Max MacEachern
CS 340 – Intro to Databases
Final Project

Outline

The database outlined in this document holds information relating to the Star Trek universe, based on the TV series created by Gene Roddenberry. The show, films and other media all follow the adventures throughout the galaxy of the starship *Enterprise* and her crew. Besides the human race, represented by the United Federation of Planets, there are multiple other species, each with their own people, ships and distinct qualities.

From all the different characters and the relationships between them to the ships the characters work on, there is a tremendous amount of entities and relationships in the universe, making it perfect to be put in a database. The sheer amount of characters alone makes understanding the nature of the universe complicated and a database would potentially help those who are interested in learning more about the series. Further, there are multiple shows featuring ships, planets and characters from different times in history, further increasing the complexity.

Database Outline

Our database has four main entities: characters, planets, ships, and skills. The character table holds rows of people who exist in Star Trek, each featuring an ID (that is auto-incrementing), first name, last name, homeworld,. The ID and first names of characters must exist, but the other attributes are not required. The last name, homeworld, and skill rows default to null. The primary key is the ID. Further, the foreign key 'homeworld' references planets.id.

In the series, while the characters live on starships for the most part, they are always adventuring to new planets or returning home to their homeworld. For this reason, planets are also an entity in the database, featuring an ID (that is auto-incrementing) and name row that each must have a value. The primary key is the ID.

Similarly, ships are also a critical part of the show and are assigned their own entity, as well. Again, these ships have ID (that is auto-incrementing) and name rows that must not be null. The primary key is the ID.

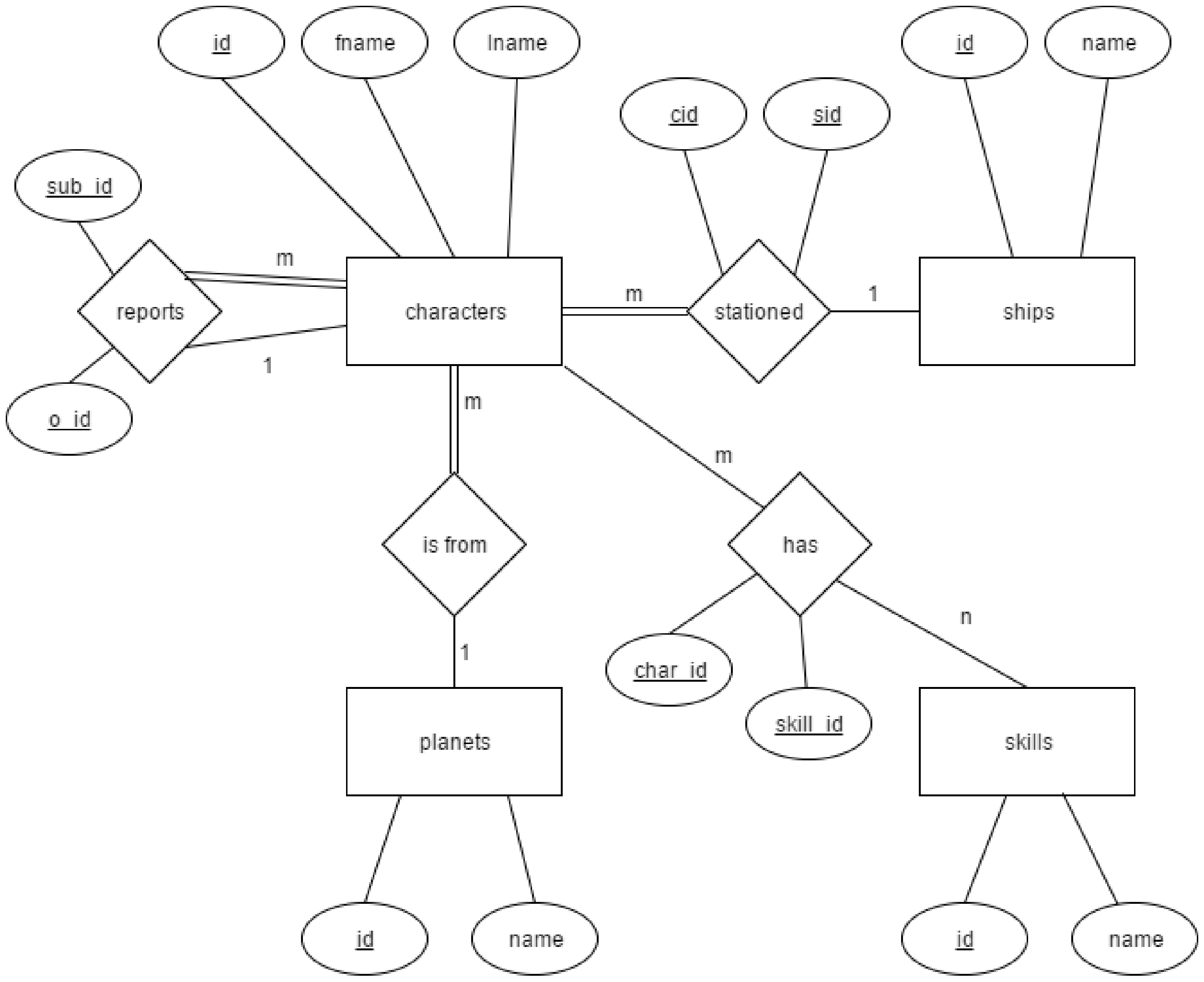
The last main entity are the skills characters possess on the show. Again, there are two rows, ID and name, that are each required. The ID row auto-increments as well and is the primary key of the table.

There are also three relationship tables: 'stationed', 'reports,' and 'stskills.' The 'stationed' table describes which ship a character is stationed on. This table features two rows, 'cid,' for the character ID, and 'sid,' for the ship ID. Both cannot be null and are the primary keys of the table. This relationship is described as a one to many relationship, as a character can only be assigned to one ship at any given time, while a ship can be the assignment of many different characters. Here both 'cid' and 'sid' are foreign keys that reference characters.id and ships.id, respectively. Both delete on cascade.

The 'reports' table describes how characters are subordinate to a single officer while a single officer may have many different characters reporting to them at any given time. There are two rows in this table: 'sub_id' and 'o_id', for subordinates and officers, respectively. Both rows must not be null and are the primary keys of the table. This relationship is described as a one to many relationship, as a character can only have one officer, while an officer can have many subordinates. Here both ‘sub_id’ and ‘o_id’ are foreign keys that reference characters.id and characters.id, respectively. Both delete on cascade.

The ‘stskills’ table describes the various skills that a given characters can have. There are two rows in this table: ‘char_id’ and ‘skill_id’, for characters and skills, respectively. Both rows must not be null and are the primary keys of the table. This relationship is described as a many to many relationship, as a character can obtain many skills and a given skill can be attributed to many different characters. Here both ‘char_id’ and ‘skill_id’ are foreign keys that reference characters.id and skills.id, respectively. Both delete on cascade.

<u>ER Diagram</u>



<u>Schema</u>

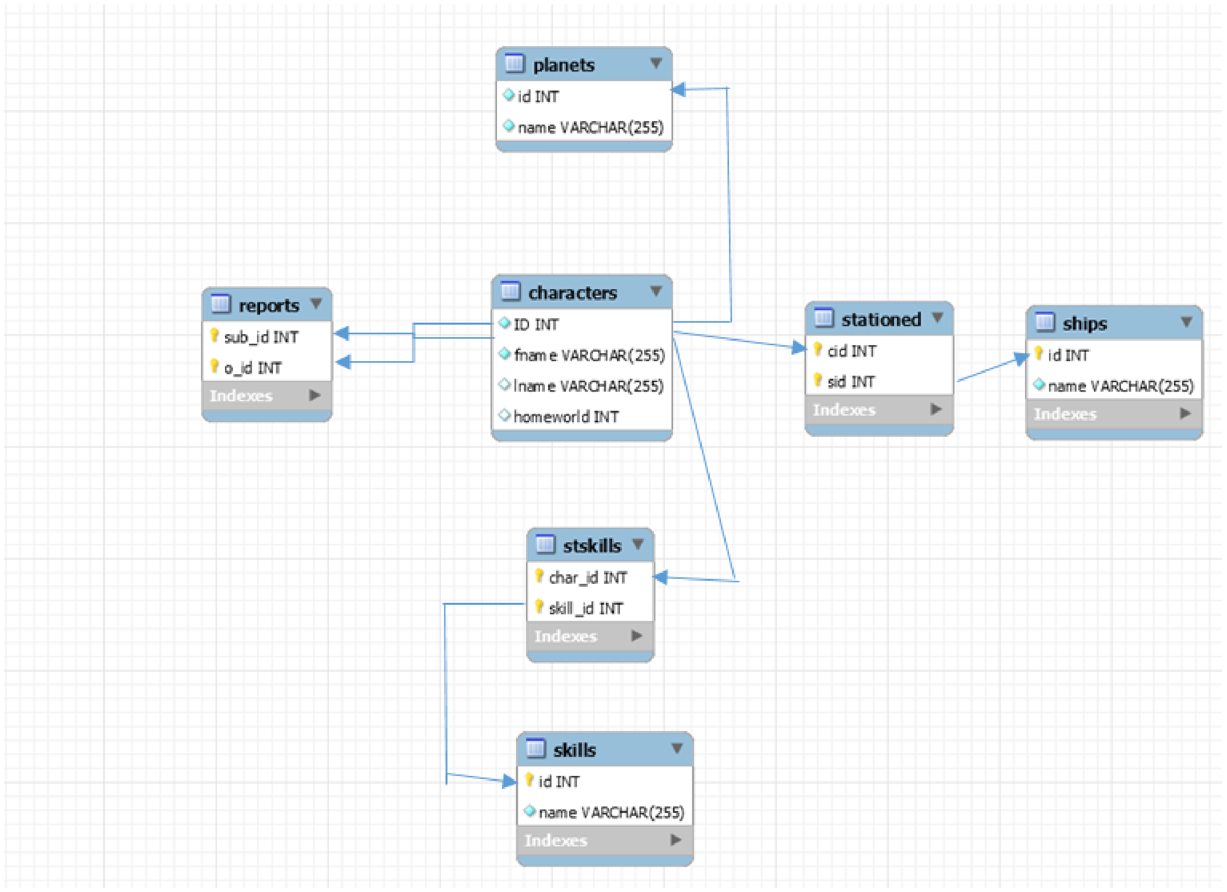


Table Creation Queries

```
DROP TABLE IF EXISTS `characters`;

CREATE TABLE `characters` (

  `id` int(11) NOT NULL AUTO_INCREMENT,

  `fname` varchar(255) NOT NULL,

  `lname` varchar(255) DEFAULT NULL,

  `homeworld` int(11) DEFAULT NULL,

  PRIMARY KEY (`id`),

  FOREIGN KEY (`homeworld`) REFERENCES `planets` (`id`))

ENGINE=INNODB

DROP TABLE IF EXISTS `planets`;


CREATE TABLE `planets` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(255) NOT NULL,
  PRIMARY KEY (`id`),
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=latin1;

DROP TABLE IF EXISTS `ships`;

CREATE TABLE `ships` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(255) NOT NULL,
  PRIMARY KEY (`id`),
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=latin1;

DROP TABLE IF EXISTS `skills`;

CREATE TABLE `skills` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(255) NOT NULL,
  PRIMARY KEY (`id`),
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=latin1;

CREATE TABLE `stationed` (
	`cid` INT NOT NULL,
	`sid` INT NOT NULL,
	PRIMARY KEY (`cid`, `sid`),
	FOREIGN KEY (`cid`) REFERENCES characters (`id`) ON DELETE CASCADE,
	FOREIGN KEY (`sid`) REFERENCES ships (`id`) ON DELETE CASCADE
) ENGINE = InnoDB;
```

```
CREATE TABLE `reports` (
	`sub_id` INT NOT NULL,
	`o_id` INT NOT NULL,
	PRIMARY KEY (`sub_id`, `o_id`),
	FOREIGN KEY (`sub_id`) REFERENCES characters (`id`) ON DELETE CASCADE,
	FOREIGN KEY (`o_id`) REFERENCES characters (`id`) ON DELETE CASCADE
) ENGINE = InnoDB;

CREATE TABLE `stskills` (
	`char_id` INT NOT NULL,
	`skill_id` INT NOT NULL,
	PRIMARY KEY (`char_id`, `skill_id`),
	FOREIGN KEY (`char_id`) REFERENCES characters (`id`) ON DELETE CASCADE,
	FOREIGN KEY (`skill_id`) REFERENCES skills (`id`) ON DELETE CASCADE
) ENGINE = InnoDB;
```

General Use Queries

INSERT

```
INSERT INTO characters(fname, lname, homeworld) VALUES ([fname], [lname],
[homeworld]);

INSERT INTO planets (name) VALUES ([name]);

INSERT INTO ships (name) VALUES ([name]);

INSERT INTO skills (name) VALUES ([name]);

INSERT INTO stationed (cid, sid) VALUES ([cid], [sid]);

INSERT INTO reports (sub_id, o_id) VALUES ([sub_id], [o_id]);

INSERT INTO stskills (char_id, skill_id) VALUES ([char_id], [skill_id]);
```

SELECT

```sql
SELECT characters.fname, characters.lname, planets.name FROM characters INNER JOIN
planets ON characters.homeworld = planets.id WHERE planets.id = [user_input]
```

DELETE

```sql
DELETE FROM characters WHERE fname = [fname] AND lname = [lname]
```