Solution du travail pratique 4.

Remarque : SQL est un langage fortement redondant. Cela signifie qu'il y a toujours plusieurs manières de résoudre une requête. Vous pouvez donc avoir une réponse parfaitement correcte qui ne figure pas dans cette correction.

1) On veut le nom des pièces qui contiennent la lettre 'c' majuscule ou minuscule dans leur nom

select distinct pname from p where (instr(upper(pname),'C')) <> 0

```
PNAME
Cam
Cog
Screw
```

2) Sachant que le poids des pièces est exprimé en livres et qu'une livre vaut 454 grammes on veut connaître les noms des pièces dont le poids est inférieur ou égal à 6 kilos.

```
select pname
 from p
where weight * 454 <=6000
```

```
PNAME
Nut
Cam
```

3) On souhaite le nom des pièces dont le poids est strictement inférieur à 18 livres et qui sont stockées soit à Rome, soit à Londres.

```
select distinct pname
 from p
 where weight <18 and( city ='Rome' or city='London')
```

```
PNAME
Nut
Screw
```

Remarque : les parenthèses sont nécessaires.

```
select distinct pname
 from p
 where weight <18 and city ='Rome' or city='London'
```

```
PNAME
Cog
Nut
Screw
```

4) Obtenir le nom des projets dont le nom contient une lettre 'i'

select jname from j where (instr(upper(jname),'I')) <> 0

```
JNAME
Display
RAID
```

5) Afficher les noms des fournisseurs qui commencent soit par 'A' soit par 'C'

select sname from s where upper(substr(sname,1,1)) in ('C','A')

```
SNAME
Adams
Clark
```

6) obtenir les triplets (id_s,id_p,id_j_) possibles tel que le fournisseur, la ville et le projet n'aient pas tous la même ville. (il ne faut pas spécialement qu'il y ait eut une livraison associée à ce triplet)

```
select id_s,id_p,id_j
 from S,P,J
 where not  (s.city = p.city and p.city=j.city)
```

extrait de la solution

ID_S	ID_P	ID_J
S1	P1	J1
S1	P1	J2
S1	P1	J3
S1	P1	J4
S1	P1	J6
S1	P1	J8
S2	P1	J1
S2	P1	J2

S1,P1,J5 ne fait pas partie de la solution puisque toutes les villes sont Londres →

← S1,P1,J6 fait partie de la réponse puisque S1,P1 sont de Londres, mais J6 d'Oslo

7) Obtenir tous les triplet (numéro de pièce1, numéro de pièce2, numéro de fournisseur) tel que le fournisseur fournisse les deux pièces du triplet.

```
select  distinct spj1.id_p, spj2.id_p,spj1. id_s
 from spj spj1, spj spj2
 where spj1.id_s=spj2.id_s and spj1.id_p > spj2.id_p
```

ID_P	ID_P	ID_S
P2	P1	S5
P3	P1	S5
P3	P2	S5
P4	P1	S5
P4	P2	S5
P4	P3	S3
P4	P3	S5
P5	P1	S5
P5	P2	S5
P5	P3	S2
P5	P3	S5
P5	P4	S5
P6	P1	S5
P6	P2	S5
P6	P3	S5
P6	P4	S5
P6	P5	S5

S5 fournit tous les couples possibles puisqu'il fournit tout.

S3 ne fournit que le couple P4,P3

S2 ne fournit que le couple P5,P3

8) obtenir les triplets (id_s,id_p,id_j_) possibles tel que le fournisseur, la pièce et le projet aient tous des villes différentes. (il ne faut pas spécialement qu'il y ait eut une livraison associée à ce triplet)

remarque : affichez aussi les villes correspondantes pour faciliter la vérification.

```
select id_s,S.city,id_p, p.city,id_j , j.city
from S,P,J
where   (s.city <> p.city and s.city<>j.city and p.city<>j.city)
```

extrait de réponses

ID_S	CITY	ID_P	CITY	ID_J	CITY
S2	Paris	P1	London	J2	Rome
S2	Paris	P1	London	J3	Athens
S2	Paris	P1	London	J4	Athens
S2	Paris	P1	London	J6	Oslo
S3	Paris	P1	London	J2	Rome
S3	Paris	P1	London	J3	Athens
S3	Paris	P1	London	J4	Athens
S3	Paris	P1	London	J6	Oslo
S5	Athens	P1	London	J1	Paris
S5	Athens	P1	London	J2	Rome
S5	Athens	P1	London	J6	Oslo
S1	London	P2	Paris	J2	Rome
S1	London	P2	Paris	J3	Athens