

A dark blue vertical bar on the left side of the page. A blue arrow-shaped banner points to the right from the bar, containing the text 'Automne 2019'.

Automne 2019

Les modèles de données

Le modèle Entités Associations(E/A)

Several thin, curved lines in shades of blue and grey originate from the bottom left corner and extend upwards and to the right, creating a decorative graphic element.

Saliha Yacoub
COLLÈGE LIONEL GROULX

Table des matières

| | |
|--|----|
| Niveau conceptuel : | 2 |
| Niveau logique : | 2 |
| Niveau externe : | 3 |
| Le modèle Entité/Relation ou entité association | 3 |
| Concepts du modèle E/A : définitions | 3 |
| Relation ou association : | 4 |
| Généralisation d'entité. | 5 |
| Définition, cardinalité | 5 |
| Avantages du modèle Conceptuel : | 6 |
| Exemple de représentation dans Entity Framework | 6 |
| Exemple Oracle : | 7 |
| Normalisation du modèle Entités-Association | 8 |
| Passage du modèle E/R au modèle Relationnel | 8 |
| Règle 1 : Entité excluant les entités surtype et sous-type : | 8 |
| Règle 2 : Relation binaire avec participation multiple : | 8 |
| Règle 3 : Relation binaire avec participation multiple d'un côté et unique de l'autre côté | 9 |
| Règle 4 : Relation binaire avec participation d'un côté et unique des deux côtés | 9 |
| Règle 5: Relation récursive avec participation multiple. | 10 |
| Règle 6: Relation récursive avec participation unique d'un côté | 10 |
| Règle7: Relation entité surtype et sous-type.(Héritage) | 11 |
| Exemple Formalisme Oracle, passage du modèle E/A au modèle Relationnel. | 12 |
| Utilisation de Visusal Studio 2019 pour réaliser un modèle E/A | 12 |

Modélisation des données

Un modèle de données est un ensemble de concepts utilisés pour décrire la structure d'une base de données. Par structure de base de données, nous entendons, les types de données, les relations, et les contraintes qui définissent le gabarit de la base de données.

Tout modèle peut être exprimé à divers niveaux de précision, conceptuel, logique physique et externe. Cette définition sera retenue lorsque nous présenterons la partie

Les techniques de modélisation de données permettent d'une part de comprendre le fonctionnement d'un système et la façon dont les données sont organisées et d'autres part de construire un système d'information qui reflète la réalité.

Il existe plusieurs niveaux de représentation (ou de modélisation des données).

Niveau conceptuel :

Le niveau conceptuel ou le schéma conceptuel de la base de données décrit la réalité du système d'information d'une organisation indépendamment du SGBD d'implantation. Le modèle utilisé à ce niveau peut-être le modèle d'Entité/Association (modèle d'Entité/Relation) ou un modèle de classes Entités. Ce niveau est de la responsabilité de l'analyste concepteur ou de l'administrateur de la base de données.

Niveau logique :

Le niveau interne spécifie comment les données sont représentées ou stockées sur les supports de stockage. Ce niveau peut être représenté par un modèle relationnel. Ce niveau est complètement pris en charge par les SGBD. Ce qui veut dire que c'est un modèle d'implémentation qui tient compte du SGBD.

Niveau externe :

Ce niveau correspond à la vision de tout ou une partie du schéma conceptuel par un groupe d'utilisateurs concernés par une application ou une partie de l'application. Il s'agit de décrire à l'aide de schémas externes (Views) la façon dont seront perçues les données par les programmes.

Le modèle Entité/Relation ou entité association

Ce modèle appelé encore le MCD (Modèle Conceptuel de Données) permet de représenter les données d'un système sans se préoccuper du SGBDR (Système de Gestion de Base de Données Relationnel) qui sera utilisé ultérieurement. C'est le modèle qui se rapproche le plus du formalisme (modèle) d'oracle

- Représente le niveau conceptuel d'une base de données
- Permet d'obtenir les caractéristiques logiques d'une base de données
- Utilise 3 (trois) concepts fondamentaux pour modéliser les données : *Entité*, *Relation* et *Attributs*.

Concepts du modèle E/A : définitions

Entité : Collection d'objets de même type concrets ou abstraits ayant une existence propre dans le système. Une entité doit avoir plus d'une occurrence et elle doit être décrite par au moins un attribut, sinon il faudra se questionner sur son existence.

Exemple : Etudiant peut être une entité dans le système de la gestion des étudiants. En effet un étudiant est représenté par plusieurs attributs comme le nom et le prénom. Et nous avons plusieurs occurrences de Etudiant. Deux occurrences de Etudiant sont Lefebvre Martin, Simba Lion

Personnages est également une entité le jeu vidéo Warcraft

En général une entité est représentée par un rectangle.

| Etudiant |
|-----------------|
| NoEtudiant |
| Nom |
| Prénom |

| Cours |
|--------------|
| CodeCours |
| Titre |

Occurrence d'un attribut : l'occurrence d'un attribut représente la valeur de cet attribut. Une occurrence pour l'attribut Prénom peut être Yanick

Occurrence d'une entité : une occurrence d'une entité est une instance de cette entité.

Exemple

Soit l'entité Etudiant définie précédemment. Les attributs de l'entité sont :

NoEtudiant

Nom

Prénom

Une occurrence de l'entité peut être : 23462007 Courtney Yanick

Relation ou association :

Les entités elles-mêmes sont porteuses de peu d'information, mais si celles-ci sont associées à d'autres entités alors l'information devient plus pertinente et plus conséquente.

En effet l'entité Etudiant est porteuse de trois informations, NoEtudiants Nom et Prénom, nous sommes renseignés sur l'identité de l'étudiant. Cependant si l'entité Etudiant est associée à l'entité Cours, alors nous pouvons être renseignés sur les cours auxquels un étudiant est inscrit.

Une relation représente une association entre deux ou plusieurs entités traduisant un fait ou un événement dont le système voudra garder trace. Le nom choisi pour une relation est un verbe d'action permettant de décrire la relation.

Pour représenter une relation, il existe plusieurs façons selon la méthode que nous utilisons pour la conception et selon les outils disponibles pour la modélisation:

- Merise : les associations sont des ovales. Dans ce cas les relations peuvent être porteuses de propriétés.
- Case D'Oracle, les associations sont des traits suivis de terminaison.
- Entity-Framework , lien avec des terminaison en chiffres.

Il existe plusieurs types de relation

- Les relations binaires simples : ce sont des relations qui lient deux entités pas plus.
- Les relations n-aires, plus de deux entités participent à la relation (formalisme de Merise)
- Les relations réflexives, c'est une relation sur la même entité

Généralisation d'entité.

La généralisation d'entité consiste à regrouper sous la même entité plusieurs entités ayant les mêmes caractéristiques. (un peu comme le concept d'héritage en POO.).

On appelle entité *surtype*, l'entité qui englobe plusieurs entités.


On appelle entité *sous-type* une entité à l'intérieur de l'entité surtype.

Définition, cardinalité

Une cardinalité est le nombre de fois minimum et maximum qu'une occurrence d'une Entité participe à une relation.

Attention : 

Le nombre minimum est tout le temps égal à 0 ou 1.

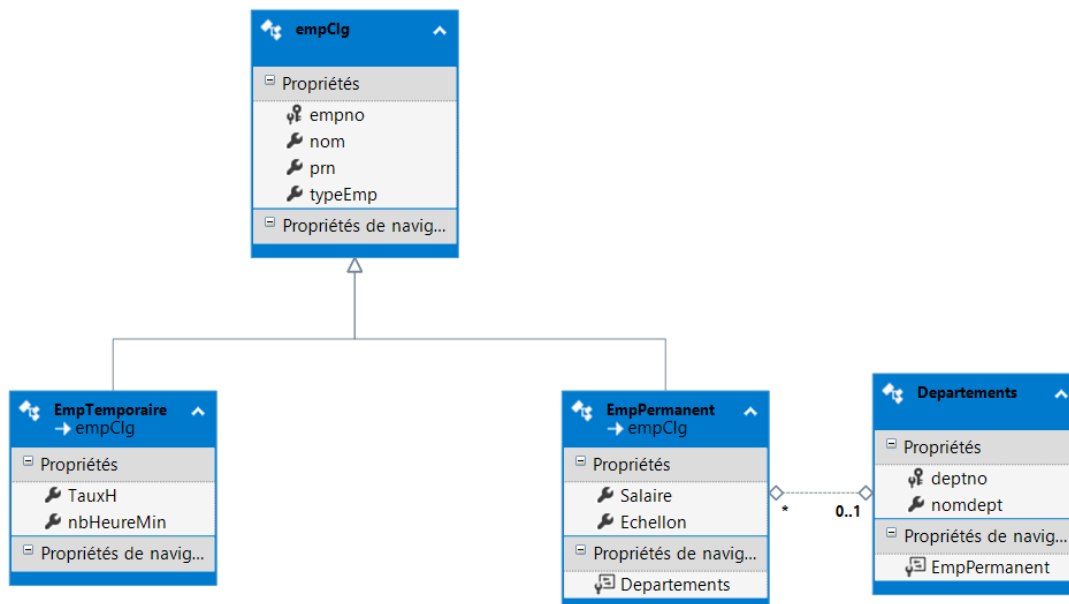
Attention : 

Le nombre maximum est tout le temps égal à 1 ou N. Selon les SGBDs, le chiffre N est représenté par * (infini) , une fourche pour Oracle..

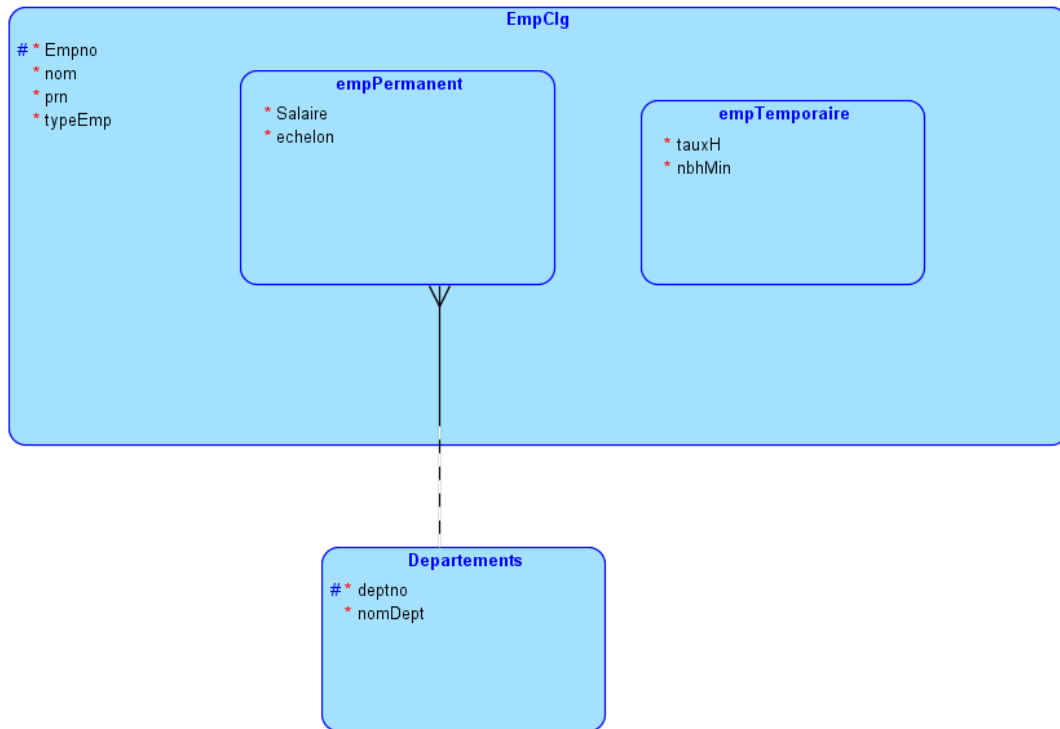
Avantages du modèle Conceptuel :

- Indépendant du SGBD
- Plus proche du monde réel
- Lorsque vous avez un système avec une base de données importante, il est difficile de voir les clés étrangères.
- Le modèle conceptuel offre une meilleure représentation du système. Exemple, nous pouvons représenter la notion de généralisation/spécialisation (Heritage) appelé aussi généralisation d'entités.

Exemple de représentation dans Entity Framework



Exemple Oracle :




Attention : 

Toutes les entités doivent avoir un identifiant. Un identifiant est un attribut qui identifie les occurrences d'une entité de manière unique.

Attention : 

Dans le modèle E/A, aucune redondance des données n'est autorisée, même celle de la clé primaire qui devient clé étrangère.

Normalisation du modèle Entités-Association

Attention : 

**Le modèle Entités/relation doit-être normalisé. Donc au moins en 3FN.
Les mêmes règles de normalisation du modèle Relationnel s'appliquent au modèle E/R**

Passage du modèle E/R au modèle Relationnel

Règle 1 : Entité excluant les entités surtype et sous-type :

- Toute entité du modèle Entité/Association est traduite par une table dans le modèle relationnel
- La clé primaire de cette table est l'identifiant de l'entité
- Les attributs de la table sont les attributs de l'entité.

Règle 2 : Relation binaire avec participation multiple :

CAS : 0:N — 0:N OU 0:N — 1:N OU 1:N — 1:N

- La relation devient t une table
- La nouvelle table hérite des clés primaires provenant des tables reliées pour devenir des clés étrangères. Ces clés étrangères forment également une clé primaire composée de la nouvelle table.
- Les clés étrangères auront des valeurs obligatoires (puisque'elles forment une clé primaire composée) et la participation est multiples. (la valeur de la clé étrangère peut revenir plusieurs fois).

Règle 3 : Relation binaire avec participation multiple d'un côté et unique de l'autre côté

Cas **0:N** — **0:1** **0:N** — **1:1**

OU **1:N** — **0:1** **1:N** — **1:1**

- La table avec participation unique hérite de la clé primaire provenant de la table opposée pour devenir une clé étrangère. Si la participation est obligatoire (1:1) alors la valeur de la clé étrangère est obligatoire. Si la relation est avec un identifiant relatif, alors la clé étrangère fait partie de la clé primaire qui devient alors une clé primaire composée.
- Le lien entre les tables pointera sur la table qui a hérité de la clé étrangère. Le lien sera obligatoire si la table qui a hérité de la clé étrangère avait une participation obligatoire à la relation et optionnel dans le cas contraire.

Règle 4 : Relation binaire avec participation d'un côté et unique des deux côtés

Cas **0:1** — **0:1** **0:N** — **1:1**

OU **1:** — **0:1** **1:** — **1:1**

Cas (0:1---1:1)

- La table avec la participation obligatoire hérite de la clé primaire provenant de la table opposée pour devenir une clé étrangère avec la participation obligatoire. Si la relation est avec un identifiant relatif, alors la clé étrangère fait partie de la clé primaire qui devient alors une clé primaire composée.
- Le lien entre les tables pointera sur la table qui a hérité de la clé étrangère. Le lien sera obligatoire avec la participation unique.

Cas (0:1—0:1)

- Dans ce cas n'importe qu'elle table peut hériter de la clé primaire de l'autre table pour devenir une clé étrangère avec valeur optionnelle. Le choix de la table qui va hériter de la clé dépendra du système.
- Le lien entre les tables pointera sur la table qui a hérité de la clé étrangère. Le lien sera optionnel avec la participation unique

Cas (1:1----1:1)

- On doit d'abord vérifier si l'on ne peut pas inclure les deux tables dans une seule, ce qui est en général le cas. Dans ce cas-ci, la clé primaire de la nouvelle table sera l'une ou l'autre des clés primaires des deux tables initiales, une clé composée des deux clés primaires, ou alors une nouvelle clé primaire
- Si l'on ne peut pas scinder les deux tables en une seule, alors les mêmes règles citées pour le cas optionnel s'appliquent, avec valeur obligatoire.

Règle 5: Relation récursive avec participation multiple.

- La relation devient une table de lien avec une clé primaire composé de deux fois l'identifiant de l'entité. Les attributs constituant la clé primaire sont également qualifiés de clé étrangères..
- Les liens entre les deux tables pointeront sur la nouvelle table. Les liens sont obligatoires car les clés étrangères sont devenues une clé primaire composée. La participation est multiple

Règle 6: Relation récursive avec participation unique d'un côté

- La clé primaire est dupliquée et devient une clé étrangère avec appellation différente de la clé primaire.
- Le lien reste récursif et sera défini en fonction de la participation de l'entité à la relation.

Règle7: Relation entité surtype et sous-type.(Héritage)

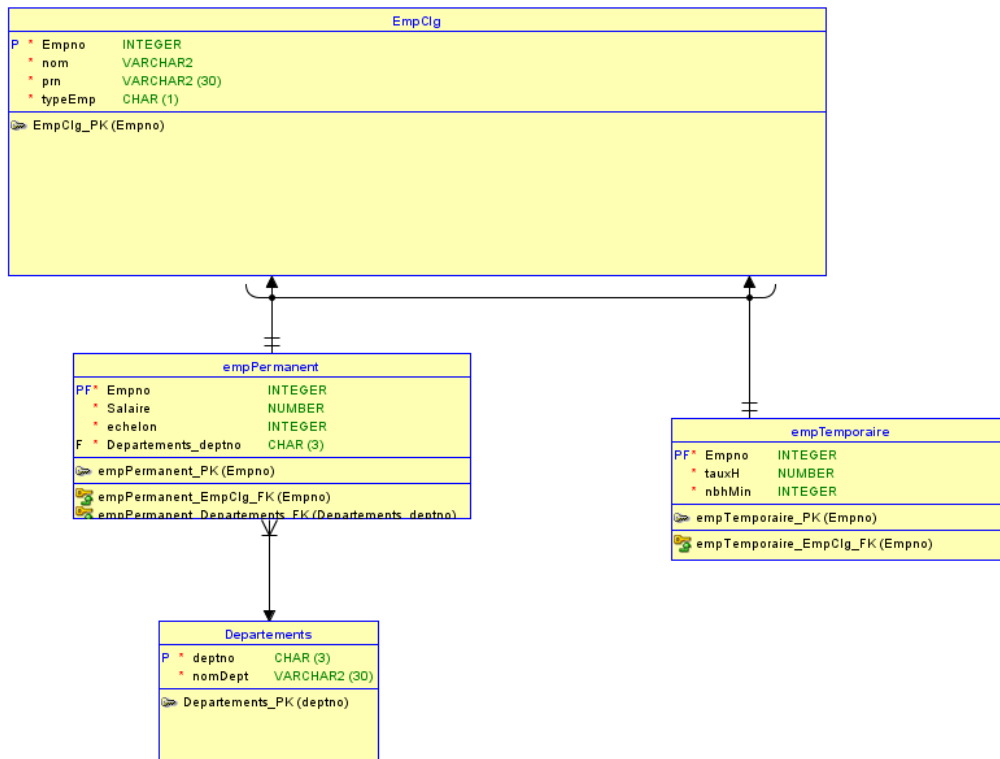
Il existe plusieurs solutions qui permettent de passer une telle représentation, la solution que l'on recommande est celle qui procède comme suit :

- Chaque entité devient une table. (les surtype et les sous-type).
- L'identifiant de l'entité surtype devient la clé primaire de la table issue de cette entité.
- Les entité sous-type hérite de l'attribut de la clé primaire de la table issue du surtype et devient :
 - Clé primaire dans chacune des tables issues des entités sous-types
 - Clé étrangère des mêmes tables.
- Dans la table issue de l'entité Surtype, ce n'est pas déjà fait dans le modèle E/R, il faudra ajouter un attribut indiquant le type d'entité.
- Certain SGBD, vous demanderont d'implanter des mécanismes garantissant l'intégrité des données au moment des insertions. (Triggers)

Il existe une solution simple mais pas optimale puisqu'elle implique de beaucoup de valeurs nulles. Personnellement, je NE la recommande PAS du tout. Elle consiste :

- Regrouper toutes les propriétés (attributs) dans une seule table.
- Un nouvel attribut est créer afin d'identifier le type d'occurrence
- L'identifiant unique du surtype devient la clé primaire de la table. Les attributs des entités sous-type deviennent les attributs de la table avec valeurs optionnelle.

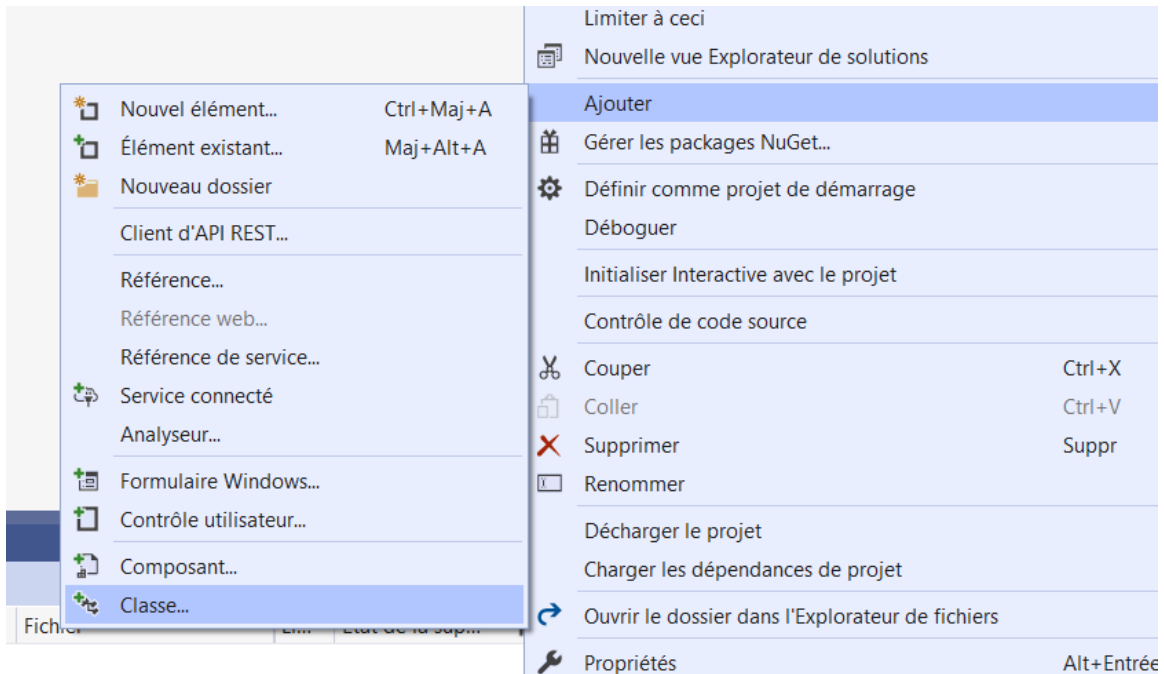
Exemple Formalisme Oracle, passage du modèle E/A au modèle Relationnel.



Utilisation de Visual Studio 2019 pour réaliser un modèle E/A

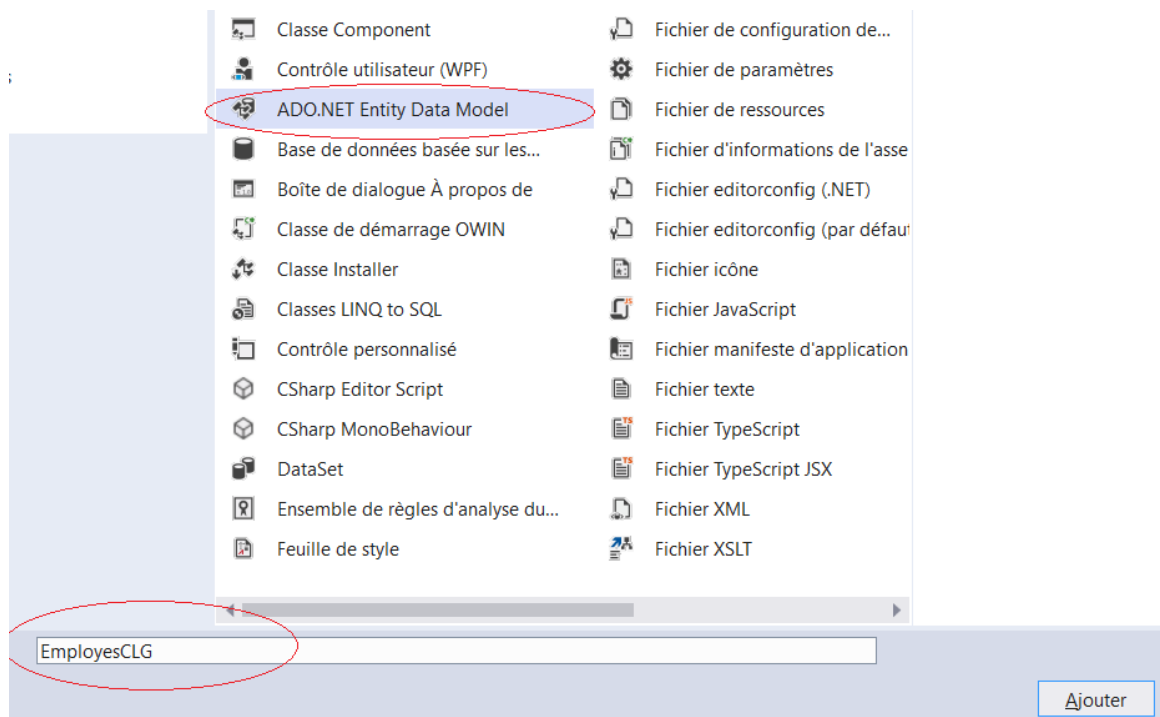
Étape 0 : Créer un projet Visual Studio Windows Forms.

Étape 2 : Cliquer sur le bouton droit de votre solution, puis ajouter une classe

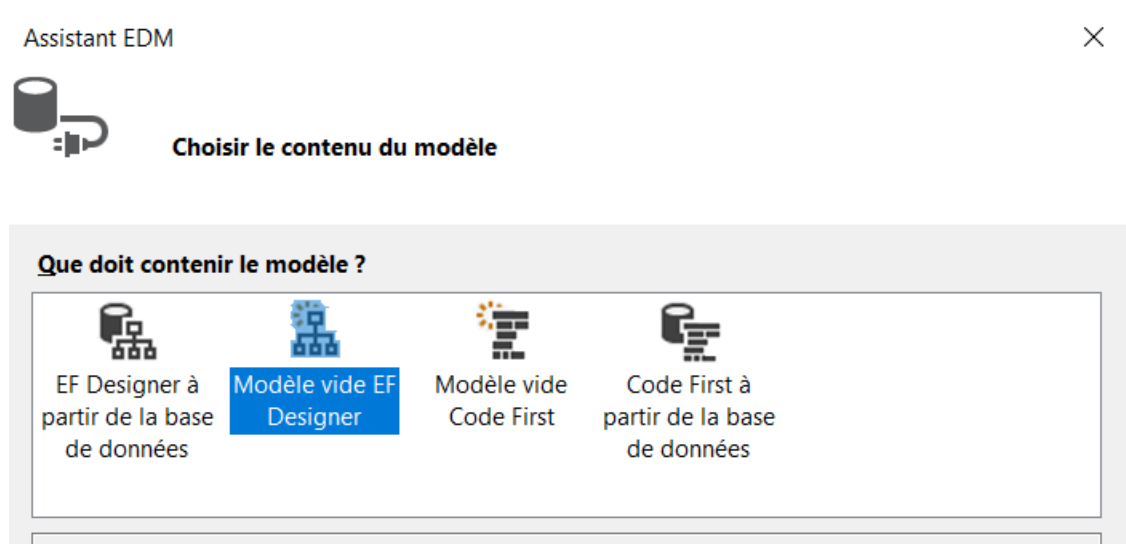


Étape 3 :

1. Choisir dans la liste ADO.NET Entity Data Model
2. Donner un nom à votre modèle.
3. Puis Ajouter



4. Choisir Modèle Vide EF Designer, puis terminer :



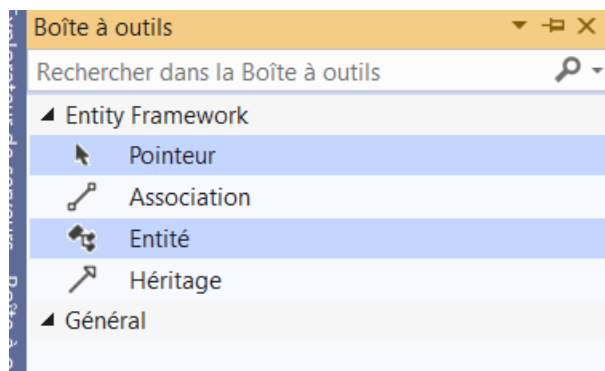
5. Choisir de créer votre modèle en y ajoutant des éléments de la boîte à Outils

Entity Data Model Designer vous permet de visualiser et concevoir votre Entity Data Model.

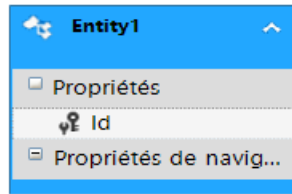
Créez de nouvelles entités dans le modèle en faisant glisser les éléments à partir de [Boîte à outils](#).

Ajoutez des entités et relations existantes à ce diagramme en les faisant glisser à partir de l'[explorateur de modèles](#).

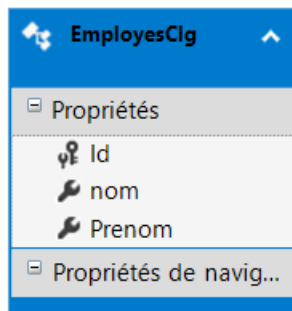
6. Faites glisser les éléments de la boîte à outils sur votre fenêtre principale pour commencer à créer votre modèle :



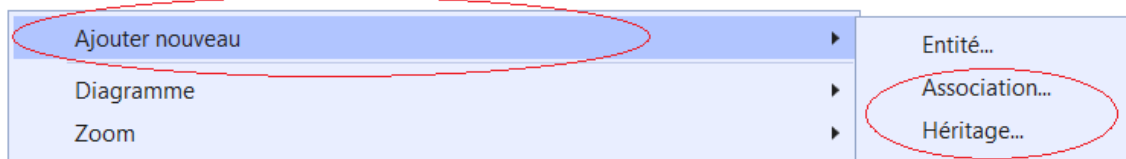
Évidemment, il faudra commencer par faire glisser vos entités. Ça prends au moins deux Entités pour réaliser une relation.



7. En utilisant le bouton droit de la souris sur votre entité, vous pouvez, la renommer, ajouter des attributs (**propriétés scalaires**)



8. Répéter ces opérations pour toutes vos entités.
9. Ajouter les associations :



10. Pour l'association d'héritage, VS2019 vas vous demander de choisir une Entité de base (Sur Type, ou la super Entité) et une Entité dérivée (Entité sous Type, ou sous Entité)

Ajouter un héritage

Sélectionnez des entités de base et dérivée pour créer une nouvelle relation d'héritage.

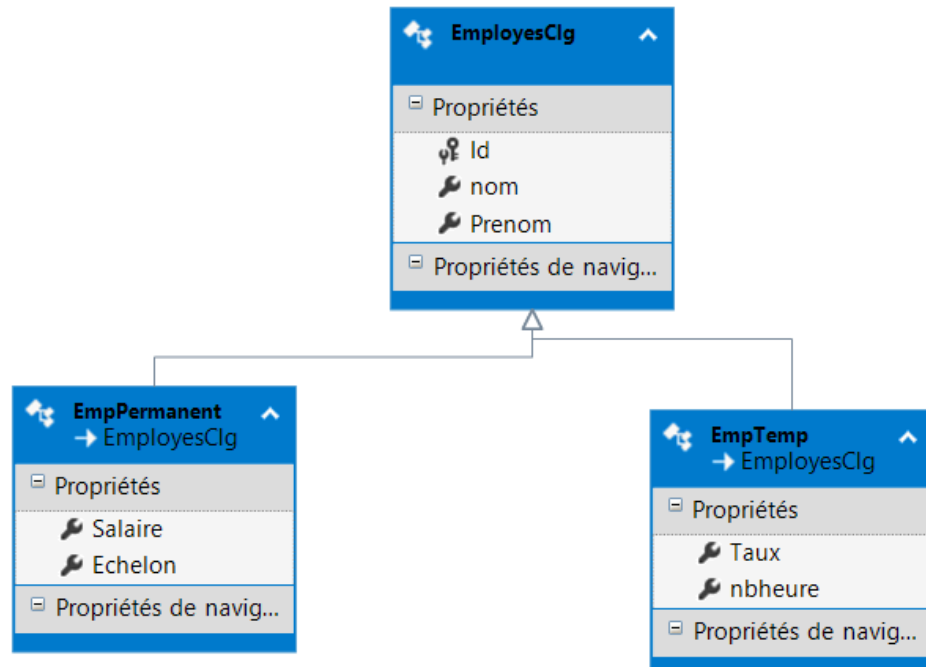
Sélectionnez une entité de base :

EmployesClg

Sélectionnez une entité dérivée :

EmpPermanent

OK Annuler



11. Pour les autres associations, binaires, procédez de la même façon en prenant soin de bien choisir :

- a. Le nom de la relation
- b. Les noms des tables
- c. Les cardinalités

Ajouter une association

Nom de l'association :
DepartementsEmpPermanent

Terminaison
Entité :
Departements
Multiplicité :
1 (un)
 Propriété de navigation :
EmpPermanent

Terminaison
Entité :
EmpPermanent
Multiplicité :
* (plusieurs)
 Propriété de navigation :
Departements

Ajouter les propriétés de la clé étrangère à l'entité 'EmpPermanent'

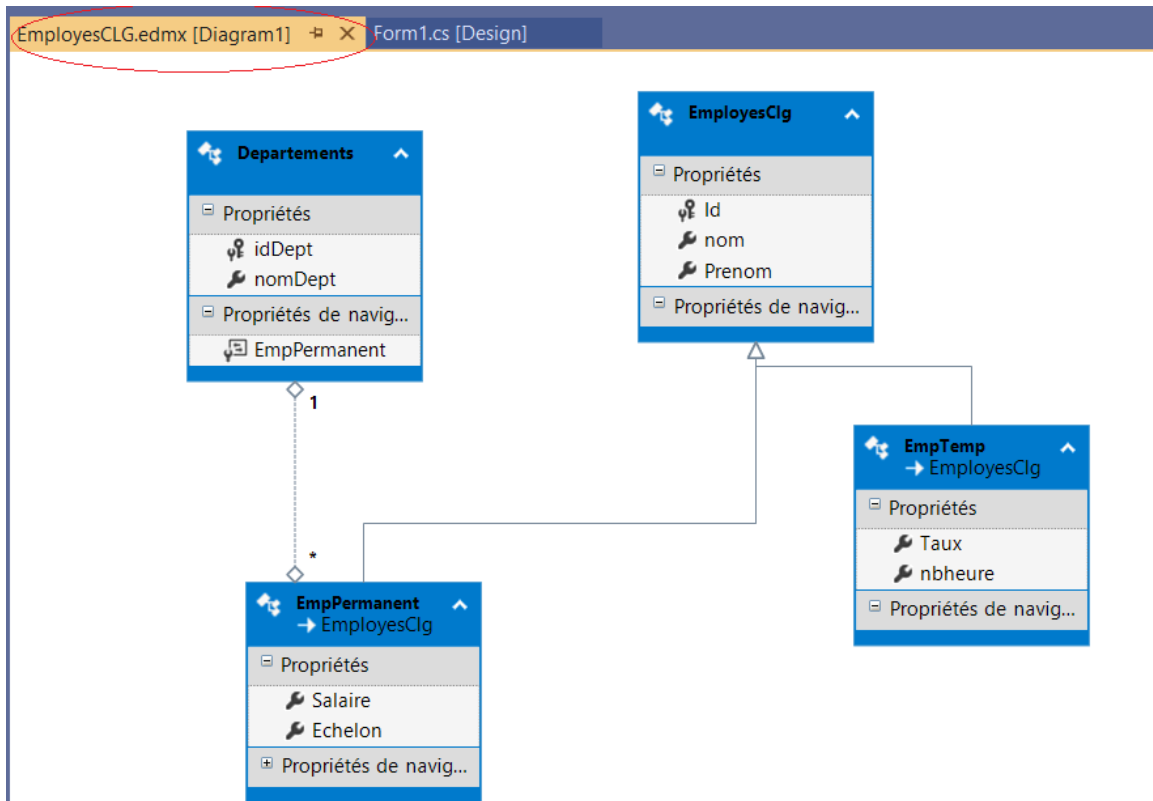
Departements peut avoir * (plusieurs) instances de EmpPermanent. Utilisez Departements.EmpPermanent pour accéder aux instances EmpPermanent.

EmpPermanent peut avoir 1 (un) instance de Departements. Utilisez EmpPermanent.Departements pour accéder à l'instance Departements.

OK Annuler

- d. Décocher la case Ajouter les propriétés de clé étrangère. Comme le montre la figure.

Une fois terminé, vous aurez votre diagramme E/A. Votre diagramme a l'Extension edmx.



12. Remarquez aussi que le nom de votre diagramme se trouve dans l'explorateur de solution.

Attention : 

Ne jamais, jamais générer le code SQL.