

# MLBDA – MU4IN801- 1<sup>ER</sup> EXAMEN REPARTI

## DU 10 NOVEMBRE 2021

Ex 1 :	Ex 2 :	Ex 3 :	
--------	--------	--------	--

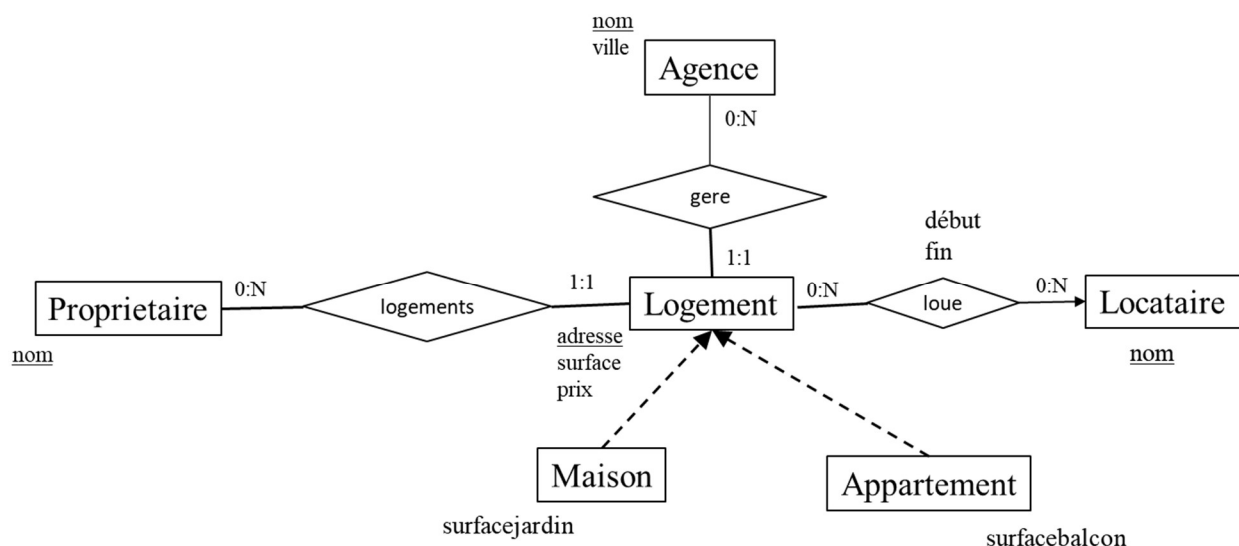
Les documents sont autorisés – Durée totale : 2h.

**Répondre aux questions sur la feuille du sujet** dans les cadres appropriés. Utiliser le dos de la dernière feuille si la réponse déborde du cadre. Le barème est donné à titre indicatif. La qualité de la rédaction sera prise en compte dans la note.

### Exercice 1. Modélisation SQL3

**10 pts**

On considère le schéma Entité-Association suivant, modélisant une base de données de locations.



Les logements ont une adresse (identifiant), une surface et un prix de location (euros/mois) et chaque logement est géré par une seule agence et a un seul propriétaire. On fait la distinction entre les maisons avec une surface de jardin et les appartements avec une surface de balcon. Les agences ont un nom et se situent dans une ville. Elles peuvent gérer plusieurs logements. Les propriétaires ont un nom et peuvent posséder plusieurs logements. Un logement peut être loué par différents locataires à différentes périodes de temps. Un locataire a un nom et peut louer plusieurs logements.

On veut modéliser ce schéma en SQL3. On définit les associations N-N dans le sens des flèches (ex : *Logement – loue* → *Locataire*) et les associations 1-N dans les deux sens. Chaque entité *E* correspond à un type SQL3 *T\_E* et il peut exister d'autres types en plus.

Les types suivants sont déjà définis (le schéma est incomplet):

```
create type T_Locataire as object (  
    nom varchar(16)  
) final instantiable;  
  
create type T_Proprietaire as object (  
    nom varchar(16),  
    logements Ens_Logements  
) final instantiable;
```

```
create type T_Logement as object (  
    adresse varchar(32),  
    surface number(3),  
    prix number(6,2),  
    proprietaire ref T_Proprietaire,  
    agence ref T_Agence,  
    locations Ens_Locations  
) not final not instantiable;
```

Il existe **quatre tables** (racines de persistance) `LesProprietaires`, `LesLocataires`, `LesLogements`, `LesAgences` pour stocker des objets de type `T_Proprietaire`, `T_Locataire`, `T_Logement` et `T_Agence`.

### 1.1 Définition des types et des tables

Définissez les types et les tables suivantes (il est possible que vous deviez définir d'autres types en plus).

**Question 1.** Définissez le type `T_Agence`.

**Question 2.** Définissez le type `T_Appartement`.

**Question 3.** Définissez le type `Ens_Locations` utilisé par le type `T_Logement`.

**Question 4.** Définissez les tables `LesProprietaires`, `LesLocataires`, `LesLogements` et `LesAgences`.

**2.1 Insertion de données :****Question 5.** Ecrivez les instruction SQL3 pour insérer

- l'agence 'Louer Pas Cher' à Paris,
- un appartement au 'App 3 - 1, rue de Paris, Gentilly', avec 75 m2 du surface loué à 500€/mois. La surface du balcon est 20 m2.
- un propriétaire qui s'appelle Max Dupont

**Question 6.** Ecrivez les instructions SQL3 qui déclarent Max Dupont comme propriétaire de l'appartement 'App3 - 1, rue de Paris, Gentilly'.

**2.2 Méthodes SQL3**

**Question 7.** On veut ajouter à ce schéma la méthode **getLocataires(dateloc Date)** qui retourne, pour une agence, l'ensemble des locataires des logements qui sont gérés par l'agence et loués à une date donnée.

Complétez la définition suivante.

```

create or replace type body ..... as
    member function getLocataires(.....) return ..... is

end;
```

**Exercice 2 : SQL3 requêtes** **7 pts**

Soit le schéma :

<pre> create type T_Service ; create type T_Contrat as object(     salaire integer,     statut varchar(16) -- stagiaire, CDD, CDI ); create type T_Personne as object (     nom varchar(32),     prenom varchar(32),     ville varchar(32),     contrat T_Contrat); create type Ens_Personnes as table of ref T_Personne; create type T_Service as object (     nom varchar(32),     responsable ref T_Personne,     membres Ens_Personnes )</pre>	<pre> create type Ens_Services as table of ref T_Service; create type T_Departement as object (     nom varchar(32),     responsable ref T_Personne,     services Ens_Services ) create table <b>LesPersonnes</b> of T_Personne; create table <b>LesServices</b> of T_Service     nested table membres store as T789; create table <b>LesDepartements</b> of T_Departement     nested table services store as T356;</pre>
--	---

Donnez les expressions SQL3 pour les requêtes suivantes.

**Question 1 :** Les noms des membres des services du département 'Informatique'.

**Question 2 :** Le nom du responsable du service 'Logiciels' du département 'Informatique'.

**Question 3 :** Les noms des responsables de département avec un contrat CDI, et des responsables de service avec un contrat CDD. Afficher le nom de la personne, le type de son contrat et son salaire

**Question 4 :** Les noms des employés qui sont responsables d'un département et membre d'un service dans un autre département.

**Question 5 :** Pour chaque service, afficher le nom du service et les noms des employés avec le plus grand salaire.

**Question 6:** Pour chaque département et service, le salaire moyen des membres du service. La requête affiche le nom du département, le nom du service et le salaire moyen.

**Exercice 3. XML et DTD****3 pts**

On considère la DTD suivante pour stocker des graphes dirigés dont les nœuds et les arcs sont étiquetés :

```
<!ELEMENT graphe (noeud*,arc*) >
```

```
<!ELEMENT noeud (label) >
```

```
<!ELEMENT arc (label) >
```

```
<!ELEMENT label (#PCDATA)>
```

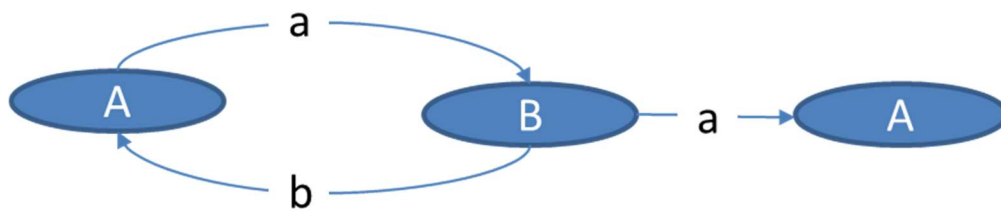
```
<!ATTLIST noeud nid ID #REQUIRED>
```

```
<!ATTLIST arc source IDREF #REQUIRED
```

```
dest IDREF #REQUIRED>
```

Les éléments label correspondent aux étiquettes des nœuds et des arcs.

**Question 1.** Transformer les deux graphes suivant en un seul document XML valide par rapport à la DTD.





**Question 2.** Est-ce qu'il est possible de définir une DTD pour empêcher qu'il y ait des nœuds sans arcs sortants (le graphe précédent n'est pas valide, car le nœud de droite n'a pas d'arc sortant) ? Si oui, donnez la DTD correspondante. Sinon, expliquez pourquoi ?