Chapter one

Query processing and Query Optimization

Query Processing

- **Query Processing** is the range of activities involved in extracting data from database.
- The activity include translation of queries in high level database language into expression that can be used at physical system of the file system.
- **Query processing** is a set of activities involving in getting the result of a query expressed in a high-level language.

Query Processing

- These activities includes **parsing** the queries and **translate** them into **expressions** that can be implemented at the physical level of the file system,
- Optimizing the query of internal form to get a suitable execution strategies for processing and then doing the actual execution of queries to get the results.
- **Query processing**: A 3-step process that transforms a **high-level query** (of relational calculus/SQL) into an **equivalent** and **more efficient lower-level query.**

Basic Steps in Processing an SQL Query

1. **Parsing and Translating:-**

✓ Parser checks **syntax**, **validates relations**, **attributes** and **access permissions**.

✓ Translate the query into an equivalent relational algebra expression.

2. **Evaluation:-**

- The query execution engine takes a physical **query plan**, **executes the plan**, and **returns the result**.
- Generate an optimal **evaluation** plan (with lowest cost) for the query plan.

Basic Steps in Processing an SQL Query

3. Optimization:

- Find the **cheapest execution plan** for a query.
- The query-execution engine takes an (optimal) evaluation plan, executes that plan, and returns the answers to the query.
- Objective of query optimization is to **minimize** the following cost function:
- **I/O cost + CPU cost + communication cost.**

cont...

- A query expressed in a high-level query language such as SQL must first be **scanned, parsed, and validated.**
- The **scanner** identifies the language token such as SQL keywords, attribute names, and relation names in the text of the query.
- Whereas the **parser checks** the query syntax to determine whether
- it is formulated according to the syntax rules of the query language.
- The query must also be **validated**, by checking that all attribute and relation names are valid and **semantically** meaningful names in the schema of the particular database being queried.

Query processing cont…

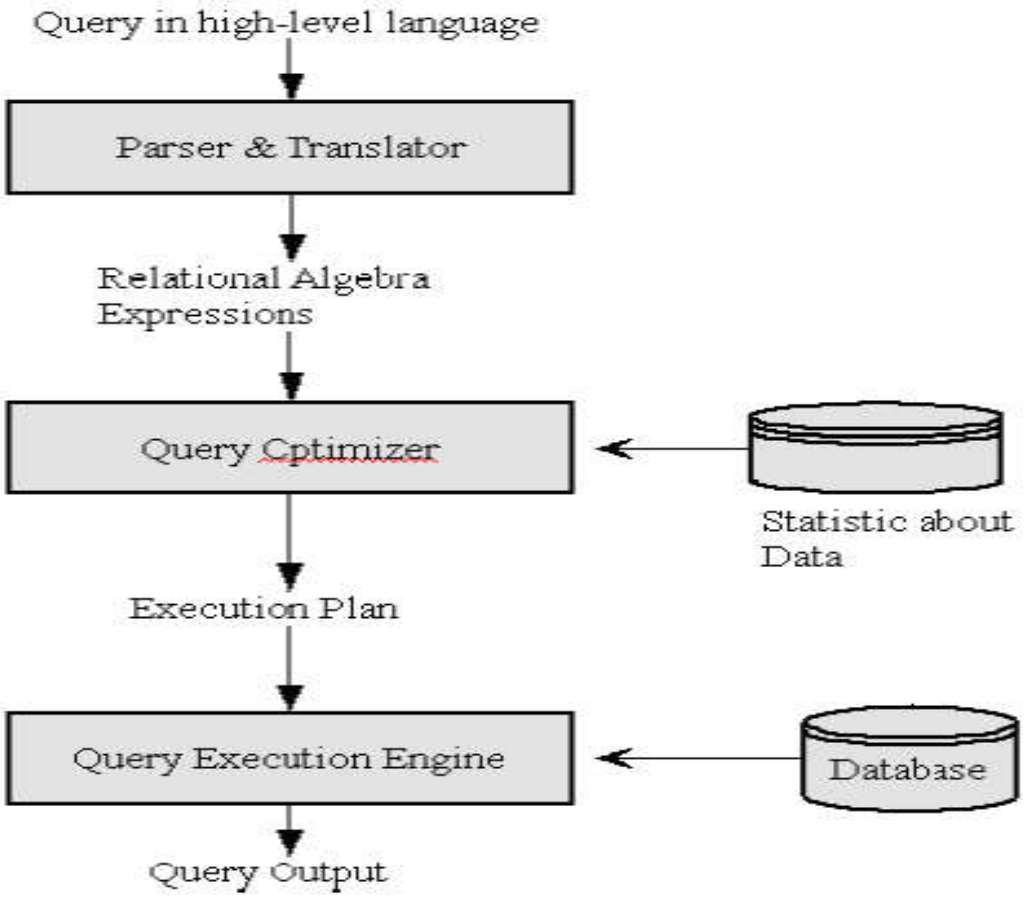


Figure 1: Steps in query processing process

Translating SQL Queries into Relational Algebra

- We need to know about relational algebra to understand query execution and optimization in a relational DBMS.
- **Relational Algebra:-** An algebra whose objects are relations and whose operators transform relations into other relations.
- Basic operators: select , project, union, set difference, Cartesian product (or cross product)

For example, consider the query:-

```
SELECT Salary
FROM EMPLOYEE
WHERE Salary >= 5000 ;
```

- The possible relational algebra expressions for this query are:

  Π Salary(σ Salary>=5000(EMPLOYEE)) or

  σ Salary>=5000(Π Salary(EMPLOYEE))

Translate SQL query into relational algebra

Example

- Instructor(ID,Fname,gender,salary,ddno).
- Department(Dno,dname,address).
- Course(course_id,title,deptname,credits).

Examples of Translate SQL query into relational algebra

1. Retrieve Fname of instructor who works in ‘cs’ department

$$\pi_{fname}(\sigma_{deptname='cs'}(department))instructor.$$

2. Find all instructor name with salary >9000

select name from instructor where salary >9000

$$\pi_{name}(\sigma_{salary>900}(instructor))$$

3. Find instructor in cs and salary>9000.

$$\pi_{name}(\sigma_{salary>900 \wedge deptname='cs'}(instructor))$$

Query Optimization

- It is the process of choosing a **suitable execution strategy** for processing a query.
- It is optimizing the query of internal form to get a suitable execution strategies for processing and then doing the actual execution of queries to get the results.
- Used to find an efficient physical query plan for an SQL query.
- Goal is minimize the evaluation time for the query,

  i.e. compute query result as fast as possible

Steps in query optimization

1. **Query Tree Generation:**
   - ✓ A **Query Tree** is a tree data structure representing a relational algebra expression.
   - ✓ The tables of the query are represented as **leaf nodes**.
   - ✓ The relational algebra operations are represented a **internal node**
   - ✓ The **root** represents the query as a whole.
2. **Query Plan Generation:**
   - ✓ After the **Query Tree** is generated, a **query plan** is made.
   - ✓ A **query plan** is an **extended query tree** that includes access paths for all operation in the query tree.
   - ✓ Access paths specify how the relational operations in the tree should be performed.

Cont'd

3. **Query Plan Code Generation:**
   - ✓ Code Generation is the final step in the Query Optimization.
   - ✓ It is the executable form of the query.
   - ✓ Once the query code is generated, the execution manager runs it and produces the results.

- A **query tree is used to represent a relational** algebra or extended relational algebra expression, whereas
- A **query graph is used to represent a relational** calculus expression.

Techniques for Query Optimization

Main techniques for query optimization

1. **Based on Heuristic Rules for ordering the** operations in query execution strategy.
2. **Systematically estimation:**
   – It estimates cost of different execution strategies and chooses the execution plan with lowest execution cost.
3. **Semantic query optimization**

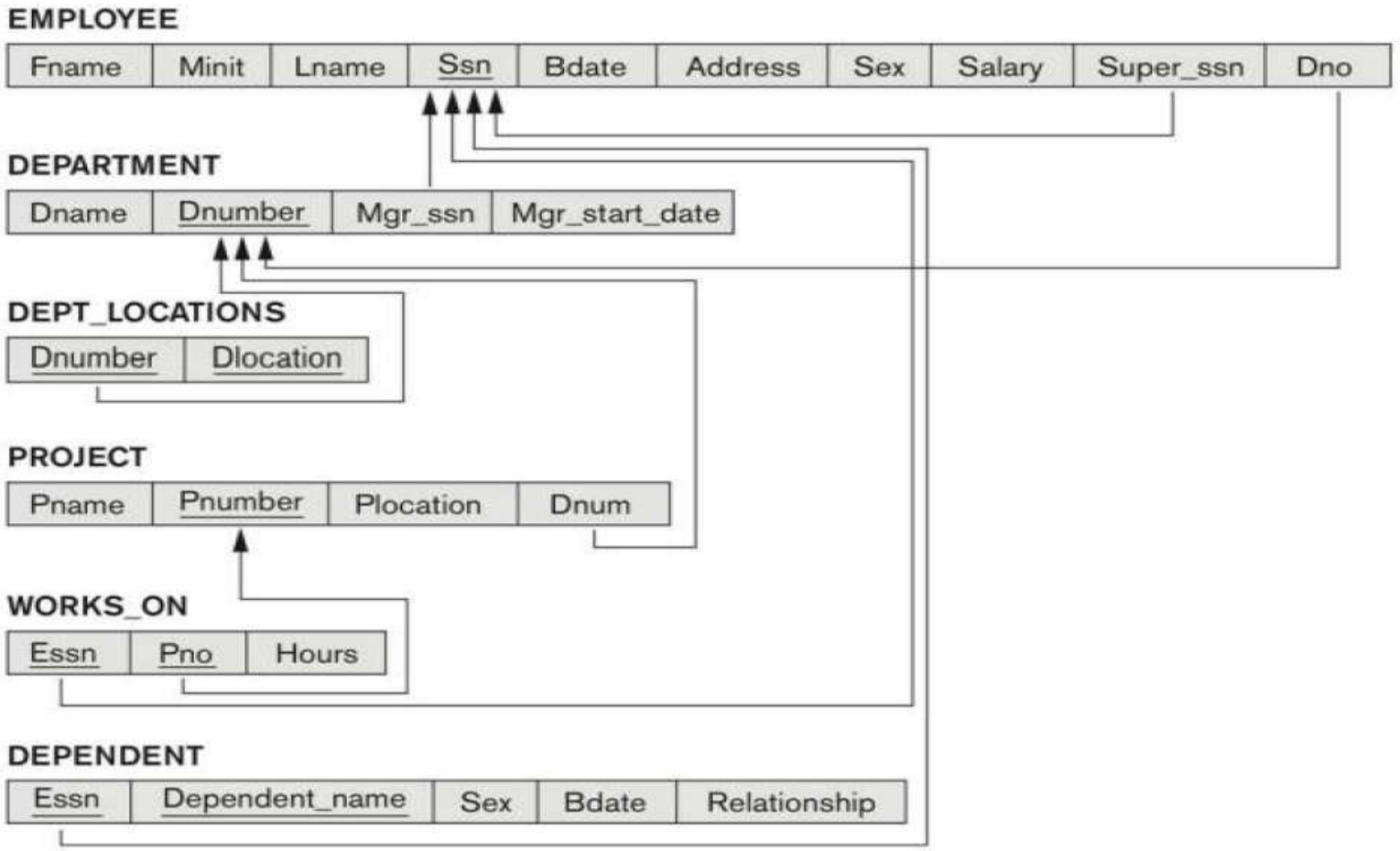
Heuristic Approach

- The heuristic rules are used as an optimization technique to modify the internal representation of query.
- Heuristic rules are used in the form of query tree of query graph data structure, to improve its performance.
- One of the main heuristic rule is to apply SELECT operation before applying the JOIN or other BINARY operations.
- This is because the size of the file resulting from a binary operation such as JOIN is usually a multi value function of the sizes of the input file

General Guideline

- A conjunctive selection condition can be broken up into a cascade of individual σ operations.
- this will allow moving selection down the tree at different branches
- Rearrange base relations so that the most restrictive selection is executed first.
- Combine Cross product X with a selection replace with JOIN
- Moving project operations down the query tree
- Execute select and join operations that are more restrictive or result in less tuples

Company database schema



Cont'd

- Q2. Find the last names of employees born after 1957-12-31 who work on a project named 'Aquarius'.

for the following question

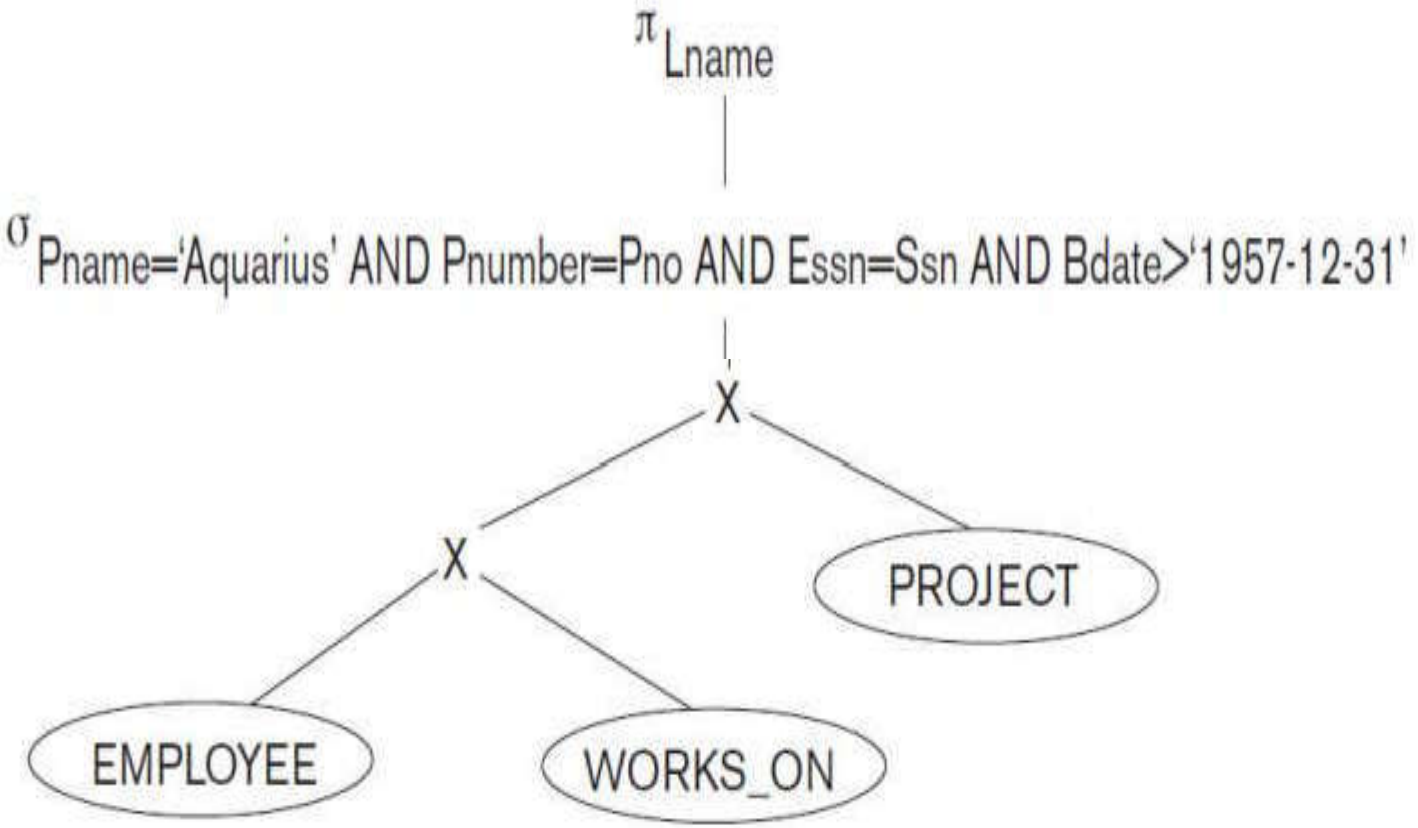
- ✓ Write SQL query
- ✓ Write relational algebraic representation
- ✓ Draw the canonical query tree
- ✓ Using the Heuristic rules optimize (show all the necessary steps)

SQL

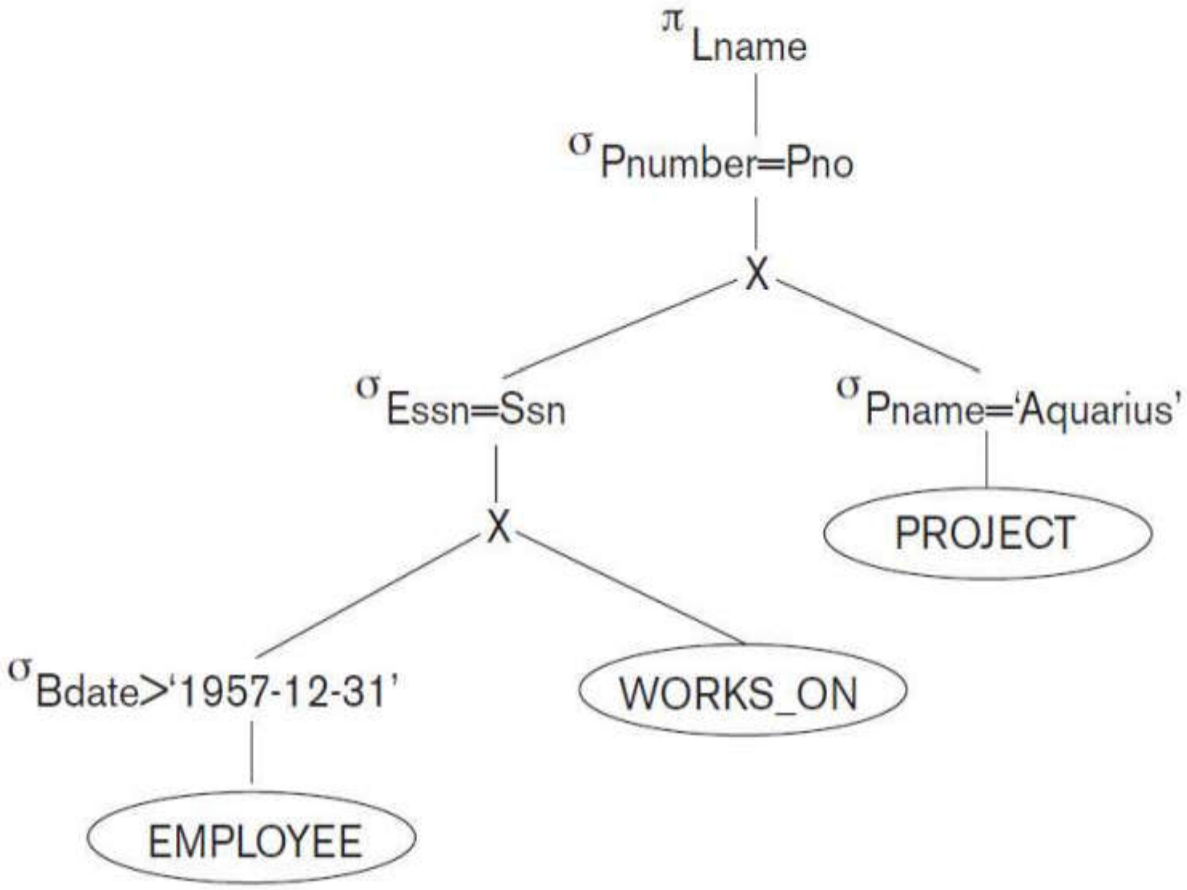
- SQL Query

```
SELECT Lname
FROM EMPLOYEE, WORKS_ON, PROJECT
WHERE Pname='Aquarius' AND Pnumber=Pno AND Essn=Ssn
AND Bdate > '1957-12-31';
```

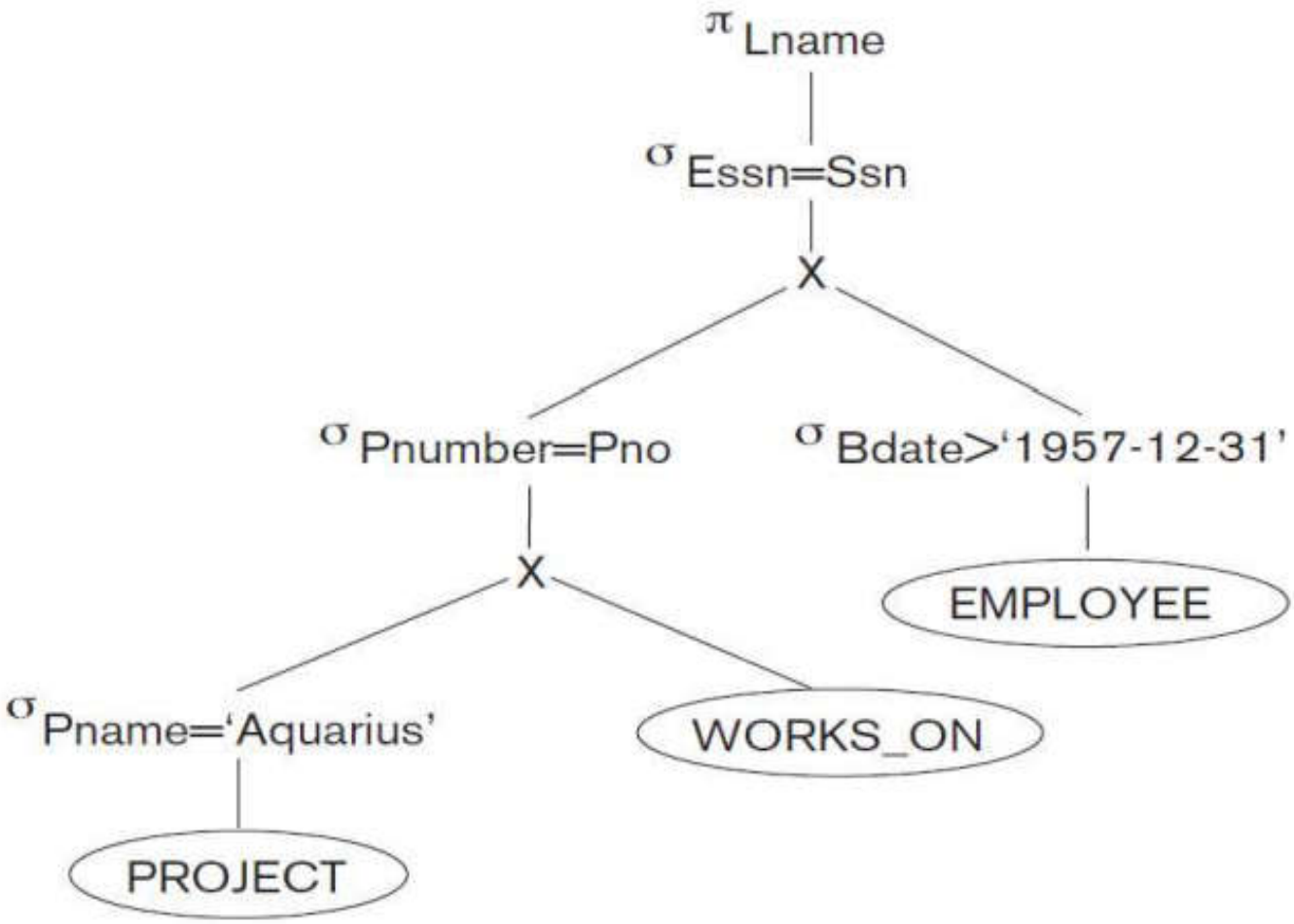
Initial (canonical) query tree for SQL query Q



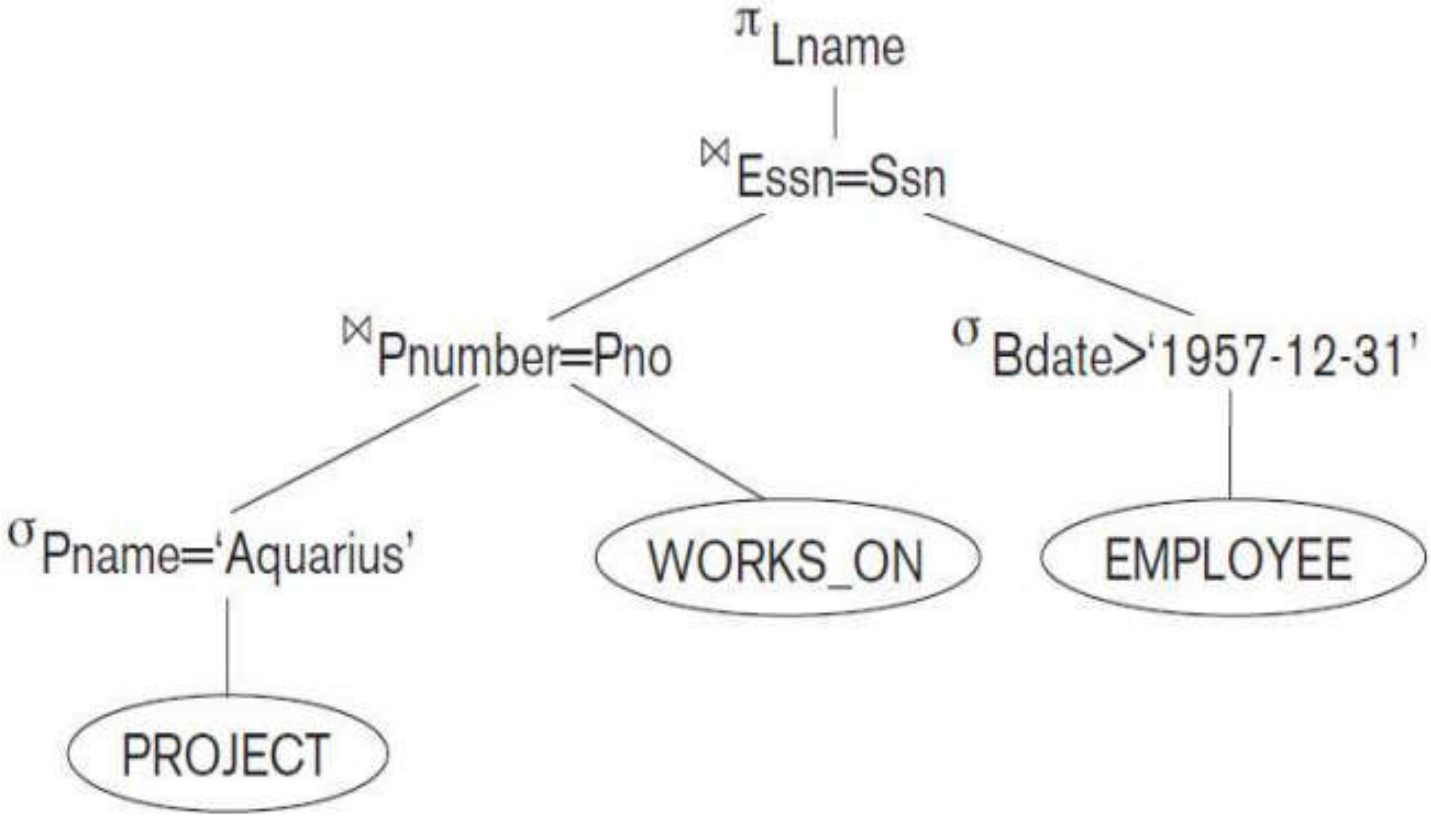
Moving select operation down the query tree



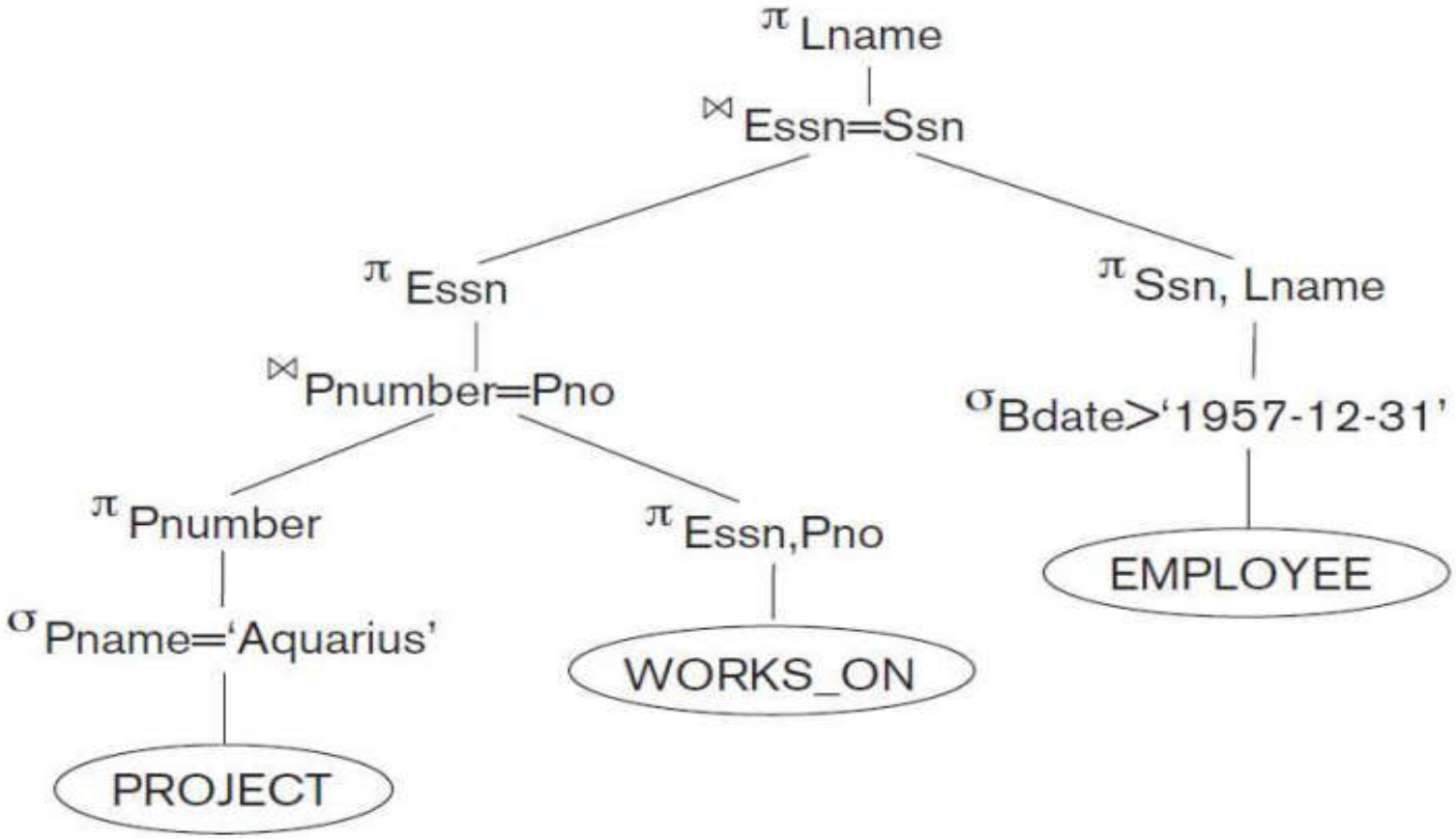
Applying the more restrictive select operation first



Replacing Cartesian product and select with join operations.



Moving project operations down the query tree



Exercise 1

for the following question use the schema give on slide 18

- Write SQL query
- Write the possible relational algebra representation
- Draw the canonical query tree
- Using the Heuristic rules optimize (show all the necessary steps)

Q1. For every project located in 'Stafford', retrieve the project number, the controlling department number, and the department manager's last name, address, and birthdate.

Exercise 2

for the following question use the schema give on slide 18

- Write SQL query
- Write the possible relational algebra representation
- Draw the canonical query tree
- Using the Heuristic rules optimize (show all the necessary steps)

Q2. Retrieve first name, birthdste and address of an employee from the research department.

Systematical Estimation(Cost Estimation)

- It uses traditional optimization techniques that search the *solution space to a problem for a* solution that minimizes an objective (cost) function.
- The cost functions used in query optimization are estimates and not exact cost functions
- Cost Estimation for Relational Algebra Expressions:
  - Estimation of relational algebra expression
  - Choosing the expression with the lowest cost

Cont'd

Cost Estimation Components:

- **Access cost to secondary storage** : is the cost of transferring data blocks between secondary disk storage and main memory buffers
- **Storage cost** – cost of storing intermediate results
- **Computation cost** : is the cost of performing in-memory operations on the records within the data buffers during query execution.
- **Memory usage cost** :t he number of main memory buffers needed during query execution.
- **Communication cost**: is the cost of shipping the query and its results from the database site

Semantic Query Optimization

- Semantic information stored in databases as integrity constraints could be used for query optimization.
- **integrity** : preserve data consistency when changes made in a database.
- This technique, which may be used in combination with the techniques discussed previously, uses constraints specified on the database schema.
  - such as unique attributes and other more complex constraints.

Advantages of Query Optimization

- Faster processing of Query
- Lesser cost per Query
- High performance of the system
- Lesser stress on the database
- Efficient usage of database engine
- Lesser memory is consumed

Reading Assignment

- What is System R or System R approach?