Week 4 Homework

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

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 , , X,  and .**

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

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

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

**Write each of the following queries in extended relational algebra. That is, you can now use aggregate functions, arithmetic in  subscripts, and entire relational algebra expressions whose value is a one-column, one-row table in  subscripts – in addition to , , X,  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).

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

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

)

l1.date = 12/01/03) and (l2.date = 12/01/03)

and (11.number\_of\_available\_seats >=3 \* l2.number\_of\_available)

and (l1.flight\_numer = flight\_leg.flight\_numer)

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.)

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

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

(leg\_instance.number\_of\_available\_seat = 20) and

(leg\_instance.date = 12/01/03) and

(leg\_instace.flight\_number = flight.number )

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

**** AIRPLANE\_TYPE.type\_name(AIREPLANE\_TYPE)) (**** (**** c1(can\_land) **X ** c2(can\_land))

(c1.airplane\_type\_name = airplane\_type.type\_name) and

[[**** count(\*) (****(c1))] [[**** c1.airport\_type\_name (c1) (****(airport)) (airport.airport\_code = c1.airport\_code) and(airport = New York) (airport = New York)] > [**** count(\*) (****(c2))] [[**** c2.airport\_type\_name (c2) (****(airport)) ] (airport.airport\_code = c2.airport\_code) and(airport = Boston)

]