CS 2441 Homework 3 and Homework 4 Sample Solutions.

HW3, Ques.1: In this question repeat Question 2 of Homework 2. Provide SQL code for the queries in Question 2, Homework 2.

Easy solution: translate directly from the Relational algebra queries from Homework 2.

SELECT *
FROM Location
WHERE MainAttraction = ‘Beach’;

SELECT Place, Name
FROM Visited
WHERE Year < 2000;

SELECT Name
FROM Visited, Person
WHERE Visited.ID = Person.ID AND Year = 2003;

SELECT P1.name, P2.name
FROM Person P1, P2, Visited V1, V2
WHERE P1.ID not= P2.ID AND V1.ID = P1.ID AND V2.ID = P2.ID AND

For the last query, find for each person all the places they visited. This set must be superset (or equal to) the set of all locations.
SELECT P1.name
FROM Person P1, Visited V1, Locations L1
WHERE
  (SELECT V1.PlaceName /*this gives set of all places visited by person P1 */
   FROM Visited V1
   WHERE V1.ID = P1.ID)
  CONTAINS /*set of all places*/
  (SELECT L1.PlaceName
   FROM Location L1);

Ques.2: In this question, provide SQL queries for the queries in Question 4 of Homework 2.

SELECT Name, Street, City
FROM EMP, DEPT
WHERE DNO = DNUM AND DNAME = ’Research’;
SELECT P.Pnum, D.DNUM, E.NAME,
FROM PROJECT P, DEPT D, EMP E
WHERE P.PLocation='Stafford' AND P.DNO=D.DNUM AND D.MGRSSN=E.SSN;

SELECT E1.Name, E2.Name
FROM EMP E1, EMP E2
WHERE E1.SuperSSN = E2.SSN;

Homework 4 Sample Solutions

Ques.1: For the COMPANY database in Homework 2, and shown below with additional table (called DEPENDENTS), provide SQL queries for the following:

- List all project numbers associated with Smith, either as manager of a controlling department or as an employee working on the project.
  
  (SELECT PNO
   FROM WORKS-ON, EMP
   WHERE ESSN=SSN AND Name= ‘Smith’)
  UNION
  (SELECT PNO
   FROM DEPT, PROJECTS,EMP
   WHERE MGRSSN=SSN AND Name=’Smith’ AND
   DEPT.DNUM=PROJECTS.DNO);

- Find names of employees who work on the same project that Smith works on.
  
  SELECT E1.Name
  FROM EMP E1, WORKS-ON W1
  WHERE E1.SSN = W1.ESSN AND W1.PNO  IN
  (SELECT W2.PNO
   FROM EMP E2, WORKS-ON W2
   WHERE E2.Name= ‘Smith’ AND
   E2.SSN = W2.ESSN);

- Find names of employees who have no dependents.
  
  SELECT E1.Name
  FROM EMP E
  WHERE not exists
    (SELECT *
     FROM DEPENDENTS D
     WHERE E.SSN=D.ESSN);

- Find names of employees with at least two dependents.
  
  SELECT E.Name
FROM EMP E
WHERE (SELECT count(*)
FROM DEPENDENT D
WHERE E.SSN=D.ESSN) >= 2;

• Find departments with at least 10 employees. The solution is written for the case when we want to find the name of the department; if only the number is desired then there is no need for the join between EMP and DEPT.

SELECT D.DNUM, D.DName
FROM DEPT D, EMP E
WHERE E.DNO= D.DNUM
GROUPBY D.DNUM
HAVING (count(*) ) >=10;

• Find average salary and department names, for each department (the average salary of a department is the average of all salaries of employees assigned to a department).

SELECT DNAME, avg(Salary)
FROM DEPT D, EMP E
WHERE E.DNO = D.DNUM
GROUPBY D.DNUM;

• Find the department number and name of the department with the largest average salary (of its employees).

SELECT DNUM, DNAME
FROM DEPT D, EMP E
WHERE D.DNUM=E.DNO
GROUPBY D.DNUM
HAVING avg(Salary) >=ALL
(SELECT avg(Salary)
FROM DEPT X, EMP Y
WHERE X.DNUM=Y.DNO
GROUPBY DNUM):

• Give a 50% pay-cut to the employee with the highest salary.

UPDATE EMP E
SET E.Salary = 0.5*E.Salary
WHERE E.Salary >=ALL
(SELECT X.Salary
FROM EMP X);
CREATE TABLE EMP (  
  SSN int PRIMARY KEY,  
  Name Varchar(50),  
  Birthdate Date,  
  Street Varchar(50),  
  City Varchar(50),  
  DNO int,  
  SuperSSN int,  
  Salary Real,  
  FOREIGN KEY (DNO) REFERENCES DEPT(DNUM),  
  FOREIGN KEY (SuperSSN) REFERENCES EMP(SSN));

CREATE TABLE DEPT (  
  DNUM int PRIMARY KEY,  
  DNAME Varchar(50),  
  MGRSSN int,  
  UNIQUE DNAME,  
  FOREIGN KEY (MGRSSN) REFERENCES EMP(SSN));

CREATE TABLE PROJECTS(  
  PNum int PRIMARY KEY,  
  Pname Varchar(50),  
  Plocation Varchar(50),  
  DNO int,  
  FOREIGN KEY (DNO) REFERENCES DEPT(DNUM));

CREATE TABLE WORKS(  
  ESSN int,  
  PNO int,  
  HOURS int,  
  PRIMARY KEY (ESSN,PNO),  
  FOREIGN KEY (ESSN) REFERENCES EMP(SSN),  
  FOREIGN KEY (PNO) REFERENCES PROJECTS(PNUM));

CREATE TABLE DEPENDENT(  
  ESSN int NOT NULL,  
  DEPNAME Varchar(50) NOT NULL,  
  SEX char,  
  BDATE Date,  
  RELATIONSHIP Varchar(10),  
  PRIMARY KEY (ESSN,DEPNAME),  
  FOREIGN KEY (ESSN) REFERENCES EMPLOYEE(SSN));