Week 4 Homework Solutions

All questions relate to the Airline Database that was the subject of HW1

In some cases lines from σ's or π's to their subscripts don't show up unless you view this document in Print Layout.

Be sure to develop complex queries in stages, using strategies written down in English – as is done in Lecture 4; without them I won't be able to understand what you've done – and you won't be able to understand if you come back to them after they've been graded. (Hint: this means that grades on complex queries will depend on use of clearly written strategies.)

Write each of the following queries in un-extended relational algebra. That is, use only σ, π, X, ρ, –, ∪ and ∩.

1. Find every flight for which there are no flight legs listed in the database.

$$\pi_{\text{FLIGHT.number}}(\text{FLIGHT}) \quad - \quad \pi_{\text{FLIGHT\_LEG.flight\_number}}(\text{FLIGHT\_LEG})$$

2. Find the flight(s) that have the highest fare(s). SHOULD BE LOWER.AMOUNT and HIGHER.AMOUNT

(set of all flights) - (set of flights that don't have the highest fare)
= (set of all flights) - (set of flights such that there's a flight with a higher fare)

$$\pi_{NUMBER}(FLIGHT) - [\pi_{LOWER.flight\_number}(\sigma_{LOWER.dnumber < HIGHER.dnumber}((\rho_{LOWER}(FARES) \times \rho_{HIGHER}(FARES)))]$$

3. Find every type of airplane that can land at all the airports.

(set of all airplane types) - (set of all airplane types for which there is at least one airport at which airplanes of that airplane type can't land)

***8
(set of all airplane types) - (set of all airplane types for which there is at least one airport at which airplanes of that airplane type can't land)
=
(set of ordered pairs (x,y) such that x is an airplane type and y is an airport code)
-
(set of ordered pairs (x,y) such that x is an airplane type and y is an airport code and airplanes of airplane type x can land at the airport with airport code y)

$$\pi_{\text{type\_name}}(\text{airplane\_type}) - \pi_{\text{type\_name}}\left[\rho_{\text{pairs(type\_name, airport\_code)}}\left[(\pi_{\text{airplane\_type.type\_name, airport.airport\_code}}(\text{airplane\_type} \times \text{airport})) - (\pi_{\text{airplane\_type, airport\_code}}(\text{can\_land}))\right]\right]$$

Write each of the following queries in extended relational algebra. That is, you can now use aggregate functions, arithmetic in σ subscripts, and entire relational algebra expressions whose value is a one-column, one-row table in σ subscripts – in addition to σ, π, X, ρ, −, ∪ and ∩. In the case of queries that you've already done in un-extended relational algebra, use the added expressive power of extended relational algebra to write simpler expressions.

4. Find the flight(s) that have the highest fare(s).

Strategy

$\pi_{\text{FARES.flight\_number}}\ \sigma_{\text{(fare for this flight) = (MAX of all fare amounts)}}\ \text{(FARES)}$

$\pi_{\text{FARES.flight\_number}}\ \sigma_{\text{(FARES.amount) = }(\pi_{\text{MAX(FARES.amount)}}\ \text{FARES})}\ \text{(FARES)}$

5. Find every type of airplane that can land at all the airports.

Strategy: The set of all airplane types such that the number of airports at which planes of that type can land is equal to the number of (all) airports

$$\pi_{\text{AIRPLANE\_TYPE.type\_name}}\ \sigma_{[\ \pi_{\text{COUNT}}\ \sigma_{(\text{CAN\_LAND.airplane\_type\_name}) = (\text{AIRPLANE\_TYPE.type\_name})}\ (\text{CAN\_LAND}) \ = \ \pi_{\text{COUNT}}(\text{AIRPORT}))]}\ (\text{AIRPLANE\_TYPE})$$

6. Find every flight whose second leg on 12/01/03 has at least three times as many available seats as its first leg.

Please note that the subscript on the σ needs two additional conditions, namely
LEG_INST1.leg_number = 1
AND
LEG_INST2.leg_number = 2

(I am unable to edit the expression, which was written in an earlier version of MSWord, in my current version of MSWord

This one doesn't use COUNT because number_of_available_seats is an attribute in the database rather than a COUNT or a SUM of attribute values.

$$\pi_{\text{LEG\_INST2.flight\_number}}\left(\sigma_{\substack{(\text{LEG\_INST1.flight\_number} = \text{LEG\_INST2.flight\_number}) \\ \text{AND } (\text{LEG\_INST1.date} = \text{LEG\_INST2.date}) \\ \text{AND } (\text{LEG\_INST1.date} = 12/01/03) \\ \text{AND } (\text{LEG\_INST2.number\_of\_available\_seats} >= 3 * \text{LEG\_INST1.number\_of\_available\_seats})}}\left(\left(\rho_{\text{LEG\_INST1}}(\text{LEG\_INSTANCE}) \text{ X } \rho_{\text{LEG\_NST2}}(\text{LEG\_INSTANCE})\right)\right)\right)$$

7. Find every flight for which there are no flight legs in the database. (Do this one using COUNT, even though the un-extended relational algebra version isn't very complicated; the version using COUNT is probably understandable by more potential readers than the un-extended relational algebra version, and readability is a positive property.)

$$\pi_{\text{FLIGHT.number}}\ \sigma_{[\ \pi_{\text{COUNT}}\ \sigma_{(\text{FLIGHT\_LEG.flight\_number} = \text{FLIGHT.number})}(\text{FLIGHT\_LEG}) \ = \ 0]}(\text{FLIGHT})$$

8. Find every flight leg for which there are twenty or more seat reservations on 12/01/03.

$\pi_{\text{FLIGHT\_LEG.flight\_number, FLIGHT\_LEG.leg\_number}}$ σ (FLIGHT_LEG)

[π_{COUNT} σ (SEAT_RESERVATIONS) >= 20]

(SEAT_RESERVATION.flight_number = FLIGHT_LEG.flight_number)
AND (SEAT_RESERVATIONS.leg_number = FLIGHT_LEG.leg_number)
AND (SEAT_RESERVATION.date = 12/01/03)

9. Find every flight leg for which there are exactly twenty seat reservations on 12/01/03.

$$\pi_{\substack{\text{FLIGHT\_LEG.flight\_number,} \\ \text{FLIGHT\_LEG.leg\_number}}} \; \sigma_{\left[\pi_{\text{COUNT}} \, \sigma_{\substack{(\text{SEAT\_RESERVATION.flight\_number} = \text{FLIGHT\_LEG.flight\_number}) \\ \text{AND } (\text{SEAT\_RESERVATIONS.leg\_number} = \text{FLIGHT\_LEG.leg\_number}) \\ \text{AND } (\text{SEAT\_RESERVATION.date} = 12/01/03)}} (\text{SEAT\_RESERVATIONS}) \;=\; 20\right]} (\text{FLIGHT\_LEG})$$

10. Find every type of airplane that can land at more airports in New York, New York than at airports in Boston, Massachusetts.

$$\pi_{\text{AIRPLANE\_TYPE.type\_name}}\ \sigma\ (\text{AIRPLANE\_TYPE})$$

$$[\ \pi_{\text{COUNT}}\ \sigma\ (\text{CAN\_LAND} \times \text{AIRPORT})]$$

(CAN_LAND.airplane_type_name = AIRPLANE_TYPE.type_name)
AND (CAN_LAND.airport_code = AIRPORT.airport_code)
AND (AIRPORT.city = “New York”)
AND (AIRPORT.state = “New York”)

$$>$$

$$[\ \pi_{\text{COUNT}}\ \sigma\ (\text{CAN\_LAND} \times \text{AIRPORT})]$$

(CAN_LAND.airplane_type_name = AIRPLANE_TYPE.type_name)
AND (CAN_LAND.airport_code = AIRPORT.airport_code)
AND (AIRPORT.city = “Boston”)
AND (AIRPORT.state = “Massachusetts”)