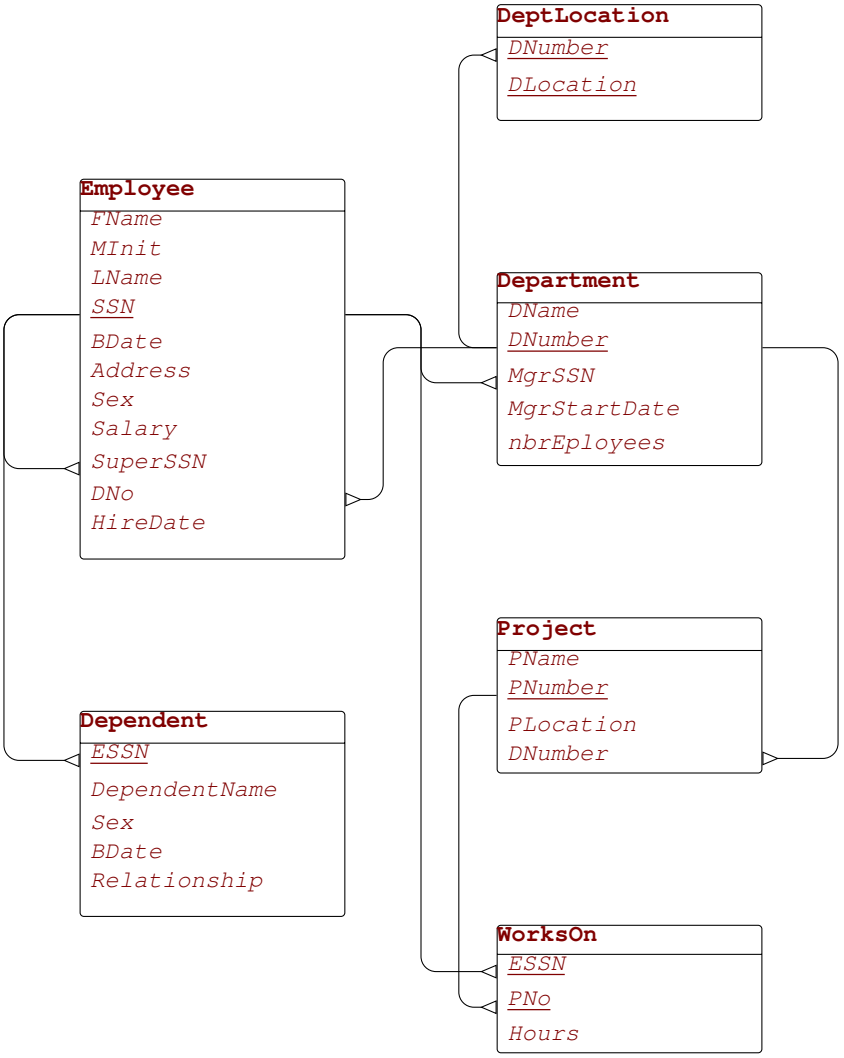
INFO-H-415 - Advanced Databases

Session 2+3 - Active Databases (2+3 of 3)

Consider the following database schema:



In SQL Server, enforce the following constraints using a set of `CHECK` constraints, referential integrity constraints, or triggers.

Exercise 1. The age of employees must be greater than 18.

Exercise 2. The supervisor of an employee must be older than the employee.

Exercise 3. The salary of an employee cannot be greater than the salary of his/her supervisor.

Exercise 4. The manager of a department must be an employee of that department.

Exercise 5. The location of a project must be one of the locations of its department.

Exercise 6. The hire date of employees must be greater than their birth date.

Exercise 7. A supervisor must be hired at least 1 year before every employee s/he supervises.

Exercise 8. The attribute Department.NbrEmployees is a derived attribute from Employee.DNo.

Exercise 9. An employee works at most in 4 projects.

Exercise 10. An employee works at least 30h/week and at most 50 h/week on all its projects.

Exercise 11. Among all employees working on a project, at most 2 can work for less than 10 hours.

Exercise 12. Only department managers can work less than 5 hours on a project.

Exercise 13. Employees that are not supervisors must work at least 10 hours on every project they work.

Exercise 14. The manager of a department must work at least 5 hours on all projects controlled by the department.

Exercise 15. The attribute Employee.SuperSSN is a derived attribute computed as follows. Department managers are supervised by the manager of Department 1 (Headquarters). Employees that are not managers are supervised by the manager of their department. Finally, the manager of Department 1 has a null value in attribute SuperSSN.

Exercise 16. The supervision relationship defined by Employee.SuperSSN must not be cyclic. (It is supposed that attribute Employee.SuperSSN is not derived as stated above.)

Details of the database for the exercises

Table creation script

```
create table Employee (
  FName varchar(15) not null,
  MInit char(1),
  LName varchar(15) not null,
  SSN char(9) not null,
  BDate smalldatetime null,
  Address varchar(30),
  Sex char(1),
  Salary decimal(18,2),
  SuperSSN char(9),
  DNo int not null,
  HireDate smalldatetime null,
  constraint PK_Employee primary key (SSN),
  constraint FK_Employee_Employee foreign key (SuperSSN) references Employee (SSN),
)
create table Department (
  DName varchar(15) not null,
  DNumber int not null,
  MgrSSN char(9) not null,
  MgrStartDate smalldatetime,
  nbrEmployees int,
  constraint PK_Department primary key (DNumber),
  constraint FK_Department_Employee foreign key (MgrSSN) references Employee (SSN)
    on delete cascade on update cascade
)
alter table Employee
  add constraint FK_Employee_Department foreign key (DNo) references Department (DNumber)
create table Project (
  PName varchar(15) not null,
  PNumber int not null,
  PLocation varchar(15),
  DNumber int not null,
  constraint PK_Project primary key (PNumber),
  constraint FK_Project_Department foreign key (DNumber) references Department (DNumber)
)
create table DeptLocations (
  DNumber int not null,
  DLocation varchar(15) not null,
  constraint PK_Dept_Locations primary key (DNumber,DLocation),
  constraint FK_Dept_Locations_Department foreign key (DNumber) references Department (DNumber)
)
create table Dependent (
  ESSN char(9) not null,
  DependentName varchar(15) not null,
  Sex char(1),
  BDate smalldatetime null,
  Relationship varchar(8),
  constraint PK_Dependent primary key (ESSN,DependentName),
  constraint FK_Dependent_Employee foreign key (ESSN) references Employee (SSN)
)
create table WorksOn (
  ESSN char(9) not null,
  PNo int not null,
  hours decimal(18,1) not null,
  constraint PK_WorksOn primary key (ESSN,PNo),
  constraint FK_WorksOn_Employee foreign key (ESSN) references Employee (SSN),
  constraint FK_WorksOn_Project foreign key (PNo) references Project (PNumber)
)
```

Initial data in the tables

Employee

FName	MInit	LName	SSN	BDate	Address	Sex	Salary	SuperSSN	DNo	HireDate
John	B	Smith	123456789	09-05-1955	731 Fondren, Houston, TX	M	30000	333445555	5	01-01-1985
Franklin	T	Wong	333445555	08-12-1945	638 Voss, Houston, TX	M	40000	888665555	5	01-01-1982
Alicia	J	Zelaya	999887777	19-07-1958	3321 Castle, Spring, TX	F	25000	987654321	4	01-01-1985
Jennifer	S	Wallace	987654321	20-06-1931	291 Berry, Bellaire, TX	F	43000	888665555	4	01-01-1982
Ramesh	K	Narayan	666884444	15-09-1952	975 Fire Oak, Humble, TX	M	38000	333445555	5	01-01-1985
Joyce	A	English	453453453	31-07-1962	5631 Rice, Houston, TX	F	25000	333445555	5	01-01-1985
Ahmad	V	Jabbar	987987987	29-03-1959	980 Dallas, Houston, TX	M	25000	987654321	4	01-01-1985
James	A	Borg	888665555	10-11-1927	450 Stone, Houston, TX	M	55000		1	01-01-1980

Department

DName	DNumber	MgrSSN	MgrStartDate	nbrEmployees
Research	5	333445555	22-05-1978	4
Administration	4	987654321	01-01-1985	3
Headquarters	1	888665555	19-06-1971	1

DeptLocations

DNumber	DLocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

Project

PName	PNumber	PLocation	DNumber
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

WorksOn

ESSN	PNo	Hours
123456789	1	32.5
123456789	2	7.5
333445555	1	10
333445555	2	10
333445555	3	20
453453453	1	20
453453453	2	20
666884444	3	40
888665555	20	30.0
987654321	10	5.0
987654321	20	15.0
987654321	30	20.0
987987987	10	35.0
987987987	30	5.0
999887777	10	10.0
999887777	30	30.0

Dependent

ESSN	DependentName	Sex	BDate	Relationship
333445555	Alice	F	05-04-1976	Daughter
333445555	Theodore	M	25-10-1973	Son
333445555	Joy	F	03-05-1948	Spouse
987654321	Abner	M	29-02-1932	Spouse
123456789	Michael	M	01-01-1978	Son
123456789	Alice	F	31-12-1978	Daughter
123456789	Elizabeth	F	05-05-1957	Spouse

Solutions for Session 2+3 - Active Databases (2+3 of 3)

➤ Solution to Exercise 1

"The age of employees must be greater than 18."

Using a CHECK constraint

```
alter table Employee
add constraint employee_Age18
check ( dateadd(year,18,BDate) <= getdate() )
```

Using a trigger

```
create trigger age18
on Employee
after insert, update
as
if exists (
    select *
      from Inserted
     where dateadd(year,18,BDate) > getdate() )
begin
    raiserror('Constraint Violation: The age of an employee
        must be greater than 18', 1, 1)
    rollback
end
```

➤ Solution to Exercise 2

"The supervisor of an employee must be older than the employee"

Using a trigger

```
create trigger supervisorAge
on Employee
after insert, update
as
if exists (
    select *
      from Inserted I,
           Employee E
     where ( I.SuperSSN = E.SSN and I.BDate < E.BDate )
        or ( E.SuperSSN = I.SSN and E.BDate < I.BDate ) )
begin
    raiserror( 'Constraint Violation:
        The age of an employee must be less than
        the age of his/her supervisor', 1, 1)
    rollback
end
```

➤ Solution to Exercise 3

"The salary of an employee cannot be greater than the salary of his/her supervisor."

Using a trigger

```
create trigger supervisorSalary
on Employee
after insert, update
as
if exists (
    select *
      from Inserted I,
           Employee E
     where ( I.SuperSSN = E.SSN and I.Salary > E.Salary )
        or ( E.SuperSSN = I.SSN and E.Salary > I.Salary ) )
begin
    raiserror('Constraint Violation:
        The salary of an employee cannot be greater than
        the salary of his/her supervisor', 1, 1)
    rollback
end
```

➤ Solution to Exercise 4

"The manager of a department must be an employee of that department."

Using `UNIQUE` and foreign key constraints

```
alter table Employee
add constraint UN_Employee_SSN_DNo
unique( SSN, DNO )

alter table Department
add constraint FK_Employee_SSN_DNo
foreign key( MgrSSN, DNumber )
references Employee( SSN, DNo )
```

➤ Solution to Exercise 5

"The location of a project must be one of the locations of its department."

Using a foreign key constraint

```
alter table Project
add constraint FK_Project_DeptLocations
foreign key( DNumber, PLocation )
references DeptLocations( DNumber, DLocation )
```

➤ Solution to Exercise 6

"The hire date of employees must be greater than their birth date."

Using a `CHECK` key constraint

```
alter table Employee
add constraint HireDate_BDate
check( HireDate > BDate )
```

➤ Solution to Exercise 7

"A supervisor must be hired at least 1 year before every employee s/he supervises."

Using a trigger

```
create trigger hireSuperv
on Employee
after insert, update
as
if exists (
    select *
      from Inserted I,
           Employee E
     where ( I.SuperSSN = E.SSN and datediff(year, E.HireDate, I.HireDate) < 1  )
        or ( E.SuperSSN = I.SSN and datediff(year, I.HireDate, E.HireDate) < 1  ) )
begin
    raiserror('Constraint Violation:
       A supervisor must be hired at least 1 year before
       every employee s/he supervises', 1, 1)
    rollback
end
```

➤ Solution to Exercise 8

"The attribute Department.NbrEmployees is a derived attribute from Employee.DNo"

Using value deriving triggers

```
create trigger DeptNbrEmp_Employee_InsUpdDel_Derive
on Employee
after insert, update, delete
as
begin
    update Department D
       set NbrEmployees = (
              select count(*)
                from Employee E
               where E.DNo = D.DNumber )
     where D.DNumber in (
              select distinct I.DNo
                from Inserted I )
        or D.DNumber in (
              select distinct D.DNo
                from Deleted D )
end
```

Incremental version

```
create trigger derived_Department_NbrEmployees_Employee
on Employee
after insert, update, delete
as
begin
    update Department
       set NbrEmployees = NbrEmployees +
              ( select count(*) from Inserted I where DNumber=I.DNo ) -
              ( select count(*) from Deleted D where DNumber=D.DNo )
     where DNumber in ( select DNo from Inserted )
        or DNumber in ( select DNo from Deleted )
end

create trigger derived_Department_NbrEmployees_Department
```

```
on Department
after insert, update
as
if exists ( select *
              from Inserted
             where NbrEmployees <> ( select count(*)
                                       from Employee E
                                      where E.DNo = DNumber ) )
begin
    raiserror('Constraint Violation:
        The attribute Department.NbrEmployees is a derived
        attribute from Employee.DNo', 1, 1)
    rollback
end
```

➤ Solution to Exercise 9

"An employee works at most in 4 projects"

Using a trigger

```
create trigger empNbrProj
on WorksOn
after insert, update
as
if exists (
    select *
      from WorksOn W
     group by W.ESSN
    having count(*) > 4 )
begin
    raiserror('Constraint Violation: An employee works at
        most in 4 projects', 1, 1)
    rollback
end
```

➤ Solution to Exercise 10

"An employee works at least 30h/week and at most 50 h/week on all its projects"

Using a trigger

```
create trigger workson_30_50
on WorksOn
after insert, update, delete
as
if exists ( select *
              from WorksOn
             group by ESSN
            having ( sum(Hours) < 30 )
                or ( sum(Hours) > 50 ) )
begin
    raiserror('Constraint Violation: An employee works at
        least 30 h/week and at most 50 h/week on all its projects', 1, 1)
    rollback
end
```

➤ Solution to Exercise 11

"A project can have at most 2 employees working on the project less than 10 hours"

Using a trigger

```
create trigger worksonLess10h
on WorksOn
after insert, update
as
if exists ( select *
              from WorksOn
             where Hours < 10
             group by PNo
            having count(*) > 2 )
begin
    raiserror('Constraint Violation: A project can have at
        most 2 employees working on the project less than 10 hours', 1, 1)
rollback end
```

➤ Solution to Exercise 12

"Only department managers can work less than 5 hours on a project"

Using a set of triggers

```
create trigger worksonLess5h_WorksOn
on WorksOn
after insert, update
as
if exists ( select *
              from Inserted
             where Hours < 5
               and ESSN not in (
                           select MgrSSN
                             from Department
                            where MgrSSN is not null ) )
begin
    raiserror('Constraint Violation: Only department managers
        can work less than 5 hours on a project', 1, 1)
    rollback
end

create trigger worksonLess5h_Department
on Department
after update, delete
as
if exists ( select *
              from Deleted
             where MgrSSN not in (
                           select MgrSSN
                             from Department )
               and MgrSSN in (
                           select ESSN
                           from WorksOn
                           where Hours < 5 ) )
begin
    raiserror('Constraint Violation: Only department managers
        can work less than 5 hours on a project', 1, 1)
    rollback
end
```

➤ Solution to Exercise 13

"Employees that are not supervisors must work at least 10 hours on every project they work"

Using a set of triggers

```
create trigger workson10h_WorksOn
on WorksOn
after insert, update
as
if exists ( select *
              from Inserted
             where Hours < 10
               and ESSN not in (
                           select SuperSSN
                             from Employee
                            where SuperSSN is not null ) )
begin
    raiserror('Constraint Violation: Employees that are not supervisors
        must work at least 10 hours on every project they work', 1, 1)
    rollback
end

create trigger workson10h_Employee
on Employee
after update, delete
as
if exists ( select *
              from Deleted
             where SuperSSN not in (
                             select SuperSSN
                               from Employee
                              where SuperSSN is not null )
               and SuperSSN in (
                             select ESSN
                               from WorksOn
                              where Hours < 10 ) )
begin
    raiserror('Constraint Violation: Employees that are not supervisors
        must work at least 10 hours on every project they work', 1, 1)
    rollback
end
```

Note about the second trigger. The logic behind that trigger is the following: A supervisor is a supervisor as long as there is at least one employee he supervises... When an employee is updated (changes the supervisor) or is deleted, there is a chance for a supervisor to loose his status and to become a "regular" employee again. When this happens, that former supervisor has to work at least 10 hours on every project.

Therefore, the query in the above trigger looks for an employee who had a supervisor (since it was a SuperSSN entry for at least one employee in the Deleted table), that is not supervisor anymore (since it is not in the set of current supervisors) and belongs to the set of employees that work less than 10 hours on a project.

➤ Solution to Exercise 14

"The manager of a department must work at least 5 hours on all projects controlled by the department."

Using a set of triggers

```
create trigger mgrProj_Department
on Department
after insert, update
as
if exists ( select *
              from ( Inserted I join Project P on I.DNumber = P.DNumber )
                   left outer join WorksOn on MgrSSN = ESSN and PNumber = PNo
             where Hours is null
                or Hours < 5 )
begin
    raiserror('Constraint Violation: A manager must work at least 5 hours
        on all projects controlled by his/her department', 1, 1)
    rollback
end
```

```
create trigger mgrProj_Project
on Project
after insert, update
as
if exists ( select *
              from ( Project P join Department D on D.DNumber = P.DNumber )
                   left outer join WorksOn on MgrSSN = ESSN and PNumber = PNo
             where P.PNumber in ( select PNumber
                                  from Inserted )
               and ( Hours is null
                     or Hours < 5 ) )
begin
    raiserror('Constraint Violation: A manager must work at least 5 hours
        on all projects controlled by his/her department', 1, 1)
    rollback
end

create trigger mgrProj_WorksOn
on WorksOn
after update, delete
as
if exists ( select *
              from ( Department D join Project P on D.DNumber=P.DNumber)
                   left outer join WorksOn on MgrSSN = ESSN and PNumber = PNo
             where D.MgrSSN in ( select ESSN
                                 from Deleted )
               and ( Hours is null
                     or Hours < 5 ) )
begin
    raiserror('Constraint Violation: A manager must work at least 5 hours
        on all projects controlled by his/her department', 1, 1)
    rollback
end
```

➤ Solution to Exercise 15

"The attribute **Employee.** *SuperSSN is a derived attribute computed as follows. Department managers are supervised by the manager of Department 1 (Headquarters). Employees that are not managers are supervised by the manager of their department. Finally, the manager of Department 1 has a* NULL *value in attribute* SuperSSN*."*

Using a set of triggers

```
create trigger derived_Employee_SuperSSN_Department
on Department
after insert, update
as
if update(MgrSSN)
begin
    update Employee
       set SuperSSN = (
                select case when SSN != D.MgrSSN
                                then D.MgrSSN
                            when SSN = D.MgrSSN and DNo != 1
                                then ( select MgrSSN
                                         from Department
                                        where DNumber = 1 )
                            else
                                null
                            end
                  from Department D
                 where DNo = D.DNumber )
    -- if the department manager changes all employees of the department
    -- must be updated
     where ( DNo in (
                select DNumber
                  from Inserted ) )
    -- if the manager of department 1 changes, all department managers
    -- must be updated
        or ( 1 in (
                select DNumber
                from Inserted )
        and SSN in (
                select MgrSSN
                from Department ) )
end

create trigger derived_Employee_SuperSSN_Employee
on Employee
after insert, update
as
if update(DNo)
begin
    update Employee
       set SuperSSN = (
                    select case when SSN != MgrSSN
                                    then D.MgrSSN
                                when SSN = MgrSSN and I.DNo != 1
                                    then ( select MgrSSN
                                             from Department
                                            where DNumber = 1 )
                                else
                                    null
                                end
                      from Inserted I,
                           Department D
                     where SSN = I.SSN
                       and I.DNo = D.DNumber )
     where SSN in (
                select SSN
```

```
                from Inserted )
end
```

➤ Solution to Exercise 16

"The supervision relationship in ***Employee****.SuperSSN must not be cyclic"*

Using a trigger

```
create trigger noncyclic_subordinates
on Employee
after insert, update
as
begin
    create table #Supervision (
        SSN char(9),
        SuperSSN char(9)
        primary key (SSN,SuperSSN) )
    insert into #Supervision
        select SSN, SuperSSN
          from Employee
         where SuperSSN is not null
    while @@rowcount != 0 -- while previous operation affected some rows
    begin
        if exists ( select *
                      from #Supervision
                     where SSN = SuperSSN )
        begin
            raiserror('Constraint Violation: The supervision
                relationship is cyclic', 1, 1)
            rollback
        end
        insert into #Supervision
            select distinct S1.SSN, S2.SuperSSN
              from #Supervision S1 join #Supervision S2
                on S1.SuperSSN = S2.SSN
             where not exists (
                          select *
                            from #Supervision S
                           where S.SSN = S1.SSN
                             and S.SuperSSN = S2.SuperSSN )
    end
end
```