

# TP 7

CORRECTION

## I - Centrale 2021

Q14.

```
1 SELECT sc_nom FROM Scene WHERE sc_creation=2021
```

Q15.

```
1 SELECT Scene.sc_id,COUNT(*) FROM Scene
2 JOIN Source ON Source.sc_id=Scene.sc_id
3 GROUP BY Scene.sc_id
```

Q16.

```
1 SELECT ob_id ,ob_x,ob_y,ob_z, sp_rayon FROM Scene
2 JOIN Objet ON Source.sc_id = Objet.sc_id
3 JOIN Sphere ON ob_id=sp_id
4 WHERE sc_nom = "woodbox"
```

Q17. On peut donc utiliser OCCULTE dans une condition. Il nous faut deux objets et une source, cela fait un certain nombre de liaisons, mais seuls les identifiants sont utiles en fait, les tables de jonction suffisent!

```
1 SELECT objr.ob_id, so_id, objo.ob_id
2 FROM Scene JOIN Source ON Source.sc_id = Scene.sc_id
3 JOIN Objet AS objr ON Source.sc_id = objr.sc_id
4 JOIN Objet AS objo ON Source.sc_id = objo.sc_id
5 WHERE sc_nom="woodbox" AND OCCULTE(sc_id, objr.ob_id, so_id, objo.ob_id)
```

## II - Mines 2021 - Marchons, marchons, marchons

Q1.

```
1 SELECT COUNT(*) FROM Participant WHERE ne>=1999 AND ne<=2003
```

Q2.

```
1 SELECT diff, AVG(duree) FROM Rando GROUP BY diff
```

Q3.

```
1 SELECT pnom
2 FROM Participant
3 WHERE diff_max < ( SELECT diff FROM Rando WHERE rid=42)
```

## Q4.

```

1 SELECT Rando. rid
2 FROM Rando JOIN Rando AS Rando2
3 ON Rando.rnom = Rando2.rnom AND Rando. rid <> Rando2. rid
4 GROUP BY Rando. rid

```

OU :

```

1 SELECT DISTINCT Rando. rid
2 FROM Rando JOIN Rando AS Rando2
3 ON Rando.rnom=Rando2.rnom AND Rando. rid<>Rando2. rid

```

## III - Centrale 2020

## Q19.

```

1 SELECT PH_id FROM Photo WHERE PH_larg*3 = PH_haut*4

```

Q20. La question est ambiguë, on décide de faire un comptage global des photos des deux mariés (et non un de chaque).

```

1 SELECT COUNT(*)
2 FROM Photo JOIN Personne
3 ON PH_auteur=PE_id
4 WHERE PE_prenom="Alice" OR PE_prenom="Bernard"

```

## Q21.

```

1 SELECT Photo.PH_id, PH_date
2 FROM Photo JOIN Decrit ON Decrit.PH_id=Photo.PH_id
3 JOIN Motcle ON Decrit.MC_id=Motcle.MC_id
4 WHERE EXTRACT (year FROM PH_date)<2006 AND MC texte="surf"

```

## Q22.

```

1 SELECT PE_prenom, PH_id
2 FROM Photo JOIN Present ON Present.PH_id=Photo.PH_id
3 JOIN Personne ON Present.PE_id=Personne.PE_id
4 WHERE PH_auteur=Personne.PE_id

```

Q23. On fait trois listes : les photos où Alice figure, celles où Bernard figure et celles où quelqu'un d'autre figure; puis on utilise l'intersection et la différence ensembliste.

```

1 (SELECT Photo.*
2 FROM Photo JOIN Present ON Present.PH_id = Photo.PH_id
3 JOIN Personne ON Present.PE_id = Personne.PE_id
4 WHERE Personne.prenom = "Alice" )
5
6 INTERSECT
7
8 (SELECT Photo.*
9 FROM Photo JOIN Present ON Present.PH_id = Photo.PH_id
10 JOIN Personne ON Present.PE_id = Personne.PE_id
11 WHERE Personne.prenom = "Bernard" )
12
13 EXCEPT
14
15 (SELECT Photo.*
16 FROM Photo JOIN Present ON Present.PH_id = Photo.PH_id
17 JOIN Personne ON Present.PE_id = Personne.PE_id
18 WHERE Personne.prenom <> "Alice" AND Personne.prenom <> "Bernard" )

```

**Q24.** Les tables concernant les mot-clef sont `Motcle` et `Decrit`; ce sont les seules qu'on envisage de faire évoluer. Pour satisfaire le point a, on conserve la même table `Decrit` mais `MC_id` désignera désormais toute une famille de mots-clef équivalents, dans les différentes langues.

La table `Motcle` est chargée de donner la traduction de cet identifiant dans les différentes langues :

Motcle	
MC_id	integer
MC_langue	integer
MC_texte	varchar(30)

La langue est indiquée par un code (0 anglais, 1 français, etc) qui est la constante à modifier dans les requêtes dont on parle au point b. Le couple (`MC_id`, `MC_langue`) caractérise l'enregistrement : c'est une clef primaire.

Par exemple en **Q21**, il suffit de préciser `MC_texte="surf" AND MC_langue=0`.

**Q25.** En supposant que anglais correspond au code 0 :

```
1 SELECT Photo.*
2 FROM Photo JOIN Decrit ON Decrit.PH_id = Photo.PH_id
3 JOIN Motcle ON Decrit.MC_id = Motcle.MC_id
4 WHERE MC_texte = "mountain" AND MC_langue = 0
```

## IV - Centrale 2018 - cinétique des gaz parfaits

**Q34.**

```
1 SELECT SI_DIM, COUNT(*) AS nombreSim FROM SIMULATION GROUP BY SI_DIM
```

**Q35.**

```
1 SELECT SI_NUM, COUNT(*) AS nombreSim, AVG(RE_VIT) as vitMoy FROM REBOND GROUP BY SI_NUM
```

**Q36.**

```
1 SELECT RE_DIR, 2*SUM(PA_M*RE_VP) as qteMvt
2 FROM REBOND JOIN PARTICULE ON PARTICULE.PA_NUM=REBOND.PA_NUM
3 WHERE SI_NUM=n
4 GROUP BY RE_DIR
```

## V - Mines 2018 - Mesure de la houle par des capteurs sur des bouées

Première question :

```
1 SELECT idBouee, nomSite FROM Bouee WHERE localisation = "Mediterranee"
```

Deuxième question :

```
1 SELECT idBouee FROM Bouee EXCEPT SELECT idBouee FROM Tempete
```

Troisième question : on part du principe qu'une même bouée peut apparaître dans plusieurs tempêtes et qu'on demande le max sur toutes les tempêtes. Le `GROUP BY` se fait sur `nomSite` car sur un même site il peut y avoir plusieurs bouées.

```
1 SELECT nomSite, MAX(Hmax)
2 FROM Bouee JOIN Tempete ON Tempete.idBouee=Bouee.idBouee
3 GROUP BY nomSite
```

## VI - X 2017 - ensemble de points

Q4.

```
1 SELECT idensemble FROM MEMBRE JOIN POINTS ON idpoint=id WHERE x=a AND y=b
```

Q5.

```
1 SELECT x, y FROM POINTS
2 JOIN MEMBRE AS ens1 ON ens1.idpoint=id
3 JOIN MEMBRE AS ens2 ON ens2.idpoint=id
4 WHERE ens1.idensemble=i AND ens2.idensemble=j
```

ou :

```
1 (SELECT x, y FROM POINTS JOIN MEMBRE ON idpoint=id WHERE idensemble=i)
2 INTERSECT
3 (SELECT x, y FROM POINTS JOIN MEMBRE ON idpoint=id WHERE idensemble=j)
```

Q6.

```
1 SELECT idpoint
2 FROM MEMBRE JOIN
3 (SELECT idensemble FROM MEMBRE
4 WHERE idpoint=(SELECT id FROM POINTS WHERE x=a AND y=b)) AS bonsensembles
5 ON bonsensembles.idensemble = MEMBRE.idensemble
```

## VII - X MP PC 2016 - réseaux sociaux

Q16.

```
1 SELECT id2 FROM LIENS WHERE id1=x
```

Q17.

```
1 SELECT nom, prenom FROM INDIVIDUS JOIN LIENS ON id=id2 WHERE id1=x
```

Q18.

```
1 SELECT LIENS.id1
2 FROM LIENS JOIN (SELECT * FROM LIENS WHERE id1=x) AS amisdex
3 ON LIENS.id2=amisdex.id2
4 WHERE LIENS.id1 <> x
```

ou :

```
1 SELECT amis2.id2
2 FROM LIENS AS amis1 JOIN LIENS AS amis2
3 ON amis1.id2=amis2.id1
4 WHERE amis1.id1=x AND amis2.id2 <> x
```

## VIII - Mines 2016 : propagation d'épidémies

Q5. Une clef primaire est un attribut ou ensemble d'attributs permettant d'identifier les enregistrements de façon unique et qui est minimale (pas d'attribut superflu).

Ici aucun attribut ne peut servir de clef primaire. Le couple (iso, annee) en est une.

Q6.

```
1 SELECT * FROM palu WHERE annee=2010 AND deces>=1000
```

Q7.

```
1 SELECT palu.nom, palu.cas/pays.pop*100000 AS incidence
2 FROM palu JOIN demographie
3 ON palu.iso=demographie.pays AND palu.annee=demographie.periode
4 WHERE annee=2011
```

On aurait aussi pu mettre deux conditions dans le WHERE (annee=2011 AND periode=2011).

**Q8.**

```
1 SELECT nom FROM palu WHERE annee=2010 ORDER BY cas DESC LIMIT 1 OFFSET 1
```

## IX - Mines 2015 - Tests de validation d'une imprimante

**Q8.**

```
1 SELECT nSerie FROM testfin WHERE Imoy<Imax AND Imoy>Imin
```

**Q9.** La condition porte sur un agrégat (AVG) donc on fait une sous-requête chargée de son calcul (la réponse est une valeur unique, ce qui permet la comparaison de chaque terme à cette valeur).

```
1 SELECT nSerie , Iec , fichierMes  
2 FROM testfin  
3 WHERE Iec <(SELECT AVG( Iec ) FROM testfin )
```

**Q10.**

```
1 SELECT testfin . nSerie , testfin . fichierMes FROM testfin JOIN  
2 (SELECT nSerie FROM testfin EXCEPT SELECT nSerie FROM production)  
3 AS lesBonsNumeros  
4 ON testfin . nSerie=lesBonsNumeros . nSerie
```