

TD Base de données - Extraits d'épreuves écrites

1 Extrait CCINP PSI 2024

Q1.

```
1 | SELECT id_Joueur FROM Joueur WHERE niveau > 1900 ;
```

Q2. On calcule dans une sous-requête le nombre total de parties commencées par la case 0.

```
1 | SELECT 100.0 * COUNT(*) / (SELECT COUNT(*) FROM Partie WHERE jeu LIKE 'a%' )  
2 | FROM Partie  
3 | WHERE jeu LIKE 'a%' AND resultat = 1
```

Q3.

```
1 | SELECT nom, prenom FROM Joueur  
2 | ORDER BY niveau DESC LIMIT 3 ;
```

Q4.

```
1 | SELECT nom, prenom, COUNT(*) AS nb  
2 | FROM Joueur JOIN Partie ON id_Joueur = id_joueur1  
3 | WHERE resultat = 1.0  
4 | GROUP BY id_Joueur  
5 | HAVING nb > 100  
6 | ORDER BY nb DESC ;
```

2 Extrait Mines-Ponts PSI 2024

Q1.

```
1 | SELECT DISTINCT auteur FROM corpus
```

Q2. On calcule dans une sous-requête le nombre total de caractères des textes de tout le corpus.

```
1 | SELECT ca.symbole, SUM(oc.nombreOccurrences) / ( SELECT SUM(nombreCaracteres) FROM corpus)  
2 | FROM caractere AS ca JOIN occurrences AS oc ON ca.idCar = oc.idCar  
3 | JOIN corpus AS co ON oc.idLivre = co.idLivre  
4 | WHERE langue = 'Français'  
5 | GROUP BY ca.symbole
```

3 Extrait CCINP PSI 2023

Q1.

```
1 | SELECT id FROM FONTES  
2 | WHERE nom = "Zurich" AND style = "romain" AND 10 <= taille AND taille <= 16
```

Q2.

```
1 SELECT c.fichier FROM CARACTERES AS C JOIN SYMBOLES AS S ON C.id_symbole = S.id
2 WHERE S.label = "A"
```

Q3.

```
1 SELECT COUNT(C.id) FROM CARACTERES AS C JOIN SYMBOLES AS S ON C.id_symbole = S.id
2 JOIN FONTES AS F ON C.id_fonte = F.id)
3 JOIN SYMBOLES AS S ON C.id_symbole = S.id
4 WHERE F.nom = "Zurich" AND F.style = "roman" AND F.taille >= 10 AND F.taille <= 16
5 GROUP BY s.label
```

4 Extrait Mines-Ponts PSI 2023

Q1.

```
1 SELECT COUNT(gid) FROM Glyphe WHERE groman = True
```

Q2.

```
1 SELECT G.gdesc
2 FROM Glyphe AS G JOIN Police AS P ON G.pid = P.pid
3 JOIN Caractere AS C ON C.code = G.code
4 WHERE C.car = 'A' AND P.pnom = 'Helvetica' AND G.groman = False
```

Q3.

```
1 SELECT F.fnom, COUNT(pid) FROM Famille AS F JOIN Police AS P ON F.fid = P.fid
2 GROUP BY P.fid
3 HAVING COUNT(pid) <>0
4 ORDER BY F.fnom
```

Remarque : dans cette dernière requête, le **HAVING** n'est pas vraiment nécessaire, puisque lors de la jointure, seule les familles qui possèdent effectivement des polices dans la table `Police` apparaîtront.

5 Mines-Ponts PSI 2022

Q1.

```
1 SELECT nom FROM materiaux WHERE t_curie > 500
```

Q2.

```
1 SELECT nom_fournisseur, prix_kg * 4.5 FROM fournisseurs
2 JOIN prix ON id_four = id_fournisseur
3 WHERE id_mat = 8713
```

Q3.

```
1 SELECT nom_fournisseur, prix_kg * 4.5 FROM fournisseurs
2 JOIN prix ON id_four = id_fournisseur
3 WHERE id_mat = 8713
4 AND prix_kg =
5 (SELECT MIN(prix_kg) FROM prix WHERE id_mat = 8713)
```

Q4.

```

1 | SELECT nom, AVG(prix_kg) AS moyenne FROM materiaux
2 | JOIN prix ON id_mat = id_materiau
3 | GROUP BY id_mat HAVING moyenne < 50

```

6 CCINP PSI 2021

Q1. L'ensemble des informations sont accessibles depuis la seule table activite.

```

1 | SELECT Ida FROM activite
2 | WHERE Idm = 1

```

Q2. De manière similaire, en nommant la vitesse moyenne Vmoyenne.

```

1 | SELECT Date, Distance, Distance/Temps*3.6 AS Vmoyenne FROM activite
2 | WHERE Idm = 1 AND Type = 'course'

```

Q3. Décrivons la requête suivante :

```

1 | SELECT activite.Ida FROM activite
2 | JOIN (SELECT membre1 AS idam1 FROM amis WHERE membre2 = 1
3 | UNION
4 | SELECT membre2 AS idam1 FROM amis WHERE membre1 = 1) AS amis1
5 | ON activite.idm = amis1.idam1
6 | WHERE Type = 'marche'

```

- La sous-requête `SELECT membre1 AS [] UNION SELECT membre2 AS []` renvoie l'union sans duplication des membres amis avec le membre n°1, autrement dit renvoie tous les membres amis avec le membre n°1.
- `activite JOIN (SELECT UNION) ON activite.idm=amis1.idam1` renvoie la jointure de la table `activite` avec la nouvelle table qui ne contient que les amis du membres n°1. Ainsi on dispose de toutes les activités des amis du membre n°1.
- `SELECT activice.Ida FROM activite [] WHERE Type='marche'` renvoie les identifiants des activités qui sont de type 'marche'.

Ainsi la requête renvoie les identifiants des activités de type 'marche' des amis du membre n°1.

7 Mines-Ponts PSI 2021

Q1 Nombre de participants nés entre 1999 et 2003 (inclus) :

```

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

```

Q2 Durée moyenne des randonnées pour chaque niveau de difficulté :

```

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

```

Q3 Nom des participants pour lesquels la randonnée n°42 est trop difficile :

- avec une sous-requête :

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

- avec un produit cartésien :

```
SELECT pnom FROM Participant,Rando
WHERE rid = 42 AND diff_max < diff
```

Q4 Clés primaires des randonnées qui ont un ou des homonymes, sans redondance :

- Première version, avec une auto-jointure :

```
SELECT DISTINCT R.rid
FROM Rando AS R JOIN Rando AS S
ON R.rnom=S.rnom
WHERE R.rid <> S.rid
```

- Deuxième version, avec un GROUP BY et un HAVING :

```
SELECT DISTINCT rid FROM Rando
WHERE rnom IN (SELECT rnom FROM rando
              GROUP BY rnom
              HAVING COUNT(*) > 1)
```

8 CCINP PSI 2019

Q1. C'est une sélection :

```
1 SELECT idpatient FROM MEDICAL WHERE etat = "hernie discale"
```

(on pourrait ajouter le mot clé DISTINCT si il y a plusieurs enregistrements pour un même patient dans la table MEDICAL).

```
1 SELECT DISTINCT idpatient FROM MEDICAL WHERE etat = "hernie discale"
```

Q2. Il faut faire une jointure avec la table PATIENT :

```
1 SELECT PATIENT.nom, PATIENT.prenom
FROM MEDICAL
3 JOIN PATIENT ON PATIENT.id = MEDICAL.idpatient
WHERE etat = "spondylolisthesis"
```

Q3. C'est une agrégation par état et l'application de la fonction d'agrégation COUNT pour chacun des états.

Si l'on considère qu'il n'y a qu'un enregistrement par patient dans MEDICAL :

```
SELECT etat, COUNT(*) FROM MEDICAL GROUP BY etat
```

sinon :

```
1 SELECT etat, COUNT(DISTINCT idpatient) FROM MEDICAL GROUP BY etat
```