

**Name:** Ya Lien

**Class:** CS340

### **Final Project**

**Website:** <http://web.engr.oregonstate.edu/~lieny/CS340/champion.php>

#### **Outline**

The database of this assignment will represent the League of Legends Universe. League of Legends is a multiplayer online battle arena game designed by the Riot Games Inc. League of Legends has a massive universe with uniquely designed champions. This database will mainly focus on the champions, including the relationship between the champions and the regions they came from, the main roles they play, the damage type they have, and the essential items that they use.

#### **Database Outline**

The database have 5 entities: Champion, Region, MainRole, Damage, and StarterItem. The Champion entity holds rows of characters who exist in League of Legends. Each champion has an unique championID, which is the primary key for this table, and other attributes that holds the champion's base information: name, baseAD for their base attack damage, baseArmor for their base armor, baseHP for base health, baseMP for base mana; and 2 foreign keys: fk\_regionID that references Region.regionID, and fk\_damageID that references to Damage.damageID.

The Region entity is the homeland where the Champions came from. It has an ID as the primary key, and a name row for the name of the Region. One champion can only come from exactly 1 region, and one region may be the homeland for many champions. This is a 1 to many relationship, so we can go to the table for entity 'Champion' and add fk\_regionID as an attribute.

The Role entity represents the types that the champions play in the game, there are 5 roles: Marksman, Support, Mage, Fighter, and Tank. This entity has an ID as its primary key, and a type column that should not be null, as every champion has at least 1 role. Each role can be played by many champions. This is a many to many relationship and will have its own relationship table.

The Damage entity represents the type of damage the champions mainly dealt in the game. This entity has an ID as its primary key, and a type that is not null. Each champion must have exactly 1 damage type, and each type of damage can be dealt by many champions. This is a 1 to many relationship and will not have a separate table. We will add fk\_damageID as an attribute to the 'Champion' table.

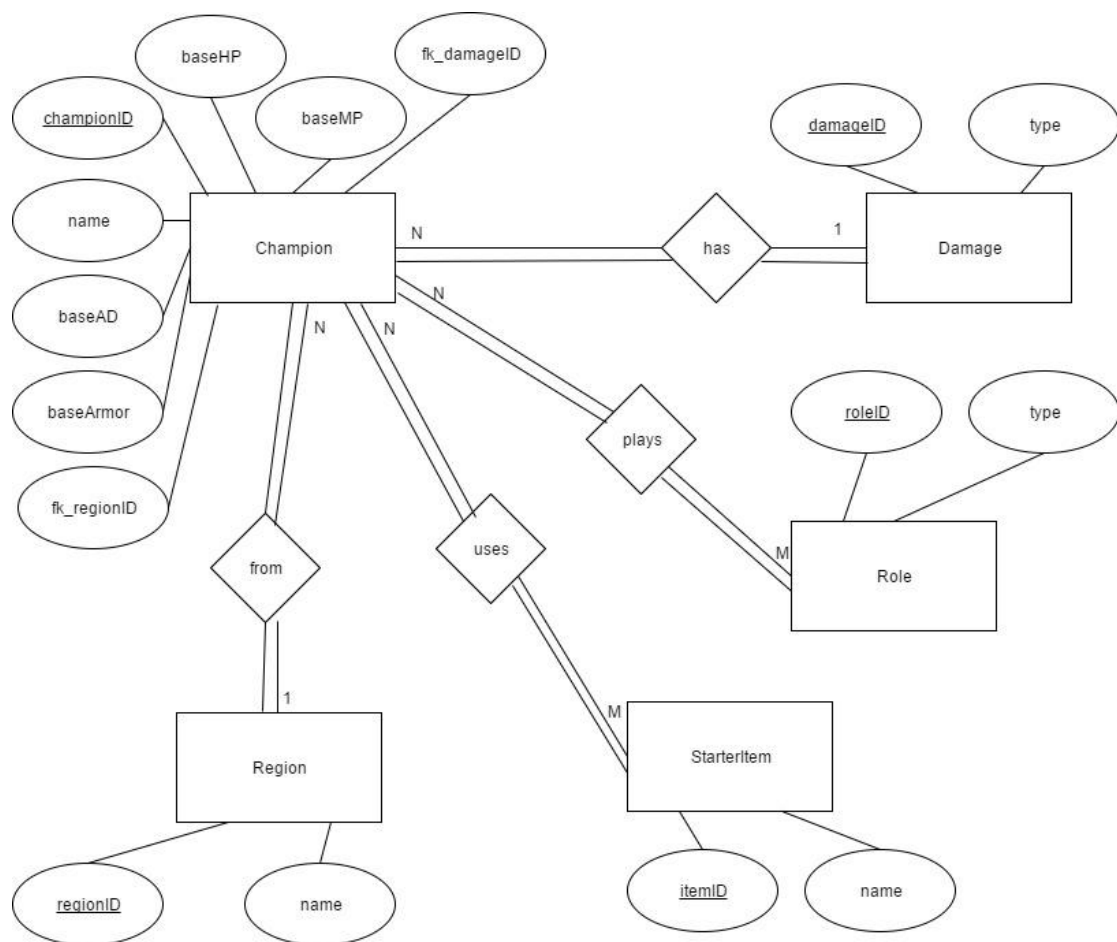
In the game, the champions can buy items as enhancements. The StarterItem entity holds the items obtainable with the starting gold, depends on each champion's

role. This entity has an ID as its primary key, and a name for each starter item. Each champion can use more than 1 starter items and each starter item can be used by many champions. This is a many to many relationship and will have a separate relationship table.

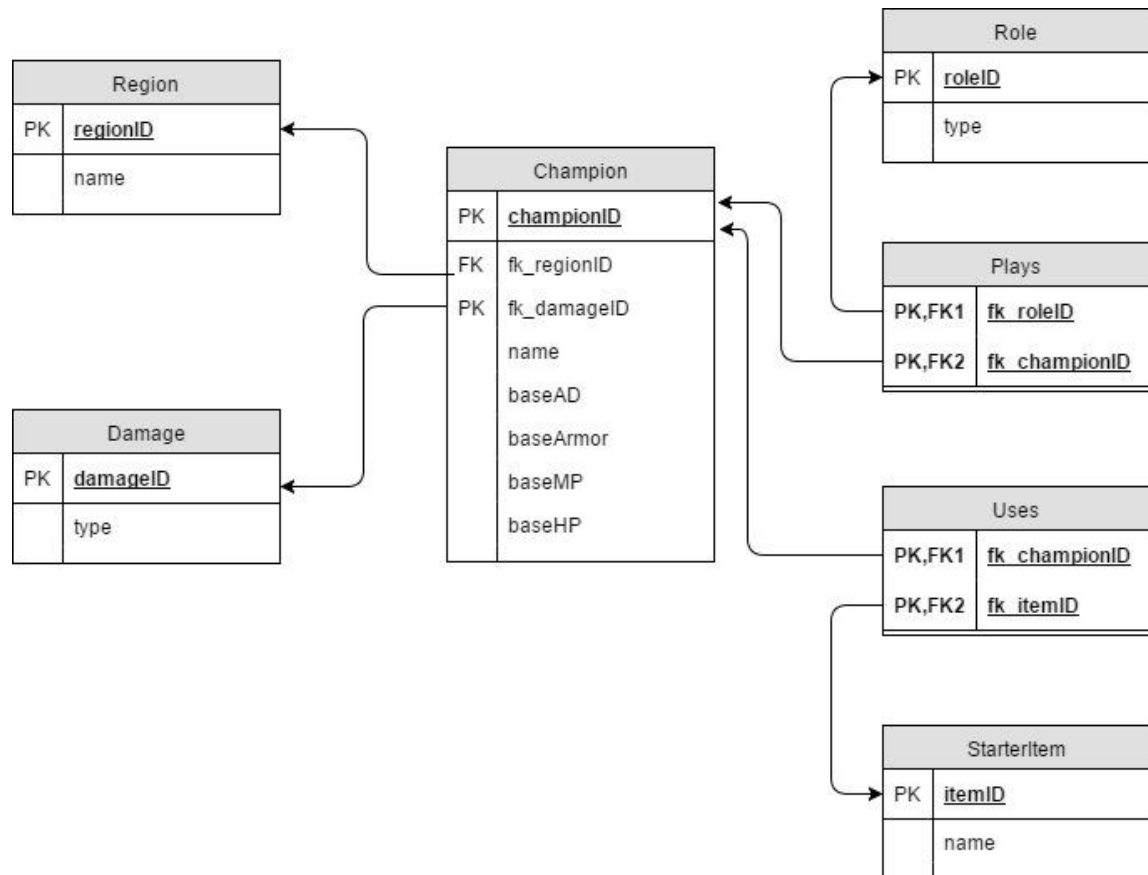
As mentioned above, there will be two relationship tables in the database: 'plays' and 'uses'. The 'plays' table describe the Roles played by the Champions. This table has two attributes, 'fk\_championID' for the champion ID and 'fk\_roleID' for the role ID. Both attributes make the primary key of table 'plays' and both attributes cannot be null.

'Uses' is a many to many relationship, so we create a separate table for that. This table has the primary keys of the entities that make up the relationship. Both attributes, 'fk\_championID' and 'fk\_itemID', make the primary key of the 'uses' table, and both should not be null.

## ER Diagram



## Schema



### Data Definition Queries

```
DROP TABLE IF EXISTS `Champion`;
```

```
DROP TABLE IF EXISTS `Region`;
```

```
DROP TABLE IF EXISTS `Damage`;
```

```
DROP TABLE IF EXISTS `Role`;
```

```
DROP TABLE IF EXISTS `Plays`;
```

```
DROP TABLE IF EXISTS `Uses`;
```

```
DROP TABLE IF EXISTS `StartItem`;
```

-- Create the Champion table with following properties:

-- championID - an auto incrementing integer which is the primary key

-- name - a varchar of maximum length 255, cannot be null

-- baseAD - an integer

-- baseArmor - an integer

-- baseHP - an integer

-- baseMP - an integer

-- fk\_regionID - an integer which is a foreign key reference to Region.regionID

-- fk\_damageID - an integer which is a foreign key reference to Damage.damageID

-- the name of the project should be unique in this table

```
CREATE TABLE `Champion`(  
  `championID` int(11) NOT NULL AUTO_INCREMENT,  
  `name` varchar(255) NOT NULL,  
  `baseAD` int(11),  
  `baseArmor` int(11),  
  `baseHP` int(11),  
  `baseMP` int(11),  
  `fk_regionID` int(11),  
  `fk_damageID` int(11),  
  PRIMARY KEY (`championID`),  
  FOREIGN KEY (`fk_regionID`) REFERENCES `Region` (`regionID`),  
  FOREIGN KEY (`fk_damageID`) REFERENCES `Damage` (`damageID`),  
  UNIQUE KEY (`name`)  
) ENGINE=InnoDB;
```

-- Create the Region Table with following properties:  
-- regionID- an auto incrementing integer which is the primary key  
-- name - a varchar of maximum length 255, cannot be null  
-- the name of the Region should be unique in this table

```
CREATE TABLE `Region`(  
  `regionID` int(11) NOT NULL AUTO_INCREMENT,  
  `name` varchar(255) NOT NULL,  
  PRIMARY KEY (`regionID`),  
  UNIQUE KEY (`name`)  
) ENGINE=InnoDB;
```

-- Create the Damage Table with following properties:  
-- damageID- an auto incrementing integer which is the primary key  
-- type - a varchar of maximum length 255, cannot be null  
-- the type of the Damage should be unique in this table

```
CREATE TABLE `Damage`(  
  `damageID` int(11) NOT NULL AUTO_INCREMENT,  
  `type` varchar(255) NOT NULL,  
  PRIMARY KEY (`damageID`),  
  UNIQUE KEY (`type`)  
) ENGINE=InnoDB;
```

-- Create the Role Table with following properties:  
-- roleID- an auto incrementing integer which is the primary key  
-- type - a varchar of maximum length 255, cannot be null  
-- the type of the Role should be unique in this table

```
CREATE TABLE `Role`(  
  `roleID` int(11) NOT NULL AUTO_INCREMENT,  
  `type` varchar(255) NOT NULL,  
  PRIMARY KEY (`roleID`),  
  UNIQUE KEY (`type`)  
) ENGINE=InnoDB;
```

-- Create the StarterItem Table with following properties:  
-- itemID- an auto incrementing integer which is the primary key  
-- name - a varchar of maximum length 255, cannot be null  
-- the name of the StarterItem should be unique in this table

```
CREATE TABLE `StarterItem`(  
  `itemID` int(11) NOT NULL AUTO_INCREMENT,  
  `name` varchar(255) NOT NULL,  
  PRIMARY KEY (`itemID`),  
  UNIQUE KEY (`name`)  
) ENGINE=InnoDB;
```

```
`itemID` int(11) NOT NULL AUTO_INCREMENT,  
`name` varchar(255) NOT NULL,  
PRIMARY KEY (`itemID`),  
UNIQUE KEY (`name`)  
) ENGINE=InnoDB;
```

-- ----- Relationship Tables -----

-- Create the Plays table with following properties:

-- fk\_roleID - an integer which is a foreign key reference to Role.roleID

-- fk\_championID - an integer which is a foreign key reference to  
Champion.championID

-- both properties are primary keys for this table

```
CREATE TABLE `Plays`(  
`fk_roleID` int(11) NOT NULL,  
`fk_championID` int(11) NOT NULL,  
PRIMARY KEY (`fk_roleID`, `fk_championID`),  
FOREIGN KEY (`fk_roleID`) REFERENCES `Role` (`roleID`) ON DELETE CASCADE,  
FOREIGN KEY (`fk_championID`) REFERENCES `Champion` (`championID`) ON DELETE  
CASCADE  
) ENGINE=InnoDB;
```

-- Create the Uses table with following properties:

-- fk\_itemID - an integer which is a foreign key reference to StarterItem.itemID

-- fk\_championID - an integer which is a foreign key reference to  
Champion.championID

-- both properties are primary keys for this table

```
CREATE TABLE `Uses`(  
`fk_itemID` int(11) NOT NULL,  
`fk_championID` int(11) NOT NULL,  
PRIMARY KEY (`fk_itemID`, `fk_championID`),  
FOREIGN KEY (`fk_itemID`) REFERENCES `StarterItem` (`itemID`) ON DELETE  
CASCADE,  
FOREIGN KEY (`fk_championID`) REFERENCES `Champion` (`championID`) ON DELETE  
CASCADE  
) ENGINE=InnoDB;
```

## **Data Manipulation Queries**

### Display All Champion Information

```
SELECT Champion.name, Champion.baseAD, Champion.baseArmor,  
Champion.baseMP, Champion.baseHP, Region.name, Damage.type FROM Champion  
    INNER JOIN Region ON Region.regionID = Champion.fk_regionID  
    INNER JOIN Damage ON Damage.damageID = Champion.fk_damageID  
GROUP BY Champion.name
```

### Add Champion

```
INSERT INTO Champion (name, baseAD, baseArmor, baseMP, baseHP, fk_regionID,  
    fk_damageID) VALUES ([name_input], [attack_damage_input],  
    [armor_input], [mana_input], [health_input], [region_input],  
    [damage_input]);
```

### Add Region

```
INSERT INTO Region(name) VALUES ([region_input]);
```

### Add Damage

```
INSERT INTO Damage(type) VALUES ([damage_input]);
```

### Add Role Type

```
INSERT INTO Role(type) VALUES ([role_input]);
```

### Add Starter Item

```
INSERT INTO StarterItem(name) VALUES ([item_input]);
```

### Add to the Plays Relationship

```
INSERT INTO Plays(fk_championID, fk_roleID) VALUES ([champion_input],  
    [role_input]);
```

### Add to the Uses Relationship

```
INSERT INTO Uses(fk_championID, fk_itemID) VALUES ([champion_input],  
    [item_input]);
```

### Delete from Uses - Delete from a many to many relationship

```
DELETE FROM Uses WHERE fk_championID = [champion_input] AND fk_itemID =  
    [item_input];
```

#### Delete a Champion

```
DELETE FROM Champion WHERE championID = [champion_input];
```

#### Filter Champions by Region

```
SELECT Champion.name, Region.name FROM Champion  
    INNER JOIN Region ON Region.regionID = Champion.fk_regionID  
    WHERE Region.regionID = [region_input];
```

#### Update Champion Information

```
UPDATE Champion SET  
    baseAD = [baseAD_input], baseArmor = [armor_input], baseMP =  
    [mana_input], baseHP = [health_input], fk_regionID = [region_input],  
    fk_damageID = [damage_input];
```