

Bases de données

Thierry Lecroq

Thierry.Lecroq@univ-rouen.fr

<http://www-igm.univ-mlv.fr/~lecroq>

PRODUIT	QUANTITE	FOURNISSEUR	ADRESSE
Vis	300	L VIS	Quai des brumes
Plaques	100	ART PLAQUES	Bd des Belges
Boulons	1000	L VIS	Quai des brumes
Pinces	50	QUIN K	Grande rue

- Si on souhaite mettre à jour l'adresse d'un fournisseur, il faut le faire partout ;
- Pour ajouter un fournisseur nouveau, il faut obligatoirement fournir des valeurs de produit et de quantité ;
- La suppression de Plaques ou Pinces fait perdre les informations concernant les fournisseurs de ces produits.

Plan du cours

1. Introduction aux systèmes de gestion de bases de données
2. Le modèle Entité - Association
3. Le modèle relationnel
4. Dépendances fonctionnelles
5. Normalisation

1. Introduction aux systèmes de gestion de bases de données

1.1 Un peu d'histoire

- les SGBD ont vu le jour dans les années 1960 ;
- gérer d'importants volume de données ;
- systèmes propriétaires appartenant à des constructeurs d'ordinateurs (ex : IBM - International Business Machine) ;
- fonctionnent sur des grands systèmes (**main frames**) ;
- schéma d'organisation « hiérarchique » ou « réseau ».

Le modèle hiérarchique

- BD construite selon un modèle en arborescence, avec une racine et plusieurs niveaux de sous-arbres ;
- Chaque élément comporte juste un lien vers le niveau inférieur ;
- Les accès aux données commencent par la racine et descendent l'arborescence jusqu'aux détails recherchés.

Le modèle réseau

- nombreux liens entre les différents éléments de données ;
- accès aux données réalisés par des cheminements divers.

Le modèle relationnel de Codd

- 1970 ;
- Codd chercheur chez IBM ;
- modèle conceptuel reposant sur une représentation unifiée de l'information sous forme de tables ;
- dispose d'un fondement mathématique solide avec l'algèbre relationnel (opérations ensemblistes) ;
- permet une grande indépendance entre les applications, les données et le support physique ;
- propose une démarche cohérente et unifiée pour :
 - la description des données (LDD : Langage de Description des Données) ;
 - l'interrogation des données (LMD : Langage de Manipulation des Données).

Le modèle relationnel de Codd (suite)

- supporte le langage SQL

Les différents SGBD

■ Micro

- dBase (Borland) : gestionnaire de fichiers structurés + langage de programmation
- FoxPro
- Access (Microsoft)
- Paradox (Borland)

Les différents SGBD

- Stations de travail et mini-ordinateurs
 - Oracle
 - Ingres
 - Informix
 - Sybase
 - DB2 (IBM)

- Les SGBD, raison d'être de l'informatique de gestion ont vu leur domaine d'application s'élargir considérablement :
 - Bases de connaissances
 - Systèmes expert
 - SIG (Systèmes d'Information Géographiques)
 - Bioinformatique
 - ...

1.2 Base de données et Système de Gestion de Base de Données

Une base de données (BD) est un ensemble structuré de données enregistrées avec le minimum de redondance pour satisfaire simultanément plusieurs utilisateurs de façon selective en un temps opportun.

L'approche BD correspond à une triple évolution :

- évolution des entreprises (volumes importants de données, centralisées ou réparties) ;
- évolution des technologies (accroissement des performances, intégration des composants, diminution des coûts) ;
- évolution des systèmes d'exploitation et des architectures (extension logicielle du matériel initial, architecture client-serveur et réseaux combinant de façon transparente des machines et des applications hétérogènes).

Un SGBD est centré sur les données contrairement aux systèmes de gestion plus anciens (et dépassés) basés sur les fonctions et les traitements (ex : chaîne de traitement de la paie, chaîne de la facturation, gestion des stocks...).

Approche gestion de fichiers

- fichiers définis pour un ou plusieurs programmes ;
- les données d'un fichier sont directement associées à un programme par une description contenue dans le programme lui-même ;
- aucune indépendance entre programme et données ;
- toute modification de la structure de données nécessite la réécriture des programmes.

Approche base de données

- structuration et description des données sont unifiées et séparées des programmes d'application ;
- la gestion des données (stockage, modification, recherche) qui est étroitement dépendante de leur structuration est fournie par le système de gestion des données ;
- les applications ne communiquent avec les données qu'au travers d'une interface de gestion.

Approche base de données (suite)

- indépendance entre données et applications (peuvent être modifiées indépendamment) ;
- le programmeur d'application (*a fortiori* l'utilisateur final) n'a pas à connaître l'organisation physique des données.

Un SGBD permet de communiquer avec une BD pour

- décrire et organiser les données sur mémoire secondaire (disques) ;
- rechercher, sélectionner et modifier les données.

Un SGBD offre la possibilité de manipuler des représentations abstraites des données indépendamment de leur organisation et de leur implantation sur les supports physiques (mémoires).

On peut considérer un SGBD comme un interpréteur d'un langage de programmation de haut niveau qui permet à l'utilisateur de décrire précisément ce qu'il veut obtenir et non comment l'obtenir : c'est-à-dire formuler une assertion et non décrire une procédure (langage assertiⁿonnel *versus* langage procédural)

Un SGBD assure

- la **description** des données ;
- la **recherche** et la **mise à jour** des données ;
- la **sûreté** des données : vérifier les droits d'accès des utilisateurs, crypter les informations sensibles ;
- la **sécurité** des données : sauvegarder et restaurer les données, limiter les erreurs de saisie, de manipulation ;
- l'**intégrité** des données : définir des règles qui maintiennent l'intégrité des données (contraintes d'intégrité) ;
- la **concurrence** des accès : détecter et traiter les cas où il y a conflit d'accès entre plusieurs utilisateurs.

1.3 Mise en oeuvre d'un SGBD

On distingue 3 niveaux. À chaque niveau correspond un schéma de représentation :

- le niveau interne avec le schéma physique ;
- le niveau conceptuel avec le schéma conceptuel ;
- le niveau externe avec les vues.

Le niveau interne

Le schéma physique spécifie comment les données sont enregistrées sur les mémoires secondaires.

La base physique de données, seule, a une existence matérielle.

Cette base est elle-même perçue à différents niveaux d'abstraction :

- fichier
- enregistrement, article (struct ou record d'un langage de programmation)
- mot machine
- octet
- bit

Le niveau conceptuel

- Le schéma conceptuel décrit en termes abstraits mais fidèles la réalité du domaine d'application.
- Pour une BD d'une compagnie aérienne le niveau conceptuel exprimera la réalité en termes de vol, équipage, passager, horaire...
- Le SGBD fournit un langage de définition des données (LDD) (**DDL - Data Description Language**) qui permet de spécifier le schéma conceptuel.
- C'est un langage de haut niveau qui décrit la BD conceptuelle par référence à un modèle de données.
- Un graphe Entité-Association constitue un exemple de modèle de données

Le niveau externe

- Le schéma externe, encore appelé « vue », permet de décrire la façon dont sont perçues les données par un programme d'application.
- Une vue est une représentation abstraite d'une partie de la BD conceptuelle (ou un sous-schéma du schéma conceptuel).

Le niveau externe (suite)

- Dans l'exemple de la compagnie aérienne, le service de réservation ne s'occupe que des vols et des passagers et n'a pas à avoir accès aux données relatives au personnel et à l'affectation des pilotes sur les différents vols.

Le niveau externe (suite)

- En général, une vue est un sous-ensemble de la base conceptuelle de données.
- Cependant, dans certains cas, une vue peut être plus abstraite que la base conceptuelle des données.
- Par exemple, les données qu'elle utilise peuvent se déduire de la base conceptuelle des données mais ne sont pas présentent dans celle-ci (ex : âge calculé à partir de la date de naissance et de la date du jour).

- Lors de la conception d'une BD on raisonne sur le schéma conceptuel.
- Lors de l'exploitation de la BD on s'intéresse aux données effectivement présentes.
- L'ensemble des données présentes à un instant déterminé est dénommé l'**extension** de la base (**instances**).

1.4 Indépendance physique - indépendance logique

La classification précédente suggère 2 types d'indépendance.

L'indépendance physique

- On doit pouvoir changer le schéma physique sans modifier le schéma conceptuel et sans redéfinir les vues.
- Les programmes d'application n'ont pas à être réécrits quand on change de configuration matérielle ou de version du système d'exploitation.

L'indépendance logique

- On doit pouvoir modifier le schéma conceptuel en ajoutant des informations à certaines entités sans modifier les programmes d'application.

1.5 SGBD et langages

- langages de programmation classiques : déclarations et instructions dans le même langage
- SGBD : 2 langages différents

Dans un SGBD les données existent en permanence et doivent être déclarées une fois pour toute contrairement aux variables des programmes classiques qui disparaissent à l'arrêt du programme.

Le langage de description des données (LDD)

- spécifie le schéma conceptuel
- descriptif des types d'entités, de leurs attributs et domaines et des associations (ou relations) entre ces entités
- utilisé
 - lors de la définition de la BD
 - lors des modifications du schéma conceptuel
 - pour préciser la façon dont les données sont enregistrées et comment y accéder (correspondance entre schéma conceptuel et schéma physique)

Le langage de manipulation des données (LMD) ou langage d'interrogation

- interroger la base
- mettre à jour les données

Généralement un programme d'application (ex : gérer les réservations) est écrit dans un langage de programmation classique, appelé « langage hôte » (C, C++, Cobol, Java, ...) et la communication avec la BD s'effectue avec des instructions du LMD activées à partir du langage hôte.

On conserve ainsi la puissance des langages de programmation pour l'application tout en profitant de la généralité et de la portabilité du LMD.

1.6 Les intervenants

- L'**administrateur** de la BD a la responsabilité de la gestion du système dans son ensemble :
 - définition du schéma original de la base
 - choix des structures de données et des méthodes d'accès au niveau physique
 - modifications éventuelles du schéma et de l'organistaion en fonction de l'évolution de la BD
 - gestion des droits d'accès et des privilèges des utilisateurs
 - spécification des contraintes d'intégrité
 - sauvegardes et restaurations
 - programmation (ou acquisition) d'applications

1.6 Les intervenants (suite)

- Les **utilisateurs occasionnels** :
 - interagissent avec le système sans écrire de programme d'application mais en formulant leurs requêtes avec le LMD
- Les **utilisateurs habituels** :
 - utilisent des programmes d'application prédéfinis

2 Le modèle Entité - Association

- L'informatisation de nombre d'activités nécessite leur modélisation, c'est-à-dire leur expression sous une forme symbolique (le plus souvent mathématique) susceptible d'être représentée en machine.
- Le modèle **Entité - Association** (EA) ou Entité - Relation fournit un outil pour analyser les situations du monde réel (entreprises, institutions, ...).

2.1 Entité

- Une **entité** est un être ou un objet (concret ou abstrait) qui existe et qui peut être distingué d'un autre objet.
- Exemple : une personne (Martin), un véhicule (bicyclette), un concept (emploi), un sentiment (joie) ...
- On regroupe les entités de même nature en ensembles d'entités, par exemple toutes les personnes, tous les véhicules...

Classes d'entités

- Une **classe d'entités** représente de manière abstraite un ensemble d'entités.
- Par exemple les classes d'entités PERSONNE, VEHICULE, SENTIMENT...

Classes d'entités (suite)

Un ensemble d'entités se définit :

- en extension, ex : { féminin, masculin }
- en intention, ex : { x est une personne telle que sexe (x) = féminin }
- par un produit cartésien

A , B et C désignent des ensembles

$A \times B \times C$ désignent l'ensemble des triplets de la forme (a, b, c) avec $a \in A$, $b \in B$ et $c \in C$

Classes d'entités (suite)

Exemple

- Au rayon produits frais d'un supermarché, les denrées peuvent être considérées comme des ensembles d'entités définis par 3 ensembles : **NOM**, **PRIX** et **POIDS**
- (carotte, 0,65, 1) est un tuple (ici un triplet) de l'ensemble d'entités **DENREE**

Attribut, valeur, domaine, clé

- Un **attribut** est une propriété caractéristique des entités d'une même classe.
- Un attribut associe à chaque entité une **valeur** appartenant à un **domaine**.
- Un **domaine** est un ensemble de valeurs acceptables pour l'attribut considéré.

Attribut, valeur, domaine, clé (suite)

Exemples

- Le domaine de l'attribut **PRIX** est l'ensemble des réels positifs.
- Les entités de l'ensemble d'entités **ETUDIANT** ont les attributs **NOM**, **DDN** (date de naissance), **ADRESSE**. Les domaines de valeurs sont ici les chaînes de caractères qui permettent de préciser le nom, la date de naissance et l'adresse.

Attribut, valeur, domaine, clé (suite)

- Une **clé** est un attribut ou un ensemble d'attributs dont les valeurs identifient de manière unique une entité au sein de l'ensemble d'entités.
- Exemple : pour l'ensemble d'entités **ETUDIANT**, **NOM**, **DDN** et **ADRESSE** ne forment pas une clé car 2 étudiants jumeaux ne sont pas distingués.

Notations

- $E = (A_1, A_2, \dots, A_n)$ est le schéma de l'ensemble d'entités E
- les A_i sont les attributs de E ($1 \leq i \leq n$)
- $D(A_i)$ est le domaine de l'attribut A_i
- $e = (a_1, a_2, \dots, a_n)$ est une entité, c'est-à-dire une **instanciation** ou **occurrence** de E avec $a_i \in D(A_i)$ pour $1 \leq i \leq n$
- $K(E) = (A_i, \dots, A_j)$ est la clé de (A_1, A_2, \dots, A_n)
- $(A_i, \dots, A_j) \rightarrow (A_1, A_2, \dots, A_n)$ signifie que (A_i, \dots, A_j) détermine de façon unique (A_1, A_2, \dots, A_n)

- Le choix des attributs, domaines et clés constitue une étape essentielle lors de la définition d'un modèle du monde réel.
- Parmi toutes les clés qui identifient une entité dans un ensemble, on appelle **clé primaire** celle qui est retenue (parmi les **clés candidates**) par le concepteur de la BD pour identifier l'ensemble considéré.

Entité dominante et entité subordonnée

- Si l'existence d'une entité y dépend de l'existence d'une entité x , x est dominante et y est subordonnée.
- Si x doit être éliminé de la BD, y doit l'être aussi.

Entité dominante et entité subordonnée (exemple)

- Une BD BANQUE, les entités CLIENT, EMPLOYE, COMPTE, TRANSACTION avec
- COMPTE définie par les attributs NUMCOMPTE et SOLDE
- TRANSACTION définie par les attributs NUMTRANSACTION, DATE et MONTANT

Un compte peut être concerné par plusieurs transactions.

Une transaction doit être associée à un compte.

Si un compte est supprimé, il faut alors supprimer toutes les transactions qui le concernent (l'inverse n'est pas vrai).

L'entité COMPTE est dominante, l'entité TRANSACTION est subordonnée.

Généralisation et hiérarchie

- Un ensemble d'entités E_1 est un **sous-ensemble** de E_2 si toute occurrence de E_1 est aussi une occurrence de E_2 .
- L'ensemble d'entités E_1 hérite des attributs de E_2 .

Exemple : PILOTE est un sous-ensemble de EMPLOYE dans la BD d'une compagnie aérienne

Généralisation et hiérarchie (suite)

- Un ensemble d'entités E est une **généralisation** de E_1, E_2, \dots, E_n si chaque occurrence de E est une occurrence d'une et une seule entité parmi E_1, E_2, \dots, E_n .
- Les ensembles d'entités E_1, E_2, \dots, E_n sont des spécialisations de l'ensemble d'entités E .

Exemple : l'ensemble VEHICULE est une généralisation des ensembles AUTOMOBILE et CYCLE.

Généralisation et hiérarchie (suite)

- Les ensembles d'entités E_1, E_2, \dots, E_n héritent des attributs de E et possèdent en outre des attributs spécifiques qui expriment leur spécialisation.

Exemple :

VEHICULE(*marque*, *modèle*)

AUTOMOBILE(*marque*, *modèle*,
immatriculation)

CYCLE(*marque*, *modèle*, *type*)

Notation

- On note A « **EST-UN** » B si l'ensemble B est une extension de A ou A est un cas particulier de B .
- Exemple : PILOTE « **EST-UN** » EMPLOYÉ

2.2 Association

- Une **association** d'entités est un regroupement d'entités traduisant une certaine réalité.
- Comme pour les entités, on regroupe les associations de même nature en classes d'associations.

Exemple : entre les entités ETUDIANT et ENSEIGNEMENT on peut considérer la classe d'assoeiation INSCRIT pour traduire le fait qu'un étudiant est inscrit à un enseignement.

Attribut d'une association

- Un **attribut** d'une classe d'association est une propriété qui dépend de toutes les entités intervenant dans l'association.

Exemple : la classe d'association **INSCRIT** définie entre les classes d'entités **ETUDIANT** et **ENSEIGNEMENT** a pour attribut l'année de première inscription de l'étudiant à cet enseignement. Cette année d'inscription est attribut de l'association et non de l'une des entités car il faut connaître l'étudiant et l'enseignement pour la déterminer.

Type d'association

- Le **type d'association** caractérise le nombre de liens autorisés entre entités.
- Il sert à distinguer les associations en fonction du nombre d'entités et de classes d'entités qu'elles mettent en jeu.

Type d'association (suite)

On distingue :

- les associations n -aires qui relient plus de 2 classes d'entités.
- les associations binaires qui relient 2 classes d'entités E et F :
 - de type 1:1 (ou un-à-un) si à une entité de E peut correspondre par l'association A au plus une entité de F et que réciproquement à une entité de F peut correspondre par l'association A au plus une entité de E ;

Type d'association (suite)

- de type $1:n$ (ou un-à-plusieurs) si à une entité de E peut correspondre par l'association A plusieurs entités de F mais à une entité de F ne peut correspondre par l'association A qu'au plus une entité de E ;
- de type $m:n$ (ou plusieurs-à-plusieurs) si à une entité de E peuvent correspondre plusieurs entités de F et réciproquement.

Cardinalité

La **cardinalité** d'un couple EA est définie de la manière suivante:

- Étant données une classe d'entités E et une classe d'association A reliant E à une (ou plusieurs) autre(s) classe(s) d'entités, on définit m (respectivement M) le nombre minimum (respectivement maximum) d'associations de classe A pouvant exister pour une entité donnée de classe E .
- Alors (m, M) est la cardinalité du couple (E, A) .

Cardinalité (suite)

Remarques :

- le minimum peut être 0, 1 ou un entier strictement plus grand que 1 ;
- le maximum M peut valoir 1 ou une valeur $n > 1$. La valeur de n n'est souvent pas précisée de manière numérique faute de connaissance suffisante.

Cardinalité (exemple)

- La relation entre PROPRIETAIRE et VEHICULE est de type $m:n$ car il est légalement possible d'acheter un véhicule à plusieurs (bien que ce soit peu fréquent) et une même personne peut posséder plusieurs véhicules.
- Quant aux cardinalités, elles sont $(0,n)$ et $(1,n)$ car certains véhicules sont abandonnés $(0,n)$ mais qu'il faut posséder au moins 1 véhicule pour être propriétaire $(1,n)$

Contraintes d'identité fonctionnelle (CIF)

- Quand on détermine, entre une association et une entité, une cardinalité présentant les valeurs (0,1) ou (1,1), l'association est particulière.
- On l'appelle **contrainte d'identité fonctionnelle (CIF)**.
- Cette association particulière n'est en général pas nommée.
- Elle indique que l'une des entités est totalement déterminée par la connaissance de l'autre.

Contraintes d'identité fonctionnelle (exemple)

- Si on connaît une commande bien précise alors on connaît un client bien précis.

Contraintes d'identité fonctionnelle (suite)

- Une CIF qui met en relation plus de 2 entités est une CIF multiple.

2.3 Diagramme Entité - Association

Un graphe EA décrit la structure d'ensemble d'une BD en combinant les objets graphiques suivants :

- des rectangles qui représentent les entités ;
- des ellipses (ou écrits en colonnes) pour les attributs ;
- des losanges pour les associations ;
- des arêtes qui relient les attributs (ellipses) aux entités (rectangles) et les entités aux associations (losanges).

Dans les diagrammes les clés sont soulignées.

2.4 Dictionnaire des données

- Le dictionnaire des données liste les entités et leurs attributs en spécifiant le domaine de chacun ainsi que leur catégorie :
- données élémentaires (informations stockées) ;
- données d'informations déduites ou calculées d'utilisation fréquente (ce qui évite de refaire le calcul plusieurs fois) ainsi que les règles de calcul ;
- données calculées de type situation ou historique (ex : total HT des commandes par mois) ;
- paramètres utilisés dans des cas particuliers (ex : TVA).

Il se présente sous forme d'une grille d'analyse.

2.5 Règles de validation

Elles doivent être respectées pour la cohérence du modèle EA :

- chaque entité possède un identifiant ;
- chaque propriété (attribut) d'une occurrence d'une entité ne possède, au plus, qu'une valeur ;
- toutes les propriétés doivent être élémentaires ;
- toutes les propriétés autres que l'identifiant doivent dépendre pleinement et directement de l'identifiant ;
- à chaque occurrence d'une association correspond une et une seule occurrence de chaque entité participant à l'association ;

2.5 Règles de validation (suite)

- pour une occurrence d'une association, il ne doit exister au plus qu'une valeur pour chaque propriété (attribut) de cette association ;
- chaque propriété d'une association doit dépendre pleinement et directement de tout identifiant (clé) et non pas d'une partie seulement de l'identifiant ;
- une cardinalité (0,1) ou (1,1) indique une CIF et réciproquement.

Exemple : modélisation conceptuelle de la publication des articles de journaux

But : conception d'une BD destinée à conserver des descriptions d'articles parus dans des journaux.

Un éditeur édite des journaux. Il est caractérisé par un nom et une adresse. Un journal est édité par un éditeur et publie des articles dans ses numéros. On conservera le nom du journal et le nom de son rédacteur en chef. Un numéro de journal contient une collection d'articles. Chaque article paru dans un numéro est signé par un auteur. On désire conserver le titre et un résumé de l'article ainsi que le nom de son auteur. Les auteurs sont connus par leur nom, leur prénom, leur adresse et leur date de naissance.

Un **éditeur** édite des journaux. Il est caractérisé par un nom et une adresse. Un journal est édité par un éditeur et publie des articles dans ses numéros. On conservera le nom du journal et le nom de son rédacteur en chef. Un numéro de journal contient une collection d'articles. Chaque article paru dans un numéro est signé par un auteur. On désire conserver le titre et un résumé de l'article ainsi que le nom de son auteur. Les auteurs sont connus par leur nom, leur prénom, leur adresse et leur date de naissance.

Un **éditeur** édite des **journaux**. Il est caractérisé par un nom et une adresse. Un journal est édité par un éditeur et publie des articles dans ses numéros. On conservera le nom du journal et le nom de son rédacteur en chef. Un numéro de journal contient une collection d'articles. Chaque article paru dans un numéro est signé par un auteur. On désire conserver le titre et un résumé de l'article ainsi que le nom de son auteur. Les auteurs sont connus par leur nom, leur prénom, leur adresse et leur date de naissance.

Un **éditeur** édite des **journaux**. Il est caractérisé par un **nom** et une **adresse**. Un journal est édité par un éditeur et publie des articles dans ses numéros. On conservera le nom du journal et le nom de son rédacteur en chef. Un numéro de journal contient une collection d'articles. Chaque article paru dans un numéro est signé par un auteur. On désire conserver le titre et un résumé de l'article ainsi que le nom de son auteur. Les auteurs sont connus par leur nom, leur prénom, leur adresse et leur date de naissance.

Un **éditeur** édite des **journaux**. Il est caractérisé par un **nom** et une **adresse**. Un journal est édité par un éditeur et publie des **articles** dans ses numéros. On conservera le nom du journal et le nom de son rédacteur en chef. Un numéro de journal contient une collection d'articles. Chaque article paru dans un numéro est signé par un auteur. On désire conserver le titre et un résumé de l'article ainsi que le nom de son auteur. Les auteurs sont connus par leur nom, leur prénom, leur adresse et leur date de naissance.

Un **éditeur** édite des **journaux**. Il est caractérisé par un **nom** et une **adresse**. Un journal est édité par un éditeur et publie des **articles** dans ses **numéros**. On conservera le nom du journal et le nom de son rédacteur en chef. Un numéro de journal contient une collection d'articles. Chaque article paru dans un numéro est signé par un auteur. On désire conserver le titre et un résumé de l'article ainsi que le nom de son auteur. Les auteurs sont connus par leur nom, leur prénom, leur adresse et leur date de naissance.

Un **éditeur** édite des **journaux**. Il est caractérisé par un **nom** et une **adresse**. Un journal est édité par un éditeur et publie des **articles** dans ses **numéros**. On conservera le **nom du journal** et le **nom** de son **rédacteur en chef**. Un numéro de journal contient une collection d'articles. Chaque article paru dans un numéro est signé par un auteur. On désire conserver le titre et un résumé de l'article ainsi que le nom de son auteur. Les auteurs sont connus par leur nom, leur prénom, leur adresse et leur date de naissance.

Un **éditeur** édite des **journaux**. Il est caractérisé par un **nom** et une **adresse**. Un journal est édité par un éditeur et publie des **articles** dans ses **numéros**. On conservera le **nom du journal** et le **nom** de son **rédacteur en chef**. Un numéro de journal contient une collection d'articles. Chaque article paru dans un numéro est signé par un **auteur**. On désire conserver le titre et un résumé de l'article ainsi que le nom de son auteur. Les auteurs sont connus par leur nom, leur prénom, leur adresse et leur date de naissance.

Un **éditeur** édite des **journaux**. Il est caractérisé par un **nom** et une **adresse**. Un journal est édité par un éditeur et publie des **articles** dans ses **numéros**. On conservera le **nom du journal** et le **nom** de son **rédacteur en chef**. Un numéro de journal contient une collection d'articles. Chaque article paru dans un numéro est signé par un **auteur**. On désire conserver le **titre** et un **résumé** de l'article ainsi que le **nom de son auteur**. Les auteurs sont connus par leur nom, leur prénom, leur adresse et leur date de naissance.

Un **éditeur** édite des **journaux**. Il est caractérisé par un **nom** et une **adresse**. Un journal est édité par un éditeur et publie des **articles** dans ses **numéros**. On conservera le **nom du journal** et le **nom** de son **rédacteur en chef**. Un numéro de journal contient une collection d'articles. Chaque article paru dans un numéro est signé par un **auteur**. On désire conserver le **titre** et un **résumé** de l'article ainsi que le **nom de son auteur**. Les auteurs sont connus par leur **nom**, leur **prénom**, leur **adresse** et leur **date de naissance**.

Un **éditeur** édite des **journaux**. Il est caractérisé par un **nom** et une **adresse**. Un journal est édité par un éditeur et publie des **articles** dans ses **numéros**. On conservera le **nom du journal** et le **nom** de son **rédacteur en chef**. Un numéro de journal contient une collection d'articles. Chaque article paru dans un numéro est signé par un **auteur**. On désire conserver le **titre** et un **résumé** de l'article ainsi que le **nom de son auteur**. Les auteurs sont connus par leur **nom**, leur **prénom**, leur **adresse** et leur **date de naissance**.

Un **éditeur** édite des **journaux**. Il est caractérisé par un **nom** et une **adresse**. Un journal est édité par un éditeur et publie des **articles** dans ses **numéros**. On conservera le **nom du journal** et le **nom** de son **rédacteur en chef**. Un numéro de journal contient une collection d'articles. Chaque article paru dans un numéro est signé par un **auteur**. On désire conserver le **titre** et un **résumé** de l'article ainsi que le **nom de son auteur**. Les auteurs sont connus par leur **nom**, leur **prénom**, leur **adresse** et leur **date de naissance**.

Un **éditeur** édite des **journaux**. Il est caractérisé par un **nom** et une **adresse**. Un journal est édité par un éditeur et publie des **articles** dans ses **numéros**. On conservera le **nom du journal** et le **nom** de son **rédacteur en chef**. Un numéro de journal contient une collection d'articles. Chaque article paru dans un numéro est signé par un **auteur**. On désire conserver le **titre** et un **résumé** de l'article ainsi que le **nom de son auteur**. Les auteurs sont connus par leur **nom**, leur **prénom**, leur **adresse** et leur **date de naissance**.

Un **éditeur** édite des **journaux**. Il est caractérisé par un **nom** et une **adresse**. Un journal est édité par un éditeur et publie des **articles** dans ses **numéros**. On conservera le **nom du journal** et le **nom** de son **rédacteur en chef**. Un numéro de journal contient une collection d'articles. Chaque article paru dans un numéro est signé par un **auteur**. On désire conserver le **titre** et un **résumé** de l'article ainsi que le **nom de son auteur**. Les auteurs sont connus par leur **nom**, leur **prénom**, leur **adresse** et leur **date de naissance**.

3 Le modèle relationnel

- proposé par Codd (IBM San Jose) en 1970 ;

Une **BD relationnelle** est une BD dont le schéma est un ensemble de schémas de **relations** et dont les occurrences sont des **tuples** (ou **n-uplets**) de ces relations.

Les entités et les associations du modèle EA sont représentées exclusivement par des relations (tables).

Une entité est représentée par sa liste d'attributs.

Une association est représentée par la liste des clés des entités qu'elle associe et ses propres attributs.

3 Le modèle relationnel (suite)

- Les objets de la base sont les tuples (lignes) des tables.
- Les objets sont manipulés (LMD) par des opérateurs algébriques relationnels (union, intersection, produit cartésien, sélection, projection, jointure...).

3 Le modèle relationnel (suite)

- Un SGBD est dit **minimalement relationnel** si :
- les informations de la base sont représentées par des tables ;
- il n'y a pas de pointeurs visibles (pour l'utilisateur) sur les tables ;
- le système supporte les opérateurs relationnels :
 - restriction (sélection) ;
 - projection ;
 - jointure.

3 Le modèle relationnel (suite)

- Un SGBD est **complètement relationnel** si de plus :
 - il réalise toutes les opérations de l'algèbre relationnel ;
 - il y a unicité des clés (pas de doublons) ;
 - il assure la contrainte référentielle (ex : pouvoir s'assurer que le produit dont on a passé commande se trouve bien dans la relation PRODUIT).

3.1 Relation

Un **domaine** est un ensemble de valeurs.

Exemples :

- $D_1 = D_2 = \{ \text{chaînes de caractères} \}$
- $D_3 = \{ \text{entiers} \}$

3.1 Relation (suite)

Un **attribut** est une variable prenant ses valeurs dans un domaine.

Exemples :

- attribut A_1 = NOM à valeurs dans D_1
- attribut A_2 = ADR à valeurs dans D_2
- attribut A_3 = NUM à valeurs dans D_3

3.1 Relation (suite)

Une **relation** sur les attributs A_1, A_2, \dots, A_n , de domaines respectifs D_1, D_2, \dots, D_n , est un sous-ensemble du produit cartésien des domaines D_1, D_2, \dots, D_n , soit un ensemble de n-uplets de D_1, D_2, \dots, D_n

Exemple :

$r = \{(MARTIN, ROUEN, 2140), (LEFEBVRE, DIEPPE, 1123), (DUPOND, LE HAVRE, 3424)\}$

Représentation d'une relation

- Chaque tuple (n-uplet) de la relation est écrit dans une ligne d'un tableau dont les noms des colonnes sont les attributs de la relation.
- Chaque tuple est unique.
- Les duplications ne sont pas autorisées.
- L'ordre des tuples est indifférents.

Représentation d'une relation (exemple)

NOM	ADR	NUM
MARTIN	ROUEN	2140
LEFEBVRE	DIEPPE	1123
DUPOND	LE HAVRE	3424

Représentation d'une relation (suite)

Le schéma de la relation r est la liste des attributs de r avec, pour chacun, son domaine, parfois sous-entendu.

Exemple :

Le schéma de r est :

$$R = (\text{NOM} : D1, \text{ADR} : D2, \text{NUM} : D3)$$

écrit en abrégé :

$$R = (\text{NOM}, \text{ADR}, \text{NUM})$$

Représentation d'une relation (suite)

- On dit que r est une relation de schéma R (attention à ne pas confondre r et R).
- Lorsqu'on repère chaque attribut d'une relation par un nom, l'ordre des colonnes est indifférent.

Clé d'une relation

- Une **clé** est une liste ordonnée d'attributs qui caractérise un tuple (n-uplet) de la relation.
- Une **clé primaire** caractérise un tuple de manière unique.

Clé d'une relation (exemple)

- $R = (\text{NOM}, \text{ADR}, \text{NUM})$

NOM	ADR	NUM
MARTIN	ROUEN	2140
LEFEBVRE	DIEPPE	1123
LEFEBVRE	ROUEN	345
DUPOND	LE HAVRE	3424

- $(\text{NOM}, \text{ADR}, \text{NUM})$, (NOM, ADR) et (NUM) sont des clés.
- (NOM, ADR) et (NUM) sont des clés primaires.
- Par contre (NOM) et (ADR) ne sont pas des clés.
- Rq : avec la clé (NOM, ADR) la modélisation ne permet pas des homonymes habitant la même ville.

3.2 Schéma de BD relationnelle

- Un schéma de BD relationnelle B est un ensemble de schémas de relation R_1, R_2, \dots, R_p .
- Une BD b de schéma B est un ensemble de relation r_1, r_2, \dots, r_p de schémas respectifs R_1, R_2, \dots, R_p .

3.3 Passage modèle EA / modèle relationnel

- La modélisation EA des données étant effectuée, il faut implanter la structure obtenue en machine, par exemple sous forme d'un SGBD relationnel.
- Nous allons donc transformer notre structure sous une forme relationnelle.
- On dit aussi que l'on transforme le diagramme EA en schéma relationnel.

Passage modèle EA / modèle relationnel : principes généraux

- Traduction d'une classe d'entités : toute classe d'entités se traduit par une table relationnelle dont les attributs sont :
 - l'identifiant de l'objet, qui forme la clé primaire du tuple correspondant ;
 - tous les attributs de la classe ;

Passage modèle EA / modèle relationnel : principes généraux (suite)

- Traductions des liens $n:1$: tout lien $n:1$ se traduit par un attribut dans la table représentant la classe de départ. Cet attribut représente l'identifiant de l'objet de la classe d'arrivée. Si le lien comporte des attributs, ceux-ci sont placés dans la table de départ.

Passage modèle EA / modèle relationnel : principes généraux (suite)

- Traductions des liens $1:n$: tout lien $1:n$ se traduit par un attribut dans la table représentant la classe d'arrivée. Cet attribut représente l'identifiant de l'objet de la classe de départ. Si le lien comporte des attributs, ceux-ci sont placés dans la table d'arrivée.

Passage modèle EA / modèle relationnel : principes généraux (suite)

- Traductions des liens $m:n$: tout lien $m:n$ se traduit par une table relationnelle, dont les attributs sont :
 - l'identifiant de l'objet de la classe de départ ;
 - l'identifiant de l'objet de la classe d'arrivée ;
 - les attributs (éventuels du lien).

Passage modèle EA / modèle relationnel : principes d'optimisation

- Il est souvent utile de minimiser le nombre de tables d'un schéma.
- C'est pourquoi, après application des principes généraux, on cherche à regrouper des tables.
- Ce sont généralement des tables issues de sous-classes d'une même classe.
- Encore faut-il qu'elles présentent un maximum d'attributs et de liens en commun.

3.4 Les langages de Manipulation de Données (LMD)

- La notion de relation correspond à un certain état de la BD.
- Pour passer d'un état à un autre, on utilise des opérations agissant sur les relations.
- Un LMD se compose d'un ensemble de commandes permettant l'interrogation de la BD et d'un ensemble de commandes permettant de modifier celle-ci (insertion, mise à jour, suppression).
- Le LMD est souvent intégré à un langage de programmation classique appelé langage hôte afin de réaliser des transactions programmées.

3.4 Les langages de Manipulation de Données (LMD) (suite)

- Le langage relationnel a engendré le développement de langages d'interrogation assertionnels qui permettent de définir les données que l'on souhaite visualiser sans dire comment y accéder.

3.4 Les langages de Manipulation de Données (LMD) (suite)

Il y a 3 grandes classes de LMD relationnels :

- les langages algébriques, où l'expression d'un besoin utilisateur se fait à l'aide d'opérations dont les opérandes sont des relations. Issus de l'algèbre relationnelle de Codd, ils consistent en une séquence d'opérateurs sur les relations ; on peut aussi en donner une représentation graphique sous forme d'arbre (ex : **SQL**).

3.4 Les langages de Manipulation de Données (LMD) (suite)

- les langages prédictifs où l'utilisation de prédictats permet de sélectionner l'ensemble des tuples souhaité. On distingue :
 - le calcul relationnel de tuples, issus de la logique des prédictats (ex : QUEL sur système Ingres) ;
 - le calcul relationnel de domaines des relations (ex : QBE).

3.5 Langages basés sur l'algèbre relationnelle

Les opérateurs de base : opérateurs ensemblistes

- union
- différence
- produit cartésien
- jointure

Union

L'**union** de deux relations r et s de même schéma
 R est une relation de même schéma contenant
l'ensemble des tuples appartenant à r ou à s .

Union (exemple)

Soient 2 relations P1 et P2 de schéma

PROD(NP,NOP,QTE,COUL)

P1

NO	NOP	QTE	COUL
100	X	10	R
110	Y	15	B
120	Z	20	R

P2

NO	NOP	QTE	COUL
100	X	10	R
110	Y	15	B
120	Z	20	R
130	W	10	V

NO	NOP	QTE	COUL
100	X	10	R
130	W	10	V

Différence

La **différence** de deux relations r et s de même schéma R est une relation de même schéma contenant l'ensemble des tuples appartenant à r et n'appartenant pas à s .

Différence (exemple)

Soient 2 relations P1 et P2 de schéma

PROD(NP,NOP,QTE,COUL)

P1

NO	NOP	QTE	COUL
100	X	10	R
110	Y	15	B
120	Z	20	R

P2

NO	NOP	QTE	COUL
110	Y	15	B
120	Z	20	R

NO	NOP	QTE	COUL
100	X	10	R
130	W	10	V

Produit cartésien

Le **produit cartésien** de deux relations r et s , de schéma quelconque R et S , est une relation ayant pour attributs la concaténation de ceux de R et de S et dont les tuples sont toutes les concaténations d'un tuple de r à un tuple de s .

Produit cartésien (exemple)

Soient $P5$ de schéma
 $PROD(NP, NOP, COUL)$ et D
de schéma $DEPOT(ND, ADR)$

$P5$

NO	NOP	COUL
100	X	B
120	Y	V

$P6 = P5 \times D$

NO	NOP	COUL	ND	ADR
100	X	B	5	A1
100	X	B	7	W
120	Y	V	5	A1
120	Y	V	7	W

D

ND	ADR
5	A1
7	W

Jointure

- La **jointure** de deux relations r et s , de schéma quelconque R et S , selon une qualification Q est l'ensemble des tuples du produit cartésien $R \times S$ satisfaisant Q .
- On peut donc la programmer comme un produit cartésien suivi d'une sélection.

Jointure (exemple)

Soient P7 de schéma
PROD(NP,NOP,COUL,ADR-F) et
 D2 de schéma **DEPOT(ND,ADR-D)**
 et Q = (ADR-F = ADR-D)

P8 = |x|Q(P7,D2)

P7

NO	NOP	COUL	ADR-F
100	X	B	A1
120	Y	V	A2
150	Z	B	A2

NO	NOP	COUL	ADR-F	ND	ADR-D
100	X	B	A1	7	A1
110	Y	V	A2	12	A2
150	Z	B	A2	12	A2

D2

ND	ADR-D
5	A3
7	A1
12	A2

Opérateur unaire : projection

La **projection** d'une relation r de schéma $R(A_1, A_2, \dots, A_n)$ sur les attributs $A_{i1}, A_{i2}, \dots, A_{ip}$ avec $1 \leq i_j \leq n$ et $p < n$ est une relation r' de schéma $R'(A_{i1}, A_{i2}, \dots, A_{ip})$ dont les tuples sont obtenus par élimination des valeurs des attributs de r n'appartenant pas à r' et par suppression des tuples en double.

Projection (exemple)

- Soit P9 de schéma PROD(NP,NOP,QTE,COUL)

P9

NO	NOP	QTE	COUL
100	X	10	R
110	Y	15	B
140	Z	10	R
160	W	5	V

P10

QTE	COUL
10	R
15	B
5	V

- La projection sur les attributs QTE et COUL donne

Restriction ou sélection

- La **restriction** d'une relation r de schéma R par une qualification Q est une relation r' de même schéma R dont les tuples sont ceux de r satisfaisant Q .
- La sélection (restriction) supprime des lignes dans la table initiale.

Restriction (exemple)

- Soit P9 de schéma $\text{PROD}(\text{NP}, \text{NOP}, \text{QTE}, \text{COUL})$ et $Q = (\text{QTE} = 10)$

P9

NO	NOP	QTE	COUL
100	X	10	R
110	Y	15	B
140	Z	10	R
160	W	5	V

- La restriction donne

$P11 = P9[Q]$

NO	NOP	QTE	COUL
100	X	10	R
140	Z	10	R

Les opérateurs supplémentaires

- Ce sont des opérateurs se déduisant des opérateurs précédents et donc redondants.
- Ils sont utilisés en pratique parce que moins coûteux en temps de calcul que la combinaison d'opérateurs qui les définit.

Cas particuliers

- L'**équi-jointure** de deux relations r et s de schémas respectifs R et S , sur les attributs A_i et B_j est la jointure selon $Q = (A_i = B_j)$.
- La **théta-jointure** de deux relations r et s de schémas respectifs R et S , sur les attributs A_i et B_j est la jointure selon $Q = (Ai \Theta Bj)$ avec $\Theta \in \{ <, >, \leq, \geq, \neq \}$.
- L'**auto-jointure** de deux relations r et s de schéma R , selon l'attribut A_i est la jointure selon $Q = (A_i = A_i)$.

Jointure naturelle

- La **jointure naturelle** de deux relations r et s de schémas respectifs R et S , notée $r \bowtie s$, est l'équi-jointure de r et s sur tous les attributs de même nom dans R et S , suivie de la projection qui élimine les tuples en doublons.

Jointure naturelle (exemple)

Soient P13 de schéma
 $\text{PROD}(\text{NP}, \text{NOP}, \text{COUL}, \text{FAB})$
et P14 de schéma
 $\text{PRIX}(\text{NP}, \text{PU}, \text{FAB})$

P13

NP	NOP	COUL	FAB
100	X	B	FA1
110	Y	V	FA2
150	Z	B	FA2

P13 |x| P14

NP	NOP	COUL	FAB	PU
100	X	B	FA1	25
110	Y	V	FA2	30
120	Z	B	FA2	10

P14

NP	PU	FAB
100	25	FA1
110	30	FA2
120	10	FA2

Intersection

- L'**intersection** de deux relations r et s de même schéma R est une relation t de schéma R contenant les tuples appartenant à la fois à r et s .
- On vérifiera que l'intersection s'exprime aussi à l'aide de la différence :
- $R \cap S = R - (R - S)$

Division

- La **division** de la relation r de schéma $R(A_1, A_2, \dots, A_n)$ par la sous-relation s de schéma $S(A_{p+1}, \dots, A_n)$ est une relation q de schéma $Q(A_1, \dots, A_p)$ formée de tous les tuples, qui concaténés à chacun des tuples de s donne toujours un tuple de r .
- Soit a_i une valeur de A_i
- $q = \{(a_1, a_2, \dots, a_p) \mid (a_{p+1}, \dots, a_n) \in s,$
 $\quad (a_1, \dots, a_p, a_{p+1}, \dots, a_n) \in r\}$
- La division permet de rechercher l'ensemble de tous les sous-tuples satisfaisant une sous-relation de la relation.

Composition d'opérateurs

- Avec les opérateurs, il est possible de composer la plupart des requêtes que l'on peut faire sur une base de données relationnelle.
- Les requêtes s'expriment à l'aide d'une succession d'opérations : union, différence, jointure, restriction et projection.