

# Database-TP1

RAMBAL JULIEN , M1-IASD

September 22, 2022



# 1 Introduction

Dans le cadre de l'unité d'enseignement Big Data présenté conjointement par M.Ulliana et Mme.Arignon nous avons été chargés d'établir une base de données cohérente pour le stockage d'oeuvre d'art sur Flickr .

Pour ce fait, nous allons procéder en 3 étapes : la présentation de l'application Flickr , la réalisation de bases de données associées aux services proposés par Flickr et finalement nous implémenterons un de ses services .

## 2 Flickr

Flickr est une plateforme basée essentiellement sur le partage de photo et de vidéo il peut faire office de vitrine à une activité professionnelle , former les nouveaux talents et surtout faire découvrir les artistes de la plateforme .Nous ne nous intéresserons pas à la partie présentation et ergonomie du site car elle n'est pas directement reliée au modèle principal de la base de données .

### 2.1 Place sur le marché

.....

### 2.2 Problématique

.....

### 2.3 Services actuellement proposés

Pour faire découvrir ses artistes Flickr proposent 2 options :

La première se matérialise sous la forme d'une barre de recherche, nous pouvons y spécifier des mots-clefs, mais également l'orientation de l'image la taille minimale, une période ou date de prise de vue, le média, la teinte globale , le style et finalement la licence.

La seconde se présente sous la forme d'une carte du monde où nous pouvons spécifier une suite de mots clés et une localisation .

Ces 2 méthodes s'appuient sur une méthode de référencement qui s'appuie sur la pertinence des résultats par rapport aux thèmes de la recherche , ces résultats sont classés grâce à un algorithme qui utilise le nombre de vues , de commentaires , les likes ainsi que la côte de l'artiste sur le site .

Pour entretenir l'aspect "réseaux social" et apprentissage de la plateforme :

Les utilisateurs peuvent participer à des discussions. Les discussions se partagent en 2 catégories principales: les événements/partage de photos et les demandes d'ordre technique ( photo + site ) .

Les événements permettent de rassembler les utilisateurs à travers un projet collaboratif qui sera ensuite mis en avant par la plateforme, du contenu exclusif peut être produit pour chacun des événements mais plus généralement les auteurs poste les photos de leur répertoire qu'ils estiment être les plus dignes d'intérêt . Les pièces choisies pour constituer un événement et leur agencement ne sont pas gérées par l'algorithme de référencement de Flickr ce sont les utilisateurs gestionnaires du groupe qui s'en occupent . Les demandes d'ordre technique servent à augmenter la qualité des œuvres .

Les utilisateurs peuvent partager leurs œuvres avec leur famille , leurs amis ou bien les personnes qu'ils suivent (qu'on suppose être des mentors , des références ou des confrères ) , suivre des personnes mais aussi intégrer des groupes d'aides ou de partage .

Finalement, nous avons une partie statistique qui permet notamment de voir les modèles d'appareil photo les plus utilisés sur la plateforme.

### 2.4 Plan d'action :

Pour encourager la productivité des artistes nous avons décidé d'implémenter un système de référencement propre aux œuvres il se basera sur le nombre de vues , de commentaires et de like.

Ensuite, nous mettrons en avant nos artistes en fonction de leur croissance en nombre de followers mais également grâce à la popularité de leur photo ( nb de vues , like et commentaires cumulés )

Finalement, nous encouragerons le référencement par commentaire et tags en proposant des outils de recherche spécifique mais aussi un auto renseignement des oeuvres via "un robot labellisateur" .

### 3 Modèle Conceptuel et Physique :

Tout d'abord nous allons formaliser le vocabulaire que nous allons utiliser pour faciliter la compréhension de notre modèle , puis nous fixerons la granularité , les dimensions et les faits pour les services précédemment évoqués . Ensuite nous proposerons un modèle conceptuel sous forme d'un schéma UML inspiré d'une construction en étoile pour répondre à chacun des 3 cahiers des charges ainsi qu'aux problématiques vues plus haut. Finalement, nous discuterons d'une implémentation possible de la base de données et nous l'implémenterons .

#### 3.1 Vocabulaire

- Album : un album est une collection de post qui appartiennent au propriétaire de l'album
- Commentaire : un commentaire est un message dans une discussion , il est écrit par un utilisateur
- Contenu numérique : un contenu numérique représente tous les objets visible par les utilisateur du site
- Discussion : une discussion est une suite potentiellement vide de commentaire a destination d'un post
- Galerie : une galerie est une collection de post de plusieurs artistes
- Oeuvre : une oeuvre est un objet artistique présent sur le site ( photo,vidéo ,ect )
- Post : un post symbolise la publication d'une oeuvre sur le site
- Tag : un tag est un mot-clef qui peut être déposé par des utilisateurs pour référencer un post

#### 3.2 Attention :

Dans les schémas UML que nous vous présentons nous ne préciserons que les clefs étrangères qui ne sont ni issu d'un héritage ni propre aux tables d'associations .

Les clefs primaires sont spécifier si et seulement si elles sont composés et ne sont pas des clefs primaires d'une table d'association . Ainsi aucune ne figure dans ces modèles .

Finalement les noms des relations et des attributs sont différents du modèle physique ( implémentation du modèle conceptuel) pour des raisons de clareté .

### 3.3 Sujet 1 :

#### 3.3.1 Cahier des Charges

##### ●● Cahier des charges #1 : les photos

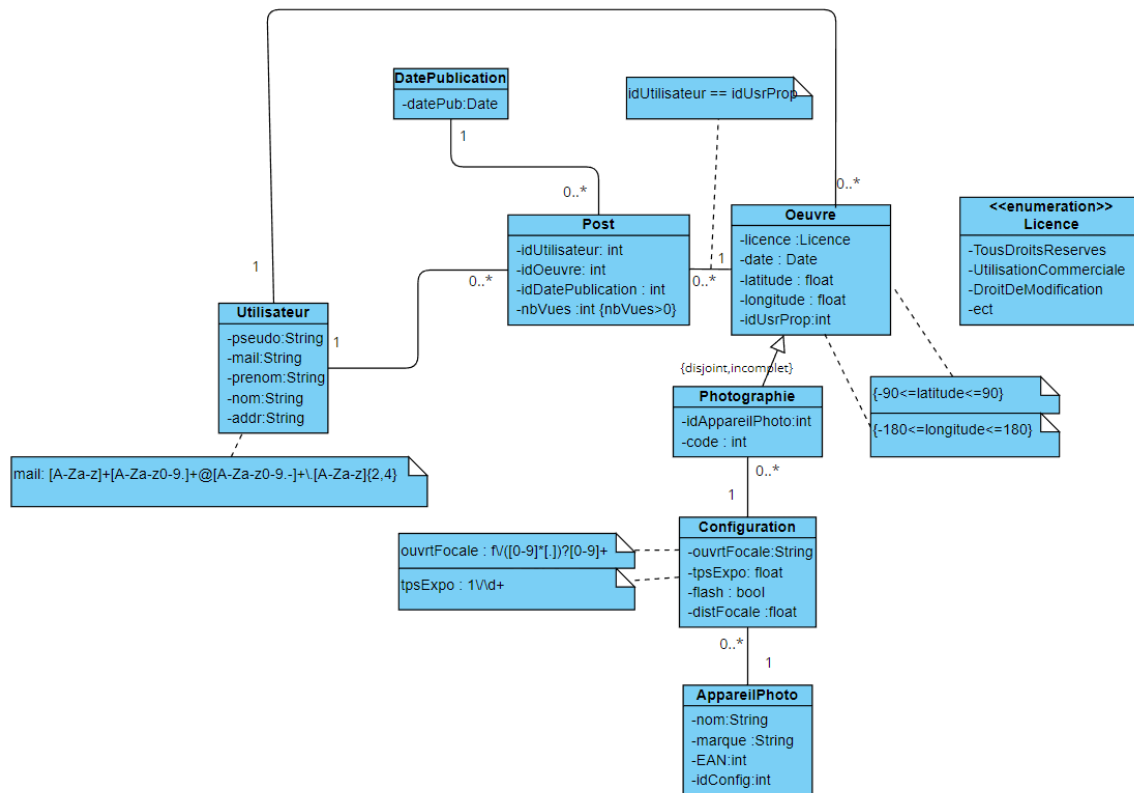
1. Une *photographie* est un objet numérique publié par un *utilisateur* à une date précise.
2. Une photographie est réalisée avec un *appareil photo* dont la *configuration* au moment du déclic prévoit plusieurs paramètres (ouverture focale, temps d'exposition, flash oui/non, distance focale, etc.). On assume qu'une configuration ait du sens seulement lorsqu'elle est associée à un dispositif.
3. Le lieu (coordonnées géographiques) et la date de réalisation de l'image sont enregistrés.
4. Une photographie contient un code pour la repérer dans le système de stockage distribuée des photos de l'application.
5. Les photos ont une *licence de distribution* (tous droits réservés, utilisation commerciale autorisée, modifications de l'image autorisées).

Ce cahier des charges concerne la mise en avant des posts

#### 3.3.2 Modèle Conceptuel :

Conformément aux objectifs de la plateforme Flickr nous avons décidé de choisir le "post" (événement de publication) pour table de fait. Un post sera assujéti à un utilisateur, une date de publication ainsi qu'à une œuvre. Le nombre de vues est l'un des attributs quantitatifs qui nous intéresse dans cette étude.

- Nous ne différencions pas 2 appareils photo distincts qui appartiennent au même modèle.
- Nous considérons que 2 modèles d'appareil photo distinct ne peuvent avoir une même configuration, car les caractéristiques techniques de l'appareil peuvent l'empêcher (= configuration potentiellement redondante).
- Nous avons créé la relation Oeuvre pour distinguer les vidéos des photos ect.
- Nous considérons qu'une unique œuvre est présentée dans chaque post mais qu'une oeuvre peut être publié plusieurs fois.
- Nous avons décidé de représenter le lieu de chaque œuvre par une latitude et longitude.
- Nous avons également choisi de ne pas créer des relations pour séparer le lieu et la date de l'oeuvre car ça nous aurait contraint à faire davantage de jointure.



Ce cahier des charges présente l'entité post ( centre de notre bdd )

### 3.3.3 Modèle Physique :

Nous allons tout d'abord présenter les contraintes que nous avons spécifier sur les relations :

- Seul l'utilisateur propriétaire de l'oeuvre pourra la poster post ne peut pas être plus ancien que l'oeuvre qui lui est associé
- Tous les paramètres d'une configuration d'appareil photo présentés dans le schéma UML sont obligatoires

#### Implementation :

Photographie aura très certainement des spécificités par rapport a vidéo mais en état il ne sert pas a grand chose ( nous verrons un cas similaire avec galerie et Album ) .Exemple de problème lié a cette modélisation:

- Pour accéder à l'ensemble des photos publié il faut faire de multiples jointure Post sur Oeuvre puis Oeuvre sur Photographie .  
Solution : un index bitmap pourrait distinguer les oeuvres photos des oeuvres vidéos ect .
- Pour connaître l'appareil qui a pris le cliché publié numéro X il sera nécessaire de faire encore plus de jointure .  
Solution : nous pourrions envisager un index ou bien de "supprimer physiquement " la table Oeuvre en faisant redescendre ses attributs sur ses classes filles .

Pour résoudre le souci de la géolocalisation des villes nous avons importer la base de données <https://sql.sh/736-base-donnees-villes-francaises> .Grace à elle nous allons pouvoir identifier la ville qui correspond aux coordonnées de l'image :

Tout d'abord, nous estimons un disque centré sur la localisation de la ville (ville\_[longitude—latitude] \_deg ) qui a la même aire que la ville (ville.\_surface ) et nous calculons son rayon .

Ensuite nous estimons grossièrement la latitude et longitude maximale et minimale de chaque ville grâce aux formule 1 à 4 ci-dessous .

Finalement savoir précisément si un point est dans une ville on vérifie que la distance entre le centre de la ville et le point est inférieur au rayon précédemment estimé ( voir 5 ème formule ci dessous ) .

Remarque : idéalement, nous aurions dû faire une recherche "nearest-neighbour".

- 1)  $max\_latitude = latitude + (rayon/rayon\_terre) * (180/pi);$
- 2)  $min\_latitude = latitude - (rayon/rayon\_terre) * (180/pi);$
- 3)  $max\_longitude = longitude + (rayon/rayon\_terre) * (180/pi)/cos(latitude * pi/180);$
- 4)  $min\_longitude = longitude - (rayon/rayon\_terre) * (180/pi)/cos(latitude * pi/180);$
- 5) si  $haversine(pointville, point) < rayon$  alors  $point$  appartient à la ville

Voir "Les publications localisées sur Montpellier" (dans la section Requête SQL et Code à la question 3) pour l'implémentation de cette méthode et ses résultats

## 3.4 Sujet 2 :

### 3.4.1 Cahier des Charges

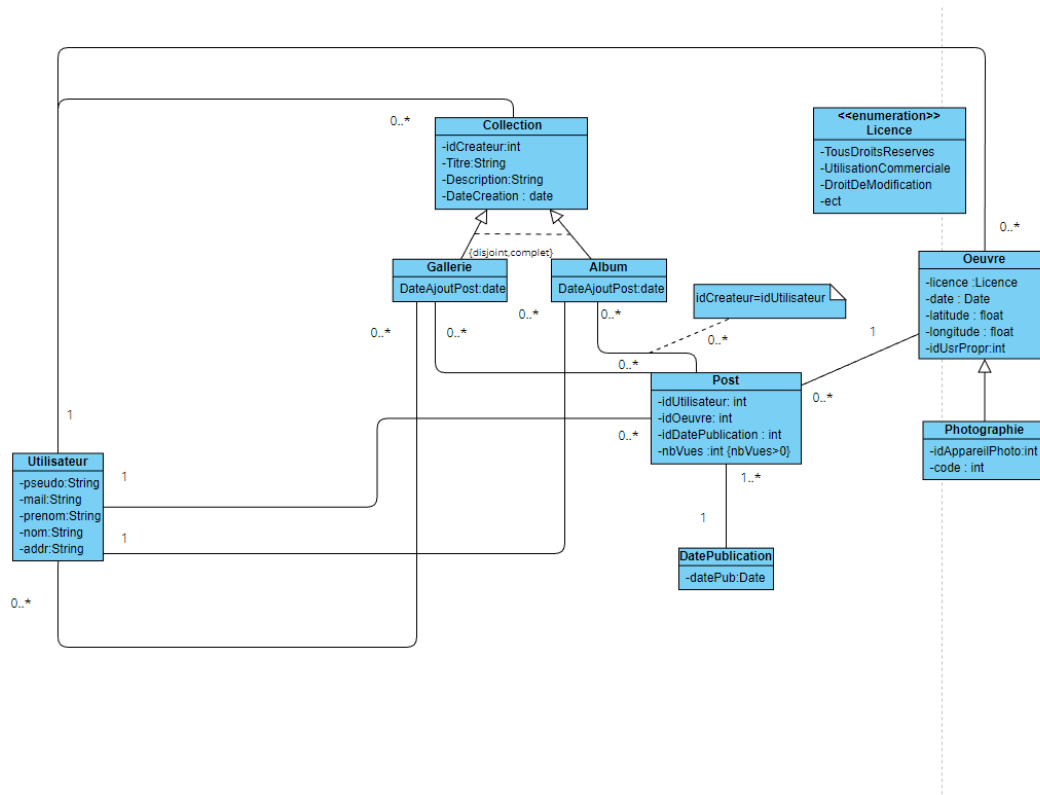
#### ●● Cahier des charges #2 : publications, albums, et galeries

1. Un utilisateur peut ranger des photos dans des *albums* ou des *galeries*.
2. Toutes les photos d'un album appartiennent au propriétaire de l'album.
3. À la différence des albums, les galeries permettent de collectionner (aussi) des images d'autres utilisateurs.
4. Albums et galeries sont les seuls cas particuliers des *Collections de Photos*.

Ce cahier des charges concerne la mise en avant des Albums (la vitrine de l'artiste ) mais aussi galerie (projet communautaire : event ) .

### 3.4.2 Modèle Conceptuel :

Flickr permet à ses artistes de réutiliser les posts d'autres artistes pour constituer une collection de photo . Il est pour nous essentiel de connaître la date d'ajout des posts aux 2 collections pour savoir si elles ont permis de populariser ces œuvres et leurs artistes.



### 3.4.3 Modèle Physique :

Les albums et les galeries pourraient être vue soit comme des objets à part entière (avec leur propre id et leur attribut ) soit comme des "tables d'associations" qui relient post à collection cela dit le cahier des charges ne fournit aucun attribut spécifique a galerie ou a album ainsi nous les avons modéliser comme des "associations" .

Une autre manière de faire aurait été de créer une association ternaire rangerAlbum/Galerie contenant la date d'ajout et impliquant la table post , la table utilisateur (redondant) et Album/Galerie (table sans attribut mise à part une id) . Nous aurions pu aussi créer une association binaire Galerie/Album - Post contenant la date .



### 3.5 Sujet 3:

Pour inciter les utilisateurs à interagir sur la plateforme mais également la rendre plus qualitative et efficace Flickr à instaurer un système de like , discussions et tag .

Le nombre de Like d'un post améliorera son référencement tout comme son nombre de commentaires . Les tags quant à eux renforceront l'engagement des utilisateurs et surtout ils permettront d'augmenter la pertinence des propositions faite par les algorithmes de référencement et de recommandation . En plus de quoi nous pourrions proposer des recherches par tag aux utilisateurs .

Finalement le système de follower permettra d'estimer la cote d'un artiste sur la plateforme . Toutes ces nouvelles fonctionnalités permettront de faire émerger des tendances sur le site et de chiffrer sa dynamique .

NB: il serait préférable de robotiser une partie de la labélisation des images par Tagg

#### 3.5.1 Cahier des Charges

##### ●● Cahier des charges #3 : les interactions entre utilisateurs

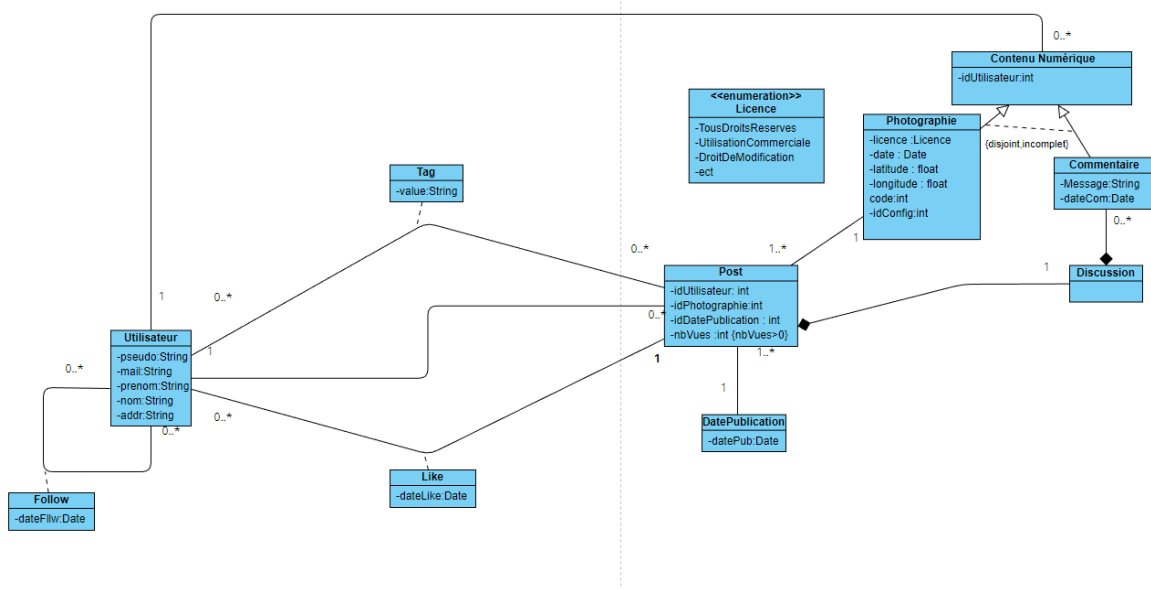
1. Les utilisateurs de la plateforme Flickr interagissent avec différentes manières.
  - (a) en appréciant des photos (le "Like", à la Facebook)
  - (b) en laissant des commentaires dans des discussions autour d'une image
  - (c) en s'abonnant aux publications d'autres utilisateurs (le "follow", à la Twitter).
2. Un utilisateur peut marquer ses photos et leur associer des balises (tags, mot-clés).
3. Le nombre de fois qu'une photographie a été visualisée est enregistré.
4. Une *discussion* autour d'une photo est constituée par un ensemble (non vide) de commentaires.
5. Un commentaire n'appartient qu'à une discussion.
6. L'effacement d'une photo de la plateforme implique aussi l'effacement de toute la discussion associée.
7. Commentaries et photographies sont des cas particuliers d'un *Contenu Numérique*.

Ce cahier des charges concerne la mise en avant des interventions entre utilisateur .

#### 3.5.2 Modèle Conceptuel :

Pour pouvoir représenter "la croissance" des posts et des artistes de la plateforme nous adjoignons à la relation Like , Commentaire et follow un attribut date . Nous considérons que chaque utilisateur du site peut labelliser chacun des posts présents sur le site . Nous avons supprimé l'entité "Oeuvres" car elle alourdirait le schéma sans répondre à une problématique formulée dans le cahier des charges 3 . Cela dit il est envisageable de re-implémenter "Oeuvre" comme sous-classe de "Contenu Numérique" si des distinctions devait être faite ( nous le verrons dans la prochaine partie ) .

L'entité suggéré mot clef n'est ici pas représenter car nous ne l'avons pas trouver sur le site .

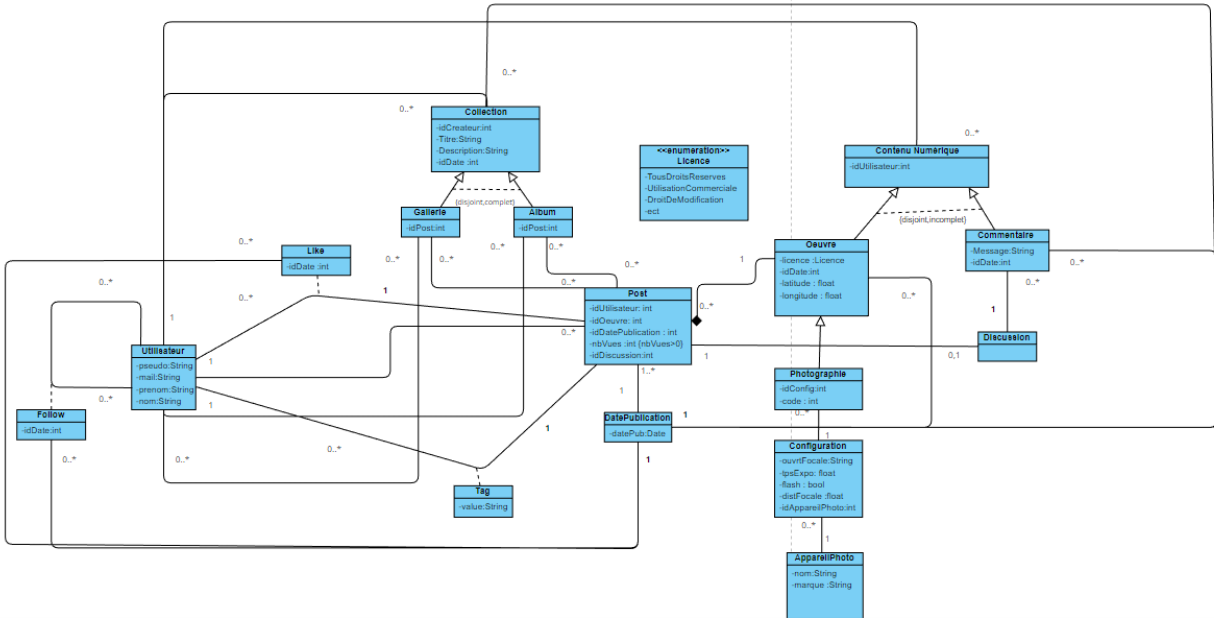


### 3.5.3 Modèle Physique :

- L'opération follow n'est pas réflexive (un utilisateur ne peut être followé par lui même) .
- Le followé ou le follower ne peut pas être nul ainsi si l'un des 2 comptes est supprimé le follow disparaît
- Le like doit cibler un post non nul et être performé par un utilisateur non nul ainsi si l'un des 2 comptes est supprimé le like disparaît.
- La date d'un like ne peut pas être antérieure au post liké .
- De même un commentaire ne peut pas être antérieur au post commenté .
- L'utilisateur associé au commentaire ne peut pas être nul
- Le commentaire appartient à une unique discussion qui ne peut être nulle
- Ainsi si l'utilisateur ou la discussion est supprimé le commentaire aussi
- Si le post disparaît, la discussion associée disparaît également,
- Le tagg doit cibler un post non null et être performé par un utilisateur non nul ainsi si l'un des 2 comptes est supprimé le tagg disparaît
- Finalement la date de publication d'un post ne peut pas être antérieure à la photographie postée . Un contenu numérique est (pour l'instant) soit une photographie soit un commentaire (mais pas les deux).

## 3.6 Implementation Finale

### 3.6.1 Modèle Conceptuel :



### 3.6.2 Modèle Physique :

Les clefs étrangères sont en italiques et les clés primaires sont soulignées .

utilisateur ( idutilisateur , pseudo , nom , prenom , mail )  
 mydate ( idmydate , datemydate )  
 contnum( idcontnum , *id\_uteur\_contnum* )  
 oeuvre( idoeuvre , *id\_contnum\_oeuvre* , *id\_mydate\_oeuvre* , licence , latitude , longitude )  
 appareil( idappareil , ean , marque )  
 config( idconfig , ouvfofocal , tpsexpo , flash , *id\_appareil\_config* , distfocal )  
 photo( idphoto , *id\_oeuvre\_photo* , *id\_config\_photo* , code )  
 discuss( iddiscus )  
 post ( idpost , *id\_uteur\_post* , *id\_mydate\_post* , *id\_oeuvre\_post* , *id\_discus\_post* , nbvues )  
 commente( idcommente , *id\_contnum\_commente* , *id\_discus\_commente* , *id\_mydate\_commente* , content )  
 tageu ( idtageu , *id\_uteur\_tageu* , *id\_post\_tageu* , tagvalue )  
 likeu ( id\_uteur\_likeu , *id\_mydate\_likeu* , *id\_post\_likeu* )  
 follow ( id\_uteur\_follow , *id\_uteur\_follow2* , *id\_mydate\_follow* )  
 collection ( idcollection , *id\_uteur\_collection* , titre , description , *id\_mydate\_collection* )  
 gallerie ( idgallerie , *id\_collection\_gallerie* , *id\_post\_gallerie* )  
 album ( idalbum , *id\_collection\_album* , *id\_post\_album* )

Explication:

- La table mydate n'est pas correctement utilisé ni construite , elle devrait être construite par une succession d'attributs non composé (ex: jour , mois , ect ) à la place de datemydate et elle devrait n'être référencé que par la table de fait post cela dit il est plus commode de faire comme ça pour cette première ébauche .
- Nous pourrions supprimer discussion mais conserver son identifiant (id\_discus\_commente) dans commentaire.
- Dans post il serait préférable d'avoir comme clef primaire une clef composite sur les clef étrangère mais c'est assez pénible quand il faut référencer l'entité post avec son id donc nous avons choisi une clef primaire simple
- Nous aurions pu mettre dans contnum un attribut date mais la relation d'héritage étant incomplète nous n'avons pas souhaiter nous prononcé sur les attributs des futures héritière .
- La relation oeuvre permet de distinguer les photos des autres oeuvres présentes sur flickr comme les vidéos et les modèles 3D (bien qu'ils ne sont pas ici présenté ).
- L'oeuvre possède une localisation en latitude et longitude en effet les appareils photos numérique spécialement les téléphones renseignent les coordonnées GPS des clichés pris . Cela dit les oeuvres faites sur logicielle exclusivement sera renseigné automatiquement par le système grâce à l'adresse de l'utilisateur (cf. conversion ville en coordonnée partie 2 question 3 a ) .
- La configuration de l'appareil est directement lié à l'appareil car l'artiste est susceptible de prendre plusieurs clichés avec la même configuration .
- Le post peut n'avoir aucune discussion associé à sa création "post : no comment " mais également suite à un débordement dans l'espace de discussion .

## 4 Requête SQL et Code

### 4.1 Question 1 :Fournir le code d'implémentation

Voir Appendice : code intégral ( partie 0,1 et 3 )

### 4.2 Question 2 : Peuplé la base de donnée

Voir Appendice : code intégral (partie 2)

```
110 |
111 | -- ICI :
112 | INSERT INTO CONFIG(IDCONFIG ,OUVFOCAL ,TPSEXPO ,FLASH ,ID_APPAREIL_CONFIG ,
    | DISTFOCAL ) VALUES(3,'f/5.6','1/250',1,3,200);
113 | INSERT INTO CONFIG(IDCONFIG ,OUVFOCAL ,TPSEXPO ,FLASH ,ID_APPAREIL_CONFIG ,DISTFOCAL
    | ) VALUES(5,'F/11','1/60',1,3,28);
114 | -- meme appareil mais 2 config differente
115 |
116 |
117 |
118 | INSERT INTO PHOTO (IDPHOTO,ID_OEUVRE_PHOTO,ID_CONFIG_PHOTO,CODE) VALUES (1,2,4,9);
119 | INSERT INTO PHOTO (IDPHOTO,ID_OEUVRE_PHOTO,ID_CONFIG_PHOTO,CODE) VALUES (2,6,7,10);
120 | INSERT INTO PHOTO (IDPHOTO,ID_OEUVRE_PHOTO,ID_CONFIG_PHOTO,CODE) VALUES (3,4,9,11);
121 | INSERT INTO PHOTO (IDPHOTO,ID_OEUVRE_PHOTO,ID_CONFIG_PHOTO,CODE) VALUES (4,5,4,12);
122 |
123 | --ICI
124 | INSERT INTO PHOTO (IDPHOTO,ID_OEUVRE_PHOTO,ID_CONFIG_PHOTO,CODE) VALUES (5,7,5,13);
125 | INSERT INTO PHOTO (IDPHOTO,ID_OEUVRE_PHOTO,ID_CONFIG_PHOTO,CODE) VALUES (6,3,3,14);
126 | -- 2 photos differentes
127 |
128 | INSERT INTO PHOTO (IDPHOTO,ID_OEUVRE_PHOTO,ID_CONFIG_PHOTO,CODE) VALUES (7,8,1,15);
129 | INSERT INTO PHOTO (IDPHOTO,ID_OEUVRE_PHOTO,ID_CONFIG_PHOTO,CODE) VALUES (8,9,6,16);
130 | INSERT INTO PHOTO (IDPHOTO,ID_OEUVRE_PHOTO,ID_CONFIG_PHOTO,CODE) VALUES (9,1,8,17);
131 |
132 | --ICI
133 | INSERT INTO POST (IDPOST ,ID_USEUR_POST,ID_MYDATE_POST,ID_OEUVRE_POST ,
    | ID_DISCUS_POST,NBVUES ) VALUES (1,1,7,2,2,120);
134 | INSERT INTO POST (IDPOST ,ID_USEUR_POST,ID_MYDATE_POST,ID_OEUVRE_POST ,
    | ID_DISCUS_POST,NBVUES ) VALUES (2,1,9,1,1,130);
135 | -- meme Utilisateur
136 |
```

## 4.3 Question 3 : requête SQL

### 4.3.1 Les publications localisées sur Montpellier

Exemple de tuple nécessaires à la réalisation de la requête : (voir <https://sql.sh/736-base-donnees-villes-francaises> pour l'ensembles des tuples )

```
01 |  
02 | -- Un exemple de la table ville_france_free :  
03 |  
04 | INSERT INTO villes_france_free (ville_id, ville_departement, ville_slug, ville_nom,  
    ville_nom_simple, ville_nom_reel, ville_nom_soundex, ville_nom_metaphone,  
    ville_code_postal, ville_commune, ville_code_commune, ville_arrondissement,  
    ville_canton, ville_surface, ville_longitude_deg, ville_latitude_deg,  
    ville_longitude_grd, ville_latitude_grd, ville_longitude_dms, ville_latitude_dms  
    , ville_zmin, ville_zmax) VALUES (13339, '34', 'montpellier', 'MONTPELLIER', '  
    montpellier', 'Montpellier', 'M3146', 'MNTPLR', '34000-34070-34080-34090', '172'  
    , '34172', 3, '99', 56.88, 3.88333, 43.6, '1711', '48457', '+35236', '433640',  
    7, 121) ;
```

Fonctions nécessaires à la réalisation de la requête :

Voir Appendice : code intégral partie 3.

Requête SQL :

```
01 |  
02 | -- Reponse pour Montpellier  
03 |  
04 | WITH t1 AS (  
05 | SELECT surfaceToDist (ville_surface) as ville_rayon ,ville_longitude_deg ,  
    ville_latitude_deg FROM villes_france_free  
06 | WHERE ville_slug LIKE 'Montpellier' OR ville_nom LIKE 'Montpellier' OR  
    ville_nom_simple LIKE 'Montpellier' OR ville_nom_reel LIKE 'Montpellier'  
07 | )  
08 | SELECT idpost,idoeuvre , longitude ,latitude FROM OEUVRE , t1  
09 | ,POST  
10 | WHERE  
11 | idoeuvre = POST.id_oeuvre_post AND longitude <= max_min_lng (1, latitude ,  
    longitude ,t1.ville_rayon ) AND longitude >= max_min_lng (0, latitude ,  
    longitude ,t1.ville_rayon )  
12 | AND latitude <= max_min_lat (1, latitude ,t1.ville_rayon ) AND latitude >=  
    max_min_lng (0, latitude , longitude , t1.ville_rayon )  
13 | AND haversine ( latitude , longitude , t1.ville_latitude_deg , t1.  
    ville_longitude_deg ) <= t1.ville_rayon ;
```

Resultat et precision obtenu :



La zone hachurer en rouge est la zone non reconnu comme etant dans Montpellier a tord , la zone colorier en jaune est reconnu a tord comme etant dans Montpellier , la zone delimiter en bleu est reconnu a raison comme etant dans Montpellier , et la croix est le "centre" de Montpellier d'après la base de donnée sql : ville de France

Observation : Le modèle n'est pas précis sur Montpellier nous obtenons de meilleur résultat dans d'autre ville comme Paris ou le point central de la ville est plus précisément placé dans la base de donnée : ville de france .

### 4.3.2 Marques les plus populaires :

Inspiré de Flickr : <https://www.flickr.com/cameras>

### Marques d'appareils photo utilisées par les membres de la communauté Flickr

Classement ▾	Marques	Modèles les plus populaires	Types de modèles	# de modèles
1	Apple	iPhone 11, iPhone 13 Pro, iPhone 13 Pro Max	Cameraphone, Point and Shoot, Tablet	45
2	Canon	EOS 5D MARK IV, EOS R5, EOS 5D Mark III	Digital SLR, Film Scanner, Mirrorless Camera, Point and Shoot, Video Camera	306
3	Nikon	D750, D850, D500	Digital SLR, Mirrorless Camera, Point and Shoot	223
4	Sony	A7 III, A7R III, A6000	Cameraphone, Digital SLR, Mirrorless Camera, Point and Shoot, Video Camera	374
5	Fujifilm	X-T3, X-T4, X-T2	Digital SLR, Film Scanner, Mirrorless Camera, Point and Shoot	205
6	Panasonic	DC-G9, DMC-FZ1000, DC-S5	Digital SLR, Mirrorless Camera, Point and Shoot, Video Camera	204
7	Samsung	Galaxy S21 Ultra, Galaxy S8, S21 5G	Cameraphone, Digital SLR, Mirrorless Camera, Point and Shoot, Video Camera	284

```
01 |  
02 | SELECT ROW_NUMBER() OVER (ORDER BY COUNT(idAppareil) DESC ) AS classement , marque ,  
    LISTAGG (DISTINCT nom , ',' ) WITHIN GROUP ( ORDER BY nom ) as modeles , COUNT(  
    idAppareil) as nb_de_modeles , COUNT(*) * 100.0 /SUM(COUNT(*)) OVER() as ratio  
    FROM APPAREIL GROUP BY marque ;  
03 |  
04 | -- TRUNCATE(ratio , 3 ) : pour avoir un resultat plus "humain"
```

[illegible]

Nous n'avons pas trier les modèles de la liste de chaque marque "par ordre d'importance" uniquement par ordre alphabétique . Comme le montre ce tableau :

[illegible]



#### 4.3.3 Troisième requête : Les publications ayant été labellisées 'blue' triées par nombre de vues

Inspiré de la recherche par tagg de Flickr :

```
01 | SELECT idpost , tagvalue ,nbVues FROM POST , TAGEU WHERE tageu.tagvalue = 'blue'
    | AND idpost= id_post_tageu ORDER BY nbVues DESC ;
02 | ;
```

IDPOST	TAGVALUE	NBVUES
4	blue	50
9	blue	45

#### 4.3.4 Question b : Les photos les plus populaires avec la licence tousDroitsReserves

```
01 | WITH t2 AS (
02 |   SELECT idpost , idphoto , COUNT (*) as nb_like FROM LIKEU ,PHOTO ,POST , oeuvre
    |   WHERE licence ='TousDroitsReserves' AND idoeuvre = id_oeuvre_post AND
    |   id_oeuvre_post = id_oeuvre_photo AND id_post_likeu = idpost
03 |   GROUP BY idpost , idphoto
04 | )
05 | SELECT idpost , idphoto , nb_like FROM t2 WHERE t2.nb_like >= ALL(SELECT t2.
    |   nb_like FROM t2);
```

IDPOST	IDPHOTO	NB_LIKE
9	2	3

(Comme nous l'avons évoqué dans la cahier des charges 1 un index nous aurait épargné la jointure avec Photo )

#### 4.3.5 Question c : Les photos contenues dans le plus de galleries

```
01 | WITH t3 AS (
02 |   SELECT idpost , idphoto , COUNT (*) as nb_gallerie FROM GALLERIE INNER JOIN POST
    |   ON idpost =id_post_gallerie INNER JOIN OEUVRE ON idoeuvre=id_oeuvre_post INNER
    |   JOIN PHOTO ON idoeuvre = id_oeuvre_photo GROUP BY idpost , idphoto
03 | )
04 | SELECT idpost , idphoto , nb_gallerie FROM t3 WHERE t3. nb_gallerie >= ALL(
    |   SELECT t3. nb_gallerie FROM t3);
```

IDPOST	IDPHOTO	NB_GALLERIE
8	7	2
2	9	2

## References

- [1] *HAI708I :Entrepot de donnée et Big Data* , Anne-Muriel Arigon et Federico Ulliana 2022.,  
<https://moodle.umontpellier.fr/course/view.php?id=22769>
- [2] *Kimball datawarehouse*,  
<https://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/>
- [3] *Multidimensional Databases and Data Warehousing* , Christian S. Jensen , 2010.,  
<https://link.springer.com/book/10.1007/978-3-031-01841-1>
- [4] *Documentation oracle* ,  
<https://oracle-base.com/articles/articles>
- [5] *Forum Oracle* ,  
<https://asktom.oracle.com/>
- [6] *Tutorial Oracle* ,  
<https://livesql.oracle.com/>
- [7] *Oracle Database 12c PLSQL Programming* , Michael McLaughlin

# Appendices

## Declaration de la table ville france free :

```
01 |
02 | BEGIN
03 | EXECUTE IMMEDIATE 'DROP TABLE villes_france_free';
04 | EXCEPTION
05 | WHEN OTHERS THEN
06 |     IF SQLCODE != -942 THEN
07 |         RAISE;
08 |     END IF;
09 | END;
10 | /
11 |
12 | CREATE TABLE villes_france_free (
13 |     ville_id NUMBER(8) ,
14 |     ville_departement varchar(3) ,
15 |     ville_slug varchar(255) ,
16 |     ville_nom varchar(45) ,
17 |     ville_nom_simple varchar(45) ,
18 |     ville_nom_reel varchar(45) ,
19 |     ville_nom_soundex varchar(20) ,
20 |     ville_nom_metaphone varchar(22) ,
21 |     ville_code_postal varchar(255) ,
22 |     ville_commune varchar(3) ,
23 |     ville_code_commune varchar(5) ,
24 |     ville_arrondissement NUMBER(3) ,
25 |     ville_canton varchar(4) ,
26 |     ville_surface float ,
27 |     ville_longitude_deg float ,
28 |     ville_latitude_deg float ,
29 |     ville_longitude_grd varchar(9) ,
30 |     ville_latitude_grd varchar(8) ,
31 |     ville_longitude_dms varchar(9) ,
32 |     ville_latitude_dms varchar(8) ,
33 |     ville_zmin NUMBER(8) ,
34 |     ville_zmax NUMBER(4)
35 | );
36 |
37 | prompt "Creation Ville france "
38 |
39 | -- Fonction pour interroger la table ville_france
40 | -- Retourne la longitude , latitude et surface associ au nom de ville pass en
   | param tre
41 | CREATE OR REPLACE FUNCTION findInfoVille (nomVille IN VARCHAR2 )
42 | RETURN VARCHAR2
43 | IS
44 | tmp VARCHAR2(300);
45 | BEGIN
46 |     tmp := 'SELECT ville_longitude_deg ,ville_latitude_deg ,ville_surface FROM
   | villes_france_free WHERE ville_slug LIKE ' || nomVille || 'OR ville_nom LIKE '
   | || nomVille || 'OR ville_nom_simple LIKE ' || nomVille || ' OR ville_nom_reel
   | LIKE ' || nomVille ;
47 |     RETURN tmp ;
48 | END;
49 | /
```

## Code integral : (mise à part les requêtes SQL )

### Partie 0 : suppression des relations

```
01 | prompt "Debut Suppression des relations"
02 |
03 |
04 | BEGIN
05 | EXECUTE IMMEDIATE 'DROP TABLE ALBUM';
06 | EXCEPTION
07 |   WHEN OTHERS THEN
08 |     IF SQLCODE != -942 THEN
09 |       RAISE;
10 |     END IF;
11 | END;
12 | /
13 | BEGIN
14 | EXECUTE IMMEDIATE 'DROP TABLE GALLERIE';
15 | EXCEPTION
16 |   WHEN OTHERS THEN
17 |     IF SQLCODE != -942 THEN
18 |       RAISE;
19 |     END IF;
20 | END;
21 | /
22 | BEGIN
23 | EXECUTE IMMEDIATE 'DROP TABLE COLLECTION';
24 | EXCEPTION
25 |   WHEN OTHERS THEN
26 |     IF SQLCODE != -942 THEN
27 |       RAISE;
28 |     END IF;
29 | END;
30 | /
31 | BEGIN
32 | EXECUTE IMMEDIATE 'DROP TABLE FOLLOW';
33 | EXCEPTION
34 |   WHEN OTHERS THEN
35 |     IF SQLCODE != -942 THEN
36 |       RAISE;
37 |     END IF;
38 | END;
39 | /
40 | BEGIN
41 | EXECUTE IMMEDIATE 'DROP TABLE LIKEU';
42 | EXCEPTION
43 |   WHEN OTHERS THEN
44 |     IF SQLCODE != -942 THEN
45 |       RAISE;
46 |     END IF;
47 | END;
48 | /
49 | BEGIN
50 | EXECUTE IMMEDIATE 'DROP TABLE TAGEU';
51 | EXCEPTION
52 |   WHEN OTHERS THEN
53 |     IF SQLCODE != -942 THEN
54 |       RAISE;
55 |     END IF;
56 | END;
57 | /
58 |
59 | BEGIN
60 | EXECUTE IMMEDIATE 'DROP TABLE COMMENTE';
61 | EXCEPTION
62 |   WHEN OTHERS THEN
63 |     IF SQLCODE != -942 THEN
64 |       RAISE;
```

```

65 |         END IF;
66 |     END;
67 | /
68 |
69 | BEGIN
70 | EXECUTE IMMEDIATE 'DROP TABLE POST';
71 | EXCEPTION
72 |     WHEN OTHERS THEN
73 |         IF SQLCODE != -942 THEN
74 |             RAISE;
75 |         END IF;
76 |     END;
77 | /
78 | BEGIN
79 | EXECUTE IMMEDIATE 'DROP TABLE DISCUS';
80 | EXCEPTION
81 |     WHEN OTHERS THEN
82 |         IF SQLCODE != -942 THEN
83 |             RAISE;
84 |         END IF;
85 |     END;
86 | /
87 |
88 | BEGIN
89 | EXECUTE IMMEDIATE 'DROP TABLE PHOTO';
90 | EXCEPTION
91 |     WHEN OTHERS THEN
92 |         IF SQLCODE != -942 THEN
93 |             RAISE;
94 |         END IF;
95 |     END;
96 | /
97 | BEGIN
98 | EXECUTE IMMEDIATE 'DROP TABLE CONFIG';
99 | EXCEPTION
100 |     WHEN OTHERS THEN
101 |         IF SQLCODE != -942 THEN
102 |             RAISE;
103 |         END IF;
104 |     END;
105 | /
106 |
107 | BEGIN
108 | EXECUTE IMMEDIATE 'DROP TABLE APPAREIL';
109 | EXCEPTION
110 |     WHEN OTHERS THEN
111 |         IF SQLCODE != -942 THEN
112 |             RAISE;
113 |         END IF;
114 |     END;
115 | /
116 |
117 | BEGIN
118 | EXECUTE IMMEDIATE 'DROP TABLE OEUVRE';
119 | EXCEPTION
120 |     WHEN OTHERS THEN
121 |         IF SQLCODE != -942 THEN
122 |             RAISE;
123 |         END IF;
124 |     END;
125 | /
126 | BEGIN
127 | EXECUTE IMMEDIATE 'DROP TABLE CONTNUM';
128 | EXCEPTION
129 |     WHEN OTHERS THEN
130 |         IF SQLCODE != -942 THEN
131 |             RAISE;
132 |         END IF;

```

```

133 | END;
134 | /
135 | BEGIN
136 | EXECUTE IMMEDIATE 'DROP TABLE MYDATE';
137 | EXCEPTION
138 |   WHEN OTHERS THEN
139 |     IF SQLCODE != -942 THEN
140 |       RAISE;
141 |     END IF;
142 | END;
143 | /
144 | BEGIN
145 | EXECUTE IMMEDIATE 'DROP TABLE USEUR';
146 | EXCEPTION
147 |   WHEN OTHERS THEN
148 |     IF SQLCODE != -942 THEN
149 |       RAISE;
150 |     END IF;
151 | END;
152 | /
153 |
154 | prompt "Fin Suppression des relations"

```

## Partie 1 : Creation des packages , tables et triggers

```
01 | set serveroutput on ;
02 |
03 |
04 | CREATE OR REPLACE PACKAGE my_global_varpackage IS
05 |     r_terre NUMBER;
06 |     pi NUMBER;
07 | END my_global_varpackage ;
08 | /
09 |
10 | my_global_varpackage.r_terre := 6378 ;
11 | my_global_varpackage.pi := 3.141592653589793238462643383279502884197169399375 ;
12 |
13 |
14 | CREATE OR REPLACE FUNCTION isIN (tableNameFk IN VARCHAR2 , colNameFk IN VARCHAR2 ,
15 |     valuer IN NUMBER )
16 | RETURN NUMBER
17 | IS
18 | tmp NUMBER;
19 | BEGIN
20 | EXECUTE IMMEDIATE 'SELECT COUNT(*) FROM ' || tableNameFk || ' WHERE ' || tableNameFk
21 |     || '.' || colNameFk || '=' || valuer INTO tmp ;
22 | return tmp ;
23 | END;
24 | /
25 |
26 | -- Utilisateur
27 | -- Utilisateur est l'ensemble des utilisateurs de la plateforme :
28 |
29 | CREATE TABLE USEUR (
30 |     IDUSEUR NUMERIC(6,0) ,
31 |     PSEUDO VARCHAR(16) UNIQUE,
32 |     NOM VARCHAR(16),
33 |     PRENOM VARCHAR(16),
34 |     MAIL VARCHAR(80),
35 |     -- ADRESSE VARCHAR(80),
36 |     CONSTRAINT PK_USEUR PRIMARY KEY (IDUSEUR)
37 | );
38 |
39 | CREATE OR REPLACE TRIGGER validate_mail
40 | BEFORE INSERT OR UPDATE ON USEUR
41 | FOR EACH ROW
42 | DECLARE
43 |     regex_validation VARCHAR(80) ;
44 | BEGIN
45 |     regex_validation := '^ [A-Za-z]+[A-Za-z0-9.]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,4}$'
46 |     ;
47 |     IF NOT REGEXP_LIKE (:NEW.mail , regex_validation) THEN
48 |         RAISE_APPLICATION_ERROR(-20500, 'Le mail n est pas valide ');
49 |     END IF ;
50 | END;
51 | /
52 | prompt "CREATION USEUR"
53 |
54 |
55 | --Mydate
56 | -- Mesure de Date sur la plateforme (suppos identique pour tous les evenemtns
57 |     dat )
58 | -- Tres important pour tracer l'volution /dynamique du site
59 |
60 | CREATE TABLE MYDATE (
61 |     IDMYDATE NUMERIC(6,0),
62 |     DATEMYDATE DATE,
```

```

62 |         CONSTRAINT PK_MYDATE PRIMARY KEY (IDMYDATE)
63 |     );
64 |
65 |
66 |     CREATE OR REPLACE FUNCTION isGreaterThenDate (id_date1 IN NUMBER , id_date2 IN
        NUMBER )
67 |     RETURN NUMBER -- pas boolean car sinon pas comparaison where clause
68 |     IS
69 |     tmp1 DATE;
70 |     tmp2 DATE;
71 |     res NUMBER ;
72 |     BEGIN
73 |     SELECT DATEMYDATE INTO tmp1 FROM MYDATE WHERE MYDATE.IDMYDATE = id_date1 ;
74 |     SELECT DATEMYDATE INTO tmp2 FROM MYDATE WHERE MYDATE.IDMYDATE = id_date2 ;
75 |     IF tmp1 > tmp2 THEN -- tmp1.month > tmp2.month OR ( tmp1.month = tmp2.month AND (
        tmp1.day > tmp2.day OR ( tmp1.day = tmp2.day AND ( tmp1.hours > tmp2.hours OR (
        tmp1.hours = tmp2.hours AND tmp1.minute > tmp2.minute OR ( tmp1.minute = tmp2.
        minute AND tmp1.second > tmp2.second ) ) ) )
76 |         return 1 ;
77 |     END IF;
78 |     return 0 ;
79 |
80 |     END;
81 |     /
82 |
83 |
84 |     prompt "CREATION MyDate"
85 |
86 |     --Contnum ("Abstract")
87 |     -- Les contenus numériques sont les "objets" du site créer entre autre par les
        utilisateurs :
88 |     -- Cette relation permet entre autre de supprimer une partie de l'activité d'un
        utilisateur sur le site suite la suppression de son compte
89 |
90 |     CREATE TABLE CONTNUM(
91 |         IDCONTNUM NUMERIC(6,0),
92 |         ID_USEUR_CONTNUM NUMERIC(6,0) NOT NULL,
93 |         CONSTRAINT PK_CONTNUM PRIMARY KEY (IDCONTNUM),
94 |         CONSTRAINT FK_CONTNUM_USEUR FOREIGN KEY (ID_USEUR_CONTNUM) REFERENCES USEUR
        (IDUSEUR) ON DELETE CASCADE
95 |     );
96 |
97 |     prompt "CREATION CONTNUM"
98 |
99 |
100 |     -- Oeuvre ("Abstract")(Sous classe de Contnum)
101 |     -- Oeuvre est les ensembles d'oeuvres artistiques
102 |
103 |     CREATE TABLE OEUVRE(
104 |         IDOEUVRE NUMERIC(6,0),
105 |         ID_CONTNUM_OEUVRE NUMERIC(6,0) UNIQUE NOT NULL,
106 |         ID_MYDATE_OEUVRE NUMERIC(6,0) ,
107 |         LICENCE VARCHAR(30) CHECK (LICENCE IN ('TousDroitsReserves','
        UtilisationCommerciale','DroitDeModification')),
108 |         LATITUDE FLOAT CHECK (LATITUDE>=-90 AND LATITUDE<= 90) ,
109 |         LONGITUDE FLOAT CHECK (LONGITUDE>=-180 AND LONGITUDE<= 180) ,
110 |         CONSTRAINT PK_OEUVRE PRIMARY KEY (IDOEUVRE),
111 |         CONSTRAINT FK_OEUVRE_CONTNUM FOREIGN KEY (ID_CONTNUM_OEUVRE) REFERENCES
        CONTNUM (IDCONTNUM) ON DELETE CASCADE,
112 |         CONSTRAINT FK_OEUVRE_MYDATE FOREIGN KEY (ID_MYDATE_OEUVRE) REFERENCES MYDATE
        (IDMYDATE)
113 |     );
114 |
115 |
116 |
117 |     CREATE OR REPLACE FUNCTION isAuteurOfOeuvre (id_auteur IN NUMBER , id_oeuvre IN
        NUMBER )
118 |     RETURN NUMBER -- pas boolean car sinon pas comparaison where clause

```



```

119 | IS
120 | tmp NUMBER;
121 | BEGIN
122 | SELECT COUNT(*) INTO tmp FROM CONTNUM , OEUVRE WHERE id_oeuvre = IDOEUVRE AND
      ID_CONTNUM_OEUVRE = IDCONTNUM AND ID_USEUR_CONTNUM = id_auteur;
123 | return tmp ;
124 | END;
125 | /
126 |
127 |
128 |
129 |
130 | prompt "CREATION OEUVRE"
131 |
132 |
133 | -- Appareil
134 | -- Appareil est l'ensemble des appareils photos recens sur la plateforme
135 |
136 | CREATE TABLE APPAREIL(
137 |     IDAPPAREIL NUMERIC(6,0),
138 |     EAN NUMERIC(6,0) UNIQUE ,
139 |     MARQUE VARCHAR(16),
140 |     CONSTRAINT PK_APPAREIL PRIMARY KEY (IDAPPAREIL)
141 |
142 | );
143 |
144 | prompt "CREATION APPAREIL"
145 |
146 |
147 | -- Config
148 | -- Config est l'ensemble des configurations de chaque appareil ( => GROUP BY
      id_appareil )
149 |
150 | CREATE TABLE CONFIG(
151 |     IDCONFIG NUMERIC(6,0),
152 |     OUVFOCAL VARCHAR(26) NOT NULL ,
153 |     TPSEXPO VARCHAR(20) NOT NULL ,
154 |     FLASH NUMBER(1) CHECK ( FLASH=0 OR FLASH=1) NOT NULL ,
155 |     ID_APPAREIL_CONFIG NUMERIC(6,0),-- ?NOT NULL ,
156 |     DISTFOCAL NUMBER(5) NOT NULL ,
157 |     CONSTRAINT PK_CONFIG PRIMARY KEY (IDCONFIG),
158 |     CONSTRAINT FK_CONFIG_APPAREIL FOREIGN KEY (ID_APPAREIL_CONFIG) REFERENCES
      APPAREIL (IDAPPAREIL) ON DELETE CASCADE
159 | );
160 |
161 |
162 |
163 | CREATE OR REPLACE TRIGGER validate_ouvfoctal
164 | BEFORE INSERT OR UPDATE ON CONFIG
165 | FOR EACH ROW
166 | DECLARE
167 |     regex_validation VARCHAR(80) ;
168 | BEGIN
169 |     regex_validation := '^(f|F)\d+((\.|,)\d+)?$';
170 |     IF NOT REGEXP_LIKE (:NEW.OUVFOCAL , regex_validation) THEN
171 |         RAISE_APPLICATION_ERROR(-20500,'L ouverture focale n est pas valide ');
172 |     END IF ;
173 | END;
174 | /
175 |
176 | CREATE OR REPLACE TRIGGER validate_tpsExpo
177 | BEFORE INSERT OR UPDATE ON CONFIG
178 | FOR EACH ROW
179 | DECLARE
180 |     regex_validation VARCHAR(80) ;
181 | BEGIN
182 |     regex_validation := '^d+(\d+)?$';
183 |     IF NOT REGEXP_LIKE (:NEW.TPSEXPO , regex_validation) THEN

```

```

184 |         RAISE_APPLICATION_ERROR(-20500, 'Le temps expo n est pas valide ');
185 |     END IF ;
186 | END;
187 | /
188 |
189 |
190 |
191 | -- Photo (Sous classe d'oeuvre )
192 | -- Photo est l'ensemble des oeuvres d'arts photos
193 |
194 | CREATE TABLE PHOTO(
195 |     IDPHOTO NUMERIC(6,0),
196 |     ID_OEUVRE_PHOTO NUMERIC(6,0) UNIQUE ,
197 |     ID_CONFIG_PHOTO NUMERIC(6,0) ,
198 |     CODE NUMERIC(6,0) NOT NULL UNIQUE ,
199 |     CONSTRAINT PK_PHOTO PRIMARY KEY (IDPHOTO, ID_OEUVRE_PHOTO),
200 |     CONSTRAINT FK_PHOTO_OEUVRE FOREIGN KEY (ID_OEUVRE_PHOTO) REFERENCES OEUVRE (
201 |         IDOEUVRE) ON DELETE CASCADE ,
202 |     CONSTRAINT FK_PHOTO_CONFIG FOREIGN KEY (ID_CONFIG_PHOTO) REFERENCES CONFIG (
203 |         IDCONFIG)
204 | );
205 |
206 | prompt "CREATION PHOTO"
207 |
208 | -- Discussion
209 | -- Une discussion est un ensemble de commentaires
210 | -- Pas forcément associe une publication/post car par exemple un topic sur le
211 | -- site possede egalement une discussion
212 |
213 | CREATE TABLE DISCUS(
214 |     IDDISCUS NUMERIC(6,0),
215 |     CONSTRAINT PK_DISCUS PRIMARY KEY (IDDISCUS)
216 | );
217 |
218 | prompt "CREATION DISCUS"
219 |
220 | -- Post
221 | -- Post est l'ensemble des publications
222 | -- Si l'oeuvre pr sent par le post est supprimer le post n'a plus raison d' tre
223 | -- et donc la discussion associ non plus
224 | -- Si le cr ateur du post cesse d'exister le commentaire aussi (par heritage)
225 |
226 | CREATE TABLE POST (
227 |     IDPOST NUMERIC(6,0),
228 |     ID_USEUR_POST NUMERIC(6,0) NOT NULL ,
229 |     ID_MYDATE_POST NUMERIC(6,0) NOT NULL , -- 2 post diff sur une meme date
230 |     ID_OEUVRE_POST NUMERIC(6,0) NOT NULL , -- 2 post diff sur une meme oeuvre
231 |     prq pas
232 |     ID_DISCUS_POST NUMERIC(6,0) UNIQUE ,
233 |     NBVUES NUMERIC(16) CHECK (NBVUES >= 0 ) ,
234 |     CONSTRAINT PK_POST PRIMARY KEY (IDPOST ),
235 |     CONSTRAINT FK_POST_USEUR FOREIGN KEY (ID_USEUR_POST) REFERENCES USEUR (
236 |         IDUSEUR),
237 |     CONSTRAINT FK_POST_MYDATE FOREIGN KEY (ID_MYDATE_POST) REFERENCES MYDATE (
238 |         IDMYDATE),
239 |     CONSTRAINT FK_POST_OEUVRE FOREIGN KEY (ID_OEUVRE_POST) REFERENCES OEUVRE (
240 |         IDOEUVRE) ON DELETE CASCADE,
241 |     CONSTRAINT FK_POST_DISCUS FOREIGN KEY (ID_DISCUS_POST) REFERENCES DISCUS (
242 |         IDDISCUS)
243 | );
244 |
245 | CREATE OR REPLACE FUNCTION isAuteurOfPost (id_auteur IN NUMBER , id_post IN NUMBER
246 | )
247 | RETURN NUMBER -- pas boolean car sinon pas comparaison where clause
248 | IS

```

```

242 | id_oeuvre NUMBER;
243 | tmp NUMBER ;
244 | BEGIN
245 | SELECT ID_OEUVRE_POST INTO id_oeuvre FROM OEUVRE POST WHERE id_post = IDPOST ;
246 | tmp:= isAuteurOfOeuvre(id_auteur,id_oeuvre);
247 | return tmp ;
248 | END;
249 | /
250 |
251 |
252 | CREATE OR REPLACE TRIGGER cascade_discus_post
253 | BEFORE DELETE ON POST
254 | FOR EACH ROW
255 | DECLARE
256 | BEGIN
257 |     DELETE FROM DISCUS WHERE IDDISCUS = :OLD.ID_DISCUS_POST ;
258 | END;
259 | /
260 |
261 | CREATE OR REPLACE TRIGGER validate_auteur
262 | BEFORE INSERT OR UPDATE ON POST
263 | FOR EACH ROW
264 | DECLARE
265 | BEGIN
266 |     IF isAuteurOfOeuvre(:NEW.ID_USEUR_POST,:NEW.ID_OEUVRE_POST) = 0 THEN
267 |         RAISE_APPLICATION_ERROR(-20500,'L oeuvre poster doit etre une oeuvre de
268 | l auteur du post ');
269 | END IF ;
270 | END;
271 | /
272 |
273 | CREATE OR REPLACE TRIGGER validate_date_post
274 | BEFORE INSERT OR UPDATE ON POST
275 | FOR EACH ROW
276 | DECLARE
277 | id_date_oeuvre NUMBER;
278 | BEGIN
279 |     SELECT ID_MYDATE_OEUVRE INTO id_date_oeuvre FROM OEUVRE WHERE IDOEUVRE = :
280 | NEW.ID_OEUVRE_POST ;
281 |
282 |     IF isGreaterThenDate (id_date_oeuvre , :NEW.ID_MYDATE_POST ) = 1 THEN --
283 | si la date de l'oeuvre est plus recente que le post qui lui ai consacr
284 | RAISE_APPLICATION_ERROR(-20500,'L oeuvre poster doit anterieure a l
285 | oeuvre ');
286 | END IF ;
287 | END;
288 | /
289 |
290 | prompt "CREATION POST"
291 |
292 | -- Commentaire (Sous classe de Contnum)
293 | -- Commentaire est l'ensemble des commentaires/message publique de chaque discussion
294 | ( => GROUP BY id_discus )
295 | -- Il peut cibler un post comme un autre contenu num rique par exemple un topic
296 | -- Si le contenu cibl ou la discussion cesse d'exister le commentaire aussi
297 | -- Si le cr ateur du commentaire cesse d'exister le commentaire aussi (par heritage
298 | )
299 |
300 | CREATE TABLE COMMENTE(
301 |     IDCOMMENTE NUMERIC(6,0),
302 |     ID_CONTNUM_COMMENTE NUMERIC(6,0) UNIQUE NOT NULL,
303 |     ID_DISCUS_COMMENTE NUMERIC(6,0) NOT NULL,
304 |     ID_MYDATE_COMMENTE NUMERIC(6,0) ,
305 |     CONTENT VARCHAR(256), --voir varchar2 pour plus grand

```

```

304 |         CONSTRAINT PK_COMMENTE PRIMARY KEY (IDCOMMENTE),
305 |         CONSTRAINT FK_COMMENTE_DISCUS FOREIGN KEY (ID_DISCUS_COMMENTE) REFERENCES
DISCUS (IDDISCUS) ON DELETE CASCADE,
306 |         CONSTRAINT FK_COMMENTE_CONTNUM FOREIGN KEY (ID_CONTNUM_COMMENTE) REFERENCES
CONTNUM (IDCONTNUM) ON DELETE CASCADE,
307 |         CONSTRAINT FK_COMMENTE_MYDATE FOREIGN KEY (ID_MYDATE_COMMENTE) REFERENCES
MYDATE (IDMYDATE)
308 | );
309 |
310 |
311 |
312 | CREATE OR REPLACE TRIGGER validate_disjoint_oeuvre
313 | BEFORE INSERT OR UPDATE ON OEUVRE
314 | FOR EACH ROW
315 | DECLARE
316 | BEGIN
317 |
318 |     IF isIN ('COMMENTE', 'ID_CONTNUM_COMMENTE' , :NEW.ID_CONTNUM_OEUVRE )> 0
THEN
319 |         RAISE_APPLICATION_ERROR(-20500,'Ce contenu est deja un commentaire ');
320 |     END IF ;
321 | END;
322 | /
323 |
324 |
325 | CREATE OR REPLACE TRIGGER validate_disjoint_commente
326 | BEFORE INSERT OR UPDATE ON COMMENTE
327 | FOR EACH ROW
328 | DECLARE
329 | BEGIN
330 |
331 |     IF isIN ('OEUVRE', 'ID_CONTNUM_OEUVRE' , :NEW.ID_CONTNUM_COMMENTE ) > 0
THEN
332 |         --si le contnum du commentaire est deja associ a une oeuvre
RAISE_APPLICATION_ERROR(-20500,'Ce contenu est deja une Oeuvre ');
333 |     END IF ;
334 | END;
335 | /
336 |
337 | CREATE OR REPLACE TRIGGER validate_date_comments
338 | BEFORE INSERT OR UPDATE ON COMMENTE
339 | FOR EACH ROW
340 | DECLARE
341 |     id_date_post POST.ID_MYDATE_POST%TYPE;
342 | BEGIN
343 |     id_date_post :=0;
344 |     SELECT ID_MYDATE_POST INTO id_date_post FROM POST WHERE ID_DISCUS_POST = :
NEW.ID_DISCUS_COMMENTE;
345 |
346 |     IF isGreaterThanDate (id_date_post ,:NEW.ID_MYDATE_COMMENTE ) = 1 THEN
--si la date du post est plus recente que le commentaire qui lui ai consacr
347 |         RAISE_APPLICATION_ERROR(-20500,'Le commentaire doit etre anterieure a l
oeuvre poster ');
348 |     END IF ;
349 |
350 | END;
351 | /
352 |
353 |
354 | prompt "CREATION COMMENTE"
355 |
356 |
357 | --Tag
358 | -- Tag est l'ensemble des qualificatifs de chaque post ( objet ?) du site ( =>
GROUP BY id_post )
359 | -- Si la cible du tagg cesse d'exister le tagg aussi en revanche le tagg peut
survivre a son utilisateur ( pratique pour conserv le referencement )
360 |
361 | CREATE TABLE TAGEU (

```

```

362 |         IDTAGEU NUMERIC(6,0),
363 |         ID_USEUR_TAGEU NUMERIC(6,0) DEFAULT NULL, --si NULL => robot site
364 |         ID_POST_TAGEU NUMERIC(6,0) NOT NULL ,
365 |         TAGVALUE VARCHAR(16) NOT NULL,
366 |         CONSTRAINT PK_TAGEU PRIMARY KEY (IDTAGEU),
367 |         CONSTRAINT FK_TAGEU_USEUR FOREIGN KEY (ID_USEUR_TAGEU) REFERENCES USEUR (
IDUSEUR),
368 |         CONSTRAINT FK_TAGEU_POST FOREIGN KEY (ID_POST_TAGEU) REFERENCES POST (IDPOST
) ON DELETE CASCADE
369 | );
370 |
371 | prompt "CREATION TAGEU"
372 |
373 |
374 | --Like
375 | -- Like est l'ensemble des likes de chaque post ( objet ?) du site (=> GROUP BY
id_post ? id_contnum)
376 | -- Il est une appreciation de l'utilisateur pour une oeuvre ainsi si l'utilisateur
disparait de la plateforme ou l'oeuvre lik alors le like disparait
377 | --(Contrairement au tagg il n'a plus de valeur une fois l'utilisateur partie du site
)
378 |
379 | CREATE TABLE LIKEU (
380 |         ID_USEUR_LIKEU NUMERIC(6,0) ,
381 |         ID_MYDATE_LIKEU NUMERIC(6,0) ,
382 |         ID_POST_LIKEU NUMERIC(6,0) NOT NULL,
383 |         CONSTRAINT PK_LIKEU PRIMARY KEY (ID_USEUR_LIKEU, ID_POST_LIKEU),
384 |         CONSTRAINT FK_LIKEU_POST FOREIGN KEY (ID_POST_LIKEU) REFERENCES POST (IDPOST
) ON DELETE CASCADE ,
385 |         CONSTRAINT FK_LIKEU_MYDATE FOREIGN KEY (ID_MYDATE_LIKEU) REFERENCES MYDATE (
IDMYDATE)
386 | );
387 |
388 |
389 | prompt "CREATION LIKEU"
390 |
391 |
392 | --Follow
393 | -- Follow est l'ensemble des follows de chaque utilisateur x 2 (=> GROUP BY id_user1
| id_user2 )
394 | -- Tout comme like il depend de ces 2 clefs etrangeres si l'une des entit s
disparait alors il perd son sens et disparait a son tour
395 | -- On dit que utilisateur numero 2 est suivie par l'utilisateur numero 1
396 |
397 | CREATE TABLE FOLLOW (
398 |         ID_USEUR_FOLLOW NUMERIC(6,0) NOT NULL, -- n1
399 |         ID_USEUR_FOLLOW2 NUMERIC(6,0) NOT NULL, -- n2
400 |         ID_MYDATE_FOLLOW NUMERIC(6,0) ,
401 |         CONSTRAINT PK_FOLLOW PRIMARY KEY (ID_USEUR_FOLLOW, ID_USEUR_FOLLOW2),
402 |         CONSTRAINT FK_FOLLOW_USEUR FOREIGN KEY (ID_USEUR_FOLLOW) REFERENCES USEUR (
IDUSEUR) ON DELETE CASCADE,
403 |         CONSTRAINT FK_FOLLOW_USEUR2 FOREIGN KEY (ID_USEUR_FOLLOW2) REFERENCES USEUR
(IDUSEUR) ON DELETE CASCADE ,
404 |         CONSTRAINT FK_FOLLOW_MYDATE FOREIGN KEY (ID_MYDATE_FOLLOW) REFERENCES MYDATE
(IDMYDATE)
405 | );
406 |
407 |
408 | CREATE OR REPLACE TRIGGER validate_follow
409 | BEFORE INSERT OR UPDATE ON FOLLOW
410 | FOR EACH ROW
411 | DECLARE
412 | BEGIN
413 |
414 |         IF :NEW.ID_USEUR_FOLLOW = :NEW.ID_USEUR_FOLLOW2 THEN --si la date du post
est plus recente que le commentaire qui lui ai consacrer
415 |             RAISE_APPLICATION_ERROR(-20500, 'On ne peut pas se follower ');
416 |         END IF ;

```

```

417 | END;
418 | /
419 |
420 |
421 | prompt "CREATION FOLLOW"
422 |
423 |
424 | --Collection ("Abstract")
425 | -- Collection est l'ensemble des collections du site a savoir soit des albums soit
      des galleries
426 | -- Si le createur de la collection disparaît la collection disparaît aussi
427 | --( nous aurions pu envisager qu'une collection ne puisse être vide en intégrant
      id_post_[album|galerie] NOT NULL a ses 2 classes filles )
428 |
429 | CREATE TABLE COLLECTION (
430 |     IDCOLLECTION NUMERIC(6,0),
431 |     ID_USEUR_COLLECTION NUMERIC(6,0) NOT NULL ,
432 |     TITRE VARCHAR(16) ,
433 |     DESCRIPTION VARCHAR(126) ,
434 |     ID_MYDATE_COLLECTION NUMERIC(6,0),
435 |     CONSTRAINT PK_COLLECTION PRIMARY KEY (IDCOLLECTION),
436 |     CONSTRAINT FK_COLLECTION_USEUR FOREIGN KEY (ID_USEUR_COLLECTION) REFERENCES
      USEUR (IDUSEUR) ON DELETE CASCADE,
437 |     CONSTRAINT FK_COLLECTION_MYDATE FOREIGN KEY (ID_MYDATE_COLLECTION)
      REFERENCES MYDATE (IDMYDATE)
438 | );
439 |
440 |
441 | prompt "CREATION COLLECTION"
442 |
443 |
444 | --Galerie (sous classe de Collection)
445 | -- Galerie est l'ensemble des posts de chaque galerie ( => GROUP BY
      id_collection_galerie )
446 | -- Si le post cesse d'exister alors la relation galerie post cesse d'exister
      également ( id_collection_galerie perd un post )
447 | CREATE TABLE GALLERIE (
448 |     IDGALLERIE NUMERIC(6,0),
449 |     ID_COLLECTION_GALLERIE NUMERIC(6,0) NOT NULL ,
450 |     ID_POST_GALLERIE NUMERIC(6,0) ,
451 |     CONSTRAINT PK_GALLERIE PRIMARY KEY (IDGALLERIE, ID_POST_GALLERIE),
452 |     CONSTRAINT FK_GALLERIE_COLLECTION FOREIGN KEY (ID_COLLECTION_GALLERIE)
      REFERENCES COLLECTION(IDCOLLECTION) ON DELETE CASCADE ,
453 |     CONSTRAINT FK_GALLERIE_POST FOREIGN KEY (ID_POST_GALLERIE) REFERENCES POST(
      IDPOST) ON DELETE CASCADE
454 | );
455 |
456 | prompt "CREATION GALLERIE"
457 |
458 |
459 | -- Album (sous classe de Collection)
460 | -- Album est l'ensemble des posts de chaque album ( => GROUP BY id_collection_album
      )
461 | -- Si le post cesse d'exister alors la relation album post cesse d'exister
      également ( id_collection_album perd un post )
462 | -- Tous les post de l'album appartiennent au détenteur de la collection ( cf super.
      id_uteur_collection )
463 | CREATE TABLE ALBUM (
464 |     IDALBUM NUMERIC(6,0),
465 |     ID_COLLECTION_ALBUM NUMERIC(6,0) NOT NULL ,
466 |     ID_POST_ALBUM NUMERIC(6,0),
467 |     CONSTRAINT PK_ALBUM PRIMARY KEY (IDALBUM, ID_POST_ALBUM ),
468 |     CONSTRAINT FK_ALBUM_COLLECTION FOREIGN KEY (ID_COLLECTION_ALBUM) REFERENCES
      COLLECTION(IDCOLLECTION) ON DELETE CASCADE,
469 |     CONSTRAINT FK_ALBUM_POST FOREIGN KEY (ID_POST_ALBUM) REFERENCES POST(IDPOST)
      ON DELETE CASCADE
470 | );
471 |

```

```

472 |
473 | CREATE OR REPLACE TRIGGER validate_gallerie
474 | BEFORE INSERT OR UPDATE ON GALLERIE
475 | FOR EACH ROW
476 | DECLARE
477 | BEGIN
478 |
479 |     IF isIN ('ALBUM', 'ID_COLLECTION_ALBUM' , :NEW.ID_COLLECTION_GALLERIE ) > 0
      THEN --si la date du post est plus recente que le commentaire qui lui ai
consacr
480 |         RAISE_APPLICATION_ERROR(-20500,'Cette collection est deja un album ');
481 |     END IF ;
482 | END;
483 | /
484 |
485 | CREATE OR REPLACE TRIGGER validate_album
486 | BEFORE INSERT OR UPDATE ON ALBUM
487 | FOR EACH ROW
488 | DECLARE
489 | BEGIN
490 |
491 |     IF isIN ('GALLERIE', 'ID_COLLECTION_GALLERIE' , :NEW.ID_COLLECTION_ALBUM ) >
      0 THEN --si la date du post est plus recente que le commentaire qui lui ai
consacr
492 |         RAISE_APPLICATION_ERROR(-20500,'Cette collection est deja une gallerie '
      );
493 |     END IF ;
494 | END;
495 | /
496 |
497 |
498 | CREATE OR REPLACE TRIGGER validate_auteur_album
499 | BEFORE INSERT OR UPDATE ON ALBUM
500 | FOR EACH ROW
501 | DECLARE
502 | id_useur_album NUMBER;
503 | BEGIN
504 |     SELECT ID_USEUR_COLLECTION INTO id_useur_album FROM COLLECTION WHERE :NEW.
      ID_COLLECTION_ALBUM =IDCOLLECTION;
505 |     IF isAuteurOfPost(id_useur_album,:NEW.ID_POST_ALBUM) = 0 THEN
506 |         RAISE_APPLICATION_ERROR(-20500,'L oeuvre poster doit etre une oeuvre de
      l auteur du post ');
507 |     END IF ;
508 | END;
509 | /
510 |
511 |
512 | prompt "CREATION ALBUM"

```

## Partie 2 : Peuplement de la base de donnée

```
01 |
02 |
03 |
04 |
05 | -- POPULATE :
06 | -- ATTENTION ne pas oublier d'egalementment importer la table villes_france_free :
07 |
08 | INSERT INTO USEUR (IDUSEUR ,PSEUDO,NOM,PRENOM,MAIL ) VALUES (1,'PierreD2','Dupont','
    Pierre','pierre.dupont@gmail.com');
09 | INSERT INTO USEUR (IDUSEUR ,PSEUDO,NOM,PRENOM,MAIL ) VALUES (2,'PatrickD2','Dupont','
    Patrick','patrick.dupont@gmail.com');
10 | INSERT INTO USEUR (IDUSEUR ,PSEUDO,NOM,PRENOM,MAIL ) VALUES (3,'PaulD2','Dupont','
    Paul','paul.dupont@gmail.com');
11 | INSERT INTO USEUR (IDUSEUR ,PSEUDO,NOM,PRENOM,MAIL ) VALUES (4,'PaqueretteD2','
    Dupont','Paquerette','Paquerette.dupont@gmail.com');
12 | INSERT INTO USEUR (IDUSEUR ,PSEUDO,NOM,PRENOM,MAIL ) VALUES (5,'PascaleD2','Dupont','
    Pascale','pascale.dupont@gmail.com');
13 | INSERT INTO USEUR (IDUSEUR ,PSEUDO,NOM,PRENOM,MAIL ) VALUES (6,'PabloD2','Dupont','
    Pablo','pablo.dupont@gmail.com');
14 | INSERT INTO USEUR (IDUSEUR ,PSEUDO,NOM,PRENOM,MAIL ) VALUES (7,'PhilippeD2','Dupont'
    , 'Philippe','philippe.dupont@gmail.com');
15 | INSERT INTO USEUR (IDUSEUR ,PSEUDO,NOM,PRENOM,MAIL ) VALUES (8,'PatriciaD2','Dupont'
    , 'Patricia','patricia.dupont@gmail.com');
16 |
17 |
18 |
19 | INSERT INTO MYDATE (IDMYDATE ,DATEMYDATE ) VALUES (1,TO_DATE('2022/09/15 17:50:11',
    'yyyy/mm/dd hh24:mi:ss'));
20 | INSERT INTO MYDATE (IDMYDATE ,DATEMYDATE ) VALUES (2,TO_DATE('2022/09/15 17:50:11',
    'yyyy/mm/dd hh24:mi:ss'));
21 | INSERT INTO MYDATE (IDMYDATE ,DATEMYDATE ) VALUES (3,TO_DATE('2022/09/15 17:50:11',
    'yyyy/mm/dd hh24:mi:ss'));
22 | INSERT INTO MYDATE (IDMYDATE ,DATEMYDATE ) VALUES (4,TO_DATE('2022/09/15 17:50:11',
    'yyyy/mm/dd hh24:mi:ss'));
23 | INSERT INTO MYDATE (IDMYDATE ,DATEMYDATE ) VALUES (5,TO_DATE('2022/09/15 17:50:11',
    'yyyy/mm/dd hh24:mi:ss'));
24 | INSERT INTO MYDATE (IDMYDATE ,DATEMYDATE ) VALUES (6,TO_DATE('2022/09/15 17:50:11',
    'yyyy/mm/dd hh24:mi:ss'));
25 | INSERT INTO MYDATE (IDMYDATE ,DATEMYDATE ) VALUES (7,TO_DATE('2022/09/15 17:50:11',
    'yyyy/mm/dd hh24:mi:ss'));
26 | INSERT INTO MYDATE (IDMYDATE ,DATEMYDATE ) VALUES (8,TO_DATE('2022/09/15 17:50:11',
    'yyyy/mm/dd hh24:mi:ss'));
27 | INSERT INTO MYDATE (IDMYDATE ,DATEMYDATE ) VALUES (9,TO_DATE('2022/09/15 17:50:11',
    'yyyy/mm/dd hh24:mi:ss'));
28 | INSERT INTO MYDATE (IDMYDATE ,DATEMYDATE ) VALUES (10,TO_DATE('2022/09/15 17:50:11'
    , 'yyyy/mm/dd hh24:mi:ss'));
29 | INSERT INTO MYDATE (IDMYDATE ,DATEMYDATE ) VALUES (11,TO_DATE('2022/09/15 17:50:11'
    , 'yyyy/mm/dd hh24:mi:ss'));
30 | INSERT INTO MYDATE (IDMYDATE ,DATEMYDATE ) VALUES (12,TO_DATE('2022/09/15 17:50:11'
    , 'yyyy/mm/dd hh24:mi:ss'));
31 | INSERT INTO MYDATE (IDMYDATE ,DATEMYDATE ) VALUES (13,TO_DATE('2022/09/15 17:50:11'
    , 'yyyy/mm/dd hh24:mi:ss'));
32 |
33 |
34 | INSERT INTO CONTNUM ( IDCONTNUM ,ID_USEUR_CONTNUM) VALUES (1,1);
35 | INSERT INTO CONTNUM ( IDCONTNUM ,ID_USEUR_CONTNUM) VALUES (2,1);
36 | INSERT INTO CONTNUM ( IDCONTNUM ,ID_USEUR_CONTNUM) VALUES (3,2);
37 | INSERT INTO CONTNUM ( IDCONTNUM ,ID_USEUR_CONTNUM) VALUES (4,3);
38 | INSERT INTO CONTNUM ( IDCONTNUM ,ID_USEUR_CONTNUM) VALUES (5,3);
39 | INSERT INTO CONTNUM ( IDCONTNUM ,ID_USEUR_CONTNUM) VALUES (6,3);
40 | INSERT INTO CONTNUM ( IDCONTNUM ,ID_USEUR_CONTNUM) VALUES (7,4);
41 | INSERT INTO CONTNUM ( IDCONTNUM ,ID_USEUR_CONTNUM) VALUES (8,4);
42 | INSERT INTO CONTNUM ( IDCONTNUM ,ID_USEUR_CONTNUM) VALUES (9,5);
43 | INSERT INTO CONTNUM ( IDCONTNUM ,ID_USEUR_CONTNUM) VALUES (10,4);
44 | INSERT INTO CONTNUM ( IDCONTNUM ,ID_USEUR_CONTNUM) VALUES (11,6);
```



```

45 | INSERT INTO CONTNUM ( IDCONTNUM ,ID_USEUR_CONTNUM) VALUES (12,7);
46 | INSERT INTO CONTNUM ( IDCONTNUM ,ID_USEUR_CONTNUM) VALUES (13,3);
47 | INSERT INTO CONTNUM ( IDCONTNUM ,ID_USEUR_CONTNUM) VALUES (14,3);
48 | INSERT INTO CONTNUM ( IDCONTNUM ,ID_USEUR_CONTNUM) VALUES (15,4);
49 | INSERT INTO CONTNUM ( IDCONTNUM ,ID_USEUR_CONTNUM) VALUES (16,4);
50 | INSERT INTO CONTNUM ( IDCONTNUM ,ID_USEUR_CONTNUM) VALUES (17,5);
51 | INSERT INTO CONTNUM ( IDCONTNUM ,ID_USEUR_CONTNUM) VALUES (18,4);
52 | INSERT INTO CONTNUM ( IDCONTNUM ,ID_USEUR_CONTNUM) VALUES (19,6);
53 | INSERT INTO CONTNUM ( IDCONTNUM ,ID_USEUR_CONTNUM) VALUES (20,7);
54 |
55 |
56 |
57 |
58 | INSERT INTO OEUVRE( IDOEUVRE ,ID_CONTNUM_OEUVRE, ID_MYDATE_OEUVRE , LICENCE ,
    LATITUDE ,LONGITUDE) VALUES (1,1,1,'TousDroitsReserves',45.7667964,4.8358581);
59 | INSERT INTO OEUVRE( IDOEUVRE ,ID_CONTNUM_OEUVRE, ID_MYDATE_OEUVRE , LICENCE ,
    LATITUDE ,LONGITUDE) VALUES (2,2,2,'TousDroitsReserves',45.245783, 2.171033);
60 | INSERT INTO OEUVRE( IDOEUVRE ,ID_CONTNUM_OEUVRE, ID_MYDATE_OEUVRE , LICENCE ,
    LATITUDE ,LONGITUDE) VALUES (3,4,4,'TousDroitsReserves',43.451651, 0.190064);
61 | INSERT INTO OEUVRE( IDOEUVRE ,ID_CONTNUM_OEUVRE, ID_MYDATE_OEUVRE , LICENCE ,
    LATITUDE ,LONGITUDE) VALUES (4,3,3,'TousDroitsReserves',43.608747, 3.889194);
62 | INSERT INTO OEUVRE( IDOEUVRE ,ID_CONTNUM_OEUVRE, ID_MYDATE_OEUVRE , LICENCE ,
    LATITUDE ,LONGITUDE) VALUES (5,9,9,'TousDroitsReserves',44.886517, 0.887887);
63 | INSERT INTO OEUVRE( IDOEUVRE ,ID_CONTNUM_OEUVRE, ID_MYDATE_OEUVRE , LICENCE ,
    LATITUDE ,LONGITUDE) VALUES (6,11,11,'TousDroitsReserves',44.097743, -1.306180);
64 | INSERT INTO OEUVRE( IDOEUVRE ,ID_CONTNUM_OEUVRE, ID_MYDATE_OEUVRE , LICENCE ,
    LATITUDE ,LONGITUDE) VALUES (7,10,10,'TousDroitsReserves',46.728759, -0.497960);
65 | INSERT INTO OEUVRE( IDOEUVRE ,ID_CONTNUM_OEUVRE, ID_MYDATE_OEUVRE , LICENCE ,
    LATITUDE ,LONGITUDE) VALUES (8,8,8,'TousDroitsReserves',46.174745, 0.567798);
66 | INSERT INTO OEUVRE( IDOEUVRE ,ID_CONTNUM_OEUVRE, ID_MYDATE_OEUVRE , LICENCE ,
    LATITUDE ,LONGITUDE) VALUES (9,7,7,'TousDroitsReserves',43.630861, 3.855133);
67 | INSERT INTO OEUVRE( IDOEUVRE ,ID_CONTNUM_OEUVRE, ID_MYDATE_OEUVRE , LICENCE ,
    LATITUDE ,LONGITUDE) VALUES (10,6,6,'TousDroitsReserves',48.535754, -0.449022);
68 | INSERT INTO OEUVRE( IDOEUVRE ,ID_CONTNUM_OEUVRE, ID_MYDATE_OEUVRE , LICENCE ,
    LATITUDE ,LONGITUDE) VALUES (11,13,13,'TousDroitsReserves',50.330492, 2.369838);
69 | INSERT INTO OEUVRE( IDOEUVRE ,ID_CONTNUM_OEUVRE, ID_MYDATE_OEUVRE , LICENCE ,
    LATITUDE ,LONGITUDE) VALUES (12,12,12,'TousDroitsReserves',43.571966, 3.861466);
70 | INSERT INTO OEUVRE( IDOEUVRE ,ID_CONTNUM_OEUVRE, ID_MYDATE_OEUVRE , LICENCE ,
    LATITUDE ,LONGITUDE) VALUES (13,5,5,'TousDroitsReserves',43.642856, 3.900339);
71 |
72 |
73 | INSERT INTO DISCUS (IDDISCUS ) VALUES (1);
74 | INSERT INTO DISCUS (IDDISCUS ) VALUES (2);
75 | INSERT INTO DISCUS (IDDISCUS ) VALUES (3);
76 | INSERT INTO DISCUS (IDDISCUS ) VALUES (4);
77 | INSERT INTO DISCUS (IDDISCUS ) VALUES (5);
78 | INSERT INTO DISCUS (IDDISCUS ) VALUES (6);
79 | INSERT INTO DISCUS (IDDISCUS ) VALUES (7);
80 | INSERT INTO DISCUS (IDDISCUS ) VALUES (8);
81 | INSERT INTO DISCUS (IDDISCUS ) VALUES (9);
82 | INSERT INTO DISCUS (IDDISCUS ) VALUES (10);
83 | INSERT INTO DISCUS (IDDISCUS ) VALUES (11);
84 | INSERT INTO DISCUS (IDDISCUS ) VALUES (12);
85 |
86 |
87 | INSERT INTO APPAREIL(IDAPPAREIL ,NOM ,EAN ,MARQUE ) VALUES (1,'modele-X',400071,'
    Nikon');
88 | INSERT INTO APPAREIL(IDAPPAREIL ,NOM ,EAN ,MARQUE ) VALUES (2,'modele-X',400071,'
    Nikon');
89 | INSERT INTO APPAREIL(IDAPPAREIL ,NOM ,EAN ,MARQUE ) VALUES (3,'modele-X',400071,'
    Nikon');
90 | INSERT INTO APPAREIL(IDAPPAREIL ,NOM ,EAN ,MARQUE ) VALUES (11,'modele-Y',400023,'
    Samsung');
91 | INSERT INTO APPAREIL(IDAPPAREIL ,NOM ,EAN ,MARQUE ) VALUES (12,'modele-Y',400023,'
    Samsung');
92 | INSERT INTO APPAREIL(IDAPPAREIL ,NOM ,EAN ,MARQUE ) VALUES (13,'modele-Z',400024,'
    Sony');
93 | INSERT INTO APPAREIL(IDAPPAREIL ,NOM ,EAN ,MARQUE ) VALUES (10,'modele-A',400032,'

```

```

        Canon');
94 | INSERT INTO APPAREIL(IDAPPAREIL ,NOM ,EAN ,MARQUE ) VALUES (15,'modele-B',400015,'
    Samsung');
95 | INSERT INTO APPAREIL(IDAPPAREIL ,NOM ,EAN ,MARQUE ) VALUES (7,'modele-X',400071,'
    Nikon');
96 | INSERT INTO APPAREIL(IDAPPAREIL ,NOM ,EAN ,MARQUE ) VALUES (4,'modele-X',400071,'
    Nikon');
97 | INSERT INTO APPAREIL(IDAPPAREIL ,NOM ,EAN ,MARQUE ) VALUES (5,'modele-X',400071,'
    Nikon');
98 | INSERT INTO APPAREIL(IDAPPAREIL ,NOM ,EAN ,MARQUE ) VALUES (14,'modele-Z',400024,'
    Sony');
99 | INSERT INTO APPAREIL(IDAPPAREIL ,NOM ,EAN ,MARQUE ) VALUES (8,'modele-A',400032,'
    Canon');
100 | INSERT INTO APPAREIL(IDAPPAREIL ,NOM ,EAN ,MARQUE ) VALUES (9,'modele-A',400032,'
    Canon');
101 | INSERT INTO APPAREIL(IDAPPAREIL ,NOM ,EAN ,MARQUE ) VALUES (6,'modele-X',400071,'
    Nikon');
102 |
103 | INSERT INTO CONFIG(IDCONFIG ,OUVFOCAL ,TPSEXPO ,FLASH ,ID_APPAREIL_CONFIG ,DISTFOCAL
    ) VALUES (1,'F/1','1/1000',1,1,28);
104 | INSERT INTO CONFIG(IDCONFIG ,OUVFOCAL ,TPSEXPO ,FLASH ,ID_APPAREIL_CONFIG ,
    DISTFOCAL ) VALUES(2,'f/4','1/500',1,4,35);
105 | INSERT INTO CONFIG(IDCONFIG ,OUVFOCAL ,TPSEXPO ,FLASH ,ID_APPAREIL_CONFIG ,
    DISTFOCAL ) VALUES(4,'f/8','1/125',0,2,35);
106 | INSERT INTO CONFIG(IDCONFIG ,OUVFOCAL ,TPSEXPO ,FLASH ,ID_APPAREIL_CONFIG ,DISTFOCAL
    ) VALUES(6,'f/16','1/30',0,4,28);
107 | INSERT INTO CONFIG(IDCONFIG ,OUVFOCAL ,TPSEXPO ,FLASH ,ID_APPAREIL_CONFIG ,DISTFOCAL
    ) VALUES(7,'f/22','1',1,5,200);
108 | INSERT INTO CONFIG(IDCONFIG ,OUVFOCAL ,TPSEXPO ,FLASH ,ID_APPAREIL_CONFIG ,DISTFOCAL
    ) VALUES(8,'f/32','1/2',1,6,200);
109 | INSERT INTO CONFIG(IDCONFIG ,OUVFOCAL ,TPSEXPO ,FLASH ,ID_APPAREIL_CONFIG ,DISTFOCAL
    ) VALUES(9,'f/2.8','1/4',0,7,35);
110 |
111 | -- ICI :
112 | INSERT INTO CONFIG(IDCONFIG ,OUVFOCAL ,TPSEXPO ,FLASH ,ID_APPAREIL_CONFIG ,
    DISTFOCAL ) VALUES(3,'f/5.6','1/250',1,3,200);
113 | INSERT INTO CONFIG(IDCONFIG ,OUVFOCAL ,TPSEXPO ,FLASH ,ID_APPAREIL_CONFIG ,DISTFOCAL
    ) VALUES(5,'F/11','1/60',1,3,28);
114 | -- meme appareil mais 2 config differente
115 |
116 |
117 |
118 | INSERT INTO PHOTO (IDPHOTO ,ID_OEUVRE_PHOTO ,ID_CONFIG_PHOTO ,CODE) VALUES (1,2,4,9);
119 | INSERT INTO PHOTO (IDPHOTO ,ID_OEUVRE_PHOTO ,ID_CONFIG_PHOTO ,CODE) VALUES (2,6,7,10);
120 | INSERT INTO PHOTO (IDPHOTO ,ID_OEUVRE_PHOTO ,ID_CONFIG_PHOTO ,CODE) VALUES (3,4,9,11);
121 | INSERT INTO PHOTO (IDPHOTO ,ID_OEUVRE_PHOTO ,ID_CONFIG_PHOTO ,CODE) VALUES (4,5,4,12);
122 |
123 | --ICI
124 | INSERT INTO PHOTO (IDPHOTO ,ID_OEUVRE_PHOTO ,ID_CONFIG_PHOTO ,CODE) VALUES (5,7,5,13);
125 | INSERT INTO PHOTO (IDPHOTO ,ID_OEUVRE_PHOTO ,ID_CONFIG_PHOTO ,CODE) VALUES (6,3,3,14);
126 | -- 2 photos differentes
127 |
128 | INSERT INTO PHOTO (IDPHOTO ,ID_OEUVRE_PHOTO ,ID_CONFIG_PHOTO ,CODE) VALUES (7,8,1,15);
129 | INSERT INTO PHOTO (IDPHOTO ,ID_OEUVRE_PHOTO ,ID_CONFIG_PHOTO ,CODE) VALUES (8,9,6,16);
130 | INSERT INTO PHOTO (IDPHOTO ,ID_OEUVRE_PHOTO ,ID_CONFIG_PHOTO ,CODE) VALUES (9,1,8,17);
131 |
132 | --ICI
133 | INSERT INTO POST (IDPOST ,ID_USEUR_POST ,ID_MYDATE_POST ,ID_OEUVRE_POST ,
    ID_DISCUS_POST ,NBVUES ) VALUES (1,1,7,2,2,120);
134 | INSERT INTO POST (IDPOST ,ID_USEUR_POST ,ID_MYDATE_POST ,ID_OEUVRE_POST ,
    ID_DISCUS_POST ,NBVUES ) VALUES (2,1,9,1,1,130);
135 | -- meme Utilisateur
136 |
137 | INSERT INTO POST (IDPOST ,ID_USEUR_POST ,ID_MYDATE_POST ,ID_OEUVRE_POST ,
    ID_DISCUS_POST ,NBVUES ) VALUES (3,4,12,9,7,0);
138 | INSERT INTO POST (IDPOST ,ID_USEUR_POST ,ID_MYDATE_POST ,ID_OEUVRE_POST ,
    ID_DISCUS_POST ,NBVUES ) VALUES (4,2,1,4,3,50);
139 |

```

```

140 |
141 | INSERT INTO POST (IDPOST ,ID_USEUR_POST ,ID_MYDATE_POST ,ID_OEUVRE_POST ,
    | ID_DISCUS_POST ,NBVUES ) VALUES (5,4,5,9,4,80);
142 | INSERT INTO POST (IDPOST ,ID_USEUR_POST ,ID_MYDATE_POST ,ID_OEUVRE_POST ,
    | ID_DISCUS_POST ,NBVUES ) VALUES (6,5,2,5,9,120);
143 | INSERT INTO POST (IDPOST ,ID_USEUR_POST ,ID_MYDATE_POST ,ID_OEUVRE_POST ,
    | ID_DISCUS_POST ,NBVUES ) VALUES (7,7,10,12,12,140);
144 | INSERT INTO POST (IDPOST ,ID_USEUR_POST ,ID_MYDATE_POST ,ID_OEUVRE_POST ,
    | ID_DISCUS_POST ,NBVUES ) VALUES (8,4,3,8,8,90);
145 | INSERT INTO POST (IDPOST ,ID_USEUR_POST ,ID_MYDATE_POST ,ID_OEUVRE_POST ,
    | ID_DISCUS_POST ,NBVUES ) VALUES (9,6,4,6,11,45);
146 |
147 |
148 | INSERT INTO COMMENTE (IDCOMMENTE ,ID_CONTNUM_COMMENTE ,ID_DISCUS_COMMENTE ,
    | ID_MYDATE_COMMENTE ,CONTENT ) VALUES (1,14,2,2,'comment1');
149 | INSERT INTO COMMENTE (IDCOMMENTE ,ID_CONTNUM_COMMENTE ,ID_DISCUS_COMMENTE ,
    | ID_MYDATE_COMMENTE ,CONTENT ) VALUES (2,20,1,1,'comment2');
150 | INSERT INTO COMMENTE (IDCOMMENTE ,ID_CONTNUM_COMMENTE ,ID_DISCUS_COMMENTE ,
    | ID_MYDATE_COMMENTE ,CONTENT ) VALUES (3,17,12,12,'comment3');
151 | INSERT INTO COMMENTE (IDCOMMENTE ,ID_CONTNUM_COMMENTE ,ID_DISCUS_COMMENTE ,
    | ID_MYDATE_COMMENTE ,CONTENT ) VALUES (4,19,11,11,'comment4');
152 | INSERT INTO COMMENTE (IDCOMMENTE ,ID_CONTNUM_COMMENTE ,ID_DISCUS_COMMENTE ,
    | ID_MYDATE_COMMENTE ,CONTENT ) VALUES (5,15,3,3,'comment5');
153 | INSERT INTO COMMENTE (IDCOMMENTE ,ID_CONTNUM_COMMENTE ,ID_DISCUS_COMMENTE ,
    | ID_MYDATE_COMMENTE ,CONTENT ) VALUES (6,16,8,8,'comment6');
154 | INSERT INTO COMMENTE (IDCOMMENTE ,ID_CONTNUM_COMMENTE ,ID_DISCUS_COMMENTE ,
    | ID_MYDATE_COMMENTE ,CONTENT ) VALUES (7,18,1,1,'comment7');
155 |
156 |
157 |
158 |
159 | INSERT INTO TAGEU (IDTAGEU ,ID_USEUR_TAGEU ,ID_POST_TAGEU , TAGVALUE) VALUES (1,4,5,'
    | Sun');
160 | INSERT INTO TAGEU (IDTAGEU ,ID_USEUR_TAGEU ,ID_POST_TAGEU , TAGVALUE) VALUES (2,8,3,'
    | Sea');
161 | INSERT INTO TAGEU (IDTAGEU ,ID_USEUR_TAGEU ,ID_POST_TAGEU , TAGVALUE) VALUES (3,8,3,'
    | Montpellier');
162 | INSERT INTO TAGEU (IDTAGEU ,ID_USEUR_TAGEU ,ID_POST_TAGEU , TAGVALUE) VALUES (4,8,4,'
    | blue');
163 | INSERT INTO TAGEU (IDTAGEU ,ID_USEUR_TAGEU ,ID_POST_TAGEU , TAGVALUE) VALUES (5,2,4,'
    | bird');
164 | INSERT INTO TAGEU (IDTAGEU ,ID_USEUR_TAGEU ,ID_POST_TAGEU , TAGVALUE) VALUES (6,3,9,'
    | blue');
165 | INSERT INTO TAGEU (IDTAGEU ,ID_USEUR_TAGEU ,ID_POST_TAGEU , TAGVALUE) VALUES (7,5,6,'
    | dog');
166 | INSERT INTO TAGEU (IDTAGEU ,ID_USEUR_TAGEU ,ID_POST_TAGEU , TAGVALUE) VALUES (8,1,8,'
    | Sun');
167 | INSERT INTO TAGEU (IDTAGEU ,ID_USEUR_TAGEU ,ID_POST_TAGEU , TAGVALUE) VALUES (9,7,9,'
    | work');
168 |
169 |
170 | INSERT INTO LIKEU (ID_USEUR_LIKEU ,ID_MYDATE_LIKEU ,ID_POST_LIKEU ) VALUES (1,1,1);
171 | INSERT INTO LIKEU (ID_USEUR_LIKEU ,ID_MYDATE_LIKEU ,ID_POST_LIKEU ) VALUES (2,2,4);
172 | INSERT INTO LIKEU (ID_USEUR_LIKEU ,ID_MYDATE_LIKEU ,ID_POST_LIKEU ) VALUES (1,2,4);
173 | INSERT INTO LIKEU (ID_USEUR_LIKEU ,ID_MYDATE_LIKEU ,ID_POST_LIKEU ) VALUES (3,3,5);
174 | INSERT INTO LIKEU (ID_USEUR_LIKEU ,ID_MYDATE_LIKEU ,ID_POST_LIKEU ) VALUES (5,5,9);
175 | INSERT INTO LIKEU (ID_USEUR_LIKEU ,ID_MYDATE_LIKEU ,ID_POST_LIKEU ) VALUES (6,6,9);
176 | INSERT INTO LIKEU (ID_USEUR_LIKEU ,ID_MYDATE_LIKEU ,ID_POST_LIKEU ) VALUES (1,3,9);
177 | INSERT INTO LIKEU (ID_USEUR_LIKEU ,ID_MYDATE_LIKEU ,ID_POST_LIKEU ) VALUES (8,8,3);
178 | INSERT INTO LIKEU (ID_USEUR_LIKEU ,ID_MYDATE_LIKEU ,ID_POST_LIKEU ) VALUES (9,8,2);
179 |
180 |
181 | INSERT INTO FOLLOW (ID_USEUR_FOLLOW ,ID_USEUR_FOLLOW2 ,ID_MYDATE_FOLLOW ) VALUES
    | (1,2,1);
182 | INSERT INTO FOLLOW (ID_USEUR_FOLLOW ,ID_USEUR_FOLLOW2 ,ID_MYDATE_FOLLOW ) VALUES
    | (1,3,1);
183 | INSERT INTO FOLLOW (ID_USEUR_FOLLOW ,ID_USEUR_FOLLOW2 ,ID_MYDATE_FOLLOW ) VALUES
    | (2,1,2);

```

```

184 | INSERT INTO FOLLOW (ID_USEUR_FOLLOW ,ID_USEUR_FOLLOW2 ,ID_MYDATE_FOLLOW ) VALUES
      (4,8,4);
185 | INSERT INTO FOLLOW (ID_USEUR_FOLLOW ,ID_USEUR_FOLLOW2 ,ID_MYDATE_FOLLOW ) VALUES
      (7,8,7);
186 | INSERT INTO FOLLOW (ID_USEUR_FOLLOW ,ID_USEUR_FOLLOW2 ,ID_MYDATE_FOLLOW ) VALUES
      (8,1,8);
187 | INSERT INTO FOLLOW (ID_USEUR_FOLLOW ,ID_USEUR_FOLLOW2 ,ID_MYDATE_FOLLOW ) VALUES
      (5,4,5);
188 | INSERT INTO FOLLOW (ID_USEUR_FOLLOW ,ID_USEUR_FOLLOW2 ,ID_MYDATE_FOLLOW ) VALUES
      (6,5,6);
189 | INSERT INTO FOLLOW (ID_USEUR_FOLLOW ,ID_USEUR_FOLLOW2 ,ID_MYDATE_FOLLOW ) VALUES
      (4,5,1);
190 |
191 |
192 | INSERT INTO COLLECTION (IDCOLLECTION, ID_USEUR_COLLECTION, TITRE, DESCRIPTION,
      ID_MYDATE_COLLECTION ) VALUES (1,3, 'Bonjour', 'ca va ', 1);
193 | INSERT INTO COLLECTION (IDCOLLECTION, ID_USEUR_COLLECTION, TITRE, DESCRIPTION,
      ID_MYDATE_COLLECTION ) VALUES (2,8, 'Bonjour2', 'ca va 2', 2);
194 | INSERT INTO COLLECTION (IDCOLLECTION, ID_USEUR_COLLECTION, TITRE, DESCRIPTION,
      ID_MYDATE_COLLECTION ) VALUES (3,5, 'Bonjour3', 'ca va 3', 3);
195 | INSERT INTO COLLECTION (IDCOLLECTION, ID_USEUR_COLLECTION, TITRE, DESCRIPTION,
      ID_MYDATE_COLLECTION ) VALUES (4,8, 'Bonjour4', 'ca va4 ', 4);
196 | INSERT INTO COLLECTION (IDCOLLECTION, ID_USEUR_COLLECTION, TITRE, DESCRIPTION,
      ID_MYDATE_COLLECTION ) VALUES (5,6, 'Bonjour', 'ca va ', 5);
197 | INSERT INTO COLLECTION (IDCOLLECTION, ID_USEUR_COLLECTION, TITRE, DESCRIPTION,
      ID_MYDATE_COLLECTION ) VALUES (6,3, 'Bonjour6', 'ca va 6', 6);
198 | INSERT INTO COLLECTION (IDCOLLECTION, ID_USEUR_COLLECTION, TITRE, DESCRIPTION,
      ID_MYDATE_COLLECTION ) VALUES (7,3, 'Bonjour', 'ca va ', 7);
199 | INSERT INTO COLLECTION (IDCOLLECTION, ID_USEUR_COLLECTION, TITRE, DESCRIPTION,
      ID_MYDATE_COLLECTION ) VALUES (8,7, 'Bonjour8', 'ca va 8', 8);
200 | INSERT INTO COLLECTION (IDCOLLECTION, ID_USEUR_COLLECTION, TITRE, DESCRIPTION,
      ID_MYDATE_COLLECTION ) VALUES (9,1, 'Bonjour9', 'ca va9 ', 9);
201 |
202 | INSERT INTO GALLERIE (IDGALLERIE ,ID_COLLECTION_GALLERIE ,ID_POST_GALLERIE ) VALUES
      (1,3,8);
203 | INSERT INTO GALLERIE (IDGALLERIE ,ID_COLLECTION_GALLERIE ,ID_POST_GALLERIE ) VALUES
      (1,3,2);
204 | INSERT INTO GALLERIE (IDGALLERIE ,ID_COLLECTION_GALLERIE ,ID_POST_GALLERIE ) VALUES
      (1,3,4);
205 | INSERT INTO GALLERIE (IDGALLERIE ,ID_COLLECTION_GALLERIE ,ID_POST_GALLERIE ) VALUES
      (2,7,8);
206 | INSERT INTO GALLERIE (IDGALLERIE ,ID_COLLECTION_GALLERIE ,ID_POST_GALLERIE ) VALUES
      (3,5,2);
207 | INSERT INTO GALLERIE (IDGALLERIE ,ID_COLLECTION_GALLERIE ,ID_POST_GALLERIE ) VALUES
      (3,5,5);
208 | INSERT INTO GALLERIE (IDGALLERIE ,ID_COLLECTION_GALLERIE ,ID_POST_GALLERIE ) VALUES
      (4,2,3);
209 | INSERT INTO GALLERIE (IDGALLERIE ,ID_COLLECTION_GALLERIE ,ID_POST_GALLERIE ) VALUES
      (5,9,1);
210 |
211 |
212 | INSERT INTO ALBUM (IDALBUM ,ID_COLLECTION_ALBUM ,ID_POST_ALBUM ) VALUES (7,4,6);
213 | INSERT INTO ALBUM (IDALBUM ,ID_COLLECTION_ALBUM ,ID_POST_ALBUM ) VALUES (2,8,7);
214 | INSERT INTO ALBUM (IDALBUM ,ID_COLLECTION_ALBUM ,ID_POST_ALBUM ) VALUES (3,4,8);
215 | INSERT INTO ALBUM (IDALBUM ,ID_COLLECTION_ALBUM ,ID_POST_ALBUM ) VALUES (4,6,4);

```

### Partie 3 : Creation des fonctions necessaires aux requêtes SQL

```
01 |
02 |
03 |
04 | CREATE OR REPLACE FUNCTION deg2rad ( deg IN FLOAT)
05 | RETURN FLOAT
06 | IS
07 | res FLOAT ;
08 | BEGIN
09 |     res := (deg * 3.141592653589793238462643383279502884197169399375) /180;
10 |     RETURN (res) ;
11 | END;
12 | /
13 | CREATE OR REPLACE FUNCTION surfaceToDist ( surface IN FLOAT)
14 | RETURN FLOAT
15 | IS
16 | res FLOAT ;
17 | BEGIN
18 |     res := SQRT(surface/3.141592653589793238462643383279502884197169399375); --
19 |     my_global_varpackage.pi);
20 |     RETURN (res) ;
21 | END;
22 | /
23 | CREATE OR REPLACE FUNCTION haversine ( lat1 IN NUMBER, lon1 IN NUMBER, lat2 IN NUMBER
24 |     , lon2 IN NUMBER)
25 | RETURN NUMBER
26 | IS
27 | dLon NUMBER;
28 | dLat NUMBER ;
29 | a1 NUMBER ;
30 | c1 NUMBER ;
31 | d1 NUMBER ;
32 | BEGIN
33 |     dLat := deg2rad(lat2-lat1);
34 |     dLon:= deg2rad(lon2-lon1);
35 |     a1:=SIN(dLat/2) * SIN(dLat/2) +COS(deg2rad(lat1)) * COS(deg2rad(lat2)) * SIN(
36 |         dLon/2) * SIN(dLon/2);
37 |     c1:=2 * ATAN2(SQRT(a1), SQRT(1-a1));
38 |     d1:= 6378 * c1;
39 |     RETURN d1;
40 | END;
41 | /
42 |
43 | CREATE OR REPLACE FUNCTION max_min_lat (maxe IN NUMBER, lat IN NUMBER , rayon IN
44 |     NUMBER )
45 | RETURN NUMBER
46 | IS
47 | tmp NUMBER;
48 | BEGIN
49 | IF maxe = 1 THEN
50 |     tmp := lat + (rayon / 6378) * (180 /
51 |         3.141592653589793238462643383279502884197169399375);
52 | ELSIF maxe = 0 THEN
53 |     tmp := lat - (rayon / 6378) * (180 /
54 |         3.141592653589793238462643383279502884197169399375);
55 | ELSE
56 |     RAISE_APPLICATION_ERROR(-20500, 'maxe correspond a rien' );
57 | END IF ;
58 | return tmp ;
59 | END;
60 | /
61 | CREATE OR REPLACE FUNCTION max_min_lng (maxe IN NUMBER , lat IN NUMBER , lng IN
62 |     NUMBER , rayon IN NUMBER )
```

```

59 | RETURN NUMBER
60 | IS
61 | tmp NUMBER;
62 | BEGIN
63 | IF maxe = 1 THEN
64 |     tmp := lng + (rayon /6378) * (180 /
        3.141592653589793238462643383279502884197169399375) / cos(lat *
        3.141592653589793238462643383279502884197169399375/180);
65 |     ELIF maxe = 0 THEN
66 |         tmp := lng - (rayon /6378) * (180 /
        3.141592653589793238462643383279502884197169399375) / cos(lat *
        3.141592653589793238462643383279502884197169399375/180);
67 | ELSE
68 |     RAISE_APPLICATION_ERROR(-20500,'maxe correspond a rien' );
69 | END IF ;
70 | return tmp ;
71 | END;
72 | /

```