Emilie Paniagua

Gestion du projet

Planning et suivi de l'activité

Etat d'avancement

Conception de la base de données

Dictionnaire des données

Règles de gestion

Modèles des données

Mise en place de la base de données

Création des tables et des autres objets de la base (vues, index, …)

Elaboration du jeu d'essai en partant des scénarios les plus représentatifs

♣ Réalisation

Requêtes et procédures les plus significatives et résultats obtenus

♣ Gestion de la sécurité

Mise en place des droits utilisateurs

Sauvegarde de la base de test.

♣ Bilan

Apports et difficultés rencontrées

Perspectives

(Le dossier de projet hors annexes doit comporter entre 15 et 20 pages)

Liste des compétences du référentiel couvertes par ce jalon :

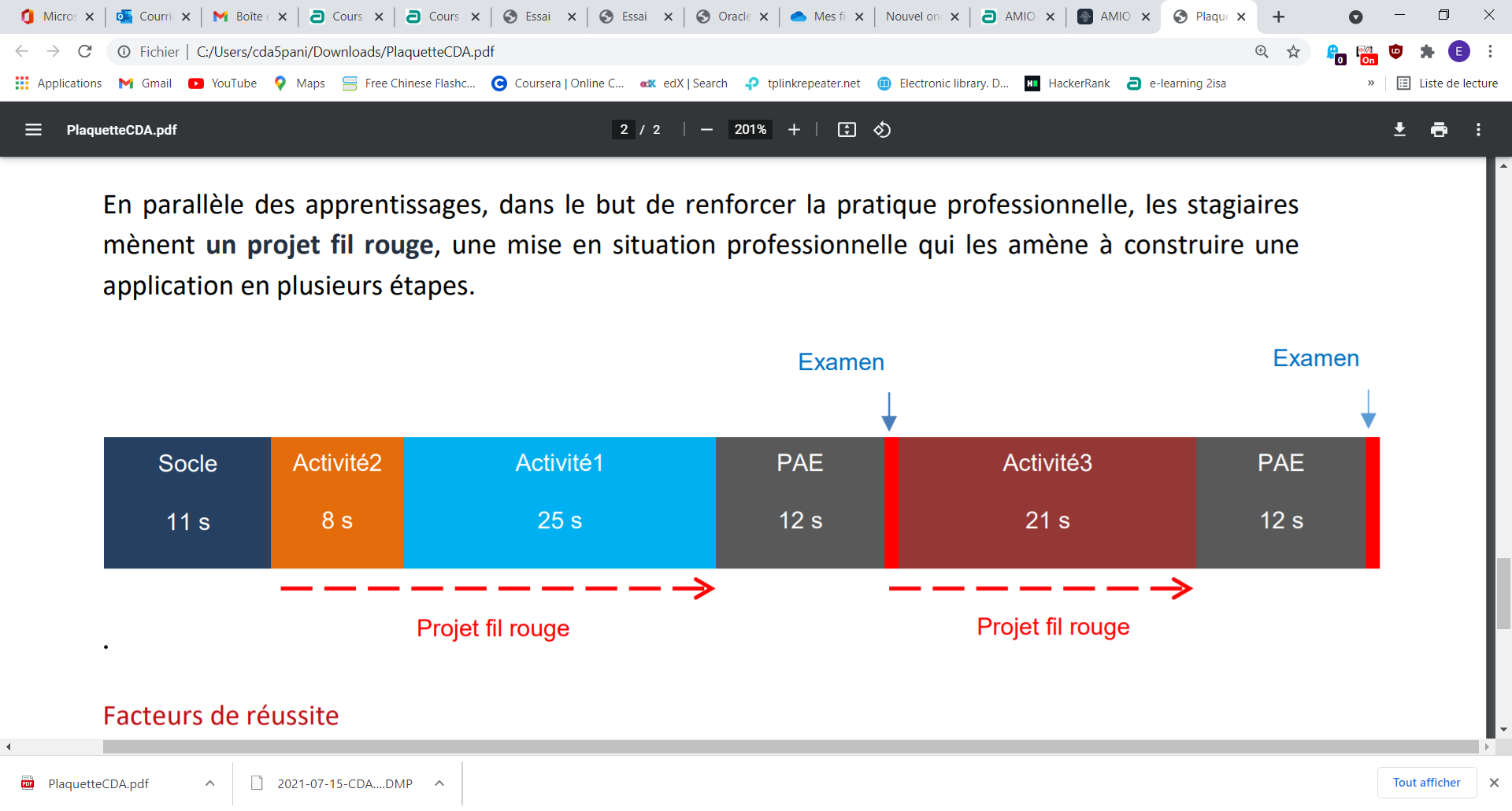
Concevoir une base de données

Mettre en place une base de données

Développer des composants dans le langage d'une base de données.

Chaque jalon se conclut par un EPCF, un examen partiel en cours de formation.

Nous en sommes au jalon 1



**Jalon 1**

Jalon 2

Jalon 3

Jalon 4

Expression des besoins du projet

Le fil rouge nous a été présenté en amphi, pour son lancement, le 21 juin 2021. C'est un projet qui permet de récapituler et d'ancrer les différentes compétences acquises en cours.

Analyse du besoin

Cas d'utilisation

Description d'un scénario significatif

Gestion du temps :

Concevoir la base de données nécessite de jongler de façon récursive avec trois étapes principales de la création du projet :

* Les règles de gestion et le dictionnaire des données issu du cahier des charges et de la FAQ
* Le MCD
* Les create table et insertions de valeurs dans la base de données

Au fur et à mesure de la lecture du cahier des charges, de la compréhension de la FAQ et des échanges avec les formateurs, j'en déduis que le travail avance suivant une boucle que l'on peut appeler boucle de feedback (ou roue de Deming, voir image ci-dessous),



Figure 1 : Boucle de Feedback : comment comprendre son fonctionnement

La boucle de feedback fonctionne suivant 4 micro-étapes:

* **planifier (plan),** où l'on envisage la tâche à réaliser, les paramètres nécessaires et la technique à employer
* **faire (do),** on met en œuvre la technique avec les paramètres envisagés
* **vérifier (check),**  on vérifie si l'étape a bien produit les résultats escomptés
* **agir (act),** suivant les résultats obtenus, on passe à l'étape suivante.

**Exemple pratique sur une boucle :**

* **Plan :** j'envisage d'ajouter des attributs à une entité Adhérents que je souhaite créer sous Looping. J'ajoute l'id, le nom, le prénom, la date de naissance, la photo de profil, le numéro de voie, le nom de rue, le code postal, la ville, le pays. J'envisage de créer une table Organisateur, qui hériterait de la table Adhérents.
* **Do :** je codifie ces informations dans le Dictionnaire des données et je les rentre dans le MCD sur Looping.
* **Check :** Je vérifie en échangeant avec mes formateurs si l'architecture est intéressante. Ceux-ci me disent que l'héritage est difficile à exploiter sous looping et Oracle.
* **Act :** Je décide donc de fusionner les deux tables Adhérent et Organisateur en une table Membre, et de recourir à un booléen (Nombre 1 pour vrai, 0 pour faux sous Oracle) pour départager si le membre est ou non un organisateur.

**Exemple avec une boucle sur les insertions de données :**

* **Plan :** Je me suis rendue compte qu'il manquait le login, le mot de passe, et un numéro de téléphone dans membres. Je planifie l'ajout de ces attributs dans le MCD et le dictionnaire de données.
* **Do :** Je codifie et rentre les données dans le MCD, et comme l'étendue de mes autres tables est prête, je copie le code de création des tables et je l'insère dans la base oracle. J'insère également des données dans mon fichier insert correspondant au login, mot de passe et numéro de téléphone de membres.
* **Check :** Je vérifie que les informations ont été écrites proprement en base.
* **Act :** Je sauvegarde mes fichiers de création de table et d'inserts de valeurs sur des fichiers texte.

En conséquence, par la nature même de ces itérations, il est très difficile d'estimer et de prévoir le temps que l'on va passer à chacune desdites étapes. Je vais donc me livrer à cet exercice mais en conscience qu'il a ses limites. Pour ce faire, j'ai établi un tableau de suivi sur Trello où j'ai répertorié l'avancée quotidienne des différentes tâches que j'ai entreprises. (nota : j'ai perdu l'accès au tableau Trello, il ne me reste plus que la prise d'écran ci-dessous pour évaluer le temps passé, je choisis donc de transférer le suivi sur une feuille excel classique)

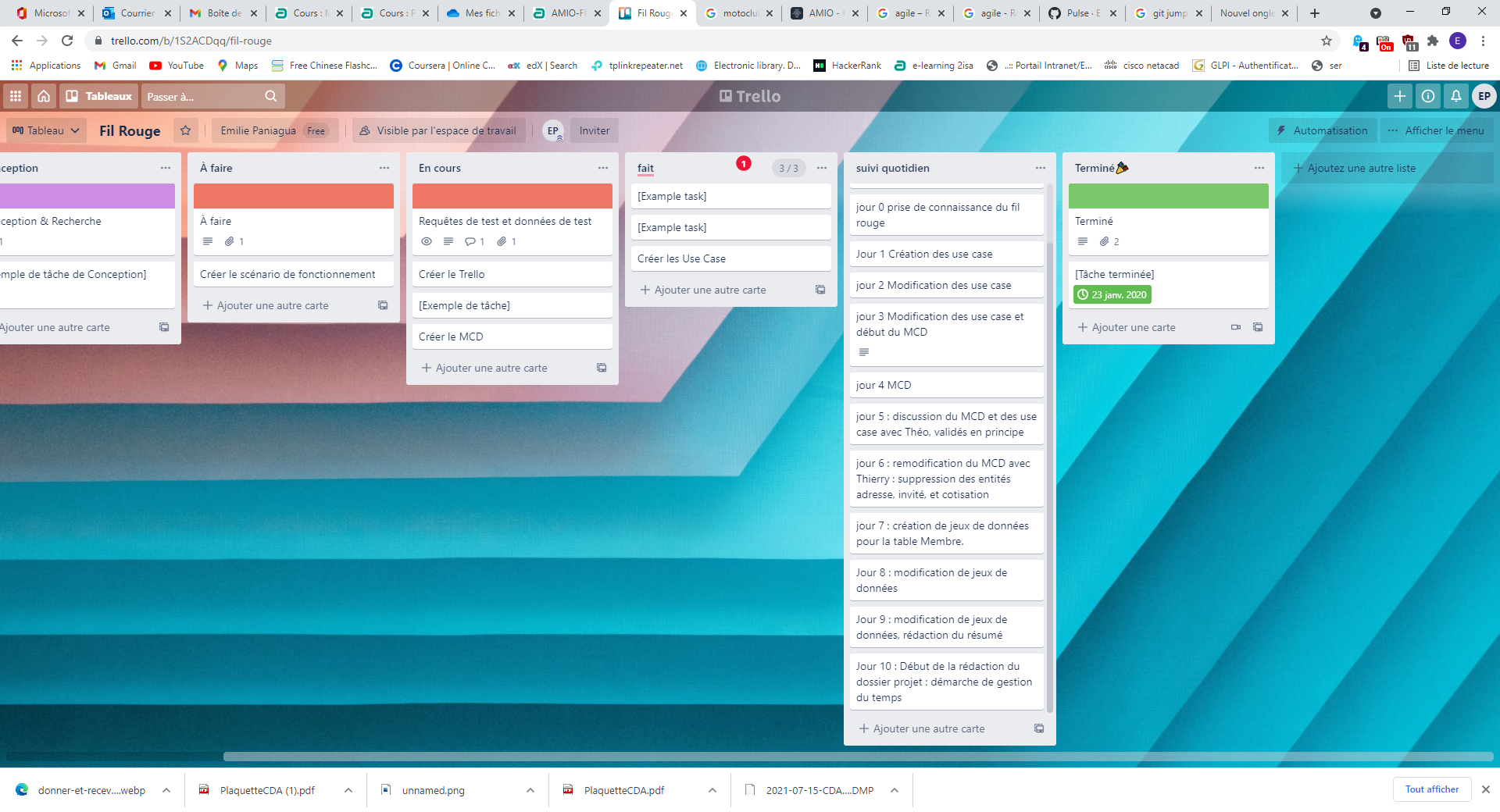


Figure 2 : impression d'écran du suivi des tâches au 15 juillet 2021

(ERRATUM : le suivi quotidien est fait en demi-journées de 4 heures et non en journées entières)

Tableau excel de gestion du temps

Suivi des modifications GIT

Pour suivre et sauvegarder les modifications que j'apporte à mon dossier fil rouge, je me sers de Git et d'un [repository sur Github](https://github.com/E1000y/FilRougeMCMP).

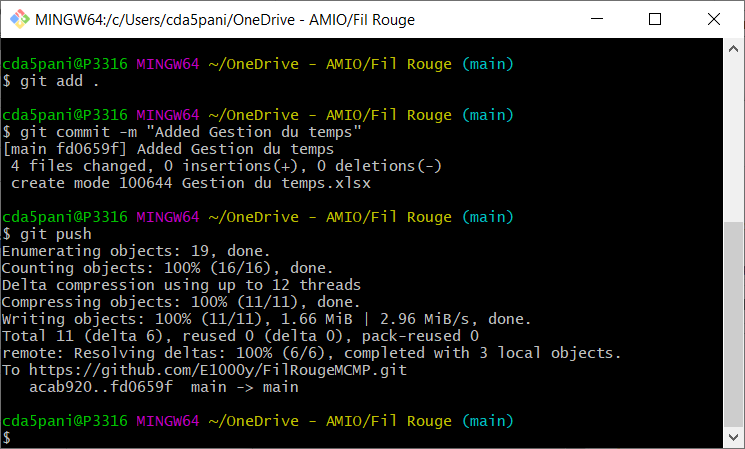


Figure : Gestion des versions avec Git, exemple

**Exemple de scénario significatif**

# **Cas :** "Ajouter sa photo de profil"

**Objectif :** Permettre d'ajouter sa photo de profil.

**Acteurs :** l'adhérent

**Date :** le 17/08/2021

**Responsable :** Emilie Paniagua

**Version :** 3.0

**Préconditions :** l'adhérent visualise ses informations personnelles. Le système présente une photo de profil par défaut.

**Scénario :**

1. L'adhérent choisit une photo depuis sa galerie ou son ordinateur
2. L'adhérent valide son choix de photo
3. Le système stocke la photo associée à l'utilisateur

**Alternatives :**

1. L'adhérent ne choisit pas de photo de profil
2. Le système laisse un avatar par défaut

**Exceptions :**

Si le système n'arrive temporairement pas à joindre la base de données, il affiche un message d'erreur du type : "connexion à la base de données impossible, veuillez réessayer plus tard".

**Postconditions :**

L'adhérent a correctement inséré sa photo

Le système affiche sa page de profil, comprenant sa photo correctement affichée.

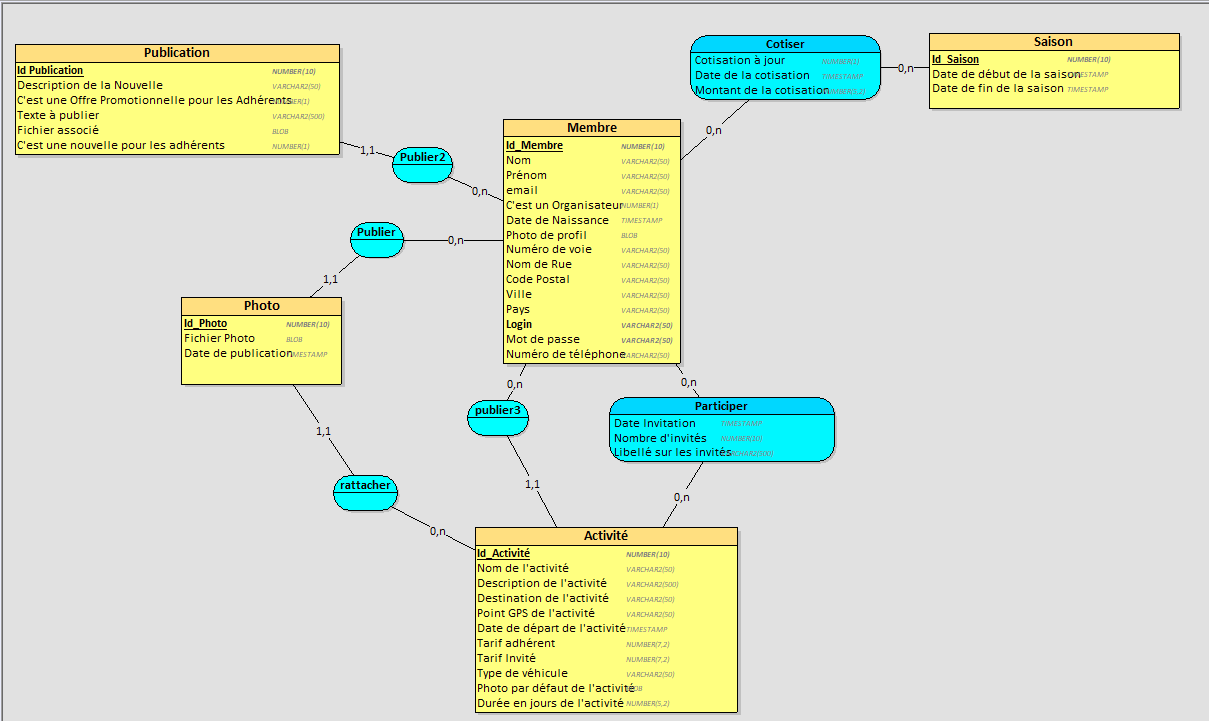
Diagramme des cas d'utilisationDictionnaire de données

La lecture du cahier des charges permet de tirer

|  |  |  |
| --- | --- | --- |
| **Libellé** | **Code** | **Type** |
|  |  |  |
| Id du membre : adhérent ou organisateur | Id\_Membre | number(10) |
| nom du membre | Nom | Varchar2(50) |
| prénom du membre | Prenom | Varchar2(50) |
| email du membre | email | Varchar2(50) |
| Est un organisateur | IsOrganisateur | Number(1) |
| Date de naissance | Date\_Naissance | Timestamp |
| Numéro de voie | No\_Voie | Varchar2(50) |
| Nom de rue | Nom\_De\_Rue | Varchar2(50) |
| Code postal | Code\_Postal | Varchar2(50) |
| Ville | Ville | Varchar2(50) |
| Pays | Pays | Varchar2(50) |
| Login | Login | Varchar2(50) |
| Mot de Passe | mdp | Varchar2(50) |
| Numéro de Téléphone | NoTel | Varchar2(50) |
|  |  |  |
| id activité | Id\_Activite | number(10) |
| nom de l'activité | Nom\_Activite | Varchar2(50) |
| Description de l'activité | Description\_Activite | Varchar2(50) |
| Destination Activité | Destination\_Activite | Varchar2(50) |
| Point GPS Activité | Point\_GPS\_Activite | Varchar2(50) |
| Date de l'activité | Date\_Activite | timestamp |
| Tarif Adhérent | Tarif\_Adherent | Number(7,2) |
| Tarif invité | Tarif\_Invite | Number(7,2) |
| Type de véhicule | Type\_Vehicule | Varchar2(50) |
| photo par défaut de l'activité | Photo\_Activite | BLOB |
| Durée en jours de l'activité | Duree\_Activite | Number(5,2) |
|  |  |  |
| Date d'invitation | Date\_Invitation | Timestamp |
| Nombre d'invités | Nombre\_invites | Number(10) |
| Libellé sur les invités | Libelle\_invites | Varchar2(500) |
|  |  |  |
| id photo | Id\_Photo | Number(10) |
| fichier photo | Fichier\_Photo | BLOB |
| date de publication | Date\_Publi | Timestamp |
|  |  |  |
| id de la publication | Id\_Publication | Number(10) |
| Description de la nouvelle | Description\_Nouvelle | Varchar2(50) |
| C'est une Offre Promotionnelle pour les Adhérents | isOffrePromotionnelleAdherents | Number(1) |
| Texte à publier | Texte | Varchar2(500) |
| C'est une nouvelle pour les adhérents | IsNouvelleAdherents | Number(1) |
| Fichier associé | Fichier\_Associe | BLOB |
|  |  |  |
| Cotisation à jour | Cotisation\_A\_Jour | Number(1) |
| Date de la cotisation | Date\_Cotisation | Timestamp |
| Montant de la cotisation | Montant\_Cotisation | Number(5,2) |
|  |  |  |
| Id Saison | Id\_Saison | Number(10) |
| Date de début de la saison | Date\_Debut\_Saison | Timestamp |
| Date de fin de la saison | Date\_Fin\_Saison | Timestamp |

Le dictionnaire de données et le MCD étant créés en même temps, l'un prenant appui sur l'autre pour la réflexion de la construction de l'ensemble, voici la version finale de celui-ci qui donne une bonne visibilité à la structure à adopter pour construire les tables du MCD.

Présentation du Modèle Conceptuel de Données MCD:



MLD

Membre

**Membre = (Id\_Membre *NUMBER(10)***, Nom *VARCHAR2(50)*, Prenom *VARCHAR2(50)*, email *VARCHAR2(50)*, IsOrganisateur *NUMBER(1)*, Date\_Naissance *TIMESTAMP*, Photo\_Profil *BLOB*, No\_Voie *VARCHAR2(50)*, Nom\_De\_Rue *VARCHAR2(50)*, Code\_Postal *VARCHAR2(50)*, Ville *VARCHAR2(50)*, Pays *VARCHAR2(50)*, **Login *VARCHAR2(50)***, mdp ***VARCHAR2(50)***, NoTel *VARCHAR2(50)***);**

**Activite = (Id\_Activite *NUMBER(10)***, Nom\_Activite *VARCHAR2(50)*, Description\_Activite *VARCHAR2(500)*, Destination\_Activite *VARCHAR2(50)*, Point\_GPS\_Activite *VARCHAR2(50)*, Date\_Activite *TIMESTAMP*, Tarif\_Adherent *NUMBER(7,2)*, Tarif\_Invite *NUMBER(7,2)*, Type\_Vehicule *VARCHAR2(50)*, Photo\_Activite *BLOB*, Duree\_Activite *NUMBER(5,2), #Id\_Membre***);**

**Photo = (Id\_Photo *NUMBER(10)***, Fichier\_Photo *BLOB*, Date\_Publi\_Photo *TIMESTAMP, #Id\_Activite, #Id\_Membre***);**

**Saison = (Id\_Saison *NUMBER(10)***, Date\_Debut\_Saison *TIMESTAMP*, Date\_Fin\_Saison *TIMESTAMP***);**

**Publication = (Id\_Publication *NUMBER(10)***, Description\_Nouvelle *VARCHAR2(50)*, IsOffrePromotionnelleAdherents *NUMBER(1)*, Texte *VARCHAR2(500)*, Fichier\_Associe *BLOB*, IsNouvelleAdherents *NUMBER(1), #Id\_Membre***);**

**Participer = (*#Id\_Membre, #Id\_Activite***, Date\_Invitation *TIMESTAMP*, Nombre\_Invites *NUMBER(10)*, Libelle\_Invites *VARCHAR2(500)***);**

**Cotiser = (*#Id\_Membre, #Id\_Saison***, Cotisation\_A\_Jour *NUMBER(1)*, Date\_Cotisation *TIMESTAMP*, Montant\_Cotisation *NUMBER(5,2)***);**