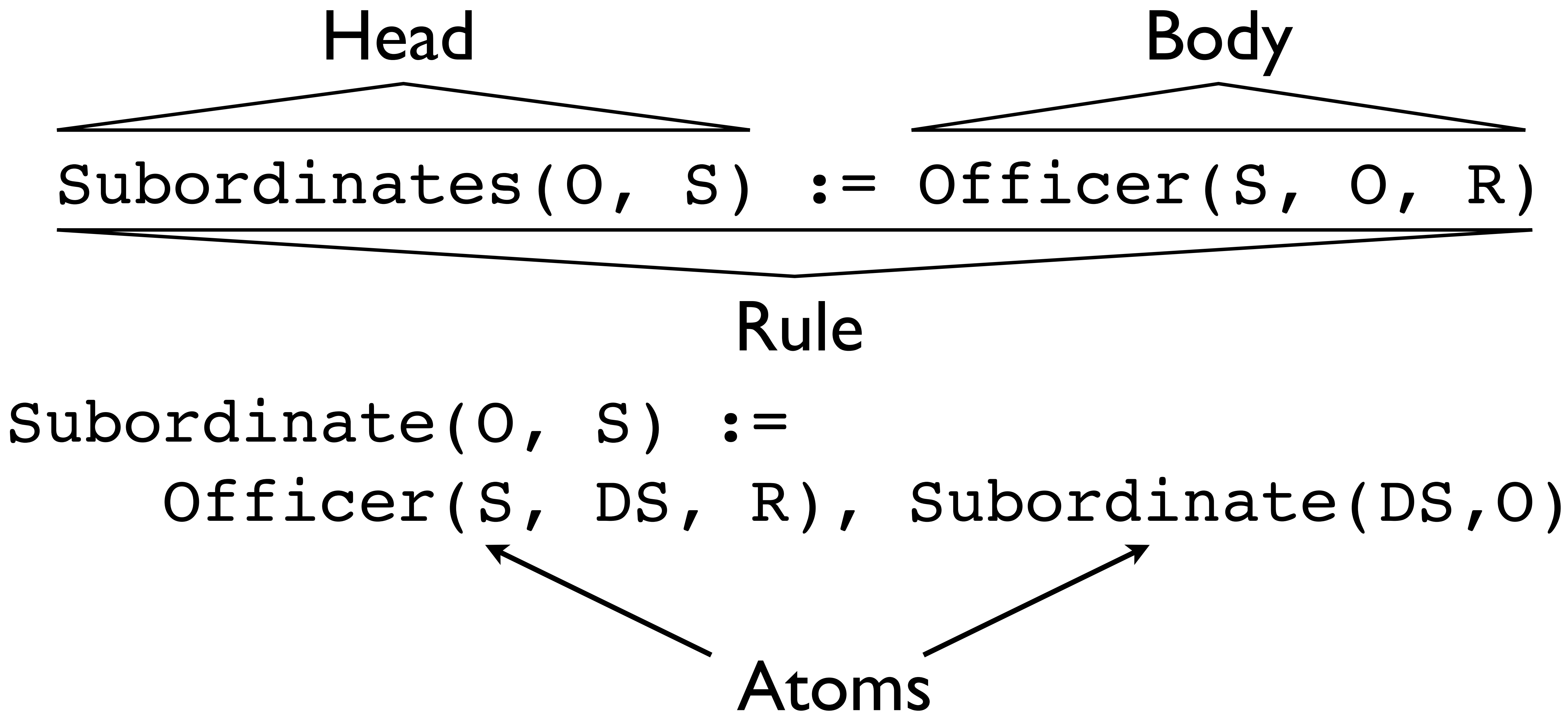
# Datalog

R&G Ch 24

# Recursion in SQL

- SQL92 does not support recursion
  - Added in SQL99 via WITH RECURSIVE
  - View Defined via Transitive Closure
- Important class of queries require recursion.
- Simpler formulation of the same principles:
  - Datalog, a recursive query language

# Datalog



Head

Body

```
Subordinates(O, S) := Officer(S, O, R)
```

Rule

```
Subordinate(O, S) :=
    Officer(S, DS, R), Subordinate(DS,O)
```

Atoms

# Safety

```
Subordinates(O, S, Opinion) :=
    Officer(S, O, R)
```

Where does ‘Opinion’ come from?

Subordinates now has an infinite number of rows.

This definition of Subordinates is unsafe.
Each variable must be range-restricted.

# Negation

```
Ham(O) :=
   Officer(O,S,R), R = 4, NOT Stoic(O)
Stoic(O) :=
   Officer(O,S,R), NOT Ham(O)
```

http://tvtropes.org/pmwiki/pmwiki.php/Main/LargeHam
http://tvtropes.org/pmwiki/pmwiki.php/Main/TheStoic

# Negation

```
Ham(O) :=
   Officer(O,S,R), R = 4, NOT Stoic(O)
Stoic(O) :=
   Officer(O,S,R), NOT Ham(O)
```

Is Spock (Rank = 3) a ‘Large Ham’, or a ‘The Stoic’?

http://tvtropes.org/pmwiki/pmwiki.php/Main/LargeHam
http://tvtropes.org/pmwiki/pmwiki.php/Main/TheStoic

# Negation

```
Ham(O) :=
   Officer(O,S,R), R = 4, NOT Stoic(O)
Stoic(O) :=
   Officer(O,S,R), NOT Ham(O)
```

Is Spock (Rank = 3) a ‘Large Ham’, or a ‘The Stoic’?

Is Kirk (Rank = 4) a ‘Large Ham’, or a ‘The Stoic’?

http://tvtropes.org/pmwiki/pmwiki.php/Main/LargeHam
http://tvtropes.org/pmwiki/pmwiki.php/Main/TheStoic

# Negation

```
Ham(O) :=
   Officer(O,S,R), R = 4, NOT Stoic(O)
Stoic(O) :=
   Officer(O,S,R), NOT Ham(O)
```

Is Spock (Rank = 3) a 'Large Ham', or a 'The Stoic'?

Is Kirk (Rank = 4) a 'Large Ham', or a 'The Stoic'?

This program has TWO fixpoints!

http://tvtropes.org/pmwiki/pmwiki.php/Main/LargeHam
http://tvtropes.org/pmwiki/pmwiki.php/Main/TheStoic

# Stratification

- Table R depends on Table S if R is in the head of a rule with table S in its body.
  - R depends negatively on S if R is in the head of a rule with NOT S in its body.
- Classify tables into layers (or strata):
  - Stratum 0 has all tables with no dependencies
  - Stratum 1 depends on tables in S1, or depends negatively on tables in S0.
  - Stratum 2 depends on tables in S2 or negatively on S1…

# Stratification

- A Stratified Program is a program that can be organized into strata.

- Is LargeHam/TheStoic a Stratified Program?
- Does Stratification ensure a single fixpoint?
  - Why/Why not?

# Term Recap

# Relational Algebra

- Basic Operators: $\sigma$, $\pi$, $\times$, $\bowtie$, /
  - Compositional: Can combine operators.
  - Operational: Expresses an evaluation strategy
- Equivalent to SQL.
- A query can be expressed in many ways.
  - Rewrite rules, or equivalencies in RA.
  - Different costs for different plans.

# Storage

- Seek time + Rotational Delay = Latency
- Transfer Time → Bandwidth
- Tradeoffs of Flash vs Hard Disk vs Memory
- Data Layouts: Per Page, Per Record
- RAID Levels: Layout per Disk

# External Algorithms

- Pipeline (Iterators) to avoid disk wherever possible.
    - (keep a small working set at all times)
    - What operations can be pipelined?
- Group-By Aggregates
- External Sort (+Replacement Sort)
- Joins: Hash, Sort/Merge (+E.Sort),

# Indexing

- Static Indexes: ISAM, Hash
- Dynamic Indexes
  - B+ Tree (Node Merge/Node Join)
  - Extendible Hashing
  - Linear Hashing
  - Consistent Hashing (Chord)
- Picking an appropriate Access Path.

# Cost Estimation

- Row Size, Selectivity/Reduction Factor
- IO Costs (Index/File Scans, Joins)
- CPU Costs (# Tuples Materialized)
- Memory Costs (Joins)
- Estimating the Selectivity of a Predicate
  - Building/Using Histograms, Sketches

# Optimization

- Pipelined, Left-deep, etc… Plans.
- Applying RA Rewrite Rules.
  - Push-down Projection, Selection ops.
  - Constructing, Reordering Joins.
- Selecting Appropriate Access Paths:
  - I-NLJ, Index Scan, File Scan, …
- Using Constraints (e.g., Foreign Key Constraints)

# Data Modeling

- E-R Model
  - Key-, Participation-, ISA-Constraints
  - Weak Entities
  - ER Aggregation
- Relationship to SQL
  - Applying E-R Constraints in a SQL DB.
  - Enforcing Constraints in a SQL DB (ON DELETE…)

# Transactions

- The A.C.I.D. Guarantees
- Transaction Schedules, Equivalence
  - Conflict-…, View-…
- Dependency Graphs
- Enforcement: Lock-Based CC
  - Avoiding Deadlock, Hierarchical Locking

# Transactions

- Enforcement: Optimistic CC
  - Validation Algorithms; What, Why?
- Enforcement: Timestamp CC
  - Write/Read Validation
  - Ignoring Out-of-date Writes (Thomas W.R.)
  - Versioned Databases

# Recovery

- When/How is a transaction “committed”
- Write-ahead-logging
    - Handling Transaction Aborts.
    - Support for Crashes During Abort.
    - Checkpointing.
    - ARIES: REDO/UNDO

# Parallel DBs

- Sequential vs Partition Parallelism
  - … and how they interact with RA Ops.
  - … especially sort, join, and aggregates.
- Shared… Memory vs Disk vs Nothing
- Partitioning Strategies
- Optimizing Parallel DB Queries.

# Distributed Xacts

- Bloom Joins
- Providing Isolation, Durability, Consistency.
- Distributed Deadlock Detection.
- 2-Phase Commit
- Recovery from failures
  - Transient… Node Failure, Link Failure

# Data Warehousing

- MOLAP vs ROLAP.
- Data Cubes; CUBE operator.
- Sequence Analysis
    - WINDOW operator

# Approximation Algos

- Sketches
  - Bloom Filters
  - Flajolet&Martin Count-Distinct Sketches
  - Count Sketch
- Online Aggregation
  - Sampling, Ripple Joins, Index Striding

# Streams & IVC

- Window Joins
    - Half-Joins (Tuple Invalidation)
- Materialized Views
- Delta Queries (Simple RA Deltas)

# Column Stores

- Column-Stores vs Row-Stores
- Data Layout (Column Optimizations)
    - Sorts, Compression
- Query Processing
    - Set Intersections
- Database Cracking

# Schema Refinement

- Functional Dependencies
  - Inference rules for FDs
- Decomposition (Lossless, Dependency-Preserving)
- Normal Forms (BCNF, 3NF)
  - Definition, Decomposition Into
- Minimal Cover

# Datalog/Recursion

- Queries/Inference Programs in Datalog
- Fixpoints, Models; Least… Model vs Fixpoint
- Variable, Rule Safety (Range Restrictions)
- Negation; Stratification
    - When is Negation ok in a program?