Final Project Phase 1

Justin Daschner and Seth Vickers

League of Legends is a MOBA that has risen in popularity over the years. Having RPG-style character building over the course of a 45 minute game, it's not surprising that many unique a item and champion combinations have been seen. That being said, the meta (most popular builds) tend to be set by the pros, who play on a level that most players will never reach. If this is known, why not take a more mathematical approach to the game? What is the most efficient use of your gold? As Talon, what is the maximum damage you can deal in a single hit? I already have two long swords, what item can I build for the most armor penetration?

Every champion has a subset of the following statistics: Attack Damage, Ability Damage, Temporary Health, Health, Mana, Energy, Rage, Magic Resist, Health Regen, Mana Regen, Armor, Attack Speed, Critical Strike Chance, Critical Strike Damage, Magic Penetration, Armor Penetration, Movement Speed, and Cooldown Reduction. Each of them are affected by champion level. These stats determine the damage/healing/shielding/etc. of a champion’s abilities, of which every champion has four and a passive. Each champion also has two Summoner Spells that are selected before the game starts that, for the purpose of this explanation, are essentially two bonus abilities chosen from a pool of ten.

With some exceptions, a champion’s 4 primary abilities are actives while the passive is, as the name implies, passive. Each ability will either create some in-game effect or modify a champion stat (buff or debuff). Some abilities affect an area, the are called AoE abilities (area of effect). Others are Crowd Control (CC) abilities, i.e., abilities that reduce movement or impair a champion’s ability to attack.

A champion can also have up to six items an one active potion at any given time. Items can build into better items using various build paths, or recipes, by combining them with other items and spending gold. Each tier of items gives additional statistics and abilities. Most items simply give stat increases, however some have additional abilities. These can be passive or active. All item actives are unique, meaning that if you build the same item twice and use the active on one, both item actives go on cooldown. Some item passives are unique, however others will stack like normal item effects.

Before a game starts, every champion has a set of masteries to choose from. There are five mastery trees. Each champion selects a four primary masteries from one tree and two secondary masteries from another. These correlate to passive abilities.

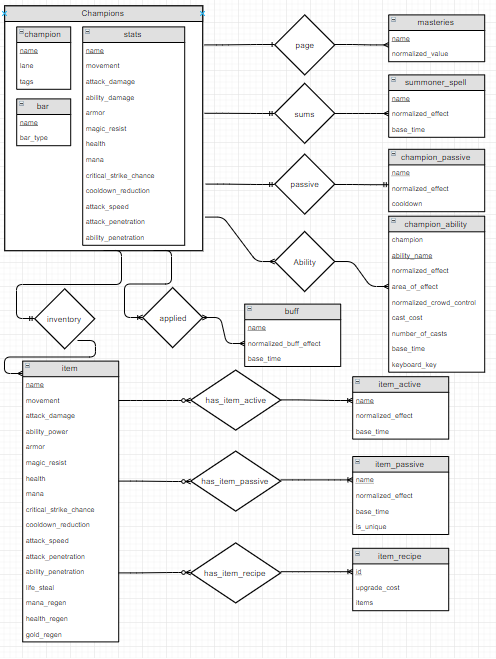
Every ability of every type has a name, icon, range, and cooldown.

This database will provide a way to determine, mathematically, how to build a champion based on the stats you want to maximize as well as efficient uses of gold.

**Example Queries:**

1. What six items give the highest flat armor?
2. What six items give the highest stats with passives and abilities?
3. Which item has the highest Attack Damage?
4. What is the total effect of masteries?
5. What is the weakest based off total effect?
6. What is the total cost?
7. What champion and item combination gives you the highest possible health?
8. What is the highest damage that can be dealt in a single hit?
9. How many times can Anivia use Crystalize in a minute?
10. What ability that doesn't use mana has the highest flat damage output?
11. What item has the shortest cooldown?
12. What abilities have CC (crowd control)?
13. What champions have no AP (ability power)?
14. Can I buy the selected items with the gold?
15. What is the total normalized score of a champion with the various selected attributes?
16. What is the Mean for Health of champions?
17. How many items give attack damage bonus?
18. How many items have passives or actives?
19. What is the most expensive build to have on a champion?
20. How many items are currently usable in Summoner's Rift?

ER Diagram:



**Relational Diagram:**

|  |  |  |  |
| --- | --- | --- | --- |
| Entity | Attribute | Data Type | Domain |
| Champion | Name (Primary Key) | varchar(32) | Name of the Champion |
|  | Lane | varchar(32) | The lanes that a champion plays in. |
|  | Tags | varchar(32) | The roles that a champion is used for. |
| Bar | Name (Primary Key) | varchar(32) | The Champion Name. |
|  | Type | varchar(32) | The type of bar a champion has, ex. Energy Bar, Mana Bar, Rage Bar. |
| Stats | Name (Primary Key) | varchar(32) | The Champion Name. |
|  | Movement | small\_int | Integer greater than 0 to show movement speed. |
|  | Attack\_Damage | small\_int | Integer greater than 0 to show attack damage. |
|  | Ability\_Power | small\_int | Integer greater than 0 to show ability damage. |
|  | Armor | small\_int | Integer greater than 0 to show armor. |
|  | Magic\_Resist | small\_int | Integer greater than 0 to show magic resist. |
|  | Health | small\_int | Integer greater than 0 to show health. |
|  | Mana | small\_int | Integer greater than 0 to show mana. |
|  | Critical\_Strike\_Chance | double(5,2) | A positive percentage to two decimals to show the critical strike chance. |
|  | Cooldown\_Reduction | double(5,2) | A positive percentage to two decimals to show the cooldown reduction. |
|  | Attack\_Speed | double(5,4) | A positive percentage to two decimals to show the attack speed. |
|  | Attack\_Penetration | small\_int | Integer greater than 0 to show attack penetration. |
|  | Ability\_Penetration | small\_int | Integer greater than 0 to show ability penetration. |
| Masteries | ID (Primary Key) | small\_int | A positive Integer used to identify the combination of the mastery page. |
|  | Normalized\_Value | small\_int | A positive Integer that is calculated to show the value of the mastery for the champion. |
| Summoner Spell | Name (Primary Key) | varchar(32) | Name of the Summoner Spell. |
|  | Normalized Effect | small\_int | A positive Integer that is calculated to show the value of the normalized effect. |
|  | Base Time | small\_int | A positive Integer that shows the base time cooldown in seconds. |
| Champion Passive | Name (Primary Key) | varchar(32) | Name of the Champion Passive. |
|  | Normalized Passive Effect | small\_int | A positive Integer that is calculated to show the value of the normalized effect. |
|  | cooldown | small\_int | A positive Integer that shows the time between casts in seconds. |
| Ability | Name (Primary Key) | varchar(32) | Name of the Ability. |
|  | Champion | varchar(32) | Name of the champion who uses the ability. |
|  | Normalized\_Ability\_Effect | small\_int | A list of positive integers to show the effects. |
|  | Area\_of\_Effect | small\_int | A normalized integer representing the area an ability effects (zero being single-target). |
|  | Normalized\_Crowd\_Control\_Value | small\_int | A positive Integer that is calculated to show the value of the crowd control effect. |
|  | Cast\_Cost | small\_int | A positive Integer that show the value of the cast cost. |
|  | Number\_of\_Casts | small\_int | A positive Integer that show the total number of casts before going on cooldown. |
|  | Base\_Time | small\_int | A positive Integer that shows the base time cooldown in seconds. |
|  | Keyboard\_Key | varchar(1) | The key (q, w, e, or r) that is mapped to the ability |
| Item | Name (Primary Key) | varchar(32) | Name of the Item |
|  | Movement | small\_int | A positive Integer that shows the bonus movement provided by the item. |
|  | Attack\_Damage | small\_int | A positive Integer that shows the bonus attack damage provided by the item. |
|  | Ability\_Power | small\_int | A positive Integer that shows the bonus ability damage provided by the item. |
|  | Armor | small\_int | A positive Integer that shows the bonus armor provided by the item. |
|  | Magic\_Resist | small\_int | A positive Integer that shows the bonus magic resist provided by the item. |
|  | Health | small\_int | A positive Integer that shows the bonus health provided by the item. |
|  | Mana | small\_int | A positive Integer that shows the bonus mana provided by the item. |
|  | Critical\_Strike\_Chance | double(5,2) | A positive percentage that shows the bonus critical strike provided by the item. |
|  | Cooldown\_Reduction | double(5,2) | A positive percentage that shows the bonus cooldown reduction provided by the item. |
|  | Attack\_Speed | double(5,2) | A positive percentage that shows the bonus attack speed provided by the item. |
|  | Attack\_Penetration | small\_int | A positive Integer that shows the bonus attack penetration provided by the item. |
|  | Ability\_Power\_Penetration | small\_int | A positive Integer that shows the bonus ability power provided by the item. |
|  | Life\_Steal\_Percentage | double (5,2) | A positive percentage that shows the bonus lifesteal percentage provided by the item. |
|  | Mana\_Regen | double(5,2) | A positive percentage that shows the bonus mana regenerated per second provided by the item. |
|  | Health\_Regen | double(5,2) | A positive percentage that shows the bonus health regenerated per second provided by the item. |
|  | Gold\_Regen | double(5,2) | A positive percentage that shows the bonus gold regenerated per second provided by the item. |
| Item Recipe | ID (Primary Key) | small\_int | A positive Integer that is the ID assigned the recipe. It isn’t always unique. |
|  | Upgrade\_Cost | small\_int | A positive integer that shows the cost to upgrade the item with all the components. |
|  | Items | list | A list of items names that go into the Item. |
| Item Active | Name (Primary Key) | varchar(32) | The name of the item active. |
|  | Normalized\_Active\_Effect | small\_int | A calculated Integer that shows the effect of the item active. |
|  | Base\_Time | small\_int | A positive Integer that shows the base time cooldown in seconds. |
| Item Passive | Name (Primary Key) | varchar(32) | The name of the item passive. |
|  | Normalized\_Passive­\_Effect | small\_int | A positive integer that is calculated to show the passive effect. |
|  | Is\_Unique | boolean | A boolean to determine if a passive is unique or not. This is important because otherwise they do not stack. |
|  | Base\_Time | small\_int | A positive Integer that shows the base time cooldown in seconds. |
| Buff | Name (Primary Key) | varchar(32) | The name of the buff or debuff. |
|  | Normalized\_Buff\_Effect | small\_int | A positive or negative value that is calculated to show the positive or negative effect of the buff. |
|  | Base\_Time | small\_int | A positive Integer that shows the base time cooldown in seconds. |
| ================ | ================ | ================ | \*\*All Foreign Keys below can be referenced above. Unless otherwise noted. |
| Inventory | Champion.Name (Foreign Key) (Primary Key) | varchar(32) | Each champion should only have one inventory so this uses a primary key to be unique. |
|  | Item1.Name (Foreign Key) | varchar(32) | May be Null. |
|  | Item2. Name (Foreign Key) | varchar(32) | May be Null. |
|  | Item3. Name (Foreign Key) | varchar(32) | May be Null. |
|  | Item4.Name (Foreign Key) | varchar(32) | May be Null. |
|  | Item5.Name (Foreign Key) | varchar(32) | May be Null. |
|  | Item6.Name (Foreign Key) | varchar(32) | May be Null. |
| Has\_Item\_Recipie | Item.Name (Foreign Key) | varchar(32) | `` |
|  | Item\_Recipe.ID (Foreign Key) | small\_int | `` |
| Has\_Item\_Actives | Item.Name (Foreign Key) | varchar(32) | `` |
|  | Item\_Active.Name (Foreign Key) | varchar(32) | `` |
| Has\_Item\_Passives | Item.Name (Foreign Key) | varchar(32) | `` |
|  | Item\_Passive.Name (Foreign Key) | varchar(32) | `` |
| Applied\_Buffs | Champion.Name (Foreign Key) | varchar(32) | `` |
|  | Buff.Name (Foreign Key) | varchar(32) | `` |
| Page | Champion.Name (Primary Key) (Foreign Key) | varchar(32) | Each champion may only have 1 page so it is a primary key and unique. |
|  | Masteries.Name (Foreign Key) | small\_int | `` |
| Sums | Champion.Name (Foreign Key) | varchar(32) | `` |
|  | Summoner\_Spell.Name (Foreign Key) | varchar(32) | `` |
| Passive | Champion.Name (Primary Key) (Foreign Key) | varchar(32) | These names are a unique relation. |
|  | Champion\_Passive.Name (Primary Key) (Foreign Key) | varchar(32) | These names are a unique relation. |
| Abilities | Champion.Name (Foreign Key) | varchar(32) | `` |
|  | Champion\_Ability.Ability\_Name (Foreign Key) | varchar(32) | `` |

**Integrity and Constraints:**

The table has been designed to fit and accept most champions. There is a lot of variance and hard things to be able to be put in a table. So the statistic is closely calculated if possible to be put into the table in relation to all other champions. If a champion lacks an attribute then the base value closest to 0 or percentage closest to 0 will be entered unless otherwise noted. No varchar may be null unless otherwise noted by the table above. If it is a boolean then it is automatically considered false unless otherwise noted. Any normalized values will be calculated before hand using a consistent formula in order to calculate the general effect it has on the champion.

Each Thursday the table should also be updated since Riot pushes patches which affect champions and items. If a new item or champion is released then their scores will be calculated and added to the table.

Everything listed under the champion entity group use the same Champion Name as primary keys. The reason for this is because there are a lot of parts to a champion and so it is hard to keep all this data in one table. The champion name must be the same for all these sub tables.

Items in the Item recipe is a list of items no greater than five items that “build into” the item. We decided to keep this as a list since the list can vary in size, have no items and be an empty list. So for this purpose, we are going to keep with a list since it makes most sense to use it here.

We are also keeping the champion sub tables separate since often these sections are called independently of each other, updated independently, and joins are rare. So this case it will make sense to keep them as is. Other dependencies have been separated in order to maintain 4NF.

**Local Design**

Below is the relational database design listed with candidate keys, primary key, functional dependencies satisfied, and normal form for each.

**Champion** (Name, lane, tags)  
*Candidate Keys*: Name   
*Primary Key*: Name  
*Functional Dependencies satisfied*: Name -> lane, tags  
*Normal Form*: 4NF

**Bar** (Name, Type)  
*Candidate Keys*: Name   
*Primary Key*: Name  
*Functional Dependencies satisfied*: Name -> Type  
*Normal Form*: 4NF

**Stats** (Name, Movement, Attack\_Damage, Ability\_Power, Armor, Magic\_Resist, Health, Mana, Critical\_Strike\_Chance, Cooldown\_Reduction, Attack\_Speed, Attack\_Penetration, Ability\_Penetration)  
*Candidate Keys*: Name   
*Primary Key*: Name  
*Functional Dependencies satisfied*: Name -> Movement, Attack\_Damage, Ability\_Power, Armor, Magic\_Resist, Health, Mana, Critical\_Strike\_Chance, Cooldown\_Reduction, Attack\_Speed, Attack\_Penetration, Ability\_Penetration  
*Normal Form*: 4NF

**Masteries** (Name, Normalized\_Vale)  
*Candidate Keys*: ID   
*Primary Key*: ID  
*Functional Dependencies satisfied*: Name -> Normalized\_Vale  
*Normal Form*: 4NF

**Summoner\_Spell**(Name, Normalized\_Effect, Base Time)  
*Candidate Keys*: Name   
*Primary Key*: Name  
*Functional Dependencies satisfied*: Name -> Normalized\_Effect, Base\_Time  
*Normal Form*: 4NF

**Champion\_Passive**(Name, Normalized\_Effec, Base\_Time)  
*Candidate Keys*: Name   
*Primary Key*: Name  
*Functional Dependencies satisfied*: Name -> Normalized\_Effec, Base\_Time   
*Normal Form*: 4NF

**Champion\_Ability** (Champion, Ability\_Name, Normalized\_Effect, Area\_of\_Effect, Normalized\_Crowd\_Control, Cast\_Cost, Number\_of\_Casts, Base\_Time)  
*Candidate Keys*: Name   
*Primary Key*: Name  
*Functional Dependencies satisfied*: Name -> Normalized\_Effect, Area\_of\_Effect, Normalized\_Crowd\_Control, Cast\_Cost, Number\_of\_Casts, Base\_Time  
*Normal Form*: 4NF

**Item** (Name, Movement, Attack\_Damage, Ability\_Power, Armor, Magic\_Resist, Health, Mana, Critical\_Strike\_Chance, Cooldown\_Reduction, Attack\_Speed, Attack\_Penetration, Ability\_Penetration)  
*Candidate Keys*: Name   
*Primary Key*: Name  
*Functional Dependencies satisfied*: Name -> Movement, Attack\_Damage, Ability\_Power, Armor, Magic\_Resist, Health, Mana, Critical\_Strike\_Chance, Cooldown\_Reduction, Attack\_Speed, Attack\_Penetration, Ability\_Penetration  
*Normal Form*: 4NF

**Item\_Recipe** (ID, Upgrade\_Cost, Items)  
*Candidate Keys*: Name   
*Primary Key*: Name  
*Functional Dependencies satisfied*: Name -> UpgradeICost, Items  
*Normal Form*: This is not in first normal form since Items is still a list but in this case we want this as noted above. Otherwise this section was considered and normalized.

**Item\_Active** (Name, Normalized\_Effect, Base\_Time)  
*Candidate Keys*: Name   
*Primary Key*: Name  
*Functional Dependencies satisfied*: Name -> Normalized\_Effect, Base\_Time  
*Normal Form*: 4NF

**Item\_Passive** (Name, Normalized\_ Effect, Base\_Time, Is\_Unique)  
*Candidate Keys*: Name   
*Primary Key*: Name  
*Functional Dependencies satisfied*: Name -> Normalized\_ Effect, Base\_Time, Is\_Unique  
*Normal Form*: 4NF

**Buff** (Name, Normalized\_Passive\_Effect, Base\_Time)  
*Candidate Keys*: Name   
*Primary Key*: Name  
*Functional Dependencies satisfied*: Name -> Normalized\_Passive\_Effect, Base\_Time  
*Normal Form*: 4NF

I**nventory** (Champion.Name, Item1.Name, Item2.Name, Item3.Name, Item4.Name, Item5.Name, Item6.Name)  
*Candidate Keys*: Champion.Name   
*Primary Key*: Champion.Name  
*Functional Dependencies satisfied*: Name -> Item1.Name, Item2.Name, Item3.Name, Item4.Name, Item5.Name, Item6.Name  
*Normal Form*: 4NF

**Has\_Item\_Recipe** (Item.Name, Item\_Recipe.ID)

*Candidate Keys*: None  
*Primary Key*: None

*Functional Dependencies satisfied*: N/A  
*Normal Form*: 4NF

**Item\_Actives**(Item.Name, Item\_Active.Name)

*Candidate Keys*: None  
*Primary Key*: None  
*Functional Dependencies satisfied*: N/A  
*Normal Form*: 4NF

**Item\_Passives**(Item.Name, Item\_Passive.Name)

*Candidate Keys*: None  
*Primary Key*: None  
*Functional Dependencies satisfied*: N/A  
*Normal Form*: 4NF

**Applied\_Buffs**(Champion.Name, Buff.Name)

*Candidate Keys*: None  
*Primary Key*: None  
*Functional Dependencies satisfied*: N/A  
*Normal Form*: 4NF

**Page***(Champion.Name, Masteries.Name)*

*Candidate Keys*: Champion.Name  
*Primary Key*: Champion.Name  
*Functional Dependencies satisfied*: Champion.Name -> Masteries.Name  
*Normal Form*: 4NF

**Sums**(Champion.Name, Summoner\_Spell.Name)

*Candidate Keys*: None  
*Primary Key*: None  
*Functional Dependencies satisfied*: N/A  
*Normal Form*: 4NF

**Passive**(Champion.Name, Champion\_Passive.Name)

*Candidate Keys*: Champion.Name, Champion\_Passive.Name  
*Primary Key*: Champion.Name, Champion\_Passive.Name  
*Functional Dependencies satisfied*: Champion.Name <-> Champion\_Passive.Name  
*Normal Form*: 4NF

**Abilities**(Champion.Name, Champion\_Ability.Ability\_Name)

*Candidate Keys*: None  
*Primary Key*: None  
*Functional Dependencies satisfied*: N/A  
*Normal Form*: 4NF