

Devoir surveillé n°3 – NSI
Correction

Exercice 1

1. Un code postal est un nombre entier, donc l'attribut CP peut être de type **INT**.
2. Le résultat de la requête est le suivant :

5

En effet, il y a 5 lignes (dont les numéros sont tous distincts).

3. L'attribut **Telephone** pourrait servir de clé primaire pour la table **Agences** à la condition qu'aucune agence n'ait le même numéro qu'une autre.
(Principe d'unicité de la clé primaire)
4. Le schéma relationnel est le suivant :
`couple_voitures_agences(#id_voiture,#id_agence)`



Sous la forme suivante :

`couple_voitures_agences(#id_agence,#id_voiture)`

le problème est que cela permettrait à une voiture d'être dans plusieurs agences.

5. On accepte la requête est la suivante (respecter l'ordre des attributs) :

```
INSERT INTO couple_voitures_agences VALUES (2, 5);
```

On peut aussi faire quelque chose comme :

```
INSERT INTO couple_voitures_agences
VALUES (2,
        (SELECT id_agence FROM Agences
         WHERE Agence = 'Deep Place'));
```

car un utilisateur humain de la base de donnée ne connaît pas nécessairement l'identifiant de l'agence.

6. On accepte la requête suivante :

```
UPDATE couple_voitures_agences
SET id_agence = 2
WHERE id_voiture = 2;
```

Sinon, si là encore on ignore les identifiants des agences :

```
UPDATE couple_voitures_agences
SET id_agence = (SELECT id_agence
                 FROM Agences
                 WHERE Agence = 'Le Carrosse')
WHERE id_voiture = 2;
```

7. La requête est la suivante :

```
SELECT type, marque, Agence FROM Voitures
JOIN couple_voitures_agences
ON Voitures.id_voiture = couple_voitures_agences.id_voiture
JOIN Agences
ON Agences.id_agence = couple_voitures_agences.id_agence;
```

Ou, avec **USING** :

```
SELECT type, marque, Agence FROM Voitures
      JOIN couple_voitures_agences USING(id_voiture)
      JOIN Agences USING(id_agence);
```

Pour préciser la provenance des attributs :

```
SELECT Voitures.type, Voitures.marque, Agences.Agence
      FROM Voitures
      JOIN couple_voitures_agences USING(id_voiture)
      JOIN Agences USING(id_agence);
```

Exercice 2

Partie A

1. Il y a des titres (comme Showbiz) qui apparaissent plus qu'une fois dans la table, donc l'attribut **titre** ne peut pas être une clé primaire pour la table **Chanson** : cela brise la règle d'unicité des clés primaires.
2. La requête retourne la réponse suivante :

```
'Welcome too the Jungle', 'Appetite for Destruction'
```

3. La requête est la suivante :

```
SELECT titre FROM Chanson WHERE album = 'Showbiz' ORDER BY titre;
```

4. La requête est la suivante :

```
INSERT INTO Chanson
      VALUES (10, 'Megalomania', 'Hullabaloo', 'Muse');
```

5. La requête est la suivante :

```
UPDATE Chanson SET titre = 'Welcome to the Jungle'
      WHERE titre = 'Welcome too the Jungle';
```

Partie B

6. Utiliser trois tables au lieu de regrouper toutes les informations dans une seule table permet de gagner en place mémoire, car il n'est pas nécessaire de répéter toutes les informations liées à un album pour chaque chanson, par exemple.
7. l'attribut **id_album** de la table **Chanson** a l'intérêt de pouvoir être une clé primaire : elle identifie de manière unique une chanson, permettant qu'un même titre soit présent plusieurs fois (y compris, potentiellement, dans un même album).
8. Le schéma relationnel est le suivant :

Groupe(id, nom)

Album(id, titre, année, #id_groupe)

Chanson(id, titre, #id_album)

9. La requête est la suivante :

```
SELECT titre FROM Album
      WHERE id in (SELECT id_album
                  FROM Chanson
                  WHERE titre = 'Showbiz');
```

ou :

```
SELECT Album.titre /* précision de la table nécessaire */  
FROM Album  
JOIN Chanson ON Album.id = Chanson.id_album  
WHERE Chanson.titre = 'Showbiz';
```

10. La requête est la suivante :

```
SELECT Chanson.titre, Album.titre  
FROM Chanson  
JOIN Album ON Chanson.id_album = Album.id  
JOIN Groupe ON Album.id_groupe = Groupe.id  
WHERE nom = 'Muse';
```

La jointure est ici nécessaire car on fait apparaître les attributs de deux tables.

11. La requête retourne le nombre d'albums du groupe Muse.