

Database Management Systems

Relational Algebra

Query Languages

- Query languages are programming languages, but not general purpose programming languages.
- Query languages are NOT Turing-complete languages.
- Advantage:
 - easy to learn and simple to use
 - leaves the compiler (optimizer) sufficient room to generate highly optimized executable code

Algebra

- Algebra in general consists of operators and atomic operands.
- In Relational Algebra, atomic operands are:
 - Variables that stand for relations
 - constants, which are finite relations.
- Expressions of relational algebra are usually referred to as queries.
- Relational algebra uses set semantics (a relation is a set of tuples and duplicates are removed automatically in the result relation).

Operations of Relational Algebra

- set operations
- operations that remove parts of a relation
- operations that combine the tuples of two relations
- operations to renaming relations or attributes

Set Operations

- set union, $\text{Relation}_1 \cup \text{Relation}_2$
- set intersection, $\text{Relation}_1 \cap \text{Relation}_2$
- set difference, $\text{Relation}_1 - \text{Relation}_2$
- they are all binary operators
- The conditions to apply these operations:
 - Both relations must be union compatible (i.e., with identical schema)
 - must have identical sets of attributes and types for each attribute must be the same
 - attributes must be ordered the same way

Other (Database) Operations

- Unary operators:
 - Projection, $\pi_{\{\text{list of attribute names}\}}(\text{Relation})$
 - Selection, $\delta_{(\text{condition})}(\text{Relation})$
 - renaming, $\rho_{S(A_1, A_2, \dots, A_n)}(\text{Relation})$
- Other binary operators
 - cartesian product (cross product), $\text{Relation}_1 \times \text{Relation}_2$
 - join, $\text{Relation}_1 \bowtie_{(\text{condition})} \text{Relation}_2$
 \equiv combination of cartesian and selection, $\delta_{(\text{condition})}(\text{Relation}_1 \times \text{Relation}_2)$
 - natural join, $\text{Relation}_1 \bowtie \text{Relation}_2$

Algebra Queries

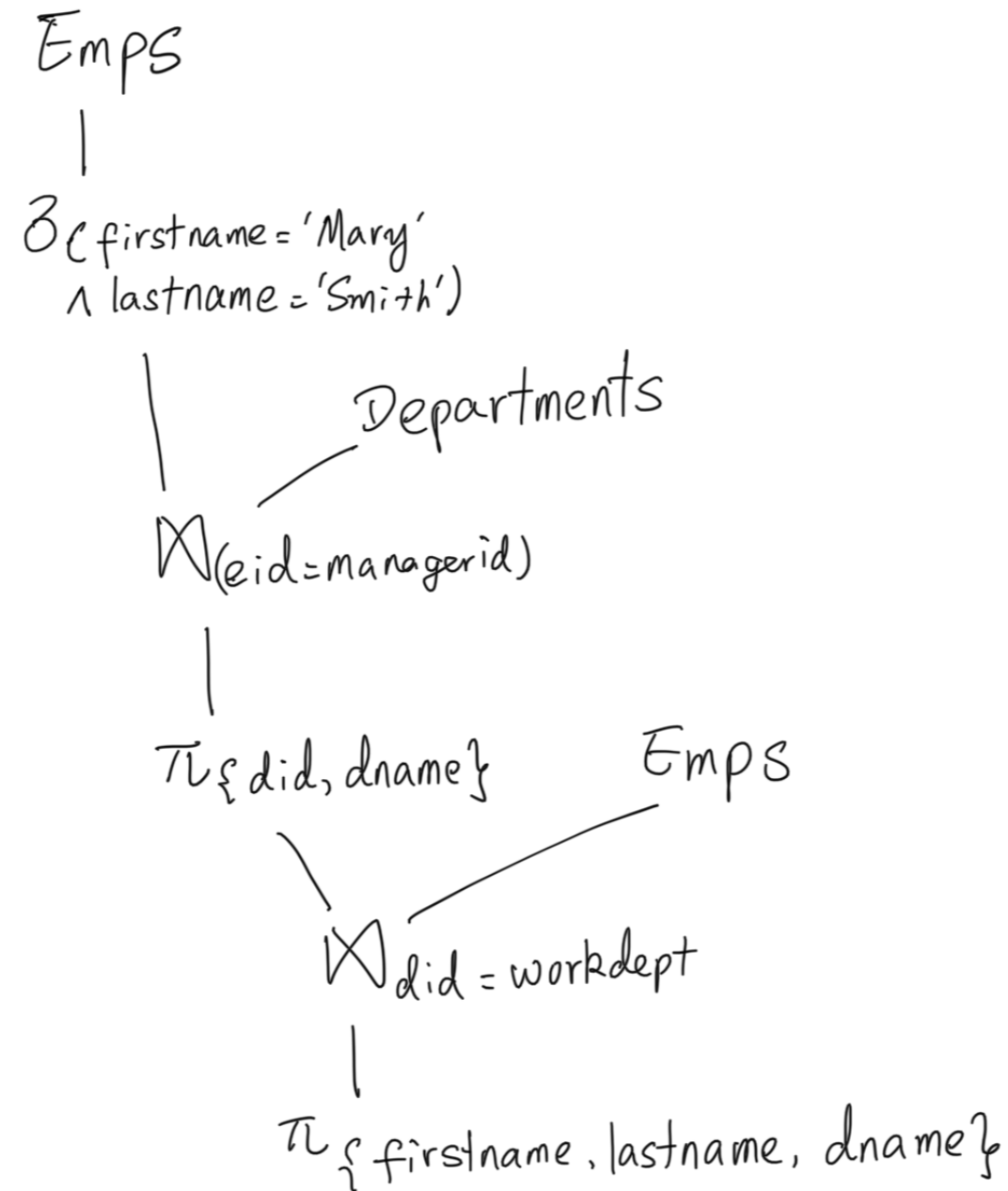
- Relational Algebra is a functional programming language
- Relation(s) in, relation out for all operators
- Nest operations — take the output of previous operation as the operand of the next operation
- Expressiveness of a query language — relational completeness: the capability of forming arbitrarily complex query to access database data without using either iteration or recursion.

Algebra Query Example

- For each employee who works in a department managed by Mary Smith, list his/her first and last name and the department's name.

```
P_{firstname, lastname, dname} (  
  P_{did,dname}(  
    S_(firstname='Mary' and lastname='Smith')Emps  
      J_(eid=managerid) Departments))  
  J_(did = workdept) Emps)
```


Algebra Query Tree



Equivalent Queries

