Summoners List

By: Noah Coley, Cameron Fisher, and Tiffany Peck

TABLE OF CONTENTS

PROJECT OVERVIEW

League of Legends is one the most popular MOBAs in the world, roughly having a player base of 115 million people. Because this game is constantly growing, our application Summoners List was created to display information of every champion to help new and already existing players.

Summoners List allows players to easily search up their favorite champions and get basic information on them. It displays their class, damage types, possible positions that you may play the champion in, and any information that would help a player excel with each champion.

BUSINESS RULES

Champions are identified by a champion ID and the system records each champion's name, damage type, position, class, prices, number of skins and when they were released

Because Champions can have multiple potential lane positions, and roles separate tables are generated and use the champion ID as a foreign key to feed the data back to the system.

Skins are identified by a skin ID, and links to the Champion via the champion ID. The system records the theme, name, tier, various prices, whether or not it can be acquired via gemstones, and whether or not it is unlockable.
Abilities are identified by the champion ID and current records the Q, W, E, R and passive skill names.

DATABASE STRUCTURE[1]

The database consists of five tables. The main tables are Champion and Skin. The other three are mostly extensions of Champion information though are displayed as their own tables. Champion is the major table which has it's primary key used as a foreign key in all other tables.

Database Table Name	Description
Champion	Contains basic information about all champions in the database.
Skin	Contains basic information about the skins of various champions.
Abilities	Contains the names of the champions' various abilities as a separate set that can be modified.
Role	Contains the champions class, or role. This gives a basic idea as to how the champion plays.
Lane	Contains the champions, lanes that they can play in. These lanes are positions that the champion is primarily used in.

EXAMPLE TABLES

Champion						
CID	Name	DamageType	SkinNumber	ReleaseDate	BEPrice	RPPrice
1	Aatrox	AD	18	2013-06-13	4800	880
2	Ahri	AP	14	2011-12-14	3150	790

[1] Values are examples and may not represent the completed database.

Skin

SID	CID	SkinTheme	Name	Tier	OEPrice[2]	RPPrice	Gemstones[3]	Unlockable
1	2	Original	Original Ahri	1	0	790	0	1
9	2	K/DA	K/DA Ahri	2	1050	1350	0	1

Role

CID	RID	Class
21	4	Diver
21	9	Assassin
1	3	Juggernaut
3	9	Assassin

Lane

CID	RID	Position
1	1	Top
1	2	Middle
2	2	Middle
4	5	Support

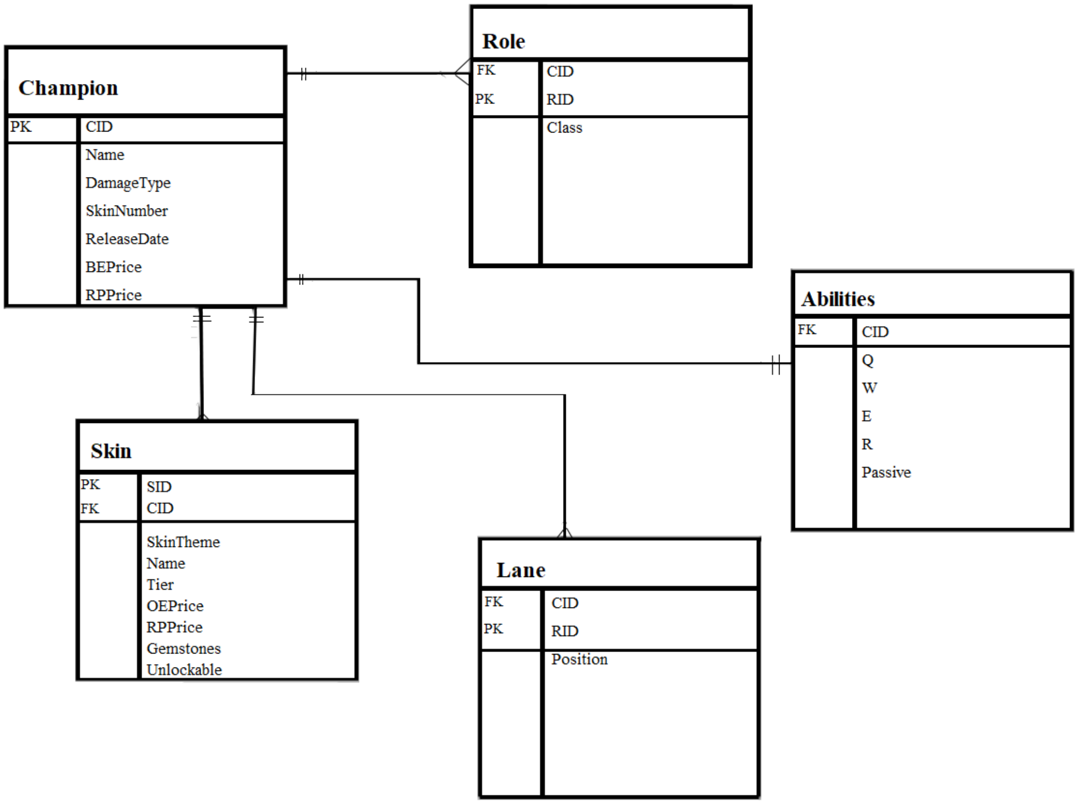
Abilities

CID	Q	W	E	R	Passive
1	“The Darkin Blade”	“Infernal Chains”	“Umbral Dash”	“World Ender”	“Deathbringer Stance”
2	“Orb of Deception”	“Fox-Fire”	“Charm”	“Spirit Rush”	“Essence Theft”
3	“Five Point Strike”	“Twilight Shroud”	“Shuriken Flip”	“Perfect Execution”	“Assassin’s Mark”

[2] 0 represents a null

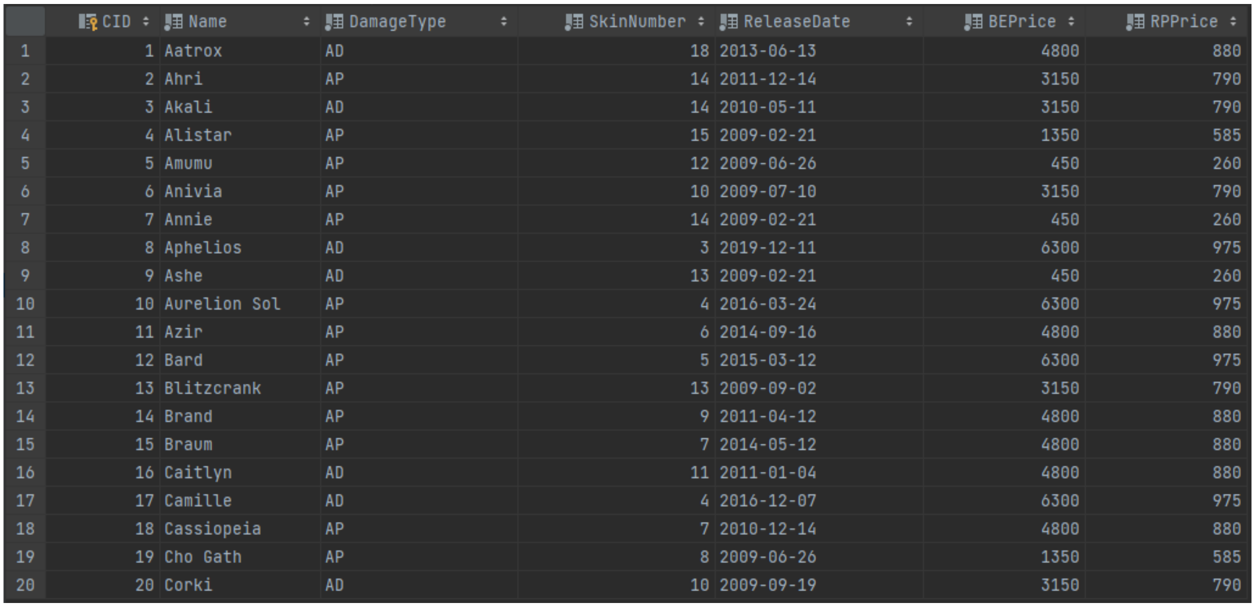
[3] For gemstones and unlockable 0 represents false, 1 represents true. SQLite does not have a boolean data type and thus we use integers to represent it in a basic manner.

ER DIAGRAM


Champion
PK
CID
Name
DamageType
SkinNumber
ReleaseDate
BEPrice
RPPrice
Role
FK
CID
PK
RID
Class
Abilities
FK
CID
Q
W
E
R
Passive
Skin
PK
SID
FK
CID
SkinTheme
Name
Tier
OEPrice
RPPrice
Gemstones
Unlockable
Lane
FK
CID
PK
RID
Position

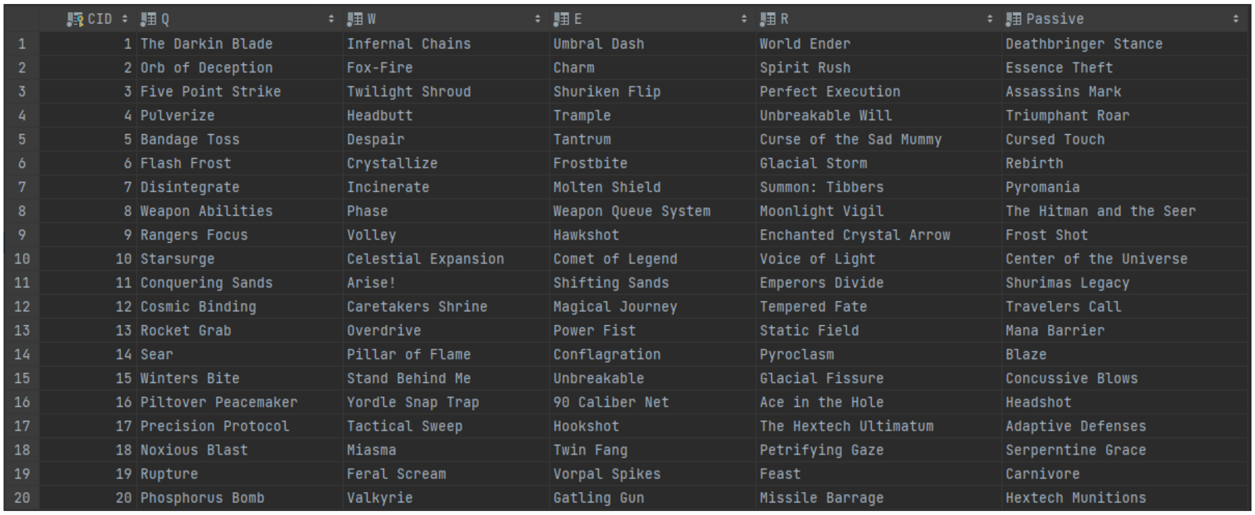
SQL TABLES

Champions



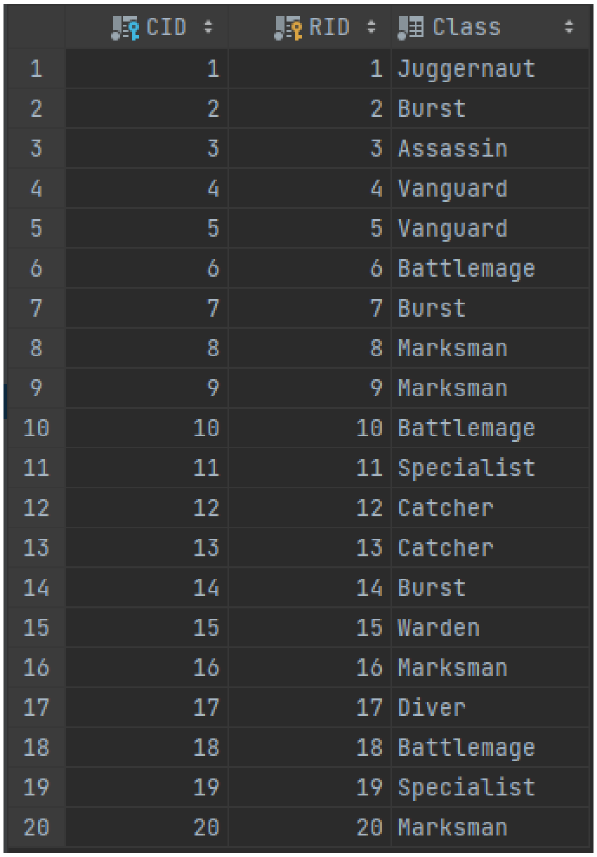
	CID	Name	DamageType	SkinNumber	ReleaseDate	BEPrice	RPPrice
1	1	Aatrox	AD	18	2013-06-13	4800	880
2	2	Ahri	AP	14	2011-12-14	3150	790
3	3	Akali	AD	14	2010-05-11	3150	790
4	4	Alistar	AP	15	2009-02-21	1350	585
5	5	Amumu	AP	12	2009-06-26	450	260
6	6	Anivia	AP	10	2009-07-10	3150	790
7	7	Annie	AP	14	2009-02-21	450	260
8	8	Aphelios	AD	3	2019-12-11	6300	975
9	9	Ashe	AD	13	2009-02-21	450	260
10	10	Aurelion Sol	AP	4	2016-03-24	6300	975
11	11	Azir	AP	6	2014-09-16	4800	880
12	12	Bard	AP	5	2015-03-12	6300	975
13	13	Blitzcrank	AP	13	2009-09-02	3150	790
14	14	Brand	AP	9	2011-04-12	4800	880
15	15	Braum	AP	7	2014-05-12	4800	880
16	16	Caitlyn	AD	11	2011-01-04	4800	880
17	17	Camille	AD	4	2016-12-07	6300	975
18	18	Cassiopeia	AP	7	2010-12-14	4800	880
19	19	Cho Gath	AP	8	2009-06-26	1350	585
20	20	Corki	AD	10	2009-09-19	3150	790

Abilities



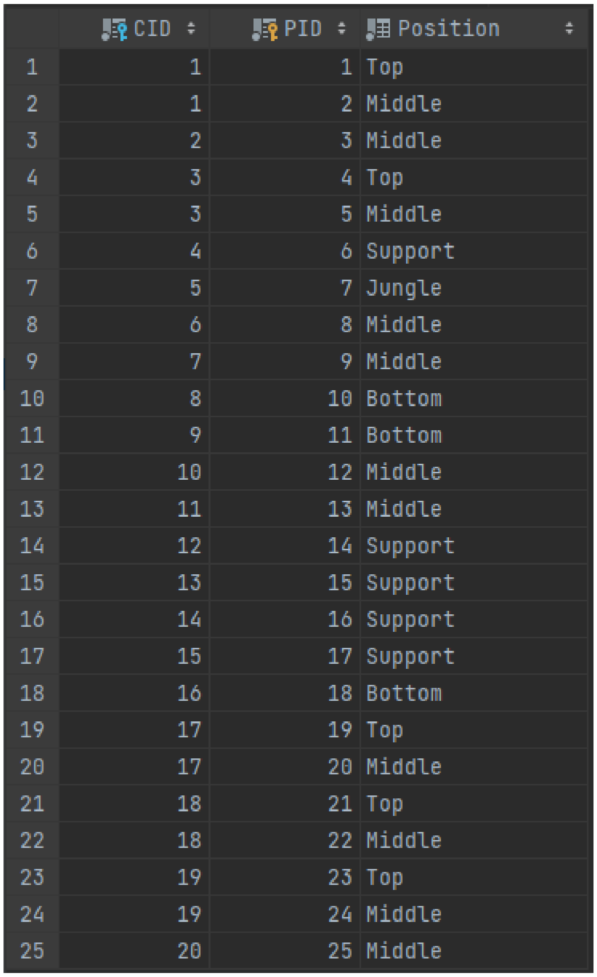
	CID	Q	W	E	R	Passive
1	1	The Darkin Blade	Infernal Chains	Umbral Dash	World Ender	Deathbringer Stance
2	2	Orb of Deception	Fox-Fire	Charm	Spirit Rush	Essence Theft
3	3	Five Point Strike	Twilight Shroud	Shuriken Flip	Perfect Execution	Assassins Mark
4	4	Pulverize	Headbutt	Trample	Unbreakable Will	Triumphant Roar
5	5	Bandage Toss	Despair	Tantrum	Curse of the Sad Mummy	Cursed Touch
6	6	Flash Frost	Crystallize	Frostbite	Glacial Storm	Rebirth
7	7	Disintegrate	Incinerate	Molten Shield	Summon: Tibbers	Pyromania
8	8	Weapon Abilities	Phase	Weapon Queue System	Moonlight Vigil	The Hitman and the Seer
9	9	Rangers Focus	Volley	Hawkshot	Enchanted Crystal Arrow	Frost Shot
10	10	Starsurge	Celestial Expansion	Comet of Legend	Voice of Light	Center of the Universe
11	11	Conquering Sands	Arise!	Shifting Sands	Emperors Divide	Shurimas Legacy
12	12	Cosmic Binding	Caretakers Shrine	Magical Journey	Tempered Fate	Travelers Call
13	13	Rocket Grab	Overdrive	Power Fist	Static Field	Mana Barrier
14	14	Sear	Pillar of Flame	Conflagration	Pyroclasm	Blaze
15	15	Winters Bite	Stand Behind Me	Unbreakable	Glacial Fissure	Concussive Blows
16	16	Piltover Peacemaker	Yordle Snap Trap	90 Caliber Net	Ace in the Hole	Headshot
17	17	Precision Protocol	Tactical Sweep	Hookshot	The Hextech Ultimatum	Adaptive Defenses
18	18	Noxious Blast	Miasma	Twin Fang	Petrifying Gaze	Serperntine Grace
19	19	Rupture	Feral Scream	Vorpal Spikes	Feast	Carnivore
20	20	Phosphorus Bomb	Valkyrie	Gatling Gun	Missile Barrage	Hextech Munitions

Role



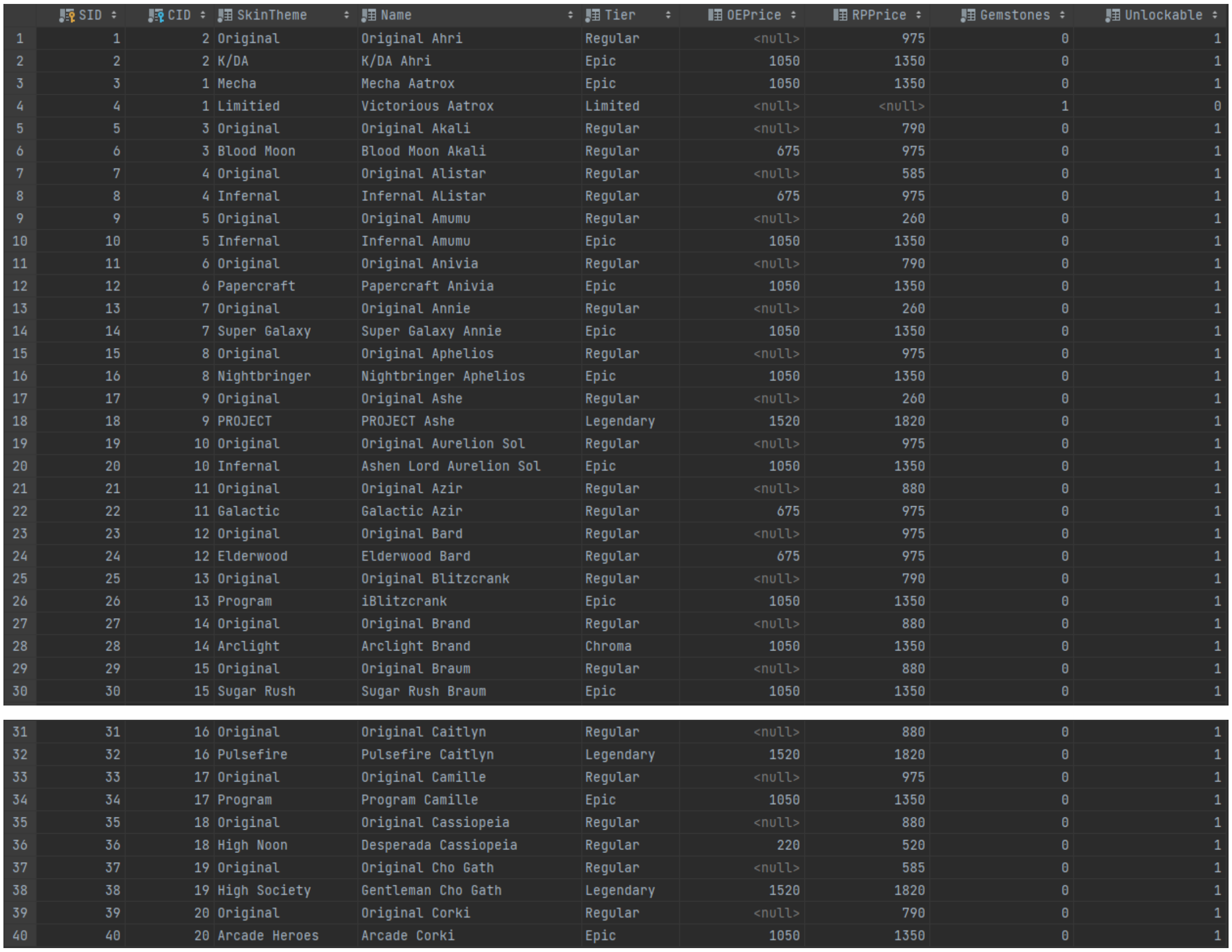
	CID	RID	Class
1	1	1	Juggernaut
2	2	2	Burst
3	3	3	Assassin
4	4	4	Vanguard
5	5	5	Vanguard
6	6	6	Battlemage
7	7	7	Burst
8	8	8	Marksman
9	9	9	Marksman
10	10	10	Battlemage
11	11	11	Specialist
12	12	12	Catcher
13	13	13	Catcher
14	14	14	Burst
15	15	15	Warden
16	16	16	Marksman
17	17	17	Diver
18	18	18	Battlemage
19	19	19	Specialist
20	20	20	Marksman

Lane



	CID	PID	Position
1	1	1	Top
2	1	2	Middle
3	2	3	Middle
4	3	4	Top
5	3	5	Middle
6	4	6	Support
7	5	7	Jungle
8	6	8	Middle
9	7	9	Middle
10	8	10	Bottom
11	9	11	Bottom
12	10	12	Middle
13	11	13	Middle
14	12	14	Support
15	13	15	Support
16	14	16	Support
17	15	17	Support
18	16	18	Bottom
19	17	19	Top
20	17	20	Middle
21	18	21	Top
22	18	22	Middle
23	19	23	Top
24	19	24	Middle
25	20	25	Middle

Skins



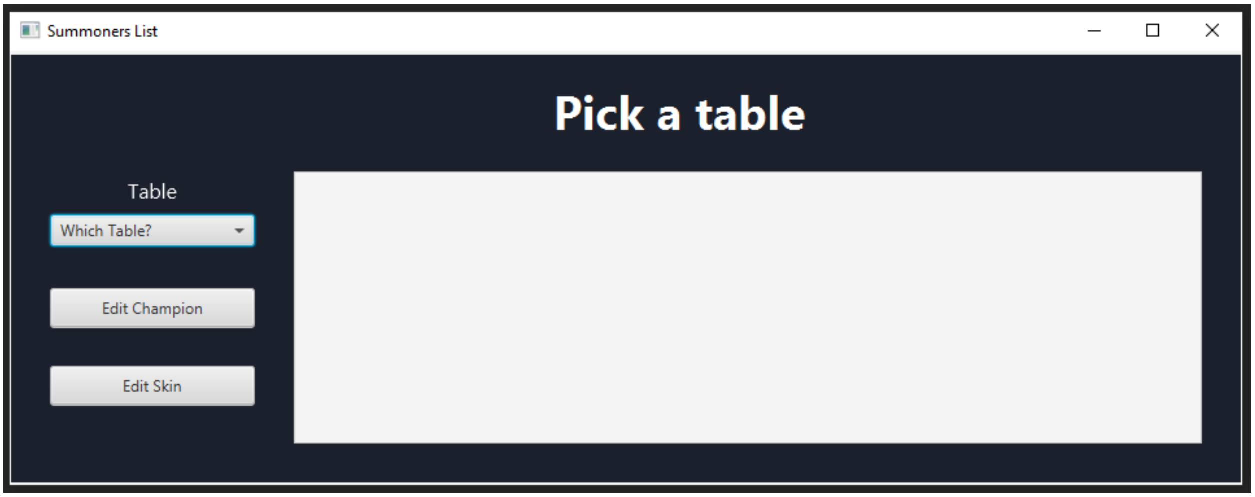
	SID	CID	SkinTheme	Name	Tier	OEPrice	RPPrice	Gemstones	Unlockable
1	1	2	Original	Original Ahri	Regular	<null>	975	0	1
2	2	2	K/DA	K/DA Ahri	Epic	1050	1350	0	1
3	3	1	Mecha	Mecha Aatrox	Epic	1050	1350	0	1
4	4	1	Limitied	Victorious Aatrox	Limited	<null>	<null>	1	0
5	5	3	Original	Original Akali	Regular	<null>	790	0	1
6	6	3	Blood Moon	Blood Moon Akali	Regular	675	975	0	1
7	7	4	Original	Original Alistar	Regular	<null>	585	0	1
8	8	4	Infernal	Infernal AListar	Regular	675	975	0	1
9	9	5	Original	Original Amumu	Regular	<null>	260	0	1
10	10	5	Infernal	Infernal Amumu	Epic	1050	1350	0	1
11	11	6	Original	Original Anivia	Regular	<null>	790	0	1
12	12	6	Papercraft	Papercraft Anivia	Epic	1050	1350	0	1
13	13	7	Original	Original Annie	Regular	<null>	260	0	1
14	14	7	Super Galaxy	Super Galaxy Annie	Epic	1050	1350	0	1
15	15	8	Original	Original Aphelios	Regular	<null>	975	0	1
16	16	8	Nightbringer	Nightbringer Aphelios	Epic	1050	1350	0	1
17	17	9	Original	Original Ashe	Regular	<null>	260	0	1
18	18	9	PROJECT	PROJECT Ashe	Legendary	1520	1820	0	1
19	19	10	Original	Original Aurelion Sol	Regular	<null>	975	0	1
20	20	10	Infernal	Ashen Lord Aurelion Sol	Epic	1050	1350	0	1
21	21	11	Original	Original Azir	Regular	<null>	880	0	1
22	22	11	Galactic	Galactic Azir	Regular	675	975	0	1
23	23	12	Original	Original Bard	Regular	<null>	975	0	1
24	24	12	Elderwood	Elderwood Bard	Regular	675	975	0	1
25	25	13	Original	Original Blitzcrank	Regular	<null>	790	0	1
26	26	13	Program	iBlitzcrank	Epic	1050	1350	0	1
27	27	14	Original	Original Brand	Regular	<null>	880	0	1
28	28	14	Arclight	Arclight Brand	Chroma	1050	1350	0	1
29	29	15	Original	Original Braum	Regular	<null>	880	0	1
30	30	15	Sugar Rush	Sugar Rush Braum	Epic	1050	1350	0	1
31	31	16	Original	Original Caitlyn	Regular	<null>	880	0	1
32	32	16	Pulsefire	Pulsefire Caitlyn	Legendary	1520	1820	0	1
33	33	17	Original	Original Camille	Regular	<null>	975	0	1
34	34	17	Program	Program Camille	Epic	1050	1350	0	1
35	35	18	Original	Original Cassiopeia	Regular	<null>	880	0	1
36	36	18	High Noon	Desperada Cassiopeia	Regular	220	520	0	1
37	37	19	Original	Original Cho Gath	Regular	<null>	585	0	1
38	38	19	High Society	Gentleman Cho Gath	Legendary	1520	1820	0	1
39	39	20	Original	Original Corki	Regular	<null>	790	0	1
40	40	20	Arcade Heroes	Arcade Corki	Epic	1050	1350	0	1

10

APPLICATION

The main application is a window that shows the table of your choice, the table selector dropdown box and the edit buttons. To sort any of the columns in ascending or descending order click the

The Main Table images:



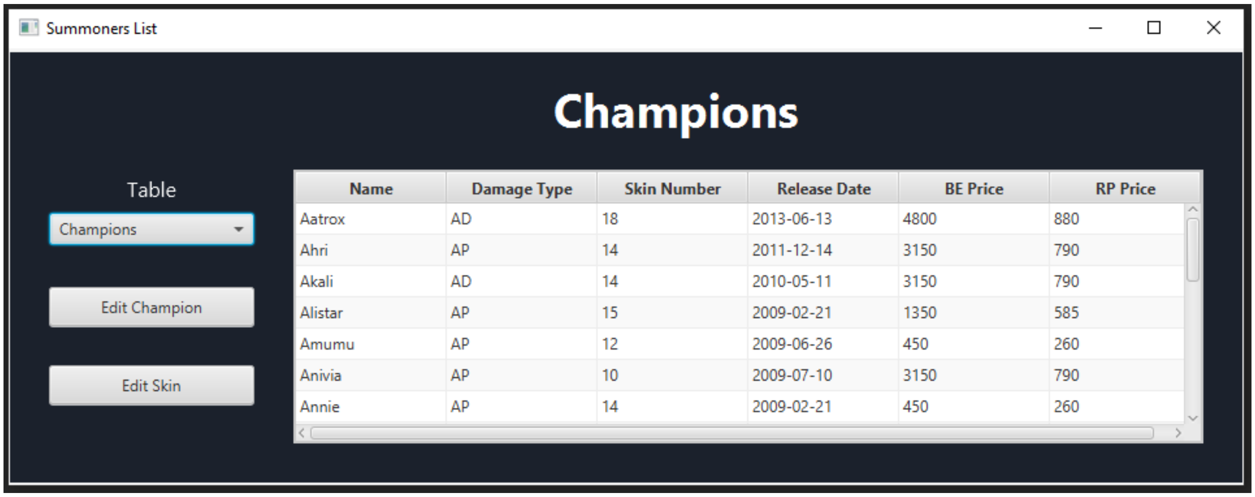
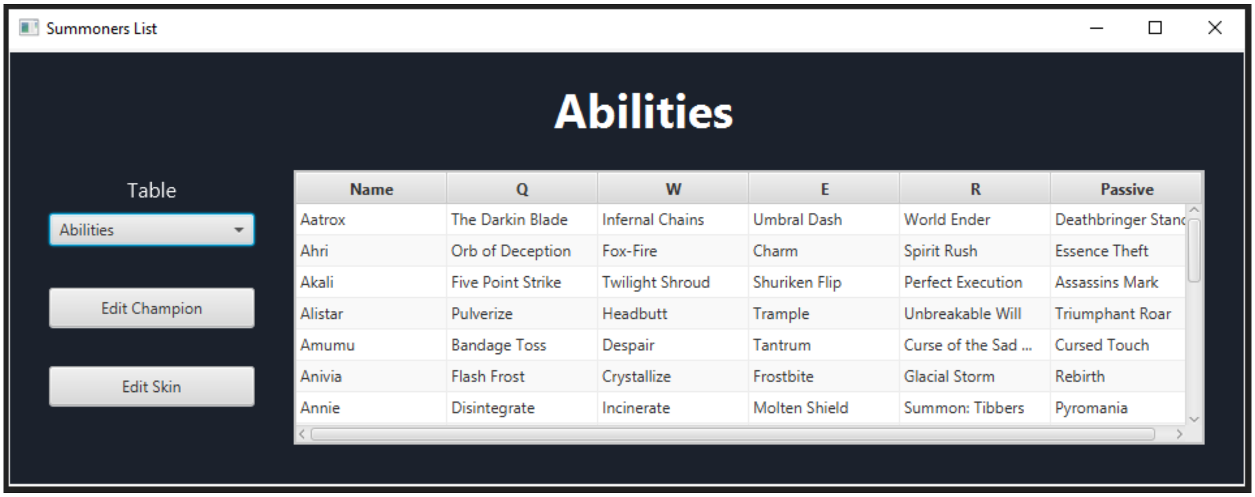


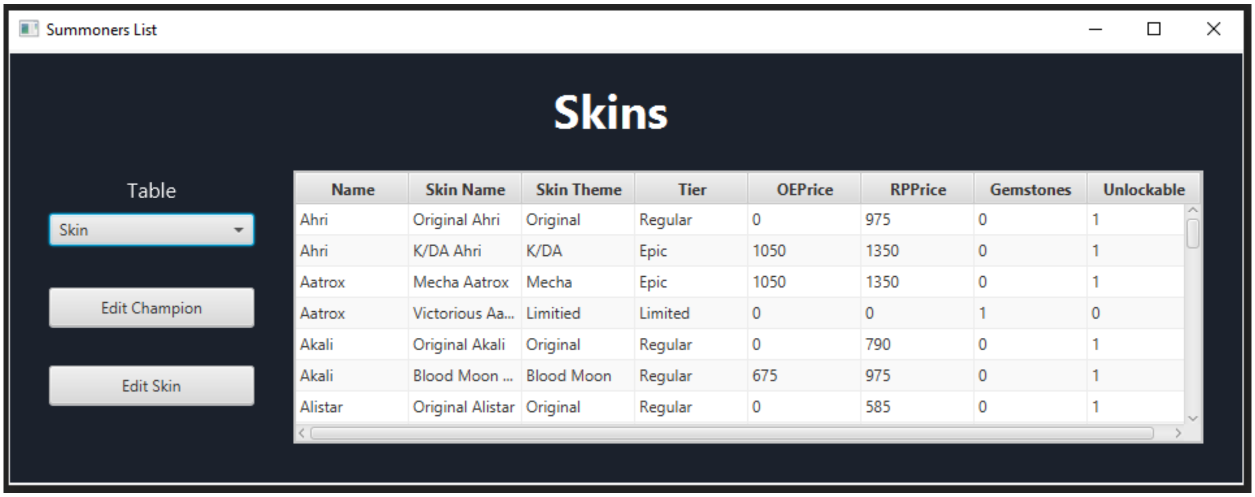
Table: Champions

Name	Damage Type	Skin Number	Release Date	BE Price	RP Price
Aatrox	AD	18	2013-06-13	4800	880
Ahri	AP	14	2011-12-14	3150	790
Akali	AD	14	2010-05-11	3150	790
Alistar	AP	15	2009-02-21	1350	585
Amumu	AP	12	2009-06-26	450	260
Anivia	AP	10	2009-07-10	3150	790
Annie	AP	14	2009-02-21	450	260



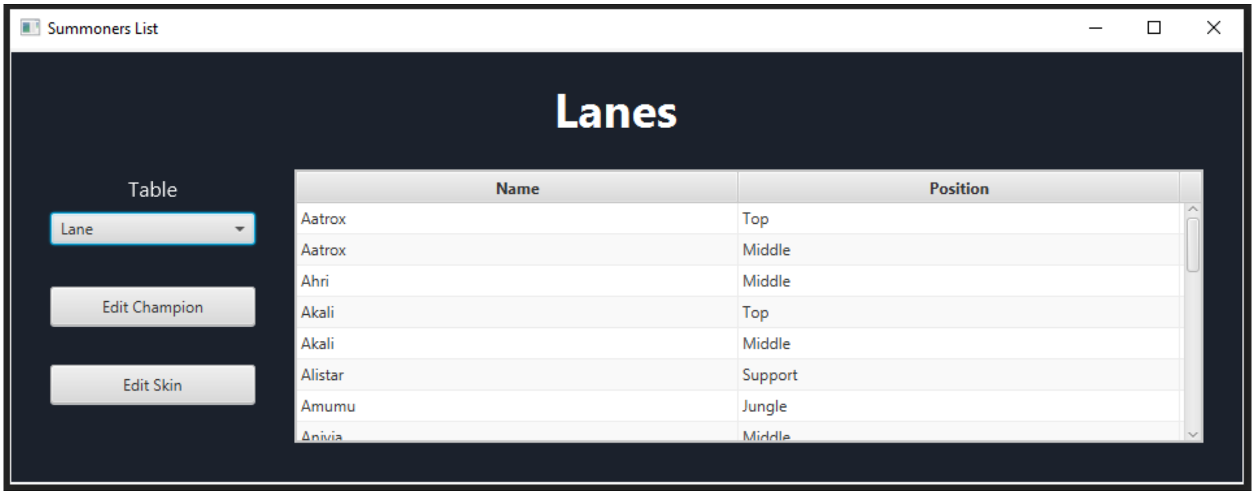
Abilities

Name	Q	W	E	R	Passive
Aatrox	The Darkin Blade	Infernal Chains	Umbral Dash	World Ender	Deathbringer Stanc
Ahri	Orb of Deception	Fox-Fire	Charm	Spirit Rush	Essence Theft
Akali	Five Point Strike	Twilight Shroud	Shuriken Flip	Perfect Execution	Assassins Mark
Alistar	Pulverize	Headbutt	Trample	Unbreakable Will	Triumphant Roar
Amumu	Bandage Toss	Despair	Tantrum	Curse of the Sad ...	Cursed Touch
Anivia	Flash Frost	Crystallize	Frostbite	Glacial Storm	Rebirth
Annie	Disintegrate	Incinerate	Molten Shield	Summon: Tibbers	Pyromania



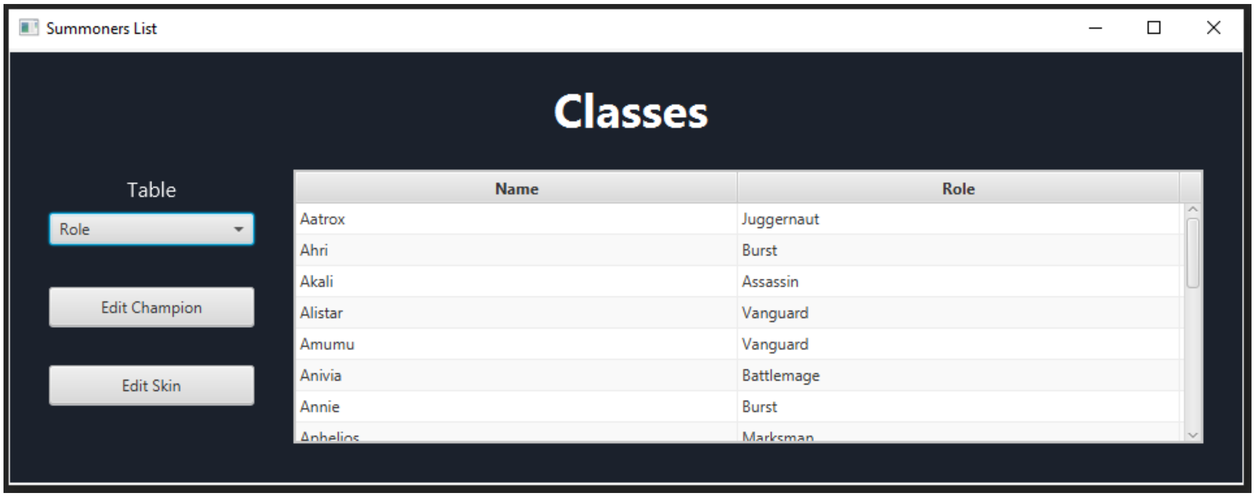
Skins

Name	Skin Name	Skin Theme	Tier	OEPrice	RPPrice	Gemstones	Unlockable
Ahri	Original Ahri	Original	Regular	0	975	0	1
Ahri	K/DA Ahri	K/DA	Epic	1050	1350	0	1
Aatrox	Mecha Aatrox	Mecha	Epic	1050	1350	0	1
Aatrox	Victorious Aa...	Limitied	Limited	0	0	1	0
Akali	Original Akali	Original	Regular	0	790	0	1
Akali	Blood Moon ...	Blood Moon	Regular	675	975	0	1
Alistar	Original Alistar	Original	Regular	0	585	0	1



Lanes

Name	Position
Aatrox	Top
Aatrox	Middle
Ahri	Middle
Akali	Top
Akali	Middle
Alistar	Support
Amumu	Jungle
Anivia	Middle



Summoners List

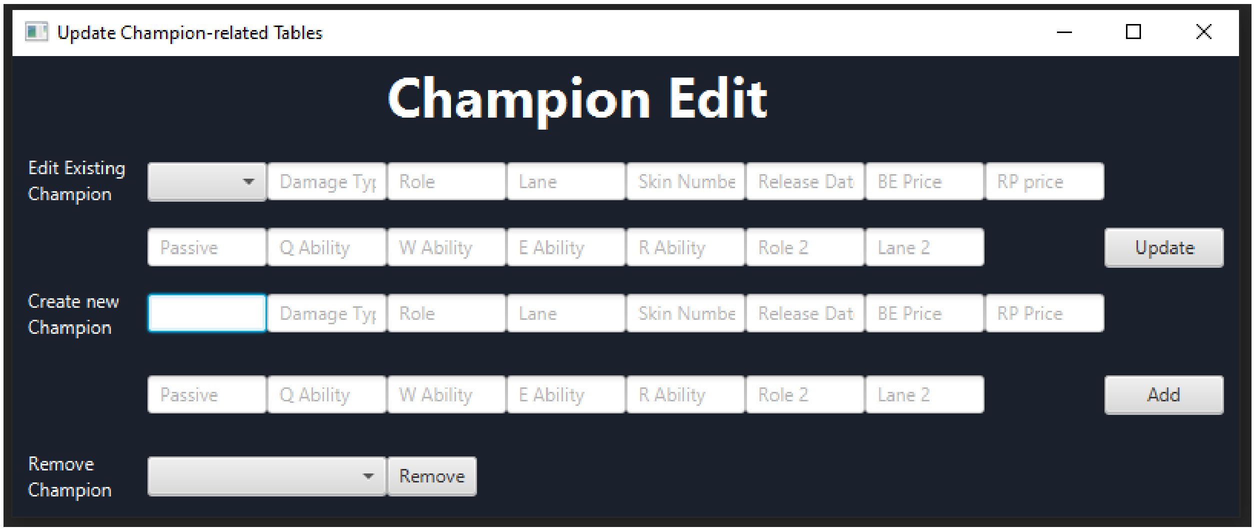
Classes

Table: Role

Name	Role
Aatrox	Juggernaut
Ahri	Burst
Akali	Assassin
Alistar	Vanguard
Amumu	Vanguard
Anivia	Battlemage
Annie	Burst
Aphelios	Marksman

Keep in mind to refresh the tables after updating, you need to select the table again from the drop down menu.

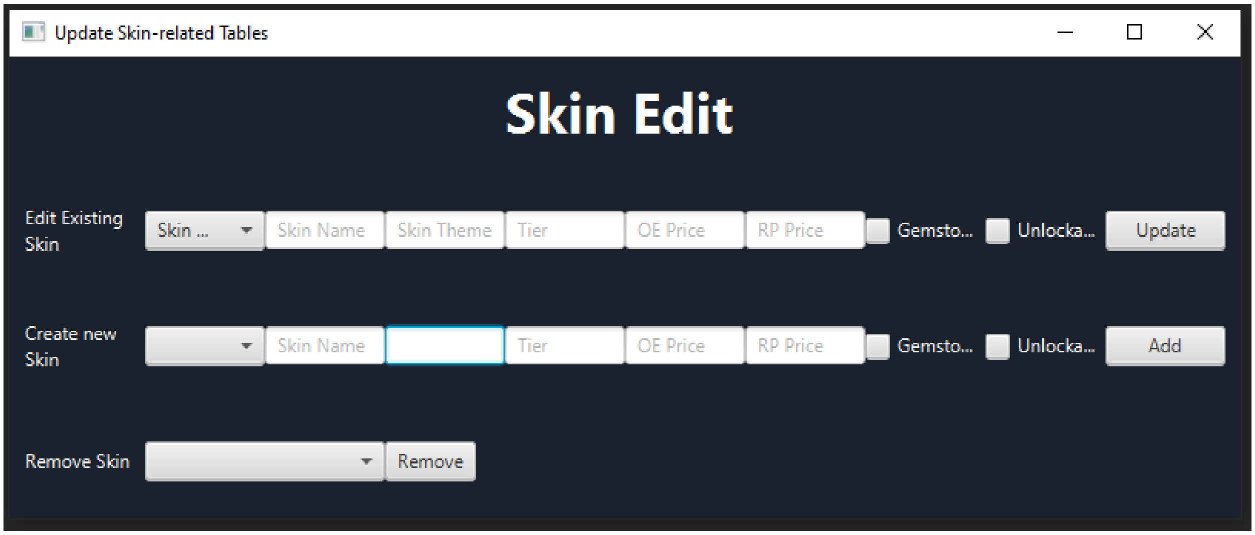
The Champion Edit Window:



Update Champion-related Tables

Champion Edit

The Edit Skin Window:



Descriptive List of Source Code

1. Main: Creates the main UI using the javafx "sample.fxml" file.
2. *Controller*: Creates the handler for the edit champion and edit skin buttons, and produces all of the tables. Controller for *"sample.fxml".*
    a. champHandler() - Upon pressing edit champion, displays the champion edit window using the **display** function.
    b. skinHandler() - Upon pressing edit skin, displays the skin edit window using the **display** function.
    c. initialize() - adds the tables to the combo box.
    d. pickTable() - Creates the tables for Champions, Abilities, Skins, Lanes, and Roles then assigns the scroll box's value depending on which table is selected using **getChampionList**, **getAbilityList**, **getSkinList**, **getLaneList** and **getRoleLis**t from the *ChampDataAccessor* respectively.
3. *ChampDataAccessor:*[4] Handles all sql use. Adding, updating, removing from the database or sending data to the application.
    a. getChampionObvList() - Returns array list of champion names for use in the combo boxes within the *ChampionTable* display.
    b. getSkinArrayList() - Returns an array list of skin names for use in the combo boxes within the *SkinTable* display.
    c. get(blank)List() - Selects specified list from sql code.
    d. add(blank)() - adds entities into a table within the database file.
    e. rm(blank)() - removes entities from the table within the database file.
    f. upd(blank)() - updates entities in a table within the database file.

[4] Generalized some methods into groups because they roughly do the same thing and there are a lot of methods.

4. *ChampionTable*: Uses text fields, combo boxes and buttons to update, add or remove champion entities from the champions table. Controller for *"EditC.fxml"*.
    a. initialize() - adds the champions to the combo boxes.
    b. updateAction() - Upon pressing update takes the values from the related combo box, text fields, and drop down and sends them to the. **updChampion, updRole, updLane, updAbility** method within *ChampDataAccessor* to update the champion, role, lane, and abilities tables.
    c. addAction() - Upon pressing Add takes the values from the text fields and sends them to the **addChampion, addRole, addLane, addAbility** method within *ChampDataAccessor* to add a champion, role, lane, and ability entity to the champions, role, lane, abilities tables respectively.
    d. removeAction() - Upon pressing Remove takes the value from the related combo box and sends the data to the **rmChampion** method within *ChampDataAccessor* to remove the champion from the table. Because the ON DELETE CASCADE refused to work in IntelliJ despite working find in a terminal environment, we brute forced a cascade delete within **rmChampion**.
5. *SkinTable*: Uses text fields, combo boxes, check boxes and buttons to update, add or remove skin entities from the skin table. Controller for *"EditS.fxml".*
    a. initialize() - adds the skins to the combo boxes
    b. updateAction() - Upon pressing update takes the values from the related combo box, text fields, check boxes, and drop down and sends them to the **updSkin** method within *ChampDataAccessor* to update the skin table.
    c. addAction() - Upon pressing Add takes the values from the text fields and check boxes and sends them to the **addSkin** method within *ChampDataAccessor* to add a skin to the skin table.
    d. removeAction() - Upon pressing Remove takes the value from the related combo box and sends the data to the **rmSkin** method within *ChampDataAccessor* to remove the skin from the table.
6. *Champion*: Takes the name of the champion, their damage type, release date, number of skins, BE price and RP price and loads them onto the display table in the GUI with the various gets and sets.
7. *Skin*: Takes the name of the champion, the name of their skin, the skin's theme, its tier, OE price, RP price, if you need gemstones and if it is unlockable and loads them onto the display table in the GUI with the various gets and sets.
8. *Abilities*: Takes the name of the champion and their ability names (e.g., q) to load onto the display table in the GUI with the various gets and sets.
9. *Class*: Takes the name of the champion and role to load onto the display table in the GUI with the various gets and sets.
10. *Lane*: Takes the name of the champion and position to load onto the display table in the GUI with the various gets and sets.
11. *sample.fxml:* The JavaFX file connected to *Controller.* Contains the main Table view and buttons to access the other windows.

12. *EditC.fxml:* The JavaFX file connected to *ChampionTable.* Contains the information for displaying the window for editing champions.
13. *EditS.fxml:* The JavaFX file connected to *SkinTable*. Contains the information for displaying the window for editing skins.