

# **Game of Thrones a Database Adaption**

## **CS340 Final Project**

**By: Alex Johnson and Chido Nguyen**

### **Outline:**

Game of Thrones is a TV show adapted from the novels “A Song of Ice and Fire” by George R.R. Martin. The fantasy show has various aspects that we thought we’d be interesting and educational in our approach to learning more about database for the final project. The show has an abundance of relationships that could be tied into creating databases. There were a few we chose to focus on :

- Many to Many (N:M) - Appearances , specifically many characters can appear in many episodes, and vice versa many episodes can have many characters appear.
- One to Many ( 1: M) - Characters can be from 1 Location, but 1 Location can have many characters. Same with houses, Characters can only be part of 1 House, but 1 House can have many characters.
- One to One ( 1:1) - Locations to Houses. Each location can only have 1 House , and each House can only be from 1 Location. We’re tracking specifically their origin location/ central command center.

### **Database Outline:**

## **Episodes**

---

The Episodes table contains the following attributes:

- ID (based off the overall episode number in the whole series, required)
- Season (the season the episode appears in, required)
- Episode (number of the season, required)
- Title (required)
- Air Date (required)

All episodes can have one or more character appearances, as represented by the Appearances table.

## **Characters**

---

The Characters table contains the following attributes:

- ID (required)
- Name (required)
- House ID (references a house)
- Birthplace (references a location)
- Biography

Characters may or may not have a birth location (they may not know where they were born). Characters are part of only one house (based off of patrilineal culture in universe).

## Houses

---

The Houses table contains the following attributes:

- ID (required)
- Name (required)
- Motto
- Sigil
- Location ID (references a location)

A house can exist in no location or one location. A house can have zero or more characters (perhaps the house has died out but is still mentioned in the show).

## Locations

---

The Locations table contains the following attributes:

- ID (required)
- Name
- Continent
- Slave (status as a slave or free state)

Locations can be associated with zero or more houses (Castle Black knows no house, while several houses may be vying for or sharing power in another location). Locations can be the birthplace of zero or more characters.

## Appearances

---

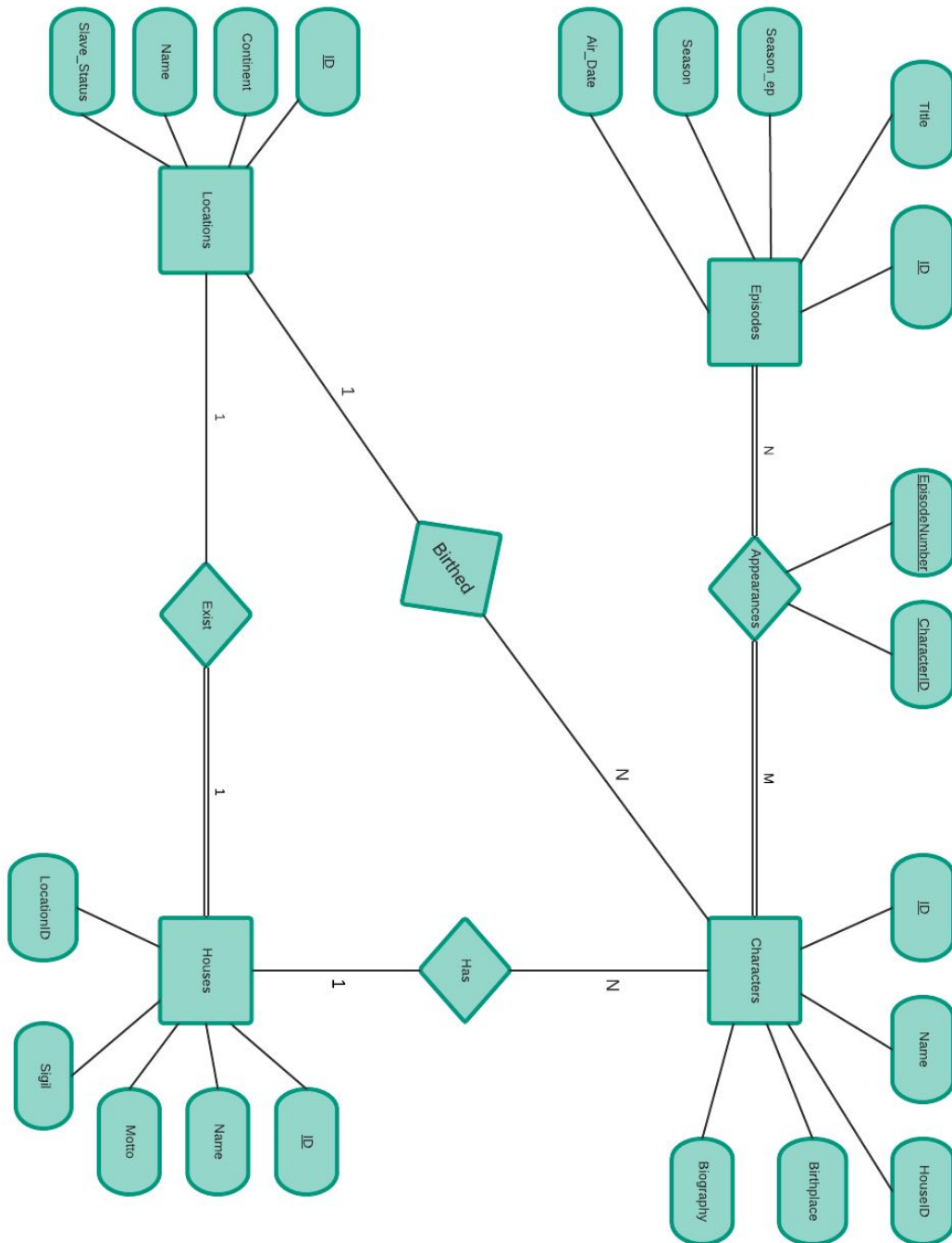
The Appearances table contains the following attributes:

- Episode ID (references Episodes, required)
- Character ID (references Characters, required)

All characters can appear in zero or more episodes (a character could be referenced by never appear). Obviously a character who does not appear in the show wouldn't be reflected in this table. Appearances are tracked by Episode's ID and Character's ID.

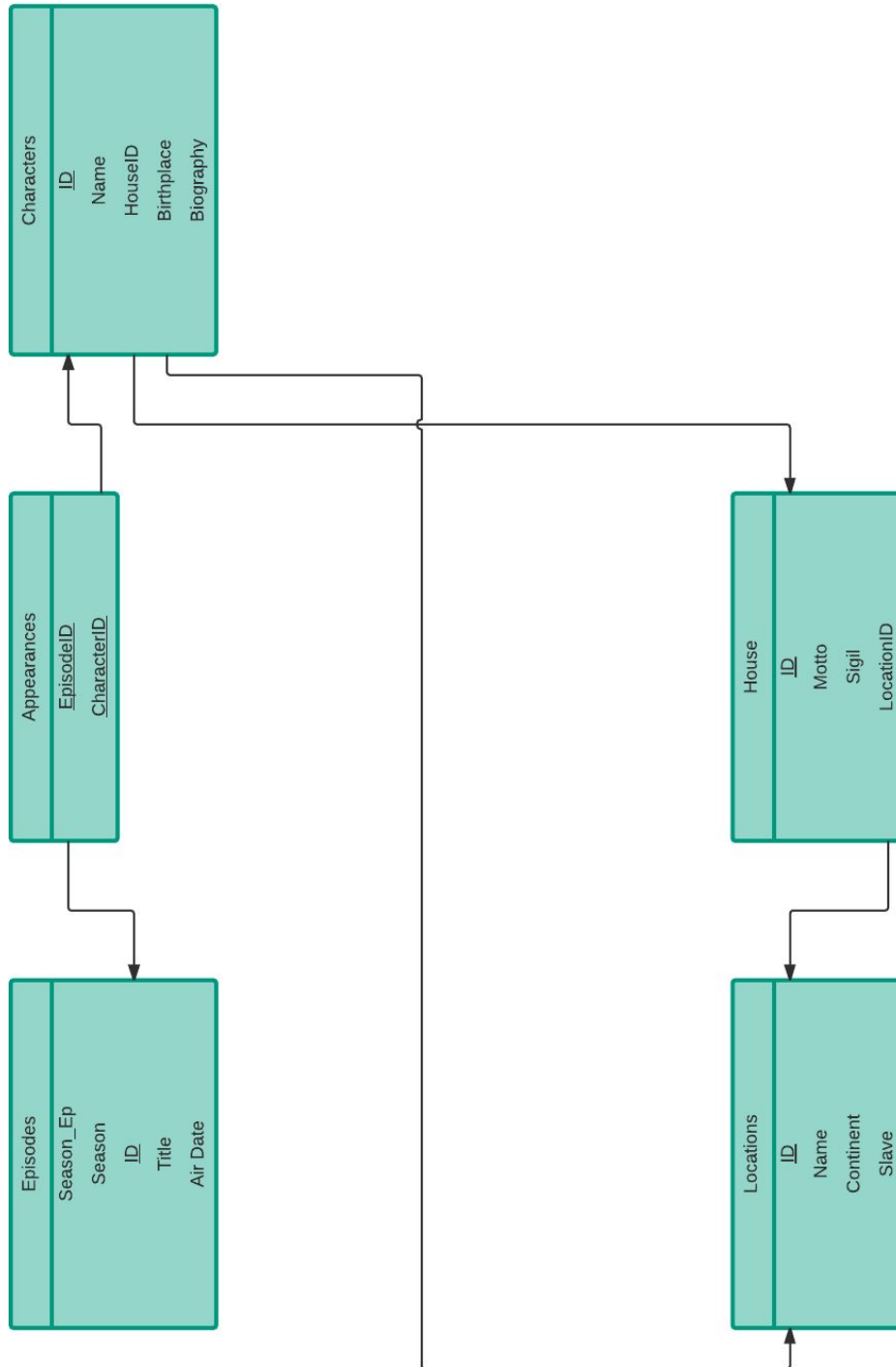
## ER Diagram: Link to ER + Schema

(<https://www.lucidchart.com/invitations/accept/6f8261c9-341f-4160-83c4-0798a7d0063e>)



Schema: Same link as above different tab at bottom of the site

(<https://www.lucidchart.com/invitations/accept/6f8261c9-341f-4160-83c4-0798a7d0063e>)



### **Data Definition Queries:**

```
SET FOREIGN_KEY_CHECKS=0;  
DROP TABLE IF EXISTS `Locations`;  
SET FOREIGN_KEY_CHECKS=1;
```

```
CREATE TABLE `Locations`(  
  `ID` INT NOT NULL AUTO_INCREMENT,  
  `Name` VARCHAR(255),  
  `Continent` VARCHAR(255),  
  `Slave` TINYINT,  
  UNIQUE(`Name`),  
  PRIMARY KEY(`ID`)  
)ENGINE=InnoDB;
```

```
SET FOREIGN_KEY_CHECKS=0;  
DROP TABLE IF EXISTS `Houses`;  
SET FOREIGN_KEY_CHECKS=1;
```

```
CREATE TABLE `Houses`(  
  `ID` INT NOT NULL AUTO_INCREMENT,  
  `Name` VARCHAR(255) NOT NULL,  
  `Motto` VARCHAR(255),  
  `Sigil` VARCHAR(255),  
  `LocationID` INT,  
  UNIQUE(`Name`),  
  PRIMARY KEY(`ID`),  
  FOREIGN KEY(`LocationID`) REFERENCES `Locations`(`ID`)  
    ON DELETE SET NULL  
)ENGINE=InnoDB;
```

```
SET FOREIGN_KEY_CHECKS=0;
DROP TABLE IF EXISTS `Characters`;
SET FOREIGN_KEY_CHECKS=1;
```

```
CREATE TABLE `Characters` (
  `ID` INT NOT NULL AUTO_INCREMENT,
  `Name` VARCHAR(255) NOT NULL,
  `HouseID` INT,
  `Birthplace` INT,
  `Biography` VARCHAR(255),
  UNIQUE(`Name`),
  PRIMARY KEY(`ID`),
  FOREIGN KEY(`HouseID`) REFERENCES `Houses`(`ID`)
    ON DELETE SET NULL,
  FOREIGN KEY(`Birthplace`) REFERENCES `Locations`(`ID`)
    ON DELETE SET NULL
)ENGINE=InnoDB;
```

```
SET FOREIGN_KEY_CHECKS=0;
DROP TABLE IF EXISTS `Episodes`;
SET FOREIGN_KEY_CHECKS=1;
```

```
CREATE TABLE `Episodes` (
  `ID` INT NOT NULL AUTO_INCREMENT,
  `Season` INT NOT NULL,
  `Season_ep` INT NOT NULL,
  `Title` VARCHAR(255) NOT NULL,
  `Air_Date` DATE NOT NULL,
  UNIQUE(`Season`, `Season_ep`),
  PRIMARY KEY(`ID`)
)ENGINE=InnoDB;
```

```

SET FOREIGN_KEY_CHECKS=0;
DROP TABLE IF EXISTS `Appearances`;
SET FOREIGN_KEY_CHECKS=1;

CREATE TABLE `Appearances`(
  `EpisodeID` INT NOT NULL,
  `CharacterID` INT NOT NULL,
  PRIMARY KEY(`EpisodeID` , `CharacterID`),
  FOREIGN KEY(`EpisodeID`) REFERENCES `Episodes`(ID)
    ON DELETE CASCADE,
  FOREIGN KEY(`CharacterID`) REFERENCES `Characters`(ID)
    ON DELETE CASCADE
)ENGINE=InnoDB;

```

### **Pre-population Queries for Data Table:**

```

INSERT INTO Locations (ID, Name, Continent, Slave)
VALUES
  (1, 'None', NULL, NULL), # Empty location for dropdown menu
  (2, 'Oldtown', 'Westeros', 0),
  (3, 'Lannisport', 'Westeros', 0),
  (4, 'Gulltown', 'Westeros', 0),
  (5, 'White Harbor', 'Westeros', 0),
  (6, 'Qarth', 'Essos', 1),
  (7, 'Braavos', 'Essos', 0),
  (8, 'Volantis', 'Essos', 1),
  (9, 'Pentos', 'Essos', 1),
  (10, 'Winterfell', 'Westeros', 0),
  (11, 'Casterly Rock', 'Westeros', 0),
  (12, 'Dragonstone', 'Westeros', 0),
  (13, 'Highgarden', 'Westeros', 0),
  (14, "Storm's End", 'Westeros', 0),
  (15, 'Bear Island', 'Westeros', 0),
  (16, 'Sunspear', 'Westeros', 0),
  (17, "King's Landing", 'Westeros', 0)
;

```



```
INSERT INTO Houses (ID, Name, Motto, Sigil, LocationID)
VALUES
```

```
(1, 'None', NULL, NULL, NULL), # Empty house for dropdown menu
(2, 'Lannister', 'Hear me roar', 'Lion', 11),
(3, 'Targaryen', 'Fire and blood', 'Dragon', 12),
(4, 'Tyrell', 'Growing strong', 'Rose', 13),
(5, 'Baratheon', 'Ours is the fury', 'Stag', 14),
(6, 'Mormont', 'Here we stand', 'Bear', 15),
(7, 'Martell', 'Unbowed, unbent, unbroken', 'Sun and spear', 16),
(8, 'Stark', 'Winter is coming', 'Direwolf', 10)
;
```

```
INSERT INTO `Characters` (ID, Name, HouseID, Birthplace, Biography)
VALUES
```

```
(1, 'Jon Snow', 8, NULL, "He knows nothing"),
(2, 'Eddard Stark', 8, 10, "Was head of House Stark, and Warden of the North"),
(3, 'Arya Stark', 8, 10, "Trained as a Faceless Man, on the path of revenge for her family"),
(4, 'Tyrion Lannister', 2, 11, "For what he lacks in brawn, he makes up in wit. Currently
assisting Daenerys Targaryen in the reclamation of her throne"),
(5, 'Daenerys Targaryen', 3, 12, "Just watch the clip of her introducing herself to Jon, enough
said"),
(6, 'Joffrey Baratheon', 5, 17, "Everyone hates him")
;
```

```
INSERT INTO `Episodes` (ID, Season, Season_ep, Title, Air_Date)
VALUES
```

```
(1, 1, 1, "Winter Is Coming", '2011-04-17'),
(20, 2, 10, "Valar Morghulis", '2012-06-03'),
(29, 3, 9, "The Rains of Castamere", '2012-06-03'),
(31, 4, 1, "Two Swords", '2014-04-06'),
(42, 5, 2, "The House of Black and White", '2015-04-19'),
(53, 6, 3, "Oathbreaker", '2016-05-08'),
(61, 7, 1, "Dragonstone", '2017-07-16'),
(62, 7, 2, "Stormborn", '2017-07-23'),
(63, 7, 3, "The Queen's Justice", '2017-07-30'),
(64, 7, 4, "The Spoils of War", '2017-07-06'),
(65, 7, 5, "Eastwatch", '2017-08-13'),
(66, 7, 6, "Beyond the Wall", '2017-08-20'),
(67, 7, 7, "The Dragon and the Wolf", '2017-08-27')
```

;

```
INSERT INTO `Appearances`(EpisodeID, CharacterID)
VALUES
  (29, 1),
  (29, 3),
  (29, 5),
  (66, 1),
  (66, 3),
  (66, 5),
  (1, 1),
  (1, 2),
  (1, 3),
  (1, 4),
  (1, 5),
  (1, 6)
;
```

#### **Data Manipulation Queries:**

- Queries are page specific:

--Appearances Page--

--Selecting initial data for table population--

```
SELECT E.ID as EID, E.Title as Title, E.Season as Season, E.season_ep as Episode, C.ID as
CID, C.Name as Name FROM `Appearances` A
  INNER JOIN `Episodes` E ON A.EpisodeID = E.ID
  INNER JOIN `Characters` C on A.CharacterID = C.ID
  ORDER BY C.Name;
```

--Filtering of episodes a CHARACTER appears in

```
SELECT E.ID as EID, E.Title as Title, E.Season as Season, E.Season_ep as Episode, C.ID as
CID, C.Name as Name FROM Appearances A
  INNER JOIN Episodes E ON A.EpisodeID = E.ID
  INNER JOIN Characters C ON A.CharacterID = C.ID
  WHERE C.ID = [USER_Inputs_Name];
```

--Get all characters that appear in our Relationship table--

```
SELECT E.ID as EID, E.Title as Title, E.Season as Season, E.Season_ep as Episode, C.ID as  
CID, C.Name as Name FROM Appearances A  
  INNER JOIN Episodes E ON A.EpisodeID = E.ID  
  INNER JOIN Characters C ON A.CharacterID = C.ID  
GROUP BY C.Name ORDER BY C.Name;
```

--Get DISTINCT seasons that appear in our Episodes table--

```
SELECT distinct(Season) From `Episodes` ORDER BY Season;
```

--Get all the characters name + ID from Characters table--

```
SELECT ID, Name FROM `Characters`;
```

--Get episodes from SPECIFIED season--

```
SELECT ID, Season, Season_ep as episodes FROM `Episodes`  
  WHERE Season = [UserChoice];
```

--Inserting an Episode + Character relationship--

```
INSERT INTO `Appearances` (EpisodeID,CharacterID)  
  VALUES [UserSelectsEpisode, UserSelectsChar];
```

-- Deleting a relationship from the table --

```
DELETE FROM `Appearances`  
  WHERE EpisodeID = [User] AND CharacterID = [User];  
/*User clicks on delete button linked with each row, row will have EpisodeID and CharacterID  
linked with the delete button accordingly*/
```

-----

--Characters page

--Select initial characters for table population --

```
SELECT C.ID, C.Name, H.Name AS House, L.Name AS Location, C.Biography FROM  
Characters C  
  LEFT JOIN Houses H ON C.HouseID = H.ID  
  LEFT JOIN Locations L ON C.Birthplace = L.ID;
```

--Gets 1 SPECIFIED character's information for update--

```
SELECT ID, Name, HouseID, Birthplace, Biography FROM `Characters`  
  WHERE id = [UserChoice];
```

--Get Location Name to replace LocationID with via Joins--

```
SELECT ID, Name FROM `Locations`;
```

--Update a specific Character and their information --

```
UPDATE `Characters`
```

```
  SET Name = [UserParam] , HouseID = [UserParam], Birthplace = [UserParam], Biography =  
[UserParam]
```

```
  WHERE ID = [UserInput];
```

--Inserting a new characters

```
INSERT INTO `Characters` (Name,HouseID,Birthplace,Biography)
```

```
  VALUES [User,User,User,User];
```

--Deleting a character--

```
DELETE FROM `Characters` WHERE ID = [UserChoice];
```

-----

--Episodes Page --

--Initial data selection to populate table

```
SELECT ID, Season, Season_ep, Title, Air_Date  
FROM `Episodes`;
```

--Grab 1 Single episode

```
SELECT ID, Season, Season_ep, Title, Air_Date  
FROM `Episodes` WHERE ID = [UserInput];
```

--Updating Episode Info--

```
UPDATE `Episodes`
```

```
  SET Season = [UserParam] , Season_ep = [UserParam], Title = [UserParam], Air_Date =  
[UserParam]
```

```
  WHERE ID = [UserChoice];
```

--Inserting new characters --

```
INSERT INTO `Episodes` (Season,Season_ep,Title,Air_Date)
```

```
  VALUES [User,User,User,UserParam];
```

--Deleting Episodes --

```
DELETE FROM `Episodes` WHERE ID = [UserChoice];
```

-----

--Houses Page --

--Selecting house information for initial table --

```
SELECT H.ID, H.Name as Name, H.Motto as Motto, H.Sigil as Sigil, L.Name as Location  
FROM `Houses` H  
INNER JOIN `Locations` L ON H.LocationID = L.ID;
```

--Gets all locations ID and Name --

```
SELECT ID, Name FROM `Locations`;
```

--Grabs specific HOUSE to update info for --

```
SELECT ID, Name, Motto, Sigil, LocationID  
FROM `Houses` WHERE ID = [UserChoice];
```

--Update House Info--

```
UPDATE `Houses` SET Name = [UserInputs], Motto = [UserInputs], Sigil = [UserInput],  
LocationID = [UserInputs]  
WHERE ID = [UserChoice];
```

--Insert New House --

```
INSERT INTO `Houses` (Name,Motto,Sigil,LocationID)  
VALUES [User,User,User,UserInputs];
```

--Delete House --

```
DELETE FROM `Houses` WHERE ID = [UserChoice];
```

-----

--Locations page

--Initial locations data selection for data table pop. --

```
SELECT ID, Name, Continent, Slave FROM `Locations`  
WHERE Name != 'None';
```

--Selects 1 location --

```
SELECT ID, Name, Continent, Slave FROM `Locations`  
WHERE ID = [UserInput];
```

--Update a location's info --

```
UPDATE `Locations` SET Name = [UserInputs], Continent = [UserInputs], Slave = [UserInputs]  
WHERE ID = [UserChoice];
```

--Add new Location --

```
INSERT INTO `Locations` (Name,Continent,Slave)  
Values [User,User,UserInput];
```

-- Delete a location --

```
DELETE FROM `Locations` WHERE ID = [UserChoice];
```