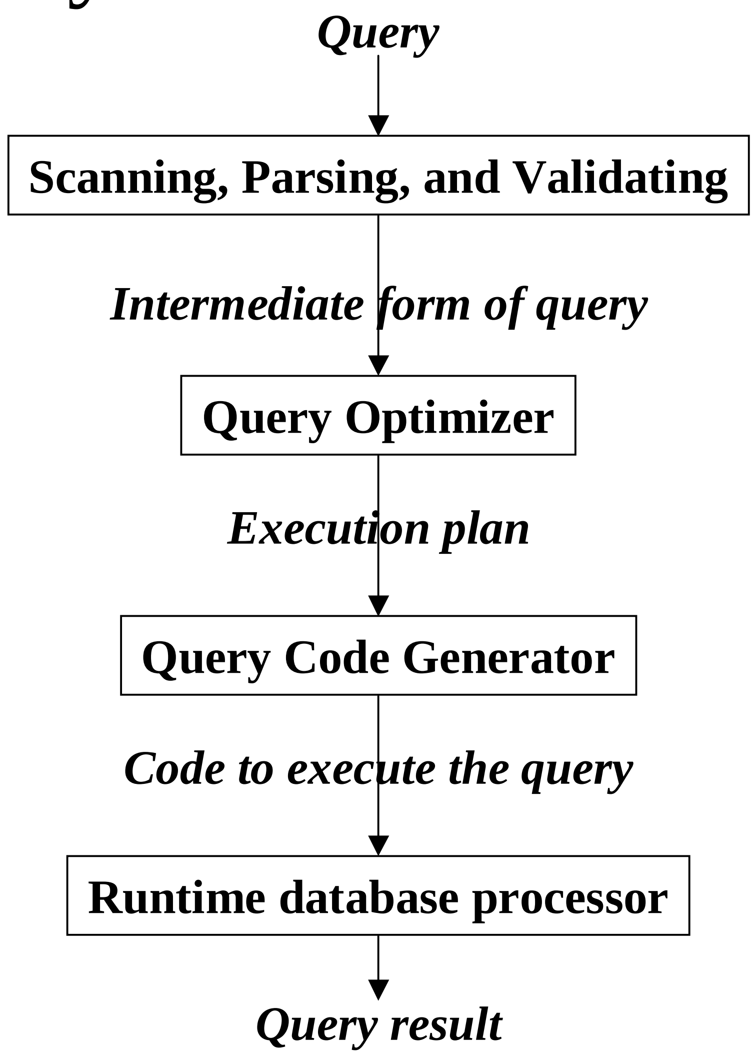
CSC 742
Database Management Systems

Topic #16: Query Optimization

Agenda

- Typical steps of query processing
- Two main techniques for query optimization
  - Heuristics based query optimization
  - Cost based query optimization
- Translating SQL queries into relational algebra
- Heuristics based query optimization

Typical Steps when Processing A High-level Query



Questions

- Difference between high-level query languages and high-level programming languages in terms of queries?
  - SQL v.s. C

- Can we find the best execution plan?

- Do we always use the best execution plan?

Two Main Techniques for Query Optimization

- Heuristic Rules
  - A heuristic is a rule that works well in most of cases, but not always.
- Cost based query optimization
  - *Estimate* the cost for each execution plan, and choose the one with the lowest cost.
- Can we get the best execution plan?

Translating SQL queries into extended relational algebra

- SQL queries are decomposed into query blocks
  - Each query block contains a single SELECT – FROM – WHERE expression as well as GROUP BY and HAVING clauses.
  - Nested queries within a query are identified as separate query blocks.
- Query optimization
  - Choose an execution plan for each block

```
SELECT LNAME, FNAME
FROM EMPLOYE
WHERE SALARY > (SELECT MAX(SALARY)
                FROM EMPLOYEE
                WHERE DNO=5);
```

```
SELECT LNAME, FNAME
FROM EMPLOYE
WHERE SALARY >C
```

```
SELECT MAX(SALARY)
FROM EMPLOYEE
WHERE DNO=5
```

```
SELECT DNAME, MAX(SALARY) AS MAX_SALARY
FROM EMPLOYE
GROUP BY DNO, DNAME
HAVING DNO > 3
```

Basic Operations

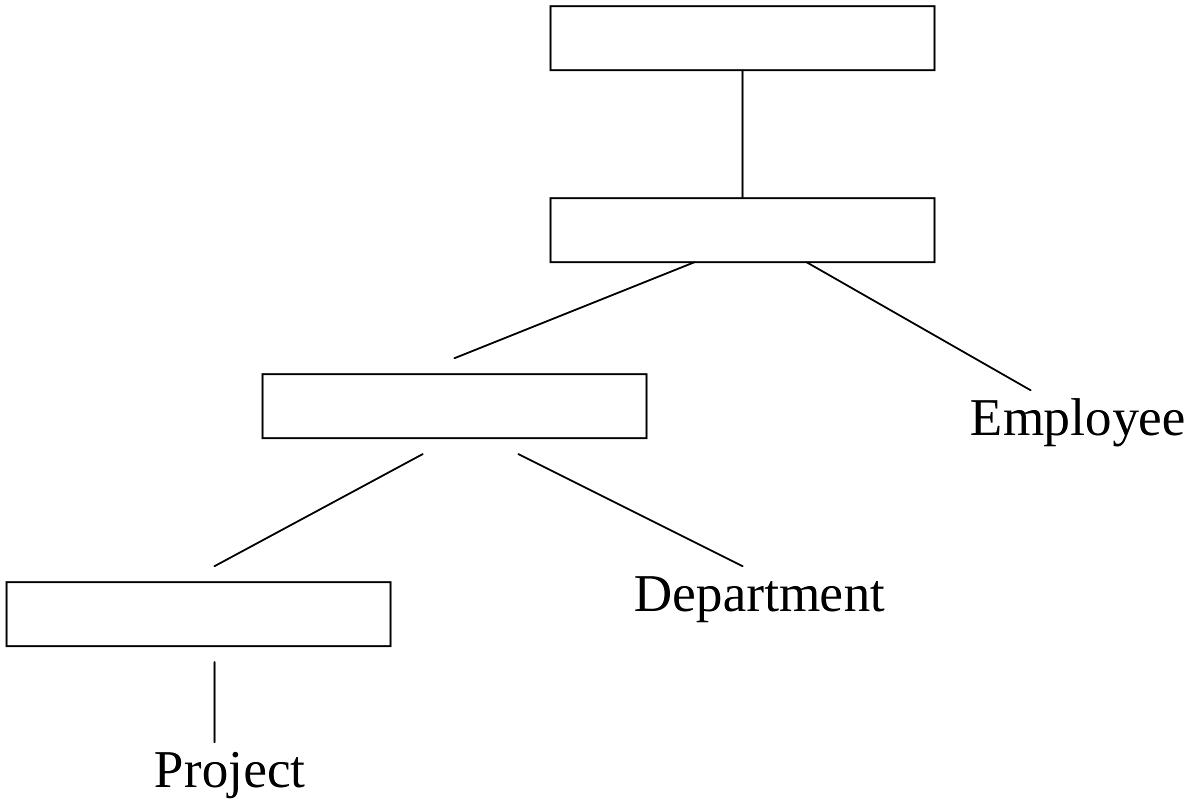
- DBMS implements basic algorithms for certain operations
  - Select
  - Project
  - Join
  - Cartesian Product
  - Set operations
  - …
- These operations are combined to form the execution plans.

Query Tree

- A query tree is a data structure that corresponds to a relational algebra expression.
  - Input relations are leaf nodes of the tree
  - Relational operations are internal nodes.
  - An execution of the query tree consists of
    - executing an internal node operation whenever its operands are available,
    - and then replacing the node by the relation that results from executing the operation.

For every project located in ‘Stafford’, retrieve the project Number, the controlling department number, and the department Manager’s last name, address, and birthdate.

$$\pi_{\text{PNUMBER, DNUM, LNAME, ADDRESS, BDATE}}(((\sigma_{\text{PLOCATION='Stafford'}}(\text{PROJECT})) \bowtie_{\text{DNUM=DNUMBER}}(\text{DEPARTMENT})) \bowtie_{\text{MGRSSN=SSN}}(\text{EMPLOYEE}))$$



Using Heuristics in Query Optimization

- An example of a heuristic
  - Apply SELECT before join.
  - Questions:
    - Does it work in most of cases?
    - Exception?

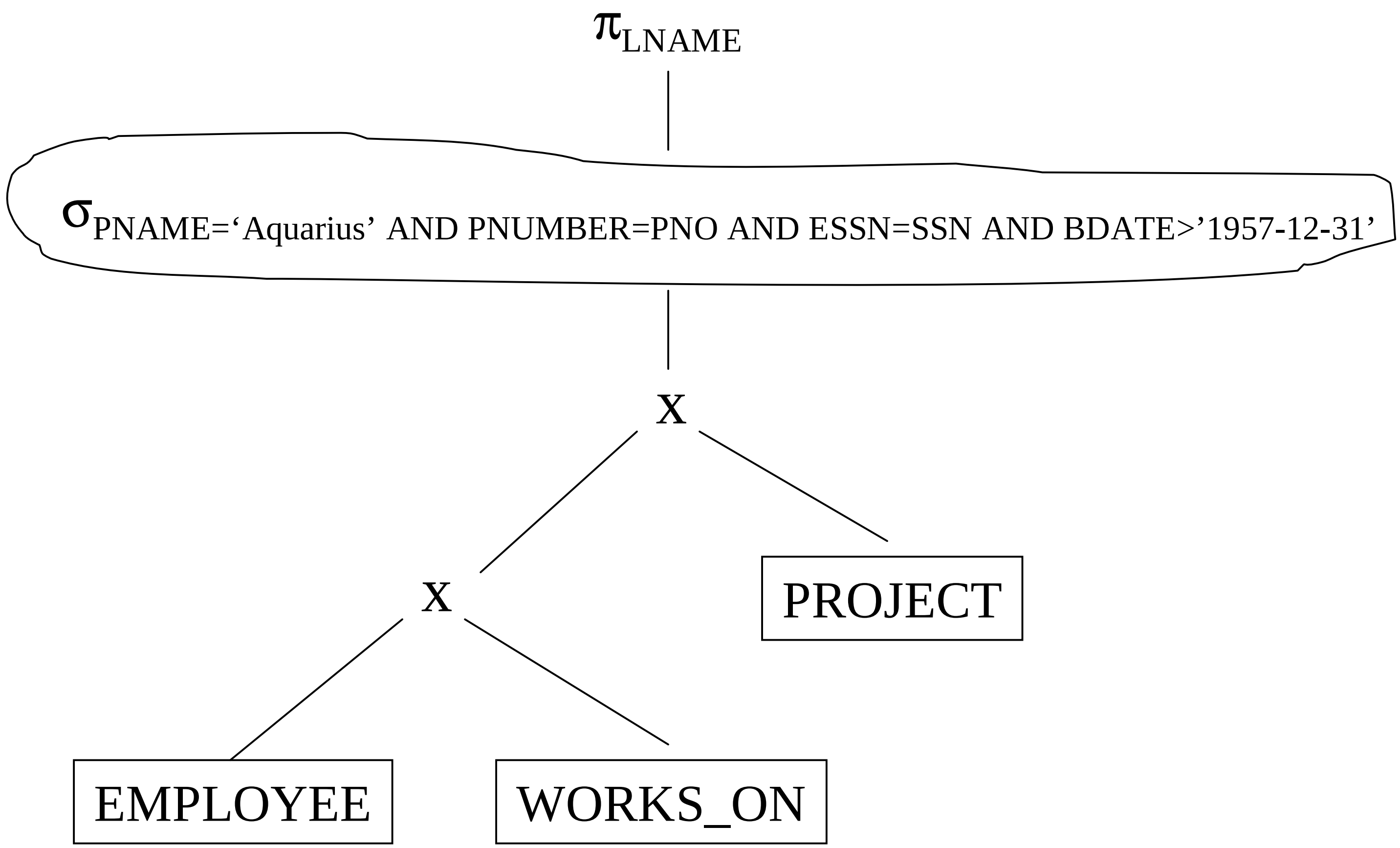
Using Heuristics in Query Optimization

- General idea
  - Many different relational algebra expressions (and thus query trees) are equivalent.
  - Transform the *initial query tree* of a query into an equivalent *final query tree* that is efficient to execute.

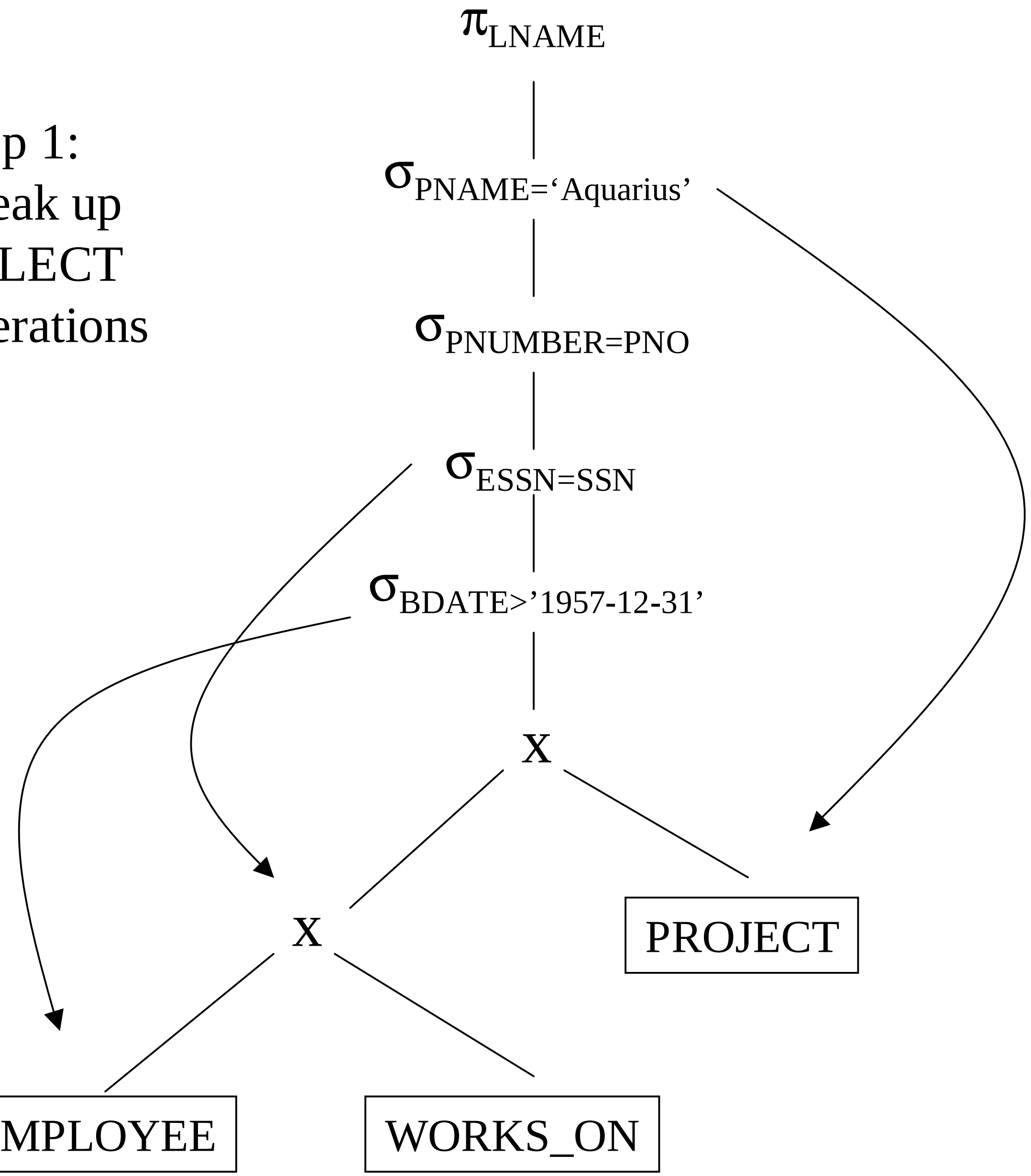
Algorithm

- Break up any SELECT operations with conjunctive conditions into a cascade of SELECT operations.
- Push SELECT operations as far down the query tree as possible
- Rearrange binary operations so that
  - the most restrictive SELECT operations are executed first
  - Avoid CARTESIAN PRODUCT
- Try to combine a CARTESIAN PRODUCT with a SELECT operation into a join operation.
- Break up PROJECT operation and move lists of projection attributes as down the tree as possible by creating new project operations.
- Identify sub-trees that represent groups of operations that can be executed by a single algorithm.

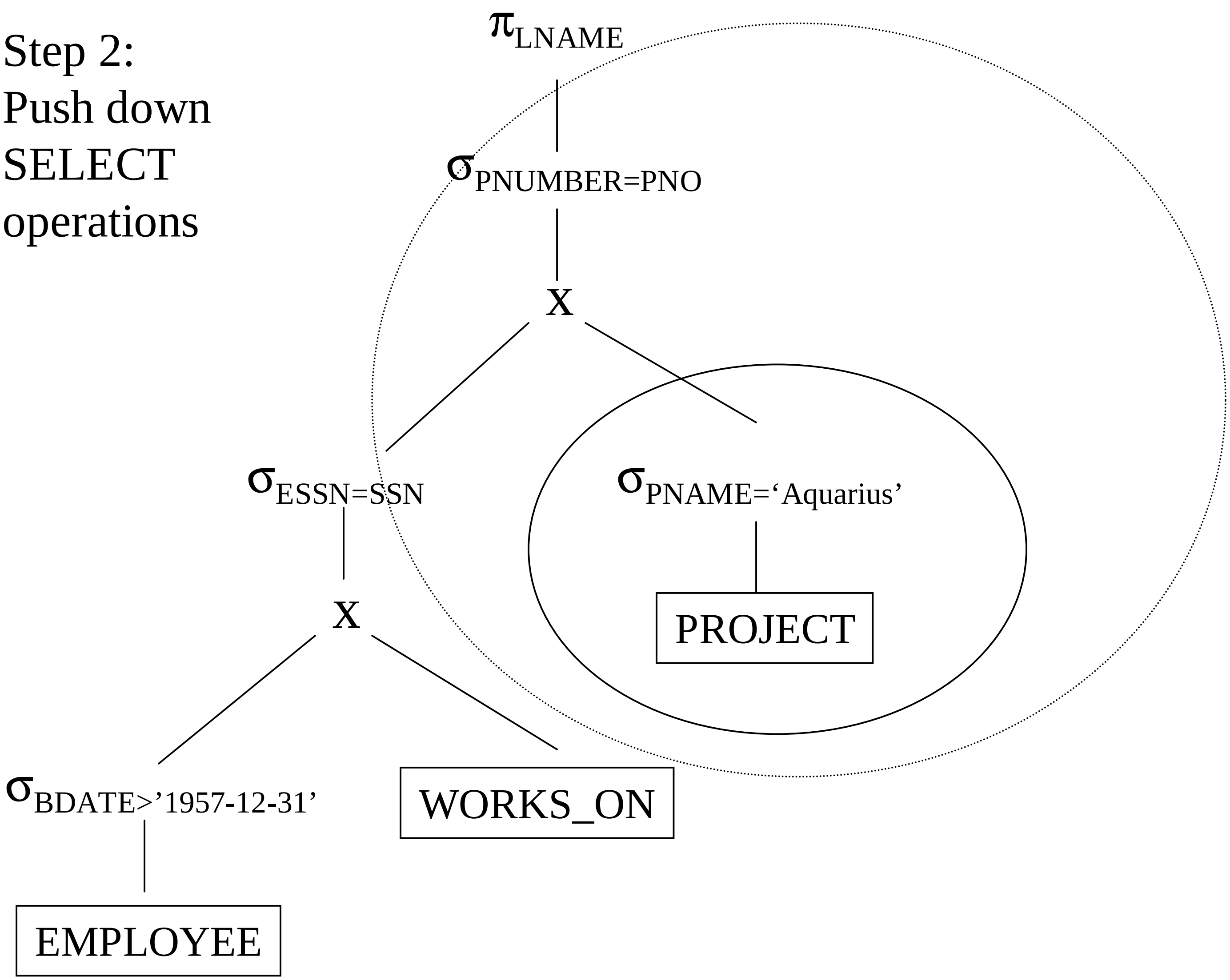
```
SELECT LNAME
FROM EMPLOYEE, WORKS_ON, PROJECT
WHERE PNAME='Aquarius' AND PNUMBER=PNO
         AND ESSN=SSN AND BDATE>'1957-12-31'
```



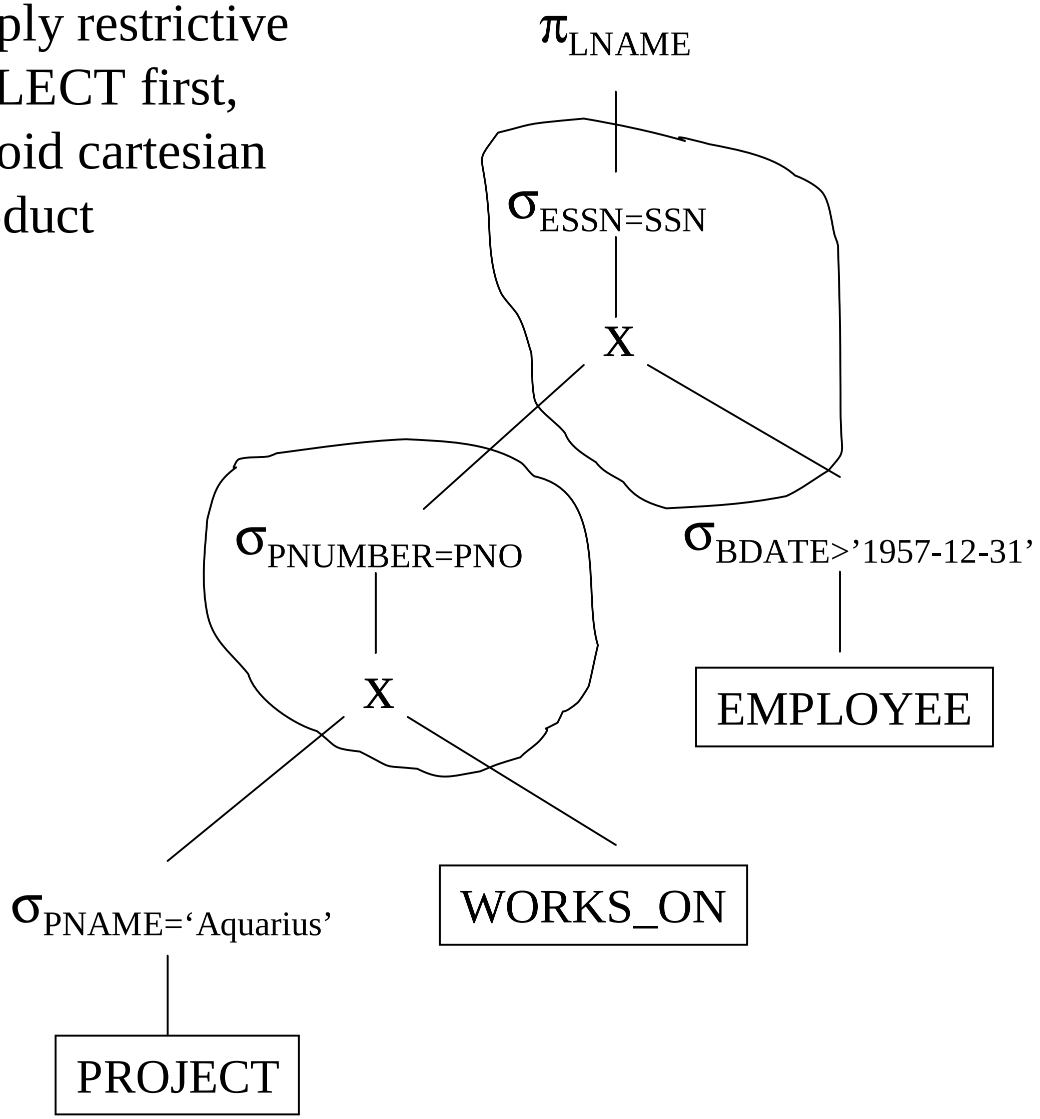
Step 1:
Break up
SELECT
operations



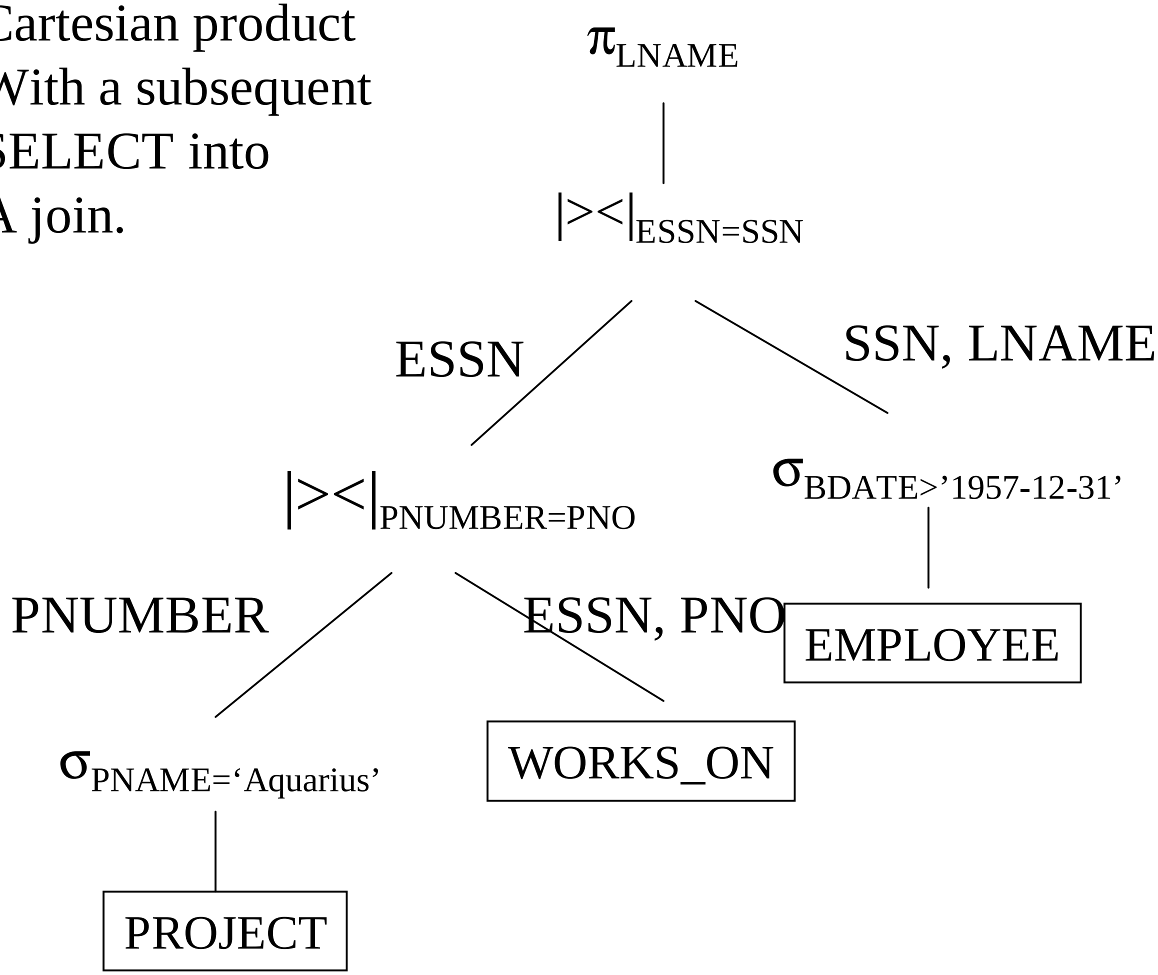
Step 2:
Push down
SELECT
operations



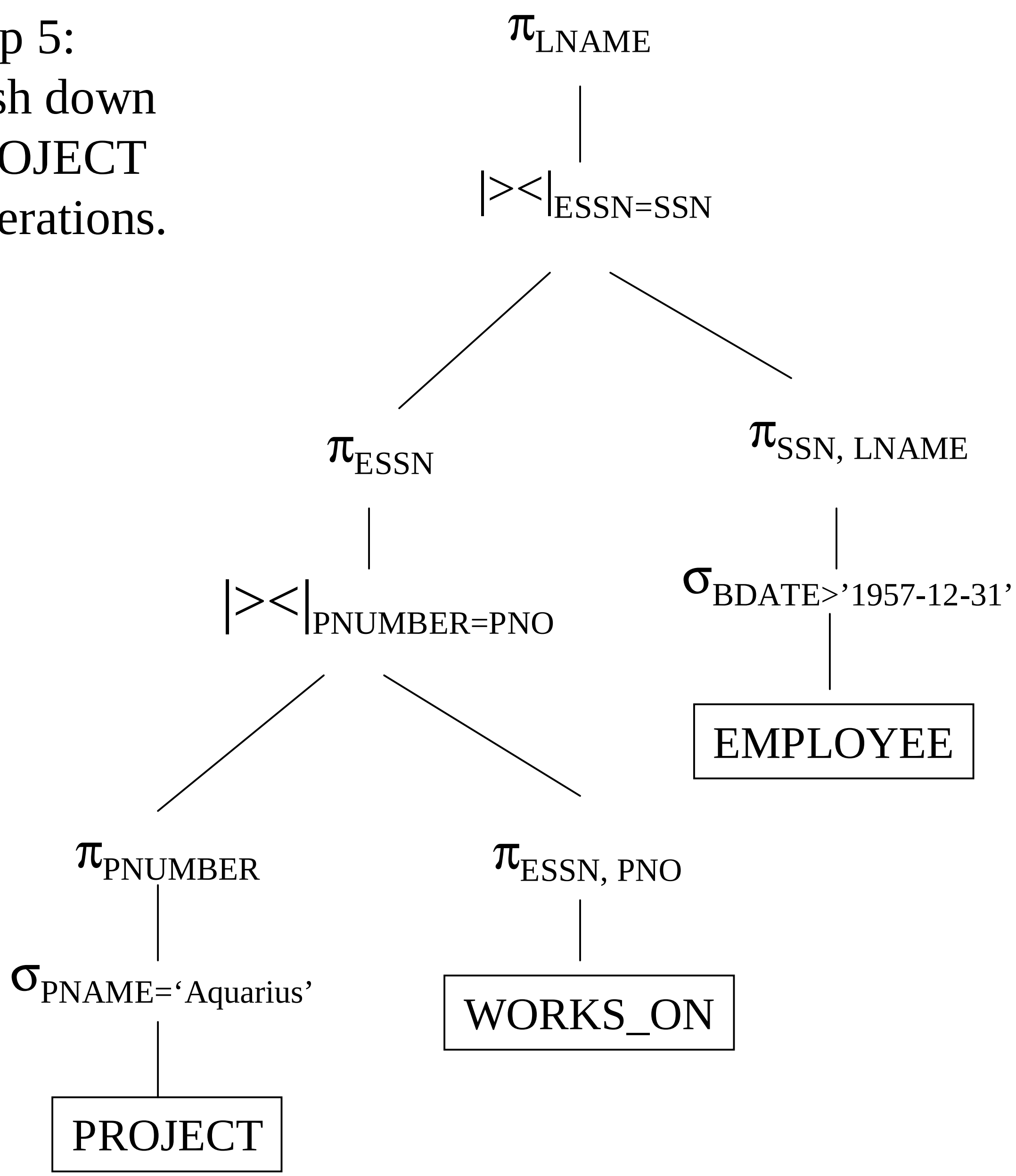
Step 3:
Apply restrictive SELECT first,
Avoid cartesian product



Step 4:
Combine a
Cartesian product
With a subsequent
SELECT into
A join.



Step 5:
Push down
PROJECT
Operations.



Step 6:
Identify sub-trees