Programming Project 1: Heuristic Query Optimization

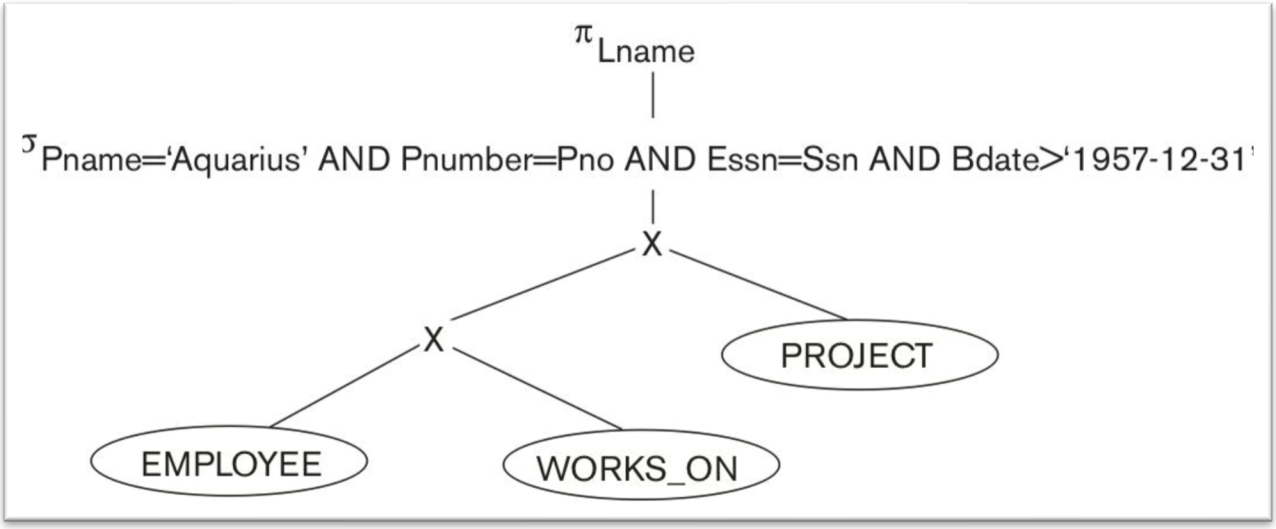
Example Inputs and Outputs

Example 1 (input1.txt)

Input SQL Query

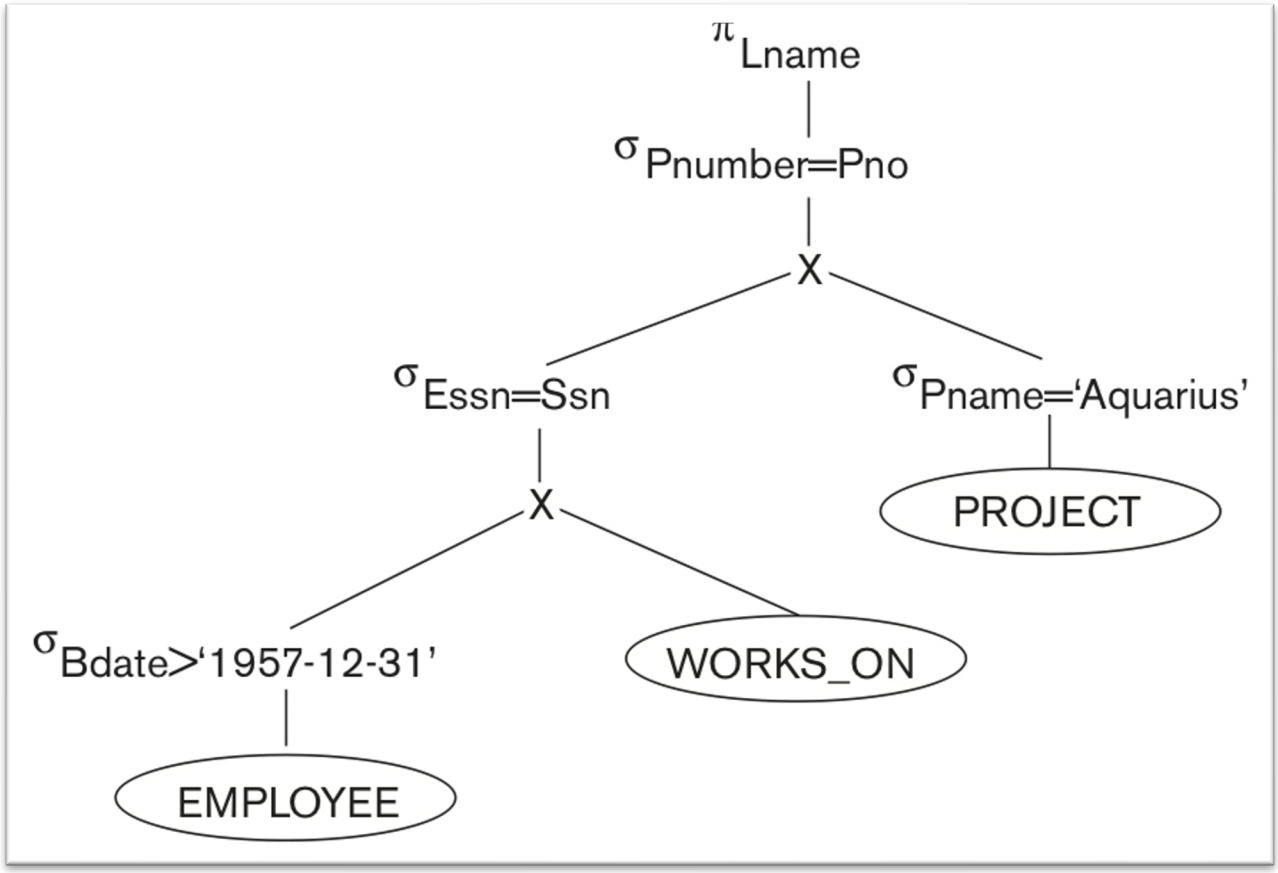
```
SELECT   E.Lname
FROM     EMPLOYEE E, WORKS_ON W, PROJECT P
WHERE    P.Pname='Aquarius' AND P.Pnumber=W.Pno
         AND E.Essn=W.Ssn
         AND E.Bdate>'1957-12-31';
```

Canonical Query Tree

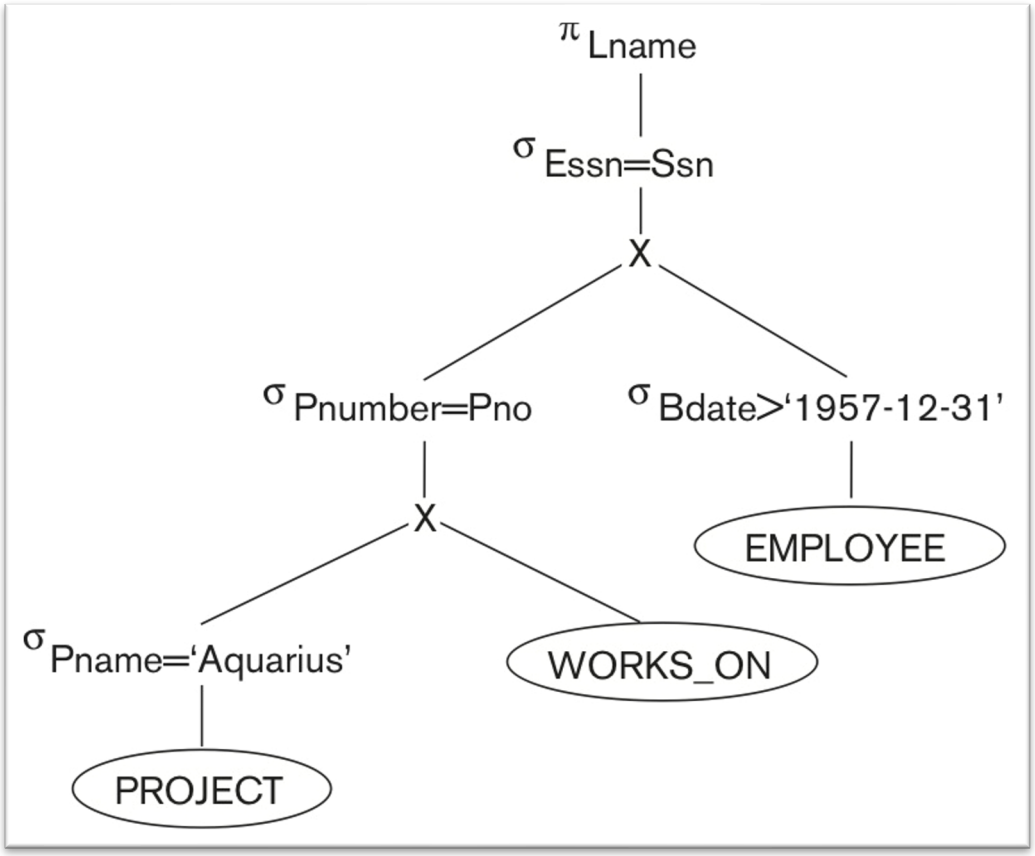


Optimized Query Tree

Applied Rule #1 & #2: Identify conjunctive selections, break them up individually, and push them down as close as possible to the leaf nodes.



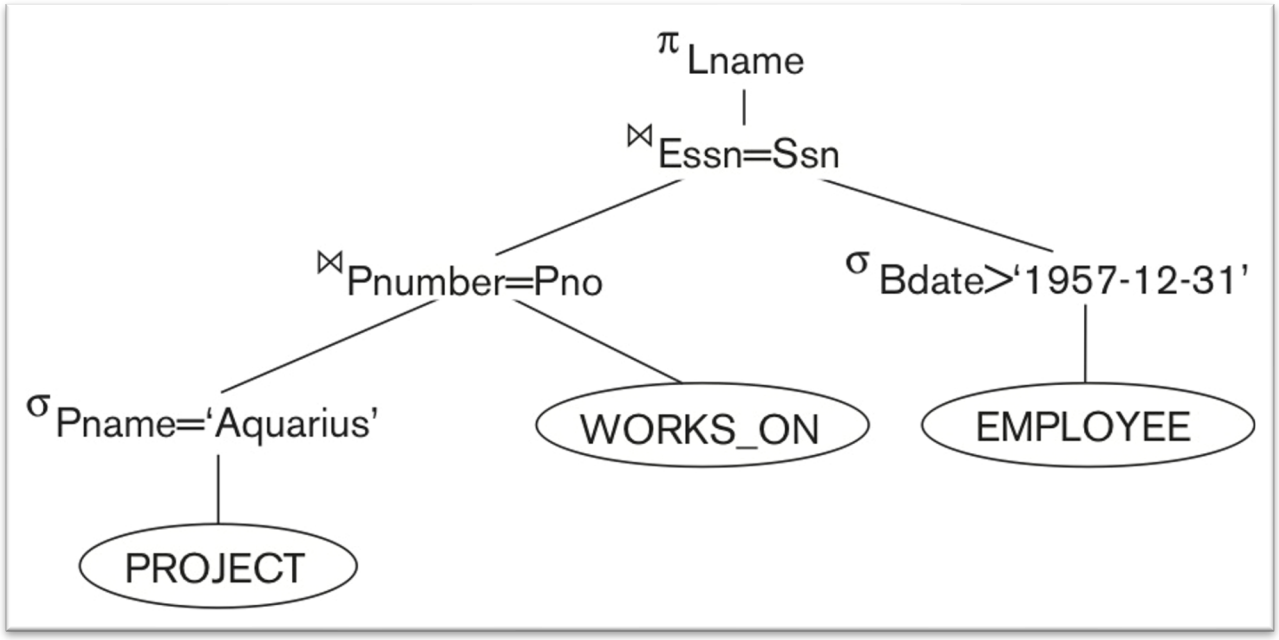
Applied Rule #3: Apply Selections with Smallest Selectivity First.



In this query tree example, selections are arranged so that **the most restrictive (smallest selectivity)** conditions are executed **as early as possible**—that is, closest to the base relations.

- **$\sigma_{Pname='Aquarius'}$ on PROJECT**
  - This is an **equality condition** on a **unique attribute** (Pname is declared UNIQUE).
  - It will likely return **only one tuple**, making it **highly selective**.
- **$\sigma_{Bdate>'1957-12-31'}$ on EMPLOYEE**
  - This is a **range predicate**, which filters moderately but not as tightly as a key-based equality.

Applied Rule #4: Replace Cartesian Product + Selection with Join



Applied Rule #5: Push Projections Down