Question 1:

1. π model(σ speed ≥ 3(PC))

Result: 1005, 1006, 1013

#### π maker(Product ⨝ σ hd ≥ 100(Laptop))

#### Result: A, B, E, F

#### π model, price(π model, price(PC) ∪ π model, price(Laptop) ∪ π model, price(Printer)) ⨝ (σ maker = 'B'(Product))

#### Result: PC.model (1004, 1005, 1006, 2007)

#### PC.price (649, 630, 1049, 1429)

#### π model(σ type = 'laser' and color = true(Printer))

#### Result: Printer.model (3003, 3007)

#### π maker(σ type = 'laptop'(Product)) - π maker(σ type = 'pc'(Product))

#### Result: Product Maker (F)

#### π A.hd( σ A.model ≠ B.model and A.hd = B.hd( ρ A(PC) ⨯ ρ B(PC)))

#### Result: 250, 80, 160

#### π PC1.model,PC2.model( σ PC1.speed = PC2.speed and PC1.ram = PC2.ram and PC1.model > PC2.model

#### (ρ PC1(PC)⨯ ρ PC2(PC)))

#### Result: 1012, 1004

#### π maker( π A.model( σ A.speed ≥ 2.8 and B.speed ≥ 2.8 and A.model ≠ B.model( ρ A( π model, speed(Laptop) ∪ πmodel, speed(PC)) ⨯ ρ B( π model, speed(Laptop) ∪ π model, speed(PC)))) ⨝ Product)

#### Result: B,D,E

#### 

#### R1: = π model, speed(PC)

#### R2: = π model, speed(Laptop)

#### R3: = R1 ∪ R2 = π model, speed(PC) U π model, speed(Laptop)

#### R4:= ρ R4( π model, speed(PC) ∪ π model, speed(Laptop))

#### R5:= π model,speed (π model, speed (PC) ∪ π model, speed (Laptop) ⨝ (speed<speed2) ρ R4 ( π model, speed (PC) ∪ π model, speed (Laptop)))

#### R6:= R3 – R5

#### R7:= π maker(R6 on Product)

#### Result: A, B

#### Question 2:

#### When we implement natural join, two relations would be combined by putting their common column of attributes together, removing their duplicate columns. However, when we implement theta join, each column from each relation despite sharing their common column of attributes will appear as their final result.

#### Question3:

#### Union: It is monotone because even if we add one arbitrary tuple either on both R or S relation, the number of whole tuples would rather increase.

#### Intersection: It is monotone because even if we add one arbitrary tuple either on both R or S relation, the number of intersection tuples would just stay the same or rather increase.

#### Natural Join: Even if new added tuple(S) does not have anything to do with given relation(R), new addition of tuple(S) will not decrease the number of previous natural join’s result.

#### Theta Join: As theta join is implemented under certain condition between the two relations, the addition of arbitrary tuple will not also decrease the number of resulting tuples unless the condition changes.

#### Selection: Selection takes place under certain condition within the relation, and addition of new tuple will not cause the previous result to decrease its number of tuples.

#### Projection: The addition of new tuple will not decrease the number of previous tuples resulting from the projection implementation.

#### \*\*Difference: If new tuple is added and being counted under the latter relation’s tuples(S), supposing R-S, then, the number of tuples resulting from new implementation would be lower than that of tuples from the previous implementation.

#### Question 4:

|  |  |  |
| --- | --- | --- |
|  | Minimum | Maximum |
| 1) | Max(m,n) | m+n |
| 2) | 0 | Min(m,n) |
| 3) | 0 | m\*n |
| 4) | 0 | n |