

## Solutions — Relational Algebra Expressions

a.

$$Answer_a = \pi_{eid}(\sigma_{aname='Boeing'}(C \bowtie A))$$

b. Let  $I = \pi_{eid}(\sigma_{aname='Boeing'}(C \bowtie A))$ . This can be used to answer the question (b) as follows.

$$Answer_b = \pi_{ename}(I \bowtie E)$$

Alternatively, you can use

$$Answer_b = \pi_{ename}(\sigma_{aname='Boeing'}(E \bowtie (C \bowtie A)))$$

c. The idea is to find aircraft with *cruiserange* greater than or equal the distance between Boon and Madras, which can be obtained from the `Flight` table. This is done as follows.

$$R_1 = \pi_{distance}(\sigma_{from='Bonn' \text{ and } to='Madras'}(F))$$

and

$$Answer_c = \pi_{aid}(\sigma_{cruiserange \geq distance}(A \times R_1))$$

d. First, we need to find all pilots whose salary is more than \$100,000 and the aircrafts that they can flight. This is done is the following query:

$$R_1 = \pi_{eid,aid}(\sigma_{salary > 100000}(E \bowtie C))$$

To find the aids of aircraft that can be controlled by by all of them, we can use the following

$$R_2 = R_1 / \pi_{eid}(R_1)$$

The expression

$$Answer_d = \pi_{flno}(\sigma_{cruiserange \geq distance}(R_2 \times F))$$

gives us all flights that can be piloted by every pilot with salary more than \$100,000.

An alternative is to first find the flights and then compute the division. This might result in different answer. The statement is not specific enough.

e. First we find all pilots who can operate planes with a range greater than 3,000 miles and then we subtract from those who are certified for some Boeing aircraft.

$$R_1 = \pi_{eid}(\sigma_{cruiserange > 3000}(C \bowtie A))$$

using the answer from (a), we have

$$Answer_e = \pi_{ename}((R_1 - Answer_a) \bowtie E)$$

Note that the projection of *ename* is necessary since the question asks for the names of the pilots.

- f. The idea is to create two tables, one does not contain the highest salary and the other one does. Set difference of the second one from the first one will give the desired result. This can be done as follows.

$$R_1 = \pi_{eid, eid1}(\sigma_{salary > salary1}(E \times E[eid1, ename1, salary1]))$$

The table  $R_1$  has two columns, the first column has the eids of employees whose salary is higher than someone, and the second column has the eids of employees whose salary is lower than someone. So, the answer is

$$Answer_f = \pi_{eid}(R_1) - \pi_{eid1}(R_1)[eid]$$

Note the renaming operator in the second operand of the above expression: without this operator, the result will be an empty set due to non-compatibility of the two tables.

- g. This is done similar to the previous query, using the result of the above query. First, we get the eids of the employee with highest salary. Next, we remove them from the eids. Then, we repeat what we have done in (f.) I leave it for you to enjoy.
- h. We do not have the capability to count using relational algebra. Hence, this query cannot be answered by a relational algebra expression.
- i. Unlike the previous query, we do not need the capability to count and can still answer this query. The idea is to find people who are certified for more than three aircraft ( $R_1$ ) and pilots who are certified for more than two aircraft ( $R_2$ ). The set difference  $R_2 - R_1$  provides the answer for the question.  $R_1$  can be obtained from the join  $C \bowtie C_1 \bowtie C_2$  where  $C_1 = C_2 = C$  by selecting the tuples with pairwise different *aids*.  $R_2$  can be computed in a similar fashion.
- j. The inability of computing the sum using relational algebra means that we cannot write a relational expression for this query.
- k. The inability of computing the transitive closure does not allow us to answer this query using relational algebra.

## Solutions — SQL

- a. 

```
Select A.aname
From   Aircraft A
Where  NOT Exists (Select * From Employee E, Certified C
                  Where  E.eid = C.eid and C.aid = A.aid
                  and E.salary < 80000)
```

The subquery finds people who is certified to operate *A.aid* and has salary less than 80000.

- b. This can be done step by step as follows. First, we find pilots who is certified for more than three aircraft.

```
Select * From Certified C Group by C.eid Having count(*) > 3
```

The next step is to join the above with the Aircraft table and find the answer.

```
Select  C.eid, A.aid, A.cruiserrange
From    Certified C, Aircraft A
Where   A.aid = C.aid
        and (C.eid IN
            Select C1.eid
            From Certified C1 Group by C1.eid Having count(*) > 3)
        and A.cruiserrange >=
            (Select Max(A1.cruiserrange)
             From Aircraft A1
             Where A.aid IN
             Select C1.aid
             From Certified C1 Group by C1.eid Having count(*) > 3)
```

c. We need to find the cheapest route from LA to Honolulu but the tables do not provide us this information.

d. 

```
Select A.aname, AVG(E.salary)
From Aircraft A, Certified E, Employee E
Where A.aid = C.aid and C.eid = E.eid
      and A.cruiserange>1000
Group By A.aid
\end{itemize}
```

```
\item[e.]
\begin{verbatim}
Select E.ename
From Aircraft A, Certified E, Employee E
Where A.aid = C.aid and C.eid = E.eid and A.aname = 'Boeing'
```

```
\item[f.] The query is not quite clear. The following answer
is for 'direct flights'.'
\begin{verbatim}
Select A.aid
From Aircraft A, Flight F
Where A.cruiserange >= F.distance and F.from = 'LAX' and F.to = 'CHI'
```

g. This is similar to the first question. Instead of writing the SQL command, that answer the question directly, we write a command that find the routes which can be operated by some pilots who makes no more than \$100,000.

h. We first find those those who can operate aircrafts with cruiserange more than 3000 and then remove those who is certified for some Boeing aircraft.

```
Select E.ename
From Employee E
Where
  E.eid IN
  (
    (Select C.eid From Certified C, Aircraft A
     Where C.eid = A.eid and A.cruiserange>3000)
    EXCEPT
    (Select C.eid From Certified C, Aircraft A
     Where C.eid = A.eid and A.aname='Boeing')
  )
```

i. This question is not complicated, it is long though.

```
Select F.departs
From Flight F
Where F.from = 'Madison' and F.To='New York' and
      F.arrives <= 6.p.m
UNION
Select F.departs
From Flight F, Flight F1
Where F.from = 'Madison' and
      F.To=F1.From and F1.To='New York' and
      F.arrives < F1.departs and
      F1.arrives <= 6 p.m.
UNION
Select F.departs
```

```
From Flight F, Flight F1, Flight F2
Where F.from = 'Madison' and
      F.To=F1.From and F1.To=F2.From and F2.To='New York' and
      F.arrives < F1.departs and
      F1.arrives < F2.departs and
      F2.arrives <= 6 p.m.
```

```
j. Select Avg(E.salary)
   From   Employee E, Certified C
   Where  E.edi = C.eid
   -
   Select Avg(E.Salary) from Employee E
```

```
k. Select E.ename, E.salary
   From   Employee E, Certified C
   Where  E.eid NOT IN (Select C.eid From Certified C)
```