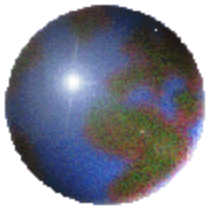
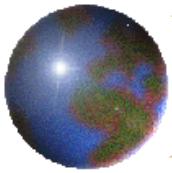


Query Optimization

Dr. Geetha Mary A
Associate Professor,
SCSE, VITU



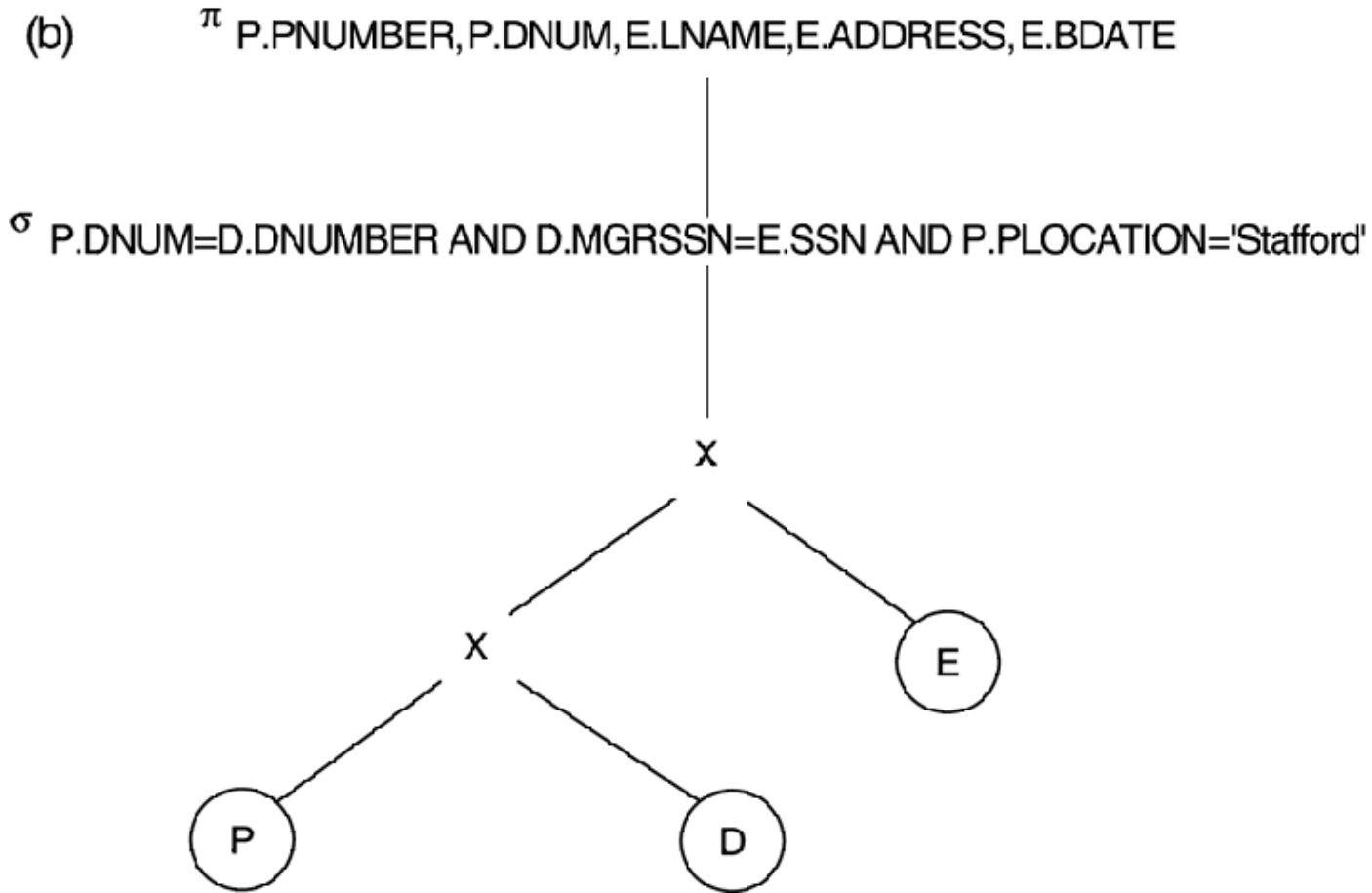
Using Heuristics in Query Optimization

- The parser first generates an initial internal representation, the uses heuristic rules to optimize
- One of the main **heuristic rules** is to apply the unary operations ‘σ’ and ‘π’ before ⋈ or other binary operations
- A **query tree** is a tree data structure that represents the input relations of the query as **leaf nodes** and the relational algebra operations as **internal nodes**.

Initial (canonical) query tree for SQL query

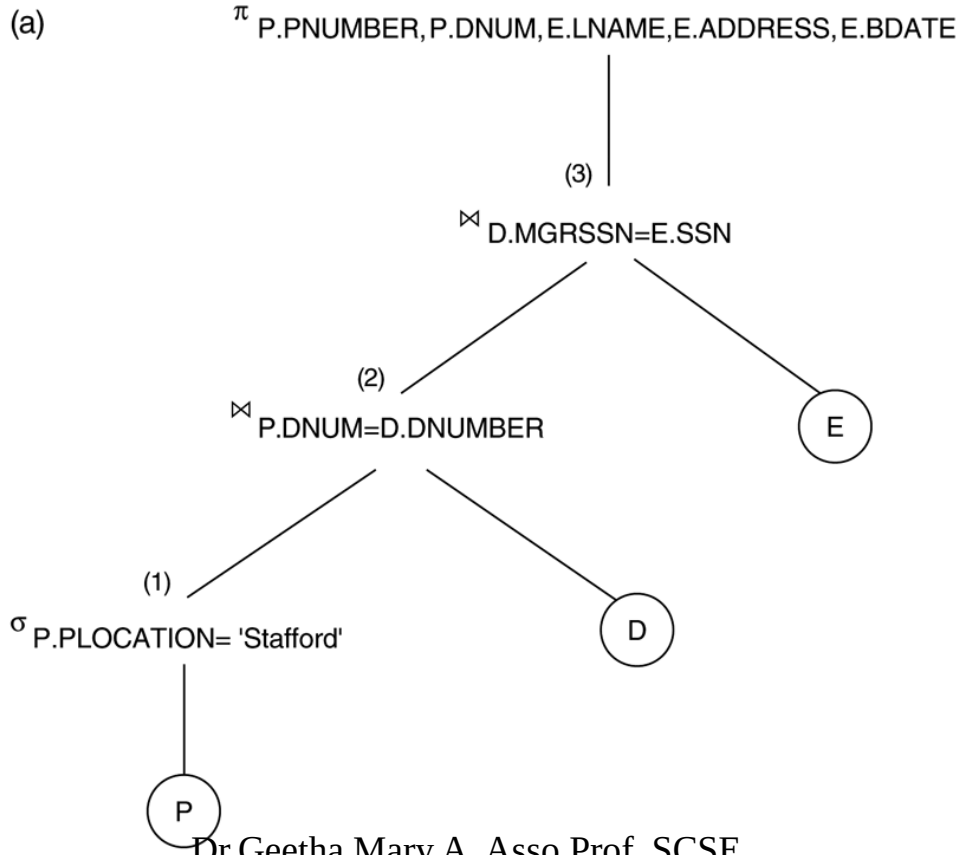
SELECT PNUMBER, DNUM, LNAME, ADDRESS, BDATE
FROM PROJECT, DEPARTMENT, EMPLOYEE
WHERE DNUM=DNUMBER AND MGRSSN=SSN AND
PLOCATION='Stafford'

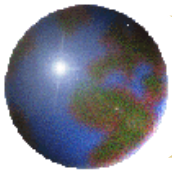
(b)



Query tree corresponding to the relational algebra expression for the SQL query

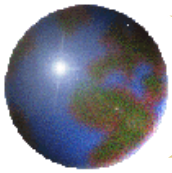
SELECT PNUMBER, DNUM, LNAME, ADDRESS, BDATE
FROM PROJECT, DEPARTMENT, EMPLOYEE
WHERE DNUM=DNUMBER AND MGRSSN=SSN AND PLOCATION='Stafford'





Using Heuristics in Query Optimization

- Execution of the query tree:
  1. Execute an internal node operation whenever its operands are available and then replace the internal node by the resulting operation.
  2. Repeat step 1 as long as there are leaves in the tree, that is, the execution terminates the root node is executed and produces the result relation for the query.
- A more natural representation of a query is the **query graph** notation.



Using Heuristics in Query Optimization

- Example of Transforming a Query:

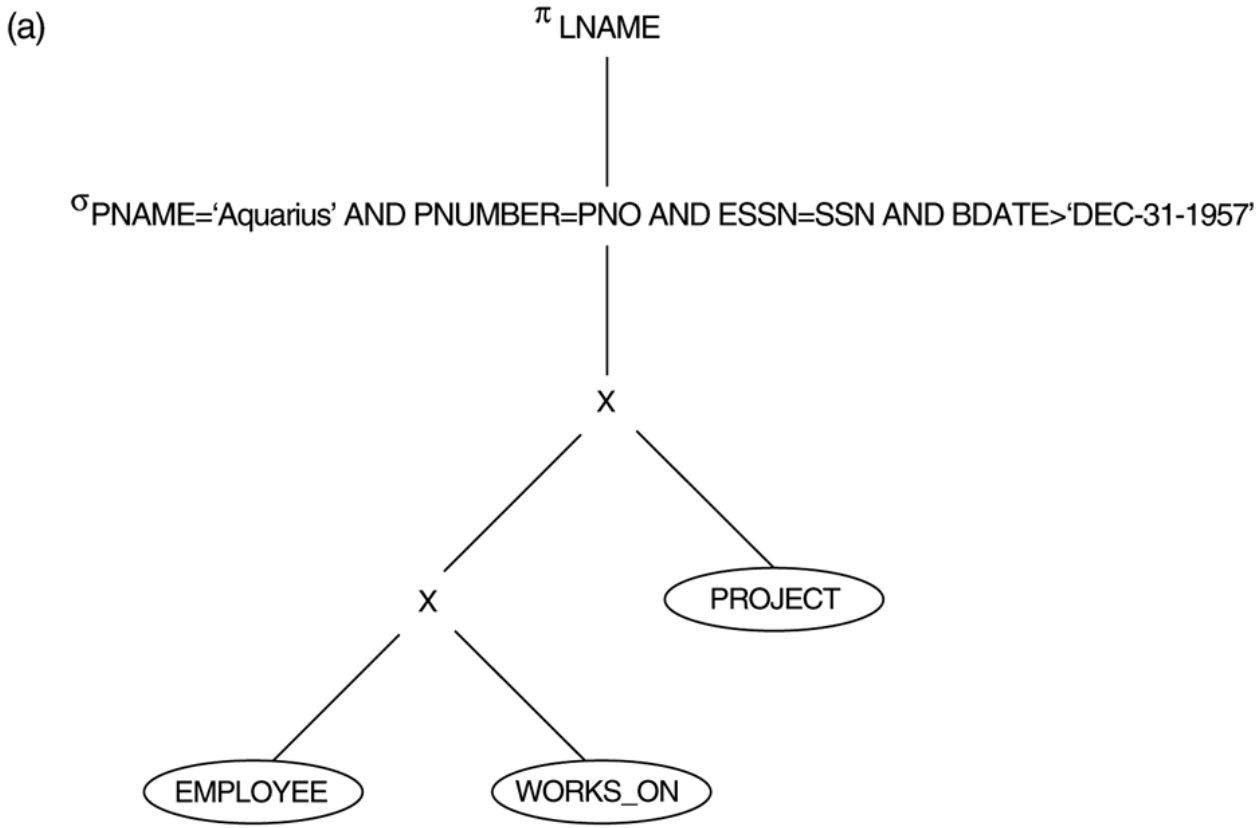
  Consider the query Q that states “Find the last names of employees born after 1957 who work on a project named ‘Aquarius’.”

In SQL, this query can be specified as:

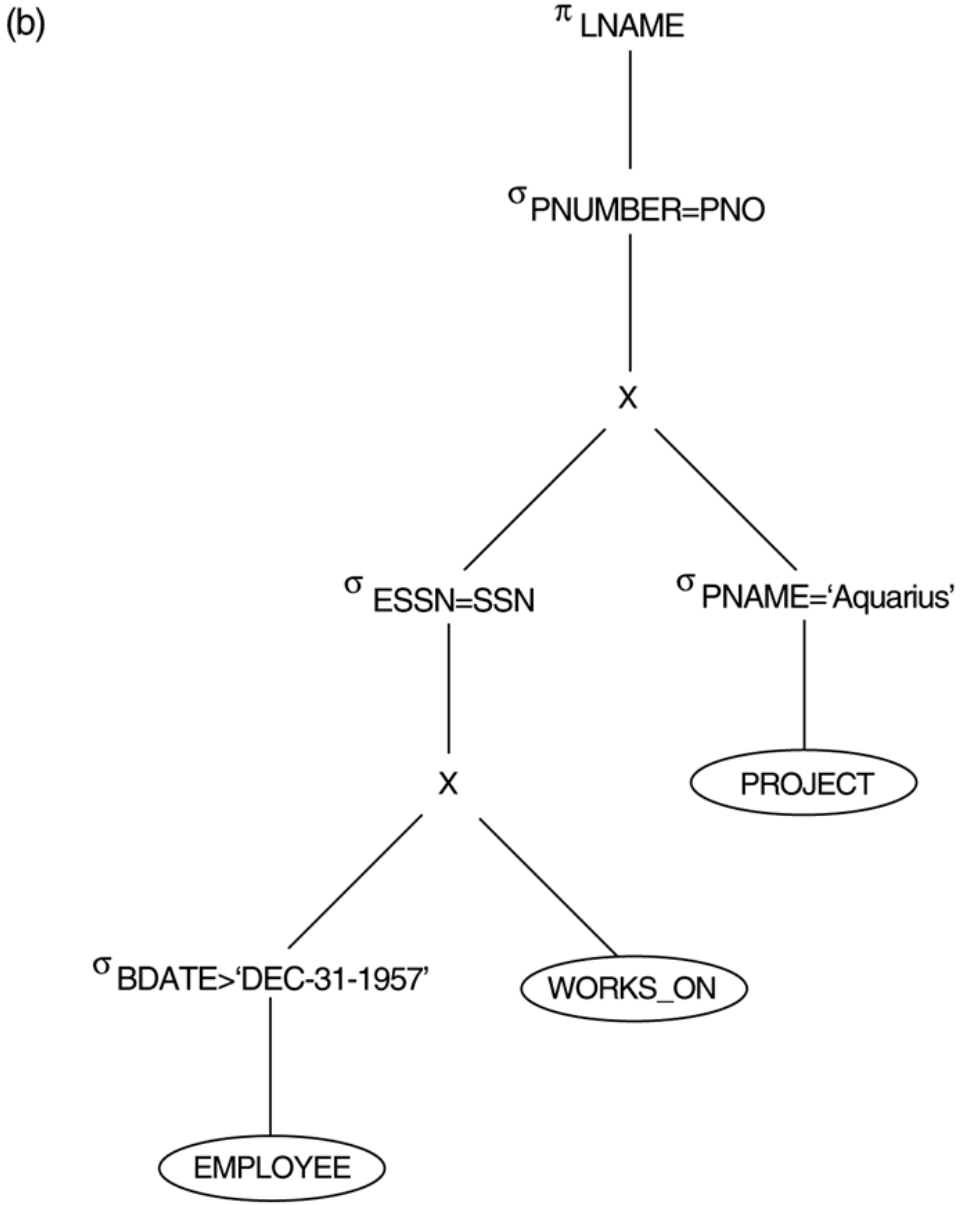
```
SELECT   LNAME  FROM   EMPLOYEE, WORKS_ON, PROJECT
WHERE    PNAME=‘Aquarius’ AND ESSN=SSN
                  AND PNUMBER=PNO
                  AND BDATE > ‘1957-12-31’;
```

Steps in converting a query tree during heuristic optimization.

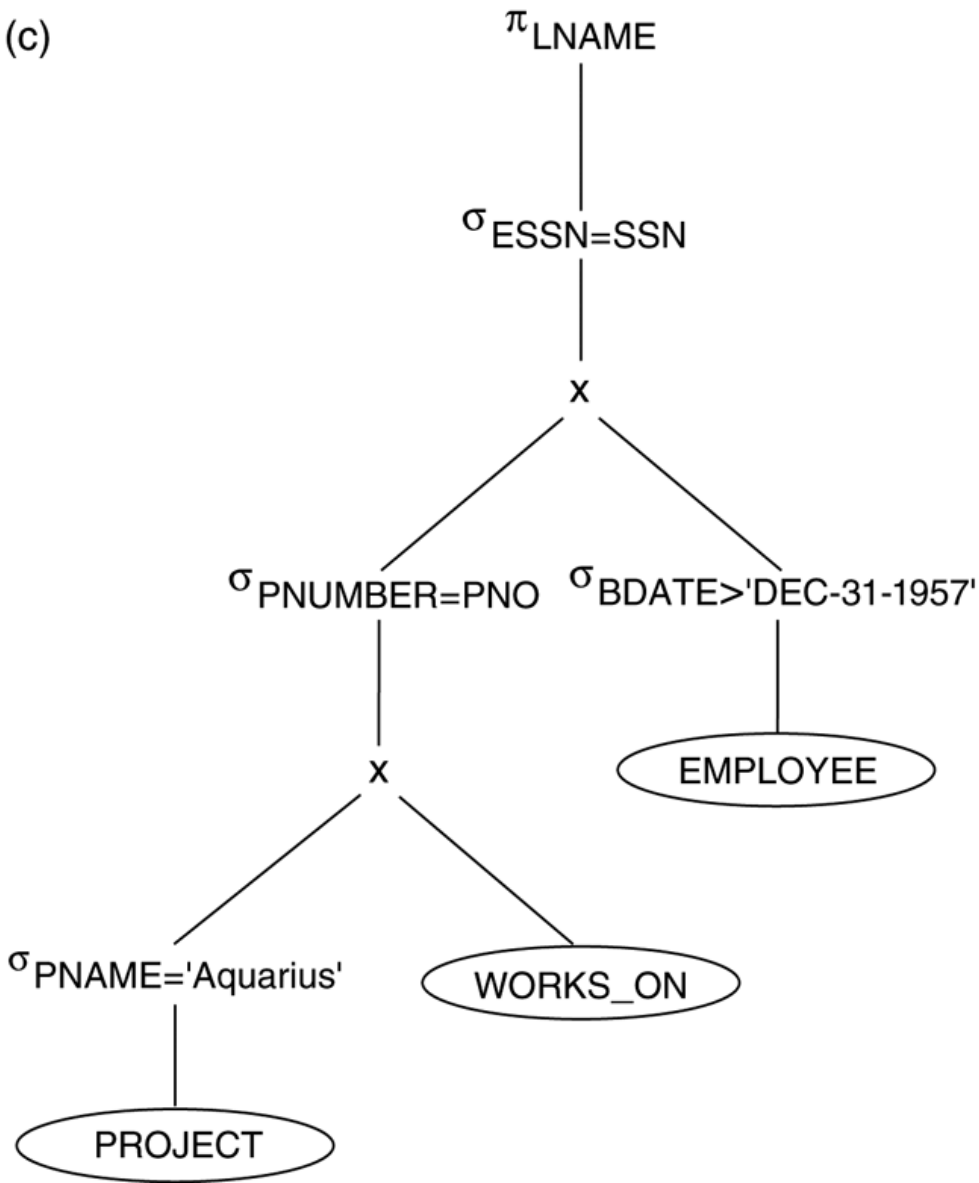
(a) Initial (canonical) query tree for SQL query Q.



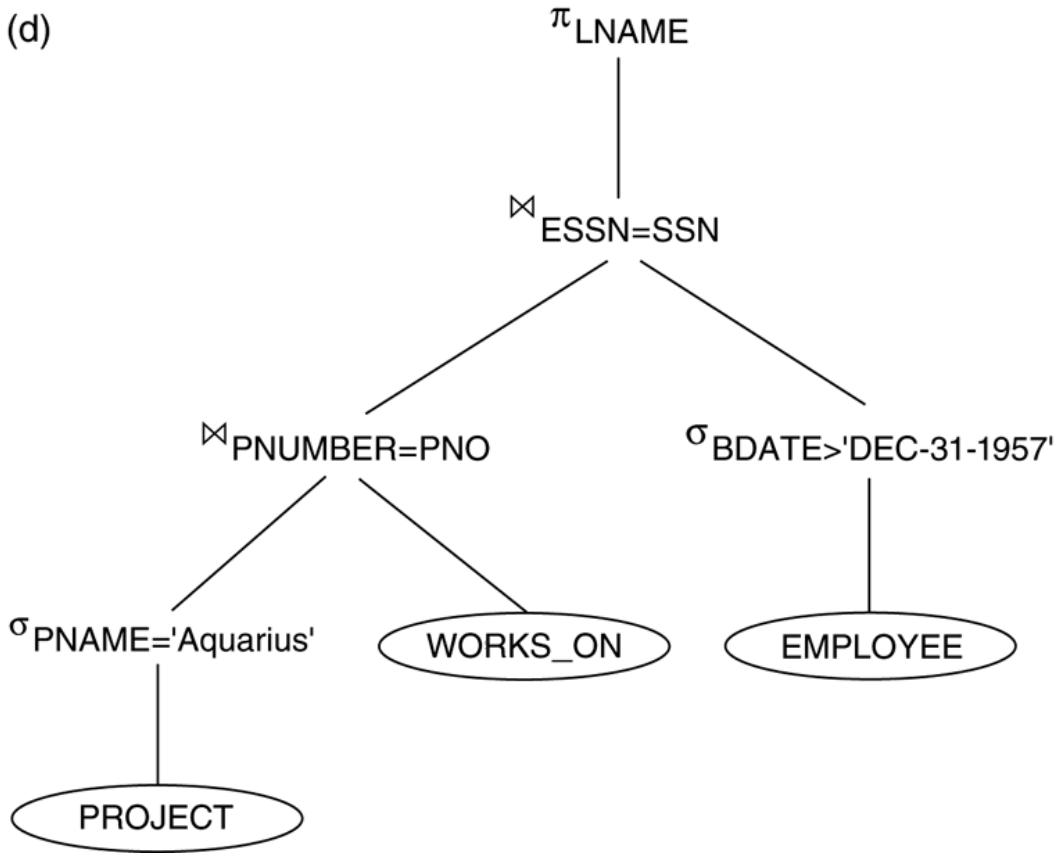
Moving SELECT operations down the query tree.



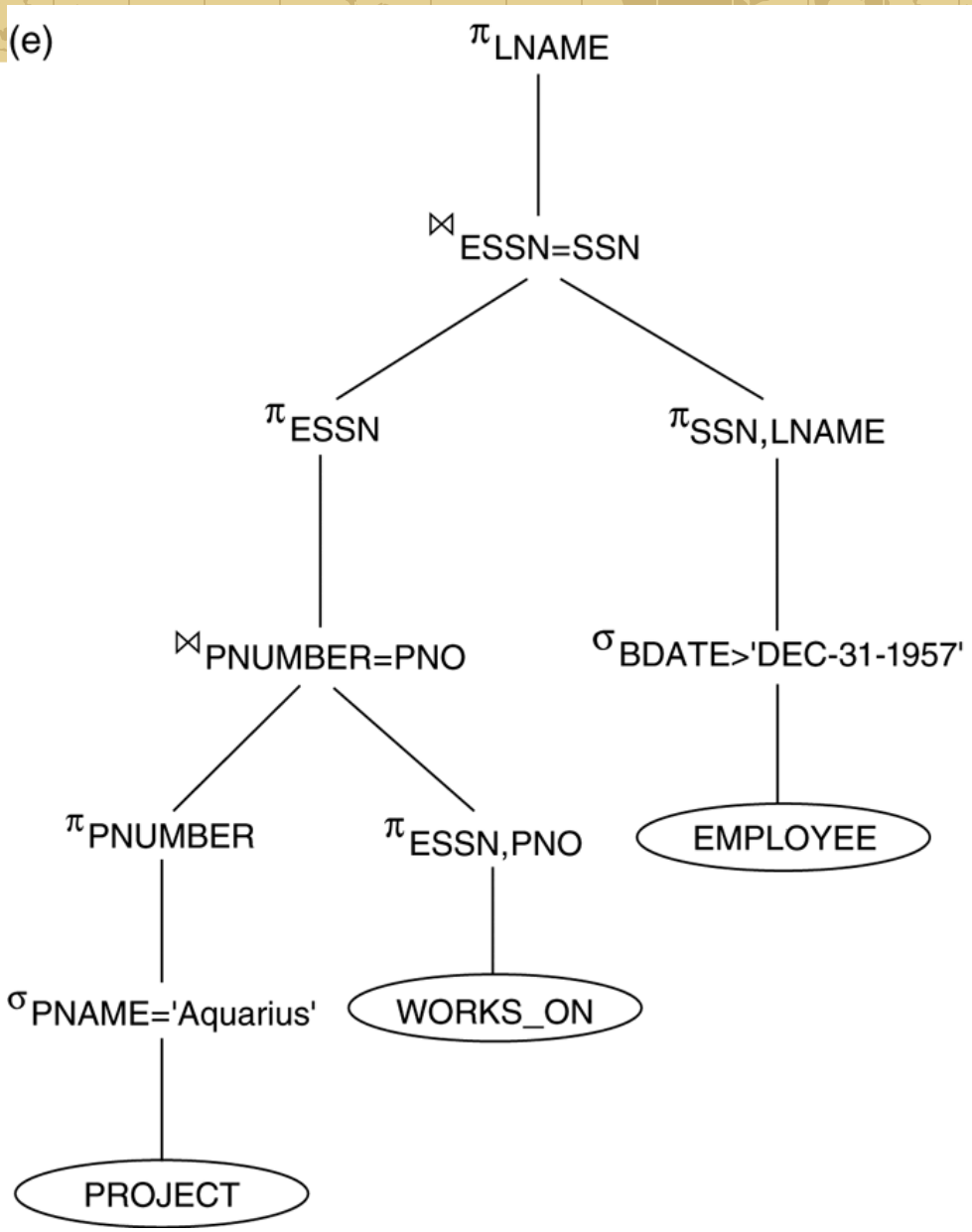
Applying the more restrictive SELECT operation first.

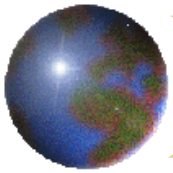


Replacing CARTESIAN PRODUCT and SELECT with JOIN operations.



Moving PROJECT operations down the query tree.





Outline of a Heuristic Algebraic optimization Algorithm

- Break up conjunctive selection condition, that is,

$\sigma_{<c1 \text{ AND } c2 \text{ AND } \ldots \text{ AND } cn>}(R) \equiv \sigma_{<c1>}(\sigma_{<c2>}(\ldots(\sigma_{<cn>}(R))\ldots))$

- Move each 'σ' operation as far down the query tree as is permitted by the attribute involved in the 'σ' condition
- Rearrange the leaf nodes of the tree using;

• Position the leaf node relation with the most restrictive σ operations so they are executed first,

• Make sure that the ordering of leaf nodes does not cause CARTESIAN PRODUCT operations

- Combine a X with a subsequent σ in the tree into a ⋈
- Break down and move lists of projection attributes down the tree as far as possible
- Identify subtrees that represent groups of operations that can be executed by a single algorithm.