

Nouvelle Calédonie – 2025 – sujet2 - Correction

Exercice 2 (6 points)

1. Rôle d'une clé primaire

Une **clé primaire** permet d'identifier **de manière unique** chaque enregistrement d'une table.

Contrainte :

- Elle doit être **unique** (aucune valeur en double).
 - Elle ne peut pas être **NULL**.
-

2. Achat d'Émile Pasteur

On repère :

Dans table **client** :

Émile Pasteur → `ref_client = 25145`

Dans table **vente** :

25634 | 2025/03/18 | 89588 | 25145 | 5 | 275

Dans table **produit** :


89588 → Sac sable

Dans table **remise** :

275 → 30%

Réponse complete:

Émile Pasteur a effectué un achat :

-  **Date** : 2025/03/18

- 🛒 **Article** : 5 sacs de sable
- 💰 **Remise** : 30 %

👉 **Commentaire:**

L'élève doit savoir faire le lien entre :

- clé étrangère
- jointure implicite entre tables

3. Ajout d'un nouveau client

Requête SQL:

```
INSERT INTO client
VALUES (25345, 'Bertaut', 'Gilles', 'gbertaut@fmail.fr', '0641424344');
```

💬 **Remarque**

On peut aussi écrire avec les noms d'attributs :

```
INSERT INTO client (ref_client, nom, prenom, email, telephone)
VALUES (25345, 'Bertaut', 'Gilles', 'gbertaut@fmail.fr', '0641424344');
```

4. Correction d'email

Requête SQL:

```
UPDATE client
SET email = 'shars@fmail.fr'
WHERE nom = 'Shar' AND prenom = 'Sofia';
```

5. Clients ayant acheté depuis le 01/01/2025

On veut :

- ref_client
- sans doublon
- date ≥ '2025/01/01'

Requête SQL:

```
SELECT DISTINCT ref_client
FROM vente
WHERE date >= '2025/01/01';
```

👉 *Commentaire:* DISTINCT évite les doublons.

6. Rappel des tondeuses 90222

On veut :

- nom
- téléphone
- clients ayant acheté produit 90222
- depuis 2024/09/15

Requête SQL:

```
SELECT client.nom, client.telephone
FROM client
JOIN vente ON client.ref_client = vente.ref_client
WHERE vente.ref_produit = 90222
AND vente.date >= '2024/09/15';
```

👉 *Commentaire:* Utilisation nécessaire de JOIN ... ON.

7. Compléter select_tel

Dictionnaire :

```
Client = {
  25123: ['Renaud', 'Martine', 'renaudm@tmail.com', '0601020304'],
  ...
}
```

Le téléphone est à l'indice **3**.

Ligne complétée: return client[ref_client][3]

8. Origine de l'erreur KeyError

L'erreur apparaît car la clé 1234 n'existe pas dans le dictionnaire.

Cela correspond à un client inexistant dans la base.

9. Modifier la fonction pour éviter KeyError

Version corrigée:

```
def select_tel(client, ref_client):
    if ref_client in client:
        return client[ref_client][3]
    else:
        return None
```

10. Fonction nb_produits

vente est un dictionnaire dont la clé est un entier et la valeur une liste:

```
vente = {
    25631 : ['2025/03/16', 86782, 25123, 2, 289],
    ...
}
```

Indice :

- ref_produit → index 1
- quantité → index 3

Fonction:

```
def nb_produits(vente, ref_produit):
    total = 0
    for v in vente:
        if vente[v][1] == ref_produit:
            total += vente[v][3]
    return total
```

11. Pourquoi l'erreur foreign key constraint failed ?

Le produit 89363 apparaît dans la table **vente**.

Si on le supprime, certaines ventes référenceraient un produit inexistant. Cela violerait l'intégrité référentielle, donc la suppression est refusée.

12. Réécrire delete_prod

Il faut vérifier qu'aucune vente n'utilise ce produit.

Version corrigée:

```
def delete_prod(produit, vente, ref_produit):
    # Vérifier si produit présent dans une vente
    for v in vente:
        if vente[v][1] == ref_produit:
            print("Suppression impossible : produit utilisé dans une
vente.")
            return

    # Sinon suppression
    if ref_produit in produit:
        del produit[ref_produit]
```

👉 *Commentaire:* On teste l'intégrité référentielle avant suppression.