Database Schema

Table 1: users

```
CREATE TABLE users (
   user_id SERIAL PRIMARY KEY,
   first_name VARCHAR(100),
   last_name VARCHAR(100),
   email VARCHAR(150) UNIQUE NOT NULL,
   phone VARCHAR(15),
   password VARCHAR(255) NOT NULL,
   role VARCHAR(50) CHECK (role IN ('member', 'admin')) NOT NULL,
   membership_id INT REFERENCES memberships(membership_id),
   created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Table 2: memberships

```
CREATE TABLE memberships (
   membership_id SERIAL PRIMARY KEY,
   membership_type VARCHAR(100),
   start_date DATE,
   end_date DATE,
   price DECIMAL(10, 2),
   status VARCHAR(50) CHECK (status IN ('active', 'expired', 'pending')),
   created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Table 3: classes

```
CREATE TABLE classes (
   class_id SERIAL PRIMARY KEY,
   class_name VARCHAR(100),
   instructor VARCHAR(100),
   start_time TIMESTAMP,
   end_time TIMESTAMP,
   max_capacity INT,
   created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Table 4: bookings

```
CREATE TABLE bookings (
   booking_id SERIAL PRIMARY KEY,
   user_id INT REFERENCES users(user_id),
   class_id INT REFERENCES classes(class_id),
   booking_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
   attendance_status VARCHAR(50) CHECK (attendance_status IN ('attended', 'missed', 'canceled')) DEFAULT 'booked'
);
```

Table 5: payments

```
CREATE TABLE payments (
   payment_id SERIAL PRIMARY KEY,
   user_id INT REFERENCES users(user_id),
   amount DECIMAL(10, 2),
   payment_method VARCHAR(50) CHECK (payment_method IN ('PayPal', 'Credit Card', 'Debit Card')),
   payment_status VARCHAR(50) CHECK (payment_status IN ('completed', 'failed', 'pending')),
   payment_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Table 6: notifications

```
CREATE TABLE notifications (
    notification_id SERIAL PRIMARY KEY,
    user_id INT REFERENCES users(user_id),
    message TEXT,
    notification_type VARCHAR(50) CHECK (notification_type IN ('class_reminder', 'membership_update')),
    sent_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

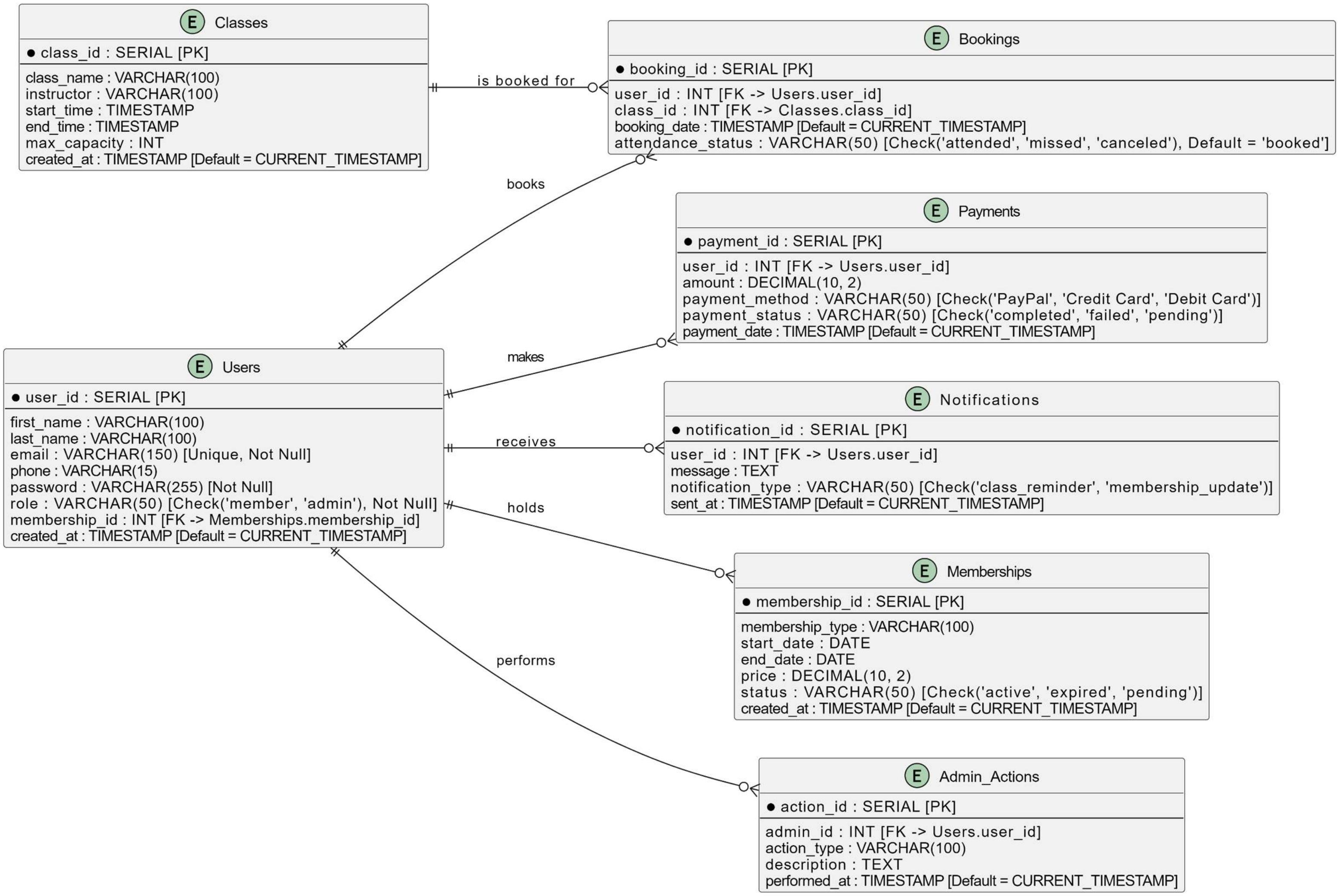
Table 7: admin_actions

```
CREATE TABLE admin_actions (
    action_id SERIAL PRIMARY KEY,
    admin_id INT REFERENCES users(user_id),
    action_type VARCHAR(100),
    description TEXT,
    performed_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Relationships:

- Users to Memberships: A user can hold one membership at a time.
- Users to Bookings: One user can book multiple classes, and a class can also be booked multiple times.
- Payments: Records payments against user membership.
- Notifications: These are in the form of notifications for reminders and updates.

ER Diagram:



E Classes

● class_id : SERIAL [PK]

class_name : VARCHAR(100)
instructor : VARCHAR(100)
start_time : TIMESTAMP
end_time : TIMESTAMP
max_capacity : INT
created_at : TIMESTAMP [Default = CURRENT_TIMESTAMP]

E Bookings

● booking_id : SERIAL [PK]

user_id : INT [FK -> Users.user_id]
class_id : INT [FK -> Classes.class_id]
booking_date : TIMESTAMP [Default = CURRENT_TIMESTAMP]
attendance_status : VARCHAR(50) [Check('attended', 'missed', 'canceled'), Default = 'booked']

E Payments

● payment_id : SERIAL [PK]

user_id : INT [FK -> Users.user_id]
amount : DECIMAL(10, 2)
payment_method : VARCHAR(50) [Check('PayPal', 'Credit Card', 'Debit Card')]
payment_status : VARCHAR(50) [Check('completed', 'failed', 'pending')]
payment_date : TIMESTAMP [Default = CURRENT_TIMESTAMP]

E Users

● user_id : SERIAL [PK]

first_name : VARCHAR(100)
last_name : VARCHAR(100)
email : VARCHAR(150) [Unique, Not Null]
phone : VARCHAR(15)
password : VARCHAR(255) [Not Null]
role : VARCHAR(50) [Check('member', 'admin'), Not Null]
membership_id : INT [FK -> Memberships.membership_id]
created_at : TIMESTAMP [Default = CURRENT_TIMESTAMP]

E Notifications

● notification_id : SERIAL [PK]

user_id : INT [FK -> Users.user_id]
message : TEXT
notification_type : VARCHAR(50) [Check('class_reminder', 'membership_update')]
sent_at : TIMESTAMP [Default = CURRENT_TIMESTAMP]

E Memberships

● membership_id : SERIAL [PK]

membership_type : VARCHAR(100)
start_date : DATE
end_date : DATE
price : DECIMAL(10, 2)
status : VARCHAR(50) [Check('active', 'expired', 'pending')]
created_at : TIMESTAMP [Default = CURRENT_TIMESTAMP]

E Admin_Actions

● action_id : SERIAL [PK]

admin_id : INT [FK -> Users.user_id]
action_type : VARCHAR(100)
description : TEXT
performed_at : TIMESTAMP [Default = CURRENT_TIMESTAMP]