

Concevoir le modèle relationnel d'une base de données

Base de données

Didier TEXIER
DTX

Concevoir le modèle relationnel d'une base de données

1	HISTORIQUE	6
1.1	FONCTION D'UN SGBD	6
1.2	TYPES DE SGBD	7
1.2.1	MODÈLE HIÉRARCHIQUE	7
1.2.2	MODÈLE RÉSEAU	7
1.3	LES DIFFÉRENTS MODÈLES	7
2	INTRODUCTION	8
3	ÉTAPES D'UN PROJET.....	10
3.1	LES DIFFÉRENTES ÉTAPES D'UN PROJET.....	10
4	FORMALISME DU MODÈLE RELATIONNEL	12
4.1	DÉFINITIONS	12
4.2	LE MODÈLE CONCEPTUEL DE DONNÉES (MCD).....	13
4.2.1	ÉTAPES DE CONSTRUCTION DU MCD.....	14
4.3	LE MODÈLE LOGIQUE DE DONNÉES (MLD).....	15
4.3.1	RÈGLES DE PASSAGE DU MCD AU MLD.....	15
4.4	LE MODÈLE PHYSIQUE DE DONNÉES (MPD)	17
4.4.1	LES OBJETS	17
4.4.2	LES TYPES.....	17
4.4.3	LES CONTRAINTES	18
4.4.4	LE DICTIONNAIRE DE DONNÉES	18
5	FORMES NORMALES :	19
5.1	OBJECTIFS	19
5.2	1 ^{ÈRE} FORME NORMALE :	19
5.3	2 ^{ÈME} FORME NORMALE :	20
5.4	3 ^{ÈME} FORME NORMALE :	20
5.5	FORME NORMALE DE BOYCE-CODD	21
6	LES OPÉRATEURS RELATIONNELS	22
6.1	PROJECTION.....	22
6.2	RESTRICTION.....	22
6.3	JOINTURE	23
6.4	PRODUIT CARTÉSIEN	23
6.5	LA DIVISION	24
6.6	OPÉRATEURS ENSEMBLISTES : UNION, INTERSECTION, DIFFÉRENCE.....	24

7	SQL, LES DIFFÉRENTS LANGAGES	25
8	LES OUTILS DE MODÉLISATION	26
9	EXERCICES	27
9.1	GESTION DES ACTIVITÉS D'UN CENTRE DE LOISIRS	27
9.2	GESTION D'UN COLLÈGE	27
9.3	GESTION DE MA PETITE ENTREPRISE.	29
9.4	FORMES NORMALES	30
9.5	GESTION D'UNE BIBLIOTHÈQUE	31
10	CORRIGÉS	33
10.1	MA PETITE ENTREPRISE	33
10.2	FORMES NORMALES	33
10.3	GESTION D'UNE BIBLIOTHÈQUE	34
11	ANNEXE	35
12	BIBLIOGRAPHIE	36
13	INDEX	38

« Le Tout est plus que la somme des parties¹ »

Edgar Morin

AVANT PROPOS

Ce support de cours est un outil personnel, il ne constitue pas un guide de référence.

C'est un outil pédagogique élaboré dans un souci de concision : il décrit les concepts essentiels à connaître pour appréhender le sujet de la formation, c'est-à-dire la conception de base de données relationnelles.

La modélisation des données fait appel à des branches des mathématiques comme la théorie des ensembles et l'algèbre relationnelle. Ces notions théoriques ne sont pas du propos de ce support.

Il apparait que les développeurs négligent l'aspect conceptuel d'une base de données, ce qui ne sera pas sans conséquences par la suite.

¹ Edgar Morin °1921, philosophe. Pensée complexe et holisme.

1 HISTORIQUE

Le début des années 70 voit le développement des bases de données.

Edgar Frank « Ted » CODD² informaticien mathématicien d'origine britannique et chercheur chez IBM, pose les fondements du modèle relationnel basé sur la théorie des ensembles et l'algèbre relationnelle.

Il énonce 12 règles (les règles de Codd) pour qu'un système de gestion de base de données soit relationnel SG BD/R.

Une base de données (BD) est en ensemble **structuré** d'informations.

Une base de données relationnelle c'est quoi ?

Une base de données relationnelle est un ensemble d'informations dont l'organisation respecte des règles précises et qui peut être interrogée par le langage SQL.

Un système de gestion de base de données relationnelle SGBR/R est un logiciel permettant la gestion des données en utilisant le modèle relationnel.

Ce logiciel est généralement complété par **un ensemble d'outils** de conception, de développement, d'administration, des pré-compilateurs, des outils de génération de formulaires et d'états et d'analyse des données.

1.1 Fonction d'un SGBD

Un SGBD doit permettre de :

- Décrire les données qui seront stockées,
- Manipuler ces données (ajouter, modifier, supprimer),
- Obtenir des renseignements à partir de ces données (sélectionner, trier, calculer, agréger,...)
- Définir des contraintes d'intégrités sur les données (contraintes de domaines, d'existence, ...)
- Définir des protections d'accès (mots de passe, autorisations, privilèges, ...)
- Résoudre les problèmes d'accès concurrentiels aux données (verrouillages, interblocages)
- Prévoir des procédures de reprise en cas d'incident (sauvegardes, journaux)

De plus, un SGBD doit permettre d'écrire des applications indépendantes de l'implémentation physique des données. On parlera d'indépendance logique et physique des données.

² 1923-2003

1.2 Types de SGBD

Historiquement, les premiers SGBD étaient du type hiérarchique, puis sont apparus les SGBD de type réseau. Aujourd'hui, la plupart des systèmes sont de type relationnel ou *objet*.

La différence essentielle entre ces types de SGBD réside dans les modèles utilisés pour représenter les données.

1.2.1 Modèle hiérarchique

Les données sont représentées sous forme d'une structure arborescente d'enregistrements. Cette structure est définie avec des **pointeurs** et détermine le chemin d'accès aux données.

1.2.2 Modèle réseau

La structure des données peut être représentée sous la forme d'un **graphe**. Comme pour le modèle hiérarchique, la structure est conçue à l'aide de pointeurs et détermine le chemin d'accès aux données.

Pour les modèles hiérarchique et réseau, les programmes ne sont pas indépendants de la structure logique de la base de données et du chemin d'accès aux données : ils doivent décrire comment retrouver les données (*naviguer* dans la base) et si, par exemple, on supprime un index, tous les programmes doivent être modifiés.

1.3 Les différents modèles

Certains noms sont associés à des théories ou des modèles,

- Modèle relationnel : **Codd**
- Théorie des systèmes : **Lemoigne**
- Modèle individu-Relation : **Tardieu**³
- Modèle Entité-Relation : **Chen**⁴

A la même période en France, le projet MERISE dont l'équipe est pilotée par H. Tardieu se met en place avec les acteurs suivants :

- Université et Centre d'Étude Technique de l'Équipement (Aix en Provence)
- ESN : SEMA, CAP GEMINI , SOGETI, CGI, GAMMA, SLIGOS, STERIA, ...
- Ministère de l'Industrie et le Centre Technique Informatique

³ C'est dans le cadre de travaux de recherche qu'Hubert Tardieu publie « Modèle individuel pour un base de données » en 1974.

⁴ C'est en 1976 que Peter Chen publie dans la revue de l'ACM (*Association for Computing Machinery*) « The Entity Relationship Model - Toward A Unified View of Data »

2 INTRODUCTION

MERISE, l'origine du terme est incertaine.

Certains y voient l'acronyme **M**éthode d'**É**tude et de **R**éalisation Informatique par les **S**ous-Ensembles mais plus vraisemblablement du merisier (*Prunus avium*) arbre particulièrement apte à la greffe...

MERISE est la greffe de plusieurs méthodes.

MERISE est une méthodologie d'informatisation dont les origines remontent à 1974. Les initiateurs de cette méthode sont des chercheurs français (*Hubert Tardieu, Jean-Louis Lemoigne*) travaillant sur des projets au ministère de l'industrie (CTI) et dans les services de l'équipement (CETE), relayés par des travaux de l'université d'Aix-Marseille et de l'INRIA. Le nom Merise est déposé en 1978. La norme AFNOR Z67-101 reprend dans ces « *recommandations pour la conduite de projets informatiques* » les éléments de la méthode. Merise est une méthode inspirée de la perception systémique des organisations. Elle repose sur une vue globale de l'organisation et intègre tous les aspects d'un SI : du pilotage stratégique à la description du système opérant.

MERISE est une **méthode d'analyse** pour les projets informatiques. MERISE est un langage qui formalise les choix à effectuer pour les systèmes d'information (et non sur l'informatique appliquée).

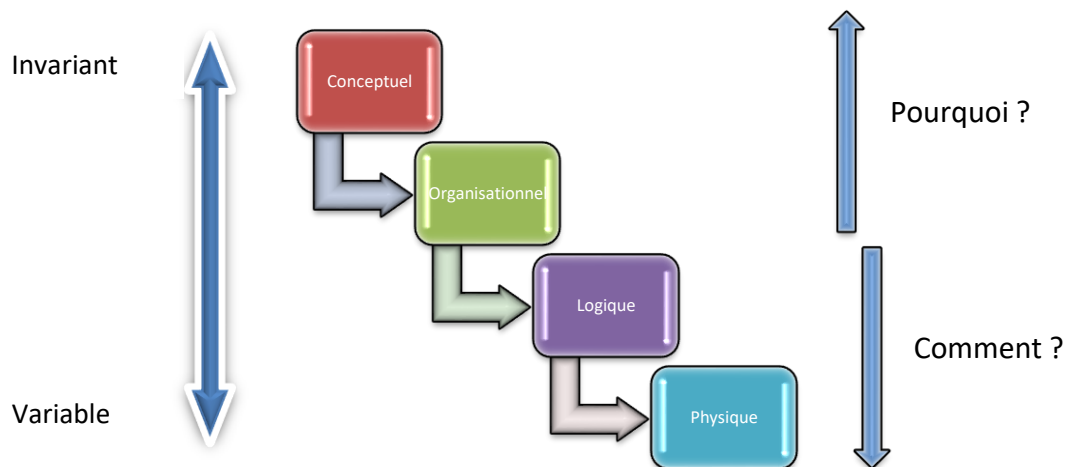
C'est une démarche (ou une méthode) qui peut être suivie pour élaborer un système d'informations. Cette méthode vise à dissocier les données des traitements. Les concepts utilisés sont peu nombreux et (relativement) simples.

De fait, MERISE pourra aider l'entreprise à modifier son organisation.

MERISE met en évidence 3 niveaux d'abstraction.

- **Le niveau conceptuel** dans lequel les contraintes organisationnelles ne sont pas considérées. Ce niveau traduit les objectifs et les contraintes de l'entreprise. On se limitera à établir le QUOI, tant au point de vue des données que des traitements. Il possède un haut niveau d'invariance. Il répond à la question : Que fait l'organisme ?
- **Le niveau organisationnel** définira l'organisation qui devra être mise en place dans l'entreprise pour atteindre les objectifs. Il précisera les postes de travail, la chronologie des opérations. On répondra aux QUI, OÙ et QUAND ou *Qui fait Quoi ?*
- **Le niveau opérationnel** encore appelé logique, technique ou physique, apportera les solutions techniques au problème : le COMMENT. Il est sujet à plus de changements ; son niveau d'invariance est fonction des évolutions technologiques. Ce niveau est découpé en 2 niveaux :
 - **Le niveau logique ou externe** (indépendant des langages de programmation ou de gestion des données)

- **Le niveau physique ou interne** (l'outil informatique)



De même, MERISE décrit les échanges internes et externes de l'entreprise en 3 domaines :

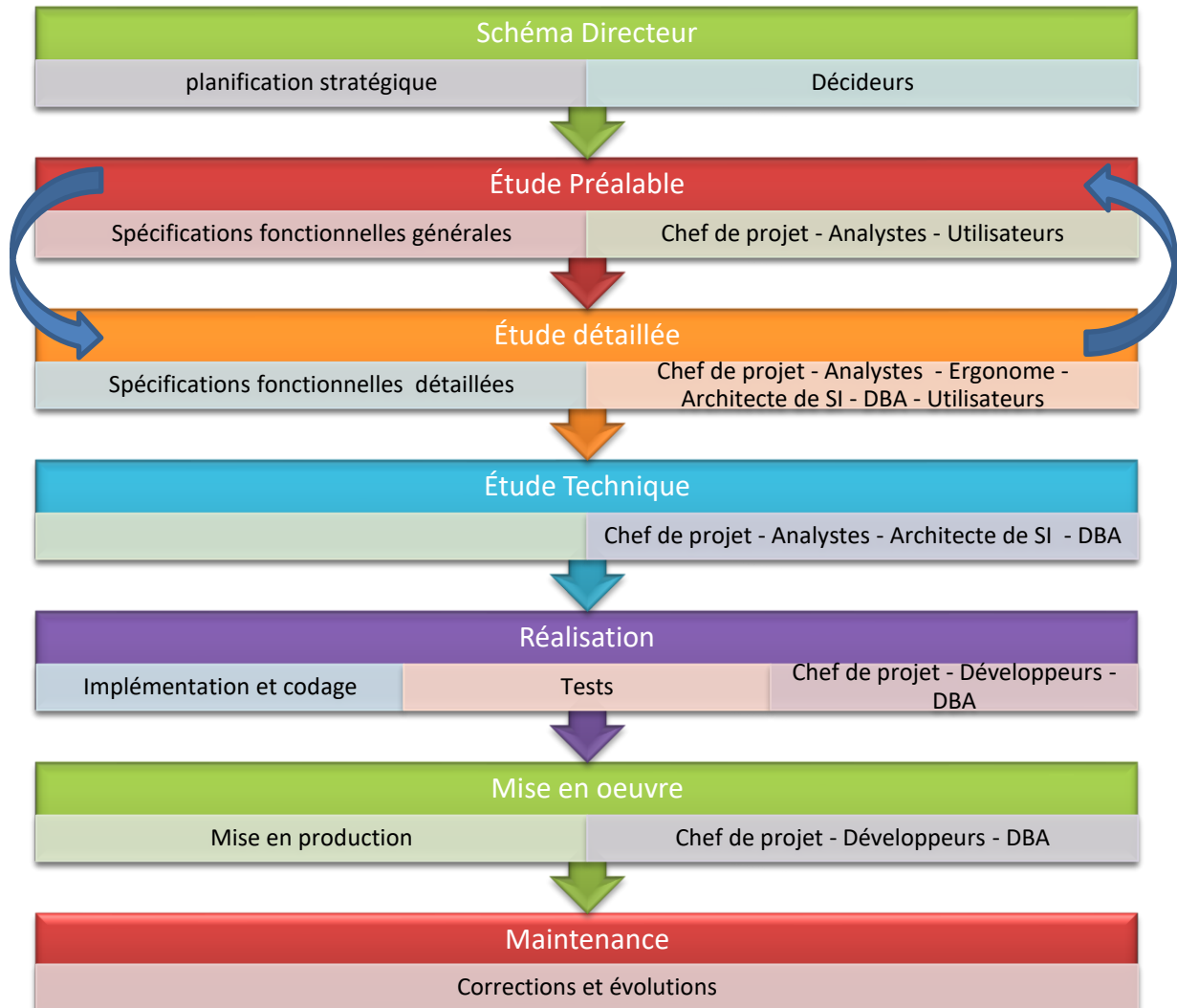
- **La communication** : les flux d'informations ou de messages entre les systèmes
- **Le traitement** : la description des tâches à effectuer à la réception ou émission des flux d'informations.
- **Les données** : la description des structures de mémorisation des données.

NIVEAUX	MODÈLES		
	Domaine des données	Domaine des communications	Domaine des traitements
CONCEPTUEL	Modèle Conceptuel MCD	Modèle Conceptuel MCA (Activités) MCC	Modèle Conceptuel MCT
ORGANISATIONNEL	Modèle Logique MLD	Modèle Logique MOC	Modèle Logique MOT
PHYSIQUE	Modèle Physique MPD	Modèle Physique MPC	Modèle Physique MPT

MERISE décrira ces modèles sous forme de schémas.

3 Étapes d'un projet

3.1 Les différentes étapes d'un projet⁵



Le schéma directeur : au cours de cette étape, la direction générale fixera la stratégie et les grandes orientations pour les années à venir. Le schéma directeur détermine les principaux projets à développer et leur enchaînement.

L'étude préalable sera menée pour chacun des domaines retenus par le schéma directeur.

L'étude préalable comporte :

- L'étude conceptuelle
- L'étude organisationnelle

⁵ Le champ d'application de MERISE s'étend à l'étude préalable et à l'étude détaillée.

- La définition des outils

Cette étude peut être décomposée en 3 parties :

- Étude de l'existant
- Scénarii futurs
- Évaluation des scénarii

Cette étude préalable réalisée après les interviews des différents intervenants de l'entreprise, permettra de construire :

- Le **DICTIONNAIRE DE DONNÉES**
- Les **RÈGLES DE GESTION**

L'étude détaillée sera réalisée pour chaque projet. Chaque domaine sera ainsi découpé en projets.

Le domaine, au sens *systemique*, est un système d'information homogène.

L'objectif de cette étude détaillée est de réaliser le **cahier des charges utilisateurs**.

L'étude technique pour chaque application afin de rédiger le **cahier des charges de réalisation**.

La réalisation, c'est le développement, les tests et la mise en œuvre.

La maintenance, c'est la mise à jour suite aux modifications demandées par l'utilisateur.

4 Formalisme du modèle relationnel

Le modèle relationnel est fondé sur une branche des mathématiques nommée *Algèbre Relationnelle*. Il conduit à une représentation simple des données sous forme de tables et non plus de pointeurs. Cette représentation et l'algèbre relationnelle ont permis le développement de langages puissants non procéduraux.

Dans ces langages, l'utilisateur ou le développeur indique quelles informations il souhaite obtenir et c'est le moteur du SGBD qui trouve le moyen d'obtenir le résultat. Ces langages peuvent être utilisés par des non-informaticiens et permettent l'écriture de programmes indépendants de la structure logique et physique des données. Un de ces langages est le **SQL** (*Structured Query Language*).

Le modèle relationnel est encore appelé :

- Entité-Relation
- Entité-Association
- Objet-Relation

4.1 Définitions

- Caractéristiques :
Simple et naturel : compris par tous.
Rigoureux et non ambigu.
- Type :
Concept abstrait générique exprimant l'essence d'une classe d'objet.
- Domaine :
Ensemble, fini ou infini, des valeurs possibles que peut prendre un attribut. Dans l'espace des bases de données, le domaine est synonyme de *type de colonnes*.
- Propriété :
Plus petit élément d'information manipulé par le système d'information **qui a un sens pour l'organisme**.
 - Simple ou composée
Exemples : le nom d'un salarié (simple) et son adresse (composée).
- Occurrence :
Élément individualisé et appartenant à un type.
- Individu :
Ou entité est un regroupement de propriétés.
- Clé candidate :

Ensemble minimal de propriétés d'un individu, telle qu'il n'existe pas deux occurrences de cet individu pour lesquelles cette propriété ou cet ensemble de propriétés puisse prendre une même valeur. Une clé candidate peut être un attribut artificiel utilisé à cette fin.

- Degré :
Nombre d'attributs d'une relation (entité)

4.2 Le Modèle Conceptuel de Données (MCD)

Le modèle conceptuel de données est une représentation graphique des informations manipulées par le système d'information (SI).

On partira du **dictionnaire de données** et des **règles de gestion** pour en faire une représentation schématique indépendante de l'organisation et des solutions techniques.

Après un recueil rigoureux d'information sur le terrain auprès des utilisateurs et bien sûr, des nouvelles demandes ; on construira le dictionnaire des données.

Typiquement, ce dictionnaire contient :

- Les données retenues non redondantes
- Leur type de données
- Des commentaires

Exemple : le nom, prénom, adresse, salaire d'un employé. Pour les 3 premières de type alphanumérique et le salaire de type numérique.

Données	Type	Commentaire	Synonymes
Nom employé	Alphabétique	Le nom d'un employé	Patronyme
Prénom employé	Alphabétique	Le prénom d'un employé	
Adresse	Alphanumérique	L'adresse postale d'un employé	
Salaire employé	Numérique	Le salaire annuel brut d'un employé	

Exemple : le nom, prénom, adresse, salaire d'un employé. Pour les 3 premières de type alphanumérique et le salaire de type numérique.

Il « suffira » de dégager à partir de ces informations (données et règles) des **ENTITÉS** et des **RELATIONS** entre-elles.

EMPLOYE
<u>Matricule</u>
Nom
Prenom
Adresse
Salaire

DEPARTEMENT
<u>Id_dept</u>
Nom_departement

← Nom de l'entité

← Propriétés

L'entité est un objet composé de propriétés dont l'une sera son identifiant. Ce sont des **objets de gestion** comme par exemple : un article, un client, un employé, une facture, ...

L'identifiant est une ou plusieurs propriétés permettant d'identifier de manière unique une occurrence de l'objet. L'identifiant sera représenté souligné et apparaîtra en premier dans la liste des propriétés.

Exemple : un matricule, un numéro de facture, un code service, ...

Puis, on déterminera les relations entre les entités. Une relation exprime une association entre des entités. Ces relations seront :

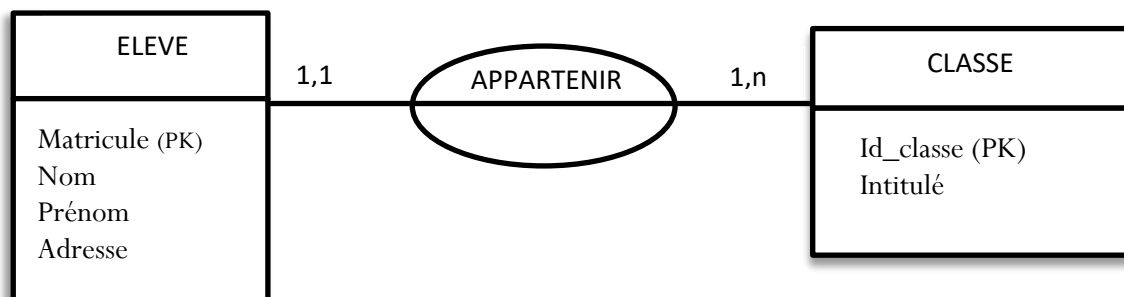
- Définies par un verbe
- De dimension binaire, ternaire ou n-aire
- Porteuses d'information (auront leurs propres propriétés)
- Renseignées par des cardinalités

La **dimension** d'une relation définit le nombre d'entité qui participe à cette relation.

La **cardinalité** exprime le nombre d'occurrences d'un objet qui participe à la relation. On s'intéressera uniquement aux cardinalités minimales et maximales.

- 0,1 : une occurrence de l'objet participe au plus une fois à la relation.
- 1,1 : une occurrence de l'objet participe une fois et une seule à la relation.
- 1,n : une occurrence de l'objet participe au moins une fois à plusieurs à la relation.
- 0,n : aucune précision quant à la participation des occurrences de l'objet.

Exemple :



4.2.1 Étapes de construction du MCD

L'ordre des étapes sera le suivant :

1. Déterminer la liste des entités.

2. Pour chacune d'elles
 - a. Établir la liste des propriétés
 - b. Trouver un identifiant (naturel ou non)
3. Construire les relations entre les entités
 - a. Trouver la dimension
 - b. Trouver les cardinalités
 - c. Faire la liste des attributs
4. Vérifier que le schéma réponde à l'étude et le valider avec les utilisateurs

4.3 Le Modèle Logique de Données (MLD)

Le MLD va nous permettre de définir l'organisation logique des données traduisant le modèle conceptuel précédemment validé et d'optimiser cette organisation par rapport aux besoins des traitements.

À ce niveau, on ne parlera pas encore de choix techniques mais de types des bases de données (hiérarchiques, réseaux, navigationnelles, relationnelles, ...).

4.3.1 Règles de passage du MCD au MLD

Avant de voir les différentes étapes de passage MCD → MLD, faisons un peu de vocabulaire.

4.3.1.1 Vocabulaire

Une base de données relationnelle est composée d'un ensemble de *tables* ou *relations*.

- La **table** possède un nombre déterminé de colonnes.
- La **colonne** correspond à une propriété et prend ses valeurs dans un domaine. Le domaine peut être simple ou composé.
- La **ligne** représente une occurrence de la table, par exemple, toutes les informations d'un employé. La ligne est identifiée par une clé primaire (**PK : Primary Key**) qui possède obligatoirement une valeur et est unique.

On définit les synonymes suivants :

- Table – Entité - Relation
- Propriété – Attribut – Colonne
- Ligne – n-uplet – Tuple (**Table UPLET**) – Occurrence - (Enregistrement)

4.3.1.2 Les entités

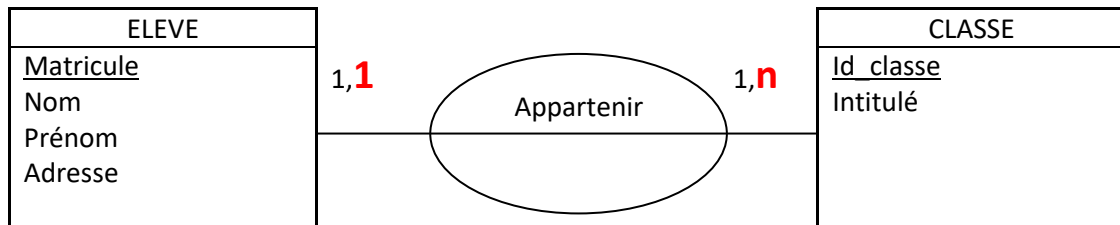
- Les entités du MCD deviennent des tables de la base de données.
- Les colonnes sont les propriétés de l'entité.



Une relation ne peut contenir 2 tuples identiques : chaque ligne est unique !

4.3.1.3 Les relations

Les relations dont les cardinalités **maximales** valent **1 et n** sont traduites par une flèche dirigée de 1 vers n ET par la migration de l'identifiant de l'entité possédant la cardinalité n vers l'entité de cardinalité 1.

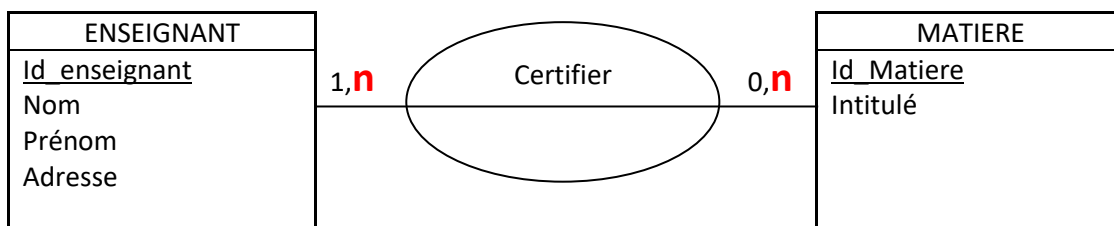


Devient :

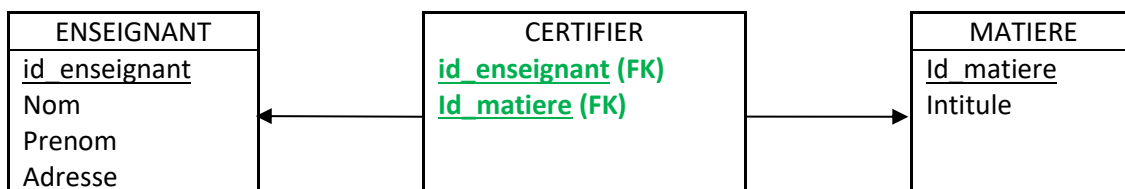


Les relations dont les cardinalités **maximales** valent **n et n** sont traduites par la création d'une table dont :

- Le nom est le nom de la relation
- Les propriétés sont les identifiants des entités qui participent à la relation.
- La clé de cette nouvelle table est constituée par la concaténation des identifiants des entités.



Devient :



4.3.1.4 Les contraintes d'intégrité

1. Choisir avec précision les domaines de chaque colonne et en déduire leur **type**⁶.

⁶ Type et Domaine sont aujourd'hui synonymes

2. Définir la **clé primaire** qui est soit un ensemble de colonnes existantes, soit une nouvelle colonne. Cette clé primaire permet de définir de manière unique une ligne dans la table. Elle sera soulignée dans le modèle et en première position.
3. **Clé étrangère** : résulte de la relation 1-n entre 2 entités. Cette clé doit être de même type que la clé primaire à laquelle elle fait référence.
4. **Contrainte obligatoire** : précise si la colonne doit être obligatoirement renseignée.
5. **Contrainte d'intégrité référentielle** : relation entre 2 colonnes (ou plus) de 2 tables. Vérifie l'existence de la clé primaire dans la table référencée. Exprime les cardinalités 1-1 ou 1-n.

4.4 Le Modèle Physique de données (MPD)

Le modèle physique de données est un ensemble de tables stockées dans des fichiers composant une base de données relationnelle.

Le langage utilisé pour interroger la base de données est le **SQL** (sigle de *Structured Query Language*, en français langage de requête structurée) est un langage informatique normalisé par l'ANSI⁷ servant à effectuer des opérations sur des bases de données relationnelles.



Attention, dans certains outils le MLP et le MPD sont parfois confondus.

4.4.1 Les objets

Le modèle physique peut être composé de :

- Tables
- Vues (tables logiques)
- Index (arbre généralement de type B-Tree)
- Synonymes (alias)
- Liens de base de données (Database Link) Chaîne de connexion vers une base distante
- Morceaux de code (fonctions, procédures, packages, déclencheurs)
- ...

4.4.2 Les types

Les types de données seront :

- Des chaînes de caractères (*CHAR* ou *VARCHAR*)
- Des numériques entiers ou décimaux (*NUMBER(p,s)*)
- Des dates (*DATE*)
- Des booléens
- Des types spéciaux utilisés pour des données de taille importante (*LOB*).
- ...

⁷ ANSI, American National Standards Institute

4.4.3 Les contraintes

Les contraintes d'intégrité vérifient les règles de validité des données conformes aux règles de gestion. Ce contrôle est effectué par le noyau avant chaque modification des données (ajout, modification ou suppression).

Les contraintes seront de nature :

- **Type** conforme au domaine et à l'usage de la propriété.
- Obligatoire (**NOT NULL**), la colonne doit avoir une valeur.
- Unicité (**UNIQUE**), la colonne ne doit pas avoir de doublon.
- Clé primaire ou identifiant (**PRIMARY KEY**), la colonne est renseignée et est n'a pas de doublon.
- Clé étrangère ou contrainte référentielle (**FOREIGN KEY**), la colonne fait référence à une autre colonne d'une autre table.
- De vérification (**CHECK**), la colonne doit vérifier une condition.

Exemple d'écriture d'une contrainte CHECK :

```
ALTER TABLE Employe
ADD CONSTRAINT emp_salary_min
CHECK (salary > 32000) ;
```

Autre exemple :

```
ALTER TABLE Employe
ADD CONSTRAINT ck_emp_courriel
CHECK (courriel LIKE '%@%.fr') ;
```

4.4.4 Le dictionnaire de données

C'est le méta-modèle.

Le contenu du Dictionnaire de Données sert à décrire les structures du modèle application.

Exprimé par la **4^{ème} règle de Codd** :

Le catalogue doit être aussi relationnel.

5 FORMES NORMALES :

5.1 Objectifs

L'objectif principal de la normalisation est de limiter le risque d'incohérence des données. C'est-à-dire éviter autant que possible, les redondances et les valeurs nulles.

La normalisation aura besoin d'une sémantique plus riche sur les données qui s'exprime par des dépendances fonctionnelles entre attributs.

La normalisation a comme avantage de simplifier la mise à jour des données d'améliorer les performances des traitements et d'augmenter la solidité du modèle.

En contrepartie, rend le schéma conceptuel plus complexe.

La normalisation étant réversible, ne pas hésitez, parfois, à dénormaliser.

Il existe 8 formes normales, mais généralement on s'intéressera aux 3 ou 4 premières.

5.2 1^{ère} Forme Normale :

Cette forme normale (**FN1**) a pour objectif d'éliminer les groupes répétitifs dans une table.

Dans tout objet de gestion (individu ou relation), si tout attribut contient une valeur atomique (du point de vue du contexte), autrement dit, une valeur non multiple ou non composée.

C'est-à-dire que pour une occurrence d'individu ou de relation, chaque propriété doit avoir un sens et ne peut posséder qu'une seule valeur.

Pas de valeurs répétitives et constantes dans le temps.

Nom	Notes
Boltzmann	12, 8, 15
De Broglie	7, 14, 18
Dirac	8,13, 16
Einstein	9, 12, 17
Kaluza	15, 17, 16

N'est pas en FN1

Nom	1 ^{er} trimestre	2 ^{ème} trimestre	3 ^{ème} trimestre
Boltzmann	12	8	15
De Broglie	7	14	18
Dirac	8	13	16
Einstein	9	12	17
Kaluza	15	17	16

Est en FN1

Solutions :

- Sortir le groupe répétitif de la table
- Transformer ce groupe en table

5.3 2^{ème} Forme Normale :

Une relation est en 2^{ème} Forme Normale (FN2) si :

- La relation est en FN1
- Et tout attribut non clé dépend totalement de la clé. Aucun attribut ne doit dépendre d'une partie seulement de la clé.

Cette Forme Normale ne doit être vérifiée que pour les relations ayant une clé composée.

On dit aussi : *tout attribut ne faisant pas partie de la clé est en dépendance fonctionnelle complète avec la clé.*

Unité	Laboratoire	Directeur	Ville
UFR 12	Quantique	Planck	Kiel
UFR 12	Nucléaire	Curie	Kiel
UFR 16	Optique	Bohr	Copenhague
UFR 7	Atomique	Minkowski	Alexotas
UFR 8	Mathématique	Klein	Stockholm

N'est pas en FN2, car il existe une relation Unité → Ville

Unité	Laboratoire	Directeur
UFR 12	Quantique	Planck
UFR 12	Nucléaire	Curie
UFR 16	Optique	Bohr
UFR 7	Atomique	Minkowski
UFR 8	Mathématique	Klein

Est en FN2

Unité	Ville
UFR 12	Kiel
UFR 16	Copenhague
UFR 7	Alexotas
UFR 8	Stockholm

Solutions :

- Regrouper dans une table tous les attributs dépendant de la totalité de la clé.
- Regrouper dans une autre table les attributs dépendant d'une partie de la clé.

5.4 3^{ème} Forme Normale :

Cette forme normale a pour objectif l'élimination des dépendances transitives.

Une relation est dite en dépendance fonctionnelle transitive (ou transitive) si pour la relation $X \rightarrow Z$, il existe un attribut (ou groupe) Y telle que :

$$\begin{aligned}
 X &\rightarrow Y \wedge Y \rightarrow Z \\
 &\Rightarrow \\
 X &\rightarrow Z
 \end{aligned}$$

Une entité est en FN3 si :

- Elle est en 2^{ème} Forme Normale
- Et tout attribut non clé doit dépendre **directement** de la clé et non d'un attribut non clé.

On dit aussi : *Aucun attribut non clé n'est en dépendance fonctionnelle transitive avec la clé.*

Unité	Ville	Pays
UFR 12	Kiel	Allemagne
UFR 16	Copenhague	Danemark
UFR 7	Alexotas	Lituanie
UFR 8	Stockholm	Suède

Solutions :

- Conserver dans la table initiale les attributs dépendant directement de la clé.
- Regrouper dans une table les attributs dépendant transitivement de la clé.

5.5 Forme normale de Boyce-Codd

Une relation est en forme normale de Boyce-Codd (FNBC) si :

- Elle est en FN3
- Et les seules DF existantes dans les relations sont celles de la clé vers les attributs non clés.

La mise en forme normale de Boyce-Codd permet d'éviter des redondances dues à l'existence de dépendances fonctionnelles autres que celle de la clé vers des attributs non-clés.

Autrement dit, les attributs d'un identifiant composé ne doivent pas dépendre d'un autre attribut de l'entité.

Exemple :

R (VILLE, DEPARTEMENT, CODE POSTAL) n'est pas en BCNF

R peut être décomposée en :

R1 (VILLE, CODE POSTAL)

R2 (CODE POSTAL, DEPARTEMENT)

La décomposition sans perte Mais qui ne préserve pas la DF :

VILLE, DEPARTEMENT → CODE.

R1 et R2 sont en BCNF.

6 Les opérateurs relationnels

6.1 Projection

La projection permet d'extraire un sous-ensemble de la table en ne sélectionnant qu'une partie des attributs de l'entité (ou la relation). Cette opération porte sur une seule relation (ou table).

A donc pour effet de réduire la table aux seules colonnes énoncées par la projection.

Elle s'écrit : $T = \sigma_{cond}(R)$

En SQL, elle s'écrit :

```
SELECT nom, salaire
FROM Employe ;
```

Id_emp	Nom	Salaire	Ville
10	Boltzmann	32000	Kiel
8	De Broglie	37000	Kiel
5	Dirac	42000	Copenhague
23	Einstein	41000	Alexotas
14	Kaluza	39000	Stockholm

6.2 Restriction

La restriction permet d'extraire un sous-ensemble de la table par une condition sur un ou plusieurs des attributs de la relation. Elle a pour conséquence d'extraire qu'une partie de ses tuples. Une restriction peut porter sur plusieurs attributs.

Elle s'écrit : $T = \Pi_{attributs}(R)$

En SQL, elle s'écrit :

```
SELECT *
FROM employe
WHERE salaire > 40000 ;
```

Id_emp	Nom	Salaire	Ville
10	Boltzmann	32000	Kiel
8	De Broglie	37000	Kiel
5	Dirac	42000	Copenhague
23	Einstein	41000	Alexotas
14	Kaluza	39000	Stockholm

6.3 Jointure

La jointure $R \bowtie S$ est une opération binaire avec 2 entités qui utilise des attributs de même type et de même sémantique avec ses 2 entités pour créer un tuple unique.

Autrement dit, c'est l'association de 2 tables à l'aide de colonnes comparables.

La jointure peut être considérée comme la combinaison d'un produit cartésien et d'une restriction !

En SQL, on utilisera l'opérateur **JOIN**.

La jointure de ces 2 tables

Unité	Laboratoire	Directeur
UFR 12	Quantique	Planck
UFR 12	Nucléaire	Curie
UFR 16	Optique	Bohr
UFR 7	Atomique	Minkowski
UFR 8	Mathématique	Klein

Unité	Ville
UFR 12	Kiel
UFR 16	Copenhague
UFR 7	Alexotas
UFR 8	Stockholm

Produira :

Unité	Laboratoire	Directeur	Ville
UFR 12	Quantique	Planck	Kiel
UFR 12	Nucléaire	Curie	Kiel
UFR 16	Optique	Bohr	Copenhague
UFR 7	Atomique	Minkowski	Alexotas
UFR 8	Mathématique	Klein	Stockholm

La jointure naturelle est une jointure mettant en jeu des colonnes de même nom et dans le cadre d'une opération d'équijointure (opérateur d'égalité). Inversement on parlera non-équijointure.

Seules les lignes qui vérifient la condition seront affichées, on parle d'**INNER JOIN**.

Les lignes qui seront situées dans l'une ou l'autre table qui ne vérifient pas la condition de jointure seront affichées dans le cas d'une **OUTER JOIN**.

L'auto-jointure, c'est l'opération de joindre une table à elle-même.

La jointure externe à gauche (ou à droite), est une jointure qui renvoie le résultat de la jointure interne ainsi que les lignes sans correspondance de la table de gauche (ou de droite).

6.4 Produit cartésien

Un produit cartésien ($A \times B$) résulte généralement, d'un oubli de la condition de jointure (**CROSS JOIN**).

Un produit génère toutes les combinaisons des lignes de la première table avec les lignes de la deuxième table.

Si la table1 contient n lignes et la table2 contient m lignes, le produit cartésien affichera $n \times m$ lignes !

6.5 La division

La division est une opération rarement utilisée dans les opérations de manipulation de base de données.

La division peut être définie comme l'opération inverse du produit cartésien.

Si $C = A \times B$ alors la division peut s'écrire : $C \div B = A$

La division n'est pas une opération de base, elle peut être réécrite en combinant le produit, la restriction et la différence.

6.6 Opérateurs ensemblistes : Union, Intersection, Différence

L'union et l'intersection sont des opérations commutatives :

$$T1 \cup T2 = T2 \cup T1$$

$$T1 \cap T2 = T2 \cap T1$$

Par contre, la différence n'est pas commutative :

$$T1 - T2 \neq T2 - T1$$



En algèbre relationnelle $A \cap B$ peut aussi s'écrire : $A - (A - B)$

En SQL,

Les opérateurs ensemblistes sont

Opérateurs	Définition
UNION	Lignes issues des 2 requêtes sans les doublons
UNION ALL	Lignes issues des 2 requêtes avec les doublons
INTERSECT	Lignes communes aux 2 requêtes
MINUS	Lignes de la 1 ^{ère} requête réduit de celles qui sont présentes dans la 2 ^{ème} .

Les 2 ensembles doivent être uni-compatibles tant au niveau nombre de colonne et type de colonne.

```
SELECT expr1, expr2, ... FROM table1
OPÉRATEUR ENSEMBLISTE
SELECT expr1, expr2, ... FROM table2 ;
```


7 SQL, les différents langages

Le langage SQL⁸ sera utilisé pour interroger la base de données ou pour effectuer des opérations sur les objets des bases de données relationnelles.

SQL est un langage :

- Non procédural
- Assertionnel ou déclaratif, permet de décrire le résultat attendu sans décrire le moyen de l'obtenir.
- Non conçus pour une logique de traitement (if, for, ...)

Il est généralement divisé en sous-langages composé d'instructions pour effectuer des tâches différentes.

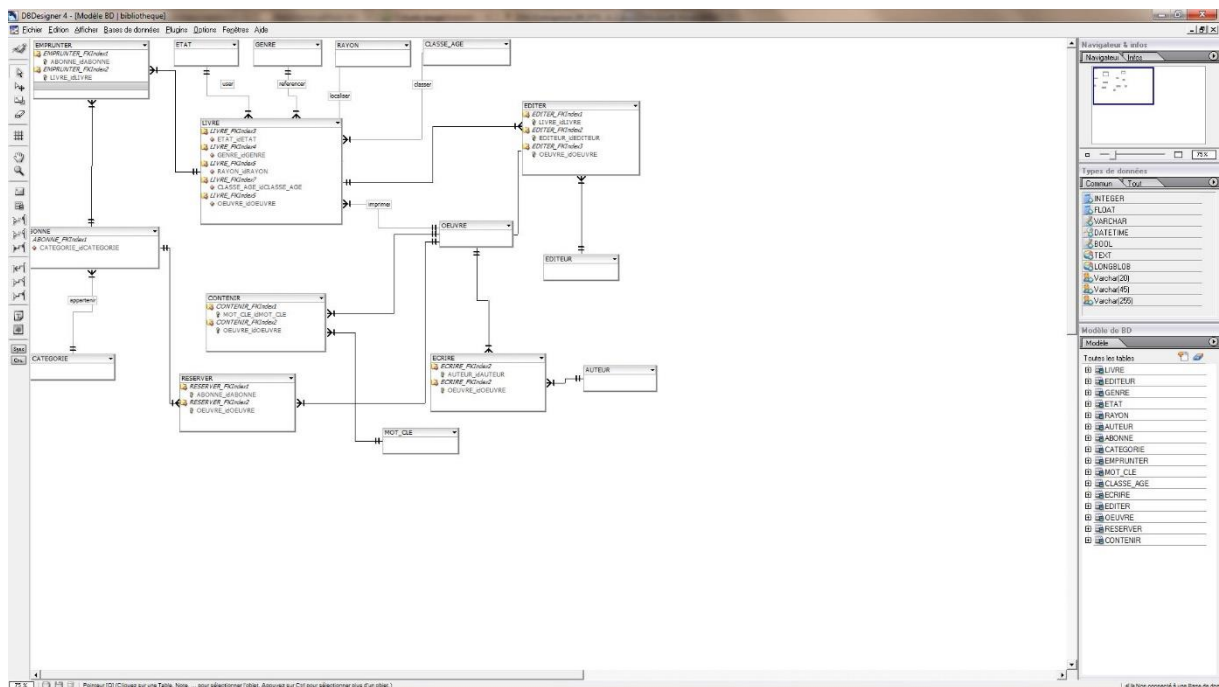
Langage de manipulation de données (DML)	SELECT INSERT UPDATE DELETE MERGE
Langage de définition de données (DDL)	CREATE ALTER DROP RENAME TRUNCATE COMMENT
Langage de contrôle de données (DCL)	GRANT REVOKE
Langage de contrôle des transactions (TCL)	COMMIT ROLLBACK SAVEPOINT

⁸ Donald D. Chamberlin (1944) est le co-inventeur avec Raymond Boyce du langage SQL

8 Les outils de modélisation

Il existe plusieurs outils de modélisation et de gestion de métadonnées. On peut citer, par exemple :

- Jmerise
- DBDesigner logiciel en ligne
- Navicat Data Modeler
- MySQL Workbench
- AnalyseSI
- Looping
- PowerAMC de Sybase®
- Datamodeler d'Oracle®
- ...



Exemple d'écran DBDesigner.

9 EXERCICES

9.1 Gestion des activités d'un centre de loisirs

Un centre de loisirs organise sur l'année, différentes activités (escalade, raquette, cirque, tricot, rock acrobatique, ...).

On s'intéresse à la gestion de la participation des adhérents aux différentes activités organisées par le centre, dans le but d'établir des statistiques sur la pratique des activités, et pouvoir ainsi mieux les planifier et mieux connaître la population qui les pratiquent.

Les informations répertoriées sur les adhérents sont : un identifiant, leur nom, adresse et âge. Pour chaque adhérent, on va enregistrer également la liste des séances d'activités auxquelles il participe. Les activités proposées au catalogue du centre sont référencées par leur nom.

Sur chaque activité, on évalue le coût unitaire d'organisation d'une séance par client, et le prix unitaire de vente à un client de la participation à une séance.

On suppose que, pour une activité donnée, le coût et prix unitaire sont fixés indépendamment des séances.

Pour chaque séance d'activité, on veut pouvoir retrouver ses dates et heures d'organisation, ainsi qu'une note d'appréciation attribuée individuellement par chaque client qui s'y est inscrit.

Construire le MCD du système d'information.

9.2 Gestion d'un collège

Vous êtes choisis pour concevoir la gestion d'un collège. On vous demande de vous intéresser seulement à la partie pédagogie.

Dans ce collège, il existe plusieurs classes par niveau. Chaque classe a son professeur principal.

Les enseignants, habilités à enseigner plusieurs matières, dispensent leurs cours dans une salle qui peut être différente suivant le cours. De plus certaines matières ont besoin de salle spécifique.

Bien sûr, une classe a plusieurs enseignants. Et un enseignant donne cours à plusieurs classes.

Les élèves sont notés plusieurs fois pour chaque matière à différentes dates.

On connaît pour chaque enseignant, leur nom et prénom, leur sexe, leur numéro de téléphone, leur adresse postale et courrielle, leur date de naissance.

Pour chaque élève, on connaît leur nom et prénom, leur date de naissance, le(s) responsable(s) légal (légaux), leur nom et prénom, leur sexe, leur adresse, le lien de parenté, un numéro de téléphone et leur adresse courrielle.

Une salle est définie par sa capacité (nombre de places) et son type d'équipement qui la réserve à des matières particulières.

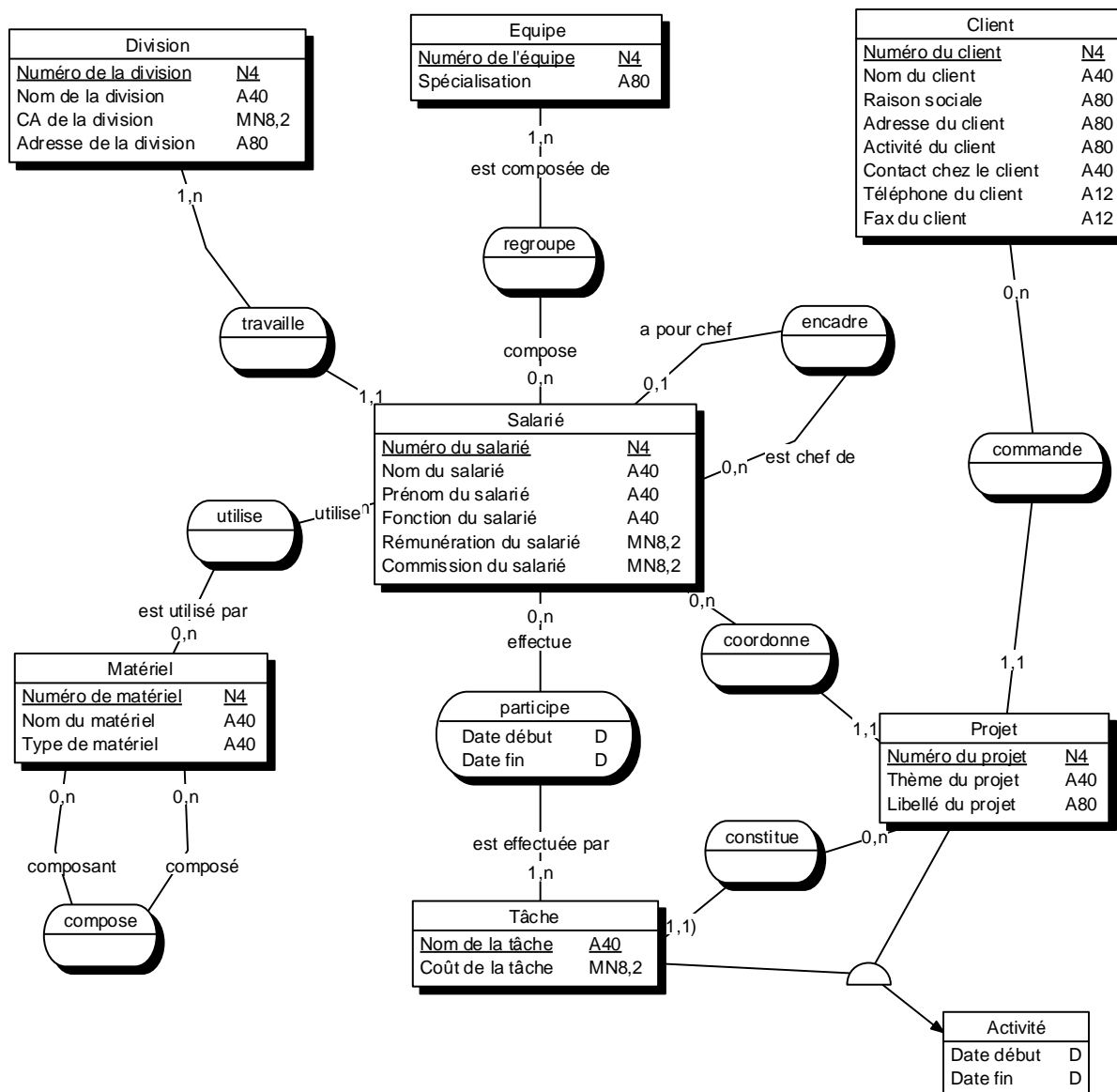
Si certaines règles de gestion vous paraissent imprécises, interviewez de nouveau votre client.

Construisez le dictionnaire de données.

Puis établissez le MCD et le MLD.

9.3 Gestion de ma petite entreprise.

À partir de ce modèle conceptuel, construisez le modèle logique.



9.4 Formes normales

Indiquez si les entités suivantes sont en première deuxième ou troisième forme normale, sinon proposez une solution.

Cas 1

COMMANDE
<u>N° commande</u>
Date
N° client
Nom
N° article
Désignation
Qté commandée

Cas 2

ARTICLE_COMMANDE
<u>N° commande</u>
<u>N° article</u>
Désignation
Qté commandée

Cas 3

COMMANDE
<u>N° commande</u>
Date
N° client
Nom

9.5 Gestion d'une bibliothèque

Le responsable d'une bibliothèque municipale souhaite informatiser et optimiser la gestion de son fond documentaire et de ses abonnés.

Il souhaite mettre en place un système informatique pour ses abonnés afin de les aider dans leurs recherches.

Un abonné pourra donc retrouver un livre dans les différents rayons de la bibliothèque. La recherche pourra s'effectuer sur le titre, un auteur, un éditeur, par genre ou encore par mot-clé.

Un auteur est connu par son nom et prénom éventuel.

Un éditeur est connu par son nom, son adresse et son numéro RCS.

Les livres sont identifiés par un code affecté à l'achat et par un code rayon qui permet de les situer dans la bibliothèque.

Chaque livre est acheté en un ou plusieurs exemplaires dont on connaîtra

- Sa date d'édition
- Sa date d'acquisition.
- Son nombre de pages
- Son format

Tous les exemplaires d'un même livre ont le même code rayon mais un code d'achat différent. On notera le code ISBN-13 des exemplaires. Les différents exemplaires d'un même livre peuvent éventuellement provenir de différents éditeurs. Les livres seront aussi recommandés par classe d'âge.

La bibliothèque veut gérer ses abonnés organisés par numéro de matricule, coordonnées (nom, adresse, téléphone, adresse courrielle) de l'abonné, sa date d'adhésion, sa date de naissance, sa catégorie professionnelle.

La gestion des prêts implique la possibilité de connaître à tout moment la liste des livres détenus par un abonné, et inversement, qu'on puisse retrouver le nom des abonnés détenant un livre non présent dans les rayons.

Les prêts sont accordés pour une durée de quinze jours, éventuellement renouvelable, si aucune demande de ce livre n'a eu lieu entre-temps. Il faudrait donc connaître pour chaque livre emprunté, la date du prêt et la date de retour.

La gestion des prêts nécessite aussi la mémorisation des livres demandés par un abonné. Cet abonné sera prioritaire lors du retour du livre en prêt. Sa priorité est maintenue pendant une semaine, à partir de la date de retour du livre.

Pour suivre de près l'état du stock, la bibliothèque utilise un code indiquant l'état d'usure de chaque livre. Ce code d'usure est éventuellement mis à jour par un bibliothécaire à chaque retour d'un livre en prêt.

La bibliothèque offre aussi la possibilité de faire des suggestions d'achat.

Pour l'instant l'informatisation d'un seul site est prévue.

On pourra s'aider de la grille de recherche suivante :

Recherche avancée

Historique de recherche

Tapez les mots les plus significatifs du titre, de l'auteur, de la collection ou du sujet que vous recherchez dans la zone adéquate, puis cliquez sur : Lancer la recherche.

Vous pouvez cliquer sur la loupe pour effectuer une recherche dans les catégories.

Titre

Auteur

Centre d'intérêt

Matière / Sujet

Collection

Site

*** tous **

Section

*** toutes **

Support

*** tous **

Lancer la recherche

Construire le MCD qui pourra répondre à la demande du responsable de la bibliothèque.

© Didier TEXIER - DTX 2022

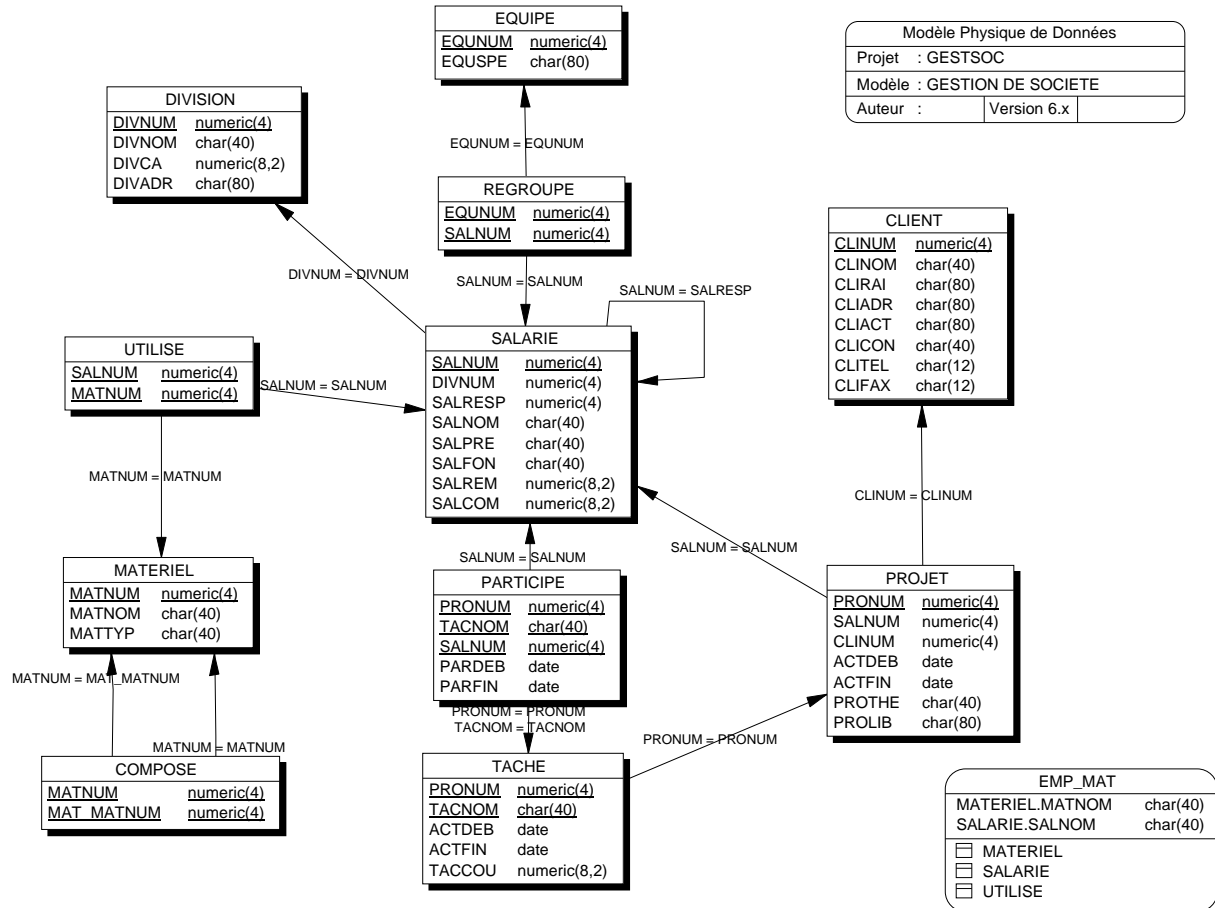
Ce document est la propriété exclusive de son auteur.
Il ne peut pas être reproduit sans autorisation écrite de l'auteur.

Base de données

Page 32
DTX pour DEMOS

10 CORRIGÉS

10.1 Ma petite entreprise



10.2 Formes Normales

COMMANDE
<u>N° commande</u>
Date
N° client
Nom

ARTICLE_COMMANDE
<u>N° commande</u>
<u>N° article</u>
Désignation
Qté commandée

LIGNE_DE_COMMANDE
<u>N° commande</u>
<u>N° article</u>
Qté commandée

ARTICLE
<u>N° article</u>
Désignation

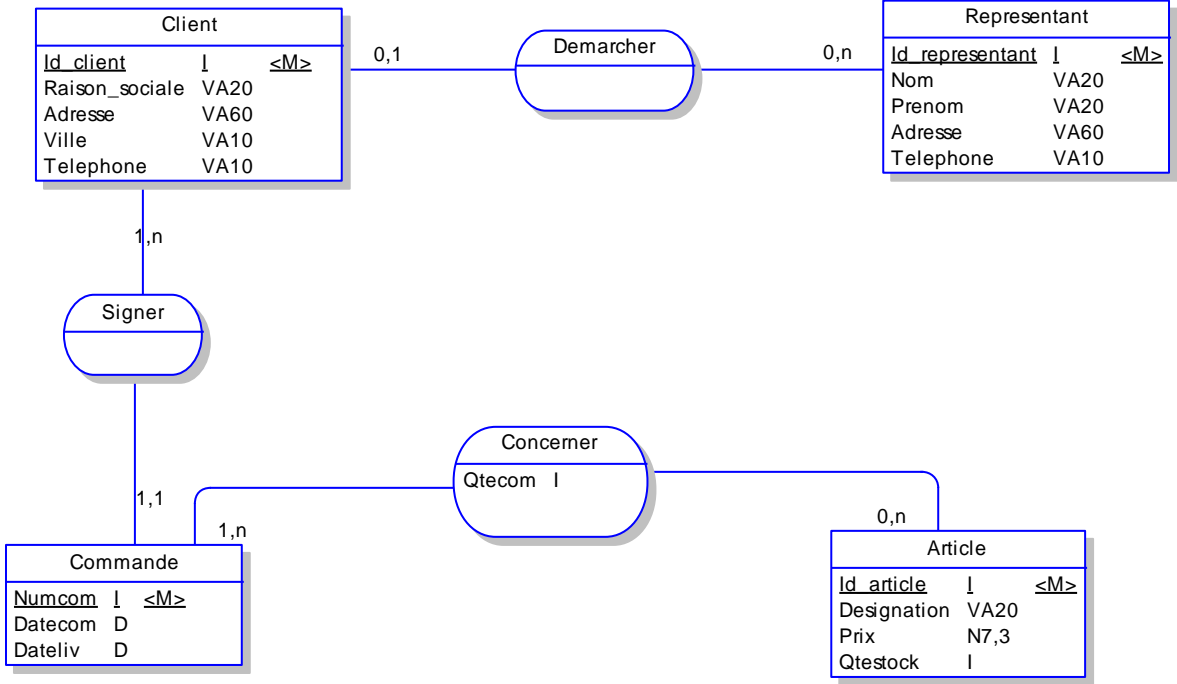
COMMANDE
<u>N° commande</u>
Date
N° Client

CLIENT
<u>N° Client</u>
Nom

10.3 Gestion d'une bibliothèqu

Abonné
Matricule
Nom
Adresse
Téléphone
Date Adhésion
Date Naissance
Catégorie Socio-professionnelle
Livre
Classe âge
Titre
Auteur
Editeur
Genre
Code d'usure
Usure
Exemplaire
Date d'achat
Code achat
Rayon
Code rayon
Genre
Réservation
Date réservation
Prêt
Date début
Date retour
Mot clé
Stock
Priorité
...

11 ANNEXE



12 BIBLIOGRAPHIE

- *Introduction aux bases de données 8^{ème} édition* Chris J. Date. (Vuibert), décembre 2004.
ISBN-13: 978-2711748389
- *Ingénierie des systèmes d'information : Merise ; 2^{ème} génération, 4^{ème} édition* Hubert Tardieu, Dominique Nanci, Bernard Espinasse, Jean-Louis Le Moigne. (Vuibert) , février 2001
ISBN-13: 978-2711786749
- *Comprendre et concevoir les bases de données relationnelles*, Serge Miranda
- *Modélisation des bases de données : UML et les modèles entité-association, 4^{ème} édition*, août 2017 Christian Soutou.
ISBN-13 : 978-2212674873

Acronymes

MCD : Modèle Conceptuel de Données	9
MERISE : <i>Méthode d'Étude et de Réalisation Informatique par les Sous-Ensembles</i>	8
MLD : Modèle Logique de Données	9
MPD : Modèle Physique de Données	9
PK : Primary Key	15
SQL : Structured Query Language	17

13 INDEX

<i>Attribut</i>	<i>Voir Propriété</i>	1ère Forme Normale	19
<i>BD</i>	6	2ème Forme Normale.....	20
<i>CAP GEMINI</i>	6	3ème Forme Normale.....	20
<i>Cardinalité</i>	14	<i>GAMMA</i>	6
<i>CGI</i>	6	<i>Identifiant</i>	13
Chen	6	<i>individu</i>	12
<i>Clé candidate</i>	12	<i>inner join</i>	23
<i>clé étrangère</i>	16	Lemoigne	6
<i>clé primaire</i>	15, 16	<i>ligne</i>	15
<i>CODD</i>	6	<i>MERISE</i>	8
<i>colonne</i>	15	<i>Niveau conceptuel</i>	8
<i>Contrainte</i>		<i>Niveau opérationnel</i>	8
CHECK	18	<i>Niveau organisationnel</i>	8
FOREIGN KEY	18	<i>NULL</i>	17
NOT NULL	17	<i>NUMBER</i>	17
PRIMARY KEY	17	<i>Occurrence</i>	12, 15
UNIQUE.....	17	<i>outer join</i>	23
<i>Contrainte d'intégrité référentielle</i>	16	<i>produit cartésien</i>	23
<i>CROSS JOIN</i>	<i>Voir Produit cartésien</i>	<i>Propriété</i>	12
<i>DCL</i>	25	<i>relations</i>	<i>Voir Table</i>
<i>DDL</i>	25	<i>SEMA</i>	6
<u><i>Degré</i></u>	12	<i>SGBR</i>	<i>Voir Système de Gestion de base de données relationnelle</i>
<i>Dépendance fonctionnelle complète</i>	20	<i>SLIGOS</i>	6
<i>Dépendance fonctionnelle transitive</i>	20	<i>SOGETI</i>	6
<i>DF</i>	<i>Voir Dépendance Fonctionnelle</i>	SQL	17
<i>Dimension</i>	14	<i>STERIA</i>	6
<i>DML</i>	25	<i>table</i>	15
<i>domaine</i>	11	Tardieu	6
<u><i>Domaine</i></u>	12	<i>Tuple</i>	15
<i>Entité</i>	13	<u><i>Type</i></u>	12
<u><i>ESN</i></u>	6	CHAR	17
<i>FN1</i>	<i>Voir Forme Normale</i>	DATE.....	17
<i>FN2</i>	<i>Voir Forme Normale</i>	VARCHAR	17
<i>FN3</i>	<i>Voir Forme Normale</i>		
<i>Formes Normales</i>	19		