

Exercice 2 :

triangles (idt:INT, ab:INT, ac:INT, bc:INT)

Déterminer à l'aide de requêtes SQL

1) la plus petite valeur des produits $AB * AC * BC$ pour les triangles ABC de périmètre supérieur ou égal à 100 ;

```
SELECT MIN(ab*ac*bc)
FROM triangles
WHERE ab+bc+ac>=100
```

2) les longueurs AB, AC et BC correspondants au(x) triangle(s) pour le(s)quel(s) le minimum précédent est atteint ;

```
SELECT ab,ac,bc
FROM triangles
WHERE ab+bc+ac>=100
AND ab*ac*bc IN ( SELECT MIN(ab*ac*bc) FROM triangles WHERE ab+bc+ac>=100)
```

3) tous les triangles rectangles en A ;

```
SELECT *
FROM triangles
WHERE ab*ab + ac*ac = bc*bc
```

4) le nombre de tels triangles ;

```
SELECT COUNT(*)
FROM triangles
WHERE ab*ab + ac*ac = bc*bc
```

5) le maximum des périmètres des triangles rectangles en A ;

```
SELECT MAX(ab+bc+ac)
FROM triangles
WHERE ab*ab + ac*ac = bc*bc
```

Exercice 1 :

1) la clé primaire : IP (unique NOT null)

2) Algèbre relationnelle

3-a) adresse de la machine s'appelant 'PC_Math' :

```
SELECT IP
FROM Parc
WHERE nom = 'PC_Math'
```

3-b) Tous les noms de machines en ordre alphabétique et avec leurs adresses IP

```
SELECT nom,IP
FROM Parc
ORDER BY nom ASC
```

3-c) Pour chaque constructeur le nombre de ses modèles différents.

```
SELECT constructeur,COUNT(DISTINCT Modele)
FROM Catalogue
GROUP BY constructeur
```

3-d) Pour chaque constructeur ayant au moins 2 modèles différents afficher le nombre de ses machines.

```
SELECT constructeur,COUNT(*)
FROM Catalogue
GROUP BY constructeur
HAVING COUNT(DISTINCT Modele)>1
```

Exercice 3 :

1. Donner l' instruction SQL permettant de supprimer tous les produits de couleur noire et de numéro compris entre 100 et 199

```
DELETE FROM Produit
WHERE NP >= 100 AND NP <= 199 AND couleur = 'noir';
```

2. Donner l' instruction SQL permettant de changer la ville du fournisseur n°1. Il a déménagé pour Rabat.

```
UPDATE Fournisseur
SET Ville = 'Rabat'
WHERE NF = 1 ;
```

3. Ecrivez en **SQL** les requêtes pour :

- a. Donner de deux manières différentes le nom et la couleur des produits livrés par le fournisseur n° 1.

Méthode1 :

```
SELECT DISTINCT P.NomP, P.Couleur
FROM Produit P , PUF
WHERE PUF.NP = P.NP AND PUF.NF = 1;
```

Méthode 2 :

```
SELECT P.NomP, P.Couleur
FROM Produit P
WHERE P.NP IN ( SELECT PUF.NP
                FROM PUF
                WHERE PUF.NF = 1 );
```

- b. Donner de deux manières différentes les numéros des fournisseurs qui approvisionnent à la fois les usines n°1 et n°2.

Méthode1 :

```
SELECT DISTINCT first.NF
FROM PUF first, PUF second
WHERE first.NF = second.NF AND first.NU = 1 AND second.NU = 2;
```

Méthode2 :

```
SELECT DISTINCT PUF.NF
FROM PUF
WHERE PUF.NU = 2
AND PUF.NF IN ( SELECT NF
                FROM PUF
                WHERE NU = 1 )
```

- c. Donner de trois manières différentes le numéro du produit le plus léger (les numéros si plusieurs produits ont ce même poids).

Méthode1 :

```
SELECT P.NP
FROM Produit P
WHERE P.Poids IN ( SELECT MIN(Q.Poids)
                  FROM Produit Q )
```

Méthode2 :

```
SELECT p1.NP
FROM Produit p1
WHERE NOT EXISTS ( SELECT *
                  FROM Produit p2
                  WHERE p1.Poids > p2.Poids )
```

Méthode3 :

```
SELECT p.NP
FROM Produit p
WHERE p.Poids <= ( SELECT q.Poids
                  FROM Produit q )
```

- d. Donner les numéros des produits qui sont livrés à toutes les usines de Casablanca.

```
SELECT DISTINCT p.NP
FROM Produit p, PUF , Usine u
WHERE p.NP = PUF.NP AND PUF.NU = u.NU AND u.Ville = 'casablanca'
GROUP BY p.NP
HAVING COUNT(DISTINCT PUF.NU) = (SELECT COUNT(u.NU)
                                FROM Usine u
                                WHERE u.Ville = 'casablanca')
```

- e. Donner les numéros des usines qui s'approvisionnent uniquement chez le fournisseur n°3.

```
SELECT u.NU
FROM Usine u
WHERE u.NU NOT IN ( SELECT PUF.NU
                    FROM PUF
                    WHERE PUF.NF <> 3 )
```

- f. Donner les numéros des usines qui utilisent au moins un produit disponible chez le fournisseur n°3 (c'est-à-dire un produit qu'il livre mais pas nécessairement à cette usine).

```
SELECT DISTINCT p.NU
FROM PUF p
WHERE p.NP IN ( SELECT k.NP
                FROM PUF k
                WHERE k.NF = 3 )
```

- g. Donner les numéros des produits livrés à une usine de Casablanca par un fournisseur de Casablanca.

```
SELECT DISTINCT PUF.NP
FROM PUF , Fournisseur f, Usine u
WHERE PUF.NF = f.NF AND PUF.NU = u.NU
AND f.Ville = 'Casablanca' and u.Ville = 'Casablanca'
```

- h. Donner les numéros des usines qui ne reçoivent aucun produit rouge d'un fournisseur de Casablanca..

```
SELECT u.NU
FROM Usine u
WHERE u.NU NOT IN ( SELECT PUF.NU
                    FROM PUF , Produit p, Fournisseur f
                    WHERE PUF.NP = p.NP AND PUF.NF = f.NF
                    AND p.Couleur = 'rouge' AND f.Ville = 'Casablanca' )
```

- i. Donner les triplets (VilleF, NP, VilleU) tels qu'un fournisseur de la première ville approvisionne une usine de la deuxième ville avec un produit NP (mais sans les triplets où les deux villes sont identiques).

```
SELECT DISTINCT f.Ville, PUF.NP , u.Ville
FROM PUF , Usine u, Fournisseur f
WHERE f.Ville <> u.Ville
      AND PUF.NF = f.NF
      AND PUF.NU = u.NU
```