Database Systems, CSCI 4380-01
Homework # 1 Practice Problem for Question 1 Version 1.03

Question 1 [48 points]. Write the following queries using relational algebra. You may use any valid relational algebra expression, break into multiple steps as needed. However, please make sure that your answers are well-formatted and are easily readable. Also, pay attention to the attributes required in the output!

(a) Find the first and last names of all passengers who flew a round trip from New York to London and back with a layover of less than or equal to one day. The outbound flight must have been on an 'Airbus A320-200' and the return flight on a 'Boeing 777-300' occurring after the outbound flight. Also, the ticket price for the return flight must have been cheaper than the outbound flight.

See next pages for solution.

Solution using the text notation.

1. Initially we can create some ”helper” relations that isolate the relevant airport codes and aircraft codes. By isolating the airport codes, we can find the specific flights (from ‘Flights‘) that depart/arrive from an airport in a given city. Extracting the code from the ‘Aircraft‘ data for the specific airplanes will allow us to easily filter by the aircraft code per ‘Flight‘.

- Note that we are renaming all of the ”code” attributes. This is necessary for the airport codes because when we are finding a flight that leaves from NY and arrives in London, we will have both the NY airport code and the London airport code in the same relation (avoiding attribute collisions). Renaming the code attribute for the aircrafts is not strictly necessary, but changing it to ‘AircraftCode‘ will allow us to do a natural join-instead of a theta-join with ‘Flights‘ later on, and also won’t create ”duplicate” attributes as a result (”Code” and ”AircraftCode” referencing the same value).

```
NY_Codes(NY_C) = project_{Code}(select_{City = 'New York'}(Airports))
Lon_Codes(Lon_C) = project_{Code}(select_{City = 'London'}(Airports))
Airbus(AircraftCode) = project_{Code}(select_{Model = 'Airbus A320-200'}(Aircraft))
Boeing(AircraftCode) = project_{Code}(select_{Model = 'Boeing 777-300'}(Aircraft))
```

2. Next, we can use the codes that we isolated to find the flights that travel from NY to London and then the flights that travel from London to NY

```
F_Out = Flights join_{DepartureAirportCode = NY_C} NY_Codes
                join_{ArrivalAirportCode = Lon_C} Lon_Codes
F_Back = Flights join_{DepartureAirportCode = Lon_C} Lon_Codes
                 join_{ArrivalAirportCode = NY_C} NY_Codes
```

3. Before we combine the flights from `NY -> London` and `London -> NY`, we should do more filtering that the question asks for.

- The only valid outgoing flights should be using an ”Airbus” and the only valid returning flights should be using a ”Boeing”.

- Since the final query output is ”Passenger” information, we can extract all of the ‘PId‘s of passengers that went on each flight. Since we know that there will be no overlapping ‘FId‘s when joining the outgoing and returning flight relations, we can safely exclude it via a projection.

- We do need to keep the price of the flight and the arrival/departure time to London so that we can do another selection for the remaining requirements of the question.

```
Out(PId1,Amt1,Arr1) =
  project_{PassengerId,Amount,ActualArrival}(F_Out * Airbus * TicketFlights * Tickets)
Back(PId2,Amt2,Dep2) =
  project_{PassengerId,Amount,ActualDeparture}(F_Back * Boeing * TicketFlights * Tickets)
```

4. As the final steps we perform a theta-join that checks for every passenger that was on an outgoing flight and a returning flight, including the final ”selection” conditions regarding pricing and layover time.

```
RoundTrip = Out join_{PId1 = PId2 and Amt2 < Amt1 and Arr1 < Dep2 <= Arr1 + '1 day'} Back
project_{FirstName,LastName}(RoundTrip join_{PId1 = PassengerId} Passengers)
```

Full solution

```
NY_Codes(NY_C) = project_{Code}(select_{City = 'New York'}(Airports))
Lon_Codes(Lon_C) = project_{Code}(select_{City = 'London'}(Airports))
Airbus(AircraftCode) = project_{Code}(select_{Model = 'Airbus A320-200'}(Aircraft))
Boeing(AircraftCode) = project_{Code}(select_{Model = 'Boeing 777-300'}(Aircraft))

F_Out = Flights join_{DepartureAirportCode = NY_C} NY_Codes
                join_{ArrivalAirportCode = Lon_C} Lon_Codes
F_Back = Flights join_{DepartureAirportCode = Lon_C} Lon_Codes
                 join_{ArrivalAirportCode = NY_C} NY_Codes

Out(PId1,Amt1,Arr1) =
  project_{PassengerId,Amount,ActualArrival}(F_Out * Airbus * TicketFlights * Tickets)
Back(PId2,Amt2,Dep2) =
  project_{PassengerId,Amount,ActualDeparture}(F_Back * Boeing * TicketFlights * Tickets)

RoundTrip = Out join_{PId1 = PId2 and Amt2 < Amt1 and Arr1 < Dep2 <= Arr1 + '1 day'} Back
project_{FirstName,LastName}(RoundTrip join_{PId1 = PassengerId} Passengers)
```