

# 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

parents  
 grandparents  
 ↗  
 Two instances of ParentOf  
 Three ↗

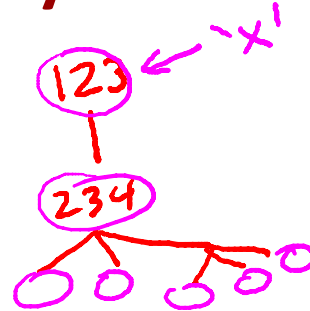
Sue Mary  
 Bob Mary  
 Fred Bob  
 Jane Bob

## 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 shows a SQL WITH statement with several annotations. The word 'with' is underlined in red. The table names 'R1', 'R2', and 'Rn' are underlined in red. The queries 'query-1', 'query-2', and 'query-n' are underlined in red. A red arrow points to the 'AS' keyword in the first line. Another red arrow points to the 'AS' keyword in the last line. A red arrow points to the '<query>' part of the final line. A red circle is drawn around the '<query>' part of the final line. There are also red double arrows pointing to the first and last lines of the WITH clause.

## 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)>