

# GIS 2A 3<sup>ème</sup> année

## Devoir surveillé bases de données - documents autorisés

© Polytech Lille - Olivier Caron

Durée 2 heures

### 1 Application de covoiturage (12 points)

Une startup désire mettre en place une application web d'aide au covoiturage. Voici la liste des fonctionnalités de cette application :

**Inscription d'un membre** Un membre peut aussi bien jouer le rôle de passager ou de conducteur de trajets. Toute personne peut s'inscrire à l'application en fournissant les informations suivantes : nom, prénom, date de naissance, le modèle de la voiture qu'il possède éventuellement, son n° de tél et son e-mail (il ne peut y avoir deux membres de même e-mail), un mot de passe. A ces informations de base, quelques informations booléennes supplémentaires permettent de définir le profil du membre : accepte ou pas les fumeurs, accepte ou pas les animaux, accepte la musique, aime peu (bla) ou beaucoup (blabla) discuter.

**Modification d'un membre** Un membre peut consulter ses informations et les modifier. Pour accéder à ses informations, il doit fournir son email et mot de passe.

**Informations sur un membre** Tout membre peut obtenir des informations (à l'exception du mot de passe) sur un autre membre, il suffit d'indiquer l'e-mail du membre recherché.

**Proposition d'un trajet** Tout membre disposant d'une voiture peut proposer des trajets. Les informations à fournir pour un trajet sont : la ville de départ, la ville d'arrivée, un commentaire facultatif pour chaque ville (ex : infos sur lieu de rendez-vous), la date et heure de départ, le nombre de places passagers disponibles, le prix d'une place, une estimation du nombre de kilomètres et la durée estimée du trajet.

**Liste des trajets** L'application doit permettre de fournir une liste de tous les trajets proposés.

**Affectation d'un trajet** Tout membre peut se proposer comme passager d'un trajet. Le passager précise s'il emporte ou pas des bagages. L'acceptation est automatique.

Pour pouvoir réaliser cette application, il faut stocker certaines informations dans une base de donnée.

**Question 1 :** Proposez un schéma conceptuel (4 points) décrit dans le formalisme UML qui permet de gérer l'ensemble de ces informations. Comme l'objectif est de créer une base de données relationnelle, précisez dans votre schéma UML à l'aide du stéréotype "<<Id>>", le ou les attributs UML qui joueront le rôle d'identifiant, ajoutez des attributs si nécessaire.

**Question 2 :** Ecrire un schéma relationnel conforme à ce schéma conceptuel (3 points), précisez les clés primaires et clés étrangères. Inutile de préciser les types des colonnes.

**Question 3 :** Ecrire les requêtes SQL suivantes (5 points) :

- liste des trajets complets (plus de places disponibles)
- liste des membres qui possèdent une voiture et qui aime beaucoup discuter
- pour chaque trajet, afficher la somme récoltée grâce aux passagers inscrits.

### 2 Composition Florale (8 points)

Une grande boutique de fleurs désire informatiser la création de bouquets (vive le progrès ?). Elle dispose d'une base de données qui regroupe les informations suivantes :

- Une fleur est identifiée par un numéro, décrite par un nom, un prix, une couleur et une date limite de vente et appartient à un type de fleur précis.
- Un type de fleur est identifié par un numéro et est caractérisé par un libellé (ex : 'rose', 'tulipe', etc). Un type de fleur peut se marier avec zéro ou plusieurs autres types de fleur.
- Un bouquet est identifié par un numéro et contient un ensemble de fleurs. Le prix du bouquet est la somme du prix des fleurs

Le schéma conceptuel correspondant à ce système d'information est conforme à celui décrit dans la figure 1.

Ci-dessous, le schéma relationnel conforme au schéma conceptuel

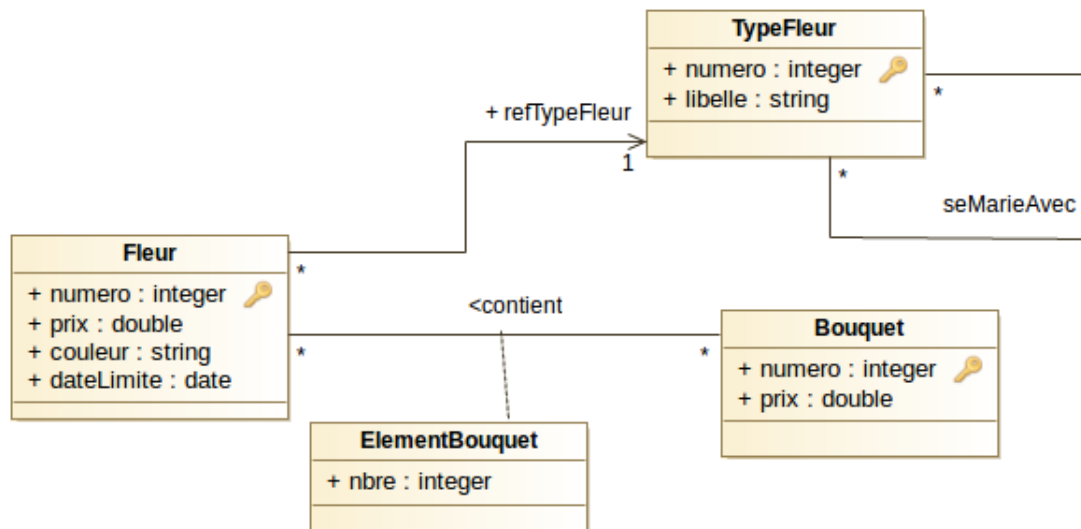


FIGURE 1 – le schéma conceptuel

```

create table typeFleur (
  numero integer primary key,
  libelle text not null unique
) ;

create table Fleur(
  numero integer primary key,
  prix float not null
  couleur text not null default 'rouge',
  dateLimite date not null,
  refTypeFleur integer not null references typeFleur
) ;

create table mariage (
  refType1 integer references typeFleur,
  refType2 integer references typeFleur,
  primary key(refType1, refType2)) ;

create table bouquet (
  numero integer primary key,
  prix float not null default 0.0
) ;

create table elementBouquet (
  refBouquet integer references bouquet,
  refFleur integer references fleur,
  nbre integer not null default 1,
  primary key(refBouquet, refFleur) ;
) ;
  
```

**Question 4 :** Ecrire la fonction PHP `ajoutFleurs($connect,$refFleur,$refBouquet, $nb)` Le but de cette fonction est d'ajouter `$nb` fleurs correspondant au no `$refFleur` dans le bouquet référencé par le no `$refBouquet`. Cette fonction vérifie d'abord si les fleurs qu'on veut ajouter sont d'un type qui peut se marier avec les autres types de fleurs du bouquet. Si c'est le cas, l'ajout est inscrit dans la base et le prix du bouquet est mis à jour dans la base en conséquence. la méthode renvoie `true` si l'ajout est réalisé, `false` sinon.

Dans le cas d'erreurs SQL ou autres, quittez le programme avec un message explicite.

Remarque : La base de données distante est déjà connectée et est accessible via la variable `$connect`