University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone # : 4

Date : Apr 5, 2024

Group Number : 22

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Matthew Wu	36535664	d7a3b	matthew6086888@hotmail.com
Allya Wellyanto	47113238	k3q9c	allwelly@student.ubc.ca
Siddarth	68727171	s3t4n	siddarth.2400@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

2. SQL setup

Note that we are using MySQL for this project. We have created a SQL script that does the following:

- Drop all tables relevant to our application (eg. get rid of all data)
- Create all tables relevant to our application
- Insert sample data into the database

This file is located in `sql/init_db.sql`

University of British Columbia, Vancouver
Department of Computer Science

3.a: A short description of the final project, and what it accomplished.

We are building an application to record games of Texas Hold'em Poker (no limit). The database will model the game itself, and some information about the players, but not information about game strategy. Using this database, users will be able to record how previous games of poker played out, and it will have enough information to accurately recreate those games.

University of British Columbia, Vancouver
Department of Computer Science

3.b: How does your final schema differ from your original schema? Why did you make these changes?

We renamed the following tables, so their names would be more interpretable:

Before	After
Game_R2	Game_Season
Game_R3	Game
Game_R4	Game_Tournament
Player_R2	Player_region
Player_R3	Player
Player_R4	Player_Country
Round_R2	Round_Type
Round_R1	Round

We also added a “net balance” attribute to the players table, to keep track of each player’s net earnings/losses.

We had to rename the “Action” table to “Actions”, because “Action” was a reserved keyword in MYSQL

University of British Columbia, Vancouver
Department of Computer Science

3.c: A copy of the schema, and screenshots of each table after running the SQL setup script

Card(
id: INTEGER UNIQUE,
value: INTEGER NOT NULL,
suite: CHAR(1) NOT NULL)

Hole_Cards(
id: INTEGER UNIQUE,
match_id: INTEGER,
player_id: INTEGER,
card1: INTEGER NOT NULL,
card2: INTEGER NOT NULL)

Round(
id: INTEGER UNIQUE,
round_number: INTEGER NOT NULL,
match_id: INTEGER NOT NULL,
round_type: TEXT NOT NULL,
pot_size: FLOAT NOT NULL,
card1_id: INTEGER,
card2_id: INTEGER,
card3_id: INTEGER,)

Actions(
id: INTEGER UNIQUE,
action_number: INTEGER NOT NULL,
round_id: INTEGER NOT NULL,
action_type: TEXT NOT NULL,
amount: FLOAT,
ending_balance: FLOAT NOT NULL,
player_id: INTEGER NOT NULL)

Match(
id: INTEGER UNIQUE,
match_number: INTEGER NOT NULL,
game_id: INTEGER NOT NULL)

Game(
id: INTEGER,
date: DATE,

season_number: INTEGER,
in_tournament: BOOLEAN,
tournament_id: INTEGER)

Player(
id: INTEGER,
name: TEXT,
home_country: TEXT,
poker_region: TEXT
net_winnings: FLOAT)

Buy_in_cash_out(
id: INTEGER,
action_type: INTEGER NOT NULL,
amount: FLOAT NOT NULL,
match_id: INTEGER NOT NULL,
player_id: INTEGER NOT NULL)

Here is a screenshot of each relation after running our database setup script:

Action:

```
mysql> SELECT * from Action ORDER BY id ASC;
```

id	action_number	round_id	player_id	action_type	amount	ending_balance
1	1	1	1	Call	50	450
2	2	1	2	Raise	100	900
3	3	1	3	Fold	0	1500
4	4	1	4	Call	100	600
5	5	1	5	Call	100	400
6	1	2	1	Check	0	450
7	2	2	2	Bet	200	700
8	3	2	4	Fold	0	600
9	4	2	5	Call	200	200
10	1	3	1	Check	0	450
11	2	3	2	Bet	300	400
12	3	3	5	Fold	0	200
13	1	4	6	Raise	100	900
14	2	4	7	Call	100	900
15	3	4	8	Fold	0	1000
16	4	4	9	Call	100	900
17	5	4	10	Call	100	900
18	1	5	6	Check	0	900
19	2	5	7	Bet	200	700
20	3	5	9	Call	200	700
21	4	5	10	Fold	0	900
22	1	6	6	Check	0	900
23	2	6	7	Check	0	700
24	3	6	9	Check	0	700
25	1	7	6	Bet	300	600
26	2	7	7	Fold	0	700
27	3	7	9	Call	300	400
28	1	8	1	Call	50	450
29	2	8	2	Raise	150	850
30	3	8	3	Call	150	850
31	4	8	4	Fold	0	1000
32	5	8	5	Call	150	350
33	6	8	6	Fold	0	1000
34	1	9	1	Check	0	450
35	2	9	2	Bet	200	650
36	3	9	3	Fold	0	850
37	4	9	5	Call	200	150
38	1	10	1	Bet	150	300
39	2	10	2	Call	150	500
40	3	10	5	Fold	0	150
41	1	11	1	Check	0	300
42	2	11	2	Check	0	500
43	1	12	7	Raise	100	900
44	2	12	8	Call	100	900
45	3	12	9	Fold	0	1000
46	4	12	10	Call	100	900
47	5	12	1	Call	100	400
48	1	13	7	Check	0	900

Buy_in_cash_out:

```
mysql> SELECT * from Buy_in_cash_out ORDER BY id ASC;
+----+-------------+--------+----------+-----------+
| id | action_type | amount | match_id | player_id |
+----+-------------+--------+----------+-----------+
|  1 |           1 |   1000 |        1 |         1 |
|  2 |           1 |   1000 |        1 |         2 |
|  3 |           1 |   1000 |        1 |         3 |
|  4 |           1 |   1000 |        1 |         4 |
|  5 |           1 |   1000 |        1 |         5 |
|  6 |           1 |   1000 |        2 |         6 |
|  7 |           1 |   1000 |        2 |         7 |
|  8 |           1 |   1000 |        2 |         8 |
|  9 |           1 |   1000 |        2 |         9 |
| 10 |           1 |   1000 |        2 |        10 |
| 11 |           2 |   1600 |        3 |         1 |
| 12 |           2 |    700 |        3 |         2 |
| 13 |           2 |    850 |        3 |         3 |
| 14 |           2 |   1300 |        3 |         4 |
| 15 |           2 |    800 |        3 |         5 |
| 16 |           2 |    750 |        3 |         6 |
| 17 |           2 |    500 |        4 |         7 |
| 18 |           2 |   1500 |        4 |         8 |
| 19 |           2 |    950 |        4 |         9 |
| 20 |           2 |   1000 |        4 |        10 |
| 21 |           2 |     50 |        4 |         1 |
+----+-------------+--------+----------+-----------+
21 rows in set (0.00 sec)
```

Card:

```
mysql> SELECT * from Card ORDER BY id ASC;
+----+-------+-------+
| id | value | suite |
+----+-------+-------+
|  1 |     1 | D     |
|  2 |     1 | C     |
|  3 |     1 | H     |
|  4 |     1 | S     |
|  5 |     2 | D     |
|  6 |     2 | C     |
|  7 |     2 | H     |
|  8 |     2 | S     |
|  9 |     3 | D     |
| 10 |     3 | C     |
| 11 |     3 | H     |
| 12 |     3 | S     |
| 13 |     4 | D     |
| 14 |     4 | C     |
| 15 |     4 | H     |
| 16 |     4 | S     |
| 17 |     5 | D     |
| 18 |     5 | C     |
| 19 |     5 | H     |
| 20 |     5 | S     |
| 21 |     6 | D     |
| 22 |     6 | C     |
| 23 |     6 | H     |
| 24 |     6 | S     |
| 25 |     7 | D     |
| 26 |     7 | C     |
| 27 |     7 | H     |
| 28 |     7 | S     |
| 29 |     8 | D     |
```

Game:

```
mysql> SELECT * from Game ORDER BY id ASC;
+----+------------+
| id | date       |
+----+------------+
|  1 | 2023-01-01 |
|  2 | 2023-02-02 |
|  3 | 2023-03-03 |
|  4 | 2023-04-04 |
|  5 | 2023-05-05 |
|  6 | 2023-06-06 |
|  7 | 2023-07-07 |
|  8 | 2023-08-08 |
+----+------------+
8 rows in set (0.01 sec)
```

Game_Season:

```
mysql> SELECT * from Game_Season ORDER BY date ASC;
+------------+---------------+
| date       | season_number |
+------------+---------------+
| 2023-01-01 |            23 |
| 2023-02-02 |            23 |
| 2023-03-03 |            23 |
| 2023-04-04 |            23 |
| 2023-05-05 |            23 |
| 2023-06-06 |            23 |
| 2023-07-07 |            24 |
| 2023-08-08 |            24 |
+------------+---------------+
8 rows in set (0.00 sec)
```

Game_tournament:

```
mysql> SELECT * from Game_Tournament ORDER BY date ASC;
+------------+---------------+---------------+
| date       | in_tournament | tournament_id |
+------------+---------------+---------------+
| 2023-01-01 |             1 |           101 |
| 2023-02-02 |             0 |          NULL |
| 2023-03-03 |             1 |           103 |
| 2023-04-04 |             1 |           104 |
| 2023-05-05 |             0 |          NULL |
| 2023-06-06 |             1 |           105 |
| 2023-07-07 |             0 |          NULL |
| 2023-08-08 |             1 |           106 |
+------------+---------------+---------------+
8 rows in set (0.00 sec)
```

Hole_Cards:

```
mysql> SELECT * from Hole_Cards ORDER BY id ASC;
+----+----------+-----------+-------+-------+
| id | match_id | player_id | card1 | card2 |
+----+----------+-----------+-------+-------+
|  1 |        1 |         1 |     1 |    14 |
|  2 |        1 |         2 |    28 |    41 |
|  3 |        1 |         3 |     4 |    17 |
|  4 |        1 |         4 |    31 |    44 |
|  5 |        1 |         5 |     7 |    20 |
|  6 |        2 |         6 |     2 |    15 |
|  7 |        2 |         7 |    29 |    42 |
|  8 |        2 |         8 |     5 |    18 |
|  9 |        2 |         9 |    32 |    45 |
| 10 |        2 |        10 |     8 |    21 |
| 11 |        3 |         1 |     3 |    16 |
| 12 |        3 |         2 |    30 |    43 |
| 13 |        3 |         3 |     6 |    19 |
| 14 |        3 |         4 |    33 |    46 |
| 15 |        3 |         5 |     9 |    22 |
| 16 |        3 |         6 |    34 |    47 |
| 17 |        4 |         7 |    10 |    23 |
| 18 |        4 |         8 |    35 |    48 |
| 19 |        4 |         9 |    12 |    25 |
| 20 |        4 |        10 |    49 |     2 |
| 21 |        4 |         1 |    13 |    26 |
| 22 |        5 |         3 |     1 |    13 |
| 23 |        5 |         4 |     2 |    14 |
| 24 |        5 |         5 |     3 |    15 |
| 25 |        5 |         6 |     4 |    16 |
| 26 |        5 |         7 |     5 |    17 |
| 27 |        6 |         8 |     6 |    18 |
| 28 |        6 |         9 |     7 |    19 |
| 29 |        6 |        10 |     8 |    20 |
| 30 |        6 |         1 |     9 |    21 |
| 31 |        6 |         2 |    10 |    22 |
| 32 |        7 |         8 |    23 |    37 |
| 33 |        7 |         9 |    24 |    38 |
| 34 |        7 |        10 |    25 |    39 |
| 35 |        7 |         1 |    26 |    40 |
| 36 |        7 |         2 |    27 |    41 |
| 37 |        8 |         3 |    28 |    42 |
| 38 |        8 |         4 |    29 |    43 |
| 39 |        8 |         5 |    30 |    44 |
| 40 |        8 |         6 |    31 |    45 |
| 41 |        8 |         7 |    32 |    46 |
| 42 |        8 |         8 |    33 |    47 |
| 43 |        9 |         9 |    34 |    48 |
```

Matches:

```
mysql> SELECT * from Matches ORDER BY id ASC;
+----+--------------+---------+
| id | match_number | game_id |
+----+--------------+---------+
|  1 |            1 |       1 |
|  2 |            2 |       1 |
|  3 |            3 |       2 |
|  4 |            4 |       2 |
|  5 |            5 |       3 |
|  6 |            6 |       3 |
|  7 |            7 |       4 |
|  8 |            8 |       4 |
|  9 |            9 |       5 |
| 10 |           10 |       5 |
| 11 |           11 |       6 |
| 12 |           12 |       6 |
| 13 |           13 |       7 |
| 14 |           14 |       7 |
| 15 |           15 |       8 |
| 16 |           16 |       8 |
+----+--------------+---------+
16 rows in set (0.01 sec)
```

Player:

```
mysql> SELECT * from Player ORDER BY id ASC;
+----+--------------------+--------------+
| id | name               | net_winnings |
+----+--------------------+--------------+
|  1 | John Doe           |          650 |
|  2 | Jane Smith         |         -300 |
|  3 | Pierre Dupont      |         -150 |
|  4 | Max Mustermann     |          300 |
|  5 | Ana Souza          |         -200 |
|  6 | Giovanni Rossi     |         -250 |
|  7 | Carlos Hernandez   |         -500 |
|  8 | Takumi Tanaka      |          500 |
|  9 | Isabella Brown     |          -50 |
| 10 | Alejandro Martinez |            0 |
+----+--------------------+--------------+
10 rows in set (0.00 sec)
```

Player_Country:

```
mysql> SELECT * from Player_Country ORDER BY id ASC;
+----+--------------+
| id | home_country |
+----+--------------+
|  1 | USA          |
|  2 | Canada       |
|  3 | France       |
|  4 | Germany      |
|  5 | Brazil       |
|  6 | Italy        |
|  7 | Spain        |
|  8 | Mexico       |
|  9 | Japan        |
| 10 | Australia    |
+----+--------------+
10 rows in set (0.00 sec)
```

Player_Region:

```
mysql> SELECT * from Player_Region;
+--------------+---------------+
| home_country | poker_region  |
+--------------+---------------+
| Australia    | Oceania       |
| Brazil       | South America |
| Canada       | North America |
| France       | Europe        |
| Germany      | Europe        |
| Italy        | Europe        |
| Japan        | Asia          |
| Mexico       | North America |
| Spain        | Europe        |
| USA          | North America |
+--------------+---------------+
10 rows in set (0.00 sec)
```

Round:

```
mysql> SELECT * from Round ORDER BY id ASC;
```

id	round_number	match_id	pot_size	card1_id	card2_id	card3_id
1	1	1	300	12	7	33
2	2	1	600	21	1	52
3	3	1	900	45	11	23
4	1	2	200	3	28	17
5	2	2	400	19	20	6
6	3	2	600	9	15	31
7	4	2	800	42	37	5
8	1	3	500	8	46	29
9	2	3	900	10	48	2
10	3	3	1300	24	35	50
11	4	3	1700	14	22	41
12	1	4	350	18	4	34
13	2	4	650	26	38	13
14	3	4	950	47	30	44
15	1	5	250	17	34	39
16	2	5	450	28	14	35
17	3	5	650	49	47	45
18	4	5	850	13	46	20
19	1	6	550	26	43	42
20	2	6	850	24	29	34
21	3	6	1150	29	8	16
22	4	6	1450	15	5	22
23	1	7	300	2	38	36
24	2	7	600	15	38	1
25	3	7	900	5	46	41
26	4	7	1200	4	15	5
27	1	8	400	3	22	5
28	2	8	700	33	16	18
29	3	8	1000	43	32	14
30	4	8	1300	35	9	47
31	1	9	500	37	31	16
32	2	9	800	51	31	52
33	3	9	1100	27	13	7
34	4	9	1400	7	43	28
35	1	10	350	23	28	27
36	2	10	650	30	47	4
37	3	10	950	44	42	7
38	4	10	1250	4	26	47
39	1	11	450	22	52	7
40	2	11	750	16	13	35
41	3	11	1050	29	9	28
42	4	11	1350	12	18	30
43	1	12	300	16	5	29
44	2	12	600	52	36	7
45	3	12	900	4	42	35
46	4	12	1200	1	6	49
47	1	13	500	16	11	27
48	2	13	850	32	31	14

Round_Type:

```
mysql> SELECT * from Round_Type;
+--------------+-------------+
| round_number | round_type  |
+--------------+-------------+
|            1 | Pre-flop    |
|            2 | Flop        |
|            3 | Turn        |
|            4 | River       |
+--------------+-------------+
4 rows in set (0.00 sec)
```

University of British Columbia, Vancouver
Department of Computer Science

3.d

List of all SQL queries, and where it can be found in the code
All SQL queries are executed on our API server (implemented with PyFlask). The server code is stored at “pyflask/__init__.py”. Here

Line #	Query
New Section	Games API
72	`INSERT INTO Game_Season (date, season_number) VALUES (%s, %s) ON DUPLICATE KEY UPDATE season_number = VALUES(season_number)`
84	`INSERT INTO Game (date) VALUES (%s)`
97	`INSERT INTO Game_Tournament (date, in_tournament, tournament_id) VALUES (%s, %s, %s) ON DUPLICATE KEY UPDATE in_tournament = VALUES(in_tournament), tournament_id = VALUES(tournament_id)`
122	`DELETE FROM Game WHERE id = %s`
150	`INSERT INTO Buy_in_cash_out (action_type, amount, match_id, player_id) VALUES (%s, %s, (SELECT id FROM Matches WHERE game_id = %s ORDER BY match_number DESC LIMIT 1), %s)`
160	`SELECT net_winnings FROM Player WHERE id = %s`
195	`INSERT INTO Buy_in_cash_out (action_type, amount, match_id, player_id) VALUES (%s, %s, (SELECT id FROM Matches WHERE game_id = %s ORDER BY match_number DESC LIMIT 1), %s)`
205	`SELECT net_winnings FROM Player WHERE id = %s`
245	`SELECT MAX(match_number) FROM Matches WHERE game_id = %s`

253	`INSERT INTO Matches (match_number, game_id) VALUES (%s, %s)`
292	`DELETE FROM Matches WHERE game_id = %s ORDER BY match_number DESC LIMIT 1`

314	`SELECT m.match_number, m.id`
	`FROM matches m`
	`WHERE m.game_id = %s`
	`ORDER BY m.match_number DESC`
	`LIMIT 1`

325	`SELECT r.round_number`
	`FROM round r`
	`INNER JOIN matches m ON r.match_id = m.id`
	`WHERE m.id = %s`
	`ORDER BY r.round_number DESC`
	`LIMIT 1`

345	`INSERT INTO round (round_number, match_id, pot_size, card1_id, card2_id, card3_id) VALUES (%s, %s, %s, %s, %s, %s)`

353	`INSERT INTO action`
	`(action_number, round_id, player_id, action_type, amount, ending_balance)`
	`VALUES {", ".join(["(%s, %s, %s, %s, %s, %s)"] * len(actions))}`

377	`SELECT r.id, r.round_number, m.id, m.match_number`
	`FROM round r`
	`INNER JOIN matches m ON r.match_id = m.id`
	`WHERE m.game_id = %s`
	`ORDER BY m.match_number DESC, r.round_number DESC`
	`LIMIT 1`

394	`DELETE FROM round WHERE round.id = %s`
New Section	Replay API
424	SELECT g.id AS game_id, g.date, gs.season_number, gt.in_tournament, gt.tournament_id FROM Game g INNER JOIN Game_Season gs ON g.date = gs.date LEFT JOIN Game_Tournament gt ON g.date = gt.date LIMIT %s OFFSET %s
464	SELECT m.id, m.match_number, m.game_id FROM Matches m WHERE m.game_id = %s ORDER BY m.match_number ASC LIMIT %s OFFSET %s
500	SELECT c.id, c.value, c.suite FROM card c
507	SELECT r.round_number, r.card1_id, r.card2_id, r.card3_id, r.pot_size FROM round r WHERE r.match_id = %s ORDER BY r.round_number ASC LIMIT %s OFFSET %s
517	SELECT r.round_number, r.card1_id, r.card2_id, r.card3_id, r.pot_size FROM round r INNER JOIN matches m ON m.id = r.match_id WHERE m.game_id = %s AND m.match_number = %s ORDER BY r.round_number ASC LIMIT %s OFFSET %s
568	SELECT DISTINCT g.id AS game_id, g.date, gs.season_number, gt.in_tournament, gt.tournament_id FROM Game g JOIN Matches m ON g.id = m.game_id LEFT JOIN Hole_Cards hc ON m.id = hc.match_id JOIN Game_Season gs ON g.date = gs.date LEFT JOIN Game_Tournament gt ON g.date = gt.date WHERE hc.player_id = %s ORDER BY g.date ASC LIMIT %s OFFSET %s

New Section	Admin API
607	`INSERT INTO player (name, net_winnings) VALUES (%s, %s)`
612	`INSERT INTO player_country (id, home_country) VALUES (%s, %s)`
616	`INSERT INTO player_region (id, region) VALUES (%s, %s)`
631	`DELETE FROM player WHERE id=%s`
651	`UPDATE player SET name=%s, net_winnings=%s WHERE id=%s`
655	`UPDATE player SET name=%s WHERE id=%s`
659	`UPDATE player SET net_winnings=%s WHERE id=%s`
664	`UPDATE player_country SET country=%s WHERE id=%s`
669	`UPDATE player_region SET region=%s WHERE id=%s`

687	`SELECT`
	`game.id AS game_id,`
	`COUNT(action.id) AS 'n_of_actions',`
	`SUM(action.action_type = 'fold') AS 'n_of_folds',`
	`SUM(action.action_type = 'check') AS 'n_of_checks',`
	`SUM(action.action_type = 'call') AS 'n_of_call',`
	`SUM(action.action_type = 'raise') AS 'n_of_raises',`
	`AVG(CASE WHEN action.action_type = 'raise' THEN action.amount END) AS 'avg_raise_size'`
	`FROM game`
	`INNER JOIN matches ON game.id = matches.game_id`
	`INNER JOIN round ON round.match_id = matches.id`
	`INNER JOIN action ON action.round_id = round.id`
	`INNER JOIN buy_in_cash_out ON`

	buy_in_cash_out.match_id = matches.id WHERE buy_in_cash_out.player_id = %s AND game.date >= %s AND game.date <= %s GROUP BY game.id;
728	SELECT DISTINCT p.id, p.name, pc.home_country, pr.poker_region, p.net_winnings FROM player p INNER JOIN player_country pc ON pc.id = p.id INNER JOIN player_region pr ON pc.home_country = pr.home_country WHERE NOT EXISTS ((SELECT game.id FROM game WHERE game.id IN ({', '.join(['%s'] * len(game_ids))})) EXCEPT (SELECT game.id FROM buy_in_cash_out bico INNER JOIN matches ON matches.id = bico.match_id INNER JOIN game ON matches.game_id = game.id WHERE bico.player_id = p.id AND game.id IN ({', '.join(['%s'] * len(game_ids))})));
774	SELECT

```
            gs.season_number AS season_number,
            COUNT(g.id) AS n_games,
            AVG(r.pot_size) AS avg_pot_size,
            MAX(r.pot_size) AS max_pot_size
        FROM Game g
        INNER JOIN game_season gs ON g.date = gs.date
        INNER JOIN matches m ON m.game_id = g.id
        INNER JOIN round r ON r.match_id = m.id
        INNER JOIN buy_in_cash_out bico ON bico.match_id
= m.id
        {"WHERE gs.season_number IN (%s)" if seasons
else "WHERE gs.season_number NOT IN (%s)"}
        GROUP BY gs.season_number
        HAVING COUNT(bico.player_id) > %s AND
            COUNT(g.id) > %s AND
            AVG(r.pot_size) > %s AND
            MAX(r.pot_size) > %s
```

University of British Columbia, Vancouver
Department of Computer Science

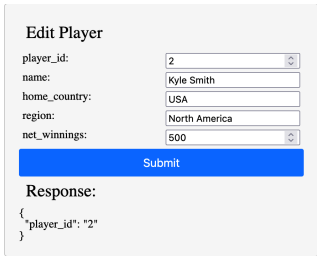
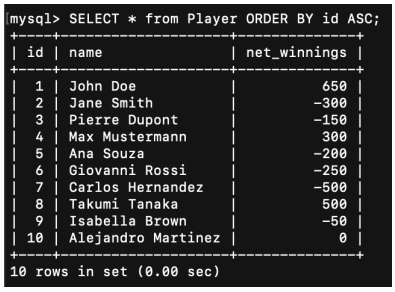
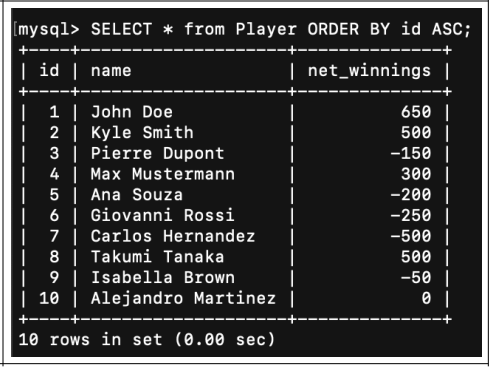
3.e

Screenshots of all SQL queries:

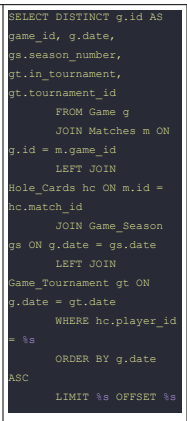
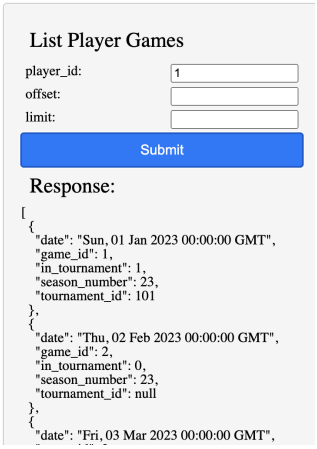
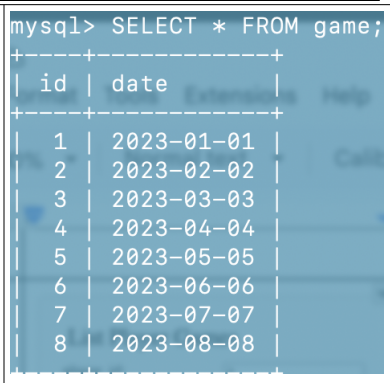
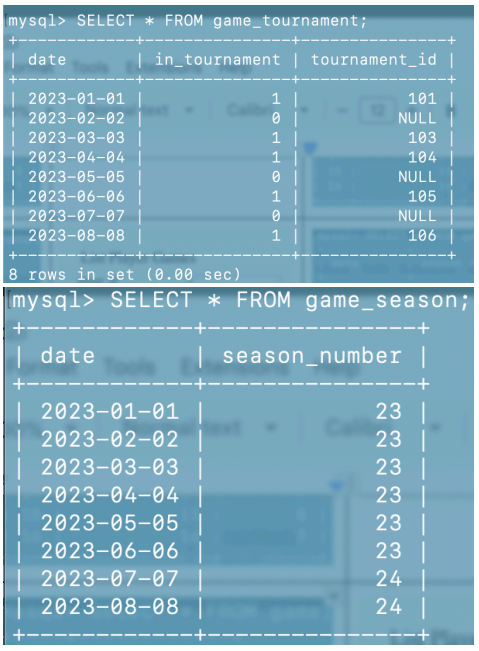
- Before: Screenshot of table before
- During: Screenshot of the GUI that triggers the query
- After: Screenshot of table after the query

We only need to do this once per query type (eg. only need to show INSERT once). All the sql code is in `pyflask/__init__.py`

Query Type	Line #	Query	GUI screenshot	Before Screenshot	After screenshot

UPDATE	651, 655, 659, 664, 669	`UPDATE player SET name=%s, net_winnings=%s WHERE id=%s`
		`UPDATE player SET name=%s WHERE id=%s`
		`UPDATE player SET net_winnings=%s WHERE id=%s`
		`UPDATE player_country SET country=%s WHERE id=%s`
		`UPDATE player_region SET region=%s WHERE id=%s`
					

selection (WHERE)	460	`SELECT m.id, m.match_number, m.game_id FROM Matches m WHERE m.game_id = %s ORDER BY m.match_number ASC LIMIT %s OFFSET %s`	List Matches game_id: 1 offset: limit: Submit Response: [{ "game_id": 1, "match_id": 1, "match_number": 1 }, { "game_id": 1, "match_id": 2, "match_number": 2 }]	`mysql> SELECT * FROM matches;` id | match_number | game_id 1 | 1 | 1 2 | 2 | 1 3 | 3 | 2 4 | 4 | 2 5 | 5 | 3 6 | 6 | 3 7 | 7 | 4 8 | 8 | 4 9 | 9 | 5 10 | 10 | 5 11 | 11 | 6 12 | 12 | 6 13 | 13 | 7 14 | 14 | 7 15 | 15 | 8 16 | 16 | 8	`mysql> SELECT * FROM matches;` id | match_number | game_id 1 | 1 | 1 2 | 2 | 1 3 | 3 | 2 4 | 4 | 2 5 | 5 | 3 6 | 6 | 3 7 | 7 | 4 8 | 8 | 4 9 | 9 | 5 10 | 10 | 5 11 | 11 | 6 12 | 12 | 6 13 | 13 | 7 14 | 14 | 7 15 | 15 | 8 16 | 16 | 8

projection (SELECT)	564				

```
SELECT DISTINCT g.id AS game_id, g.date,
gs.season_number,
gt.in_tournament,
gt.tournament_id
        FROM Game g
        JOIN Matches m ON g.id = m.game_id
        LEFT JOIN Hole_Cards hc ON m.id = hc.match_id
        JOIN Game_Seasons gs ON g.date = gs.date
        LEFT JOIN Game_Tournament gt ON g.date = gt.date
        WHERE hc.player_id = %s
        ORDER BY g.date ASC
        LIMIT %s OFFSET %s
```

List Player Games

player_id: 1

offset:

limit:

Submit

Response:


```
[
  {
    "date": "Sun, 01 Jan 2023 00:00:00 GMT",
    "game_id": 1,
    "in_tournament": 1,
    "season_number": 23,
    "tournament_id": 101
  },
  {
    "date": "Thu, 02 Feb 2023 00:00:00 GMT",
    "game_id": 2,
    "in_tournament": 0,
    "season_number": 23,
    "tournament_id": null
  },
  {
    "date": "Fri, 03 Mar 2023 00:00:00 GMT",
```


```
mysql> SELECT * FROM game;
+----+------------+
| id | date       |
+----+------------+
|  1 | 2023-01-01 |
|  2 | 2023-02-02 |
|  3 | 2023-03-03 |
|  4 | 2023-04-04 |
|  5 | 2023-05-05 |
|  6 | 2023-06-06 |
|  7 | 2023-07-07 |
|  8 | 2023-08-08 |
+----+------------+
```

```
mysql> SELECT * FROM game_tournament;
+------------+---------------+---------------+
| date       | in_tournament | tournament_id |
+------------+---------------+---------------+
| 2023-01-01 |             1 |           101 |
| 2023-02-02 |             0 |          NULL |
| 2023-03-03 |             1 |           103 |
| 2023-04-04 |             1 |           104 |
| 2023-05-05 |             0 |          NULL |
| 2023-06-06 |             1 |           105 |
| 2023-07-07 |             0 |          NULL |
| 2023-08-08 |             1 |           106 |
+------------+---------------+---------------+
8 rows in set (0.00 sec)
```

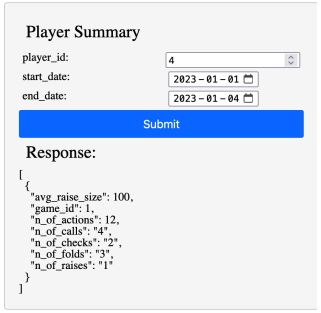
```
mysql> SELECT * FROM game_season;
+------------+---------------+
| date       | season_number |
+------------+---------------+
| 2023-01-01 |            23 |
| 2023-02-02 |            23 |
| 2023-03-03 |            23 |
| 2023-04-04 |            23 |
| 2023-05-05 |            23 |
| 2023-06-06 |            23 |
| 2023-07-07 |            24 |
| 2023-08-08 |            24 |
+------------+---------------+
```

JOIN	564	`SELECT DISTINCT g.id AS game_id, g.date, gs.season_number, gt.in_tournament, gt.tournament_id FROM Game g JOIN Matches m ON g.id = m.game_id LEFT JOIN Hole_Cards hc ON m.id = hc.match_id JOIN Game_Seasons gs ON g.date = gs.date LEFT JOIN Game_Tournament gt ON g.date = gt.date WHERE hc.player_id = %s ORDER BY g.date ASC LIMIT %s OFFSET %s`	List Player Games
			player_id: 1
			offset:
			limit:
			Submit
			Response:
			[{ "date": "Sun, 01 Jan 2023 00:00:00 GMT", "game_id": 1, "in_tournament": 1, "season_number": 23, "tournament_id": 101 }, { "date": "Thu, 02 Feb 2023 00:00:00 GMT", "game_id": 2, "in_tournament": 0, "season_number": 23, "tournament_id": null }, { "date": "Fri, 03 Mar 2023 00:00:00 GMT",
				`mysql> SELECT * FROM game;`
				`+----+------------+`
				`| id | date |`
				`+----+------------+`
				`| 1 | 2023-01-01 |`
				`| 2 | 2023-02-02 |`
				`| 3 | 2023-03-03 |`
				`| 4 | 2023-04-04 |`
				`| 5 | 2023-05-05 |`
				`| 6 | 2023-06-06 |`
				`| 7 | 2023-07-07 |`
				`| 8 | 2023-08-08 |`
				`+----+------------+`
					`mysql> SELECT * FROM game_tournament;`
					`+------------+---------------+---------------+`
					`| date | in_tournament | tournament_id |`
					`+------------+---------------+---------------+`
					`| 2023-01-01 | 1 | 101 |`
					`| 2023-02-02 | 0 | NULL |`
					`| 2023-03-03 | 1 | 103 |`
					`| 2023-04-04 | 1 | 104 |`
					`| 2023-05-05 | 0 | NULL |`
					`| 2023-06-06 | 1 | 105 |`
					`| 2023-07-07 | 0 | NULL |`
					`| 2023-08-08 | 1 | 106 |`
					`+------------+---------------+---------------+`
					`8 rows in set (0.00 sec)`
					
					`mysql> SELECT * FROM game_season;`
					`+------------+---------------+`
					`| date | season_number |`
					`+------------+---------------+`
					`| 2023-01-01 | 23 |`
					`| 2023-02-02 | 23 |`
					`| 2023-03-03 | 23 |`
					`| 2023-04-04 | 23 |`
					`| 2023-05-05 | 23 |`
					`| 2023-06-06 | 23 |`
					`| 2023-07-07 | 24 |`
					`| 2023-08-08 | 24 |`
					`+------------+---------------+`

GROUP BY | 687

```
SELECT
        game.id AS game_id,
        COUNT(action.id) AS 'n_of_actions',
        SUM(action.action_type = 'fold') AS 'n_of_folds',
        SUM(action.action_type = 'check') AS 'n_of_checks',
        SUM(action.action_type = 'call') AS 'n_of_call',
        SUM(action.action_type = 'raise') AS 'n_of_raises',
        AVG(CASE WHEN action.action_type = 'raise' THEN action.amount END) AS 'avg_raise_size'
        FROM game
        INNER JOIN matches ON game.id = matches.game_id
        INNER JOIN round ON round.match_id = matches.id
        INNER JOIN action ON action.round_id = round.id
        INNER JOIN buy_in_cash_out ON buy_in_cash_out.match_id = matches.id
        WHERE

buy_in_cash_out.player_id = %s AND
        game.date >= %s AND
        game.date <= %s
        GROUP BY game.id;
```



id	match_number	game_id
1	1	1
2	2	1
3	3	2
4	4	2
5	5	3
6	6	3
7	7	4

id	round_number	match_id	pot_size	card1_id	card2_id	card3_id
1	1	1	300	12	7	33
2	2	1	600	21	1	52
3	3	1	900	45	11	23
4	1	2	200	3	28	17
5	2	2	400	19	20	6
6	3	2	600	9	15	31
7	4	2	800	42	37	5
8	1	3	500	8	46	29
9	2	3	900	10	48	2
10	3	3	1300	24	35	50
11	4	3	1700	14	22	41
12	1	4	350	18	4	34

id	match_number	game_id
1	1	1
2	2	1
3	3	2
4	4	2
5	5	3
6	6	3
7	7	4

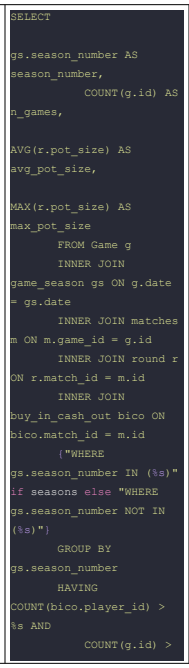
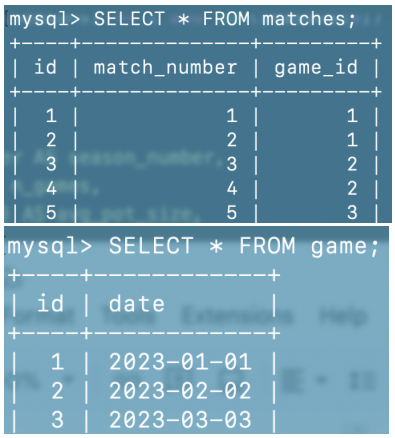
id	round_number	match_id	pot_size	card1_id	card2_id	card3_id
1	1	1	300	12	7	33
2	2	1	600	21	1	52
3	3	1	900	45	11	23
4	1	2	200	3	28	17
5	2	2	400	19	20	6
6	3	2	600	9	15	31
7	4	2	800	42	37	5
8	1	3	500	8	46	29
9	2	3	900	10	48	2
10	3	3	1300	24	35	50
11	4	3	1700	14	22	41
12	1	4	350	18	4	34

id	action_number	round_id	player_id	action_type	amount	ending_balance
1	1	1	1	Call	50	450
2	2	1	2	Raise	100	900
3	3	1	3	Fold	0	1500
4	4	1	4	Call	100	400
5	5	1	5	Call	100	400
6	1	2	1	Check	0	450
7	2	2	2	Bet	200	700
8	3	2	4	Fold	0	600
9	4	2	5	Call	200	200
10	1	3	1	Check	0	450
11	2	3	2	Bet	300	400
12	3	3	5	Fold	0	200
13	1	4	6	Raise	100	900
14	2	4	7	Call	100	900
15	3	4	8	Fold	0	1000
16	4	4	9	Call	100	900
17	5	4	10	Call	100	900
18	1	5	6	Check	0	900
19	2	5	7	Bet	200	700
20	3	5	9	Call	200	700
21	4	5	10	Fold	0	900
22	1	6	6	Check	0	900
23	2	6	7	Check	0	700
24	3	6	9	Check	0	700
25	1	7	6	Bet	300	600
26	2	7	7	Fold	0	700
27	3	7	9	Call	300	400
28	1	8	1	Call	50	450

id	action_type	amount	match_id	player_id
1	1	1000	1	1
2	1	1000	1	2
3	1	1000	1	3
4	1	1000	1	4
5	1	1000	1	5
6	1	1000	2	6
7	1	1000	2	7
8	1	1000	2	8
9	1	1000	2	9
10	1	1000	2	10
11	2	1600	3	1
12	2	700	3	2
13	2	850	3	3
14	2	1300	3	4

id	action_number	round_id	player_id	action_type	amount	ending_balance
1	1	1	1	Call	50	450
2	2	1	2	Raise	100	900
3	3	1	3	Fold	0	1500
4	4	1	4	Call	100	400
5	5	1	5	Call	100	400
6	1	2	1	Check	0	450
7	2	2	2	Bet	200	700
8	3	2	4	Fold	0	600
9	4	2	5	Call	200	200
10	1	3	1	Check	0	450
11	2	3	2	Bet	300	400
12	3	3	5	Fold	0	200
13	1	4	6	Raise	100	900
14	2	4	7	Call	100	900
15	3	4	8	Fold	0	1000
16	4	4	9	Call	100	900
17	5	4	10	Call	100	900
18	1	5	6	Check	0	900
19	2	5	7	Bet	200	700
20	3	5	9	Call	200	700
21	4	5	10	Fold	0	900
22	1	6	6	Check	0	900
23	2	6	7	Check	0	700
24	3	6	9	Check	0	700
25	1	7	6	Bet	300	600
26	2	7	7	Fold	0	700
27	3	7	9	Call	300	400
28	1	8	1	Call	50	450

id	action_type	amount	match_id	player_id
1	1	1000	1	1
2	1	1000	1	2
3	1	1000	1	3
4	1	1000	1	4
5	1	1000	1	5
6	1	1000	2	6
7	1	1000	2	7
8	1	1000	2	8
9	1	1000	2	9
10	1	1000	2	10
11	2	1600	3	1
12	2	700	3	2
13	2	850	3	3
14	2	1300	3	4

HAVING	774				


```
mysql> SELECT * FROM matches;
+----+--------------+---------+
| id | match_number | game_id |
+----+--------------+---------+
|  1 |            1 |       1 |
|  2 |            2 |       1 |
|  3 |            3 |       2 |
|  4 |            4 |       2 |
|  5 |            5 |       3 |
+----+--------------+---------+
```

```
mysql> SELECT * FROM game;
+----+------------+
| id | date       |
+----+------------+
|  1 | 2023-01-01 |
|  2 | 2023-02-02 |
|  3 | 2023-03-03 |
```

```
%s AND

AVG(r.pot_size) > %s AND

MAX(r.pot_size) > %s
```

```
mysql> SELECT * FROM game_Season;
+------------+---------------+
| date       | season_number |
+------------+---------------+
| 2023-01-01 |            23 |
| 2023-02-02 |            23 |
| 2023-03-03 |            23 |
| 2023-04-04 |            23 |
| 2023-05-05 |            23 |
| 2023-06-06 |            23 |
| 2023-07-07 |            24 |
| 2023-08-08 |            24 |
+------------+---------------+
```

```
mysql> SELECT * FROM round;
+----+--------------+----------+----------+----------+----------+----------+
| id | round_number | match_id | pot_size | card1_id | card2_id | card3_id |
+----+--------------+----------+----------+----------+----------+----------+
|  1 |            1 |        1 |      300 |       12 |        7 |       33 |
|  4 |            1 |        2 |      200 |        3 |       28 |       17 |
|  8 |            1 |        3 |      500 |        8 |       46 |       29 |
| 12 |            1 |        4 |      350 |       18 |        4 |       34 |
| 16 |            1 |        5 |      250 |       17 |       34 |       39 |
| 19 |            1 |        6 |      550 |       26 |       43 |       42 |
| 23 |            1 |        7 |      300 |        2 |       38 |       36 |
| 27 |            1 |        8 |      400 |        3 |       22 |        5 |
| 31 |            1 |        9 |      500 |       37 |       31 |       16 |
| 35 |            1 |       10 |      350 |       23 |       28 |       27 |
| 39 |            1 |       11 |      450 |       22 |       52 |        7 |
| 43 |            1 |       12 |      300 |       16 |        5 |       29 |
| 47 |            1 |       13 |      500 |       16 |       11 |       27 |
```

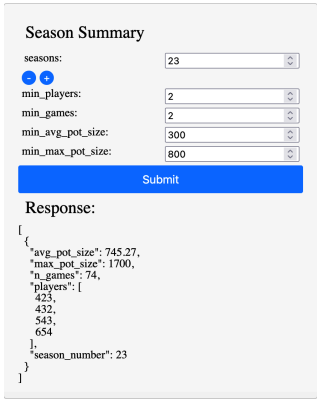
<table>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>

```
mysql> SELECT * FROM buy_in_cash_out;
+----+-------------+--------+----------+-----------+
| id | action_type | amount | match_id | player_id |
+----+-------------+--------+----------+-----------+
|  1 |           1 |   1000 |        1 |         1 |
|  2 |           1 |   1000 |        1 |         2 |
|  3 |           1 |   1000 |        1 |         3 |
|  4 |           1 |   1000 |        1 |         4 |
|  5 |           1 |   1000 |        1 |         5 |
|  6 |           1 |   1000 |        2 |         6 |
|  7 |           1 |   1000 |        2 |         7 |
|  8 |           1 |   1000 |        2 |         8 |
+----+-------------+--------+----------+-----------+
```

</td>
<td></td>
</tr>
<tr>
<td>nested GROUP BY</td>
<td>774</td>
<td>

```
SELECT
        gs.season_number AS season_number,
        COUNT(g.id) AS n_games,
        AVG(r.pot_size) AS avg_pot_size,
        MAX(r.pot_size) AS max_pot_size
        FROM Game g
        INNER JOIN game_seasons ON g.date = gs.date
        INNER JOIN matches m ON m.game_id = g.id
        INNER JOIN round r ON r.match_id = m.id
        INNER JOIN buy_in_cash_out bico ON bico.match_id = m.id
        {"WHERE gs.season_number IN (%s)" if seasons else "WHERE gs.season_number NOT IN (%s)"}
        GROUP BY gs.season_number
        HAVING COUNT(bico.player_id) > %s AND
        COUNT(g.id) > %s AND
        AVG(r.pot_size) > %s AND
        MAX(r.pot_size) > %s
```

</td>
<td>



</td>
<td>

```
+------------+----------------+
| date       | season_number  |
+------------+----------------+
| 2023-01-01 |             23 |
| 2023-02-02 |             23 |
| 2023-03-03 |             23 |
| 2023-04-04 |             23 |
| 2023-05-05 |             23 |
| 2023-06-06 |             23 |
+------------+----------------+
```

```
+----+--------------+----------+
| id | match_number | game_id  |
+----+--------------+----------+
|  1 |            1 |        1 |
|  2 |            2 |        1 |
|  3 |            3 |        2 |
|  4 |            4 |        2 |
|  5 |            5 |        3 |
|  6 |            6 |        3 |
|  7 |            7 |        4 |
+----+--------------+----------+
```

</td>
<td>

```
+------------+----------------+
| date       | season_number  |
+------------+----------------+
| 2023-01-01 |             23 |
| 2023-02-02 |             23 |
| 2023-03-03 |             23 |
| 2023-04-04 |             23 |
| 2023-05-05 |             23 |
| 2023-06-06 |             23 |
+------------+----------------+
```

```
+----+--------------+----------+
| id | match_number | game_id  |
+----+--------------+----------+
|  1 |            1 |        1 |
|  2 |            2 |        1 |
|  3 |            3 |        2 |
|  4 |            4 |        2 |
|  5 |            5 |        3 |
|  6 |            6 |        3 |
|  7 |            7 |        4 |
+----+--------------+----------+
```

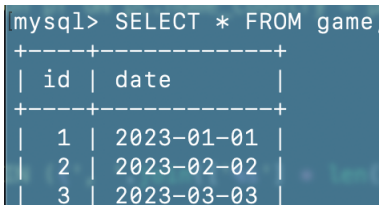
</td>
</tr>
</table>

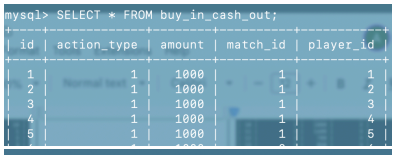
id	round_number	match_id	pot_size	card1_id	card2_id	card3_id
1	1	1	300	12	7	33
2	2	1	600	21	1	52
3	3	1	900	45	11	23
4	1	2	200	3	28	17
5	2	2	400	19	20	6
6	3	2	600	9	15	31
7	4	2	800	42	37	5
8	1	3	500	8	46	29
9	2	3	900	10	48	2
10	3	3	1300	24	35	50
11	4	3	1700	14	22	41
12	1	4	350	18	4	34

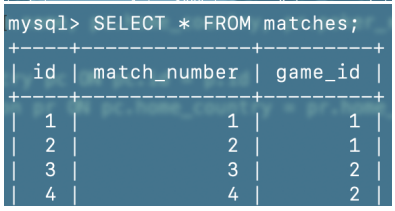
id	action_type	amount	match_id	player_id
1	1	1000	1	1
2	1	1000	1	2
3	1	1000	1	3
4	1	1000	1	4
5	1	1000	1	5
6	1	1000	2	6
7	1	1000	2	7
8	1	1000	2	8
9	1	1000	2	9
10	1	1000	2	10
11	2	1600	3	1
12	2	700	3	2
13	2	850	3	3
14	2	1300	3	4

id	round_number	match_id	pot_size	card1_id	card2_id	card3_id
1	1	1	300	12	7	33
2	2	1	600	21	1	52
3	3	1	900	45	11	23
4	1	2	200	3	28	17
5	2	2	400	19	20	6
6	3	2	600	9	15	31
7	4	2	800	42	37	5
8	1	3	500	8	46	29
9	2	3	900	10	48	2
10	3	3	1300	24	35	50
11	4	3	1700	14	22	41
12	1	4	350	18	4	34

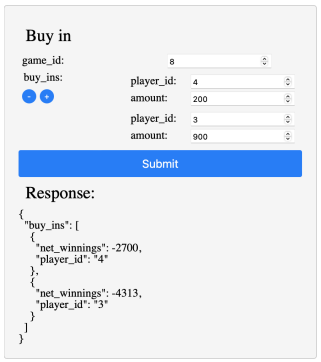
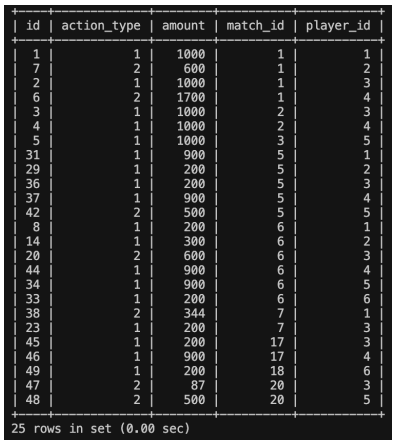
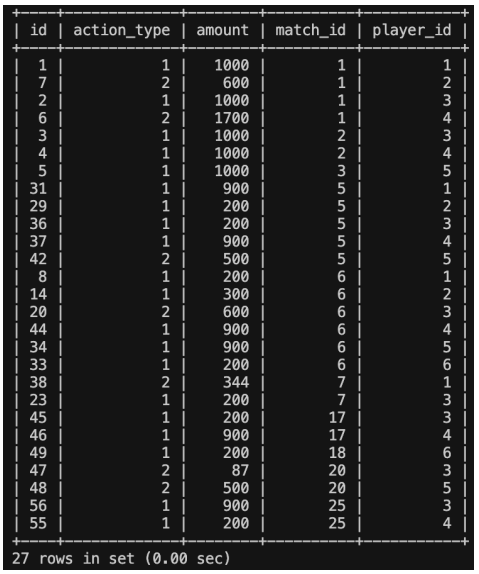
id	action_type	amount	match_id	player_id
1	1	1000	1	1
2	1	1000	1	2
3	1	1000	1	3
4	1	1000	1	4
5	1	1000	1	5
6	1	1000	2	6
7	1	1000	2	7
8	1	1000	2	8
9	1	1000	2	9
10	1	1000	2	10
11	2	1600	3	1
12	2	700	3	2
13	2	850	3	3
14	2	1300	3	4

DIVISION	728
```
SELECT DISTINCT p.id,
p.name, pc.home_country,
pr.poker_region,
p.net_winnings
   FROM player p
   INNER JOIN
player_country pc ON
pc.id = p.id
   INNER JOIN
player_region pr ON
pc.home_country =
pr.home_country
   WHERE NOT EXISTS (
      (
          SELECT game.id
          FROM game
          WHERE game.id
IN ({', '.join(['%s'] *
len(game_ids))})
      )
      EXCEPT
      (
          SELECT game.id
          FROM
buy_in_cash_out bico
          INNER JOIN
matches ON matches.id =
bico.match_id
          INNER JOIN
game ON matches.game_id =
game.id
          WHERE
bico.player_id = p.id AND
             game.id IN
```
 | 
 | 




 | |

		`(', '.join(['%s'] * len(game_ids)))` `)` `);`			

INSERT	150	`INSERT INTO Buy_in_cash_out (action_type, amount, match_id, player_id) VALUES (%s, %s, (SELECT id FROM Matches WHERE game_id = %s ORDER BY match_number DESC LIMIT 1), %s)`			

DELETE	122	`DELETE FROM Game WHERE id = %s`	Delete Game
			game_id: 6
			Submit
			Response:
			{
			"game_id": "6"
			}
				+----+------------+
				| id | date |
				+----+------------+
				| 1 | 2023-01-01 |
				| 2 | 2023-02-02 |
				| 3 | 2023-03-03 |
				| 6 | 2024-04-09 |
				| 8 | 2024-04-09 |
				+----+------------+
				5 rows in set (0.00 sec)
					+----+------------+
					| id | date |
					+----+------------+
					| 1 | 2023-01-01 |
					| 2 | 2023-02-02 |
					| 3 | 2023-03-03 |
					| 8 | 2024-04-09 |
					+----+------------+
					4 rows in set (0.00 sec)