

Name:

Mark:

Database Assignment

Purpose and End User of my database

For people that are interested in league of legends to learn about the champions, their play style, difficulty, and where they are from.
The purpose is to encourage people to learn about league of legends and get them interested in the game.

Describe at least 3 implications that are relevant to your database and its use by the end user and why they are important

Social - Is about how it positively impacts the end users, this could be by making them more outgoing or willing to do activities, for example advertisements with famous athletes make young children aspire to be like said athlete and go play sport while buying the product advertised like a basketball. This is important to me as the whole point of my project is to get people invested in League of Legends, and I can do this by making it easy for them to learn. Social is also about if it negatively affects the user, like being offensive. An example of this are the advertisements in the 90's of Coke and Pepsi competing against each other, this could be offensive as they would portray users of the other company's product as lesser, this could end up making those users that see that ad never purchase from their company. This affects my project as if someone doesn't like what I'm doing, they'll most likely not ever want to play League of Legends.

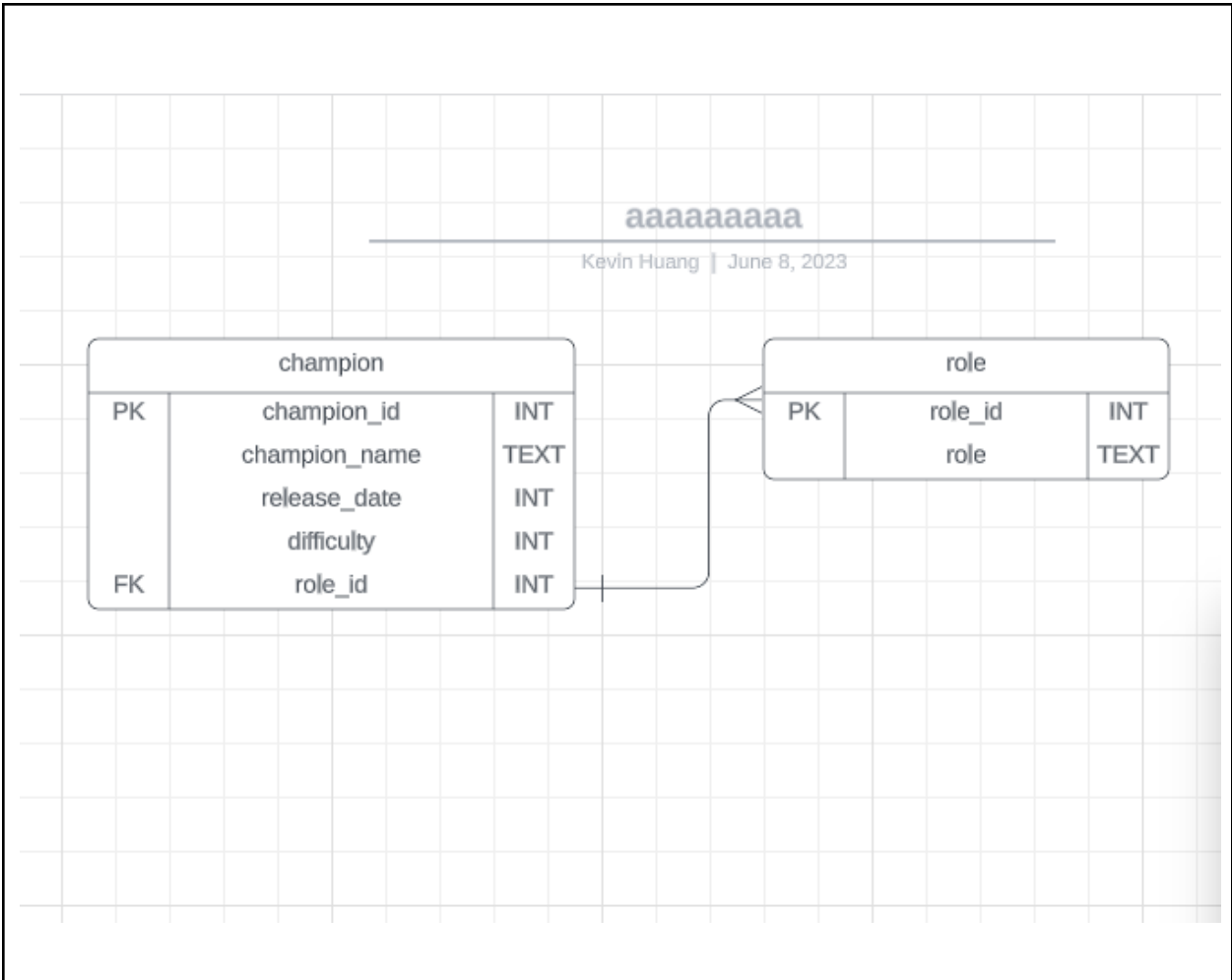
Usability - Usability is about how easy it is to use. This can be by making it easier for the user to understand and use it. The more cohesive it is. For example things with high usability would be websites with a user interface. The usability of a product is related to how well its error detection is. This is important because if the usability is very bad the users will not be interested in League of Legends.

Functionality - Functionality is about how well the product works and if it works as intended. The functionality shows how well and capable your product which can attract customers e.g the quality of decades old cars still being well are why Japanese car brands like Toyota are popular. If it works as intended it means it is of high quality. This is important as if my project is not working people will get disappointed at how abysmal my project is and never experience the grace that is League of Legends.

Name:

Mark:

Database Design- Your Entity Relationship Diagram.



Database Testing Table: SQL Statements

Purpose	SQL Statement	Result Success?
See all the champions alphabetically descending	<pre>SELECT champion_name FROM champion ORDER BY champion_name ASC;</pre>	Yes

Name:

Mark:

See all existing roles	<pre>SELECT roles FROM role</pre>	Yes
See all champions by release date earliest to oldest	<pre>SELECT * FROM champion ORDER BY release_date DESC;</pre>	Yes
See all champions by release date oldest to earliest	<pre>SELECT * FROM champion ORDER BY release_date ASC;</pre>	Yes
Random champion	<pre>SELECT champion_name FROM champion ORDER BY RANDOM() LIMIT 1;</pre>	Yes
Random champion from a selected role	<pre>SELECT champion_id, champion_name, release_date, difficulty FROM champion WHERE role_id = ? ORDER BY RANDOM() LIMIT 1</pre>	Yes
Print champions from a role	<pre>SELECT * FROM champion WHERE role_id = ?</pre>	Yes
See all champions by difficulty easiest to hardest	<pre>SELECT * FROM champion ORDER BY difficulty ASC;</pre>	Yes
See all champions by difficulty from hardest to easiest	<pre>SELECT * FROM champion ORDER BY difficulty DESC;</pre>	Yes
See all champions alphabetically reverse	<pre>SELECT champion_name FROM champion ORDER BY champion_name DESC;</pre>	Yes

Name:

Mark:

Relevant Implications- Explain how your database addresses the relevant implications that you identified at the start.

Functionality - My data shows high functionality by working extremely well. It can print it out alphabetically, all the champions in a selected role, a random champion, and a random champion from a selected role. It is specific but not too specific where it becomes redundant. This is because I have a simple submenu that categorises everything into smaller bite sized pieces for the user to understand, instead of just asking the user to choose which function they want out of the many that I have. Not only this but I have chosen where you'd want to see specific information, as if you were to see all 154 champions and all their information, it would overload your brain, so I have chosen to only give extra information in some places like in the functions where you select a random champion or selecting champions from one of the 13 roles, as it is easier to understand the information in bite sized pieces. I have also got wonderful error detection as you can type something like "drkhldufihgdf78ygt78e354" in it and you will not be able to encounter any errors. This is because I have coded it to not break as I know some people can be clumsy when they type.

Social - My product has created a positive social impact, it encourages learning of a hobby which can be done with friends. My product lets newer players of League of Legends have more fun as they understand the game easier with my product. The random champion and random champion from a role easily can be used by newer players to find a champion they'd want to play, as it tells them how hard it is and the role.

Usability - My product is wonderful for people to use, this is because the functions to choose a random champion and a random champion from a role that the user inputted can be wonderful for new players. This is because using both of these will show you the champion's difficulty and the ability to select a role can let players choose their play style. This role is descriptive of the playstyle of the champion e.g assassin, juggernaut. The difficulty lets new players gauge their mechanical skills as if they want to relax and have a nice relaxing champion to play; they can choose the ones with a lower difficulty.

Legal - Legal is about if someone breaks the law in some way for example copying someone's intellectual property and claiming it's yours, or using it as an advertisement or a way to sell products. This is important because it limits my ability to advertise the wonders of League of Legends and will have consequences.

Name:

Mark:

Showcase:

Give evidence of your database and the Python code that interfaces with it. Use screenshots or a short video. Explain how it improved, how it functions, how it was tested etc.

I had improved my database by making it a submenu, instead of having one input with a large hunk of text I categorised my functions in the submenu. The point of the submenu is to make it easy for the users, also having to type numbers further up is annoying on keyboards with no numberpad as your right hand is on the mouse. The submenu can be seen below.

```
while True:
    try:
        #main menu
        user_input = int(input("\nWhat do you want to do?\n1. See champions\n2. See all roles\n3. Exit\n"))

        # Submenu for seeing champions
        if user_input == 1:
            champion_input = input("\nWhat do you want to do?\n\n1. See all champions\n2. See a random champion\n3. Go back\n")

            # Submenu for seeing all champions in different ways
            if champion_input == "1":
                champion_all_input = input("\n\n1. See all champions alphabetically\n2. See all champions in reverse alphabetical order\n3. See all champions in one role\n4. Exit\n")
```

This is the main menu and the submenu for the category of seeing all champions.

```
# All functions for champion input 1
# See all champions alphabetically
if champion_all_input == "1":
    print_champions_asc()
    break

# See all champions in reverse alphabetical order
elif champion_all_input == "2":
    print_champions_desc()
    break

# Seeing every champion of one role
elif champion_all_input == "3":
    show_roles()
    while True:
        try:
            champion_by_role_input = int(input("Which role do you want to see every champion of?\n"))
            if champion_by_role_input in range(1, 14):
                print_champions_by_role(champion_by_role_input)
                break
            else:
                print("Invalid role ID. Please enter a valid role ID.")
        except ValueError:
            print("Invalid input. Please enter a valid role ID as a number.")
    break

# Exit
elif champion_all_input == "4":
    print("Exiting, Have a good day!")
    break
else:
    print("Invalid input. Please enter a valid option (1, 2, 3, or 4).")

# random champion submenu, champion input 2
elif champion_input == "2":
    random_champion_input = input("\n\n")
```

This is the code for the submenu of seeing all champions, I added error detection, this is because it'll improve my product, by making it not break.

Name:

Mark:

```
205         # random champion submenu, champion input 2
206         elif champion_input == "2":
207             random_champion_input = input("\n
208 1. See a random champion\n\
209 2. See a random champion from a selected role\n\
210 3. Exit\n")
211
212         # print random champion
213         if random_champion_input == "1":
214             print_random_champion()
215             break
216         #random champion from a role
217         elif random_champion_input == "2":
218             show_roles()
219             while True:
220                 try:
221                     random_champion_role_input = int(input("Which role do you want to select a random champion from?\n"))
222                     if random_champion_role_input in range(1, 14):
223                         print_random_champion_from_role(random_champion_role_input)
224                         break
225                     else:
226                         print("Invalid role ID. Please enter a valid role ID.")
227                 except ValueError:
228                     print("Invalid input. Please enter a valid role ID as a number.")
229                 break
230         elif random_champion_input == "3":
231             print("Have a nice day!")
232             break
233         else:
234             print("Invalid input. Please enter a valid option (1, 2, or 3).")
235
```

This is the second submenu, this is for the random selection. It also has error detection.

```
229             break
230         elif random_champion_input == "3":
231             print("Have a nice day!")
232             break
233         else:
234             print("Invalid input. Please enter a valid option (1, 2, or 3).")
235
236         # Show roles
237         elif user_input == 2:
238             show_roles()
239             break
240         # Exit
241         elif user_input == 3:
242             print("Have a nice day!")
243             break
244         else:
245             print("Invalid input. Please enter a valid option (1, 2, or 3).")
246             continue
247         #when you type something and are clumsy
248         # error detection
249         except ValueError:
250             print("Invalid input. Please enter a valid option as a number.")
251             continue
252
```

More error detection and my kindness when exiting.

```
What do you want to do?
1. See champions
2. See all roles
3. Exit
dtfhfghdxdhdfihguidfshgdsiuohgdxfg
Invalid input. Please enter a valid option as a number.
```

Error detection

Name:

Mark:

```
5
Here are all the available Roles:
Role ID: 1, Role Name: Vanguard
Role ID: 2, Role Name: Juggernaut
Role ID: 3, Role Name: Burst
Role ID: 4, Role Name: Enchanter
Role ID: 5, Role Name: Artillery
Role ID: 6, Role Name: Skirmisher
Role ID: 7, Role Name: Diver
Role ID: 8, Role Name: Warden
Role ID: 9, Role Name: Catcher
Role ID: 10, Role Name: Battlemage
Role ID: 11, Role Name: Specialist
Role ID: 12, Role Name: Marksman
Role ID: 13, Role Name: Assassin
Which role do you want to see every champion of?
16
Invalid role ID. Please enter a valid role ID.
```

```
What do you want to do?
1. See all champions
2. See a random champion
3. Go back
1

1. See all champions alphabetically
2. See all champions in reverse alphabetical order
3. See all champions in one role
4. Exit
3
Here are all the available Roles:
Role ID: 1, Role Name: Vanguard
Role ID: 2, Role Name: Juggernaut
Role ID: 3, Role Name: Burst
Role ID: 4, Role Name: Enchanter
Role ID: 5, Role Name: Artillery
Role ID: 6, Role Name: Skirmisher
Role ID: 7, Role Name: Diver
Role ID: 8, Role Name: Warden
Role ID: 9, Role Name: Catcher
Role ID: 10, Role Name: Battlemage
Role ID: 11, Role Name: Specialist
Role ID: 12, Role Name: Marksman
Role ID: 13, Role Name: Assassin
Which role do you want to see every champion of?
tfdhfgdfg
Invalid input. Please enter a valid role ID as a number.
```

Name:

Mark:

Error detection

```
What do you want to do?
1. See champions
2. See all roles
3. Exit
1

What do you want to do?
1. See all champions
2. See a random champion
3. Go back
1

1. See all champions alphabetically
2. See all champions in reverse alphabetical order
3. See all champions in one role
4. Exit
fghfghg
Invalid input. Please enter a valid option (1, 2, 3, or 4).
```

Error detection

```
What do you want to do?
1. See champions
2. See all roles
3. Exit
2
Here are all the available Roles:
Role ID: 1, Role Name: Vanguard
Role ID: 2, Role Name: Juggernaut
Role ID: 3, Role Name: Burst
Role ID: 4, Role Name: Enchanter
Role ID: 5, Role Name: Artillery
Role ID: 6, Role Name: Skirmisher
Role ID: 7, Role Name: Diver
Role ID: 8, Role Name: Warden
Role ID: 9, Role Name: Catcher
Role ID: 10, Role Name: Battlemage
Role ID: 11, Role Name: Specialist
Role ID: 12, Role Name: Marksman
Role ID: 13, Role Name: Assassin
PS C:\Users\GGPC\OneDrive\Desktop\school> |
```


Name:

Mark:

```
Here are all the available Roles:
Role ID: 1, Role Name: Vanguard
Role ID: 2, Role Name: Juggernaut
Role ID: 3, Role Name: Burst
Role ID: 4, Role Name: Enchanter
Role ID: 5, Role Name: Artillery
Role ID: 6, Role Name: Skirmisher
Role ID: 7, Role Name: Diver
Role ID: 8, Role Name: Warden
Role ID: 9, Role Name: Catcher
Role ID: 10, Role Name: Battlemage
Role ID: 11, Role Name: Specialist
Role ID: 12, Role Name: Marksman
Role ID: 13, Role Name: Assassin
Which role do you want to select a random champion from?
234h534uih
Invalid input. Please enter a valid role ID as a number.
```

Error detection

The way I did error detection is by smashing my head into my keyboard instead of putting the right inputs. It functions using a submenu system in which you only need to type the numbers that are from minimum one and maximum four. This means it's easy for people to use. I improved it over time by making more submenus and adding error detection.

Name:
















Mark:

	champion_id	champion_name	release_date	difficulty	role_id
1	1	Aatrox	2013	2	2
2	2	Ahri	2011	2	3
3	3	Akali	2010	2	13
4	4	Alistar	2009	1	1
5	5	Amumu	2009	1	1
6	6	Anivia	2009	3	10
7	7	Annie	2009	1	3
8	8	Aphelios	2019	3	12
9	9	Ashe	2009	1	12
10	10	Aurelion Sol	2016	3	10
11	11	Azir	2014	3	11
12	12	Bard	2015	3	9
13	13	Blitzcrank	2009	2	9
14	14	Brand	2011	2	3
15	15	Braum	2014	2	8
16	16	Caitlyn	2011	1	12
17	17	Camille	2016	3	13
18	18	Cassiopeia	2010	3	10
19	19	Cho'Gath	2009	2	11
20	20	Corki	2009	2	12
21	21	Darius	2012	2	2
22	22	Diana	2012	2	13
23	23	Draven	2012	3	12
24	24	Dr. Mundo	2009	1	2
25	25	Ekkko	2015	2	13
26	26	Elise	2012	3	7
27	27	Evelynn	2009	2	13
28	28	Ezreal	2010	2	12
29	29	Fiddlesticks	2009	2	11
30	30	Fiora	2012	3	6
31	31	Fizz	2011	2	13
32	32	Galio	2010	2	8
33	33	Gangplank	2009	3	11
34	34	Garen	2010	1	2
35	35	Gnar	2014	3	11

Database content for table champion

Name:

Mark:

Structure			Data	Constraints	Indexes	Triggers	DDL											
Grid view			Form view															
																<input type="text" value="Filter data"/>		Total rows loaded: 13
	role_id	roles																
1	1	Vanguard																
2	2	Juggernaut																
3	3	Burst																
4	4	Enchanter																
5	5	Artillery																
6	6	Skirmisher																
7	7	Diver																
8	8	Warden																
9	9	Catcher																
10	10	Battlemage																
11	11	Specialist																
12	12	Marksman																
13	13	Assassin																

Name:

Mark:

[illegible]

Table role's structure

Name:

Mark:

Teacher Checklists:

AS91879- Develop a digital outcome to manage data

Credits: 4

NZQA: <https://www.nzqa.govt.nz/nqfdocs/ncea-resource/achievements/2019/as91879.pdf>

Achieved- Develop a digital outcome to manage data	Evidence	
using appropriate tools and techniques to structure, organise, query and present data for a purpose and end user		✓
applying appropriate data integrity and testing procedures		✓
describing relevant implications.		✓
Merit- Develop an informed digital outcome to manage data		
using information from testing procedures to improve the quality and functionality of the outcome		✓
structuring, organising and querying the data logically		✓
addressing relevant implications.		✓
Excellence- Develop a refined digital outcome to manage data		
iterative improvement throughout the development and testing process		✓
presenting the data effectively for the purpose and to meet end-user requirements.	Not all data available to the user, some formatting issues with table output	

Name:

Mark:

Develop a computer program

Credits: 4 (Internal)

NZQA: <http://www.nzqa.govt.nz/nqfdocs/ncea-resource/achievements/2018/as91883.pdf>

Achieved Develop a computer program	Evidence	
Wrote a program that performs a specific task using a suitable programming language		✓
Set out the program code clearly		✓
Documented the program with comments		✓
Tested and debugged to ensure that it works on a sample of expected cases		✓
Merit Develop an informed computer program		
Documented the program with variable names and comments that describe code function and behaviour		✓
Following conventions of the chosen programming language		✓
Tested and debugged the program in an organised way to ensure it works on expected and relevant boundary cases		✓
Excellence Develop a refined computer program		
Ensured the program is a well structured logical solution to the task	Doesn't loop- exits prematurely after certain options	?
Making the program flexible and robust	Hard coded checks like the 1-14 for the ID's of the champion role. Makes it hard to change if you add new roles or delete/edit and the id's change.	?
Comprehensively tested and debugged the program		

Comments:

Final grades will be decided using professional judgement based on a holistic examination of the evidence provided against the criteria in the Achievement Standard.