



Recursion in SQL

Basic recursive
WITH statement

SQL is not a “Turing complete” language

- Simple, convenient, declarative
- Expressive enough for most database queries
- But basic SQL can't express unbounded computations

Example 1: Ancestors

ParentOf(parent, child)

➤ Find all of Mary's ancestors

Sue Mary

Bob Mary

Fred Bob

Jane Bob

parents

grandparents



Two instances of ParentOf
Three

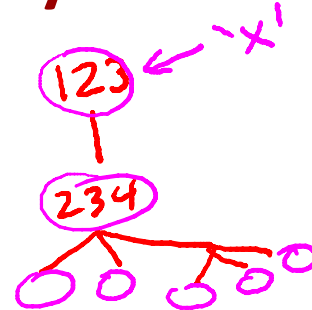


Example 2: Company hierarchy

Employee(ID, salary) ←

Manager(mID, eID) ←

Project(name, mgrID)



➤ Find total salary cost of project 'X'

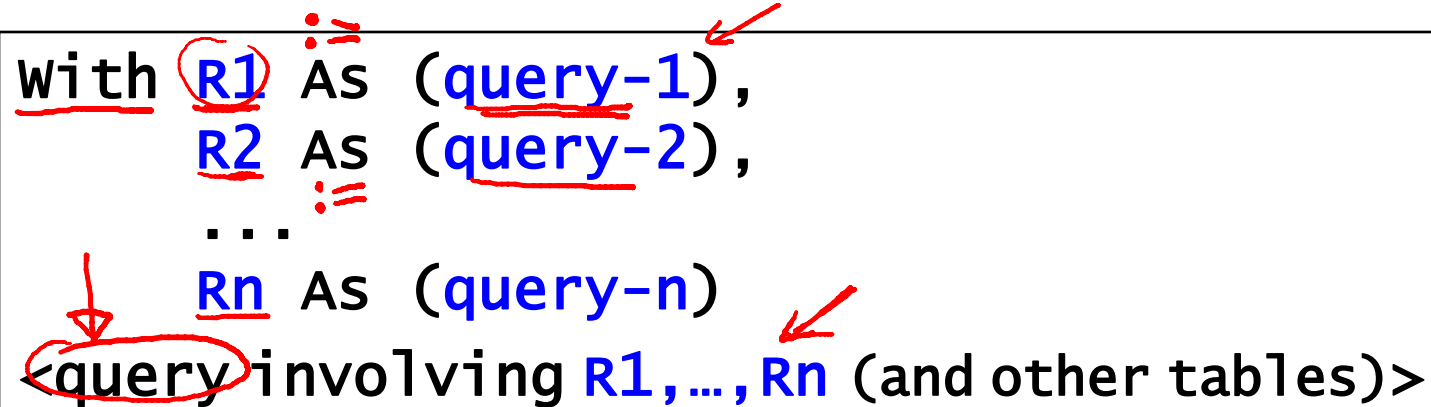
Example 3: Airline flights

Flight(orig, dest, airline, cost)

➤ Find cheapest way to fly from 'A' to 'B'

SQL With Statement

```
with R1 AS (query-1),  
      R2 AS (query-2),  
      ...  
      Rn AS (query-n)  
<query involving R1, ..., Rn (and other tables)>
```



The diagram illustrates the syntax of a SQL With Statement. It shows a sequence of table aliases (R1, R2, ..., Rn) each followed by an AS clause and a query (query-1, query-2, ..., query-n). The word 'with' is underlined. Red arrows point to the AS keyword, the query text, and the table aliases. A red circle highlights the word 'query' in the final clause, and another red circle highlights the table aliases R1, ..., Rn.

SQL With Statement

```
With R1(A1,A2,...,Am) AS (query-1),  
    R2 AS (query-2),  
    ...  
    Rn AS (query-n)  
<query involving R1,...,Rn (and other tables)>
```

SQL With Recursive Statement

With Recursive

R1 AS (query-1),

R2 AS (query-2),

...

Recursive Rn AS (query-n)

<query involving R1,...,Rn (and other tables)>

SQL With Recursive Statement

With Recursive

→ R As (base query ← not R
| Union all ← R
| recursive query)
<query involving R (and other tables)>