

Bases de données

Table des matières

1. Introduction.....	2
2. Problématique : la conception.....	2
3. Objectif et démarche.....	3
3.1. Modèle conceptuel.....	3
3.2. Modèle relationnel.....	3
4. Modèle entité-association.....	4
4.1. Formalisme.....	4
4.2. Les entités.....	5
4.3. Attribut.....	5
4.4. L'Association.....	6
4.4.1. Cardinalité.....	7
4.4.2. Attribut (d'une association).....	8
4.4.3. Contraintes d'intégrité.....	8

La base de données est au centre des dispositifs informatiques de collecte, mise en forme, stockage, et utilisation d'informations. Le dispositif comporte un système de gestion de base de données (SGBD) : un logiciel moteur qui manipule la base de données et dirige l'accès à son contenu. De tels dispositifs — souvent appelés base de données — comportent également des logiciels applicatifs, et un ensemble de règles relatives à l'accès et l'utilisation des informations.



1. Introduction

Les BD¹ sont nées vers la fin des années 1960 pour combler les lacunes des systèmes de fichiers et faciliter la gestion qualitative et quantitative des données informatiques. Les SGBD² sont des applications informatiques permettant de créer et de gérer des BD (comme Oracle ou PostgreSQL par exemple).

Les BD relationnelles, issues de la recherche de Codd, sont celles qui ont connu le plus grand essor depuis plus de 20 ans, et qui reste encore aujourd'hui les plus utilisées. On utilise des SGBDR³ pour les implémenter. Le langage SQL est le langage commun à tous les SGBDR, ce qui permet de concevoir des BD relativement indépendamment des systèmes utilisés.

Les usages de BD se sont aujourd'hui généralisés pour entrer dans tous les secteurs de l'entreprise, depuis les "petites" BD utilisées par quelques personnes dans un service pour des besoins de gestion de données locales, jusqu'aux "grosses" BD qui gèrent de façon centralisée des données partagées par tous les acteurs de l'entreprise. Parallèlement l'accroissement de l'utilisation du numérique comme outil de manipulation de toutes données (bureautique, informatique applicative, etc.) et comme outil d'extension des moyens de communication (réseaux) d'une part; et les évolutions technologiques (puissance des PC, Internet, etc.) d'autre part ont à la fois rendu indispensable et complexifié la problématique des BD.

Les conséquences de cette généralisation et de cette diversification des usages se retrouvent dans l'émergence de solutions conceptuelles et technologiques nouvelles.

2. Problématique : la conception

Selon Wikipédia : « En informatique, une base de données (ou DB⁴ en anglais) est un lot d'informations stockées dans un dispositif informatique. »

Selon Marc Grange : « Une base de données est un ensemble structuré d'informations **non redondantes** dont l'organisation est régie par un **modèle de données**. »

Nous ne savons pas toujours dans quelle table placer certaines colonnes (par exemple, l'adresse de livraison se met dans la table des clients ou dans la table des commandes?);

Ainsi il y a un risque de redondance qui peut engendrer des incohérences en modification, insertion et suppression de données, et rend la base peu performante.

Il y donc nécessité d'une étape préliminaire de conception :

- Définition des données permanentes nécessaires aux besoins d'un ensemble d'utilisateurs.
- La base de données sera relative à un domaine d'application (bibliothèque, service hospitalier, département marketing d'une entreprise, production d'énergie, etc).
- La BD doit contenir toutes les données nécessaires à la représentation du domaine d'application.

1 Bases de Données

2 Système de Gestion de Base de Données

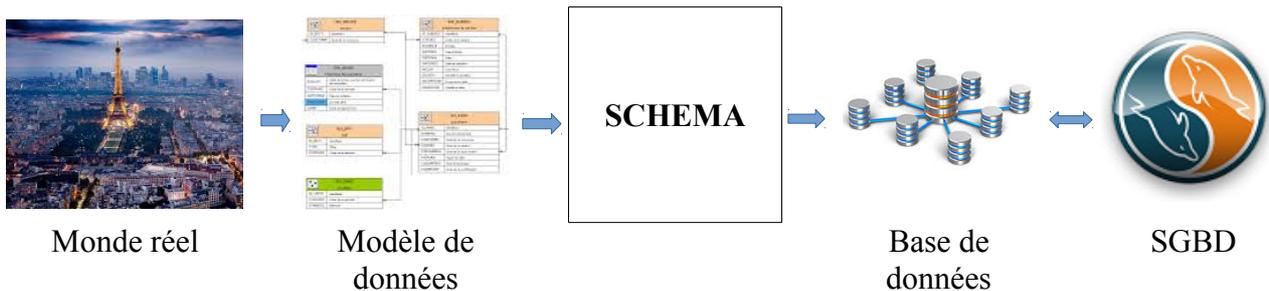
3 Système de Gestion de Base de Données Relationnelles

4 DataBase

3. Objectif et démarche

1. Observation de la réalité : Collecte d'informations
2. Réflexion et modélisation : Méthodologie
3. Définition des tables (relations) d'une BD relationnelle

Les bases de données sont basées sur une approche de « STRUCTURATION » du monde réel qui donne une représentation abstraite (le SCHEMA) résultant de l'application d'un MODELE de DONNEES (data model).



La modélisation d'un problème, c'est-à-dire le passage du monde réel à sa représentation informatique, se définit en plusieurs étapes pour parvenir à son intégration dans un SGBD-R et permettre la manipulation des données par le langage SQL.

Classiquement, le processus de conception des données passe par deux phases :

1. Réalisation d'un modèle conceptuel
2. Traduction en un modèle relationnel

3.1. Modèle conceptuel

C'est la phase d'analyse du problème réel. On cherche à définir les données à utiliser, leur mode d'évolution dans le temps et les relations entre elles.

Quel usage on destine le modèle informatique que l'on est entrain de constituer ?

Il s'exprime dans un formalisme graphique de type entité-association. Il existe d'autres types de formalismes comme le formalisme MCD⁵ de Merise ou UML⁶.

3.2. Modèle relationnel

Il conduit à élaborer l'ensemble des objets manipulables par un SGBD-R et peut être découpé en deux étapes :

1. la conception de modèle logique (représentation en tables indépendantes du SGBD).
2. la traduction en un modèle physique (propre à un SGBD spécifique). Tous les SGBD n'ont pas les mêmes caractéristiques du langage SQL.

Exemple de cahier des charges :

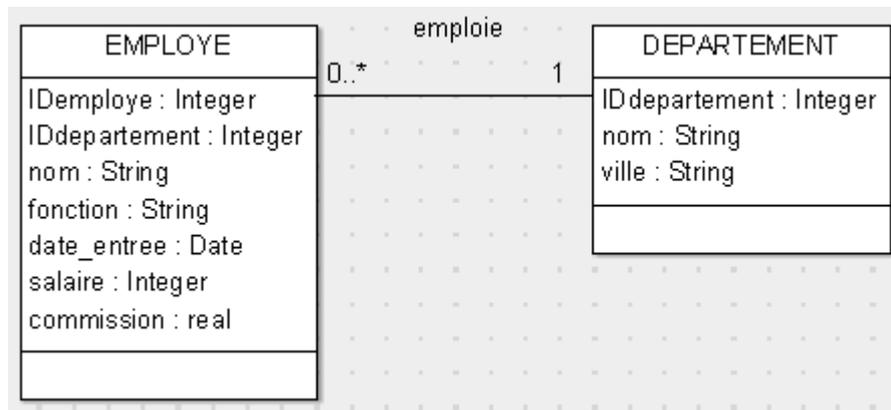
Un service de ressources humaines dans une entreprise veut gérer le personnel. Dans un premier temps, on veut pouvoir connaître le nom, la fonction, la date d'entrée, le salaire, la commission (part de salaire variable) de chaque employé et le numéro du département dans lequel travaille chaque

⁵ Modèle Conceptuel de Données

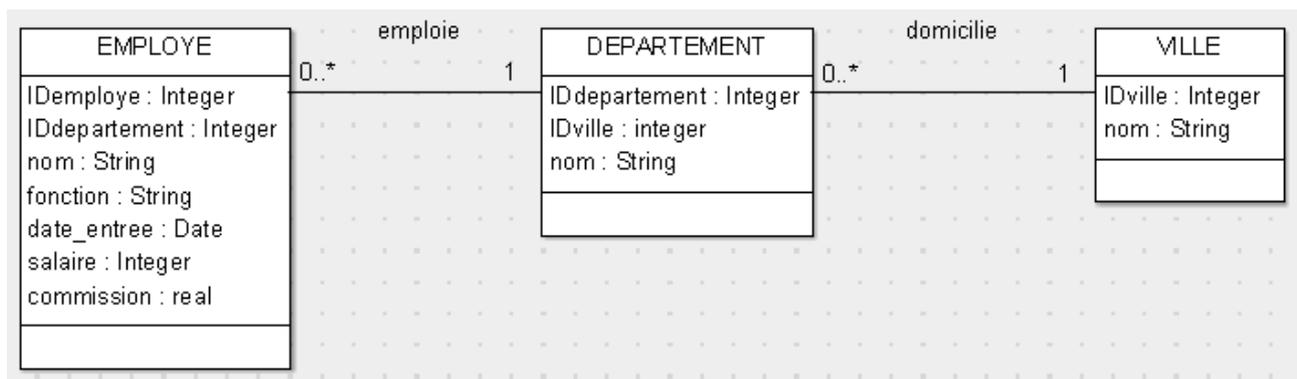
⁶ Unified Modelling Language

employé. Chaque employé a donc les caractéristiques suivantes : **Nom, fonction, date d'entrée, salaire, commission**, numéro du département.

Le service du personnel souhaite aussi connaître le **nom du département** dans lequel l'employé travaille. L'entreprise est répartie dans plusieurs **villes**. Les départements sont donc caractérisés par leur nom et par leur ville. Un employé travaille dans un département et un seul. Il peut y avoir plusieurs départements qui ont le même nom.



modèle UML



version améliorée

EMPLOYE(**IDemploye**, #IDdepartement, nom, fonction, date_entree, salaire, commission)

DEPARTEMENT(**IDdepartement**, #IDville, nom)

VILLE(**IDville**, nom)

4. Modèle entité-association

4.1. Formalisme

Le langage UML définit le diagramme de classes.

Une classe est une description abstraite (condensée) d'un ensemble d'objets du domaine de l'application : elle définit leur structure, leur comportement et leurs relations.

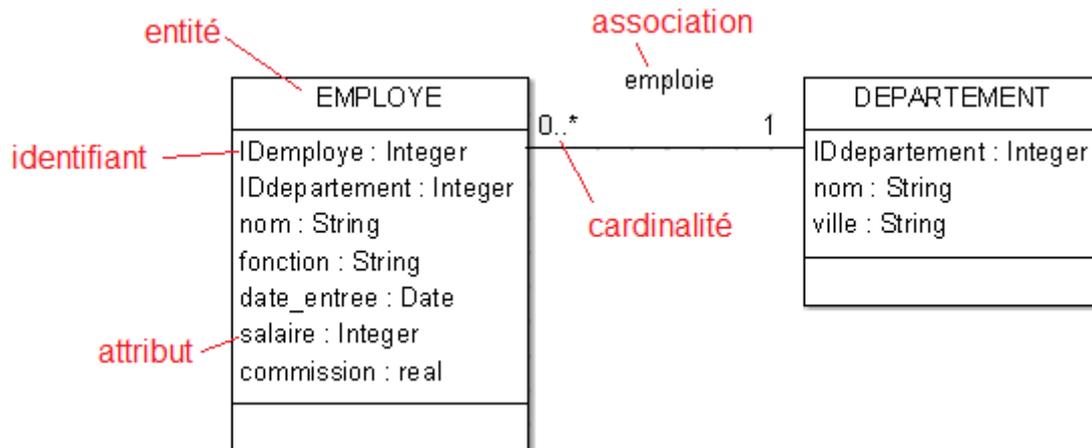
Représentation : les classes sont représentées par des rectangles compartimentés :

- le 1^{er} compartiment représente le nom de la classe
- le 2^{ème} compartiment représente les attributs de la classe

- le 3^{ème} compartiment représente les opérations de la classe

Le modèle sera décrit par:

- Des Types d'Entités (TE)
- Des Types d'Associations (TA)



4.2. Les entités

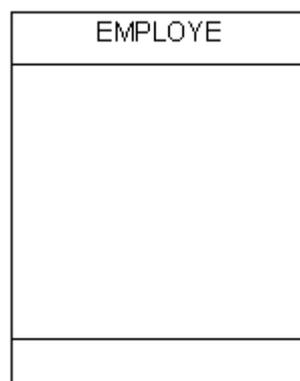
Une entité désigne un ensemble d'objets qui possèdent une sémantique et propriétés communes.

Exemple:

- Le département X a embauché M. Maque
- Le département Y a embauché M. Théon
- Le département X a embauché Mlle. Arya

Il y a 5 individus pouvant être ici regroupés en 2 entités.

Une entités est représentée par un rectangle ; le nom est en majuscule.



4.3. Attribut

Un attribut est la modélisation d'une information élémentaire présente dans le discours.

Exemple:

- nom d'une personne: Maque, Théon, Arya

- date de naissance: 16/08/01, 24/02/88

Il est l'élément descriptif (ou caractéristique) de l'entité ou l'association.

Il est unique dans un modèle conceptuel et ne peut être rattaché qu'à un seul concept (entité ou association).

Un attribut prend ses valeurs dans un ensemble de valeurs possibles appelé "domaine".

ATTRIBUT	DOMAINE
Nom d'une personne	Caractères alphanumériques
Fonction	Caractères alphanumériques
Date d'entrée	Jour/mois/année
Salaire	Nombre entier positif
Commission	Nombre avec 2 décimales

Un attribut est une donnée élémentaire, ce qui exclut les données calculées ou dérivées.

Un identifiant d'entité permet de repérer une entité de manière unique et sans ambiguïté parmi toutes les entités.

Un identifiant peut être constitué d'un ou plusieurs attributs. Le ou les attributs identifiant une entité sont soulignés.

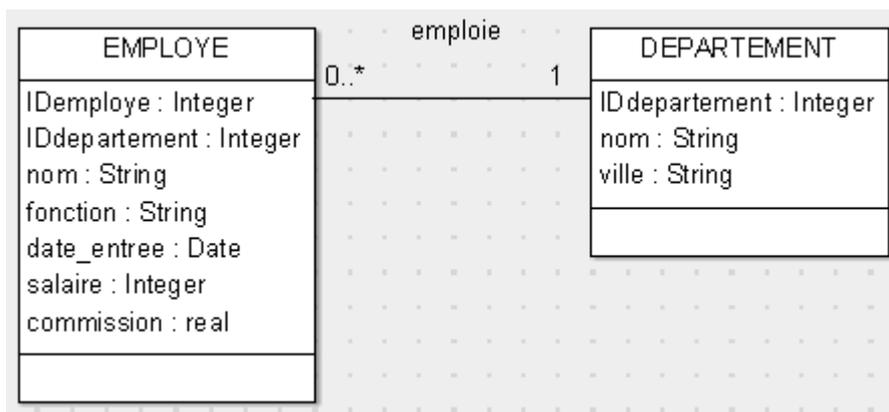
EMPLOYE
<u>IDemploye</u> : Integer
IDdepartement : Integer
nom : String
fonction : String
date_entree : Date
salaire : Integer
commission : real

Remarques :

- Éviter les identifiants composés de plusieurs attributs (comme par exemple un identifiant formé par les attributs nom du client et prénom)
- Préférer un identifiant court pour rendre la recherche la plus rapide possible (éviter par exemple les chaînes de caractères comme le numéro de sécurité sociale ou la plaque d'immatriculation)
- Dans le modèle physique de données, on utilise une clé numérique(un entier) incrémenté automatiquement.

4.4. L'Association

Une Association est une liaison entre plusieurs entités, avec une signification précise. Exemple : un employé travaille dans un département.



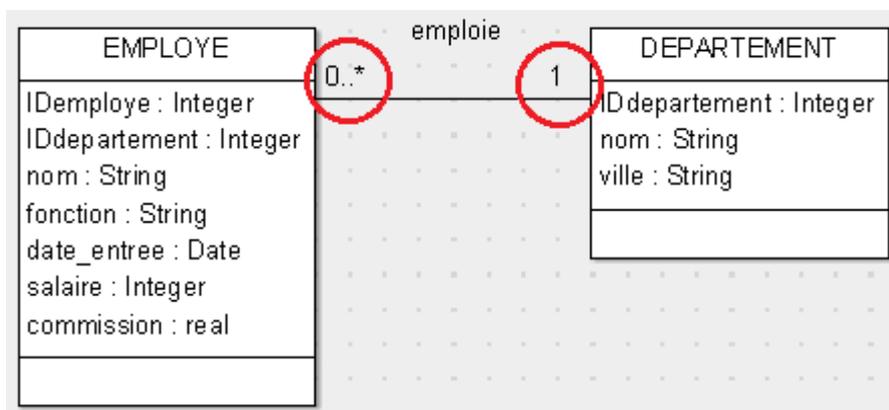
Une association peut être porteuse d'informations :

- Les cardinalités
- Les attributs
- Les contraintes (CIF)

4.4.1. Cardinalité

La cardinalité exprime la façon dont sont associées les entités : c'est à dire les nombres minimal (min) et maximal (max) d'associations possibles entre deux entités.

Exemple:



La cardinalité se lit comme suit :

- Un département emploie plusieurs employés (min=0, max=n).
- Un employé ne peut être employé que par un seul département (min=1, max=1).

Il existe plusieurs types de cardinalité :

- 0,1
- 1,1 (1)
- 0,N (0,*)
- 1,N (1,*)

Le premier nombre de la cardinalité représente le nombre minimal d'associations possibles (min) et

le second nombre, le nombre maximal d'associations possibles (max).

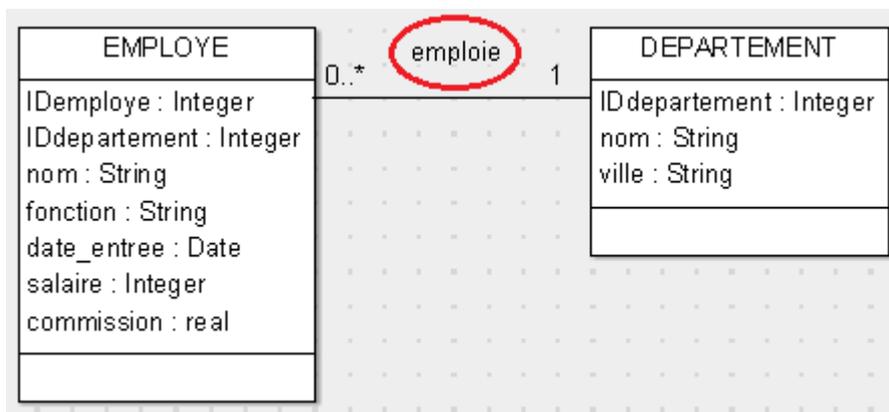
Règles :

1. L'expression de la cardinalité est obligatoire.
2. Il ne peut y avoir de cardinalité maximale égale à 0.
3. Si une cardinalité est connue et vaut 2 ou plus, alors nous considérons qu'elle est indéterminée et vaut n. En effet, si cette valeur est définie lors de la conception, il se peut qu'elle évolue dans le futur. Il faut donc considérer n comme inconnue dès la conception.

4.4.2. Attribut (d'une association)

L'attribut d'une association représente les informations qui ne peuvent prendre de sens qu'avec la présence de l'ensemble des entités constituant cette relation.

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



4.4.3. Contraintes d'intégrité

Les contraintes d'intégrité représente toutes les règles implicites ou explicites que doivent suivre les données :

- Contraintes d'entité : toute entité doit posséder un identifiant.
- Contraintes de domaine : les valeurs de certains attributs doivent être prises dans un ensemble donné.
"La fonction d'un employé prend sa valeur dans l'ensemble {Secrétaire, Technicien, Ingénieur, Chef de projet}"
- Contraintes d'unicité : une valeur d'attribut ne peut pas être affectée deux fois à deux entités différentes.
"Un département, identifié par son numéro, a un nom unique (il n'y a pas deux départements de même nom)"
- Contraintes générales : règle permettant de conserver la cohérence de la base de manière générale.
"Un même examen ne peut pas avoir lieu dans deux salles différentes à la même date et à la même heure"