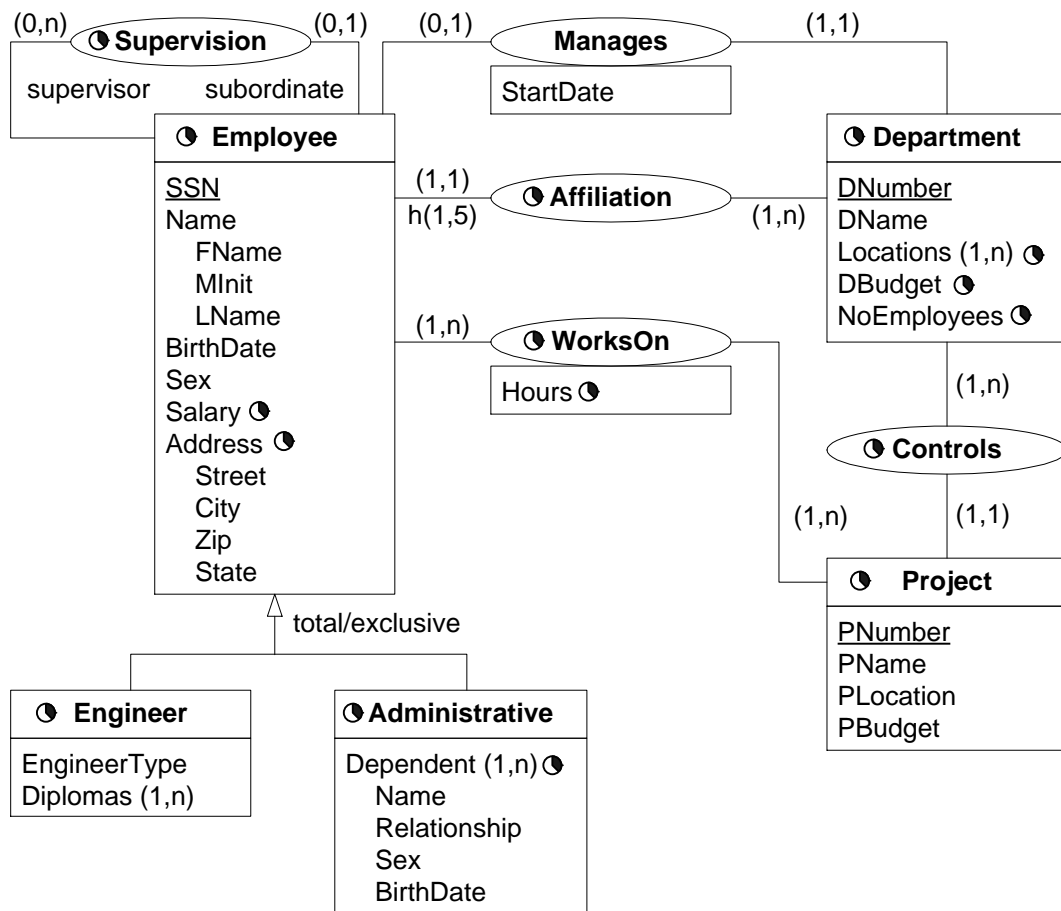


MobilityDB

Temporal Databases



Consider the above temporal conceptual schema

- Define a relational schema corresponding to the conceptual schema.
- Write the following queries in SQL:
 1. Name of managers currently living in Houston
 2. Name of employees currently working in the 'Research' department who currently have a salary greater or equal than 45000
 3. Name of current employees who do not currently work in any department
 4. Name of employees who had the highest salary in 2002-01-01
 5. Joint evolution of salary and affiliation of employees
 6. Name of employees and periods when they were supervisors but did not work in any project during the same period
 7. Name of supervisors who ever worked on a project
 8. Name of employees and dates when they changed their affiliation
 9. Name of employees and periods when they worked on any project
 10. Evolution of the maximum salary
 11. Evolution of the maximum salary by department
 12. Evolution of the number of projects by department
 13. Name of employees and periods when they worked on all projects of their department

Relational Schema

- Employee(SSN, FName, MInit, LName, BirthDate, Sex, Lifespan)

- EmpSalary(SSN, Salary, VT)

SSN references Employee(SSN)

- EmpAddress(SSN, Street, City, Zip, Country, VT)

SSN references Employee(SSN)

- Engineer(SSN, EngineerType, Lifespan)

SSN references Employee(SSN)

In this table the lifespan of Engineer is kept as well as the attribute EngineerType.

- EngineerDiplomas(SSN, Diploma)

SSN references Engineer(SSN)

- AdminLifespan(SSN, Lifespan)

SSN references Employee(SSN)

- AdminDependent(SSN, Name, Relationship, Sex, BirthDate, VT)

SSN references AdminLifespan(SSN)

It is supposed that an employee does not have two dependents of the same name and the same relationship. An alternative will be to put BirthDate instead of Relationship as part of the key.

- Supervision(SSN, SuperSSN, VT)

SSN references Employee(SSN)

SuperSSN references Employee(SSN)

- Affiliation(SSN, DNumber, VT)

SSN references Employee(SSN)

- Department(DNumber, DName, MgrSSN, MgrStartDate, Lifespan)

MgrSSN references Employee(SSN)

It is supposed that the lifespan of departments is continuous. In this case an additional table for the lifespan is not necessary.

- DeptLocations(DNumber, Location, VT)

DNumber references Department(DNumber)

- DepartmentBudget(DNumber, DBudget, VT)

DNumber references Department(DNumber)

- DepartmentNbEmp(DNumber, NbEmp, VT)

DNumber references Department(DNumber)

- Project(PNumber, PName, PLocation, PBudget, VT)

It is supposed that the lifespan of Project is continuous.

- Controls(PNumber, DNumber, VT)

PNumber references Project(PNumber)

DNumber references Department(DNumber)

- WorksOn(SSN, PNumber, Hours, VT)

PNumber references Project(PNumber)

SSN references Employee(SSN)

It is supposed that the temporality of attribute Hours is the same as the lifespan of the association. In this case two different tables are not necessary. To obtain the lifespan of the association independently of the attribute hours a temporal projection is needed.

Example Database

Partial schema where not all entities and attributes are taken into account.

Employee

SSN	FName	MInit	LName	BirthDate	Sex	Lifespan
123456789	John	B	Smith	1955-05-09	M	{[1985-01-01, 9999-12-31]}
333445555	Franklin	T	Wong	1945-12-08	M	{[1982-01-01, 9999-12-31]}
999887777	Alicia	J	Zelaya	1958-07-19	F	{[1985-01-01, 9999-12-31]}
987654321	Jennifer	S	Wallace	1931-06-20	F	{[1982-01-01, 9999-12-31]}
666884444	Ramesh	K	Narayan	1952-09-15	M	{[1985-01-01, 9999-12-31]}
453453453	Joyce	A	English	1962-07-31	F	{[1985-01-01, 9999-12-31]}
987987987	Ahmad	V	Jabbar	1959-03-29	M	{[1985-01-01, 9999-12-31]}
888665555	James	A	Borg	1927-11-10	M	{[1980-01-01, 9999-12-31]}

EmpSalary

SSN	Salary	VT
123456789	30000	{[1985-01-01, 9999-12-31]}
333445555	40000	{[1982-01-01, 1983-01-01]}
333445555	45000	{[1983-01-01, 9999-12-31]}
999887777	25000	{[1985-01-01, 9999-12-31]}
987654321	43000	{[1982-01-01, 9999-12-31]}
666884444	38000	{[1985-01-01, 9999-12-31]}
453453453	25000	{[1985-01-01, 9999-12-31]}
987987987	25000	{[1985-01-01, 9999-12-31]}
888665555	55000	{[1980-01-01, 1981-01-01]}
888665555	58000	{[1981-01-01, 9999-12-31]}

EmpAddress

SSN	Street	City	Zip	State	VT
123456789	731 Fondren	Houston	1000	TX	{[1985-01-01, 9999-12-31]}
333445555	638 Voss	Houston	1000	TX	{[1982-01-01, 9999-12-31]}
999887777	3321 Castle	Spring	1000	TX	{[1985-01-01, 9999-12-31]}
987654321	291 Berry	Bellaire	1000	TX	{[1982-01-01, 9999-12-31]}
666884444	975 Fire Oak	Humble	1000	TX	{[1985-01-01, 9999-12-31]}
453453453	5631 Rice	Houston	1000	TX	{[1985-01-01, 9999-12-31]}
987987987	980 Dallas	Houston	1000	TX	{[1985-01-01, 9999-12-31]}
888665555	450 Stone	Houston	1000	TX	{[1980-01-01, 9999-12-31]}

Supervision

SSN	SuperSSN	VT
123456789	333445555	{[1985-01-01, 9999-12-31]}
333445555	888665555	{[1982-01-01, 9999-12-31]}
999887777	987654321	{[1985-01-01, 9999-12-31]}
987654321	888665555	{[1982-01-01, 9999-12-31]}
666884444	333445555	{[1985-01-01, 9999-12-31]}
453453453	333445555	{[1985-01-01, 9999-12-31]}
987987987	987654321	{[1985-01-01, 9999-12-31]}

Affiliation

SSN	DNumber	VT
123456789	1	{[1985-01-01, 1986-01-01]}
123456789	5	{[1986-01-01, 9999-12-31]}
333445555	4	{[1982-01-01, 1984-01-01]}
333445555	5	{[1984-01-01, 9999-12-31]}
999887777	4	{[1985-01-01, 9999-12-31]}
987654321	4	{[1982-01-01, 9999-12-31]}
666884444	5	{[1985-01-01, 9999-12-31]}
453453453	5	{[1985-01-01, 9999-12-31]}
987987987	4	{[1985-01-01, 9999-12-31]}
888665555	1	{[1980-01-01, 9999-12-31]}

Department

<u>DNumber</u>	<u>DName</u>	<u>MgrSSN</u>	<u>MgrStartDate</u>	<u>Lifespan</u>
1	Headquarters	888665555	1980-06-19	{[1980-01-01, 9999-12-31]}
4	Administration	987654321	1982-01-01	{[1981-01-01, 9999-12-31]}
5	Research	333445555	1984-05-22	{[1982-01-01, 9999-12-31]}

DepartmentNbEmp

<u>DNumber</u>	<u>NbEmp</u>	<u>VT</u>
1	1	{[1980-01-01, 9999-12-31]}
4	3	{[1980-01-01, 9999-12-31]}
5	4	{[1980-01-01, 9999-12-31]}

DeptLocations

<u>DNumber</u>	<u>DLocation</u>	<u>VT</u>
1	Houston	{[1980-01-01, 9999-12-31]}
4	Stafford	{[1980-01-01, 9999-12-31]}
5	Bellaire	{[1980-01-01, 9999-12-31]}
5	Sugarland	{[1980-01-01, 9999-12-31]}
5	Houston	{[1980-01-01, 9999-12-31]}

Project

<u>PNumber</u>	<u>PName</u>	<u>PLocation</u>	<u>Lifespan</u>
1	ProductX	Bellaire	{[1980-01-01, 9999-12-31]}
2	ProductY	Sugarland	{[1980-01-01, 9999-12-31]}
3	ProductZ	Houston	{[1980-01-01, 9999-12-31]}
10	Computerization	Stafford	{[1980-01-01, 9999-12-31]}
20	Reorganization	Houston	{[1980-01-01, 9999-12-31]}
30	Newbenefits	Stafford	{[1980-01-01, 9999-12-31]}

Controls

<u>PNumber</u>	<u>DNumber</u>	<u>VT</u>
1	5	{[1980-01-01, 9999-12-31]}
2	5	{[1980-01-01, 9999-12-31]}
3	5	{[1980-01-01, 9999-12-31]}
10	4	{[1980-01-01, 9999-12-31]}
20	1	{[1980-01-01, 9999-12-31]}
30	4	{[1980-01-01, 9999-12-31]}

WorksOn

<u>SSN</u>	<u>PNumber</u>	<u>Hours</u>	<u>VT</u>
123456789	1	32.5	{[1985-01-01, 9999-12-31]}
123456789	2	7.5	{[1985-01-01, 9999-12-31]}
333445555	1	10	{[1982-01-01, 2000-01-01]}
333445555	2	10	{[1982-01-01, 2002-01-01]}
333445555	3	20	{[2005-01-01, 9999-12-31]}
453453453	1	20	{[1985-01-01, 9999-12-31]}
453453453	2	20	{[1985-01-01, 9999-12-31]}
666884444	3	40	{[1985-01-01, 9999-12-31]}
888665555	20	30.0	{[1983-01-01, 9999-12-31]}
987654321	10	5.0	{[1982-01-01, 2000-01-01]}
987654321	20	15.0	{[1982-01-01, 2001-01-01]}
987654321	30	20.0	{[1982-01-01, 2002-01-01]}
987987987	10	35.0	{[1985-01-01, 9999-12-31]}
987987987	30	5.0	{[1985-01-01, 9999-12-31]}
999887777	10	10.0	{[1985-01-01, 9999-12-31]}
999887777	30	30.0	{[1985-01-01, 9999-12-31]}

Queries

1. Name of managers currently living in Houston

```
SELECT e.FName, e.LName
FROM Employee e, EmpAddress a, Department d
WHERE e.SSN = a.SSN and e.SSN = d.MgrSSN AND a.City = 'Houston' AND
a.VT @> CURRENT_DATE AND d.VT @> CURRENT_DATE;
```

2. Name of employees currently working in the 'Research' department who currently have a salary greater or equal than 45000

```
SELECT e.FName, e.LName
FROM Employee e, EmpSalary s, Affiliation a, Department d
WHERE e.SSN = s.SSN AND e.SSN = a.SSN AND a.DNumber = d.DNumber AND
d.DName = 'Research' AND s.Salary >= 45000 AND
s.VT @> CURRENT_DATE AND a.VT @> CURRENT_DATE;
```

3. Name of current employees who do not currently work in any department

```
SELECT e.FName, e.LName
FROM Employee e
WHERE e.Lifespan @> CURRENT_DATE AND NOT EXISTS (
SELECT * FROM Affiliation a
WHERE e.SSN = a.SSN AND a.VT @> CURRENT_DATE );
```

4. Name of employees who had the highest salary in 2002-01-01

```
SELECT e.FName, e.LName
FROM Employee e, EmpSalary s
WHERE e.SSN = s.SSN AND s.VT @> DATE '2002-01-01' AND salary = (
SELECT MAX(salary)
FROM EmpSalary
WHERE VT @> DATE '2002-01-01' );
```

5. Joint evolution of salary and affiliation of employees

```
SELECT e.FName, e.LName, d.DName, s.Salary, s.VT * a.VT AS VT
FROM Employee e, EmpSalary s, Affiliation a, Department d
WHERE e.SSN = s.SSN AND e.SSN = a.SSN AND a.DNumber = d.DNumber AND
s.VT * a.VT IS NOT NULL
ORDER BY e.FName, e.LName, s.VT * a.VT;
```

6. Name of employees and periods when they were supervisors but did not work in any project during the same period

```
WITH SupNoProj AS (
SELECT e.FName, e.LName, s.VT - w.VT AS VT
FROM Employee e, Supervision s, WorksOn w
WHERE e.SSN = s.SuperSSN AND e.SSN = w.SSN
UNION
SELECT e.FName, e.LName, s.VT
FROM Employee e, Supervision s
WHERE e.SSN = s.SuperSSN AND NOT EXISTS (
SELECT * FROM WorksOn w
WHERE e.SSN = w.SSN ) )
SELECT FName, LName, spanUnion(VT)
FROM SupNoProj
GROUP BY FName, LName
ORDER BY FName, LName;
```

7. Name of supervisors who ever worked on a project

```
SELECT DISTINCT e.FName, e.LName
FROM Employee e, Supervision s, WorksOn w
WHERE e.SSN = s.SuperSSN and e.SSN = w.SSN;
```

8. Name of employees and dates when they changed their affiliation

```
SELECT e.FName, e.LName, setUnion(dates(a1.VT) * dates(a2.VT))
FROM Employee e, Affiliation a1, Affiliation a2
WHERE E.SSN = a1.SSN and e.SSN = a2.SSN AND a1.DNumber <> a2.DNumber
GROUP BY e.SSN, e.FName, e.LName
ORDER BY e.SSN;
```

9. Name of employees and periods when they worked on any project

```
SELECT e.FName, e.LName, spanUnion(w.VT)
FROM Employee e, WorksOn w
WHERE e.SSN = w.SSN
GROUP BY e.SSN, e.FName, e.LName
ORDER BY e.SSN;
```

10. Evolution of the maximum salary

```
SELECT tMax(tint(Salary, tstzspanset(VT)))
FROM EmpSalary s;
/* {[55000@1980-01-01 00:00:00+01, 58000@1981-01-01 00:00:00+01,
58000@9999-12-31 00:00:00+01]} */
```

Alternative version where the answer is expressed using tuple timestamping.

```
SELECT (rec).value AS Salary, datespanset((rec).time) AS VT
FROM (
  SELECT unnest(tMax(tint(Salary, tstzspanset(VT)))) AS rec
  FROM EmpSalary s ) AS t
ORDER BY VT;
/* 55000 | {[1980-01-01, 1981-01-01]}
58000 | {[1981-01-01, 9999-12-31]} */
```

11. Evolution of the maximum salary by department

```
SELECT a.DNumber, tMax(tint(Salary, tstzspanset(a.VT * s.VT)))
FROM Affiliation a, EmpSalary s
WHERE a.SSN = s.SSN
GROUP BY a.DNumber
ORDER BY a.DNumber;
/* 1 | {[55000@1980-01-01 00:00:00+01, 58000@1981-01-01 00:00:00+01,
58000@9999-12-31 00:00:00+01]}
4 | {[43000@1982-01-01 00:00:00+01, 45000@1983-01-01 00:00:00+01,
43000@1984-01-01 00:00:00+01, 43000@9999-12-31 00:00:00+01]}
5 | {[45000@1984-01-01 00:00:00+01, 45000@9999-12-31 00:00:00+01]} */
```

Alternative version where the answer is expressed using tuple timestamping.

```
SELECT DNumber, (rec).value AS Salary, datespanset((rec).time) AS VT
FROM (
  SELECT a.DNumber, unnest(tMax(tint(Salary, tstzspanset(a.VT * s.VT)))) AS rec
  FROM Affiliation a, EmpSalary s
  WHERE a.SSN = s.SSN
  GROUP BY a.DNumber ) AS t
ORDER BY DNumber, VT;
/* 1 | 55000 | {[1980-01-01, 1981-01-01]}
1 | 58000 | {[1981-01-01, 9999-12-31]}
4 | 43000 | {[1982-01-01, 1983-01-01), [1984-01-01, 9999-12-31]}
```

```

4 | 45000 | {[1983-01-01, 1984-01-01]}
5 | 45000 | {[1984-01-01, 9999-12-31]} */

```

12. Evolution of the number of projects by department

```

SELECT c.DNumber, tCount(tstzspanset(c.VT))
FROM Controls c
GROUP BY c.DNumber
ORDER BY c.DNumber;
/* 1 | {[1@1980-01-01 00:00:00+01, 1@9999-12-31 00:00:00+01]}
   4 | {[2@1980-01-01 00:00:00+01, 2@9999-12-31 00:00:00+01]}
   5 | {[3@1980-01-01 00:00:00+01, 3@9999-12-31 00:00:00+01]} */

```

Alternative version where the answer is expressed using tuple timestamping.

```

SELECT DNumber, (rec).value AS Count, datespanset((rec).time) AS VT
FROM (
  SELECT c.DNumber, unnest(tCount(tstzspanset(c.VT))) AS rec
  FROM Controls c
  GROUP BY c.DNumber
  ORDER BY c.DNumber ) AS t
ORDER BY DNumber, VT;
/* 1 | 1 | {[1980-01-01, 9999-12-31]}
   4 | 2 | {[1980-01-01, 9999-12-31]}
   5 | 3 | {[1980-01-01, 9999-12-31]} */

```

13. Name of employees and periods when they worked on all projects of their department

```

/* Temporal join of Affiliation and Controls
   Each period in the result should be fully covered by a tuple in WorksOn */

```

```

WITH AffCont AS (
  SELECT a.SSN, a.DNumber, c.PNumber, a.VT * c.VT AS VT
  FROM Affiliation a, Controls c
  WHERE a.DNumber = c.DNumber ),
/* Temporal difference of AffCont and WorksOn
   These are the missing periods */
MissingPeriods AS (
  SELECT SSN, DNumber, spanUnion(VT) AS VT
  FROM (
    SELECT c.SSN, c.DNumber, c.VT - w.VT AS VT
    FROM AffCont c, WorksOn w
    WHERE c.SSN = w.SSN AND c.PNumber = w.PNumber AND
           c.VT - w.VT IS NOT NULL
    UNION
    SELECT c.SSN, c.DNumber, c.VT
    FROM AffCont c WHERE NOT EXISTS (
      SELECT * FROM WorksOn w
      WHERE c.SSN = w.SSN AND c.PNumber = w.PNumber ) )
  GROUP BY SSN, DNumber ),

```

```

-- Temporal Difference of Affiliation and MissingPeriods

```

```

AffMinusMissing AS (
  SELECT a.SSN, p.DNumber, a.VT - p.VT AS VT
  FROM Affiliation a, MissingPeriods p
  WHERE a.SSN = p.SSN AND a.DNumber = p.DNumber
        AND a.VT - p.VT IS NOT NULL
  UNION
  SELECT a.SSN, a.DNumber, a.VT
  FROM Affiliation a
  WHERE NOT EXISTS (

```

```
SELECT *
FROM MissingPeriods p
WHERE a.SSN = p.SSN AND a.DNumber = p.DNumber ) )
SELECT FName, LName, spanUnion(VT)
FROM Employee e, AffMinusMissing a WHERE e.SSN = a.SSN
GROUP BY e.SSN, FName, LName
ORDER BY FName, LName;
```