Isn't Implementing a Database System Simple ?

- Based on novel GOOFIE algorithms
- *Storage Structure*:
  - each relation is a file
  - each tuple is a line
- *Query Processing*:
  - DML statements reduced to C code by preprocessor
  - queries reduced to nested loops
- *Transaction Management*:
  - huh ?


```
Wild;Lynch;Winger\ nSky;...
```

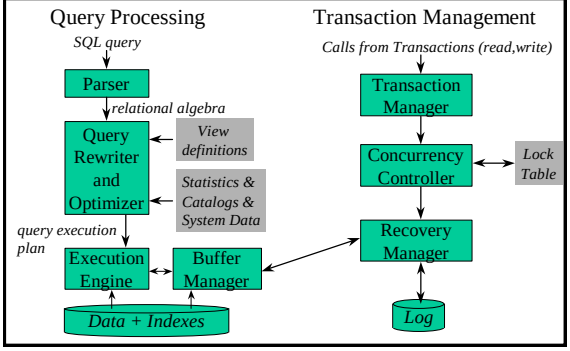
```
SELECT t.Actor FROM Movie t,s
WHERE t.Title=s.Title
        AND s.Actor="Winger"

read Movie file; for each line t
        read Movie file; for each line s
                create joint tuple t;s
                check condition
                output if OK
```

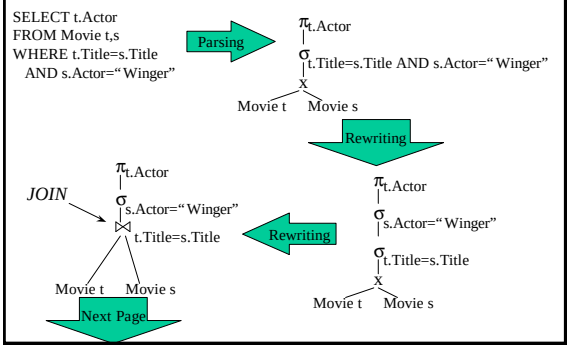

GOOFIE's Future is Bleak

- Absence of *indexes*
  - can not quickly find a tuple with given value for Actor
- Brute Force *Query Processing;* No *Optimizer*
  - could join the two relations using index on Title
  - could sort the two relations and require only one pass
  - could use main memory effectively (*buffer management*)
- *Not reliable* under *concurrency* and *failures*
  - what if system fails before insertion is completed ?
  - mutual exclusion is not guaranteed

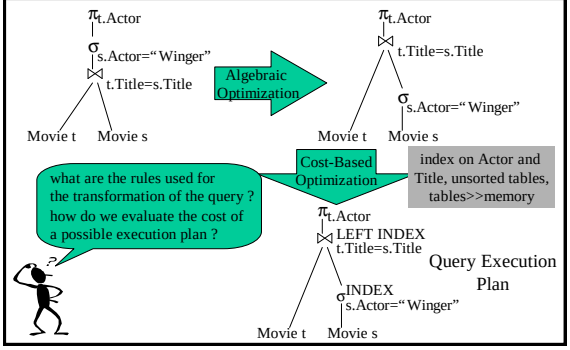
Database System Architecture



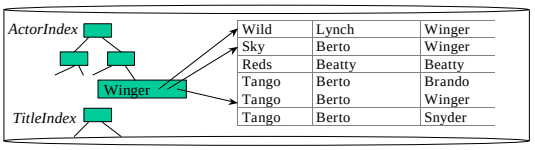
The Journey of a Query (Example)



The Journey of a Query (cont'd)



The Journey of a Query (cont'd)



EXECUTION ENGINE

find "Winger" tuples using Actorindex
  for each "Winger" tuple
    find tuples t with the same title
    using TitleIndex
    project the attribute Actor of t

How is the table arranged on the disk ? Are tuples with the same Actor value clustered (consecutive) ? What is the exact structure of the index (tree, hash table,…) ?

Course Overview

- File Structure
- Indexing
- Query Optimization
- Concurrency Control
- Recovery
- Beyond transactional centralized databases