4/22/2014

RIOT GAME

LEAGUE OF LEGEND DATABASE PROPOSAL

For Alan Labouseur | Jing Tai Wu

Table of Contents

Executive Summary

Overview

League of Legends (LoL) is a multiplayer online battle arena video game developed by Riot Games. It is inspired by a very famous Tower defense map Dota, Defense of the Ancients, powered by the Warcraft game engine. Unlike Dota, League of Legend is much easier to pick up and very difficult to master. Due to its "beginner-friendly" characteristic, LoL's player community grew rapidly in the past few years.

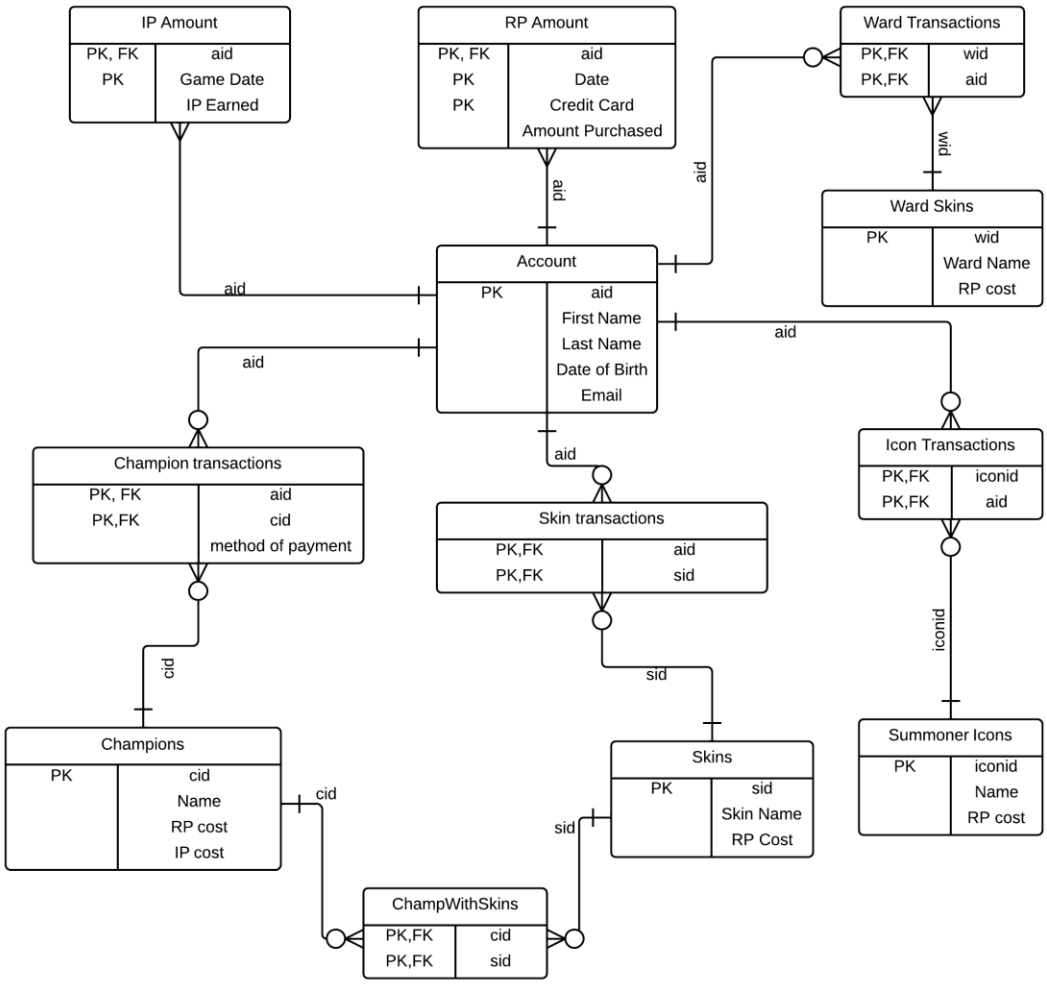
With its large player base, Riot Games must seek a way to earn profit and expand the company. One major way of earning profit is through selling Champions, skins and themes. There are two types of currencies in League of Legend: IP (influential points) and RP (Riot Points). Players earn IP through battling against other players or computers and RP is only earned through purchasing with real world currency. We must build an efficient data base that will manage all transactions made for each player's account.

Objectives

The purpose of this document is to describe and outline a database system that will manage player's account information and various transactions. It will keep track of the champions and skins purchased by each account. This allows the player to be able to play a champion with a skin that he or she purchased.

This report will provide an E/R diagram of all the tables and its relations in this database along with their functional dependencies. It will also provide stored procedures, triggers and views. This database is designed and tested on PostgreSQL.

Entity Relationship Diagram



Tables

Account Table

This table holds user's account information including FirstName, LastName, DoB and Email. I created aid as a primary key for this table.

Functional dependency:

aid -> FirstName, LastName, DoB, Email

SQL script:

```
CREATE TABLE Account(
        aid             char(7) not null,
        FirstName       text,
        LastName        text,
        DoB             date,
        Email           text,


        primary key(aid)
);
```

IPAmt Table

Since IP(influence points) is only earned through playing games with other people, this table will tell us what date and time the player played game at and the amount of IP earned for that game.

Functional Dependency:

aid, GameDate -> IPErned

SQL script:

```
CREATE TABLE IPAmt(
        aid             char(7) not null references Account(aid),
        GameDate    timestamp,
        IPErned             integer,


        primary key(aid, GameDate)
);
```

RPAmt Table:

This table shows the RP(Riot Point) purchased for each account using a credit card.

Functional Dependency:

aid, Date, CrdCd -> Amt

SQL script:

```
CREATE TABLE RPAmt(
        aid             char(7) not null references Account(aid),
        Date            timestamp,
        CrdCd                   text,
        Amt             integer,


        primary key(aid, Date, CrdCd, Amt)
);
```

Champions Table:

This table records all the champions that are being released so far.

Functional Dependency:

cid -> Name, RPCost, IPCost

SQL script:

```
CREATE TABLE Champions(
        cid             char(7) not null,
        Name            text,
        RPCost                  integer,
        IPCost          integer,


        primary key(cid)
);
```

ChmpTrans Table

This table shows what account bought what champion using either RP or IP. Aid and cid are the primary keys for this table because an account can only purchase the same champion once.

Functional Dependency:

Aid,cid -> PymtMthd

SQL script:

```
CREATE TABLE ChmpTrans(
        aid             char(7) not null references Account(aid),
        cid             char(7) not null references Champions(cid),
        PymtMthd        text,


        primary key(aid, cid)
);
```

Skins Table

This table shows the skin names and the RP cost of the skin. Note that it is only RP Cost because players cannot purchase any skins with IP. (We must make money somehow and this is the most lucrative way)

Functional Dependency:

sid -> Name, RPCost

SQL script:

```
CREATE TABLE Skins(
        sid             char(7) not null,
        Name            text not null unique,
        RPCost                  integer,


        primary key(sid)
);
```

ChampsWithSkins Table

This table shows what champions have what skins.

Functional Dependencies:

cid, sid ->

SQL script:

```
CREATE TABLE ChampWithSkins(
        cid             char(7) not null references Champions(cid),
        sid             char(7) not null references Skins(sid),


        primary key(cid, sid)
);
```

SkinTrans Table

This table shows what account bought which skins.

Functional Dependencies:

aid, sid ->

SQL script:

```
CREATE TABLE SkinTrans(
        aid             char(7) not null references Account(aid),
        sid             char(7) not null references Skins(sid),


        primary key(aid, sid)
);
```

SummonerIcons Table

This table shows available summoner icons for purchase. Same as the skin, summoner icons are only purchasable by RP(aka real money).

Functional Dependency:

Iconid -> Name, RPCost

SQL Script:

```
CREATE TABLE SummonerIcons(
        iconid          char(7) not null,
        Name            text,
        RPCost                  integer,


        primary key(iconid)
);
```

IconTrans Table

This table shows which account bought what icon.

Functional Dependency:

iconid, aid ->

SQL Script:

```
CREATE TABLE IconTrans(
        iconid          char(7) not null references SummonerIcons(iconid),
        aid             char(7) not null references Account(aid),


        primary key(iconid, aid)
);
```

WardSkins Table

This table contains different skins for wards. A ward is an item in LoL that provides vision on the map. Of course we can make it look fancier and grab a few more bucks from the players.

Functional Dependency:

wid -> Name, RPCost

SQL Script:

```
CREATE TABLE WardSkins(
        wid             char(7) not null,
        Name            text,
        RPCost                  integer,


        primary key(wid)
);
```

WardTrans Table

This table shows which accounts bought what ward skins.

Functional Dependency:

Wid, aid ->

SQL Script:

```
CREATE TABLE WardTrans(
        wid             char(7) not null references WardSkins(wid),
        aid             char(7) not null references Account(aid),


        primary key(wid, aid)
);
```

Insert Statements

Fill this database with sample data.

Account Table:

	aid character(7)	firstname text	lastname text	dob date	email text
1	a001	Jing Tai	Wu	1993-03-23	jingthebest@gmail.com
2	a002	Alan	Labouseur	1800-03-23	YI-ER-SAN@gmail.com
3	a003	Brian	Dones	1993-09-18	ILoveShakira@gmail.com
4	a004	Matt	Ancona	1992-02-19	AccountingSucks@gmail.com
5	a005	Robert	Paquini	1990-03-23	AmericanWannabe@gmail.com
6	a006	Matthew	Johnson	1700-07-23	TheIndianChief@gmail.com
7	a007	James	Bond	1930-08-25	StrongerThanChuckNorris@gmail.com

RPAmt Table:

	aid character(7)	date timestamp without time zone	crdcd text	amt integer
1	a001	2014-04-24 12:12:12	American Express	1000
2	a002	2014-04-10 12:12:12	VISA	2000
3	a003	2014-04-12 12:12:12	MasterCard	3000
4	a004	2014-01-24 12:12:12	VISA GiftCard	500
5	a005	2014-03-24 12:12:12	Discover	800
6	a006	2013-02-24 12:12:12	Captial One	6000
7	a007	2014-04-22 12:12:12	Citi Bank	1000
8	a001	2014-04-23 12:12:12	American Express	3000

IPAmt:

	aid character(7)	gamedate timestamp without time zone	iperned integer
1	a001	2013-03-24 11:11:11	90
2	a002	2013-03-24 11:11:11	80
3	a003	2013-03-24 11:11:11	40
4	a004	2013-03-24 11:11:11	60
5	a005	2013-03-24 11:11:11	30
6	a006	2013-03-24 11:11:11	60
7	a007	2013-03-24 11:11:11	40

Champions Table:

	cid character(7)	name text	rpcost integer	ipcost integer
1	c001	Aatrox	975	6300
2	c002	Ahri	975	6300
3	c003	Akali	480	3150
4	c004	Alistar	260	450
5	c005	Amumu	480	3150
6	c006	Anivia	480	3150
7	c007	Annie	260	450
8	c008	Ashe	260	450

ChmpTrans Table:

	aid character(7)	cid character(7)	pymtmthd text
1	a001	c001	RP
2	a002	c002	RP
3	a003	c001	RP
4	a004	c003	RP
5	a005	c003	RP
6	a006	c004	RP
7	a007	c007	RP
8	a002	c008	RP

Skins Table:

	sid character(7)	name text	rpcost integer
1	s001	Justicar Aatrox	975
2	s002	Dynasty Ahri	975
3	s003	Blood Moon Akali	975
4	s004	Nurse Akali	975
5	s005	Infernal Alistar	975
6	s006	LongHorn	520
7	s007	Golden Alistar	390
8	s008	Sad Robot Amumu	1350
9	s009	Emumu	520

ChampWithSkins:

	cid character(7)	sid character(7)
1	c001	s001
2	c002	s002
3	c003	s003
4	c003	s004
5	c004	s005
6	c004	s006
7	c004	s007
8	c005	s008
9	c005	s009

SkinTrans

	aid character(7)	sid character(7)
1	a001	s001
2	a001	s002
3	a002	s007
4	a002	s004
5	a003	s002
6	a003	s003
7	a004	s004
8	a005	s005
9	a006	s007

SummonerIcon Table

	iconid character(7)	name text	rpcost integer
1	cn001	CLG	260
2	cn002	TSM	260
3	cn003	DNG	260
4	cn004	WE	260
5	cn005	SK1	260
6	cn006	OMG	260
7	cn007	LGD	260

IconTrans Table

	iconid character(7)	aid character(7)
1	cn001	a001
2	cn002	a001
3	cn003	a001
4	cn004	a001
5	cn005	a001
6	cn006	a001
7	cn001	a002
8	cn001	a003

WardSkins Table

	wid character(7)	name text	rpcost integer
1	w001	Season 3 Victorious Ward	640
2	w002	Starcall Ward	640
3	w003	Luminosity Ward	640
4	w004	Ward of Draven	640

WardTrans Table

	wid character(7)	aid character(7)
1	w001	a001
2	w002	a002
3	w003	a005
4	w004	a004
5	w002	a003

Views

Display all the RP purchased by Different Accounts (The accounting department probably needs this, we need to know how much money we are making.)

View RPIncome

```
create view RPIncome

as

select r.Date as "Date Purchased", r.Amt as "RP Purchased", a.FirstName,
a.LastName

from Account a, RPAmt r

where a.aid = r.aid
```

	Date Purchased timestamp without time zone	RP Purchased integer	firstname text	lastname text
1	2014-04-24 12:12:12	1000	Jing Tai	Wu
2	2014-04-10 12:12:12	2000	Alan	Labouseur
3	2014-04-12 12:12:12	3000	Brian	Dones
4	2014-01-24 12:12:12	500	Matt	Ancona
5	2014-03-24 12:12:12	800	Robert	Paquini
6	2013-02-24 12:12:12	6000	Matthew	Johnson
7	2014-04-22 12:12:12	1000	James	Bond
8	2014-04-23 12:12:12	3000	Jing Tai	Wu

Display Champion purchased by different account so we know what's popular and what's not. (Champion Design Department needs this to make a decision on whether to buff a champion or not).

View ChampRec

```
create view ChampRec

as

select a.FirstName, a.LastName, c.Name as "Champion Name"

from Account a, Champions c, ChmpTrans ct

where a.aid = ct.aid

and   c.cid = ct.cid
```

	firstname text	lastname text	Champion Name text
1	Jing Tai	Wu	Aatrox
2	Alan	Labouseur	Ahri
3	Brian	Dones	Aatrox
4	Matt	Ancona	Akali
5	Robert	Paquini	Akali
6	Matthew	Johnson	Alistar
7	James	Bond	Annie
8	Alan	Labouseur	Ashe

Queries

Total RP Purchased:

We must know what the total amount of RP is being purchased for each account so that we can double check the player's RP before they make a purchase.

SQL Script

```
select a.FirstName, a.LastName, sum(r.Amt) as "Total RP"

from Account a, RPAmt r

where a.aid = r.aid

group by a.FirstName, a.LastName
```

How it looks like:

	firstname text	lastname text	Total RP bigint
1	Matthew	Johnson	6000
2	Jing Tai	Wu	4000
3	Brian	Dones	3000
4	Alan	Labouseu	2000
5	Robert	Paquini	800
6	James	Bond	1000
7	Matt	Ancona	500

Total IP Earned:

We must also know the total amount of IP is being earned in each account.

SQL Script

```
select a.FirstName, a.LastName, sum(i.IPErned) as 'Total IP'

from Account a, IPAmt i

where a.aid = i.aid

group by a.FirstName, a.LastName
```

How it looks like:

	firstname text	lastname text	Total IP bigint
1	Matthew	Johnson	60
2	Jing Tai	Wu	90
3	Brian	Dones	40
4	Alan	Labouseu	80
5	Robert	Paquini	30
6	James	Bond	40
7	Matt	Ancona	60

Stored Procedure

isSkinFor

We want to know the skin name passed in is for what champion specifically.

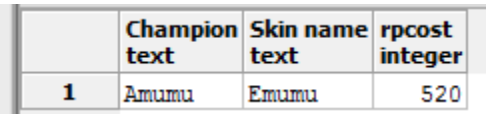
SQL Script:

```
create or replace function isSkinFor (text, refcursor) returns refcursor as
$$
declare
        SkinName text := $1;
        resultset refcursor := $2;
begin
        open resultset for
                select Champions.Name as "Champion", Skins.Name as "Skin name",
Skin.RPCost
                from Champions, Skins, ChampsWithSkins
                where Champions.cid = ChampsWithSkins.cid
                and   Skins.sid = ChampsWithSkins.sid
                and   Skins.Name = SkinName;
```

```
        return resultset;

end;

$$

language plpgsql;
```

Example and how it looks like:

```
select isSkinFor('Emumu','results');

Fetch all from results;
```



	Champion text	Skin name text	rpcost integer
1	Amumu	Emumu	520

HasSkins

We want to know what skins does a champion have.

SQL Script

```
create or replace function isSkinFor (text, refcursor) returns refcursor as

$$

declare

        ChampName text := $1;
```

```
        resultset refcursor := $2;
begin
        open resultset for
                select Champions.Name as "Champion", Skins.Name as "Skin name",
Skins.RPCost
                from Champions, Skins, ChampWithSkins
                where Champions.cid = ChampWithSkins.cid
                and   Skins.sid = ChampWithSkins.sid
                and   Champions.Name = ChampName;
        return resultset;
end;
$$
language plpgsql;
```

Test and see how it looks like

```
select isSkinFor('Akali','results');
Fetch all from results;
```

	Champion text	Skin name text	rpcost integer
1	Akali	Blood Moon Akali	975
2	Akali	Nurse Akali	975

Security

We want to control the select, insert and update privileges for various types of users.

There are three types of users as of now:

Admin-grant access to all privileges

User-grant access to update their account information

Salesman-grant access to insert, update various transactions

SQL Script:

```
CREATE ROLE admin;

GRANT select, insert, update

on all tables in schema public

to admin



CREATE ROLE normal_user

grant update

on Account to normal_user
```

```
CREATE ROLE salesman

grant insert, update

on ChmpTrans,IconTrans,WardTrans,SkinTrans to salesman
```

Known Issues/Future Enhancements

This database will do well in terms of knowing what types of transactions each account had made. The admin will be able to query all the transactions and grant access to different champions and skins according to account's purchases. However, there are a few things to consider:

1. A Skin is only created and released after a champion is released. In the current database system, I can insert a skin into the Skins table without having to worry about if this skin belongs to the champion that has been released or not. A possible solution to that could be adding a trigger in which it asks for a champion ID and searches in the champion table to see if it exists or not, revoked the update if the champion is not found.
2. Although we can query about the total amount of RP and IP a single account have, the database system does not check if the account's IP and RP amount if sufficient before making the purchase. However, since only the salesmen and the admin have the access to insert and update the table, there still could be errors. If such error is made, it will give away champions and skins for free. A solution to that is create another table that just have the sum of the RP purchased and IP earned, and a trigger that will update the table after every insert in any transaction tables.