TP n° 15 : Correction

1 Avec une seule table

1.
```
SELECT code_dep
FROM communes
WHERE nom_commune = "TROYES"
```

2. Comme trois villes françaises s'appellent Allonnes, l'attribut `nom-commune` ne constitue pas une clé primaire.

3.
```
SELECT superficie, population
FROM communes
WHERE nom_commune = "ALLONNES" AND code_dep = "72"
```

4.
```
SELECT SUM(population)
FROM communes
```

5.
```
SELECT code_dep, SUM(population)
FROM communes
GROUP BY code_dep
```

6.
```
SELECT COUNT(*)
FROM communes
WHERE population >= 20000 AND population < 100000
```

7.
```
SELECT nom_commune, population
FROM communes
ORDER BY population
ASC LIMIT 10
```

2 Avec plusieurs tables

8.
```
SELECT code_dep, nom_commune, votants
FROM pres2017_t2
WHERE abstentions = 0;
```

9.
```
SELECT 100*SUM(votants)/SUM(inscrits)
FROM pres2017_t2
```

10.

```
SELECT 100*SUM(votants)/SUM(inscrits)
FROM communes AS c
JOIN pres2017_t2 AS p
ON c.code_dep = p.code_dep AND c.code_commune = p.code_commune
WHERE population < 1000
```

11.

```
SELECT COUNT(*)
FROM pres2017_t1 AS t1
JOIN pres2017_t2 AS t2
ON t1.code_dep = t2.code_dep AND t1.code_commune = t2.code_commune
WHERE t2.votants > t1.votants
```

12.

```
SELECT p17.nom_dep, COUNT(*)
FROM pres2017_t1 AS p17
JOIN pres2012_T1 AS p12
ON p17.code_dep = p12.code_dep AND p17.code_commune = p12.code_commune
WHERE p17.blancs + p17.nuls >= 2*p12.blancsnuls
GROUP BY p17.code_dep
```

13.

```
SELECT 100*SUM(macron)/SUM(votants)
FROM pres2017_t2 AS p17
JOIN communes AS c
ON p17.code_dep = c.code_dep AND p17.code_commune = c.code_commune
JOIN salaires AS s
ON c.code_dep = s.code_dep AND c.code_commune = s.code_commune
WHERE population > 1000 AND salaire > 20
```

14.

```
SELECT code_dep, SUM(population)
FROM communes
GROUP BY code_dep
HAVING SUM(population) > 1000000
```

15.

```
SELECT nom_commune, COUNT(*) AS nb
FROM communes
GROUP BY nom_commune
HAVING nb >= 10
```

3 Pour aller plus loin

16. Cette requête permet de déterminer le nom des communes qui ont la même altitude moyenne que Troyes.

17.

```
SELECT AVG(pop_tot)
FROM (SELECT SUM(population) AS pop_tot
      FROM communes
      GROUP BY code_dep)
```

18. Avec sous-requête :

```
SELECT code_dep, MIN(pop_tot)
FROM (SELECT code_dep, SUM(population) AS pop_tot
      FROM communes
      GROUP BY code_dep)
```

ou sans sous-requête :

```
SELECT code_dep, SUM(population) AS pop_tot
FROM communes
GROUP BY code_dep
ORDER BY pop_tot
LIMIT 1
```

19. Avec sous-requête :

```
SELECT soustable.code_dep, nom_dep, MIN(pop_tot)
FROM (SELECT code_dep, SUM(population) AS pop_tot
      FROM communes
      GROUP BY code_dep) AS soustable
JOIN pres2017_t2 AS pres
ON pres.code_dep = soustable.code_dep
```

ou sans :

```
SELECT com.code_dep, nom_dep, SUM(population) AS pop_tot
FROM communes AS com
JOIN pres2017_t2 AS pres
ON com.code_dep=pres.code_dep AND com.code_commune=pres.code_commune
GROUP BY com.code_dep
ORDER BY pop_tot
LIMIT 1
```