



Riichi Mahjong

game tracking and analytics

Nikita Paniukhin



Game overview

My database models competitive Riichi Mahjong games. It tracks key gameplay elements such as **players**, **games**, **player hands**, **tile compositions**, and **scoring patterns** known as "Yaku".

Each game involves **four players**, each occupying a seat with a **wind direction**. Players start with a **score**, which changes over the course of the game. Each player maintains a **hand of 13 tiles**, with a 14th tile drawn to complete the hand when declaring a win. Some tiles may be part of **melds** — **groups** of revealed tiles.

Yaku are used to evaluate whether a player's hand is valid and how much it scores. The game also has a "**Riichi**" **mechanic**, which allows a player to pay a bounty and declare, that they're ready to win.

This project is inspired by the existing system called Pantheon and aims to fully replicate it.



Scenarios

- Recording a new game and its participants, including scores and outcomes
- Tracking each player's hand and declared Riichi
- Storing which Yaku combinations were achieved by players during a game
- Efficient querying of past game results, player histories, or specific hand patterns



Entities

- `players` — registered users with a rating, username, and optional country
- `games` — individual Mahjong matches with time data, wind round, and winner/dealer references
- `yaku` — valid winning hand patterns, each with name, score value (Han), and restrictions
- `tiles` — definitions of Mahjong tiles (e.g., 1m, 5p, 0s, E, C)

Meta-entities:

- `player_status` — each player's state in a game — wind, score, Riichi status, etc.
- `player_yaku` — maps which `player_status` earned which Yaku in a game
- `hand_tile` — maps which tiles are in each player's hand, their position, and meld status

yaku	
id	SERIAL
name	TEXT NN
han_value	INT
menzenchin_only	BOOLEAN
description	TEXT

player_yakus	
id	SERIAL
player_status_id	INT
yaku_id	INT

hand_tile	
id	SERIAL
player_status_id	INT
tile_code	TEXT NN
is_in_meld	BOOLEAN
position	INT

players	
id	SERIAL
username	TEXT NN
display_name	TEXT
rating	INTEGER
country	TEXT

player_status	
id	SERIAL
game_id	INT
player_id	INT
wind	TEXT
cur_score	INT
declared_riichi	BOOLEAN

games	
id	SERIAL
start_time	TIMESTAMP
end_time	TIMESTAMP
winner_id	INT
wind	TEXT
dealer_id	INT

Schema



DBML & SQL: Players, Games

```
Table players {  
  id SERIAL [pk]  
  username TEXT [unique, not null]  
  display_name TEXT  
  rating INTEGER [default: 1500]  
  country TEXT  
}
```

```
Table games {  
  id SERIAL [pk]  
  start_time TIMESTAMP  
  end_time TIMESTAMP  
  winner_id INT [ref: > players.id]  
  wind TEXT  
  dealer_id INT [ref: > players.id]  
}
```

```
CREATE TABLE players (  
  id INTEGER PRIMARY KEY AUTOINCREMENT,  
  username TEXT UNIQUE NOT NULL,  
  display_name TEXT,  
  rating INTEGER DEFAULT 1500,  
  country TEXT  
);
```


```
CREATE TABLE games (  
  id INTEGER PRIMARY KEY AUTOINCREMENT,  
  start_time TIMESTAMP,  
  end_time TIMESTAMP,  
  winner_id INTEGER,  
  wind TEXT CHECK (wind IN ('East', 'South')),  
  dealer_id INTEGER,  
  FOREIGN KEY (winner_id) REFERENCES players(id),  
  FOREIGN KEY (dealer_id) REFERENCES players(id)  
);
```



DBML & SQL: Yaku, Hand Tiles

```
Table yaku {  
  id SERIAL [pk]  
  name TEXT [unique, not null]  
  han_value INT  
  menzenchin_only BOOLEAN  
  description TEXT  
}  
  
Table hand_tile {  
  id SERIAL [pk]  
  player_status_id INT  
    [ref: > player_status.id]  
  tile_id INT [ref: > tiles.id]  
  is_in_meld BOOLEAN  
  position INT  
}
```

```
CREATE TABLE yaku (  
  id INTEGER PRIMARY KEY AUTOINCREMENT,  
  name TEXT UNIQUE NOT NULL,  
  han_value INTEGER,  
  menzenchin_only BOOLEAN,  
  description TEXT  
);  
  
CREATE TABLE hand_tile (  
  id INTEGER PRIMARY KEY AUTOINCREMENT,  
  player_status_id INTEGER,  
  tile_id INTEGER,  
  is_in_meld BOOLEAN,  
  position INTEGER,  
  FOREIGN KEY (player_status_id) REFERENCES player_status(id),  
  FOREIGN KEY (tile_id) REFERENCES tiles(id),  
  CHECK (is_in_meld = TRUE OR position IS NOT NULL)  
);
```



DBML & SQL: Player Status, Player Yaku

```
Table player_status {  
  id SERIAL [pk]  
  game_id INT [ref: > games.id]  
  player_id INT [ref: > players.id]  
  wind TEXT  
  cur_score INT [default: 25000]  
  declared_riichi BOOLEAN  
}
```

```
Table player_yakus {  
  id SERIAL [pk]  
  player_status_id INT  
    [ref: > player_status.id]  
  yaku_id INT [ref: > yaku.id]  
}
```

```
CREATE TABLE player_status (  
  id INTEGER PRIMARY KEY AUTOINCREMENT,  
  game_id INTEGER,  
  player_id INTEGER,  
  wind TEXT CHECK (wind IN ('East', 'South', 'West', 'North')),  
  cur_score INTEGER DEFAULT 25000,  
  declared_riichi BOOLEAN,  
  FOREIGN KEY (game_id) REFERENCES games(id),  
  FOREIGN KEY (player_id) REFERENCES players(id)  
);
```

```
CREATE TABLE player_yakus (  
  id INTEGER PRIMARY KEY AUTOINCREMENT,  
  player_status_id INTEGER,  
  yaku_id INTEGER,  
  FOREIGN KEY (player_status_id) REFERENCES player_status(id),  
  FOREIGN KEY (yaku_id) REFERENCES yaku(id)  
);
```




Indexes

Data integrity is enforced through foreign keys, constraints (e.g., tile position validity for non-melded tiles), and unique indexes (e.g., one tile per position in a closed hand). Indexes are optimized for efficient querying of player records, hands, and scoring combinations.

```
CREATE UNIQUE INDEX unique_hand_position ON hand_tile(player_status_id, position) WHERE is_in_meld = FALSE;
```

```
CREATE INDEX idx_games_winner_id ON games(winner_id);  
CREATE INDEX idx_games_dealer_id ON games(dealer_id);  
CREATE INDEX idx_player_status_game_id ON player_status(game_id);  
CREATE INDEX idx_player_status_player_id ON player_status(player_id);  
CREATE INDEX idx_player_yaku_status_id ON player_yaku(player_status_id);  
CREATE INDEX idx_player_yaku_yaku_id ON player_yaku(yaku_id);  
CREATE INDEX idx_hand_tile_player_status_id ON hand_tile(player_status_id);  
CREATE INDEX idx_hand_tile_tile_code ON hand_tile(tile_code);
```

Sample data

in cloud-hosted database



games

Add data

Manage table ▾

Columns

Rows

6

100

hand_tile

Add data

Manage table ▾

Columns

Rows

5

14000

player_status

Add data

Manage table ▾

Columns

Rows

6

1000

1 - 25 of 100 rows

<< < 1 of 4 > >>

Rows per page: 25 ▾

Delete (0) rows

<input type="checkbox"/>	id ↕	start_time ↕	end_time ↕	winner_id ↕	wind ↕	dealer_id ↕	
<input type="checkbox"/>	1	2025-04-13	<null>	<null>	South	3	...
<input type="checkbox"/>	2	2025-01-03	2025-02-06	10	South	5	...
<input type="checkbox"/>	3	2025-02-10	<null>	<null>	South	4	...
<input type="checkbox"/>	4	2025-03-31	2025-04-07	1	East	9	...
<input type="checkbox"/>	5	2025-05-27	<null>	<null>	East	5	...
<input type="checkbox"/>	6	2025-02-14	2025-02-27	9	South	9	...
<input type="checkbox"/>	7	2025-05-17	<null>	<null>	East	9	...
<input type="checkbox"/>	8	2025-01-31	2025-02-28	4	South	4	...
<input type="checkbox"/>	9	2025-05-25	<null>	<null>	South	2	...
<input type="checkbox"/>	10	2025-03-30	<null>	<null>	East	4	...
<input type="checkbox"/>	11	2025-01-11	2025-02-04	10	South	10	...
<input type="checkbox"/>	12	2025-04-24	2025-05-15	1	South	5	...
<input type="checkbox"/>	13	2025-03-30	<null>	<null>	South	10	...
<input type="checkbox"/>	14	2025-05-27	2025-04-01	9	East	9	...
<input type="checkbox"/>	15	2025-03-05	<null>	<null>	East	2	...
<input type="checkbox"/>	16	2025-01-26	<null>	<null>	South	8	...
<input type="checkbox"/>	17	2025-01-27	<null>	<null>	East	2	...
<input type="checkbox"/>	18	2025-01-13	<null>	<null>	South	5	...
<input type="checkbox"/>	19	2025-03-15	2025-01-04	5	South	5	...
<input type="checkbox"/>	20	2025-02-22	2025-05-18	4	South	2	...
<input type="checkbox"/>	21	2025-05-23	2025-02-05	6	East	7	...
<input type="checkbox"/>	22	2025-03-08	<null>	<null>	South	3	...
<input type="checkbox"/>	23	2025-02-25	2025-04-03	7	East	7	...
<input type="checkbox"/>	24	2025-04-14	<null>	<null>	East	9	...
<input type="checkbox"/>	25	2025-03-18	2025-03-07	6	South	6	...

1 - 25 of 14,000 rows

<< < 1 of 560 > >>

Rows per page: 25 ▾

Delete (0) rows

<input type="checkbox"/>	id ↕	player_status_id ↕	tile_code ↕	is_in_meld ↕	position ↕	
<input type="checkbox"/>	1	5	3b	false	0	...
<input type="checkbox"/>	2	5	6b	false	1	...
<input type="checkbox"/>	3	5	4m	false	2	...
<input type="checkbox"/>	4	5	9b	false	3	...
<input type="checkbox"/>	5	5	1m	false	4	...
<input type="checkbox"/>	6	5	4b	false	5	...
<input type="checkbox"/>	7	5	6m	false	6	...
<input type="checkbox"/>	8	5	6b	true	7	...
<input type="checkbox"/>	9	5	1b	false	8	...
<input type="checkbox"/>	10	5	9m	false	9	...
<input type="checkbox"/>	11	5	E	false	10	...
<input type="checkbox"/>	12	5	6p	true	11	...
<input type="checkbox"/>	13	5	3b	false	12	...
<input type="checkbox"/>	14	5	7b	true	13	...
<input type="checkbox"/>	15	15	5m	false	0	...
<input type="checkbox"/>	16	15	5b	false	1	...
<input type="checkbox"/>	17	15	9b	false	2	...
<input type="checkbox"/>	18	15	9p	false	3	...
<input type="checkbox"/>	19	15	9m	true	4	...
<input type="checkbox"/>	20	15	9p	true	5	...
<input type="checkbox"/>	21	15	2b	true	6	...
<input type="checkbox"/>	22	15	0p	false	7	...
<input type="checkbox"/>	23	15	7b	false	8	...
<input type="checkbox"/>	24	15	2b	false	9	...
<input type="checkbox"/>	25	15	9m	true	10	...

1 - 25 of 1,000 rows

<< < 1 of 40 > >>

Rows per page: 25 ▾

Delete (0) rows

<input type="checkbox"/>	id ↕	game_id ↕	player_id ↕	wind ↕	cur_score ↕	declared_riichi ↕	
<input type="checkbox"/>	1	26	6	West	-8447	true	...
<input type="checkbox"/>	2	26	3	North	15788	false	...
<input type="checkbox"/>	3	26	2	West	7333	false	...
<input type="checkbox"/>	4	26	9	North	97047	true	...
<input type="checkbox"/>	5	26	1	East	33342	true	...
<input type="checkbox"/>	6	26	5	West	-8647	false	...
<input type="checkbox"/>	7	26	8	South	74913	true	...
<input type="checkbox"/>	8	26	10	North	68894	false	...
<input type="checkbox"/>	9	26	7	West	77063	false	...
<input type="checkbox"/>	10	26	4	South	78520	true	...
<input type="checkbox"/>	11	35	6	South	93999	false	...
<input type="checkbox"/>	12	35	3	South	80657	true	...
<input type="checkbox"/>	13	35	2	East	95625	false	...
<input type="checkbox"/>	14	35	9	East	-22306	false	...
<input type="checkbox"/>	15	35	1	North	44615	false	...
<input type="checkbox"/>	16	35	5	West	56959	false	...
<input type="checkbox"/>	17	35	8	South	3095	true	...
<input type="checkbox"/>	18	35	10	West	32042	true	...
<input type="checkbox"/>	19	35	7	East	2188	true	...
<input type="checkbox"/>	20	35	4	East	34284	false	...
<input type="checkbox"/>	21	42	6	West	-11982	false	...
<input type="checkbox"/>	22	42	3	South	98440	false	...
<input type="checkbox"/>	23	42	2	South	7509	true	...
<input type="checkbox"/>	24	42	9	West	20312	true	...
<input type="checkbox"/>	25	42	1	West	-128	true	...

player_yaku

Add data

Manage table

Columns

Rows

3

495

players

Add data

Manage table

Columns

Rows

5

10

yaku

Add data

Manage table

Columns

Rows

5

34

1 - 25 of 495 rows

<< < 1 of 20 > >>

Rows per page: 25

Delete (0) rows

<input type="checkbox"/>	id	player_status_id	yaku_id	
<input type="checkbox"/>	1	5	25	...
<input type="checkbox"/>	2	15	30	...
<input type="checkbox"/>	3	35	3	...
<input type="checkbox"/>	4	65	6	...
<input type="checkbox"/>	5	75	25	...
<input type="checkbox"/>	6	85	34	...
<input type="checkbox"/>	7	95	4	...
<input type="checkbox"/>	8	125	29	...
<input type="checkbox"/>	9	155	4	...
<input type="checkbox"/>	10	165	12	...
<input type="checkbox"/>	11	175	14	...
<input type="checkbox"/>	12	195	33	...
<input type="checkbox"/>	13	205	6	...
<input type="checkbox"/>	14	245	13	...
<input type="checkbox"/>	15	255	13	...
<input type="checkbox"/>	16	265	29	...
<input type="checkbox"/>	17	295	11	...
<input type="checkbox"/>	18	305	24	...
<input type="checkbox"/>	19	315	28	...
<input type="checkbox"/>	20	335	11	...
<input type="checkbox"/>	21	345	30	...
<input type="checkbox"/>	22	355	18	...
<input type="checkbox"/>	23	365	22	...
<input type="checkbox"/>	24	375	24	...
<input type="checkbox"/>	25	385	32	...

1 - 10 of 10 rows

<< < 1 of 1 > >>

Rows per page: 10

Delete (0) rows

<input type="checkbox"/>	id	username	display_name	rating	country	
<input type="checkbox"/>	1	khunter	Sharon Manning	1689	Uganda	...
<input type="checkbox"/>	2	clarkjay	Kristin Phillips	1348	Vietnam	...
<input type="checkbox"/>	3	bjohnson	Anthony Humphrey	1538	Turkmenistan	...
<input type="checkbox"/>	4	vincent91	Daniel Sandoval	1470	Bermuda	...
<input type="checkbox"/>	5	kingfelicia	Monica Barton	1722	Mali	...
<input type="checkbox"/>	6	bgomez	Susan Scott	1574	Kazakhstan	...
<input type="checkbox"/>	7	vasquezlauren	Lisa Webb	1535	New Caledonia	...
<input type="checkbox"/>	8	meganyoung	Stephanie Owens	1700	Panama	...
<input type="checkbox"/>	9	fnelson	Jason Cross	1377	Saint Martin	...
<input type="checkbox"/>	10	michaelhamilton	Deborah Richards	1482	Malaysia	...

1 - 25 of 34 rows

<< < 1 of 2 > >>

Rows per page: 25

Delete (0) rows

Players: 10

Games: 100

500 completed yaku

14'000 individual tiles



Sample requests

```
SELECT p.username, p.display_name, COUNT(g.id) AS win_count
FROM games g
JOIN players p ON g.winner_id = p.id
GROUP BY g.winner_id
ORDER BY win_count DESC
LIMIT 1;
```

> Top winner:

Username: vincent91, Display name: Daniel Sandoval, Wins: 8

```
SELECT y.name, COUNT(py.id) AS count
FROM player_yaku py
JOIN yaku y ON py.yaku_id = y.id
GROUP BY py.yaku_id
ORDER BY count DESC
LIMIT 5;
```

> Top 5 most common Yaku:

Riichi:	24 times
SanshokuDoujun:	23 times
Toitoi:	21 times
Sankantsu:	20 times
Ikkitsuukan:	19 times



Sample requests

```
SELECT ht.tile_code, COUNT(*) AS usage_count
FROM hand_tile ht
GROUP BY ht.tile_code
ORDER BY usage_count DESC
LIMIT 10;
```

> Top 10 most used tiles:

```
4p: 433 times
7b: 421 times
N: 420 times
0p: 414 times
1b: 398 times
...
```

```
SELECT ht.tile_code, COUNT(*) AS meld_count
FROM hand_tile ht
WHERE ht.is_in_meld == "true"
GROUP BY ht.tile_code
ORDER BY meld_count DESC
LIMIT 10;
```

> Top 10 tiles used in melds:

```
4p: 227 times
7b: 211 times
2p: 210 times
6b: 210 times
9b: 207 times
...
```




Conclusion

Over the course of this project, I have:

- Designed and documented the structure of a system for tracking competitive Riichi Mahjong games.
- Developed a relational database schema tailored to the entities and gameplay of Riichi Mahjong.
- Implemented the schema in DBML and `runnable SQL` formats, making it and deployable.
- Deployed the database to a `cloud environment` with preconfigured API access.
- Wrote a script that connects to the database and populates it with a large set of randomized game data.
- Wrote a script that queries the `live database` to demonstrate sample analytics and data retrieval.

This project lays the groundwork for building full-featured applications on top of the existing database. The database can be easily extended in the future to support new features and additional gameplay logic.

<https://github.com/npanuhin/CU-Databases/tree/master/project>



DBML, SQL,
Python-scripts,
etc.