CS 340 Final Project for Sarah Cawley

Hosted At: http://web.engr.oregonstate.edu/~cawleys/CS340/startrek.php

Outline

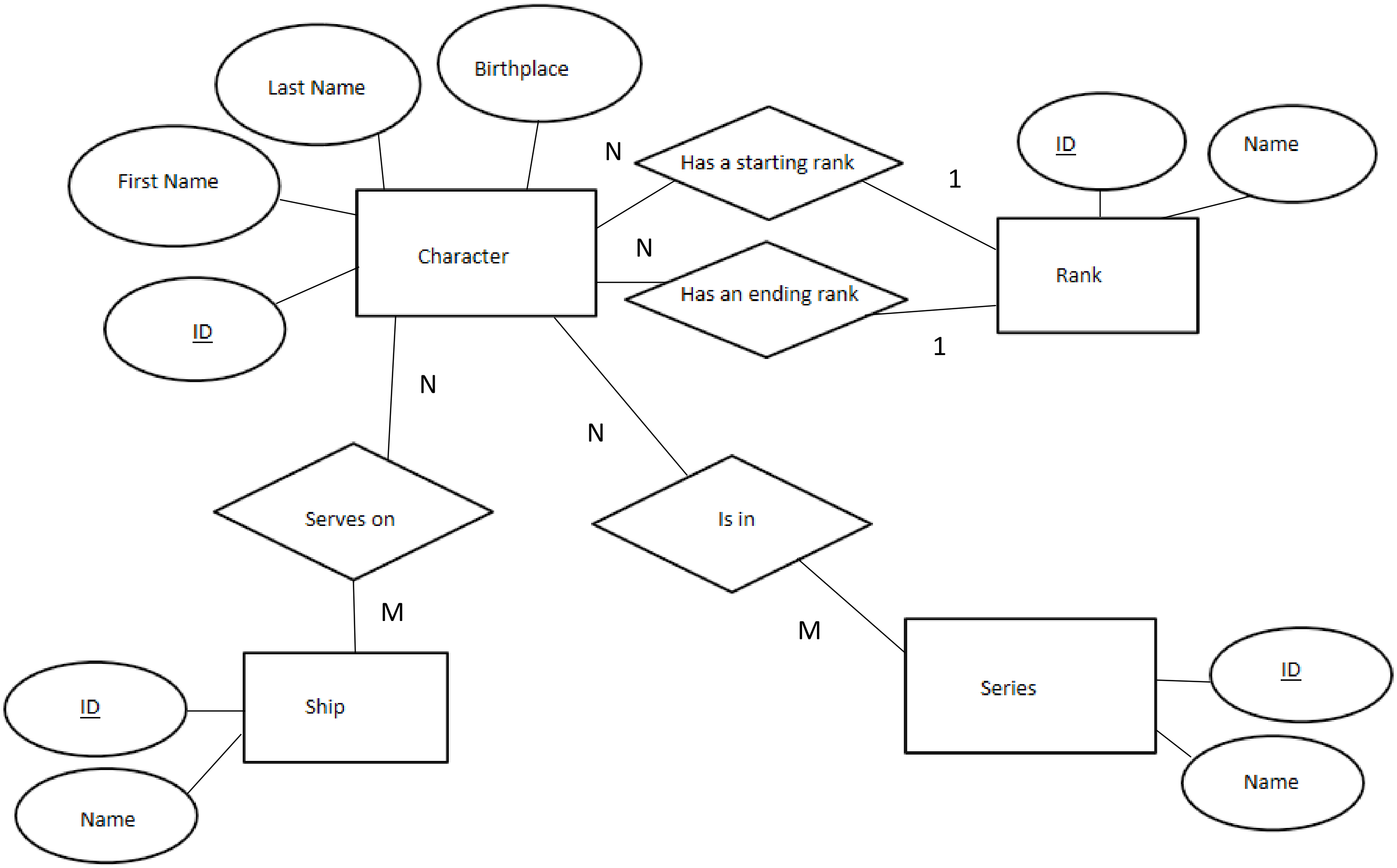
I have created a Star Trek data base that tracks character's interactions with ships, series, starting and ending rankings, and birth planets. I think this is valuable and fun to see which characters have appeared in series outside of their stand show, also to see where the characters were born (or created).

The database is able to view information about characters, add new characters, edit information about the character's series appeared on, ships served on, and starting and ending ranks. Characters can also be deleted from the database.

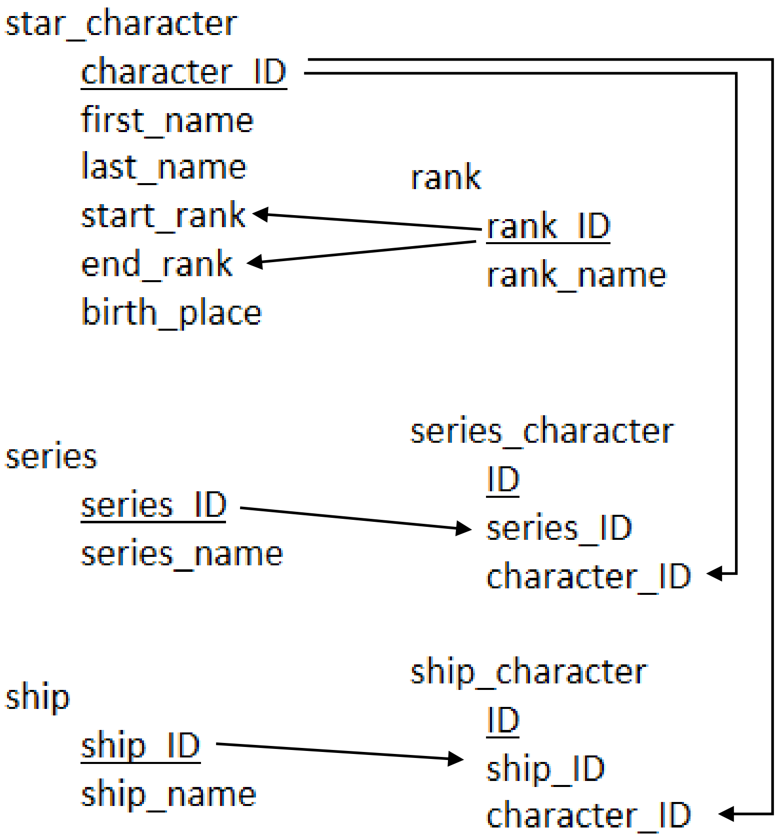
Database Outline in Words

- A character has a, ID, first name, last name, starting rank, ending rank, and birthplace. A character is uniquely identified by their ID
- A series has a series ID and a series name. A series is uniquely identified by its ID
- A ship has an ID and a name. A ship is uniquely identified by its ID
- A rank has a rank ID and a rank name. A rank is uniquely identified by its ID
- A character can be related to multiple ships and series, a character can be related to at most 1 starting and ending rank
- A rank may be associated with many characters
- A ship may be associated with many characters
- A series may be associated with many characters

ER Diagram of Database



Database Schema



star_character
- character_ID
- first_name
- last_name
- start_rank
- end_rank
- birth_place

rank
- rank_ID
- rank_name

series
- series_ID
- series_name

series_character
- ID
- series_ID
- character_ID

ship
- ship_ID
- ship_name

ship_character
- ID
- ship_ID
- character_ID

Table Creation Queries

```
-- Holds all ship information
CREATE TABLE ship (
	ship_ID int  AUTO_INCREMENT NOT NULL,
	ship_name  VARCHAR(255) NOT NULL ,
	PRIMARY KEY (ship_ID)
)ENGINE=InnoDB;

-- holds ranking information (admiral, captain, etc.)
CREATE TABLE rank (
	rank_id int AUTO_INCREMENT NOT NULL,
	rank_name VARCHAR(255),
	PRIMARY KEY (rank_id)
)ENGINE=InnoDB;

-- main table that everything is connected to.
CREATE TABLE star_character (
	character_ID int AUTO_INCREMENT NOT NULL,
	first_name VARCHAR(255),
	last_name VARCHAR(255) ,
	start_rank int,
	end_rank int,
	birth_place VARCHAR(255),
	PRIMARY KEY (character_ID),
	FOREIGN KEY (start_rank) REFERENCES rank(rank_id),
	FOREIGN KEY (end_rank) REFERENCES rank(rank_id)
)ENGINE=InnoDB DEFAULT CHARSET = utf8;

-- connecting table for characters and which ships they have served on
CREATE TABLE ship_character (
	id int AUTO_INCREMENT NOT NULL,
	character_ID int,
	ship_ID int,
	PRIMARY KEY (id),
	FOREIGN KEY (character_ID) REFERENCES star_character(character_ID),
	FOREIGN KEY (ship_ID) REFERENCES ship(ship_ID)
)ENGINE=InnoDB DEFAULT CHARSET = utf8;

-- holds series information (id and name)
CREATE TABLE series (
	series_ID int AUTO_INCREMENT NOT NULL,
	series_name VARCHAR(255) NOT NULL,
	PRIMARY KEY (series_ID)
)ENGINE=InnoDB DEFAULT CHARSET = utf8;
```

```
-- connecting table for characters and the series they have been on
CREATE TABLE series_character (
	id int AUTO_INCREMENT NOT NULL,
	series_ID int,
	character_ID int,
	PRIMARY KEY (id),
	FOREIGN KEY (series_ID) REFERENCES series(series_ID),
	FOREIGN KEY (character_ID) REFERENCES star_character(character_ID)
)ENGINE=InnoDB DEFAULT CHARSET = utf8;
```

General Use Queries

```
//calls first and last names from star_character line 31
SELECT first_name AS 'First Name', last_name AS 'Last Name' FROM star_character;

//shows first name, last name, starting rank, ending rank, and birthplace for character line 92
SELECT first_name AS 'First Name',
	last_name AS 'Last Name',
	(select rank_name from rank where rank_id = start_rank) as 'Starting Rank',
	(select rank_name from rank where rank_id = end_rank) as 'Ending Rank',
	birth_place AS 'Birth Place'
	FROM star_character
	WHERE character_ID =  [character_ID];

//Show ships served on by a character line 121
SELECT ship_name FROM ship
	INNER JOIN ship_character ON ship_character.ship_ID = ship.ship_ID
	WHERE ship_character.character_ID = [ship_character.character_ID];

//Show series appeared on by a character line 141
SELECT series_name FROM series
	INNER JOIN series_character ON series_character.series_ID = series.series_ID
	WHERE series_character.character_ID = [series_character.character_ID];

//insert first, last, birthplace line 188
INSERT INTO star_character(first_name, last_name, birth_place)
	VALUES ([first_name], [last_name] , [birth_place]);

//for pull down for series line 230
SELECT series_ID, series_name FROM series;

//for pull down for ship names line 253
SELECT ship_ID, ship_name FROM ship

//for dropdown for starting rank line 272
SELECT rank_ID, rank_name FROM rank
```

```
//connect a character to a series line 333
INSERT INTO series_character(character_ID, series_ID)
            VALUES ([character], [series ]);

//connect a character to a ship line 352
INSERT INTO ship_character(character_ID, ship_ID)
            VALUES ([character],[ship]);

//update the beginning rank of a character line 371
UPDATE star_character SET start_rank =  [startingRank] WHERE character_ID = [character];

//update the ending rank of a character line 379
UPDATE star_character SET end_rank =  [endingRank] WHERE character_ID = [character];

//create a new ranking level line 395
INSERT INTO rank(rank_name) VALUES ([newRank']);

//create a new series line 423
INSERT INTO series(series_name) VALUES ([newSeries]);

//create new ship line 448
INSERT INTO ship(ship_name) VALUES ([newShip]);

//delete character line 494
DELETE FROM star_character WHERE character_ID = [character];
```