Consider the database schema that you have designed for Project 4 which supports the information needs of the airline information system at various international airports in the states.

The airline information system must be implemented with the following functions using an embedded SQL language: (If there are more than one possible answer to the query, you must retrieve all of them.)

1. Search for the cheapest fare between any two input cities (i.e., any source and destination cities), which cannot be the same, in the database. The results must include the total fare, the flights involved (identified by flight numbers), and the departure and arrival time of each flight. The following options must be provided in searching for the cheapest fare:

   Option (a): The fare may require any number of stops in between the two input cities.
   Option (b): The fare involves at most \(n\) stops in between the two cities, where \(n\) is an input integer.
   Option (c): The fare should not require any stop-over for longer than \(n\) hours, where \(n\) is an input integer.
   Option (d): The fare cannot involve a stop in a particular city \(C\), where \(C\) is an input string.

2. Search for the shortest route between any two input cities (i.e., any source and destination cities), which cannot be the same, in the database. Four different options must be provided in searching for the shortest route according to:

   Option (a): The minimal distance from the source to the destination city. The results must include the total mileages, flight(s) involved, and the departure and arrival time of each flight.
   Option (b): The minimal number of stop-overs from the source to the destination city. The results must include the number of stop-overs, flight(s) involved, and the departure and arrival time of each flight.
   Option (c): The shortest time from the source to the destination city without counting the lay over time at the intermediate cities. The results must include the shortest time, flight(s) involved, and the departure and arrival time of each flight.
   Option (d): The shortest time from the source to the destination city, counting the lay over time at each intermediate city. The results must include the shortest time, flight(s) involved, and the departure and arrival time of each flight.

3. Search for the special routes between any two input cities (i.e., any source city \(S\) and destination city \(D\), and \(S \neq D\)) in the database such that

   Option (a): There are at most \(n\) \((n \leq 3)\) stops in between \(S\) and \(D\) and the total mileage cannot exceed \(m\), where \(m\) is an integer.
   Option (b): The airfare cannot exceed \(m\) dollars in between \(S\) and \(D\), where \(m\) is any positive numerical value less than 10,000.
Option (c): The departure time from S cannot be earlier than \( hh:mm \) and the arrival time at D cannot be later than \( hh:mm \), where \( 00 \leq hh \leq 23 \) and \( 00 \leq mm \leq 59 \).

Option (d): All the passengers, identified by names, who fly from S to D via city C are retrieved.

4. Insertion, deletion, and modification on the data in the database:

Option (a): Insert a new crew member record. Allow the new crew member to be assigned to one or more existing flights.

Option (b): Insert a new flight record. Allow the new flight to be assigned to one or more crew members.

Option (c): Modify an existing crew member record. (Any existing crew member record is allowed to be modified. Referential integrity constraints must be enforced.)

Option (d): Modify an existing flight record. (Any existing flight record is allowed to be modified. Referential integrity constraints must be enforced.)

Option (e): Insert a new or modify an existing maintenance record. (Any existing maintenance record is allowed to be modified. Referential integrity constraints must be enforced.)

Option (f): Delete an existing flight record. (Any existing flight record is allowed to be deleted. Referential integrity constraints must be enforced.)

Each item, i.e., (a), (b), …, listed under Parts (1) and (2) is an option, same as Parts (3) and (4), and they are mutually exclusive, and you must implement all the options in each part.

- **Evaluation:** your project will be evaluated based on the following criteria:

  - Disseminate and integrate the data provided for Project 5 into your database according to your database schema defined in Project 4.
  - Correct answer for each query, and invalid input error checking.
  - A good user interface (which should be menu-driven), and query results should be displayed in a proper tabular format (i.e., with appropriate title, column names, etc.)

- **Policy:**

  - Each student must set up an appointment with the TA to pass off the project on the due date. The sign-up sheet for appointments will be passed around in class on Tuesday, December 2.
  - You should make sure that the TA passes off your project in the TA office during your appointment, regardless the programming language you have chosen to implement this assignment. You should have your project ready to be passed off at least a few minutes before your appointment. Note that you are given only 15 minutes during your appointment to show that all the functions in this project assignment are correctly implemented.
  - You have ONLY ONE chance if you wait until the due date to pass off your project. However, you can come pass off your project as many times as you want with the TA during her regular office hours any time between now and December 5, the due date of this project assignment. DON’T WAIT UNTIL THE LAST MINUTE.

This project assignment is worth 150 points.