

## PostgreSQL Code

```
CREATE TYPE game_status_enum AS ENUM ('waiting', 'in_progress', 'finished');
```

```
CREATE TYPE card_color_enum AS ENUM ('red', 'blue', 'green', 'yellow', 'wild');
```

```
CREATE TYPE card_value_enum AS ENUM (  
    '0','1','2','3','4','5','6','7','8','9',  
    'skip','reverse','draw_two','wild','wild_draw_four'  
);
```

```
CREATE TYPE card_location_enum AS ENUM ('deck', 'discard', 'player_hand');
```

```
CREATE TYPE action_type_enum AS ENUM ('play','draw','skip','reverse','draw_two','wild');
```

```
CREATE TABLE users (  
    id BIGSERIAL PRIMARY KEY,  
    username VARCHAR(50) NOT NULL UNIQUE,  
    email VARCHAR(100) NOT NULL UNIQUE,  
    password VARCHAR(255) NOT NULL,  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

```
CREATE TABLE game_rooms (  
    id BIGSERIAL PRIMARY KEY,  
    title VARCHAR(100),  
    max_players INT NOT NULL DEFAULT 4,  
    password VARCHAR(100),  
    status game_status_enum NOT NULL DEFAULT 'waiting',  
    created_by BIGINT NOT NULL REFERENCES users(id) ON DELETE CASCADE,  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    started_at TIMESTAMP,  
    ended_at TIMESTAMP  
);
```

```
CREATE TABLE game_room_players (  
    id BIGSERIAL PRIMARY KEY,  
    user_id BIGINT NOT NULL REFERENCES users(id) ON DELETE CASCADE,  
    game_room_id BIGINT NOT NULL REFERENCES game_rooms(id) ON DELETE  
CASCADE,  
    is_game_master BOOLEAN DEFAULT FALSE,  
    player_order INT,  
    cards_in_hand INT DEFAULT 0,
```

```
joined_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

```
CREATE TABLE uno_cards (  
  id BIGSERIAL PRIMARY KEY,  
  color card_color_enum NOT NULL,  
  value card_value_enum NOT NULL  
);
```

```
CREATE TABLE game_room_decks (  
  id BIGSERIAL PRIMARY KEY,  
  game_room_id BIGINT NOT NULL REFERENCES game_rooms(id) ON DELETE  
CASCADE,  
  card_id BIGINT NOT NULL REFERENCES uno_cards(id) ON DELETE CASCADE,  
  location card_location_enum NOT NULL,  
  owner_player_id BIGINT REFERENCES game_room_players(id) ON DELETE CASCADE,  
  position_index INT  
);
```

```
CREATE TABLE game_turns (  
  id BIGSERIAL PRIMARY KEY,  
  game_room_id BIGINT NOT NULL REFERENCES game_rooms(id) ON DELETE  
CASCADE,  
  player_id BIGINT NOT NULL REFERENCES game_room_players(id) ON DELETE  
CASCADE,  
  card_played_id BIGINT REFERENCES uno_cards(id) ON DELETE SET NULL,  
  action_type action_type_enum NOT NULL,  
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

```
CREATE TABLE game_results (  
  id BIGSERIAL PRIMARY KEY,  
  game_room_id BIGINT NOT NULL REFERENCES game_rooms(id) ON DELETE  
CASCADE,  
  winner_id BIGINT REFERENCES users(id) ON DELETE SET NULL,  
  total_turns INT,  
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

```
CREATE TABLE game_result_players (  
  id BIGSERIAL PRIMARY KEY,
```

```

    game_result_id BIGINT NOT NULL REFERENCES game_results(id) ON DELETE
CASCADE,
    user_id BIGINT NOT NULL REFERENCES users(id) ON DELETE CASCADE,
    rank INT NOT NULL,
    cards_left INT DEFAULT 0
);

```

```

CREATE TABLE chat_messages (
    id BIGSERIAL PRIMARY KEY,
    user_id BIGINT NOT NULL REFERENCES users(id) ON DELETE CASCADE,
    game_room_id BIGINT REFERENCES game_rooms(id) ON DELETE CASCADE, -- NULL =
main lobby
    message TEXT NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

```

## DB Diagram Code

```

Enum game_status_enum {
    waiting
    in_progress
    finished
}

```

```

Enum card_color_enum {
    red
    blue
    green
    yellow
    wild
}

```

```

Enum card_value_enum {
    "0"
    "1"
    "2"
    "3"
    "4"
    "5"
    "6"
    "7"
    "8"
}

```

```
"g"  
skip  
reverse  
draw_two  
wild  
wild_draw_four  
}
```

```
Enum card_location_enum {  
    deck  
    discard  
    player_hand  
}
```

```
Enum action_type_enum {  
    play  
    draw  
    skip  
    reverse  
    draw_two  
    wild  
}
```

```
Table users {  
    id BIGSERIAL [pk]  
    username varchar(50) [not null, unique]  
    email varchar(100) [not null, unique]  
    password varchar(255) [not null]  
    created_at timestamp [default: `CURRENT_TIMESTAMP`]  
}
```

```
Table game_rooms {  
    id BIGSERIAL [pk]  
    title varchar(100)  
    max_players int [not null, default: 4]  
    password varchar(100)  
    status game_status_enum [not null, default: 'waiting']  
    created_by bigint [not null, ref: > users.id]  
    created_at timestamp [default: `CURRENT_TIMESTAMP`]  
    started_at timestamp  
    ended_at timestamp  
}
```

```
Table game_room_players {
```

```
id BIGSERIAL [pk]
user_id bigint [not null, ref: > users.id]
game_room_id bigint [not null, ref: > game_rooms.id]
is_game_master boolean [default: false]
player_order int
cards_in_hand int [default: 0]
joined_at timestamp [default: `CURRENT_TIMESTAMP`]
}
```

```
Table uno_cards {
  id BIGSERIAL [pk]
  color card_color_enum [not null]
  value card_value_enum [not null]
}
```

```
Table game_room_decks {
  id BIGSERIAL [pk]
  game_room_id bigint [not null, ref: > game_rooms.id]
  card_id bigint [not null, ref: > uno_cards.id]
  location card_location_enum [not null]
  owner_player_id bigint [ref: > game_room_players.id]
  position_index int
}
```

```
Table game_turns {
  id BIGSERIAL [pk]
  game_room_id bigint [not null, ref: > game_rooms.id]
  player_id bigint [not null, ref: > game_room_players.id]
  card_played_id bigint [ref: > uno_cards.id]
  action_type action_type_enum [not null]
  created_at timestamp [default: `CURRENT_TIMESTAMP`]
}
```

```
Table game_results {
  id BIGSERIAL [pk]
  game_room_id bigint [not null, ref: > game_rooms.id]
  winner_id bigint [ref: > users.id]
  total_turns int
  created_at timestamp [default: `CURRENT_TIMESTAMP`]
}
```

```
Table game_result_players {
  id BIGSERIAL [pk]
  game_result_id bigint [not null, ref: > game_results.id]
```

```
user_id bigint [not null, ref: > users.id]
rank int [not null]
cards_left int [default: 0]
}
```

```
Table chat_messages {
  id BIGSERIAL [pk]
  user_id bigint [not null, ref: > users.id]
  game_room_id bigint [ref: > game_rooms.id]
  message text [not null]
  created_at timestamp [default: `CURRENT_TIMESTAMP`]
}
```



# Specification Questions

## 1. Entities and Attributes

- users - Authentication and Profile
  - id BIGSERIAL PK
  - username VARCHAR
  - email VARCHAR
  - password VARCHAR
  - created\_at TIMESTAMP
- game\_rooms - Active UNO lobby/game
  - id BIGSERIAL PK
  - title VARCHAR
  - title VARCHAR
  - max\_players INT
  - password VARCHAR
  - status game\_status\_enum
  - created\_by BIGINT -> users FK
  - created\_at TIMESTAMP
  - started\_at TIMESTAMP
  - ended\_at TIMESTAMP
- game\_room\_players - Join table for users and rooms
  - id BIGSERIAL PK
  - user\_id BIGINT -> users FK
  - game\_room\_id BIGINT -> game\_rooms FK
  - is\_game\_master BOOLEAN
  - player\_order INT
  - cards\_in\_hand INT
  - joined\_at TIMESTAMP
- uno\_cards - master deck definition
  - id BIGSERIAL PK
  - color card\_color\_enum
  - value card\_value\_enum
- game\_room\_decks - per-room deck state
  - id BIGSERIAL PK
  - game\_room\_id BIGINT -> game\_rooms FK
  - card\_id BIGINT -> uno\_cards FK
  - location card\_location\_enum
  - owner\_player\_id BIGINT -> game\_room\_players FK
  - position\_index INT
- game\_turns - action log for each turn
  - id BIGSERIAL PK
  - game\_room\_id BIGINT -> game\_rooms FK
  - player\_id BIGINT -> game\_room\_players FK
  - card\_played\_id BIGINT NULL -> uno\_cards FK



- action\_type action\_type\_enum
- created\_at TIMESTAMP DEFAULT
- game\_results — final summary of a finished game
  - id BIGSERIAL PK
  - game\_room\_id BIGINT -> game\_rooms FK
  - winner\_id BIGINT NULL -> users FK
  - total\_turns INT
  - created\_at TIMESTAMP
- game\_result\_players — per-player final standing
  - id BIGSERIAL PK
  - game\_result\_id BIGINT -> game\_results FK
  - user\_id BIGINT -> users FK
  - rank INT
  - cards\_left INT
- chat\_messages — lobby-wide chat
  - id BIGSERIALPK
  - user\_id BIGINT -> users FK
  - message TEXT
  - created\_at TIMESTAMP

## 2. Relationships

- Users -> Game\_rooms : 1 - 0..1
  - Each room has one creator, a user can create 0 or 1 room
- Users -> game\_result\_players : 1 - \*
  - A user can show up in many game results
- Users -> game\_results : 1 - \*
  - A user can win multiple games
- Users -> chat\_messages: 1 - \*
  - A user can send many chat messages
- Game\_rooms -> game\_room\_players : 1 - \*
  - one room has many participants
- Game\_room -> Game\_Turns : 1 - \*
  - One room logs many turns
- Game\_rooms -> game\_results : 1 - 0..1
  - Each room may have 0 or 1 final result
- Game\_room\_players -> game\_turns : 1 - \*
  - A player can perform multiple turns
- Game\_room\_players -> Game\_room\_decks : 1 - \*
  - A player can own multiple cards in hand
- Game\_room\_players -> Users : 1 - \*
  - Many users can join one room
- Uno\_cards -> Game\_room\_decks : 1 - \*
  - A single card type can exist in many room decks
- Uno\_Cards -> Game\_turns : 1 - 0..\*

- A card may appear in multiple turns
- Game\_Results -> Game\_Result\_Players : 1 - \*
  - One game result has multiple player standings

### **Normalization and Design Rationale**

The database is designed to separate users, rooms, players, cards, turns, and results into distinct entities and keep clarity. The design supports multiplayer UNO with clear ownership, turn tracking, and end results.