

DBMS DIGITAL ASSIGNMENT 2

Name: Om Ashish Mishra

Registration Number: 16BCE0789

Slot: D1+TD1

The Question:

1. Consider the following database schema where the primary keys are underlined
Employee(EId, EName, ESal, PId)
Project(PId, PName, PDeadline, PBudget).

Which of the following is the most optimized query while retrieving employee

names

whose salary is more than 1000 and are allocated in a project having a budget more than 50000? Justify your answer. (2 Marks)

- A. $\Pi_{EName}(\Pi_{EName, ESal, PId}(\sigma_{ESal > 1000}(Employee)) \bowtie \Pi_{PId, PBudget}(\sigma_{PBudget > 50000}(Project)))$
- B. $\Pi_{EName}((\sigma_{ESal > 1000}(Employee)) \bowtie \Pi_{PId}(\sigma_{PBudget > 50000}(Project)))$
- C. $\Pi_{EName}(\Pi_{EName, PId}(\sigma_{ESal > 1000}(Employee)) \bowtie \Pi_{PId}(\sigma_{PBudget > 50000}(Project)))$
- D. $\Pi_{EName}(\sigma_{ESal > 1000 \wedge PBudget > 50000}(Employee \bowtie PIdProject))$

Ans:

(C) $\Pi_{EName}(\Pi_{EName, PId}(\sigma_{ESal > 1000}(Employee)) \bowtie \Pi_{PId}(\sigma_{PBudget > 50000}(Project)))$

Verification:

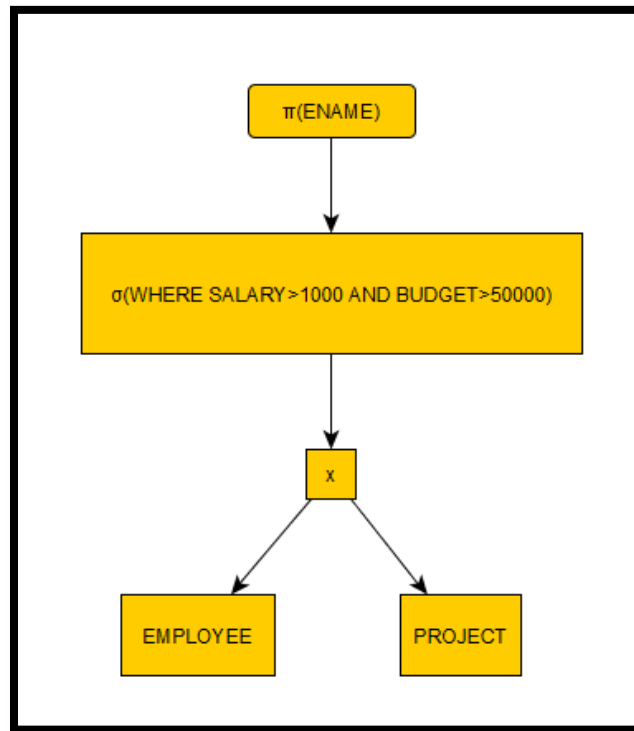
Employee(EId, EName, ESal, PId)

Project(PId, PName, PDeadline, PBudget)

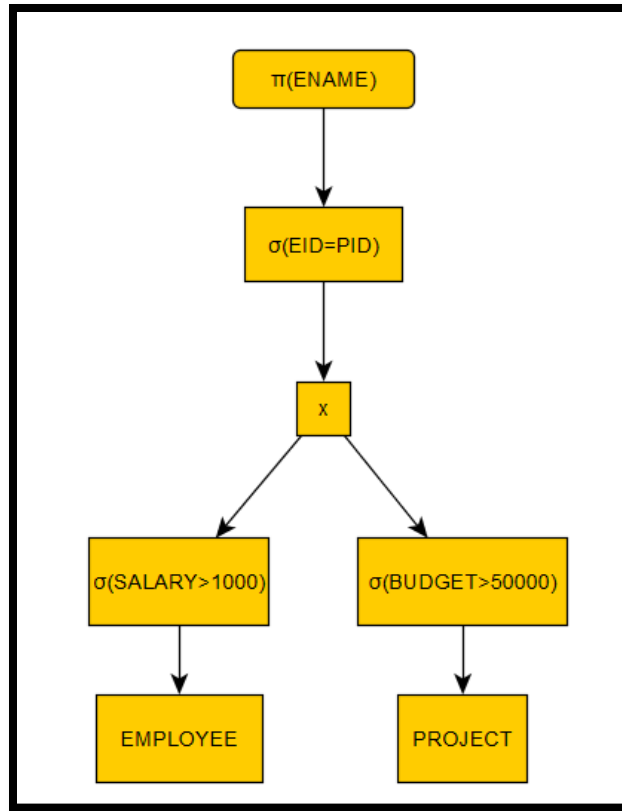
The Query:

```
SELECT EName
FROM EMPLOYEE, PROJECT
WHERE SALARY > 1000 AND BUDGET > 50000 AND PID=EID;
```

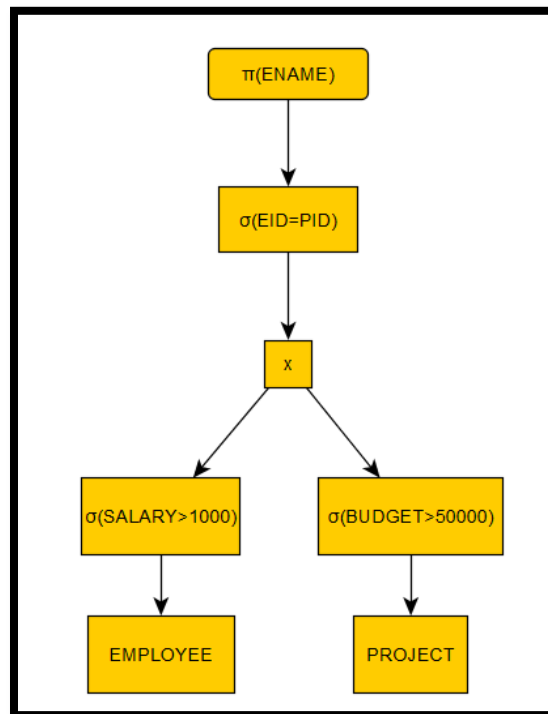
Step 1: Canonical query tree:



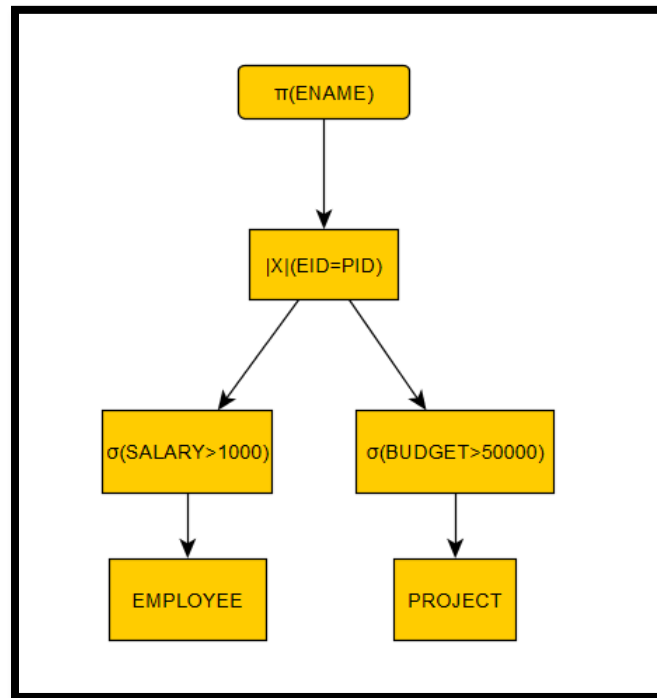
Step 2: Moving SELECT operations down the query tree



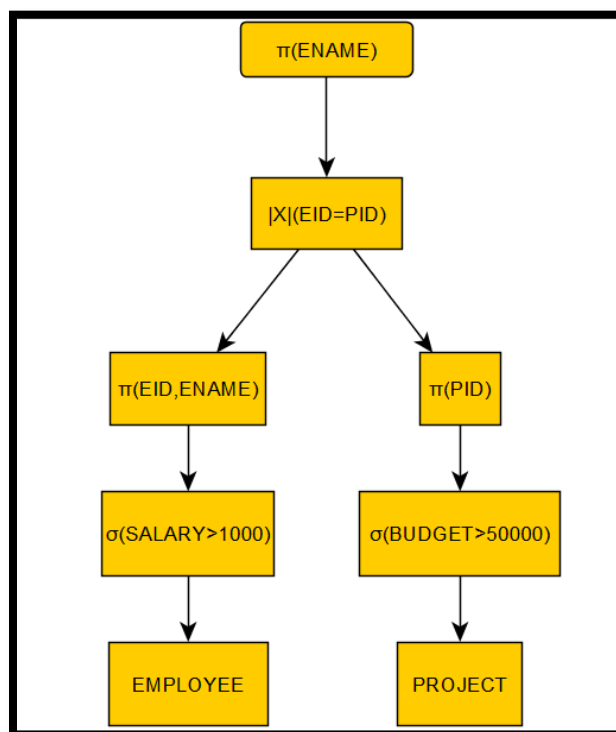
Step 3: Applying more restrictive SELECT operation first (Same as 2)



Step 4: Replacing CARTESIAN PRODUCT and SELECT with JOIN($|X|$)



Step 5: Moving PROJECT operations down



Hence proved.

2. Choose the correct set of replacements for operation (+) for which following equivalence relation holds: $\sigma_{\theta(E1(+)E2)} \equiv \sigma_{\theta(E1)(+)E2}$. (1 Mark)

- A. \cup
- B. \cap
- C. $-$
- D. Both \cap and \cup

Ans: Both B and C. Verification: Ans 3.

3. Choose the correct set of replacements for operation (+) for which following equivalence relation holds: $\sigma_{\theta(E1 (+) E2)} \equiv \sigma_{\theta(E1) (+) E2}$. Justify your answer. (2 Marks)

- A. Union and Intersection
- B. Intersection and Union
- C. Intersection and Set Difference
- D. Union only

Ans: C. Intersection and Set Difference

Verification:

The operation (+) in mathematics is “**mod 2 operation**” thus it has both divide and consideration of remainder part.

Therefore,

Divide: Represents that **intersection** is taking place as we are basically taking the common part by division.

Remainder: Represents the **set difference** as we consider the remainder part after division and not the quotient part.

Now in DBMS,

The operation (+) thus has both intersection and set difference as it takes the common of both tables and in the given case it does Left Outer Join for the relations.

Thus we can say: $\sigma_{\theta(E1 (+) E2)} \equiv \sigma_{\theta(E1) (+) E2}$ and thus C is the answer.

Hence proved.

4. Consider the query evaluation tree of the following SQL statement. (5 Marks)

**Select P.PNumber, P.DNum, E.Last, E.Address, E.DOB
From Employee as E, Department as D, Project as P
Where P.DNum = D.DNumber and D.MgrSSN = E.SSN and
P.Location='PGH';**

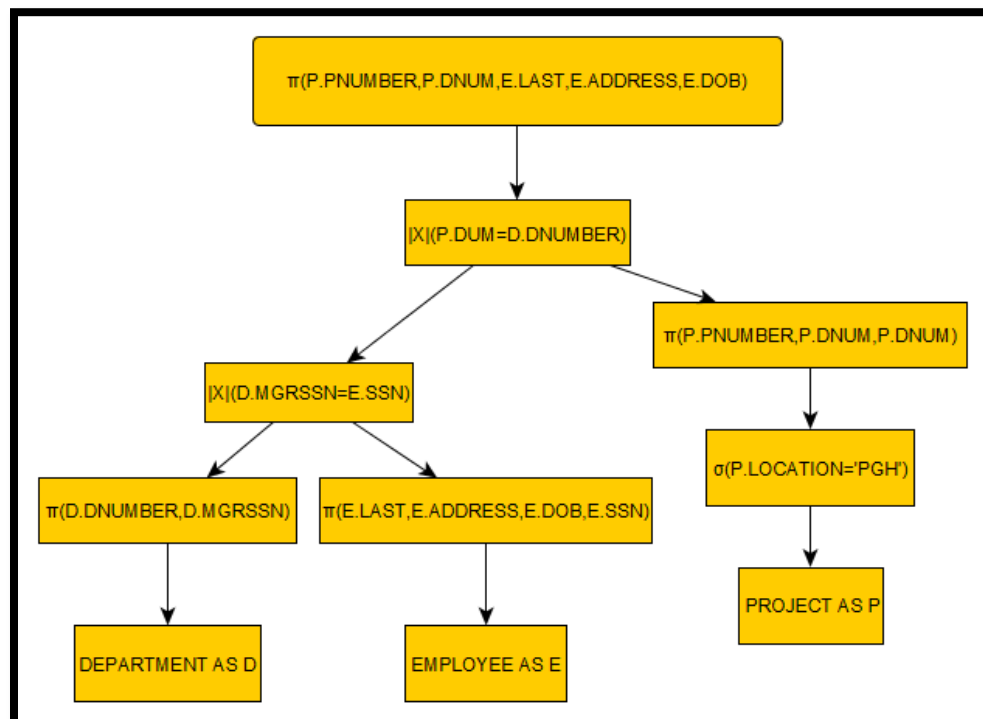
Further, consider four possible organizations of the Employee-Department-Project database. For each database organizations above, identify subtrees that represent groups of operations which can be executed by a single algorithm. For each such subtree, there are potentially multiple methods that can be combined to implement it as a single algorithm.

Ans:

The given query:

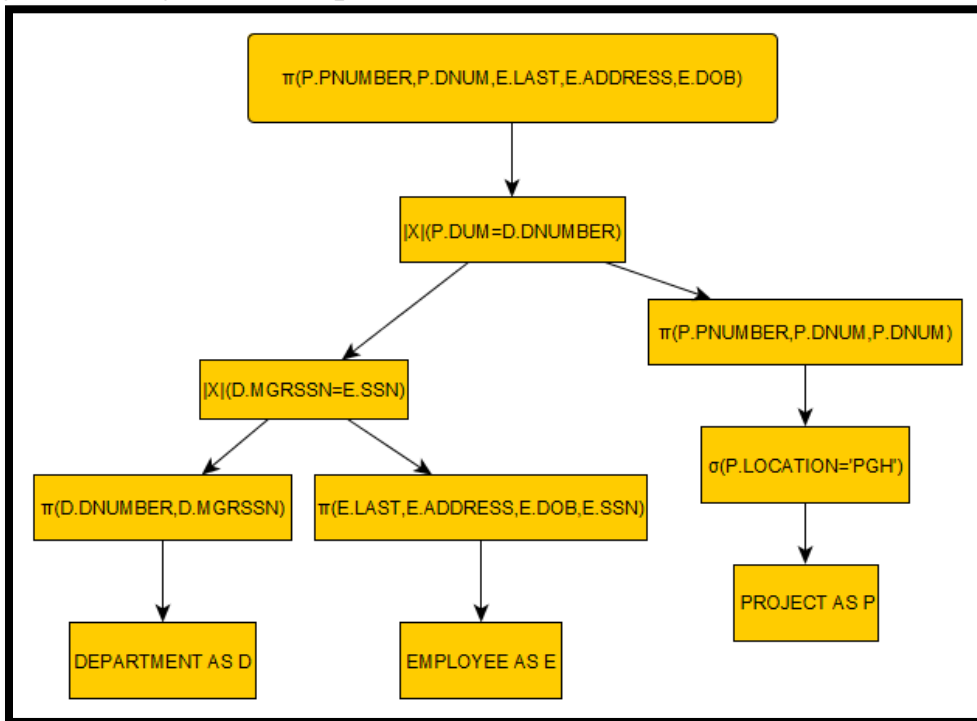
**Select P.PNumber, P.DNum, E.Last, E.Address, E.DOB
From Employee as E, Department as D, Project as P
Where P.DNum = D.DNumber and D.MgrSSN = E.SSN and
P.Location='PGH';**

The Final Stage of the Query Evaluation Tree:

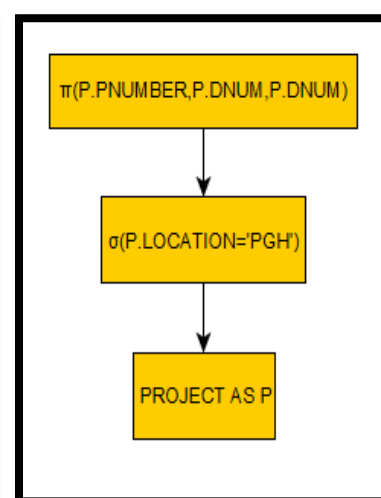
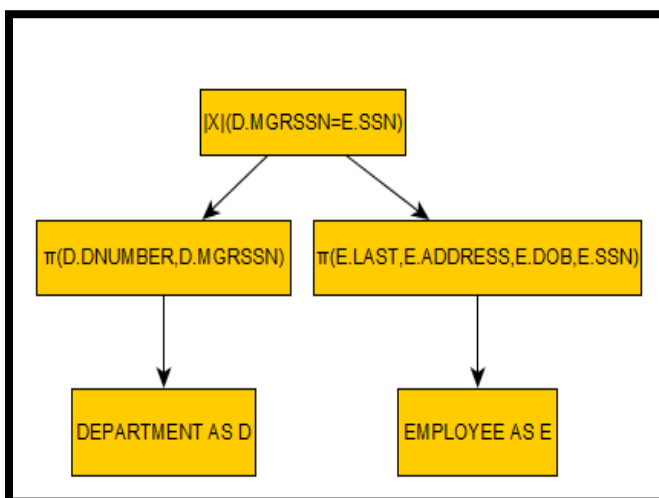


Four possible organizations of the Employee-Department-Project database can be:- (Note here right side represents the higher priority subtree)

A. Here we are checking for the Project's Location = 'PGH' first and then we check the join operation of Department and Employee Table followed by join of Project and Department table.



Therefore the sub trees are:

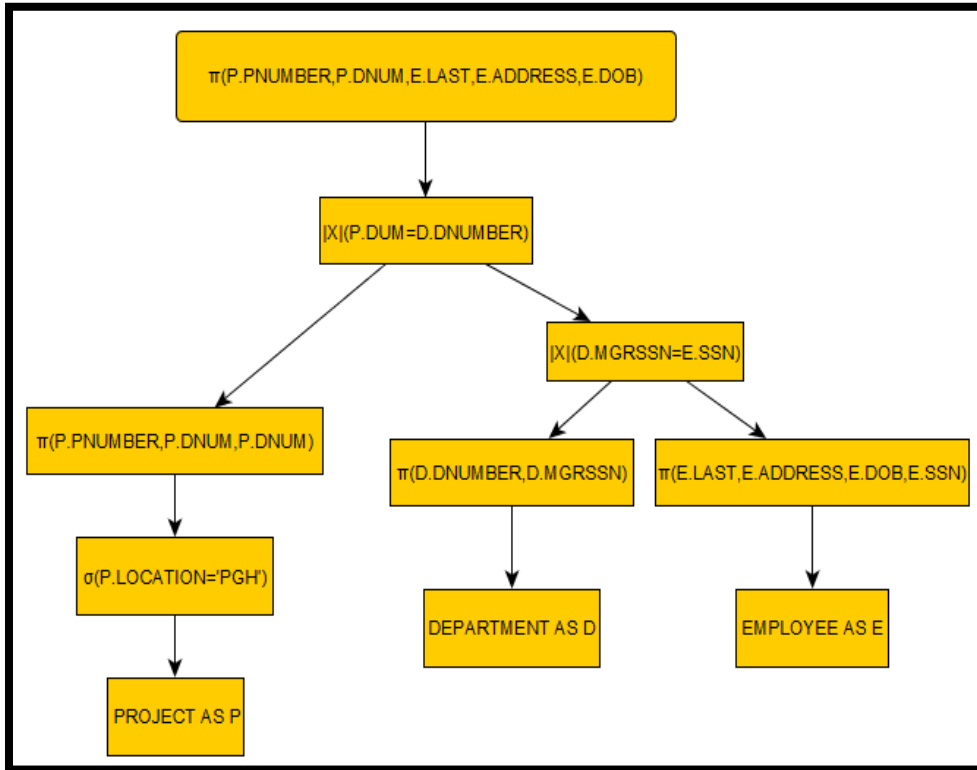


These two sub trees are joined by the P.DUM=D.DNUMBER.

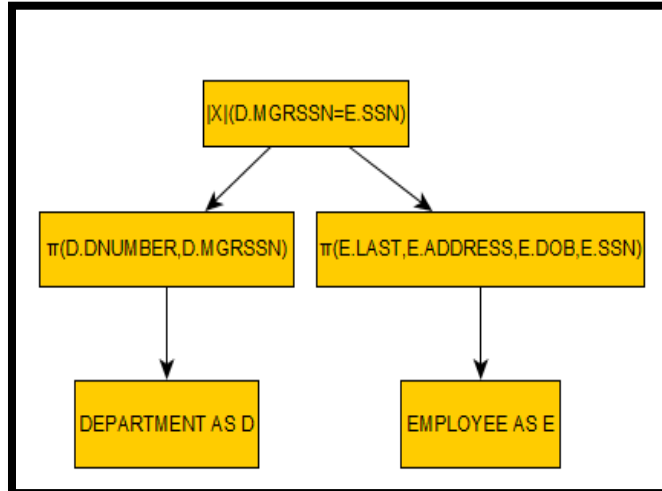
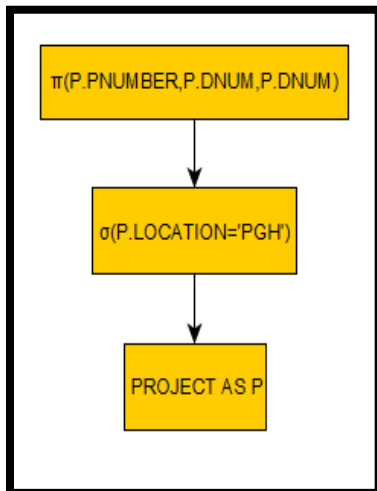
THE ALGORITHM:

```
<Returntype> function project(...), department(...), employee(...);
Class DBMS
{
.
.
.
If(P.Pnumber==D.DNUMBER)
{
If(P.Location=="PGH")           //Sub tree 1
{
Store the values of the P.Pnumber,P.dum in respective variables from
project(...) function;
.
.
If(E.SSN==D.MGRSSN)             //Sub tree 2
{
Store the values of the E.Last, E.DOB, E.SSN, E.Address from
employee(...) function;
Store the values of the D.MGRSSN, D.DNUMBER from department(...)
function;
}
}
}
.
.
.
}
```


B. Here we are checking for the join operation of Department and Employee Table first followed by join of Project and Department table and then we check for the Project's Location = 'PGH'



Therefore the sub trees are:

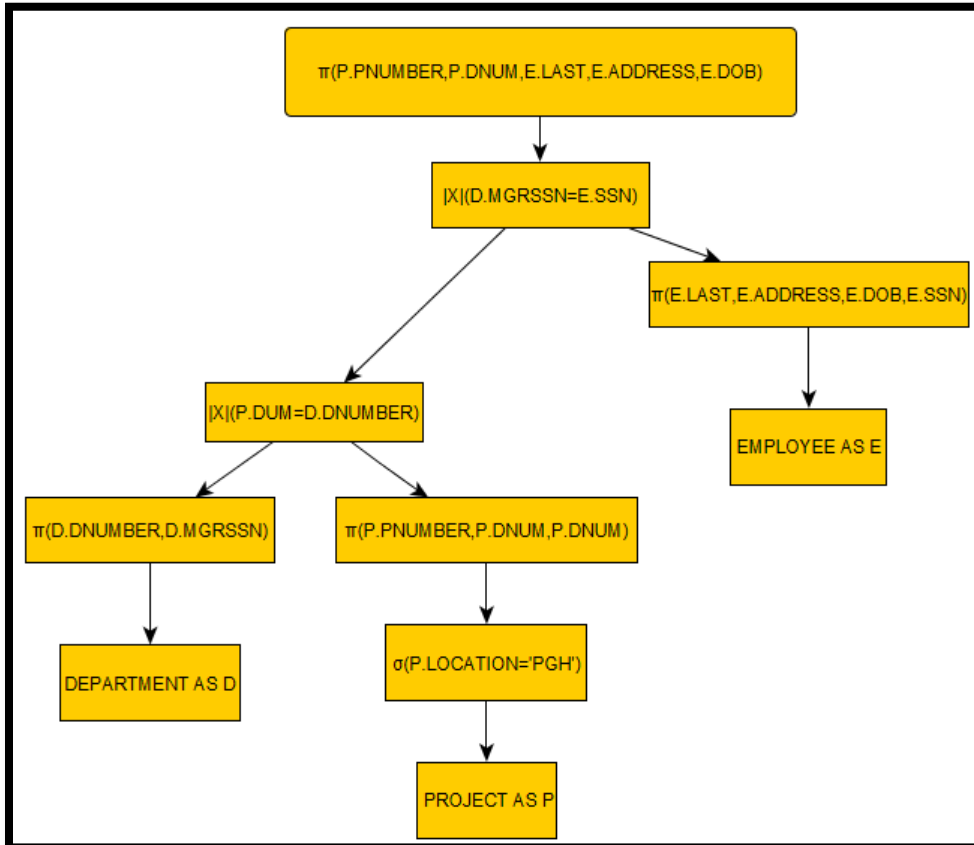


These two sub trees are joined by the P.DUM=D.DNUMBER.

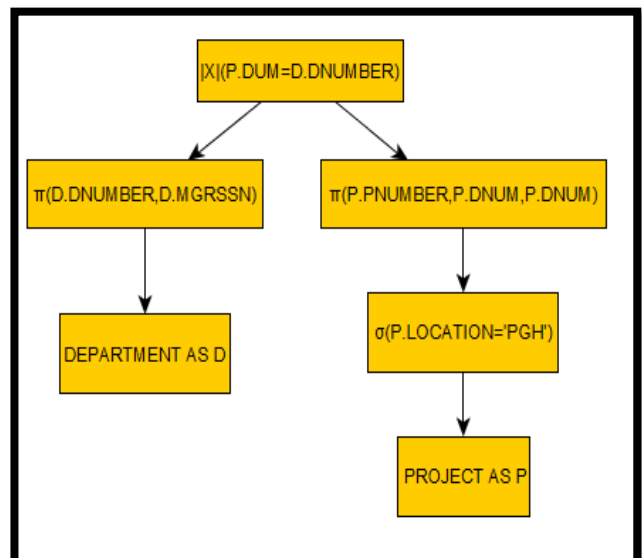
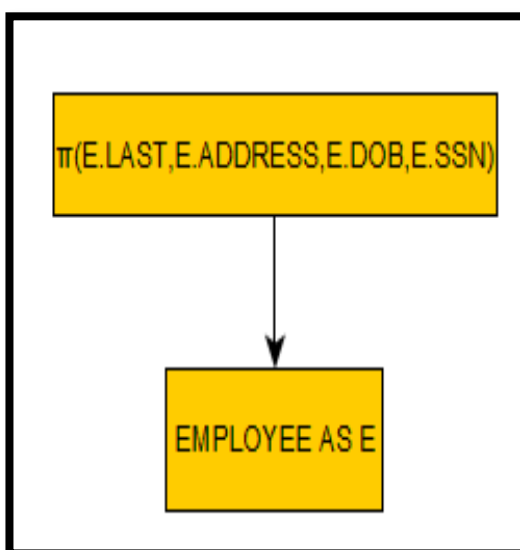
THE ALGORITHM:

```
<Returntype> function project(...), department(...), employee(...);
Class DBMS
{
.
.
.
If(P.Pnumber==D.DNUMBER)
{
If(E.SSN==D.MGRSSN)           //Sub tree 1
{
Store the values of the E.Last, E.DOB, E.SSN, E.Address from
employee(...) function;
Store the values of the D.MGRSSN, D.DNUMBER from department(...)
function;
.
.
If(P.Location=="PGH")         //Sub tree 2
{
Store the values of the P.Pnumber,P.dum in respective variables from
project(...) function;
}
}
}
.
.
.
}
```

C. Here we are taking the values of employee table first and then we check for the Project's Location = 'PGH' followed by the join operation of Department and Project Table and then the join operation of Employee and Department table takes place.



Therefore the sub trees are:

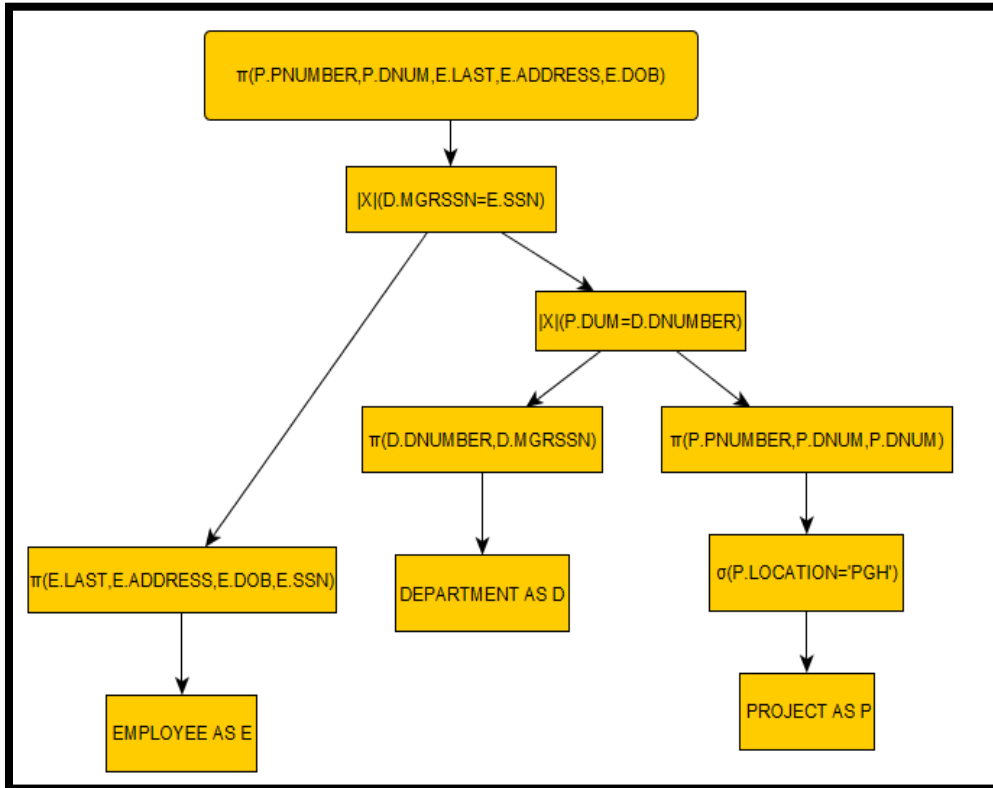


These two sub trees are joined by the D.MGRSSN=E.SSN.

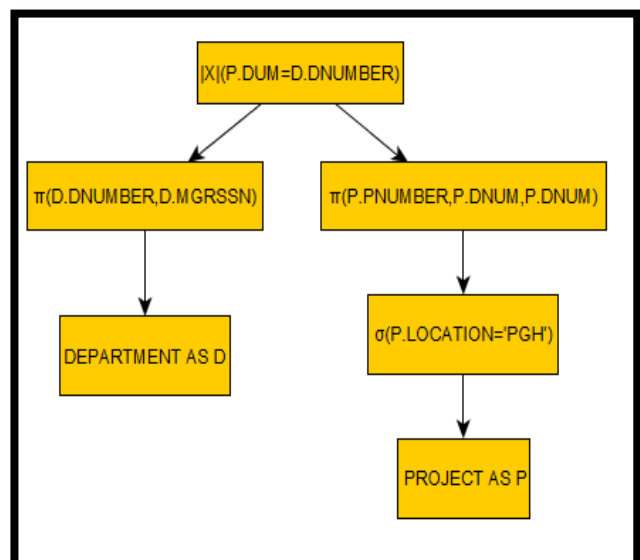
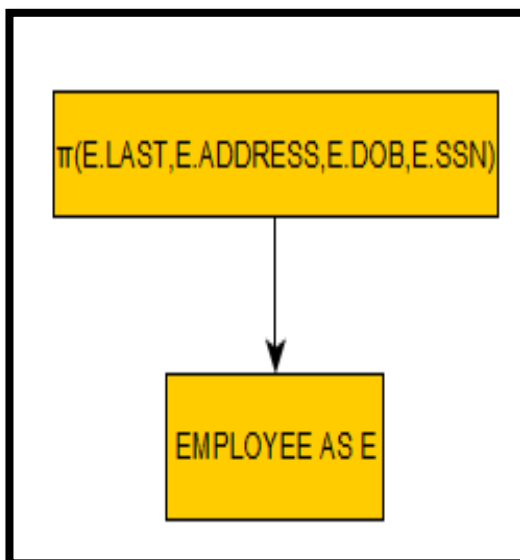
THE ALGORITHM:

```
<Returntype> function project(...), department(...), employee(...);
Class DBMS
{
.
.
.
If(E.SSN==D.MGRSSN)
{
Store the values of the E.Last, E.DOB, E.SSN, E.Address from
employee(...) function;                                     //Sub tree 1
.
.
If(P.Location=="PGH" && P.Pnumber==D.DNUMBER)              //Sub tree 2
{
Store the values of the P.Pnumber,P.dum in respective variables from
project(...) function;
Store the values of the D.MGRSSN, D.DNUMBER from department(...)
function;
}
}
}
.
.
.
}
```

D. Here we check for the Project's Location = 'PGH' followed by the join operation of Department and Project Table and then we take the values of Employee and the join operation of Employee and Department table takes place.



Therefore the sub trees are:



These two sub trees are joined by the D.MGRSSN=E.SSN.

THE ALGORITHM:

```
<Returntype> function project(...), department(...), employee(...);
Class DBMS
{
.
.
.
If(E.SSN==D.MGRSSN)
{
If(P.Location=="PGH" && P.Pnumber==D.DNUMBER)           //Sub tree 1
{
Store the values of the P.Pnumber,P.dum in respective variables from
project(...) function;
Store the values of the D.MGRSSN, D.DNUMBER from department(...)
function;
}
.
.
Store the values of the E.Last, E.DOB, E.SSN, E.Address from
employee(...) function;                                   //Sub tree 2
}
.
.
.
}
```

-----XXXX-----