University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project

Milestone #: 4
Date: April 4th 2023

Group Number: 48

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Eric Leung	47358213	leung147	ericleung147258369@hotmail.com
Anna Wang	59754044	annaw245	annatqwang@gmail.com
Chris Lee	59422501	clee422	leechris422@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 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

University of British Columbia, Vancouver
Department of Computer Science

Project Summary

This database models characteristics of events and players in the competitive Valorant esports scene. This database could be used by coaches, fans, and analysts to analyse and predict future outcomes. People using the database will be able to query for and view previous events and the organisations participating in them.

Based on our current implementation of the project, we are able to:

1. Check for the average damage per round for all players
2. Find player's in-game-name with kills above a specified threshold
3. Check the matches for which two players have played in together
4. See the headshot percentage by weapon of an inputted player in-game-name
5. Show all the attributes currently under Organizations
6. Add new organizations into the database
7. Delete matches from the database
8. Update winrates of specified organizations
9. Show the region that has the highest win rate, indicated as leading region
10. Select a table and its attributes to output a table with those targeted attributes

Changed Schema

We took out the Contract table and attributes under agentPlayed because they were never used. AvgCombatScore represented some of the attributes that we wanted to implement into agentPlayed but we ended up taking it out. We were able to instead find some similar stats based off of our weaponsUsed table such as kills, which were the most important for our needs.

Copy of the Schema and Data Screenshots

```
CREATE TABLE OrganizationID (
      o_id INTEGER,
      name CHAR(255) NOT NULL UNIQUE,
      PRIMARY KEY (o_id)
);
```

```
SET PAGESIZE 1000
COLUMN o_id FORMAT 999
COLUMN name FORMAT A20
SELECT o_id, name FROM OrganizationID;

O_ID|NAME
----|--------------------
   1|Sentinels

   2|LOUD

   3|Fnatic

   4|Paper Rex

   5|NRG
```

```
CREATE TABLE Organization (
      name CHAR(255) NOT NULL PRIMARY KEY,
      ranking INTEGER UNIQUE,
      region CHAR(255) NOT NULL,
      win_rate REAL
);
```

```
SET LINESIZE 200
SET PAGESIZE 1000
SET COLSEP '|'
COLUMN name FORMAT A20
COLUMN ranking FORMAT 999
COLUMN region FORMAT A20
COLUMN win_rate FORMAT 999 .99
SELECT name, ranking, region, win_rate FROM organization;

NAME                |RANKING|REGION              |WIN_RATE
--------------------|-------|--------------------|--------
Sentinels           |     37|Americas            |     .48
                    |       |                    |

LOUD                |      2|Americas            |     .87
                    |       |                    |

Fnatic              |      6|EMEA                |     .74
                    |       |                    |

Paper Rex           |      4|Pacific             |     .78
                    |       |                    |

NRG                 |     31|Americas            |     .56
                    |       |                    |
```

```
CREATE TABLE HistoricalMatchup (
      o_id1 INTEGER,
      o_id2 INTEGER,
      win_loss_ratio CHAR(255),
      PRIMARY KEY (o_id1, o_id2)
);
```

```
SELECT o_id1, o_id2, win_loss_ratio FROM HistoricalMatchup;


O_ID1|O_ID2|WIN_LOSS_RATIO
-----|-----|-------------------
   5|    2|0:1
    |     |

   1|    5|2:1
    |     |

   3|    4|0:1
    |     |

   3|    1|1:2
    |     |

   4|    5|0:0
    |     |
```

```
CREATE TABLE Event (
      e_id INTEGER PRIMARY KEY,
      name CHAR(255) NOT NULL,
      start_date DATE,
      end_date DATE,
      winning_organization INTEGER,
      prize_pool REAL NOT NULL,
      FOREIGN KEY (winning_organization) REFERENCES OrganizationID
(o_id));
```

```
SELECT e_id, name, start_date, end_date, winning_organization, prize_pool
FROM Event;
```

E_ID	NAME	START_DATE	END_DATE	WINNING_ORGANIZATION	PRIZE_POOL
1	VCT 2023: LOCK//IN S??o Paulo	13-FEB-23	04-MAR-23		500000.00
2	VCT 2021: Stage 2 Masters - Reykjav??k	24-MAY-21	30-MAY-21	1	600000.00
3	2022: Stage 2 Masters - Copenhagen	10-JUL-22	24-JUL-22		650000.00
4	VCT 2021: Brazil Stage 1 Masters	13-MAR-21	21-MAR-21	2	45538.00
5	VCT 2022: Game Changers Korea	22-SEP-22	23-SEP-22	4	7038.00

```
CREATE TABLE ParticipatingIn (
      o_id INTEGER,
      e_id INTEGER,
      final_placement INTEGER,
      PRIMARY KEY (o_id, e_id),
      FOREIGN KEY (o_id) REFERENCES OrganizationID (o_id) ON DELETE
CASCADE,
      FOREIGN KEY (e_id) REFERENCES Event (e_id) ON DELETE SET NULL
);
```

```
SELECT o_id, e_id, final_placement FROM ParticipatingIn;
```

O_ID	E_ID	FINAL_PLACEMENT
1	3	1
1	1	2
2	1	1
3	1	1
4	3	1

```
CREATE TABLE TeamMemberContract (
      tm_id INTEGER,
      in_game_name CHAR(255) NOT NULL UNIQUE,
      real_name CHAR(255) UNIQUE,
      start_date DATE,
      end_date DATE,
      salary INTEGER,
      o_id INTEGER,
      PRIMARY KEY (tm_id),
      FOREIGN KEY (o_id) REFERENCES OrganizationID (o_id) ON DELETE
SET NULL);
```

```
SELECT tm_id, in_game_name, real_name, sSQL> tart_date, end_date, salary, o_id
FROM TeamMemberContract;

TM_ID|IN_GAME_NAME        |REAL_NAME           |START_DAT|END_DATE | SALARY|O_ID
-----|--------------------|--------------------|---------|---------|-------|----
    1|TenZ                |Tyson Ngo           |01-JUN-20|         |       |   1
     |                    |                    |         |         |       |

    2|aspas               |Erick Santos        |01-JAN-22|         |       |   2
     |                    |                    |         |         |       |

    3|Enzo                |Enzo Mestari        |09-MAY-22|30-NOV-22|       |   3
     |                    |                    |         |         |       |

    4|SyykoNT             |Don Muir            |03-OCT-22|         |       |   1
     |                    |                    |         |         |       |

    5|s0m                 |Sam Oh              |07-OCT-20|         |       |   5
     |                    |                    |         |         |       |

    6|f0rsakeN            |Jason Susanto       |08-FEB-21|         |       |   4
     |                    |                    |         |         |       |
```

```
CREATE TABLE Player (
      tm_id INTEGER PRIMARY KEY,
      rank CHAR(255),
      role CHAR(255),
      FOREIGN KEY (tm_id) REFERENCES TeamMemberContract (tm_id) ON
DELETE CASCADE);
```

```
SELECT tm_id, rank, role FROM Player;

TM_ID|RANK               |ROLE
-----|-------------------|-------------------
   1|Radiant #1          |Duelist
    |                    |
    |                    |

   2|Radiant #1          |Duelist
    |                    |
    |                    |

   3|Radiant #220        |Initiator
    |                    |
    |                    |

   5|Radiant #24         |Controller
    |                    |
    |                    |

   6|Radiant #1          |Duelist
    |                    |
    |                    |
```

```
CREATE TABLE Weapon (
      weapon_name CHAR(255),
      damage INTEGER,
      PRIMARY KEY (weapon_name)
);
```

```
SELECT weapon_name, damage  FROM Weapon;

WEAPON_NAME         |DAMAGE
--------------------|------
Sheriff             |   159

Marshal             |   202

Vandal              |   160

Classic             |    78

Guardian            |   195
```

```
CREATE TABLE UsesWeapon (
      tm_id INTEGER,
      weapon_name CHAR(255),
      average_damage_per_round REAL,
      kills REAL,
      headshot_percentage REAL,
      PRIMARY KEY (tm_id, weapon_name),
      FOREIGN KEY (tm_id) REFERENCES TeamMemberContract (tm_id) ON
DELETE CASCADE,
      FOREIGN KEY (weapon_name) REFERENCES Weapon (weapon_name) ON
DELETE SET NULL
);
```

```
SELECT tm_id, weapon_name, average_damage_per_round, kills, headshot_percentage
FROM UsesWeapon;
```

TM_ID	WEAPON_NAME	AVERAGE_DAMAGE_PER_ROUND	KILLS	HEADSHOT_PERCENTAGE
1	Vandal	177.10	483.00	.36
1	Marshal	120.90	323.00	.25
1	Classic	28.30	4545.00	.34
2	Sheriff	88.40	444.00	.56
2	Vandal	175.90	.00	.46

```
CREATE TABLE StagedEvent (
      e_id INTEGER PRIMARY KEY,
      num_attendees INTEGER,
      venue_location CHAR(255) NOT NULL
);
```

```
SELECT e_id, num_attendees, venue_location
FROM StagedEvent;

  E_ID|NUM_ATTENDEES|VENUE_LOCATION
------|-------------|--------------------
     1|        10332|Gin??sio do Ibirapue
      |             |ra
      |             |

     2|         4215|Laugardalsh??ll
      |             |
      |             |

     3|         6922|Forum Copenhagen
      |             |
      |             |

     6|         2352|Marlene-Dietrich-Hal
      |             |le
      |             |

     8|         1702|Volkswagen Arena
      |             |
      |             |
```

```
CREATE TABLE OnlineEvent (
      e_id INTEGER PRIMARY KEY,
      num_viewers INTEGER,
      broadcast_platform CHAR(255) NOT NULL
);
```

```
SELECT e_id, num_viewers, broadcast_platform FROM OnlineEvent;

E_ID|NUM_VIEWERS|BROADCAST_PLATFORM
----|-----------|--------------------
   1|     372184|Twitch
    |           |
    |           |

   2|     488364|Twitch
    |           |
    |           |

   3|     317604|Twitch
    |           |
    |           |

   4|     372184|Twitch
    |           |
    |           |

   5|     372184|Twitch
    |           |
    |           |
```

```
CREATE TABLE SeriesInEvent (
      s_id INTEGER PRIMARY KEY,
      game_date DATE NOT NULL,
      winning_organization INTEGER,
      e_id INTEGER,
      FOREIGN KEY (e_id) REFERENCES Event (e_id) ON DELETE SET
NULL,
      FOREIGN KEY (winning_organization) REFERENCES OrganizationID
(o_id)
);
```

```
SELECT s_id, game_date, winning_organization, e_id FROM SeriesInEvent;

S_ID|GAME_DATE      |WINNING_ORGANIZATION|E_ID
----|--------------|--------------------|----
   1|19-FEB-23     |                   2|   1
   2|24-FEB-23     |                   3|   1
   3|30-MAY-21     |                   1|   2
   4|14-JUL-22     |                   5|   3
   5|17-JUL-22     |                   4|   3
```

```
CREATE TABLE MatchInSeries (
      m_id INTEGER PRIMARY KEY,
      num_rounds INTEGER,
      winning_organization INTEGER,
      scoreline CHAR(255),
      s_id INTEGER,
      FOREIGN KEY (s_id) REFERENCES SeriesInEvent (s_id) ON DELETE
SET NULL,
      FOREIGN KEY (winning_organization) REFERENCES OrganizationID
(o_id)
);
```

```
SELECT m_id, num_rounds, winning_organization, scoreline, s_id FROM MatchInSeries;


M_ID|NUM_ROUNDS|WINNING_ORGANIZATION|SCORELINE          |S_ID
----|----------|--------------------|--------------------|----
   1|        19|                   3|13:6                |   2
    |          |                    |                    |

   2|        20|                   3|13:7                |   2
    |          |                    |                    |

   3|        16|                   2|13:3                |   1
    |          |                    |                    |

   4|        28|                   5|15:13               |   1
    |          |                    |                    |

   5|        34|                   2|18:16               |   1
    |          |                    |                    |
```

```
CREATE TABLE Map (
      map_name CHAR(255),
      PRIMARY KEY (map_name)
);
```

```
SELECT map_name FROM Map;

MAP_NAME
--------------------
Ascent

Breeze

Fracture

Haven

Lotus

Pearl

Split
```

```
 CREATE TABLE MapPicks (
    map_name CHAR(255),
    o_id INTEGER,
    num_times_played INTEGER,
    PRIMARY KEY (map_name, o_id),
    FOREIGN KEY (map_name) REFERENCES Map ON DELETE SET NULL,
    FOREIGN KEY (o_id) REFERENCES OrganizationID ON DELETE CASCADE
);
```

```
SELECT map_name, o_id, num_times_played FROM MapPicks;

MAP_NAME           |O_ID|NUM_TIMES_PLAYED
-------------------|----|---------------
Ascent             |   1|             25
                   |    |

Haven              |   2|             11
                   |    |

Ascent             |   5|             18
                   |    |

Lotus              |   3|              2
                   |    |

Breeze             |   4|              0
                   |    |
```

```
CREATE TABLE MapPlayed (
    m_id INTEGER,
    map_name CHAR(255) NOT NULL,
    starting_side CHAR(255) NOT NULL,
    PRIMARY KEY (m_id, map_name),
    FOREIGN KEY (m_id) REFERENCES MatchInSeries ON DELETE CASCADE,
    FOREIGN KEY (map_name) REFERENCES Map ON DELETE SET NULL
);
```

```
SELECT m_id, map_name, starting_side FROM MapPlayed;

M_ID|MAP_NAME           |STARTING_S
----|-------------------|----------
   1|Haven               |Defend
    |                    |

   2|Split               |Defend
    |                    |

   3|Split               |Attack
    |                    |

   4|Pearl               |Defend
    |                    |

   5|Fracture            |Attack
    |                    |
```

```
CREATE TABLE Matchup (
    s_id INTEGER,
    o_id INTEGER,
    PRIMARY KEY (s_id, o_id),
    FOREIGN KEY (s_id) REFERENCES SeriesInEvent ON DELETE CASCADE,
    FOREIGN KEY (o_id) REFERENCES OrganizationID ON DELETE CASCADE
);
```

```
SELECT s_id, o_id FROM Matchup;

S_ID|O_ID
----|----
   1|   1
   1|   2
   2|   2
   2|   3
   2|   4
```

```
CREATE TABLE Agent (
    agent_number INTEGER,
    name CHAR(255) UNIQUE NOT NULL,
    pick_rate REAL NOT NULL,
    PRIMARY KEY (agent_number)
);
```

```
SELECT agent_number, name, pick_rate FROM Agent;


AGENT_NUMBER|NAME                |PICK_RATE
------------|--------------------|---------
           4|Killjoy             |      .66
            |                    |

          10|Jett                |      .55
            |                    |

           6|Sova                |      .47
            |                    |

          13|Breach              |      .41
            |                    |

           3|Omen                |      .41
            |                    |
```

```
CREATE TABLE AgentPlayed (
    tm_id INTEGER,
    m_id INTEGER,
    agent_number INTEGER,
    PRIMARY KEY (tm_id, m_id, agent_number)
);
```

```
SELECT tm_id, m_id, agent_number FROM AgentPlayed;

 TM_ID|M_ID|AGENT_NUMBER
------|----|------------
     1|   1|          10
     1|   3|           8
     2|   1|           6
     2|   3|           1
     2|   4|           3
```

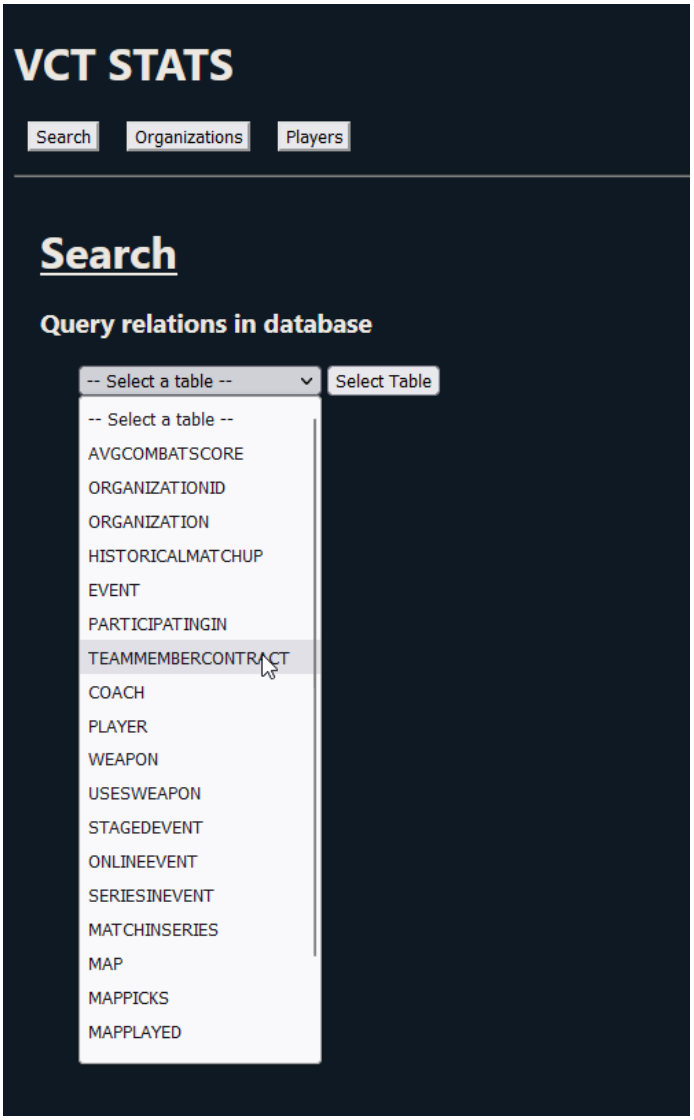
List of all SQL queries used

a) INSERT Operation
   i) Organizations.php, line 223
b) DELETE Operation
   i) Organizations.php, line 214

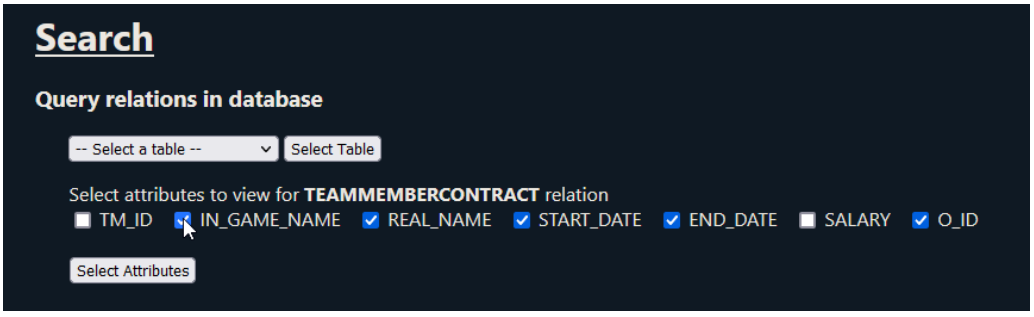
c) UPDATE Operation
    i) Organizations.php, line 205
d) SELECTION Operation
    i) Organizations.php, line 292
e) PROJECTION Operation
    i) Search.php, line 102
f) JOIN Operation
    i) Players.php, line 323
g) GROUP BY Operation
    i) Organizations.php, line 278
h) HAVING Operation
    i) Players.php, line 246
i) Nested GROUP BY Operation
    i) Organizations.php, line 278
j) DIVISION Operation
    i) Players.php, line 284

GUI Screenshots

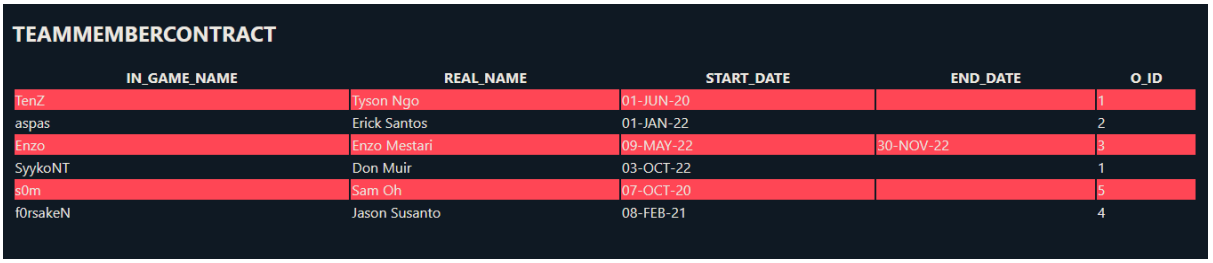
Search Tool (Projection)



User selects table they want to project on



User selects attributes to project for selected table

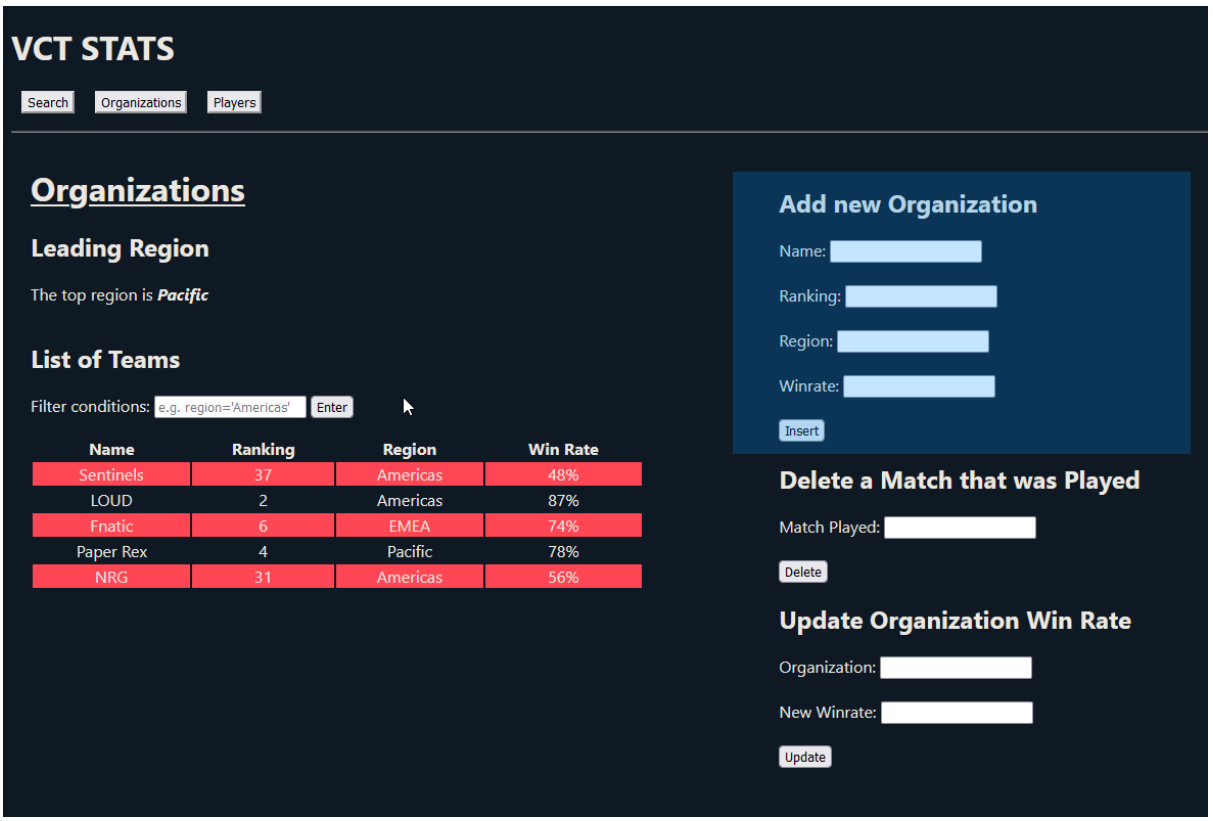


TEAMMEMBERCONTRACT

IN_GAME_NAME	REAL_NAME	START_DATE	END_DATE	O_ID
TenZ	Tyson Ngo	01-JUN-20		1
aspas	Erick Santos	01-JAN-22		2
Enzo	Enzo Mestari	09-MAY-22	30-NOV-22	3
SyykoNT	Don Muir	03-OCT-22		1
s0m	Sam Oh	07-OCT-20		5
f0rsakeN	Jason Susanto	08-FEB-21		4

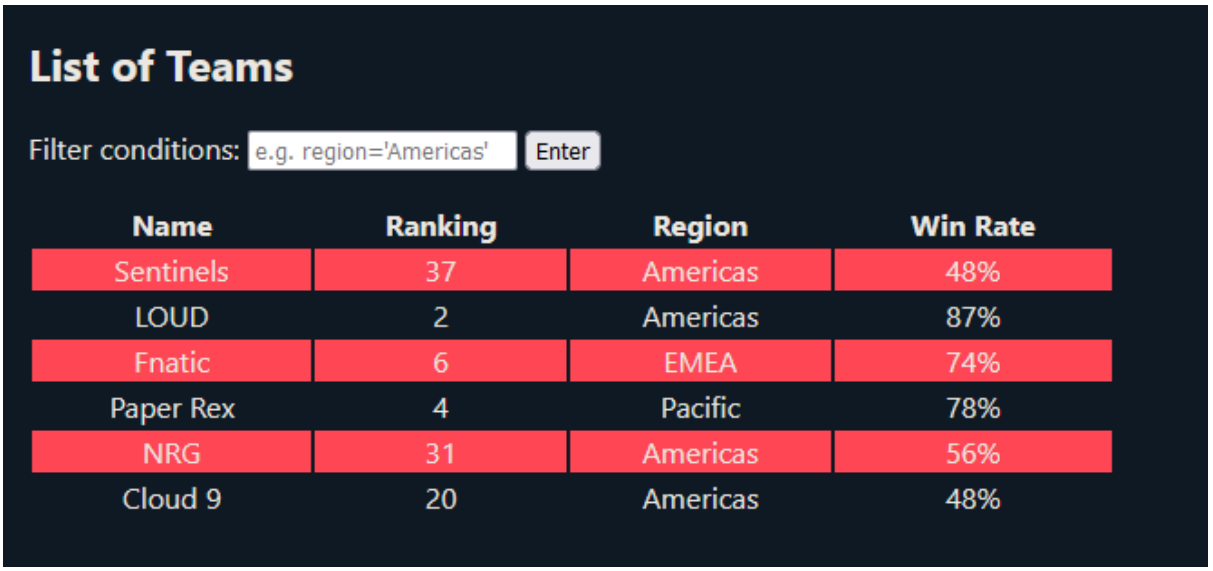
Resulting table with projected attributes

Add New Organization (Insert)



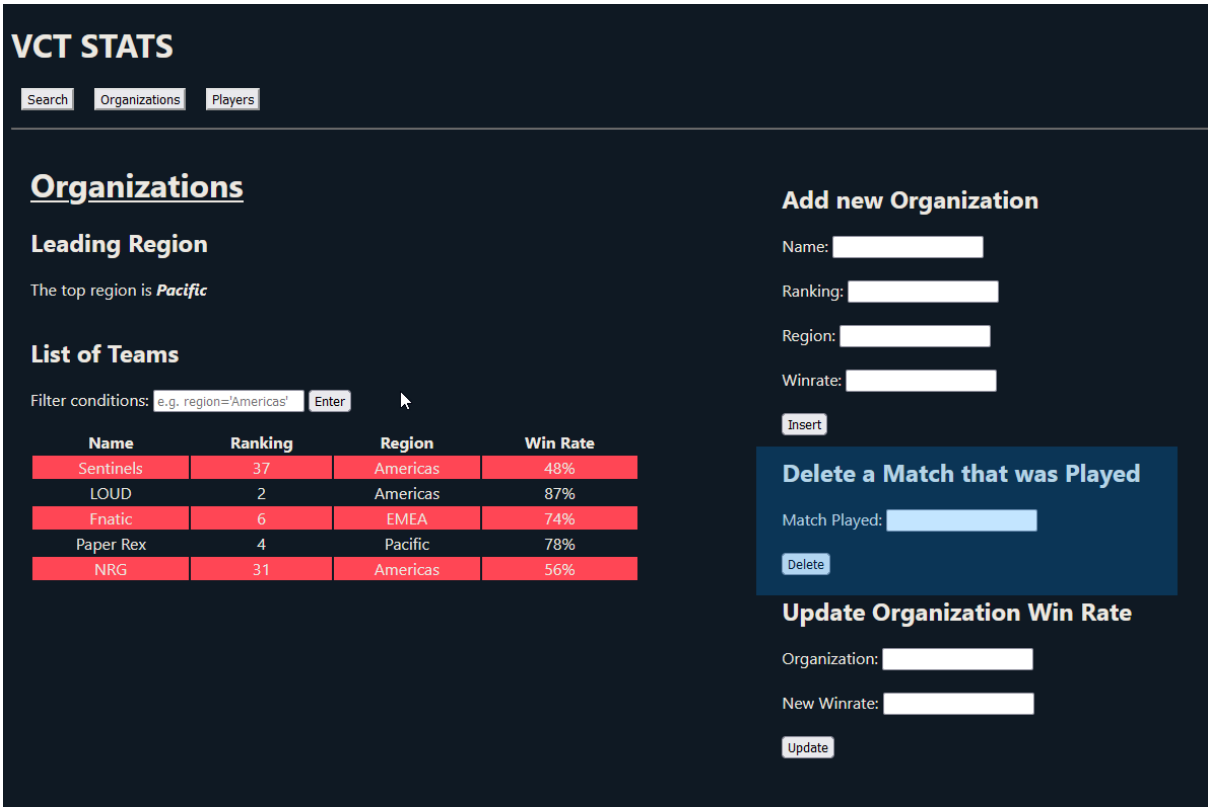
The highlighted area is for the user to add a new organization into the database with the organization name, ranking, region, and winrate. User can see the organizations already in the database by looking at the table under List of Teams in red and black on the middle left-hand side (5 teams)

User inputs organization name, ranking, region, and winrate

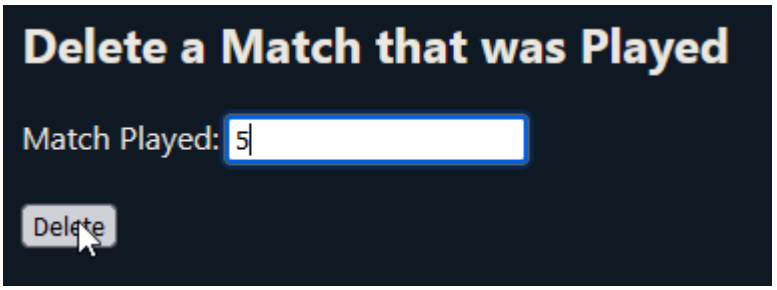


Resulting table shows 6 teams (+1 teams from the previous picture) with the added team being the inputs from the previous step

Delete an Existing Match (Delete)



The highlighted area is for the user to delete a match that has been played



User inputs match ID (m_id) which user wants to delete

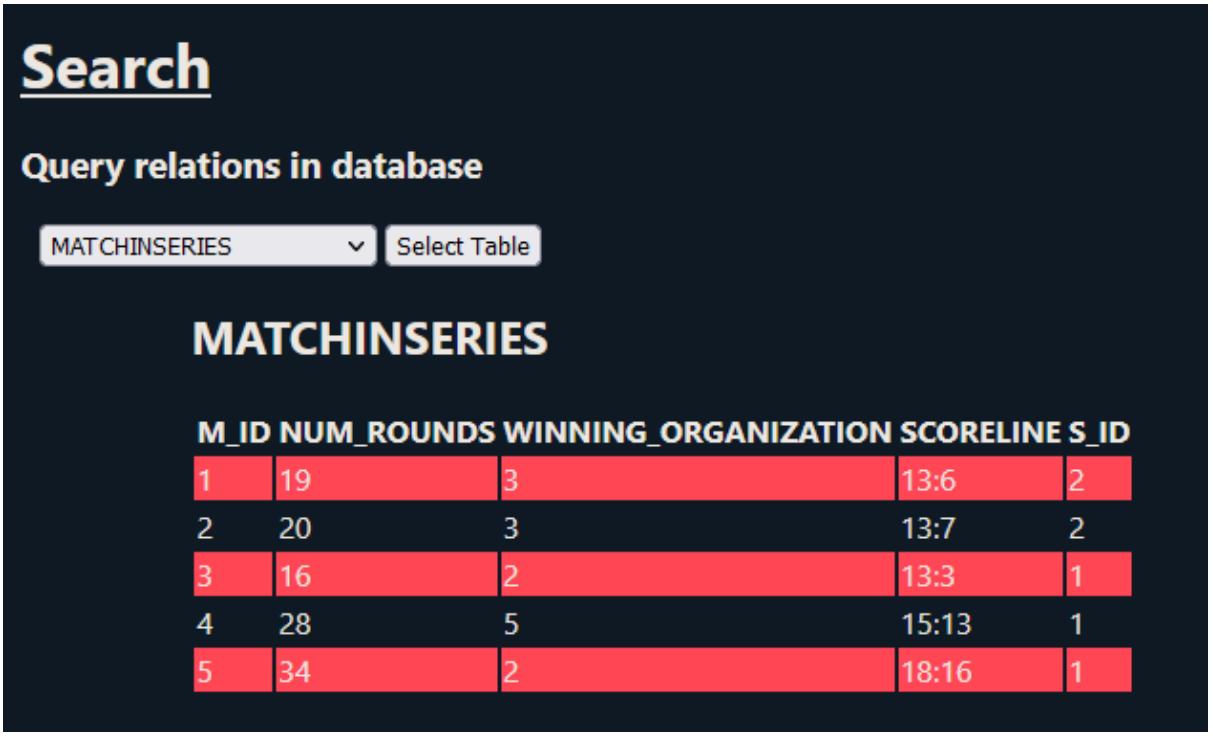


Table showing matches BEFORE the deletion

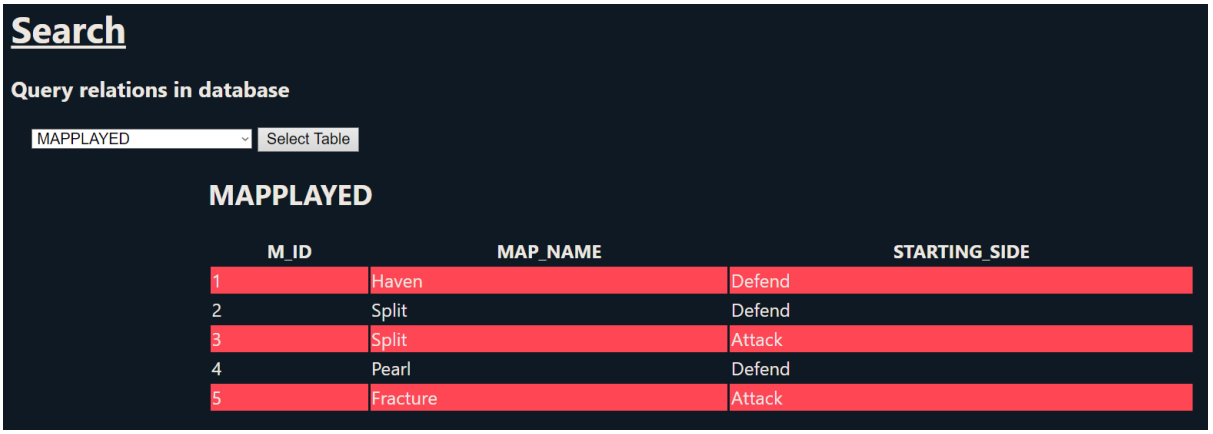


Table showing MapPlayed BEFORE the deletion. note that MapPlayed has a foreign key m_id to MatchInSeries. So when a row with an m_id is removed from MatchInSeries, the entry corresponding to that m_id should also be removed from MapPlayed.

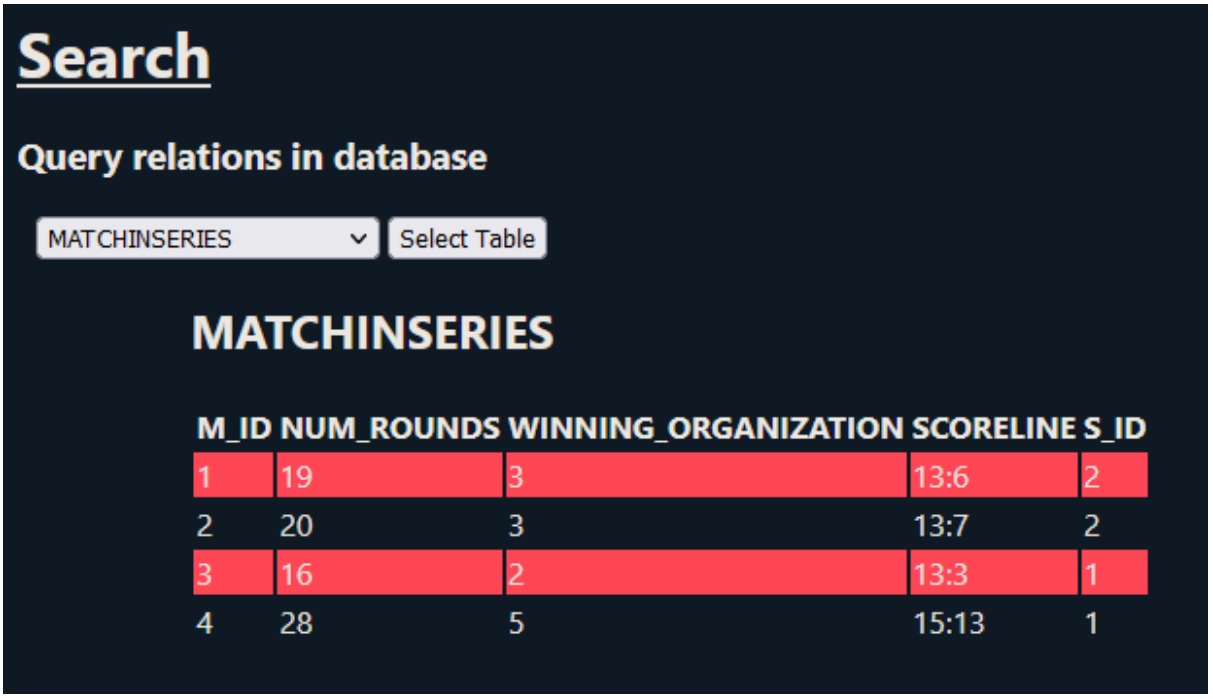
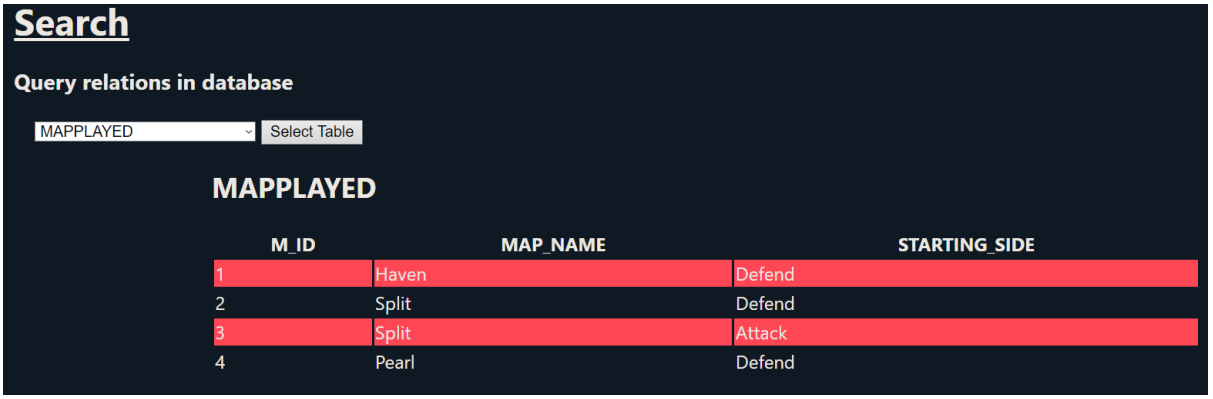
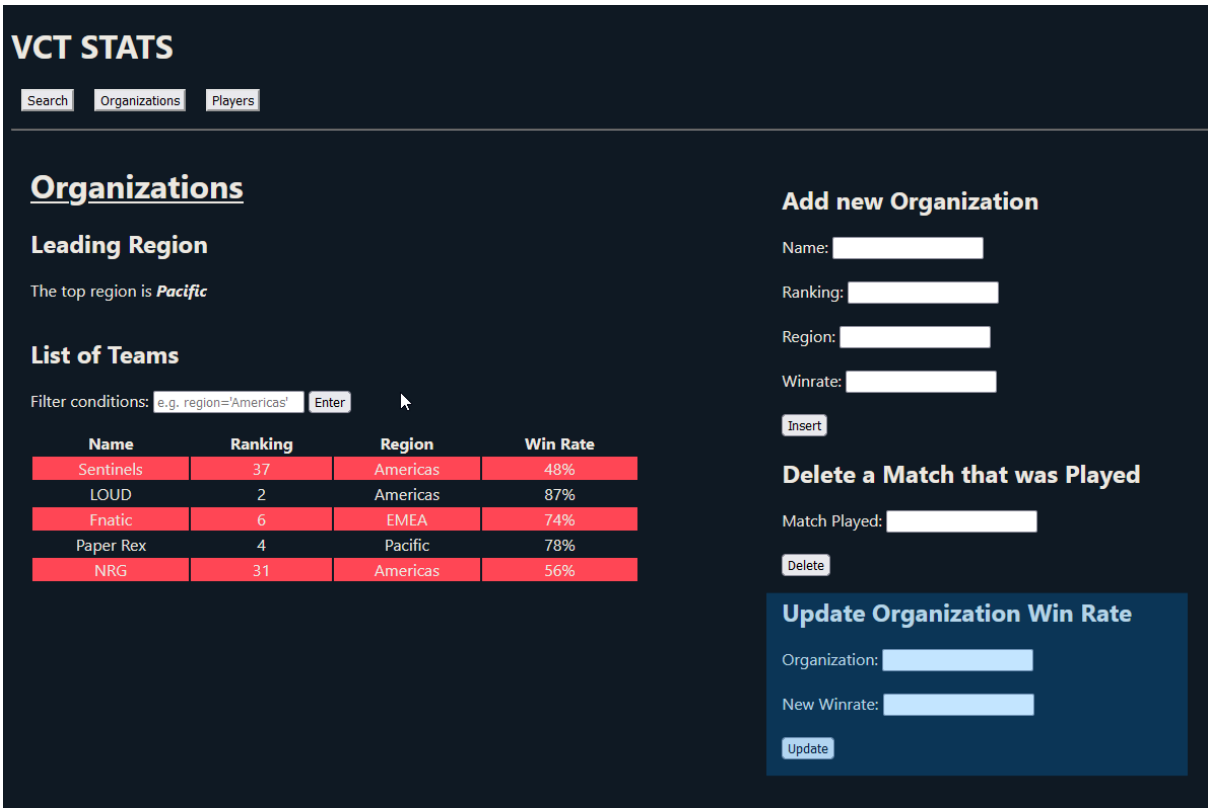


Table showing matches AFTER the deletion of m_id = 5

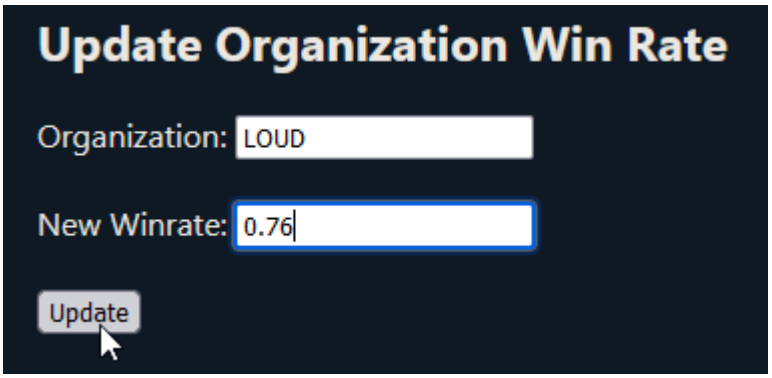


The entry corresponding to m_id = 5 is also removed from MapPlayed.

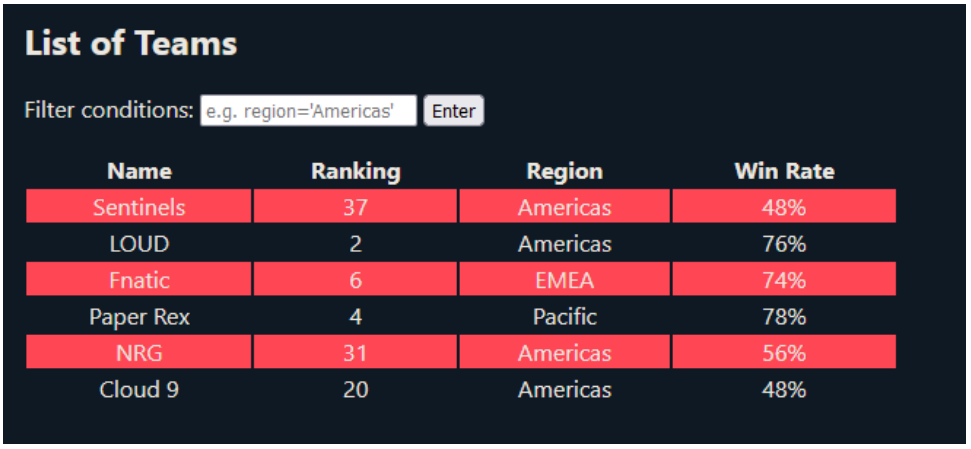
Update Organization Win Rate



The highlighted area is for the user to update the win rate of an organization. We can see the win rate for LOUD is currently 87%.

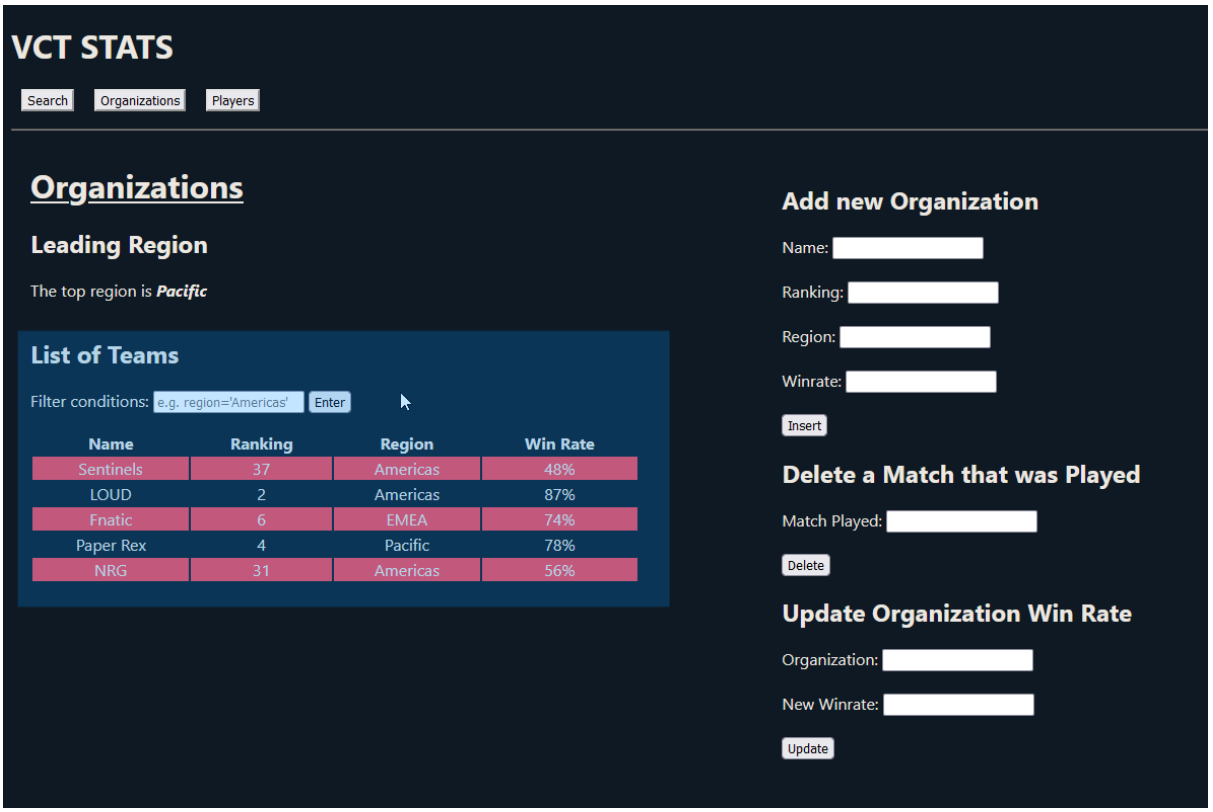


User inputs the organization of which its win rate is to be changed, and the new win rate for the organization

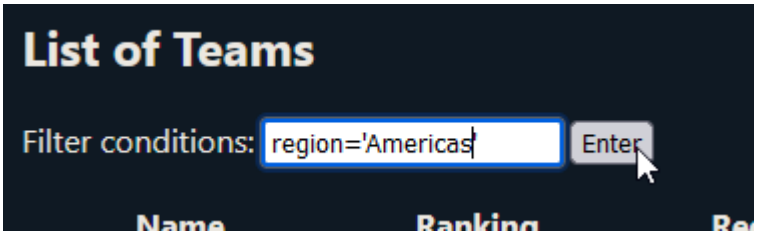


The resulting table shows LOUD with its new win rate at 76%.

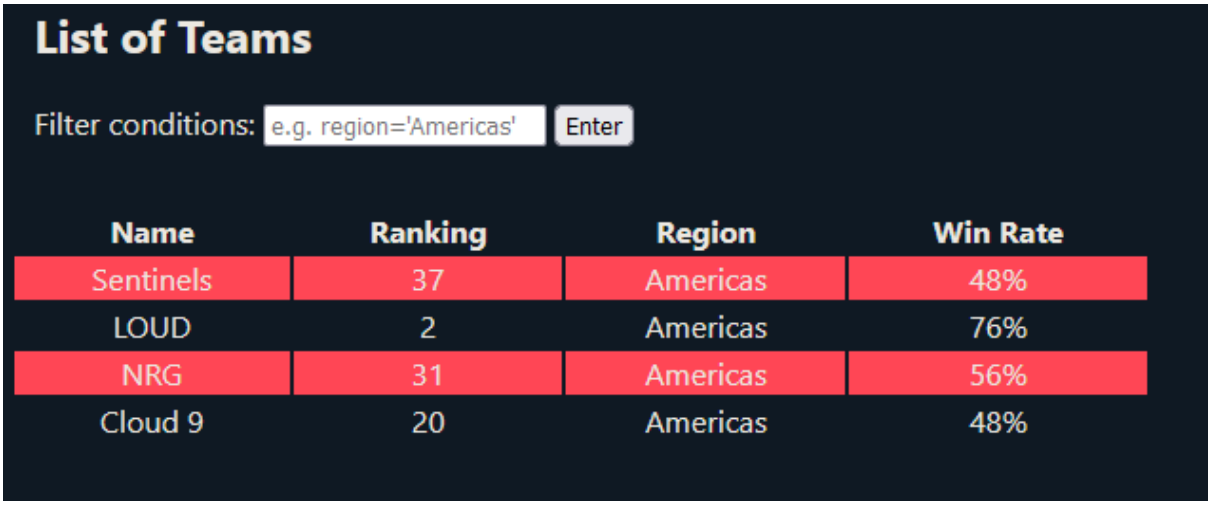
List Teams with Input Filter Conditions (Selection)



The highlighted region allows the user to filter the table based on a specified condition.

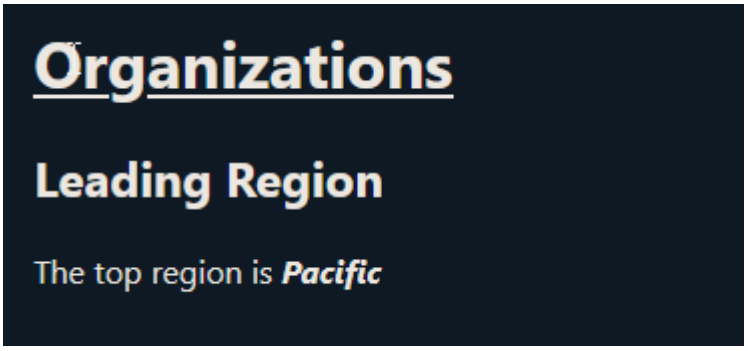


*User indicates **region='Americas'** to be the filter in the WHERE clause for the table shown*



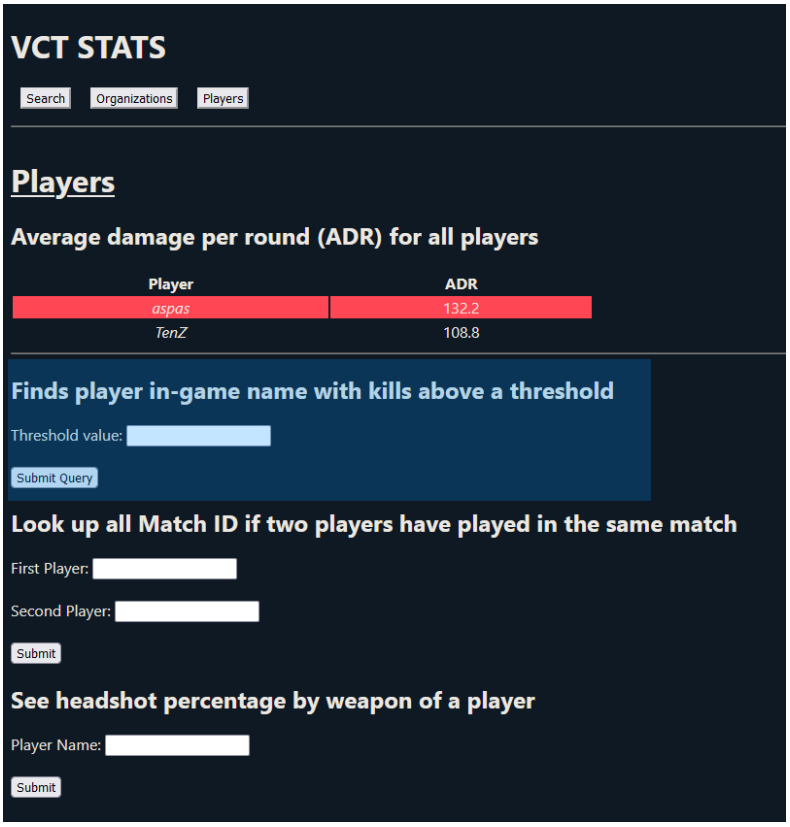
*The resulting table is filtered for as shown **regions='Americas'** under the **Region** column*

Find the Region with the Highest Average Win Rate (Group By, Nested Group By)

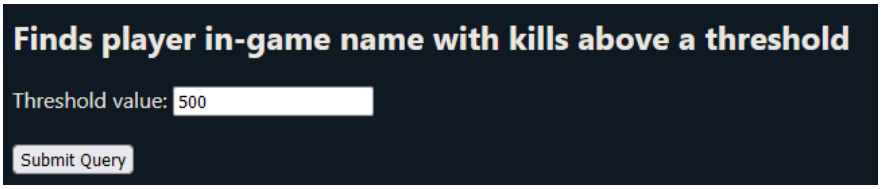


User can see the leading region based on the highest individual team win rate and its region

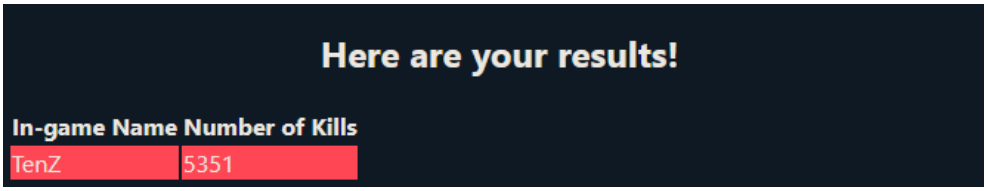
Find players with kills above a threshold (Having, Aggregation)



The highlighted area has threshold value represents the number of kills which filters for players that have kills above that value

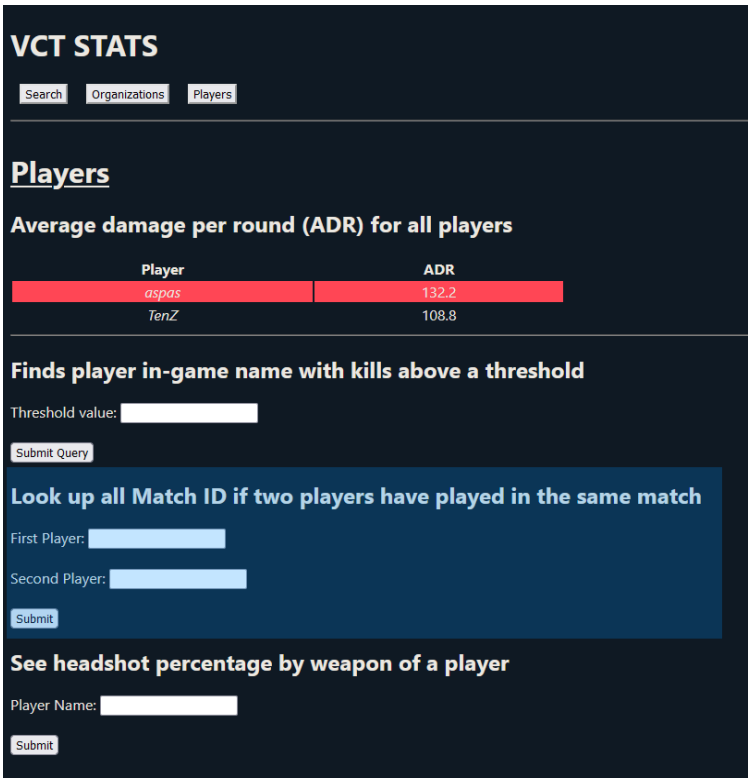


User inputs the value desired as a threshold

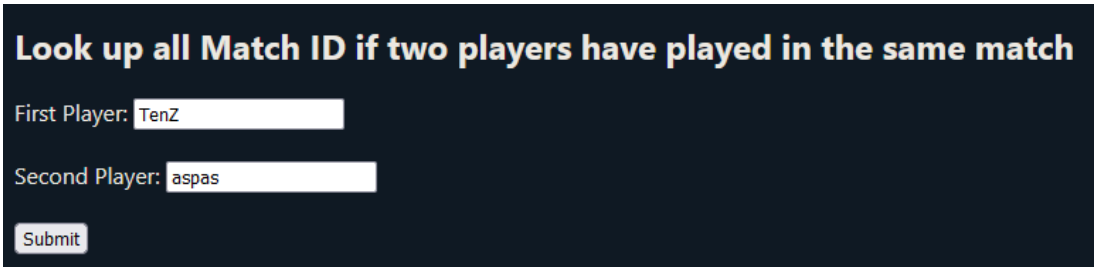


Resulting table shows player with total number of kills above the threshold value, and shows the player's number of kills

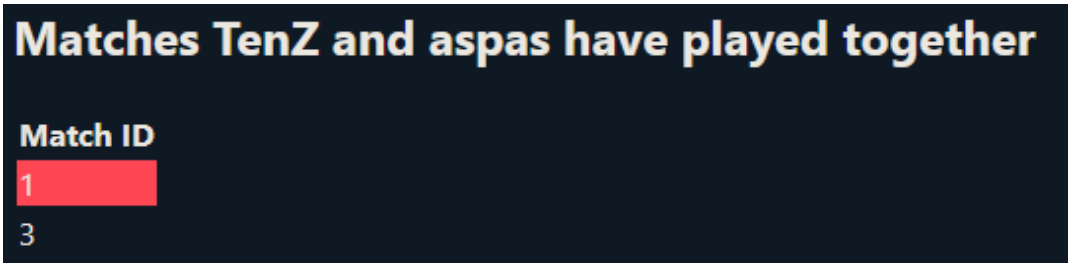
Find all matches where two players played in the same match (Division)



The highlighted area shows boxes where user can input two different player in-game-names already in the database to find matches which both players have played together in

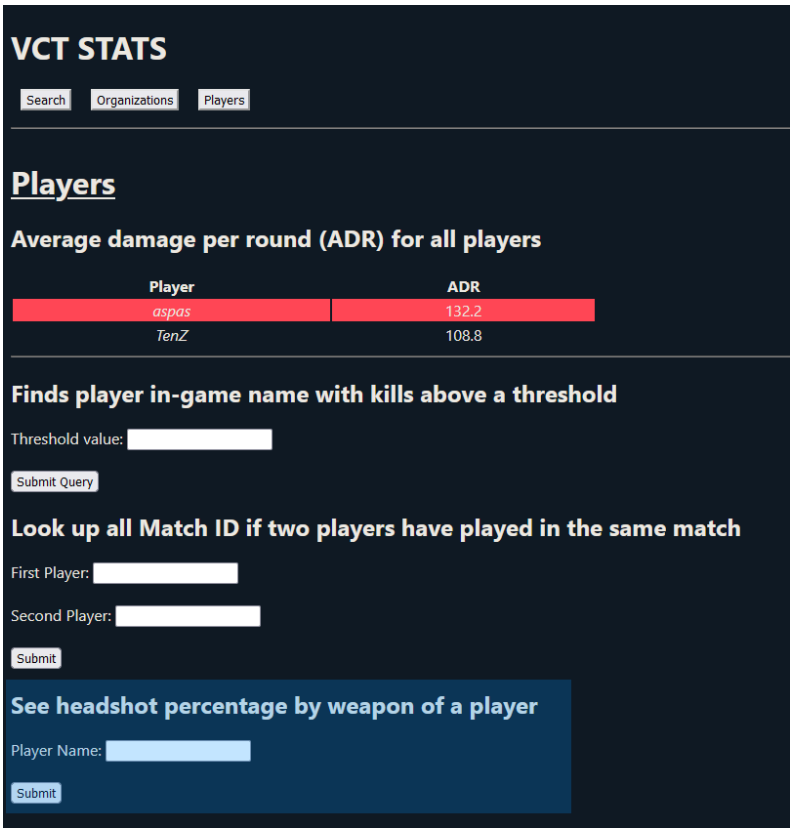


User inputs two player names into the boxes

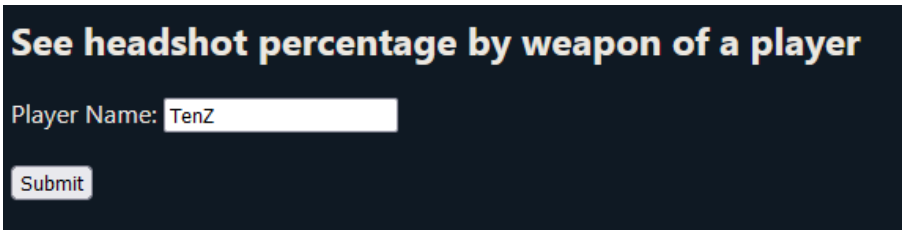


Resulting table is the matches based on Match ID (m_id) which both players have played together in

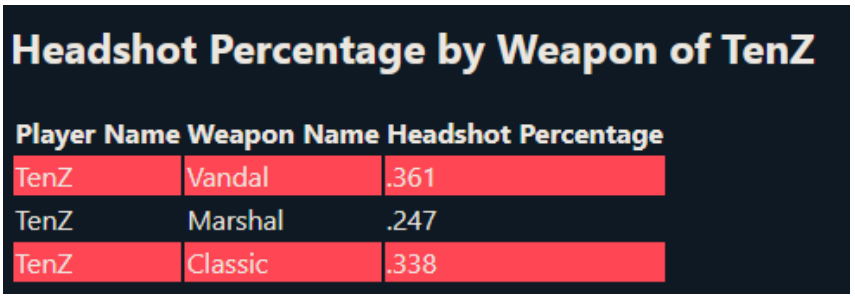
See headshot percentage by weapon of a player (Join)



The highlighted area asks for input for player in-game-name that is already in database to find their headshot percentage based on weapons used by the player



User inputs a player name



Resulting table shows the player name, the different weapons used, and the corresponding headshot percentage