

## LA MÉTHODE MERISE

Ce tutoriel a pour but de savoir résoudre un problème à l'aide de MERISE et de développer la base de donnée correspondante..

### L'Analyse des données :

Prenons l'exemple d'une entreprise de maintenance de machines : « Cette entreprise a des intervenants qui sont repérés par leur identifiant, leur nom et leur numéro de téléphone. Ils interviennent dans des usines qui possèdent des machines ayant un numéro, un nom et une marque. Les usines possèdent un numéro, une adresse et un nom. Le prix de l'intervention dépend du lieu et de l'heure. Les intervenant sont capables de réparer n'importe quelle machine. »

Nous devons, en fonction de cette description, repérer les différents **attributs** (données) et leur type :

Un intervenant : id\_i (un séquence alphanumérique de 5 caractères), nom\_i (Une séquence alphabétique de 30 caractères maximum), tel\_i (un séquence de 10 chiffres).

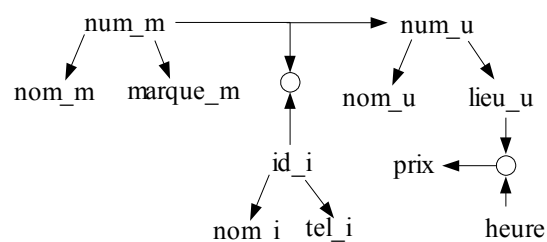
Une machine : num\_m (nombre entre 1 et 9999), nom\_m (10 caractères maximum), marque\_m (20 caractère maximum)

Une usine : num\_u (nombre entre 1 et 999), lieu\_u (maximum 40 caractères), nom\_u (10 maximum 20 caractères).

Notez l'effort fait pour rendre les noms significatifs, même si rien n'est imposé.

### Graphique des Dépendances Fonctionnelles (DF) :

Il s'agit simplement de relier les attributs entre eux. Dans la plus part des cas on relie les attributs qui représentent un détails de la même chose (détails d'une machine, d'un intervenant), à un identifiant unique (dans ce cas les num\_m, id\_i...). Ils sont unique et toujours existant !

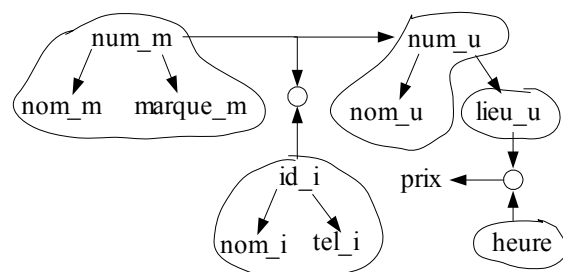


Lorsque plusieurs chose peuvent être en relation avec plusieurs autres, j'utilise une **association** comme ici entre num\_m et id\_i ou lieu\_u et heure car un intervenant peut intervenir sur plusieurs machines, et une machine peut être réparée par n'importe quel intervenant. Une intervention dans un lieu\_u peut être faite à plusieurs moment et plusieurs interventions dans différents lieux peuvent être opérées en

même temps.

Ici le prix dépend à la fois du lieu\_u et de l'heure, il est donc issue d'une association.

### Passage du DF au Schéma Entités-Association :



Nous commençons par entourer ce qui va être plus tard les entités.

Une **entité** est un objet (une table de notre base), par exemple une « machine », ou une « usine ». On peut utiliser plusieurs fois le même objet (chaque machine représente une ligne de la table MACHINE).

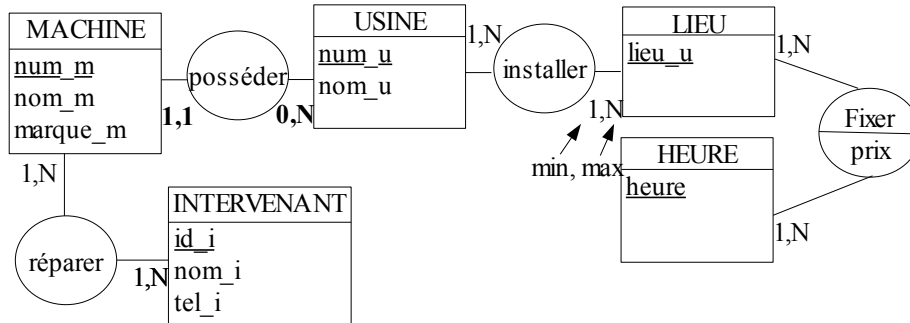
Il faut donc leur donner une clé unique : ici ce sont nos numéros de machine ou nos identifiants d'intervenant (id\_i). Les clés sont toujours soulignées.

Ici, il est tout à fait possible qu'il y ait plusieurs usines sur le même lieu. De plus on nous dit que le prix dépend du lieu, on a donc tout intérêt à sortir le lieu\_u de l'usine et à en faire une autre entité. Nous l'entourons donc à part.

Ensuite nous allons transformer les liens et les attributs (ou groupes d'attributs) en entités : en les regroupant avec un nom. Il faut aussi veiller à souligner les identifiants (clés).

Nous ajoutons aussi les **cardinalités**, le nombre maximum et minimum de relation. Par exemple, ici, une machine n'appartenant (posséder) qu'à une seule usine, mais au minimum à une (**1,1**). Dans l'autre sens, une usine peut avoir un infinité machines ou bien 0 (**0,N**), le zéro est toujours discutable, mais partons d'un principe que l'on ajoute une usine avant de connaître ses machines.

Nous obtenons donc le schéma entité/associations suivant.



Les types de relations, et leur équivalent pour le schéma entité-association :

<p><b>Relation</b> : c'est le lien le plus courant on l'utilise lorsqu'une entité (1 au maximum) peut donner lieu à plusieurs (N au maximum) : du côté de la flèche.</p> <p>num_machine → num_usine</p> <p>MACHINE ?1- ?N USINE</p> <p>Ici ? représente le minimum, sa valeur dépend du sujet.</p>	<p><b>Association simple</b> : cette association est utilisée quand 1 entité peut donner lieu à N entités, et <u>réciroquement</u>.</p> <p>id_intervenant → num_mach</p> <p>MACHINE ?N- ?N INTERVENANT</p> <p>1 intervenant peut correspondre à N machine, et 1 machine peut correspondre à N intervenant.</p>	<p><b>Association avec entité résultante</b> :</p> <p>Il s'agit de la même relation que l'association simple. On l'utilise quand un attribut est défini par deux entités à la fois.</p> <p>Ici le prix dépend du lieu et de l'heure. Pour le schéma entité association on place la variable dans le cercle.</p>

Pour ces trois entités nous plaçons un cercle contenant un verbe, qui définit la relation. Par exemple entre MACHINE et USINE, nous utilisons le verbe *posséder*. Dans le cas d'une association avec entité résultante nous plaçons aussi les variables dans le cercle.

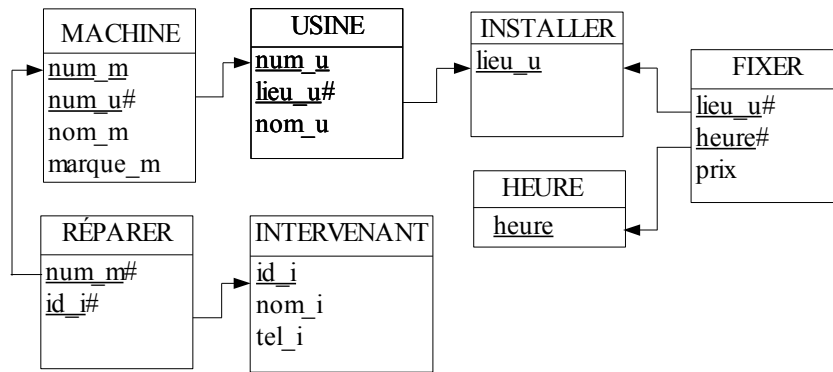
On utilise une flèche  $\longleftrightarrow$  pour les relations ?1 — ?1

**Passage au schéma relationnel** : La forme des bases de données :

Dans un schéma relationnel **les associations sont transformées en entités** : L'association *réparer* devient l'entité RÉPARER. Elle est repérée par un numéro de machine et un identifiant d'intervenant. Nous pouvons noter que ces deux clés proviennent des entités MACHINE et INTERVENANT qui servent à définir RÉPARER. On les appelle donc **clés étrangères** et on les repère par un # à la fin. **La flèche part toujours de la clé étrangère vers sa clé d'origine.**

De même, notre prix de l'association *fixer* devient l'entité FIXER qui a deux clés : lieu\_u# et heure# et l'association installer devient l'entité INSTALLER dont la clé lieu\_u sert de clé étrangères à USINE.

Ce qui donne :



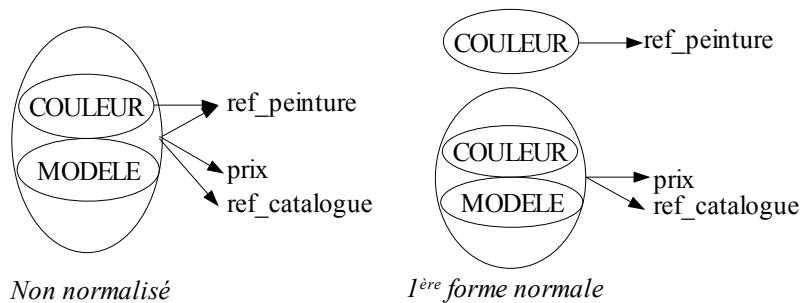
**La normalisation :**

Avec le schéma relationnel nous avons obtenu une forme facilement transformable en base de donnée. Pour des soucis d'optimisation, il est possible d'améliorer encore notre base avant de l'informatiser.

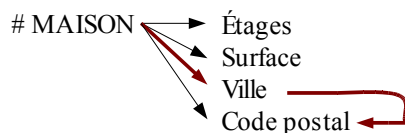
**1<sup>ère</sup> forme normale** : chaque ligne d'une relation (table) doit avoir un valeur pour chaque colonne (attribut). Les structures de données ne doivent pas être répétées plus d'une fois et les attributs doivent être décomposés (préférer : rue, code\_postal, ville au lieu de adresse)

On est en **2<sup>nd</sup> forme normale** (2NF) quand lorsque la clé dépend de plusieurs attributs (on dit parfois que l'on a plusieurs clé), tous les attributs doivent dépendre de toute la clé.

Exemple :

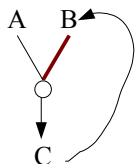


On est en **3<sup>ème</sup> forme normale** (3NF) lorsqu'il n'y a pas de transitivité :



La ville nous permet de connaître le Code Postal, la maison aussi. On peut supprimer un des deux liens rouge.

La forme suivante s'appelle BCNF (Boyce Codd). C'est lorsque la clé n'est pas transitive :



C dépend de A et de B, et B dépend de C.

$A \rightarrow (C,B)$  et  $C \rightarrow B$

On peut donc supprimer le lien rouge.

Donc C dépend de A, et B dépend de C

$A \rightarrow C \rightarrow B$