

EXERCICE 1 (6 points)

Cet exercice porte sur la programmation Python, les bases de données relationnelles et les requêtes SQL.

En particulier, les mots-clés suivants peuvent être utilisés :

SELECT, CREATE TABLE, FROM, WHERE, JOIN ON, INSERT INTO, VALUES, UPDATE, SET, COUNT, DELETE, DISTINCT, AND, OR, AS.

Partie A

Dans cette partie, on utilise une base de données relationnelle.

Une entreprise de location de voitures propose à ses clients la possibilité de rapporter le véhicule dans une autre agence. Les informations correspondantes sont rangées dans une base de données. Voici les extraits de deux tables utilisées.

Agences					
Agence	Ville	CP	Departement	Adresse	Telephone
Licorne	Paris	75001	Paris	123 Rue de la Révolution	0123456789
Le Carosse	Paris	75001	Paris	456 Virtual Street	0156789012
Vroum	Lyon	69001	Rhône	789 Virtual Lane	0456789034
Rapide	Toulouse	31000	Haute Garonne	321 Virtual Avenue	0567890123
Deep Place	Bordeaux	33000	Gironde	987 Virtual Road	0567890145

Voitures						
id_ voiture	marque	modele	kilome trage	nombre_ places	type	carburant
1	Renault	Clio	64022	5	Berline	Essence
2	Renault	Clio	50350	5	Berline	Essence
3	Dacia	Sandero	62031	5	Berline	Essence
4	Dacia	Sandero	58955	5	Berline	Essence
5	Dacia	Sandero	65779	5	Berline	Essence
6	Dacia	Sandero	56253	5	Berline	Essence
7	Renault	Clio	49660	5	Berline	Essence
8	Fiat	500	2545	4	Citadine	Electrique
9	Fiat	500	1953	4	Citadine	Electrique
10	Fiat	500	549	4	Citadine	Electrique

1. Donner pour la table `Agences` un type possible pour l'attribut `CP` qui indique le code postal.

La fonction `COUNT` permet de compter le nombre d'enregistrements et le mot-clé `DISTINCT` permet de ne pas prendre en compte les doublons.

2. Donner le résultat de la requête suivante pour les extraits des tables données :

```
SELECT COUNT(DISTINCT 'Telephone')
FROM 'Agences' ;
```
3. Expliquer à quelle condition l'attribut `Telephone` pourrait servir de clé primaire pour la table `Agences`.

Certaines agences ont décidé de partager un même standard téléphonique. On ajoute à la table `Agences` l'attribut `id_agence` qui est utilisé comme clé primaire. Voici la nouvelle table obtenue :

Agences						
id_agence	Agence	Ville	CP	Departement	Adresse	Telephone
1	Licorne	Paris	75001	Paris	123 Rue de la Révolution	0123456789
2	Le Carosse	Paris	75001	Paris	456 Virtual Street	0123456789
3	Vroum	Lyon	69001	Rhône	789 Virtual Lane	0456789034
4	Rapide	Toulouse	31000	Haute Garonne	321 Virtual Avenue	0567890123
5	Deep Place	Bordeaux	33000	Gironde	987 Virtual Road	0567890145

Nous souhaitons créer une table `couple_voitures_agences` qui permettra d'associer la table `Agences` à la table `Voitures`. On précise que la table `Voitures` a pour clé primaire `id_voiture`.

4. Décrire les attributs de cette nouvelle table ; une clé primaire sera soulignée et une clé étrangère commencera par un dièse (#).
5. Écrire une requête qui permet d'enregistrer dans la table `couple_voitures_agences` le fait que le véhicule 2 est associé à l'agence `Deep Place`.
6. Écrire une requête qui permet d'actualiser la table pour indiquer que le véhicule 2 se trouve maintenant à l'agence `Le Carrosse`.
7. Écrire une requête qui permet d'afficher le type, la marque et le nom de l'agence pour l'ensemble des véhicules.

Partie B

Pour cette seconde partie, on admet que pour chacune des trois tables, la clé primaire est auto-incrémentée. On utilise par la suite la programmation en Python.

On dispose de la fonction `execute_requete_insert` :

```
1  def execute_requete_insert(requete):
2      """
3      Exécute la requête d'insertion passée en paramètre.
4      Paramètre :
5      - requete : STR, la chaîne formatée de la requête
6      Résultat :
7      - BOOL, booléen de contrôle : Vrai si la requête s'est bien
passée, Faux sinon
8      """
9      ...
```

Exemple d'utilisation :

```
>>> execute_requete_insert('INSERT INTO couple_voitures_agences (id,
id_agence, id_voiture) VALUES (9, 3, 3)')
>>> True
```

On souhaite disposer d'une fonction Python qui insère un nouveau véhicule dans une agence. Cette fonction prend en paramètres les valeurs des attributs de la voiture à insérer et l'entier `id_agence`. Elle renvoie un booléen de contrôle.

```
def insert_voiture(liste_valeurs, id_agence):
    ...
```

Exemple d'utilisation pour une Fiat dans l'agence Vroum :

```
>>> insert_voiture(('Fiat', '500', 18, 4, 'Citadine',
'Electrique'),3)
>>> True
```

8. Écrire en Python la fonction `insert_voiture`.
9. Préciser les conditions que doivent vérifier les paramètres transmis à cette fonction.