

Compléments sur le Modèle Entité / Association

1 Les associations n-aires

Nous avons vu jusqu'à présent des associations qui reliaient uniquement deux entités. On parle alors d'association binaires.

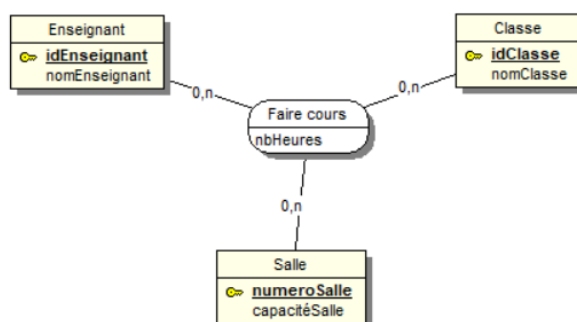
Mais il existe aussi des associations plus complexes qui peuvent relier N entités (avec $N > 2$). On parle alors d'associations n-aires. Les plus communes de ces associations n-aires sont les associations ternaires qui relient trois entités. Tout comme les associations binaires, les associations n-aires (et ternaires) peuvent être non porteuses de propriétés ou bien porteuses.

1.1 Les associations n-aires porteuses

Les associations n-aires porteuses sont généralement faciles à trouver car la (ou les) propriété portée par l'association dépend de toutes les entités reliées à l'association n-aire ; et cette propriété ne peut donc se trouver que dans cette ternaire.

Par exemple, on a les 3 entités suivantes : Classe – Salle – Enseignant. Et on souhaite connaître le nombre d'heures de cours (nbHeures) que dispense un enseignant avec une classe dans une salle donnée. En utilisant des associations binaires, il n'est pas possible de placer correctement la propriété nbHeures. En effet, si on place nbHeures dans une binaire entre **Classe et Enseignant**, on aura alors le nombre d'heures que dispense un enseignant à une classe toutes salles confondues. Si on place nbHeures dans une binaire entre **Classe et Salle**, on aura alors le nombre d'heures que passe une classe dans une salle quel que soit l'enseignant. Enfin, si on place nbHeures dans une binaire entre **Enseignant et Salle**, on aura alors le nombre d'heures que passe un enseignant dans une salle quel que soit la classe.

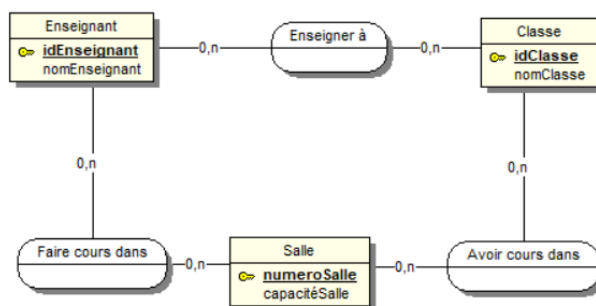
La seule solution est donc de mettre la propriété nbHeures dans une ternaire qui relie **Classe, Salle et Enseignant**.



1.2 Les associations n-aires non porteuses

Les association n-aires non porteuses sont beaucoup plus difficiles à trouver.

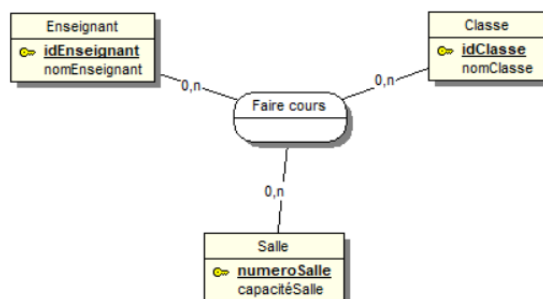
Par exemple, avec les mêmes 3 entités Classe – Salle – Enseignant, on souhaite ici connaître les salles dans lesquelles un enseignant fait cours à une classe. Si on essaye de modéliser ce problème avec des binaires on peut obtenir le modèle suivant :



Mais cette solution n'est pas satisfaisante.

En effet, avec ces trois binaires, on peut savoir à quelle classe enseigne un enseignant, dans quelle salle fait cours un enseignant et dans quelle salle une classe suit ses cours. Mais on ne sait pas dans quelle salle un enseignant fait cours à une classe. Par exemple, si on sait que l'enseignant M. Latronche fait cours à la classe S2 (et à d'autres classes). Qu'il fait cours dans la salle 018 (et dans d'autres salles avec d'autres classes). Et que la classe S2 a des cours en salle 018 (et dans d'autres salles avec d'autres enseignants). On ne peut pas en déduire que M. Latronche fait forcément cours en salle 018 avec la classe S2.

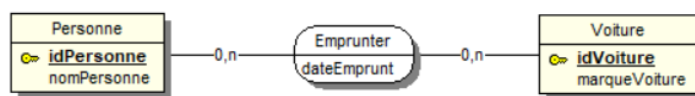
La seule modélisation qui permet de savoir dans quelle salle un enseignant fait cours à une classe est la ternaire suivante :



1.3 Les associations n-aires permettant de gérer l'historique

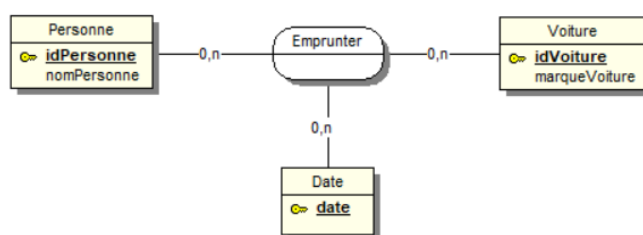
Les associations n-aires (notamment les ternaires) sont souvent utilisées pour gérer des historiques par rapport à des dates (ou des mois, des années, ...). A ce moment-là, une des entités reliée à la ternaire va être une entité 'spatiotemporelle' (Date, Mois ou Année).

Par exemple sur le modèle suivant :

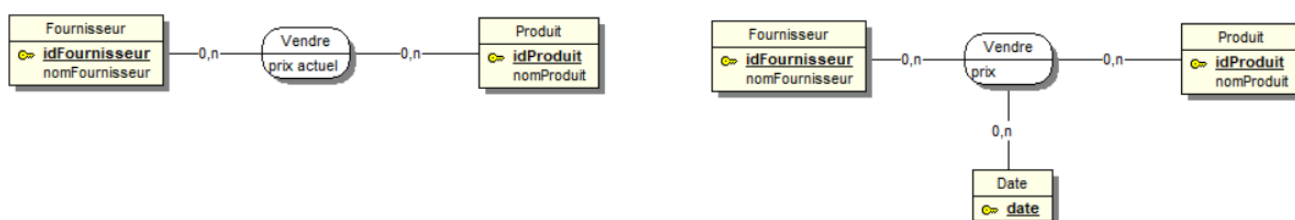


Ici, une personne peut emprunter plusieurs voitures et une voiture peut être empruntée par plusieurs personnes. Mais pour une personne donnée et une voiture donnée, on n'a qu'un seul emprunt et donc une seule date d'emprunt. On ne peut donc stocker que le dernier emprunt (et la date de cet emprunt) qu'une personne a fait avec une voiture.

Si on souhaite avoir l'historique de tous les emprunts d'une personne avec une voiture, une solution peut être de transformer la binaire en une ternaire en rajoutant une patte sur une entité date.



Autre exemple. Dans la solution de gauche, on n'a que le prix actuel pratiqué par un fournisseur sur un produit (pour un fournisseur donné et un produit donné, on n'a qu'un seul prix). Si on veut l'historique des prix pratiqués par un fournisseur sur un produit, on peut alors modéliser la ternaire de droite.



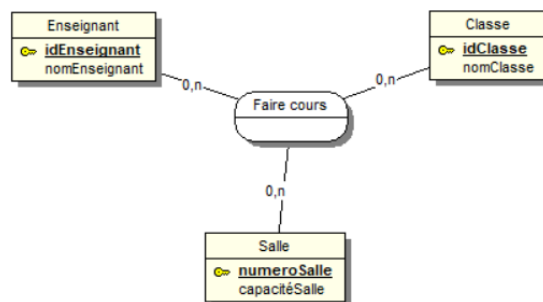
1.4 Quelques remarques sur les associations n-aires

Pour être juste, une association n-aire ne doit pas être décomposable en association possédant moins de pattes. Par exemple une association ternaire juste ne doit pas être décomposable en association binaire.

Si elle est décomposable, elle est forcément fausse.

On peut vérifier cela en s'assurant que toutes les occurrences des entités reliées à la n-aire doivent pouvoir correspondre à plusieurs occurrences de toutes les autres entités de la n-aire.

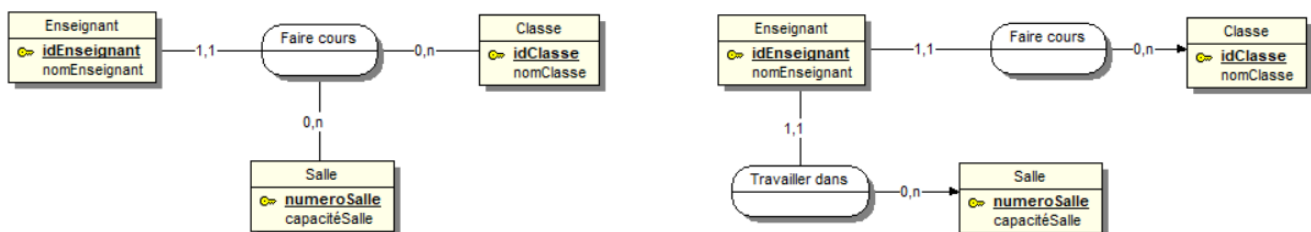
Par exemple, avec la ternaire suivante :



Pour que cette ternaire soit juste, il faut qu'un enseignant puisse enseigner à plusieurs classes. Qu'il puisse également enseigner dans plusieurs salles. Il faut aussi qu'une classe puisse avoir plusieurs enseignants et qu'une classe puisse avoir des cours dans plusieurs salles. Enfin, il faut aussi qu'une salle puisse accueillir plusieurs enseignants et qu'elle puisse également accueillir plusieurs classes. Si toutes ces conditions ne sont pas réunies, la ternaire est forcément fausse.

On en déduit donc que les cardinalités maximales de toutes les pattes d'une ternaire doivent être obligatoirement plusieurs. Si une d'entre elles vaut 1, la ternaire est forcément fausse et décomposable.

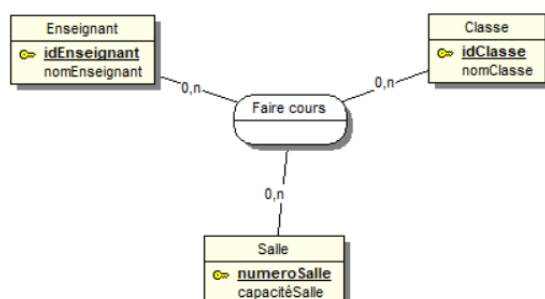
Par exemple, la ternaire de gauche est fausse. En effet, si un enseignant ne peut faire qu'un seul cours au maximum, cela veut dire qu'il ne fait cours qu'à une seule classe et dans une seule salle. La ternaire est donc fausse et doit être décomposée en deux binaires (à droite).



1.5 Passage au schéma relationnel

En l'absence de contraintes que nous verrons au paragraphe suivant, le passage au Schéma Relationnel d'une n-aire s'opère comme pour une association binaire de type plusieurs-plusieurs. On crée donc une nouvelle table et on met en clé étrangère et primaire tous les identifiants des entités reliées à l'association n-aire.

Par exemple, le modèle e/a de gauche donnera le schéma relationnel de droite :



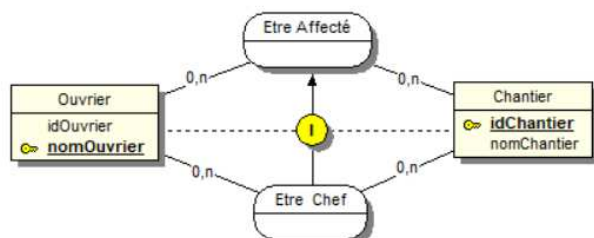
Enseignants (idEnseignant, nomEnseignant)
 Classes (idClasse, nomClasse)
 Salle (numeroSalle, capacitéSalle)
Cours (idEnseignant#, idClasse#, numeroSalle#)

2 Contraintes sur les associations

2.1 Contrainte d'inclusion

Une contrainte d'inclusion entre deux associations permet d'indiquer qu'une occurrence d'une (ou plusieurs) entité ne peut pas avoir une occurrence d'une association s'il n'a pas une occurrence de l'autre association.

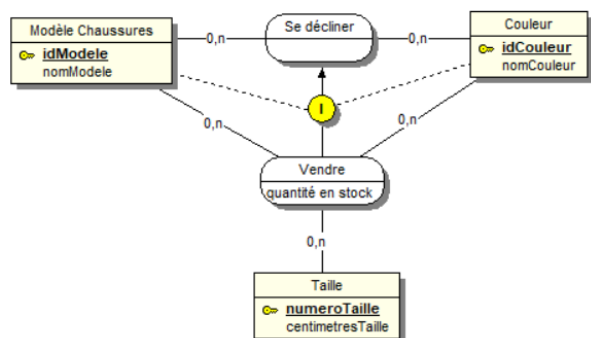
Ici, la contrainte d'inclusion permet d'indiquer qu'un ouvrier ne peut pas être chef d'un chantier sur lequel il n'est pas affecté. Cette contrainte peut facilement être représentée au niveau du Schéma Relationnel grâce à une clé étrangère



Ouvriers (idOuvrier, nomOuvrier)
 Chantiers (idChantier, nomChantier)
 EtreAffecte (idOuvrier#, idChantier#)
 EtreChef ((idOuvrier, idChantier)#)

On peut également trouver des contraintes d'inclusion entre des binaires et des ternaires.

Ici la contrainte d'inclusion permet d'indiquer qu'il n'est pas possible de vendre un modèle de chaussure dans une taille et une couleur (et donc de posséder un stock pour une modèle une taille et une couleur) si ce modèle de chaussures ne se décline pas dans la couleur en question.

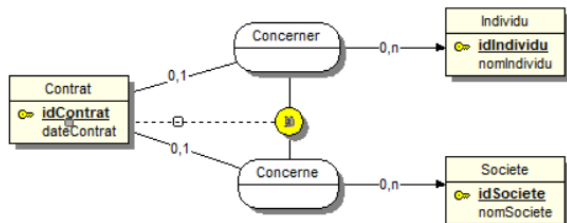


Modele (idModele, nomModele)
 Couleurs (idCouleur, nomCouleur)
 Tailles (numeroTaille, cmTaille)
 SeDecliner (idModele#, idCouleur#)
 Vendre ((idModele, idCouleur)#, numeroTaille#, qte)

2.2 Contrainte d'exclusion

Une contrainte d'exclusion entre deux associations permet d'indiquer qu'une occurrence d'une entité ne peut pas avoir d'occurrences des deux associations.

Par exemple, dans le modèle suivant, la contrainte d'exclusion indique qu'un contrat ne peut pas concerner à la fois un individu et une société. Il concerne soit l'un soit l'autre. Au niveau du Schéma Relationnel cette contrainte ne peut pas être représentée, mais il est possible de l'implémenter en SQL au niveau physique grâce aux contraintes CHECK.



Individus (idIndividu, nomIndividu)
 Societes (idSociete, nomSociete)
 Contrats (idContrat, dateContrat, idIndividu#, idSociete#)

```
CREATE TABLE Individus
(idIndividu VARCHAR(4), nomIndividu VARCHAR(20),
CONSTRAINT pk_Individu PRIMARY KEY (idIndividu));

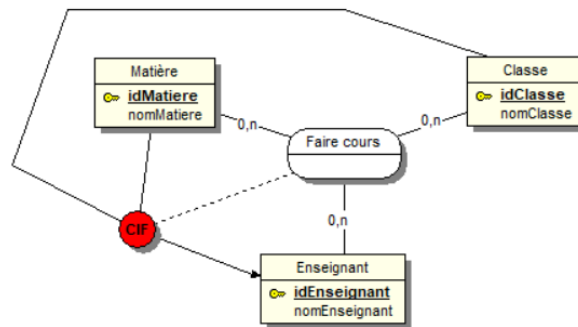
CREATE TABLE Societes
(idSociete VARCHAR(4), nomSociete VARCHAR(20),
CONSTRAINT pk_Societe PRIMARY KEY (idSociete));

CREATE TABLE Contrats
(idContrat VARCHAR(4), dateContrat DATE, idIndividu VARCHAR(4), idSociete VARCHAR(20),
CONSTRAINT pk_Contrat PRIMARY KEY (idContrat),
CONSTRAINT fk_Contrat_Societe FOREIGN KEY (idSociete) REFERENCES Societes(idSociete),
CONSTRAINT fk_Contrat_Individu FOREIGN KEY (idIndividu) REFERENCES Individus(idIndividu),
CONSTRAINT ck_Contrat_eXclusion CHECK (idSociete IS NULL OR idIndividu IS NULL));
```

2.3 Contrainte d'unicité sur une ternaire (ou CIF : contrainte d'intégrité fonctionnelle)

La Contrainte d'Intégrité Fonctionnelle sur une association ternaire (ou n-aire) permet d'indiquer que les occurrences de deux entités reliées à la ternaire permettent de donner l'occurrence de la troisième entité reliée à la ternaire. On dit qu'il y a une contrainte d'unicité sur la ternaire.

Sur la ternaire suivante, il est possible de représenter la CIF qui indique que pour une classe et une matière donnée, il n'y a qu'un seul enseignant (unicité). Par exemple, en classe 6°3, les mathématiques ne peuvent être enseignés que par un seul enseignant : M. Latronche.



Au niveau du Schéma Relationnel, cette contrainte d'unicité va changer la clé primaire de la table déduite par la ternaire (ici la table cours). Pour justement empêcher que pour une classe et une matière donnée on puisse avoir plusieurs enseignants.

Enseignants (idEnseignant, nomEnseignant)

Classes (idClasse, nomClasse)

Matieres (idMatiere, nomMatiere)

Cours (idClasse#, idMatiere#, idEnseignant#)

C1	M1	E1
C2	M2	E2
C2	M2	E1

impossible, cette ligne a la même clé primaire que la précédente
Il ne peut y avoir qu'un enseignant de la matière M2 dans la classe C2