:
DELETE FROM Player WHERE inGameName = “Ninja”;

“Deletion: Remove an employee from an organization”
SQL DDL:
DELETE FROM Employer WHERE employeeID = 0839;

“Deletion: Delete an organization”
SQL DDL:
DELETE FROM Organization WHERE orgName = “NetEase”;
This also would additionally delete all the employees in the organization named NetEase

“Update: Update player information”.
SQL DDL:
UPDATE Player SET birthDate = “dateOfBirth” WHERE inGameName = “Ninja”;

“Select: Select all employees that work for a specific organization”

SQL DDL:

SELECT Employee WHERE orgName= “NetEase”

Examples for the following queries were not included since they have not yet been covered in lecture :

- Projection
- Join
- Aggregation with Group by
- Aggregation with Having
- Nested Aggregation with Group By
- Division