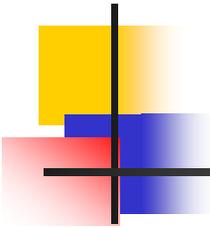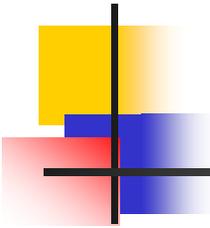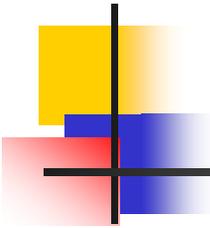# Query Processing and Optimization

Objectives

- Introduction +
- Major Phases in Query Processing +
- Optimizing Queries +
  - Steps in Query Optimization +
  - Query Trees and Graphs +
  - Relational Algebra Transformation +
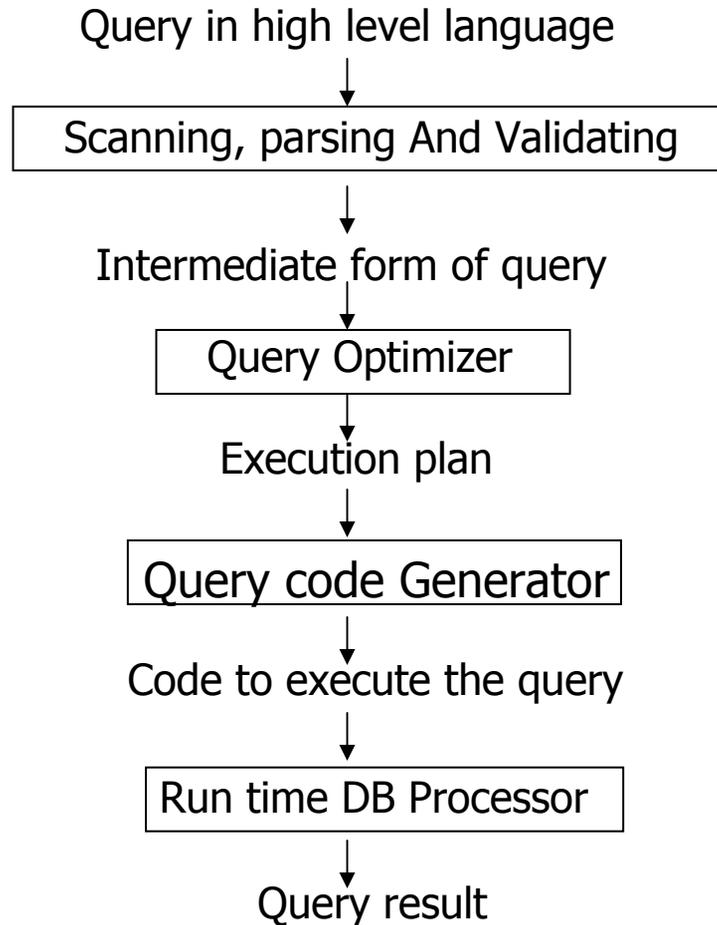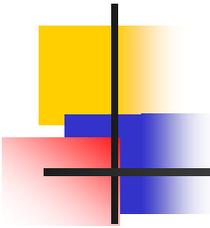 - Query Processing and Optimization +

- Introduction

- With declarative languages such as SQL, the user specifies what data is required rather than how it is retrieved

- Relieves user of knowing what constitutes a good execution strategy

- Also gives the DBMS more control over system performance.

-- Processing a High-Level Query

Query in high level language

↓

| Scanning, parsing And Validating |

↓

Intermediate form of query

↓

| Query Optimizer |

↓

Execution plan

↓

| Query code Generator |

↓

Code to execute the query

↓

| Run time DB Processor |

↓

Query result

- Major Phases in Query Processing

- Scan, Parse, and Validate Query: Transform query written in high level language (e g. SQL), into correct efficient execution strategy expressed in low-level language (e. g. Relational Algebra)

- Optimize query and generate code: produce an execution strategy or plan and run-time code.

- Execute query: Implement the plan to retrieve required data.
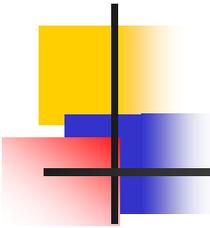
- Optimizing Queries

- The objective in query optimization is to select an efficient execution strategy.

- As there are many equivalent transformations from same high-level query, aim is to choose the one that minimizes resource usage.

- Generally, an efficient execution plan reduces the total execution time of a query; thereby reducing the response time of a query.

- The problem is computationally intractable with large number of relations, so the strategy adopted is reduced to finding a near optimum solution.

- Two main techniques for query optimization are:
  - Heuristic rules that order operations in a query
  - Comparing different strategies based on relative cost, and selecting one that minimizes resource usage.
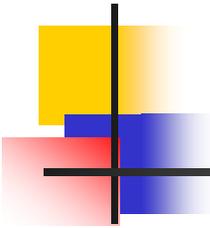
- Steps in Query Optimization

- In using heuristics during query optimization, the following steps must be followed:

    1. Translate queries into query trees or query graphs

    2. Apply transformation rules for relational algebra expression.

    3. Perform heuristic optimization
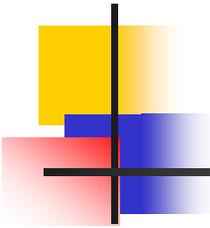
-- Query Trees and Graphs

- **A query tree** is a tree data structure that corresponds to a relational algebra expression. It represents the input relations of the query as leaf nodes of the tree, and represents the relational algebra operations as internal nodes

- An execution of the query tree consists of executing an internal node operation whenever its operands are available and then replacing that internal node by the relation that results from executing the operation.

- The execution terminates when the root node is executed and produces the result relation for the query.
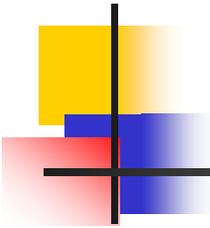
--- Query Tree Example ...

- For every project located in 'Stafford', list the project number, the controlling department number and the department manager's last name, address, and birth date.
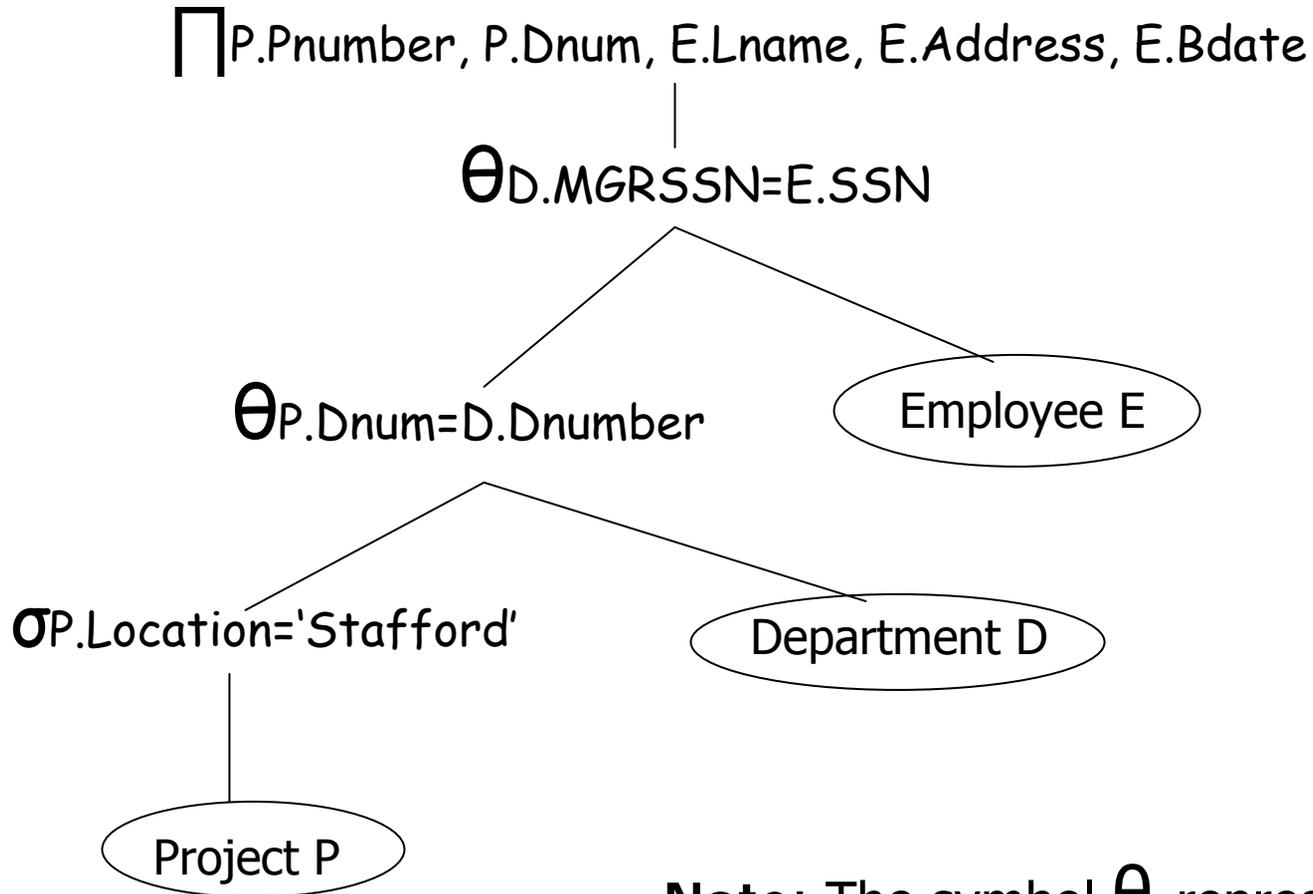
    SELECT Pnumber, Dnum, Lname, Address, Bdate
    FROM Project, Department, Employee
    WHERE Dnum=Dnumber
    AND MgrSSN = SSN
    AND Plocation = 'Stafford'

- Equivalent relational algebra expression is:

$$\Pi_{Pnumber, Dnum, Lname, Address, Bdate}$$

$$(((\sigma_{Plocation = \text{'Stafford'}}(Project))$$

$$\theta_{Dnum=Dnumber}(Department))$$

$$\theta_{MgrSSN=SSN}(Employee))$$

Note: θ stands for JOIN

... --- Query Tree: Example

$$\prod_{\text{P.Pnumber, P.Dnum, E.Lname, E.Address, E.Bdate}}$$

$$\theta_{\text{D.MGRSSN=E.SSN}}$$

$$\theta_{\text{P.Dnum=D.Dnumber}}$$

Employee E

$$\sigma_{\text{P.Location='Stafford'}}$$

Department D

Project P

Note: The symbol θ represents JOIN

$\prod$ P.Pnumber, P.Dnum, E.Lname, E.Address, E.Bdate

σ P.Dnum=D.Dnumber AND D.MGRSSN=E.SSN AND P.Location='Stafford'

X

X   E

P   D
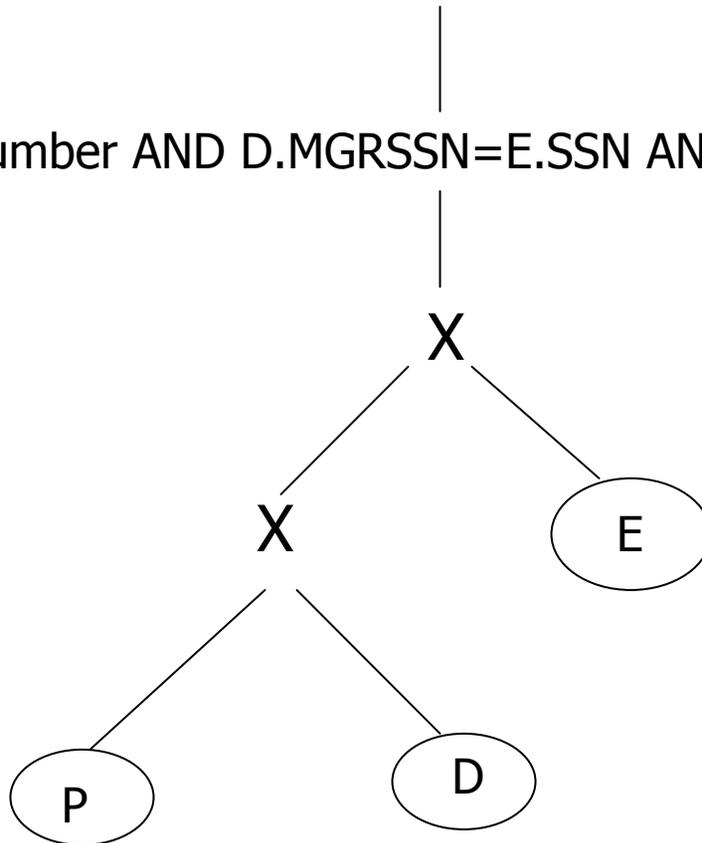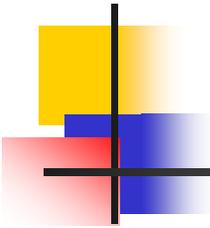
--- Query Graph Example ...

- For every project located in 'Stafford', list the project number, the controlling department number, and the department manager's last name, address, and birth date.

```
SELECT Pnumber, Dnum, Lname, Address, Bdate
FROM Project, Department, Employee
WHERE Dnum = Dnumber
AND MgrSSN = SSN
AND Plocation = 'Stafford'
```

... --- Query Graph Example

[P.Pnumber, P.Dum]   [E.lname,E.aAddress, E.Bdate]

P —— P.Dum=D.Numbr —— D —— D.MgrSSN=E.SSN —— E

P.Plocation = 'Stafford'

'Stafford'
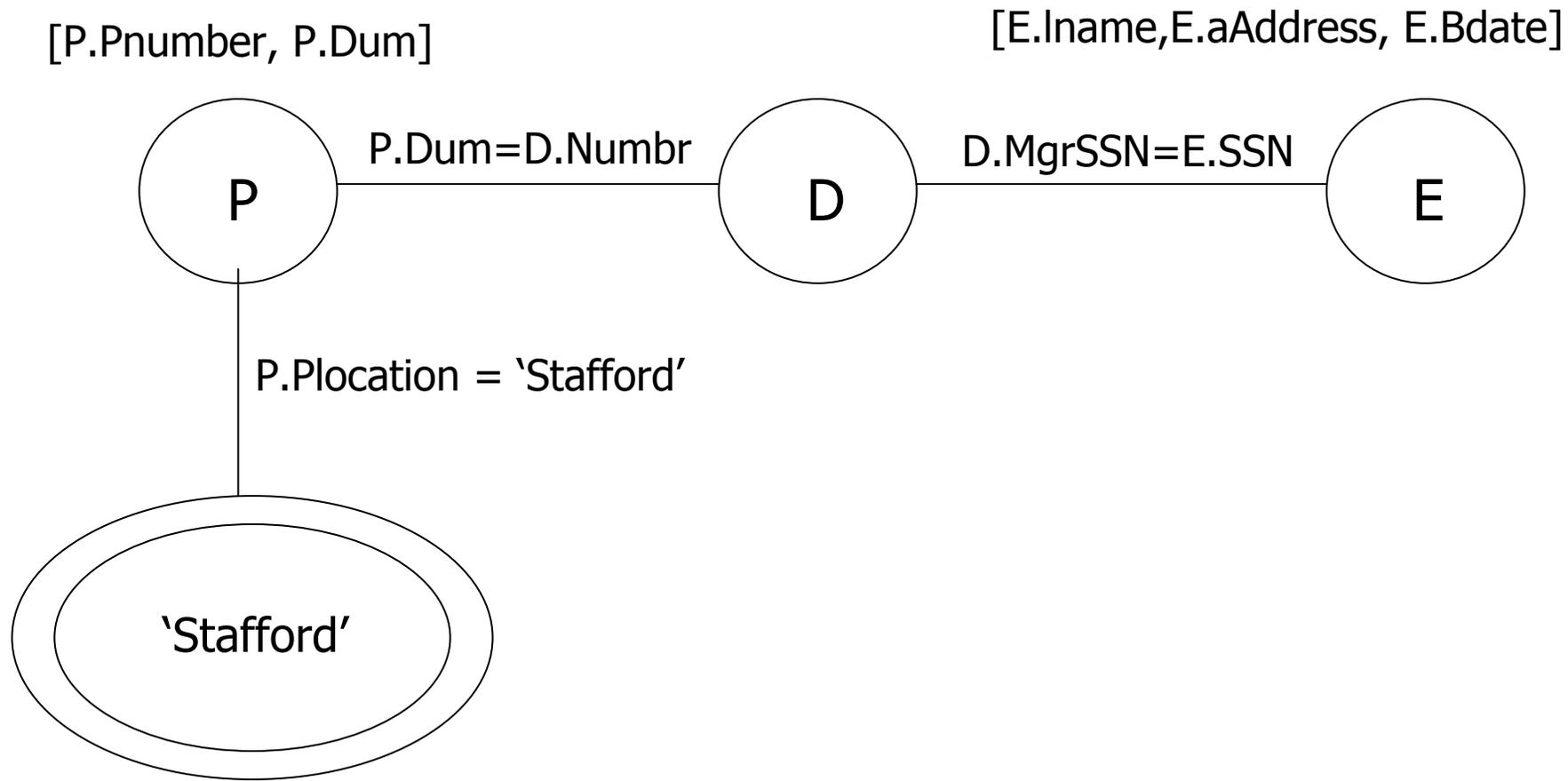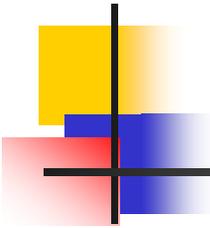
- Relational Algebra Transformation …

- The following list gives a basic selection of relational algebra transformation:
  - **Cascade of selection**: Conjunctive selection operations can cascade into individual selection operations

$$\sigma_{p \text{ AND } q \text{ AND } r}(R) = \sigma_p(\sigma_q(\sigma_r(R)))$$

  - **Cascade of projection**: In a sequence of projection operations, only the last in sequence is required.

$$\Pi_L(\Pi_{M(\ldots(}\Pi_N(R)) = \Pi_L(R)$$

  - **Commutatively of selection**: A sequence of selection operations are commutative

$$\sigma_q(\sigma_p(R)) = \sigma_p(\sigma_q(R))$$

  - **Associativity of Natural Join and Cross Product**: Natural join (*) and Cross Product (X) are associative:
    $$(R * S) * T = R * (S * T)$$
    $$(R \times S) \times T = R \times (S \times T)$$
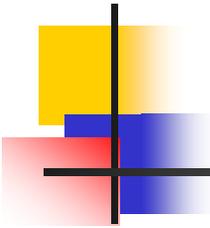
-- Heuristic Optimization

- Perform selection operations as early as possible
  - Keep predicates on same relation together
- Combine Cartesian product with subsequent selection whose predicate represents join condition into a join operation.
- Use associative of binary operations to rearrange leaf nodes with most restrictive selection operations first.
- Perform projection as early as possible
  - Keep projection attributes on same relations together.
- Compute common expressions once.
  - If common expression appears more than once, and result not too large, store result and reuse it when required.
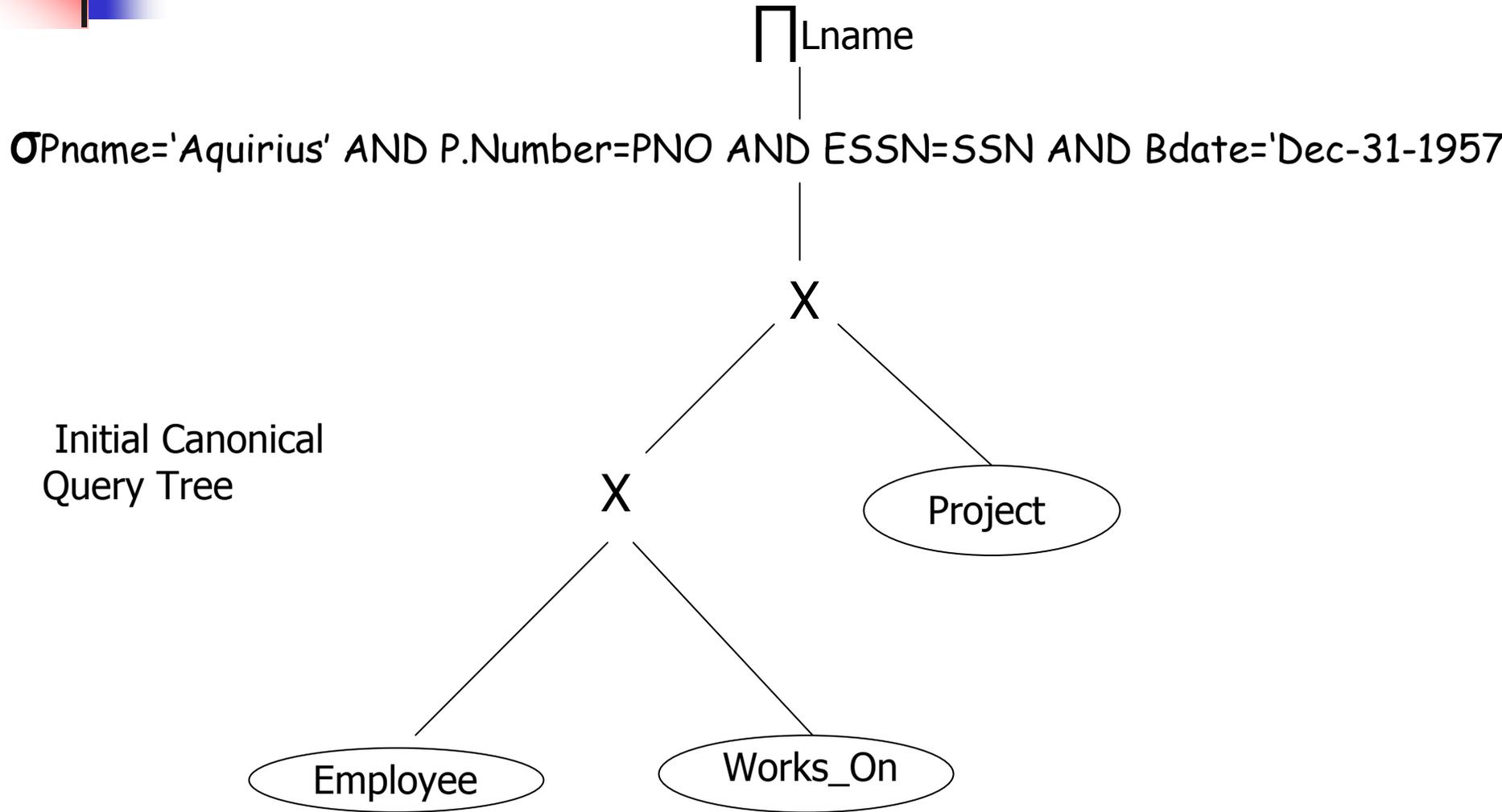 - Useful when querying views, as same expression is used to construct view each time.
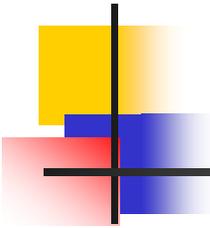
-- Query Optimization: Example …

- Query: Find the last name of employees born after 1975 who work on a project named 'Aquarius'

```
SELECT Lname
FROM Employee, Works_on, Project
WHERE Pname = 'Aquarius'
AND ESSN = SSN
AND Pnumber = PNO
AND Bdate > '1975-12-31'
```
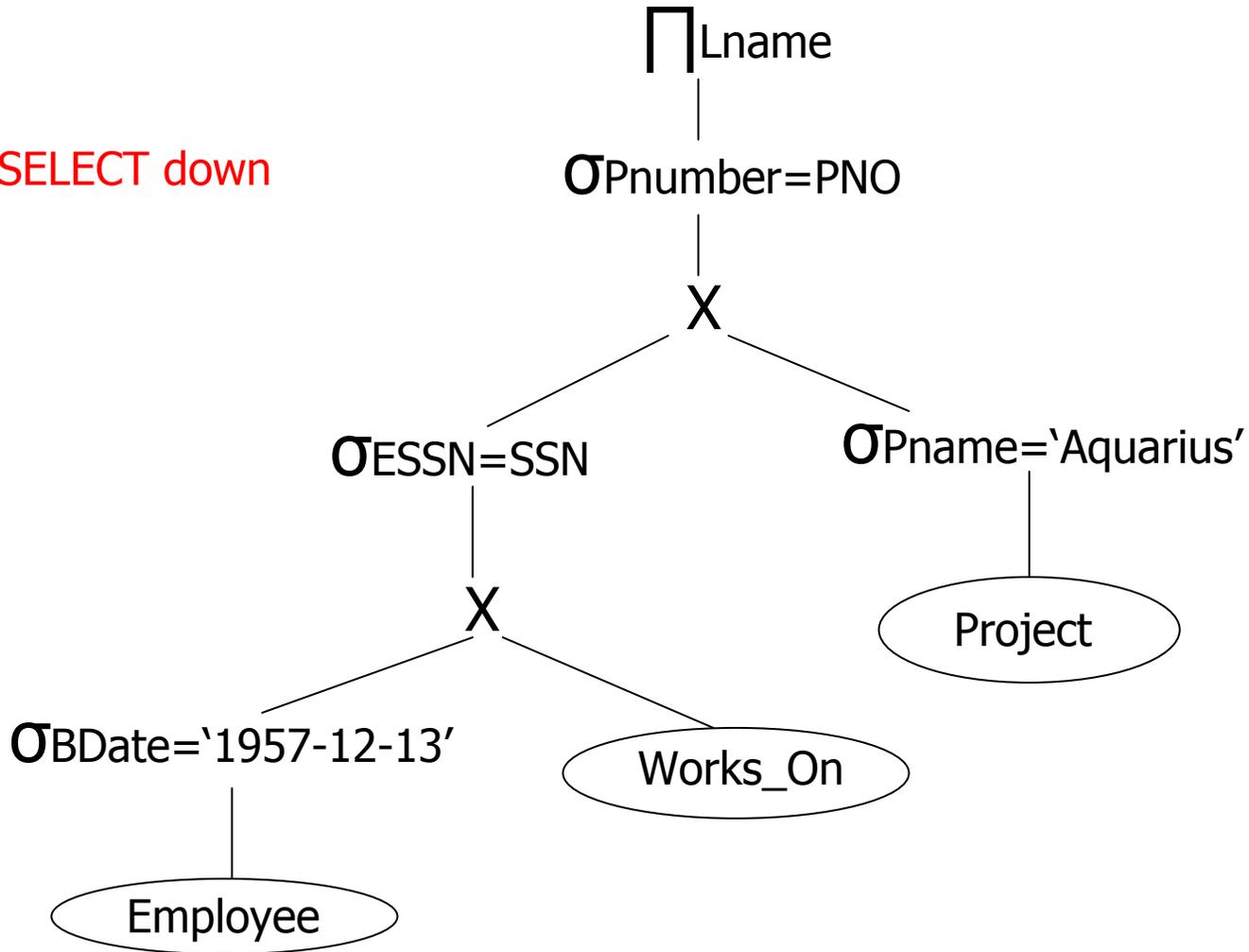
... -- Query Optimization: Example ...

$$\prod Lname$$

$$\sigma Pname='Aquirius' \text{ AND } P.Number=PNO \text{ AND } ESSN=SSN \text{ AND } Bdate='Dec-31-1957'$$

X

Initial Canonical
Query Tree

X

Project

Employee

Works_On

... -- Query Optimization: Example ...

Moving SELECT down

$$\prod Lname$$

$$\sigma Pnumber=PNO$$

X

$$\sigma ESSN=SSN$$ $$\sigma Pname=`Aquarius'$$

X   Project

$$\sigma BDate=`1957-12-13'$$ Works_On

Employee

$$\prod Lname$$

$$\sigma ESSN=SSN$$

Apply more restrictive SELECT

X

$$\sigma Pnumber=PNO \qquad \sigma BDate='1957-12-13'$$

X

Employee

$$\sigma Pname='Aquirius'$$

Works_On
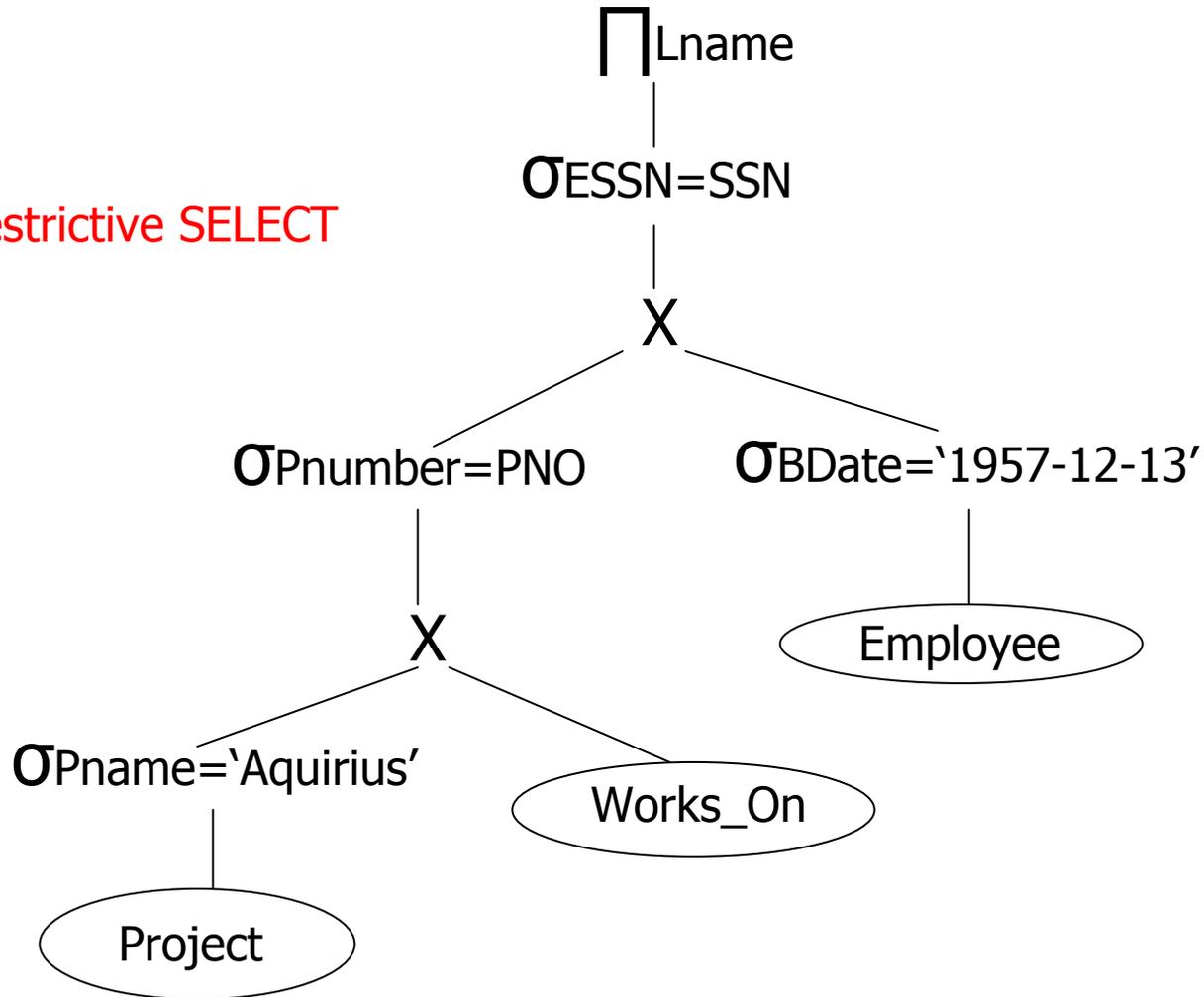
Project

Replace CARTESIAN PRDUCT
And SELECT with JOIN

$\prod$Lname
|
σESSN=SSN

θPnumber=Pno σBDate='1957-12-13'

σPname='Aquirius'   Works_On   Employee
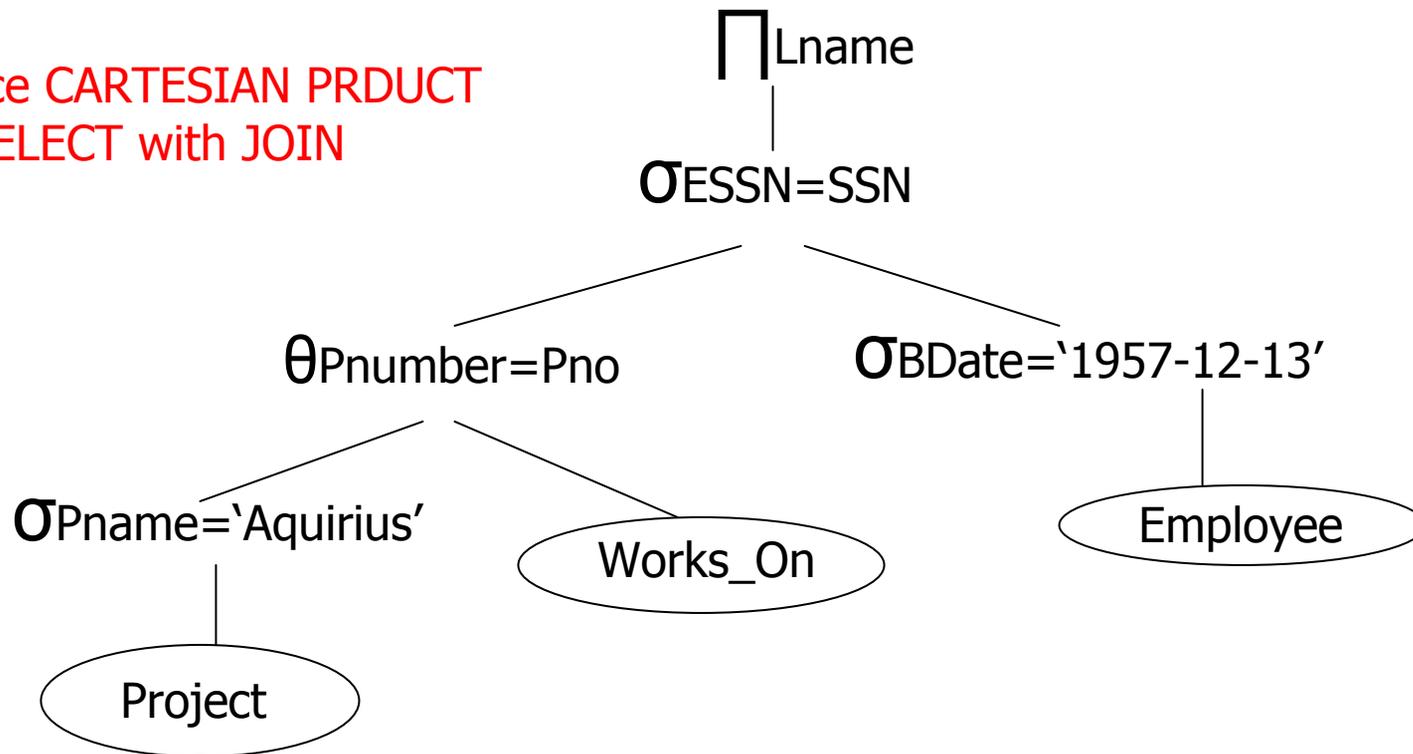
Project

… -- Query Optimization: Example

Move PROJECTION down

$\prod$Lname
θESSN=SSN
$\prod$ESSN
$\prod$SSN,Lname
θPnumber=Pno
σBDate='1957-12-13'
$\prod$Pnumber
$\prod$ESSN,PNO
σPname='Aquirius'
Works_On
Employee
Project