

MERISE

Guillaume Rodrigues

[NOM DE LA SOCIETE] [Adresse de la société]

Table des matières

1 Merise c'est quoi ?.....	2
Système d'Information (SI)	2
La méthode Merise	2
2 Modélisation d'une BDD au niveau conceptuel : le MCD	3
Elaboration des règles de gestion métier	3
Le dictionnaire de données	4
Exemple	4
Données calculées et non-calculées	6
Le Modèle Conceptuel des Données (MCD)	7
Les entités	7
Les associations.....	8
TD : Elaboration du MCD	10
3 Modélisation d'une BDD au niveau logique : le MLD	11
Passage du MCD au MLD	11
Règles de conversion.....	12
Conversion d'entités ayant au moins une cardinalité 1,1	12
Conversion d'entités ayant au moins une cardinalité « 0,1 ».....	13
Conversion d'entités n'ayant que des cardinalités de type « 0,N » ou « 1,N »	13
Cas particulier : associations réflexives	15
Cas particulier : associations n-aires (n>2).....	16
Exercices	17
4 Normalisation : Les Formes Normales.....	18
1 ^{ère} Forme Normale (1FN).....	18
2 ^{ème} Forme Normale (2FN).....	18
3 ^{ème} Forme Normale (3FN).....	18
Exercices Formes Normales.....	20
Exercices Formes Normales : Correction	21

1 Merise c'est quoi ?

Merise est une méthode d'analyse, de conception et de réalisation de système d'information

Système d'Information (SI)

Le système d'information ou SI, peut être défini comme étant l'ensemble des moyens humains, matériels et immatériels mis en œuvre afin de gérer l'information au sein d'une unité, une entreprise par exemple.

Le SI possède 4 fonctions principales :

- La saisie / collecte d'informations
- Le stockage d'information (Fichiers ou Base De Données)
- Le traitement de l'information (consultation, calculs, mises à jour)
- La diffusion de l'information

La méthode Merise

Méthode française développée dans les années 70 par Hubert Tardieu.

Elle fut mise en avant dans les années 80 à la demande du ministère de l'Industrie qui souhaitait une méthode de conception des SI.

Merise est donc une méthode d'analyse et de conception des SI basée sur le principe de la séparation des données et des traitements. Elle possède un certain nombre de modèles (ou schémas) qui sont répartis sur trois niveaux :

- Le niveau conceptuel
- Le niveau logique ou organisationnel
- Le niveau physique

2 Modélisation d'une BDD au niveau conceptuel : le MCD

Il s'agit de l'élaboration du **Modèle Conceptuel de Données (MCD)** : représentation graphique et structurée des données stockées sur un SI.

Le MCD est basé sur 2 éléments principaux : les **entités** et les **associations**.

L'élaboration d'un MCD passe par les étapes suivantes :

1. La mise en place des règles de gestion (si elles ne sont pas données)
2. L'élaboration du dictionnaire de données
3. La recherche de dépendances fonctionnelles entre ces données
4. L'élaboration du MCD (création des **entités**, des **associations**, puis ajout des **cardinalités**)

Elaboration des règles de gestion métier

Avant de créer entités et relations il faut centraliser les besoins métier et créer les règles de gestion.

Par exemple, si nous souhaitons informatiser le SI d'une ludothèque :

- Pour chaque jeu, on doit connaître le **titre**, l'**année** de parution, le/les **types** de jeu (jeu de plateau, jeu de dé, jeu de carte, jeu de coopération, jeu d'adresse, rôle, gestion, rôle caché, deck building)
- Un jeu peut avoir été créé par aucun, un ou plusieurs auteurs, et peut avoir été illustré par aucun, un ou plusieurs illustrateurs.
- Chaque exemplaire du jeu est identifié par une **référence** composée de lettres et de chiffres et ne peut être paru que dans une et une seule édition.
- Un adhérent à la ludothèque est identifié par un **numéro** et on doit stocker son **nom, prénom, adresse, numéro de téléphone, adresse mail**.
- Un adhérent peut faire 0, 1 ou plusieurs emprunts qui concernent chacun un et un seul exemplaire de jeu. Pour chaque emprunt, on connaît la **date d'emprunt** et la **durée maximale**.

Ces règles sont parfois données, mais il est bien souvent nécessaire de les créer soi-même ou de les préciser.

Le dictionnaire de données

Etape intermédiaire qui facilite les étapes suivantes.

Elle consiste à lister, à partir des règles de gestion métier, toutes les données qui devront être stockées en BDD et qui figureront dans le MCD. Pour chaque donnée, il indique :

- Le **code mnémonique** : libellé désignant une donnée (exemple : « titre_j » pour le titre du jeu)
- La **désignation** : description de la données (exemple : « Titre du jeu »)
- Le **type de donnée** :
 - A ou Alphabétique
 - N ou Numérique
 - AN ou Alphanumérique
 - Date (AAAA-MM-JJ)
 - Booléen
- La **taille** exprimée en nombre de caractères ou de chiffres.
- Des **remarques/observations**

Exemple

Si on reprend notre cas d'étude, on peut établir le dictionnaire de données suivant :

Code mnémonique	Désignation	Type	Taille	Observations
id_a	Identifiant numérique d'un adhérent	N		
nom_a	Nom d'un adhérent	A	50	
prenom_a	Prénom d'un adhérent	A	50	
rue_a	Rue de l'adhérent	AN	150	
ville_a	Ville de l'adhérent	A	50	
cp_a	Code Postal de l'adhérent	AN	5	
tel_a	Tel fixe de l'adhérent	AN	12	
tel_port_a	Tel mobile de l'adhérent	AN	12	
email_a	Email de l'adhérent	AN	100	
id_j	Identifiant numérique du jeu	N		
titre_j	Titre du jeu	A	50	
annee_j	Année de 1 ^{ère} parution du jeu	Date	4	AAAA
ref_e	Code de référence d'un exemplaire du jeu	AN	30	

Code mnémonique	Désignation	Type	Taille	Observations
id_t	Identifiant numérique d'un type de jeu	N		
libelle_t	Libellé d'un type de jeu	A	50	
id_ed	Identifiant numérique de l'édition d'un jeu	N		

nom_ed	Nom d'une édition d'un jeu	A	50	
annee_ed	Année de l'édition d'un jeu	Date	4	AAAA
id_au	Identifiant numérique d'un auteur de jeu	N		
nom_au	Nom d'un auteur de jeu	A	50	
prenom_au	Prénom d'un auteur de jeu	A	50	
id_i	Identifiant numérique d'un illustrateur de jeu	N		
nom_i	Nom d'un illustrateur de jeu	A	50	
prenom_i	Prénom d'un illustrateur de jeu	A	50	
id_p	Identifiant numérique d'un pays	N		
nom_p	Nom d'un pays	A	50	
id_em	Identifiant numérique d'un emprunt	N		
date_em	Date d'un emprunt	Date	10	AAAA-MM-JJ
delai_em	Délai maximal autorisé pour un emprunt	N	3	Nb de jours

Données calculées et non-calculées

Les données figurant dans le dictionnaire de données doivent la plupart du temps être des données élémentaires :

- Elles ne doivent pas être des données calculées, sauf lorsqu'il est nécessaire de les stocker pour des raisons pertinentes (optimisation).
- Elles ne doivent pas être composées : des données composées sont des données obtenues par concaténation de plusieurs données (ex adresse)

Certaines données de type numérique doivent être stockées en tant que donnée de type alphanumérique (exemple code postal, numéro de téléphone).

Le Modèle Conceptuel des Données (MCD)

Les entités

Chaque entité est unique et décrite par un ensemble de **propriétés**.

Une des propriétés de l'entité est **l'identifiant**. Cette propriété doit posséder des occurrences **uniques** et doit être source des **dépendances fonctionnelles** avec toutes les autres propriétés de l'entité. Bien souvent, on utilise une donnée de type entier qui s'incrémente pour chaque occurrence, ou encore un code unique spécifique du contexte.

Le formalisme d'une entité est le suivant :

Nom de l'entité	
<u>Identifiant</u>	
propriété	1
propriété	2
...	

Ainsi, en reprenant notre étude de cas, on peut schématiser une entité « Auteur » de la façon suivante :

Auteur
<u>id_au</u>
prenom_au
nom_au

On peut retomber sur la règle de gestion suivante : un auteur est identifié par un numéro unique (id_au), et caractérisé par un prénom et un nom.

Une entité peut n'avoir aucune, une ou plusieurs **occurrences**. Voici par exemple une table d'occurrences :

<u>id_au</u>	prenom_au	nom_au
1	Richard	Garfield
2	Klaus	Teuber
3	Donald X.	Vaccarino

Cette table est composée de 3 occurrences de la table auteur.

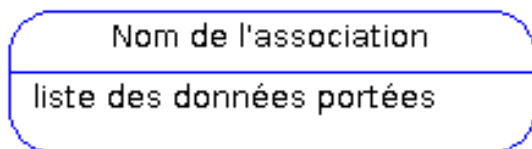
Les occurrences peuvent être appelées tuples, n-uplet, enregistrements

Les associations

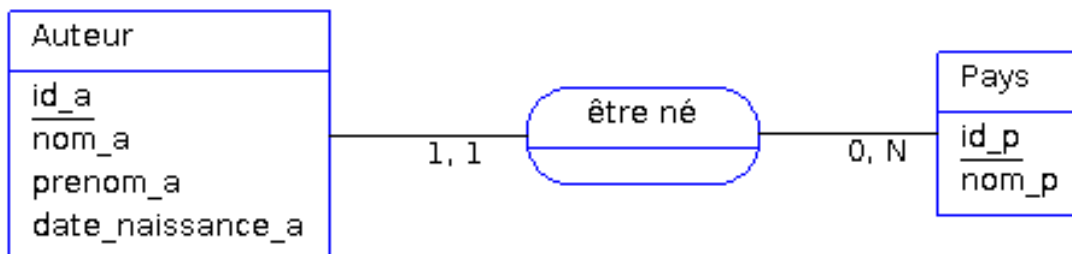
Une association définit un lien sémantique entre plusieurs entités.

La définition de liens entre entités permet de traduire une partie des règles de gestion qui n'ont pas été satisfaites par la simple définition des entités.

Le formalisme d'une association est le suivant :



Généralement le nom de l'association est un verbe définissant le lien entre les entités qui sont reliées par cette dernière. Par exemple :



Ici l'association « être né » traduit les deux règles de gestion suivantes :

- Un auteur est né dans un et un seul pays
- Dans un pays, sont nés aucun, un ou plusieurs auteurs.

Vous remarquerez que cette association est caractérisée par ces annotations 1,1 et 0,N qui nous ont permis de définir les règles de gestions précédentes. **Ces annotations sont appelées les cardinalités.**

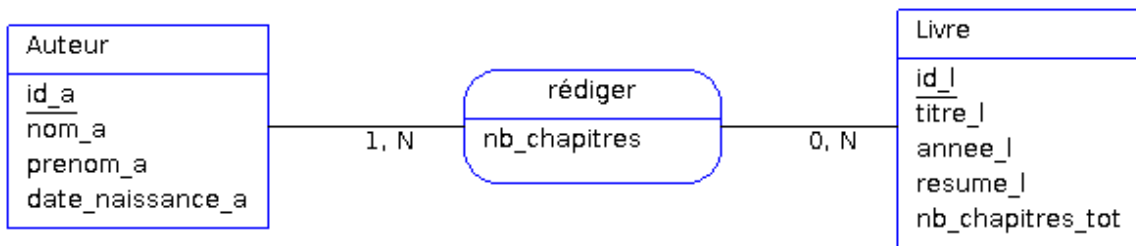
Une cardinalité est définie comme ceci : minimum, maximum

Les cardinalités les plus répandues sont les suivantes : 0,N ; 1,N ; 0,1 ; 1,1.

On peut toutefois tomber sur des règles de gestion imposant des cardinalités avec des valeurs particulières, mais cela reste assez exceptionnel et la présence de ces cardinalités imposera l'implantation de traitements supplémentaires.

L'identifiant d'une association ayant des cardinalités 0,N/1,N de part et d'autre, est obtenu par la concaténation des entités qui participent à l'association.

Imaginons l'association suivante :



Ici un auteur rédige au moins un ou plusieurs livres et pour chaque livre, on connaît le nombre de chapitres rédigés par l'auteur (on connaît aussi le nombre total de chapitres pour chaque livre).

L'association « rédiger » peut donc être identifiée par la concaténation des propriétés id_a et id_l.

Ainsi, le couple id_a, id_l doit être unique pour chaque occurrence de l'association.

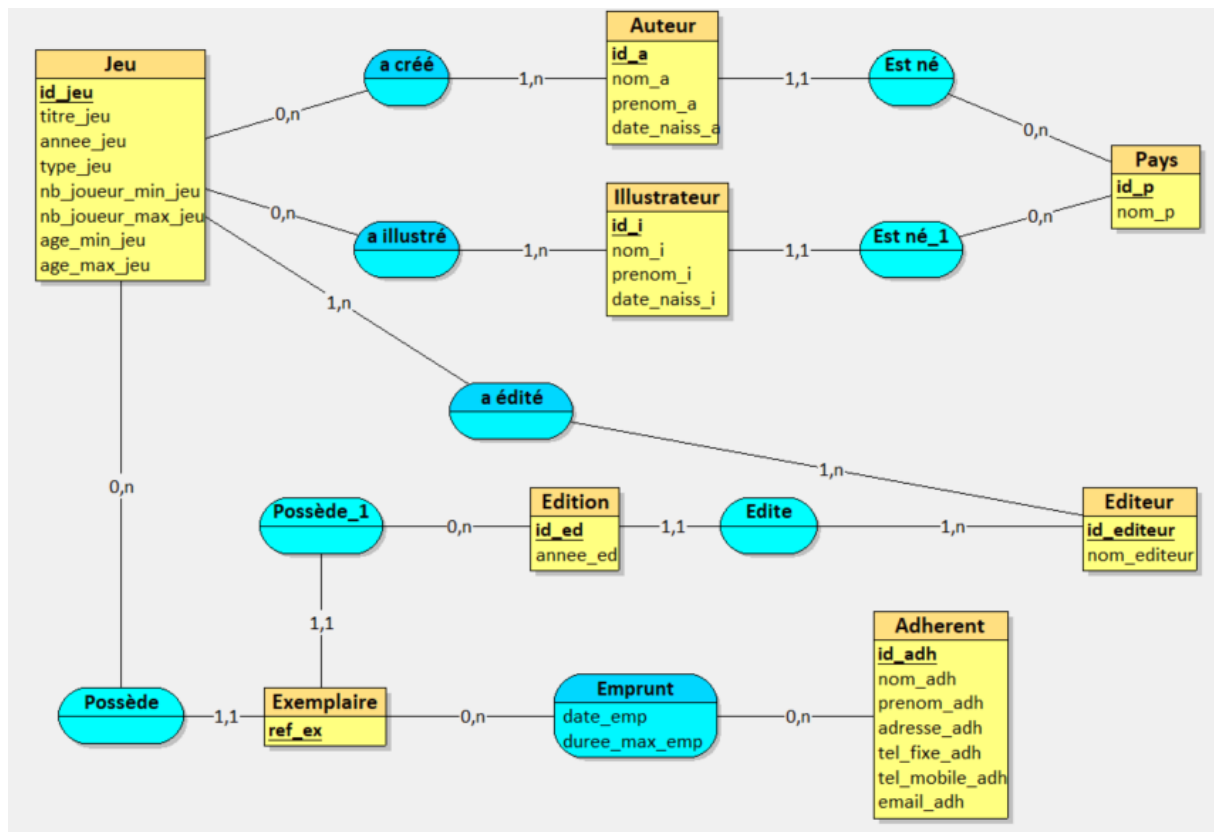
On dit que nb_chapitres (nombre de chapitres rédigés par un auteur, pour un livre) est une **donnée portée** par l'association « rédiger ». Cette association est donc une **association porteuse de données**.

Pour une association ayant au moins une cardinalité de type 0,1 ou 1,1 considérons dans un premier temps que cette dernière ne peut être porteuse de données et qu'elle est identifiée par l'identifiant de l'entité porteuse de la cardinalité 0,1 ou 1,1.

Nous reviendrons plus en détail sur la notion d'identification d'une association lors du passage au modèle logique.

TD : Elaboration du MCD

Nous pouvons maintenant élaborer le MCD à partir des données du dictionnaire de données.



3 Modélisation d'une BDD au niveau logique : le MLD

Une fois le MCD réalisé, nous voulons maintenant le transformer en Modèle Logique de Données (MLD) qui sera la représentation de notre Système d'Information en SQL.

Passage du MCD au MLD

Le modèle logique de données (MLD) est composé uniquement de ce que l'on appelle des **tables**.

Les **tables** sont principalement la résultante des **entités** précédemment modélisées dans le MCD, mais également de certaines **associations**. Les relations seront les tables de notre Base De Données.

Les **propriétés** des entités précédemment créées deviendront des **attributs**. Les attributs sont des données élémentaires mais aussi des identifiants et des données portées par certaines relations.

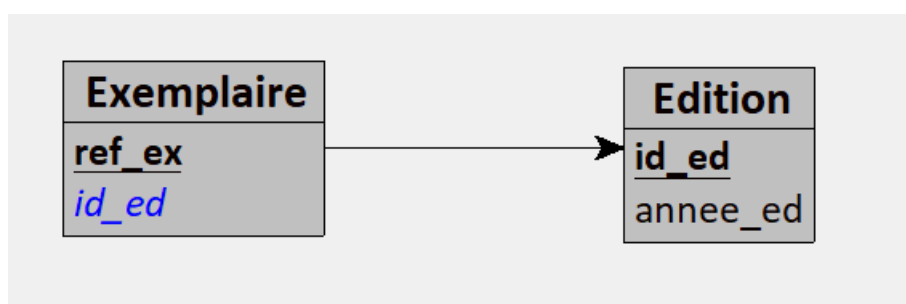
Une **table possède un nom** qui correspond généralement à celui de l'entité ou de l'association lui correspondant.

Une **entité possède une clé primaire**, qui permet **d'identifier** sans ambiguïté une **occurrence** de la table.

La clé primaire sera souvent un identifiant numérique unique mais pourra parfois être une clé composée de plusieurs attributs.

Le passage du MCD au MLD va également faire apparaître la notion de **clé étrangère**, représentant les liens entre les différentes tables. Une clé étrangère est un **attribut** faisant **référence** à une **clé primaire** d'une autre table. Elles sont les résultantes directes des **cardinalités**.

Voici une conversion en MLD de 2 entités de l'exemple précédent :



Ici, le passage du MCD au MLD a créé une clé étrangère `id_ed` dans la table `Exemplaire`, directement liée à l'identifiant de la table `Edition`.

Règles de conversion

En règle générale, toute entité du MCD devient une table dont la clef primaire est l'identifiant de cette entité. Chaque propriété de l'entité devient un attribut de la table correspondante.

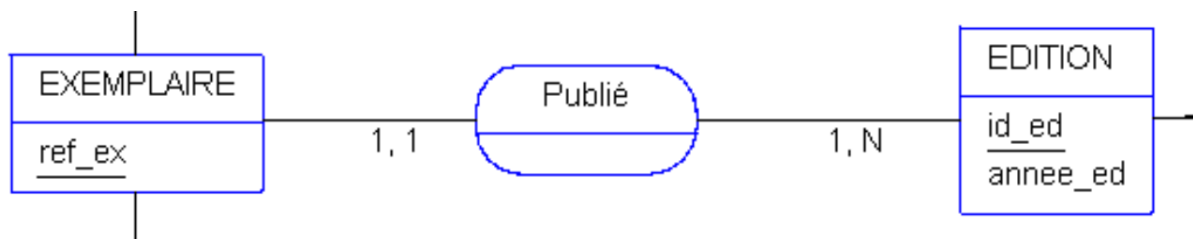
Pour définir les règles de conversion, **on se basera toujours sur les cardinalités maximales** (1 ou N) de chaque côté de l'association.

Conversion d'entités ayant au moins une cardinalité 1,1

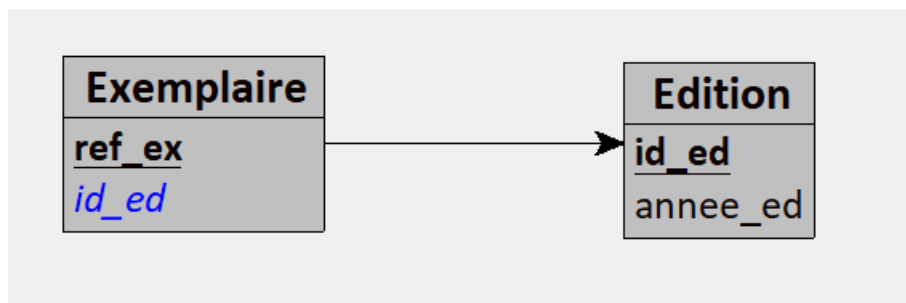
Dans ce cas, nous allons ajouter une clé étrangère dans la table qui correspond à l'entité se situant du côté de la cardinalité « 1,1 ».

C'est le cas pour les tables « Exemple » et « Edition » vues précédemment.

1. MCD des entités Exemple et Edition



2. MLD des tables Exemple et Edition



Lorsque l'on applique cette règle de conversion, deux restrictions s'imposent :

- L'association **ne peut être porteuse de données**. Les données portées sont en dépendances fonctionnelles directes avec l'identifiant de l'entité dont la clef correspondante sera référencée par une clef étrangère dans une autre table.
- L'association doit être binaire (c'est-à-dire relier uniquement deux entités et pas plus).

Lorsque deux entités sont toutes deux reliées avec une cardinalité 1,1 par une même association, sauf exception les entités devraient le plus souvent fusionner. Mais dans des cas spécifiques, on peut souhaiter conserver une clé étrangère. Dans ce cas elle sera placée d'un côté ou de l'autre de la table.

Conversion d'entités ayant au moins une cardinalité « 0,1 »

Si la cardinalité est « 0,1 », nous avons 2 possibilités :

- Nous pouvons reprendre le formalisme précédent : une clé étrangère fait directement référence à la clé primaire d'une autre table. Mais cette clé étrangère pourra être « nullable ».
- Il existe aussi la solution de passer par une table intermédiaire, qui serait dans l'exemple ci-dessous nommée « publie » ou « exemplaire_edition » et composée de : Publie(ref_ex,id_ed).

Le principal avantage par rapport à la solution précédente est de n'avoir à aucun moment de clé étrangère « null » en base de données, mais cela peut alourdir la compréhension du modèle.

Conversion d'entités n'ayant que des cardinalités de type « 0,N » ou « 1,N »

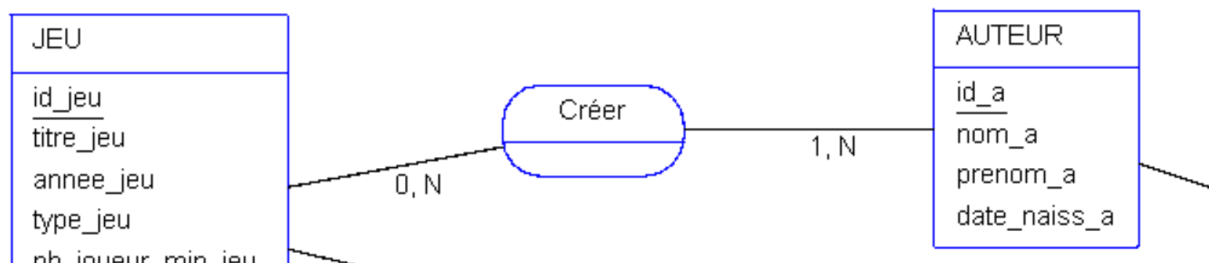
Une association ayant des cardinalités « 0,N » ou « 1,N » de chaque côté deviendra une table dont la clé primaire sera constituée des identifiants des entités reliées par cette association.

On parle dans ce cas de tables associatives.

Ces tables peuvent être porteuses de données.

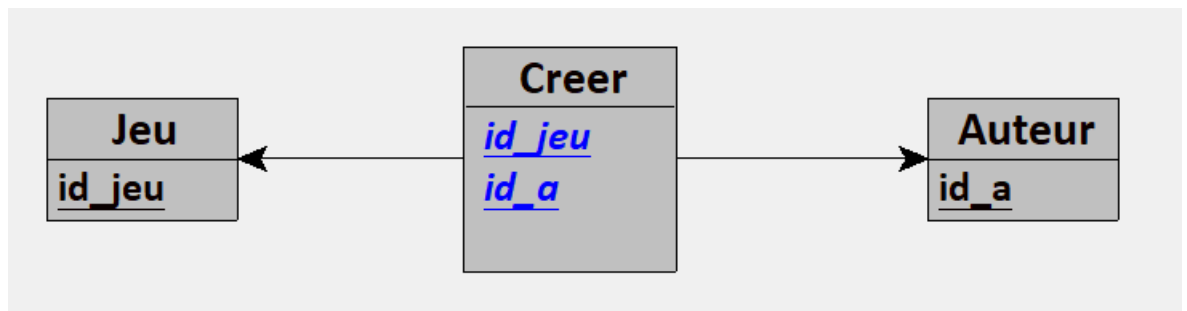
Dans notre MCD, nous avons les entités « Jeu » et « Auteur » qui correspondent à ce cas.

1. MCD des entités Jeu et Auteur



2. MLD des tables Jeu et Auteur

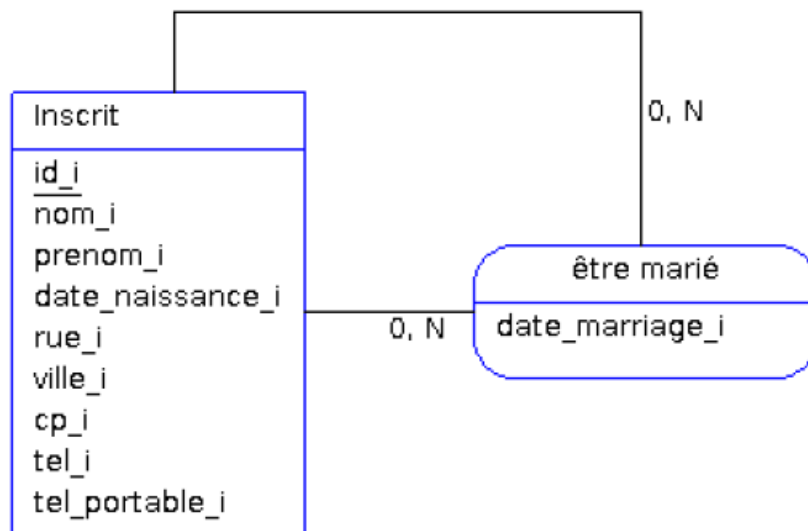
Le MLD résultant de cette association sera le suivant (je n'ai pas recopié les propriétés titre_jeu, annee_jeu, type_jeu, mais dans un cas réel il faudra évidemment les reprendre) :



Cas particulier : associations réflexives

Il est possible de relier une entité à elle-même par une association, on parle dans ce cas-là d'association réflexive.

Imaginons que l'on veuille connaître les inscrits qui sont mariés entre eux tout en conservant leur date de mariage, voici ce que l'on obtiendrait au niveau conceptuel :



Dans ce cas, c'est la même. Il faudra cependant différencier les noms des clefs étrangères de la table associative correspondantes tout en référençant la même clef primaire :

Inscrit (id_i, nom_i, prenom_i, date_naissance_i, rue_i, ville_i, cp_i, email_i, tel_i, tel_portable_i)

EtreMarie (id_epoux#, id_epouse#, date_mariage_i)

Légende :

x : table

x : clef primaire

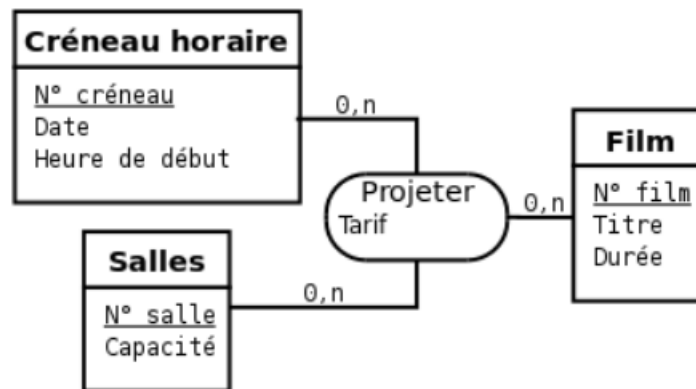
x# : clef étrangère

Nous reviendrons plus en détail sur ce cas particulier lors du cours SQL.

Cas particulier : associations n-aires (n>2)

Ce type association met en relation n type entité. Même s'il n'y a, en principe, pas de limite sur le nombre d'entités d'une association, dans la pratique on va rarement au-delà de trois.

Exemple d'une association ternaire

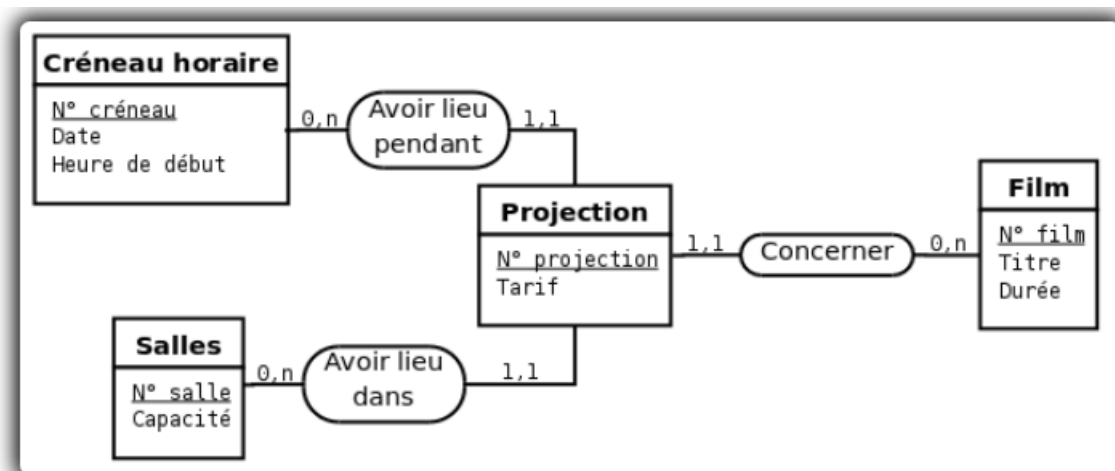


Ici, l'association porteuse de données « Projeter » nous permettra de connaître le créneau horaire de la projection, la salle dans laquelle aura lieu la projection et le film projeté.

Il est toujours possible de s'affranchir d'une association n-aire en procédant de la façon suivante :

- On remplace l'association n-aire par une entité auquel on attribue un identifiant.
- On crée des associations binaires entre la nouvelle entité et toutes les entités de l'association n-aire précédente.
- La cardinalité de chaque association binaire créée sera « 1,1 » du côté de la nouvelle entité créée et « 0,n » ou « 1,n » du côté des entités de la collection de l'ancienne association n-aire.

Le résultat de la conversion donnerait ceci :



Exercices

Faire la conversion du MCD vers le MLD pour les entités « Jeu », « Adhérent » et « Exemple ».

4 Normalisation : Les Formes Normales

Une mauvaise conception des entités et associations représentant le monde réel modélisé conduit à des relations problématiques.

Une redondance des données conduit à des risques d'incohérences lors des mises à jour.

Il est donc important d'éliminer toute anomalie afin de faciliter la manipulation des relations.

Pour cela, il est nécessaire de normaliser les relations.

Il existe 9 formes normales mais seules les 3 premières sont fondamentales pour s'assurer de la cohérence et de l'intégrité des données dans un SGBDr (Système de Gestion de Base de Données Relationnelles).

1^{ère} Forme Normale (1FN)

Conditions :

- Tous les attributs doivent posséder une valeur atomique (valeur non-subdivisible)
- Par exemple, un numéro de sécu n'est pas atomique (composé de parties distinctes : sexe, date de naissance, identifiant commune, ...)
- En revanche une date est atomique

2^{ème} Forme Normale (2FN)

Conditions :

- Respecter la 1FN
- Un attribut non-identifiant ne dépend pas d'une partie de l'identifiant mais de tout l'identifiant (cas des clés composées)

La deuxième forme normale évite une des principales sources de redondance dans une relation : la présence d'attributs qui ne dépendent que d'une partie de la clé. De cette définition, on peut déduire immédiatement qu'une relation dont la clé est composée d'un unique attribut est toujours en 2NF (si elle est aussi en 1NF).

3^{ème} Forme Normale (3FN)

Conditions :

- Respecter la 2FN
- Un attribut non-identifiant ne dépend pas d'un ou plusieurs attributs de participant pas à l'identifiant

Outre qu'elle permet d'identifier une nouvelle source de redondance dans laquelle un attribut non-clé dépend fonctionnellement d'un (ensemble de) d'attribut(s) non-clé(s), il est important de souligner, comme pour la 2NF, la progressivité forcée des formes normales. En effet, le critère de passage en 3NF est totalement orthogonal à celui de la 2NF.

Exercices Formes Normales

Exercice 1

Pour chacune des relations suivantes et d'après vos connaissances en SQL, indiquez si la relation est 1NF en identifiant en quoi chaque structure serait complexe à interroger en SQL.

1. LIVRAISON(n°fournisseur, listeVilles)
2. LIVRAISON (n°fournisseur, ville)
3. CLIENT (n°client, nom, prénoms)
4. CLIENT (n°client, nom, prénom1, prénom2)
5. CLIENT (n°client, nom, prénom, adresse)

Exercice 2

Pour chacune des relations suivantes, et d'après vos connaissances sur les données concernées, proposez des dépendances fonctionnelles et indiquez si la relation est 2NF :

1. PRET (n°isbn, n°adherent, date, nom_adherent, ville_adherent, titre_livre)
2. PRET (n°isbn, date, n°adherent, nom_adherent, ville_adherent, titre_livre)
3. PRET (n°isbn, n°adherent, date)
4. PRET (n°exemplaire, date, n°adherent)

Exercice 3

Pour chacune des relations suivantes et d'après vos connaissances sur les données concernées, identifiez les dépendances fonctionnelles et indiquez si la relation est en troisième forme normale ou non. Concluez en spécifiant le risque de redondance.

1. FOURNISSEUR (n°fournisseur, ville, pays) dans le contexte où un fournisseur n'est établi que dans une seule ville (considérez le cas où chaque nom de ville est unique et le cas où des villes de même nom peuvent se retrouver dans différents pays).
2. PERSONNEL (n°agent, nom, département_recherche, bâtiment)
3. PERSONNEL (n°agent, nom, département_recherche, statut_agent)
4. VOL(n°vol, compagnie, heure, destination, modele_avion, nombre_passagers)
5. VOL(n°vol, compagnie, heure, destination, modele_avion, nombre_places)

Exercices Formes Normales : Correction

Exercice 1

1. LIVRAISON(n°fournisseur, listeVilles) : **pas en 1NF**, le stockage d'un attribut qui contient une liste impose une syntaxe SQL complexe pour \"sortir\" chaque ville et l'interroger.
2. LIVRAISON (n°fournisseur, ville) : **1NF**, la clé sur fournisseur implique qu'un fournisseur ne fournisse qu'une ville, ou alors, il faut mettre la clé sur fournisseur et ville.
3. CLIENT (n°client, nom, prénoms) : **pas en 1NF**, même explication que 1.
4. CLIENT (n°client, nom, prénom1, prénom2) : **1NF**, les prénoms sont indiqués dans des champs différents.
5. CLIENT (n°client, nom, prénom, adresse) : **pas en 1NF**, si on veut récupérer en SQL certains éléments de l'adresse comme la ville par exemple. Si l'on veut récupérer l'adresse comme un tout, la relation peut alors être considérée en 1NF.\" %}

Exercice 2

1. PRET (n°isbn, n°adherent, date, nom_adherent, ville_adherent, titre_livre) : **pas en 2NF**, la relation est en 1NF car n°adhérent → nom_adherent et n°isbn → titre_livre.
2. PRET (n°isbn, date, n°adherent, nom_adherent, ville_adherent, titre_livre) : **pas en 2NF**, la relation est en 1NF car n°isbn → titre_livre.
3. PRET (n°isbn, n°adherent, date) : **2NF**, tous les attributs font partie de la clé !
4. PRET (n°exemplaire, date, n°adherent) : **2NF**

Exercice 3

1. FOURNISSEUR (n°fournisseur, ville, pays) : **pas en 3NF**, si ville → pays, sinon 3NF, cas de Brest qui peut être en France ou en Russie...
2. PERSONNEL (n°agent, nom, département_recherche, bâtiment) : **3NF**, si l'on prend le cas de notre école où un département peut être réparti sur plusieurs bâtiments. **2NF**, uniquement si département_recherche → bâtiment.
3. PERSONNEL (n°agent, nom, département_recherche, statut_agent) : **3NF**
4. VOL(n°vol, compagnie, heure, destination, modele_avion, nombre_passagers) : **3NF**
5. VOL(n°vol, compagnie, heure, destination, modele_avion, nombre_places) : **pas en 3NF**, uniquement en 2NF car modele_avion → nombre_places.