

Database Systems, CSCI 4380-01

Homework #1 Practice Problem for Question 1

Jordyn Young

February 2026

Practice problem for Homework 1 Question 1

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!

Database Description. This is an Airline Data Analysis database. The database contains information about aircraft, airports, flights, bookings, and tickets. It contains the following relations:

Aircraft(Code, Model, Range)
Airports(Code, Name, City, Latitude, Longitude, Timezone)
Flights(FId, FlightNumber, ScheduledDeparture, ScheduledArrival, DepartureAirportCode, ArrivalAirportCode, AircraftCode, ActualDeparture, ActualArrival)
Tickets(TicketNo, PassengerId)
TicketFlights(TicketNo, FId, Fare, Amount)
Passengers(PassengerId, FirstName, LastName, DOB)

The key of each relation is underlined.

- (1) Return the flight number, scheduled departure, departure airport name and city, and the aircraft model for all flights scheduled to depart in February 2026 and the aircraft used for the flight has a range of at least 5000.

Solution using standard notation

- Rename Airports to represent departure airports

$$\rho_{DepAirports(DepCode, DepName, DepCity, DepLatitude, DepLongitude, DepTimezone)}(Airports)$$

This step creates a renamed copy of the *Airports* relation that represents departure airports. This lets us distinguish departure airport attributes from any other airport info. Only attribute names are modified here.

- Select flights scheduled to depart in February 2026

$$C1 = \text{ScheduledDeparture} :: \text{date} \geq \text{date}'02/01/2026' \wedge \text{ScheduledDeparture} :: \text{date} \leq \text{date}'02/28/2026'$$

$$R1 = \sigma_{C1}(Flights)$$

This step keeps only those flights whose scheduled departure date occurs in February 2026. All flights outside this time window are excluded.

- Select aircraft with range at least 5000

$$C2 = Range \geq 5000$$

$$R2 = \sigma_{C2}(Aircraft)$$

This step restricts the *Aircraft* relation to only have aircrafts that satisfy the minimum range requirement of at least 5000.

- Join flights with aircraft to attach the aircraft model

$$C3 = AircraftCode = Code$$

$$R3 = R1 \bowtie_{C3} R2$$

This join matches each flight in the relation with the aircraft used for that flight by comparing the flight's Aircraft Code with the aircraft's Code. The result attaches additional aircraft information to each remaining flight. At this point, every tuple represents a February 2026 flight that uses an aircraft with at least 5000.

- Join with departure airports to get departure name/city

$$C4 = DepartureAirportCode = DepCode$$

$$R4 = R3 \bowtie_{C4} DepAirports$$

This join connects each flight to its corresponding departure airport using the departure airport code. The departure airport's name and city are added to each flight tuple.

- Project flight number, scheduled departure, departure airport name, departure city, and model

$$Result = \pi_{FlightNumber, ScheduledDeparture, DepName, DepCity, Model}(R4)$$

The resulting relation contains one tuple per flight with the specified output attributes.

Final answer

$$\begin{aligned} & \rho_{DepAirports}(DepCode, DepName, DepCity, DepLatitude, DepLongitude, DepTimezone)(Airports) \\ R1 = & \sigma_{ScheduledDeparture::date \geq date '02/01/2026' \wedge ScheduledDeparture::date \leq date '02/28/2026'}(Flights) \\ R2 = & \sigma_{Range \geq 5000}(Aircraft) \\ R3 = & R1 \bowtie_{AircraftCode=Code} R2 \\ R4 = & R3 \bowtie_{DepartureAirportCode=DepCode} DepAirports \\ Result = & \pi_{FlightNumber, ScheduledDeparture, DepName, DepCity, Model}(R4) \end{aligned}$$

Solution using text notation

- Rename Airports to represent departure airports

```
DepAirports(DepCode,DepName,DepCity,DepLatitude,DepLongitude,DepTimezone) = Airports
```

This step creates a renamed copy of the *Airports* relation that represents departure airports. This lets us distinguish departure airport attributes from any other airport info. Only attribute names are modified here.

- Select flights scheduled to depart in February 2026

```
C1 = ScheduledDeparture::date >= date '02/01/2026' AND ScheduledDeparture::date <= date '02/28/2026'
```

```
R1 = select_C1(Flights)
```

This step keeps only those flights whose scheduled departure date occurs in February 2026. All flights outside this time window are excluded.

- Select aircraft with range at least 5000

```
C2 = Range >= 5000
```

```
R2 = select_C2(Aircraft)
```

This step restricts the *Aircraft* relation to only have aircraft that satisfy the minimum range requirement of at least 5000.

- Join flights with aircraft to attach the aircraft model

```
C3 = AircraftCode = Code
```

```
R3 = R1 join_C3 R2
```

This join matches each flight in the relation with the aircraft used for that flight by comparing the flight's AircraftCode with the aircraft's Code. The result attaches additional aircraft information to each remaining flight.

- Join with departure airports to get departure name/city

```
C4 = DepartureAirportCode = DepCode
```

```
R4 = R3 join_C4 DepAirports
```

This join connects each flight to its corresponding departure airport using the departure airport code. The departure airport's name and city are added to each flight tuple.

- Project flight number, scheduled departure, departure airport name, departure city, and model

```
Result = project_(FlightNumber, ScheduledDeparture, DepName, DepCity, Model)(R4)
```

The resulting relation contains one tuple per flight with the specified output attributes.

Final answer

```
DepAirports(DepCode,DepName,DepCity,DepLatitude,DepLongitude,DepTimezone) = Airports  
R1 = select_(ScheduledDeparture::date >= date '02/01/2026' AND ScheduledDeparture::date <= date '02/28/2026')  
    R2 = select_(Range >= 5000)(Aircraft)  
    R3 = R1 join_(AircraftCode = Code) R2  
    R4 = R3 join_(DepartureAirportCode = DepCode) DepAirports  
Result = project_(FlightNumber, ScheduledDeparture, DepName, DepCity, Model)(R4)
```