SQL: Recursion

Introduction to Databases
CompSci 316 Fall 2018
Announcements (Tue., Oct. 2)

• Homework 2 Problems 1-4 and X1 due tonight
• Homework 2 Problems 5 & 6 due tomorrow night
• Homework 2 Problem 7 due after break
• Extra office hours for midterm preparation
  • UTA Wed. (Oct. 3) 6-7:45pm
  • Jun Thu. (Oct. 4) 9-10am
• Midterm in class Thursday
  • Open-book, open-notes
  • Same format as sample midterm

• A last-minute, informal mixer Thu. (Oct. 4) 8-10pm in LSRC D243 for those still looking for members/teams
• Project Milestone #1 due next Thursday
A motivating example

Example: find Bart’s ancestors

“Ancestor” has a recursive definition

- $X$ is $Y$’s ancestor if
  - $X$ is $Y$’s parent, or
  - $X$ is $Z$’s ancestor and $Z$ is $Y$’s ancestor
Recursion in SQL

• SQL2 had no recursion
  • You can find Bart’s parents, grandparents, great grandparents, etc.
    SELECT p1.parent AS grandparent
    FROM Parent p1, Parent p2
    WHERE p1.child = p2.parent
    AND p2.child = 'Bart';
  • But you cannot find all his ancestors with a single query

• SQL3 introduces recursion
  • WITH clause
  • Implemented in PostgreSQL (common table expressions)
Ancestor query in SQL3

WITH RECURSIVE Ancestor(anc, desc) AS
((SELECT parent, child FROM Parent)
UNION
(SELECT a1.anc, a2.desc
FROM Ancestor a1, Ancestor a2
WHERE a1.desc = a2.anc))

SELECT anc
FROM Ancestor
WHERE desc = 'Bart';
Fixed point of a function

• If \( f : D \rightarrow D \) is a function from a type \( D \) to itself, a **fixed point** of \( f \) is a value \( x \) such that \( f(x) = x \)

• Example: What is the fixed point of \( f(x) = x / 2 \)?
  • 0, because \( f(0) = 0/2 = 0 \)

• To compute a fixed point of \( f \)
  • Start with a “seed”: \( x \leftarrow x_0 \)
  • Compute \( f(x) \)
    • If \( f(x) = x \), stop; \( x \) is fixed point of \( f \)
    • Otherwise, \( x \leftarrow f(x) \); repeat

• Example: compute the fixed point of \( f(x) = x / 2 \)
  • With seed 1: 1, 1/2, 1/4, 1/8, 1/16, ... \( \rightarrow 0 \)

\( \textbullet \) Doesn’t always work, but happens to work for us!
Fixed point of a query

• A query $q$ is just a function that maps an input table to an output table, so a **fixed point** of $q$ is a table $T$ such that $q(T) = T$

• To compute fixed point of $q$
  • Start with an empty table: $T \leftarrow \emptyset$
  • Evaluate $q$ over $T$
    • If the result is identical to $T$, stop; $T$ is a fixed point
    • Otherwise, let $T$ be the new result; repeat

Starting from $\emptyset$ produces the **unique minimal fixed point** (assuming $q$ is monotone)
Finding ancestors

• WITH RECURSIVE
  Ancestor(anc, desc) AS
  ((SELECT parent, child FROM Parent)
  UNION
  (SELECT a1.anc, a2.desc
   FROM Ancestor a1, Ancestor a2
   WHERE a1.desc = a2.anc))
• Think of the definition as Ancestor = q(Ancestor)
Intuition behind fixed-point iteration

• Initially, we know nothing about ancestor-descendent relationships
• In the first step, we deduce that parents and children form ancestor-descendent relationships
• In each subsequent steps, we use the facts deduced in previous steps to get more ancestor-descendent relationships
• We stop when no new facts can be proven