

Devoir d'informatique

Durée : 2 heures

1 Reconnaissance de caractères

Q1. Écrire une requête SQL permettant d'extraire les identifiants des fontes dont le nom est "Zurich", de style "romain" et dont la taille est comprise entre 10 et 16 pixels.

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

Q2. Écrire une requête SQL permettant d'extraire tous les noms de fichiers des caractères qui correspondent au symbole de label "A".

```
1 | SELECT fichier
2 | FROM CARACTERES JOIN SYMBOLES
3 |   ON CARACTERES.id_symbole = SYMBOLES.id
4 | WHERE label = "A"
```

Q3. Écrire une requête SQL permettant d'indiquer le nombre de caractères correspondant à la fonte "Zurich", de style "romain" et dont la taille est comprise entre 10 et 16 pixels groupés selon les labels des symboles.

```
1 | SELECT count(CARACTERES.id)
2 | FROM FONTES
3 |   JOIN CARACTERES ON FONTES.id = CARACTERES.id_fonte
4 |   JOIN SYMBOLES ON SYMBOLES.id = CARACTERES.id_symbole
5 | WHERE nom = "Zurich" and taille >= 10 and taille <=16 and style = "romain"
6 | GROUP BY label
```

2 Données géographiques

Q1. Écrire la requête SQL qui permet d'avoir le nom, la latitude et la longitude des photos faites le 2020/06/16.

```
1 | SELECT Nom ,Latitude ,Longitude
2 | FROM Photos
3 | WHERE Date = "2020/06/16"
```

Q2. Écrire la requête SQL qui permet d'avoir le nom des photos faites le 2020/06/16 et entre les latitudes 35935 et 35940.

```
1 | SELECT Nom
2 | FROM Photos
3 | WHERE Date = "2020/06/16" and Latitude > 35935 and Latitude > 35940
```

Q3. Écrire la requête SQL qui permet de trouver le nom des photos prises le 2020/06/17 et situées dans le dossier C:\Images\Las Vegas\Bellagio.

```
1 | SELECT Nom
2 | FROM Photos
```

```

3 JOIN Dir
4 ON Photos.Nom = Dir.NomPhoto
5 WHERE Date = "2020/06/17" and Dossier = "C:\Images\Las Vegas\Bellagio"

```

Q4. Écrire la requête SQL qui permet de compter le nombre de photos situées dans le dossier C:\Images\Las Vegas\Bellagio.

```

1 SELECT COUNT(NomPhoto)
2 FROM Dir
3 WHERE Dossier = "C:\Images\Las Vegas\Bellagio"

```

Q5. Écrire la requête SQL qui permet de compter le nombre de photos situées dans chaque dossier. Cette requête renverra donc la table composée d'une première colonne dont le nom est le nom du dossier et d'une seconde colonne qui donne le nombre de photo pour le dossier en question.

```

1 SELECT Dossier ,COUNT(NomPhoto)
2 FROM Dir
3 GROUP BY Dossier

```

Q6. Écrire la requête SQL correspondant à la question précédente mais en limitant l'affichage aux dossiers contenant plus de 50 photos en les classant en partant du dossier le moins rempli vers le dossier le plus rempli.

```

1 SELECT Dossier ,COUNT(NomPhoto)
2 FROM Dir
3 GROUP BY Dossier
4 HAVING COUNT(NomPhoto)>50
5 ORDER BY COUNT(NomPhoto)

```

3 Gestion d'une entreprise

Q1. Écrire, en SQL, la requête (1) qui permet d'obtenir le numéro de carte de crédit de toutes les personnes référencées dans la base de données de l'entreprise dont le numéro de sécurité sociale commence par 2. On utilisera le caractère "_" comme le séparateur des milliers. Par exemple 10000000 sera réécrit comme 10_000_000.

```

1 SELECT num_CB
2 FROM clients
3 WHERE num_secu < 199_999_999_999_999 and num_secu < 300_000_000_000_000

```

Q2. Écrire, en SQL, la requête (2) permettant d'obtenir le nom et le prénom de toutes les personnes ayant effectué un achat avec un résultat sans doublon.

```

1 SELECT distinct(nom , prenom)
2 FROM clients

```

Q3. Écrire, en SQL, la requête (3) qui permet d'obtenir les produits associés à chaque numéro de carte de crédit du client et qui ont été vendus entre le 1 juin 2020 et le 30 juillet 2020. On rappelle que SQL compare les variables de type TEXT grâce à l'ordre lexicographique : par exemple "13-06-1989 < 13-07-1999" est vrai.

```

1 SELECT nom_produits , num_CB
2 FROM clients
3 JOIN ventes ON clients.id = ventes.num_client
4 JOIN produits ON ventes.ref_produit = produits.ref_produit

```

```
5 || WHERE date > "01-06-2020" and date < "30-07-2020"
```

4 Résultats de championnat de formule 1

Q1. Écrire une requête SQL qui liste, par ordre chronologique, la date et le nom du circuit de toutes les courses qui se sont déroulées en France (ci_pays = 'France').

```
1 || SELECT gp_date , ci_nom
2 || FROM Circuit AS CI JOIN GrandPrix AS GP ON CI.ci_id = GP.ci_id
3 || WHERE ci_pays = "France"
4 || ORDER BY GP.date
```

Q2. Écrire une requête SQL qui liste, pour chaque course de l'année 2021, le nom du circuit, le nom du pilote gagnant et son temps de course.

```
1 || SELECT ci_nom , pi_nom , sum(to_temps)
2 || FROM Participation
3 ||     JOIN Pilote ON Participation.pi_id = Pilote.pi_id
4 ||     JOIN GrandPrix ON Participation.gp_id = GrandPrix.gp_id
5 ||     JOIN Circuit ON GrandPrix.ci_id = Circuit.ci_id
6 ||     JOIN Tour ON Participation.pa_id = Tour.pa_id
7 || WHERE gp_date >= "01/01/2021" and gp_date <= "31/12/2021" and pa_cla = 1
8 || GROUP BY gp_id , pi_id
```

Q3. Écrire une requête SQL qui affiche, dans le sens décroissant, le nom des dix pilotes ayant le plus de victoire et le nombre de ces victoires.

```
1 || SELECT pi_nom , sum(pa_cla)
2 || FROM Participation
3 ||     JOIN Pilote ON Participation.pi_id = Pilote.pi_id
4 || WHERE pa_cla = 1
5 || GROUP BY pi_id
6 || ORDER BY sum(pa_cla) DESC
7 || LIMIT 10
```

Q4. Expliquer le résultat de la requête donnée.

La requête renvoie le nom du circuit, le nom du pilote, la date du grand prix et le temps du meilleur tour.

Le résultat de cette requête est donc le record historique de chacun des circuits ainsi que le pilote qui a réalisé cette performance.

5 Reconnaissance de caractères

Q1 Proposer une requête en SQL sur cette base de données pour compter le nombre de glyphes en roman (cf. description précédente).

```
1 || SELECT count(*)
2 || FROM Glyphe
3 || WHERE groman = True
```

Q2 Proposer une requête en SQL afin d'extraire la description vectorielle du caractère A dans la police nommée Helvetica en italique.

```
1 | SELECT gdesc
2 | FROM Glyphe
3 |     JOIN Caractere ON Glyphe.code = Caractere.code
4 |     JOIN Police ON Glyphe.pid = Police.pid
5 | WHERE groman = False and pnom = "Helvetica" and car = "A"
```

Q3 Proposer une requête en SQL pour extraire les noms des familles, classés par ordre alphabétique, ainsi que leur nombre de polices en limitant l’affichage aux familles ayant au moins deux polices.

```
1 | SELECT fnom , COUNT(pid)
2 | FROM Famille
3 |     JOIN Police ON Famille.fid = Police.fid
4 | WHERE groman = False and pnom = "Helvetica" and car = "A"
5 | GROUP BY fid
6 | HAVING COUNT(pid) >= 2
7 | ORDER BY fnom
```