

**SOMMAIRE**

<b>1 - Présentation</b>	<b>2</b>
<b>1 1 - Une base de données, c'est quoi ?</b>	<b>2</b>
1 1 1 Notion de base de données	2
1 1 2 Modèle de base de données	3
<b>1 2 - Un SGBD, c'est quoi ?</b>	<b>5</b>
1 2 1 Principes de fonctionnement	5
1 2 2 Les objectifs	5
1 2 3 Exemples de SGBD	7
<b>2 - Conception d'une base de données, modèle entités-associations</b>	<b>8</b>
<b>2 1 - Introduction</b>	<b>8</b>
2 1 1 Modélisation préalable	8
2 1 2 Merise	8
<b>2 2 - Les éléments du modèle entités-associations</b>	<b>10</b>
2 2 1 L'entité	10
2 2 2 Attribut ou propriété, valeur	11
2 2 3 Identifiant ou clé	12
2 2 4 Association	13
2 2 5 Cardinalité	14
<b>3 - Introduction aux bases de données relationnelles</b>	<b>15</b>
<b>3 1 - Objectifs et définitions</b>	<b>15</b>
<b>3 2 - Eléments principaux du modèle relationnel</b>	<b>17</b>
<b>3 3 - Passage du modèle entités-associations au modèle relationnel</b>	<b>19</b>
<b>4 - Exemple de mise en oeuvre</b>	<b>20</b>
<b>4 1 - Les informations recueillies</b>	<b>20</b>
<b>4 2 - Modéliser le MCD</b>	<b>21</b>
<b>4 3 - Générer le MPD</b>	<b>23</b>
<b>4 4 - Générer le script de création de la base de données</b>	<b>24</b>

# 1 - Présentation

## 1 1 - Une base de données, c'est quoi ?

### 1 1 1 Notion de base de données

Une base de données est un ensemble structuré et organisé de données enregistrées pouvant être interrogées (recherche) et mises à jour (ajout, mise à jour, suppression) par une communauté d'utilisateurs.

L'accès à une base de données et sa gestion, sont assurés par un ensemble de programmes qui constituent le Système de Gestion de Base de Données (SGBD).

Les bases de données doivent pouvoir gérer de grandes quantités de données, géographiquement distribuées à l'échelle d'Internet, utilisées par un grand nombre d'utilisateurs dans un contexte d'exploitation changeant.

La technologie des bases de données s'oriente vers des architectures de base de données distribuées.

## 1 1 2 Modèle de base de données

### Modèle hiérarchique

Dans une base de données de type hiérarchique les enregistrements sont liés dans une structure arborescente de façon à ce que chaque enregistrement n'ait qu'un seul possesseur ou "père".

Ce modèle a été mis au point à l'occasion du programme Apollo (années 60).

### Modèle réseau

Le modèle réseau (ou navigationnel) permet d'établir des liaisons de type n-n.

Pour retrouver une donnée dans une telle modélisation, il faut connaître le chemin d'accès (les liens) ce qui rend les programmes dépendants de la structure de données.

Ce modèle de bases de données a été inventé par Bachman (1973).

### Modèle relationnel

Une base de données relationnelle est une base de données structurée suivant les principes de l'algèbre relationnelle.

Codd, chercheur chez IBM (fin des années 1960), mathématicien de formation, publia en 1970 un article où il proposait de stocker des données hétérogènes dans des tables, permettant d'établir des relations entre elles.

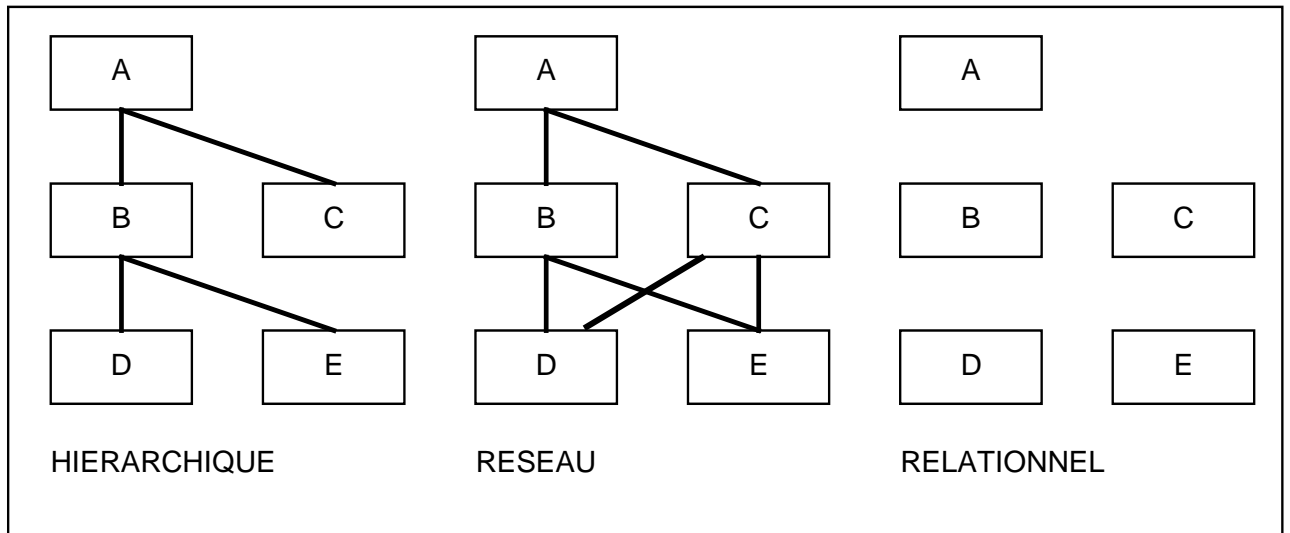
Le premier prototype de SGBD Relationnelles (SGBDR) a été construit dans les laboratoires d'IBM.

En 1987, le langage SQL, qui étend l'algèbre relationnelle, a été standardisé.

### Modèle objet

Les modèles de bases de données "objet" et "objet relationnel" sont plus récents.

L'histoire ... (en 3 schémas) des SGBD ...



— : pointeurs physiques

## 1 2 - Un SGBD, c'est quoi ?

### 1 2 1 Principes de fonctionnement

La plupart des SGBD fonctionnent selon un mode client/serveur.

Le serveur, la machine qui stocke les données, reçoit des requêtes de plusieurs clients et ceci de manière concurrente.

Le serveur analyse les requêtes, les traite et retourne les résultats au client. Le modèle client/serveur est souvent implémenté dans un réseau qui peut être Internet.

Un des points importants à prendre en compte est la cohérence des données.

Si plusieurs utilisateurs peuvent accéder en même temps à une ligne d'une table pour la lire ou pour la mettre à jour, il faut s'accorder sur la politique de mise à jour.

### 1 2 2 Les objectifs

Indépendance physique : la définition des données doit être indépendante des structures de stockage.

Indépendance logique : les données peuvent être vues différemment par des utilisateurs différents.

Accès aux données : l'accès aux données se fait par l'intermédiaire d'un Langage de Manipulation de Données (LMD) qui doit permettre d'obtenir des réponses aux requêtes en un temps "raisonnable". Il doit donc être optimisé et minimiser le nombre d'accès disques de façon transparente pour l'utilisateur.

Intégration : les données doivent être centralisées dans un réservoir unique commun aux applications.

Non redondance des données : afin d'éviter les problèmes lors des mises à jour, chaque donnée ne doit être présente qu'une seule fois dans la base.

## LES BASES DE DONNEES

Cohérence des données : les données sont soumises à un certain nombre de contraintes d'intégrité qui définissent un état cohérent de la base.

Ces contraintes d'intégrité sont vérifiées automatiquement à chaque insertion, modification ou suppression des données.

Partage des données : plusieurs utilisateurs doivent accéder aux mêmes données au même moment de manière transparente.

Ce problème est simple à résoudre quand il s'agit uniquement d'interrogations, quand il s'agit de modifications dans un contexte multi-utilisateurs il faut permettre à plusieurs utilisateurs de modifier la même donnée "en même temps" et assurer un résultat d'interrogation cohérent pour un utilisateur consultant une table pendant qu'un autre la modifie.

Sécurité des données : les données doivent être protégées contre les accès non autorisés.

Résistance aux pannes : Il faut pouvoir récupérer une base dans un état "sain" ou "cohérent". Ainsi, après une panne intervenant au milieu d'une modification deux solutions sont possibles : soit récupérer les données dans l'état où elles étaient avant le début de la modification, soit "terminer" de manière forcée l'opération interrompue.

## LES BASES DE DONNEES

### 1 2 3 Exemples de SGBD (liste non exhaustive)

PostgreSQL	<a href="http://www.postgresql.org/">http://www.postgresql.org/</a>	dans le domaine public
MySQL	<a href="http://www.mysql.org/">http://www.mysql.org/</a>	dans le domaine public
Oracle	<a href="http://www.oracle.com/">http://www.oracle.com/</a>	
IBM DB2	<a href="http://www-01.ibm.com/software/data/db2/">http://www-01.ibm.com/software/data/db2/</a>	
Microsoft SQL	<a href="http://www.microsoft.com/sql/">http://www.microsoft.com/sql/</a>	
Sybase	<a href="http://www.sybase.com/linux">http://www.sybase.com/linux</a>	
Informix	<a href="http://www-01.ibm.com/software/data/informix/">http://www-01.ibm.com/software/data/informix/</a>	
...	...	...

## 2 - Conception d'une base de données, modèle entités-associations

### 2 1 - Introduction

#### 2 1 1 Modélisation préalable

Ce modèle, (Chen), permet une description du monde réel à partir des concepts d'entité et d'association.

Ce modèle, utilisé pour la phase de conception, s'inscrit notamment dans le cadre d'une méthode plus générale et très répandue, la méthode Merise.

#### 2 1 2 Merise

La méthode Merise (Méthode d'Étude et de Réalisation Informatique pour les Systèmes d'Entreprise - 1977) permet de valider des choix par rapport aux objectifs, de quantifier les solutions retenues, de mettre en œuvre des techniques d'optimisation et enfin de guider jusqu'à l'implémentation.

Merise devient un outil de communication qui offre un moyen de communication accessible aux non-informaticiens.

Un des concepts de la méthode Merise est la séparation des données et des traitements.

L'expression conceptuelle des données conduit à une modélisation des données en "entités" et en "associations".

Remarque : la méthode Merise est évoquée ici uniquement dans la perspective de la modélisation des bases de données sans tenir compte de la modélisation des traitements.



Merise propose une démarche par niveaux, dans laquelle il s'agit de hiérarchiser les préoccupations de modélisation qui sont de trois ordres : la conception, l'organisation et la technique.

Niveau conceptuel : le modèle conceptuel des données (MCD) décrit les entités du monde réel, en terme d'objets, de propriétés et de relations, indépendamment de toute technique d'organisation et d'implantation des données.

Ce modèle se concrétise par un schéma entités-associations du point de vue des données.

Niveau logique : le modèle logique des données (MLD) précise le modèle conceptuel par des choix organisationnels.

Il s'agit d'une transcription du MCD dans un formalisme adapté à une implémentation ultérieure, au niveau physique, sous forme de base de données relationnelles, réseaux ou autres.

Niveau physique : le modèle physique des données (MPD) permet d'établir la manière concrète dont le système sera mis en place avec le SGBD retenu.

## 2 2 - Les éléments du modèle entités-associations

La représentation du modèle entités-associations s'appuie sur trois notions :

- l'entité,
- l'association,
- la propriété.

L'entité est un objet ayant une existence propre.

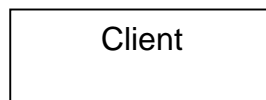
L'association est un lien ou relation entre objets sans existence propre.

La propriété est la plus petite donnée d'information décrivant un objet ou une association.

### 2 2 1 L'entité

Entité : une entité est un objet, une chose concrète ou abstraite qui peut être reconnue distinctement et qui est caractérisée par son unicité.

Représentation graphique de l'entité Client sans ses propriétés.



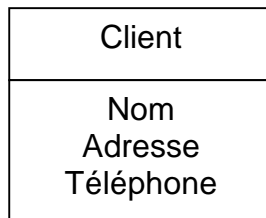
## 2 2 2 Attribut ou propriété, valeur

attribut, propriété : un attribut ou une propriété est une caractéristique associée à une entité ou à une association

exemple : le nom d'un client, le prix d'un produit, etc ...

valeur : au niveau de l'entité ou de l'association, chaque attribut possède un domaine qui définit l'ensemble des valeurs possibles (entier, chaîne de caractères, booléen, ...).

Représentation graphique de l'entité Client avec trois attributs.



Un attribut est une donnée élémentaire.

Une entité et ses attributs doivent être cohérents entre eux, c'est à dire ne traiter que d'un seul sujet.

Par exemple, si le modèle doit comporter des informations relatives à des produits et à leur fournisseur, ces informations ne doivent pas coexister au sein d'une même entité.

Il est préférable de mettre les informations relatives aux produits dans une entité "Produit" et les informations relatives aux fournisseurs dans une entité "Fournisseur".

Ces deux entités seront reliées par une association.

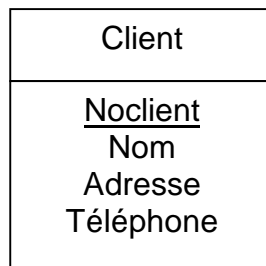
### □ 2 2 3 Identifiant ou clé

identifiant, clé : un identifiant ou clé d'une entité ou d'une association est constitué par un ou plusieurs de ses attributs qui doivent avoir une valeur unique pour chaque entité ou association.

Exemples d'identifiant : le numéro de sécurité sociale pour une personne, le numéro d'immatriculation pour une voiture, etc ...

Chaque entité doit posséder au moins un identifiant, éventuellement formé de plusieurs attributs.

Représentation graphique de l'entité Client avec l'identifiant et trois attributs.



## 2 2 4 Association

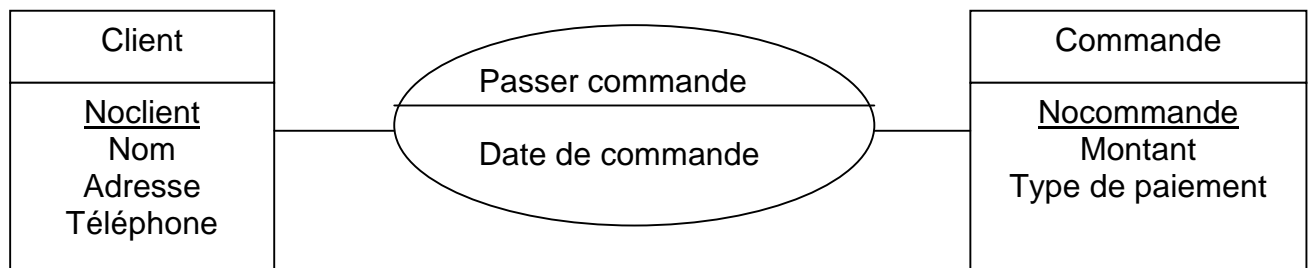
association : une association est un lien entre plusieurs entités.

Exemple d'association : l'emprunt, par l'étudiant Jean Dupont, du 25 ème exemplaire du livre "Les bases de données".

Comme les entités, les associations sont définis à l'aide d'attributs.

Un attribut peut être placé dans une association uniquement lorsqu'il dépend de toutes les entités liées par l'association.

Représentation graphique d'une association.



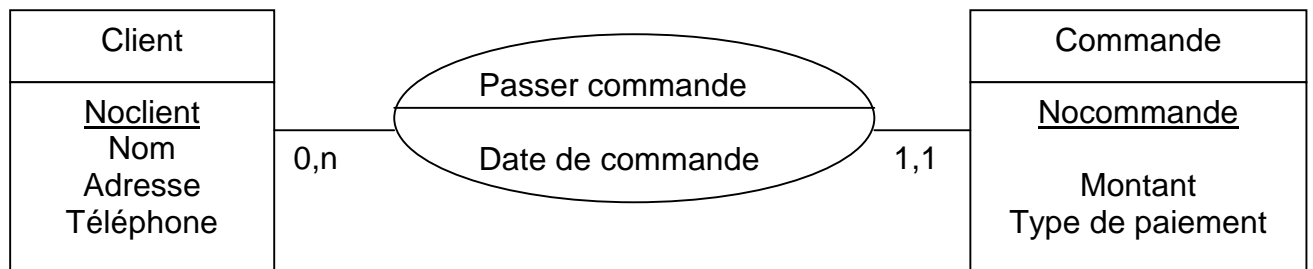
Comme une entité, une association possède un identifiant, qu'il soit explicite ou non.

La concaténation des identifiants des entités liées à une association constitue un identifiant de cette association.

## 2 2 5 Cardinalité

cardinalité : la cardinalité d'une patte reliant une association et une entité précise le nombre de fois minimal et maximal d'interventions d'une entité dans une association.

Représentation graphique des cardinalités d'une association.



L'expression de la cardinalité est obligatoire pour chaque patte d'une association.

Une cardinalité minimal est toujours 0 ou 1 et une cardinalité maximale est toujours 1 ou n.

Une cardinalité minimale de 1 doit se justifier par le fait que les entités ont besoin de l'association pour exister.

Dans tous les autres cas, la cardinalité minimale vaut 0.

## **3 - Introduction aux bases de données relationnelles**

### **3 1 - Objectifs et définitions**

Dans le modèle relationnel, les données sont représentées par des tables, sans préjuger de la façon dont les informations sont stockées dans la machine.

Les tables constituent donc la structure logique du modèle relationnel.

Au niveau physique, le système est libre d'utiliser n'importe quelle technique de stockage dès lors qu'il est possible de relier ces structures à des tables au niveau logique.

Les tables ne représentent donc qu'une abstraction de l'enregistrement physique des données.

Le modèle relationnel repose sur des bases théoriques solides, notamment la théorie des ensembles et la logique des prédicats (Codd).

Les objectifs du modèle relationnel sont :

- proposer des schémas de données faciles à utiliser,
- améliorer l'indépendance logique et physique,
- mettre à la disposition des utilisateurs des langages de haut niveau,
- optimiser les accès à la base de données,
- améliorer l'intégrité et la confidentialité,
- fournir une approche méthodologique dans la construction des schémas.

Avec le modèle relationnel :

- les données sont organisées sous forme de tables à deux dimensions, encore appelées relations, dont les lignes sont appelées n-uplet ou tuple,
- les données sont manipulées par des opérateurs de l'algèbre relationnelle,
- l'état cohérent de la base est défini par un ensemble de contraintes d'intégrité.

Au modèle relationnel est associée la théorie de la normalisation des relations qui permet de se débarrasser des incohérences au moment de la conception d'une base de données relationnelle.



## 3 2 - Eléments principaux du modèle relationnel

attribut : un attribut est un nom décrivant une information stockée dans une base.

Exemple : l'âge d'une personne, le nom d'une personne, le numéro de sécurité sociale.

domaine : le domaine d'un attribut est l'ensemble, fini ou non, de ses valeurs possibles.

Exemple : l'attribut "numéro de sécurité sociale" a pour domaine l'ensemble des combinaisons (suivant des règles précises) de quinze chiffres et "nom" a pour domaine l'ensemble des combinaisons de lettres ou chaîne de caractères.

relation : une relation est représentée sous la forme d'un tableau à deux dimensions dans lequel les n attributs correspondent aux titres des n colonnes.

schéma de relation : un schéma de relation précise le nom de la relation ainsi que la liste des attributs avec leurs domaines.

Exemple de relation et son schéma :

Client
<u>Noclient</u> Nom Adresse Téléphone

Schéma de la relation Client

```
(
    Noclient    numérique ,
    Nom         chaîne   ,
    Adresse     chaîne   ,
    Téléphone   chaîne
)
```

occurrence : c'est une ligne du tableau qui représente la relation.

cardinalité : la cardinalité d'une relation est son nombre d'occurrences.

clé candidate : une clé candidate d'une relation est un ensemble minimal d'attributs de la relation dont les valeurs identifient à coup sûr une occurrence.

La valeur d'une clé candidate est donc distincte pour toutes les occurrences de la relation.

Toute relation a au moins une clé candidate et peut en avoir plusieurs.

clé primaire : la clé primaire d'une relation est une de ses clés candidates. Pour signaler la clé primaire, l'attribut est généralement souligné.

clé étrangère : une clé étrangère dans une relation est formée d'un ou plusieurs attributs qui "correspondent" à une clé primaire dans une autre relation.

### **3 3 - Passage du modèle entités-associations au modèle relationnel**

Pour traduire un schéma du modèle entités-associations vers le modèle relationnel, on doit appliquer les règles suivantes :

- 1 La normalisation doit toujours être effectuée avant le passage au modèle relationnel.
- 2 Chaque entité donne naissance à une relation. Chaque attribut de l'entité devient un attribut de la relation. L'identifiant est conservé en tant que clé de la relation.
- 3 Chaque association dont aucune patte n'a pour cardinalité maximale 1 donne naissance à une relation. Chaque attribut de l'association devient un attribut de la relation. L'identifiant, s'il est précisé, est conservé en tant que clé de la relation, sinon cette clé est formée par la concaténation des identifiants des entités qui interviennent dans l'association.
- 4 Une association dont au moins une patte a pour cardinalité maximale 1 ne devient pas une relation. Elle décrit en effet une dépendance fonctionnelle. La relation correspondant à l'entité dont la patte vers l'association a une cardinalité maximale valant 1, se voit simplement ajouter comme attribut, et donc comme clé étrangère, l'identifiant de l'autre entité.

## 4 - Exemple de mise en oeuvre

### 4 1 - Les informations recueillies

...

*"Chaque salarié, est identifié par un matricule, nommé par son nom et son prénom..., il perçoit un salaire. Les employés perçoivent en plus une prime alors que les vendeurs perçoivent en plus une commission qui est fonction de leur chiffre d'affaires."*

...

*"L'entreprise est organisée en divisions identifiées par un numéro, désignées par un nom ... Chaque salarié, ... travaille dans une seule division ..."*

...

*"Un projet, piloté par un chef de projet, est caractérisé par un numéro, un thème, des dates de début et de fin de réalisation. Il est constitué de tâches caractérisées par un numéro à l'intérieur du projet, un nom symbolique, une description, une date au plus tôt et une date au plus tard de planification et une durée en jours."*

...

## 4 2 - Modéliser le MCD

### entité SALARIE

Propriété	Code	Type	Identifiant	Obligatoire
Matricule du salarié	SALNUM	N6	Oui	Oui
Nom de salarié	SALNOM	VA40		Oui
Prénom de salarié	SALPREN	VA40		Oui
Salaire	SALSAL	N7		Oui

### entité EMPLOYE

Propriété	Code	Type	Identifiant	Obligatoire
Prime	SALPRIM	N7		

### entité VENDEUR

Propriété	Code	Type	Identifiant	Obligatoire
Chiffre d'affaire	SALCAF	N7		
Commission	SALCOM	N7		

### entité DIVISION

Nom de Propriété	Code	Type	Identifiant	Obligatoire
Numéro de division	DIVNUM	N6	Oui	Oui
Nom de division	DIVNOM	VA40		

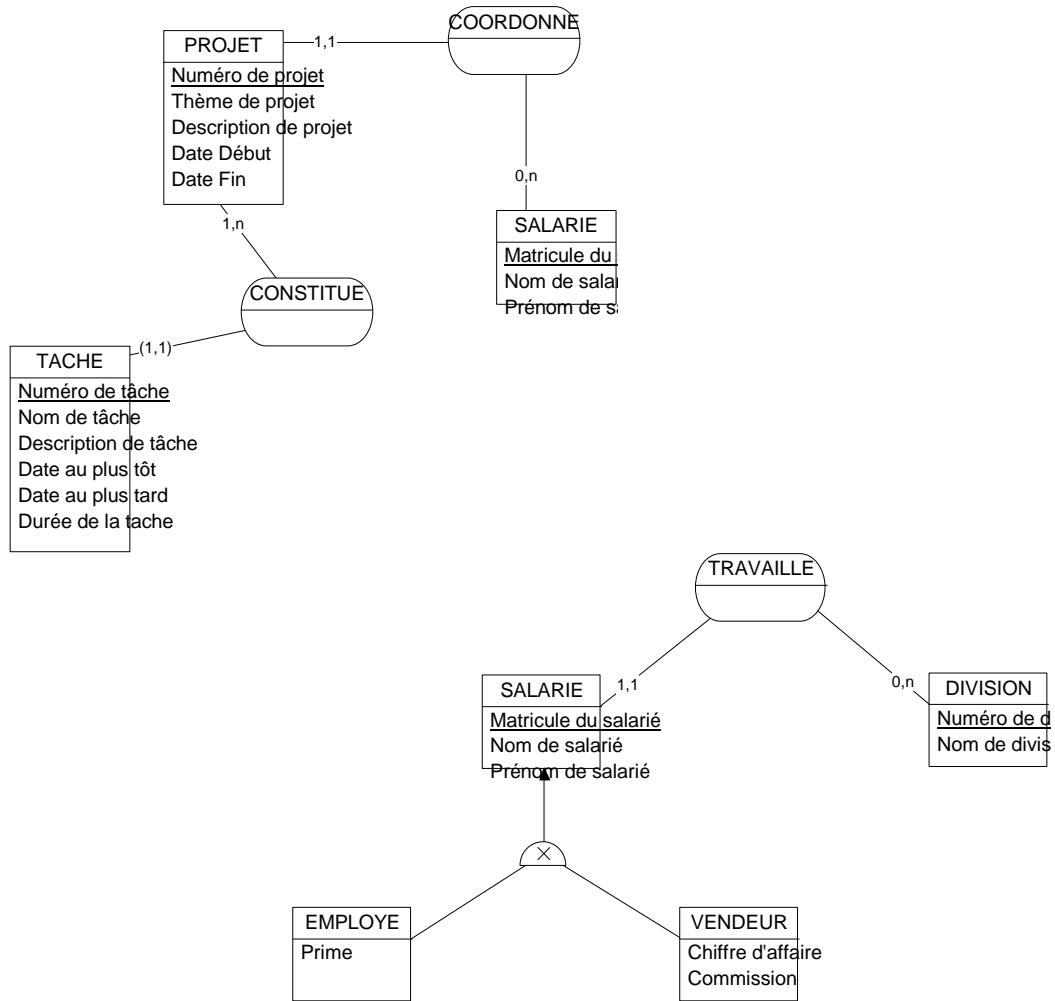
### entité PROJET

Nom de Propriété	Code	Type	Identifiant	Obligatoire
Numéro de projet	PRONUM	N6	Oui	Oui
Thème de projet	PROTHEM	VA40		Oui
Description de projet	PRODESC	VA120		
Date Début	PRODEB	D		Oui
Date Fin	PROFIN	D		Oui

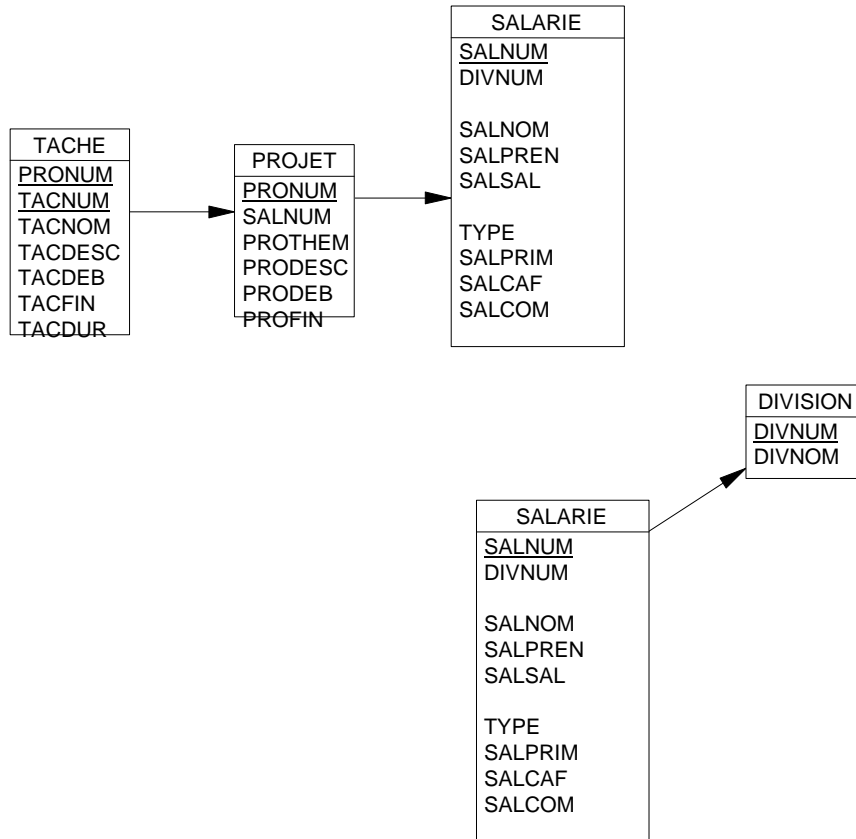
### entité TACHE

Nom de Propriété	Code	Type	Identifiant	Obligatoire
Numéro de tâche	TACNUM	N6	O	Oui
Nom de tâche	TACNOM	VA40		Oui
Description de tâche	TACDESC	VA120		
Date au plus tôt	TACDEB	D		Oui
Date au plus tard	TACFIN	D		Oui
Durée de la tâche	TACDUR	I		Oui

# LES BASES DE DONNEES



## 4 3 - Générer le MPD



## 4 4 - Générer le script de création de la base de données

```
-- =====
--   Table : DIVISION
-- =====
create table DIVISION
(
    DIVNUM      NUMBER(6)           ,
    DIVNOM      VARCHAR2(40)       ,
                                constraint pk_division primary key (DIVNUM)
);

-- =====
--   Table : SALARIE
-- =====
create table SALARIE
(
    SALNUM      NUMBER(6)           ,
    DIVNUM      NUMBER(6)           not null,
    SALNOM      VARCHAR2(40)       not null,
    SALPREN     VARCHAR2(40)       not null,
    SALSAL      NUMBER(7)           not null,
    SALPRIM     NUMBER(7)           ,
    SALCAF      NUMBER(7)           ,
    SALCOM      NUMBER(7)           ,
    TYPE        VARCHAR2(1)        ,
                                constraint pk_salarie primary key (SALNUM)
);

-- =====
--   Table : PROJET
-- =====
create table PROJET
(
    PRONUM      NUMBER(6)           ,
    SALNUM      NUMBER(6)           not null,
    PROTHEM     VARCHAR2(40)       not null,
    PRODESC     VARCHAR2(120)      ,
    PRODEB      DATE               not null,
    PROFIN      DATE               not null,
                                constraint pk_projet primary key (PRONUM)
);
```



```
-- =====
--   Table : TACHE
-- =====
create table TACHE
(
    PRONUM          NUMBER(6)          ,
    TACNUM          NUMBER(6)          ,
    TACNOM          VARCHAR2(40)       not null,
    TACDESC         VARCHAR2(120)      ,
    TACDEB          DATE               not null,
    TACFIN          DATE               not null,
    TACDUR          INTEGER             not null,
    constraint pk_tache primary key (PRONUM, TACNUM)
);

alter table SALARIE
    add constraint fk1_salarie foreign key (DIVNUM)
        references DIVISION (DIVNUM);

alter table PROJET
    add constraint fk1_projet foreign key (SALNUM)
        references SALARIE (SALNUM);

alter table TACHE
    add constraint fk1_tache foreign key (PRONUM)
        references PROJET (PRONUM);
```