

2014- 2015

Projet commun

HAMONNAIS Raphael
COTTIN Thomas

[OUTIL D'EXERCICE A LA METHODE MERISE]

[Tapez le résumé du document ici. Il s'agit généralement d'une courte synthèse du document. Tapez le résumé du document ici. Il s'agit généralement d'une courte synthèse du document.]

Introduction : présentation générale du projet.

Le projet a pour but de familiariser l'utilisateur (étudiant) à la méthode d'analyse MERISE. Tout en lui proposant différents exercices de difficulté variable, il couvrira deux aspects fondamentaux de la méthode, à savoir le Dictionnaire de Données et le Modèle Entités Associations.

Via une interface WEB, l'utilisateur se voit présenté une liste d'exercices, chacun possédant son propre énoncé. Lorsqu'il en effectue un, une liste conséquente d'entités et de rubriques lui est proposée. Il lui faut donc choisir lesquels garder.

Dans la partie Dictionnaire de Données, l'utilisateur doit sélectionner le type de rubrique (attribut, calculée ou paramètre) ainsi que le type de donnée (integer, varchar, ...) pour chacune des rubriques précédemment sélectionnée.

Pour la partie Modèle Entités Associations, l'utilisateur doit sélectionner et insérer des attributs dans les entités de son choix puis créer des associations en spécifiant les cardinalités minimum et maximum (choix dans liste déroulante).

Ces deux parties sont soumises à une correction automatique, délivrant à l'élève un score sur cet exercice. A lui de réessayer plus tard s'il le souhaite afin de l'améliorer.

Table des matières

I.	Partie ACSI	2
a.	Le dictionnaire des données	2
b.	Le MEA	3
c.	Modèle de contexte	4
d.	Modèle conceptuel de traitement analytique	5
e.	Cycle de vie des objets	9
II.	Partie base de donnée	11
a.	Explications liées au passage MEA / schéma relationnel et commentaires concernant le modèle relationnel	11
b.	Commentaires sur le script de création	12
c.	Commentaires sur le script de remplissage	12
d.	Scenarii d'utilisation	12

I. Partie ACSI

a. Le dictionnaire des données

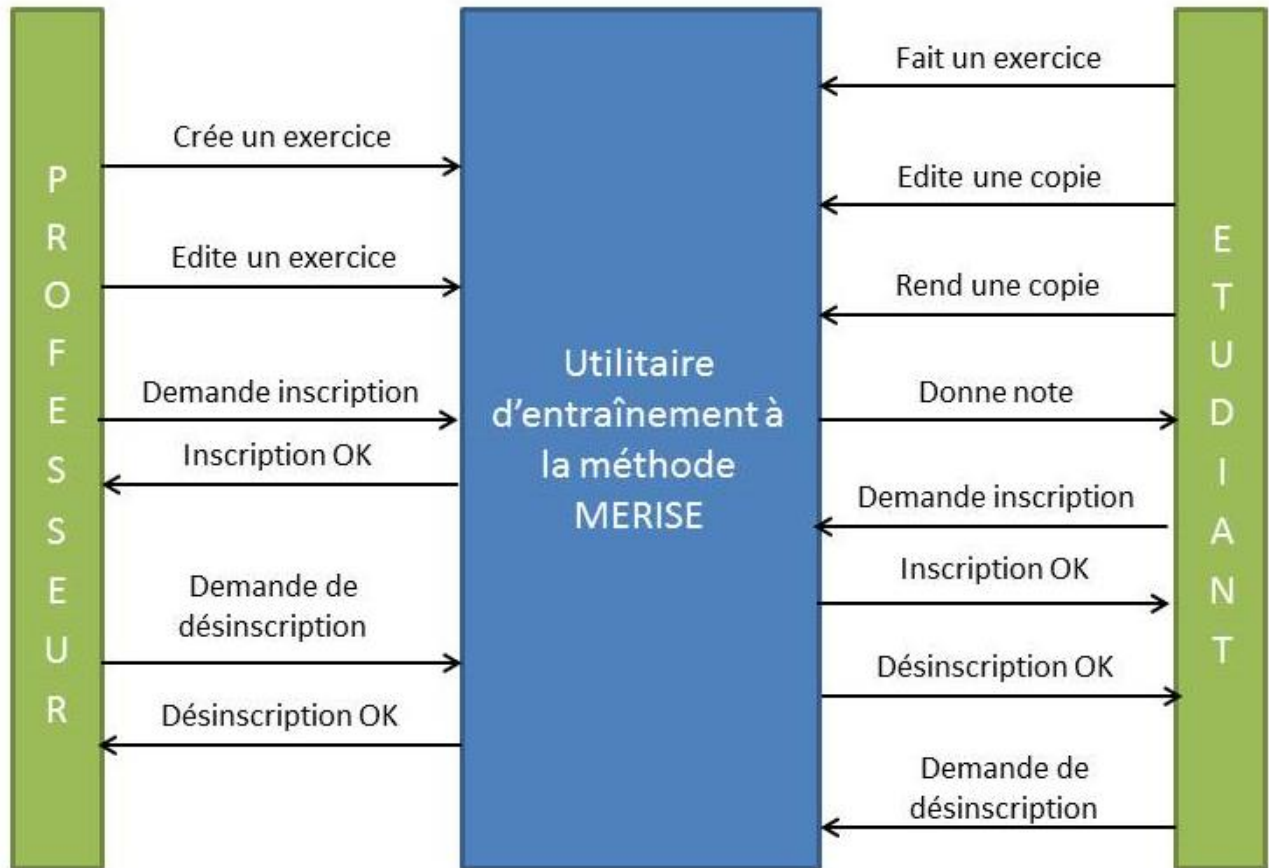
Nom rubrique	Type rubrique	Type donnée	Commentaire
ID_Utilisateur	Attribut	INTEGER	identifiant unique de l'utilisateur
Password	Attribut	VARCHAR	mot de passe (secret) de l'utilisateur. Nécessite cryptage.
Nom_Utilisateur	Attribut	VARCHAR	nom de l'utilisateur
Prenom_Utilisateur	Attribut	VARCHAR	prénom de l'utilisateur
Mail_Utilisateur	Attribut	VARCHAR	adresse email de l'utilisateur
ID_Groupe	Attribut	INTEGER	identifiant unique du groupe
Nom_Groupe	Attribut	VARCHAR	nom du groupe
Commentaire_Groupe	Attribut	TEXT	ajoute des informations utiles
ID_Promotion	Attribut	INTEGER	identifiant d'une promotion
Nom_Promotion	Attribut	VARCHAR	nom de la promotion
Annee_Promotion	Attribut	YEAR	année de la promotion
ID_Edition_Exercice	Attribut	INTEGER	identifiant de la table servant de log des éditions
Date_Edition_Exercice	Attribut	DATE	date de l'édition
ID_Type_Exercice	Attribut	INTEGER	identifiant unique du type d'exercice (DD ou MEA)
Commentaire_Type_Exercice	Attribut	TEXT	ajoute des informations utiles
ID_Exercice	Attribut	INTEGER	identifiant unique de l'exercice
Nom_Exercice	Attribut	VARCHAR	nom de l'exercice
Enonce_Exercice	Attribut	VARCHAR	chemin d'accès au fichier contenant l'énoncé
ID_MEA_Correction	Attribut	INTEGER	identifiant du MEA créé par le professeur, correction de l'exercice
ID_MEA_Fake	Attribut	INTEGER	identifiant du MEA contenant toutes les rubriques, même fausses
ID_DD_Correction	Attribut	INTEGER	identifiant du DD créé par le professeur, correction de l'exercice
ID_DD_Fake	Attribut	INTEGER	identifiant du DD contenant toutes les rubriques, même fausses
Exercice_Pret	Attribut	BOOLEAN	vrai quand l'exercice est validé par le professeur et donc visible
ID_MEA	Attribut	INTEGER	identifiant unique du MEA

ID_DD	Attribut	INTEGER	identifiant unique de DD
ID_Copie_Eleve	Attribut	INTEGER	identifiant unique d'une copie élève, CAD de la réalisation d'un exercice
Nom_Copie_Eleve	Attribut	VARCHAR	nom que l'élève peut donner à sa copie pour s'y retrouver
Date_Derniere_Modif_Copie_Eleve	Attribut	DATE	date de la dernière modification de l'exercice
Date_Envoi_Copie_Eleve	Attribut	DATE	date de l'envoi définitif de la copie
Commentaire_Copie_Eleve	Attribut	TEXT	s'il le souhaite, l'étudiant peut ajouter un commentaire
Note_Copie_Eleve	Attribut	FLOAT	note attribuée quand le rendu à été effectué
Nom_Entite	Attribut	VARCHAR	nom de l'entité
Nom_Association	Attribut	VARCHAR	nom de l'association
Libelle_Cardinalite	Attribut	VARCHAR	couple cardinalité min/max
Nom_Attribut	Attribut	VARCHAR	nom de l'attribut
Cle_Primaire	Attribut	BOOLEAN	défini si l'attribut est ou non PK
ID_Type_Donnee	Attribut	INTEGER	identifiant du type de donnée
Libelle_Type_Donnee	Attribut	VARCHAR	libelle du type de donnée (boolean, integer, varchar...)
Nom_Calculée	Attribut	VARCHAR	nom de la calculée
Nom_Parametre	Attribut	VARCHAR	nom du paramètre
Valeur	Attribut	VARCHAR	Contiens la valeur du paramètre
Commentaire_Rubrique	Attribut	TEXT	Commentaire de la rubrique

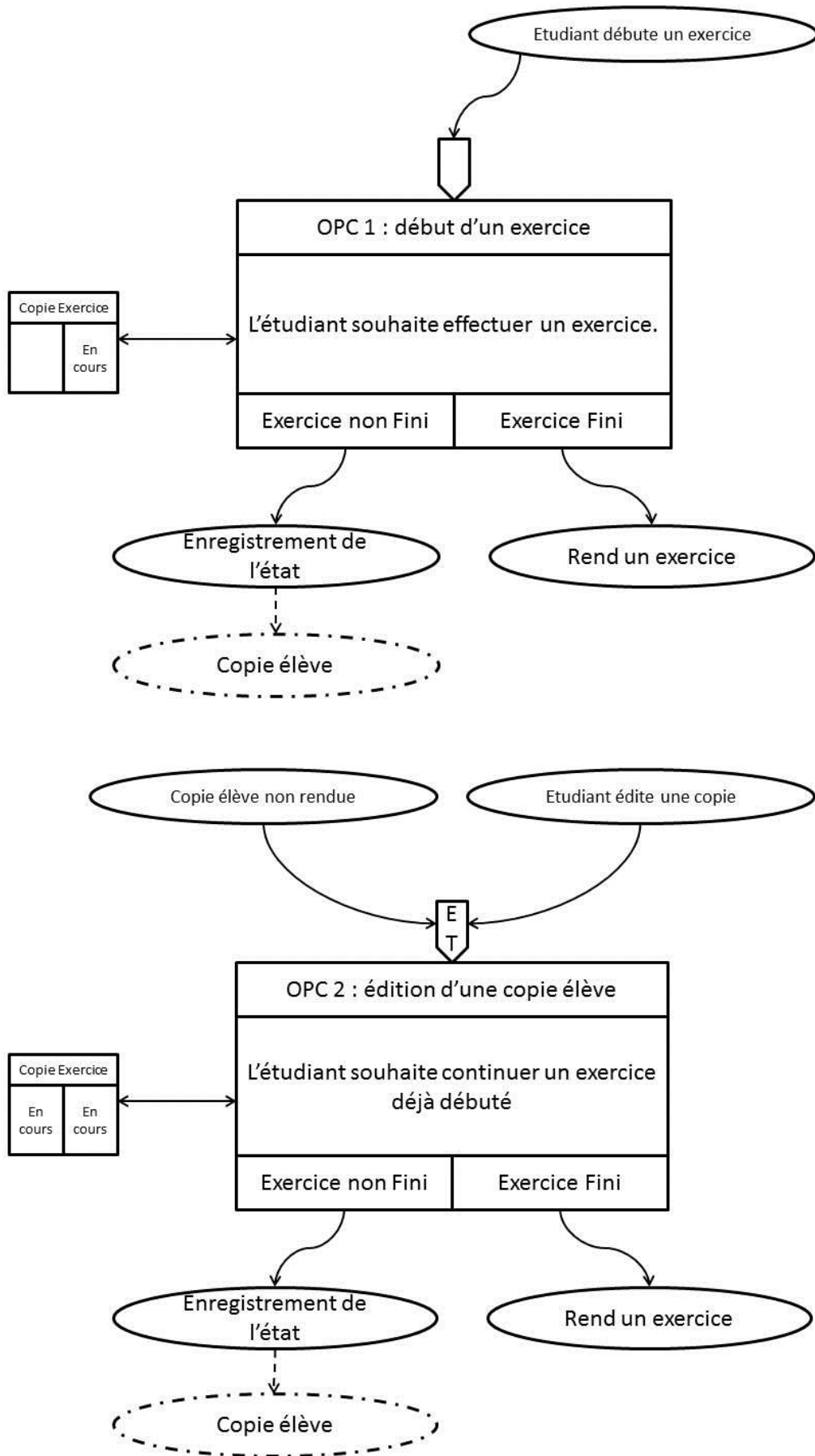
b. Le MEA

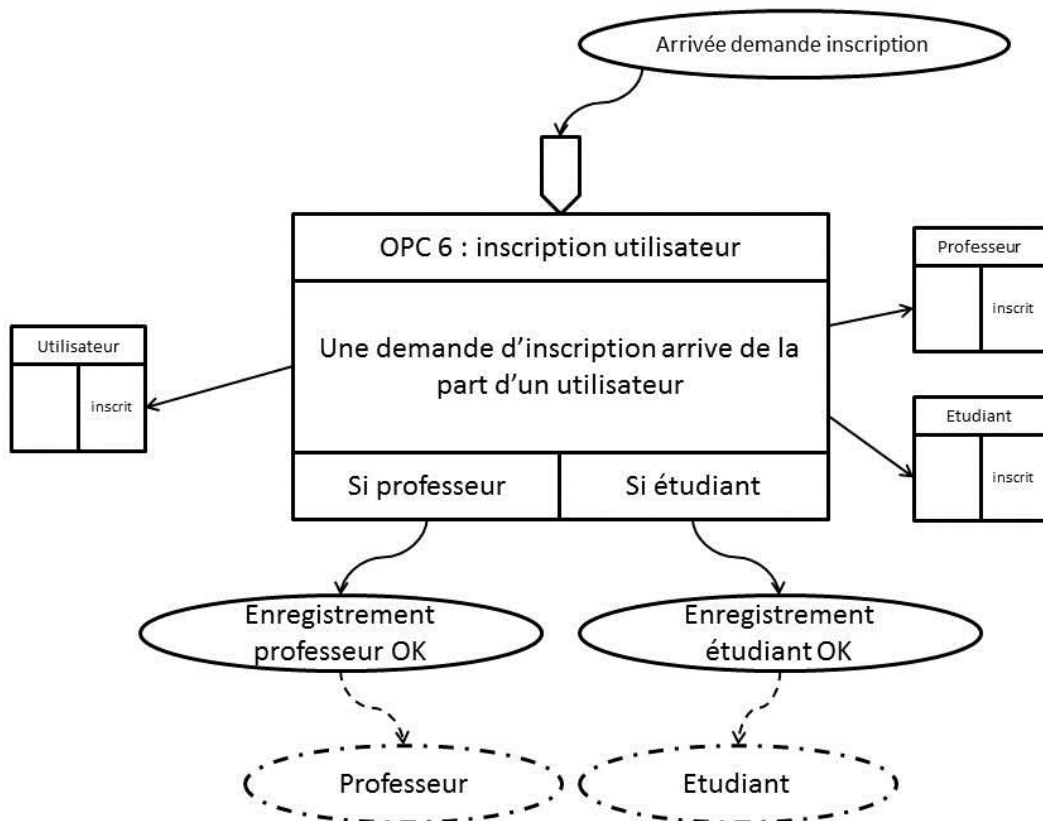
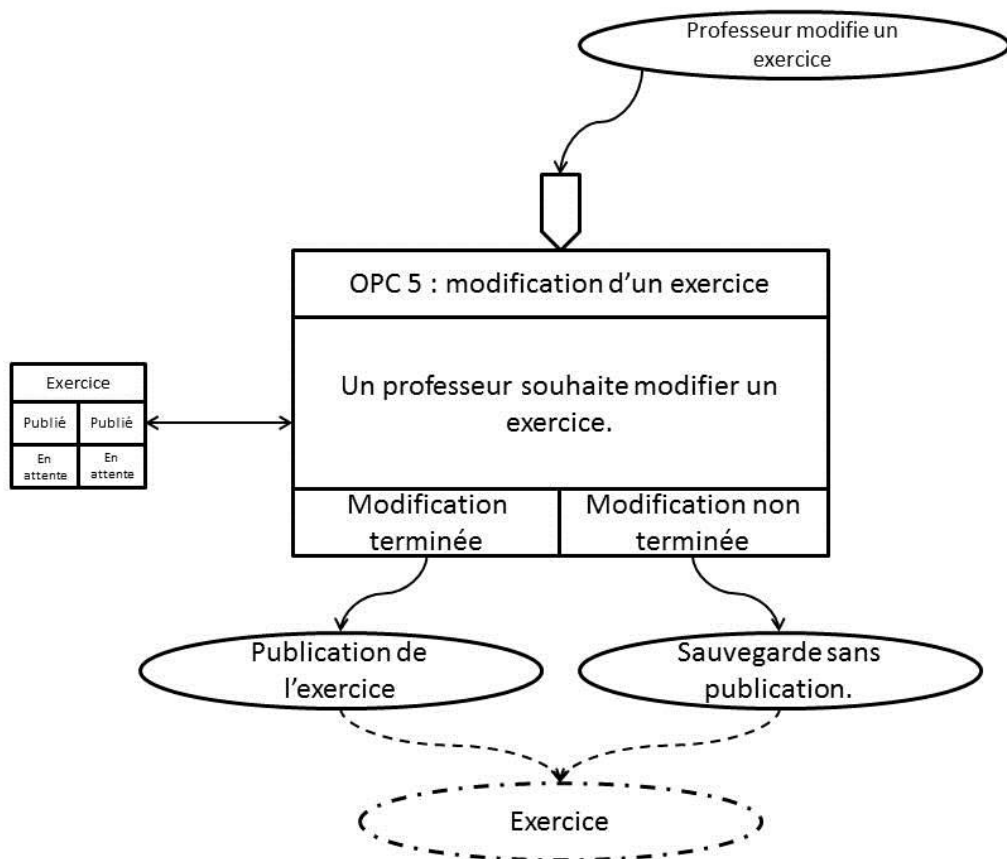
Voir en annexe

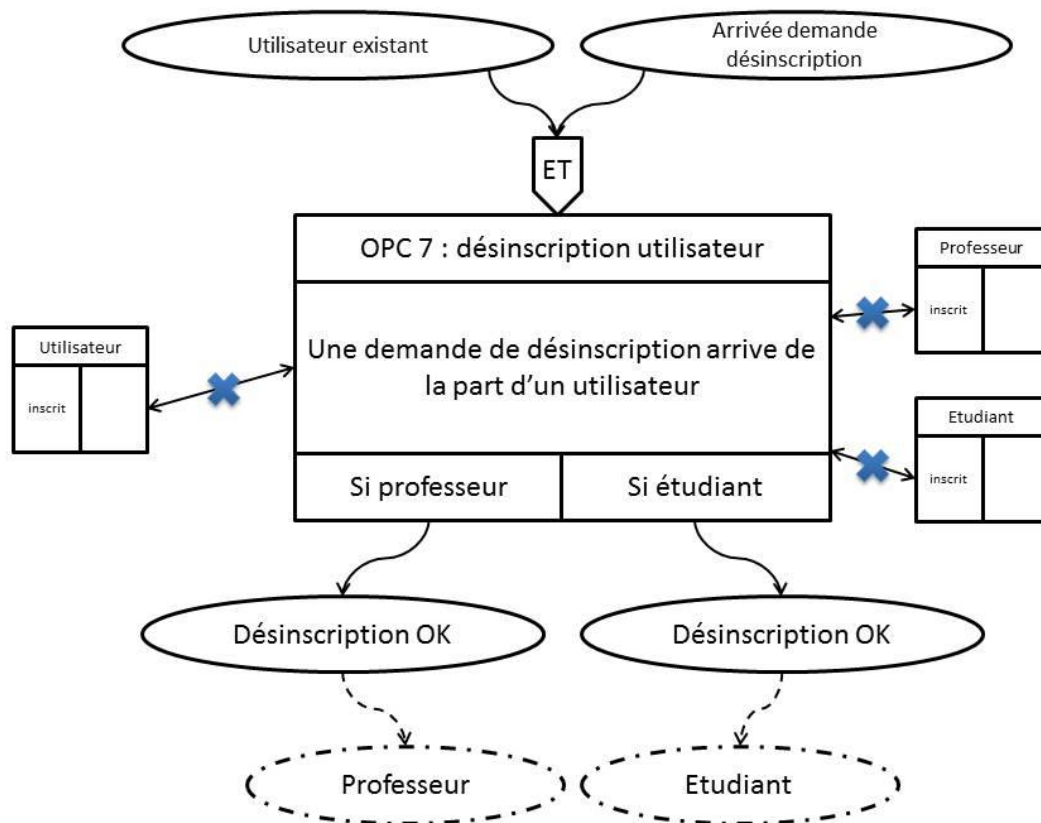
c. Modèle de contexte



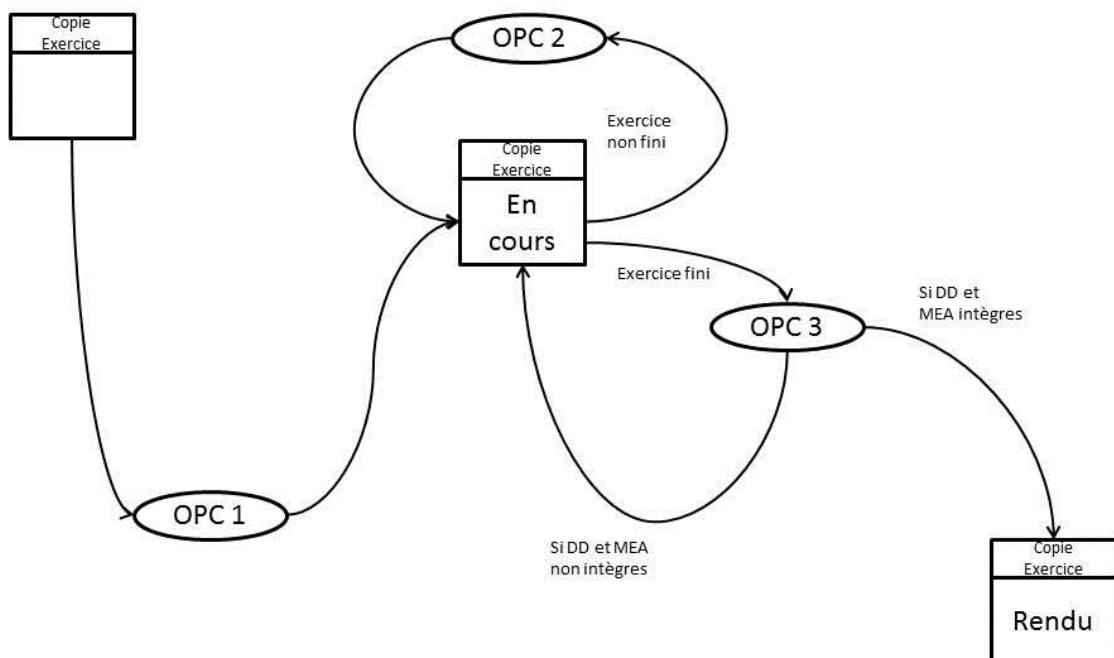
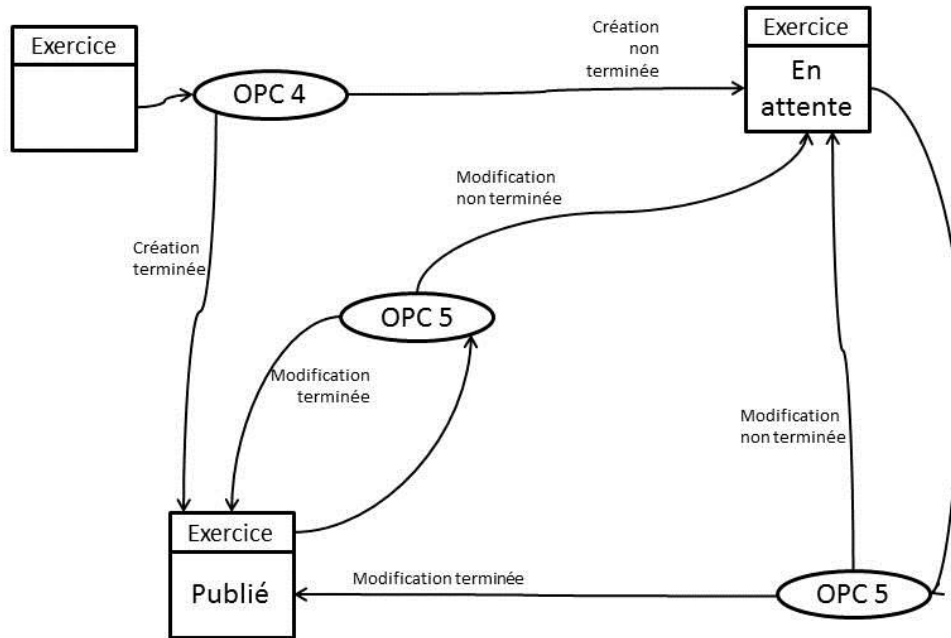
d. Modèle conceptuel de traitement analytique

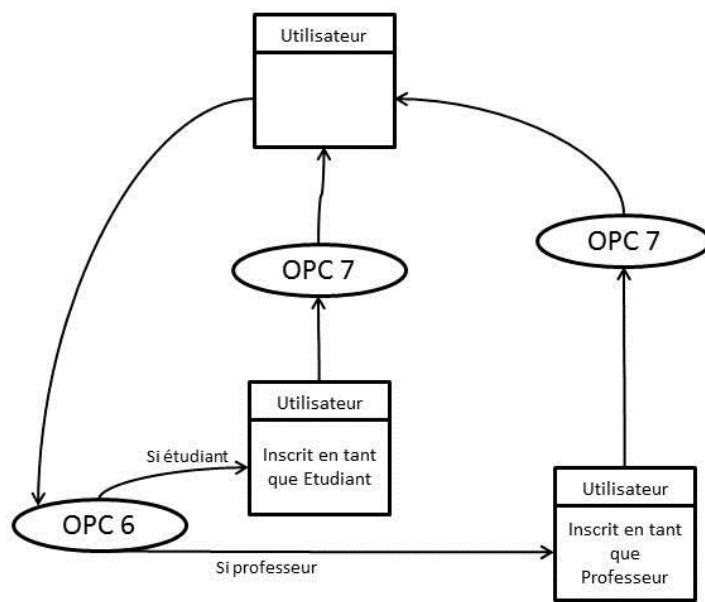






e. Cycle de vie des objets





II. Partie base de donnée

a. Explications liées au passage MEA / schéma relationnel et commentaires concernant le modèle relationnel

En annexe se trouve le schéma relationnel.

Les relations de type "X,1—X,n" ou "X,1—X,1" se transforment en Foreign Key.

Les relations de type "X,n—X,n" se transforment en table associative.

(X signifie que la cardinalité minimale n'a pas d'importance).

Couple de clé primaire dans les tables ENTITE et ASSOCIATION

- Permet d'avoir plusieurs fois une Entité ou une Association du même nom à condition qu'elles n'appartiennent pas au même MEA
- Permet pour un MEA donné d'avoir une unicité des noms d'entité et d'association

Triplet de clé primaire dans la table CALCULEE A PARTIR DE

- Permet d'avoir une Calculée calculée à partir de plusieurs rubriques tout en empêchant l'insertion d'une ligne déjà existante

Implémentation en langage SQL de la généralisation spécialisation des entités RUBRIQUE, ATTRIBUT, CALCULEE et PARAMETRE :

- Nom_Rubrique se transforme en Nom_Attribut, Nom_Parametre et Nom_Calculée suivant la table de manière à bien les différencier
- Rajout de l'attribut ID_DD dans les tables PARAMETRE, CALCULEE et ATTRIBUT
 - Sans cela, il aurait été impossible d'avoir des rubriques ayant le même nom et appartenant à un DD différent : couple de clé primaire
 - Cela permet aussi de simplifier les requêtes de correction (pas de JOIN à faire pour avoir l'ID_DD)
- Rajout de l'attribut ID_MEA dans la table ATTRIBUT
 - Cela permet de savoir à quel MEA on se réfère lors de l'attribution d'un attribut donné à une Entité ou une Association
 - Cela simplifie aussi les requêtes de correction (pas de JOIN à faire pour avoir l'ID_MEA)

Relations entre les tables EXERCICE, DD et MEA

- Choix de mettre la Foreign Key sur l'ID_DD_Correction et ID_DD_Fake de la table EXERCICE qui référencent chacun un ID_DD déjà existants dans la table DD : on est sûr que ID_DD_Correction et ID_DD_Fake référencent chacun un DD qui existe
- Choix de mettre la Foreign Key sur l'ID_MEA_Correction et ID_MEA_Fake de la table EXERCICE qui référencent chacun un ID_MEA déjà existants dans la table MEA : on est sûr que ID_MEA_Correction et ID_MEA_Fake référencent chacun un MEA qui existe

Relations entre les tables COPIE_ELEVE, DD et MEA

- Choix de mettre la Foreign Key sur l'ID_DD de la table COPIE_ELEVE qui référence un ID_DD déjà existant dans la table DD : on est sûr qu'une copie élève donnée référence un MEA qui existe
- Choix de mettre la Foreign Key sur l'ID_MEA de la table COPIE_ELEVE qui référence un ID_MEA déjà existant dans la table MEA : on est sûr qu'une copie élève donnée référence un MEA qui existe

b. Commentaires sur le script de création

Table EXERCICE

- Enoncé de l'exercice lue depuis un fichier dont le chemin d'accès est spécifié par la rubrique Enonce_Exercice.
- Attribut Exercice_Pret de type booléen permettant de gérer les exercices visibles par les élèves

Table COPIE_ELEVE

- Possibilité pour un élève de faire plusieurs copies d'un même exercice.
- Copie considérée comme rendue et non modifiable lorsque le champ Date_Envoi_Copie_Eleve est rempli.
- Champ Note_Copie_Eleve rempli lorsque la copie a été envoyée et corrigée.

Remarque amélioration future

- Créer une table CORRECTION dans laquelle se trouve la correction automatique d'une copie avec la note et un détail des corrections. Le professeur référent de l'élève peut y accéder et vérifier la correction, moduler la note si besoin.
- Mettre en place des requêtes de synthèse qui donnent par exemple la moyenne générale d'un élève, la moyenne d'un élève par exercice, la moyenne générale (et par exercice) d'un groupe ou d'une promotion.

c. Commentaires sur le script de remplissage

Le script de remplissage ne contient pas des données extrêmement exhaustives, seulement les données nécessaires à la vérification du bon fonctionnement des requêtes

- Seulement 3 exercices créés
- Tous les types de données ne sont pas présents
- Seulement un exercice de test et pas extrêmement compliqué
- Les insertions dans la copie élève au niveau du DD et du MEA ne couvrent pas tous les cas
 - Il a été nécessaire pendant les tests des requêtes de modifier des lignes, d'en supprimer certaines puis de les remettre, le tout afin de couvrir un panel maximum de cas possibles
 - Requetes concernées : vérification de l'intégrité du DD et du MEA, correction simples et fines du DD et du MEA

d. Scénarii d'utilisation

Les différents scénarii présentés en annexe permettent de répondre aux besoins du projet, tant au niveau de l'efficacité que de l'ergonomie.

Les premières requêtes, somme toute basiques, permettent un affichage des différentes données nécessaires : liste des professeurs ou des élèves avec conditions de groupes / promotions ou non, affichage de différentes propriétés d'un exercice ou des données d'un élève... Ces requêtes ne se veulent pas exhaustives au possible : il est fort possible que d'autres soient nécessaires au cours de l'élaboration de la partie applicative du projet.

Les scénarii d'utilisation présentés par la suite sont un peu plus complexes et sont décrits en détails en annexe. En voici les principaux aspects.

Création d'un exercice de la part d'un utilisateur autorisé à le faire, typiquement un Professeur

Insertion dans la table EXERCICE

- Deux insertions sont faites dans la table DD qui correspondront à l'ID_DD_Correction et l'ID_DD_Fake de l'exercice
- Deux insertions sont faites dans la table MEA qui correspondront à l'ID_MEA_Correction et l'ID_MEA_Fake de l'exercice
- Finalement une insertion est faite dans la table EXERCICE
 - l'ID_Exercice est en AUTO_INCREMENT : ID unique et pas d'erreur d'insertion
 - L'énoncé de l'exercice correspond au chemin du fichier énoncé (uploadé par exemple)
 - Le nom de l'exercice est demandé à l'utilisateur
 - Le type de l'exercice est proposé à l'utilisateur (liste déroulante)

Mise en place des DD et des MEA de l'exercice

De façon automatique, dès qu'une insertion correspondant à la correction est faite dans la table ENTITE ou RUBRIQUE, la même insertion est faite dans les même table mais pour le DD fake ou le MEA fake. La réciproque n'est pas vraie.

Création d'entités

- Le nom est demandé à l'utilisateur
- L'insertion est ensuite faite dans la table ENTITE avec l'ID_MEA qui convient

Création d'association

- Le nom est demandé à l'utilisateur
- L'insertion est ensuite faite dans la table ASSOCIATION avec l'ID_MEA qui convient

Création de relations entre Associations et Entités

- L'utilisateur relie une Entité à une Association et choisit la cardinalité qui lui convient
- Relier deux entités par une association se fait donc en deux fois
- La base de données créée en fait les pattes de l'association

Spécialisation des Rubriques en Attributs

- Cela passe par l'attribution de rubriques à une entité ou une association : elles deviennent alors des attributs. Lorsque l'utilisateur veut insérer une rubrique dans une entité ou une association, une liste de toutes les rubriques dont il dispose dans son DD lui est proposée et il choisit celle qui lui convient
- L'insertion de fait alors dans la table ATTRIBUT
 - ID_DD et ID_MEA qui correspondent au DD et au MEA en cours
 - Nom_Attribut prend la valeur du nom de la rubrique sélectionnée
 - Nom_Entite ou Nom_Association prend la valeur de l'entité ou de l'association dans laquelle on insère un attribut
 - Bien entendu un attribut ne peut être à la fois dans une entité et une association : un de ces deux champs doit forcément rester à NULL

Spécialisation des Rubriques en PARAMETRES

- Lorsque l'utilisateur veut définir une Rubrique en Paramètre, il lui est alors demandé une valeur (obligatoire)
- L'insertion est ensuite faite dans la table PARAMETRE
 - ID_DD correspond au DD en cours
 - Nom_Parametre correspond au nom de la Rubrique que l'on modifie

Spécialisation des Rubriques en CALCULEES

- Il est alors demandé un type de donnée à choisir parmi toutes les données présentes dans la table TYPE_DONNEE puis une insertion est faite dans la table CALCULEE

- ID_DD correspond au DD en cours
- ID_Type_Donnee correspond l'ID de la donnée choisie
- Nom_Calculée correspond au nom de la Rubrique que l'on modifie
- Il est également demandé de choisir les rubriques qui permettent de calculer la Calculée qui est en train d'être créée
 - Toutes les Rubriques du DD en cours sont proposées dans une liste déroulante
 - Pour chaque Rubrique choisie une insertion est faite dans la table CALCULEE_A_PARTIR_DE
 - ID_DD correspond au DD en cours
 - Nom_Calculée correspond au nom de la Rubrique que l'on modifie
 - Nom_Rubrique correspond à la rubrique choisie parmi la liste

Mise à jour des ATTRIBUTS lorsqu'on veut leur affecter un type de donnée ou un caractère de clé primaire

- Option "définir en tant que clé primaire" disponible pour les attributs
 - le champ Cle_Primaire passe alors à "TRUE"
- Option "définit un type de donnée"
 - La liste des données est tirée de la table TYPE_DONNEE
 - La modification est faite suivant le type de donnée choisi par l'utilisateur

Finalisation de la création des DD et des MEA fake de l'exercice

- L'utilisateur peut rentrer autant de Rubriques ou d'Entités différentes qu'il veut pour le DD_Fake et le MEA_Fake
- Pas besoin de remplir les tables ASSOCIATION, PATTE, ATTRIBUT, CALCULEE, PARAMETRE ou CALCULEE_A_PARTIR_DE pour le DD et le MEA fake.

Finalisation de la création de l'exercice

- Une entrée est faite dans la table EDITION_EXERCICE
 - ID_Utilisateur prend la valeur de l'ID de l'utilisateur courant
 - ID_Exercice prend la valeur de l'exercice courant
 - Date_Edition_Exercice prend la valeur de la date courante
- La date de création d'un exercice est donc la Date_Edition_Exercice la plus ancienne
- Un exercice est en fait sauvegardé automatiquement à chaque changement qu'effectue l'utilisateur, on ne peut pas avoir deux versions d'un même exercice en même temps, à moins de créer deux exercices ayant une ID différente mais étant en fait deux versions d'un même exercice

Rendre l'exercice visible pour les élèves

Un exercice ne peut être visible que si l'intégrité du DD et de MEA sont vérifiées (requêtes de vérification détaillées dans la partie suivante)

- Mise à jour de la table EXERCICE : Exercice_Pret prend la valeur "TRUE"

Vérification de l'intégrité du DD et du MEA (pour un ID_DD et un ID_MEA donné)

Le MEA ne doit pas posséder d'associations ayant moins de deux pattes

- Compter le nombre de couple (Nom_Entite, Nom_Association) avec un NATURAL LEFT JOIN des tables ASSOCIATION et PATTE en groupant par Nom_Association
- Si une association a moins de deux pattes alors le MEA n'est pas intègre

Le MEA ne doit pas posséder d'entités non reliées à au moins une association

- Compter le nombre de couple (Nom_Entite, Nom_Association) avec un NATURAL LEFT JOIN des tables ENTITE et PATTE en groupant par Nom_Entite
- Si une entité n'est reliée à aucune association alors le MEA n'est pas intègre

Toutes les rubriques du DD doivent être spécialisées en attribut, calculée ou paramètre

- Si le nom d'une rubrique (Nom_Rubrique) n'apparaît pas dans l'union des noms des calculées, paramètres et attributs, alors le DD n'est pas intègre

Un attribut ne peut en aucun cas ne pas être dans une association ou une entité

- Chercher dans la table ATTRIBUT les attributs qui n'ont pas de Nom_Entite ou de Nom_Association définis
- Si la recherche est fructueuse alors le DD n'est pas intègre

Un attribut doit obligatoirement posséder un type de donnée

- Si un des attributs n'en possède pas, alors le DD n'est pas intègre
- Le type de donnée est obligatoire pour créer une Calculée, il n'y a donc pas besoin de faire la même vérification dans la table CALCULEE

Une Calculée doit obligatoirement être calculée à partir de quelque chose

- Si une Calculée existant dans la table CALCULEE n'apparaît pas dans la table CALCULEE_A_PARTIR_DE alors le DD n'est pas intègre

Création d'une copie élève

Insertion dans la table COPIE_ELEVE

- Deux insertions sont faites dans les tables DD et MEA qui correspondront à l'ID_DD et l'ID_MEA de la copie élève
- Finalement une insertion est faite dans la table COPIE_ELEVE
 - l'ID_Exercice prend la valeur de l'ID de l'exercice que l'élève effectue
 - Le nom de la copie est rentré par l'élève ainsi que les commentaires éventuels
 - ID_Utilisateur prend la valeur de l'ID de l'élève
 - Date_Derniere_Modif_Copie_Eleve prend la valeur de la date actuelle

Mise en place du DD et du MEA

La mise en place du DD et du MEA de la copie de l'élève se font exactement de la même manière que lorsqu'un professeur crée un exercice à quelque différences près

- Un élève ne crée pas de rubriques, il sélectionne celles qui lui semblent pertinentes depuis l'ensemble des rubriques du DD fake de l'exercice
- Un élève ne crée pas d'entités, il sélectionne celles qui lui semblent pertinentes depuis l'ensemble des entités du MEA fake de l'exercice
- Il crée par contre les associations comme un professeur le ferait
- De même, la spécialisation des attributs ou la création de pattes est identique à ce qui est fait par le professeur

Modification et Sauvegarde des copies d'élèves

Un élève ne peut rendre sa copie si son DD et son MEA ne sont pas intègres (cf. la vérification de l'intégrité du DD et du MEA)

- Un élève peut afficher toutes les copies qu'il n'a pas encore rendues
- Si un élève a terminé et veut rendre une copie
 - Date_Derniere_Modif_Copie_Eleve et Date_Envoi_Copie_Eleve prennent la valeur de la date actuelle
- Si un élève veut simplement sauvegarder sa copie pour la rendre plus tard
 - Date_Derniere_Modif_Copie_Eleve prend la valeur de la date actuelle

Réflexion sur l'enregistrement des copies élèves

On pourrait surement améliorer cette partie de la base de données

- Couple de clé primaire dans la table COPIE_ELEVE sur ID_Copie_Eleve et Date_Sauvegarde_Copie par exemple (Date_Sauvegarde_Copie de type TIMEDATE) et possibilité de changer le nom de la copie élève lorsqu'on sauvegarde

- Lorsqu'un élève sauvegarde sa copie, celle-ci peut être copiée (avec son DD et son MEA) de manière à pouvoir la retrouver plus tard

Requêtes de correction

Correction du Dictionnaire de Données

Il a été spécifié précédemment qu'un élève ne peut rendre sa copie si son DD n'est pas intègre : toutes les rubriques sont donc "spécialisées" en attributs, calculées ou paramètres.

La correction du DD se fait en plusieurs étapes.

Correction simple du dictionnaire de données

- Partie 1 : Trouver les rubriques présentes dans le DD de correction et non présentes dans le DD de la copie de l'élève
- Partie 2 : Trouver les rubriques présentes dans le DD de la copie de l'élève et non présentes dans le DD de correction

En faisant la somme des deux, on a une approximation de correction du dictionnaire de données. On verra par la suite que la correction fine se charge aussi de vérifier la partie 2 de la correction simple. On ne gardera donc in fine que la partie 1.

Correction Fine du dictionnaire de données

Correction des Attributs

Un attribut de la copie de l'élève est correct quand :

- Il apparaît dans le DD de correction
- Il appartient à la même entité
- Il possède la même valeur que la correction pour le booléen "Cle_Primaire"
- Il possède le même type de donnée que dans la correction

Le nombre de points en moins pour une copie élève sur les attributs sera donc égal à l'ensemble de ses attributs moins les attributs corrects. Si tous ses attributs sont dans la correction et que tous sont corrects, il perd zéro point par exemple.

Correction des Paramètres

Un paramètre ne possède pas vraiment de caractéristiques permettant une correction fine. Son attribut "Valeur" est de type VARCHAR et permet à l'élève d'y rentrer des informations diverses de type "20%" ou bien des chiffres, une date, ...

La correction des paramètres est donc simple : étant donné que la "partie 1" de la correction des rubriques est faite, il suffit de rajouter la "partie 2" mais seulement pour les paramètres : le nombre de fautes de l'élève sera le nombre de paramètres présents dans sa copie et non présents dans la correction.

Correction des Calculées

Une calculée de l'élève est correcte si

- Elle est présente dans le DD de correction
- Elle possède le même type de données
- Elle est calculée à partir du même nombre de rubriques que dans la correction
- Elle est calculée à partir des mêmes rubriques que dans la correction

En résumé, pour une calculée donnée, il faut que toutes les lignes soient identiques à celles de la correction et qu'il y ait le même nombre de lignes que dans la correction.

Pour savoir combien l'élève perd de point on calcule donc le nombre total de ses calculées présentes dans la table CALCULEE, auquel on soustrait le nombre de calculées correctes (qui vérifient les conditions ci-dessus). Si l'élève a le bon nombre de calculées et que toutes sont correctes, alors il perd zéro points, par exemple.

Remarque :

Sans PHP, la correction des Calculées ne marche qu'à condition qu'il y ait une seule calculée. S'il y a plus d'une calculée alors les sous-requêtes retournent plusieurs lignes et la requête principale ne marche plus. Il faut donc faire une correction "ciblée" : lancer une même requête pour chacune des calculées, et ce de façon individuelle.

Voici une idée préalable de comment cela marcherait en langage algorithmique.

```

si (présentDansCorrection(uneCalculée))
    alors nbCalculéesJustes <— nbCalculéesJustes + verifCalculée(uneCalculée)
  
```

fin si

- `présentDansCorrection(uneCalculée)` vérifie que "uneCalculée" (calculée du DD de la copie de l'élève) soit présente dans la correction
- `verifCalculée(uneCalculée)` vérifie que "uneCalculée" (calculée du DD de la copie de l'élève) réponde à toutes les conditions de correction d'une calculée. Retourne 1 si la calculée est juste, 0 sinon

Cette condition "si...fin si" est à répéter tant qu'on n'a pas vérifié toutes les calculées de l'élève.

Le total de points en moins sera le nombre total de ses calculées moins le nombre de calculées justes. Si toutes sont bonnes, il ne perd pas de points.

Correction du MEA

Correction simple du MEA

- Partie 1 : Trouver les entités présentes dans le MEA de correction et non présentes dans le MEA de la copie de l'élève
- Partie 2 : Trouver les entités présentes dans le MEA de la copie de l'élève et non présentes dans le MEA de correction

En faisant la somme des deux, on a une approximation de correction du MEA. On verra par la suite que la correction fine se charge aussi de vérifier la partie 2 de la correction simple. On ne gardera donc in fine que la partie 1.

Correction fine du MEA

Nous n'avons pas réussi à mettre en place une correction fine du MEA avec des noms d'associations inconnus choisis par l'utilisateur. La correction qui suit est donc basée sur l'hypothèse que les noms d'associations sont définis automatiquement lorsque que deux entités ou plus sont reliées ensemble.

Le nom de l'association est déterminé de la façon suivante :

- L'utilisateur indique quelles entités il veut lier par une association
- Le nom de l'association est alors composé des deux noms des entités triés par ordre alphabétique et séparés par un underscore
- Il est alors demandé à l'utilisateur les cardinalités pour chaque entité
- S'il veut lier 3 entités, le nom de l'association est alors composé des trois noms des entités triés par ordre alphabétique et séparés par un underscore
- Possibilité de lier une entité à une association déjà existante pour transformer une binaire en ternaire, voire une ternaire en quaternaire. Le nom de l'association est recalculé automatiquement

Une entité est correcte si

- L'entité est présente dans le MEA de correction de l'exercice
- Elle est liée à la même association que dans le MEA de correction avec la même cardinalité
- Chaque association est liée au même nombre d'entités et aux mêmes entités que dans le MEA de correction

Cela permet de vérifier à la fois les Entités, les Pattes et les Associations

Le total des points en moins sera égal au nombre total d'entités du MEA de l'élève moins le nombre d'entités correctes. Si tout est bon, il a donc zéro point en moins.

Remarque :

Cette correction "fine" du MEA marche en théorie mais n'a pu être testé de façon satisfaisante. Il est donc fort possible que des ajustements soient nécessaires.

Sans PHP, la correction ne marche qu'à condition qu'il y ait une seule association. S'il y a plus d'une association alors les sous-requêtes retournent plusieurs lignes et la requête principale ne marche plus. Il faut donc faire une correction "ciblée" : lancer une même requête pour chacune des associations, et ce de façon individuelle.

Voici une idée préalable de comment cela marcherait en langage algorithmique.

si (`présentDansCorrection(uneEntité)`)

alors `nbEntitésJustes <— nbEntitésJustes + verifEntitéAssociation(uneAssociation)`

fin si

- `présentDansCorrection(uneEntité)` vérifie que " uneEntité " (du MEA de l'élève) soit présente dans la correction
- `verifEntitéAssociation(uneAssociation)` vérifie que "uneAssociation" (du MEA de l'élève) réponde à toutes les conditions de correction du MEA. Retourne 1 si le ensemble entités et association est juste, 0 sinon

Cette condition "si...fin si" est à répéter tant qu'on n'a pas vérifié toutes les entités de l'élève.

Le total de points en moins sera le nombre total de ses entités moins le nombre d'entités justes. Si toutes sont bonnes, il ne perd pas de points.

Remarque : Dans le cas où les noms d'associations seraient générés automatiquement, il faudrait alors penser à adapter la requête de correction des attributs afin de vérifier le nom d'association (une association peut être porteuse d'attributs).

Conclusion

Contraintes non modélisables

Contraintes de OU exclusif entre deux entités

Prenons deux tables table1 et table2 :

- Un même "objet" existe soit dans l'une, soit dans l'autre, mais en aucun cas dans les deux
- Trigger qui vérifie, lors d'une insertion, que ce que l'on veut insérer dans la table1 n'existe pas déjà dans la table2 et inversement

Ces contraintes de vérification doivent être implémentées :

- Entre les tables PROFESSEUR, ELEVE et UTILISATEUR lors de l'insertion d'un nouvel utilisateur
 - Un utilisateur est soit Professeur, soit Elève, soit autre (possibilité d'avoir un utilisateur Administrateur) mais en aucun cas il ne peut avoir plusieurs statuts à la fois
- Entre les tables ENTITE, ASSOCIATION et ATTRIBUT lors de l'insertion d'attributs
 - Un Attribut peut être associé soit à un Entité, soit à une Association et en aucun cas aux deux à la fois
 - Il peut aussi n'être associé à aucun des deux, le MEA n'est alors pas intègre, mais cela est vérifié par la suite
- Entre les tables EXERCICE, MEA et COPIE_ELEVE
 - un MEA donné (ID_MEA) appartient soit à la correction d'un exercice, soit au "fake" d'un exercice, soit à une copie élève
 - Il ne peut "exister" (il peut être créé dans la table MEA mais ne sera pas utilisé ou référencé) s'il n'appartient pas à l'une de ces trois catégories et ne peut absolument pas appartenir à plus d'une catégorie à la fois
- Entre les tables EXERCICE, DD et COPIE_ELEVE
 - Un DD donné (ID_DD) appartient soit à la correction d'un exercice, soit au "fake" d'un exercice, soit à une copie élève
 - Il ne peut "exister" (il peut être créé dans la table DD mais ne sera pas utilisé ou référencé) s'il n'appartient pas à l'une de ces trois catégories et ne peut absolument pas appartenir à plus d'une catégorie à la fois
- Entre les tables ATTRIBUT, CALCULEE ET PARAMETRE
 - Une rubrique donnée ne peut être à la fois de type ATTRIBUT et de type CALCULEE ou PARAMETRE
 - Une rubrique donnée appartient à l'un des trois types de façon exclusive

A approfondir lors du développement de l'application

Se pencher plus avant sur les requêtes de modification des exercices de la part d'un utilisateur PROFESSEUR

- Possibilité de supprimer un exercice
 - Auquel cas qu'advient-il des copies élèves qui référencent cet exercice
- Possibilité de modifier un DD et/ou un MEA de correction
 - Auquel cas qu'advient-il des copies élèves déjà rendues et corrigées
 - Elève notifié qu'une nouvelle correction est disponible
 - Nouvelle correction faite à la demande de l'élève (si automatique, risque d'être lourd si beaucoup de copie d'élèves relatives à l'exercice modifié)

Réfléchir à la possibilité pour un utilisateur PROFESSEUR de vérifier une correction faite automatiquement par l'ordinateur

- Création d'une table correction avec génération d'un fichier de correction ?

- Possibilité pour le professeur de moduler la note de l'élève
- La correction peut générer une note globale, une note par élément (DD et MEA)

Approfondir l'automatisation de la correction

- Ajouter à la création de l'exercice un système de spécification du barème de notation. Le professeur spécifie :
 - Pour le DD, enlever tant de points par rubrique fausse,
 - Pour le MEA, enlever tant de points par faute.
- Ramener la note de l'élève sur 20 en tenant compte du nombre maximal de point de l'exercice
 - Si l'exercice est sur 267 et que l'élève a eu la note de 136, sa note finale sera :
$$136 * 20 / 267 = 10,19$$

ANNEXES

