**Question #2**

EMP(EID: integer primary key, EName: string, Age: integer, Salary: real)

WORKS(EID: integer, DID: integer, primary key (EID, DID), Pct time: integer)

DEPT(DID: integer primary key, Budget: real, ManagerID: integer)

2.1) Employees must make a minimum salary of 2000$.

Because we are given this condition that needs to be satisfied, the CHECK constraint is the best way to make sure the condition is satisfied

EMP(EID: integer primary key, EName: string, Age: integer, Salary: real **CHECK (Salary >= 2000)**)

This will make sure that any salary input is at least 2000.

2.2) Every manager must also be an employee

Since every manager must also be an employee, we will need a combination of an assertion as well as a check constraint

CREATE ASSERTION IsManagerAnEmployee

CHECK ((SELECT COUNT (\*)

FROM DEPT D

WHERE D.ManagerID NOT IN (SELECT \* FROM EMP)) = 0)

2.3) The total percentage of all appointments for an employee must be under 100.

In this case, once again we must use a check constraint in the WORKS table with the Pct time attribute.

WORKS(EID: integer, DID: integer, primary key (EID, DID), Pct time: integer,

CHECK ( (SELECT COUNT (EID)

FROM Works

GROUP BY EID

HAVING Sum(Pct time) > 100) = 0)))

2.4) A manager must always have a higher salary than any employee that he or she managers. Now since the manager has to make higher salary than his own employees, we will make sure of assertion and check constraint at the same time

CREATE ASSERTION ManagerHasHigherSalary

CHECK (SELECT E.EID

FROM EMP E, EMP M, WORKS W, DEPT D

WHERE E.WID = W.EID AND W.DID = D.DID AND D.ManagerID = M.EID AND E.Salary > M.Salary)

2.5) Whenever an employee is given a raise, the manager’s salary must be increased to be at least as much. This can be satisfied by creating a trigger that will increase the manager’s salary equal to the amount the employee who got a raise if the manager’s salary is less then the employee’s new salary

CREATE TRIGGER GiveRaise AFTER UPDATE ON EMP

WHEN old.Salary < new.Salary

FOR EACH ROW

BEGIN

UPDATE EMP M

SET M.Salary = new.Salary

WHERE M.Salary < new.Salary

AND M.EID IN (SELECT D.ManagerID

FROM EMP E, WORKS W, DEPT D

WHERE E.EID = new.EID

AND E.EID = W.EID AND W.DID = D.DID);

END

2.6) Whenever an employee is given a raise, the salary of his/her manager must be increased to be at least as much. Further, whenever an employee is given a raise, the department’s budget must be increased to be greater than the sum of salaries of all employees in the department.

We can see that this condition is similar to the one above but with further conditions so we can extend the trigger to apply the new condition. We will basically add an Update command to increase the budget of the department by the amount of the raise.

CREATE TRIGGER GiveRaise AFTER UPDATE ON EMP

WHEN old.Salary < new.Salary

FOR EACH ROW

DECLARE raise REAL;

BEGIN

raise := new.Salary – old.Salary;

UPDATE EMP M

SET M.Salary = new.Salary

WHERE M.Salary < new.Salary

AND M.EID IN (SELECT D.ManagerID

FROM EMP E, WORKS W, DEPT D

WHERE E.EID = new.EID

AND E.EID = W.EID AND W.DID = D.DID);

UPDATE Dept D

SET D.Budget = D.Budget + raise

WHERE D.DID IN (SELECT W.DID

FROM EMP E, WORKS W, DEPT D

WHERE E.EID = new.EID AND E.EID = W.EID

AND D.DID = W.DID AND D.Budget < (SELECT Sum(E2.Salary)

FROM EMP E2, WORKS W2

WHERE E2.EID = W2.EID AND W2.DEPT = D.DID));

END