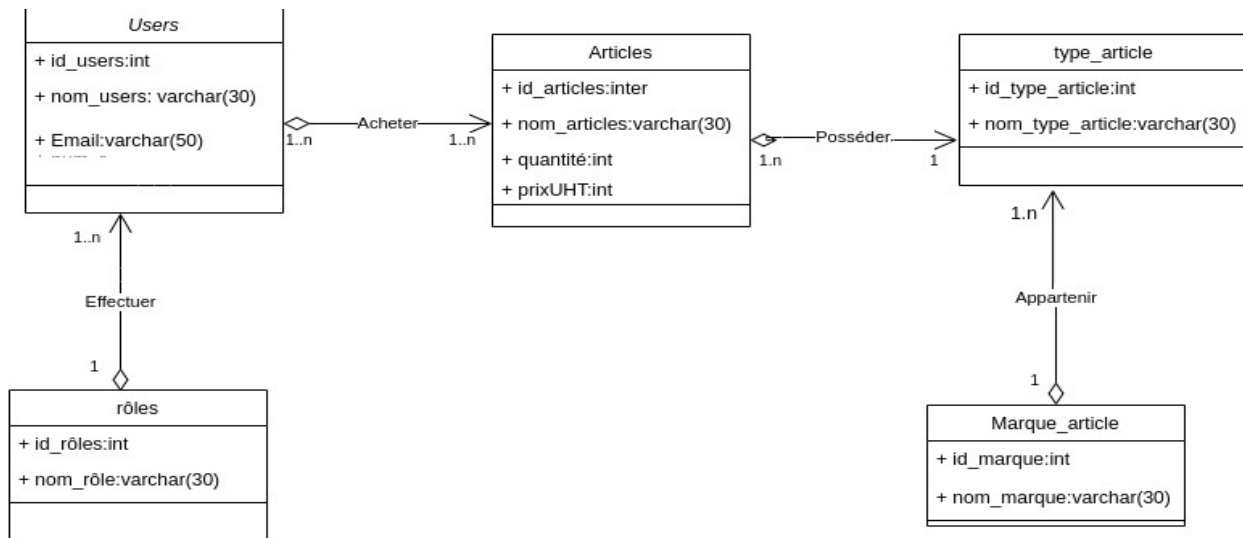
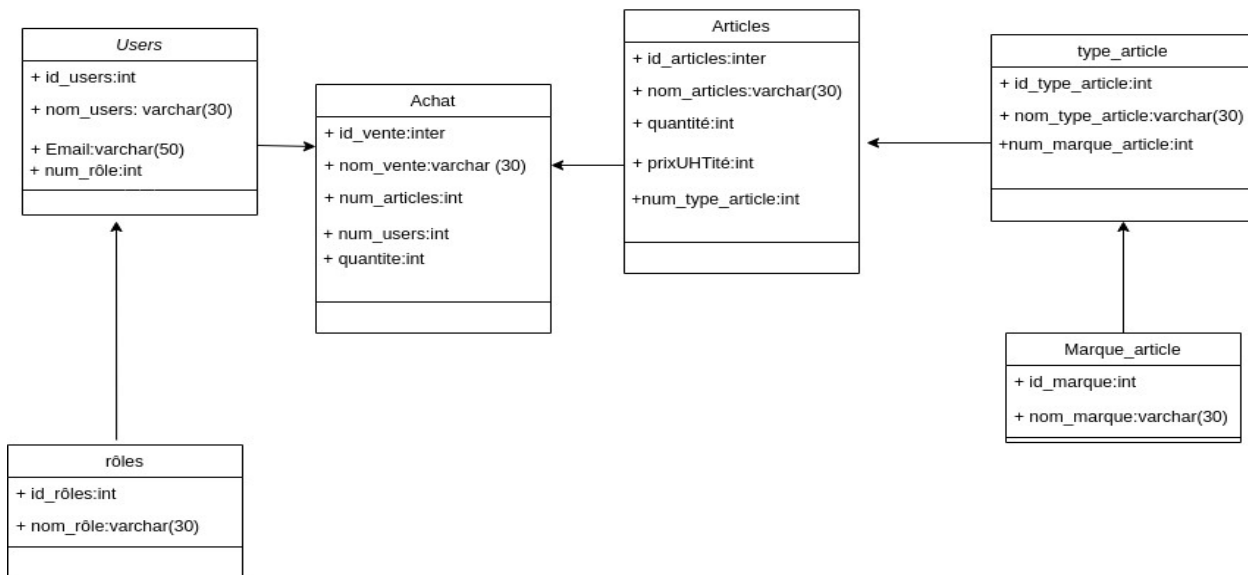


## Travail demandé

1. Créer un diagramme de la base de données avec UML ou une représentation du formalisme Entité-Relation



And



## 2. Créez la base de données avec le langage SQL

```
mysql> create database jojo_fashion;  
Query OK, 1 row affected (0.00 sec)  
  
mysql> use jojo_fashion;  
Database changed
```

## 3. Créez les tables avec des requêtes SQL + Modèle crée avec le diagramme des classes UML

```
mysql> create table rôles(id_rôles int not null auto_increment primary key,nom_rôles varchar(30) not null) ENGINE=InnoDB;  
Query OK, 0 rows affected (0.29 sec)  
  
mysql> create table Users(id_users int not null auto_increment primary key,nom_users varchar(30) not null, Email varchar(50) not null, num_rôles int not null,constraint id_rôles foreign key (num_rôles) references rôles(id_rôles) ON DELETE CASCADE ON UPDATE CASCADE) ENGINE=InnoDB;  
Query OK, 0 rows affected (0.42 sec)  
  
mysql> create table marque_article(id_marque int not null auto_increment primary key,nom_marque varchar(30) not null) ENGINE=InnoDB;  
Query OK, 0 rows affected (0.36 sec)  
  
mysql> create table type_article(id_type_article int not null auto_increment primary key,nom_type_article varchar(30) not null, num_marque_article int not null,constraint id_marque_article foreign key(num_marque_article) references marque_article(id_marque) ON DELETE CASCADE ON UPDATE CASCADE) ENGINE=InnoDB;  
Query OK, 0 rows affected (0.39 sec)  
  
mysql> create table articles(id_articles int not null auto_increment primary key,nom_articles varchar(30) not null, quantite int not null,prix UHT int not null,num_type_article int not null,constraint id_type_article foreign key(num_type_article) references type_article(id_type_article) ON DELETE CASCADE ON UPDATE CASCADE) ENGINE=InnoDB;  
Query OK, 0 rows affected (0.29 sec)  
  
mysql> create table achat(id_vente int not null auto_increment primary key,nom_vente varchar(30) not null, quantite int not null,num_articles int not null,num_users int not null,constraint id_articles foreign key(num_articles) references articles(id_articles) ON DELETE CASCADE ON UPDATE CASCADE) ENGINE=InnoDB;  
Query OK, 0 rows affected (0.51 sec)  
  
mysql> alter table achat add constraint id_users foreign key (num_users) references Users(id_users) ON DELETE CASCADE ON UPDATE CASCADE;  
Query OK, 0 rows affected (0.74 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

j'ai utilisé comme option: **ON DELETE** : pour effectuer une suppression d'un enregistrement qui est référencé dans une autre table;

**ON UPDATE** : Idm que le premier juste différent dans le cas de la mise à jour de l'enregistrement qui est référencé.

Et cela en utilisant comme paramètre **CASCADE** : qui est une mise à jour en cascade. Il mettra à jour / supprimera automatiquement les enregistrements qui référencent l'enregistrement qui a été modifié / supprimé.

#### 4. Insérez les données dans toutes les tables avec SQL tout en tenant compte des relations

```
mysql> insert into rôles(nom_rôles) values ("client"), ("gerant"), ("Agent_commercial");
Query OK, 3 rows affected (0.04 sec)
Records: 3 Duplicates: 0 Warnings: 0

mysql> insert into marque_article(nom_marque) values ("etham"), ("nike"), ("adidas"), ("louis_vuitton");
Query OK, 4 rows affected (0.04 sec)
Records: 4 Duplicates: 0 Warnings: 0

mysql> insert into type_article(nom_type_article, num_marque_article) values ("jupe", "1"), ("robe", "4"), ("blouse", "2"), ("chaussure", "3");
Query OK, 4 rows affected (0.03 sec)
Records: 4 Duplicates: 0 Warnings: 0

mysql> insert into articles(nom_articles, quantite, prixUHT, num_type_article) values ("jupe", "30", "25", "1"), ("robe", "9", "50", "4"), ("blouse", "5", "15", "2"), ("chaussure", "3", "35", "3");
Query OK, 4 rows affected (0.06 sec)
Records: 4 Duplicates: 0 Warnings: 0

mysql> insert into Users(nom_users, Email, num_rôles) value ("thythy", "nyembothesy@gmail.com", "2"), ("dimercia", "dimaria@gmail.com", "3"), ("chrichri", "mutaonachrichri@gmail.com", "1");
Query OK, 3 rows affected (0.03 sec)
Records: 3 Duplicates: 0 Warnings: 0
```

#### 5. Supprimez une catégorie d'articles tout en supprimant aussi tous les articles associés

```
mysql> delete from `type_article` where `type_article`.`id_type_article`=1;
Query OK, 1 row affected (0.06 sec)
```

```
mysql>
```

```
mysql> delete from `type_article` where `id_type_article`=2;
Query OK, 1 row affected (0.04 sec)
```

#### 6. Supprimer des enregistrements avec le langage SQL

```
mysql> delete from `Users` where `id_users`= 2;
Query OK, 0 rows affected (0.02 sec)
```

## 7. Lister tous les articles de la marque nike et de type chaussures basses avec le langage SQ

```
mysql> select * from `articles` join `type_article` join `marque_article` where
`articles`.`num_type_article` = `type_article`.`id_type_article` && `marque_artic
le`.`id_marque` = `type_article`.`num_marque_article` && `marque_article`.`nom_ma
rque` = "nike" && `type_article`.`nom_type_article` = "chaussures_basses";
Empty set (0.00 sec)
```

```
mysql> select * from articles join type_article join marque_article where num_ty
pe_article= id_type_article && id_marque= num_marque_article && nom_marque= "nik
e" && nom_type_article="chaussures_basses";
Empty set (0.00 sec)
```

**Empty set** renvoie un ensemble vide parce qu'il n'existe pas dans notre base de données les chaussures basses de marque nike.

**Exemple:** Juste pour illustrer une liste tous les articles de la marque nike et de type blouse avec le langage SQ pour démontrer que l'article existe.

```
mysql> select * from articles join type_article join marque_article where num_ty
pe_article= id_type_article && id_marque= num_marque_article && nom_marque= "nik
e" && nom_type_article="blouse";
+-----+-----+-----+-----+-----+-----+-----+-----+
| id_articles | nom_articles | quantite | prixUHT | num_type_article | id_type_a |
rticle | nom_type_article | num_marque_article | id_marque | nom_marque |
+-----+-----+-----+-----+-----+-----+-----+-----+
|          4 | chaussure    |          3 |      35 |          2 |          3 |
3 | blouse      |          2 |          2 |          2 | nike      |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

## 8. Lister tous les clients de Jojo Fashion en sachant qu'il n'existe pas de tables clients et qu'il faut faire une requête ou l'on va partir de la catégorie de l'utilisateur

```
mysql> select * from `Users` join `rôles` where `Users`.`num_rôles` = `rôles`.`id_rôles`;
+-----+-----+-----+-----+-----+-----+
| id_users | nom_users | Email | num_rôles | id_rôles | nom_rôles |
+-----+-----+-----+-----+-----+-----+
|          3 | chrichri | mutaonachrichri@gmail.com |          1 |          1 | client |
|          1 | thythy | nyembothesy@gmail.com |          2 |          2 | gerant |
|          2 | dimercia | dimaria@gmail.com |          3 |          3 | Agent_commercial |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.04 sec)
```

```
mysql> select * from `Users` join `rôles` where `num_rôles` = `id_rôles`;
+-----+-----+-----+-----+-----+-----+
| id_users | nom_users | Email | num_rôles | id_rôles | nom_rôles |
+-----+-----+-----+-----+-----+-----+
|          3 | chrichri | mutaonachrichri@gmail.com |          1 |          1 | client |
|          1 | thythy | nyembothesy@gmail.com |          2 |          2 | gerant |
|          2 | dimercia | dimaria@gmail.com |          3 |          3 | Agent_commercial |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

9. Liste toutes les culottes de marque nike se trouvant dans la base de données

```
mysql> select * from type_article join marque_article where id_marque= num_marque_article && nom_marque= "nike" && nom_type_article="culottes";
```

id_type_article	nom_type_article	num_marque_article	id_marque	nom_marque
7	culottes	8	8	nike
8	culottes	8	8	nike
9	culottes	2	2	nike

```
3 rows in set (0.00 sec)
```