Erin Alltop
CS340 – Spring 2017
World of Warcraft Database – Final Project

Outline

The database created here is a database of some aspects of the popular game World of Warcraft (WoW). WoW is a mass-multiplayer roleplaying game that consists of many ways to play. One of the more popular ways for people to play together in this game is to form a player group called a guild which is a mini-database of characters who are affiliated with each other under their guild name and can help each other, form smaller groups to do quests, or form larger groups to take down difficult "bosses" in raid encounters.

This database specifically will represent a database of a player guild. It is one small section of a tremendous world, 10 years in the making. While WoW itself has almost limitless possibilities for databases, this database will most benefit a Guild Leader to keep track of their players and their characters. A guild leader using this database will be able to tell what level their guild members are, what professions they have (important for guild crafting!), what race and class they are, and what raid achievements they have under their belt. If the guild does regular "raiding" (taking down bosses as a team), then it will be important to know what experience a character has before raid night.

In essence, this database will give an overview of characters in a specific guild – "Ancient Knowledge" - and their achievements. It also allows a raid leader to sort through the database by guild rank, add and remove characters from the guild, and add professions, raids, and continents and the game progresses with new expansions. In a world as large as that of Warcraft, it is a handy tool to keep track of everyone in the guild and their abilities.

Database Outline

This database has six entities, each with their own attributes:

1. Characters
    - ID (primary key), Name (unique), Level, Race, Class, main_spec, Item Level, Guild Rank, continent, and profession
2. Professions
    - ID (primary key), Name
3. Continents
    - ID (primary key), Name
4. Raids
    - ID (primary key), Name
5. Specs
    - ID (primary key), Name

Entity Tables

Characters are the main entity in this database. Each character row in the Characters table has an auto-incrementing ID that is the primary key; This is required and cannot be null. Their name, level, guild rank, class, and race. Name in this table also has a unique constraint. The main_spec, continent, and profession foreign keys are not required and are NULL by default. Note that iLevel is an aggregate number of the quality of a character's gear and can be useful in determining readiness for a certain raid.

Professions take time for a player character to cultivate, but are invaluable in a guild. They are not required, so they are not a required attribute in the character table and are their own entity as there are many professions. This table has an auto-incrementing primary key and required name. Since characters in WoW can have multiple professions, and many players can have the same professions, this entity has a many-to-many relationship with the character table.

The Continents table is also an entity and it has a one-to-many relationship with characters. Characters can only be from one continent, but continents of course can have many characters originate from them. Continents have a required name and an auto-incrementing primary key.

One of the most important aspects of this database (for a raid leader) is the Raids table. Raids are collection of difficult bosses that the most seasoned characters can attempt to defeat in a team put together by a Raid Leader. Having skill in taking down some of the most notorious raid bosses tells the Raid Leader that you are worthy to take on his team. Characters can have many "raid achievements" (i.e. raid completions / boss kills), but this is not required. Similarly, raids can be completed by many different characters. Thus, there is a many-to-many relationship between these two entities as well. The Raids table has an auto-incrementing primary key and a required name.

In addition to the raids table is the "completed" table for the relationship between characters and their raid achievements. Each raid has its own difficulty rating – Normal, Heroic, and Mythic. Normal is the "easiest" (but not easy!), and Mythic is the most challenging raid. This can tell a raid leader how seasoned a character is in raiding.

Lastly, we have an entity "Specs" table. This table is a list of the possible specs (or character specializations) in the game. Like most MMOs, a player can be a healer, a DPS (damage dealer), or a tank (takes the most beating!). This entity has a auto-incrementing primary key and a required name. This database does not have an option to add to this table as it is an extra table for use with other table and there are not any other specializations able to be added to it.

Relationship Tables

This database also consists of four relationship tables between the main entities to connect our data and give it more useful features.

1. **`main_spec`** – Describes the main spec of each character. This table has two columns, the `char_id` and the `spec_id` columns. `char_id` references the character ID table and `spec_id` references the Specs table. Neither can be null and both are primary keys. In addition, the `char_id` must be unique in the table. This relationship represents a one-to-many relationship. Many characters can have the same spec but an individual character can only have one spec.

2. **`is_from`** – Describes what continent a character originates from. This table also has two columns, the `char_id` and the `cont_id`. ` char_id` references the characters table and `cont_id` references the continents table. Neither can be null and both are primary keys. Same as the main_spec table, the `char_id` must be unique in this table. Many characters can be from the same continent, but a character can only be from one continent. There is a one-to-many relationship between this table and the characters table.
3. **`knows`** – Describes what professions a character has (if any). This table has two columns, the `char_id` column which is required and cannot be null (and is the primary key), and the `prof_id` which is NOT required and CAN be null. This table has a many-to-many relationship with the professions table. Many characters can know the same professions and characters can know many professions.
4. **`completed`** – Describes what raids a character has completed (if any) and their difficulty. This table has three columns, the `char_id` which is required and cannot be null (and is the primary key), raid_id` (required, cannot be null, also a primary key), and `difficulty` (required, cannot be null). A character is not required to have completed any raids, but if he or she has, then all fields must have a value. In addition, this table has a unique key combination `raid_unique`. The combination of the char_id, raid_id, and difficulty can only occur once in the table, though each can exist multiple times in different rows. This table has a many-to-many relationship with the raids table. Characters can have completed many raid achievements and many characters can have completed the same raid achievements.

Database Creation Queries

```
DROP TABLE IF EXISTS `continents`;
DROP TABLE IF EXISTS `professions`;
DROP TABLE IF EXISTS `raids`;
DROP TABLE IF EXISTS `characters`;
DROP TABLE IF EXISTS `specs`;

DROP TABLE IF EXISTS `main_spec`;
DROP TABLE IF EXISTS `completed`;
DROP TABLE IF EXISTS `is_from`;
DROP TABLE IF EXISTS `knows`;

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

CREATE TABLE `professions` (
        `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,
	`name` varchar(255) NOT NULL UNIQUE,
	`level` int NOT NULL,
	`race` varchar(255) NOT NULL,
	`class` varchar(255) NOT NULL,
	 `main_spec` int DEFAULT NULL,
	`iLevel` int DEFAULT NULL,
	`guild_rank` varchar(255) DEFAULT NULL,
	`continent` int DEFAULT NULL,
	`profession` int DEFAULT NULL,
	PRIMARY KEY(`id`),
	FOREIGN KEY(`continent`) REFERENCES continents(`id`),
	FOREIGN KEY(`main_spec`) REFERENCES specs(`id`),
	FOREIGN KEY(`profession`) REFERENCES professions(`id`)
)ENGINE = InnoDB;

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

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

CREATE TABLE `completed` (
	`char_id` int NOT NULL,
	`raid_id` int NOT NULL,
	`difficulty` varchar(255),
	PRIMARY KEY(`char_id`, `raid_id`, `difficulty`),
	FOREIGN KEY(`char_id`) REFERENCES characters(`id`) ON DELETE CASCADE,
	FOREIGN KEY(`raid_id`) REFERENCES raids(`id`) ON DELETE CASCADE,
)ENGINE = InnoDB;



CREATE TABLE `knows` (
	`char_id` int NOT NULL,
	`profession_id` int NOT NULL,
	PRIMARY KEY(`char_id`, `profession_id`),
        FOREIGN KEY(`char_id`) REFERENCES characters(`id`) ON DELETE CASCADE,
        FOREIGN KEY(`profession_id`) REFERENCES professions(`id`) ON DELETE CASCADE
)ENGINE = InnoDB;

CREATE TABLE `is_from` (
        `char_id` int NOT NULL UNIQUE,
        `continent_id` int NOT NULL,
        PRIMARY KEY(`char_id`, `continent_id`),
        FOREIGN KEY(`char_id`) REFERENCES characters(`id`) ON DELETE CASCADE,
        FOREIGN KEY(`continent_id`) REFERENCES continents(`id`) ON DELETE CASCADE
)ENGINE = InnoDB;

CREATE TABLE `main_spec` (
        `char_id` int NOT NULL UNIQUE,
        `spec_id` int NOT NULL,
        PRIMARY KEY(`char_id`, `spec_id`),
        FOREIGN KEY(`char_id`) REFERENCES characters(`id`) ON DELETE CASCADE,
        FOREIGN KEY(`spec_id`) REFERENCES specs(`id`) ON DELETE CASCADE
)ENGINE = InnoDB;
```

Database Manipulation Queries

Add a new character to the `characters` table:

```
INSERT INTO `characters` (name, level, race, class, iLevel, guild_rank) VALUES
([nameInput], [levelInput], [raceInput], [classInput], [iLevelInput], [guild_rankInput]);
```

Delete a character from the `characters` table:

```
DELETE FROM `characters` WHERE id = [charID];
```

Add a new continent to the `continents` table:

```
INSERT INTO `continents` (name) VALUES [continentName];
```

Add a new raid to the `raids` table:

```
INSERT INTO `raids` (name) VALUES [raidName];
```

Add a new profession to the `professions` table:

```
INSERT INTO `professions` (name) VALUES [professionName];
```

Insert a character and continent into the `is_from` relationship table:

```
INSERT INTO `is_from` (char_id, continent_id) VALUES ([charInput], [continentInput])
ON DUPLICATE KEY UPDATE continent_id = VALUES(continent_id);
```

Insert a character and profession into the `knows` relationship table:

```
INSERT INTO `knows` (char_id, profession_id) VALUES ([charInput], [continentInput]);
```

Insert a character, raid, and difficulty into the `completed` relationship table:

```
INSERT INTO `completed` (char_id, raid_id, difficulty) VALUES ([charInput], [raidInput],
[difficultyInput]);
```

Insert a character and main spec into the `main_spec` relationship table:

```
INSERT INTO `main_spec` (char_id, spec_id) VALUES ([charInput], [specInput])
ON DUPLICATE KEY UPDATE spec_id = VALUES(spec_id);
```

Display the filtered `characters` table to the user:

```
SELECT characters.name, characters.level, characters.race, characters.class, characters.iLevel,
characters.guild_rank FROM `characters` WHERE characters.guild_rank = [guild_rankInput];
```

Display the `completed` table to the user:

```
SELECT characters.name, raids,name, difficulty FROM `completed`
INNER JOIN `characters` ON char_id = characters.id
INNER JOIN `raids` ON raid_id = raids.id
ORDER BY characters.name;
```

Display the `is_from` table to the user:

```
SELECT characters.name, continents.name FROM `is_from`
INNER JOIN `characters` ON char_id = characters.id
INNER JOIN `continents` ON continent_id = continents.id
ORDER BY characters.name;
```

Display the `knows` table to the user:

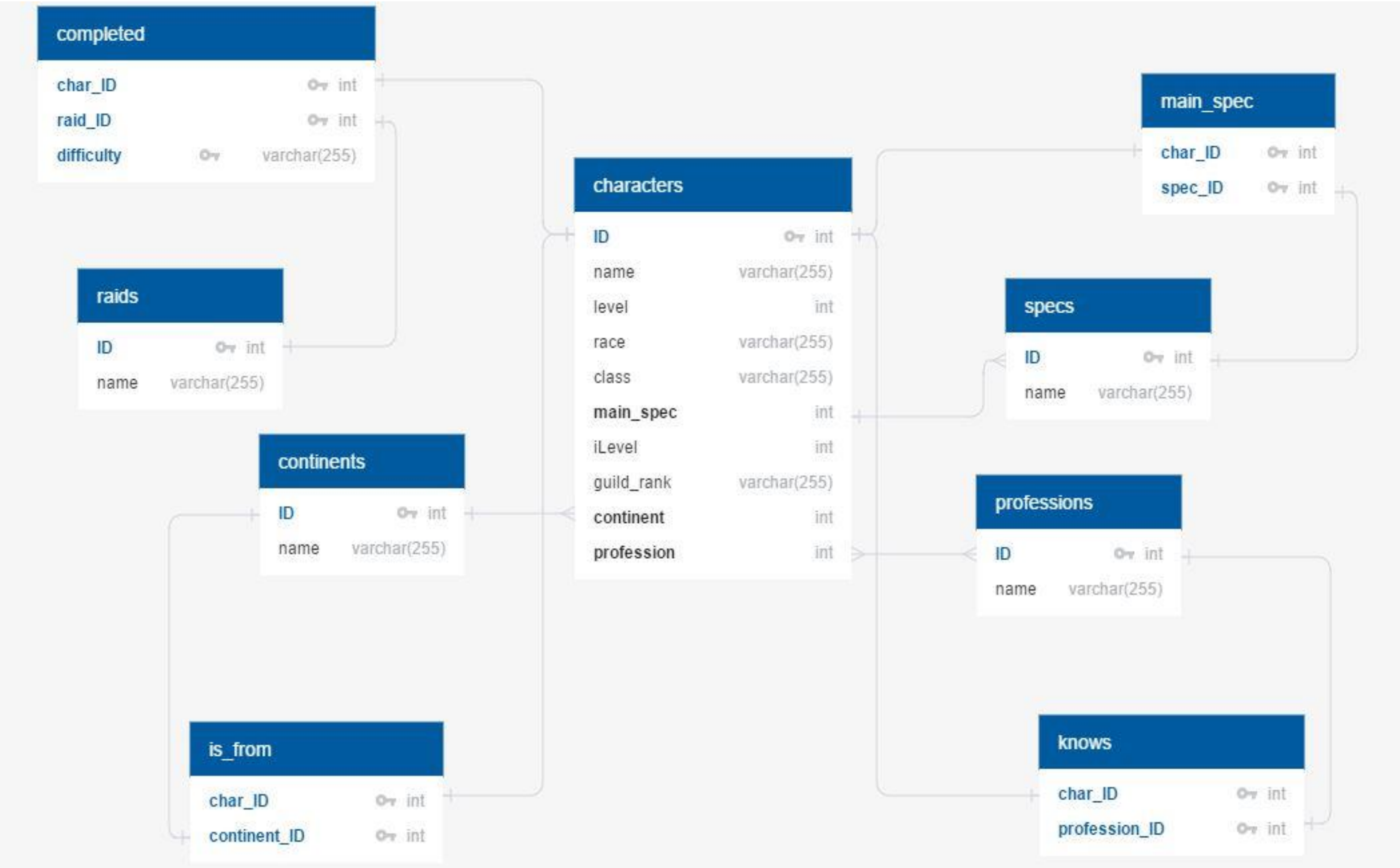
SELECT characters.name, professions.name FROM `knows`
INNER JOIN `characters` ON char_id = characters.id
INNER JOIN `professions` ON profession_id = professions.id
ORDER BY characters.name;

Display the `main_spec` table to the user:

SELECT characters.name, specs.name FROM `main_spec`
INNER JOIN `characters` ON char_id = characters.id
INNER JOIN `specs` ON spec_id = specs.id
ORDER BY specs.name;

Database Schema

Note: bolded items are foreign key references, light blue items are primary keys.



completed		
char_ID	🔑	int
raid_ID	🔑	int
difficulty	🔑	varchar(255)

main_spec		
char_ID	🔑	int
spec_ID	🔑	int

characters		
ID	🔑	int
name		varchar(255)
level		int
race		varchar(255)
class		varchar(255)
main_spec		int
iLevel		int
guild_rank		varchar(255)
continent		int
profession		int

raids		
ID	🔑	int
name		varchar(255)

specs		
ID	🔑	int
name		varchar(255)

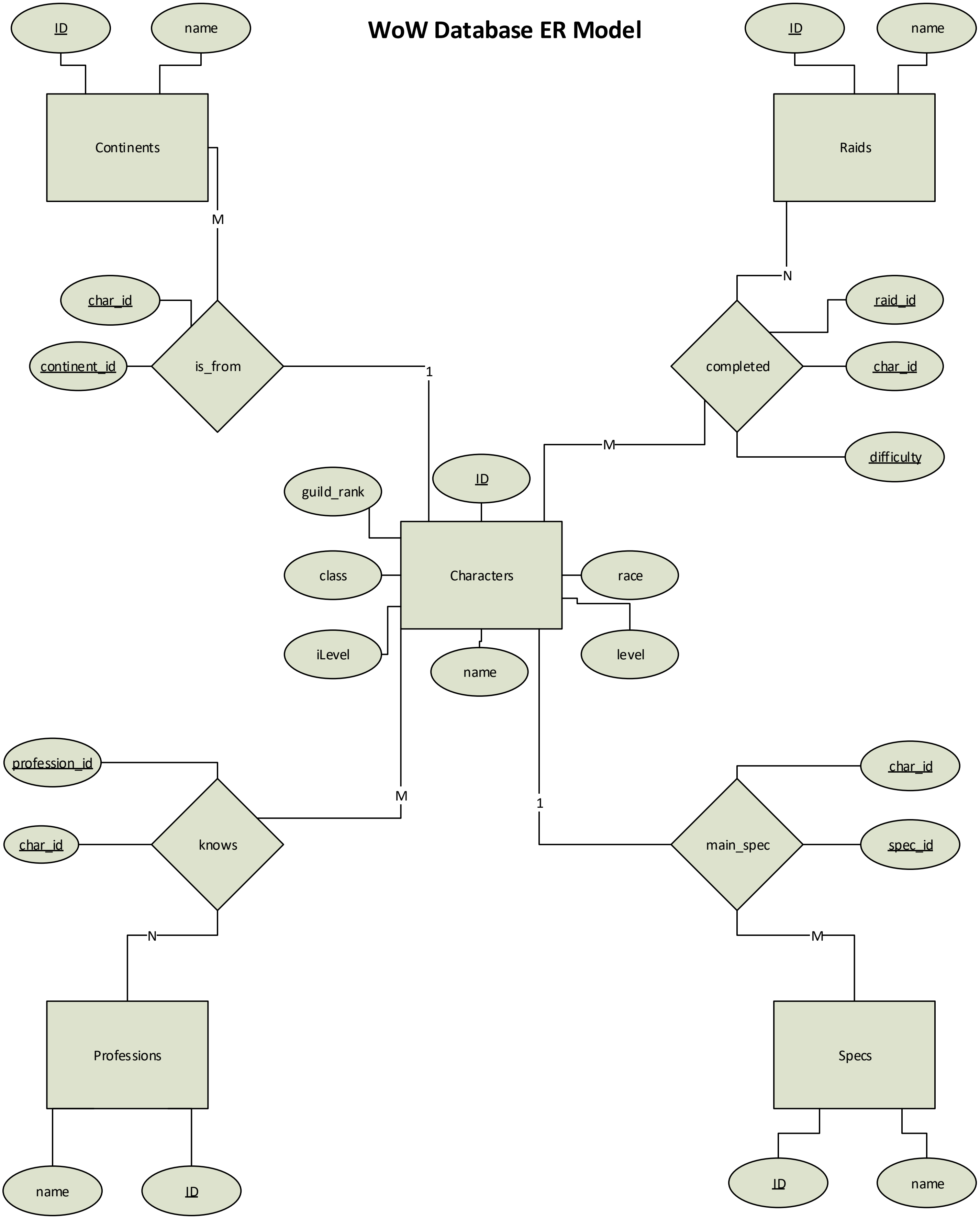
continents		
ID	🔑	int
name		varchar(255)

professions		
ID	🔑	int
name		varchar(255)

is_from		
char_ID	🔑	int
continent_ID	🔑	int

knows		
char_ID	🔑	int
profession_ID	🔑	int

WoW Database ER Model


ID
name
Continents
ID
name
Raids
M
N
char_id
continent_id
is_from
1
raid_id
completed
char_id
M
difficulty
guild_rank
ID
class
Characters
race
iLevel
name
level
profession_id
M
1
char_id
char_id
knows
main_spec
spec_id
N
M
Professions
Specs
name
ID
ID
name