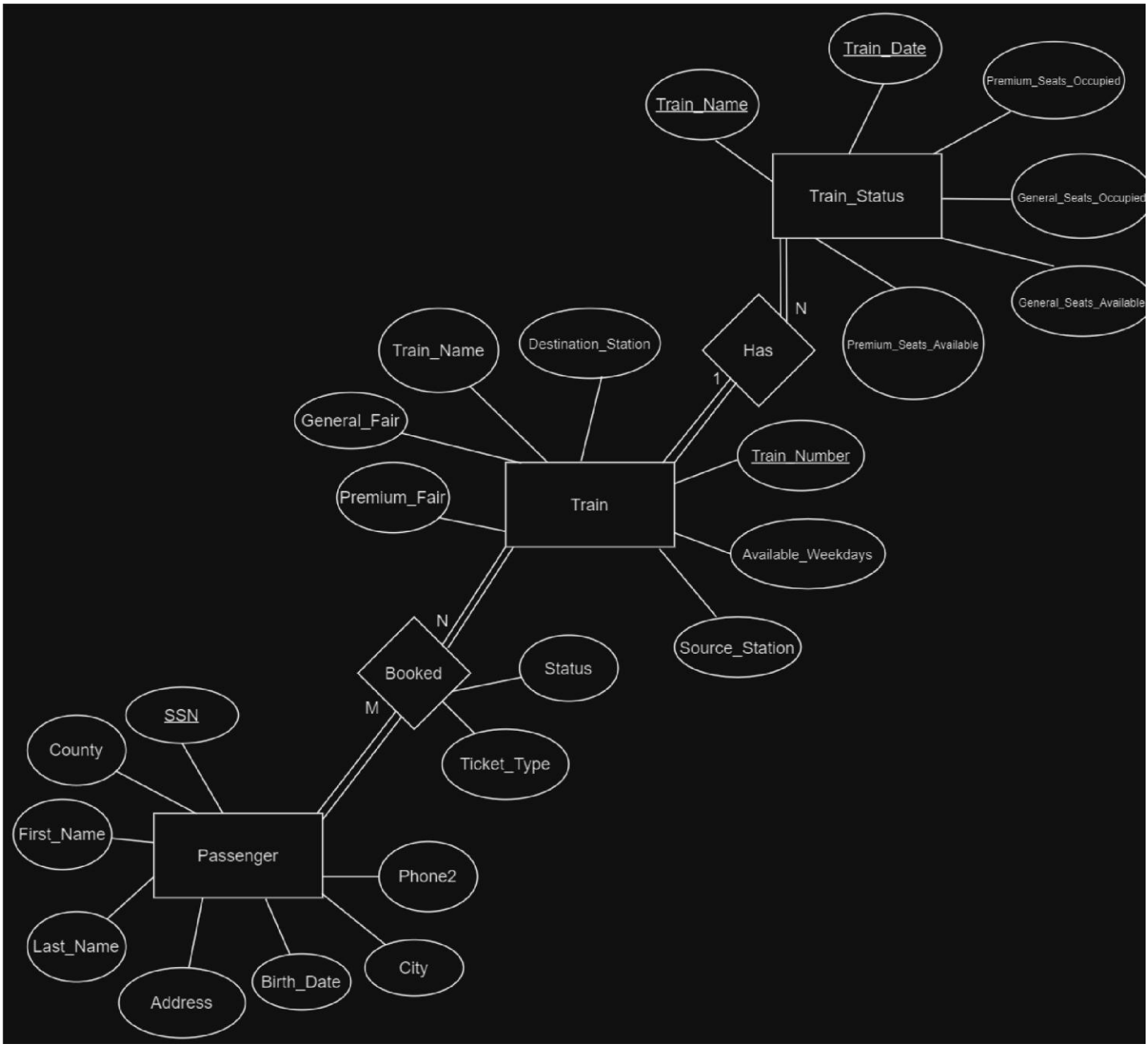
Project 1 - CSE 3330-004 – Project #1 – Railway Reservation System

Dev Patel

ER DIAGRAM


Train_Date
Premium_Seats_Occupied
Train_Name
Train_Status
General_Seats_Occupied
General_Seats_Available
N
Premium_Seats_Available
Has
1
Train_Name
Destination_Station
General_Fair
Train_Number
Premium_Fair
Train
Available_Weekdays
N
Source_Station
Booked
Status
M
SSN
County
Ticket_Type
First_Name
Passenger
Phone2
Last_Name
City
Address
Birth_Date

LOAD DATA METHOD

To load the data into the tables, we used the INSERT INTO SQL command. This method explicitly adds rows of data into the specified tables by defining the columns and their corresponding values. Each INSERT INTO statement consists of the table name followed by a list of column names and a VALUES clause that includes the data for each column in the order they are defined.

For example, for our project we have 4 INSERT INTO SQL commands, one for each table, which are

1) INSERT INTO Passenger (First_Name, Last_Name, Address, City, County, Phone2, SSN, Birth_Date) VALUES
2) INSERT INTO Booked (Passenger_SSN, Train_Number, Ticket_Type, Status) VALUES
3) INSERT INTO Train (Train_Number, Train_Name, Premium_Fair, General_Fair, Source_Station, Destination_Station, Available_Weekdays) VALUES
4) INSERT INTO Train_status (Train_Date, Train_Name, Premium_Seats_Available, General_Seats_Available, Premium_Seats_Occupied, General_Seats_Occupied) VALUES

For each table, separate INSERT INTO statements were constructed to load the respective data, ensuring that each row of data matches the schema of the table (taken from the provided zip file containing 4 excel files on Canvas submission module for Project 1). This approach allows for precise insertion of multiple records all at once into the database tables.

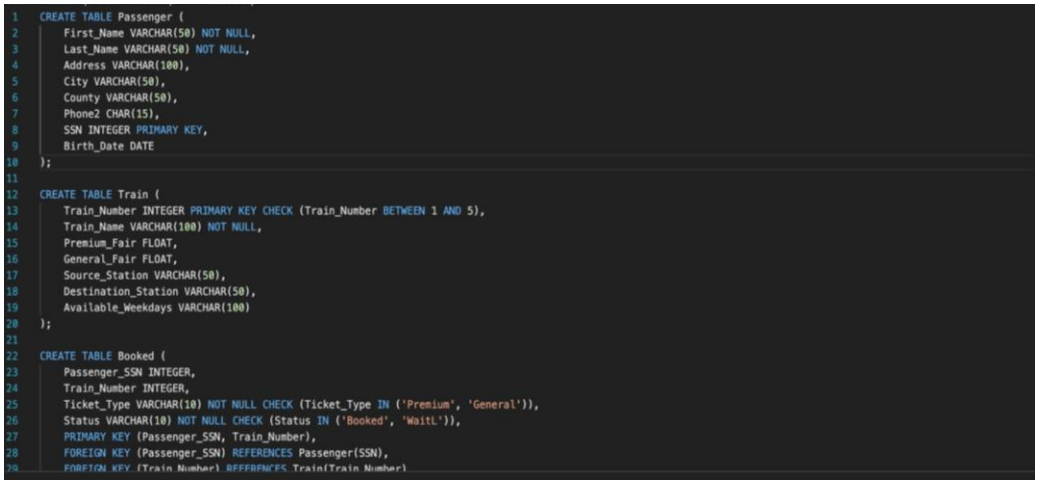
ReadMe (TOOLS USED FOR PROJECT)

- Visual Studio Code version 1.93.1 (IDE to run program)
- SQLiteStudio version 3.4.4 (to access database)
- SQLite version 0.14.1 (SQL extension from VSCode)
- draw.io (to create ER Diagram)

SQL SELECT STATEMENTS - QUERY RESULT SCREENSHOTS

1) **Given a passenger's last name and first name and retrieve all trains they are booked on.**

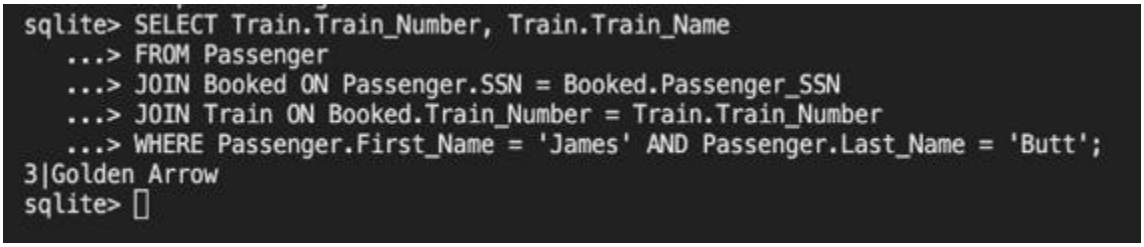
   - used James Butt as an example to run query (specified custom input for name is correct/allowed)



```sql
CREATE TABLE Passenger (
    First_Name VARCHAR(50) NOT NULL,
    Last_Name VARCHAR(50) NOT NULL,
    Address VARCHAR(100),
    City VARCHAR(50),
    County VARCHAR(50),
    Phone2 CHAR(15),
    SSN INTEGER PRIMARY KEY,
    Birth_Date DATE
);

CREATE TABLE Train (
    Train_Number INTEGER PRIMARY KEY CHECK (Train_Number BETWEEN 1 AND 5),
    Train_Name VARCHAR(100) NOT NULL,
    Premium_Fair FLOAT,
    General_Fair FLOAT,
    Source_Station VARCHAR(50),
    Destination_Station VARCHAR(50),
    Available_Weekdays VARCHAR(100)
);

CREATE TABLE Booked (
    Passenger_SSN INTEGER,
    Train_Number INTEGER,
    Ticket_Type VARCHAR(10) NOT NULL CHECK (Ticket_Type IN ('Premium', 'General')),
    Status VARCHAR(10) NOT NULL CHECK (Status IN ('Booked', 'WaitL')),
    PRIMARY KEY (Passenger_SSN, Train_Number),
    FOREIGN KEY (Passenger_SSN) REFERENCES Passenger(SSN),
    FOREIGN KEY (Train_Number) REFERENCES Train(Train_Number)
```



```
sqlite> SELECT Train.Train_Number, Train.Train_Name
   ...> FROM Passenger
   ...> JOIN Booked ON Passenger.SSN = Booked.Passenger_SSN
   ...> JOIN Train ON Booked.Train_Number = Train.Train_Number
   ...> WHERE Passenger.First_Name = 'James' AND Passenger.Last_Name = 'Butt';
3|Golden Arrow
sqlite>
```

2)

Given a day list the passengers traveling on that day with confirmed tickets. - *used Friday as example to run query (specified custom input for name is correct/allowed - TA Priyanka)*

```sql
CREATE TABLE Passenger (
    First_Name VARCHAR(50) NOT NULL,
    Last_Name VARCHAR(50) NOT NULL,
    Address VARCHAR(100),
    City VARCHAR(50),
    County VARCHAR(50),
    Phone2 CHAR(15),
    SSN INTEGER PRIMARY KEY,
    Birth_Date DATE
);

CREATE TABLE Train (
    Train_Number INTEGER PRIMARY KEY CHECK (Train_Number BETWEEN 1 AND 5),
    Train_Name VARCHAR(100) NOT NULL,
    Premium_Fair FLOAT,
    General_Fair FLOAT,
    Source_Station VARCHAR(50),
    Destination_Station VARCHAR(50),
    Available_Weekdays VARCHAR(100)
);

CREATE TABLE Booked (
    Passenger_SSN INTEGER,
    Train_Number INTEGER,
    Ticket_Type VARCHAR(10) NOT NULL CHECK (Ticket_Type IN ('Premium', 'General')),
    Status VARCHAR(10) NOT NULL CHECK (Status IN ('Booked', 'WaitL')),
    PRIMARY KEY (Passenger_SSN, Train_Number),
    FOREIGN KEY (Passenger_SSN) REFERENCES Passenger(SSN),
    FOREIGN KEY (Train_Number) REFERENCES Train(Train_Number)
```

```
sqlite> SELECT Passenger.First_Name, Passenger.Last_Name
   ...> FROM Passenger
   ...> JOIN Booked ON Passenger.SSN = Booked.Passenger_SSN
   ...> JOIN Train ON Booked.Train_Number = Train.Train_Number
   ...> WHERE Booked.Status = 'Booked' AND Train.Available_Weekdays LIKE '%Friday%';
Kiley|Caldarera
Fletcher|Flosi
Josephine|Darakjy
Sage|Wieser
Kris|Marrier
Graciela|Ruta
sqlite>
```

3)

Display the train information (Train Number, Train Name, Source and Destination) and passenger information (Name, Address, Category, ticket status) of passengers who are between the ages of 50 to 60.

```sql
CREATE TABLE Passenger (
    First_Name VARCHAR(50) NOT NULL,
    Last_Name VARCHAR(50) NOT NULL,
    Address VARCHAR(100),
    City VARCHAR(50),
    County VARCHAR(50),
    Phone2 CHAR(15),
    SSN INTEGER PRIMARY KEY,
    Birth_Date DATE
);

CREATE TABLE Train (
    Train_Number INTEGER PRIMARY KEY CHECK (Train_Number BETWEEN 1 AND 5),
    Train_Name VARCHAR(100) NOT NULL,
    Premium_Fair FLOAT,
    General_Fair FLOAT,
    Source_Station VARCHAR(50),
    Destination_Station VARCHAR(50),
    Available_Weekdays VARCHAR(100)
);

CREATE TABLE Booked (
    Passenger_SSN INTEGER,
    Train_Number INTEGER,
    Ticket_Type VARCHAR(10) NOT NULL CHECK (Ticket_Type IN ('Premium', 'General')),
    Status VARCHAR(10) NOT NULL CHECK (Status IN ('Booked', 'WaitL')),
    PRIMARY KEY (Passenger_SSN, Train_Number),
    FOREIGN KEY (Passenger_SSN) REFERENCES Passenger(SSN),
    FOREIGN KEY (Train_Number) REFERENCES Train(Train_Number)
```

```
sqlite> SELECT Train.Train_Number, Train.Train_Name, Train.Source_Station, Train.Destination_Statiosqlite> SELECT Train.Train_Number, Train.Trai
n_Name, Train.Soun, Passenger.First_Name, Passenger  on, Passenger.First_Name, Passenger.Last_Name, Passenger.Address, Booked.Ticket_Type, Booke
d.Status
   ...> From Passenger
   ...> JOIN Booked ON Passenger.SSN = Booked.Passenger_SSN
   ...> JOIN Train ON Booked.Train_Number = Train.Train_Number
   ...> WHERE strftime('%Y', 'now') - strftime('%Y', Passenger.Birth_Date) BETWEEN 50 AND 60;
3|Golden Arrow|Victoria|Dover|James|Butt|6649 N Blue Gum St|Premium|Booked
4|Golden Chariot|Bangalore|Goa|Michael|Smith|5678 Oak Avenue|General|Booked
sqlite>
```

4)

List train name, day and number of passengers on that train. - *listed all 4 train names with day and number of passengers*

```sql
CREATE TABLE Passenger (
    First_Name VARCHAR(50) NOT NULL,
    Last_Name VARCHAR(50) NOT NULL,
    Address VARCHAR(100),
    City VARCHAR(50),
    County VARCHAR(50),
    Phone2 CHAR(15),
    SSN INTEGER PRIMARY KEY,
    Birth_Date DATE
);

CREATE TABLE Train (
    Train_Number INTEGER PRIMARY KEY CHECK (Train_Number BETWEEN 1 AND 5),
    Train_Name VARCHAR(100) NOT NULL,
    Premium_Fair FLOAT,
    General_Fair FLOAT,
    Source_Station VARCHAR(50),
    Destination_Station VARCHAR(50),
    Available_Weekdays VARCHAR(100)
);

CREATE TABLE Booked (
    Passenger_SSN INTEGER,
    Train_Number INTEGER,
    Ticket_Type VARCHAR(10) NOT NULL CHECK (Ticket_Type IN ('Premium', 'General')),
    Status VARCHAR(10) NOT NULL CHECK (Status IN ('Booked', 'WaitL')),
    PRIMARY KEY (Passenger_SSN, Train_Number),
```

```
sqlite> SELECT Train.Train_Name, Train_Status.Train_Date, Train_Status.Premium_Seats_Occupied + Train_Status.General_Seats_Occupied
   ...> FROM Train
   ...> JOIN Train_Status ON Train.Train_Name = Train_Status.Train_Name;
Orient Express|Friday|0
Flying Scottsman|Sunday|6
Golden Arrow|Tuesday|7
Golden Chariot|Saturday|6
sqlite>
```

5)

Enter a train name and retrieve all the passengers with confirmed status traveling on that train. - *used Flying Scottsman as an example for the train (specified custom input for name is correct/allowed)*

```
CREATE TABLE Passenger (
    First_Name VARCHAR(50) NOT NULL,
    Last_Name VARCHAR(50) NOT NULL,
    Address VARCHAR(100),
    City VARCHAR(50),
    County VARCHAR(50),
    Phone2 CHAR(15),
    SSN INTEGER PRIMARY KEY,
    Birth_Date DATE
);

CREATE TABLE Train (
    Train_Number INTEGER PRIMARY KEY CHECK (Train_Number BETWEEN 1 AND 5),
    Train_Name VARCHAR(100) NOT NULL,
    Premium_Fair FLOAT,
    General_Fair FLOAT,
    Source_Station VARCHAR(50),
    Destination_Station VARCHAR(50),
    Available_Weekdays VARCHAR(100)
);

CREATE TABLE Booked (
    Passenger_SSN INTEGER,
    Train_Number INTEGER,
    Ticket_Type VARCHAR(10) NOT NULL CHECK (Ticket_Type IN ('Premium', 'General')),
    Status VARCHAR(10) NOT NULL CHECK (Status IN ('Booked', 'WaitL')),
    PRIMARY KEY (Passenger_SSN, Train_Number),
    FOREIGN KEY (Passenger_SSN) REFERENCES Passenger(SSN),
    FOREIGN KEY (Train_Number) REFERENCES Train(Train_Number)
);
```

```
sqlite> SELECT Passenger.First_Name, Passenger.Last_Name
   ...> FROM Passenger
   ...> JOIN Booked ON Passenger.SSN = Booked.Passenger_SSN
   ...> JOIN Train ON Booked.Train_Number = Train.Train_Number
   ...> WHERE Train.Train_Name = 'Flying Scottsman' AND Booked.Status = 'Booked';
Kiley|Caldarera
Fletcher|Flosi
Josephine|Darakjy
Sage|Wieser
Kris|Marrier
Graciela|Ruta
sqlite>
```

6)

List passengers that are waitlisted including the name of the train.

```sql
CREATE TABLE Passenger (
    First_Name VARCHAR(50) NOT NULL,
    Last_Name VARCHAR(50) NOT NULL,
    Address VARCHAR(100),
    City VARCHAR(50),
    County VARCHAR(50),
    Phone2 CHAR(15),
    SSN INTEGER PRIMARY KEY,
    Birth_Date DATE
);

CREATE TABLE Train (
    Train_Number INTEGER PRIMARY KEY CHECK (Train_Number BETWEEN 1 AND 5),
    Train_Name VARCHAR(100) NOT NULL,
    Premium_Fair FLOAT,
    General_Fair FLOAT,
    Source_Station VARCHAR(50),
    Destination_Station VARCHAR(50),
    Available_Weekdays VARCHAR(100)
);

CREATE TABLE Booked (
    Passenger_SSN INTEGER,
    Train_Number INTEGER,
    Ticket_Type VARCHAR(10) NOT NULL CHECK (Ticket_Type IN ('Premium', 'General')),
    Status VARCHAR(10) NOT NULL CHECK (Status IN ('Booked', 'WaitL')),
    PRIMARY KEY (Passenger_SSN, Train_Number),
    FOREIGN KEY (Passenger_SSN) REFERENCES Passenger(SSN),
    FOREIGN KEY (Train_Number) REFERENCES Train(Train_Number)
);
```

```
sqlite> SELECT Passenger.First_Name, Passenger.Last_Name, Train.Train_Name
   ...> FROM Passenger
   ...> JOIN Booked ON Passenger.SSN = Booked.Passenger_SSN
   ...> JOIN Train ON Booked.Train_Number = Train.Train_Number
   ...> WHERE Booked.Status = 'WaitL';
Abel|Maclead|Flying Scottsman
Mattie|Poquette|Flying Scottsman
Simona|Morasca|Golden Arrow
Sarah|Johnson|Golden Chariot
David|Willson|Golden Chariot
Jennifer|David|Golden Chariot
Mathew|Alison|Golden Chariot
sqlite>
```

7)

List passenger names in descending order that have '605' phone area code. *assorted passengers by first name (first name is allowed specified) in descending order*

```sql
CREATE TABLE Passenger (
    First_Name VARCHAR(50) NOT NULL,
    Last_Name VARCHAR(50) NOT NULL,
    Address VARCHAR(100),
    City VARCHAR(50),
    County VARCHAR(50),
    Phone2 CHAR(15),
    SSN INTEGER PRIMARY KEY,
    Birth_Date DATE
);

CREATE TABLE Train (
    Train_Number INTEGER PRIMARY KEY CHECK (Train_Number BETWEEN 1 AND 5),
    Train_Name VARCHAR(100) NOT NULL,
    Premium_Fair FLOAT,
    General_Fair FLOAT,
    Source_Station VARCHAR(50),
    Destination_Station VARCHAR(50),
    Available_Weekdays VARCHAR(100)
);

CREATE TABLE Booked (
    Passenger_SSN INTEGER,
    Train_Number INTEGER,
    Ticket_Type VARCHAR(10) NOT NULL CHECK (Ticket_Type IN ('Premium', 'General')),
    Status VARCHAR(10) NOT NULL CHECK (Status IN ('Booked', 'WaitL')),
    PRIMARY KEY (Passenger_SSN, Train_Number),
    FOREIGN KEY (Passenger_SSN) REFERENCES Passenger(SSN),
    FOREIGN KEY (Train_Number) REFERENCES Train(Train_Number)
);

CREATE TABLE Train_Status (
    Train_Date DATE,
    Train_Name VARCHAR(100),
```

```
sqlite> SELECT First_Name, Last_Name
   ...> FROM Passenger
   ...> WHERE Phone2 LIKE '605%'
   ...> ORDER BY First_Name DESC, Last_Name DESC;
Sage|Wieser
Mattie|Poquette
Art|Venere
sqlite>
```

8) List name of passengers that are traveling on Thursdays in ascending order. *assorted passengers by first name in ascending order. Used Thursday as input (specified by question) and Booked as input - since those are the only passengers that confirmed traveling, not waitlisted. Results are empty which is correct due to in table Booked, none of the passengers have a Train Number of 1 (Train Number 1 is only the train that has Thursday as its available weekday/s)). Assorting passengers by first name, and adding Booked as input is correct/allowed)*

```
CREATE TABLE Passenger (
    County VARCHAR(50),
    Phone2 CHAR(15),
    SSN INTEGER PRIMARY KEY,
    Birth_Date DATE
);

CREATE TABLE Train (
    Train_Number INTEGER PRIMARY KEY CHECK (Train_Number BETWEEN 1 AND 5),
    Train_Name VARCHAR(100) NOT NULL,
    Premium_Fair FLOAT,
    General_Fair FLOAT,
    Source_Station VARCHAR(50),
    Destination_Station VARCHAR(50),
    Available_Weekdays VARCHAR(100)
);

CREATE TABLE Booked (
    Passenger_SSN INTEGER,
    Train_Number INTEGER,
    Ticket_Type VARCHAR(10) NOT NULL CHECK (Ticket_Type IN ('Premium', 'General')),
    Status VARCHAR(10) NOT NULL CHECK (Status IN ('Booked', 'WaitL')),
    PRIMARY KEY (Passenger_SSN, Train_Number),
    FOREIGN KEY (Passenger_SSN) REFERENCES Passenger(SSN),
    FOREIGN KEY (Train_Number) REFERENCES Train(Train_Number)
);
```

```
SQLite version 3.32.3 2020-06-18 14:16:19
Enter ".help" for usage hints.
sqlite> SELECT Passenger.First_Name, Passenger.Last_Name
   ...> FROM Passenger
   ...> JOIN Booked ON Passenger.SSN = Booked.Passenger_SSN
   ...> JOIN Train ON Booked.Train_Number = Train.Train_Number
   ...> WHERE Train.Available_Weekdays LIKE '%Thursday%' AND Booked.Status = 'Booked'
   ...> ORDER BY Passenger.First_Name ASC;
sqlite>
```