

Nouvelle Calédonie – 2023 – sujet2 - Correction

Exercice 1 (6 points)

1

a. Rôle de `idbenevole`

`idbenevole` est la **clé primaire** de la table `Benevole`.

Elle permet :

- d'identifier **de manière unique** chaque bénévole ;
- d'éviter les doublons ;
- d'être référencée par d'autres tables.

👉 *Commentaire : une clé primaire doit être unique et non nulle. Elle garantit l'intégrité d'une table.*

b. Rôle de `idmission`

`idmission` est une **clé étrangère** dans la table `Benevole`.

Elle fait référence à la clé primaire `idmission` de la table `Mission`.

Elle permet :

- d'associer chaque bénévole à une mission ;
- d'assurer la cohérence entre les tables.

👉 *Commentaire : une clé étrangère établit un lien entre deux tables et garantit l'intégrité référentielle.*

2. Objectif de la requête INSERT

La requête :

```
INSERT INTO Benevole
(nom, prenom, rue, code_postal, ville, telephone, idmission)
VALUES
('POHU', 'Dominique', '12 rue de la Frairie', '49000', 'Angers',
'+336123456', 1),
('MOISY', 'Camille', '33 rue Maurice Bot', '49400', 'Cholet',
'+337987654', 2);
```

permet :

- d'insérer **deux nouveaux bénévoles** dans la table `Benevole` ;
- d'enregistrer leurs coordonnées ;
- d'indiquer la mission associée via `idmission`.

👉 *Commentaire : la commande `INSERT` permet d'ajouter des enregistrements dans une table.*

3.

a. Requête : ventes réalisées en 2020

```
SELECT *
FROM Collecte
WHERE annee_ventes = 2020;
```

👉 *Commentaire : la clause `WHERE` permet de filtrer les enregistrements selon une condition.*

b. Noms et prénoms des bénévoles ayant apporté plus de 50 € en 2020

Il faut faire une jointure entre `Benevole` et `Collecte`.

```
SELECT nom, prenom
FROM Benevole
JOIN Collecte ON Benevole.idbenevole = Collecte.idbenevole
WHERE annee_ventes = 2020
AND montant > 50;
```

👉 *Commentaire : une jointure est nécessaire car les informations sont réparties dans deux tables.*

c. Correction du montant erroné

```
UPDATE Collecte
SET montant = 300
WHERE idbenevole = 1
AND annee_ventes = 2020;
```

👉 *Commentaire : la commande UPDATE permet de modifier des données existantes. La clause WHERE est indispensable pour ne pas modifier toute la table.*

4.

a. Pourquoi la requête INSERT dans Presence échoue ?

Requête :

```
INSERT INTO Presence(idpresence, jour , nb_heures)
VALUES ('11-02-2020', 2);
```

La table `Presence` contient **trois attributs** :

- `idpresence`
- `jour`
- `nb_heures`

Or la requête ne fournit que **deux valeurs**, il manque donc une valeur.

👉 *Commentaire : le nombre de valeurs doit correspondre exactement au nombre d'attributs spécifiés.*

b. Pourquoi la requête SELECT échoue ?

Requête :

```
SELECT nom FROM Presence WHERE nb_heures > 2;
```

Problème : La table `Presence` ne contient pas l'attribut `nom`.

Elle contient seulement :

- idpresence
- jour
- nb_heures

👉 *Commentaire : on ne peut sélectionner que des attributs existant dans la table interrogée.*

Correction possible:

Il faut faire une jointure avec la table `Benevole`, par exemple :

```
SELECT nom
FROM Benevole
JOIN Presence ON Benevole.idbenevole = Presence.idpresence
WHERE nb_heures > 2;
```

👉 *Commentaire : lorsqu'une information est dans une autre table, une jointure est nécessaire pour l'obtenir.*
