

## Corrigé du DM n°2 ITC Partie SQL

## 1 Banque d'images. Extrait Centrale 2020

## III.A – Quelques requêtes

- Q 19. On peut utiliser les opérations +, -, \* et / dans les clauses **WHERE**. On obtient donc ici :

```
SELECT PH_id FROM Photo
WHERE PH_larg/PH_haut = 4/3;
```

- Q 20. On a besoin d'une jointure entre la table **Photo** et la table **Personne**. D'après la figure 4, il y a égalité entre les attributs **PH\_auteur** de la table **Photo** et **PE\_id** de la table **Personne** et on peut réaliser cette jointure sur cette égalité sans passer par la table **Present**. On obtient donc :

```
SELECT COUNT(*) AS nombrePhotos FROM Photo
        JOIN Personne
        ON Photo.PE_auteur = Personne.PE_id
WHERE PE_prenom = 'Alice' OR PE_prenom = 'Bernard';
```

**Remarque :**

Comme il n'y a pas d'ambiguïté sur la table à laquelle appartient l'attribut **PH\_auteur** ou **PE\_id**, on aurait pu écrire directement **PH\_auteur = PE\_id** après le **ON**. C'est ce qui a d'ailleurs été fait pour la clause **WHERE** où on a écrit directement **PE\_prenom** à la place de **Personne.PE\_prenom**.

- Q 21. La fonction d'agrégation **EXTRACT** n'est pas au programme d'IPT. La présence de cette question dans un sujet de concours est donc très critiquable!

Néanmoins, pour répondre à cette question, il faut commencer par faire une jointure entre les tables **Photo** et **MotCle**, ce qui ne peut se faire que par l'intermédiaire de la table **Decrit** : donc jointure de trois tables. On a donc :

```
SELECT PH_id, EXTRACT(year FROM PH_date) AS annee FROM Photo
        JOIN Decrit
        ON Photo.PH_id = Decrit.PH_id
        JOIN MotCle
        ON Decrit.MC_id = MotCle.MC_id
WHERE MC_texte = 'surf'
HAVING annee < 2006;
```

**Remarque :** la condition de sélection sur **annee** qui est le résultat d'une fonction d'agrégation se fait à l'aide du mot clé **HAVING** et jamais **WHERE**.

- Q 22.** Il faut faire une jointure entre les trois tables **Photo**, **Present** et **Personne** basée sur les égalités **Photo.PH\_id = Present.PH\_id** et **Personne.PE\_id = Present.PE\_id**. Pour chaque photo, cette jointure va indiquer les identifiants de toutes les personnes présentes sur la photo. Il faut ensuite sélectionner les lignes qui correspondent à l'égalité **Photo.PH\_auteur = Present.PE\_id**. Cela donne :

```

SELECT PE_prenom, Photo.PH_id FROM Photo
           JOIN Present
           ON Photo.PH_id = Present.PH_id
           JOIN Personne
           ON Present.PE_id = Personne.PE_id
WHERE PH_auteur = Present.PE_id;

```

- Q 23.** On commence par afficher les photos dans lesquelles sont présents "Alice" et "Bernard" en compagnie éventuellement d'autres personnes. Cela se fait en réalisant la jointure suivante :

```

SELECT Photo.PH_id FROM Photo
           JOIN Present
           ON Photo.PH_id = Present.PH_id
           JOIN Personne
           ON Present.PE_id = Personne.PE_id
WHERE PE_prenom = 'Alice' OR PE_prenom = 'Bernard';

```

On sélectionne ensuite les photos où sont présentes toutes les personnes différentes d'Alice ou de Bernard mais d'où ces deux personnes ne sont pas forcément exclues. Cela se fait avec une requête presque identique à la requête précédente mais où on a modifié la deuxième jointure. On a restreint la table **Personne** aux individus différents de Alice et Bernard.

```

SELECT Photo.PH_id FROM Photo
JOIN Present
ON Photo.PH_id = Present.PH_id
JOIN Personne WHERE PE_prenom != 'Alice' AND PE_prenom != 'Bernard'
ON Present.PE_id = Personne.PE_id;

```

On utilise ensuite une clause **EXCEPT** pour faire la différence de ces deux ensembles de photos, ce qui donne :

```
(SELECT Photo.PH_id FROM Photo
      JOIN Present
      ON Photo.PH_id = Present.PH_id
      JOIN Personne
      ON Present.PE_id = Personne.PE_id
 WHERE PE_prenom = 'Alice' OR PE_prenom = 'Bernard' )
EXCEPT
(SELECT Photo.PH_id FROM Photo
      JOIN Present
      ON Photo.PH_id = Present.PH_id
      JOIN Personne WHERE PE_prenom != 'Alice' AND PE_prenom != 'Bernard'
      ON Present.PE_id = Personne.PE_id );
```