Chapter 10 Advanced topics in relational databases

- Security and user authorization in SQL
- Recursion in SQL
- Object-relational model

1. User-defined types in SQL
2. Operations on object-relational data

- Online analytic processing & data cubes

Question?

- EDB: Par(c,p) = *p* is a parent of *c*.
- We want to find generalized cousins: people with common ancestors one or more generations back.

Select p1.c,p2.c

From Par P1,Par P2

Where P1.p=P2.p and P1.c <>p2.c

What is the result of this query ? sibling

Recursive example

```
Sib(x,y) <- Par(x,p) AND Par(y,p) AND x<>y
Cousin(x,y) <- Sib(x,y)
Cousin(x,y) <- Par(x,xp) AND Par(y,yp)
                    AND Cousin(xp,yp)
```

Example: Evaluation of Cousin

- We'll proceed in rounds to infer Sib facts (red) and Cousin facts (green).
- Remember the rules:

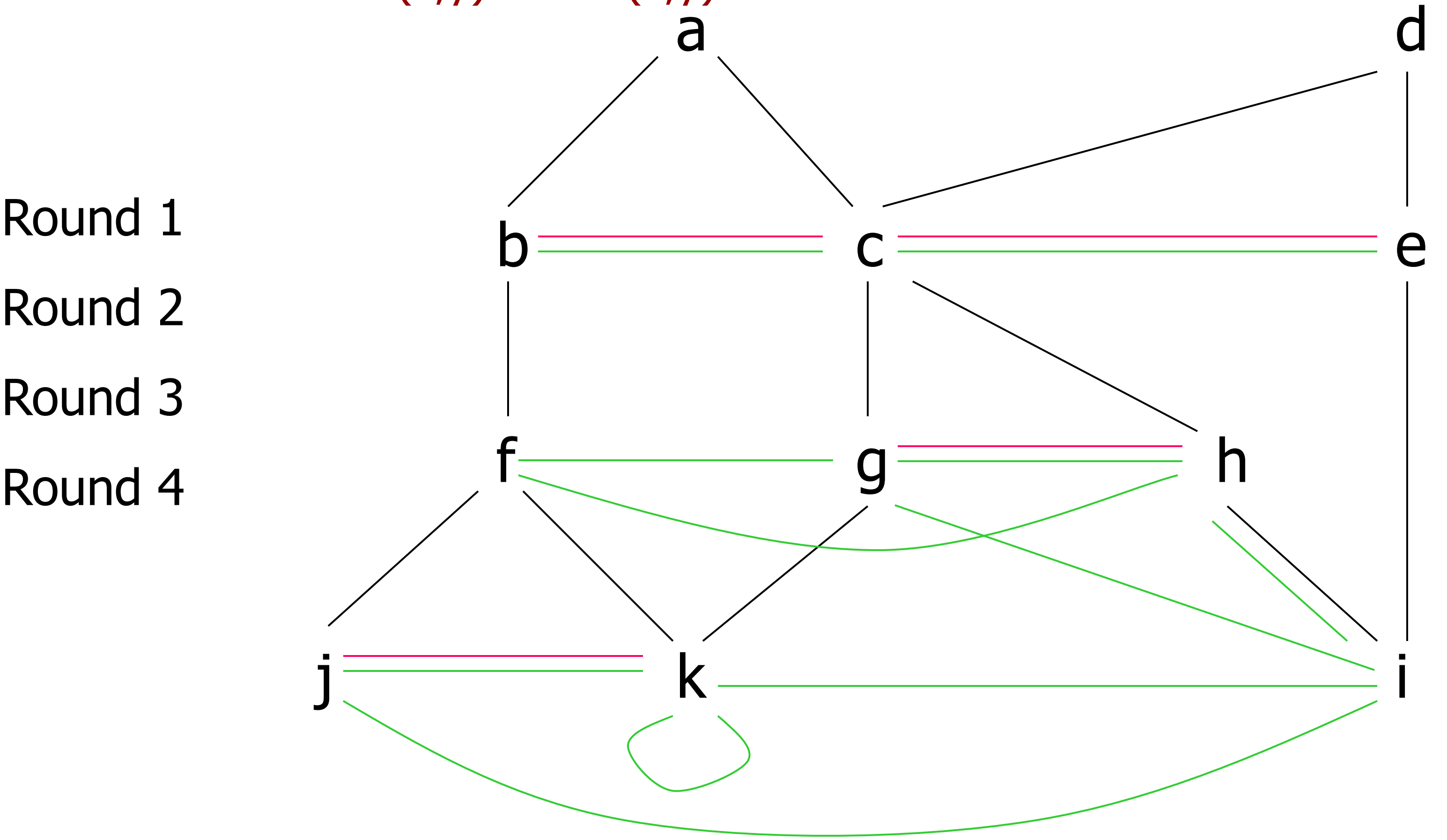
```
Sib(x,y) <- Par(x,p) AND Par(y,p) AND x<>y
Cousin(x,y) <- Sib(x,y)
Cousin(x,y) <- Par(x,xp) AND Par(y,yp)
                         AND Cousin(xp,yp)
```

Par Data: Parent Above Child

Sib(x,y) <- Par(x,p) AND Par(y,p) AND x<>y

Cousin(x,y) <- Par(x,xp) AND Par(y,yp) AND Cousin(xp,yp)

Cousin(x,y) <- Sib(x,y)



SQL-99 Recursion

- Datalog recursion has inspired the addition of recursion to the SQL-99 standard.
- IBM DB2 does implement the SQL-99 proposal.

Form of SQL Recursive Queries

```
WITH
   [RECURSIVE] R1 AS <Definition of R1>
   [RECURSIVE] R2 AS <Definition of R2>

 <a SQL query about EDB,R1,R2,…>
```

Example: SQL Recursion – (1)

- Find Sally’s cousins, using SQL like the recursive Datalog example.
- Par(child,parent) is the EDB.

```
WITH Sib(x,y) AS
        SELECT p1.child, p2.child
        FROM Par p1, Par p2
        WHERE p1.parent = p2.parent AND
                p1.child <> p2.child;
```

Like Sib(x,y) <-
  Par(x,p) AND
  Par(y,p) AND
  x <> y

Example: SQL Recursion – (2)

Required – Cousin is recursive

Reflects Cousin(x,y) <- Sib(x,y)

```
WITH
RECURSIVE Cousin(x,y) AS
   (SELECT * FROM Sib)
      UNION
  (SELECT p1.child, p2.child
   FROM Par p1, Par p2, Cousin
   WHERE p1.parent = Cousin.x AND
   p2.parent = Cousin.y);
```

Reflects Cousin(x,y) <- Par(x,xp) AND Par(y,yp) AND Cousin(xp,yp)

Example: SQL Recursion – (3)

- With those definitions, we can add the query, which is about the virtual view Cousin(x,y):

```
SELECT y
FROM Cousin
WHERE x = ’Sally’;
```

Legal SQL Recursion

- It is possible to define SQL recursions that do not have a meaning.
- The SQL standard restricts recursion so there is a meaning.

Another Example

```
create table Employee(ID int, salary int);
create table Manager(mID int, eID int);
create table Project(name text, mgrID int);
```

- Find total salary cost of project 'X'

Solution 1:

Employee(ID , salary)
Manager(mID, eID)
Project(name,mgrID)

```
with recursive
  Superior as (select * from Manager
               union
               select S.mID, M.eID
               from Superior S, Manager M
               where S.eID = M.mID )

select sum(salary)
from Employee
where ID in
  (select mgrID from Project where name = 'X'
   union
   select eID from Project, Superior
   where Project.name = 'X' AND Project.mgrID = Superior.mID );
```

Solution 2:

Employee(ID , salary)
Manager(mID, eID)
Project(name,mgrID)

```
with recursive
  Xemps(ID) as (select mgrID as ID from
   Project where name = 'X'
                  union
                  select eID as ID
                  from Manager M, Xemps X
                  where M.mID = X.ID)
select sum(salary)
from Employee
where ID in (select ID from Xemps);
```