DATABASE SYSTEM DESIGN

Joseph McDonough

Dr. Schwartz

CMPT 308N - 113

25 November 2019

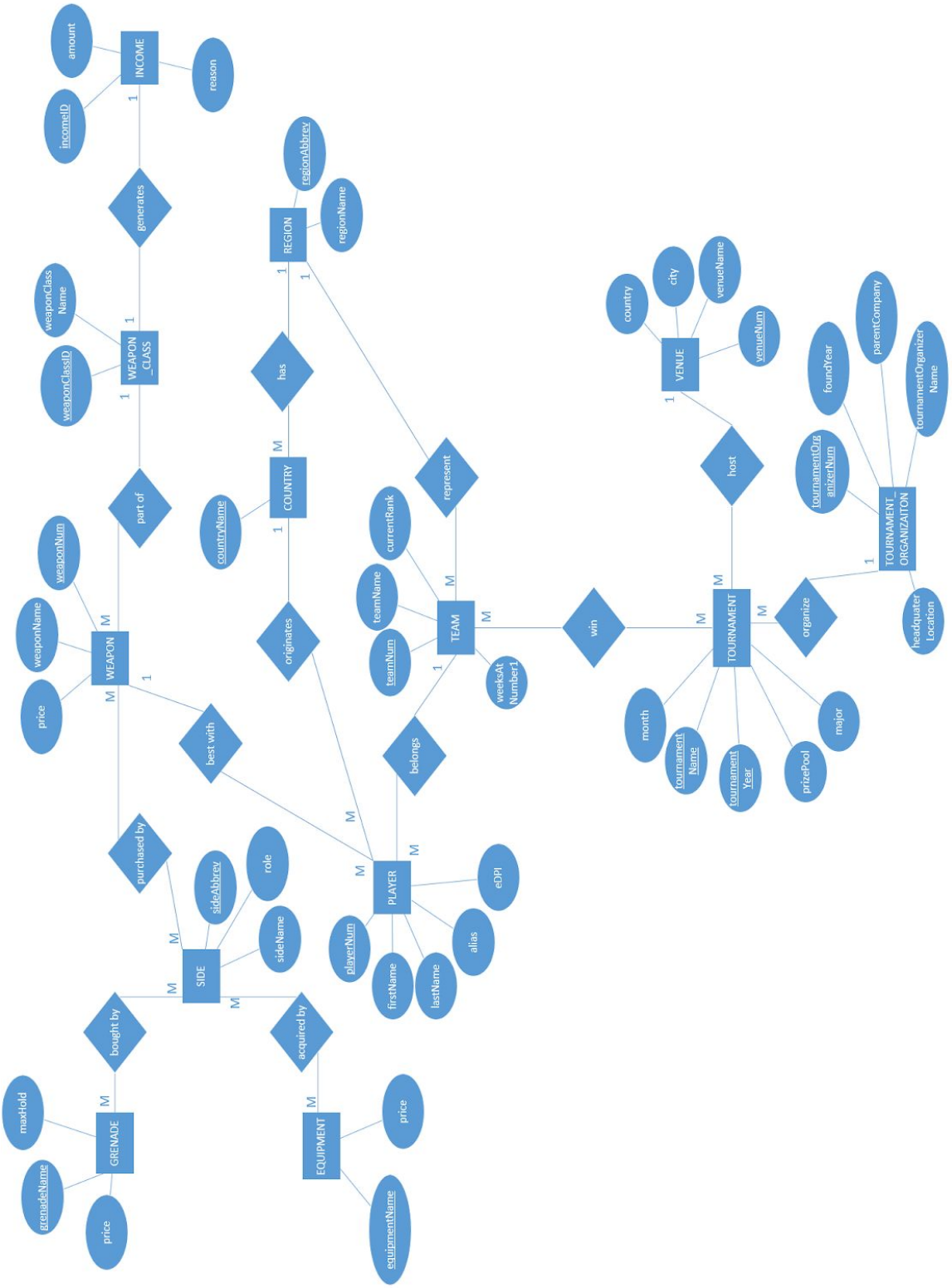
TABLE OF CONTENTS

I. Model Description

As the esports and the Counter-Strike:Global Offensive (CS:GO) scene grow to a worldwide audience, there is a constant influx of new players, countries, and teams as well as the contents inside the game. Unlike traditional sports that are broken into seasons, CS:GO is almost exclusively tournaments. There are a lot of big tournaments each month with some recurring every year. These tournaments (differentiated by a combination of name and year) have prize pools as well as a winning team. Generally twice a year, there is a tournament that is more prestigious than the rest and those are referred to as the majors. It is important to differentiate the two majors from the rest of the big tournaments. Each tournament is hosted in a venue and organized by one tournament organizer. The tournament organizers are given a name and number in addition to the year they were founded, their parent company (if any), and the current country their headquarters is located. The venues are various arenas and stadiums around the world and are referred to by their venue number, but the name of the venue as well as the city and country are noted. These tournaments, as expected, are played by many teams and these teams are playing many tournaments, such that the same set of teams will face each other several times. Teams are given a team number as well as their team name. A team has a world ranking and to know who has been the best, the weeks spent at the number one spot in the world is also kept. A team is made up of five players and since people, although extremely rare, can have the same first name, last name, and choose the same alias (in-game name), each player is given a number. Each player plays with different settings and since they can vary greatly, each player's mouse sensitivity (measured as eDPI) is also recorded. A player originates from a country (recorded by its name) and each country belongs to a region (recorded by its abbreviation).

Since CS:GO is a worldwide esport, they use regions to group teams of similar locations together. Regions are generally just the continent, but Asia gets divided into three: CIS, Ocenia, and the rest of the continent. Each team represents a region (which is determined by the region of the majority of its players). In CS:GO, there are many weapons a player can play with, but everyone has their preference, so each player's best weapon is noted. In-game, everything costs money to buy, so a weapon has a price along with its name and number. There are many weapons of the same type and so a weapon class is made to group like weapons together. The classes are differentiated by their class name. In order to spend money, a player has to be able to make money. Each weapon class generates a different amount of income. There are many different reasons a player can gain money and these different reasons result in different dollar amounts. For simplicity, each reason and dollar value is given an ID number. CS:GO is a game of two sides, offense and defense. Each side has a name and abbreviation. The offense is only able to purchase a certain set of weapons, grenades, and equipment. That set contains some of the same stuff offered to the defense, but both sides have weapons, grenades, and equipment exclusive to them. Grenades have a price, a name, and the amount a player can hold at one time. Equipment is given a name and price. Ultimately, this database could be used to get stats about various players and teams. There are over a hundred tournaments each year and an easy way to get data about how a team is performing is needed. Yet as the game grows and more items get added, this database can also be a way to keep track of all that new data. It can be used in figuring out how much money a player would like to spend in a given round and how they can go about making that money.

II. Entity-Relationship Diagram



III. Table Statements

Note that all table definition names are prefaced by ‘a’ so that they appear at the top of the list of tables on the Oracle server.

A. Country

1. Description

Country table servers to mainly place the countries that appear in the database to a region. Players are born to a country, not a region and teams represent region, not necessarily countries. This table servers as a way to link countries to their region.

2. Definition

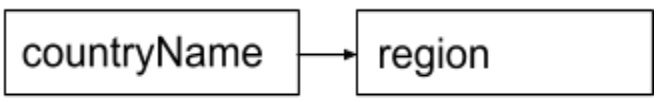
```
CREATE TABLE aCountry (
        countryName NVARCHAR2(40),
        region NVARCHAR2(20),
        CONSTRAINT pk_aCountry PRIMARY KEY (countryName) );
```

3. Foreign Keys

```
ALTER TABLE aCountry ADD CONSTRAINT fk_aCountryRegion
FOREIGN KEY (region) REFERENCES aRegion(regionAbbreviation);
```

4. Third-Normal Form Justification

The country table contains two attributes with only one serving as the primary key. The other attribute is a foregin key used to link to the region table. There are no partial or transitive dependencies.



B. Equipment

1. Description

This table holds the various equipment each player can buy. Each piece of equipment has a unique name and a price to go with it. Since some equipment can only be bought be one side, their exclusive side is recorded.

2. Definition

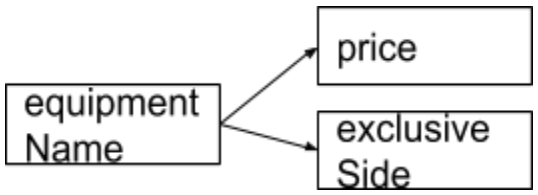
```
CREATE TABLE aEquipment (
        equipmentName NVARCHAR2(20),
        exclusiveSide NVARCHAR2(2),
        price INTEGER,
        CONSTRAINT pk_aEquipment PRIMARY KEY (equipmentName) );
```

3. Foreign Keys

```
ALTER TABLE aEquipment ADD CONSTRAINT fk_aEquipmentSIDE
FOREIGN KEY (exclusiveSide) REFERENCES aSide(sideAbbreviation);
```

4. Third-Normal Form Justification

The equipment table has three attributes, with one of them serving as the primary key. One of the remaining two attributes serves as a foreign key to the side table. There are no partial or transitive dependencies.



C. Grenade

1. Description

This table contains the list of all the grenades available for purchase. Each grenade has a unique name. They also have a price and a maximum amount a player can hold at a given point in time. Since both sides have grenades exclusive to them, that side is noted.

2. Definition

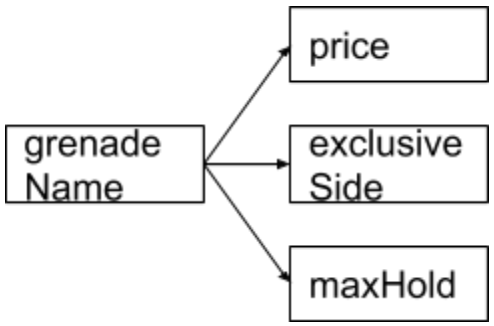
```
CREATE TABLE aGrenade (
        grenadeName NVARCHAR2(20),
        exclusiveSide NVARCHAR2(2),
        price INTEGER,
        maxHold INTEGER,
        CONSTRAINT pk_aGrenade PRIMARY KEY (grenadeName) );
```

3. Foreign Keys

```
ALTER TABLE aGrenadeADD CONSTRAINT fk_aGrenadeSIDE
FOREIGN KEY (exclusiveSide) REFERENCES aSide(sideAbbreviation);
```

4. Third-Normal Form Justification

The grenade table contains four attributes, with only one serving as a primary key and one a foreign key (to side). There are no partial or transitive dependencies.



D. Income

1. Description

There are many ways that players can gain money and this table is where that data is maintained. The way a player can gain money is explained in the reason column, but since that data is can be wordy, each reason has an incomeID, mainly to serve as a more effective foregin key in other tables. Each reason generates income, which is recorded in the amount column.

2. Definition

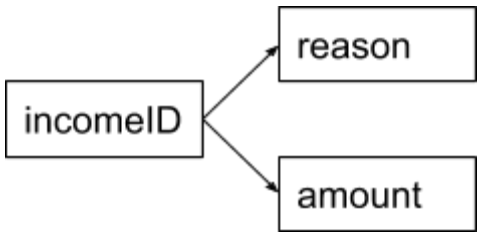
```
CREATE TABLE aIncome (
        incomeID INTEGER,
        reason NVARCHAR2(40),
        amount INTEGER,
        CONSTRAINT pk_aIncome PRIMARY KEY (incomeID) );
```

3. Foreign Keys

None

4. Third-Normal Form Justification

The income table has three values with one as the primary key. The primary key points to all these values and there are not partial or transitive dependencies present.



E. Player

1. Description

The player table holds the list of numerous professional players. In the event a player had the same first name, last name, and alias, the playerNum serves as the primary key. A player belongs to represents a country and has a current team. Each players favorite weapon and eDPI, which is the sensitivity they play on, is recorded.

2. Definition

```
CREATE TABLE aPlayer (
        playerNum INTEGER,
        firstName NVARCHAR2(20),
        lastName NVARCHAR2(20),
        alias NVARCHAR2(20),
        country NVARCHAR2(40),
        currentTeam INTEGER,
        bestWeapon INTEGER,
        eDPI INTEGER,
        CONSTRAINT pk_aPlayer PRIMARY KEY (playerNum) );
```

3. Foreign Keys

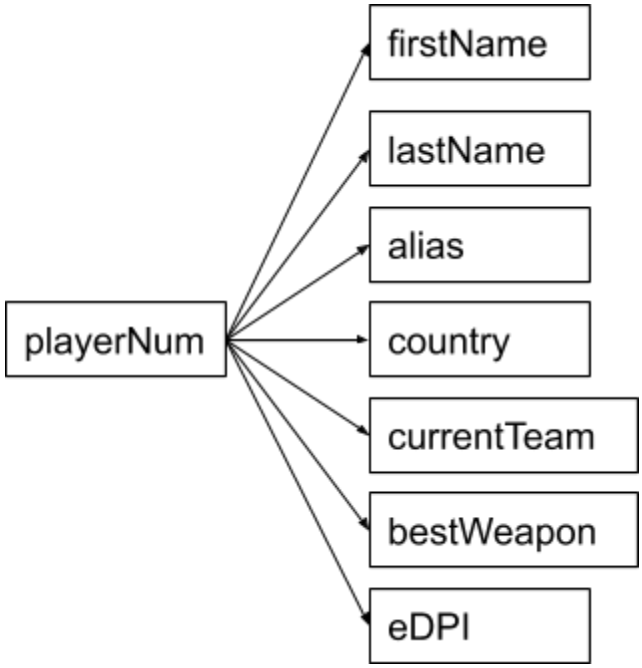
```
ALTER TABLE aPlayer ADD CONSTRAINT fk_aPlayerCOUNTRY
FOREIGN KEY (country) REFERENCES aCountry(countryName);

ALTER TABLE aPlayer ADD CONSTRAINT fk_aPlayerTEAM
FOREIGN KEY (currentTeam) REFERENCES aTeam(teamNum);
```

ALTER TABLE aPlayer ADD CONSTRAINT fk_aPlayerWEAPON

FOREIGN KEY (bestWeapon) REFERENCES aWeapon(weaponNum);

4. Third-Normal Form Justification

The player table holds eight attributes, one as a primary key and three as a foreign key (one to country, one to team, one to weapon). The primary key points to the other attributes and nothing else. There are no partial or transitive dependencies.



F. Region

1. Description

The region table converts the region abbreviations that are used in the other tables into the full region name. Since the region names can be lengthy and the abbreviation is what is commonly used, this table serves as a way to use the abbreviations throughout the database while still having a way to convert it to its proper name.

2. Definition

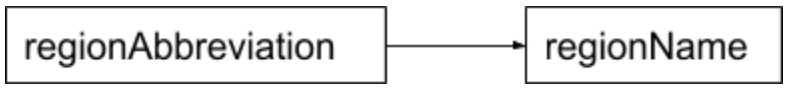
```
CREATE TABLE aRegion(
        regionAbbreviation NVARCHAR2(3),
        regionName NVARCHAR2(40),
        CONSTRAINT pk_aRegion PRIMARY KEY (regionAbbreviation) );
```

3. Foreign Keys

None

4. Third-Normal Form Justification

The region table has two attributes with one of them being the primary key and no foregin keys. There are no partial or transitive dependencies.



G. Side

1. Description

This game is played between the two sides and each side has things to purchase that are exclusive to them. This table generates the sideAbbreviation foreign key that is used in those tables. Also recorded here are the full side names and a note on whether it is the offense or defense (defined in role).

2. Definition

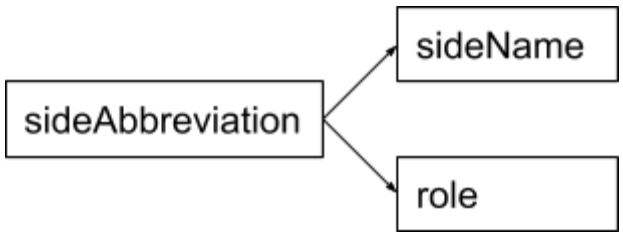
```
CREATE TABLE aSide (
        sideName NVARCHAR2(20),
        sideAbbreviation NVARCHAR2(2),
        role NVARCHAR2(10),
        CONSTRAINT pk_aSide PRIMARY KEY (sideAbbreviation) );
```

3. Foreign Keys

None

4. Third-Normal Form Justification

The side table holds three attributes with one serving as the primary key and none as foreign keys. There are no partial or transitive dependencies.



H. Team

1. Description

This table holds the data of each team that appears in this database. Each team has a number since team names change over time and the number ensures that only the name has to change in this table. Each team also represents a region and a current rank. Since some teams disbanded and no longer exists, not every team has a current rank. The weeks the team has spent at #1 in the world is also noted.

2. Definition

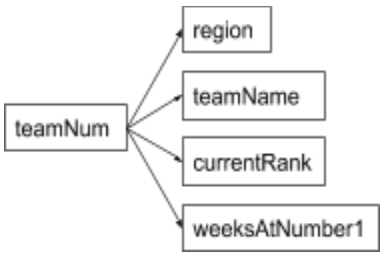
```
CREATE TABLE aTeam (
        teamNum INTEGER,
        region NVARCHAR2(20),
        teamName NVARCHAR2(20),
        currentRank INTEGER,
        weeksAtNumber1 INTEGER,
        CONSTRAINT pk_aTeam PRIMARY KEY (teamNum) );
```

3. Foreign Keys

```
ALTER TABLE aTeam ADD CONSTRAINT fk_aTeamRegion
FOREIGN KEY (region) REFERENCES aRegion(regionAbbreviation);
```

4. Third-Normal Form Justification

The team table hosts five attributes, one serving as primary key and one as a foreign key to the region table. There are no partial or transitive dependencies.



I. Tournament

1. Description

The tournament table holds some of the tournaments that have been played over the last 6 years. Each tournament has an organizer as well as a name and a year. A tournament will never have the same name and be played twice in the same year, so a combination of the organizer, name, and year serve as a composite primary key. The month of the tournament is noted. The venue, winning team, and prize pool for each tournament are also recorded. Since the majors are more prestigious than other tournaments, it is important to note which were majors.

2. Definition

```
CREATE TABLE aTournament (
        tournamentName NVARCHAR2(20),
        tournamentYear INTEGER,
        tournamentMonth NVARCHAR2(20),
        tournamentOrganizer INTEGER,
        venue INTEGER,
        winningTeam INTEGER,
        prizePool INTEGER,
        major NVARCHAR2(3),
        CONSTRAINT pk_aTournament PRIMARY KEY (tournamentName,
        tournamentYear) );
```

3. Foreign Keys

```
ALTER TABLE aTournament ADD CONSTRAINT fk_aTournamentTEAM
FOREIGN KEY (winningTeam) REFERENCES aTeam(teamNum);
```

ALTER TABLE aTournament ADD CONSTRAINT fk_aTournamentVENUE

FOREIGN KEY (venue) REFERENCES aVenue(venueNum);

ALTER TABLE aTournament ADD CONSTRAINT

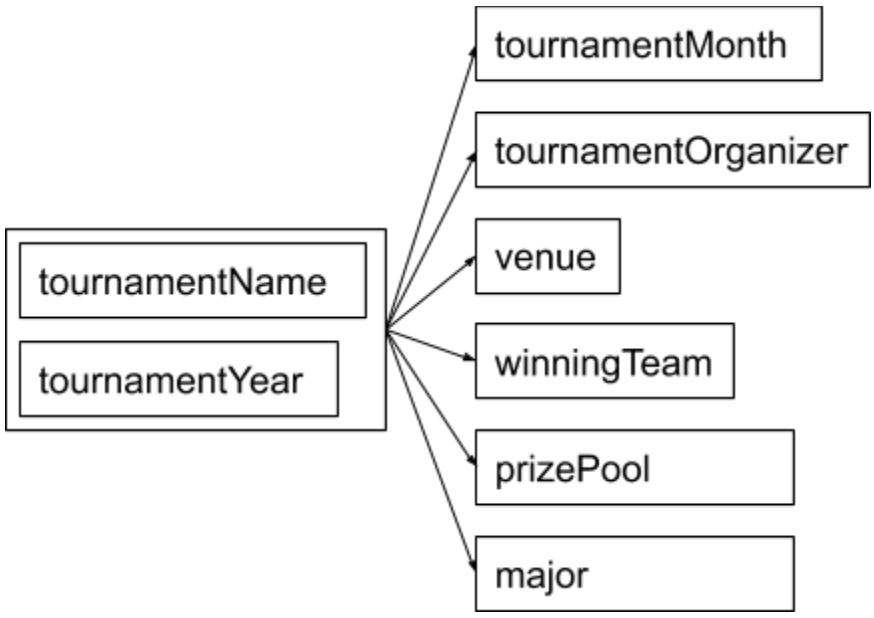
fk_aTournamentTOURNEYORG

FOREIGN KEY (tournamentOrganizer) REFERENCES

aTournamentOrganizer(tournamentOrganizerNum);

4. Third-Normal Form Justification

The tournament table has eight attributes, two come together to form a composite primary key and three others are used as foregin keys (one to team, one to venue, one to tournamentOrganizer). There are no partial or transitive dependencies.



J. Tournament Organizer

1. Description

Tournament organizers prepare for multiple tournaments each year. Each organizer is given a number to serve as a better foreign key in the rest of the tables. Each organizer has a name along with its founding year and current headquarters location. Since only some organizers have parent companies, those that do have it listed.

2. Definition

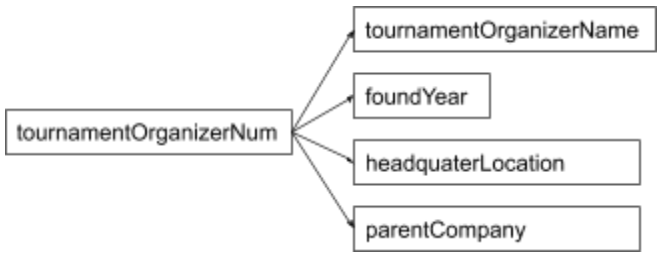
```
CREATE TABLE aTournamentOrganizer (
        tournamentOrganizerNum INTEGER,
        tournamentOrganizerName NVARCHAR2(20),
        foundYear INTEGER,
        headquaterLocation NVARCHAR2(40),
        parentCompany NVARCHAR2(20),
        CONSTRAINT pk_aTournamentOrganizer PRIMARY KEY
        (tournamentOrganizerNum) );
```

3. Foreign Keys

None

4. Third-Normal Form Justification

The tournament organizer table has five attributes with one serving as primary key and no foreign keys. There are no partial or transitive dependencies.



K. Venue

1. Description

Venues can host multiple tournaments and can be utilized by multiple organizers. Each venue was given a number to go with its name. Since the venues are physical arenas and stadiums located around the world, the city and country of the place is recorded.

2. Definition

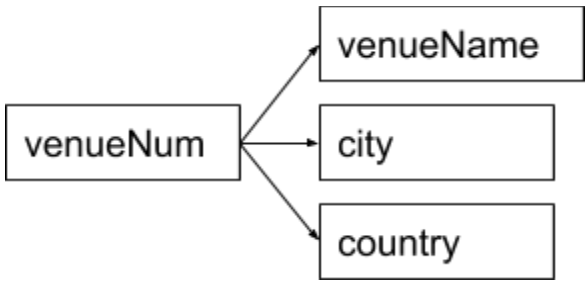
```
CREATE TABLE aVenue (
        venueNum INTEGER,
        venueName NVARCHAR2(40),
        city NVARCHAR2(20),
        country NVARCHAR2(40),
        CONSTRAINT pk_aVenue PRIMARY KEY (venueNum) );
```

3. Foreign Keys

None

4. Third-Normal Form Justification

The venue table has four attributes with one serving as primary key and no foreign keys. There are no partial or transitive dependencies.



L. Weapon

1. Description

The weapon table is a list of every current weapon available for purchase in game. Each weapon has a number and name. They also all have a price and weapon class. Since not every weapon is available for purchase on both sides, the exclusive side is recorded.

2. Definition

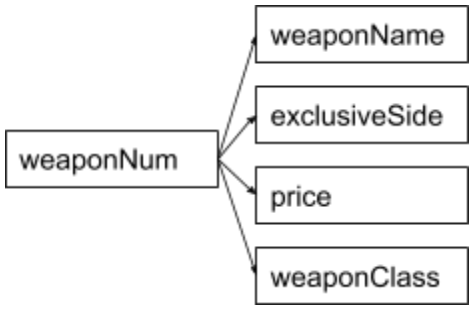
```
CREATE TABLE aWeapon (
        weaponNum INTEGER,
        weaponName NVARCHAR2(20),
        exclusiveSide NVARCHAR2(2),
        price INTEGER,
        weaponClass INTEGER,
        CONSTRAINT pk_aWeapon PRIMARY KEY (weaponNum) );
```

3. Foreign Keys

```
ALTER TABLE aWeapon ADD CONSTRAINT fk_aWeaponSIDE
FOREIGN KEY (exclusiveSide) REFERENCES aSide(sideAbbreviation);

ALTER TABLE aWeapon ADD CONSTRAINT fk_aWeaponWEAPONCLASS
FOREIGN KEY (weaponClass) REFERENCES
aWeaponClass(weaponClassID);
```

4. Third-Normal Form Justification

The weapon table has five attributes, one primary key and two foreign keys (one to side, one to weapon class). There are no partial or transitive dependencies.



M. Weapon Class

1. Description

Weapon classes serve as a way to group like weapons together. This table also serves as a join table between the weapon and income tables. The weapon class has an ID as well as the name. The weaponClassID links it to the weapon table and the eliminationReward links it to the income table.

2. Definition

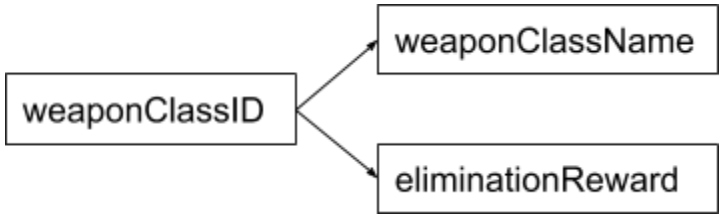
```
CREATE TABLE aWeaponClass (
        weaponClassID INTEGER,
        weaponClassName NVARCHAR2(20),
        eliminationReward INTEGER,
        CONSTRAINT pk_aWeaponClass PRIMARY KEY (weaponClassID) );
```

3. Foreign Keys

```
ALTER TABLE aWeaponClass ADD CONSTRAINT
fk_aWeaponClassINCOME
FOREIGN KEY (eliminationReward) REFERENCES aIncome(incomeID);
```

4. Third-Normal Form Justification

The weapon class table has three attributes, one serves as the primary key and another serves as a foreign key to the income table. There are no partial or transitive dependencies.



IV. Queries

A. Universal Quantifier

Get weapon name and class for weapons that are favorited by at least one player from every country in the database.

```
SELECT aWeapon.weaponName, aWeaponClass.weaponClassName
FROM aWeapon, aWeaponClass
WHERE NOT EXISTS(
        SELECT *
        FROM aCOUNTRY
        WHERE NOT EXISTS(
                SELECT *
                FROM aPlayer
                WHERE aPlayer.country = aCountry.countryname
                AND aPlayer.bestWeapon = aWeapon.weaponNum
                AND aWeaponClass.weaponClassID = aWeapon.weaponClass))
```

WEAPONNAME	WEAPONCLASSNAME
sg 553	rifle

Cardinality: 1

B. Only

For those that only won at the Laxness-Arena, give the team name, their region, and the year(s) they won it.

```
SELECT aTeam.teamName, aTeam.region
FROM aTeam
WHERE aTeam.teamNum IN(
        SELECT tourney1.winningTeam
        FROM aTournament tourney1
        WHERE tourney1.winningTeam NOT IN(
                SELECT tourney2.winningteam
                FROM aTournament tourney2
                WHERE tourney2.venue NOT IN(
                        SELECT aVenue.venuenum
                        FROM aVenue
                        WHERE aVenue.venueName = 'Laxness-Arena')))
```

TEAMNAME	REGION
NiP	EU
Avangar	CIS

Cardinality: 2

C. None

For weapon classes in which none of the Swedish players are best with any of its weapons, give the weapon class name and the income amount it generates

```
SELECT aWeaponClass.weaponClassName, aIncome.amount
FROM aWeaponClass, aIncome
WHERE aWeaponClass.eliminationReward = aIncome.incomeID
AND aWeaponClass.weaponClassID NOT IN(
        SELECT aWeapon.weaponClass
        FROM aWeapon
        WHERE aWeapon.weaponNum IN(
                SELECT aPlayer.bestWeapon
                FROM aPlayer
                WHERE aPlayer.country = 'Sweden'))
```

WEAPONCLASSNAME	AMOUNT
light machine gun	200
submachine gun	600
shotgun	900
melee	1500

Cardinality: 4

D. Outer Join

1. Left Outer Join

For all reasons other than losing a round, give the income amount, income reason, and weaponClass name, if any.

```
SELECT aIncome.amount, aIncome.reason,
aWeaponClass.weaponClassName
FROM aIncome LEFT OUTER JOIN aWeaponClass ON
(aIncome.incomeID = aWeaponClass.eliminationReward)
WHERE aIncome.reason NOT LIKE 'lose%'
```

AMOUNT	REASON	WEAPONCLASSNAME
100	sniper elimination	sniper
200	light machine gun elimination	light machine gun
300	rifle elimination	rifle
300	pistol elimination	pistol
600	submachine gun elimination	submachine gun
800	bomb plant	
900	shotgun elimination	shotgun
1500	knife elimination	melee
3250	win via elimination	
3250	win via round expire	
3500	win via explosion	
3600	win via defuse	

Cardinality: 12

2. Right Outer Join

For teams that have won at least one major after 2015, give the tournament name, tournament year, team name, as well as the team's current rank, if any.

```
SELECT aTeam.currentRank, aTeam.teamName,
aTournament.tournamentName, aTournament.tournamentYear
FROM aTournament RIGHT OUTER JOIN aTeam ON
(aTournament.winningTeam = aTeam.teamNum)
WHERE aTournament.major = 'Yes'
AND aTournament.tournamentYear > 2015
```

CURRENTRANK	TEAMNAME	TOURNAMENTNAME	TOURNAMENTYEAR
	SK	One Cologne	2016
	Luminosity	Columbus	2016
1	Astralis	Major Atlanta	2017
	Gambit	Major Krakow	2017
1	Astralis	Major London	2018
4	Liquid	Major Boston	2018
1	Astralis	Berlin Major	2019
1	Astralis	IEM Katowice	2019

Cardinality: 8

3. Full Outer Join

For venues or tournament organizers that are not from the United States of America, give the venue name and country, if any, as well as the organizer name and headquarters location, if any, such that if both exists, the venue's country is the same as the headquarters'.

```
SELECT aVenue.venueName, aVenue.country,
aTournamentOrganizer.tournamentOrganizerName,
aTournamentOrganizer.headquaterLocation
FROM aVenue FULL OUTER JOIN aTournamentOrganizer ON
(aVenue.country = aTournamentOrganizer.headquaterLocation)
WHERE aVenue.country <> 'United States of America'
OR aTournamentOrganizer.headquaterLocation <> 'United States of
America'
```

VENUENAME	COUNTRY	TOURNAMENTORGANIZERNAME	HEADQUATERLOCATION
Qudos Bank Arena	Australia		
SSE Arena	England	FACEIT	England
Laxness-Arena	Germany	ESL	Germany
Mercedes-Benz Arena	Germany	ESL	Germany
TAURON Arena	Poland		
Spodex	Poland		
Sala Polivalenta	Romania	PGL	Romania
Polyvalent Hall	Romania	PGL	Romania
Yubileyny Sports Palace	Russia	Epic Esports Events	Russia
Elmia	Sweden	DreamHack	Sweden
		StarLadder	Ukraine

Cardinality: 11

E. At Least Six Tables

North American players have a tendency to thrive playing in front of a home crowd. Give the player's alias, best weapon, their country and team, as well as the tournament name, year, and venue name such that it is also located in North America.

```
SELECT aPlayer.alias, aWeapon.weaponName, aPlayer.country,
aTeam.teamName, aTournament.tournamentName,
aTournament.tournamentYear, aVenue.venueName
FROM aRegion, aTeam, aTournament, aPlayer, aVenue, aWeapon
WHERE aRegion.regionName = 'North America'
AND aRegion.regionAbbreviation = aTeam.region
AND aTeam.teamNum = aPlayer.currentTeam
AND aTeam.teamNum = aTournament.winningTeam
AND aVenue.venueNum = aTournament.venue
AND aPlayer.bestWeapon = aWeapon.weaponNum
```

ALIAS	WEAPONNAME	COUNTRY	TEAMNAME	TOURNAMENTNAME	TOURNAMENTYEAR
Brehze	sg 553	United States of America	Evil Geniuses	One New York	2019
Elige	usp-s	United States of America	Liquid	Major Boston	2018
Ethan	aug	United States of America	Evil Geniuses	One New York	2019
Twistzz	sg 553	Canada	Liquid	Major Boston	2018
nitr0	knife	United States of America	Liquid	Major Boston	2018
tarik	pp-bizon	United States of America	Evil Geniuses	One New York	2019

Cardinality: 6

F. Lab 2/Lab 3 Complexity

1. query1

Get weapon names for weapons that generate less income than the pistol class

```
SELECT aWeapon.weaponName
FROM aWeapon, aWeaponClass
WHERE aWeapon.weaponClass = aWeaponClass.weaponClassID
AND aWeaponClass.eliminationReward IN(
        SELECT income1.incomeID
        FROM aWeaponClass, aIncome income1
        WHERE income1.incomeID = aWeaponClass.eliminationReward
        GROUP BY income1.incomeID, income1.amount
        HAVING income1.amount < (SELECT income2.amount
                FROM aIncome income2, aWeaponClass wepClass2
                WHERE wepClass2.weaponClassName = 'pistol'
                AND wepClass2.eliminationReward =
                income2.incomeID))
```

WEAPONNAME
awp
g3sg1
m249
negev
scar-20
ssg 08

Cardinality: 6

2. query2

Name the countries in which a tournament organizer and venue are co-located. Include the amount of tournaments hosted in that country.

```
SELECT aVenue.country,
COUNT(aTournament.tournamentName)TOURNAMENT_AMOUNT
FROM aTournament, aVenue JOIN aTournamentOrganizer ON
(aVenue.country = aTournamentOrganizer.headquaterLocation)
WHERE aTournament.venue = aVenue.venueNum
AND aTournament.tournamentOrganizer =
aTournamentOrganizer.tournamentOrganizerNum
GROUP BY aVenue.country
```

COUNTRY	TOURNAMENT_AMOUNT
England	1
Germany	4
Russia	1
Sweden	2
United States of America	3

Cardinality: 5

3. query3

For team's whose average weapon price is less than the current #1 ranked team's weapon price, give the team number, name, their average as well as most expensive weapon weapon price

```
SELECT aTeam.teamNum, aTeam.teamName,
to_char((AVG(aWeapon.price)),'9999.99') average_price,
MAX(aWeapon.price)most_expensive
FROM aTeam, aPlayer, aWeapon
WHERE aWeapon.weaponNum = aPlayer.bestWeapon
AND aPlayer.currentTeam = aTeam.teamNum
GROUP BY aTeam.teamNum, aTeam.teamName
HAVING  AVG(aWeapon.Price) <(SELECT AVG(aWeapon.price)
        FROM aTeam, aPlayer, aWeapon
        WHERE aWeapon.weaponNum = aPlayer.bestWeapon
        AND aPlayer.currentTeam = aTeam.teamNum
        AND aTeam.currentRank = 1)
```

TEAMNUM	TEAMNAME	AVERAGE_PRICE	MOST_EXPENSIVE
4	Liquid	1066.67	3000
7	Faze	1733.33	3000
8	Avangar	2266.67	3100

Cardinality: 3

V. Updates

A. ER Diagram

Changed the relationship between weapon class and income from many-to-many to one-to-one because realized it was an error made and the correct relationship should be one-to-one.

Modified certain attributes on tables to fit with the new table structures.

Changed primary keys on some tables to fit with new table structures.

B. Table Structure

Added weaponNum to the weapon table in the event a weapon name was changed over time, it would make it easier to change the data in one place. Also wanted to ensure third normal form.

Changed primary key of tournament table to a unique and minimal composite key as well as ensuring third normal form.

Added foundYear and headquaterLocation to tournamentOrganizer table to give it more data and make some queries better.

Added regionAbbreviation as the primary key to region table and made regionName an attribute. Helped in making queries better and ensuring third normal form.

Removed MVP from tournament table because it should result in a link between tournament and players that is not desired at this current time.

C. Table Data

Changed the winners of some tournaments to make queries better.

Deleted columns to fit new table structure.

Added ID number columns to fit new table structures.