

CPSC 304 Project Cover Page

Milestone # : 2

Date : Mar 1, 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.)

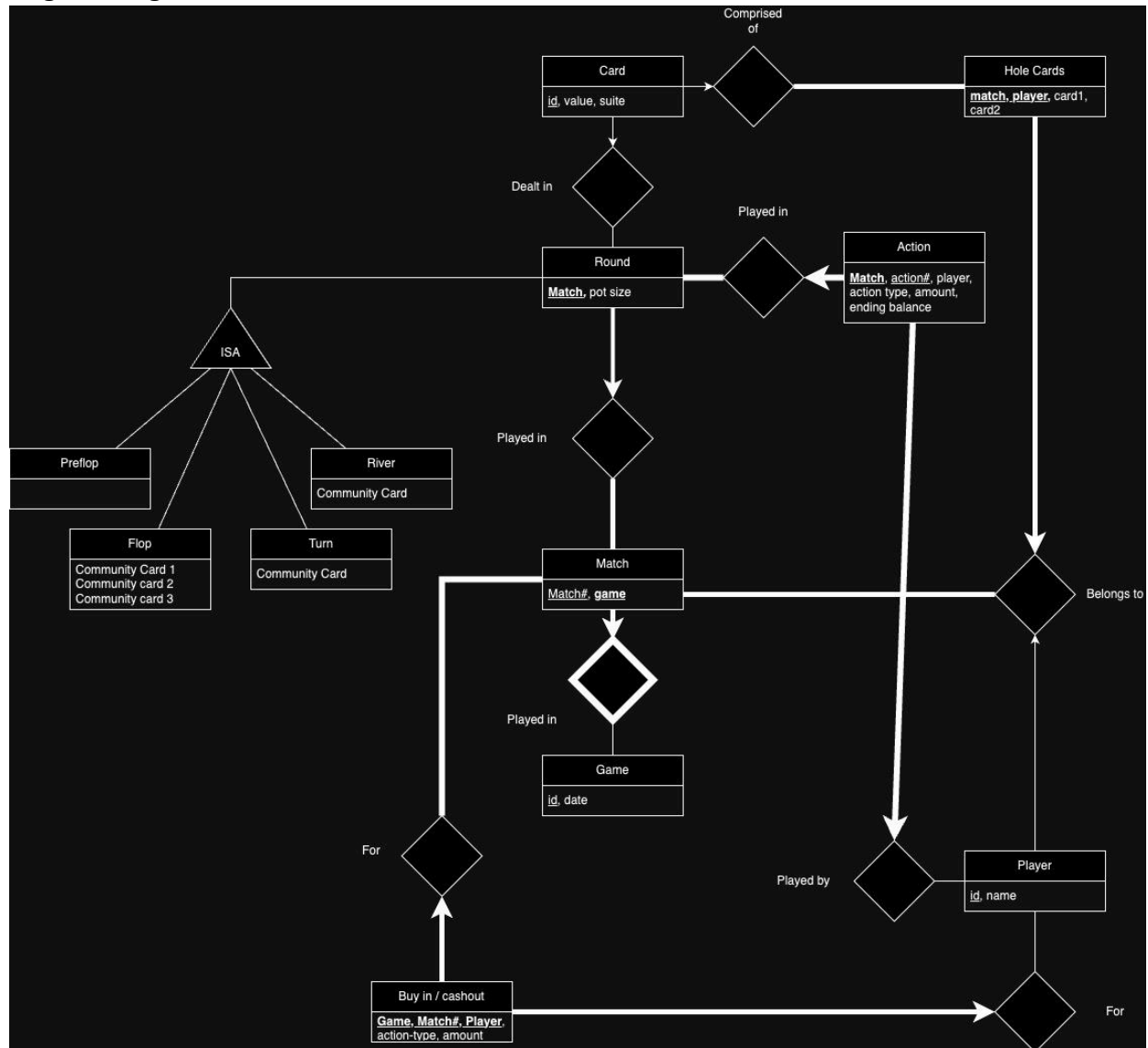
In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

2. Project Summary

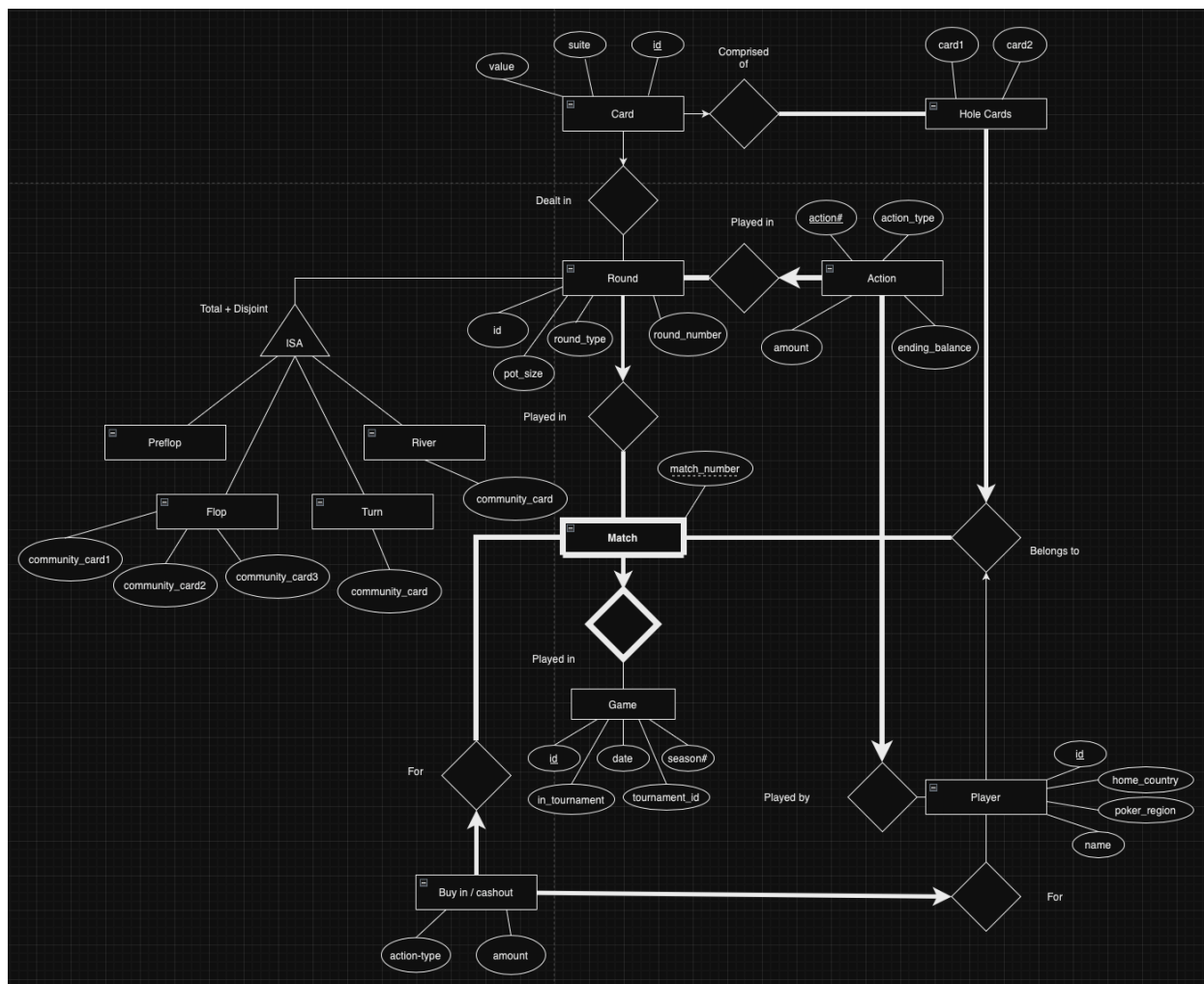
We are building an application to record games of Texas Hold'em Poker (no limit). The database will model the game itself, and some information about the players, but not information about game strategy. Using this database, users will be able to record how previous games of poker played out, and it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

3. Updated Diagram

Original Diagram



Revised Diagram



We made the following changes:

- Changed the attributes to follow the convention showed in the textbook
- Clearly defined Match to be a weak entity
- Added note to ISA specifying the participation constraint

If the diagram is hard to see, you can look at the link [here](#)

4. Relational Schema

Note that all Primary Keys are implicitly NOT NULL, and their combinations are UNIQUE.

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,)

Action(

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)

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)

5. Functional Dependencies

1. Card

- value, suite \rightarrow id

2. Hole Cards

- match_id, player_id \rightarrow card1, card2, id

3. Round

- round_number, match_id \rightarrow id, round_type, pot_size, card1_id, card2_id, card3_id
- round_number \rightarrow round_type

4. Action

- player_id, action_number, round_id \rightarrow id, action_type, amount, ending_balance

5. Game

- id \rightarrow date, season_number, in_tournament, tournament_id
- date \rightarrow season_number
- in_tournament, date \rightarrow tournament_id

6. Player

- id \rightarrow name, home_country, poker_region
- home_country \rightarrow poker region

7. Buy in / cash out

- match_id, player_id \rightarrow id, action_type, amount,

8. Match

- match_number, game_id \rightarrow id

6. Normalization

Tables **Card**, **Hole Cards**, **Action**, **Buy in / cash out**, and **Match** do not require decomposition since none of their FDs violate any BCNF conditions.

Round:

For the table Round, FD2 violates BCNF because round_number is not a superkey, yet it determines round_type. Therefore, we decompose the relation into two relations.

R1(id, round_number, pot_size, match_id, card1_id, card2_id, card3_id)

R2(round_number, round_type)

So, we have two relations for table Round: R1(id, round_number, match_id, card1_id, card2_id, card3_id), R2(round_number, round_type)

Game

For the table Game, FD2 and FD3 violate BCNF because date and (in_tournament, date) are not a superkey, yet they determine season_number and tournament_id. Therefore, we decompose the relation into two relations.

R1(id, date, in_tournament, tournament_id)

R2(date, season_number)

We need to decompose R1 further as it violates FD1.

R3(id, date)

R4(date, in_tournament, tournament_id)

So, we have three relations for table Game: R2(date, season_number), R3(id, date), R4(date, in_tournament, tournament_id).

Player

For the table Player, FD2 violates BCNF because home_country is not a superkey, yet it determines poker_region. Therefore, we decompose the relation into two relations.

R1(id, name, home_country)

R2(home_country, poker_region)

We need to decompose R1 further as it violates FD1.

R3(id, name)

R4(id, home_country)

University of British Columbia, Vancouver

Department of Computer Science

So, we have three relations for table Player: R2(home_country, poker_region), R3(id, name), R4(id, home_country).

Final Tables

List of tables, their primary keys, their candidate keys, and their foreign keys after normalization:

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_R2(

round_number: INTEGER,
round_type: TEXT NOT NULL)

Round_R1(

id: INTEGER UNIQUE,
pot_size: FLOAT NOT NULL
round_number: INTEGER NOT NULL,
match_id: INTEGER NOT NULL,
card1_id: INTEGER,
card2_id: INTEGER,
card3_id: INTEGER)

Action(

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

Match(

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

Game_R2(
 date: DATE,
 season_number: INTEGER)

Game_R3(
 id: INTEGER,
 date: DATE)

Game_R4(
 date: DATE,
 in_tournament: BOOLEAN,
 tournament_id: INTEGER)

Player_R2(
 home_country: TEXT,
 poker_region: TEXT)

Player_R3(
 id: INTEGER,
 name: TEXT)

Player_R4(
 id: INTEGER,
 home_country: TEXT)

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)

7. SQL DDL

```
CREATE TABLE IF NOT EXISTS Card(  
    id INT UNIQUE AUTO_INCREMENT,  
    value INT NOT NULL,  
    suite CHAR(1) NOT NULL,  
    PRIMARY KEY (value, suite)  
);
```

```
CREATE TABLE IF NOT EXISTS Game_R2(  
    date DATE PRIMARY KEY,  
    season_number INT  
);
```

```
CREATE TABLE IF NOT EXISTS Game_R3(  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    date DATE,  
    FOREIGN KEY (date) REFERENCES Game_R2(date) ON DELETE SET NULL  
);
```

```
CREATE TABLE IF NOT EXISTS Game_R4(  
    date DATE PRIMARY KEY,  
    in_tournament BOOLEAN,  
    tournament_id INT,  
    FOREIGN KEY (date) REFERENCES Game_R2(date) ON DELETE RESTRICT  
);
```

```
CREATE TABLE IF NOT EXISTS Match(  
    id INT UNIQUE AUTO_INCREMENT,  
    match_number INT NOT NULL,  
    game_id INT NOT NULL,  
    PRIMARY KEY (match_number, game_id),  
    FOREIGN KEY (game_id) REFERENCES Game_R3(id) ON DELETE CASCADE  
);
```

```
CREATE TABLE IF NOT EXISTS Player_R2(  
    home_country VARCHAR PRIMARY KEY,  
    poker_region VARCHAR  
);
```

```
CREATE TABLE IF NOT EXISTS Player_R3(  
    id INT PRIMARY KEY AUTO_INCREMENT,
```

University of British Columbia, Vancouver

Department of Computer Science

```
        name VARCHAR
    );

CREATE TABLE IF NOT EXISTS Player_R4(
    id INT PRIMARY KEY AUTO_INCREMENT,
    home_country VARCHAR,
    FOREIGN KEY (id) REFERENCES PLAYER_R3(id) ON DELETE CASCADE
);

CREATE TABLE IF NOT EXISTS Hole_Cards(
    id UNIQUE AUTO_INCREMENT,
    match_id INT,
    player_id INT,
    card1 INT NOT NULL,
    card2 INT NOT NULL,
    PRIMARY KEY (match_id, player_id),
    FOREIGN KEY (match_id) REFERENCES Match(id) ON DELETE CASCADE,
    FOREIGN KEY (player_id) REFERENCES Player_R3(id) ON DELETE RESTRICT,
    FOREIGN KEY (card1) REFERENCES Card(id) ON DELETE RESTRICT,
    FOREIGN KEY (card2) REFERENCES Card(id) ON DELETE RESTRICT
);

CREATE TABLE IF NOT EXISTS Round_R2(
    round_number INT PRIMARY KEY,
    round_type VARCHAR NOT NULL
);

CREATE TABLE IF NOT EXISTS Round_R1(
    id INT PRIMARY KEY AUTO_INCREMENT,
    round_number INT NOT NULL,
    match_id INT NOT NULL,
    pot_size FLOAT NOT NULL,
    card1_id INT,
    card2_id INT,
    card3_id INT,
    UNIQUE KEY (round_number, match_id),
    FOREIGN KEY (round_number) REFERENCES Round_R2(round_number) ON DELETE
    RESTRICT,
    FOREIGN KEY (match_id) REFERENCES Match(id) ON DELETE CASCADE,
    FOREIGN KEY (card1_id) REFERENCES Card(id) ON DELETE RESTRICT,
    FOREIGN KEY (card2_id) REFERENCES Card(id) ON DELETE RESTRICT,
    FOREIGN KEY (card3_id) REFERENCES Card(id) ON DELETE RESTRICT
);
```

```
CREATE TABLE IF NOT EXISTS Action(  
    id INT UNIQUE AUTO_INCREMENT,  
    action_number INT,  
    round_id INT NOT NULL,  
    player_id INT NOT NULL,  
    action_type VARCHAR NOT NULL,  
    amount FLOAT,  
    ending_balance FLOAT NOT NULL,  
    PRIMARY KEY (action_number, round_id, player_id),  
    FOREIGN KEY (round_id) REFERENCES Round_R1(id) ON DELETE CASCADE,  
    FOREIGN KEY (player_id) REFERENCES Player_R3(id) ON DELETE CASCADE,  
    FOREIGN KEY (action_number) REFERENCES Round_R2(action_number) ON DELETE  
CASCADE  
);
```

```
CREATE TABLE IF NOT EXISTS Buy_in_cash_out(  
    id INT UNIQUE AUTO_INCREMENT,  
    action_type INT NOT NULL,  
    amount FLOAT NOT NULL,  
    match_id INT NOT NULL,  
    player_id INT NOT NULL),  
    PRIMARY KEY (match_id, player_id),  
    FOREIGN KEY (match_id) REFERENCES match(id) ON DELETE RESTRICT,  
    FOREIGN KEY (player_id) REFERENCES player(id) ON DELETE RESTRICT  
);
```

8. Insert Statements

Card

```
INSERT INTO Card (value, suite) VALUES
```

```
(1, D),
```

```
(1, C),
```

```
(1, H),
```

```
(1, S),
```

```
(2, D);
```

Game

```
INSERT INTO Game_R2 (date, season_number) VALUES
```

```
('2023-01-01', 1),
```

```
('2022-02-15', 1),
```

```
('2021-03-20', 2),
```

```
('2020-04-10', 2),
```

```
('2022-05-05', 3);
```

```
INSERT INTO Game_R3 (date) VALUES
```

```
(1, '2024-01-01'),
```

```
(2, '2024-02-15'),
```

```
(3, '2024-03-20'),
```

```
(4, '2024-04-10'),
```

```
(5, '2024-05-05');
```

```
INSERT INTO Game_R4 (date, in_tournament, tournament_id) VALUES
```

```
('2024-01-01', true, 101),
```

```
('2024-02-15', false, null),
```

```
('2024-03-20', true, 102),
```

```
('2024-04-10', false, null),
```

```
('2024-05-05', true, 103);
```

Match

```
INSERT INTO Match (match_number, game_id) VALUES
```

```
(1, 1),
```

```
(2, 2),
```

```
(3, 3),
```

```
(4, 4),
```

```
(5, 5);
```

Player

```
INSERT INTO Player_R2 (home_country, poker_region) VALUES
```

```
('USA', 'North America'),
```

University of British Columbia, Vancouver

Department of Computer Science

```
('Canada', 'North America'),  
( 'Germany', 'Europe'),  
( 'Australia', 'Oceania'),  
( 'Japan', 'Asia');
```

```
INSERT INTO Player_R3 (name) VALUES  
(1, 'Player A'),  
(2, 'Player B'),  
(3, 'Player C'),  
(4, 'Player D'),  
(5, 'Player E');
```

```
INSERT INTO Player_R4 (home_country) VALUES  
(1, 'USA'),  
(2, 'Canada'),  
(3, 'Germany'),  
(4, 'Australia'),  
(5, 'Japan');
```

Hole_Cards

```
INSERT INTO Hole_Cards (match_id, player_id, card1, card2) VALUES  
(1, 1, 1, 5),  
(2, 2, 2, 3),  
(3, 3, 3, 4),  
(4, 4, 4, 2),  
(5, 5, 5, 1);
```

Round

```
INSERT INTO Round_R2 (round_number, round_type) VALUES  
(1, 'Pre-flop'),  
(2, 'Flop'),  
(3, 'Turn'),  
(4, 'River'),  
(5, 'Showdown');
```

```
INSERT INTO Round_R1 (round_number, match_id, pot_size, card1_id, card2_id, card3_id)  
VALUES  
(1, 1, 100.0, NULL, NULL, NULL),  
(2, 2, 150.0, 2, 3, 4),  
(3, 3, 200.0, 3, NULL, NULL),  
(4, 4, 120.0, 4, NULL, NULL),  
(5, 5, 180.0, NULL, NULL, NULL);
```

Action

```
INSERT INTO Action (action_number, round_id, player_id, action_type, amount,
ending_balance) VALUES
(1, 1, 1, 'Bet', 50.0, 50.0),
(2, 2, 2, 'Raise', 30.0, 80.0),
(3, 3, 3, 'Call', 40.0, 120.0),
(4, 4, 4, 'Fold', NULL, 80.0),
(5, 5, 5, 'All-in', 180.0, 0.0);
```

Buy_in_cash_out

```
INSERT INTO Buy_in_cash_out (action_type, amount, match_id, player_id) VALUES
(1, 100.0, 1, 1),
(2, 50.0, 2, 2),
(1, 200.0, 3, 3),
(2, 30.0, 4, 4),
(1, 150.0, 5, 5);
```