

Modélisation de BD

Exercices

Modèle Relationnel

Modèle Sagittal-MEA-UML

Modèles normalisés et valorisés

Logiciel de modélisation

Bertrand LIAUDET

SOMMAIRE

SOMMAIRE	1
MR-MEA-EXERCICES	3
Série 1 - Conception	3
1 Association et donateurs	5
2 La maison de disques	7
3 La bibliothèque	8
3 bis : Retail_BD	10
4 La cinémathèque	11
5 Gestion de projet	12
6 Les employés et leurs carrières	12
7 ASSURAUTO	13
8 Les enchères	14
9 Centres d'animation	14
Série 1 bis – Conception spéciale	15
1 Le cyclisme	15
2 Rolland Garros	15
Série 2 – Critiques et corrections	16
1 Contrat et client	16
2 Employé et fonction	16
3 Stock	17
4 Etudiant et Groupe de TD	17
Série 3 – Conception avancée	18
1 Les chantiers d'été	18
2 Le chenil	19
Série 4 – A partir du MR	20
1 La clinique vétérinaire	20

Série 5 – A partir du SQL	21
1. DDL Chantiers	21
Série 6 - départ MEA	24
1 MEA-171	24
2 MCD-173	24
3 MCD-175	25
Série 7 – Départ UML	26
1. UML-utc	26
2. UML-174	26
3. UML-147	27
4. UML-172	27
5. UML-147	28
6. UML-148	28
7. UML-149	29

Edition février 2019

MR-MEA-EXERICES

Série 1 - Conception

Chaque exercice présente une situation. D'après la situation, il faut concevoir la BD en suivant les étapes ci-dessous (toutes les étapes ne doivent pas forcément être traitées dans l'ordre).

➤ **Etape 1 : MEA-UML / MR-GT**

1. Faites le MEA normalisé. Précisez les clés primaires et les clés significatives des entités. Vérifiez qu'il n'y a pas d'attribut calculés.
2. Faire le MR normalisé correspondant. Précisez les clés primaires des nouvelles tables issues du MEA.
3. Faites le GT correspondant.
4. Faites l'UML correspondant au MEA.

➤ **Etape 2 : Valorisation**

5. Valoriser le modèle et faites le dictionnaire des attributs : type, obligatoire (oui par défaut), unique (non par défaut), valeur par défaut, modifiable (non par défaut), énuméré, valeurs limites, calculé).
6. Modèle externe : certaines questions nécessitent la définition d'un modèle externe pour gérer les fonctionnalités attendues. Dans ce cas, proposer le modèle et la ou les vues SQL associées.

➤ **Etape 3 : Construire un jeu de données**

7. Remplissez les tables avec des données.
8. Quels attributs calculés peut-on envisager ?
9. Si vous connaissez le SQL, écrivez les requêtes SQL permettant de calculer les attributs calculés envisagés.

➤ **Etape 4 : Ontologie**

10. Pour chaque table, préciser le type de la clé primaire et la sémantique de la table (ontologie relationnelle).

➤ **Etape 5 : Analyse fonctionnelle**

11. A partir du graphe des tables, réfléchissez aux usages nécessaire pour pouvoir utiliser la base (cycle de vie des données).
12. Si vous connaissez les diagrammes de cas d'utilisation UML, à partir du GT, faites une analyse fonctionnelle avec un diagramme des cas d'utilisation pour simuler un fonctionnement du SI.

➤ **Etape 6 : Logiciel de modélisation**

13. Faites le MEA valorisé au mieux avec un logiciel de modélisation.
14. Transformez le MEA en MR. Vérifiez que les attributs sont bien valorisés et que les clés primaires sont justes.
15. Transformez le MR en code SQL.
16. Transformer le MEA en UML.

➤ **Etape 7 : charger des données dans une BD**

17. A partir d'un jeu de données sous Excel construit de telle sorte que chaque table Excel corresponde à un table de la BD, importez les tables Excel dans la BD (avec phpMyAdmin par exemple).

Première partie

Une association reçoit des dons de donateurs privés.

Pour chaque don, l'association enregistre la date du don, son montant ainsi que le nom et l'adresse du donateur, le téléphone et l'adresse mail si possible. Les dons ont un numéro identifiant qui les distingue les uns des autres.

L'association veut pouvoir afficher en permanence la liste des donateurs avec leur ancienneté (l'année de leur premier don), le total de dons effectués, la moyenne du nombre de dons effectué par an, la moyenne du total des montants donnés par an, le nombre de dons effectués pour l'année en cours et le montant total des dons effectués pour l'année en cours.

Deuxième partie

Vers le mois de février, l'association envoie des reçus fiscaux aux donateurs de l'année. On conserve la date d'envoi du reçu. Les reçus fiscaux concernent uniquement les dons de l'année précédente. Ils précisent le nom et l'adresse du donateur, l'année fiscale (qui correspond à l'année du don), le montant et la date du don. Si un même donateur fait plusieurs dons dans l'année, il reçoit un reçu par don. Si on doit renvoyer le reçu une deuxième fois (le donateur appelle pour dire qu'il ne l'a pas reçu), on ne change pas les données dans la BD et on renvoie un reçu identique.

Pour les reçus fiscaux, l'association veut pouvoir, avant d'envoyer les reçus pour l'année précédente, lister les donateurs avec le nombre de dons de l'année précédente, le montant total des dons et la liste des dons avec le numéro, le montant et la date (pour cette liste, MySQL offre la fonction « group-concat » qui permet de lister dans un attribut la liste des tuples regroupés par un group by).

Troisième partie

L'association envoie des courriers aux donateurs pour les solliciter. Un courrier est caractérisé par un libellé (un titre), une date d'envoi et son texte. Plusieurs mailings différents peuvent être envoyés à la même date. Il arrive que plusieurs courriers aient le même titre sans avoir le même texte. Un don peut faire référence à un courrier ou pas. Pour chaque courrier, on connaît le montant total des dons qui s'y rapporte.

Quatrième partie (variante de la deuxième).

La gestion des reçus fiscale a changé : l'administration fiscale demande à avoir un reçu par an et par donateur avec la liste des dons (montant et date), le nombre total de dons et le total des dons. Elle demande aussi que chaque reçu ait un numéro identifiant.

L'association envoie ce reçu vers le mois de février et enregistre la date d'envoi. Si on doit renvoyer le reçu une deuxième fois (le donateur appelle pour dire qu'il ne l'a pas reçu), on ne change pas les données dans la BD et on renvoie un reçu identique.

Modifier le modèle issu de la deuxième partie pour arriver à une solution.

Travail à faire

MEA et/ou UML, MR, GT, MR valorisé, Données, SQL.

Jeu de données

Pour le jeu de données, on travaille sur 2 années : l'année en cours (année n) pour laquelle il n'y a pas eu de reçu fiscaux et l'année précédente pour laquelle il y a eu des reçus fiscaux. On se dote d'au moins 4 donateurs : 3 sur l'année n dont 2 déjà présent à l'année n-1 et 3 sur l'année n-1 (dont 2 retrouvés sur l'année n). 1 donateur de l'année n-1 à fait plus de 1 don.

2 La maison de disques

Une maison de disques gère des disques. Un disque contient des chansons. On connaît le nombre des chansons du disque. On connaît aussi sa durée. Les chansons ont un numéro sur le disque. Les chansons ont un titre et une durée. Pour chaque chanson, on connaît la liste des musiciens qui y participent ainsi que le nom du ou des chanteurs, le nom du ou des auteurs et le nom du ou des compositeurs. Pour chaque personne, on connaît son nom et sa nationalité.

Pour le jeu de données, on se donne au moins 3 Cds dont l'un reprend au moins une chanson de l'autre. On se donne au moins 2 artistes principaux différents.

Travail

MEA, UML, MR, GT, MR valorisé, Données, SQL,

3 La bibliothèque

Le code de création de la BD est le suivant :

```
drop database if exists biblio;
create database biblio;

use biblio;

CREATE TABLE oeuvres(
  NO          integer primary key auto_increment,
  titre       varchar(150),
  auteur      varchar(100)
) ENGINE InnoDB;

CREATE TABLE adherents (
  NA          int PRIMARY KEY AUTO_INCREMENT,
  nom         varchar(30) not null,
  prenom      varchar(30),
  adr         varchar(100) not null,
  tel        char(10)
) ENGINE InnoDB;

CREATE TABLE livres (
  NL          integer primary key auto_increment,
  editeur     varchar(50),
  NO          integer not null,
  foreign key(NO) references oeuvres(NO)
) ENGINE InnoDB;

CREATE TABLE categories(
  NC          integer primary key auto_increment,
  categorie   varchar(150)
) ENGINE InnoDB;

CREATE TABLE thematique (
  NO          integer not null,
  NC          integer not null,
  foreign key(NO) references oeuvres(NO),
  foreign key(NC) references categories(NC),
  primary key (NO, NC)
) ENGINE InnoDB;

CREATE TABLE emprunter (
  NL          integer not null,
  datEmp      date not null,
  dureeMax    integer not null,
  dateRet     date,
  NA          integer not null,
  foreign key(NA) references adherents(NA),
  foreign key(NL) references livres(NL),
  primary key (NL, datEmp)
) ENGINE InnoDB;
```

Ce code répond à la situation suivante :

Une bibliothèque gère les emprunts des livres de ses adhérents. Les livres ont un titre, un auteur, un éditeur, une année d'édition. On peut avoir plusieurs exemplaire du même livre mais aussi plusieurs éditions différentes de la même œuvre. Les adhérents ont un nom, une adresse, une adresse mail et éventuellement un téléphone. On souhaite archiver tous les emprunts. Un livre ne peut pas être rendu le jour même de son emprunt. La durée maximum d'emprunt est de 21 jours. Un adhérent ne peut pas avoir plus de 15 emprunts en cours. Tout retard dans les rendus bloque la possibilité de nouveaux emprunts.

Les livres sont rattachés à des genres (jeunesse, SF, fantastique, roman, bande dessinée, philosophie, politique, histoire, etc.). Une œuvre peut être rattachée à plusieurs genres.

A tout moment, on connaît le nombre de livres actuellement empruntés par chaque adhérent ainsi que le nombre de livres actuellement en retard de rendu.

La bibliothèque souhaite pouvoir faire des statistiques sur la pratique des abonnés (nombre de livres empruntés par an, répartition des emprunts par genre, nombre d'emprunts par livre, etc.

De plus, les abonnés peuvent commander des livres. Ils peuvent en commander 5 au maximum. Une commande peut être annulée ou honorée si le livre commandé a finalement été emprunté.

La bibliothèque souhaite garder la trace de toutes les commandes effectuées, qu'elles aient été annulées ou honorées.

Travail

GT, MR, MR valorisé, ontologie, Données, MEA, UML

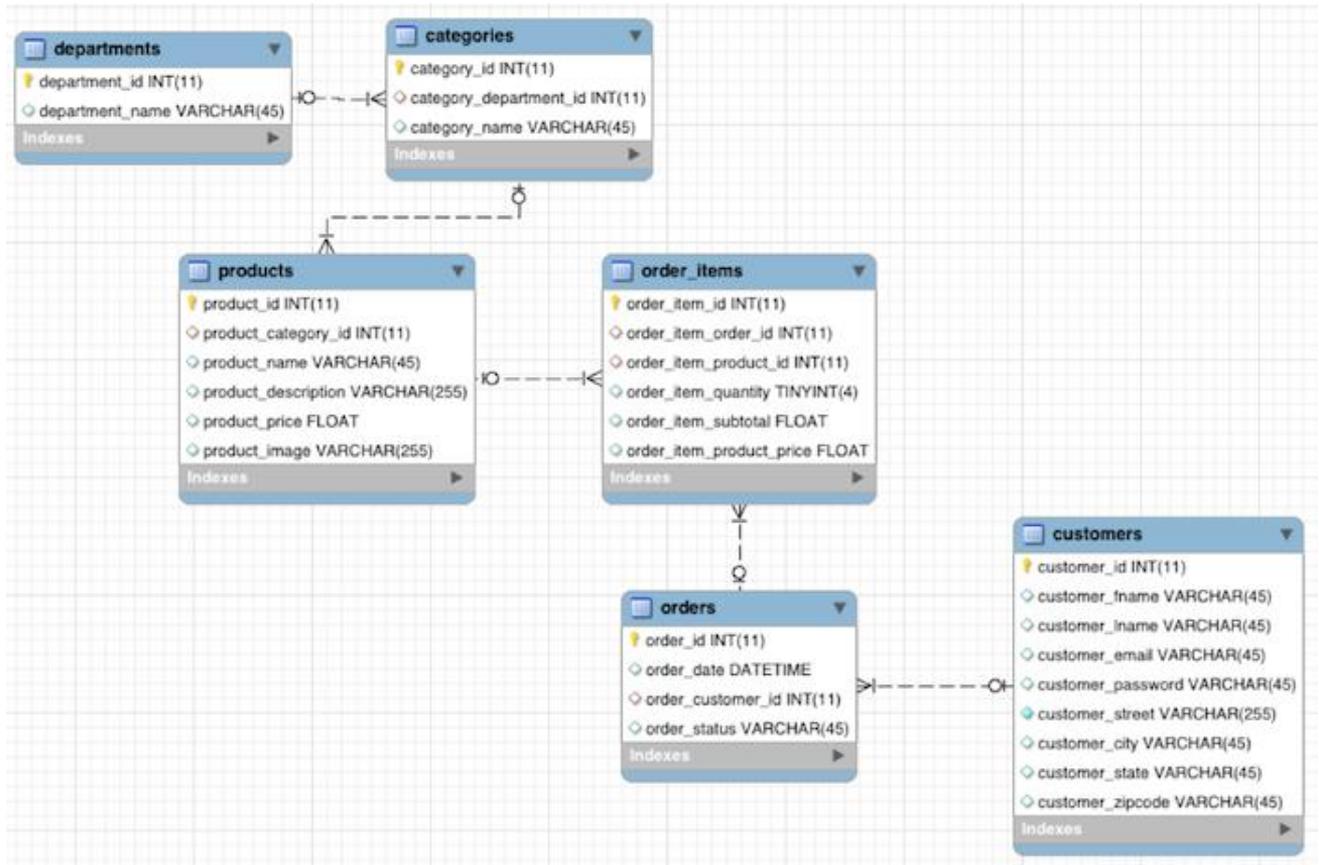
Prioritairement :

GT, MR textuel valorisé, MEA ou UML.

3 bis : Retail_BD

Retail_DB est un exemple de BD qu'on trouve sur internet.

Voici le schéma proposé pour cette BD :



Vous pouvez charger cette BD à l'adresse suivante : http://bliaudet.free.fr/IMG/zip/retail_db.zip

Mais ce n'est pas utile pour l'exercice.

Travail à faire :

- 1) De quel type de modèle s'agit-il ?
- 2) Faites un GT
- 3) Que pensez-vous des cardinalités ?
- 4) A quel type de table correspond « order_item »
- 5) Que pensez-vous de l'attribut « order_item_id » ?
- 6) Faites un MEA (ou UML)

4 La cinémathèque

Une cinémathèque veut mettre en place un système d'information concernant des films et leurs diffusions. La cinémathèque gère plusieurs sites dans différentes villes (un site par ville). Pour chaque film on connaît son titre, sa date de création, ses acteurs avec leur nom, leur prénom et leur nationalité, les réalisateurs avec les mêmes informations, la langue du film. On connaît aussi les salles dans lesquelles il a été projeté, avec l'heure et le jour de la projection, la version projetée (version originale ou version française). On connaît le nombre de places des salles, le type d'écran et le type de son, ainsi que le site correspondant, avec son nom, son adresse, son téléphone et le nombre de salles du cinéma. On veut pouvoir faire un classement hebdomadaire, mensuel et annuel du nombre d'entrées par film, par réalisateur et par acteur.

Pour le jeu de données, on se donne au moins 2 films projetés sur 2 journées. Un film avec 1 réalisateur qui n'est pas acteur. Un film avec un réalisateur qui est aussi acteur dans son film.

Travail

MEA, UML, MR, GT, MR valorisé, Données, SQL

5 Gestion de projet

On modélise l'organisation des projets en tâches dans une entreprise. Un projet est caractérisé par un nom, une date début, une date de fin et un identifiant unique. Un projet est constitué de plusieurs tâches numérotées de 1 à N. Chaque tâche a un intitulé, une date de début et une date de fin. Elle est précédée par 0 ou plusieurs tâches. Elle est suivie par 0 ou plusieurs tâches. Deux tâches d'un même projet peuvent démarrer à la même date. Deux tâches d'un même projet peuvent se terminer à la même date. Une tâche démarre au plus tôt le lendemain de la fin de la tâche qu'elle suit. Une tâche ne peut pas commencer avant le début du projet, ni finir après la fin du projet.

Dans l'entreprise, les membres du personnel ont un nom, une fonction et un identifiant. A chaque projet, on affecte des membres du personnel avec une fonction spécifique pour chaque projet. Ensuite, on affecte les membres du projet à des tâches du projet.

Chaque tâche peut requérir des ressources spécifiques. L'allocation d'une ressource par une tâche se fait à partir d'un certain jour et pour une certaine durée. Une ressource peut être allouée plusieurs fois par une même tâche à des périodes différentes. Une ressource a un intitulé et coût journalier.

6 Les employés et leurs carrières

On souhaite gérer le personnel d'une société. Chaque membre du personnel a un nom, une fonction, un salaire, une commission (part de salaire variable), un salaire totale (salaire + commission), une date d'entrée dans la société. Chaque membre du personnel travaille dans un département caractérisé par son nom (commercial, production, personnel, comptable et recherche) et la ville dans laquelle il se trouve.

Chaque membre du personnel a un supérieur hiérarchique et un seul lui-même membre du personnel, sauf le président qui n'a pas de supérieur hiérarchique.

A cette situation, qui donne lieu au schéma de la BD abordé en cours, on ajoute les éléments suivants :

Le service des Ressources Humaines veut gérer la carrière des employés. Pour chaque employé, on considère chaque modification de salaire, chaque changement de poste, chaque changement de supérieur hiérarchique et chaque changement de département comme une étape dans sa carrière. L'historique des commissions n'est, par contre, pas gardé.

On souhaite construire un MR destiné à la gestion de sinistres pour la compagnie d'assurance-automobile. ASSURAUTO à partir de la lettre de demande d'expertise envoyée par un de ses agents d'assurance.

Voici la lettre :

A Paris, le 10 juin 2009

De la part de M. Pierre Durand, Agent d'assurance
ASSURAUTO
10, rue Saint-Antoine, 75001 Paris
tél : 01 42 42 24 24

A M. Henri Dupond
Expert en assurance

Monsieur,

Je vous saisis par le présent courrier pour vous demander de suivre le dossier du sinistre SIS20020934 constitué à ce jour suite à l'accident survenu le 02 juin 2005 au croisement de la rue de Rivoli et de la rue du Renard à Paris 1^{er} arrondissement, à 11 heures du soir, par temps de pluie, d'où l'état glissant de la chaussée à ce moment-là. M. André Danlemur est un des protagonistes de l'accident. Il a souscrit chez nous, depuis le 17 décembre 2002, un contrat enregistré sous le n° CTR2002120 et de type tout risque sur le véhicule impliqué dans le sinistre et immatriculé 123 ABC 75. Ce véhicule est une Renault Espace de 100 chevaux. M. Oliveau habite à Paris, 25 rue de Rivoli, dans le 1^{er} arrondissement. Il est joignable au 01 42 23 12 20 chez lui, ou au 06 12 12 12 12 sur son portable, ou au 01 23 43 23 34 à son travail. Vous voudrez bien prendre contact avec lui pour convenir d'un rendez-vous en vue de l'expertise que vous voudrez bien effectuer au garage « Saroul », 45, rue Rousseau, 75002 Paris, dans un délai de 15 jours.

Dans l'attente de votre rapport d'expertise, nous vous prions de croire, Monsieur l'expert, à nos sentiments distingués.

M. Durant, Agent d'assurance

On ajoute les spécifications suivantes : un accident est expertisé une ou plusieurs fois. En cas de ré-expertise, on change forcément d'expert. Un accident est suivi par l'agent responsable du contrat. Pour un accident, on enregistre aussi, si nécessaire, la liste de tous les véhicules impliqués un accident concerne donc des véhicules assurés ou pas par ASSURAUTO. Le contrat d'assurance est passé par un client et un seul et concerne une voiture et une seule.

8 Les enchères

On souhaite développer un système d'enchères électroniques. Les premières fonctionnalités décrites sont les suivantes :

Tout utilisateur du système (vendeur ou acheteur) doit préalablement s'inscrire en fournissant une adresse email qui lui servira d'identificateur unique pour ses futures interventions, un mot de passe choisi par lui qui lui servira d'authentification pour ses futures interventions et des renseignements d'état-civil : nom, prénom, adresse.

Les utilisateurs peuvent mettre en vente des objets. Pour cela, ils doivent fournir leur identificateur, leur mot de passe, la catégorie de l'objet pour une recherche par thème, un intitulé court de l'objet, une description détaillée de l'objet avec ses défauts et ses qualités, une mise à prix en euros, la date de début de l'enchère, la durée de l'annonce en jours, une ou plusieurs photos, s'il le souhaite. Le système fournit en retour un numéro de lot unique pour l'objet à vendre.

Un utilisateur ayant proposé un objet à la vente peut réviser la mise à prix à la baisse tant qu'il n'y a aucunes enchères sur l'objet.

Les utilisateurs peuvent effectuer des recherches parmi les objets mis en vente selon les critères suivants : catégorie de l'objet, mots de la description de l'objet, fourchette de prix.

Un utilisateur peut enchérir sur un objet. Pour être valable, l'enchère doit être supérieure de 0,5 euro à la plus haute enchère déjà effectuée ou à la mise à prix si c'est la première enchère ; l'annonce ne doit pas être expirée ; l'utilisateur ne doit pas être le vendeur de l'objet. Pour suivre l'évolution de l'enchère, le système garde aussi la date de chaque enchère.

Quand l'échéance de fin d'enchère tombe, le système informe le vendeur dans tous les cas, et l'acheteur s'il y a lieu, du résultat de l'enchère.

Si un vendeur veut remettre en vente un objet non vendu, il doit créer un nouvel objet dans le système avec ses caractéristiques propres.

Par ailleurs, les utilisateurs peuvent aussi sélectionner certains objets dont ils veulent suivre l'enchère. Le système enverra un mail d'alerte à l'utilisateur 24 heures avant la fin de l'enchère. L'utilisateur pourra supprimer les éléments de sa sélection.

9 Centres d'animation

La Mairie de Paris veut gérer les membres de ses centres d'animations et les animations qu'ils pratiquent. La Mairie possède plusieurs centres. Ils ont un nom, une adresse, un arrondissement. Chaque centre propose plusieurs animations. Une animation peut être proposée par plusieurs centres. Les animations ont un intitulé, un coût semestriel et un public (enfants, adulte, tout public). Un membre ne peut s'inscrire que dans un centre. Un membre peut pratiquer plusieurs activités dans plusieurs centres différents.

Série 1 bis – Conception spéciale

1 Le cyclisme

Cet exercice suit un vieux principe de logique appelé : « Le rasoir d'Occam » (1287-1349) et qui dit : Entia non sunt multiplicanda praeter necessitatem, c'est-à-dire : il ne faut pas multiplier les entités plus que nécessaire.

Une entreprise de fabrication de produits nécessaires pour le cyclisme veut automatiser la gestion de son stock et de ses coûts de fabrication. Elle a besoin de connaître à tout moment la quantité en stock de chacun de ses produits, la composition détaillée de chacun des produits composés, etc.

La liste de ses produits se présente comme suit :

1. Un vélo est composé de deux roues, d'un guidon, de deux gardes boue, d'une chaîne, d'une selle, etc.
2. Un tricycle est composé de trois roues, d'un guidon, de trois gardes boue, d'une chaîne, d'une selle, etc.
3. Une roue est composée de cinquante rayons, d'une jante, d'une chambre à air, etc.
4. Etc.

2 Roland Garros

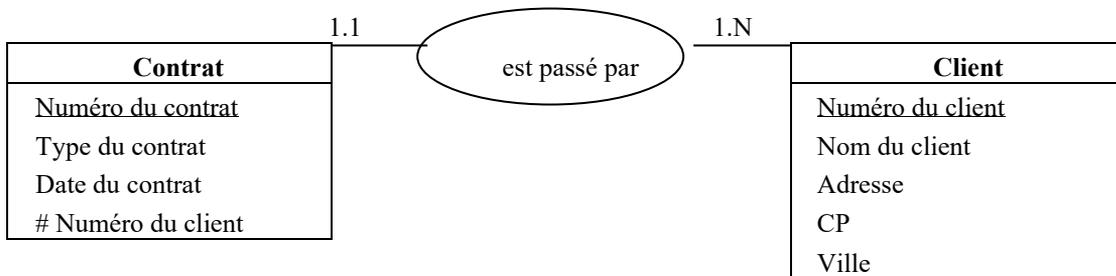
On veut enregistrer et archiver les données d'une compétition de tennis. Les matchs ont lieu à une certaine date, sur un certain court et correspondent à un certain niveau dans la compétition (finale, demi, quart, etc.).

On veut pouvoir connaître les résultats de matchs et pouvoir compter le nombre de jeux et de sets pour chaque match et pour chaque joueur.

Série 2 – Critiques et corrections

1 Contrat et client

Soit le modèle entité-association suivant :



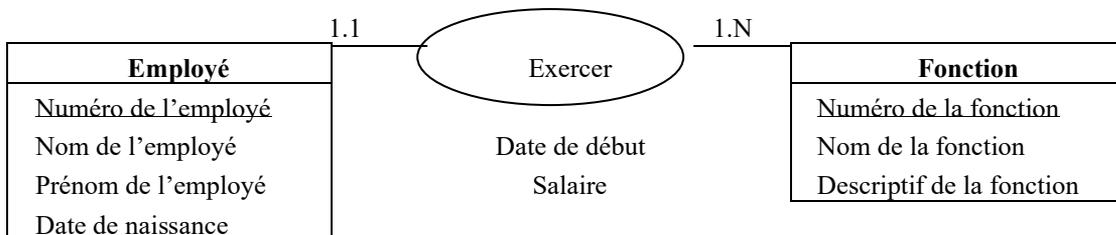
Un contrat est passé par un client et un seul. Un client peut passer plusieurs contrats.

Quelle critique formelle pouvez-vous faire au MEA présenté ci-dessous ?

Proposez un modèle corrigé.

2 Employé et fonction

Soit le modèle entité-association suivant :

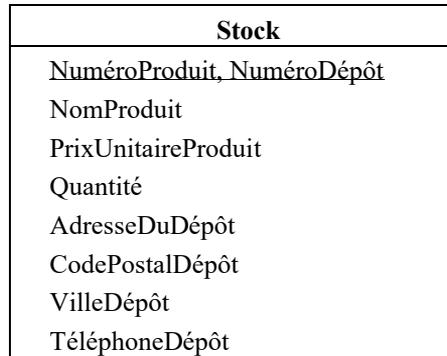


Un employé peut exercer plusieurs fonctions successivement avec différents salaires, mais pas en même temps.

- Quelle critique formelle (= syntaxique, = non sémantique = est-ce que ça « compile » ?) pouvez-vous faire au MEA présenté ci-dessous ?
- Proposez un modèle corrigé.
- Proposez le MR correspondant et le graphe des tables.

3 Stock

Soit le modèle entité-association suivant :

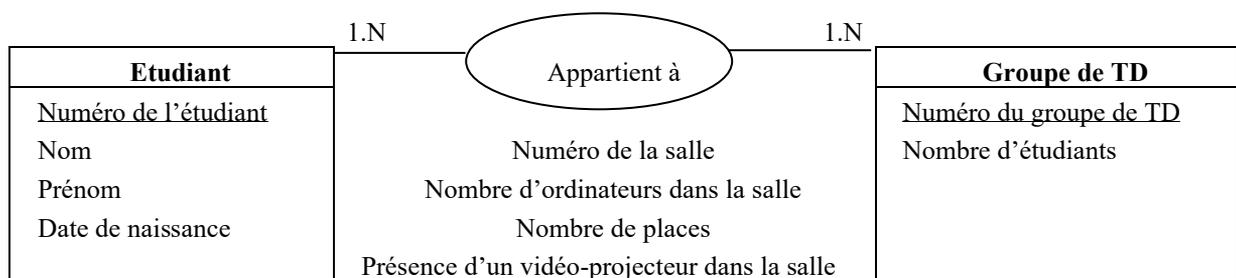


La quantité correspond à la quantité de produit stocké pour un numéro de produit et un numéro de dépôt. Dans un dépôt, il peut y avoir plusieurs produits. Un dépôt peut être vide. Un produit peut être stocké dans plusieurs dépôts. Un produit peut être en rupture de stock.

- Quelle critique formelle (=syntaxique, = non sémantique = est-ce que ça « compile » ?) pouvez-vous faire au MEA présenté ci-dessous ?
- Proposez un modèle corrigé.
- Proposez le MR correspondant et le graphe des tables.

4 Etudiant et Groupe de TD

Soit le modèle entité-association suivant :



Un étudiant appartient à un groupe de TD et un seul. Un groupe de TD est associé à une salle et une seule.

- Quelle critique formelle (=syntaxique, = non sémantique = est-ce que ça « compile » ?) pouvez-vous faire au MEA présenté ci-dessous ?
- Proposez un modèle corrigé.
- Proposez le MR correspondant et le graphe des tables.

Série 3 – Conception avancée

Pour cette série, on suit les mêmes étapes que pour la série 1.

1 Les chantiers d'été

Une association organise des chantiers d'été. Il y en a de 3 sortes : chantiers de réhabilitation, chantiers de construction ou chantiers archéologiques. Pour les chantiers de construction et de réhabilitation, on connaît le type de bâtiment. Pour les chantiers de réhabilitation, on connaît la date de création de l'ancien bâtiment et des informations sur l'ancien bâtiment. Pour les chantiers archéologiques, on connaît la période archéologique et la surface du chantier (qui peut être très grande). Les chantiers ont une date de début et une date de fin (qui peut être plusieurs années après la date de fin). Des équipes viennent sur le chantier. Une équipe est constituée pour une durée limitée (environ un mois pendant l'été). Elle est affectée à un chantier et un seul. Les équipes sont constituées de participants. Les participants aux chantiers sont des personnes. Leur participation à l'équipe peut être inférieure à la durée de vie de l'équipe elle-même. Les personnes ont un nom, un prénom, une adresse, un email et un téléphone.

L'association peut recevoir des dons de personnes. Les dons ont un montant et une date de don. Ils sont faits par une personne. Ils donnent lieu à un reçu fiscal envoyé à une certaine date.

L'association envoie des courriers aux personnes. Les courriers ont un titre, une date et un libellé.

L'association gère aussi des membres. Les membres payent des cotisations annuelles. Les cotisations ne donnent pas lieu à un reçu fiscal. On distingue donc deux types de dons : ceux avec reçu et les cotisations.

Les membres de l'association peuvent participer au conseil d'administration (CA). Le CA est renouvelé en général tous les ans. Les participants au CA ont une fonction particulière. Cette fonction peut changer en cours de CA. On conserve les dates de début et de fin de chaque fonction occupée. Chaque CA a un certain nombre de participants.

1) On suit les mêmes étapes que pour la série 1 pour le sujet suivant :

Un réseau de chenils souhaite informatiser son activité et vous demande de créer un modèle conceptuel de sa future base de données. Un chenil est un établissement destiné à l'élevage ou à la pension des chiens. Dans ce réseau, les chenils s'occupent des chiens de leur naissance jusqu'à la fin de leur apprentissage. L'identité du propriétaire étant privée, aucun chien ne sera relié à un propriétaire dans cette base de données qui sert aussi de vitrine sur le savoir-faire du réseau.

Un chenil du réseau possède un nom unique permettant de l'identifier, un nom de contact ainsi qu'un numéro de téléphone, et une description de ses activités. Chaque chien est identifié par son nom et celui du chenil qui gère son apprentissage ; il possède une date de naissance et est associé à une race. Certains des chiens suivront un apprentissage plus spécifique : ils deviendront guides d'aveugle, chiens de garde ou chiens de course. Chaque chien de garde possède une spécialité : attaque, défense, pistage ou détection. Un chien de course quand à lui possède une vitesse maximum mesurée.

Le chenil gère également l'historique des poids de chaque animal, afin d'enregistrer régulièrement la courbe de croissance. Pour cela chaque relevé de poids est identifié par le nom du chien et la date du relevé.

Les chiens aux parcours plus spécifiques (guides, chiens de garde ou chiens de course) sont liés à des entités qui les géreront après l'apprentissage. Ces entités sont identifiées par un nom, et possèdent une description ainsi qu'un nom de contact avec un numéro de téléphone. Les chiens guides d'aveugle sont ainsi gérés par une association qui propose ensuite les chiens aux personnes malvoyantes. Les chiens de gardes sont utilisés par des entreprises (le chenil souhaite pouvoir ajouter une description de l'activité principale de chaque entreprise). Et un chien de course est possédé par une écurie (qui a une date de création). Bien sûr, chacune de ces entités peut gérer plusieurs animaux.

2) On suit les mêmes étapes que pour la série 1 pour le sujet suivant. On distinguera bien les nouveautés par rapport à l'exercice précédent.

Le réseau souhaiterait compléter cette première modélisation en ajoutant des informations sur les chiens de course, pour améliorer sa publicité. L'objectif est de garder en mémoire pour chaque épreuve sportive auquel participe un chien de course son numéro de dossard (de 1 à 6), sa position à l'issue de l'épreuve (de 1 à 6), son temps de course, et s'il a abandonné ou non.

Une épreuve est identifiée par la date et le nom du tournoi à laquelle elle appartient (un tournoi est identifié uniquement par son nom). Il y a toujours 6 chiens qui courent dans une épreuve, mais ils ne proviennent pas tous d'un des chenils du réseau (nous ne gérons pas les

chiens en dehors du réseau). Chaque épreuve a lieu dans un cynodrome identifié par son nom, et chaque cynodrome se situe dans un département, identifié par son numéro et possédant un nom.

Série 4 – A partir du MR

1 La clinique vétérinaire

Animal(Id, nom, DateNaissance, DateDécès, idPropriétaire)

Chien (idAnimal, taille, poids, idRace)

Chat (idAnimal, idRace)

Race_chien (id, nom)

Race_chat (id, nom)

Propriétaire (id, nom, prenom, rue, codepostal, ville, telephone, telephoneMobile)

Visite (id, dateVisite, heureVisite, raison, idDossier, idAnimal, idVeterinaire)

Dossier (id, antecedents)

Veterinaire (id, nom, prenom, telephone, telephoneMobile)

Medicament (id, nom, dosage, indications, effetsSecondaires, laboratoire)

Prescrire (idVisite, idMedicament, posologie)

Horaire (id, jour, heureDebut, heureFin, idVeterinaire)

Garde (id, date, heure_debut, heure_fin)

Effectuer (idGarde, idVeterinaire)

Travail

GT, MR, MR valorisé, ontologie, MEA

Améliorer le modèle.

Série 5 – A partir du SQL

1. DDL Chantiers

Le DDL ci-dessous est associé aux règles de gestion suivantes :

- La BD gère des visites de chantier
- Une visite est faite dans une voiture. Un employé est conducteur.
- D'autres employés sont visiteurs.
- Une visite peut faire intervenir plusieurs voitures.
- Il ne peut y avoir qu'une visite par jour pour un chantier.
- Un véhicule a une capacité, qui exclut le conducteur

- 1) Faire le Graphe des Tables (de façon arborescence, flèches vers le bas autant que possible, en évitant les croisements de branches autant que possible).
- 2) Faire le MEA correspondant au modèle.
- 3) Combien y a-t-il de tuples dans la table « VISITE » après l'exécution de la requête suivante exécutée juste après l'exécution du script de création des tables et des tuples. Vous devez donner le nombre de tuples et justifier votre réponse.

```
INSERT INTO visite (n_chantier, n_vehicule, date, n_conducteur)
VALUES ('CH4', 'V2', '2018-04-02 09:00:00', 'E1');
```

- 4) Idem pour la requête :

```
INSERT INTO visite (n_chantier, n_vehicule, date, n_conducteur)
VALUES ('CH4', 'V2', '2018-04-10 09:00:00', 'E11');
```

- 5) Combien y a-t-il de tuples dans la table « CHANTIER » après l'exécution de la requête suivante exécutée juste après l'exécution du script de création des tables et des tuples. Vous devez donner le nombre de tuples et justifier votre réponse.

```
DELETE FROM chantier
WHERE n_chantier = 'CH4';
```

- 6) Combien de tuples ont-ils été modifiés dans la table « VISITE » après l'exécution de la requête suivante exécutée juste après l'exécution du script de création des tables et des tuples. Vous devez donner le nombre de tuples modifiés et justifier votre réponse.

```
UPDATE visite
SET date_heure = '2018-04-03 14:00:00'
WHERE n_chantier = 'CH4'
AND n_vehicule = 'V5';
```

```
-- Creation des tables

CREATE TABLE employe (
    n_emp          VARCHAR(4) PRIMARY KEY,
    nom_emp        VARCHAR(20) not null,
    qualif_emp     VARCHAR(12) not null
) type innodb;

CREATE TABLE chantier (
```

```

    n_chantier    VARCHAR(10) PRIMARY KEY,
    nom_ch        VARCHAR(10) not null,
    adresse_ch    VARCHAR(15) not null,
    kilometres    INTEGER not null
) type innodb;

CREATE TABLE vehicule (
    n_vehicule    VARCHAR(10) PRIMARY KEY,
    type_vehicule VARCHAR(1) not null,
    capacite      INTEGER not null check(capacite >0)
) type innodb;

CREATE TABLE visite(
    n_chantier    VARCHAR(10),
    n_vehicule    VARCHAR(10),
    date_heure    DATETIME not null,
    n_conducteur  VARCHAR(4) not null,
    FOREIGN KEY(n_chantier) REFERENCES chantier(n_chantier),
    FOREIGN KEY(n_vehicule) REFERENCES vehicule(n_vehicule),
    FOREIGN KEY(n_conducteur) REFERENCES employe(n_emp),
    PRIMARY KEY(n_chantier,n_vehicule, date_heure)
) type innodb;

CREATE TABLE transporter (
    n_chantier VARCHAR(10),
    n_vehicule VARCHAR(10),
    date_heure DATETIME,
    n_transporte VARCHAR(4) not null,
    PRIMARY KEY(n_chantier,n_vehicule,date_heure,n_transporte),
    FOREIGN KEY(n_chantier,n_vehicule,date_heure)
        REFERENCES visite(n_chantier,n_vehicule,date_heure),
    FOREIGN KEY(n_transporte) REFERENCES employe(n_emp)
) type innodb;

-----
-- 10 tuples "employe"

INSERT INTO employe VALUES('E1','Victor','OS');
INSERT INTO employe VALUES('E2','Henri','OS');
INSERT INTO employe VALUES('E3','Michel','Assistant');
INSERT INTO employe VALUES('E4','Pierre','Assistant');
INSERT INTO employe VALUES('E5','Paul','Architecte');
INSERT INTO employe VALUES('E6','Fred','Ingénieur');
INSERT INTO employe VALUES('E7','Alain','Ingénieur');
INSERT INTO employe VALUES('E8','Jack','Ingénieur');
INSERT INTO employe VALUES('E9','Fred','Ingénieur');
INSERT INTO employe VALUES('E10','Christian','OS');

-----
-- 4 tuples "chantier"

INSERT INTO chantier VALUES('CH1','IUT','Blagnac', 40);
INSERT INTO chantier VALUES('CH2','UPS','Toulouse', 80);
INSERT INTO chantier VALUES('CH3','UTM','Toulouse', 70);
INSERT INTO chantier VALUES('CH4','INSA','Castanet', 50);

-----
-- 5 tuples "vehicule"

INSERT INTO vehicule VALUES('V1','0', 2);
INSERT INTO vehicule VALUES('V2','2', 6);
INSERT INTO vehicule VALUES('V3','1', 3);
INSERT INTO vehicule VALUES('V4','2', 1);
INSERT INTO vehicule VALUES('V5','2', 1);

```

```

-----
-- 14 tuples "visite"

INSERT INTO visite VALUES('CH1','V1','2018-04-01 09:00:00','E1');
INSERT INTO visite VALUES('CH2','V1','2018-04-01 14:00:00','E1');
INSERT INTO visite VALUES('CH3','V1','2018-04-01 16:00:00','E1');
INSERT INTO visite VALUES('CH2','V2','2018-04-01 09:00:00','E2');
INSERT INTO visite VALUES('CH4','V2','2018-04-01 15:00:00','E2');

INSERT INTO visite VALUES('CH2','V1','2018-04-02 09:00:00','E2');
INSERT INTO visite VALUES('CH4','V1','2018-04-02 14:00:00','E2');
INSERT INTO visite VALUES('CH1','V2','2018-04-02 09:00:00','E10');
INSERT INTO visite VALUES('CH2','V2','2018-04-02 13:30:00','E10');
INSERT INTO visite VALUES('CH3','V4','2018-04-02 09:00:00','E5');
INSERT INTO visite VALUES('CH4','V2','2018-04-02 17:00:00','E10');

INSERT INTO visite VALUES('CH3','V2','2018-04-03 09:00:00','E1');
INSERT INTO visite VALUES('CH2','V5','2018-04-03 09:00:00','E3');
INSERT INTO visite VALUES('CH4','V5','2018-04-03 14:15:00','E3');

-----
-- 18 tuples "transporter"

INSERT INTO transporter VALUES ('CH1','V1','2018-04-01
    09:00:00','E7');
INSERT INTO transporter VALUES ('CH1','V1','2018-04-01
    09:00:00','E8');
INSERT INTO transporter VALUES ('CH2','V1','2018-04-01
    14:00:00','E7');
INSERT INTO transporter VALUES ('CH3','V1','2018-04-01
    16:00:00','E8');

INSERT INTO transporter VALUES ('CH4','V2','2018-04-01
    15:00:00','E9');
INSERT INTO transporter VALUES ('CH2','V2','2018-04-01
    09:00:00','E9');

INSERT INTO transporter VALUES ('CH1','V2','2018-04-02
    09:00:00','E1');
INSERT INTO transporter VALUES ('CH1','V2','2018-04-02
    09:00:00','E2');
INSERT INTO transporter VALUES ('CH1','V2','2018-04-02
    09:00:00','E3');
INSERT INTO transporter VALUES ('CH2','V2','2018-04-02
    13:30:00','E1');
INSERT INTO transporter VALUES ('CH2','V2','2018-04-02
    13:30:00','E2');
INSERT INTO transporter VALUES ('CH4','V2','2018-04-02
    17:00:00','E3');
INSERT INTO transporter VALUES ('CH4','V1','2018-04-02
    14:00:00','E8');
INSERT INTO transporter VALUES ('CH2','V1','2018-04-02
    09:00:00','E7');
INSERT INTO transporter VALUES ('CH3','V4','2018-04-02
    09:00:00','E9');

INSERT INTO transporter VALUES ('CH2','V5','2018-04-03
    09:00:00','E7');
INSERT INTO transporter VALUES ('CH3','V2','2018-04-03
    09:00:00','E2');
INSERT INTO transporter VALUES ('CH4','V5','2018-04-03
    14:15:00','E9');

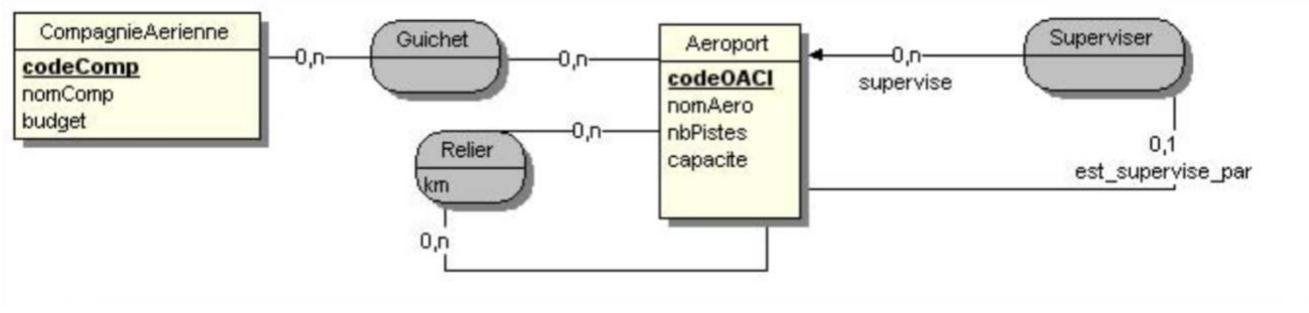
```

Série 6 - départ MEA

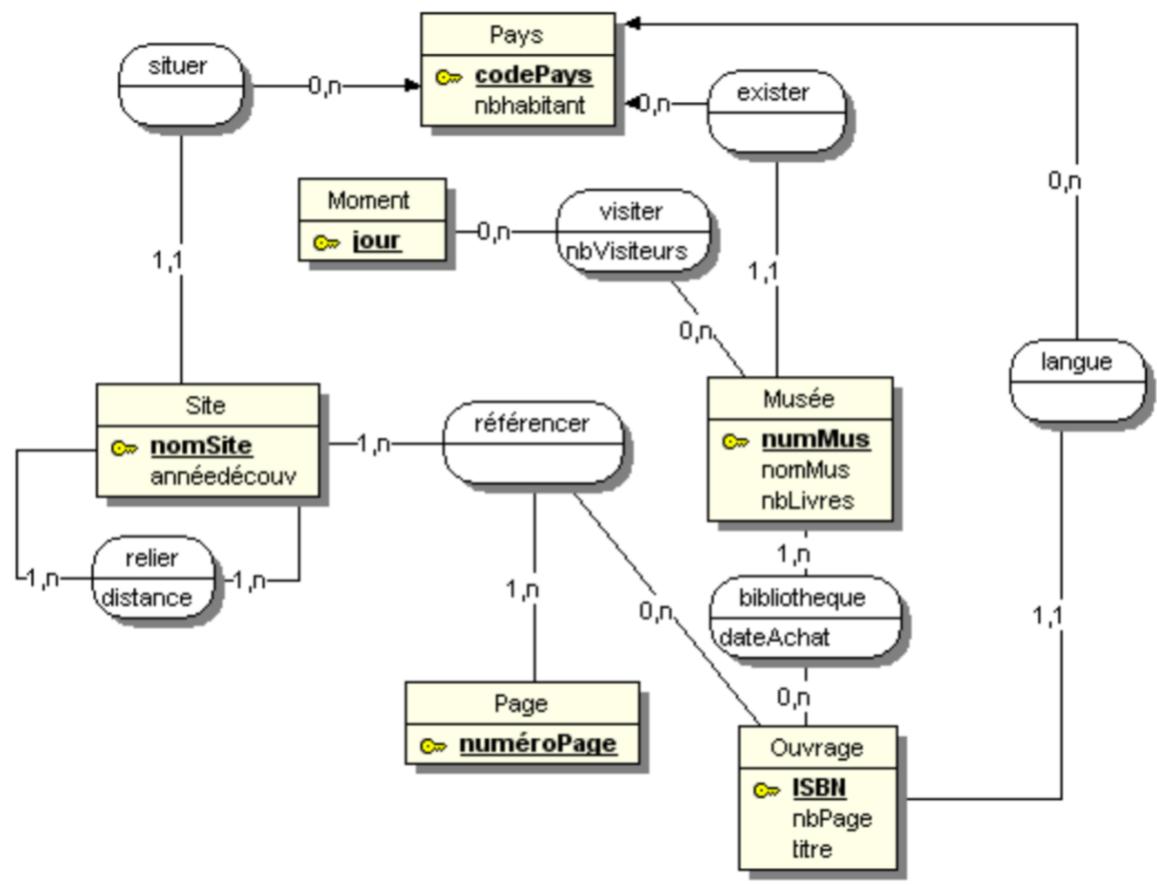
Pour cette série, on suit les mêmes étapes que pour la série 1.

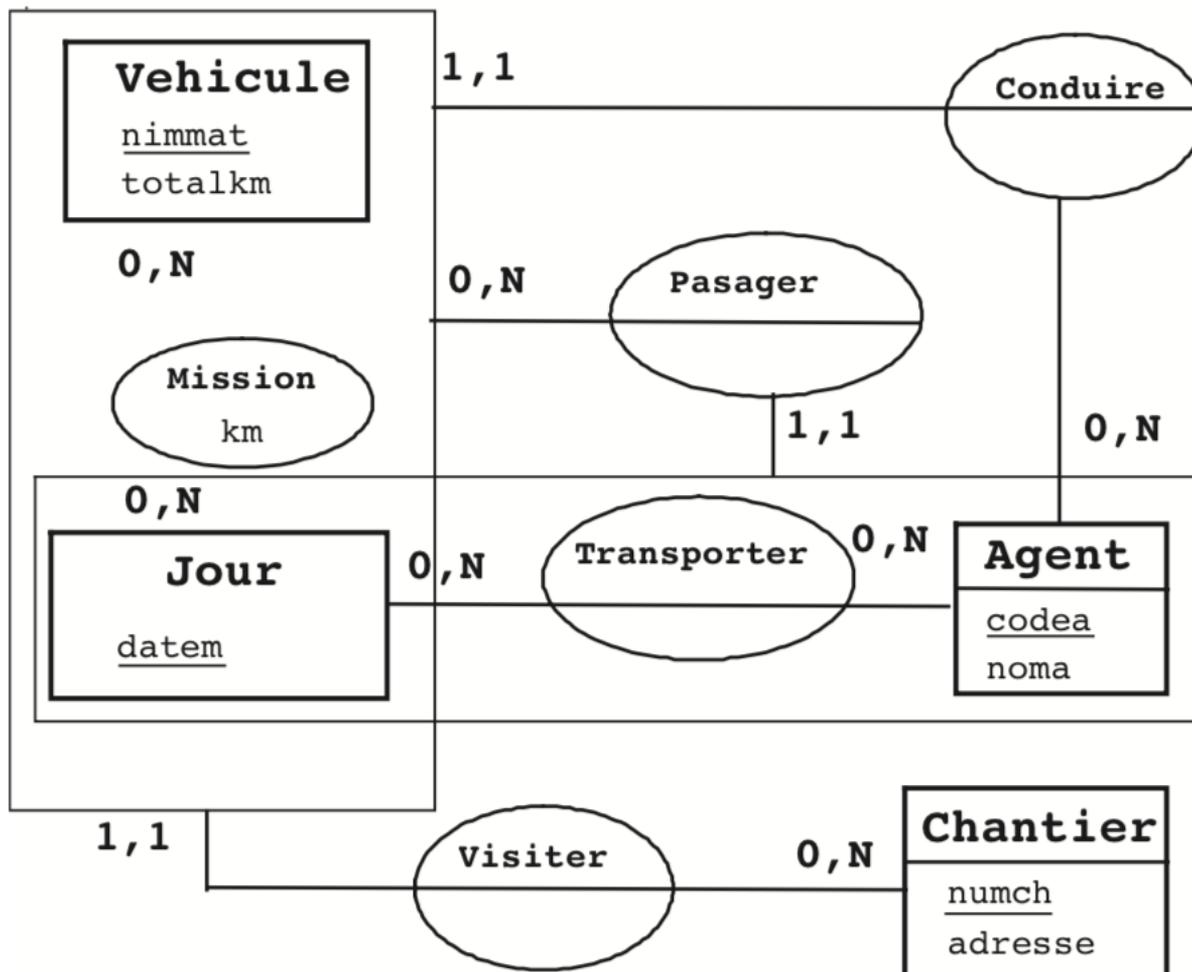
Les modèles proviennent pour la plupart de « UML 2 pour les bases de données – Christian Sautou ».

1 MEA-171



2 MCD-173

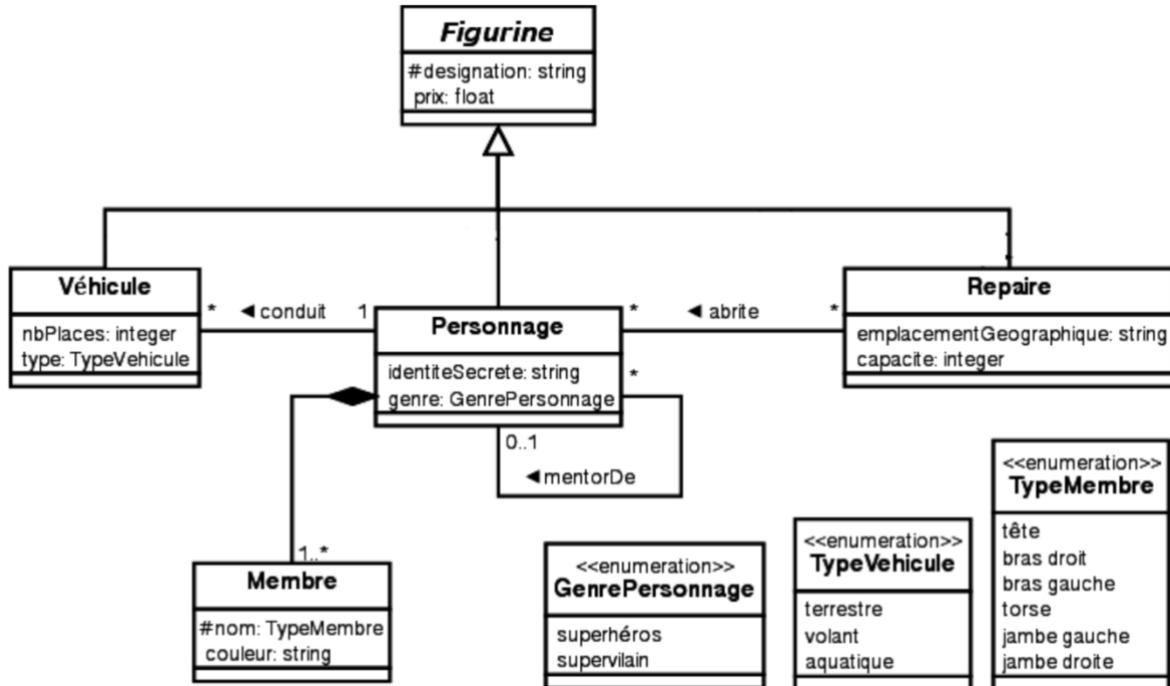




Série 7 – Départ UML

Pour cette série, on suit les mêmes étapes que pour la série 1. L'objectif n°1 est de produire un MR textuel, puis un MEA. Les modèles proviennent pour la plupart de « UML 2 pour les bases de données – Christian Sautou ».

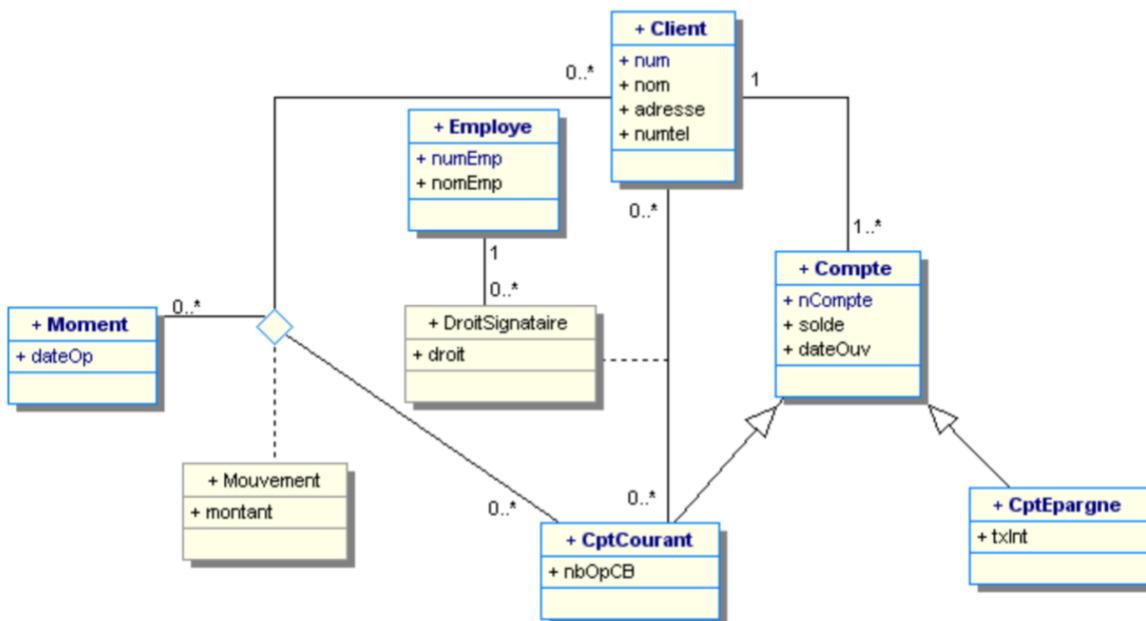
1. UML-utc



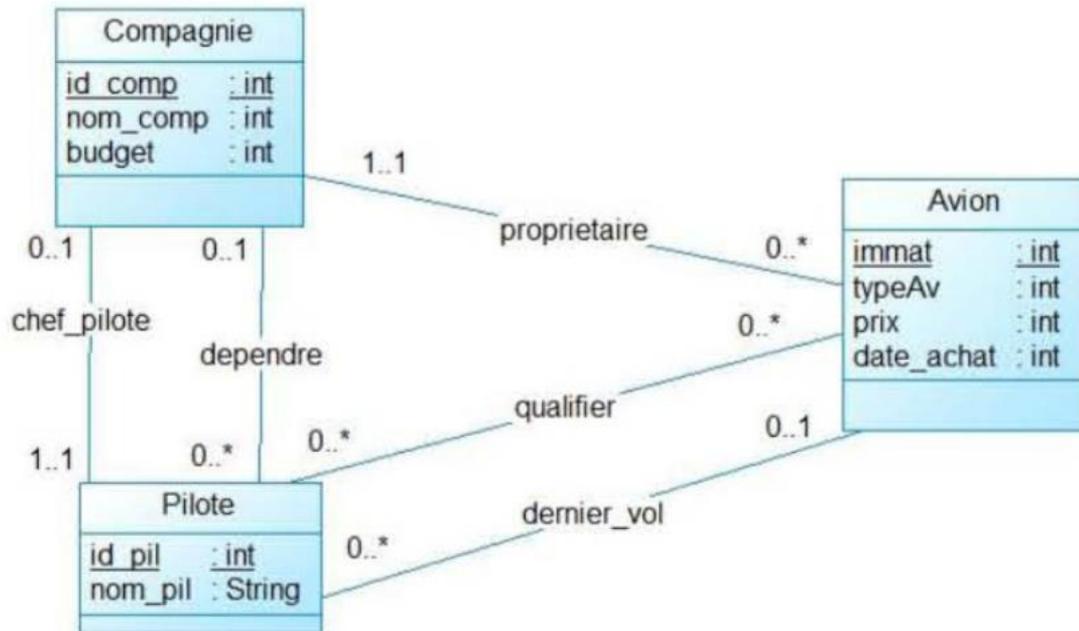
2. UML-174

Décrire un schéma relationnel à partir de la modélisation UML suivante. On suppose qu'un compte courant ne peut pas être rémunéré.

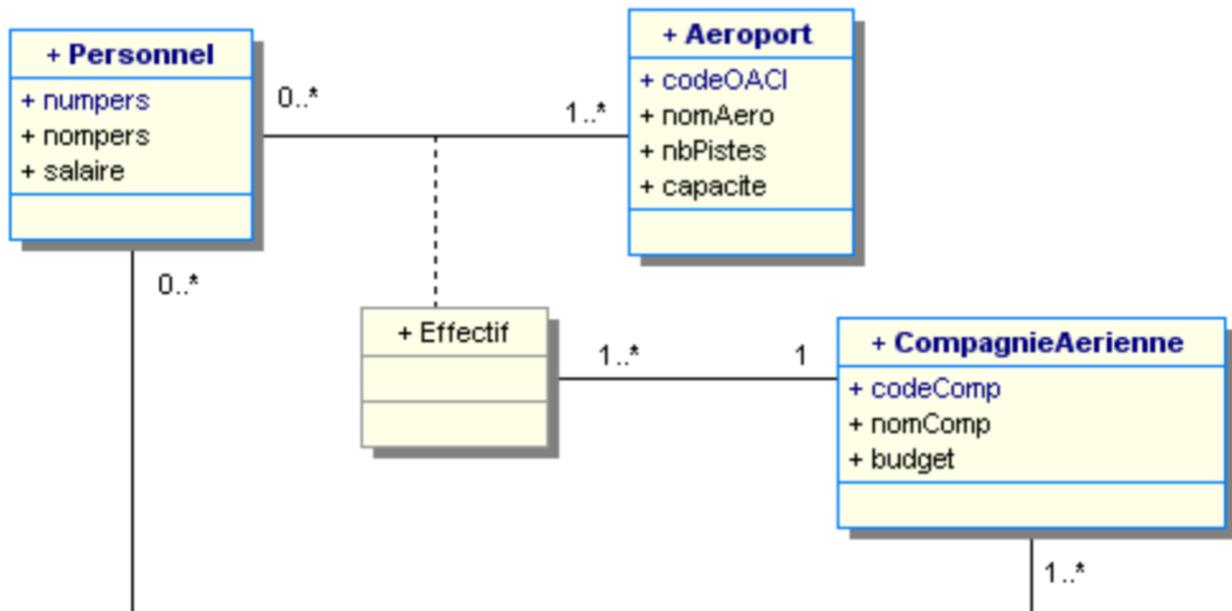
Figure 2-90 Association n-aire UML



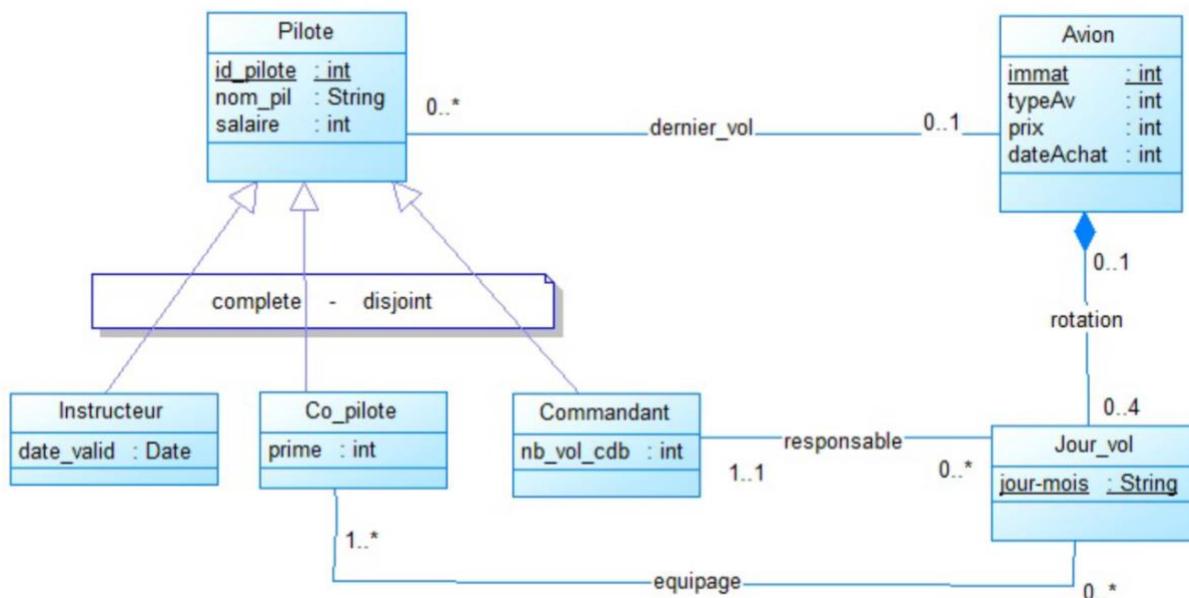
3. UML-147



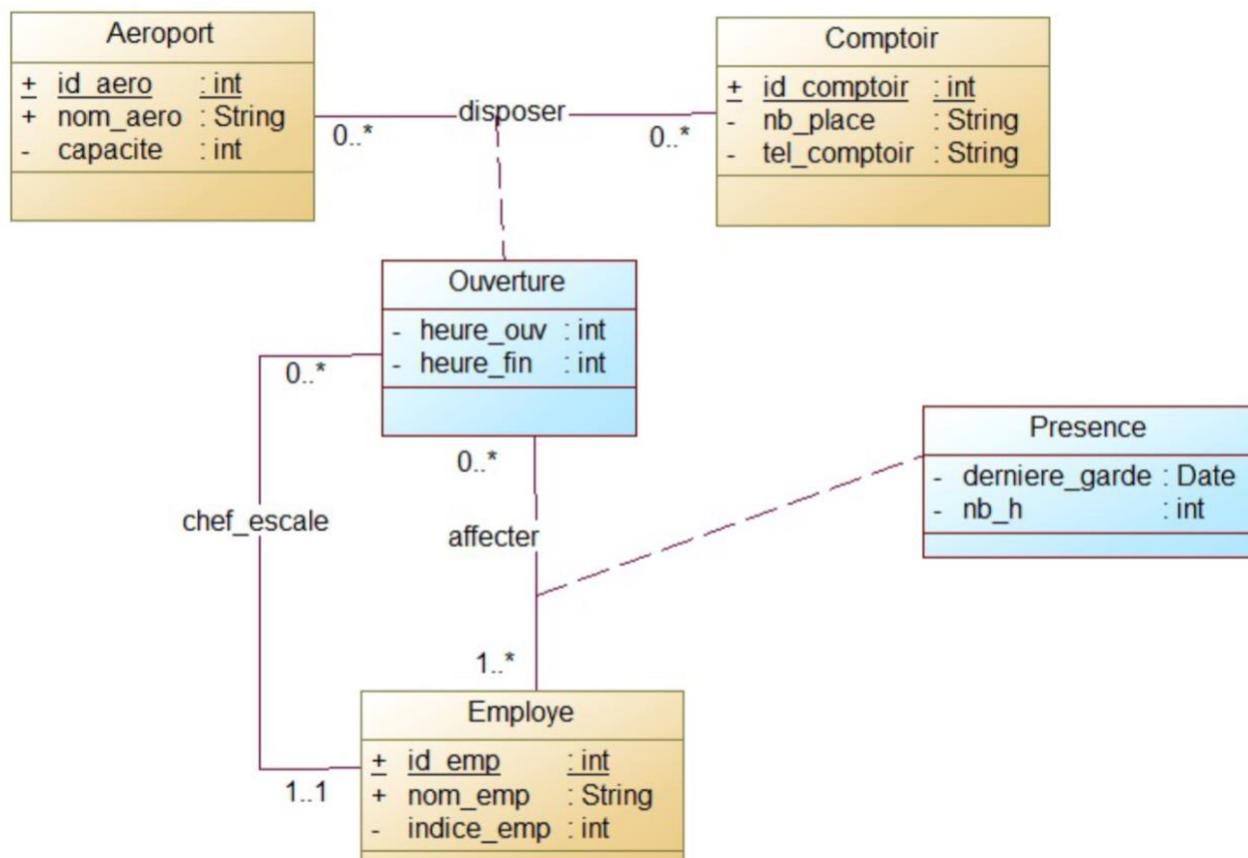
4. UML-172



5. UML-147



6. UML-148



7. UML-149

