League of Legends

Champion database

By Brad Huntington

12/2/13

Table of contents

Executive Summary 3

**Entity Relationship Diagram4**

**Table Create Statements 5**

**League Champions5**

**Rolls6**

**Champion roll7**

**Item Counters8**

**Champion item Counters9**

**Lane10**

**Champion lane11**

**Counters12**

**Champion counter13**

**Views14**

**Reports & Queries15**

Stored Procedures16

Triggers17

Security18,19

Implementation Notes 20

Known Problems 21

Future Enhancements 22

Executive Summary

This database is designed to aid in champion selection for League of Legends a Multi Player Online Battle Arena. In a typical league of legends match, there will be two teams of five and at the start of the match each player more commonly referred to as “Summoner” will chose his champion. The purpose of this database is to make it easier for a player to counter pick the enemy team’s champions. The database allows strategic game data to be stored such as: champion names, specific roles each champion can satisfy, items that counter champions in game, typical lanes that each champion goes to, and champions that will gain the advantaged when matched up against other champions.

Entity Relationship Diagram

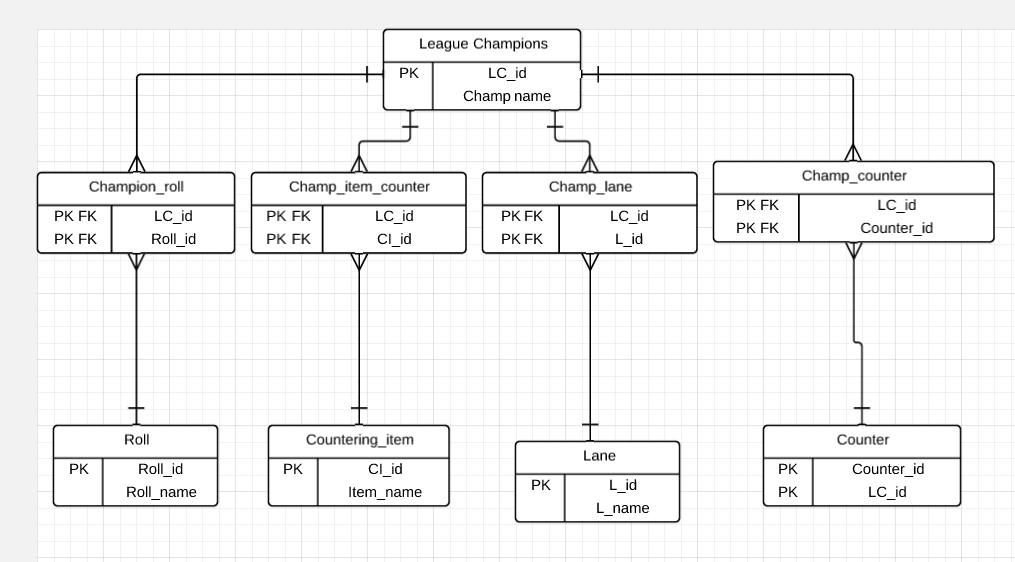


Table Create Statements

League Champions Table

Table create statements



The League Champions table has one unique field, Champ\_name. This field paired with the LC\_id will help us keep track of the champions.

Functional dependencies



LC\_id🡪Champ\_name

Create Statement

drop table if exists League\_champions;



create table if not exists League\_champions

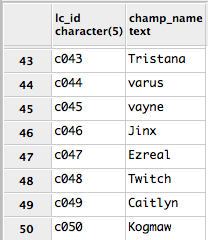
(

LC\_id CHAR(5) NOT NULL,

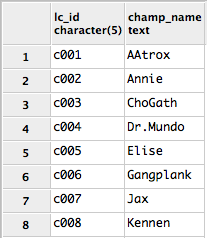
Champ\_name TEXT NOT NULL,

Primary key (LC\_id)

);

Sample data





Roll Table



The Roll table has one unique field, Roll\_name. This field paired with the Roll\_id will help us keep track of the champion rolls.

Functional dependencies



Roll\_id🡪Roll\_name

Create Statement

drop table if exists Roll;



create table if not exists Roll

(

Roll\_id CHAR(4) NOT NULL,

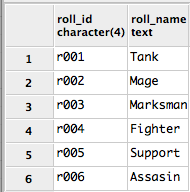
Roll\_name TEXT,

Primary key (Roll\_id)

);

Sample data





Champion Roll Table



The Champion Roll table has no unique fields. Its primary purpose is to eliminate the many to many relationship of Champions and Rolls. This table will link each champion id to a roll id.

Functional dependencies



None

Create Statement

drop table if exists Champion\_roll;



create table if not exists Champion\_roll

(

LC\_id CHAR(5) not null REFERENCES League\_champions (LC\_id),

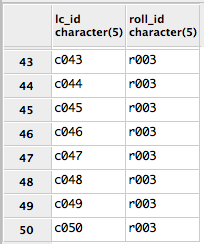
Roll\_id CHAR (5) not null REFERENCES Roll (Roll\_id),

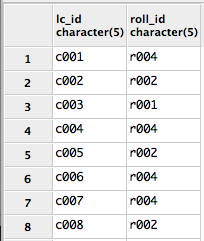
primary key (LC\_id, Roll\_id)

);

Sample data







Cou

Countering item Table



The Countering item table has one unique field, Countering\_item\_name. This field paired with the CI\_id will help us keep track of some key items made specifically to counter certain types of champions.

Functional dependencies



CI\_id🡪Countering\_item\_name

Create Statement

drop table if exists Countering\_item;



create table if not exists Countering\_item

(

CI\_id CHAR(6) NOT NULL,

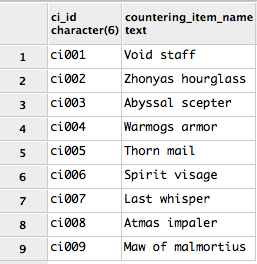
Countering\_item\_name TEXT,

Primary key (CI\_id)

);

Sample data





Champion Item Counter Table



The Champion item counter table has no unique fields. Its primary purpose is to eliminate the many to many relationship between Champions and countering items. This table will link each champion with three specific items that counter the champion’s role.

Functional dependencies



None

Create Statement

drop table if exists Champion\_item\_counter;



create table if not exists Champion\_item\_counter

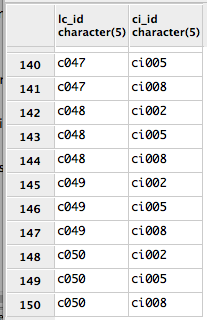
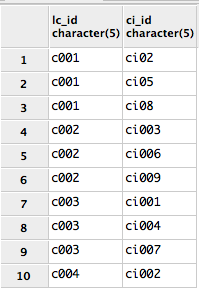
(

LC\_id CHAR(5) not null REFERENCES League\_champions (LC\_id),

CI\_id CHAR (5) not null REFERENCES Countering\_item (CL\_id),

primary key (LC\_id, CL\_id)

);

Sample data



Lane Table



The Lane table has one unique field, Lane\_name. This field paired with the L\_id will help us keep track of the Lanes that each champion can go to.

Functional dependencies



L\_id🡪Lane\_name

Create Statement

drop table if exists Lane;



create table if not exists Lane

(

L\_id CHAR(4) NOT NULL,

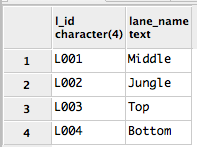
Lane\_name TEXT,

Primary key (L\_id)

);

Sample data





Champ lane Table



The Champ lane table has no unique fields. Its primary purpose is to eliminate the many to many relationship between Champions and lane table. This table will link each champion id to the id of the lanes that the champion is viable to be played in.

Functional dependencies



None

Create Statement

drop table if exists Champ\_lane;



create table if not exists Champ\_lane

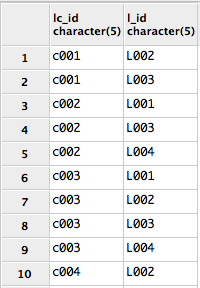
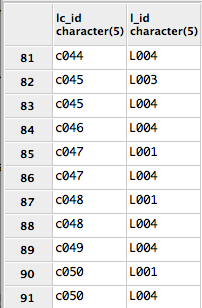
(

LC\_id CHAR(5) not null REFERENCES League\_champions (LC\_id),

L\_id CHAR (5) not null REFERENCES Lane (L\_id),

primary key (LC\_id, L\_id)

);

Sample data



Counter Table



The Counter table has no unique fields. Its primary purpose is to assign a Counter id to each champion so that it may be referenced in the Champion counter table.

Functional dependencies



None

Create Statement

drop table if exists Counter;



create table if not exists Counter

(

LC\_id CHAR(5) not null REFERENCES League\_champions (LC\_id),

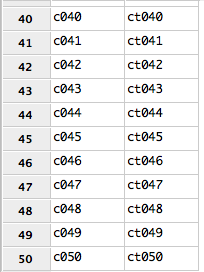
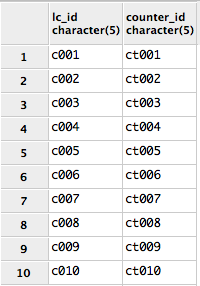
Counter\_id CHAR (5) NOT NULL,

primary key (LC\_id,Counter\_id)

);

Sample data





Champ Counter Table



The Champ counter table has no unique fields. Its primary purpose is to take an assigned Counter id from the counter table and link it to a champion id. This links each champion to a countering champion.

Functional dependencies



None

Create Statement

drop table if exists Champ\_counter;



create table if not exists Champ\_counter

(

Counter\_id CHAR (5) not null,

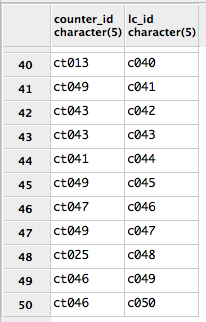
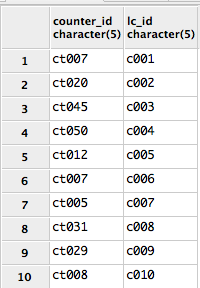
LC\_id CHAR(5) not null REFERENCES League\_champions (LC\_id),

primary key (LC\_id, Counter\_id)

);

Sample data





Views

**This view uses the whole database to bring useful information to the person picking a champion. The user will be supplied with the champion name, the roll of the champion, the items that counter the champion, and the champion counter.**

**Create View super\_query as**

SELECT DISTINCT

lc.LC\_id, lc1.Champ\_name,

r.roll\_name, l.Lane\_name,

CI.countering\_Item\_name,

c.LC\_id, lc2.Champ\_name

FROM

Champ\_lane cl, lane l,

Champion\_item\_counter cic,

Countering\_item ci, Champion\_roll cr,

Roll r, Champ\_Counter cc,

League\_champions lc, League\_champions lc1,

League\_champions lc2, counter c

WHERE

r.Roll\_id=cr.Roll\_id

AND cr.LC\_id=lc.LC\_id

AND c.Counter\_id=cc.counter\_id

AND cc.LC\_id=lc.LC\_id

AND cc.LC\_id = lc1.lc\_id

AND c.LC\_id = lc2.lc\_id

AND ci.CI\_id=CIC.CI\_id

AND cic.LC\_id=lc.LC\_id

AND l.L\_id=cl.l\_id

AND cl.LC\_id=lc.LC\_id

ORDER BY

lc.LC\_id

**These four reports will be helpful to any Summoner that wishes to learn more specific aspects each champion.**

Reports and Queries

**This query will generate the name of the champion along with specific items that counter the champion.**

Select distinct

lc.LC\_id,

lc.Champ\_name,

CI.countering\_Item\_name

from
Champion\_item\_counter cic,

League\_champions lc,

Countering\_item ci

where

CI.CI\_id=CIC.CI\_id

AND

CIC.LC\_id=lc.LC\_id

order by
lc.LC\_id

**This query will generate the Champion name and the roll that the champion fulfills.**

Select distinct

lc.LC\_id,

lc.Champ\_name,

r.Roll\_name

from

Champion\_roll cr,

League\_champions lc,

Roll r

where

r.Roll\_id=cr.Roll\_id

AND

cr.LC\_id=lc.LC\_id

order by
lc.LC\_id

**This query will generate the name of the champion along with specific counter picks of the champion**

select distinct

lc.LC\_id, lc1.Champ\_name,
c.LC\_id,

lc2.Champ\_name

from

Champ\_Counter cc,
League\_champions lc,

League\_champions lc1,

League\_champions lc2,

counter c

where
c.Counter\_id=cc.counter\_id

AND
cc.LC\_id=lc.LC\_id

and
cc.lc\_id = lc1.lc\_id

and
c.lc\_id = lc2.lc\_id

order by
lc.LC\_id

**This query will generate the name of the champion along with specific Lanes the champion is viable to be played in.**

Select distinct

lc.LC\_id,

lc.Champ\_name,

l.Lane\_name

from

Champ\_lane cl,

League\_champions lc,

lane l

where

l.l\_id=cl.l\_id

AND

cl.LC\_id=lc.LC\_id

order by
lc.LC\_id

Stored procedures

**This stored procedure gives helpful champion data. It will return the champion name, the role the champion plays and the lane the champion goes to.**

CREATE FUNCTION CHAMP\_INFO (Champ\_name text)

returns table (Champ\_name text, roll\_name text, lane\_name text) as $$

select

lc.Champ\_name, r.Roll\_name, l.lane\_name

from

Champ\_lane cl, lane l, League\_champions lc, Champion\_roll cr, roll r where

r.Roll\_id=cr.Roll\_id

AND cr.LC\_id=lc.LC\_id

AND l.l\_id=cl.l\_id

AND cl.lc\_id=lc.lc\_id

$$ language 'sql';

select \* from CHAMP\_INFO('Champ\_name')

**The following trigger calls this stored procedure, it checks to see if a champion has an empty value in the role field.**

CREATE

FUNCTION new\_data\_check()

returns trigger as $$

BEGIN if (roll\_id is null)

Then

UPDATE

set Roll\_id = 'NA'

where Roll\_id = null;

END

$$language plpgsql;

Triggers

**This trigger is used when a new champion is entered into the database. Once a new champion is released it takes a while to determine certain characteristics. For example, what roll the champion will excel at. Therefore we need to fill in potential missing data.**

CREATE

trigger New\_champion\_data\_check

After insert

on

champion\_roll,

for each row

execute new\_data\_check

Security

**To ensure security we make a user and an admin. The Admin’s control over the database is unconstrained while the user can only view data.**

SQL Code for security



CREATE USER League\_admin WITH PASSWORD ‘alpaca’

Grant Select, update, delete, insert on League\_champions to League\_admin

Grant Select, update, delete, insert on Champ\_counter to League\_admin

Grant Select, update, delete, insert on Champ\_lane to League\_admin

Grant Select, update, delete, insert on Champion\_item\_counter to League\_admin

Grant Select, update, delete, insert on Champion\_lane to League\_admin

Grant Select, update, delete, insert on Counter to League\_admin

Grant Select, update, delete, insert on Countering\_item to League\_admin

Grant Select, update, delete, insert on Lane to League\_admin

Grant Select, update, delete, insert on Roll to League\_admin

-- This code will give the user League\_admin all the unrestricted access to the database

CREATE USER League\_user WITH PASSWORD ‘alpaca’

Revoke all on League\_champions from League\_user

Revoke all on Champ\_counter from League\_user

Revoke all on Champ\_lane from League\_user

Revoke all on Champion\_item\_counter from League\_user

Revoke all on Champion\_lane from League\_user

Revoke all on Counter from League\_user

Revoke all on Countering\_item from League\_user

Revoke all on Lane from League\_user

Revoke all on Roll from League\_user

Grant Select on League\_champions from League\_user

Grant Select on Champ\_counter from League\_user

Grant Select on Champ\_lane from League\_user

Grant Select on Champion\_item\_counter from League\_user

Grant Select on Champion\_lane from League\_user

Grant Select on Counter from League\_user

Grant Select on Countering\_item from League\_user

Grant Select on Lane from League\_user

Grant Select on Roll from League\_user

Implementation Notes

* Implementation is as easy as copying the tables provided.
* Once tables are created, all that needs to be done is insert desired champion data.
* The 67 not included champions may also be added into the database before implementation.

Known Problems

* The amount of champions is always growing the space provided by LC\_id (league Champion) and Counter\_id will be insufficient if the number of champions grows past 999.
* As a result of the ever growing champion pool new items are inserted and removed from the game in order to balance the power of all champions, if an item on the countering item list were to be removed it would effect multiple tables.

Future Enhancements

* Currently the database includes only critical data to allow the user to make an educated decision in champion selection, however in the future more non-critical data could be added.
  + Specific Champion spells
  + Summoner spells
  + 168 more items could be added
* The rest of the champions should be added to complete the database
* Triggers for every table will be implimented
* Multiple Champion counters and countering items should be added in order to supply the user with more data and allow people to have a choice when choosing a countering champion