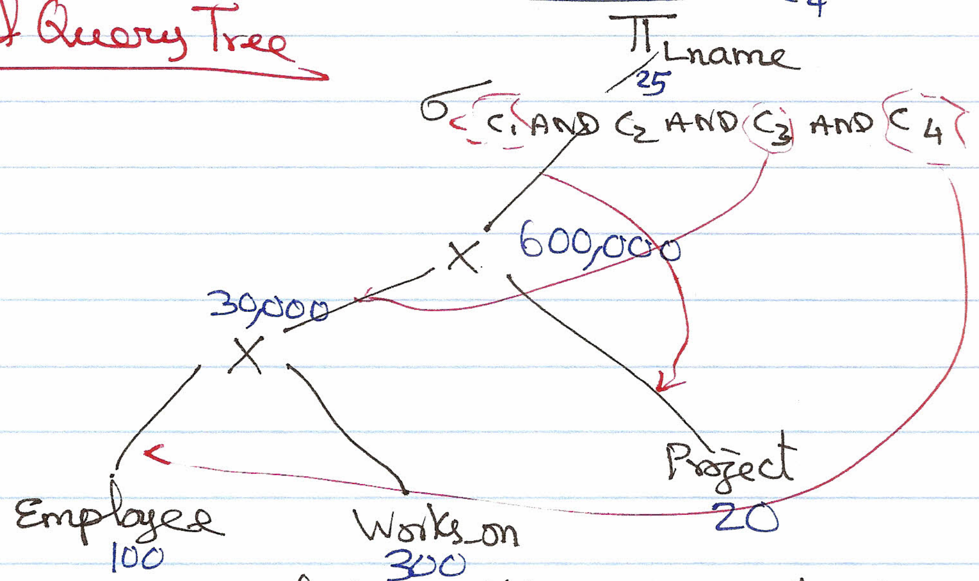
Query Optimization illustration

List the lastname of employees who work for 'AQUARIUS' Project and born after '1982-12-31'.

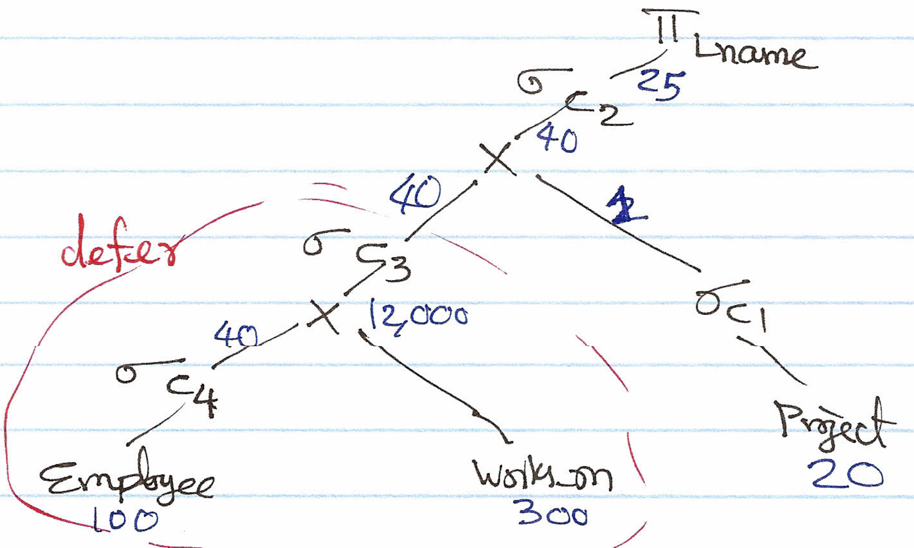
Given SQL Query

```
③ SELECT  Lname
① FROM    Employee, works_on, Project
② WHERE   Pname='AQUARIUS'        AND   c1
          Pnumber= Pno            AND   c2
          ESSN = SSN              AND   c3
          Bdate > '1982-12-31'          c4
```

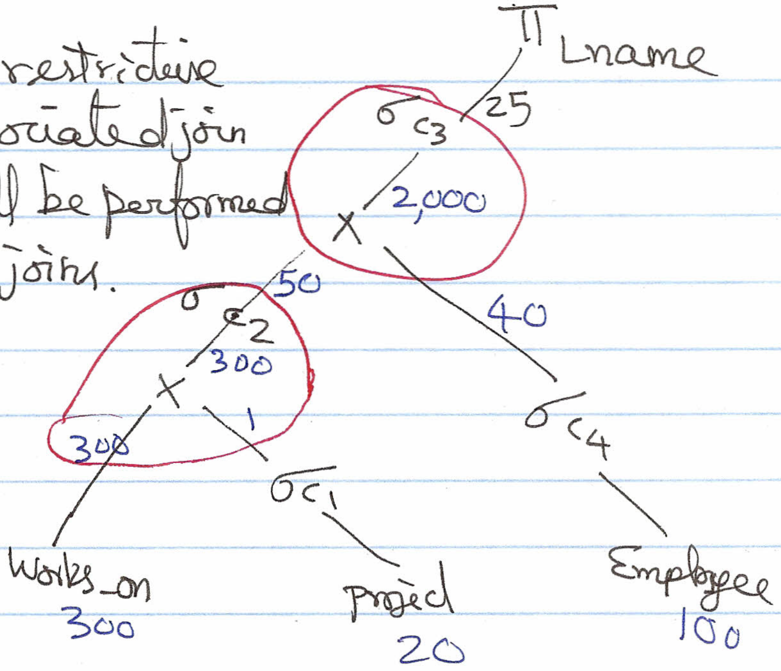
Initial Query Tree



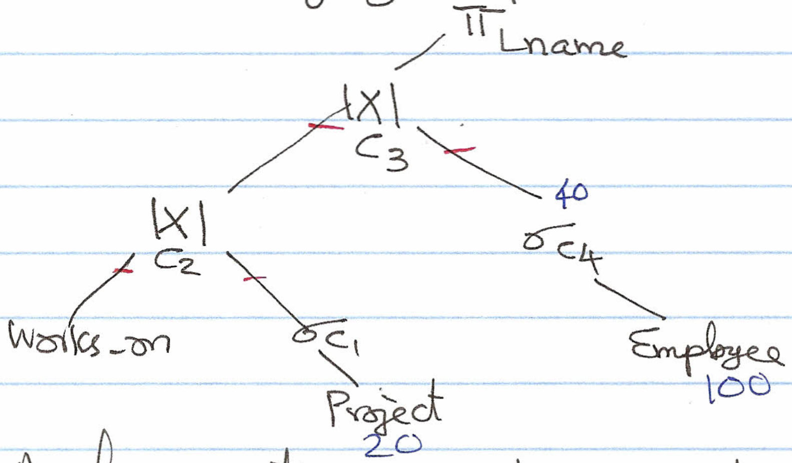
Step1: Move select conditions down the tree.



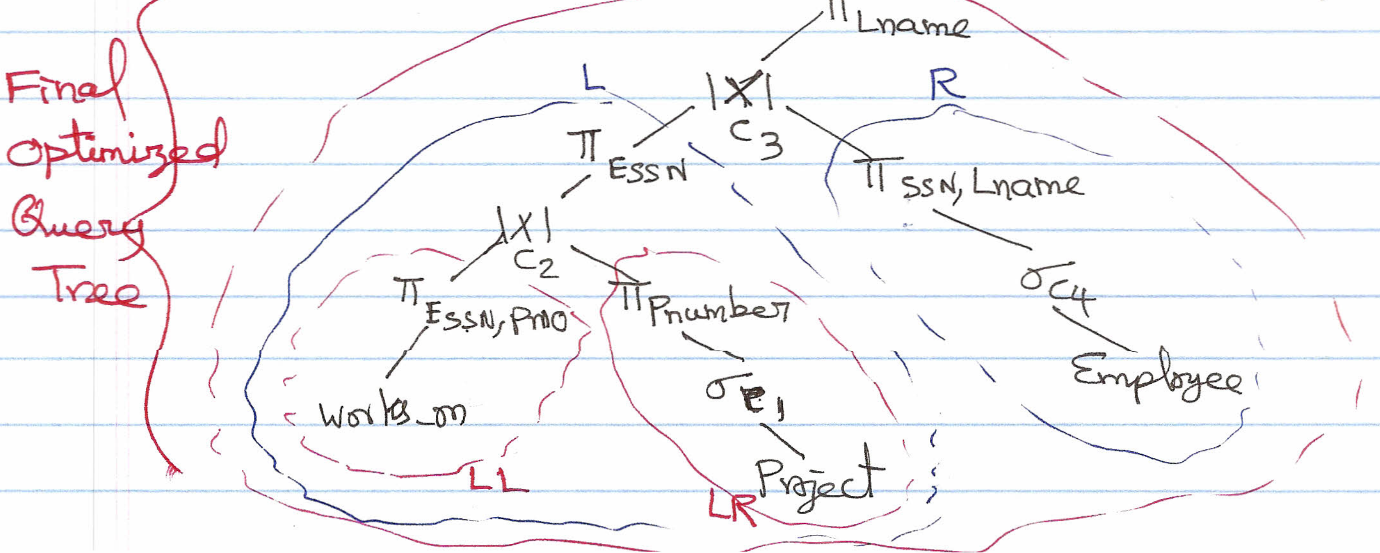
Step 2: Swap the most restrictive selection & its associated join so that this join will be performed earlier than other joins.



Step 3: Replace cross product & selection pair with sort-merge join operation



Step 4: Apply projection on each incoming branch of join operations.



```
SELECT  Lname
FROM  (  SELECT ESSN
         FROM  (
                   LL   )
               JOIN
                (
                   LR  )
               ON  C2
                                  )  as L
      JOIN
         (
               R  )
      ON   C3
```

Complete SQL Query for the optimized tree:

```
SELECT  Lname
FROM  (  Select ESSN
         FROM  (Select ESSN,PNO
     LL         From works_on) as LL
 L             JOIN
               ( Select Pnumber
     LR           From Project
                  Where  Pname='AQUARIUS') as LR
               ON  LL.PNO=LR.Pnumber) as L
      JOIN
         ( Select SSN, Lname
 R         From Employee
           Where  Bdate > '1982-12-31') as R
      ON  L.ESSN=R.SSN;
```

Tuning Queries

- Partitioning a table and store them on multiple physical disks to exploit concurrent disk I/O operations.

- Creating relevant indexes for retrieval queries.
- Apply deNormalization and store derived results to speed up future queries on the same set of tables.