

**James Le**  
**Adam James Kniffin**  
**CS 340 Intro to Databases**  
**Final Project**

## **Outline**

The database outlined in this document holds information relating to the World of Warcraft universe. This is a universe based on a popular online role-playing game created by a company called Blizzard Entertainment. The game centers around two factions known as the alliance and the horde that are at constant war with one another. The world is massive and contains many different entities.

From all the different characters and the relationships between them to the zones they came from, the mounts that they used, the bosses that they face, and the skills that they may acquire would result in 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 game.

## **Database Outline**

Our database has five main entities: characters, bosses, zones, skills, and mounts. The character table holds rows of people who exist in Warcraft, each featuring an ID that is auto-incrementing, display name, class, race, and homeland. The ID, display name, race, and class must exist, but not other foreign attributes, such as homeland. The race, class, and homeland rows are default to null. The primary key is the ID. Also, the foreign key homeland references zones.id.

In the game, while characters may engage in activities in a certain zone for a period of time, they may also travel to other zones in the world, and return to their homeland location. For this reason, zones are also an entity in the database, featuring an ID that is auto-incrementing and a name row that must have a value to indicate that it exists in the world. The primary key is the ID. One zone can have many characters, but characters can only exist at one zone at any given point in time. This makes it a many to one relationship between characters and zones. A zone can contain zero or more characters, but a character needs to exist in a zone, thus making it a full participation for characters.

While traveling in the world, characters most likely would use mounts to make it more accessible to certain areas. These mounts are either acquired by summoning or purchasing in stores. Mounts have an ID that is auto-incrementing, and a name row that can't be null. The primary key is the ID.

We can't call it a role-playing game without enemies. In the World of Warcraft, bosses are a powerful entity that exists for characters to kill, level up, and acquire new skills. Bosses may exist in zones that are shared with characters. They may also have skills like characters do. Bosses will have an ID that is auto-incrementing, and a name row that is not null. A zone can contain zero or more bosses but bosses can only be at one zone at any given point in time. This

is a many to one relationship between bosses and the zones. Bosses also needs full participation as they must exist in a zone.

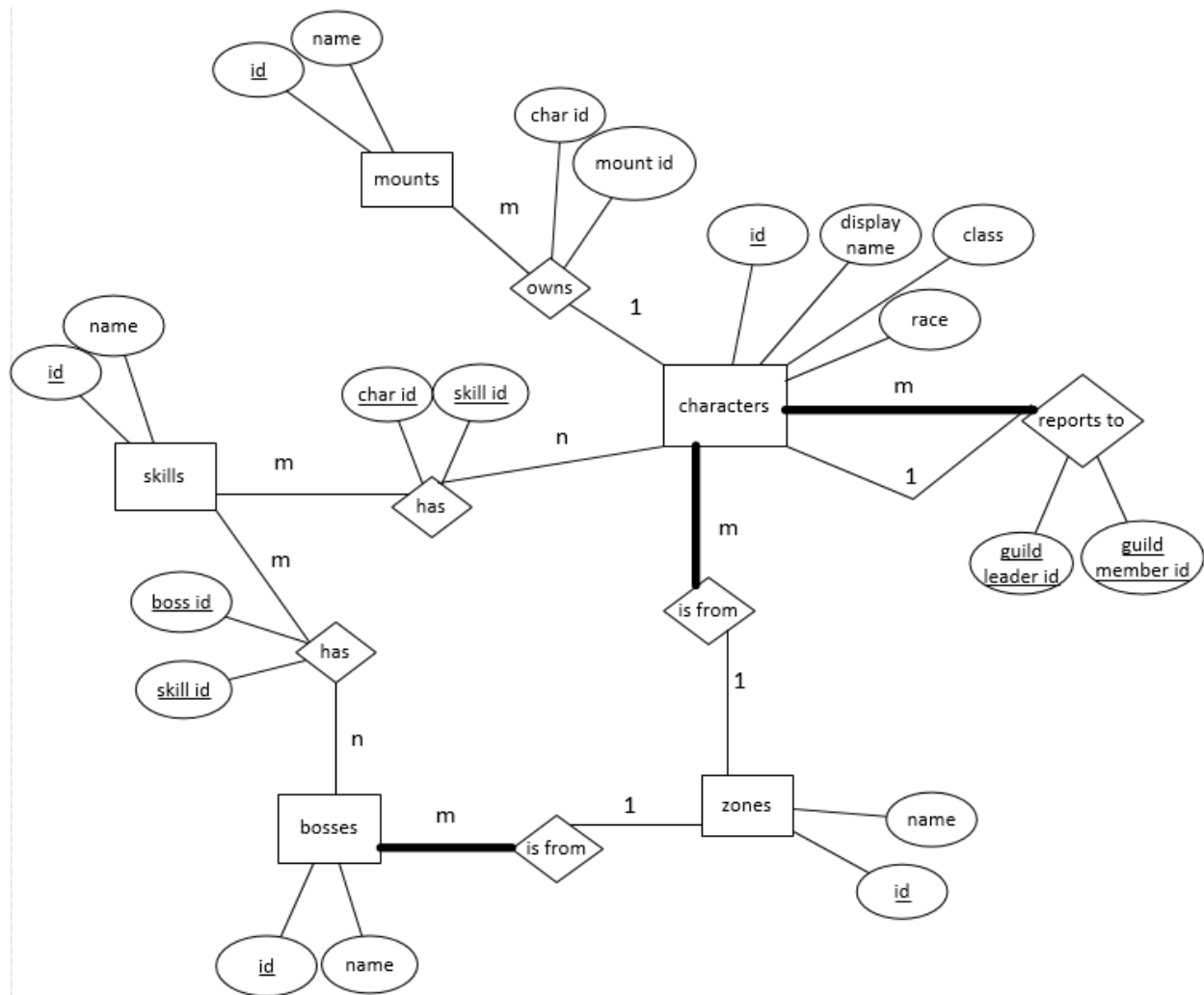
The last main entity are the skills characters and bosses possess in the game. Skills will have an ID that is auto-incrementing, and a name that are not null. The ID row is the primary key of the table.

There are four relationship tables in this database: 'owns', 'reports', 'chskills', and 'bskills'. The 'owns' table describes which mounts belongs to which character. This table features two rows, 'char\_id' for the character ID, and 'mount\_id' for the mount ID. Both cannot be null and are the primary keys of the table. This relationship is described as a many to one relationship as only one mount can be assigned to a character at any given time, but a character can have many mounts. Here both 'mount\_id' and 'char\_id' are foreign keys that references mounts.id and characters.id, respectively. Both delete on cascade.

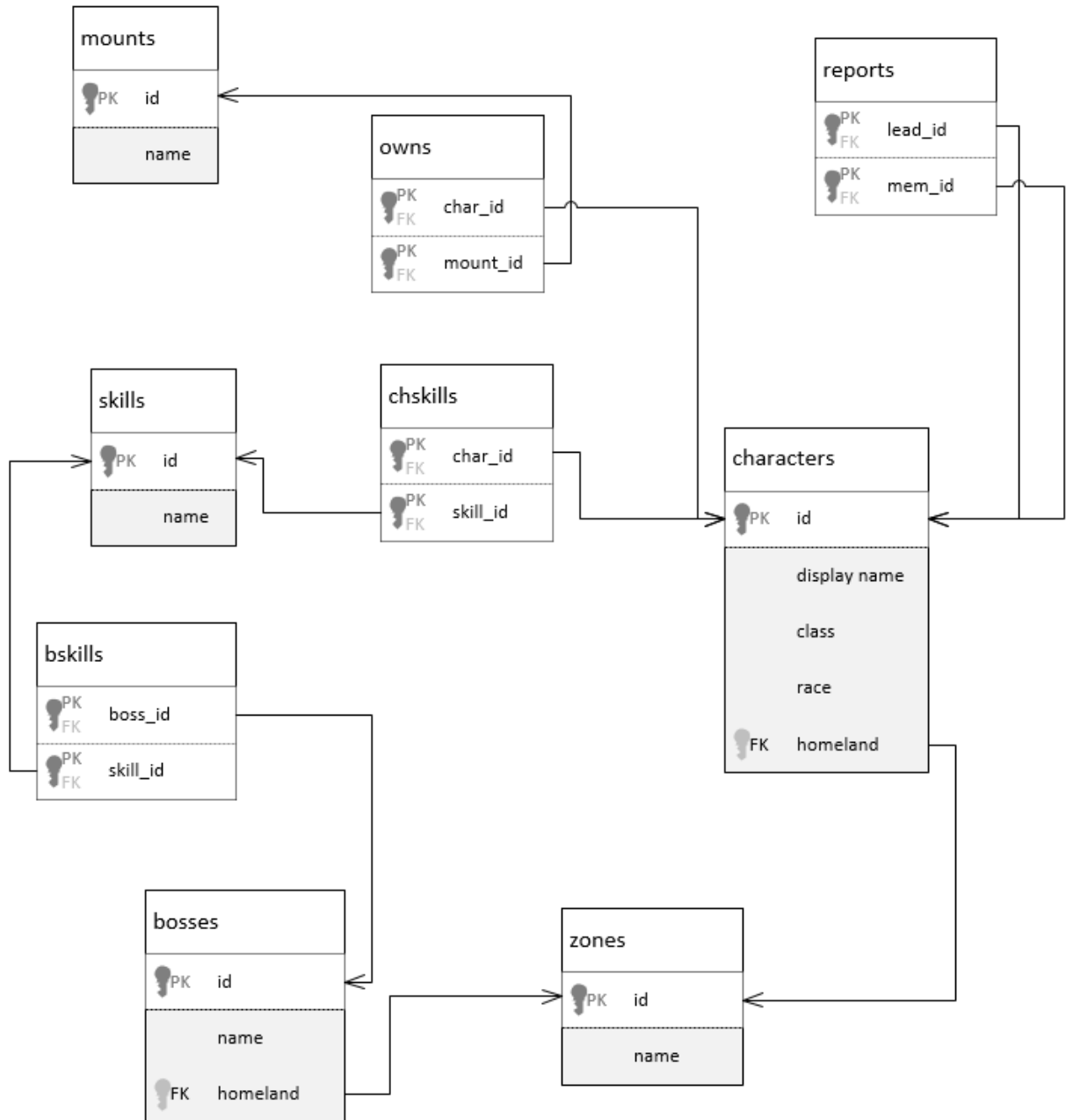
The 'reports' table describes how characters are members in a guild and follows the instruction of a single leader. Many members reports to a single guild leader, while a single guild leader can have many members reporting to them at any given time. There are two rows in this table: 'lead\_id' and 'mem\_id', for the leader and the members, respectively. Both rows can't 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 leader, while a leader can have many different members. Here both 'lead\_id' and 'mem\_id' are foreign keys that references to characters.id and characters.id, respectively. Both delete on cascade.

The 'chskills' and 'bskills' table describes the various skills that a given characters and bosses can have, respectively. There are two rows in both tables: 'char\_id' and 'skill\_id', for characters and skills table 'chskills', and 'boss\_id' and 'skill\_id' for bosses and skills table 'bskills'. These relationship tables are describe as a many to many relationship, as both characters and bosses can have many skills and a given skill can be attributed to many different characters and bosses. Here 'char\_id' and 'skill\_id' are foreign keys that references to characters.id and skills.id, and likewise, 'boss\_id' and 'skill\_id' are foreign keys that references to bosses.id and skills.id, both respectively. All are delete on cascade.

## ER Diagram of Database



## Database Schema



## Table Creation Queries

```
DROP TABLE IF EXISTS `zones`;
DROP TABLE IF EXISTS `characters`;
DROP TABLE IF EXISTS `bosses`;
DROP TABLE IF EXISTS `mounts`;
DROP TABLE IF EXISTS `skills`;
```

```
DROP TABLE IF EXISTS `owns`;
DROP TABLE IF EXISTS 'reports';
DROP TABLE IF EXISTS 'chskills';
DROP TABLE IF EXISTS 'bskills';
```

```
CREATE TABLE zones (
  id int NOT NULL AUTO_INCREMENT,
  name varchar(255) NOT NULL,
  PRIMARY KEY (id)
)ENGINE = InnoDB;
```

```
CREATE TABLE characters (
  id int NOT NULL AUTO_INCREMENT,
  dname varchar(255) NOT NULL,
  class varchar(255) DEFAULT NULL,
  race varchar(255) DEFAULT NULL,
  homeland int DEFAULT NULL,
  PRIMARY KEY (id),
  FOREIGN KEY (homeland) REFERENCES zones(id)
)ENGINE = InnoDB;
```

```
CREATE TABLE bosses (
  id int NOT NULL AUTO_INCREMENT,
  name varchar(255) NOT NULL,
  homeland int DEFAULT NULL,
  PRIMARY KEY (id)
  FOREIGN KEY (homeland) REFERENCES zones(id)
)ENGINE = InnoDB;
```

```
CREATE TABLE mounts (
  id int NOT NULL AUTO_INCREMENT,
  name varchar(255) NOT NULL,
  PRIMARY KEY (id)
)ENGINE = InnoDB;
```

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

```
CREATE TABLE owns (  
  mount_id int NOT NULL,  
  char_id int NOT NULL,  
  PRIMARY KEY (mount_id, char_id),  
  FOREIGN KEY (mount_id) REFERENCES mounts (id) ON DELETE CASCADE,  
  FOREIGN KEY (char_id) REFERENCES characters (id) ON DELETE CASCADE  
)ENGINE = InnoDB;
```

```
CREATE TABLE reports (  
  lead_id int NOT NULL,  
  mem_id int NOT NULL,  
  PRIMARY KEY (lead_id, mem_id),  
  FOREIGN KEY (lead_id) REFERENCES characters (id) ON DELETE CASCADE,  
  FOREIGN KEY (mem_id) REFERENCES characters (id) ON DELETE CASCADE  
)ENGINE = InnoDB;
```

```
CREATE TABLE chskills (  
  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;
```

```
CREATE TABLE bskills (  
  boss_id int NOT NULL,  
  skill_id int NOT NULL,  
  PRIMARY KEY (boss_id, skill_id),  
  FOREIGN KEY (boss_id) REFERENCES bosses (id) ON DELETE CASCADE,  
  FOREIGN KEY (skill_id) REFERENCES skills (id) ON DELETE CASCADE  
)ENGINE = InnoDB;
```

## General Use Queries

### INSERT

```
INSERT INTO characters (dname, class, race, homeland) VALUES ([dname], [class], [race], [homeland]);
```

```
INSERT INTO zones (name) VALUES ([name]);
```

```
INSERT INTO bosses (name) VALUES ([name]);
```

```
INSERT INTO mounts (name) VALUES ([name]);
```

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

```
INSERT INTO reports (lead_id, mem_id) VALUES ([lead_id], [mem_id]);
```

```
INSERT INTO owns (mount_id, char_id) VALUES ([mount_id], [char_id]);
```

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

```
INSERT INTO bskills (boss_id, skill_id) VALUES ([boss_id], [skill_id]);
```

### SELECT

```
SELECT characters.dname, characters.class, characters.race, zones.name  
FROM characters  
INNER JOIN zones ON characters.homeland = zones.id  
WHERE zones.id = [user_input]
```

```
SELECT bosses.name, zones.name  
FROM bosses  
INNER JOIN zones ON bosses.homeland = zones.id  
WHERE zones.id = [user_input]
```

### DELETE

```
DELETE FROM characters  
WHERE dname= [dname]
```

```
DELETE FROM bosses  
WHERE name = [name]
```