

1. 1) Simplify where clause

Based on the given query let's assume

$$P_1 \Rightarrow R \text{ESP} = \text{"Analyst"}$$

$$P_2 \Rightarrow PNO = \text{"P2"}$$

$$P_3 \Rightarrow DUR = 12$$

Now, the 'where' clause is

$$\Rightarrow P_1 \wedge \neg(P_2 \vee P_3) \wedge \neg P_2 \wedge P_3$$

$$\Rightarrow P_1 \wedge \neg P_2 \wedge \neg P_3 \wedge \neg P_2 \wedge P_3 \quad (\text{using Normalization Rule 8})$$

$$\neg(P_2 \vee P_3) = \neg P_2 \wedge \neg P_3$$

$$\Rightarrow P_1 \wedge \neg P_2 \wedge \neg P_2 \wedge \text{false} \quad (\text{using Simplification Rule 7})$$

$$\neg P_3 \wedge P_3 = \text{false}$$

$$\Rightarrow \text{false} \quad (\text{using Simplification Rule 5, } P \wedge \text{false} = \text{false})$$

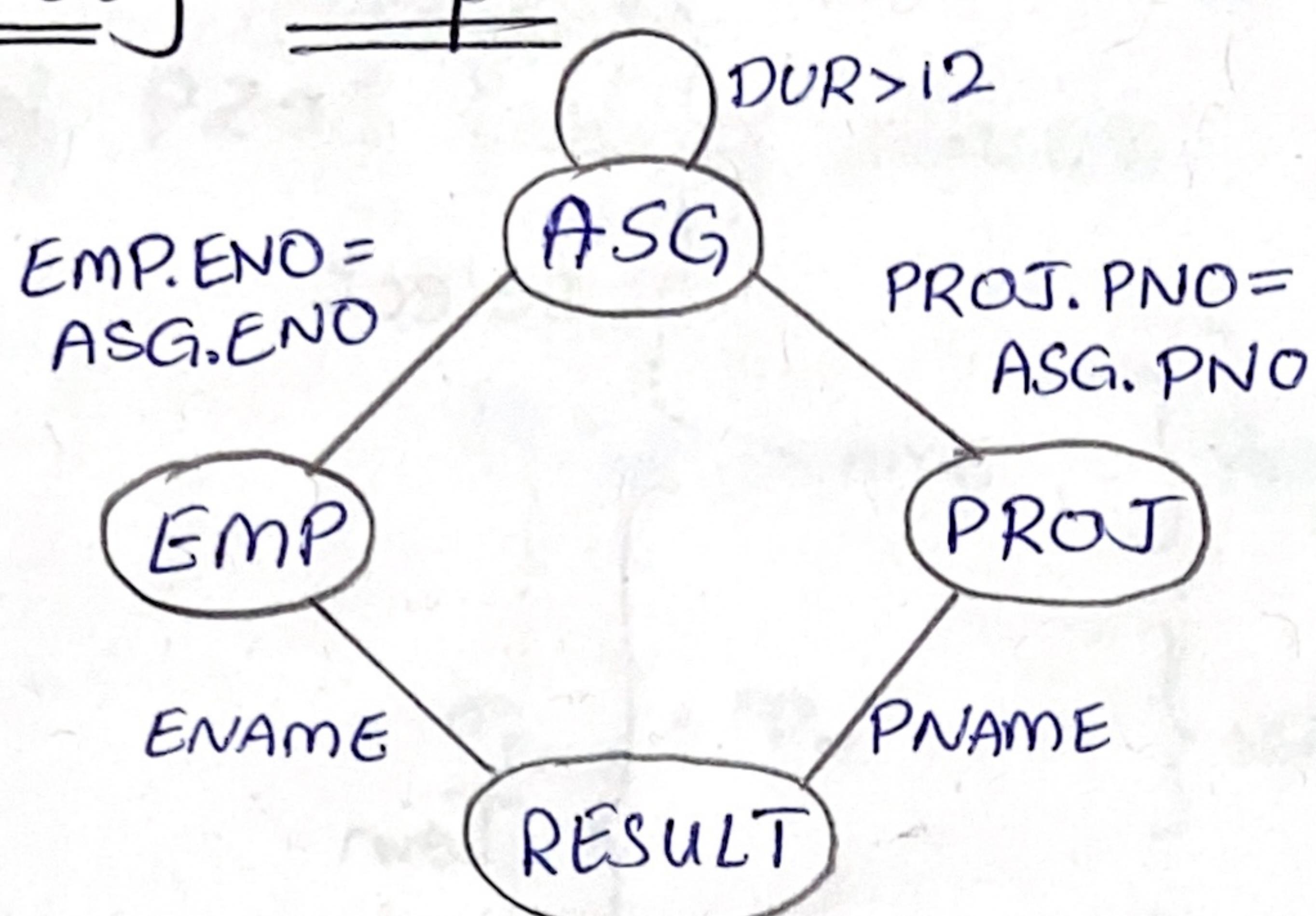
1. 2) Simplified Query is as follows

```
SELECT ENO
FROM ASG
WHERE FALSE.
```

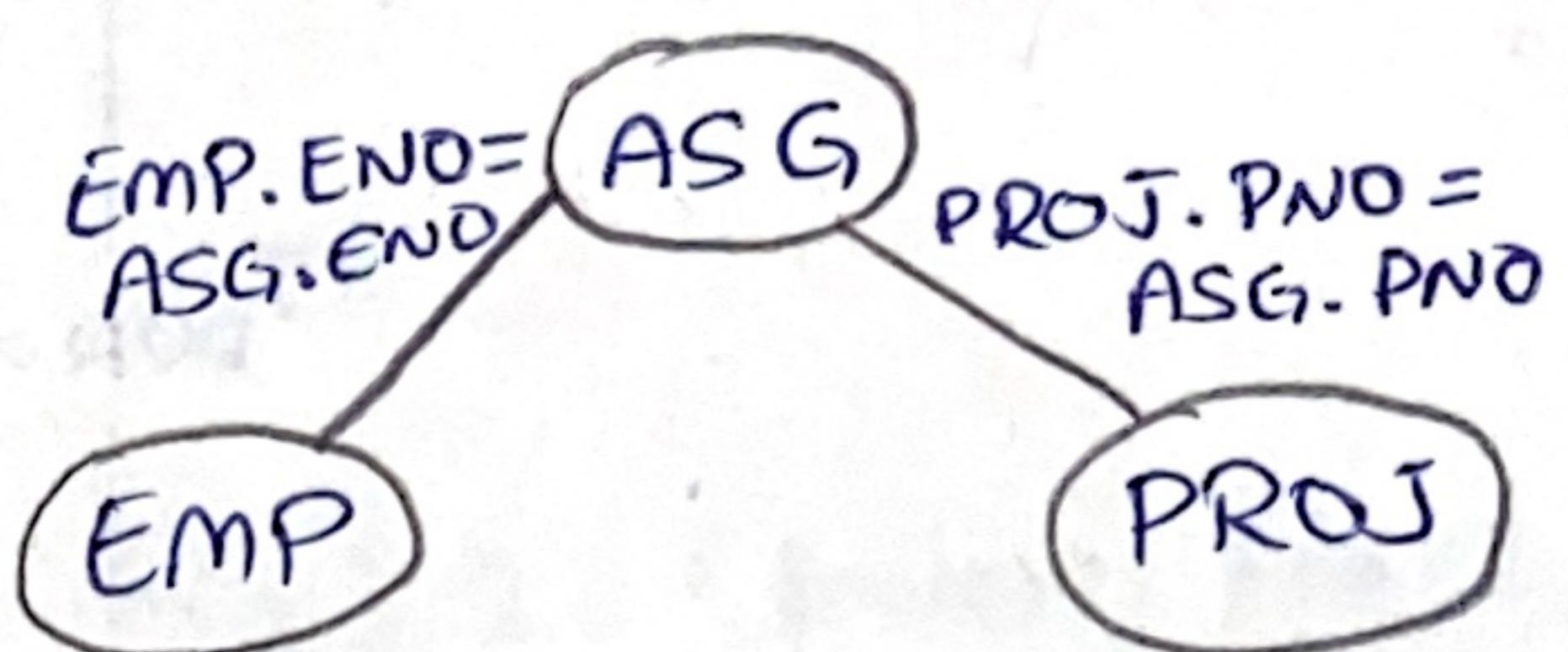
The query will not return anything because the where clause condition is never true.

So, the query result is an EMPTY SET.

2. 1) Query Graph:



Join graph:



2. 2) Generic Query (operator) tree:

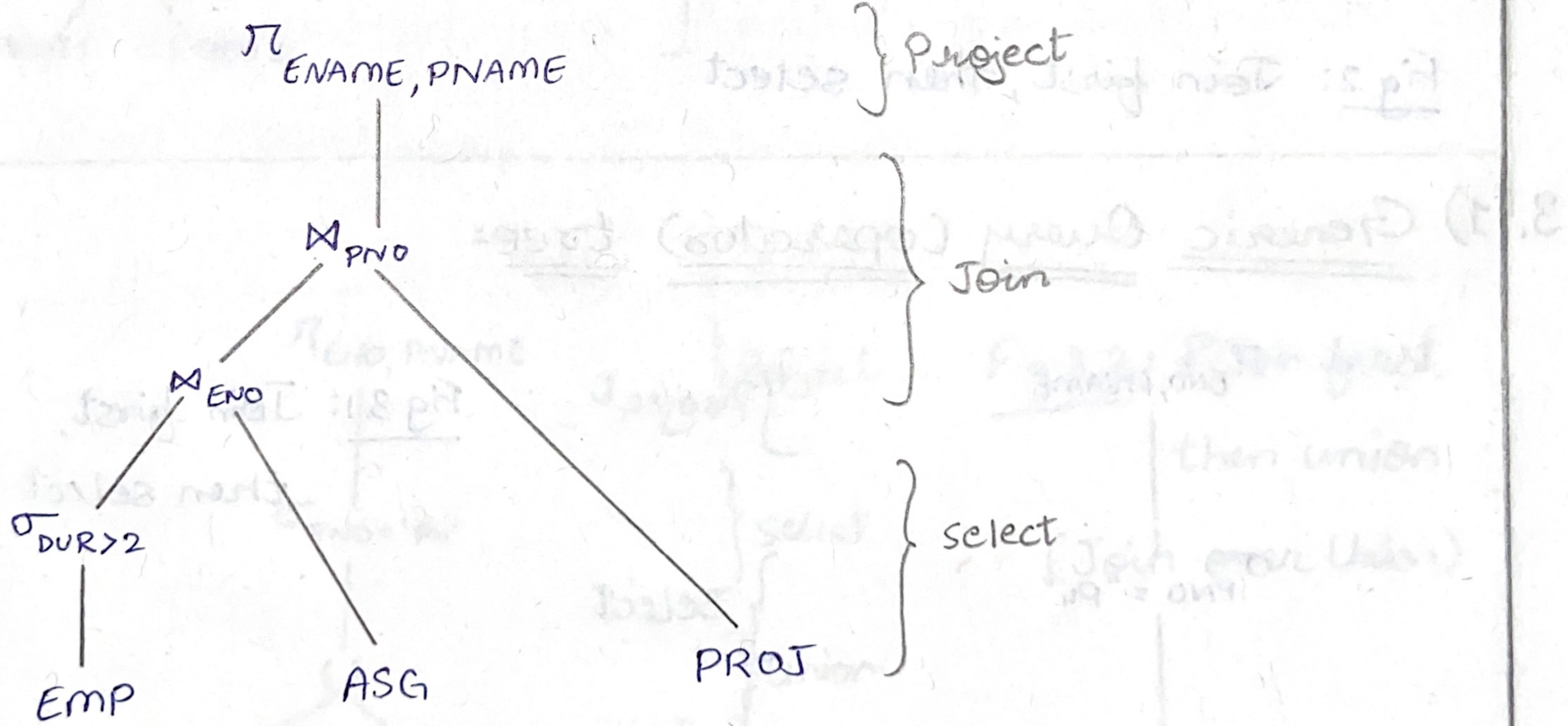


Fig 1: First select, then join.

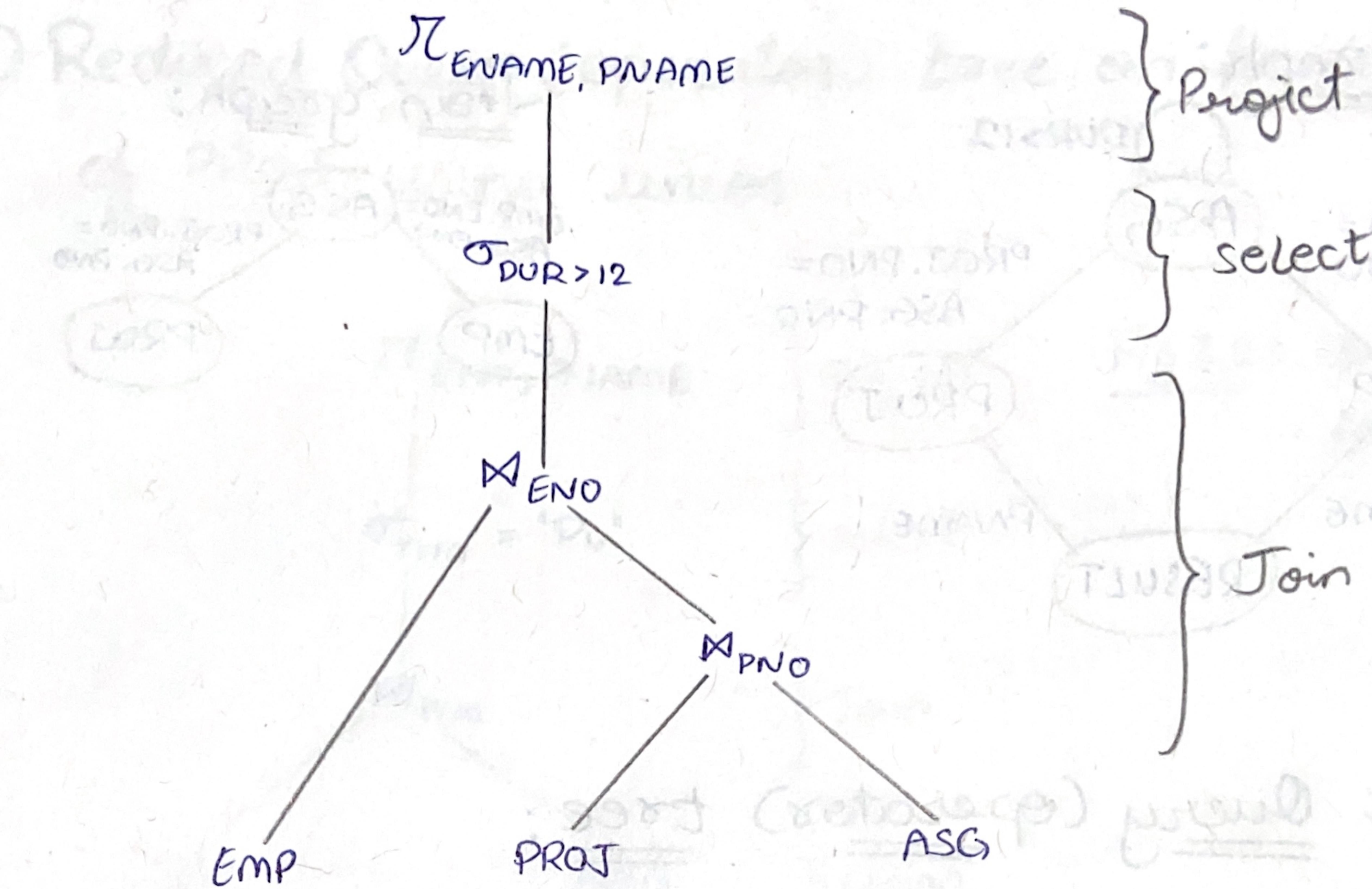


Fig 2: Join first, then select.

3. 1) Generic Query (operators) tree:

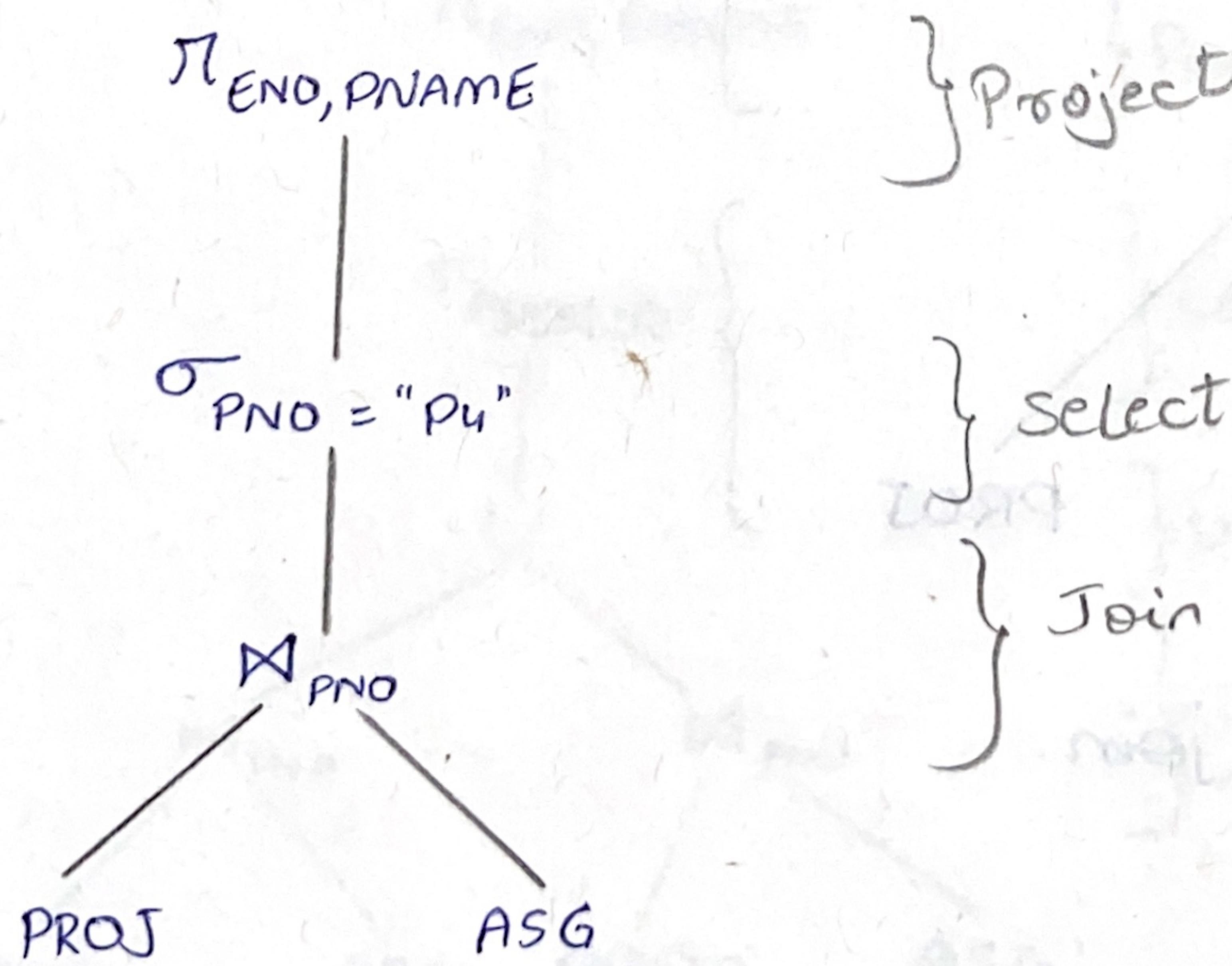


Fig 3.1: Join first,
then select.

3. 2) Reduced Query (operator) tree on fragments of PROJ using union.

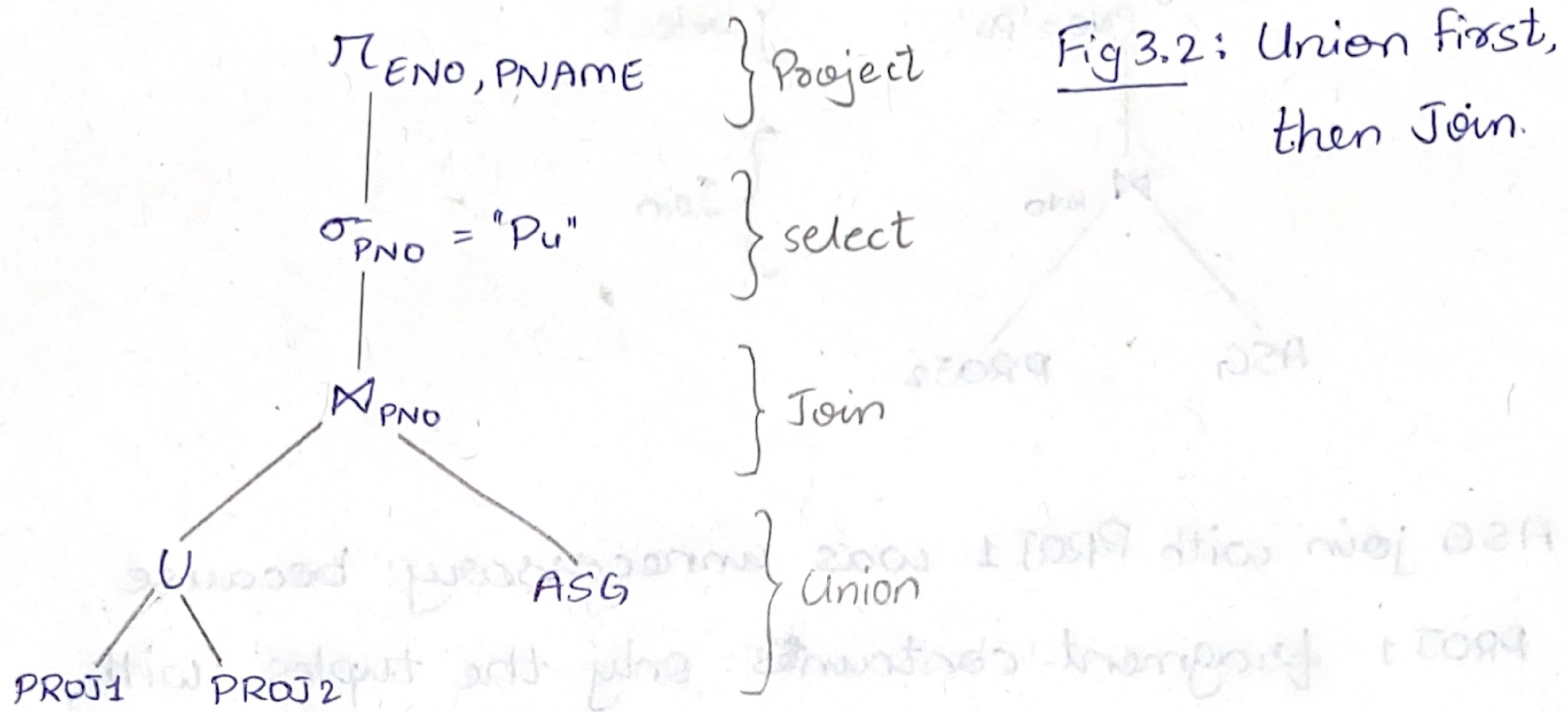


Fig 3.2: Union first, then Join.

3. 3)

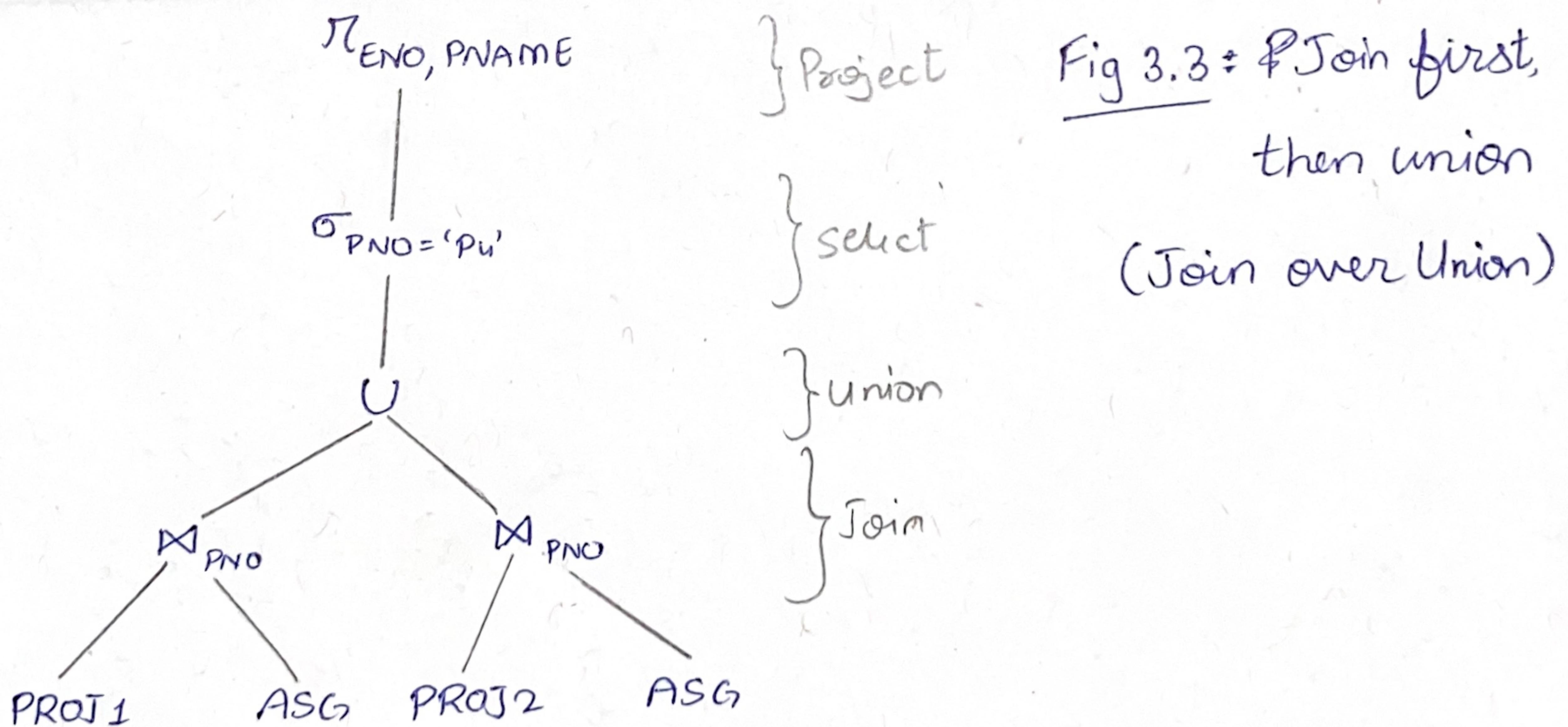
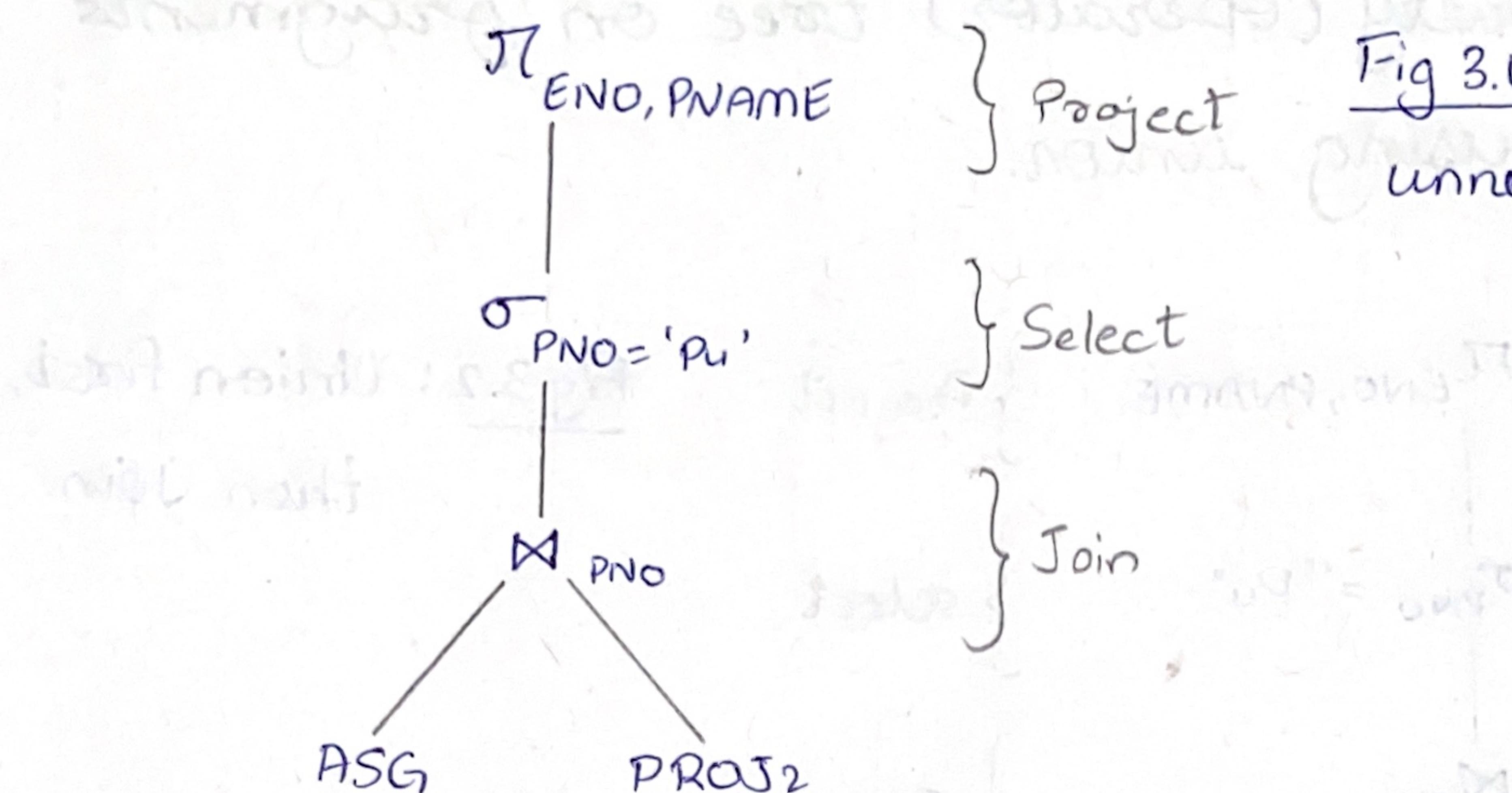


Fig 3.3: Join first, then union
(Join over Union)

3.4)



ASG join with PROJ1 was unnecessary because PROJ1 fragment contains only the tuples with $PNO \leq 'P2'$ and we want to select tuples with $PNO = 'P1'$.