

Université de Montpellier

Année 2023-2024

Faculté des Sciences

30 Place E.Bataillon, 34095 Montpellier

Rapport de Travaux Pratiques

TP n°1

Bases de données relationnelles et UML

par

Romain GALLERNE
Loris BENAÏTIER

Encadrant de TP : Mme Anne-Muriel Chifolleau

Responsable du module : Mme Anne-Muriel Chifolleau

Table des matières

1	Introduction	2
2	Modélisation conceptuelle : la gestion des photos dans la plateforme flickr	3
2.1	UML n°1	3
2.2	UML n°2	5
2.3	UML n°3	6
3	Implémentation du cahier des charges n°1	7
3.1	Schéma logique	7
3.2	Implémentation Oracle	8
4	Conclusion	9

Introduction

Dans ce premier TP, on se propose d'étudier les bases de la modélisation de bases de données en UML ainsi que son implémentation en utilisant pour cela le modèle de la plateforme flickr (www.flickr.com), un site d'hébergement de photos. Nous allons donc étudier la base de données d'une populaire application Web permettant le partage de photos.

Modélisation conceptuelle : la gestion des photos dans la plateforme flickr

Dans cette première partie, nous allons présenter les différents modèles UML réalisés pour chaque cahier des charges fournis ainsi que détailler les choix de modélisation que nous avons faits.

2.1 UML n°1

D'après ce premier cahier des charges, nous avons choisi de modéliser les photographies, les utilisateurs, les appareils photos et les configurations des photographies comme des classes, ajoutons à cela la classe parent de photographie : contenu numérique. La présence de celle-ci n'est justifiée que par soucis de cohérence de la classe photographie avec le modèle UML3, nous détaillerons donc cette classe plus tard.

Pour modéliser le fait qu'un utilisateur puisse publier une photographie à une date précise, nous enregistrons simplement la date de publication de la photographie par un attribut, nous pourrons ensuite, dans le modèle physique, retrouver aisément la date de publication de l'utilisateur par une simple jointure.

Les configurations de l'appareil photo sont sauvegardés séparément dans une classé à part dont la clé est l'ensemble des paramètres (chaque configuration est unique).

Une photographie comporte une licence, celle-ci est modélisé par une énumération des diverses licences possibles.

Le lieu et la date de la photographie sont enregistrés dans l'association "réalise".

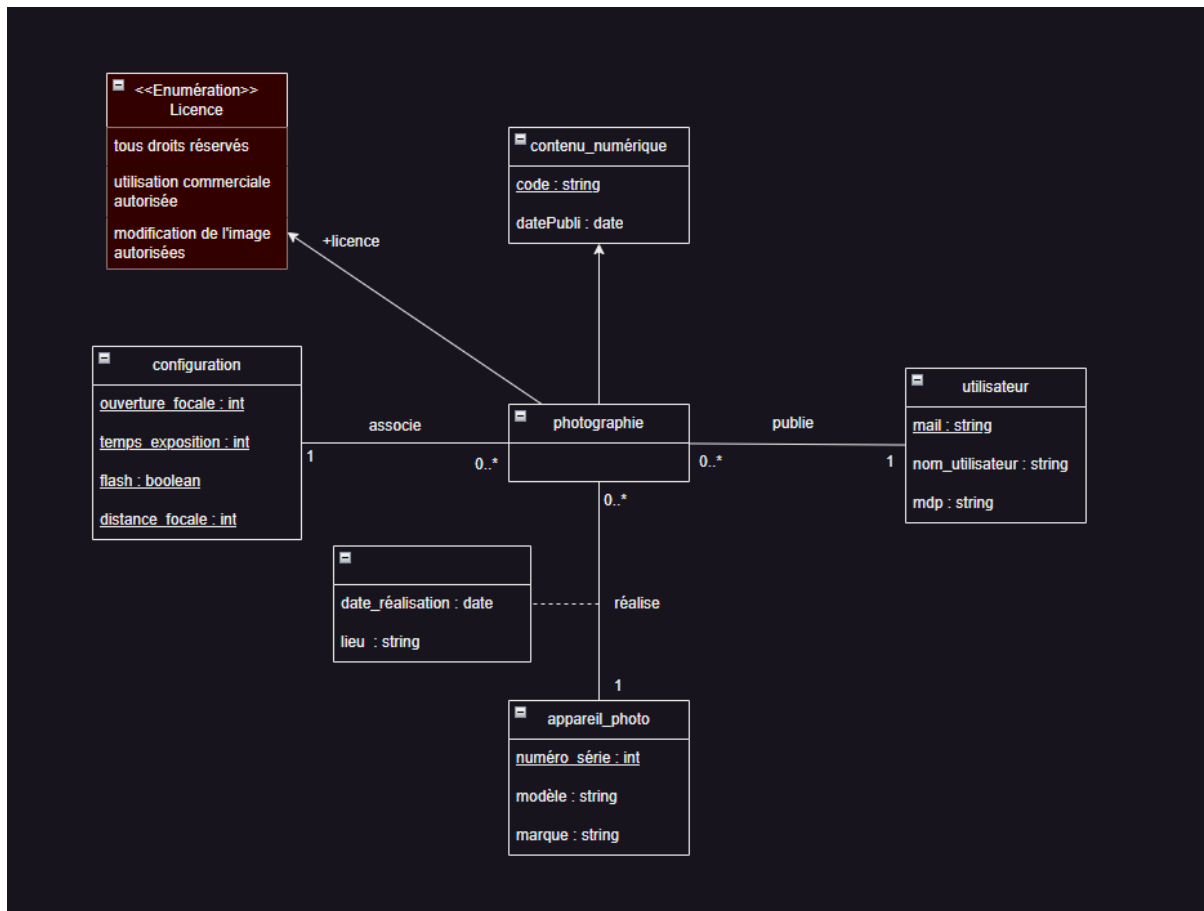


Schéma UML du 1er cahier des charges d'après les choix de modélisations effectués.

2.2 UML n°2

Dans le second cahier des charges, nous avons choisi de modéliser les classes : photographie et sa classe parent contenu_numérique, utilisateurs, galeries et les albums comme enfants des collections_photos.

La possibilité de ranger des photos dans des albums et des galeries a été une étape difficile de la modélisation de ce schéma, en effet nous souhaitions pouvoir représenter le fait qu'une galerie puisse contenir les images d'autres utilisateurs. Nous avons donc abouti à une solution dans laquelle les photographies peuvent appartenir à une ou plusieurs galeries et cela sans contrainte particulière. En revanche, pour les albums, nous avons modélisé cela par une association ternaire entre l'album, l'utilisateur et la photographie dans laquelle il faudra bien vérifier au moment de l'ajout que l'auteur de la photographie est le même que le propriétaire de la galerie.

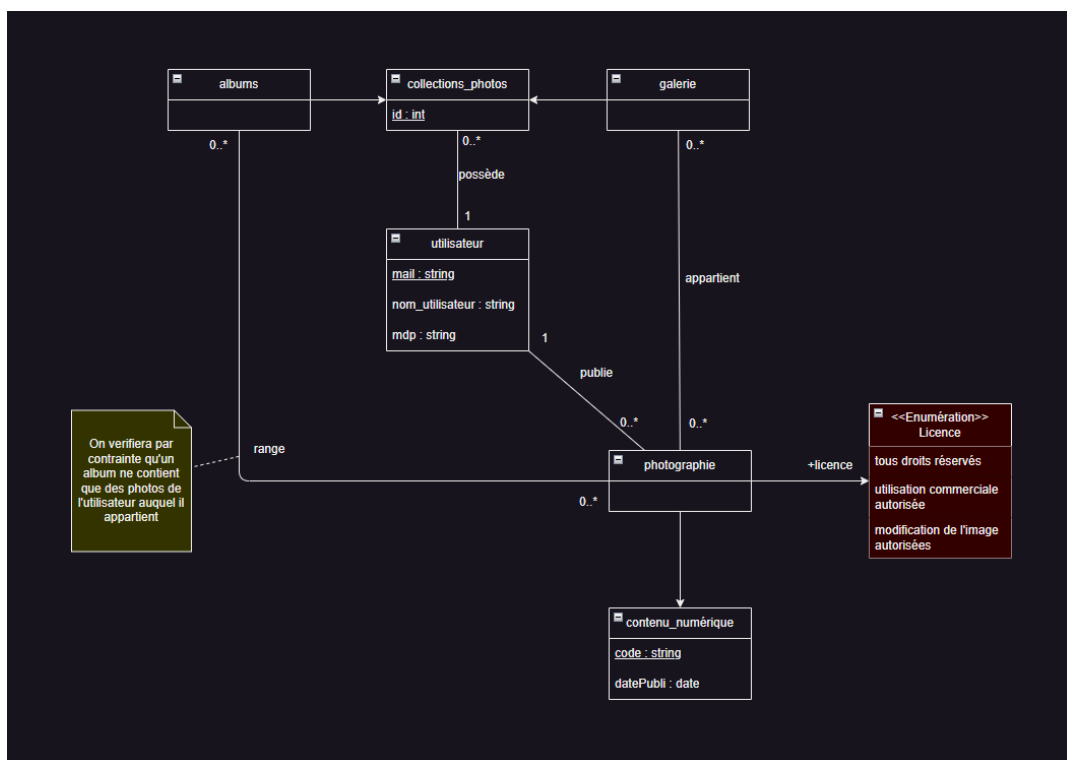


Schéma UML du 2nd cahier des charges d'après les choix de modélisations effectués.

2.3 UML n°3

Enfin, pour le dernier cahier des charges, nous avons choisi de modéliser les classes : photographie et commentaire ainsi que leur classe parent contenu_numérique, utilisateurs, mot_cle et discussion.

Nous avons choisi de modéliser les commentaire par deux classes : une classe discussion qui concerne une photographie et une classe commentaire. Ainsi, une discussion est créée à l'écriture du premier commentaire par un utilisateur. La discussion doit être supprimé en cascade lorsque la photographie qui la concerne l'est.

Nous avons aussi choisi de modéliser contenu_numérique comme une super classe de commentaire et de photographie afin de gérer leurs attributs communs.

Enfin, pour les mots-clés, nous avons modélisé une classe mot_cle avec pour seul attribut le mot clé en question. Chaque utilisateur peut crée un nouveau mot-clé à condition que celui-ci n'existe pas et un utilisateur peut décrire ses photographies à l'aide de ces mots-clés.

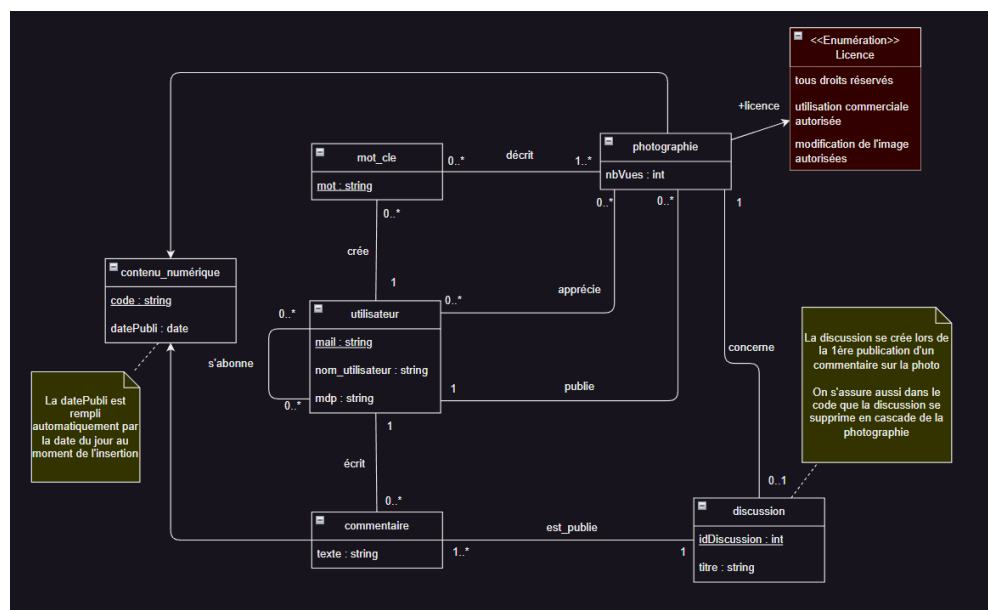


Schéma UML du 3ème cahier des charges d'après les choix de modélisations effectués.

Implémentation du cahier des charges n°1

Dans cette partie, nous nous proposons d'implémenter le modèle UML n°1 d'abord dans un schéma relationnel, puis, en SQL sous Oracle.

3.1 Schéma logique

Nous avons donc décidé d'implémenter le premier modèle UML, en voici le schéma logique (ou MLD) :

```
utilisateur = (mail STRING, nom_utilisateur STRING, mdp STRING);  
  
contenu_numerique = (code STRING, date_publie STRING);  
  
appareil_photo = (numero INT, modele STRING, marque STRING);  
  
configuration = (ouverture_focal INT, temps_exposition INT, flash INT, distance_focal INT);  
  
photographie = (#code STRING, #date_publie DATE, licence  
{'tous_droits_réserves','utilisation_commerciale_autorisee','modification_de_l_image_autorisees'}, lieu  
STRING, date_realisation DATE, #mail STRING, #ouverture_focal INT, temps_exposition INT, flash  
BOOLEAN, distance_focal INT, #numero INT);
```

Modèle MLD du schéma n°1

3.2 Implémentation Oracle

Nous avons ensuite implémenter le modèle sous Oracle (voir fichiers ci-joints). Nous avons tenter d'implémenter l'héritage dans les création de table SQL mais celui-ci ne semble pas faisable en Oracle, tout du moins nous n'avons pas trouvé comment le faire. Nous avons donc choisi de répéter les attributs de la super-classe dans les sous-classes correspondantes afin de modéliser cette héritage. La clé primaire d'une photographie est donc, par exemple, bien aussi une clé étrangère de la clé primaire d'un contenu numérique.

Nous avons ensuite écrits les requêtes SQL demandés, les voici (celles-ci sont également jointes via le fichier "requetes.sql") :

Les photos de Montpellier :

```
select * from photographie where lieu='Montpellier';
```

Les photos prisent par l'utilisateur nomme "Bob" :

```
select * from photographie join utilisateur on utilisateur.mail=photographie.mail where  
utilisateur.nom_utilisateur='Bob';
```

Les photos n'ayant pas la licence "tous droits réservés" et n'ayant pas été prises à Montpellier :

```
select * from photographie where licence!='tous_droits_réserves' and lieu!='Montpel-  
lier';
```

Requête qui permet de trouver les photos les plus appréciées avec la licence de distribution 'tous droits réservés' :

```
select * from photographie join apprecie on photographie.code=apprecie.code where li-  
cense='tous_droits_réserves' group by photographie.code having count(*)>=all(select  
count(*) from photographie join apprecie on photographie.code=apprecie.code where  
licence='tous_droits_réserves' group by apprecie.code);
```

Requête qui permet de trouver les photos incluses dans le plus grande nombre de galeries :

```
select code from appartient group by code having count(*)>=all(select count(*) from  
appartient group by code);
```

Conclusion

Dans ce TP, nous avons donc revu l'usage des outils de base de la modélisation et de la création de base de données SQL, en particulier Oracle. Nous avons pu revoir en détail la modélisation UML des bases de données ainsi que l'écriture d'un schéma relationnel.