

# TD6: BDD1.

Nous nous intéressons à un service de petites annonces, *le bon endroit*. Un membre de ce service est identifié par un numéro et possède un nom, un prénom et un email. Il peut publier des annonces. Celles ci sont identifiées par un numéro. Elles possèdent un type qui peut être soit 'offre' soit 'demande', un titre, un contenu, une ville et un prix. Elles concernent une rubrique du service (immobilier, voitures, vacances, ...). Celle ci est décrite par un libellé et modérée par un ou plusieurs employés du bon endroit appelés modérateurs. Ceux ci ont à leur charge plusieurs rubriques.

Voici le schéma pour ce problème :

```
MEMBRE(id_membre (entier) ,nom (chaîne), prenom (chaîne), email (chaîne))

MODERATEUR(id_moderateur (entier) ,utilisateur (chaîne))

RUBRIQUE(id_rubrique (entier), libelle (chaîne))

ANNONCE(id_annonce (entier), letype (chaîne), titre (chaîne), contenu (chaîne),
ville (chaîne), prix (réel avec deux chiffres après virgule),
id_rubrique (entier), id_membre (entier))

MODERER(id_moderateur (entier), id_rubrique (entier))
```

## Question 1

Quelle est la clef primaire de la table MEMBRE ? Quelles sont les contraintes à définir sur cette table ? Justifiez vos réponses.

La clé primaire est id-membre  
car il est unique pour chaque membre  
par construction. la contrainte à ajouter  
est : PRIMARY KEY

## Question 2

Quelle peut être la clef primaire de la table MODERER ? Justifiez.

C'est une clé composée, il faut prendre id modérateur et id rubrique

## Question 3

Ecrivez les instructions SQL pour la création des tables RUBRIQUE et ANNONCE. N'oubliez pas les contraintes de clefs primaires, d'intégrité référentielle et autres. Vous utiliserez des clefs autogénérées pour ces deux tables.

```
create table RUBRIQUE (
  id_rubrique serial primary key,
  libelle varchar(30)
);
```

```
create table ANNONCES (
  id-annonces serial primary key,
  letype, titre, ... varchar(30),
  prix numeric(2),
  id_rubrique int,
  id_membre int
FOREIGN KEY(id_rubrique) references
Rubrique(id_rubrique);
--
```

## Question 4

Ecrivez les instructions SQL pour la création de la table MODERER. De même, n'oubliez pas les contraintes.

create table moderer (

id\_moderateur int primary key ,

id\_rubrique int primary key ,

foreign key (id\_m, id\_r) references (rubriques(id), mod.(id)).

⚠️ S: une clé primaire est composée, la clé étrangère associée l'est aussi.  
Pour référencer la FK, on utilise une seule instruction !!

- (a) Liste des annonces pour la ville de *Dunkerque* (titre, contenu, prix).
- (b) Liste des annonces qui ont pour rubrique *immobilier* pour la ville de *Limoges* (titre, contenu, prix).
- (c) Nombre d'annonces de chaque rubrique (libelle, nombre). Une rubrique peut n'avoir aucune annonce.
- (d) Liste des membres n'ayant publié aucune annonce (nom, prenom) ✓
- (e) Pour chaque rubrique, le nombre d'annonces publiées dans la ville de *Nyons* pour les rubriques ayant plus de 1000 annonces dans cette ville (libelle, nombre\_annonces)
- (f) Le chiffre d'affaire espéré pour chaque membre (nom, prenom, chiffre\_affaire). On dispose de la fonction COALESCE(expr, v) qui est équivalent à (si expr est null alors v sinon expr).
- (g) Liste des modérateurs qui modèrent le plus grand nombre de rubriques (utilisateur, nombre\_rubriques)

```
MEMBRE(id_membre (entier), nom (chaîne), prenom (chaîne), email (chaîne))  
  
MODERATEUR(id_moderateur (entier), utilisateur (chaîne))  
  
RUBRIQUE(id_rubrique (entier), libelle (chaîne))  
  
ANNONCE(id_annonce (entier), letype (chaîne), titre (chaîne), contenu (chaîne),  
ville (chaîne), prix (réel avec deux chiffres après virgule),  
id_rubrique (entier), id_membre (entier))  
  
MODERER(id_moderateur (entier), id_rubrique (entier))
```

- a) select titre, contenu, prix from annonce where ville = 'Dunkerque';
- b) select titre, contenu, prix from annonce as a join rubrique as r on  
a.id\_rubrique = r.id\_rubrique where ville = 'Limoges' and libelle = 'immo...';
- c) select libelle, count(id\_annonces) as nombre from rubrique as r  
left join annonce as a on a.id\_rubrique = r.id\_rubrique  
group by (libelle);
- d) select nom, prenom from membre as m right join annonce as a on  
m.id\_membre = a.id\_membre where id\_annonce is null;