

CPSC 304 Project Cover Page

Milestone #: 4

Date: April 5, 2023

Group Number: 39

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Kitty Liu	57347221	f9d9y	kittyliu113@gmail.com
Vanessa Lee	45559374	b5n2b	vanessacarlinglee@gmail.com
Isaac Chung	87754206	y8j3b	isaacchung1990@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

Project Description

Our project focuses on collecting and analyzing data from Nintendo's popular video game, Super Smash Bros. The data includes information about players' rankings, wins, and losses in tournaments, as well as details about the characters they play, the stages they compete on, and the game modes they use.

The application is designed for Super Smash Bros players, tournament organizers, and fans who are interested in staying updated on the latest matches, tournament information, and player statistics. By storing and analyzing this data, our application provides valuable insights into players' career statistics and tournament performance, allowing fans and players alike to closely follow the Super Smash Bros esports scene.

In addition, our application can also be helpful for tournament organizers, as it provides them with information for event planning, such as live streams and event schedules. Furthermore, players can use the application to keep track of their own progress and improve their gameplay by analyzing their performance in previous tournaments.

Overall, our project aims to provide a comprehensive and user-friendly platform for Super Smash Bros players, tournament organizers, and fans to access and analyze data related to the game. By offering valuable insights and statistics, our application aims to enhance the overall experience of the Super Smash Bros esports community. Our project not only aims to depict data from the game but also serves as a valuable tool for players, fans, and organizers alike. It also serves as a one-stop solution for Super Smash Bros enthusiasts to stay updated and engaged with the game and aims to cater to the needs of Super Smash Bros players, tournament organizers, and fans alike, providing them with a comprehensive platform to access and analyze data related to the game.

Project Description — Changes

Our final project has undergone several changes from our original schema. Firstly, we implemented feedback from Milestone 2 and fixed errors such as missing foreign keys, attribute name typos, methods of combining 1:1 relationships, etc. The application's goal and use cases remain the same. However, we restructured the schema to include more robust information and for information across tables to connect more seamlessly, namely adding attributes and centralizing foreign keys to the "Game" table, rather than referring to "game_id" in the majority of the tables. Furthermore, we used another way to write our IsA relationship SQL DDL statements to make it more robust for users.

Secondly, we introduced new entities called "Game_Tournament" and "Game_Player". "Game_Tournament" represents a game played in a tournament, while "Game_Player" represents an individual player participating in a game. We realized that there was no established relationship between games and tournaments, so we added these entities to represent the possibility of a game being part of a tournament. These relations also helped our "Player" table to be less convoluted.

Thirdly, another modification we made was to transform "plays_on_console" from a standalone table into an attribute. After considering that players use the same consoles in a game, we decided to remove it as a separate table and instead include it as an attribute within the "Game" entity. In reality, a separate table for consoles would not be very informative or useful in real-life applications.

The final change we made to our original schema was to replace the "ranking" attribute in the "Player" entity with the attribute "ranking_level". We realized that devising a ranking system based on a win/loss ratio could be challenging, especially in cases of ties where we would need to develop an algorithm to handle such situations. Therefore, we opted for a more straightforward approach by changing "ranking" to "ranking_level", representing different tiers players can be in (i.e., beginner, bronze, silver, gold, platinum). Furthermore, we added attributes such as "age" and "country" to make our player database more robust and have more interesting and meaningful queries. "Country" represents the country of origin for each player and "age" is the age of the player.

Overall, these changes enhance the clarity and functionality of our final project. The redefined entities and attributes provide a more comprehensive representation of the relationships and characteristics within the schema, and the changes reflect our team's thoughtful consideration and strategic decision-making throughout the project development process.

Schema

Please note: **foreign keys are bolded** and primary keys are underlined

Player (username: CHAR(20), country: CHAR(20), ranking_level: CHAR(20), age: INTEGER, wins: INTEGER, losses: INTEGER)

Stage (stage_name: CHAR(20), form: CHAR(20))

Smash_Character (character_name: CHAR(20), costume: CHAR(20))

Ruleset (ruleset_type: CHAR(20), win_criteria: CHAR(20))

Spirits (spirits_name: CHAR(20), spirits_ability: CHAR(50), spirits_type: CHAR(20))

Game (game_id: INTEGER, console: CHAR(20), game_mode: CHAR(20), **stage_name**: CHAR(20), **ruleset_type**: CHAR(20), **spirits_name**: CHAR(20)), stage_name, ruleset_type, and spirits_name NOT NULL

Game_Player (**game_id**: INTEGER, username: CHAR(20), **character_name**: CHAR(20)), character_name NOT NULL

Tournament (tournament_id: CHAR(20), tournament_name: CHAR(20), organizer: CHAR(20), startdate: DATE, end_date: DATE, prize_pool: INTEGER)

Game_Tournament (**game_id**: INTEGER, **tournament_id**: INTEGER), tournament_id NOT NULL

Ability (**character_name**: CHAR(20), ultimate_attack: CHAR(20), up_attack: CHAR(20), neutral_attack: CHAR(20), down_attack: CHAR(20))

Single_Player_Game (**game_id**: INTEGER, competitor_type: CHAR(20), console: CHAR(20), game_mode: CHAR(20), **stage_name**: CHAR(20), **ruleset_type**: CHAR(20), **spirits_name**: CHAR(20)), stage_name, ruleset_type, and spirits_name NOT NULL

Multiplayer_Game (**game_id**: INTEGER, number_of_players: INTEGER, console: CHAR(20), game_mode: CHAR(20), **stage_name**: CHAR(20), **ruleset_type**: CHAR(20), **spirits_name**: CHAR(20)), stage_name, ruleset_type, and spirits_name NOT NULL

Screenshots of data present (after the SQL script is run) on MySQL

All tables

```
mysql> show tables;
+----------------+
| Tables_in_smash |
+----------------+
| Ability          |
| Game             |
| Game_Player      |
| Game_Tournament   |
| Multiplayer_Game |
| Player            |
| Ruleset           |
| Single_Player_Game |
| Smash_Character  |
| Spirits           |
| Stage             |
| Tournament        |
+----------------+
12 rows in set (0.02 sec)
```

Player, 23 rows

```
mysql> SELECT * FROM Player;
+-----+-----+-----+-----+-----+-----+
| username | country | ranking_level | age | wins | losses |
+-----+-----+-----+-----+-----+-----+
| Adam     | China   | gold          | 16  | 49   | 6      |
| Anywehe  | South Korea | platinum    | 21  | 64   | 12     |
| Christian | Philippines | bronze       | 26  | 23   | 5      |
| CyberKnight | Canada   | bronze       | 28  | 29   | 7      |
| EliteGamingPro | UK       | gold          | 22  | 45   | 9      |
| EpicGamer88 | Australia | beginner     | 12  | 17   | 11     |
| Faker     | South Korea | platinum    | 20  | 54   | 2      |
| GameOn92  | USA      | beginner     | 40  | 15   | 8      |
| Hauntzer  | Philippines | gold          | 27  | 45   | 46     |
| Ninja     | USA      | platinum    | 20  | 53   | 5      |
| NinjaGamer | Canada   | bronze       | 29  | 20   | 5      |
| Overtaker  | Canada   | platinum    | 22  | 59   | 5      |
| PewDiePie  | Sweden   | platinum    | 20  | 60   | 12     |
| PixelWarrior | UK       | beginner     | 13  | 14   | 10     |
| RetroGamerX | Australia | silver        | 35  | 33   | 12     |
| Sally     | Japan    | Silver        | 18  | 30   | 8      |
| Shroud    | Canada   | silver        | 20  | 33   | 8      |
| Summit1g   | USA      | diamond       | 18  | 101  | 3      |
| TheGamingGeek | USA      | beginner     | 15  | 18   | 6      |
| ThunderBolt | USA      | silver        | 31  | 34   | 4      |
| Victory    | Australia | silver        | 30  | 35   | 9      |
| VirtualAssassin | Canada | bronze       | 16  | 22   | 3      |
| Zahgrd    | Australia | beginner     | 14  | 5    | 86     |
+-----+-----+-----+-----+-----+-----+
23 rows in set (0.02 sec)
```

Stage, 10 rows

stage_name	form
Battlefield	Battle
Big Blue	Battle
Final Destination	Omega
Lylat Cruise	64
Mario Galaxy	Normal
Mushroom Kingdom I	Normal
Spring Stadium	Omega
Super Happy Tree	Omega
Windy Hill Zone	Battle
Yoshi Island	Melee

10 rows in set (0.01 sec)

Smash_Character, 21 rows

character_name	costume
Boo	Red
Bowser	orange
Captain Falcon	blue
Daisy	yellow
Donkey Kong	Blue
Falco	blue
Fox	orange
Ice Climbers	red
Jigglypuff	pink
Kirby	pink
Link	green
Luigi	green
Mario	red
Mewtwo	purple
Ness	blue
Pikachu	green
Princess Peach	pink
Samus	orange
Wario	purple
Yoshi	green
Zelda	Green

21 rows in set (0.00 sec)

Ruleset, 8 rows

```
mysql> SELECT * FROM ruleset;
+-----+-----+
| ruleset_type | win_criteria |
+-----+-----+
| All-Star Mode | Set opponents
| Coin          | Practice
| Squad Strike | Last team eliminated
| Stamina       | Highest health meter
| Stock          | 3 stock
| Target Blast  | Damage dealt points
| Time limit    | 6 minutes
| Tournament    | Bracketed style
+-----+
8 rows in set (0.01 sec)
```

Spirits, 10 rows

```
mysql> SELECT * FROM Spirits;
+-----+-----+-----+
| spirits_name      | spirits_ability   | spirits_type |
+-----+-----+-----+
| Cappy             | Midair jump      | Ace
| Captain Toad     | Item Gravitation | Primary
| Cat Princess Peach | Lower Weight     | Primary
| Diddy             | Power Increase   | Novice
| Dry Bowser        | Flame proof      | Advanced
| Fawful            | Resist status    | Advanced
| Huey              | Freezing Resist  | Support
| Mario             | Fight             | Fighter
| Vivian            | Lower Weight     | Primary
| Waluigi           | Foot Attack      | Ace
+-----+
10 rows in set (0.00 sec)
```

Game, 25 rows

```
mysql> SELECT * FROM Game;
```

game_id	console	game_mode	stage_name	ruleset_type	spirits_name
1	Nintendo Switch	Classic	Battlefield	Stock	Mario
2	Nintendo Switch	Training	Big Blue	Time limit	Waluigi
3	Playstation 4	Classic	Yoshi Island	Squad Strike	Huey
4	Nintendo Switch	All-Star	Yoshi Island	Time limit	Vivian
5	Xbox One	Stadium	Final Destination	Stock	Captain Toad
6	Playstation 4	Training	Mushroom Kingdom I	Coin	Cappy
7	Nintendo Switch	Event	Lylat Cruise	Time limit	Fawful
8	Xbox One	Event	Mushroom Kingdom I	Stock	Diddy
9	Playstation 4	Mario Galaxy	Battlefield	Coin	Cat Princess Peach
10	Nintendo Switch	Mario Galaxy	Final Destination	Time limit	Fawful
11	Xbox One	Stadium	Lylat Cruise	Stock	Diddy
12	Playstation 4	All-Star	Windy Hill Zone	Coin	Diddy
13	Nintendo Switch	Training	Spring Stadium	Target Blast	Vivian
14	Xbox One	Mario Galaxy	Windy Hill Zone	Stock	Mario
15	Playstation 4	Classic	Final Destination	Coin	Fawful
16	Xbox One	Training	Yoshi Island	Target Blast	Huey
17	Playstation 4	Event	Final Destination	Coin	Huey
18	Xbox One	Classic	Lylat Cruise	Stock	Huey
19	Playstation 4	Stadium	Final Destination	Coin	Vivian
20	Playstation 4	All-Star	Battlefield	Coin	Dry Bowser
21	Nintendo Switch	Classic	Final Destination	Target Blast	Fawful
22	Wii U	Stadium	Spring Stadium	Time limit	Diddy
23	Wii U	All-Star	Yoshi Island	Coin	Captain Toad
24	Xbox One	Training	Mushroom Kingdom I	Target Blast	Mario
25	Playstation 4	Event	Spring Stadium	Stock	Waluigi

```
25 rows in set (0.00 sec)
```

Game_Player, 59 rows

mysql> SELECT * FROM Game_Player;		
game_id	username	character_name
6	PewDiePie	Boo
8	PewDiePie	Boo
14	PixelWarrior	Boo
19	PewDiePie	Boo
20	TheGamingGeek	Boo
6	Overtaker	Bowser
9	Summit1g	Bowser
24	Ninja	Bowser
4	GameOn92	Daisy
7	Shroud	Daisy
14	GameOn92	Daisy
18	Faker	Daisy
23	Ninja	Daisy
4	Adam	Donkey Kong
13	Hauntzer	Donkey Kong
22	Overtaker	Falco
21	RetroGamerX	Ice Climbers
22	Ninja	JigglyPuff
3	Zahgrd	Kirby
6	CyberKnight	Kirby
6	EpicGamer88	Kirby
12	Zahgrd	Kirby
18	Shroud	Kirby
20	Overtaker	Kirby
22	Anywehe	Kirby
24	Faker	Link
24	PewDiePie	Luigi
1	Overtaker	Mario
4	Christian	Mario
4	NinjaGamer	Mario
6	Ninja	Mario
10	Overtaker	Mario
12	Christian	Mario
14	Overtaker	Mario
14	VirtualAssassin	Mario
17	EpicGamer88	Mario
24	Overtaker	Mario
22	Adam	Ness
2	Adam	Pikachu
2	Overtaker	Pikachu
5	Hauntzer	Pikachu
10	Sally	Pikachu
11	Adam	Pikachu
14	RetroGamerX	Pikachu
14	TheGamingGeek	Pikachu
6	ThunderBolt	Princess Peach
8	Faker	Princess Peach
16	Sally	Princess Peach
18	Ninja	Princess Peach
25	Ninja	Samus
4	Victory	Wario
6	EliteGamingPro	Wario
12	Victory	Wario
16	CyberKnight	Wario
2	Hauntzer	Yoshi
6	RetroGamerX	Yoshi
10	Anywehe	Yoshi
15	Sally	Yoshi
24	Anywehe	Yoshi

59 rows in set (0.00 sec)

Tournament, 14 rows

tournament_id	tournament_name	organizer	startdate	enddate	prize_pool
101	ESL One Hamburg 2015	ESL Gaming	2015-11-16	2015-11-17	155000
102	Get On My Line 2016	Even Matchup Gaming	2016-06-25	2016-06-26	10000
103	Low Tier City 8	Tourney Locator	2017-07-29	2017-07-31	10000
104	Smash Ultimate World Tour 2018	Nintendo	2018-03-02	2018-12-08	250000
105	Smash Worlds 2019	Riot Games	2019-10-02	2019-11-10	2500000
106	ESL One LA 2020	ESL Gaming	2020-03-28	2020-04-19	400000
107	EVO 2020 Online	EVO	2020-07-04	2020-08-02	39000
108	Asia Championships 2021	Perfect World	2021-04-01	2021-04-04	500000
109	PUBG Global Invitational 2021	PUBG Corporation	2021-02-05	2021-03-28	7000000
110	Smash Ultimate Summit 3	Beyond The Summit	2021-05-06	2021-05-09	150000
111	Smash Championships 2021	Activision Blizzard	2021-08-19	2021-08-22	2500000
112	Rainbow Six Siege Invitational 2022	Ubisoft	2022-02-09	2022-02-20	3000000
113	Smash World Cup 2022	Epic Games	2022-07-22	2022-07-24	50000000
114	The International 2023	Valve Corporation	2023-08-18	2023-08-28	30000000

14 rows in set (0.00 sec)

Game_Tournament, 25 rows

```
mysql> SELECT * FROM Game_Tournament;
```

game_id	tournament_id
1	101
2	101
3	102
4	103
5	103
6	103
7	103
8	104
9	105
10	105
11	105
12	106
13	106
14	107
15	108
16	108
17	109
18	109
19	109
20	110
21	111
22	112
23	112
24	112
25	112

25 rows in set (0.01 sec)

Ability, 16 rows

character_name	ultimate_attack	up_attack	neutral_attack	down_attack
Boo	Quick Shift	Ghost Mode	Sting	Teleport
Bowser	Final Cutter	Stone	Inhale	Flame
Daisy	Daisy Parasol	Vegetable	Toad	Slash
Donkey Kong	Power	Jump	Spin	Crouch
Falco	Landmaster	Fire Bird	Blaster	Reflector
Ice Climbers	Iceberg	Hammer	Ice Shot	Blizzard
Jigglypuff	Puff Up	Sing	Rollout	Rest
Kirby	Quick Attack	Thunder	Jolt	Jump
Luigi	Poltergust	Jump Punch	Fireball	Missile
Mario	Super Jump Punch	FLUDD	Fireball	Hero
Mewtwo	Psystrike	Teleport	Shadow Ball	Disable
Pikachu	Throw	Fire Spit	Jump	Thunderbolt
Princess Peach	Peach Parasol	Vegetable	Toad	Spin
Wario	Corkscrew	Wario Waft	Chomp	Spin
Yoshi	Egg Throw	Yoshi Bomb	Egg Lay	Crouch
Zelda	Arrow	Transform	Wind	Fire

16 rows in set (0.01 sec)

Single_Player_Game, 13 rows

game_id	competitor_type	console	game_mode	stage_name	ruleset_type	spirits_name
1	CPU	Nintendo Wii	Classic	Battlefield	Stock	Mario
3	CPU	Playstation 4	Classic	Yoshi Island	Squad Strike	Huey
5	Player	Xbox One	Stadium	Final Destination	Stock	Captain Toad
7	Player	Nintendo Switch	Event	Lylat Cruise	Time limit	Fawful
9	CPU	Playstation 4	Mario Galaxy	Battlefield	Coin	Cat Princess Peach
11	Player	Xbox One	Stadium	Lylat Cruise	Stock	Diddy
13	CPU	Nintendo Switch	Training	Spring Stadium	Target Blast	Vivian
15	Player	Playstation 4	Event	Final Destination	Coin	Fawful
17	CPU	Playstation 4	Event	Final Destination	Coin	Huey
19	Player	Playstation 4	Stadium	Final Destination	Coin	Vivian
21	Player	Nintendo Switch	Classic	Final Destination	Target Blast	Fawful
23	CPU	Wii U	All-Star	Yoshi Island	Coin	Captain Toad
25	CPU	Playstation 4	Event	Spring Stadium	Stock	Waluigi

13 rows in set (0.01 sec)

Multiplayer_Game, 12 rows

game_id	number_of_players	console	game_mode	stage_name	ruleset_type	spirits_name
2	3	Nintendo Switch	Training	Big Blue	Time limit	Waluigi
4	5	Nintendo Switch	All-Star	Yoshi Island	Time limit	Vivian
6	8	Playstation 4	Training	Mushroom Kingdom I	Coin	Cappy
8	2	Xbox One	Event	Mushroom Kingdom I	Stock	Diddy
10	3	Nintendo Switch	Mario Galaxy	Final Destination	Time limit	Fawful
12	3	Playstation 4	All-Star	Windy Hill Zone	Coin	Diddy
14	6	Xbox One	Mario Galaxy	Windy Hill Zone	Stock	Mario
16	2	Xbox One	Training	Yoshi Island	Target Blast	Huey
18	3	Xbox One	Classic	Lylat Cruise	Stock	Huey
20	2	Playstation 4	All-Star	Battlefield	Coin	Dry Bowser
22	4	Wii U	Stadium	Spring Stadium	Time limit	Diddy
24	5	Xbox One	Training	Mushroom Kingdom I	Target Blast	Mario

12 rows in set (0.01 sec)

Screenshots of data present (after the SQL script is run) on GUI

Player

PLAYERS
PLAYER SELECTION

USERNAME: <input type="text"/>	COUNTRY: <input type="text"/>	RANKING LEVEL: <input type="text"/>
AGE: <input type="text"/>	WINS: <input type="text"/>	LOSSES: <input type="text"/>
<input type="button" value="SEARCH"/> <input type="button" value="UPDATE"/> <input type="checkbox"/> USERNAME <input type="checkbox"/> COUNTRY <input type="checkbox"/> RANKING LEVEL <input type="checkbox"/> AGE <input type="checkbox"/> WINS <input type="checkbox"/> LOSSES <input type="button" value="SUBMIT"/>		

USERNAME	COUNTRY	RANKING LEVEL	AGE	WINS	LOSSES
ADAM	CHINA	GOLD	16	49	6
ANYWEHE	SOUTH KOREA	PLATINUM	21	64	12
CHRISTIAN	PHILIPPINES	BRONZE	26	23	5
CYBERKNIGHT	CANADA	BRONZE	28	29	7
ELITEGAMINGPRO	UK	GOLD	22	45	9
EPICGAMER88	AUSTRALIA	BEGINNER	12	17	11
FAKER	SOUTH KOREA	PLATINUM	20	54	2
GAMEON92	USA	BEGINNER	40	15	8
HAUNTZER	PHILIPPINES	GOLD	27	45	46
NINJA	USA	PLATINUM	20	53	5
NINJAGAMER	CANADA	BRONZE	29	20	5
OVERTAKER	CANADA	PLATINUM	22	59	5
PEWDIEPIE	SWEDEN	PLATINUM	20	60	12
PIXELWARRIOR	UK	BEGINNER	13	14	10
RETROGAMERX	AUSTRALIA	SILVER	35	33	12
SALLY	JAPAN	SILVER	18	30	8
SHROUD	CANADA	SILVER	20	33	8
SUMMIT1G	USA	DIAMOND	18	101	3
THEGAMINGGEEK	USA	BEGINNER	15	18	6
THUNDERBOLT	USA	SILVER	31	34	4
VICTORY	AUSTRALIA	SILVER	30	35	9
VIRTUALASSASSIN	CANADA	BRONZE	16	22	3
ZAHGRD	AUSTRALIA	BEGINNER	14	5	86

Tournaments

TOURNAMENTS

TOURNAMENT NAME	ORGANIZER	START DATE	END DATE	PRIZE POOL	
ESL ONE HAMBURG 2015	ESL GAMING	2015-11-16	2015-11-17	155000	<button>DELETE</button>
GET ON MY LINE 2016	EVEN MATCHUP GAMING	2016-06-25	2016-06-26	10000	<button>DELETE</button>
LOW TIER CITY 8	TOURNEY LOCATOR	2017-07-29	2017-07-31	10000	<button>DELETE</button>
SMASH ULTIMATE WORLD TOUR 2018	NINTENDO	2018-03-02	2018-12-08	250000	<button>DELETE</button>
SMASH WORLDS 2019	RIOT GAMES	2019-10-02	2019-11-10	2500000	<button>DELETE</button>
ESL ONE LA 2020	ESL GAMING	2020-03-28	2020-04-19	400000	<button>DELETE</button>
EVO 2020 ONLINE	EVO	2020-07-04	2020-08-02	39000	<button>DELETE</button>
ASIA CHAMPIONSHIPS 2021	PERFECT WORLD	2021-04-01	2021-04-04	500000	<button>DELETE</button>
PUBG GLOBAL INVITATIONAL 2021	PUBG CORPORATION	2021-02-05	2021-03-28	7000000	<button>DELETE</button>
SMASH ULTIMATE SUMMIT 3	BEYOND THE SUMMIT	2021-05-06	2021-05-09	150000	<button>DELETE</button>
SMASH CHAMPIONSHIPS 2021	ACTIVISION BLIZZARD	2021-08-19	2021-08-22	2500000	<button>DELETE</button>
RAINBOW SIX SIEGE INVITATIONAL 2022	UBISOFT	2022-02-09	2022-02-20	3000000	<button>DELETE</button>
SMASH WORLD CUP 2022	EPIC GAMES	2022-07-22	2022-07-24	50000000	<button>DELETE</button>
THE INTERNATIONAL 2023	VALVE CORPORATION	2023-08-18	2023-08-28	30000000	<button>DELETE</button>

Characters

CHARACTERS

NAME	COSTUME
BOO	RED
BOWSER	ORANGE
CAPTAIN FALCON	BLUE
DAISY	YELLOW
DONKEY KONG	BLUE
FALCO	BLUE
FOX	ORANGE
ICE CLIMBERS	RED
JIGGLYPUFF	PINK
KIRBY	PINK
LINK	GREEN
LUIGI	GREEN
MARIO	RED
MEWTWO	PURPLE
NESS	BLUE
PIKACHU	GREEN
PRINCESS PEACH	PINK
SAMUS	ORANGE
WARIO	PURPLE
YOSHI	GREEN
ZELDA	GREEN

Abilities

 **ABILITIES**

CHARACTER NAME: ULTIMATE ATTACK: UP ATTACK:

NEUTRAL ATTACK: DOWN ATTACK: **SUBMIT**

CHARACTER NAME	ULTIMATE ATTACK	UP ATTACK	NEUTRAL ATTACK	DOWN ATTACK
BOO	QUICK SHIFT	GHOST MODE	STING	TELEPORT
BOWSER	FINAL CUTTER	STONE	INHALE	FLAME
DAISY	DAISY PARASOL	VEGETABLE	TOAD	SLASH
DONKEY KONG	POWER	JUMP	SPIN	CROUCH
FALCO	LANDMASTER	FIRE BIRD	BLASTER	REFLECTOR
ICE CLIMBERS	ICEBERG	HAMMER	ICE SHOT	BLIZZARD
JIGGLYPUFF	PUFF UP	SING	ROLLOUT	REST
KIRBY	QUICK ATTACK	THUNDER	JOLT	JUMP
LUIGI	POLTERGUST	JUMP PUNCH	FIREBALL	MISSILE
MARIO	SUPER JUMP PUNCH	FLUDD	FIREBALL	HERO
MEWTWO	PSYSTRIKE	TELEPORT	SHADOW BALL	DISABLE
PIKACHU	THROW	FIRE SPIT	JUMP	THUNDERBOLT
PRINCESS PEACH	PEACH PARASOL	VEGETABLE	TOAD	SPIN
WARIO	CORKSCREW	WARIO WAFT	CHOMP	SPIN
YOSHI	EGGTHROW	YOSHI BOMB	EGG LAY	CROUCH
ZELDA	ARROW	TRANSFORM	WIND	FIRE

Game Players

GAME PLAYER

GAME ID	USERNAME	CHARACTER NAME
1	OVERTAKER	MARIO
2	ADAM	PIKACHU
2	HAUNTZER	YOSHI
2	OVERTAKER	PIKACHU
3	ZAHGRD	KIRBY
4	ADAM	DONKEY KONG
4	CHRISTIAN	MARIO
4	GAMEON92	DAISY
4	NINJAGAMER	MARIO
4	VICTORY	WARIO
5	HAUNTZER	PIKACHU
6	CYBERKNIGHT	KIRBY
6	ELITEGAMINGPRO	WARIO
6	EPICGAMER88	KIRBY
6	NINJA	MARIO
6	OVERTAKER	BOWSER
6	PEWDIEPIE	BOO
6	RETROGAMERX	YOSHI
6	THUNDERBOLT	PRINCESS PEACH
7	SHROUD	DAISY
8	FAKER	PRINCESS PEACH
8	PEWDIEPIE	BOO
9	SUMMIT1G	BOWSER
10	ANYWBHE	YOSHI
10	OVERTAKER	MARIO
10	SALLY	PIKACHU
11	ADAM	PIKACHU
12	CHRISTIAN	MARIO
12	VICTORY	WARIO

12	ZAHGRD	KIRBY
13	HAUNTZER	DONKEY KONG
14	GAMEON92	DAISY
14	OVERTAKER	MARIO
14	PIXELWARRIOR	BOO
14	RETROGAMERX	PIKACHU
14	THEGAMINGGEEK	PIKACHU
14	VIRTUALASSASSIN	MARIO
15	SALLY	YOSHI
16	CYBERKNIGHT	WARIO
16	SALLY	PRINCESS PEACH
17	EPICGAMER88	MARIO
18	FAKER	DAISY
18	NINJA	PRINCESS PEACH
18	SHROUD	KIRBY
19	PEWDIEPIE	BOO
20	OVERTAKER	KIRBY
20	THEGAMINGGEEK	BOO
21	RETROGAMERX	ICE CLIMBERS
22	ADAM	NESS
22	ANYWEHE	KIRBY
22	NINJA	JIGGLYPUFF
22	OVERTAKER	FALCO
23	NINJA	DAISY
24	ANYWEHE	YOSHI
24	FAKER	LINK
24	NINJA	BOWSER
24	OVERTAKER	MARIO
24	PEWDIEPIE	LUIGI
25	NINJA	SAMUS

Game_Tournament

GAME TOURNAMENT	
GAME ID	TOURNAMENT ID
1	101
2	101
3	102
4	103
5	103
6	103
7	103
8	104
9	105
10	105
11	105
12	106
13	106
14	107
15	108
16	108
17	109
18	109
19	109
20	110
21	111
22	112
23	112
24	112
25	112

Other tables (for specific queries)

GAMES PLAYED	
SELECT A USER:	
<input type="button" value="-- SELECT A USER --"/>	
USERNAME	NUMBER OF GAMES PLAYED
WINS BY ATTRIBUTE	
SELECT AN ATTRIBUTE:	
<input type="button" value="-- SELECT AN ATTRIBUTE --"/>	
TOTAL WINS	
TEAMS BY ATTRIBUTE	
SELECT AN ATTRIBUTE:	
<input type="button" value="-- SELECT AN ATTRIBUTE --"/>	
NUMBER OF PLAYERS	
AVERAGE AGE BY ATTRIBUTE	
SELECT AN ATTRIBUTE:	
<input type="button" value="-- SELECT AN ATTRIBUTE --"/>	
AVERAGE AGE	
PLAYERS WHO PLAYED EVERY GAME IN A TOURNAMENT	
SELECT A TOURNAMENT:	
<input type="button" value="-- SELECT A TOURNAMENT --"/>	
USERNAME	

SQL Queries Used

File name: routes/apiRouter.js

Line #	SQL Query	Description
85	<code>SELECT * FROM Player WHERE</code>	Selection query: User is able to search for tuples with specific keys. For example, a user can search for Players from USA or with the username "Adam", etc. Essentially, the user can specify the WHERE clause.
130	<code>SELECT \${selectClause} FROM Player</code>	Projection query: User can filter specific columns they want to see in the Players table. For example, users can check the username and country checkboxes if they only want to see data from those columns in the Players table.
163	<code>INSERT INTO ABILITY VALUES ("\${body.character_name}", "\${body.ultimate_attack}", "\${body.up_attack}", "\${body.neutral_attack}", "\${body.down_attack}")</code>	Insert query: User can insert abilities of a character in the Abilities table. However, please note this only works if the character exists in the Characters table.
190	<code>UPDATE Player SET WHERE username="\${req.params.username}"</code>	Update query: User can update an existing player's information in the Players table. For example, if a user has moved countries, the user can update their country.
220	<code>DELETE FROM Tournament WHERE tournament_id="\${req.params.id}"</code>	Delete query: User can delete a tournament and this will also delete the games in that deleted tournament.
243	<code>SELECT Player.username, COUNT(Game_Player.game_id) AS num_games_played</code>	Join query: Joins the game_player and player table. User can search for a player and the number of games they have played

	<pre> FROM Player JOIN Game_Player ON Player.username = Game_Player.username WHERE Player.username = "\${username}" GROUP BY Player.username </pre>	
272	<pre> SELECT \${attribute}, SUM(wins) as total_wins FROM Player GROUP BY \${attribute} </pre>	<p>Aggregation with GROUP BY: User can find the total number of wins for an attribute. For example, user can see various countries and how many wins they have.</p>
298	<pre> SELECT \${attribute}, COUNT(username) AS num_players FROM Player GROUP BY \${attribute} HAVING num_players > 2 ORDER BY num_players DESC </pre>	<p>Aggregation with HAVING: This query calculates the number of players in each attribute that has more than 2 players and orders the results by the number of players in descending order.</p>
333	<pre> SELECT p1.\${attribute}, AVG(p1.age) as avg_age FROM Player p1 GROUP BY p1.\${attribute} HAVING 1 < (SELECT COUNT(*) FROM Player p2 WHERE p1.\${attribute} = p2.\${attribute}) </pre>	<p>Nested Aggregation with GROUP BY: Retrieves the average age of players for an attribute, but only for ranking levels that have more than one player.</p>
370	<pre> SELECT username FROM Player WHERE NOT EXISTS (SELECT game_id FROM Game_Tournament WHERE tournament_id = \${tournament_id} EXCEPT SELECT game_id) </pre>	<p>Division query: Selects the usernames of players that have played in every game of the specified X tournament.</p> <ol style="list-style-type: none"> the query uses a subquery that finds all the game IDs in the Game_Tournament table for tournament X compares it with the game IDs in the Game_Player table for each individual player

```
        FROM Game_Player
        WHERE Game_Player.username
= Player.username)
```

3) If a player has played in all games for the tournament, their username will be returned by the outer SELECT statement.
The EXCEPT keyword is used to find the game IDs in the first subquery that are not present in the second subquery, which helps determine if a player has played in all the games or not.

Screenshots of Query Functionality using GUI

1. INSERT operation:

- **Scenario #1:** Foreign key value already exists in the table so you can insert the tuple with no problems.
- **Scenario #2:** Foreign key value does not exist yet so you need to first run an insert into the table before inserting the value the user has chosen.

1a) Scenario #1: INSERT operation - BEFORE

Abilities table before INSERT operation, view below.

CHARACTER NAME	ULTIMATE ATTACK	UP ATTACK	NEUTRAL ATTACK	DOWN ATTACK
BOO	QUICK SHIFT	GHOST MODE	STING	TELEPORT
BOWSER	FINAL CUTTER	STONE	INHALE	FLAME
DAISY	DAISY PARASOL	VEGETABLE	TOAD	SLASH
DONKEY KONG	POWER	JUMP	SPIN	CROUCH
FALCO	LANDMASTER	FIRE BIRD	BLASTER	REFLECTOR
ICE CLIMBERS	ICEBERG	HAMMER	ICE SHOT	BLIZZARD
JIGGLYPUFF	PUFF UP	SING	ROLLOUT	REST
KIRBY	QUICK ATTACK	THUNDER	JOLT	JUMP
LUIGI	POLTERGUST	JUMP PUNCH	FIREBALL	MISSILE
MARIO	SUPER JUMP PUNCH	FLUDD	FIREBALL	HERO
MEWTWO	PSYSTRIKE	TELEPORT	SHADOW BALL	DISABLE
PIKACHU	THROW	FIRE SPIT	JUMP	THUNDERBOLT
PRINCESS PEACH	PEACH PARASOL	VEGETABLE	TOAD	SPIN
WARIO	CORKSCREW	WARIO WAFT	CHOMP	SPIN
YOSHI	EGGTHROW	YOSHI BOMB	EGG LAY	CROUCH

Please note the intended character name you would like to enter in Abilities must exist in the Characters table (view next screenshot) for the INSERT operation to work.

Note that Zelda is a Character name in the Characters table (view last tuple):

CHARACTERS

NAME	COSTUME
BOO	RED
BOWSER	ORANGE
CAPTAIN FALCON	BLUE
DAISY	YELLOW
DONKEY KONG	BLUE
FALCO	BLUE
FOX	ORANGE
ICE CLIMBERS	RED
JIGGLYPUFF	PINK
KIRBY	PINK
LINK	GREEN
LUIGI	GREEN
MARIO	RED
MEWTWO	PURPLE
NESS	BLUE
PIKACHU	GREEN
PRINCESS PEACH	PINK
SAMUS	ORANGE
WARIO	PURPLE
YOSHI	GREEN
ZELDA	GREEN

1b) Scenario #1: INSERT operation - DURING

The process of entering Zelda (which exists in Characters) and their abilities before pressing submit.

CHARACTER NAME	ULTIMATE ATTACK	UP ATTACK	NEUTRAL ATTACK	DOWN ATTACK
BOO	QUICK SHIFT	GHOST MODE	STING	TELEPORT
BOWSER	FINAL CUTTER	STONE	INHALE	FLAME
DAISY	DAISY PARASOL	VEGETABLE	TOAD	SLASH
DONKEY KONG	POWER	JUMP	SPIN	CROUCH
FALCO	LANDMASTER	FIRE BIRD	BLASTER	REFLECTOR
ICE CLIMBERS	ICEBERG	HAMMER	ICE SHOT	BLIZZARD
JIGGLYPUFF	PUFF UP	SING	ROLLOUT	REST
KIRBY	QUICK ATTACK	THUNDER	JOLT	JUMP
LUIGI	POLTERGUST	JUMP PUNCH	FIREBALL	MISSILE
MARIO	SUPER JUMP PUNCH	FLUDD	FIREBALL	HERO
MEWTWO	PSYSTRIKE	TELEPORT	SHADOW BALL	DISABLE
PIKACHU	THROW	FIRE SPIT	JUMP	THUNDERBOLT
PRINCESS PEACH	PEACH PARASOL	VEGETABLE	TOAD	SPIN
WARIO	CORKSCREW	WARIO WAFT	CHOMP	SPIN
YOSHI	EGGTHROW	YOSHI BOMB	EGG LAY	CROUCH

1c) Scenario #1: INSERT operation - AFTER

Pressed submit button after entering Zelda's abilities:

The screenshot shows a web application interface for entering character abilities. At the top, there are four input fields: 'CHARACTER NAME' (containing 'ZELDA'), 'ULTIMATE ATTACK' (containing 'ARROW'), 'UP ATTACK' (containing 'TRANSFORM'), 'NEUTRAL ATTACK' (containing 'WIND'), and 'DOWN ATTACK' (containing 'FIRE'). Below these is a 'SUBMIT' button. The main area displays a table of Super Smash Bros. character abilities. The row for 'ZELDA' is highlighted with a red border.

CHARACTER NAME	ULTIMATE ATTACK	UP ATTACK	NEUTRAL ATTACK	DOWN ATTACK
BOO	QUICK SHIFT	GHOST MODE	STING	TELEPORT
BOWSER	FINAL CUTTER	STONE	INHALE	FLAME
DAISY	DAISY PARASOL	VEGETABLE	TOAD	SLASH
DONKEY KONG	POWER	JUMP	SPIN	CROUCH
FALCO	LANDMASTER	FIRE BIRD	BLASTER	REFLECTOR
ICE CLIMBERS	ICEBERG	HAMMER	ICE SHOT	BLIZZARD
JIGGLYPUFF	PUFF UP	SING	ROLLOUT	REST
KIRBY	QUICK ATTACK	THUNDER	JOLT	JUMP
LUIGI	POLTERGUST	JUMP PUNCH	FIREBALL	MISSILE
MARIO	SUPER JUMP PUNCH	FLUDD	FIREBALL	HERO
MEWTWO	PSYSTRIKE	TELEPORT	SHADOW BALL	DISABLE
PIKACHU	THROW	FIRE SPIT	JUMP	THUNDERBOLT
PRINCESS PEACH	PEACH PARASOL	VEGETABLE	TOAD	SPIN
WARIO	CORKSCREW	WARIO WAFT	CHOMP	SPIN
YOSHI	EGGTHROW	YOSHI BOMB	EGG LAY	CROUCH
ZELDA	ARROW	TRANSFORM	WIND	FIRE

The screenshot shows the browser's developer tools Network tab. It lists several failed requests (404 Not Found) for various API endpoints like '/api/wins', '/api/games-played', and '/api/teams'. Following these errors, a successful POST request is shown with the message: "Query: INSERT INTO ABILITY VALUES ('Zelda', 'Abilities.js:90', 'Arrow', 'Transform', 'Wind', 'Fire')".

```

1 Issue: 1
⚠ DevTools failed to load source map: Could not load content for chrome-extension://gighmmppioklfepojocnamgkbbiolidom/browser-polyfill.js.map: System error: net::ERR_FILE_NOT_FOUND
react-dom.development.js:29840
Download the React DevTools for a better development experience: https://reactjs.org/link/react-devtools
✖ Warning: ReactDOM.render is no longer supported in React 18. Use createRoot instead. Until you switch to the new API, your app will behave as if it's running React 17. Learn more: https://reactjs.org/link/switch-to-createroot
✖ GET http://localhost:3001/api/wins/ 404 (Not Found)
✖ Error: Error: Network response was not ok at fetchData (Wins.js:52:1)
✖ GET http://localhost:3001/api/games-played/ 404 (Not Found)
✖ Error: Error: Network response was not ok at fetchData (GamesPlayed.js:46:1)
✖ GET http://localhost:3001/api/teams/ 404 (Not Found)
✖ Error: Error: Network response was not ok at fetchData (Teams.js:19:1)
✖ GET http://localhost:3001/api/avg-age/ 404 (Not Found)
✖ Error: Error: Network response was not ok at fetchData (AvgAge.js:17:1)
Query: INSERT INTO ABILITY VALUES ("Zelda", "Abilities.js:90", "Arrow", "Transform", "Wind", "Fire")
  
```

1a) Scenario #2: INSERT operation - BEFORE

We want to enter abilities for a character called “Wii”, but the application will not let us.

The screenshot shows the Abilities page with a table of character abilities. A red box highlights the row for 'WII'. The table has columns: CHARACTER NAME, ULTIMATE ATTACK, UP ATTACK, NEUTRAL ATTACK, and DOWN ATTACK. The 'WII' row contains: CHARACTER NAME: WII, ULTIMATE ATTACK: A, UP ATTACK: S, NEUTRAL ATTACK: C, DOWN ATTACK: D. Below the table is a form with fields: CHARACTER NAME: WII, ULTIMATE ATTACK: A, UP ATTACK: S, NEUTRAL ATTACK: C, DOWN ATTACK: D, and a SUBMIT button. To the right of the browser window is a DevTools console window showing several error messages related to React DOM development.

```

at Object.invokeGuardedCallbackDev (react-dom.development.js:4213:1)
at invokeGuardedCallback (react-dom.development.js:4277:1)
at invokeGuardedCallbackAndCatchFirstError (react-dom.development.js:4291:1)
at executeDispatch (react-dom.development.js:9041:1)
at processDispatchQueueItemsInOrder (react-dom.development.js:9073:1)
at processDispatchQueue (react-dom.development.js:9086:1)
at dispatchEventsForPlugins (react-dom.development.js:9097:1)
at react-dom.development.js:9288:1
x > POST http://localhost:3001/api/a Abilities.js:99 ↗
net::ERR_CONNECTION_REFUSED
x > Error: TypeError: Failed to fetch Abilities.js:117
at handleSubmit (Abilities.js:99:1)
at HTMLUnknownElement.callCallback (react-dom.development.js:4164:1)
at Object.invokeGuardedCallbackDev (react-dom.development.js:4213:1)
at invokeGuardedCallback (react-dom.development.js:4277:1)
at invokeGuardedCallbackAndCatchFirstError (react-dom.development.js:4291:1)
at executeDispatch (react-dom.development.js:9041:1)

```

We need to create a Character named “Wii”.

The screenshot shows the Characters page with a table of character names and costumes. A red box highlights the row for 'WII'. The table has columns: NAME and COSTUME. The 'WII' row contains: NAME: WII, COSTUME: WHITE. Below the table is a form with fields: CHARACTER NAME: WII, COSTUME: WHITE, and a SUBMIT button. To the right of the browser window is a DevTools console window showing several error messages related to React DOM development.

```

[16807388337438211] Script Started content.js:8
⚠ DevTools failed to load source map: Could not load content for chrome-extension://hfnankncofebbddgijnnhnfknaad/requestProvider.js.map: System error: net::ERR_BLOCKED_BY_CLIENT
⚠ DevTools failed to load source map: Could not load content for chrome-extension://hfnankncofebbddgijnnhnfknaad/requestSolanProvider.js.map: System error: net::ERR_BLOCKED_BY_CLIENT
⚠ DevTools failed to load source map: Could not load content for chrome-extension://hfnankncofebbddgijnnhnfknaad/content.js.map: System error: net::ERR_BLOCKED_BY_CLIENT
✖ Warning: ReactDOM.render is no longer supported in react-dom.development.js:86
React 18. Use createRoot instead. Until you switch to the new API, your app will behave as if it's running React 17. Learn more: https://reactjs.org/link/switch-to-createRoot
⚠ DevTools failed to load source map: Could not load content for chrome-extension://hfnankncofebbddgijnnhnfknaad/browser-polyfills.js.map: System error: net::ERR_BLOCKED_BY_CLIENT
Query: INSERT INTO Smash_Character VALUES ("Wii", "white") Character.js:100
>

```

Then we can enter Wii’s abilities.

1b) Scenario #2: INSERT operation - DURING

The screenshot shows a web-based application interface. At the top, there is a form titled "ABILITIES" with fields for "CHARACTER NAME" (containing "WII"), "ULTIMATE ATTACK" (containing "A"), "UP ATTACK" (containing "B"), "NEUTRAL ATTACK" (containing "C"), "DOWN ATTACK" (containing "D"), and a "SUBMIT" button. A red box highlights the input fields for the character name, ultimate attack, up attack, neutral attack, and down attack. Below the form is a table titled "CHARACTER NAME" with columns for "CHARACTER NAME", "ULTIMATE ATTACK", "UP ATTACK", "NEUTRAL ATTACK", and "DOWN ATTACK". The table lists various Super Smash Bros. characters with their corresponding attack names.

CHARACTER NAME	ULTIMATE ATTACK	UP ATTACK	NEUTRAL ATTACK	DOWN ATTACK
BOO	QUICK SHIFT	GHOST MODE	STING	TELEPORT
BOWSER	FINAL CUTTER	STONE	INHALE	FLAME
DAISY	DAISY PARASOL	VEGETABLE	TOAD	SLASH
DONKEY KONG	POWER	JUMP	SPIN	CROUCH
FALCO	LANDMASTER	FIRE BIRD	BLASTER	REFLECTOR
ICE CLIMBERS	ICEBERG	HAMMER	ICE SHOT	BLIZZARD
JIGGLYPUFF	PUFF UP	SING	ROLLOUT	REST
KIRBY	QUICK ATTACK	THUNDER	JOLT	JUMP
LUIGI	POLTERGUST	JUMP PUNCH	FIREBALL	MISSILE
MARIO	SUPER JUMP PUNCH	FLUDD	FIREBALL	HERO
MEWTWO	PSYSTRIKE	TELEPORT	SHADOW BALL	DISABLE
PIKACHU	THROW	FIRE SPIT	JUMP	THUNDERBOLT
PRINCESS PEACH	PEACH PARASOL	VEGETABLE	TOAD	SPIN
WARIO	CORKSCREW	WARIO WAFT	CHOMP	SPIN
YOSHI	EGGTHROW	YOSHI BOMB	EGG LAY	CROUCH

1c) Scenario #2: INSERT operation - AFTER

Wii's abilities are inserted.

The screenshot shows the same web application after the "WII" character has been successfully inserted. The character database table now includes a new row for "WII" with attacks "A", "B", "C", and "D". The "CHARACTER NAME" column is highlighted with a red box. To the right of the table, the browser's developer tools are open, showing the "Console" tab with several error messages related to DevTools failing to load source maps due to network issues. The "Network" tab shows a successful "POST" request to the "/abilities" endpoint with a status of 200 OK. The "Sources" tab shows the file "content.js" with some code and a warning about React 18. The bottom of the screen features three navigation links: "GAME PLAYER", "GAME TOURNAMENT", and "GAMES PLAYED".

2a) DELETE operation - BEFORE

Tournaments table before deletion. There are red delete buttons on the table to delete tuples.

Tournaments						
Tournament ID	Tournament Name	Organizer	Start Date	End Date	Prize Pool	
101	ESL ONE HAMBURG 2015	ESL GAMING	2015-11-16	2015-11-17	155000	<button>DELETE</button>
102	GET ON MY LINE 2016	EVEN MATCHUP GAMING	2016-06-25	2016-06-26	10000	<button>DELETE</button>
103	LOW TIER CITY 8	TOURNEY LOCATOR	2017-07-29	2017-07-31	10000	<button>DELETE</button>
104	SMASH ULTIMATE WORLD TOUR 2018	NINTENDO	2018-03-02	2018-12-08	250000	<button>DELETE</button>
105	SMASH WORLDS 2019	RIOT GAMES	2019-10-02	2019-11-10	2500000	<button>DELETE</button>
106	ESL ONE LA 2020	ESL GAMING	2020-03-28	2020-04-19	400000	<button>DELETE</button>
107	EVO 2020 ONLINE	EVO	2020-07-04	2020-08-02	39000	<button>DELETE</button>
108	ASIA CHAMPIONSHIPS 2021	PERFECT WORLD	2021-04-01	2021-04-04	500000	<button>DELETE</button>
109	PUBG GLOBAL INVITATIONAL 2021	PUBG CORPORATION	2021-02-05	2021-03-28	7000000	<button>DELETE</button>
110	SMASH ULTIMATE SUMMIT 3	BEYOND THE SUMMIT	2021-05-06	2021-05-09	150000	<button>DELETE</button>
111	SMASH CHAMPIONSHIPS 2021	ACTIVISION BLIZZARD	2021-08-19	2021-08-22	2500000	<button>DELETE</button>
112	RAINBOW SIX SIEGE INVITATIONAL 2022	UBISOFT	2022-02-09	2022-02-20	3000000	<button>DELETE</button>
113	SMASH WORLD CUP 2022	EPIC GAMES	2022-07-22	2022-07-24	50000000	<button>DELETE</button>
114	THE INTERNATIONAL 2023	VALVE CORPORATION	2023-08-18	2023-08-28	30000000	<button>DELETE</button>
115	ESL AMSTERDAM 2023	VALVE CORPORATION	2024-01-19	2024-01-29	2000000	<button>DELETE</button>

2b) DELETE operation - DURING

Hover over delete button for a specific tuple, brown color indicates feedback of hovering over the tuple.

TOURNAMENTS						
Tournament ID	Tournament Name	Organizer	Start Date	End Date	Prize Pool	
101	ESL ONE HAMBURG 2015	ESL GAMING	2015-11-16	2015-11-17	155000	<button>DELETE</button>
102	GET ON MY LINE 2016	EVEN MATCHUP GAMING	2016-06-25	2016-06-26	10000	<button>DELETE</button>
103	LOW TIER CITY 8	TOURNEY LOCATOR	2017-07-29	2017-07-31	10000	<button>DELETE</button>
104	SMASH ULTIMATE WORLD TOUR 2018	NINTENDO	2018-03-02	2018-12-08	250000	<button>DELETE</button>
105	SMASH WORLDS 2019	RIOT GAMES	2019-10-02	2019-11-10	2500000	<button>DELETE</button>
106	ESL ONE LA 2020	ESL GAMING	2020-03-28	2020-04-18	400000	<button>DELETE</button>
107	EVO 2020 ONLINE	EVO	2020-07-04	2020-08-02	39000	<button>DELETE</button>
108	ASIA CHAMPIONSHIPS 2021	PERFECT WORLD	2021-04-01	2021-04-04	500000	<button>DELETE</button>
109	PUBG GLOBAL INVITATIONAL 2021	PUBG CORPORATION	2021-02-05	2021-03-28	7000000	<button>DELETE</button>
110	SMASH ULTIMATE SUMMIT 3	BEYOND THE SUMMIT	2021-05-06	2021-05-09	150000	<button>DELETE</button>
111	SMASH CHAMPIONSHIPS 2021	ACTIVISION BLIZZARD	2021-08-19	2021-08-22	2500000	<button>DELETE</button>
112	RAINBOW SIX SIEGE INVITATIONAL 2022	UBISOFT	2022-02-09	2022-02-20	3000000	<button>DELETE</button>
113	SMASH WORLD CUP 2022	EPIC GAMES	2022-07-22	2022-07-24	50000000	<button>DELETE</button>
114	THE INTERNATIONAL 2023	VALVE CORPORATION	2023-08-18	2023-08-28	30000000	<button>DELETE</button>
115	ESL AMSTERDAM 2023	VALVE CORPORATION	2024-01-19	2024-01-29	2000000	<button>DELETE</button>

2c) DELETE operation - AFTER

Delete button pressed and tuple is deleted.

Tournament ID	Tournament Name	Organizer	Start Date	End Date	Prize Pool	
102	GET ON MY LINE 2016	EVEN MATCHUP GAMING	2016-06-25	2016-06-26	10000	<button style="background-color: red; color: white; padding: 2px 5px;">DELETE</button>
103	LOW TIER CITY 8	TOURNAMENT LOCATOR	2017-07-29	2017-07-31	10000	<button style="background-color: red; color: white; padding: 2px 5px;">DELETE</button>
104	SMASH ULTIMATE WORLD TOUR 2018	NINTENDO	2018-03-02	2018-12-08	250000	<button style="background-color: red; color: white; padding: 2px 5px;">DELETE</button>
105	SMASH WORLDS 2019	RIOT GAMES	2019-10-02	2019-11-10	2500000	<button style="background-color: red; color: white; padding: 2px 5px;">DELETE</button>
106	ESL ONE LA 2020	ESL GAMING	2020-03-28	2020-04-19	400000	<button style="background-color: red; color: white; padding: 2px 5px;">DELETE</button>
107	EVO 2020 ONLINE	EVO	2020-07-04	2020-08-02	39000	<button style="background-color: red; color: white; padding: 2px 5px;">DELETE</button>
108	ASIA CHAMPIONSHIPS 2021	PERFECT WORLD	2021-04-01	2021-04-04	500000	<button style="background-color: red; color: white; padding: 2px 5px;">DELETE</button>
109	PUBG GLOBAL INVITATIONAL 2021	PUBG CORPORATION	2021-02-05	2021-03-28	7000000	<button style="background-color: red; color: white; padding: 2px 5px;">DELETE</button>
110	SMASH ULTIMATE SUMMIT 3	BEYOND THE SUMMIT	2021-05-06	2021-05-09	150000	<button style="background-color: red; color: white; padding: 2px 5px;">DELETE</button>
111	SMASH CHAMPIONSHIPS 2021	ACTIVISION BLIZZARD	2021-08-19	2021-08-22	2500000	<button style="background-color: red; color: white; padding: 2px 5px;">DELETE</button>
112	RAINBOW SIX SIEGE INVITATIONAL 2022	UBISOFT	2022-02-09	2022-02-20	3000000	<button style="background-color: red; color: white; padding: 2px 5px;">DELETE</button>
113	SMASH WORLD CUP 2022	EPIC GAMES	2022-07-22	2022-07-24	5000000	<button style="background-color: red; color: white; padding: 2px 5px;">DELETE</button>
114	THE INTERNATIONAL 2023	VALVE CORPORATION	2023-08-18	2023-08-28	30000000	<button style="background-color: red; color: white; padding: 2px 5px;">DELETE</button>
115	ESL AMSTERDAM 2023	VALVE CORPORATION	2024-01-19	2024-01-29	2000000	<button style="background-color: red; color: white; padding: 2px 5px;">DELETE</button>

DELETE SUCCESSFUL

```

DELETE FROM Tournament WHERE tournament_id="101"
Tournament.js:43
    
```

Note that this will also delete games that are associated with this tournament in Game_Tournament.

Game_Tournament Before Delete

GAME ID	TOURNAMENT ID
1	101
2	101
3	102
4	103
5	103
6	103
7	103
8	104
9	105
10	105
11	105
12	106
13	106
14	107
15	108
16	108
17	109
18	109
19	109
20	110
21	111
22	112
23	112
24	112
25	112

Game_Tournament After Delete

GAME ID	TOURNAMENT ID
3	102
4	103
5	103
6	103
7	103
8	104
9	105
10	

3a) UPDATE operation - BEFORE

Players table before update operation.

PLAYERS PLAYER SELECTION						
USERNAME: <input type="text"/>	COUNTRY: <input type="text"/>	RANKING LEVEL: <input type="text"/>				
AGE: <input type="text"/>	WINS: <input type="text"/>	LOSSES: <input type="text"/>	SEARCH	UPDATE		
<input type="checkbox"/> USERNAME	<input type="checkbox"/> COUNTRY	<input type="checkbox"/> RANKING LEVEL	<input type="checkbox"/> AGE	<input type="checkbox"/> WINS	<input type="checkbox"/> LOSSES	<input type="button" value="SUBMIT"/>
USERNAME	COUNTRY	RANKING LEVEL	AGE	WINS	LOSSES	
ADAM	CHINA	GOLD	16	49	6	
ANYWEHE	SOUTH KOREA	PLATINUM	21	64	12	
CHRISTIAN	PHILIPPINES	BRONZE	26	23	5	
CYBERKNIGHT	CANADA	BRONZE	28	29	7	
ELITEGAMINGPRO	UK	GOLD	22	45	9	
EPICGAMER88	AUSTRALIA	BEGINNER	12	17	11	
FAKER	SOUTH KOREA	PLATINUM	20	54	2	
GAMEON92	USA	BEGINNER	40	15	8	
HAUNTZER	PHILIPPINES	GOLD	27	45	46	
NINJA	USA	PLATINUM	20	53	5	
NINJAGAMER	CANADA	BRONZE	29	20	5	
OVERTAKER	CANADA	PLATINUM	22	59	5	
PEWDIEPIE	SWEDEN	PLATINUM	20	60	12	
PIXELWARRIOR	UK	BEGINNER	13	14	10	
RETROGAMERX	AUSTRALIA	SILVER	35	33	12	
SALLY	INDIA	BRONZE	17	23	6	
SHROUD	CANADA	SILVER	20	33	8	
SUMMIT1G	USA	DIAMOND	18	101	3	
THEGAMINGGEEK	USA	BEGINNER	15	18	6	
THUNDERBOLT	USA	SILVER	31	34	4	
VICTORY	AUSTRALIA	SILVER	30	35	9	
VIRTUALASSASSIN	CANADA	BRONZE	16	22	3	
ZAHGRD	AUSTRALIA	BEGINNER	14	5	86	

3b) UPDATE operation - DURING

User enters information to update for Adam.

The screenshot shows a web-based application for managing player statistics. At the top, there's a header with the word 'PLAYERS' and 'PLAYER SELECTION'. Below the header is a search bar with fields for 'USERNAME' (ADAM), 'COUNTRY' (USA), 'RANKING LEVEL' (SILVER), 'AGE' (20), 'WINS' (4), and 'LOSSES' (4). There are 'SEARCH' and 'UPDATE' buttons. Below the search bar is a row of checkboxes for filtering results by 'USERNAME', 'COUNTRY', 'RANKING LEVEL', 'AGE', 'WINS', and 'LOSSES', followed by a 'SUBMIT' button. A red box highlights the search bar and filter row. The main area contains a table with columns: USERNAME, COUNTRY, RANKING LEVEL, AGE, WINS, and LOSSES. The table lists various players with their respective details. The row for 'ADAM' is highlighted in pink.

USERNAME	COUNTRY	RANKING LEVEL	AGE	WINS	LOSSES
ADAM	CHINA	GOLD	16	49	6
ANYWEHE	SOUTH KOREA	PLATINUM	21	64	12
CHRISTIAN	PHILIPPINES	BRONZE	26	23	5
CHUNGUS	USA	DIAMOND	20	69	40
CYBERKNIGHT	CANADA	BRONZE	28	29	7
ELITEGAMINGPRO	UK	GOLD	22	45	9
EPICGAMER88	AUSTRALIA	BEGINNER	12	17	11
FAKER	SOUTH KOREA	DIAMOND	20	54	2
GAMEONS2	USA	BEGINNER	40	15	8
HAUNTER	PHILIPPINES	GOLD	27	45	46
HUNGRY	BRAZIL	SILVER	20	5	13
NINJA	USA	PLATINUM	50	53	5
NINJAGAMER	CANADA	BRONZE	29	20	5
OVERTAKER	CANADA	PLATINUM	22	59	5
PENDIEPIE	SWEDEN	PLATINUM	35	60	12
PIXELWARRIOR	UK	BEGINNER	13	14	10
RETROGAMERX	AUSTRALIA	SILVER	35	33	12
SALLY	INDIA	BRONZE	17	23	6
SHROUD	CANADA	SILVER	33	33	8
SUMMITIG	USA	DIAMOND	18	101	3
THEGAMINGEEK	USA	BEGINNER	15	18	6
THUNDERBOLT	USA	SILVER	31	34	4
VICTORY	AUSTRALIA	SILVER	30	35	9
VIRTUALASSASSIN	CANADA	BRONZE	16	22	3
XQC	CANADA	DIAMOND	20	100	0
ZAHGRD	AUSTRALIA	BEGINNER	14	5	96

3c) UPDATE operation - AFTER

User clicks update, Adam's information is updated.

The screenshot shows a web application interface for 'PLAYERS' with a header 'CPSC 304 GROUP 39' and a sub-header 'VANESSA LEE, ISAAC CHANG, KITTY LIU'. A green success message 'UPDATE SUCCESSFUL!' is displayed at the top right. Below it is a search form with fields for 'USERNAME' (ADAM), 'COUNTRY' (USA), 'RANKING LEVEL' (SILVER), 'AGE' (20), 'WINS' (4), and 'LOSSES' (4). There are 'SEARCH' and 'UPDATE' buttons. Below the form is a table of player data. The row for 'ADAM' (USA, SILVER, 20, 4, 4) is highlighted with a red border. In the browser's developer tools (Console tab), several errors are listed, including 'DevTools failed to load source map: Could not load content for chrome-extension://hnmankncofebbddgclmhnfnkdnad/requestProvider.js.map: System error: net::ERR_BLOCKED_BY_CLIENT' and a warning about React DOM render being deprecated. The bottom of the developer tools shows a log entry for an 'UPDATE' query:

```
Query: UPDATE Player SET username='Adam', country='USA', ranking_level='Silver', age=20, wins=4, losses=4 WHERE username='Adam'
```

4a) Selection query - BEFORE

Players table before selection query.



PLAYER SELECTION

USERNAME: <input type="text"/>	COUNTRY: <input type="text"/>	RANKING LEVEL: <input type="text"/>
AGE: <input type="text"/>	WINS: <input type="text"/>	LOSSES: <input type="text"/>
<input type="button" value="SEARCH"/> <input type="button" value="UPDATE"/>		
<input type="checkbox"/> USERNAME <input type="checkbox"/> COUNTRY <input type="checkbox"/> RANKING LEVEL <input type="checkbox"/> AGE <input type="checkbox"/> WINS <input type="checkbox"/> LOSSES <input type="button" value="SUBMIT"/>		

USERNAME	COUNTRY	RANKING LEVEL	AGE	WINS	LOSSES
ADAM	CHINA	GOLD	16	49	6
ANYWEHE	SOUTH KOREA	PLATINUM	21	64	12
CHRISTIAN	PHILIPPINES	BRONZE	26	23	5
CYBERKNIGHT	CANADA	BRONZE	28	29	7
ELITEGAMINGPRO	UK	GOLD	22	45	9
EPICGAMER88	AUSTRALIA	BEGINNER	12	17	11
FAKER	INDIA	BRONZE	4	4	4
GAMEON92	USA	BEGINNER	40	15	8
HAUNTZER	PHILIPPINES	GOLD	27	45	46
NINJA	USA	PLATINUM	20	53	5
NINJAGAMER	CANADA	BRONZE	29	20	5
OVERTAKER	CANADA	PLATINUM	22	59	5
PEWDIEPIE	SWEDEN	PLATINUM	20	60	12
PIXELWARRIOR	UK	BEGINNER	13	14	10
RETROGAMERX	AUSTRALIA	SILVER	35	33	12
SALLY	JAPAN	SILVER	18	30	8
SHROUD	CANADA	SILVER	20	33	8
SUMMIT1G	USA	DIAMOND	18	101	3
THEGAMINGGEEK	USA	BEGINNER	15	18	6
THUNDERBOLT	USA	SILVER	31	34	4
VICTORY	AUSTRALIA	SILVER	30	35	9
VIRTUALASSASSIN	CANADA	BRONZE	16	22	3
ZAHGRD	AUSTRALIA	BEGINNER	14	5	86

4b) Selection query- DURING

Search for players named Adam.

The screenshot shows a web application titled "PLAYERS" with a sub-section "PLAYER SELECTION". At the top, there are input fields for "USERNAME" (containing "ADAM"), "COUNTRY", and "RANKING LEVEL". Below these are fields for "AGE", "WINS", and "LOSSES", each with a corresponding input field and a checkbox. There are also "SEARCH" and "UPDATE" buttons. Below the search bar is a row of checkboxes labeled "USERNAME", "COUNTRY", "RANKING LEVEL", "AGE", "WINS", "LOSSES", and "SUBMIT". The main area displays a table of player statistics:

USERNAME	COUNTRY	RANKING LEVEL	AGE	WINS	LOSSES
ADAM	CHINA	GOLD	16	49	6
ANYWHERE	SOUTH KOREA	PLATINUM	21	64	12
CHRISTIAN	PHILIPPINES	BRONZE	26	23	5
CYBERKNIGHT	CANADA	BRONZE	28	29	7
ELITEGAMINGPRO	UK	GOLD	22	45	9
EPICGAMER88	AUSTRALIA	BEGINNER	12	17	11
FAKER	INDIA	BRONZE	4	4	4
GAMEON82	USA	BEGINNER	40	15	8
HAUNTER	PHILIPPINES	GOLD	27	45	46
NINJA	USA	PLATINUM	20	53	5
NINJAGAMER	CANADA	BRONZE	29	20	5
OVERTAKER	CANADA	PLATINUM	22	59	5
PENDIEPIE	SWEDEN	PLATINUM	20	60	12
PIXELWARRIOR	UK	BEGINNER	13	14	10
RETROGAMERX	AUSTRALIA	SILVER	35	33	12
SALLY	JAPAN	SILVER	18	30	8
SHROUD	CANADA	SILVER	20	33	8
SUMMIT1G	USA	DIAMOND	18	101	3
THEGAMINGEEK	USA	BEGINNER	15	18	6
THUNDERBOLT	USA	SILVER	31	34	4
VICTORY	AUSTRALIA	SILVER	30	35	9
VIRTUALASSASSIN	CANADA	BRONZE	16	22	3
ZAHGRD	AUSTRALIA	BEGINNER	14	5	86

4c) Selection query- AFTER

Click Search to view Adam and their statistics.

The screenshot shows the same "PLAYERS" application interface. The "USERNAME" field now contains "ADAM". The "SEARCH" button is highlighted in red. To the right of the search bar, a message says "Download the React DevTools for a better development experience: <https://reactjs.org/link/react-devtools>". Below this, a warning message states: "Warning: ReactDOM.render react-dom.development.js:86 is no longer supported in React 18. Use createRoot instead. Until you switch to the new API, your app will behave as if it's running React 17. Learn more: https://reactjs.org/link/switch-to-createRoot". Further down, an error message indicates: "DevTools failed to load source map: Could not load content for chrome-extension://gighmmiobioklfepjocnamgkbbilidom/browser-polyfill.js.map: System error: net::ERR_FILE_NOT_FOUND". A red box highlights the command "Query: SELECT * FROM Player WHERE username = 'adam'" in the developer tools log, followed by "Player.js:108".

USERNAME	COUNTRY	RANKING LEVEL	AGE	WINS	LOSSES
ADAM	CHINA	GOLD	16	49	6

5a) Projection query - BEFORE

Characters table before projection query.

PLAYERS PLAYER SELECTION						
USERNAME:	COUNTRY:	RANKING LEVEL:	AGE:	WINS:	LOSSES:	SEARCH
<input type="checkbox"/> USERNAME	<input type="checkbox"/> COUNTRY	<input type="checkbox"/> RANKING LEVEL	<input type="checkbox"/> AGE	<input type="checkbox"/> WINS	<input type="checkbox"/> LOSSES	SUBMIT
USERNAME	COUNTRY	RANKING LEVEL	AGE	WINS	LOSSES	
ADAM	CHINA	GOLD	16	49	6	
ANYWEHE	SOUTH KOREA	PLATINUM	21	64	12	
CHRISTIAN	PHILIPPINES	BRONZE	26	23	5	
CYBERKNIGHT	CANADA	BRONZE	28	29	7	
ELITEGAMINGPRO	UK	GOLD	22	45	9	
EPICGAMER88	AUSTRALIA	BEGINNER	12	17	11	
FAKER	SOUTH KOREA	PLATINUM	20	54	2	
GAMEON92	USA	BEGINNER	40	15	8	
HAUNTZER	PHILIPPINES	GOLD	27	45	46	
NINJA	USA	PLATINUM	20	53	5	
NINJAGAMER	CANADA	BRONZE	29	20	5	
OVERTAKER	CANADA	PLATINUM	22	59	5	
PEWDIEPIE	SWEDEN	PLATINUM	20	60	12	
PIXELWARRIOR	UK	BEGINNER	13	14	10	
RETROGAMERX	AUSTRALIA	SILVER	35	33	12	
SALLY	INDIA	BRONZE	17	23	6	
SHROUD	CANADA	SILVER	20	33	8	
SUMMIT1G	USA	DIAMOND	18	101	3	
THEGAMINGGEEK	USA	BEGINNER	15	18	6	
THUNDERBOLT	USA	SILVER	31	34	4	
VICTORY	AUSTRALIA	SILVER	30	35	9	
VIRTUALASSASSIN	CANADA	BRONZE	16	22	3	
ZAHGRD	AUSTRALIA	BEGINNER	14	5	86	

5b) Projection query - DURING

User checks columns they want to see and clicks submit. In this example, user only wants to see username and country.

PLAYERS
PLAYER SELECTION

USERNAME: COUNTRY: RANKING LEVEL:
AGE: WINS: LOSSES:

USERNAME COUNTRY RANKING LEVEL AGE WINS LOSSES

USERNAME	COUNTRY	RANKING LEVEL	AGE	WINS	LOSSES
ADAM	CHINA	GOLD	16	49	6
ANYWEHE	SOUTH KOREA	PLATINUM	21	64	12
CHRISTIAN	PHILIPPINES	BRONZE	26	23	5
CYBERKNIGHT	CANADA	BRONZE	28	29	7
ELITEGAMINGPRO	UK	GOLD	22	45	9
EPICGAMER88	AUSTRALIA	BEGINNER	12	17	11
FAKER	SOUTH KOREA	PLATINUM	20	54	2
GAMEON92	USA	BEGINNER	40	15	8
HAUNTZER	PHILIPPINES	GOLD	27	45	46
NINJA	USA	PLATINUM	20	53	5
NINJAGAMER	CANADA	BRONZE	29	20	5
OVERTAKER	CANADA	PLATINUM	22	59	5
PEWDIEPIE	SWEDEN	PLATINUM	20	60	12
PIXELWARRIOR	UK	BEGINNER	13	14	10
RETROGAMERX	AUSTRALIA	SILVER	35	33	12
SALLY	JAPAN	SILVER	18	30	8
SHROUD	CANADA	SILVER	20	33	8
SUMMIT1G	USA	DIAMOND	18	101	3
THEGAMINGGEEK	USA	BEGINNER	15	18	6
THUNDERBOLT	USA	SILVER	31	34	4
VICTORY	AUSTRALIA	SILVER	30	35	9
VIRTUALASSASSIN	CANADA	BRONZE	16	22	3
ZAHGRD	AUSTRALIA	BEGINNER	14	5	86

5c) Projection query - AFTER

User clicks the “update” button to only view the chosen columns.

The screenshot shows a web application titled "PLAYERS" with a "PLAYER SELECTION" section. It includes input fields for "USERNAME", "COUNTRY", "RANKING LEVEL", "AGE", "WINS", and "LOSSES", along with "SEARCH" and "UPDATE" buttons. Below this is a table with columns: USERNAME, COUNTRY, RANKING LEVEL, AGE, WINS, and LOSSES. The table contains 20 rows of player data. To the right of the table, a developer console window is open, displaying a series of error messages from the "Player.js" file, indicating network requests and their responses.

USERNAME	COUNTRY	RANKING LEVEL	AGE	WINS	LOSSES
ADAM	CHINA				
ANYWHERE	SOUTH KOREA				
CHRISTIAN	PHILIPPINES				
CYBERKNIGHT	CANADA				
ELTEGAMINGPRO	UK				
EPICOGAMER88	AUSTRALIA				
FAKER	SOUTH KOREA				
GAMEON82	USA				
HAUNTER	PHILIPPINES				
NINJA	USA				
NINJAGAMER	CANADA				
OVERTAKER	CANADA				
PENDIEPIE	SWEDEN				
PIXELWARRIOR	UK				
RETROGAMERX	AUSTRALIA				
SALLY	JAPAN				
SHROUD	CANADA				
SUMMIT19	USA				
THEGAMINGGEEK	USA				
THUNDERBOLT	USA				
VICTORY	AUSTRALIA				
VIRTUALASSASSIN	CANADA				
ZAHGRD	AUSTRALIA				

6a) JOIN - BEFORE

Joining Player and Game_Player to find the number of games a player has played.

The screenshot shows a web application titled "GAMES PLAYED" with a "SELECT A USER:" dropdown menu. Below it is a table with two columns: "USERNAME" and "NUMBER OF GAMES PLAYED". The "NUMBER OF GAMES PLAYED" column is currently empty, indicating no data has been selected or loaded.

USERNAME	NUMBER OF GAMES PLAYED
~ SELECT A USER ~	

6b) JOIN - DURING

Click drop down to choose a player.

The screenshot shows a user interface for a game statistics application. At the top, there are three tabs: "USERNAME", "NUMBER OF GAMES PLAYED", and "WINS". Below these tabs are two dropdown menus labeled "SELECT AN ATTRIBUTE" and "SELECT A USER". The "SELECT A USER" dropdown is open, displaying a list of player names. The name "Overtaker" is highlighted in blue, indicating it is the selected value. The list includes: -- Select a user --, Adam, Anywehe, Christian, CyberKnight, EliteGamingPro, EpicGamer88, Faker, GameOn92, Hauntzer, Ninja, NinjaGamer, Overtaker, PewDiePie, PixelWarrior, RetroGamerX, Sally, Shroud, Summit1g, and TheGamingGeek.

6c) JOIN - AFTER

Clicked on a player and returns the number of games they have played.

The screenshot shows the same application interface after a selection has been made. The "SELECT A USER" dropdown now shows "OVERTAKER" as the selected value. The "WINS BY ATTRIBUTE" section below it contains a single dropdown menu labeled "SELECT AN ATTRIBUTE" with the option "-- SELECT A USER --". To the right of the application, a red-bordered box highlights a portion of the browser's developer tools' console output. The console shows several error messages related to network requests, followed by a successful SQL query:

```
SELECT Player.username,
       COUNT(Game_Player.game_id) AS num_games_played
  FROM Player
  JOIN Game_Player ON Player.username =
    Game_Player.username
 WHERE Player.username = "Overtaker"
 GROUP BY Player.username
```

7a) Aggregation with GROUP BY - BEFORE

Find the total number of wins based on a player's information.

WINS BY ATTRIBUTE

SELECT AN ATTRIBUTE:

-- SELECT A USER -- ▾

TOTAL WINS

7b) Aggregation with GROUP BY - DURING

Click dropdown to either group by a player's country, ranking level, or age to see their wins.
Click country.

WINS BY ATTRIBUTE

SELECT AN ATTRIBUTE:

✓ -- Select a user --

Country

Ranking Level

Age

WINS

TEAMS BY ATTRIBUTE

7c) Aggregation with GROUP BY - AFTER

Displays countries and their total wins.

COUNTRY

TOTAL WINS

CHINA	49
SOUTH KOREA	19
PHILIPPINES	68
CANADA	163
UK	59
AUSTRALIA	90
USA	221
SWEDEN	60
JAPAN	30

```
at fetchData (Wins.js:52:1)
④ > GET http://localhost:3001/api/teams/ 404 (Not Found) Teams.js:17 ↵
④ > Error: Error: Network response was not ok Teams.js:25
      at fetchData (Teams.js:19:1)
④ > GET http://localhost:3001/api/avg-age/ 404 (Not Found) AvgAge.js:17 ↵
④ > Error: Error: Network response was not ok AvgAge.js:25
      at fetchData (AvgAge.js:19:1)
④ > Warning: Each child react-jsx-dev-runtime.development.js:87
      in a list should have a unique "key" prop.
Check the render method of 'WinsTable'. See https://reactjs.org/
/guide/warning-keys for more information.
      at 0 (http://localhost:3000/static/js/bundle.js:43297:8)
      at WinsTable (http://localhost:3000/static/js/bundle.js:19
81:5)
      at Wins (http://localhost:3000/static/js/bundle.js:3939:7)
4)   at section
      at 0 (http://localhost:3000/static/js/bundle.js:43297:8)
      at App
SELECT country, SUM(wins) as total_wins
  FROM Player
  GROUP BY country
Wins.js:56
```

TEAMS BY ATTRIBUTE

NUMBER OF PLAYERS

AVERAGE AGE BY ATTRIBUTE

8a) Aggregation with HAVING - BEFORE

To view number of players with a chosen attribute that has more than 2 players.

The screenshot shows a user interface for a game application. At the top, there's a logo of a character holding a sword and shield, followed by the text 'TEAMS BY ATTRIBUTE'. Below this is a section titled 'SELECT AN ATTRIBUTE:' with a dropdown menu. The dropdown currently displays the placeholder text '-- Select an attribute --'. To the right of the dropdown, there's a green horizontal bar with the text 'NUMBER OF PLAYERS' in white. The background is light blue with some faint cloud-like patterns.

8b) Aggregation with HAVING - DURING

User clicks dropdown to choose attribute, such as country.

This screenshot shows the same application after a selection has been made. The 'SELECT AN ATTRIBUTE:' dropdown now has a checkmark next to 'Country', indicating it is the selected attribute. The green bar below the dropdown now displays the word 'PLAYERS' in large, bold, black letters. The rest of the interface remains the same, with the character logo and the 'TEAMS BY ATTRIBUTE' title at the top.

8c) Aggregation with HAVING - AFTER

We can see the number of players from each country, which have more than 2 players.

This screenshot shows the final state of the application where the data has been aggregated. On the left, there's a table with two columns: 'COUNTRY' and 'NUMBER OF PLAYERS'. The table contains three rows: USA (6), CANADA (6), and AUSTRALIA (4). On the right side of the screen, a red box highlights a portion of the application's code. The code is a SQL query written in a language like JavaScript or TypeScript. It selects 'country' and 'COUNT(username)' as 'num_players' from the 'Player' table, groups the results by 'country', applies a 'HAVING' clause to filter rows where 'num_players' is greater than 2, and finally orders the results by 'num_players' in descending order. The code is annotated with line numbers: 5:8, 5:88, and 5:89, and includes a note 'Teams.js:79'.

```
5:8) at App (http://localhost:3000/static/js/bundle.js:12)
5:88)
5:89)
SELECT country, COUNT(username) AS num_players
FROM Player
GROUP BY country
HAVING num_players > 2
ORDER BY num_players DESC
```

9a) Nested Aggregation with GROUP BY - BEFORE

Retrieves the average age of players for an attribute, but only for attributes that have more than one player.

The screenshot shows a web application titled "AVERAGE AGE BY ATTRIBUTE". At the top, there is a logo of a cartoon character and the title. Below it is a section labeled "SELECT AN ATTRIBUTE" with a dropdown menu. The dropdown menu is currently open, showing four options: "Country", "Ranking Level", "Wins", and "Losses". A large blue bar at the bottom of the screen is labeled "AVERAGE AGE".

9b) Nested Aggregation with GROUP BY - DURING

User selects country to see the average of age of players in that country, granting that country has more than 1 player.

The screenshot shows the same application after the user has selected "Country" from the dropdown menu. The dropdown menu is now closed, and the "Country" option is highlighted. The large blue bar at the bottom is labeled "AGE".

9c) Nested Aggregation with GROUP BY - AFTER

User can view the average of age of players in that country, granting that country has more than 1 player.

The screenshot shows the final state of the application. A table titled "COUNTRY" and "AVERAGE AGE" is displayed, showing the average age for five countries: Philippines (26.5000), Canada (23.0000), UK (17.5000), Australia (22.7500), and USA (24.8000). To the right of the table, a red box highlights a warning message from the React.js developer tools: "Each child in a list should have a unique "key" prop. Check the render method of 'AvgAgeTable'. See https://reactjs.org/link/warning-keys for more information." Below the table, a footer bar is visible with the text "PLAYERS WHO PLAYED EVERY GAME IN A TOURNAMENT".

```
SELECT p1.country, Avg(p1.age) as avg_age
FROM Player p1
GROUP BY p1.country
HAVING 1 < (SELECT COUNT(*)
              FROM Player p2
              WHERE p1.country = p2.country)
```

10a) Division query- BEFORE

Selects the usernames of players that have played in every game of a specified tournament.

The screenshot shows a web page with a light blue background featuring a cartoon character at the top left. The main title is "PLAYERS WHO PLAYED EVERY GAME IN A TOURNAMENT". Below it is a dropdown menu labeled "SELECT A TOURNAMENT" with the placeholder "-- SELECT A TOURNAMENT --". A yellow horizontal bar across the middle contains the word "USERNAME".

10b) Division query - DURING

User wants to view all players who played in tournament 112.

The screenshot shows the same web page as above, but the dropdown menu is now open, displaying a list of tournament IDs from 103 to 113. The option "112" is highlighted with a blue selection bar, indicating it has been chosen.

10c) Division query - AFTER

User views all players who played in tournament 112.

The screenshot shows the results of the query for tournament 112. The "USERNAME" field contains the value "NINJA" and is highlighted with a red border. To the right, a red box contains the SQL code:

```
SELECT username
FROM Player
WHERE NOT EXISTS (
    SELECT game_id
    FROM Game_Tournament
    WHERE tournament_id = 112
    EXCEPT
    SELECT game_id
    FROM Game_Player
    WHERE Game_Player.username =
        Player.username)
```