

League of Legends Champion database

By Brad Huntington
12/2/13

Table of contents

Executive Summary	3
Entity Relationship Diagram	4
Table Create Statements	5
League Champions.....	5
Rolls	6
Champion roll.....	7
Item Counters	8
Champion item Counters	9
Lane	10
Champion lane.....	11
Counters	12
Champion counter.....	13
Views.....	14
Reports & Queries.....	15
STORED PROCEDURES	16
TRIGGERS.....	17
SECURITY	18,19
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

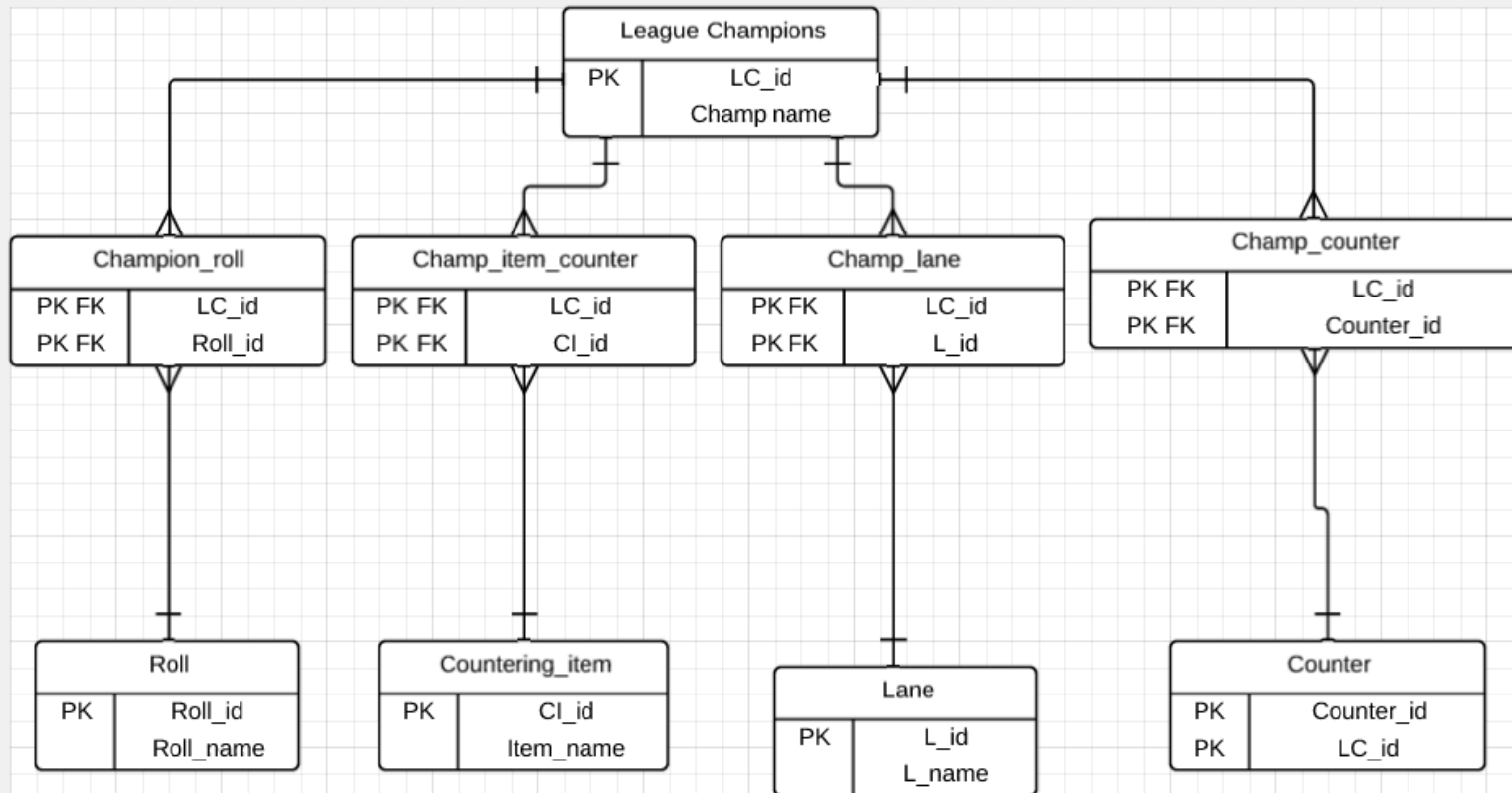


Table create statements

League Champions Table

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

	lc_id character(5)	champ_name text
1	c001	AAatrox
2	c002	Annie
3	c003	ChoGath
4	c004	Dr.Mundo
5	c005	Elise
6	c006	Gangplank
7	c007	Jax

	lc_id character(5)	champ_name text
43	c043	Tristana
44	c044	varus
45	c045	vayne
46	c046	Jinx
47	c047	Ezreal
48	c048	Twitch

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

	roll_id character(4)	roll_name text
1	r001	Tank
2	r002	Mage
3	r003	Marksman
4	r004	Fighter
5	r005	Support
6	r006	Assasin

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

	lc_id character(5)	roll_id character(5)
1	c001	r004
2	c002	r002
3	c003	r001
4	c004	r004
5	c005	r002
6	c006	r004
7	c007	r004

C
ou

	lc_id character(5)	roll_id character(5)
43	c043	r003
44	c044	r003
45	c045	r003
46	c046	r003
47	c047	r003
48	c048	r003
49	c049	r003

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

	ci_id character(6)	countering_item_name text
1	ci001	Void staff
2	ci002	Zhonyas hourglass
3	ci003	Abyssal scepter
4	ci004	Warmogs armor
5	ci005	Thorn mail
6	ci006	Spirit visage
7	ci007	Last whisper
8	ci008	Atmas impaler
9	ci009	Maw of malmortius

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

	lc_id character(5)	ci_id character(5)
1	c001	ci02
2	c001	ci05
3	c001	ci08
4	c002	ci003
5	c002	ci006
6	c002	ci009
7	c003	ci001
8	c003	ci004
9	c003	ci007
10	c004	ci002

	lc_id character(5)	ci_id character(5)
140	c047	ci005
141	c047	ci008
142	c048	ci002
143	c048	ci005
144	c048	ci008
145	c049	ci002
146	c049	ci005
147	c049	ci008
148	c050	ci002
149	c050	ci005
150	c050	ci008

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

$L_id \rightarrow Lane_name$

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

	L_id character(4)	lane_name text
1	L001	Middle
2	L002	Jungle
3	L003	Top
4	L004	Bottom

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

	lc_id character(5)	L_id character(5)
1	c001	L002
2	c001	L003
3	c002	L001
4	c002	L003
5	c002	L004
6	c003	L001
7	c003	L002
8	c003	L003
9	c003	L004
10	c004	L002

	lc_id character(5)	L_id character(5)
81	c044	L004
82	c045	L003
83	c045	L004
84	c046	L004
85	c047	L001
86	c047	L004
87	c048	L001
88	c048	L004
89	c049	L004
90	c050	L001
91	c050	L004

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

	lc_id character(5)	counter_id character(5)
1	c001	ct001
2	c002	ct002
3	c003	ct003
4	c004	ct004
5	c005	ct005
6	c006	ct006
7	c007	ct007
8	c008	ct008
9	c009	ct009

40	c040	ct040
41	c041	ct041
42	c042	ct042
43	c043	ct043
44	c044	ct044
45	c045	ct045
46	c046	ct046
47	c047	ct047
48	c048	ct048
49	c049	ct049

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

	counter_id character(5)	lc_id character(5)
1	ct007	c001
2	ct020	c002
3	ct045	c003
4	ct050	c004
5	ct012	c005
6	ct007	c006
7	ct005	c007
8	ct031	c008
9	ct029	c009

	counter_id character(5)	lc_id character(5)
40	ct013	c040
41	ct049	c041
42	ct043	c042
43	ct043	c043
44	ct041	c044
45	ct049	c045
46	ct047	c046
47	ct049	c047
48	ct025	c048
49	ct046	c049

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
```

Reports and Queries

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

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 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
```

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
```

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 implemented
- 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