Database Management Systems

Vijaya Saradhi

IIT Guwahati

Mon, 06th Apr 2020

Recursive Programming

Contents from

Contents of this lecture are from section 10.3 of the text book **Database systems The complete book**

Datalog: Recursive Programming

Recursive Rules

1. `Reaches(x, y) ← Flights(a, x, y, d, r)`
2. `Reaches(x, y) ← Reaches(x, z) AND Reaches(z, y)`

Flights

airline	from	to	departs	arrives
UA	SF	DEN	0930	1230
AA	SF	DAL	0900	1430
UA	DEN	CHI	1500	1800
UA	DEN	DAL	1400	1700
AA	DAL	CHI	1530	1730
AA	DAL	NY	1500	1930
AA	CHI	NY	1900	2200
UA	CHI	NY	1830	2130

Datalog: Recursive Programming

Recursive Rules

1. `Reaches(x, y) ← Flights(a, x, y, d, r)`
2. `Reaches(x, y) ← Reaches(x, z) AND Reaches(z, y)`

Flights				
airline	from	to	departs	arrives
UA	SF	DEN	0930	1230
AA	SF	DAL	0900	1430
UA	DEN	CHI	1500	1800
UA	DEN	DAL	1400	1700
AA	DAL	CHI	1530	1730
AA	DAL	NY	1500	1930
AA	CHI	NY	1900	2200
UA	CHI	NY	1830	2130

Round # 1	
Reaches	
x	y
SF	DEN
SF	DAL
DEN	CHI
DEN	DAL
DAL	CHI
DAL	NY
CHI	NY

Datalog: Recursive Programming

Recursive Rules

1. `Reaches(x, y) ← Reaches(x, z) AND Reaches(z, y)`

Reaches R1		Reaches R2	
R1.x	R1.z	R2.z	R2.y
SF	DEN	SF	DEN
SF	DEN	SF	DAL
SF	DEN	DEN	CHI
SF	DEN	DEN	DAL
SF	DEN	DAL	CHI
SF	DEN	DAL	NY
SF	DEN	CHI	NY

Round # 1	
Reaches	
x	y
SF	DEN
SF	DAL
DEN	CHI
DEN	DAL
DAL	CHI
DAL	NY
CHI	NY

Datalog: Recursive Programming

Result

1. `Reaches(x, y) ← Reaches(x, z) AND Reaches(z, y)`

Reaches R1		Reaches R2	
R1.x	R1.z	R2.z	R2.y
SF	DEN	SF	DEN
SF	DEN	SF	DAL
SF	DEN	DEN	CHI
SF	DEN	DEN	DAL
SF	DEN	DAL	CHI
SF	DEN	DAL	NY
SF	DEN	CHI	NY

Round # 1	
Reaches	
x	y
SF	DEN
SF	DAL
DEN	CHI
DEN	DAL
DAL	CHI
DAL	NY
CHI	NY
Round #2	
SF	CHI

Datalog: Recursive Programming

Recursive Rules

1. Reaches(x, y) ← Reaches(x, z) AND Reaches(z, y)

Reaches R1		Reaches R2	
R1.x	R1.z	R2.z	R2.y
SF	DAL	SF	DEN
SF	DAL	SF	DAL
SF	DAL	DEN	CHI
SF	DAL	DEN	DAL
SF	DAL	DAL	CHI
SF	DAL	DAL	NY
SF	DAL	CHI	NY

Round # 1	
Reaches	
x	y
SF	DEN
SF	DAL
DEN	CHI
DEN	DAL
DAL	CHI
DAL	NY
CHI	NY
Round #2	
SF	CHI
SF	NY

Datalog: Recursive Programming

Recursive Result

1. Reaches(x, y) ← Reaches(x, z) AND Reaches(z, y)

Reaches R1		Reaches R2	
R1.x	R1.z	R2.z	R2.y
DEN	CHI	SF	DEN
DEN	CHI	SF	DAL
DEN	CHI	DEN	CHI
DEN	CHI	DEN	DAL
DEN	CHI	DAL	CHI
DEN	CHI	DAL	NY
DEN	CHI	CHI	NY

Round # 1	
Reaches	
x	y
SF	DEN
SF	DAL
DEN	CHI
DEN	DAL
DAL	CHI
DAL	NY
CHI	NY
Round #2	
SF	CHI
SF	NY
DEN	NY

Datalog: Recursive Programming

Recursive Result

1. `Reaches(x, y) ← Reaches(x, z) AND Reaches(z, y)`

Reaches R1		Reaches R2	
R1.x	R1.z	R2.z	R2.y
DEN	DAL	SF	DEN
DEN	DAL	SF	DAL
DEN	DAL	DEN	CHI
DEN	DAL	DEN	DAL
DEN	DAL	DAL	CHI
DEN	DAL	DAL	NY
DEN	DAL	CHI	NY

Round # 1	
Reaches	
x	y
SF	DEN
SF	DAL
DEN	CHI
DEN	DAL
DAL	CHI
DAL	NY
CHI	NY
Round #2	
SF	CHI
SF	NY
DEN	NY
DEN	CHI

Datalog: Recursive Programming

Recursive Result

1. Reaches(x, y) ← Reaches(x, z) AND Reaches(z, y)

Reaches R1		Reaches R2	
R1.x	R1.z	R2.z	R2.y
DEN	DAL	SF	DEN
DEN	DAL	SF	DAL
DEN	DAL	DEN	CHI
DEN	DAL	DEN	DAL
DEN	DAL	DAL	CHI
DEN	DAL	DAL	NY
DEN	DAL	CHI	NY

Round # 1	
Reaches	
x	y
SF	DEN
SF	DAL
DEN	CHI
DEN	DAL
DAL	CHI
DAL	NY
CHI	NY
Round #2	
SF	CHI
SF	NY
DEN	NY

Datalog: Recursive Programming

Recursive Result

1. Reaches(x, y) ← Reaches(x, z) AND Reaches(z, y)

Reaches R1		Reaches R2	
R1.x	R1.z	R2.z	R2.y
DAL	CHI	SF	DEN
DAL	CHI	SF	DAL
DAL	CHI	DEN	CHI
DAL	CHI	DEN	DAL
DAL	CHI	DAL	CHI
DAL	CHI	DAL	NY
DAL	CHI	CHI	NY

Round # 1	
Reaches	
x	y
SF	DEN
SF	DAL
DEN	CHI
DEN	DAL
DAL	CHI
DAL	NY
CHI	NY
Round #2	
SF	CHI
SF	NY
DEN	NY

Datalog: Recursive Programming

Recursive Result

1. Reaches(x, y) ← Reaches(x, z) AND Reaches(z, y)

Reaches R1		Reaches R2	
R1.x	R1.z	R2.z	R2.y
DAL	NY	SF	DEN
DAL	NY	SF	DAL
DAL	NY	DEN	CHI
DAL	NY	DEN	DAL
DAL	NY	DAL	CHI
DAL	NY	DAL	NY
DAL	NY	CHI	NY

Round # 1	
Reaches	
x	y
SF	DEN
SF	DAL
DEN	CHI
DEN	DAL
DAL	CHI
DAL	NY
CHI	NY
Round #2	
SF	CHI
SF	NY
DEN	NY

Datalog: Recursive Programming

Recursive Result

1. `Reaches(x, y) ← Reaches(x, z) AND Reaches(z, y)`

Reaches R1		Reaches R2	
R1.x	R1.z	R2.z	R2.y
CHI	NY	SF	DEN
CHI	NY	SF	DAL
CHI	NY	DEN	CHI
CHI	NY	DEN	DAL
CHI	NY	DAL	CHI
CHI	NY	DAL	NY
CHI	NY	CHI	NY

Round # 1	
Reaches	
x	y
SF	DEN
SF	DAL
DEN	CHI
DEN	DAL
DAL	CHI
DAL	NY
CHI	NY
Round #2	
SF	CHI
SF	NY
DEN	NY

Datalog: Recursive Programming

Recursive Result

1. `Reaches(x, y) ← Reaches(x, z) AND Reaches(z, y)`

Reaches R1		Reaches R2	
R1.x	R1.z	R2.z	R2.y
SF	DEN	SF	DEN
SF	DEN	SF	DAL
SF	DEN	DEN	CHI
SF	DEN	DEN	DAL
SF	DEN	DAL	CHI
SF	DEN	DAL	NY
SF	DEN	CHI	NY
SF	DEN	DAL	CHI
SF	DEN	DAL	NY
SF	DEN	CHI	NY

Round # 2	
Reaches	
x	y
SF	DEN
SF	DAL
DEN	CHI
DEN	DAL
DAL	CHI
DAL	NY
CHI	NY
SF	CHI
SF	NY
DEN	NY

Datalog: Recursive Programming

Recursive Result

1. `Reaches(x, y) ← Reaches(x, z) AND Reaches(z, y)`

Reaches R1		Reaches R2	
R1.x	R1.z	R2.z	R2.y
SF	DAL	SF	DEN
SF	DAL	SF	DAL
SF	DAL	DEN	CHI
SF	DAL	DEN	DAL
SF	DAL	DAL	CHI
SF	DAL	DAL	NY
SF	DAL	CHI	NY
SF	DAL	DAL	CHI
SF	DAL	DAL	NY
SF	DAL	CHI	NY

Round # 2

Reaches	
x	y
SF	DEN
SF	DAL
DEN	CHI
DEN	DAL
DAL	CHI
DAL	NY
CHI	NY
SF	CHI
SF	NY
DEN	NY

Datalog: Recursive Programming

Recursive Result

1. `Reaches(x, y) ← Reaches(x, z) AND Reaches(z, y)`

Reaches R1		Reaches R2	
R1.x	R1.z	R2.z	R2.y
DEN	CHI	SF	DEN
DEN	CHI	SF	DAL
DEN	CHI	DEN	CHI
DEN	CHI	DEN	DAL
DEN	CHI	DAL	CHI
DEN	CHI	DAL	NY
DEN	CHI	CHI	NY
DEN	CHI	DAL	CHI
DEN	CHI	DAL	NY
DEN	CHI	CHI	NY

Round # 2	
Reaches	
x	y
SF	DEN
SF	DAL
DEN	CHI
DEN	DAL
DAL	CHI
DAL	NY
CHI	NY
SF	CHI
SF	NY
DEN	NY

Datalog: Recursive Programming

Recursive Result

1. `Reaches(x, y) ← Reaches(x, z) AND Reaches(z, y)`

Reaches R1		Reaches R2	
R1.x	R1.z	R2.z	R2.y
DEN	DAL	SF	DEN
DEN	DAL	SF	DAL
DEN	DAL	DEN	CHI
DEN	DAL	DEN	DAL
DEN	DAL	DAL	CHI
DEN	DAL	DAL	NY
DEN	DAL	CHI	NY
DEN	DAL	DAL	CHI
DEN	DAL	DAL	NY
DEN	DAL	CHI	NY

Round # 2	
Reaches	
x	y
SF	DEN
SF	DAL
DEN	CHI
DEN	DAL
DAL	CHI
DAL	NY
CHI	NY
SF	CHI
SF	NY
DEN	NY

Datalog: Recursive Programming

Recursive Result

1. `Reaches(x, y) ← Reaches(x, z) AND Reaches(z, y)`

Reaches R1		Reaches R2	
R1.x	R1.z	R2.z	R2.y
DAL	CHI	SF	DEN
DAL	CHI	SF	DAL
DAL	CHI	DEN	CHI
DAL	CHI	DEN	DAL
DAL	CHI	DAL	CHI
DAL	CHI	DAL	NY
DAL	CHI	CHI	NY
DAL	CHI	DAL	CHI
DAL	CHI	DAL	NY
DAL	CHI	CHI	NY

Round # 2

Reaches	
x	y
SF	DEN
SF	DAL
DEN	CHI
DEN	DAL
DAL	CHI
DAL	NY
CHI	NY
SF	CHI
SF	NY
DEN	NY

Datalog: Recursive Programming

Recursive Result

1. `Reaches(x, y) ← Reaches(x, z) AND Reaches(z, y)`

Reaches R1		Reaches R2	
R1.x	R1.z	R2.z	R2.y
DAL	NY	SF	DEN
DAL	NY	SF	DAL
DAL	NY	DEN	CHI
DAL	NY	DEN	DAL
DAL	NY	DAL	CHI
DAL	NY	DAL	NY
DAL	NY	CHI	NY
DAL	NY	DAL	CHI
DAL	NY	DAL	NY
DAL	NY	CHI	NY

Round # 2	
Reaches	
x	y
SF	DEN
SF	DAL
DEN	CHI
DEN	DAL
DAL	CHI
DAL	NY
CHI	NY
SF	CHI
SF	NY
DEN	NY

Datalog: Recursive Programming

Recursive Result

1. `Reaches(x, y) ← Reaches(x, z) AND Reaches(z, y)`

Reaches R1		Reaches R2	
R1.x	R1.z	R2.z	R2.y
CHI	NY	SF	DEN
CHI	NY	SF	DAL
CHI	NY	DEN	CHI
CHI	NY	DEN	DAL
CHI	NY	DAL	CHI
CHI	NY	DAL	NY
CHI	NY	CHI	NY
CHI	NY	DAL	CHI
CHI	NY	DAL	NY
CHI	NY	CHI	NY

Round # 2

Reaches	
x	y
SF	DEN
SF	DAL
DEN	CHI
DEN	DAL
DAL	CHI
DAL	NY
CHI	NY
SF	CHI
SF	NY
DEN	NY

Datalog: Recursive Programming

Recursive Result

1. `Reaches(x, y) ← Reaches(x, z) AND Reaches(z, y)`

Reaches R1		Reaches R2	
R1.x	R1.z	R2.z	R2.y
SF	CHI	SF	DEN
SF	CHI	SF	DAL
SF	CHI	DEN	CHI
SF	CHI	DEN	DAL
SF	CHI	DAL	CHI
SF	CHI	DAL	NY
SF	CHI	CHI	NY
SF	CHI	DAL	CHI
SF	CHI	DAL	NY
SF	CHI	CHI	NY

Round # 2	
Reaches	
x	y
SF	DEN
SF	DAL
DEN	CHI
DEN	DAL
DAL	CHI
DAL	NY
CHI	NY
SF	CHI
SF	NY
DEN	NY

Datalog: Recursive Programming

Recursive Result

1. `Reaches(x, y) ← Reaches(x, z) AND Reaches(z, y)`

Reaches R1		Reaches R2	
R1.x	R1.z	R2.z	R2.y
SF	NY	SF	DEN
SF	NY	SF	DAL
SF	NY	DEN	CHI
SF	NY	DEN	DAL
SF	NY	DAL	CHI
SF	NY	DAL	NY
SF	NY	CHI	NY
SF	NY	DAL	CHI
SF	NY	DAL	NY
SF	NY	CHI	NY

Round # 2

Reaches	
x	y
SF	DEN
SF	DAL
DEN	CHI
DEN	DAL
DAL	CHI
DAL	NY
CHI	NY
SF	CHI
SF	NY
DEN	NY

Datalog: Recursive Programming

Recursive Result

1. `Reaches(x, y) ← Reaches(x, z) AND Reaches(z, y)`

Reaches R1		Reaches R2	
R1.x	R1.z	R2.z	R2.y
DEN	NY	SF	DEN
DEN	NY	SF	DAL
DEN	NY	DEN	CHI
DEN	NY	DEN	DAL
DEN	NY	DAL	CHI
DEN	NY	DAL	NY
DEN	NY	CHI	NY
DEN	NY	DAL	CHI
DEN	NY	DAL	NY
DEN	NY	CHI	NY

Round # 2

Reaches	
x	y
SF	DEN
SF	DAL
DEN	CHI
DEN	DAL
DAL	CHI
DAL	NY
CHI	NY
SF	CHI
SF	NY
DEN	NY

Datalog: Recursive Programming

Recursive Rules

1. Difference between Departure time at next hop and arrival time should be at least 100 minutes
2. `Connects(x, y, d, r)` ← `Flights(a, x, y, d, r)`
3. `Connects(x, y, d, r)` ← `Connects(a, x, z, d, t1) AND Connects(a, z, y, t2, r) AND t1` ≤ `t2 - 100`

Flights				
airline	from	to	departs	arrives
UA	SF	DEN	0930	1230
AA	SF	DAL	0900	1430
UA	DEN	CHI	1500	1800
UA	DEN	DAL	1400	1700
AA	DAL	CHI	1530	1730
AA	DAL	NY	1500	1930
AA	CHI	NY	1900	2200
UA	CHI	NY	1830	2130

Connects			
x	y	d	r
Round #1			
SF	DEN	0930	1230
SF	DAL	0900	1430
DEN	CHI	1500	1800
DEN	DAL	1400	1700
DAL	CHI	1530	1730
DAL	NY	1500	1930
CHI	NY	1900	2200
Round #2			
SF	CHI	0900	1730
SF	CHI	0930	1800
SF	DAL	0930	1700
DEN	NY	1500	2200
DAL	NY	1530	2130
DAL	NY	1530	2200
Round #3			
SF	NY	0900	2130
SF	NY	0900	2200
SF	NY	0930	2200

Datalog: Recursive Programming

Recursive Rules

1. Find those pairs of cities (x, y) in which only UA operates but not AA
2. `UAReaches(x, y) ← Flights(a, x, y, d, r)`
3. `UAReaches(x, y) ← UAReaches(x, z) AND UAReaches(z, y)`
4. `AAReaches(x, y) ← Flights(a, x, y, d, r)`
5. `AAReaches(x, y) ← AAReaches(x, z) AND AAReaches(z, y)`
6. `UAOnly(x, y) ← UAReaches(x, y) AND NOT AAReaches(x, y)`

UAReaches		UAOnly	
x	y	x	y
SF	DEN	SF	DEN
SF	DAL	DEN	DAL
SF	CHI	DEN	CHI
SF	NY	DEN	NY
DEN	DAL		
DEN	CHI		
DEN	NY		
CHI	NY		
AAReaches			
SF	DAL		
SF	CHI		
SF	NY		
DAL	CHI		
DAL	NY		
CHI	NY		

Flights				
airline	from	to	departs	arrives
UA	SF	DEN	0930	1230
AA	SF	DAL	0900	1430
UA	DEN	CHI	1500	1800
UA	DEN	DAL	1400	1700
AA	DAL	CHI	1530	1730
AA	DAL	NY	1500	1930
AA	CHI	NY	1900	2200
UA	CHI	NY	1830	2130

SQL: Recursive Programming

Introduction

- SQL has grown to be an expressive data-oriented language
- Intentionally it has not been designed as a general-purpose programming language
- SQL does not loop forever.
- Any SQL query is expected to terminate, regardless of size/contents of the input tables
- SQL queries are evaluated efficiently

SQL: Addition of recursion

Expressive

SQL becomes a Turing-complete language thus a general-purpose programming lanugage

Efficiency

No longer queries are guaranteed to terminate.

SQL: WITH RECURSIVE

Recursive common table expression (CTE)

```
WITH RECURSIVE T(c1, c2, ..., ck)  -- common schema of q0 and q.(.)
AS(
    q0    -- base case query, evaluated once

    UNION [ALL]

    q0(T) -- recursive query refers to T itself
)       -- evaluated repeatedly

q(T)    -- final post processing query
```

SQL: WITH RECURSIVE

Initial query

Forms the base result set of the CTE structure. The initial query part is referred to as an anchor member.

Recursive query

References to the CTE name, therefore, it is called a recursive member. The recursive member is joined with the anchor member by UNION or UNION ALL

Termination

A termination condition that ensures the recursion stops when the recursive member returns no row

SQL: WITH RECURSIVE

Step 1

Separate the members into anchor and recursive members

Step 2

Execute the anchor member to form the base result set R0. Use this base result set for next iteration

Step 3

Execute the recursive member with Ri result set as input and make Ri+1 as an output

Step 4

Repeat step 3 till recursive member returns an empty result set

Step 5

Combine result sets from R0 to Rn using UNION [ALL] operator

Example - 01

Recursive Rules

1. Reaches(x, y) ← Flights(a, x, y, d, r)
2. Reaches(x, y) ← Reaches(x, z) AND Reaches(z, y)

Recursive SQL Query

```
WITH RECURSIVE Reaches(frm, to)  -- T(c1, c2)
AS(
        (SELECT frm, to FROM Flights) -- q0

    UNION
        (SELECT R1.frm, R2.to
         FROM    Reaches AS R1, Reaches AS R2
         WHERE   R1.to = R2.frm
        )
)
SELECT * FROM Reaches;
```