Jessica Spokoyny
CS 340
Summer 2016
Final Project

Outline:

My database represents the Game of Thrones universe. Game of Thrones is a medieval fantasy drama tv series which chronicles the struggles among a kingdom's noble families for control of the Iron Throne. The characters within this world have complex relationships and dynamics which make this universe a good candidate for a database. As the show is ongoing, this database changes frequently as new information is discovered. People fight often, die often, and can change from allies to enemies at the drop of a hat. This database could be useful for adding new characters as they are introduced in the show and adding their relationships to other existing characters.

Database Outline in Words:

A character has an ID, a first name, a last name, a title, and a living/dead attribute. It is uniquely identified by its ID. The combination of a character's first name and last name must be unique. A character must have a first name.

A city has an ID, and a name. It is uniquely identified by its ID. A city's name must also be unique. A city must have a name.

A region has an ID, a name, and a continent. It is uniquely identified by its ID. A region's name must also be unique. A region must have a name.

A house has an ID, a name, and a motto. It is uniquely identified by its ID. A house's name must also be unique. A house must have a name.

A connection type has an ID, and a description. It is uniquely identified by its ID. A connection type's description must also be unique. A connection type must have a description.

A character belongs to at most 1 house. A house is related to 0 or more characters. (1-many)

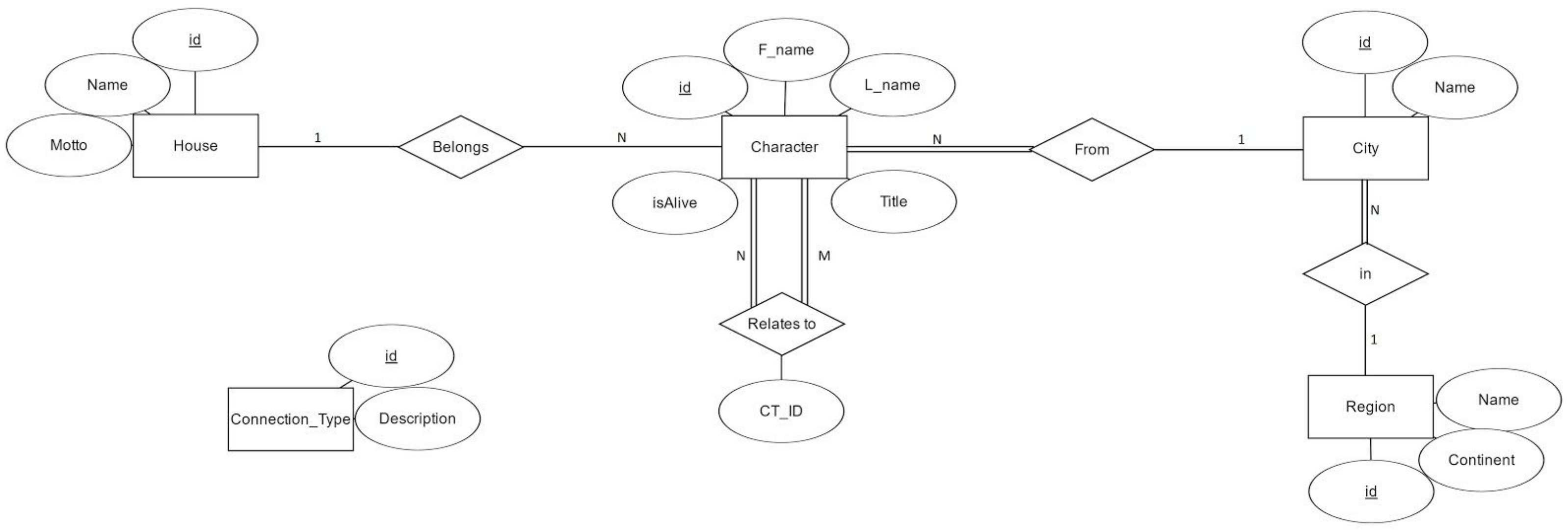
A character is from exactly 1 home city. A city is related to 0 or more characters. (1-many)

A city is in exactly 1 region. A region is related to 0 or more cities. (1-many)

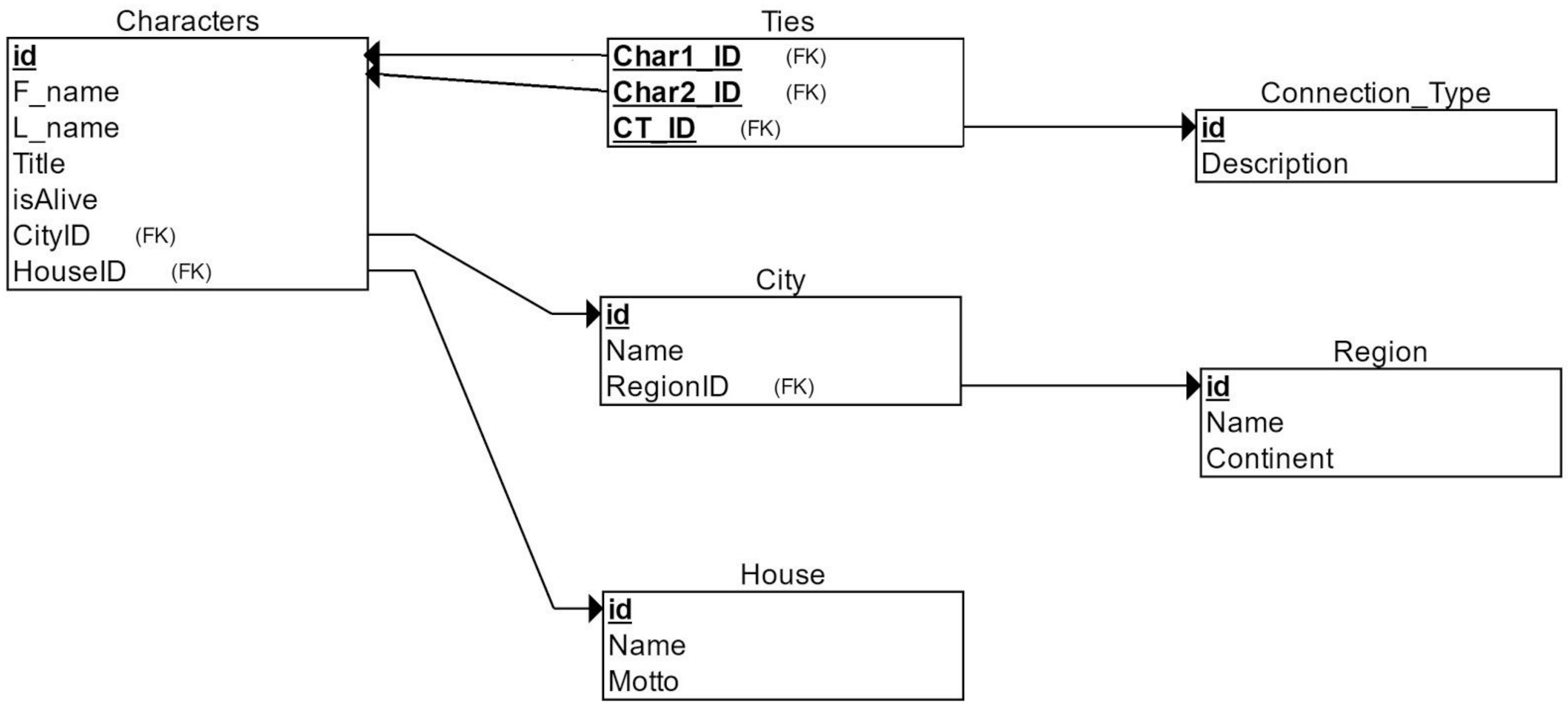
A character is tied to at least 1 other character. Every character is a tie of at least 1 other character. The connection type of the tie is recorded. A tie is uniquely identified by Char1_ID, Char2_ID, and CT_ID. (many-many)

Jessica Spokoyny
CS 340
Summer 2016
Final Project

ER Diagram of Database:



Database Schema:



Jessica Spokoyny
CS 340
Summer 2016
Final Project

Table Creation Queries:

```
-- Create Characters Table
CREATE TABLE Characters (
	id INT NOT NULL AUTO_INCREMENT,
	F_name VARCHAR(255) NOT NULL,
	L_name VARCHAR(255),
	Title VARCHAR(255),
	isAlive INT,
	CityID INT,
	HouseID INT,
	PRIMARY KEY (id),
	CONSTRAINT CityFK FOREIGN KEY (CityID) REFERENCES City (id) ON DELETE
CASCADE ON UPDATE CASCADE,
	CONSTRAINT HouseFK FOREIGN KEY (HouseID) REFERENCES House (id) ON
DELETE CASCADE ON UPDATE CASCADE,
	UNIQUE KEY (F_name, L_name)
) ENGINE=InnoDB;

-- Create City Table
CREATE TABLE City (
	id INT NOT NULL AUTO_INCREMENT,
	Name VARCHAR(255) NOT NULL UNIQUE,
	RegionID INT,
	PRIMARY KEY(id),
	CONSTRAINT RegionFK FOREIGN KEY (RegionID) REFERENCES Region (id) ON
DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB;

-- Create Region Table
CREATE TABLE Region (
	id INT NOT NULL AUTO_INCREMENT,
	Name VARCHAR(255) NOT NULL UNIQUE,
	Continent VARCHAR(255) NOT NULL,
	PRIMARY KEY(id)
) ENGINE=InnoDB;

-- Create House Table
CREATE TABLE House (
	id INT NOT NULL AUTO_INCREMENT,
	Name VARCHAR(255) NOT NULL UNIQUE,
	Motto VARCHAR(255),
	PRIMARY KEY (id)
) ENGINE=InnoDB;
```

-- Create Connection Type Table

```
CREATE TABLE Connection_Type (
	id INT NOT NULL AUTO_INCREMENT,
	Description VARCHAR(255) NOT NULL UNIQUE,
	PRIMARY KEY(id)
) ENGINE=InnoDB;
```

-- Create Ties Table

```
CREATE TABLE Ties (
	Char1_ID INT NOT NULL,
	Char2_ID INT NOT NULL,
	CT_ID INT NOT NULL,
	PRIMARY KEY(Char1_ID, Char2_ID, CT_ID),
	CONSTRAINT Char1FK FOREIGN KEY (Char1_ID) REFERENCES Characters (id) ON
DELETE CASCADE ON UPDATE CASCADE,
	CONSTRAINT Char2FK FOREIGN KEY (Char2_ID) REFERENCES Characters (id) ON
DELETE CASCADE ON UPDATE CASCADE,
	CONSTRAINT ConnTypeFK FOREIGN KEY (CT_ID) REFERENCES Connection_Type
(id) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB;
```

General Use Queries:

SELECT Queries

-- Dropdown Menus

```
SELECT id, F_name FROM Characters;
SELECT id, Name FROM City;
SELECT id, Name FROM Region;
SELECT id, Name FROM House;
SELECT id, Description FROM Connection_Type;
```

-- Show Characters

```
SELECT F_name, L_name, Title, isAlive, City.Name, House.Name FROM Characters
	INNER JOIN City ON CityID = City.id
	LEFT JOIN House ON HouseID = House.id
	ORDER BY F_name;
```

Jessica Spokoyny
CS 340
Summer 2016
Final Project

```
-- Show Cities
SELECT City.Name, Region.Name, Region.Continent FROM City
        INNER JOIN Region ON City.RegionID = Region.id
        ORDER BY Region.Name;

-- Show Regions
SELECT Name, Continent FROM Region
        ORDER BY Continent;

-- Show Houses
SELECT Name, Motto FROM House
        ORDER BY Name;

-- Show Connection Types
SELECT Description FROM Connection_Type
        ORDER BY Description;

-- Show Ties
SELECT DISTINCT c1.F_name, c1.L_name, Connection_Type.Description, c2.F_name,
c2.L_name FROM Ties
        INNER JOIN Characters AS c1 ON Ties.Char1_ID = c1.id
        INNER JOIN Characters AS c2 ON Ties.Char2_ID = c2.id
        INNER JOIN Connection_Type ON CT_ID = Connection_Type.id
        ORDER BY c1.F_name;

-- Search Character by Name
SELECT F_name, L_name, Title, isAlive, City.Name, House.Name FROM Characters
        INNER JOIN City ON CityID = City.id LEFT JOIN House ON HouseID = House.id
        WHERE Characters.id = [charInput];

-- Search Characters by City
SELECT F_name, L_name, Title, isAlive, City.Name, House.Name FROM Characters
        INNER JOIN City ON CityID = City.id LEFT JOIN House ON HouseID = House.id
        WHERE City.id = [cityInput];

-- Search Characters by Region
SELECT F_name, L_name, Title, isAlive, City.Name, Region.Name, Region.Continent,
House.Name FROM Characters
        INNER JOIN City ON CityID = City.id INNER JOIN Region ON City.RegionID =
Region.id
        LEFT JOIN House ON HouseID = House.id
        WHERE Region.id = [regionInput]
        ORDER BY F_name;
-- Search Characters by House
SELECT F_name, L_name, Title, isAlive, City.Name, House.Name, House.Motto FROM
Characters
        INNER JOIN City ON CityID = City.id LEFT JOIN House ON HouseID = House.id
        WHERE House.id = [houseInput];

-- Search Ties by Connection Type
SELECT DISTINCT c1.F_name, c1.L_name, Connection_Type.Description, c2.F_name,
c2.L_name FROM Ties
        INNER JOIN Characters AS c1 ON Ties.Char1_ID = c1.id
        INNER JOIN Characters AS c2 ON Ties.Char2_ID = c2.id
        INNER JOIN Connection_Type ON CT_ID = Connection_Type.id
        WHERE CT_ID = [ctInput];

-- Search Ties by Character
SELECT DISTINCT c1.F_name, c1.L_name, Connection_Type.Description, c2.F_name,
c2.L_name FROM Ties
        INNER JOIN Characters AS c1 ON Ties.Char1_ID = c1.id
        INNER JOIN Characters AS c2 ON Ties.Char2_ID = c2.id
        INNER JOIN Connection_Type ON CT_ID = Connection_Type.id
        WHERE c1.id = [c1Input];
```

ADD Queries

```
--  Add Character
INSERT INTO Characters (F_name, L_name, Title, isAlive, CityID, HouseID)
        VALUES ([firstNameInput], [lastNameInput], [titleInput], [aliveInput],
        (SELECT id FROM City WHERE Name = [homeCityInput]),
        (SELECT id FROM House WHERE Name = [houseInput]));

--  Add City
INSERT INTO City (Name, RegionID)
        VALUES ([nameInput],
        (SELECT id FROM Region WHERE Name =[regionInput]));

--  Add Region
INSERT INTO Region (Name, Continent)
        VALUES ([nameInput], [continentInput]);

--  Add House
INSERT INTO House (Name, Motto)
```

```
        VALUES ([nameInput], [mottoInput]);
```

-- Add Connection Type

```
INSERT INTO Connection_Type (Description)
        VALUES ([descriptionInput]);
```

-- Add Tie

```
INSERT INTO Ties (Char1_ID, Char2_ID, CT_ID)
        VALUES (
        (SELECT id FROM Characters WHERE F_name = [char1FirstNameInput]),
        (SELECT id FROM Characters WHERE F_name = [char2FirstNameInput]),
        (SELECT id FROM Connection_Type WHERE Description = [descriptionInput])),
```

DELETE Queries

-- Delete Character

```
DELETE FROM Characters WHERE F_name = [charInput];
```

-- Delete City

```
DELETE FROM City WHERE Name = [cityInput];
```

-- Delete Region

```
DELETE FROM Region WHERE Name = [regionInput];
```

-- Delete House

```
DELETE FROM House WHERE Name = [houseInput];
```

-- Delete Connection Type

```
DELETE FROM Connection_Type WHERE Description = [ctInput];
```

* all delete queries use forms that bind id to Name/F_name/Description. The user selects a name from a list of dropdowns but the query actually removes the row based on id