**SNOOGLE**

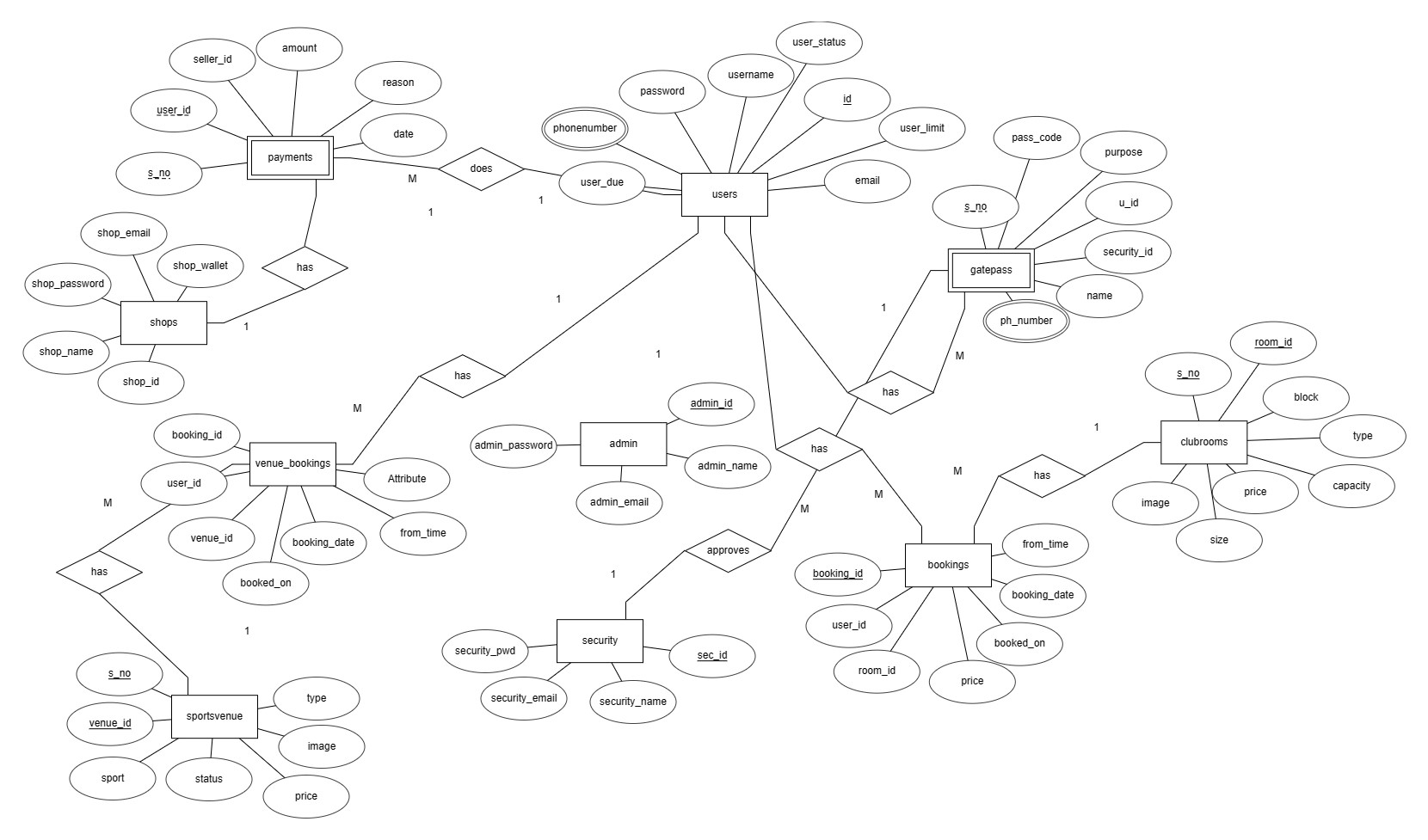
**Snehith Madala - 2010110371**

**Sai Sri Maddirala-2010110761**

SNoogle is an all-in-one platform to centralize all the finances spread throughout the campus and also aims at taking away hassle in organizing and managing common facilities in the campus.

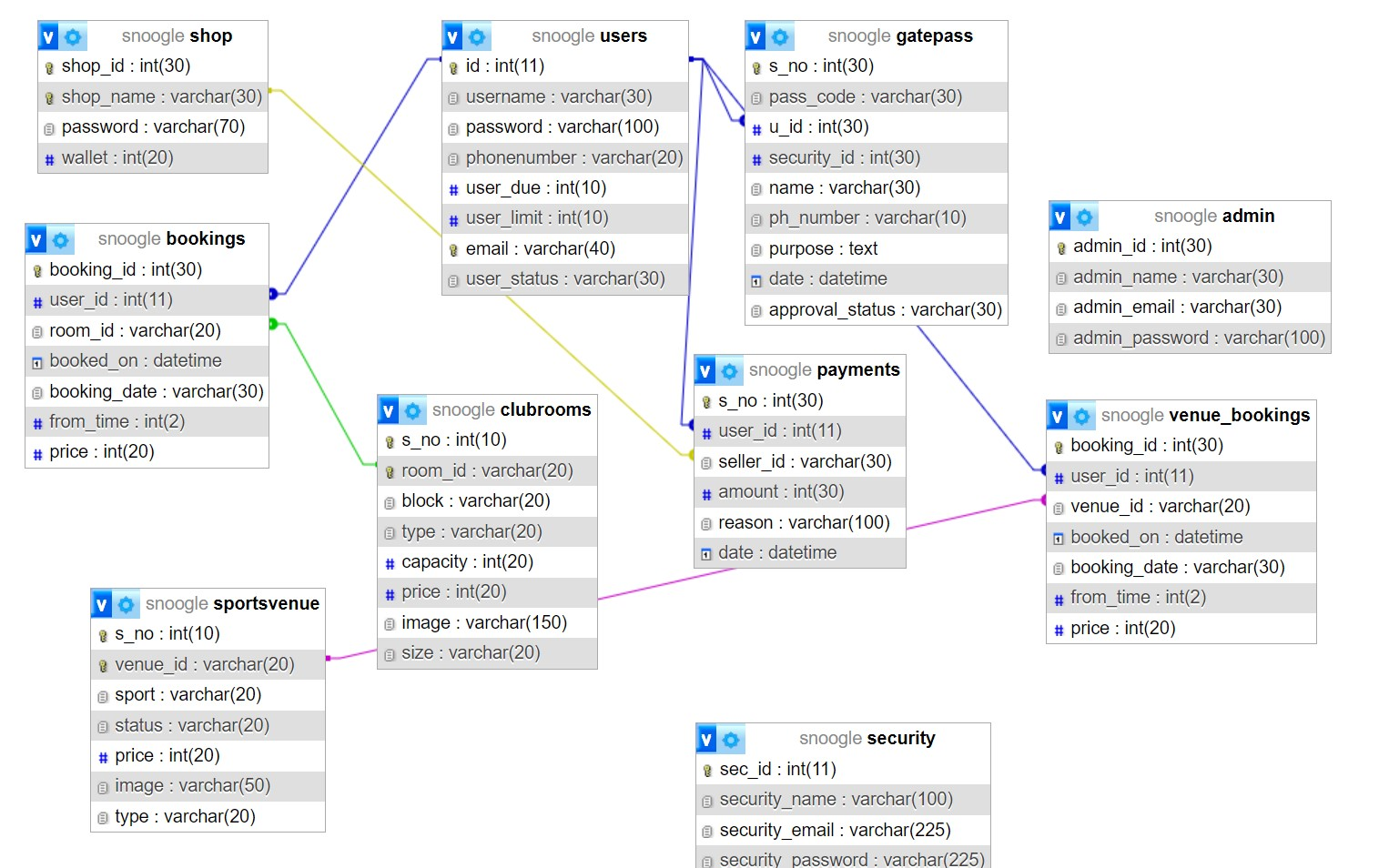
The Snoogle database aims to efficiently manage various aspects of a dynamic environment, including room bookings, security, payments, and sports venue reservations. The features include user management, room bookings, gate pass tracking, payment transactions, and sports venue availability.

**ER MODEL**

****

The Entity-Relationship (ER) model illustrates the relationships between different entities in the database. Entities such as users, bookings, payments, and more are connected through well-defined relationships, providing a clear representation of the data structure.

**RELATIONAL MODEL**



The relational model is derived from the ER model and represents how data is organized in tables. Each table corresponds to an entity, and relationships are maintained through keys. For instance, the bookings table is linked to the users and clubrooms tables.

**Admin Table (admin)**: This table stores information about administrators. It has a primary key admin\_id, and administrators are identified by their names and email addresses. This table doesn't have direct relationships with other tables in the provided SQL dump.

**Bookings Table (bookings):** This table manages booking information. It has relationships with the users and clubrooms tables. The user\_id column is a foreign key referencing the users table, and the room\_id column references the clubrooms table. This indicates which user made a booking and the room that was booked.

**Clubrooms Table (clubrooms):** This table contains details about clubrooms, including room IDs, block information, capacity, price, etc. The primary key is room\_id, and it is referenced by the bookings table.

**Gatepass Table (gatepass):** This table manages gatepass information, including pass codes, user IDs, security IDs, and approval status. It has a foreign key relationship with the users table.

**Payments Table (payments):** This table records payment transactions. It has relationships with the users and shops tables through foreign keys.

**Security Table (security):** Stores information about security personnel. It has a primary key sec\_id.

**Shops Table (shops):** Contains details about shops, including shop IDs, names, passwords, emails, and wallet amounts. The primary key is shop\_name.

**Sportsvenue Table (sportsvenue):** Manages information about sports venues. It has a primary key venue\_id and is referenced by the venue\_bookings table.

**Users Table (users):** Stores user information, including user IDs, usernames, passwords, contact details, due amounts, limits, and status. The primary key is id.

**Venue Bookings Table (venue\_bookings):** Manages reservations for sports venues. It has relationships with the users and sportsvenue tables.

**Normalized Relations:**

The database adheres to normalization principles, minimizing data redundancy and ensuring data integrity. Normalized relations contribute to efficient data management and reduce the risk of anomalies during data operations.

**Snoogle is in 3NF form because:**

**Elimination of Duplicate Data:** Tables such as admin, clubrooms, gatepass, payments, security, shops, sportsvenue, users, and venue\_bookings seem to avoid redundant information. Each table appears to store data that is unique to its purpose.

**Functional Dependencies:** Columns in the tables appear to be functionally dependent on the primary key. For example, in the users table, the columns id, username, password, etc., are functionally dependent on the primary key id.

**Separation of Concerns:** Different tables handle different aspects of data. For instance, clubrooms manages information about available rooms, payments handles financial transactions, and venue\_bookings manages reservations for sports venues.

**SQL COMMANDS:**

**Database and Table Creation:**

**CREATE DATABASE:**Creates the snoogle database.

**CREATE TABLE:** Creates tables such as admin, bookings, clubrooms, gatepass, payments, security, shops, sportsvenue, users, and venue\_bookings.

**Data Insertion:**

**INSERT INTO**: Inserts data into tables like admin, bookings, clubrooms, gatepass, payments, security, shops, sportsvenue, users, and venue\_bookings.

Stored Procedures Creation:

**CREATE PROCEDURE:** Defines stored procedures like insert\_record, insert\_gatepass, and insert\_payments.

**Foreign Key Constraints**:

**FOREIGN KEY:** Establishes foreign key relationships between tables, such as between bookings and users, bookings and clubrooms, gatepass and users, payments and users, payments and shops, and venue\_bookings and users.

Indexes:

**ADD PRIMARY KEY:** Adds primary key constraints to columns, ensuring uniqueness.

**ADD KEY:** Adds indexes to columns for optimization, for example, in the bookings table.

**Auto-Increment:**

**AUTO\_INCREMENT:** Configures columns to auto-increment for primary key generation.

**Delimiter and Stored Procedure Call:**

**DELIMITER:** Sets the delimiter for defining stored procedures.

**CALL**: Invokes stored procedures, such as insert\_gatepass and insert\_payments.

**PROCEDURES**

procedures are used to encapsulate and execute predefined logic. Three procedures, namely **insert\_record**, **insert\_gatepass**, and **insert\_payments**, are defined. These procedures serve specific purposes, such as inserting records into tables like bookings, gatepass, and payments. Procedures enhance modularity, readability, and reusability in database operations by encapsulating SQL statements and business logic.

**REFERENCES**

* <https://www.w3schools.com/html/>
* <https://www.w3schools.com/php/default.asp>
* <https://stackoverflow.com/questions/tagged/php>
* <https://getbootstrap.com/docs/4.6/getting-started/introduction/>
* <https://developer.mozilla.org/en-US/docs/Web/JavaScript>