Questions

1. The following relations keep track of airline flight information:

   Flights(fno: integer, from: string, to: string, distance: integer, departs: time, arrives: time, price: integer)
   Aircraft(aid: integer, aname: string, cruisingrange: integer)
   Certified(eid: integer, aid: integer)
   Employees(eid: integer, ename: string, salary: integer)

Note that the Employees relation describes pilots and other kinds of employees as well; every pilot is certified for some aircraft, and only pilots are certified to fly. Write each of the following queries in SQL.

   a. Find the names of aircraft such that all pilots certified to operate them earn more than $80,000.
   b. For each pilot who is certified for more than three aircraft, find the eid and the maximum cruisingrange of the aircraft for which she or he is certified.
   c. Find the names of pilots whose salary is less than the price of the cheapest route from Los Angeles to Honolulu.
   d. For all aircraft with cruisingrange over 1000 miles, find the name of the aircraft and the average salary of all pilots certified for this aircraft.
   e. Find the names of pilots certified for some Boeing aircraft.
   f. Find the aids of all aircraft that can be used on routes from Los Angeles to Chicago.
   g. Identify the routes that can be piloted by every pilot who makes more than $100,000.
   h. Print the enames of pilots who can operate planes with cruisingrange greater than 3000 miles but are not certified on any Boeing aircraft.
   i. A customer wants to travel from Madison to New York with no more than two changes of flight. List the choice of departure times from Madison if the customer wants to arrive in New York by 6 p.m.
   j. Compute the difference between the average salary of a pilot and the average salary of all employees (including pilots).
   k. Print the name and salary of every nonpilot whose salary is more than the average salary for pilots.
l. Print the names of employees who are certified only on aircrafts with cruising range longer than 1000 miles.

m. Print the names of employees who are certified only on aircrafts with cruising range longer than 1000 miles, but on at least two such aircrafts.

n. Print the names of employees who are certified only on aircrafts with cruising range longer than 1000 miles and who are certified on some Boeing aircraft.

Ans:

a. SELECT DISTINCT A.aname
   FROM Aircraft A
   WHERE A.aid IN (SELECT C.aid
                    FROM Certified C, Employees E
                    WHERE C.eid = E.eid
                    AND NOT EXISTS (SELECT *
                                    FROM Employees E1
                                    WHERE E1.eid = E.eid
                                    AND E1.salary < 80000 ))

b. SELECT C.eid, MAX (A.cruisingrange)
   FROM Certified C, Aircraft A
   WHERE C.aid = A.aid
   GROUP BY C.eid
   HAVING COUNT(*) > 3

c. SELECT DISTINCT E.ename
   FROM Employees E
   WHERE E.salary < (SELECT MIN (F.price)
                      FROM Flights F
                      WHERE F.from = 'Los Angeles' AND F.to = 'Honolulu')

d. Observe that aid is the key for Aircraft, but the question asks for aircraft names; we deal with this complication by using an intermediate relation Temp:
   SELECT Temp.name, Temp.AvgSalary
   FROM (SELECT A.aid, A.aname AS name, AVG (E.salary) AS AvgSalary
          FROM Aircraft A, Certified C, Employees E
          WHERE A.aid = C.aid AND C.eid = E.eid AND A.cruisingrange > 1000
          GROUP BY A.aid, A.aname ) AS Temp

e. SELECT DISTINCT E.ename
   FROM Employees E, Certified C, Aircraft A
   WHERE E.eid = C.eid AND C.aid = A.aid AND A.aname LIKE 'Boeing%'

f. SELECT A.aid
   FROM Aircraft A
   WHERE A.cruisingrange > (SELECT MIN (F.distance)
                             FROM Flights F
                             WHERE F.from = 'Los Angeles' AND F.to = 'Chicago')
g. SELECT DISTINCT F.from, F.to
FROM Flights F
WHERE NOT EXISTS (SELECT * 
FROM Employees E 
WHERE E.salary > 100000 AND NOT EXISTS (SELECT * 
FROM Aircraft A, Certified C 
WHERE A.cruisingrange > F.distance AND E.eid = C.eid AND A.aid = C.aid))

h. SELECT DISTINCT E.ename
FROM Employees E 
WHERE E.eid IN ((SELECT C.eid 
FROM Certified C 
WHERE EXISTS (SELECT A.aid 
FROM Aircraft A 
WHERE A.aid = C.aid AND A.cruisingrange > 3000) AND NOT EXISTS (SELECT A1.aid 
FROM Aircraft A1 
WHERE A1.aid = C.aid AND A1.aname LIKE 'Boeing%')))

i. SELECT F.departs
FROM Flights F 
WHERE F.fno IN ((SELECT F0.fno 
FROM Flights F0 
WHERE F0.from = 'Madison' AND F0.to = 'New York' AND F0.arrives < '18:00')
UNION (SELECT F0.fno 
FROM Flights F0, Flights F1 
WHERE F0.from = 'Madison' AND F0.to <> 'New York' AND F0.to = F1.from AND F1.to = 'New York' AND F1.departs > F0.arrives AND F1.arrives < '18:00')
UNION (SELECT F0.fno 
FROM Flights F0, Flights F1, Flights F2 
WHERE F0.from = 'Madison' AND F0.to = F1.from AND F1.to = F2.from AND F2.to = 'New York' AND F0.to <> 'New York' AND F1.to <> 'New York' AND F1.departs > F0.arrives AND F2.departs > F1.arrives AND F2.arrives < '18:00'))
j. \[ \text{SELECT} \quad \text{Temp1.avg} - \text{Temp2.avg} \\
\text{FROM} \quad (\text{SELECT} \quad \text{AVG} \text{ (E.salary)} \quad \text{AS avg} \\
\text{FROM} \quad \text{Employees E} \\
\text{WHERE} \quad \text{E.eid} \quad \text{IN} \quad (\text{SELECT} \quad \text{DISTINCT} \text{ C.eid} \\
\text{FROM} \quad \text{Certified C}) \quad \text{AS Temp1} ) \quad \text{AS} \quad \text{Temp2} ) \\
\]

k. \[ \text{SELECT} \quad \text{E.ename, E.salary} \\
\text{FROM} \quad \text{Employees E} \\
\text{WHERE} \quad \text{E.eid} \quad \text{NOT IN} \quad (\text{SELECT} \quad \text{DISTINCT} \text{ C.eid} \\
\text{FROM} \quad \text{Certified C}) \quad \text{AND} \quad \text{E.salary} \quad \text{>} \quad (\text{SELECT} \quad \text{AVG} \text{ (E1.salary)} \\
\text{FROM} \quad \text{Employees E1} \\
\text{WHERE} \quad \text{E1.eid} \quad \text{IN} \quad (\text{SELECT} \quad \text{DISTINCT} \text{ C1.eid} \\
\text{FROM} \quad \text{Certified C1}) ) \\
\]

l. \[ \text{SELECT} \quad \text{E.ename} \\
\text{FROM} \quad \text{Employees E, Certified C, Aircraft A} \\
\text{WHERE} \quad \text{C.aid} = \text{A.aid} \quad \text{AND} \quad \text{E.eid} = \text{C.eid} \\
\text{GROUP BY} \quad \text{E.eid, E.ename} \\
\text{HAVING} \quad \text{EVERY} \quad (\text{A.cruisingrange} > 1000) \]

m. \[ \text{SELECT} \quad \text{E.ename} \\
\text{FROM} \quad \text{Employees E, Certified C, Aircraft A} \\
\text{WHERE} \quad \text{C.aid} = \text{A.aid} \quad \text{AND} \quad \text{E.eid} = \text{C.eid} \\
\text{GROUP BY} \quad \text{E.eid, E.ename} \\
\text{HAVING} \quad \text{EVERY} \quad (\text{A.cruisingrange} > 1000) \quad \text{AND COUNT (*)} \quad > \quad 1 \]

n. \[ \text{SELECT} \quad \text{E.ename} \\
\text{FROM} \quad \text{Employees E, Certified C, Aircraft A} \\
\text{WHERE} \quad \text{C.aid} = \text{A.aid} \quad \text{AND} \quad \text{E.eid} = \text{C.eid} \\
\text{GROUP BY} \quad \text{E.eid, E.ename} \\
\text{HAVING} \quad \text{EVERY} \quad (\text{A.cruisingrange} > 1000) \quad \text{AND ANY} \quad (\text{A.aname} = \text{'Boeing'}) \]
2. Consider the following relational schema. An employee can work in more than one department; the pct_time filed of the Works relation shows the percentage of time that a given employee works in a given department.

\[
\text{Emp}(\text{eid}: \text{integer}, \text{ename}: \text{string}, \text{age}: \text{integer}, \text{salary}: \text{real}) \\
\text{Works}(\text{eid}: \text{integer}, \text{did}: \text{integer}, \text{pct}\_\text{time}: \text{integer}) \\
\text{Dept}(\text{did}: \text{integer}, \text{dname}: \text{string}, \text{budget}: \text{real}, \text{managerid}: \text{integer})
\]

Write the following queries in SQL:

a. Print the names and ages of each employee who works in both the Hardware department and the Software department.

\[
\text{SELECT E.ename, E.age} \\
\text{FROM Emp E, Works W1, Works W2, Dept D1, Dept D2} \\
\text{WHERE E.eid = W1.eid} \land \text{W1.did = D1.did} \land \text{D1.dname = ‘Hardware’} \land \text{E.eid = W2.eid} \land \text{W2.did = D2.did} \land \text{D2.dname = ‘Software’}
\]

b. For each department with more than 20 full-time-equivalent employees (i.e., where the part-time and full-time employees add up to at least that many full-time employees), print the did together with the number of employees that work in that department.

\[
\text{SELECT W.did, COUNT (W.eid)} \\
\text{FROM Works W} \\
\text{GROUP BY W.did} \\
\text{HAVING 2000 < (SELECT SUM (W1.pct\_time)} \\
\text{FROM Works W1} \\
\text{WHERE W1.did = W.did )}
\]

Ans:

a. SELECT E.ename, E.age \\
FROM Emp E, Works W1, Works W2, Dept D1, Dept D2 \\
WHERE E.eid = W1.eid AND W1.did = D1.did AND D1.dname = ‘Hardware’ AND E.eid = W2.eid AND W2.did = D2.did AND D2.dname = ‘Software’

b. SELECT W.did, COUNT (W.eid) \\
FROM Works W \\
GROUP BY W.did \\
HAVING 2000 < (SELECT SUM (W1.pct\_time) \\
FROM Works W1 \\
WHERE W1.did = W.did )
c. SELECT E.ename
FROM Emp E
WHERE E.salary > ALL (SELECT D.budget
FROM Dept D, Works W
WHERE E.eid = W.eid AND D.did = W.did)

d. SELECT DISTINCT D.managerid
FROM Dept D
WHERE 1000000 < ALL (SELECT D2.budget
FROM Dept D2
WHERE D2.managerid = D.managerid)

e. SELECT E.ename
FROM Emp E
WHERE E.eid IN (SELECT D.managerid
FROM Dept D
WHERE D.budget = (SELECT MAX (D2.budget)
FROM Dept D2))

f. SELECT D.managerid
FROM Dept D
WHERE 5000000 < (SELECT SUM (D2.budget)
FROM Dept D2
WHERE D2.managerid = D.managerid)

g. SELECT DISTINCT tempD.managerid
FROM (SELECT DISTINCT D.managerid, SUM (D.budget) AS tempBudget
FROM Dept D
GROUP BY D.managerid ) AS tempD
WHERE tempD.tempBudget = (SELECT MAX (tempD.tempBudget)
FROM tempD)

h. SELECT E.ename
FROM Emp E, Dept D
WHERE E.eid = D.managerid GROUP BY E.eid, E.ename
HAVING EVERY (D.budget > 1000000) AND ANY (D.budget < 5000000)