Proyecto Final

Javier,Samuel,Ines

League of Legends es un MOBA que ha ganado popularidad a lo largo de los años. Con la creación de personajes al estilo RPG en el transcurso de un juego de 45 minutos.

Dicho esto, el meta (construcciones más populares) tiende a ser establecido por los profesionales, que juegan en un nivel que la mayoría de los jugadores nunca alcanzarán. Si se sabe esto, ¿por qué no adoptar un enfoque más matemático del juego? ¿Cuál es el uso más eficiente de su oro? Como Talon, ¿cuál es el daño máximo que puedes infligir en un solo golpe? Ya tengo dos espadas largas, ¿qué elemento puedo construir para la mayor penetración de armadura?

Cada campeón tiene un conjunto de las siguientes estadísticas: Daño de ataque, Daño de habilidad, Salud temporal, Salud, Maná, Energía, Rabia, Resistencia mágica, Regeneración de salud, Regeneración de maná, Armadura, Velocidad de ataque, Probabilidad de golpe crítico, Daño de golpe crítico, Magia Penetración, Penetración de armadura, Velocidad de movimiento y Reducción de enfriamiento.

Cada uno de ellos se ve afectado por el nivel del campeón. Estas estadísticas determinan el daño/curación/protección/etc. de las habilidades de un campeón, de las cuales cada campeón tiene cuatro y una pasiva. Cada campeón también tiene dos hechizos de invocador que se seleccionan antes de que comience el juego que, para el propósito de esta explicación, son esencialmente dos habilidades adicionales elegidas de un grupo de diez.

Icono de Validado por la comunidad

Con algunas excepciones, las 4 habilidades principales de un campeón son activas mientras que la pasiva es, como su nombre lo indica, pasiva. Cada habilidad creará algún efecto en el juego o modificará una estadística de campeón (mejora o desventaja). Algunas habilidades afectan un área, se llaman habilidades AoE (área de efecto). Otras son habilidades de control de multitudes (CC), es decir, habilidades que reducen el movimiento o afectan la capacidad de ataque de un campeón.

Un campeón también puede tener hasta seis elementos y una poción activa en un momento dado. Los artículos pueden convertirse en mejores artículos usando varias rutas de construcción o recetas, combinándolos con otros artículos y gastando oro. Cada nivel de elementos proporciona estadísticas y habilidades adicionales. La mayoría de los elementos simplemente dan aumentos de estadísticas, sin embargo, algunos tienen habilidades adicionales. Estos pueden ser pasivos o activos. Todos los elementos activos son únicos, lo que significa que si construyes el mismo elemento dos veces y usas el elemento activo en uno, ambos elementos activos se enfriarán. Algunas pasivas de elementos son únicas, sin embargo, otras se acumularán como efectos de elementos normales.

Antes de que comience un juego, cada campeón tiene un conjunto de maestrías para elegir. Hay cinco árboles de maestría. Cada campeón selecciona cuatro dominios primarios de un árbol y dos dominios secundarios de otro. Estos se correlacionan con las habilidades pasivas.

Cada habilidad de cada tipo tiene un nombre, ícono, rango y tiempo de reutilización.

Esta base de datos proporcionará una forma de determinar, matemáticamente, cómo construir un campeón en función de las estadísticas que desea maximizar, así como de los usos eficientes del oro.

**Consultas :**

1. ¿Qué seis elementos dan la armadura plana más alta?

2. ¿Qué seis elementos dan las estadísticas más altas con pasivos y habilidades?

3. ¿Qué objeto tiene el daño de ataque más alto?

4. ¿Cuál es el efecto total de las maestrías?

5. ¿Cuál es el efecto total más débil basado en?

6. ¿Cuál es el costo total?

7. ¿Qué combinación de campeón y objeto te da la salud más alta posible?

8. ¿Cuál es el mayor daño que se puede infligir en un solo golpe?

9. ¿Cuántas veces puede Anivia usar Crystalize en un minuto?

10. ¿Qué habilidad que no usa maná tiene la producción de daño fijo más alta?

11. ¿Qué elemento tiene el tiempo de reutilización más corto?

12. ¿Qué habilidades tiene CC (control de multitudes)?

13. ¿Qué campeones no tienen AP (poder de habilidad)?

14. ¿Puedo comprar los artículos seleccionados con el oro?

15. ¿Cuál es la puntuación normalizada total de un campeón con los distintos atributos seleccionados?

16. ¿Cuál es el Medio para la Salud de los campeones?

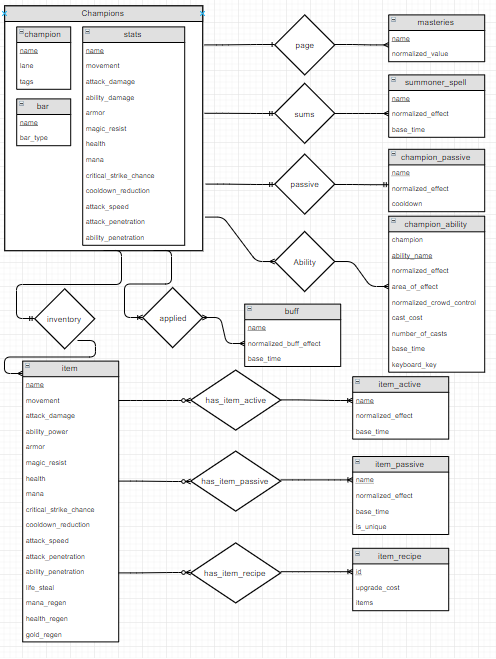
17. ¿Cuántos elementos dan bonificación de daño de ataque?

18. ¿Cuántos elementos tienen pasivos o activos?

19. ¿Cuál es la configuración más costosa para un campeón?

20. ¿Cuántos elementos se pueden usar actualmente en Summoner's Rift?

Diagramas ER.



**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) | `` |

**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