

Exercices - Bases de données

1- EXERCICE 1 :

Deux relations modélisent la flotte d'un réseau de location de voitures, la relation *agence* et la relation *voiture*. On donne ci-dessous leur contenu :

id_agence	ville	département
1	Paris	75
2	Lyon	69
3	Marseille	13
4	Aubagne	13

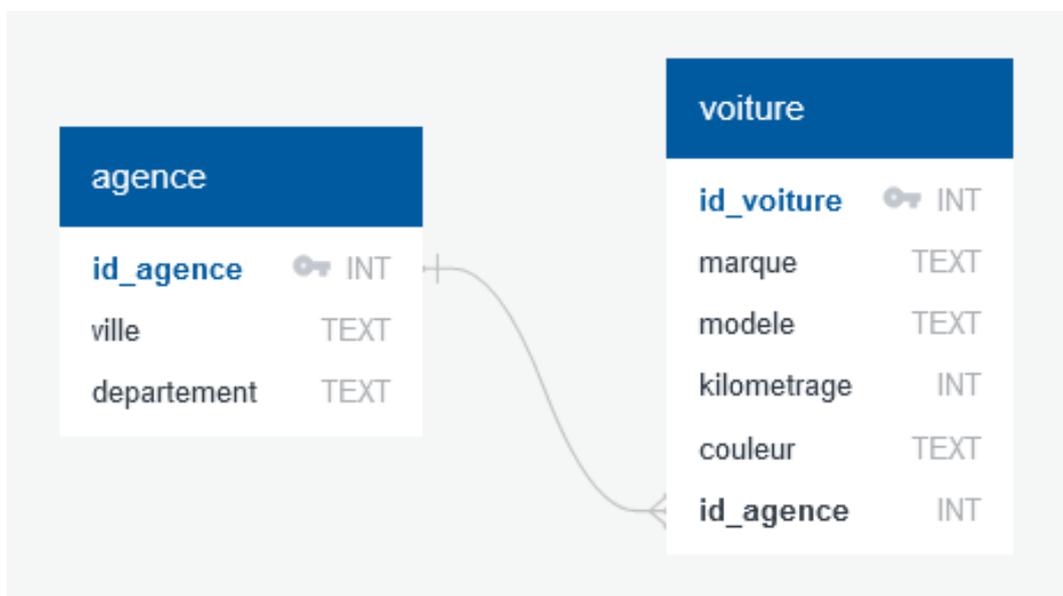
id_voiture	marque	modèle	kilométrage	couleur	id_agence
1	Renault	Clio	12000	Rouge	2
2	Peugeot	205	22000	Noir	3
3	Toyota	Yaris	33000	Noir	3

Le schéma relationnel de la table *agence* est : `agence (id_agence INTEGER , ville TEXT , departement INTEGER)`

- 1- Compléter ci-dessous le schéma relationnel de la table *voiture* (ne pas repérer le type des attributs). Repérer en rouge la clé primaire et la clé étrangère.

`voiture (id_voiture INTEGER , marque TEXT , modele TEXT , kilometrage TEXT , couleur TEXT , #id_agence INTEGER)`

- 2- Compléter ci-dessous le diagramme relationnel de la bdd :



- 3- Donner ci-dessous, les requêtes Sql qui permettent de créer chacune des relations :

Relation <i>agence</i>	Relation <i>voiture</i>
<pre>CREATE TABLE agence (id_agence INTEGER AUTOINCREMENT, ville TEXT, departement TEXT, PRIMARY KEY (id_agence));</pre>	<pre>CREATE TABLE voiture (id_voiture INTEGER PRIMARY KEY , marque TEXT, modele TEXT, kilometrage INTEGER, couleur TEXT, id_agence INTEGER, FOREIGN KEY(id_agence) REFERENCES agence(id_agence));</pre>

4- Donner ci-dessous, les requêtes Sql qui permettent de remplir chacune des relations :

Relation <i>agence</i>	Relation <i>voiture</i>
<pre>INSERT INTO agence(ville,departement) VALUES ('Paris',75), ('Lyon',69), ('Marseille',13) ('Aubagne',13);</pre>	<pre>INSERT INTO voiture VALUES (1,'Renault','Clio',12000,'Rouge',2), (2,'Peugeot','205',22000,'Noir',3), (3,'Toyota','Yaris',33000,'Noir',3);</pre>

5- Donner la requête Sql qui permet de retourner la ville et le kilométrage de Toyota Yaris :

```
SELECT agence.ville , v.kilometrage FROM agence
JOIN voiture AS v ON agence.id_agence = v.id_agence
WHERE v.marque = 'Toyota' AND v.modele = 'Yaris';
```

6- Donner la requête Sql qui permet de retourner le kilométrage moyen de la flotte :

```
SELECT AVG(kilometrage) FROM voiture
```

2- EXERCICE 2 :

La Bdd *tourDeFrance* contient les relations suivantes :

Table *equipe* :

code_equipe	nom_equipe
ALM	AG2R La Mondiale
AST	Astana Pro Team
TBM	Bahrain - McLaren
BOH	BORA - hansgrohe
CCC	CCC Team
COF	Cofidis, Solutions Crédits
DQT	Deceuninck - Quick Step
EF1	EF Pro Cycling
GFC	Groupama - FDJ
LTS	Lotto Soudal
...	...

Table *coureur* :

dossard	nom_coureur	prenom_coureur	code_equipe
141	LÓPEZ	Miguel Ángel	AST
142	FRAILE	Omar	AST
143	HOULE	Hugo	AST
11	ROGLIČ	Primož	TJV
12	BENNETT	George	TJV
41	ALAPHILIPPE	Julian	DQT
44	CAVAGNA	Rémi	DQT
45	DECLERCQ	Tim	DQT
121	MARTIN	Guillaume	COF
122	CONSONNI	Simone	COF
123	EDET	Nicolas	COF
...

Table *etapes* :

num_etape	ville_depart	ville_arrivee	km
1	Nice	Nice	156
2	Nice	Nice	185
3	Nice	Sisteron	198
4	Sisteron	Orcières-Merlette	160
5	Gap	Privas	198
...

Table *temps* :

dossard	num_etape	temps
41	2	04:55:27
121	4	04:07:47
11	5	04:21:22
122	5	04:21:22

1- Donner le schéma relationnel de chaque relation :

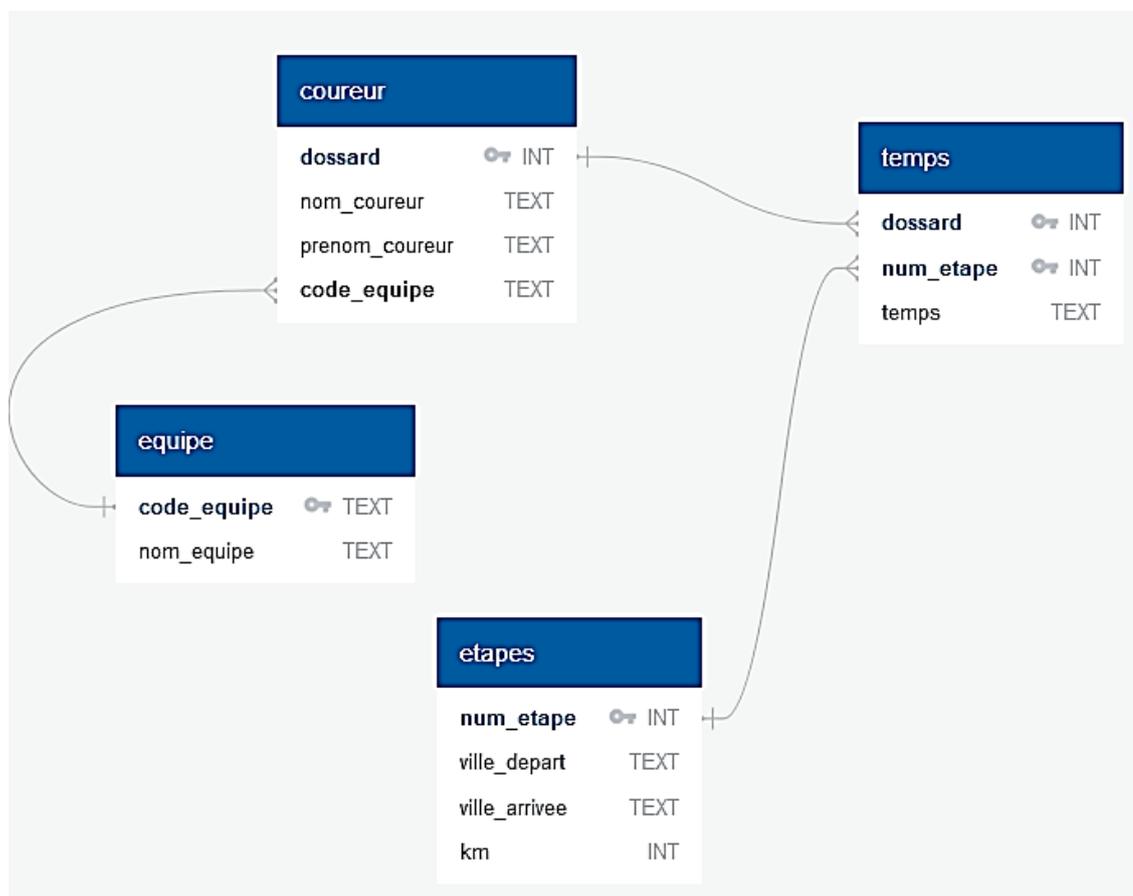
equipe (code_equipe TEXT , nom_equipe TEXT)

coureur (dossard TEXT , nom_coureur TEXT , prenom_coureur TEXT , #code_equipe TEXT)

etapes (num_etape INTEGER , ville_depart TEXT , ville_arrivee TEXT , km INTEGER)

temps (#dossard INTEGER, #num_etape INTEGER, temps TEXT)

2- Donner le diagramme relationnel de la bdd.



- 3- Donner la requête sql qui retourne le temps qu'a réalisé Guillaume MARTIN sur l'étape Sisteron ⇨ Orcières-Merlette.

```
SELECT temps FROM temps AS t
JOIN etapes AS e ON t.num_etape = e.num_etape
JOIN coureur AS c ON t.dossard = c.dossard
WHERE c.prenom_coureur = 'Guillaume' AND c.nom_coureur = 'MARTIN'
AND e.ville_depart = 'Sisteron' AND e.ville_arrivee = 'Orcières-Merlette';
```

3- EXERCICE 3 :

On considère une bdd permettant de gérer des réservations avec une compagnie d'Hotels. Le schéma de cette base est donné ci-dessous :

Client(nom TEXT, prenom TEXT)

Reservation(id_reservation INT, num_chambre INT, nom_hotel TEXT)

Hotel(id_hotel TEXT, nom_hotel TEXT, adresse TEXT)

Chambre(num_chambre INT, nom_hotel TEXT, prix INT)

⇨ Repérer et corriger ci-dessous toutes les anomalies dans le schéma relationnel de cette base.

Client(nom TEXT, prenom TEXT)

Reservation(id_reservation INT, num_chambre INT, nom_hotel TEXT)

Hotel(id_hotel TEXT, nom_hotel TEXT, adresse TEXT)

Chambre(num_chambre INT, nom_hotel TEXT, prix INT)