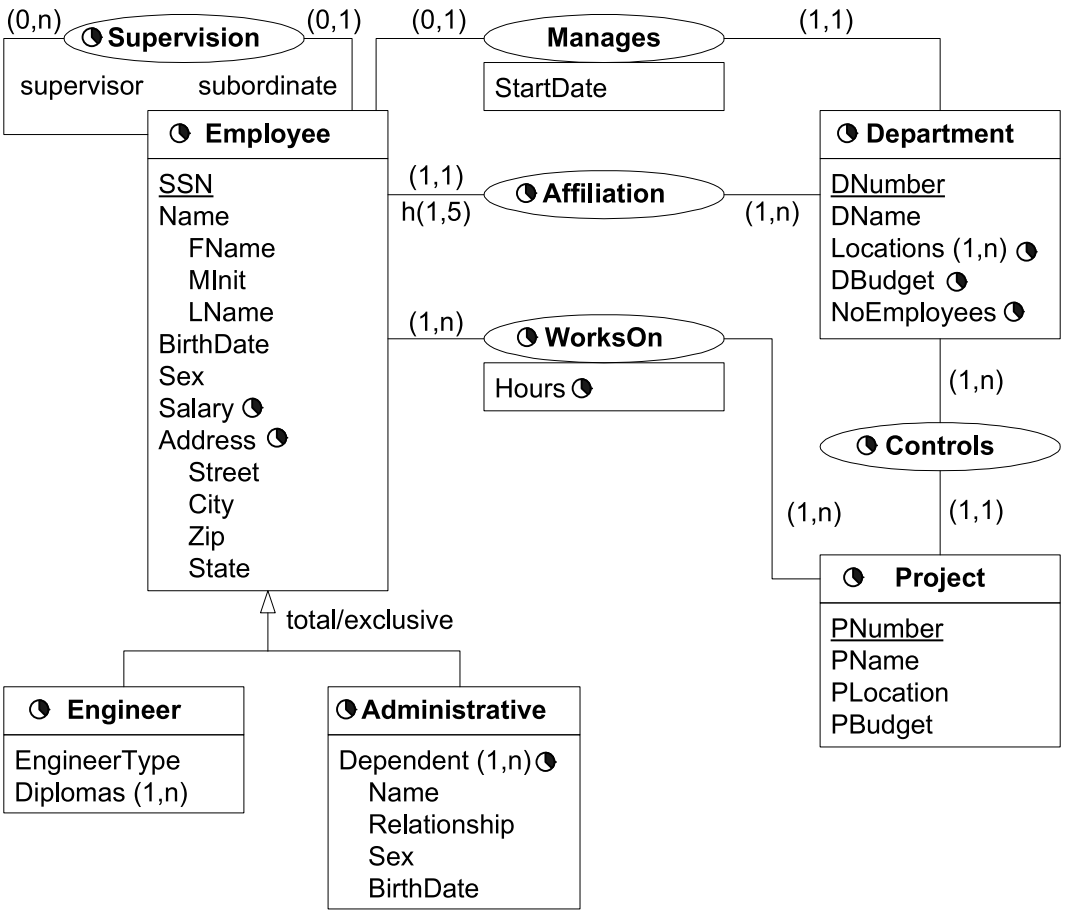
SQL Server
Temporal Databases: Constraints



Consider the above temporal conceptual schema.
Ensure the following integrity constraints.

1. An employee works in at most one department at any point in time.
2. At any point in time an employee cannot work more than once in a project.
3. The lifecycle of affiliation is included in the lifecycle of employee
4. The lifecycle of an employee is equal to the union of his/her affiliations.
5. Employees have a contiguous lifecycle.
6. The lifecycle of an employee is equal to the union of his/her affiliations, now taking into account that the lifecycle of employees is contiguous.

Constraints

1. An employee works in at most one department at any point in time.
   In other terms SSN is a sequenced primary key for Affiliation.

```
create trigger Seq_PK_Affiliation on Affiliation
  after insert, update  as
if exists ( select * from Inserted A1
  where 1 < ( select count(*) from Affiliation A2
    where A1.SSN = A2.SSN
    and A1.FromDate < A2.ToDate and A2.FromDate < A1.ToDate ) )
begin
  raiserror 13000
    'An employee works in at most one department at any point in time'
  rollback transaction
end
```

2. At any point in time an employee cannot work more than once in a project.
   In other terms (SSN,PNumber) is a sequenced primary key for WorksOn

```
create trigger Seq_PK_WorksOn on WorksOn
  after insert, update as
if exists ( select * from Inserted W1
  where 1 < ( select count(*) from WorksOn W2
    where W1.SSN = W2.SSN and W1.PNumber = W2.PNumber
    and W1.FromDate < W2.ToDate and W2.FromDate < W1.ToDate ) )
begin
  raiserror 13000
    'At any point in time an employee cannot work more than once in a project'
  rollback transaction
end
```

3. The lifecycle of affiliation is included in the lifecycle of employee.
   In the following triggers it is assumed that the table EmployeeLifecycle is coalesced.
   Therefore, every line in Affiliation must be covered by one line in EmployeeLifecycle.

```
create trigger Seq_FK_Affiliation_EmployeeLifecycle_1 on Affiliation
  after insert, update as
if exists ( select * from Inserted A
  where not exists ( select * from EmployeeLifecycle E
    where A.SSN = E.SSN
    and E.FromDate <= A.FromDate and A.ToDate <= E.ToDate ) )
begin
  raiserror 13000
    'The lifecycle of affiliation must be included in the lifecycle of employee'
  rollback transaction
end

create trigger Seq_FK_Affiliation_EmployeeLifecycle_2 on EmployeeLifecycle
  after update, delete as
if exists ( select * from Affiliation A
  where A.SSN IN ( select SSN from Deleted)
    and not exists ( select * from EmployeeLifecycle E
    where A.SSN = E.SSN
    and E.FromDate <= A.FromDate and A.ToDate <= E.ToDate ) )
begin
  raiserror 13000
    'The lifecycle of affiliation must be included in the lifecycle of employee'
  rollback transaction
end
```

4. The lifecycle of an employee is equal to the union of his/her affiliations. It is supposed that the previous trigger is activated, therefore it is sufficient to monitor that an employee must be affiliated to a department throughout his/her lifecycle.

```
create trigger Seq_FK_EmployeeLifecycle_Affiliation_1 on Affiliation
  after update, delete as
if exists ( select * from EmployeeLifecycle E
  where E.SSN in ( select SSN from Deleted )
  and not exists ( select * from Affiliation A
    where E.SSN = A.SSN
    and A.FromDate <= E.FromDate and E.FromDate < A.ToDate )
  or not exists ( select * from Affiliation A
    where E.SSN = A.SSN
    and A.FromDate < E.ToDate and E.ToDate <= A.ToDate )
  or exists ( select * from Affiliation A
    where E.SSN = A.SSN
    and E.FromDate < A.ToDate and A.ToDate < E.ToDate
    and not exists ( select * from Affiliation A2
      where A2.SSN = A.SSN
      and A2.FromDate <= A.ToDate and A.ToDate < A2.ToDate ) ) )
begin
  raiserror 13000
    'An employee must be affiliated to a department throughout his/her lifecycle'
  rollback transaction
end

create trigger Seq_FK_EmployeeLifecycle_Affiliation_2 on EmployeeLifecycle
  after insert, update as
if exists ( select * from Inserted E
  where not exists ( select * from Affiliation A
    where E.SSN = A.SSN
    and A.FromDate <= E.FromDate and E.FromDate < A.ToDate )
  or not exists ( select * from Affiliation A
    where E.SSN = A.SSN
    and A.FromDate < E.ToDate and E.ToDate <= A.ToDate )
  or exists ( select * from Affiliation A
    where E.SSN = A.SSN
    and E.FromDate < A.ToDate and A.ToDate < E.ToDate
    and not exists ( select * from Affiliation A2
      where A2.SSN = A.SSN
      and A2.FromDate <= A.ToDate and A.ToDate < A2.ToDate ) ) )
begin
  raiserror 13000
    'An employee must be affiliated to a department throughout his/her lifecycle'
  rollback transaction
end
```

5. Employees have a contiguous lifecycle.

```
alter table EmployeeLifecycle
drop constraint PK_EmployeeLifecycle
alter table EmployeeLifecycle
add constraint PK_EmployeeLifecycle primary key (SSN)
```

6. The lifecycle of an employee is equal to the union of his/her affiliations, now taking into account that the lifecycle of employees is contiguous.
   It is necessary to ensure that (1) the affiliations of an employee define a contiguous history, and (2) an employee must be affiliated to a department throughout his/her lifecycle.

The following trigger ensures that the affiliations of an employee define a contiguous history.

```
create trigger Contiguous_Hist_Affiliation on Affiliation
  after insert, update, delete as
if exists ( select * from Affiliation A1, Affiliation A2
  where A1.SSN = A2.SSN and A1.ToDate < A2.FromDate
  and not exists ( select * from Affiliation A3
     where A1.SSN = A3.SSN
     and ( ( A3.FromDate <= A1.ToDate and A1.ToDate < A3.ToDate )
        or ( A3.FromDate < A2.FromDate and A2.FromDate <= A3.ToDate ) ) ) )
begin
  raiserror 13000
    'The affiliations of an employee define a contiguous history'
  rollback transaction
end
```

The following two triggers replaces those of question (4).

```
alter trigger Seq_FK_EmployeeLifecycle_Affiliation_1 on Affiliation
  after update, delete as
if exists ( select * from EmployeeLifecycle E
  where E.SSN in ( select SSN from Deleted )
  and not exists ( select * from Affiliation A
    where E.SSN = A.SSN
    and A.FromDate <= E.FromDate and E.FromDate < A.ToDate )
  or not exists ( select * from Affiliation A
    where E.SSN = A.SSN
    and A.FromDate < E.ToDate and E.ToDate <= A.ToDate ) )
begin
  raiserror 13000
    'An employee must be affiliated to a department throughout his/her lifecycle'
  rollback transaction
end

alter trigger Seq_FK_EmployeeLifecycle_Affiliation_2 on EmployeeLifecycle
  after insert, update  as
if exists ( select * from Inserted E
  where not exists ( select * from Affiliation A
    where E.SSN = A.SSN
    and A.FromDate <= E.FromDate and E.FromDate < A.ToDate )
  or not exists ( select * from Affiliation A
    where E.SSN = A.SSN
    and A.FromDate < E.ToDate and E.ToDate <= A.ToDate ) )
begin
  raiserror 13000
    'An employee must be affiliated to a department throughout his/her lifecycle'
  rollback transaction
end
```