

Rapport Entrepôt de Données et Big Data
TP 1 - Bases de données relationnelles et UML

Sujet du TP

Morgan NAVEL, Eric GILLES

19 septembre 2023

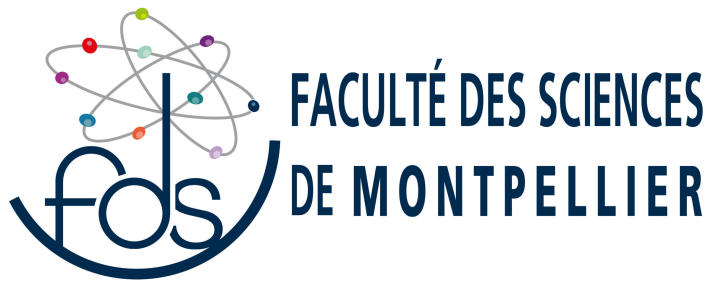


Table des matières

1	Objectifs du TP-TD	3
1.1	Introduction	3
2	Exercice 1	3
2.1	Cahier des charges 1 : Les photos	3
2.2	Cahier des charges 2 : publications, albums et galeries	4
2.3	Cahier des charges 3 : Interactions entre utilisateurs	5
2.4	Choix du modèle & Schéma relationnel	6
3	Exercice 2	6
3.1	Implémentation de la Base de Données sur Oracle	6
3.2	Insertion des données	7
3.3	Requêtes sur la Base de Données	7
4	Annexe	8
4.1	Code de Création des Tables	8
4.2	Code des Insertions des Données	11

1 Objectifs du TP-TD

1.1 Introduction

Le but de ce TD est de se questionner sur la façon dont on modélise et on implémente une base de données en se basant sur le fonctionnement de la plateforme de gestion de photos [Flickr](#).

En effet, il est très important de suivre les spécifications données par le client, pour cela un schéma tel qu'un modèle UML (ou E/A) est très important pour prendre en compte toutes les spécifications données par le client, et permettre de définir les différentes associations entre les entités du discours.

2 Exercice 1

Avant de commencer tous les schémas des modèles ont été réalisés en utilisant drawio, vous retrouverez facilement ces fichiers dans ce [Repository GitHub](#) consacré à cette matière.

2.1 Cahier des charges 1 : Les photos

Dans ce schéma, on retrouve quatre entités et une énumération.

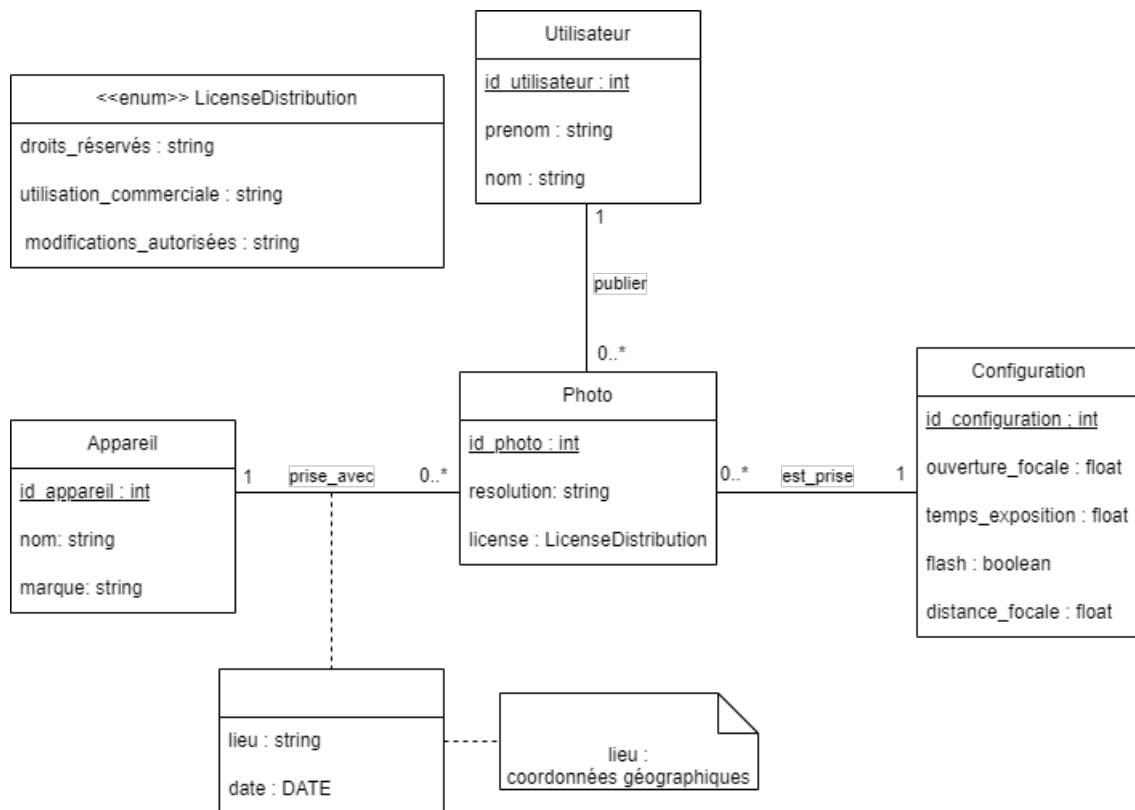


FIGURE 1 – Schéma UML Cahier des Charges 1 : Les photos

Une photo est une entité, qui contient des attributs tels qu'un nom, une résolution et une licence de droit. La licence sera décrite par une énumération. Cette entité est centrée dans notre représentation.

Un appareil photo est caractérisé par un identifiant, un nom et une marque. Un appareil capture plusieurs photos, éventuellement aucune (s'il n'a jamais servi) et une photo ne peut être capturée que par un seul appareil. Cette capture est caractérisée par un lieu (coordonnées géographiques) et une date.

Une configuration est décrite par plusieurs attributs donner dans l'énoncé. Un identifiant de la configuration est ajouté. La photo est prise avec une et une seule configuration, cette configuration peut être utilisée dans plusieurs ou aucune des photos.

Un utilisateur a des attributs nom, prénom (des attributs de base) et un identifiant d'utilisateur en cas d'homonyme, pour pouvoir quand même les identifiés.

Un utilisateur peut publier plusieurs photos, éventuellement aucune, si une photo est publiée, elle a forcément été publiée par un seul utilisateur.

2.2 Cahier des charges 2 : publications, albums et galeries

Ce schéma était un peu plus difficile à représenter en raison des associations impliquant plusieurs entités.

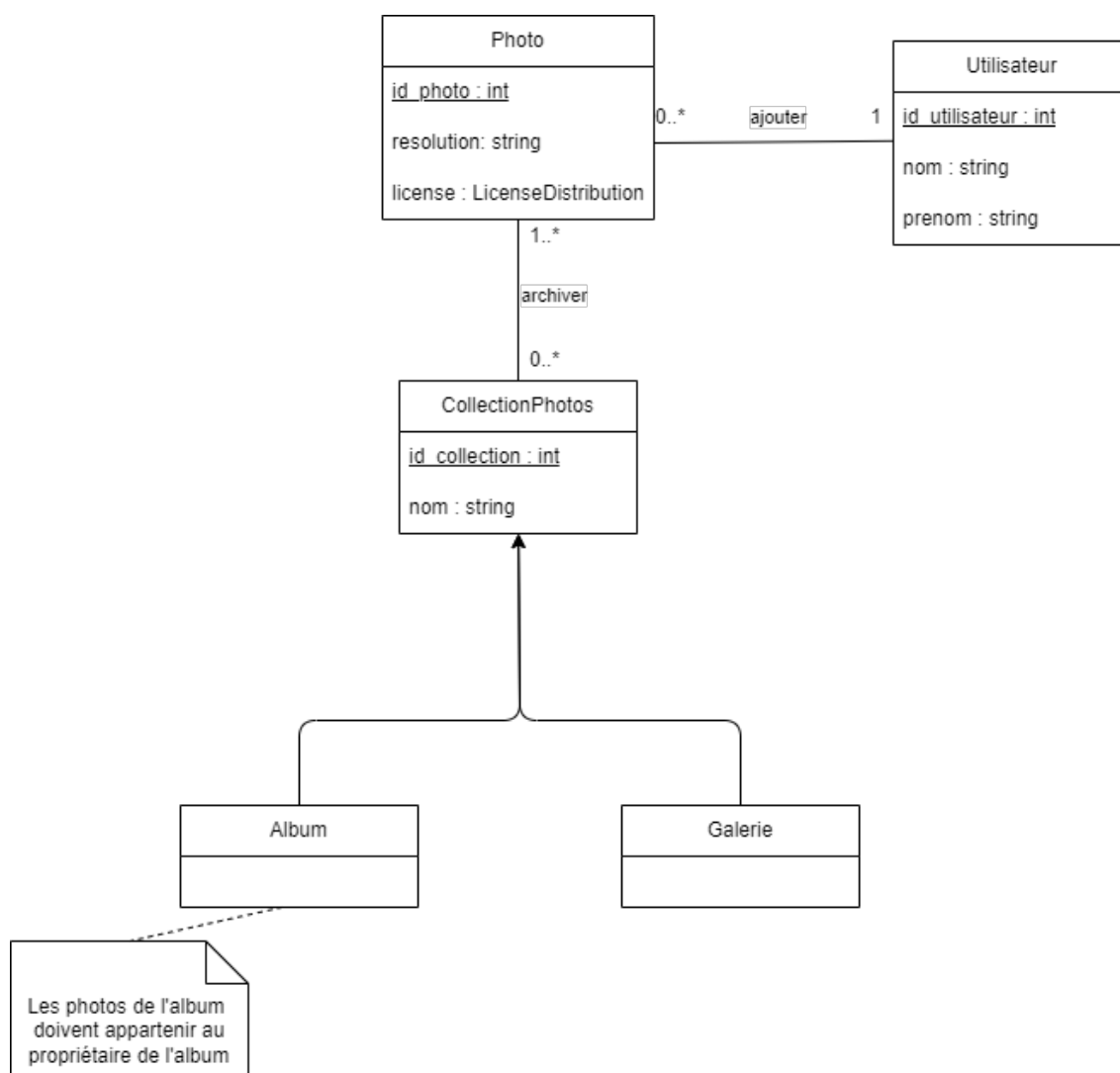


FIGURE 2 – Schéma UML Cahier des Charges 2 : publications, albums et galeries

Nous avons réalisé une modélisation dans laquelle on considérait les photos comme des entités à part entière de la modélisation et l'utilisateur peut ajouter des photos dans une collection.

Album et Galerie sont des collections (Héritage). Grâce à l'énoncé, on peut donc faire les relations. Les collections ont un identifiant et un nom.

Pour photo, deux choix possibles, on peut décider de lier uniquement photo à collection, cela reliera photo à Album et Galerie, soit on les relie une par une, cela dépendra de l'utilisation future de photo dans la base de données. Une collection archive plusieurs photos (éventuellement aucune), et inversement une photo est archivée dans au moins une collection.

Une galerie archive plusieurs photos au moins une et une photo peut être archivée dans une ou plusieurs galeries (éventuellement aucune).

L'appartenance des photos exclusives à un utilisateur pour les albums a été modélisé par une note que le développeur devra prendre en compte pour l'implémentation.

2.3 Cahier des charges 3 : Interactions entre utilisateurs

Dans cette représentation, il y a huit entités.

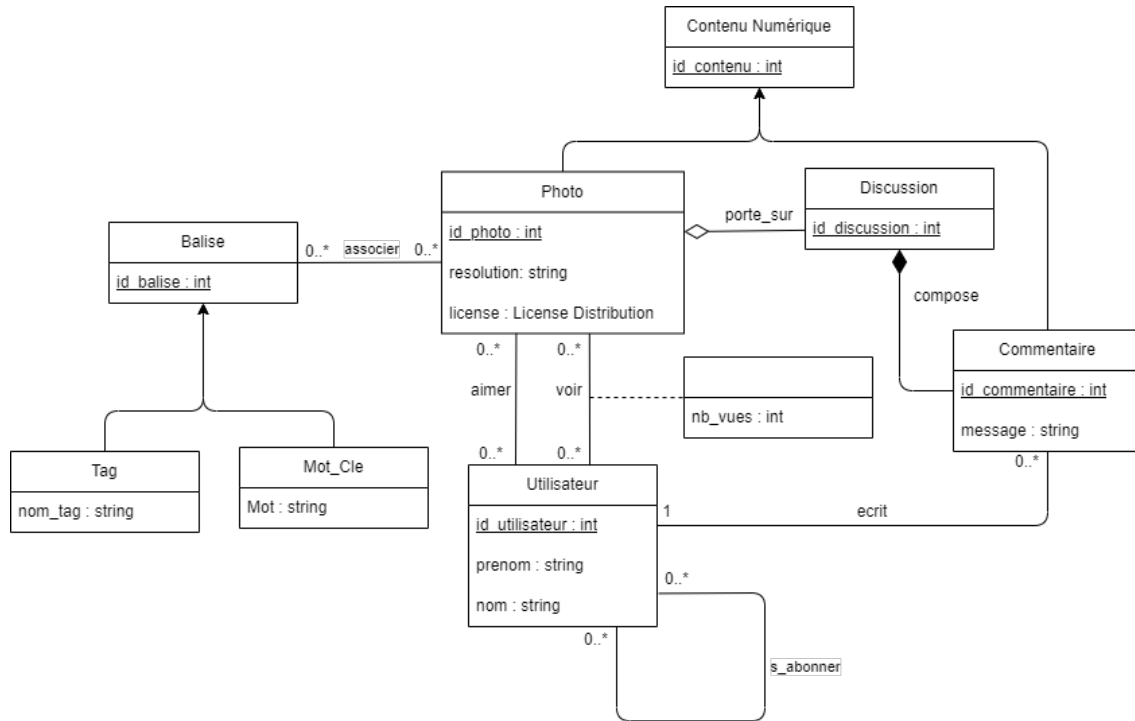


FIGURE 3 – Schéma UML Cahier des Charges 3 : Interactions entre utilisateurs

Commençons tout d'abord par l'entité "Balise", des balises sont associées à des photos. Ces balises sont des tags ou des mots clefs (Héritage), des photos n'ont pas forcément de balises associées. Un tag est une entité avec un nom de tag, et un Mot clef est constitué d'un mot.

Une Photo est un contenu numérique (Héritage) comme les commentaires.

Une Photo est reliée à l'entité Discussion, elle peut contenir au plus une discussion (déduit par le cahier des charges). Une photo est reliée à utilisateur avec deux associations. La première association, lorsqu'un utilisateur aime une photo. Une photo peut avoir 0 "J'aime" et l'utilisateur peut n'avoir aimé aucune photo. La deuxième association, lorsqu'une photo est vue par un utilisateur. Une photo peut ne pas être vue, et un utilisateur peut n'avoir jamais vu de photo.

Un utilisateur peut s'abonner à d'autre utilisateur. Un utilisateur peut avoir aucun abonné ou plusieurs, et peut s'abonner à plusieurs utilisateurs (éventuellement aucun).

Un utilisateur peut aussi écrire des commentaires (éventuellement aucun) contenu dans une discussion, chacun de ces commentaires est écrit par un seul utilisateur. Un commentaire est composé d'un identifiant de commentaire et d'un message.

Une discussion peut contenir plusieurs commentaires au moins un. Une discussion est simplement décrite par un identifiant de discussion.

2.4 Choix du modèle & Schéma relationnel

Rappel : Les termes soulignés correspondent aux clés primaires et les termes précédés d'un #.

UTILISATEUR(id_utilisateur, prenom, nom)

APPAREIL(id_appareil, nom, marque)

CONFIGURATION(id_configuration, ouverture_focale, temps_exposition, flash, distance_focale)

PHOTO(#id_photo, nom, resolution, license, lieu, date_, #id_user, #id_appareil, #id_configuration)

COLLECTION(id_collection, nom)

ALBUM(#id_album)

GALERIE(#id_galerie)

ABONNE(#id_utilisateur1, #id_utilisateur2)

AIMER(#id_photo, #id_utilisateur)

CONTENUNUMERIQUE(id_contenu)

DISCUSSION(id_discussion, #id_photo)

COMMENTAIRE(id_commentaire, #id_discussion, contenu, #id_utilisateur)

BALISE(id_balise, #id_photo)

TAG(#id_tag, nom_tag, #id_photo)

MOT_CLE(#id_tag, mot, #id_photo)

VUES(#id_utilisateur, #id_photo)

3 Exercice 2

3.1 Implémentation de la Base de Données sur Oracle

Pour l'implémentation de la base de données, la contrainte des albums sur les utilisateurs n'a pas été implémentée. Cette contrainte nécessite une réalisation en PL/SQL.

[voir l'annexe](#) pour le code de création des tables en SQL.

3.2 Insertion des données

[voir l'annexe](#) pour le code d'insertion des données dans les tables en SQL.

3.3 Requêtes sur la Base de Données

1. — Les photos de Montpellier

```
SELECT * FROM Photo
WHERE lieu = 'Montpellier';
```

- Les utilisateurs qui sont abonnés à un utilisateur et qui ont pris une photo avec un Canon

```
SELECT Utilisateur.id_utilisateur,nom,prenom FROM Utilisateur
JOIN Abonne ON Utilisateur.id_utilisateur = Abonne.id_utilisateur1
JOIN Photo ON Abonne.id_utilisateur2 = Photo.id_utilisateur
JOIN Appareil ON Photo.id_appareil = Appareil.id_appareil
WHERE Appareil.marque = 'Canon';
```

- Le nombre de Commentaires d'une photo qui a le tag #summer

```
SELECT COUNT(*) FROM Commentaire
JOIN Discussion ON Commentaire.id_discussion = Discussion.id_discussion
JOIN PhotoBalises ON Discussion.id_photo = PhotoBalises.id_photo
JOIN Tag ON PhotoBalises.id_balise = Tag.id_tag
WHERE Tag.tag = '#summer';
```

2. Les photos les plus appréciées avec la licence de distribution 'tous droits réservés'.

```
SELECT Photo.id_photo FROM Photo
JOIN PhotoCollection ON Photo.id_photo = PhotoCollection.id_photo
JOIN Aimer ON Photo.id_photo = Aimer.id_photo
WHERE Photo.licence LIKE 'TousDroitsReserves'
GROUP BY Photo.id_photo,Photo.lieu
HAVING COUNT(*) >= ALL (
    SELECT COUNT(*) FROM Photo P1
    JOIN Aimer ON P1.id_photo = Aimer.id_photo
    WHERE P1.licence LIKE 'TousDroitsReserves'
    GROUP BY P1.id_photo
);
```

3. Les photos incluses dans le plus grand nombre de galeries

```
SELECT Photo.id_photo FROM Photo
JOIN PhotoCollection ON Photo.id_photo = PhotoCollection.id_photo
JOIN Galerie ON PhotoCollection.id_collection = Galerie.id_galerie
GROUP BY Photo.id_photo
HAVING COUNT(*) >= ALL (
    SELECT COUNT(*) FROM Photo P1
    JOIN PhotoCollection ON P1.id_photo = PhotoCollection.id_photo
    JOIN Galerie ON PhotoCollection.id_collection = Galerie.id_galerie
    GROUP BY P1.id_photo
);
```

4 Annexe

4.1 Code de Création des Tables

```
/*Creation de la base de donnees TP1*/

/*creation des tables*/

/*table des appareils*/
CREATE TABLE Appareil(
    id_appareil INTEGER PRIMARY KEY,
    marque VARCHAR(50),
    modele VARCHAR(50)
);

/*table des utilisateurs*/
CREATE TABLE Utilisateur(
    id_utilisateur INTEGER PRIMARY KEY,
    nom VARCHAR(50),
    prenom VARCHAR(50)
);

/*table des abonnements*/
CREATE TABLE Abonne(
    id_utilisateur1 INTEGER NOT NULL,
    id_utilisateur2 INTEGER NOT NULL,
    PRIMARY KEY(id_utilisateur1, id_utilisateur2),
    FOREIGN KEY(id_utilisateur1) REFERENCES Utilisateur(id_utilisateur
),
    FOREIGN KEY(id_utilisateur2) REFERENCES Utilisateur(id_utilisateur
)
);

/*table des configurations*/
CREATE TABLE Configuration(
    id_configuration INTEGER PRIMARY KEY,
    ouverture_focale FLOAT,
    temps_exposition FLOAT,
    flash VARCHAR(1),
    distance_focale float,
    CONSTRAINT Check_flash CHECK(flash IN ('Y','N'))
);

/*tables des Collections*/
CREATE TABLE Collection(
    id_collection INTEGER PRIMARY KEY
);

/*table des Albums*/
CREATE TABLE Album(
    id_album INTEGER PRIMARY KEY,
    id_utilisateur INTEGER,
    FOREIGN KEY(id_album) REFERENCES Collection(id_collection),
    FOREIGN KEY(id_utilisateur) REFERENCES Utilisateur(id_utilisateur)
```



```

);

/*table des Galeries*/
CREATE TABLE Galerie(
    id_galerie INTEGER PRIMARY KEY,
    id_utilisateur INTEGER,
    FOREIGN KEY(id_galerie) REFERENCES Collection(id_collection),
    FOREIGN KEY(id_utilisateur) REFERENCES Utilisateur(id_utilisateur)
);

/*table des contenus numeriques*/
CREATE TABLE ContenuNumerique(
    id_contenu INTEGER PRIMARY KEY
);

/*table des photos*/
CREATE TABLE Photo(
    id_photo INTEGER PRIMARY KEY,
    resolution VARCHAR(50),
    date_photo DATE NOT NULL,
    lieu VARCHAR(50) NOT NULL,
    licence VARCHAR(50),
    id_utilisateur INTEGER,
    id_appareil INTEGER,
    id_configuration INTEGER,
    FOREIGN KEY(id_photo) REFERENCES ContenuNumerique(id_contenu),
    FOREIGN KEY(id_utilisateur) REFERENCES Utilisateur(id_utilisateur)
,
    FOREIGN KEY(id_appareil) REFERENCES Appareil(id_appareil),
    FOREIGN KEY(id_configuration) REFERENCES Configuration(
id_configuration),
    CONSTRAINT Check_licence CHECK ( licence IN ('TousDroitsReserves',
'UtilisationCommercialeAutorisee', 'ModificationImageAutorisee'))
);

/*table des balises*/
CREATE TABLE Balise(
    id_balise INTEGER PRIMARY KEY
);

/*table des tags*/
CREATE TABLE Tag(
    id_tag INTEGER PRIMARY KEY,
    tag VARCHAR(50) NOT NULL,
    id_photo INTEGER,
    FOREIGN KEY(id_tag) REFERENCES Balise(id_balise)
);

/*table des mots cles*/
CREATE TABLE MotCle(
    id_mot_cle INTEGER PRIMARY KEY,
    mot_cle VARCHAR(50) NOT NULL,
    FOREIGN KEY(id_mot_cle) REFERENCES Balise(id_balise)
);

/*table des balises sur une photo*/

```

```

CREATE TABLE PhotoBalises(
    id_photo INTEGER,
    id_balise INTEGER,
    PRIMARY KEY(id_balise,id_photo),
    FOREIGN KEY(id_photo) REFERENCES Photo(id_photo),
    FOREIGN KEY(id_balise) REFERENCES Balise(id_balise)
);

/*table des photos dans les collections*/
CREATE TABLE PhotoCollection(
    id_photo INTEGER,
    id_collection INTEGER,
    PRIMARY KEY(id_photo, id_collection),
    FOREIGN KEY(id_photo) REFERENCES Photo(id_photo),
    FOREIGN KEY(id_collection) REFERENCES Collection(id_collection)
);

/*table des photos aimees par un utilisateur*/
CREATE TABLE Aimer(
    id_photo INTEGER,
    id_utilisateur INTEGER,
    PRIMARY KEY(id_photo, id_utilisateur),
    FOREIGN KEY(id_photo) REFERENCES Photo(id_photo),
    FOREIGN KEY(id_utilisateur) REFERENCES Utilisateur(id_utilisateur)
);

/*table des discussions sur une photo*/
CREATE TABLE Discussion(
    id_discussion INTEGER PRIMARY KEY,
    id_photo INTEGER,
    FOREIGN KEY(id_photo) REFERENCES Photo(id_photo)
);

/*table des commentaires sur une discussion*/
CREATE TABLE Commentaire(
    id_commentaire INTEGER PRIMARY KEY,
    contenu VARCHAR(250) NOT NULL,
    id_discussion INTEGER,
    id_utilisateur INTEGER,
    FOREIGN KEY(id_commentaire) REFERENCES ContenuNumerique(id_contenu
),
    FOREIGN KEY(id_discussion) REFERENCES Discussion(id_discussion),
    FOREIGN KEY(id_utilisateur) REFERENCES Utilisateur(id_utilisateur)
);

CREATE TABLE Vues(
    id_photo INTEGER,
    id_utilisateur INTEGER,
    PRIMARY KEY(id_photo, id_utilisateur),
    FOREIGN KEY(id_photo) REFERENCES Photo(id_photo),
    FOREIGN KEY(id_utilisateur) REFERENCES Utilisateur(id_utilisateur)
);

/*fin de la creation de la base de donnees*/

```

4.2 Code des Insertions des Données

```
/*Insertion des donnees TP1*/
/*insertion des donnees*/
INSERT INTO Appareil(id_appareil,marque, modele) VALUES(1,'Canon', '
EOS 5D Mark IV');
INSERT INTO Appareil(id_appareil,marque, modele) VALUES(2,'Nikon', '
D750');
INSERT INTO Appareil(id_appareil,marque, modele) VALUES(3,'Sony', '
ILCE-7M3');
INSERT INTO Appareil(id_appareil,marque, modele) VALUES(4,'Panasonic',
'Lumix GH5');

INSERT INTO Utilisateur(id_utilisateur, nom, prenom) VALUES(1, 'Gilles
', 'Eric');
INSERT INTO Utilisateur(id_utilisateur, nom, prenom) VALUES(2, 'Navel'
, 'Morgan');
INSERT INTO Utilisateur(id_utilisateur, nom, prenom) VALUES(3, 'Nemard
', 'Jean');
INSERT INTO Utilisateur(id_utilisateur, nom, prenom) VALUES(4, '
Zeblouse', 'Agathe');

INSERT INTO Abonne(id_utilisateur1, id_utilisateur2) VALUES(1, 2);
INSERT INTO Abonne(id_utilisateur1, id_utilisateur2) VALUES(1, 3);
INSERT INTO Abonne(id_utilisateur1, id_utilisateur2) VALUES(1, 4);
INSERT INTO Abonne(id_utilisateur1, id_utilisateur2) VALUES(2, 1);
INSERT INTO Abonne(id_utilisateur1, id_utilisateur2) VALUES(2, 3);
INSERT INTO Abonne(id_utilisateur1, id_utilisateur2) VALUES(4, 3);

INSERT INTO Configuration(id_configuration, ouverture_focale,
temps_exposition, flash, distance_focale) VALUES(1, 1.8, 1/100, 'N'
, 50);
INSERT INTO Configuration(id_configuration, ouverture_focale,
temps_exposition, flash, distance_focale) VALUES(2, 2.8, 1/200, 'Y'
, 100);
INSERT INTO Configuration(id_configuration, ouverture_focale,
temps_exposition, flash, distance_focale) VALUES(3, 3.5, 1/400, 'N'
, 200);

INSERT INTO Collection(id_collection) VALUES(1);
INSERT INTO Collection(id_collection) VALUES(2);
INSERT INTO Collection(id_collection) VALUES(3);
INSERT INTO Collection(id_collection) VALUES(4);

INSERT INTO Album(id_album, id_utilisateur) VALUES(1, 1);
INSERT INTO Album(id_album, id_utilisateur) VALUES(2, 2);

INSERT INTO Galerie(id_galerie, id_utilisateur) VALUES(3, 3);
INSERT INTO Galerie(id_galerie, id_utilisateur) VALUES(4, 4);

INSERT INTO ContenuNumerique(id_contenu) VALUES(1);
INSERT INTO ContenuNumerique(id_contenu) VALUES(2);
INSERT INTO ContenuNumerique(id_contenu) VALUES(3);
```

```

INSERT INTO ContenuNumerique(id_contenu) VALUES(4);

INSERT INTO Photo(id_photo, resolution, date_photo, lieu, licence,
id_utilisateur, id_appareil, id_configuration) VALUES(1, '1280x720',
TO_DATE('2023-06-03','YYYY-MM-DD'),'Montpellier','
TousDroitsReserves', 1, 1, 1);
INSERT INTO Photo(id_photo, resolution, date_photo, lieu, licence,
id_utilisateur, id_appareil, id_configuration) VALUES(2, '1280x720',
TO_DATE('2023-08-03','YYYY-MM-DD'),'Paris','
UtilisationCommercialeAutorisee', 2, 2, 2);
INSERT INTO Photo(id_photo, resolution, date_photo, lieu, licence,
id_utilisateur, id_appareil, id_configuration) VALUES(3, '1280x720',
TO_DATE('2023-12-03','YYYY-MM-DD'),'Los Angeles','
ModificationImageAutorisee', 3, 2, 3);
INSERT INTO Photo(id_photo, resolution, date_photo, lieu, licence,
id_utilisateur, id_appareil, id_configuration) VALUES(4, '1280x720',
TO_DATE('2023-04-03','YYYY-MM-DD'),'Incheon','TousDroitsReserves'
, 3, 3, 3);

INSERT INTO Balise(id_balise) VALUES(1);
INSERT INTO Balise(id_balise) VALUES(2);
INSERT INTO Balise(id_balise) VALUES(3);
INSERT INTO Balise(id_balise) VALUES(4);
INSERT INTO Balise(id_balise) VALUES(5);

INSERT INTO Tag(id_tag,tag) VALUES(1,'#summer');
INSERT INTO Tag(id_tag,tag) VALUES(2,'#view');
INSERT INTO Tag(id_tag,tag) VALUES(3,'#montpellier');

INSERT INTO MotCle(id_mot_cle,mot_cle) VALUES(4,'Korea');
INSERT INTO MotCle(id_mot_cle,mot_cle) VALUES(5,'Baguette');

INSERT INTO PhotoBalises(id_photo,id_balise) VALUES(1,1);
INSERT INTO PhotoBalises(id_photo,id_balise) VALUES(2,5);
INSERT INTO PhotoBalises(id_photo,id_balise) VALUES(3,2);
INSERT INTO PhotoBalises(id_photo,id_balise) VALUES(4,4);
INSERT INTO PhotoBalises(id_photo,id_balise) VALUES(1,3);

INSERT INTO PhotoCollection(id_photo, id_collection) VALUES(1, 1);
INSERT INTO PhotoCollection(id_photo, id_collection) VALUES(2, 2);
INSERT INTO PhotoCollection(id_photo, id_collection) VALUES(3, 3);
INSERT INTO PhotoCollection(id_photo, id_collection) VALUES(3, 4);

INSERT INTO Aimer(id_photo, id_utilisateur) VALUES(1, 1);
INSERT INTO Aimer(id_photo, id_utilisateur) VALUES(1, 2);
INSERT INTO Aimer(id_photo, id_utilisateur) VALUES(1, 3);
INSERT INTO Aimer(id_photo, id_utilisateur) VALUES(1, 4);
INSERT INTO Aimer(id_photo, id_utilisateur) VALUES(2, 2);
INSERT INTO Aimer(id_photo, id_utilisateur) VALUES(3, 3);
INSERT INTO Aimer(id_photo, id_utilisateur) VALUES(4, 1);

```

```

INSERT INTO Discussion(id_discussion, id_photo) VALUES(1, 1);
INSERT INTO Discussion(id_discussion, id_photo) VALUES(3, 1);
INSERT INTO Discussion(id_discussion, id_photo) VALUES(2, 2);

INSERT INTO Commentaire(id_commentaire, id_discussion, id_utilisateur,
    contenu) VALUES(1, 1, 1, 'Super photo !');
INSERT INTO Commentaire(id_commentaire, id_discussion, id_utilisateur,
    contenu) VALUES(2, 2, 2, 'Magnifique !');
INSERT INTO Commentaire(id_commentaire, id_discussion, id_utilisateur,
    contenu) VALUES(3, 1, 3, 'Complettement incoherente, je ne
    comprends pas comment cette photo a ete prise. Travaillez votre
    cadrage ! ');

INSERT INTO Vues(id_photo, id_utilisateur) VALUES(1, 1);
INSERT INTO Vues(id_photo, id_utilisateur) VALUES(1, 2);
INSERT INTO Vues(id_photo, id_utilisateur) VALUES(1, 3);
INSERT INTO Vues(id_photo, id_utilisateur) VALUES(2, 1);
INSERT INTO Vues(id_photo, id_utilisateur) VALUES(2, 4);
INSERT INTO Vues(id_photo, id_utilisateur) VALUES(3, 1);
INSERT INTO Vues(id_photo, id_utilisateur) VALUES(3, 3);
/*fin de l'insertion des donnees*/

```