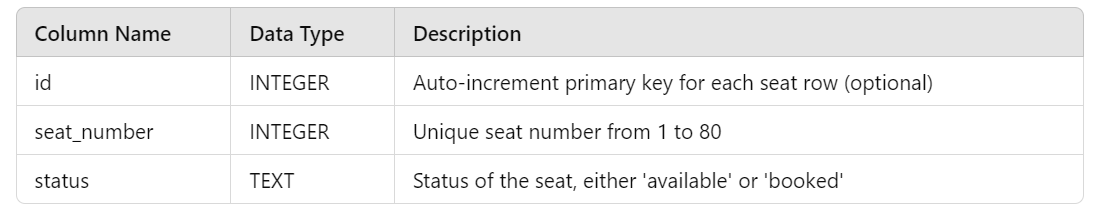
The database for the train seat booking system is relatively simple, consisting of one table: seats. This table holds information about each seat's number and its booking status. Below is the database structure for the SQLite database used in the code.

**Database Structure:**

**Table: seats**

This table will have 80 rows, one for each seat in the coach. The key columns are seat\_number (the number of the seat) and status (whether the seat is available or booked).



SQL Query to Create the seats Table:

CREATE TABLE IF NOT EXISTS seats

(

id INTEGER PRIMARY KEY AUTOINCREMENT,

seat\_number INTEGER NOT NULL,

status TEXT NOT NULL CHECK(status IN ('available', 'booked'))

);

* **id**: A unique identifier for each seat, with auto-increment enabled. This is mainly for internal use and could be optional.
* **seat\_number**: The actual seat number, ranging from 1 to 80. The first 77 seats represent 7-seat rows, while seat numbers 78 to 80 represent the last row of 3 seats.
* **status**: The seat's current booking status, which can either be 'available' or 'booked'.

**Database Initialization:**

The initialize\_seats() function in the code ensures that the table is only created if it doesn't already exist. It populates the table with 80 seats, all marked as 'available' initially.

**Example Record:**

In this scenario:

* **Seat numbers 1 to 10** are booked.
* **Seat numbers 11 to 80** are available.

This structure allows the system to easily query for available seats, and users can book based on the availability reflected in the status field.

**Code Structure**

1. **Database Initialization**:
   * The function initialize\_seats() creates the seats table if it doesn't exist and populates it with 80 entries, marking all seats as available.
2. **Finding Available Seats**:
   * The function find\_available\_seats(num\_seats) checks for available seats. It first attempts to find seats in the same row and, if not enough are found, looks for adjacent available seats.
3. **Booking Seats**:
   * The function book\_seats(seat\_numbers) updates the status of the selected seats in the database from available to booked.
4. **Displaying Seat Status**:
   * The function display\_seats() shows the current status of all seats in the console, using emojis to indicate whether a seat is available (🟢) or booked (🔴).
5. **Checking Available Seats**:
   * The function available\_seats\_count() returns the number of available seats left in the coach.
6. **Main Booking Process**:
   * The function book\_tickets() manages the entire booking process:
     + It initializes the seats.
     + It enters a loop that allows users to book seats multiple times.
     + Users can input the number of seats they want to book (up to 7).
     + If enough seats are available, they are booked, and the current seat status is displayed.
     + The user is asked if they want to continue booking until the coach is full or they choose to stop.