

TD1 : Modèle entités associations avec looping

Le modèle Entité-Association a été conçu aux Etats-Unis en 1976 par le Taiwanais Peter Chen. Il est aujourd'hui à la base de la plupart des méthodes de modélisation des données.

Le modèle Entité-Association est utile pour la modélisation, et consiste à décrire les informations représentant les données de manière simple. Il permet ensuite un passage facile au modèle relationnel qui permet alors de programmer la base de données (hors programme) et d'effectuer des requêtes dessus (au programme).

1 Entités

- Une **entité** est un élément physique, concret, ou abstrait.
Une entité est donc un objet possédant des propriétés, appelées **attributs** servant à le décrire. Chaque attribut possède une valeur simple.
Et l'ensemble des valeurs des attributs constituent un n-uplet.
- L'ensemble des objets qui ont des caractéristiques communes, les attributs, est un **type d'entité** ou une **classe d'entité**.
- Un attribut particulier possède une valeur différente pour chaque entité et permet de distinguer deux entités. On l'appelle **identifiant**. Chaque classe d'entité possède au moins un attribut identifiant.
On représente un type d'entité avec son nom et les noms des différents attributs dans un tableau.

Exercice :

- Lancer le logiciel Looping
- Créer un nouveau modèle que vous pouvez nommer bibliothèque
- Créer l'entité Auteur avec les attributs idAuteur, nom, prénom, naissance, nationalité.
L'attribut idAuteur est l'identifiant que l'on peut souligner.

Auteur
<u>idAuteur</u>
Nom
Prénom
naissance
décès
nationalité

2 Associations

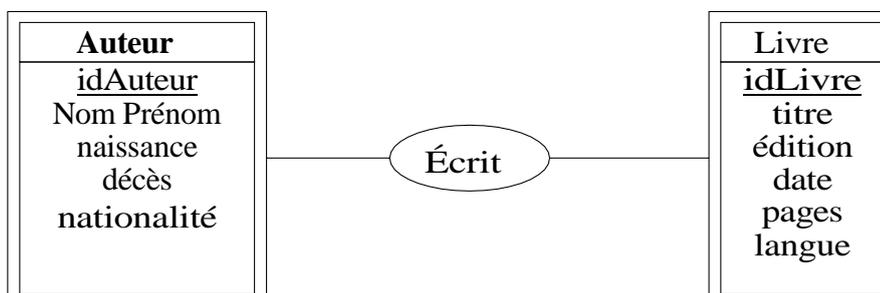
- Des liens ou relations existent entre les entités. Une **classe de relation** entre deux types d'entités contient toutes les relations du même type entre les entités appartenant à ces deux types.
- La relation ou le lien entre entités s'appelle une **association**.
- On parle donc de modèle **entité-association**.

Exercice :

- Créer une deuxième entité Livre. L'attribut idLivre est l'identifiant que l'on peut souligner.
- Créer une association entre Auteur et Livre

Le type d'association est décrit par un titre comme **Écrit** puisqu'un auteur a écrit un livre ou **A été écrit** puisqu'un livre a été écrit par un auteur.

Livre
<u>idLivre</u>
titre
édition
date
pages
langue



3 Cardinalité

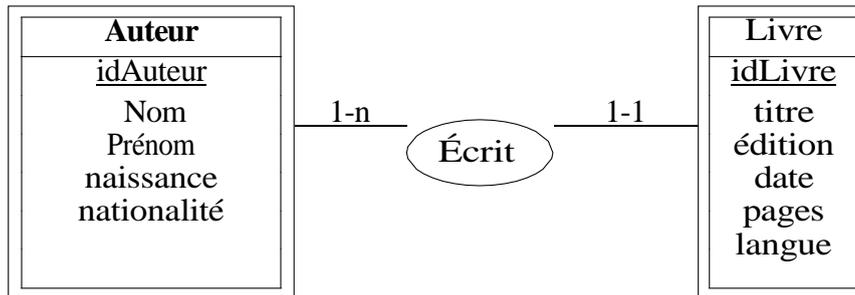
- Le lien entre une entité et une association est caractérisé par la **cardinalité**. Celle-ci est un couple minimum-maximum qui indique le nombre minimum de fois et le nombre maximum de fois qu'une entité peut participer à l'association.

On distingue 3 cardinalités :

- Le couple est 1-1 si chaque entité participe exactement une fois à la relation.
- Le couple est 1-n si chaque entité participe au moins une fois à la relation.
- Le couple est 0-n si certaines entités ne participent pas à la relation.

Exemple : Chaque auteur peut avoir écrit plusieurs livres, donc la cardinalité est 1-n entre Auteur et Écrit.

Prenons, dans un premier temps le cas où chaque livre n'a été écrit que par un unique auteur, la cardinalité est donc 1-1 entre Livre et Écrit.



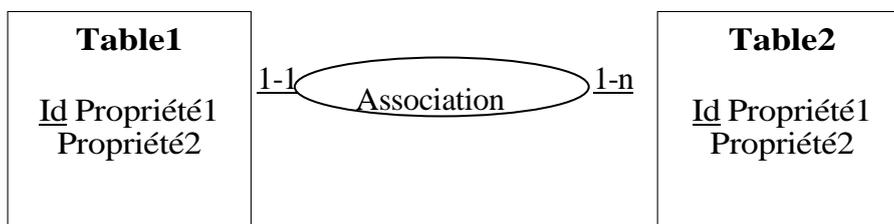
L'association est 1-* entre Livre et Auteur : un auteur peut avoir écrit plusieurs livres mais un livre n'a été écrit que par un seul auteur.

- On peut présenter le modèle entité-association en parlant d'association 1-*, 1-1 ou *-*

4 Lien avec le modèle relationnel

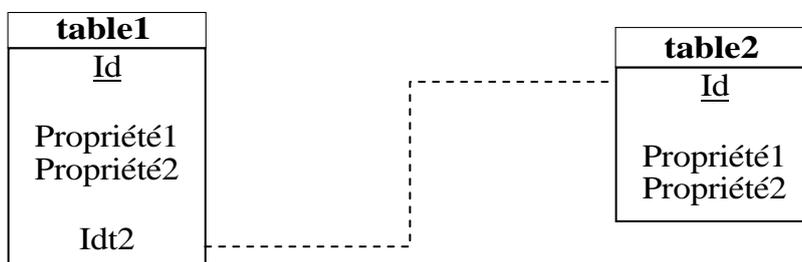
4.1 Association 1-*

On considère l'association suivante pour laquelle les entités de la table1 participent une seule fois à la relation, tandis que des entités de la table2 participent plusieurs fois à la relation.



Une association 1-* entre deux types d'entités table1 et table2 donne deux tables dans le modèle relationnel, les identifiants devenant les clés primaires de chacune des deux tables.

Cette association 1-* conduit à ajouter un attribut à la table table1 et à poser sur cet attribut une contrainte de **clé étrangère** en relation avec la clé primaire de la table table2.



Exemple. Dans le modèle relationnel, les deux types d'entités deviennent des tables Auteur et Livre avec leurs attributs respectifs et les identifiants idAuteur et idLivre deviennent les clés primaires des deux tables.

L'association 1-* conduit à ajouter un attribut à la table Livre, idAuteur et à poser sur cet attribut une contrainte de clé étrangère en relation avec la clé primaire idAuteur de la table Auteur.



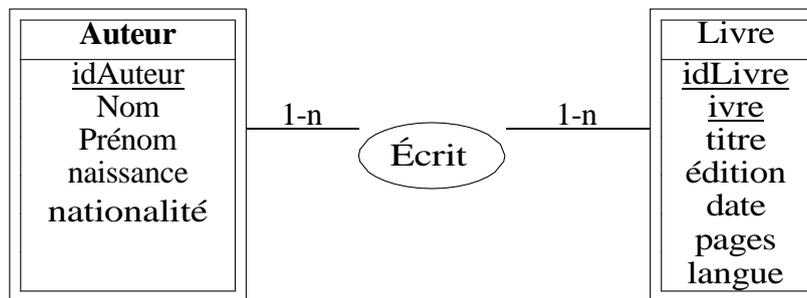
4.2 Association *-*

Une association *-* entre deux types d'entité table1 et table2 est décomposée en deux associations 1-*, **en ajoutant un type d'entités**. La traduction de ce modèle entités-associations en un modèle relationnel nécessite **trois tables**. La troisième table ajoutée contient deux clés étrangères chacune en relation avec une clé primaire de chaque table table1 et table2.

Exemple : Dans votre bibliothèque certains livres ont été écrits par plusieurs auteurs

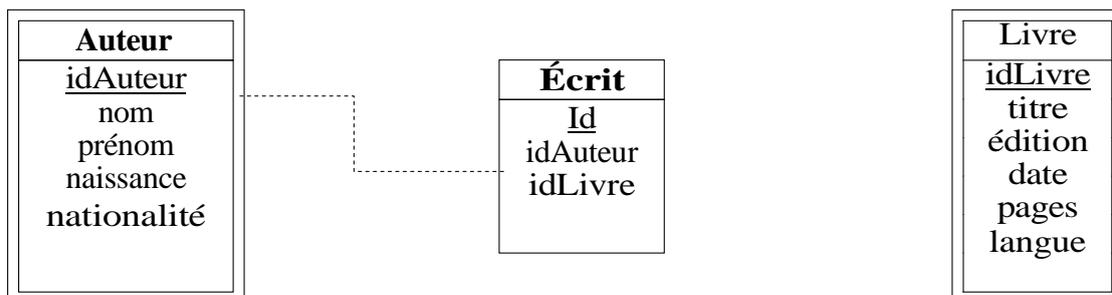
Chaque auteur a écrit au moins un livre, donc la cardinalité de l'association entre Auteur et Écrit est 1-n.

Un livre peut avoir été écrit par plusieurs auteurs, donc l'association entre Livre et Écrit est aussi 1-n.



Il s'agit d'une association *-* entre Auteur et Livre.

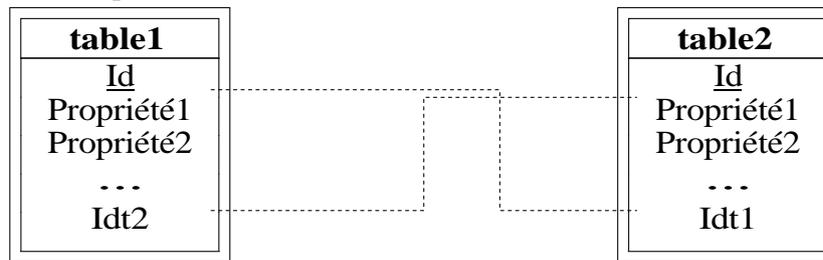
La traduction dans le modèle relationnel nécessite d'ajouter une table, Écrit, qui contient deux clés étrangères : idAuteur en relation avec la clé primaire idAuteur de la table Auteur, et idLivre en relation avec la clé primaire idLivre de la table Livre.



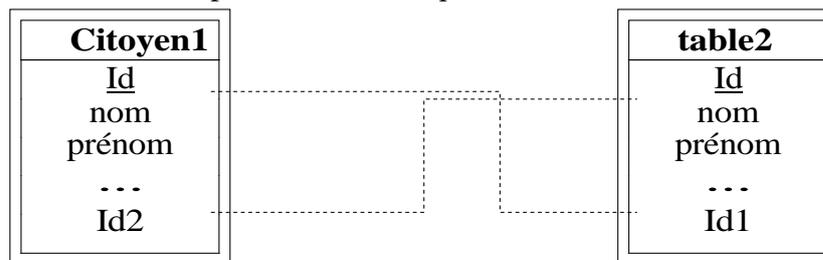
4.3 Association 1-1 :

Une association 1-1 entre deux types d'entité table1 et table2 signifie que chaque entité de table1 peut être en relation avec une et unique entité de table2.

Une association 1-1 entre deux entités est traduite par deux tables, et dans chaque table par la présence d'une clé étrangère en lien avec la clé primaire de l'autre table.



Exemple : Prenons par exemple le cas d'une mairie qui souhaite gérer les mariages entre ses administré.e.s. Chaque administré.e ne peut être marié.e qu'à un.e autre administré.e



5 Pour s'exercer: Conception Entité-Association traduction en modèle relationnel et implémentation

Une entreprise souhaite gérer les commandes de ses clients et a besoin d'une base de données pour ses applications. Tous les produits vendus disposent d'un code unique propre au produit, une désignation, et relèvent d'une ou plusieurs catégories. Chaque produit a un prix à l'unité, ce prix peut varier en fonction du temps mais il convient de ne pas perdre l'historique. Les catégories possèdent un nom, et seront identifiées par un numéro unique.

Les clients sont identifiés par un numéro unique interne à cette application, et une ville de résidence. On distinguera les particuliers, qui ont un nom et un prénom, des professionnels, qui ont une raison sociale et un numéro de SIRET. Un client ne pourra pas appartenir aux deux catégories à la fois.

Chaque client peut effectuer une commande à une date donnée (date et heure de validation de la commande, d'une précision à la minute). Une adresse de livraison est également enregistrée à la validation de la commande. Une commande contient un ensemble de produits ; on précise à chaque fois la quantité d'un produit dans une commande.

Chaque commande peut donner lieu à un ensemble de livraisons. Pour une commande donnée, une livraison est identifiée par sa date et l'heure de livraison. Pour chaque livraison on précise le transporteur, et la possibilité de rajouter des commentaires sur la livraison. Chaque livraison porte sur un ensemble de produits dans une quantité à préciser.

Pour la bonne cohérence des données, on cherchera à assurer que les produits livrés font bien partie de la commande concernée par la livraison.

Q1. Réaliser un schéma conceptuel dans le formalisme Entité-Association.

Q2. Faites apparaître le schéma relationnel (MLD textuel) correspondant à votre conception.

Q3. Observez ce schéma pour bien observer les mécanismes de traduction vus en cours.

Q4. Faites apparaître le code SQL permettant d'implémenter le schéma obtenu.

Le schéma relationnel à obtenir finalement doit être le suivant :

```
categories = (num_categorie SERIAL, nom_categorie VARCHAR(60) );
clients = (num_client SERIAL, ville_client VARCHAR(50) );
commandes = (#num_client, date_commande TIMESTAMP, adresselivraison_commande VARCHAR(100) );
professionnels = (#num_client, siret_professionnel VARCHAR(17), raisonsociale_professionnel VARCHAR(50) );
particuliers = (#num_client, nom_particulier VARCHAR(50), prenom_particulier VARCHAR(50) );
livraisons = (#(num_client, date_commande), date_livraison TIMESTAMP, transporteur_livraison VARCHAR(50), commentaire_livraison VARCHAR(50) );
produits = (code_produit VARCHAR(6), designation_produit VARCHAR(50), num_categorie);
historiquesprix = (#code_produit, datedebut_prix DATE, montant_prix NUMERIC(11,2) );
commandes_contient_produits = (#code_produit, #(num_client, date_commande), quantite_produit_commande INTEGER);
livraisons_livre_produits = (#code_produit, #(num_client, date_commande), date_livraison), quantite_produit_livraison INTEGER);
```