CS2102 Database Systems
2013/2014 Semester I

Tutorial #2 Relational Calculus

Consider the following relations containing airline flight information:

Flight (*flno*:integer, *from*:string, *to*:string, *distance*:integer, *departs*:time, *arrives*:time)
Aircraft (*aid*:integer, *aname*:string, *cruiserange*:string)
Certified (*eid*:integer, *aid*:integer)
Employee (*eid*:integer, *ename*:string, *salary*:integer)

The Employee relation describes pilots and other kinds of employees; every pilot is certified for some aircraft (otherwise, s/he would not qualify as a pilot), and only pilots are certified to fly.

Write the following queries in tuple relational calculus. Note that some of these queries may not be expressible in relational calculus. For such queries, explain why they cannot be expressed.

1. Find the *eid*s of pilots certified for some Boeing aircraft.

{T | ∃C ∈ Certified ∃A∈ Aircraft (A.aid = C.aid ∧ A.aname = 'Boeing' ∧ C.eid=T.eid)}

2. Find the names of pilots certified for some Boeing aircraft.

{T| ∃E∈ Employee ∃C∈ Certified ∃A∈ Aircraft (E.eid = C.eid ∧ A.aid = C.aid ∧ A.aname = 'Boeing' ∧ E.ename=T.ename)}

3. Find the *aid*s of all aircrafts that can be used on non-stop flights from 'L.A.' to 'N.Y.'.

{T| ∃A∈ Aircraft ∃F∈ Flight (F.from='L.A.' ∧ F.to='N.Y.' ∧ A.cruiserange > F.distance ∧ A.aid=T.aid)}

4. Identify the flights that can be piloted by every pilot whose salary is more than 100,000.

{T| ∃F∈ Flight ∀E∈ Employee ((∃C∈ Certified (C.eid =E.eid∧ E.salary > 100,000)) ⟹(∃A∈Aircraft ∃C1∈ Certified(E.eid = C1.eid ∧c1.aid=A.aid^ A.cruiserange > F.distance)) ∧ F.flno=T.flno))}

5. Find the names of pilots who can operate planes with a range greater than 3000 miles but are not certified on any Boeing aircraft.

{T| ∃ E∈ Employee ∃ C1∈ Certified ∃ A1∈ Aircraft (A1.aid=C1.aid ∧ E.eid=C1.eid ∧ A1.cruiserange>3000) ∧ ¬(∃ C2∈ Certified ∃ A2∈ Aircraft (A2.aname='Boeing' ∧ C2.aid=A2.aid ∧ C2.eid=E.eid)) ∧ E.ename=T.ename)}

6. Find the *eid*s of employees who make the highest salary.

{T| ∃ E1∈ Employee ∀ E2∈ Employee (E2.salary≤E1.salary ∧ E1.eid=T.eid)}

7. Find the *eid*s of employees who make the second highest salary.

{T| ∃ E∈ Employee∃ E1∈ Employee ∀ E2∈ Employee ∀ E3∈ Employee (E1.salary > E.salary ∧ (E2.salary > E.salary) ⟹ (E3.salary<=E2.salary) ∧ E.eid=T.eid) }

8. Find the *eid*s of employees who are certified for the largest number of aircrafts.

Cannot be expressed in calculus (or algebra) because there is no operator count. Query can be expressed in SQL as follows:

```
SELECT Temp.eid
FROM (SELECT C.eid AS eid, COUNT (C.aid) AS num,
        FROM Certified C
        GROUP BY C.eid) AS Temp
WHERE Temp.num = (SELECT MAX (Temp.num) FROM Temp
```

9. Find the total amount paid to employees as salaries.

Cannot be expressed in calculus (or algebra) because there is no operator to sum values. Query can be expressed in SQL as follows:

```
SELECT SUM (E.salaries)
FROM Employee E
```