User
(user_id, first_name, last_name, email, phone_no, password)
user_id -> determines all of them

Airports
(IATA_code, airport_name, city_name)
IATA Pkey (3 letter code uniquely identifies airport) Like HYD, BLR, DXB

Flight_data
(Flight_num, airline_name, aircraft_type, src_airport, dst_airport)
Flight_num -> airline_name, aircraft_type src_airport, dst_airport (pkey) like AI2000 EK356 etc
Src_airport, dst_airport are foreign key to Airports

Schedule
(Schedule_id, Flight_num, src_airport, dst_airport, dept_date, arrival_date, dept_time, arrival_time, base_price, seats, status)
(src_airport, dst_airport fkey on Airports
Flight_num is (fkey on Flight data)
Flight_num, Schedule_id is (pkey) everything

Use a stored procedure Find_flight(src, dst, dept_date, max_layover)
Returns schedule_ids, use those schedule ids to compute data and send as json to frontend

Seats
(Schedule_id, seat_no, Flight_num, , status (avl/booked))
Since a flight has many seats the the Json object has (seats, price, status)
Like {1A , :200, Booked, 1B : 100 Available }
Since its hard to model seats for all planes we store as Json object
(Schedule_id, Flight_id) -> Json Object of seat

Single person can book flight for multiple ppl hence select multiple seats
After u select seats and confirm, u go to pay u generate trip data, reduce number of seats in Schedule table for corresponding flights

Traveler_info (all travellers irrespective of registration status)
(social_security_num, name, phone, email)

Trip
(SSN, PNR, Schedule_id, booked_by, seat_num, date_of_booking, status)
Status (confired, pending, cancelled)
In case of multicity each person is identified these 3 pkey as a single person under same pnr can have different flights (multicity)
Booked_by fkey on User_info

(you go to pay and it generates a transaction)

Transactions (user logged in pays)
(Transaction_id, user_id, price, Time_stamp, comment) pay immediately
Comment = payment or refund

After the person has selected seats and has paid, Enter the passenger info to PAX table (status confirmed), Trip data and Booking Table, and update seats of respective flights

Bookings
(PNR, user_id, transaction_id, status of booking) Pending confirmed canceled
(PNR, user_id) -> transaction_id, status of booking

If a user wants to see his bookings, query to Booking table, find corresponding PNRs, obtain Trip_info, from it get ssn and their details
(could use stored Prcedure)

If U need to cancel in Booking table change status, make a transaction back to user_id , Passenge status, seats, and seats in schedule Id

User Feedback
(PNR, user_id, rating)