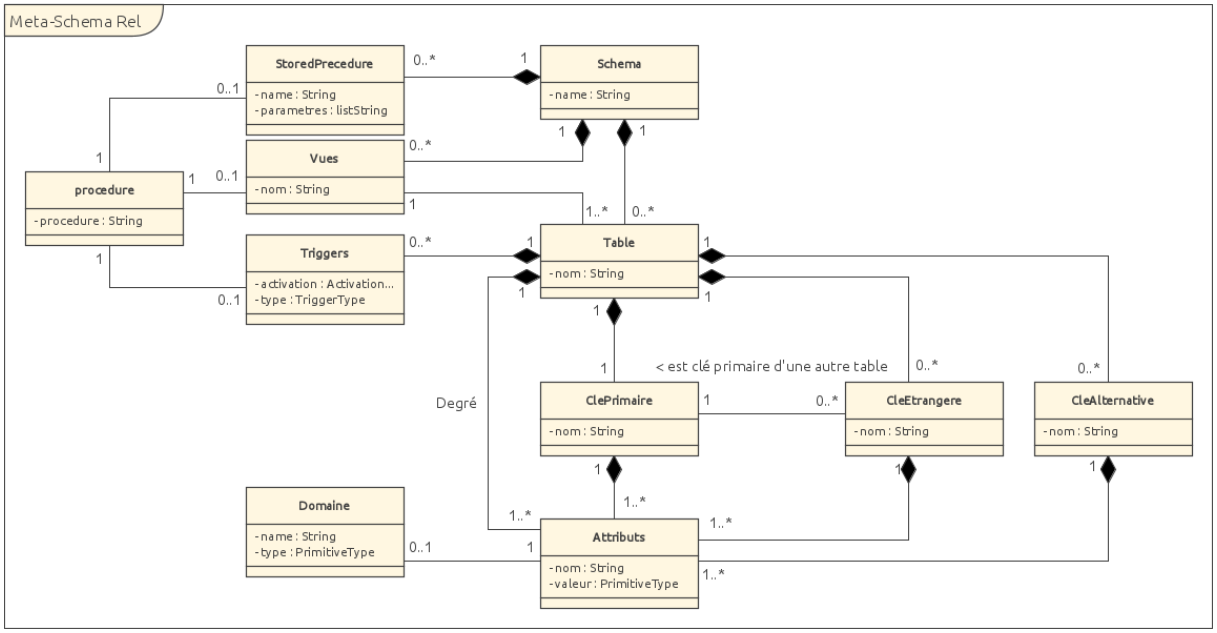
Modélisation Slyum Relationnel

Meta-Schéma Relationnel

Le meta-schéma sert à représenter les éléments du schéma relationnel qui seront présent dans Slyum Relationnel.



Schéma

Schéma est une classe englobante qui sert à différentier les schémas relationnels au sein du logiciel

Table

Elément basique d'une base de données relationnel

Clés Primaires/Etrangères/Alternatives

Les différentes clés des tables doivent être uniques et sont composées d'un ou plusieurs attributs

Attributs

Les différentes colonnes de la table

Procédures Stockées/Vues/Triggers

Eléments du script qui ne font pas partie du modèle relationnel strict mais sont présent dans beaucoup de SGBD et sont intéressant à inclure pour la génération du script SQL

Procédures

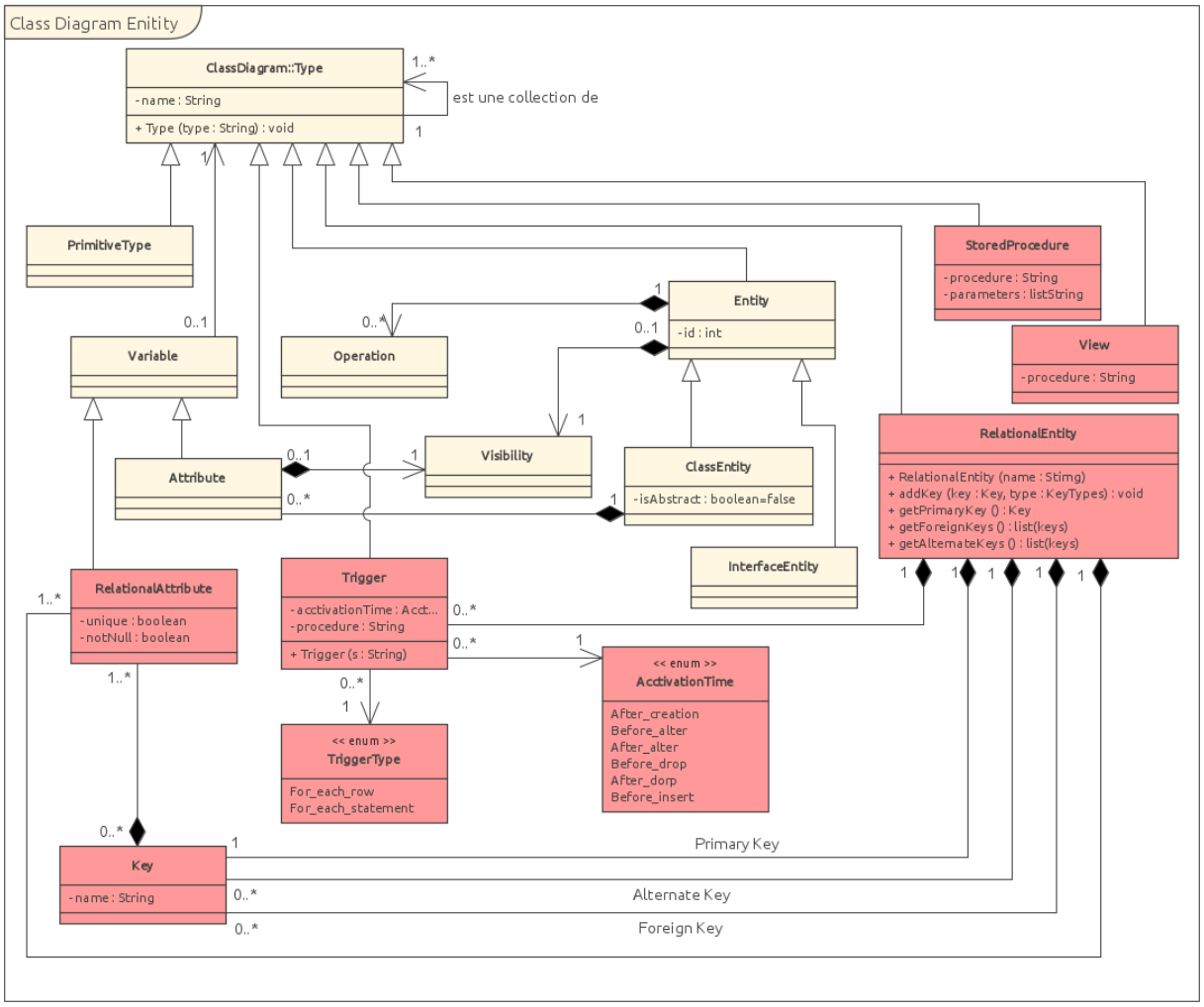
Les procédures génériques sont, pour le moment, des strings qui correspondent aux procédures/requêtes associées aux triggers, vues et procédures stockées. Pour le moment ce sont de simples strings mais elles pourraient, si le temps le permet, être améliorés en un(des) objet(s) plus complexe(s) qui pourrai(ent) stocker la procédure indépendamment de la SGBD visé.

Insertion dans Slyum

Les sections ci-dessous servent à illustrer comment les éléments de mon met-schéma s'insèrent dans le schéma existant de Slyum.

Diagramme de classe – entité

Ce diagramme contient tous les éléments du meta-schéma sauf la relation entre clé primaire et clé étrangère



RelationalEntity

Correspond à la table du meta-schéma

RelationalAttribute

Correspond à l'attribut du meta-schéma

Key

Correspond au 3 types de clés du meta-schéma qui seront stockées dans 3 attributs de RelationalEntity

Trigger, TriggerTypes et ActivationTime

Correspond au Trigger du meta-schéma

StoredProcedure et View

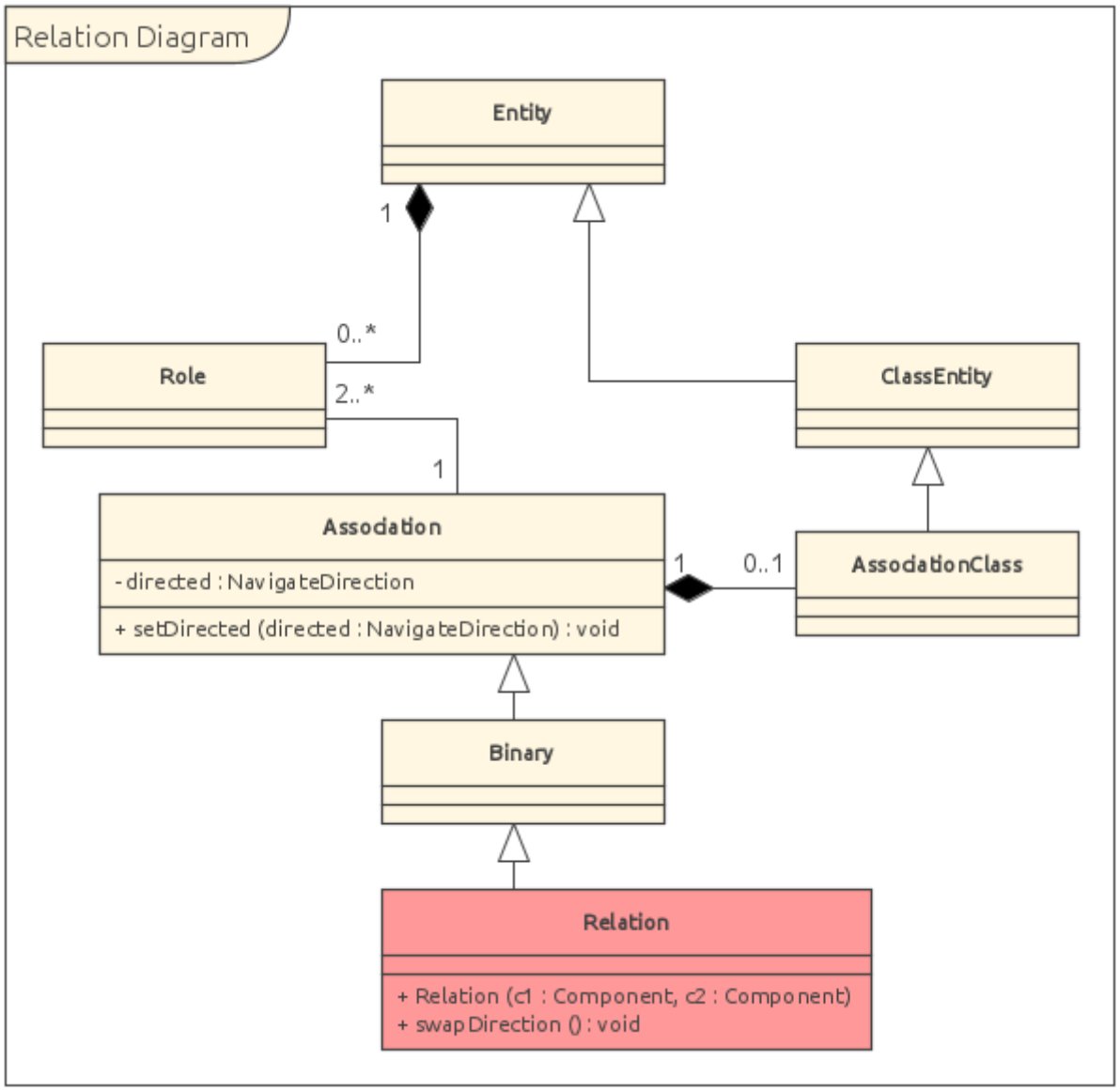
Correspondent respectivement à procédure stockée et vue du meta-schéma

Procédure

L'élément procédure du meta-schéma est représenté par l'attribut "procédure" dans les classes Trigger, StoredProcedure et View.

Diagramme de classe - relation

Ce diagramme montre comment la relation entre les tables est défini



La classe relation est le seul moyen de créer des clés étrangères dans les tables. Quand une relation est créée, la table d'origine du lien est définie comme clé primaire et la table de destination comme clé étrangère. La clé primaire de la table d'origine est ajoutée à la table de destination comme clé étrangère. La méthode swapDirrection permet de changer la "direction" du lien et ainsi changer quelle table recevra la clé étrangère.

Règles de conversion

Classes, interfaces et classes d'association deviennent des **Tables**

Attributs deviennent des **Attributs Relationnels**

Les opérations sont négligées à la conversion car elles n'ont pas d'équivalence directe en relationnel

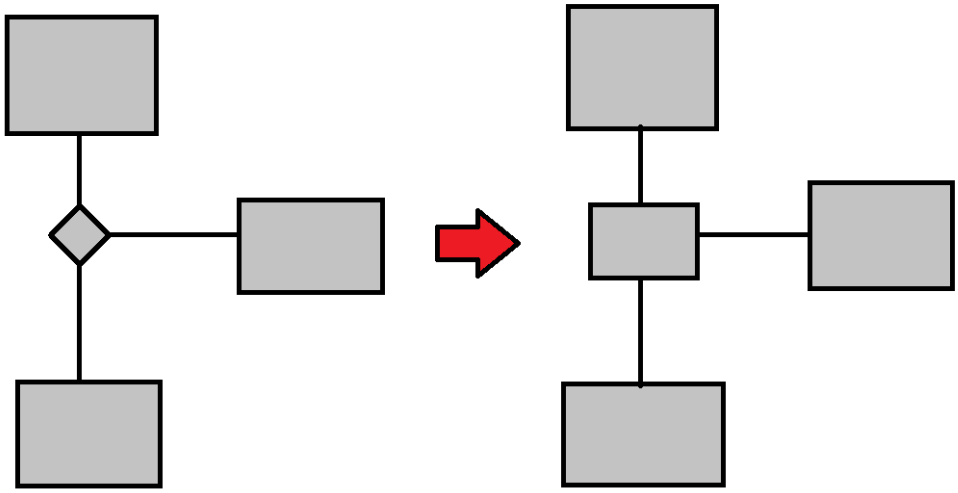
Associations simples, agrégations et compositions deviennent des **Relations.**

La conversion des **cardinalités** des relations se fait selon les règles standard (celles du cours de BDR)

Si une association a une **classe d'association,** les attributs de cette classe sont ajoutés à une des tables de la relation selon les règles standard (BDR)

Une **association multiple** est convertie en une table liée aux autres tables de la relation via des relations 1-N

Un avertissement est donné à l'utilisateur pour lui permettre de le faire manuellement



Les liens d'**héritage** sont convertis en **association simple** avec la sous-classe ayant la même clé que la classe parent.

Les classes internes deviennent des classes normales et suivent les mêmes règles que celles-ci.

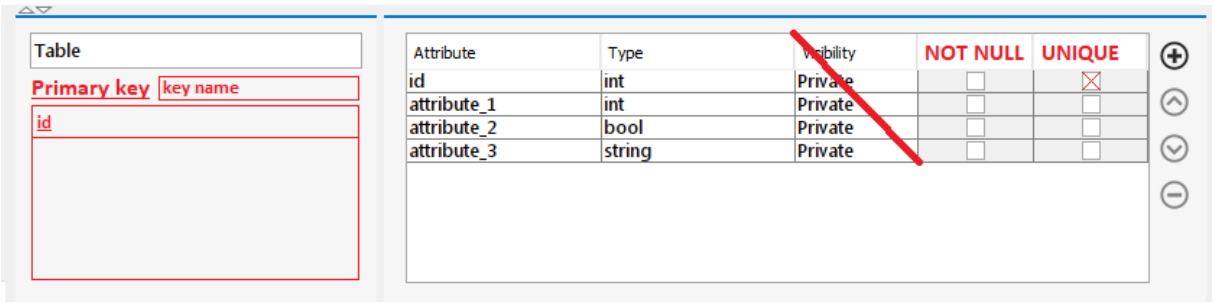
La conversion a 2 modes, rajouter un attribut id a toutes les classes qui sera la clé primaire ou, pour chaque classe un popup laissera à l'utilisateur le choix de définir sa propre clé (attribut simple, composition d'attributs ou id par défaut).

Validation Schéma

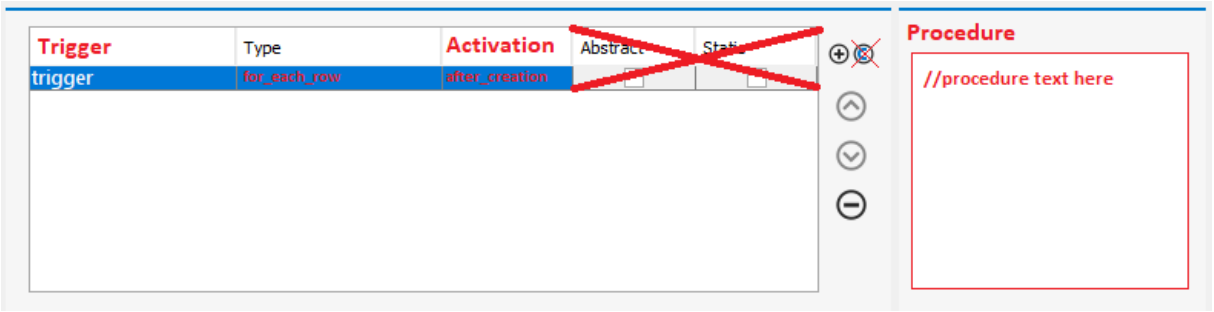
Un validateur du schéma relationnel sera lancé automatiquement à la conversion en relationnel et avant la conversion en SQL. La validation peut aussi être lancée manuellement via l'interface. Les résultats de cette analyse peuvent se trouver sur l'interface de projet (voir plus bas).

Interface

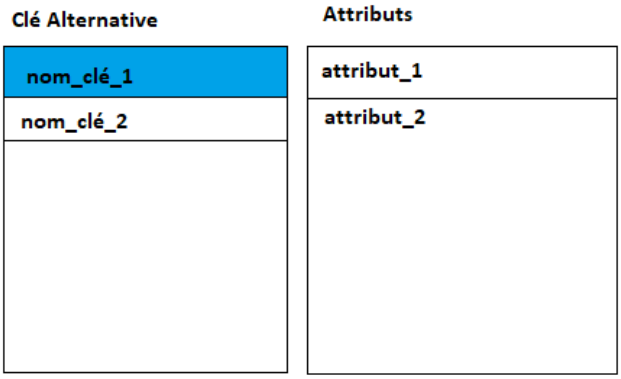
Mockup interface tables, attributs et clés primaire



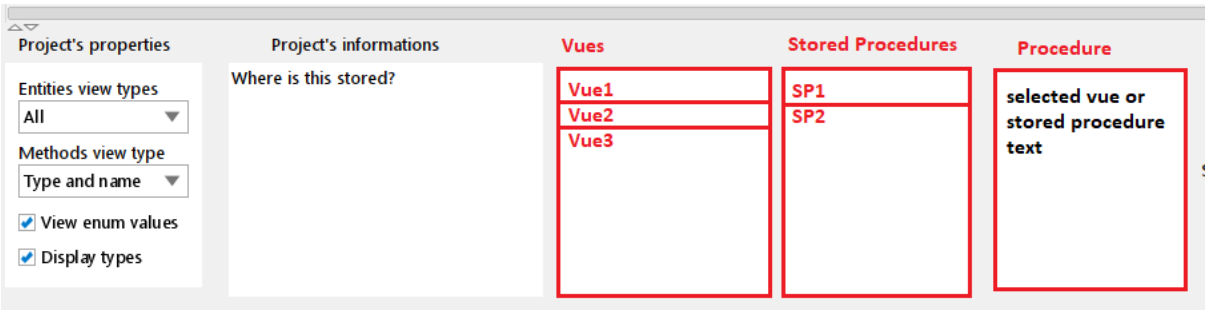
Mockup interface triggers



Mockup interface clés alternatives



Mockup interface de projet (vue et procédures stockées).



La partie de l'interface dédiée aux éléments de la classe est adapté pour fonctionner avec les tables relationnelles.

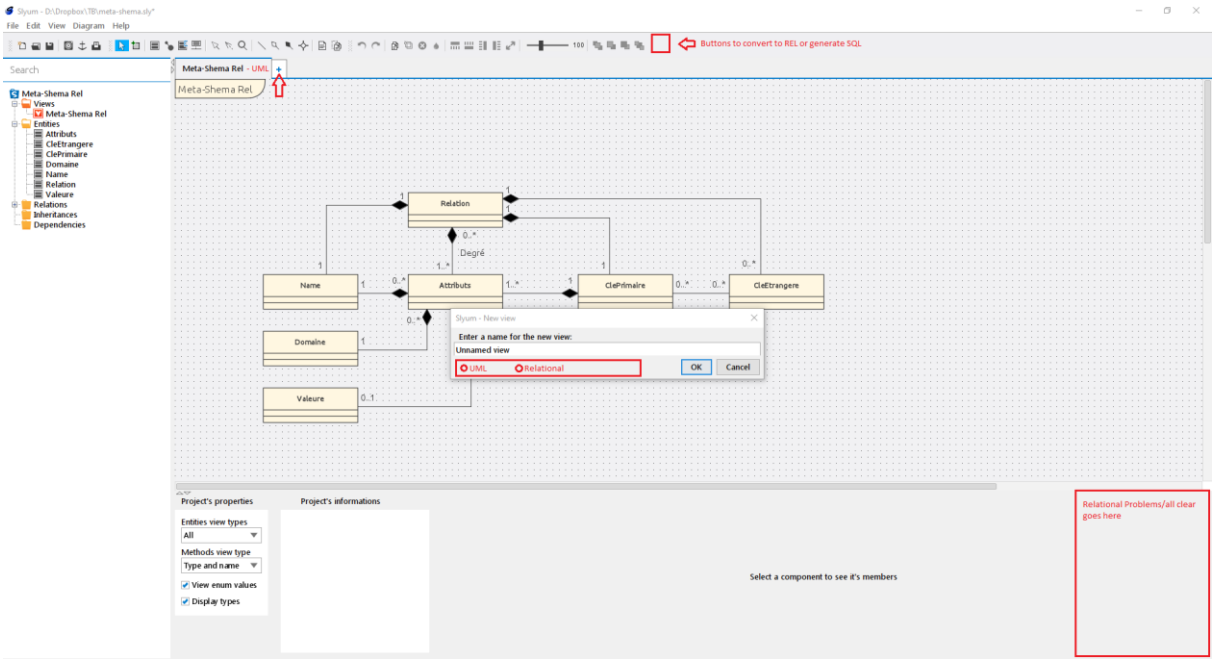
Une table pour définir une clé primaire est rajoutée. Cette table peut contenir un ou plusieurs attributs. A la création d'une classe, un attribut "id" est créé pour créer une clé par défaut.

La partie contenant les méthodes de classes est changée pour pouvoir stocker des triggers. La partie montrant les paramètres de la méthode montre maintenant une zone de texte pour la procédure.

Une troisième partie est rajoutée pour définir les clés alternatives et ses attributs.

Si aucune table n'est sélectionnée, on voit l'interface du projet avec les vues et les procédures stockées à côté.

Mockup interface création de vue "uml" ou "relationnel"

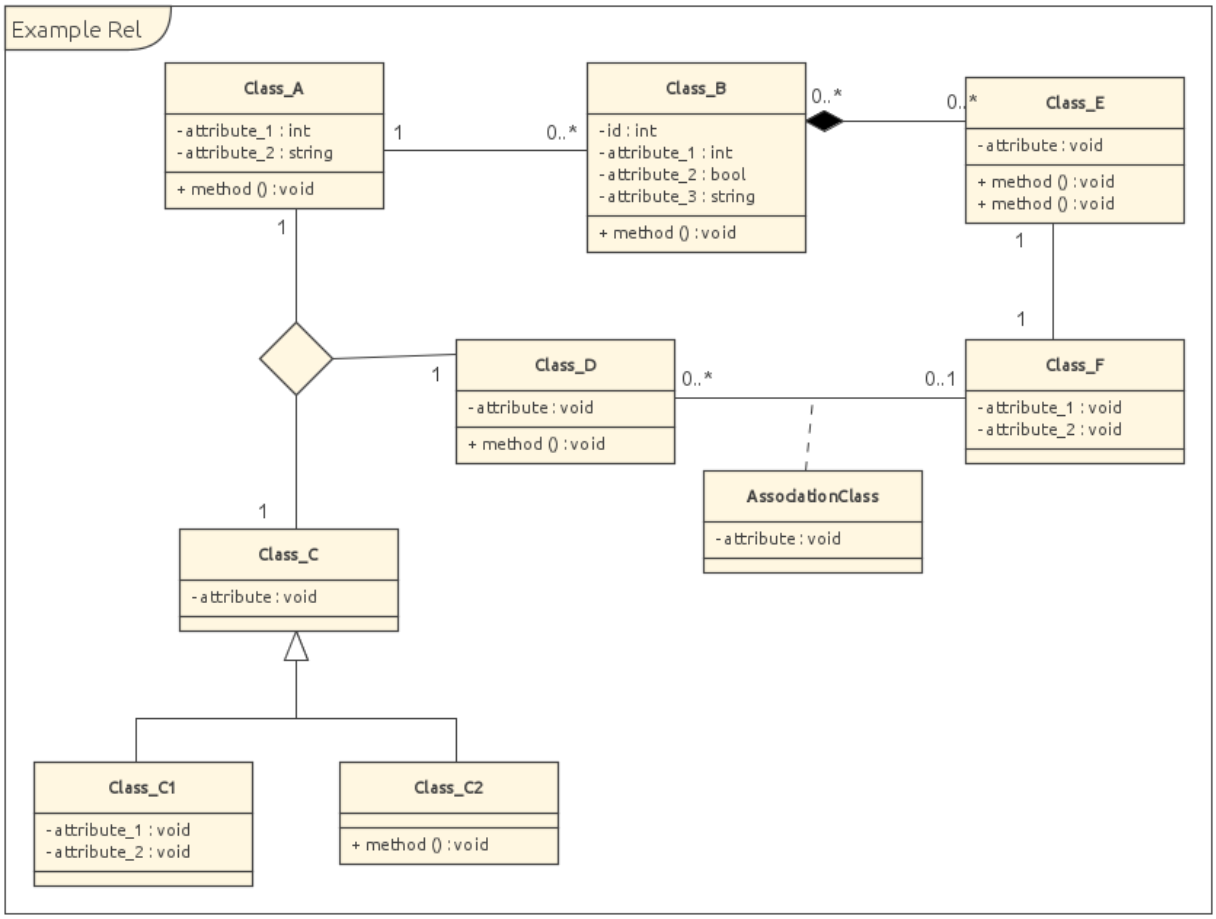


Le choix de uml ou relationnel quand on crée une nouvelle vue ne change pas les classes utilisées mais détermines les fonctions utilisables sur la GUI (ex : on peut créer des liens d'héritage en uml mais pas en relationnel)

Je profite aussi de cette vue globale pour montrer où se situent les résultats de la validation relationnel.

Use Case

Supposez un utilisateur avec ce schéma UML qu'il veut convertir en schéma relationnel puis en scripte MYSQL.



- L'utilisateur choisi de convertir son schéma avec l'option définir les clés soi-même
- L'utilisateur est averti que son association multiple sera changée en classe lui donne le choix d'arrêter la conversion pour la changer manuellement.
- Il décide de continuer et donc l'association est convertie en une classe avec un nom par défaut et des associations 1-N vers les 3 classe qu'elle liait.
- La classe A est convertie et l'utilisateur choisi les attributs 1 et 2 comme clé primaire composite.
- La classe B est convertie et id est choisi comme clé primaire
- La classe C est convertie et attribute est choisi comme clé primaire
- Les classes C1 et C2 sont converties et vu qu'elles sont sous classes de C on y rajoute les attributs de la clé de C et la même clé primaire est choisie pour les 2
- Les classes D et E sont converties avec leur attribute respectif comme clé primaire
- La classe F est convertie et l'utilisateur décide de rajouter une clé par défaut nommée id
- Les associations sont converties
    - L'association 1-N entre A et B est transformé en relation avec la clé étrangère dans B
    - La composition N-M entre B et E est transformé en une table qui a comme clé primaire les clés de B et E
    - L'association 1-1 entre E et F est transformé en relation avec la clé étrangère dans E

    - L’association 1-N entre D et F est transformé en relation avec la clé étrangère et les attributs de la classe d’association dans D
    - Les associations 1-N résultant de la transformation de l’association multiple sont transformés de la même façon qu’entre A et B
    - Les liens d’héritage entre C, C1 et C2 sont transformé en relation avec une clé primaire et clé étrangère identique
- A l’issue de la transformation le validateur relationnel analyse le schéma, une boucle est détectée [A, B, E, F, D, assocMulti, A] et une erreur est rajoutée dans la liste des erreurs
- L’utilisateur rajoute une clé alternative à B avec comme attributs 2 et 3.
- L’utilisateur rajoute une vue
- L’utilisateur rajoute une procédure stockée
- L’utilisateur rajoute un trigger sur les classes A et D
- L’utilisateur essaye de convertir son schéma relationnel en SQL. Une erreur est levée et l’utilisateur est averti qu’il ne peut créer de script avant la résolution de toutes les erreurs
- L’utilisateur supprime le lien entre E et F pour “réparer” l’erreur
- L’utilisateur ressaye de générer le script SQL. Le validateur ne trouve pas d’erreur.
- L’utilisateur choisi Postgress comme SGBD, un script est généré et est sauvegardé ou l’utilisateur choisi.

Annexes

Schéma d'entité Slyum original

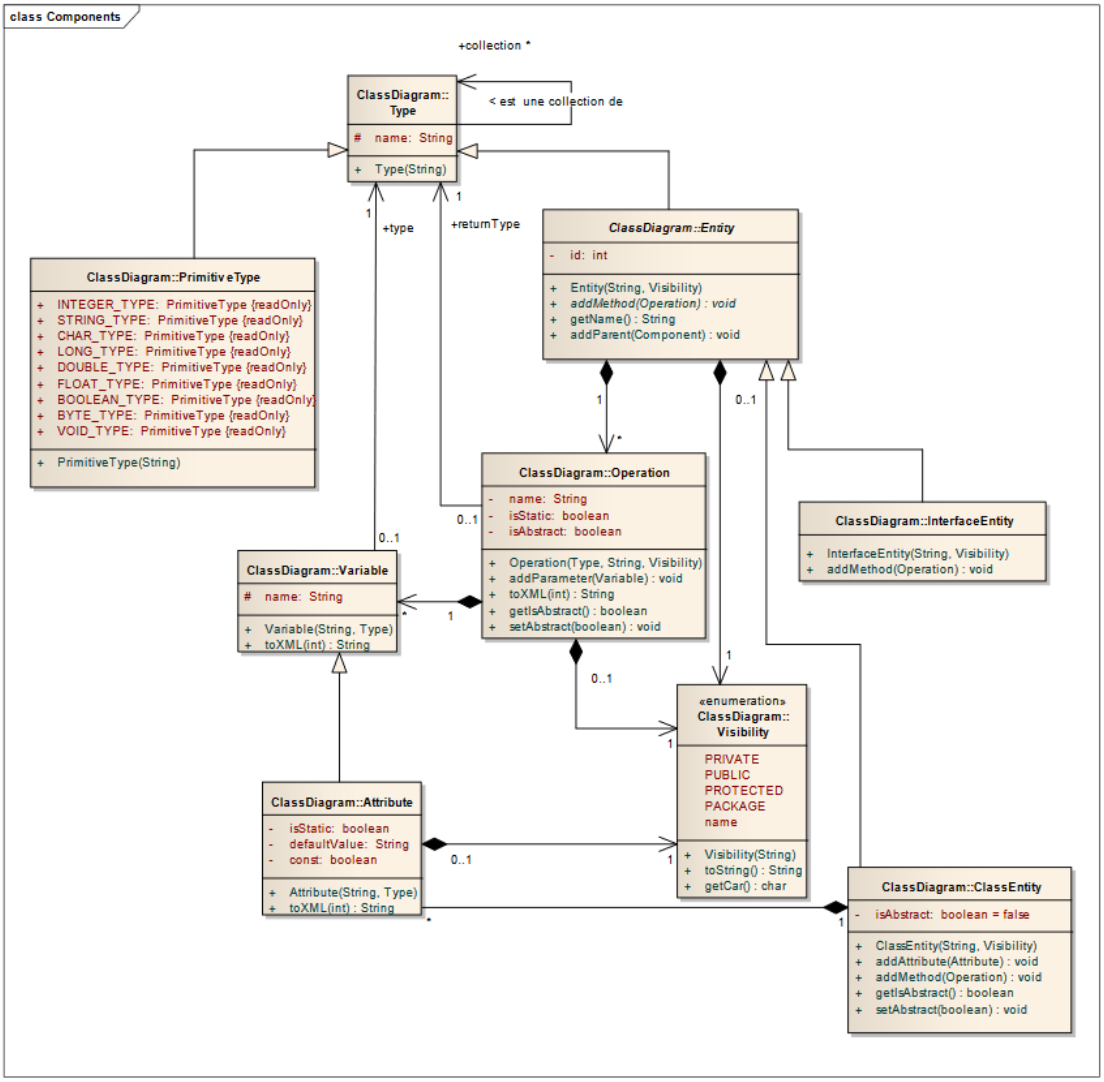


Schéma de relation Slyum original