Exercices : Exercices de base de données

Exercice 1 Une agence de voyage gère ses données grâce à une base de données formée des tables (ou relations) suivantes :

- **Vehicule**(*IdVehicule, Type, Compagnie*) : enregistre les véhicules disponibles - l'identifiant du véhicule, son type et sa compagnie.
- **Trajet**(*IdTrajet, VilleD, VilleA, IdVehicule*) : enregistre les trajets élémentaires possibles - l'identifiant du trajet, la ville de départ, la ville d'arrivée ainsi que le véhicule utilisé.
- **Ticket**(*IdTicket, IdTrajet, Place, Date, Heure, Prix*) : enregistre les tickets disponibles - l'identifiant du ticket, celui du trajet auquel ce ticket donne accès, le numéro de la place, la date, l'horaire et le prix.
- **Hotel**(*IdHotel, Classe, Ville*) : enregistre les hôtels connus - l'identifiant de l'hôtel, sa classe et sa ville.
- **Chambre**(*IdReservation, IdHotel, Date, Prix*) : enregistre les chambres d'hôtels qui sont disponibles - l'identifiant de réservation à utiliser, l'identifiant de l'hôtel où se trouve la chambre, la date et le prix.

1. donner les clés étrangères de cette base de données
2. écrire une requête SQL qui renvoie tous les trajets dont la ville de départ est Brest.
3. écrire une requête qui renvoie le prix moyen de tous les tickets.
4. écrire une requête qui renvoie le nombre de réservations de chambres d'hôtel à Bayonne le 07/07/2020.
5. écrire une requête qui renvoie la liste formée des identifiants d'hôtel et de leur classe dans la ville d'arrivée de chaque trajet.
6. écrire une requête qui donne les différents identifiants de véhicules qui circulent le 07/07/2020.
7. écrire une requête qui renvoie l'identifiant du véhicule le plus utilisé aujourd'hui.
8. Que donne la requête SQL ci-dessous ?

```
SELECT * FROM Chambre WHERE Chambre.Prix = '100'  
AND Chambre.IdHotel IN (SELECT Hotel.IdHotel  
FROM Hotel JOIN Trajet ON Hotel.Ville = Trajet.VilleA JOIN Ticket  
ON Trajet.IdTrajet = Ticket.IdTrajet WHERE Ticket.Prix = '50'  
AND Trajet.VilleD = 'Brest')
```

Exercice 2 Soit la BDD définie par le modèle logique suivant :

- ACTOR (id, name)
- MOVIE (id, title, year, score, votes, director)
où director réfère à ACTOR(id)
- CASTING (movieid, actorid, ord)
où movieid réfère à MOVIE(id) et actorid réfère à ACTOR(id)

1. Trouver les films dont le score est le plus élevé.
2. Sortir la liste des films ayant comptabilisé le moins de votes mais dont le score est supérieur à 8.
3. Trouver les réalisateurs qui comptabilisent les meilleurs scores moyens pour leurs films.
4. Pareil pour les acteurs.
5. Sortir le casting du film "Star Wars"
6. Sortir la liste des films dont le réalisateur est aussi acteur.

Exercice 3 On a une base de donnée avec une seule classe

`enregistrement_etat_civil(prenom, nombre, sexe, annee)`

Cette base de données contient les prénoms des bébés nés à Paris depuis 1955. `nombre` donne le nombre de fois où un prénom est donné par sexe et par année.

1. Pour chacune de requêtes SQL suivantes, donner la traduction "en français".
 - (a) `SELECT DISTINCT prenom FROM enregistrement_etat_civil`
 - (b) `SELECT SUM(nombre) FROM enregistrement_etat_civil`
 - (c) `SELECT COUNT(DISTINCT prenom) FROM enregistrement_etat_civil`
 - (d) `SELECT annee, SUM(nombre) FROM enregistrement_etat_civil GROUP BY annee`
 - (e) `SELECT prenom, annee, SUM(nombre) as s
FROM enregistrement_etat_civil
GROUP BY prenom, annee
HAVING s >= 300`
 - (f) `SELECT prenom, SUM(nombre) as somme
FROM enregistrement_etat_civil
GROUP BY prenom
HAVING somme >= 2000`
2. Donner les commandes SQL qui
 - (a) donne le nombre de naissances de filles qui ont été enregistrées.
 - (b) donne le nombre de fois votre prénom a-t-il été donné à Paris pendant les années concernées.
 - (c) donne, pour chaque année, le nombre de prénoms différents qui ont été enregistrés.
 - (d) donne les prénoms qui ont été donnés exactement 100 fois.
 - (e) donne le prénom qui a été le plus donné sur l'ensemble de la période.
 - (f) donne le prénom féminin qui a été le plus donné sur l'ensemble de la période ?

Exercice 4 La base de données d'une bibliothèque est constituée du modèle logique suivante :

- Auteur(CODE_AUTEUR, NOM_AUTEUR, PRENOM_AUTEUR)
contenant la liste des auteurs.
- Livre(NO_LIVRE, ISBN, TITRE, EDITEUR, DATE_EDITION)
contenant la liste des livres.
- Ecrit(CODE_AUTEUR, NO_LIVRE)
associant à chaque auteur les livres qu'il a écrits.
- Exemplaire(NO_LIVRE, NO_EXEMPL, ETAT, DATE_ACQ)
contenant la liste des exemplaires des livres de la bibliothèque.
- Emprunteur(NO_INSCR, NOM_EMPR, PRENOM_EMPR, ADR_EMPR)
contenant la liste des emprunteurs.
- Emprunt(NO_LIVRE, NO_EXEMPL, NO_INSCR, DATE_EMPR)
contenant la liste des emprunts en cours.

1. En regardant ce modèle, donner l'ensemble des clefs primaires et des clefs étrangères avec leur référence.
2. Donner son modèle conceptuel.
3. Exprimer les requêtes suivantes en SQL.
 - (a) Rechercher le titre des livres écrits par Yves Fonctionrecurs.
 - (b) Rechercher les numéro, nom et prénom des emprunteurs qui ont en prêt un exemplaire du livre numéro 10.
 - (c) Rechercher le nombre d'emprunts en cours.
 - (d) Pour chaque livre, rechercher son numéro, son titre et le nombre de ses exemplaires en cours de prêt.
 - (e) Rechercher le numéro et le titre des livres de la bibliothèque édités depuis le 1 septembre 2014.
 - (f) Rechercher le nom, le prénom et le code des auteurs ayant écrit le plus de livres de la bibliothèque.
 - (g) Rechercher le numéro et le titre des livres disponibles (c-à-d dont au moins un exemplaire n'est pas emprunté).
 - (h) Rechercher le numéro et le titre des livres dont tous les exemplaires sont empruntés.
 - (i) Rechercher le numéro et le titre des livres disponibles (c-à-d dont au moins un exemplaire n'est pas emprunté) édités après le 1 septembre 2010 dont le titre contient le mot "motivation".

Exercice 5 La base de données du registre national est constituée du modèle conceptuel suivant :

- Personne(NUMERO_NATIONAL, NOM, PRENOM, SEXE, RUE, NUMERO, CODE_POSTAL, CODE_LOCALITE)
contenant la liste des habitants de Belgique.
- Commune(CODE_POSTAL, NOM_COMMUNE)
contenant la liste des communes.
- Localité(CODE_POSTAL, CODE_LOCALITE, NOM_LOCALITE)
contenant la liste des localités.
- Mariage(CODE_MARIAGE, NUMERO_NATIONAL_MARI, NUMERO_NATIONAL_FEMME, DATE_MARIAGE, DATE_DIVORCE, CODE_POSTAL)
contenant la liste des mariages.
- Enfant(NUMERO_NATIONAL_ENFANT, NUMERO_NATIONAL_MERE, NUMERO_NATIONAL_PERE, DATE_NAISSANCE)
contenant un historique des naissances.

1. Donner les clefs primaires et étrangères avec leurs références.
2. Donner le modèle conceptuel.
3. Exprimer les requêtes suivantes en SQL.
 - (a) Rechercher le nom et prénom des personnes s'étant mariées à Brest.
 - (b) Rechercher le nom et prénom des personnes n'ayant aucun enfant.
 - (c) Rechercher le nom et prénom des enfants nés de père inconnu.
 - (d) Rechercher le nombre de localités par commune.
 - (e) Rechercher le nom et prénom des enfants étant nés après le mariage (éventuel) de leur parents.
 - (f) Rechercher le pourcentage de divorces par commune depuis la création de la base de données.

Exercice 6 On a une base de données `base_triangles.db`, constituée d'une seule table, dont le modèle logique est :

triangle(idt :integer, ab :integer, ac :integer, bc :integer)

Chaque élément de la table représente les longueurs d'un triangle ABC, ainsi qu'un identificateur unique.

1. Indiquer ce que fait les requêtes SQL suivantes :

```
SELECT COUNT(*) FROM triangles
SELECT * FROM triangles WHERE ab+ac+bc=100
SELECT COUNT(*) FROM triangles WHERE ab=ac
```
2. Déterminer, à l'aide de requêtes SQL :
 - (a) la plus petite valeur des produits $AB \times AC \times BC$, pour les triangles (ABC) de périmètre supérieur ou égal à 100 ;
 - (b) les longueurs correspondants au(x) triangle(s) pour le(s)quel(s) le minimum précédent est atteint ;
 - (c) tous les triangles rectangles en A ;
 - (d) le nombre de tels triangles ;
 - (e) le maximum des périmètres des triangles rectangles en A ;
 - (f) tous les triangles équilatéraux.