

Bolan Peng
CS 340 Sec 400
Fall 2015

Introduction

This database is about World of Warcraft, an open world, massively multiplayer online role-playing game. Players can control an avatar to explore the landscape, solve quests, fight monsters, and interact with other players and non-player characters. This is a very popular game that is played by people from many different countries, so it's interesting to see who are the most avid players and where are they from.

Database Outline

The game has some characters. Each character is uniquely identified by an username. They have race, class and level.

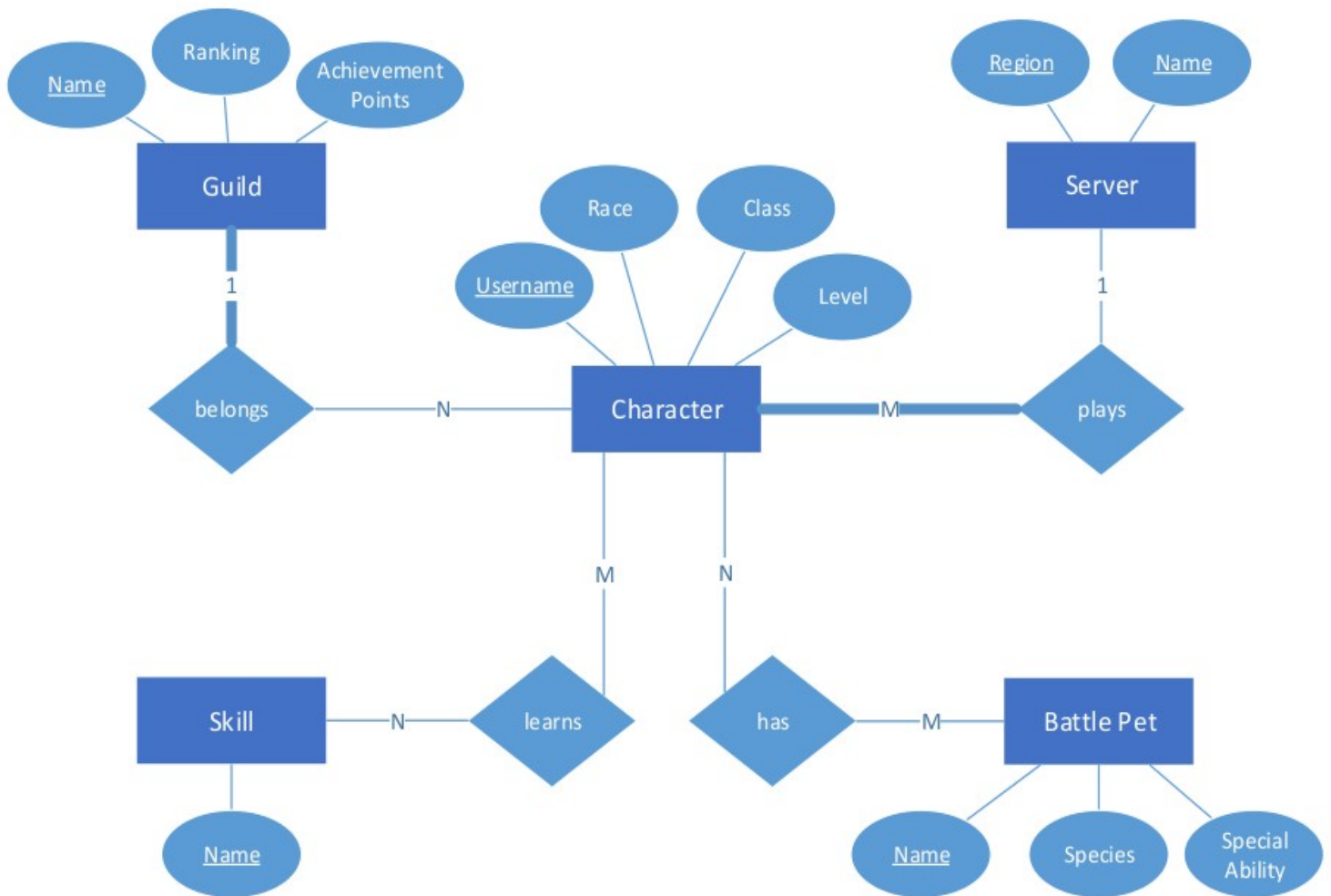
A character can belong to zero or one guild, but a guild has to have at least one character. A guild has name, ranking, achievement points.

A character must be played on one server at a time, but a server can contain zero (most extreme case) or more characters. Each server is uniquely identified by a region and a name.

Characters can learn zero or more skills. Skills may be learned by zero or more characters. Each skill has a name.

Characters can have zero or more battle pets. Battle pets can belong to zero or more characters (Battle pets are like animals, each type of battle pets have more than one in number). Battle pets have name, species and maybe a special ability.

ER Diagram



Schema

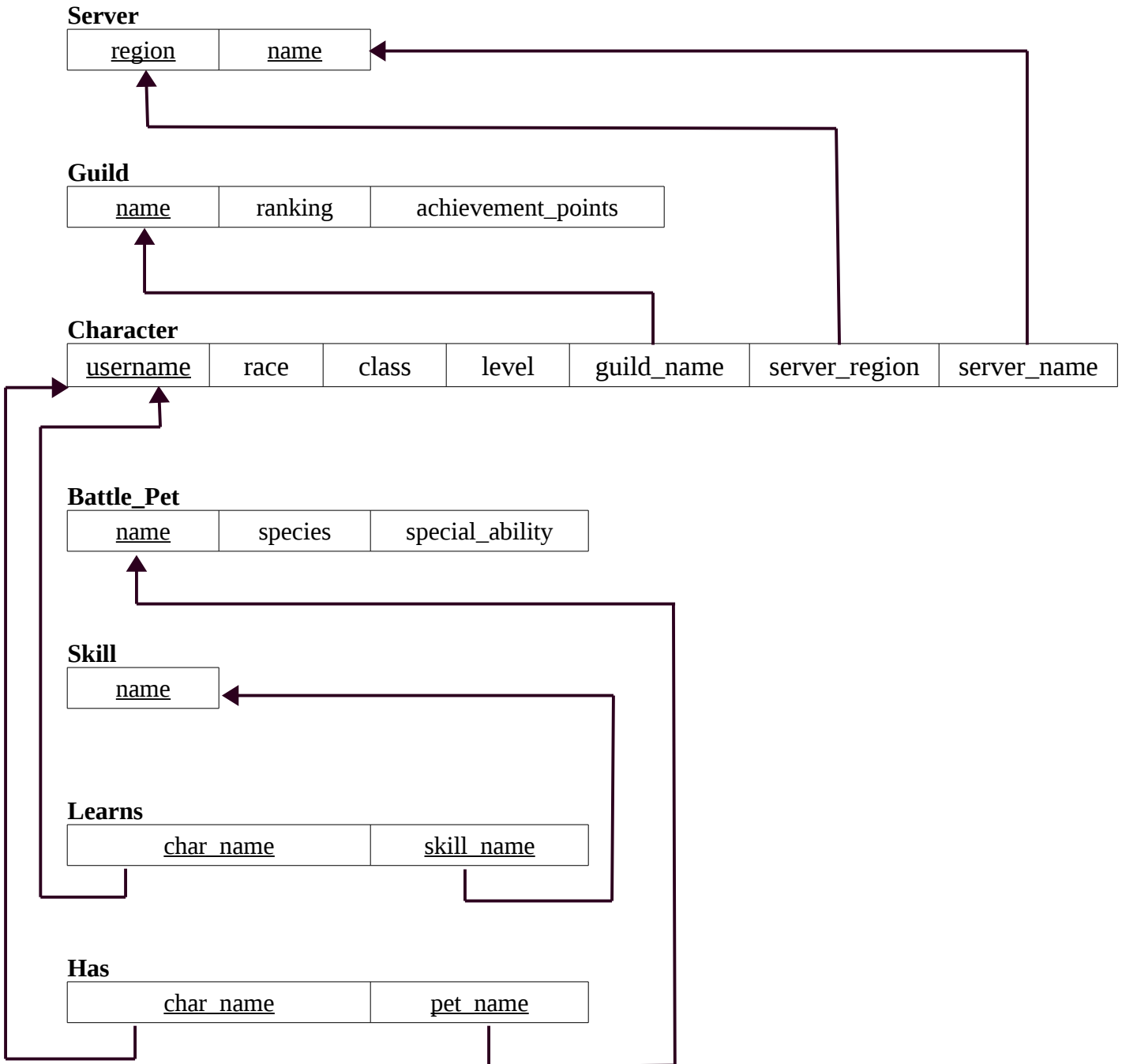


Table Creation Queries

```
CREATE TABLE `server` (  
    `region` varchar(255) NOT NULL,  
    `name` varchar(255) NOT NULL,  
    PRIMARY KEY (`region`, `name`)  
) ENGINE=InnoDB;
```

```
CREATE TABLE `guild` (  
    `name` varchar(255) PRIMARY KEY NOT NULL,  
    `ranking` int NOT NULL,  
    `ach_points` int NOT NULL,  
    UNIQUE (`ranking`)  
) ENGINE=InnoDB;
```

```
CREATE TABLE `character` (  
    `username` varchar(255) PRIMARY KEY NOT NULL,  
    `race` varchar(255) NOT NULL,  
    `class` varchar(255) NOT NULL,  
    `level` int NOT NULL,  
    `guild_name` varchar(255),  
    `server_region` varchar(255) NOT NULL,  
    `server_name` varchar(255) NOT NULL,  
    FOREIGN KEY (`guild_name`) REFERENCES `guild` (`name`) ON DELETE SET  
    NULL ON UPDATE CASCADE,  
    FOREIGN KEY (`server_region`, `server_name`) REFERENCES `server`  
    (`region`, `name`) ON UPDATE CASCADE  
) ENGINE=InnoDB;
```

```
CREATE TABLE `battle_pet` (  
    `name` varchar(255) PRIMARY KEY NOT NULL,  
    `species` varchar(255) NOT NULL,  
    `special_ability` varchar(255)  
) ENGINE=InnoDB;
```

```
CREATE TABLE `skill` (  
    `name` varchar(255) PRIMARY KEY NOT NULL  
) ENGINE=InnoDB;
```

```

CREATE TABLE `learns` (
  `char_name` varchar(255) NOT NULL,
  `skill_name` varchar(255) NOT NULL,
  PRIMARY KEY (`char_name`, `skill_name`),
  FOREIGN KEY (`char_name`) REFERENCES `character` (`username`) ON UPDATE
  CASCADE,
  FOREIGN KEY (`skill_name`) REFERENCES `skill` (`name`)
) ENGINE=InnoDB;

```

```

CREATE TABLE `has` (
  `char_name` varchar(255) NOT NULL,
  `pet_name` varchar(255) NOT NULL,
  PRIMARY KEY (`char_name`, `pet_name`),
  FOREIGN KEY (`char_name`) REFERENCES `character` (`username`) ON UPDATE
  CASCADE,
  FOREIGN KEY (`pet_name`) REFERENCES `battle_pet` (`name`) ON UPDATE
  CASCADE
) ENGINE=InnoDB;

```

General Use Queries

```

INSERT INTO `server` (region, name) VALUES ([region], [name]);

```

```

INSERT INTO `guild` (name, ranking, ach_points) VALUES ([name], [ranking],
[ach_points]);

```

```

INSERT INTO `character` (username, race, class, level, guild_name,
server_region, server_name) VALUES ([username], [race], [class], [level],
[guild_name], [server_region], [server_name]);

```

```

INSERT INTO `battle_pet` (name, species, special_ability) VALUES ([name],
[species], [special_ability]);

```

```

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

```

```

INSERT INTO `learns` (char_name, skill_name) VALUES ([char_name],
[skill_name]);

```

```

INSERT INTO `has` (char_name, pet_name) VALUES ([char_name], [pet_name]);

```

-- Server

-- the server with the highest ranking guild

```
SELECT s.region, s.name FROM `guild` g
      INNER JOIN `character` c ON c.guild_name=g.name
      INNER JOIN `server` s ON s.name=c.server_name
      GROUP BY g.name
      ORDER BY g.ranking ASC
      LIMIT 1;
```

-- Guild

-- List the name of the players that belong to the number one guild.

```
SELECT c.username FROM `guild` g
      INNER JOIN `character` c ON c.guild_name=g.name
      WHERE g.ranking=1;
```

-- Find all the guilds that have players that are not max level (level < 100).

```
SELECT g.name FROM `guild` g
      INNER JOIN `character` c ON c.guild_name=g.name
      WHERE c.level < [maxLevel]
      ORDER BY g.name ASC;
```

-- Character

-- Find the most popular race.

```
SELECT c.race, COUNT(c.race) AS num
      FROM `character` c
      GROUP BY c.race
      ORDER BY num DESC
      LIMIT 1;
```

-- Find the least popular race.

```
SELECT c.race, COUNT(c.race) AS num
      FROM `character` c
      GROUP BY c.race
      ORDER BY num ASC
      LIMIT 1;
```

-- Find the most popular class.

```
SELECT c.class, COUNT(c.class) AS num
      FROM `character` c
```

```
GROUP BY c.class
ORDER BY num DESC
LIMIT 1;
```

-- Find the least popular class.

```
SELECT c.class, COUNT(c.class) AS num
FROM `character` c
GROUP BY c.class
ORDER BY num ASC
LIMIT 1;
```

-- Battle Pet --

-- how many of each battle pet is owned by characters

```
SELECT bp.name, COUNT(has.pet_name) AS num FROM `battle_pet` bp
LEFT JOIN `has` ON has.pet_name=bp.name
GROUP BY bp.name;
```

-- rank battle pets according to their popularity, in descending order

```
SELECT bp.name, COUNT(has.pet_name) AS num FROM `battle_pet` bp
LEFT JOIN `has` ON has.pet_name=bp.name
GROUP BY bp.name
ORDER BY num DESC;
```

-- find the least popular battle pet

```
SELECT bp.name, COUNT(has.pet_name) AS num FROM `battle_pet` bp
LEFT JOIN `has` ON has.pet_name=bp.name
GROUP BY bp.name
ORDER BY num ASC
LIMIT 1;
```

-- the name of players that don't have any battle pets

```
SELECT c.username FROM `character` c
WHERE c.username NOT IN
(SELECT has.char_name FROM `battle_pet` bp
INNER JOIN `has` ON has.pet_name=bp.name)
ORDER BY c.username ASC;
```

-- Skills -

-- most popular skill

```
SELECT skill.name, COUNT(learns.skill_name) as num FROM `skill`
```

```
INNER JOIN `learns` ON learns.skill_name=skill.name
GROUP BY skill.name
ORDER BY num DESC
LIMIT 1;
```

-- least popular skill

```
SELECT skill.name, COUNT(learns.skill_name) as num FROM `skill`
INNER JOIN `learns` ON learns.skill_name=skill.name
GROUP BY skill.name
ORDER BY num ASC
LIMIT 1;
```

-- name of the players that learned all the skills

```
SELECT c.username, COUNT(learns.skill_name) as num FROM `skill`
INNER JOIN `learns` ON learns.skill_name=skill.name
INNER JOIN `character` c ON c.username=learns.char_name
GROUP BY c.username
HAVING num=(SELECT COUNT(skill.name) as num FROM `skill`);
```

-- name of players that didn't learn any skills

```
SELECT c.username, COUNT(learns.skill_name) as num FROM `skill`
INNER JOIN `learns` ON learns.skill_name=skill.name
INNER JOIN `character` c ON c.username=learns.char_name
GROUP BY c.username
HAVING num=0;
```