

---

# Introduction aux Bases de Données



# Objectifs de la formation

## « Introduction aux Bases de Données »

### ► Savoirs à acquérir :

- définir et comprendre ce qu'est une base de données, son utilité au sein d'un système d'information et ses cas d'utilisation
- définir et comprendre ce qu'est un système de gestion de bases de données, ses fonctionnalités et son importance au sein d'un système informatique
- comprendre l'intérêt de la couche conceptuelle, et ses liens avec la couche logique et la couche physique
- connaître les constructeurs et la grammaire du modèle entité-association
- comprendre l'utilité du modèle relationnel et son fonctionnement

### ► Savoir-faire à acquérir :

- savoir structurer une base de données à l'aide du modèle entité-association
- savoir comprendre et expliquer la structure d'une base de données modélisée avec le langage entité-association
- pouvoir transformer une base de données exprimée avec le modèle entité-association vers son schéma relationnel
- savoir utiliser les fonctionnalités de bases de l'outil BD-Main

# Cas d'utilisation

- Gestion administrative et encodage
- Système de réservation utilisant Internet: les "formulaire"
- Comptes
- Paiements (e-Banking)
- E-gouvernement (FedICT): Tax-On-Web

# Autres cas d'utilisation

- ▶ Gestion de stocks, commandes,...
- ▶ Gestion de réservations
- ▶ Enregistrement des achats dans les magasins
- ▶ Enregistrement des données citoyens
- ▶ Sites Web 2.0 (dynamique): Forum, news, RSS,...
- ▶ ...

« Partout tout le temps! »

# Table des matières

- ▶ Chapitre 1: Introduction
  1. Un peu d'histoire
  2. Réponses aux besoins de stocker des données (Fichiers, BD et SGBD)
  3. La notion de donnée
  4. Architecture d'un SGBD
  5. Objectifs et rôles d'une modélisation de la BD
  6. Les différents niveaux de schéma d'une BD
  7. Les contraintes
  8. Les transactions
- ▶ Chapitre 2: Le modèle Entité–Association
- ▶ Chapitre 3: Le schéma Relationnel
- ▶ (Chapitre 4: SQL)



# Chapitre 1: Introduction

## 1. Un peu d'histoire

# Un peu d'histoire

Le **concept de base de données** est né dans les années 60, avec la généralisation des **disques magnétiques** permettant l'accès direct à une donnée, contrairement aux bandes magnétiques, limitées à l'accès séquentiel.

**Premier disque magnétique commercial** : l'IBM 350, équipant le Ramac 305

- **Date** : de 1956 à 1969
- **Capacité** : 5 millions de mots (= 4,4 Mo)
- **Vitesse** : 1.200 rpm. Déplacement moyen : 0,6 sec (temps de commutation inconnu mais sans doute très important)
- **Géométrie** : 50 plateaux de 60 cm mais seulement 2 têtes. 100 cylindres, 100 pistes par cylindre. 5 secteurs par piste. 100 mots par secteur. 7 bits par mot.
- **Volume** : 2 m<sup>3</sup>. Poids : 1.075 kg
- **Nombre d'exemplaires fabriqués** : 100
- **Coût de location** : 35.000 \$/an (en 1956)

# Un peu d'histoire: 60'

## Années 1960

- 1963 : Integrated Data Store (IDS) chez Honeywell-Bull. Développé par Ch. Bachman. Ancêtre des architectures standard de SGBD.
- 1965 : Information Management System (IMS) chez IBM. Au départ conçu pour North American Rockwell pour supporter le projet du 1er vol humain vers la lune. Initialement sur bande magnétique. Sera commercialisé en 1969 sous le nom IMS/360.



- 1968 : premier projet de recommandation CODASYL, inspirée d'IDS.



# Un peu d'histoire: 70'

## ► Années 1970

- 1970 : article de E. F. Codd d'IBM définissant les principes des **bases de données relationnelles** (Codd, E., F., A Relational Model of Data for Large Shared Data Banks, in *Comm. ACM*, Vol. 13, No 6, June 1970).
- 1971 : première recommandation officielle CODASYL. Les rapports de 1973 et 1978 affineront et clarifieront les propositions: ***origine de nombreux SGBD encore en activité aujourd'hui.***
- 1973 : Première définition du langage SEQUEL, qui sera renommé **SQL**.
- 1974 : **premier SGBD** relationnel expérimental, le System/R d'IBM. Donnera naissance aux SGBD SQL/DS et DB2 dans les années 1980.
- 1975 : première version opérationnelle de INGRES, SGBD relationnel expérimental de l'université Berkeley. Langage QUEL concurrent de SQL. Donnera naissance à PostgreSQL. Des transfuges de l'équipe créeront **Sybase**.
- 1979 : première version commerciale d'un SGBD relationnel proposant SQL : **Oracle**.

# Un peu d'histoire: 80'

## ► Années 1980

- 1982 : commercialisation de SQL/DS, SGBD relationnel d'IBM (machines MV).
- 1983 : commercialisation de DB2, SGBD relationnel d'IBM (machines MVS).
- 1987 : après 10 ans de travaux, **premier standard effectif destiné aux *SGBD relationnels***.
- Les SGBD relationnels remplacent progressivement les SGBD traditionnels (CODASYL, IMS).
- Développement de SGBD expérimentaux admettant des valeurs structurées (tables *non plates*). Ces principes seront repris par les SGBD objet et relationnels objet.
- Développement de SGBD déductifs expérimentaux, basés sur les principes de déduction de la logique. Pas de succès commercial.
- Expérimentation sur les bases de données distribuées.

# Un peu d'histoire: 90'

## ► Années 1990

- 1992 : Microsoft commercialise Sybase sous le nom **SQL Server**.
- Standard SQL2 ou SQL 1992. Introduit les notions de **primary key** et *foreign key*.
- Développement de SGBD expérimentaux **orientés objet**. Faible succès commercial mais certains de leurs principes seront repris par les SGBD relationnels objet.
- Microsoft envisage d'intégrer SQL Server comme composant système de Windows. Ce projet ne sera réalisé qu'avec Vista.
- Des versions légères sont proposées pour les systèmes mobiles et embarqués : cartes à puce, PDA, systèmes de capteurs, etc.

# Un peu d'histoire: 00'

## Années 2000

- 2000 : Apparition des SGBD gérant de manière native des documents **XML**. Faible succès commercial mais certains de leurs principes seront repris par les SGBD relationnels modernes.
- Standard SQL3 ou **SQL:1999**. Introduit notamment les requêtes récursives et les structures relationnelles objet.
- **SQL 2003** en préparation. Spécifie notamment les aspects multimedia, XML, modélisation spatiale et analyse de données (fouille de données, entrepôts de données).
- Les SGBD IMS et CODASYL issus des années 1960 sont toujours en activité!

# Et ensuite?

*Trois questions fondamentales :*

**Qu'est-ce qu'une base de données ?**

**Comment construit-on une base de données ?**

**Comment utilise-t-on une base de données ?**



# Chapitre 1: Introduction

2. Réponses aux besoins de stocker des données: Fichiers, BD et SGBD

# Approche historique: Les systèmes de documentation papier

- Farde, papier, classeurs, intercalaires,...

# Approche historique: Les fichiers informatiques

- ▶ Utilisation de **fichiers** = suite de données les unes après les autres
- ▶ Classement séquentiel = seule structure possible
- ▶ Gestion difficile
  - Sécurité
  - Accès concurrents
  - Codage explicite dans chaque application
    - => **Dépendance des données!**
  - Partage de données entre applications
  - Information agrégée difficile à obtenir
  - ...



# Les Bases de données

- ▶ Une **base de données** (BD) contient l'ensemble des données nécessaires au fonctionnement d'une organisation (ou d'une partie de l'organisation)  
Sa gestion est assurée par un logiciel appelé système de gestion de bases de données (SGBD)
- ▶ Une base de données = une collection de données "reliées" (related data)
- ▶ Auto-description du contenu dans une BD
- ▶ Indépendance des données
  - Relation conceptuelle entre les données (pas de pointer!)  
=> Pas de relation directe avec les applications les utilisant
  - Pas de format particulier propre à une application particulière

# Les systèmes de gestion de BD

- ▶ Un **système de gestion de bases de données** (SGBD, en anglais *DBMS*) est un (ensemble de) logiciel(s) fonctionnant sur un système hardware
- ▶ Un SGBD est responsable de mettre en œuvre toutes les manipulations réalisées sur les bases de données, tout en garantissant leur bonne gestion
- ▶ Le composant central est le moteur de base de données
- ▶ Indépendant des applications pour lesquelles le SGBD gère et stocke les données

# Quelques SGBD's

## dBase (1985)



1995



1989

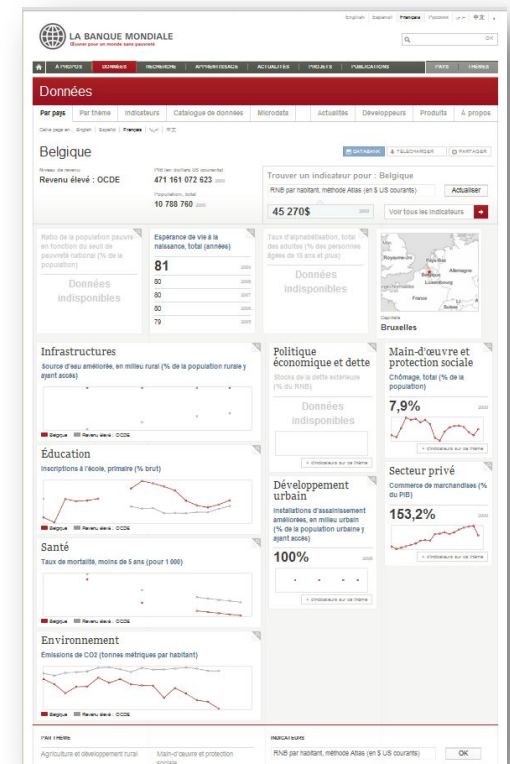
Microsoft®  
SQL Server® 2008

ORACLE®

1979

# La banque de données

- ▶ Une **banque de données** est un ensemble de données, propres à un domaine, que des "producteurs de données" réunissent pour ensuite en commercialiser son usage à un public extérieur
  - Exemple: les banques de données juridiques, économiques, médicales, des brevets, des propriétés des matériaux, ...
- ▶ Concept différent d'une BD, mais devant souvent utilisé les outils de gestion de BD (SGBD)



# Les propriétés d'une BD

## Structuré

- Structure logique et cohérente par rapport au domaine d'application

## Contextualisé

- Contextualisation des données dans un objectif précis (pourquoi créer une DB?)

## Taille adéquate

- Différentes tailles pour différentes utilisations (envergure et contenu)

## Organisation physique

- Différentes possibilités de mise en place de l'hardware et du software

## Persistance des données

- Les données sont stockées sur un long terme

# Les fonctions d'un SGBD

## Définir la structure

- Construire la base de données (structure et contraintes), et définition d'une représentation des données stockées
- Définition et gestion des méta données

## Manipulation

- Insérer, modifier et supprimer les données (intégrité!)
- Requêtes simples et complexes (Reporting)

## Gérer la BD

- Organiser le stockage des données et leur pérennité
- Sauvegarder et restaurer les données
- Performance et optimisation de l'allocation des ressources
- Droit d'accès des utilisateurs

## Monitoring

- Permettre une analyse sur l'utilisation

## Gérer les problèmes

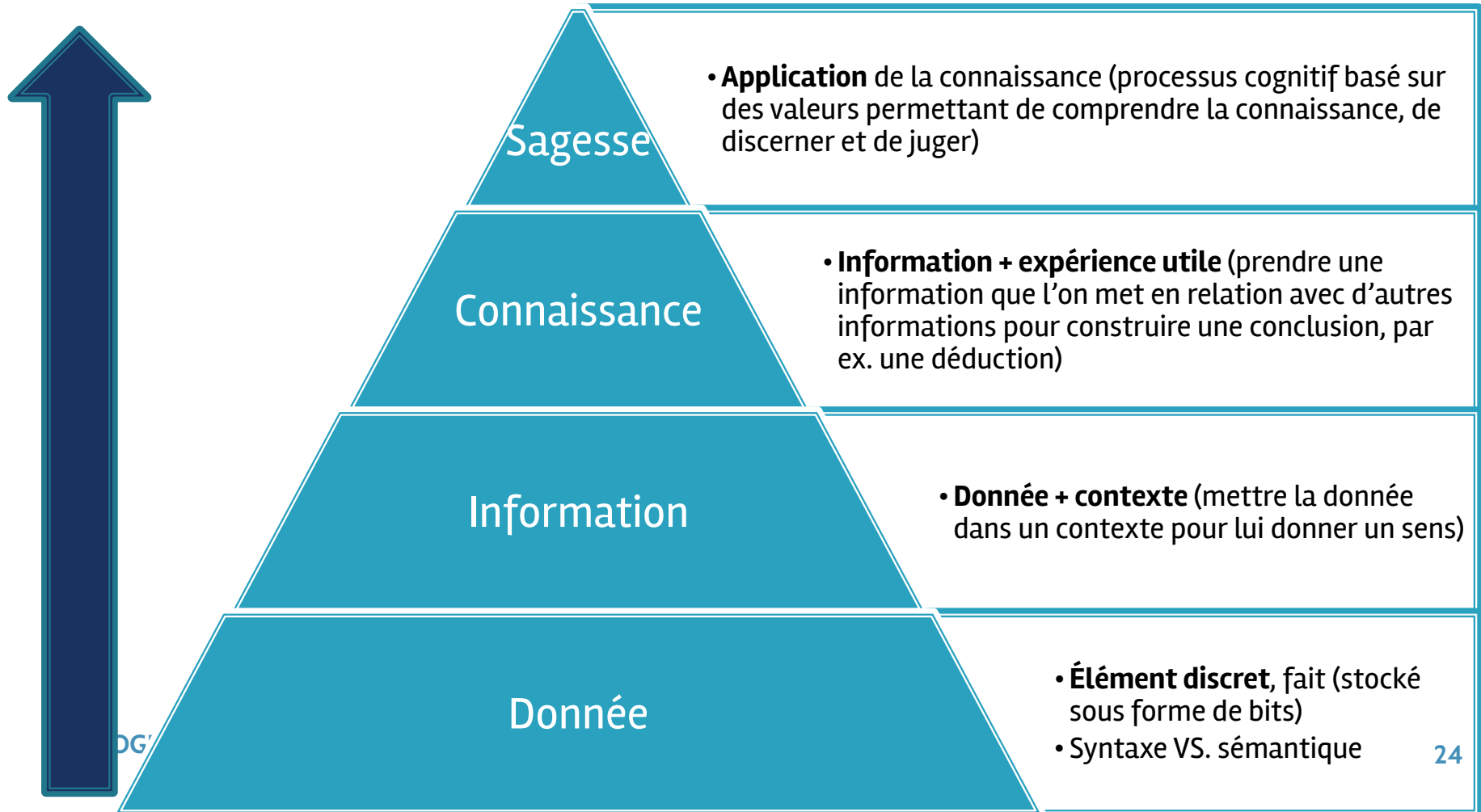
- Sécurité et accès concurrents
- Confidentialité des données



# Chapitre 1: Introduction

## 3. La notion de donnée

# Structure informationnelle





# Donnée persistante VS. donnée transitoire

- ▶ Donnée **persistante**:
  - Stockée dans les bases de données
  - Durable dans le temps
- ▶ Donnée **transitoire**: créée par une application (au départ d'autres données) pendant son fonctionnement, et non nécessaire après son fonctionnement

# Les metadonnées

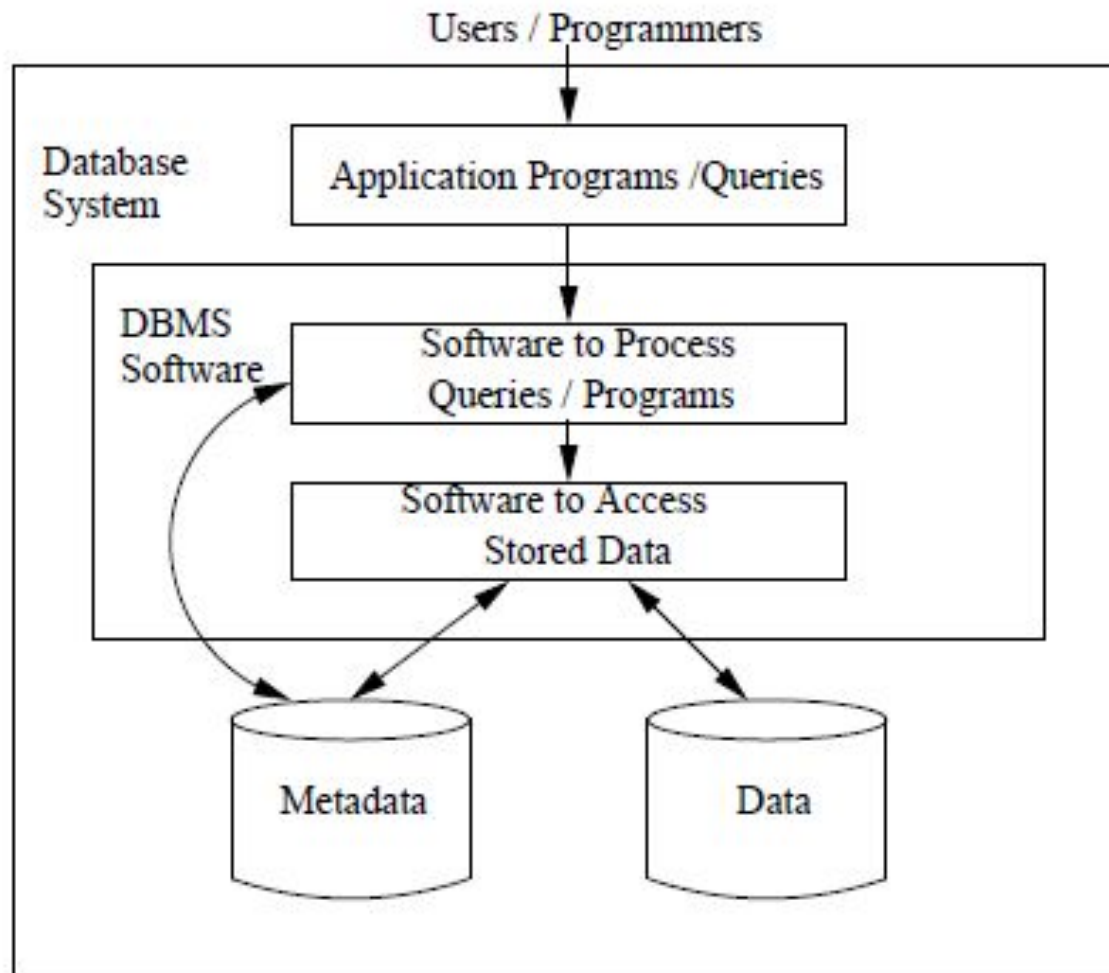
- ▶ Metadonnée: "donnée sur les données"
- ▶ Définition de la structure des données stockées dans la DB
  - Le typage
  - Les identifiants
  - Un lien générique
  - Un attribut générique
- ▶ Stockées par le SGBD (data catalog & data dictionary)
  - une table répertoriant les tables de la base de données
  - une table décrivant les colonnes de ces tables
  - une table décrivant les clés (PK et FK) et une autre décrivant leurs composants
  - une table décrivant les vues
  - une table décrivant les utilisateurs et une autre décrivant les privilèges
  - et *plusieurs dizaines d'autres tables/informations ...*



# Chapitre 1: Introduction

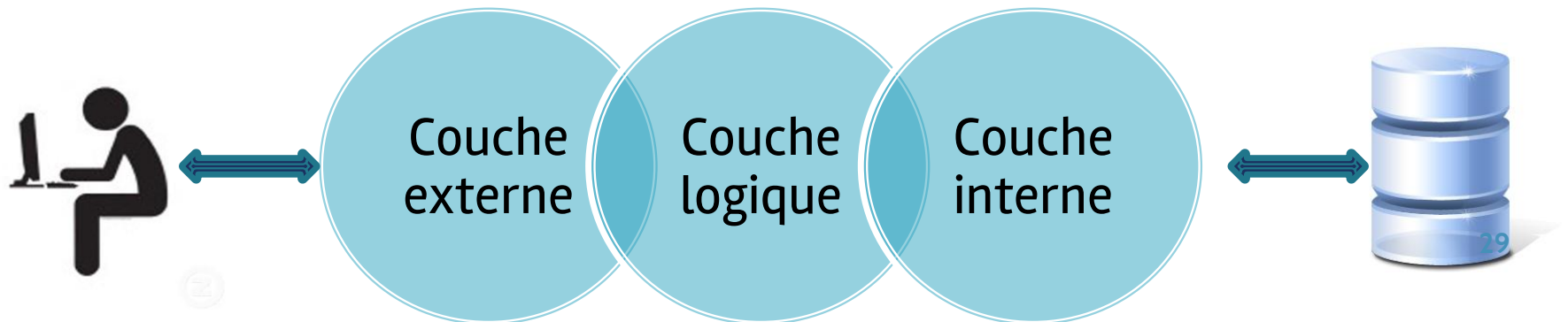
## 4. Architecture d'un SGBD

# Une architecture globale



# Trois couches dans un SGBD

- ▶ Couche **externe**: dialogue avec les utilisateurs, dont les « vues » associées à chacun
- ▶ Couche **interne**
  - Stockage sur des supports physiques
  - Gestion des structures de mémorisation et d'accès
- ▶ Couche **logique**: contrôle des données et liens entre la couche externe et interne (SGBD)

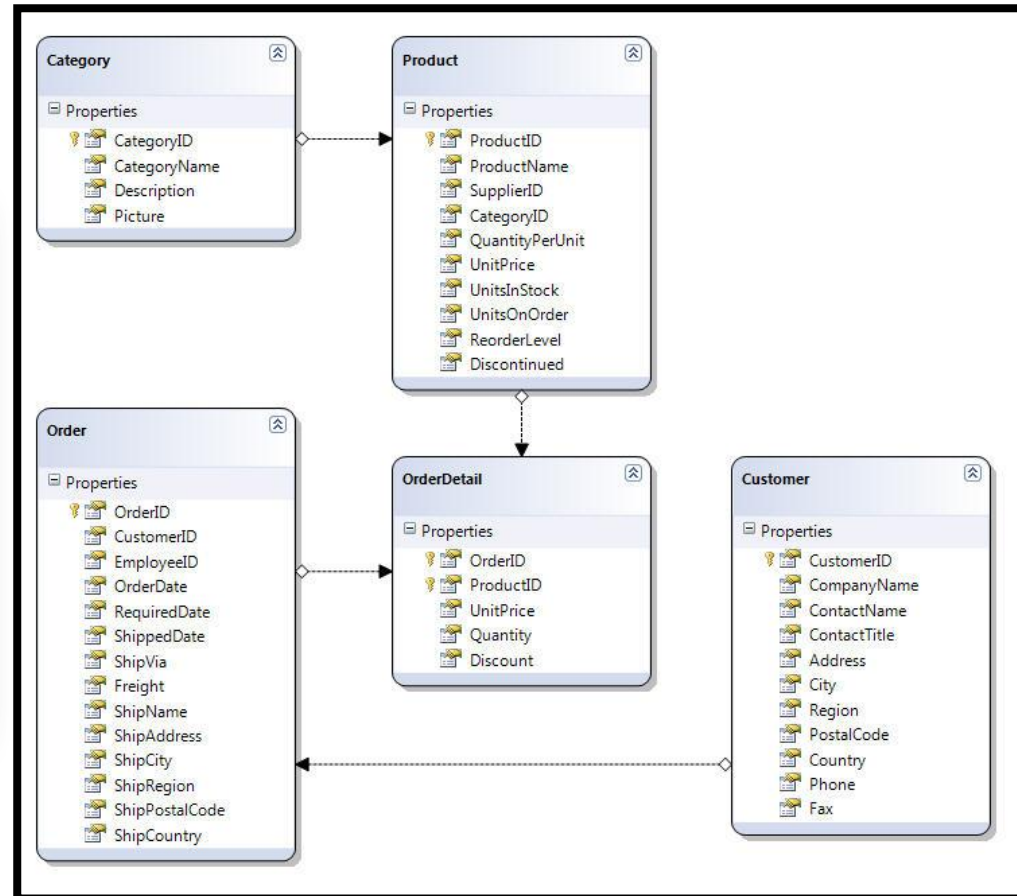




# Chapitre 1: Introduction

5. Objectifs et rôles d'une modélisation de la BD

# Modélisation d'une BD



# Un modèle de données

- ▶ Pour chaque couche (externe – logique – interne), il faut un modèle de données
- ▶ **Modèle de données:** ensemble de concepts permettant de décrire les données d'un point de vue générale, ainsi que les règles d'utilisation de ces données générales
  - Statique: structure des données et règles d'utilisation
  - Dynamique: opérations sur les données
- ▶ Travail nécessaire: *analyse des besoins* (du domaine d'application de la BD)
- ▶ Utile pour "cacher" les détails
  - => Vue abstraite du domaine d'application



# Besoins de description

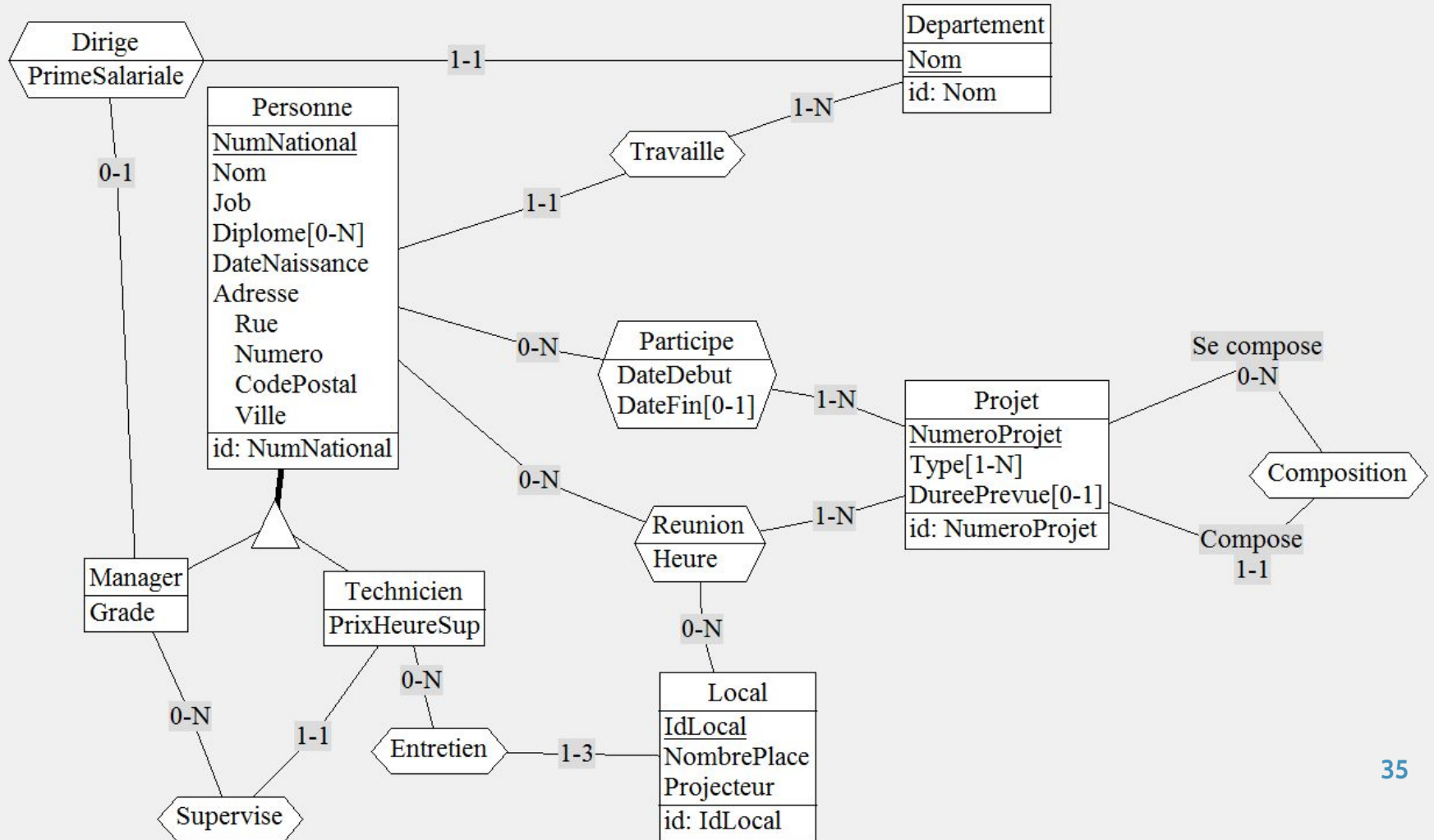
- ▶ **Modélisation conceptuelle** ( $\Leftrightarrow$  couche externe)
  - Décrire les données du monde réel (domaine d'application)
  - Capturer les données, sans référence à une solution informatique particulière
  - Utilisé par les analystes/architectes de DB
- ▶ **Modélisation logique** ( $\Leftrightarrow$  couche logique)
  - Élaborer une description dans un schéma relationnel
  - Réalisé pour le SGBD
- ▶ **Modélisation physique** ( $\Leftrightarrow$  couche interne/physique)
  - Création d'un script pour réaliser la base de données
  - Implémentation (organisation et types de fichiers, index, chemin d'accès,...) réalisé par les administrateurs système

# Les technologies et langages

- ▶ Modélisation conceptuelle
  - Entité – Association (E-A)
  - UML diagramme de classes
- ▶ Modélisation logique
  - Schéma relationnelle
  - Modèle orienté objets
- ▶ Modélisation physique/utilisation
  - SQL
    - Data Definition Language (DDL)
    - Data Manipulation Language (DML)
    - Data Retrieval Language (DRL)

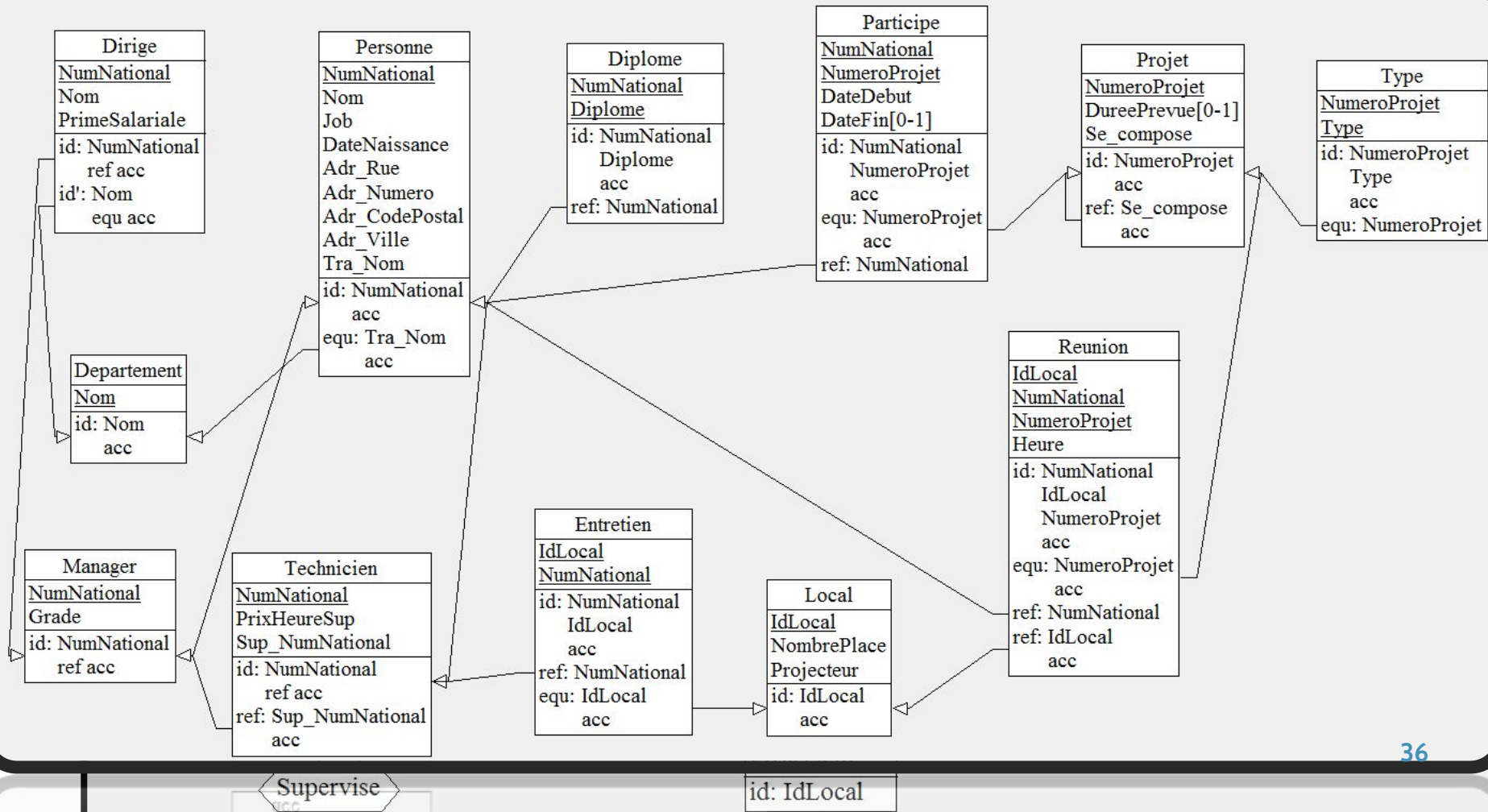
# Exemple de modèle conceptuel

## Entité – Association



# Exemple de modèle logique

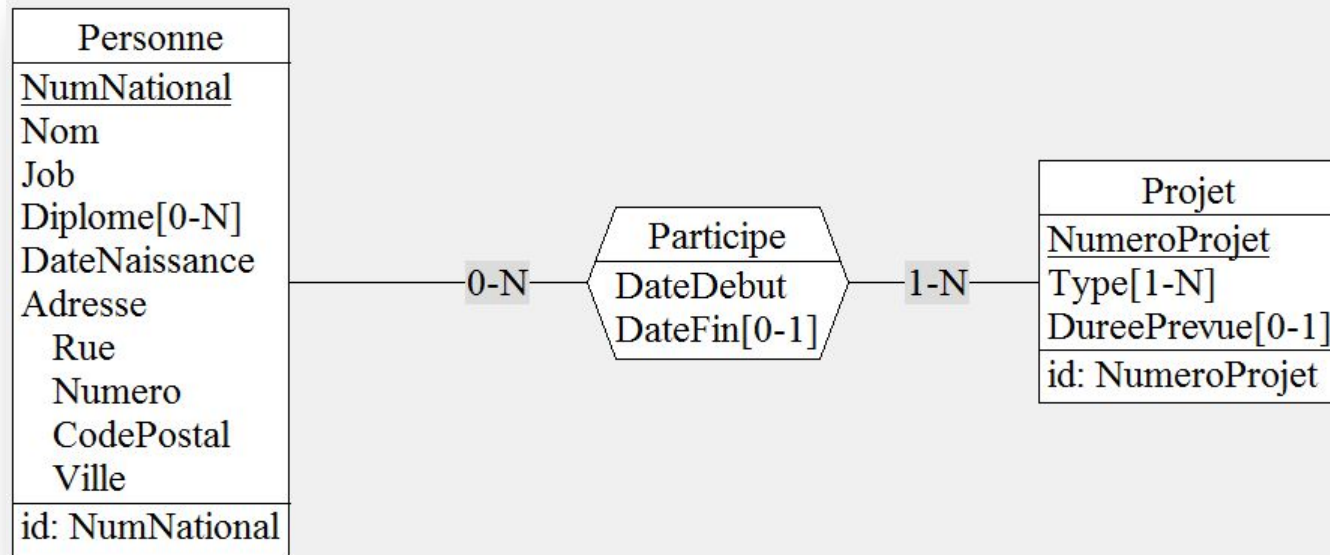
## Schéma Relationnel



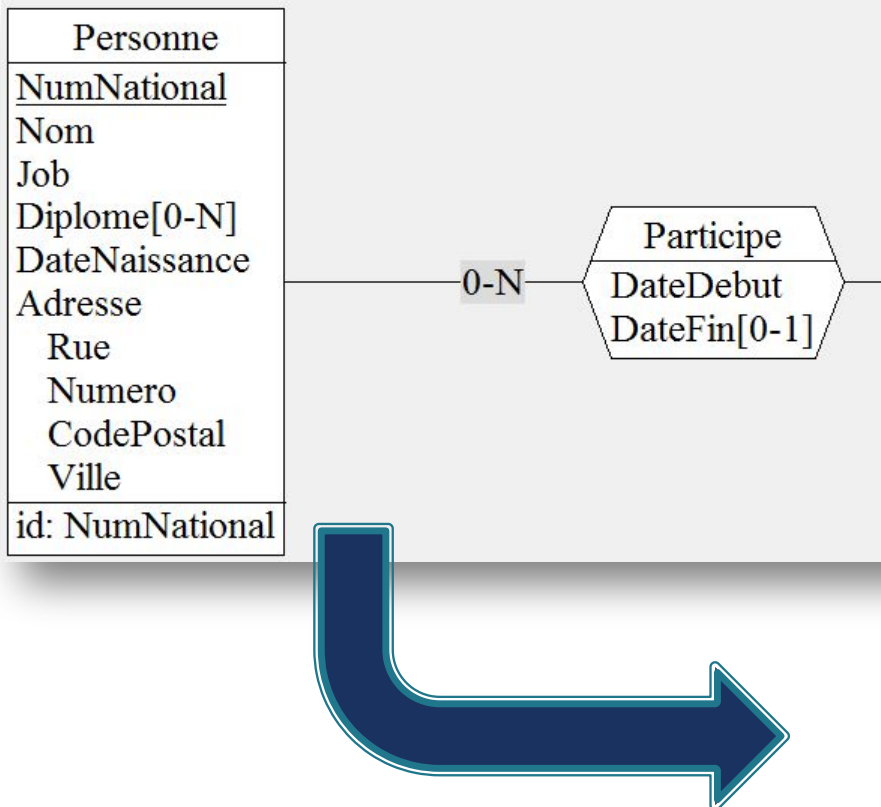
# SQL: un bref coup d'œil

- ▶ SQL (Structured Query Language)
  - Langage de haut niveau dit *déclaratif* (>< langages procéduraux: retrouver des données individuelles et séparément)
  - Spécification de QUOI retrouver (plutôt de comment retrouver des données)
  - Utilisation en tant que tel ou imbriqué dans des langages de programmation (ex: PHP)

# Exemple de modèle physique et utilisation SQL – DDL



# Exemple de modèle physique et utilisation SQL – DDL



```

create table Diplome (
    NumNational int(11) not null,
    Diplome varchar(30) not null,
    constraint ID_Diplome_ID primary key (NumNational, Diplome));

create table Participe (
    NumNational int(11) not null,
    NumeroProjet int(8) not null,
    DateDebut date not null,
    DateFin date,
    constraint ID_Participe_ID primary key (NumNational, NumeroProjet));

create table Personne (
    NumNational int(11) not null,
    Nom varchar(15) not null,
    Job varchar(50) not null,
    DateNaissance date not null,
    Ad_Rue varchar(30) not null,
    Ad_Numero int(6) not null,
    Ad_CodePostal int(4) not null,
    Ad_Ville varchar(30) not null,
    constraint ID_Personne_ID primary key (NumNational));

create table Projet (
    NumeroProjet int(8) not null,
    DureePrevue int(3),
    constraint ID_Projet_ID primary key (NumeroProjet));

create table Type (
    NumeroProjet int(8) not null,
    Type varchar(18) not null,
    constraint ID_Type_ID primary key (NumeroProjet, Type));
  
```

# Exemple de modèle physique et utilisation SQL – DML

Personne
<u>NumNational</u>
Nom
Job
DateNaissance
Ad_Rue
Ad_Numero
Ad_CodePostal
Ad_Ville
id: NumNational
acc

```
INSERT INTO Personne VALUES ('76031231429',  
'Georges', 'Soudeur', '12-03-1976', 'Rue de l'Église', 54,  
5000, 'Namur');
```

```
INSERT INTO Personne VALUES ('81093095829',  
'Lucas', 'Analyste en supply chain', '30-09-1981', 'Rue  
Point Carré', 389, 1020, 'Bruxelles');
```

scc
id: NumNational
Ad_Ville



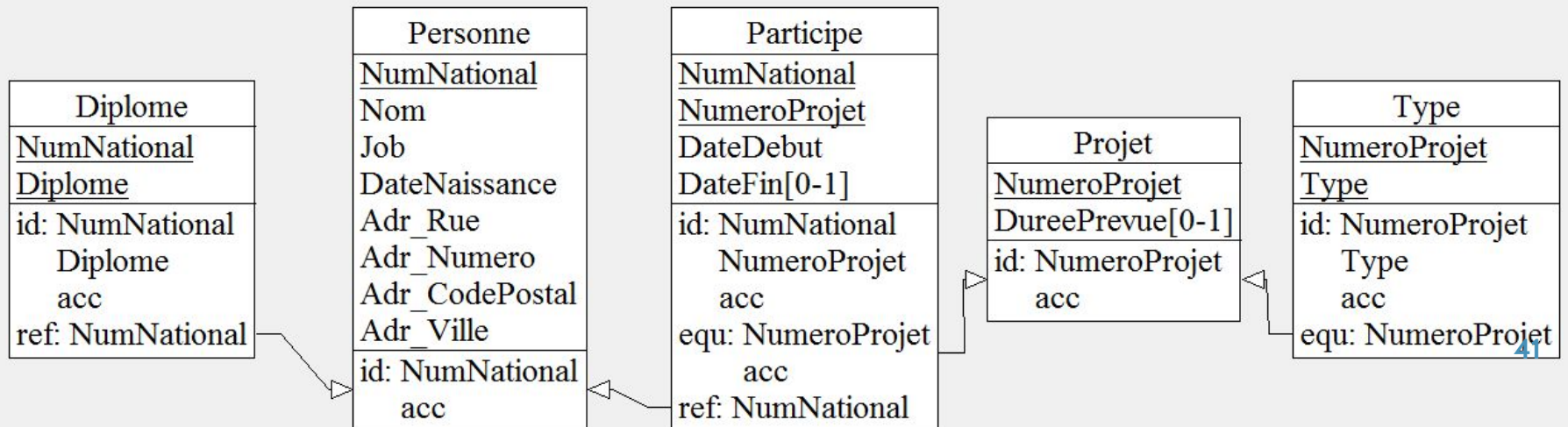
# Exemple de modèle physique et utilisation SQL – DRL

- Sélectionner toutes les personnes ayant dans leur job le terme 'Analyste'

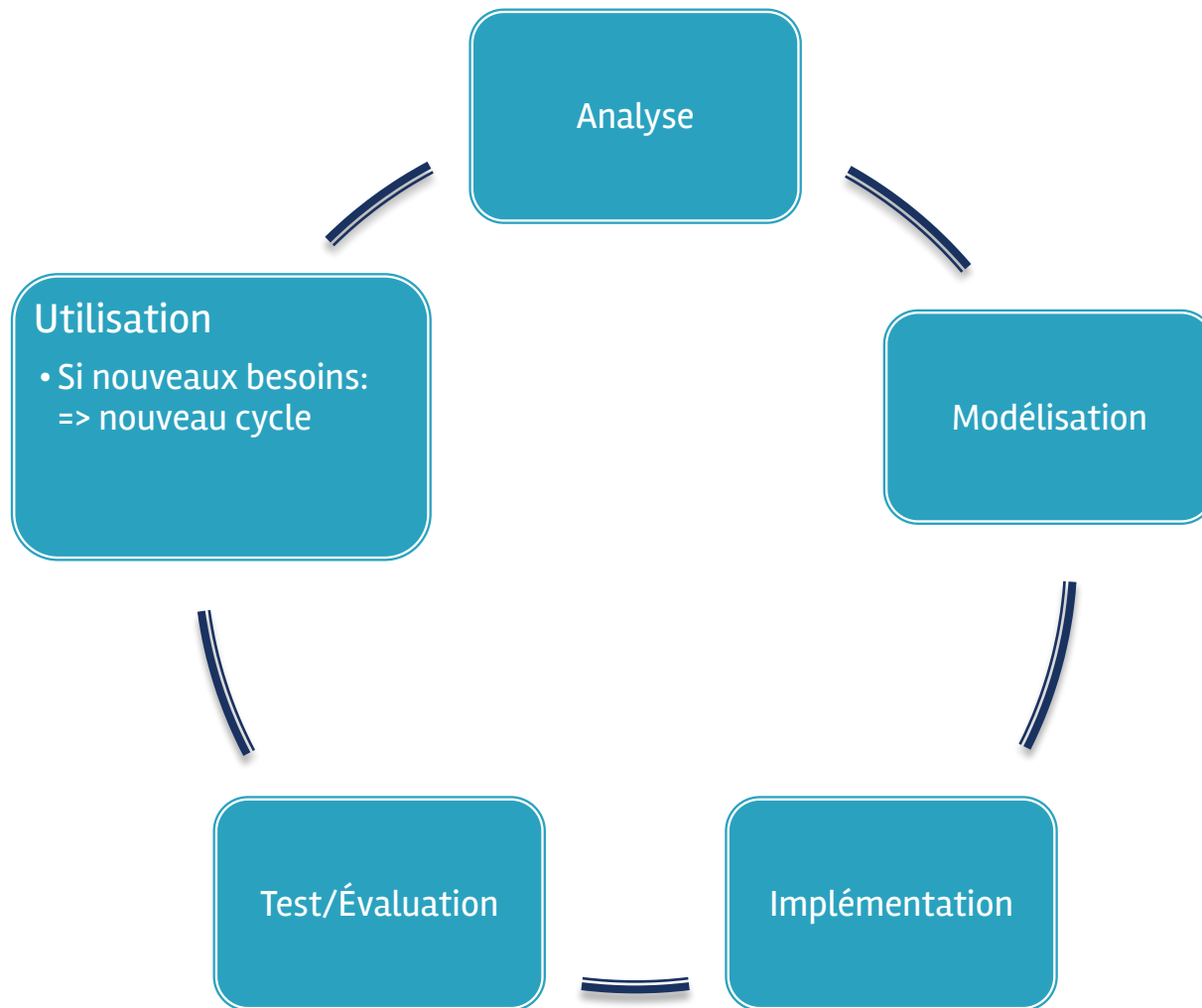
Select NumNational, Nom, job

From Personne

Where Job like '%Analyste%'



# Le cycle de vie des bases de données





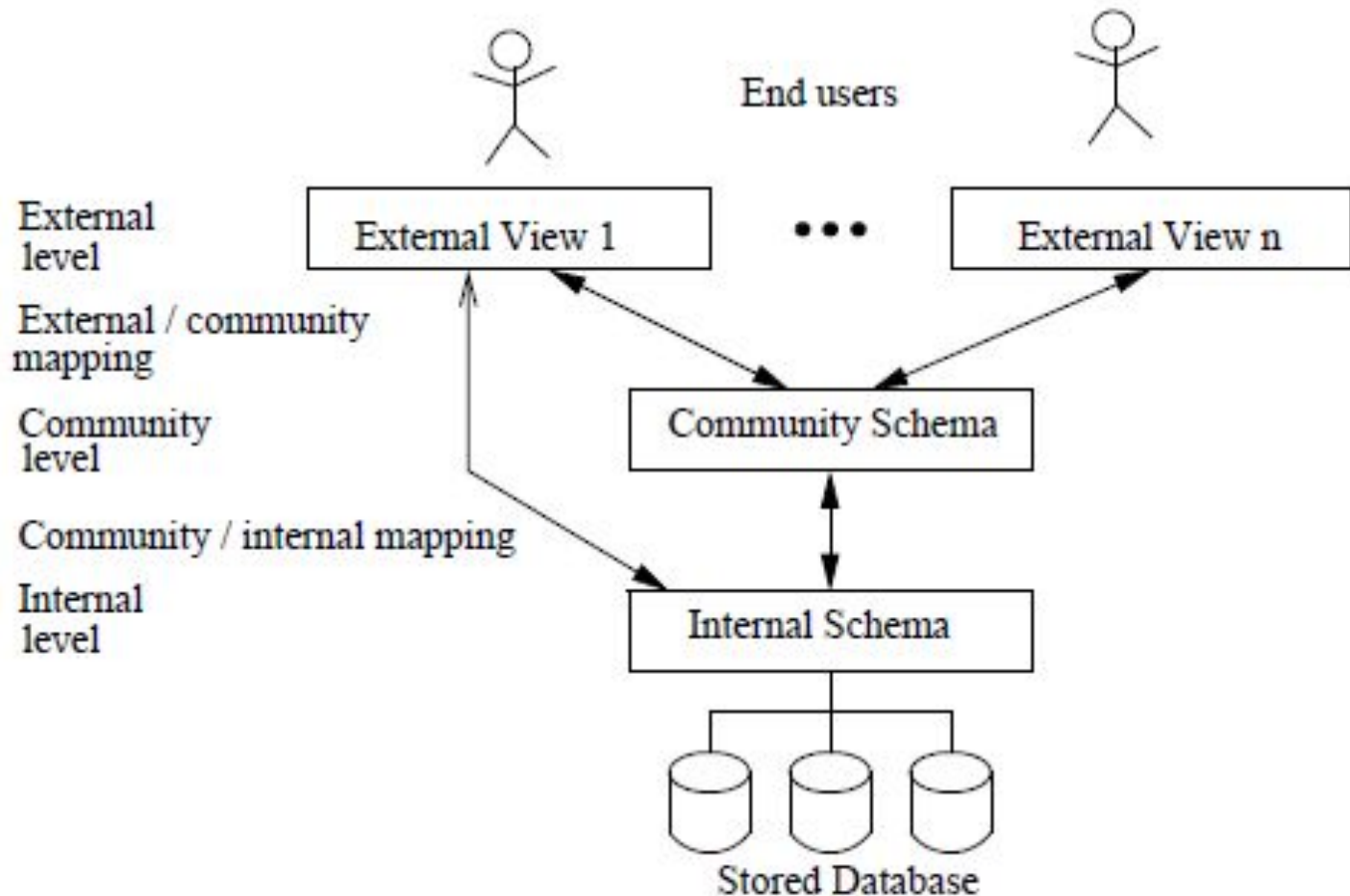
# Chapitre 1: Introduction

6. Les différents niveaux de schéma d'une BD

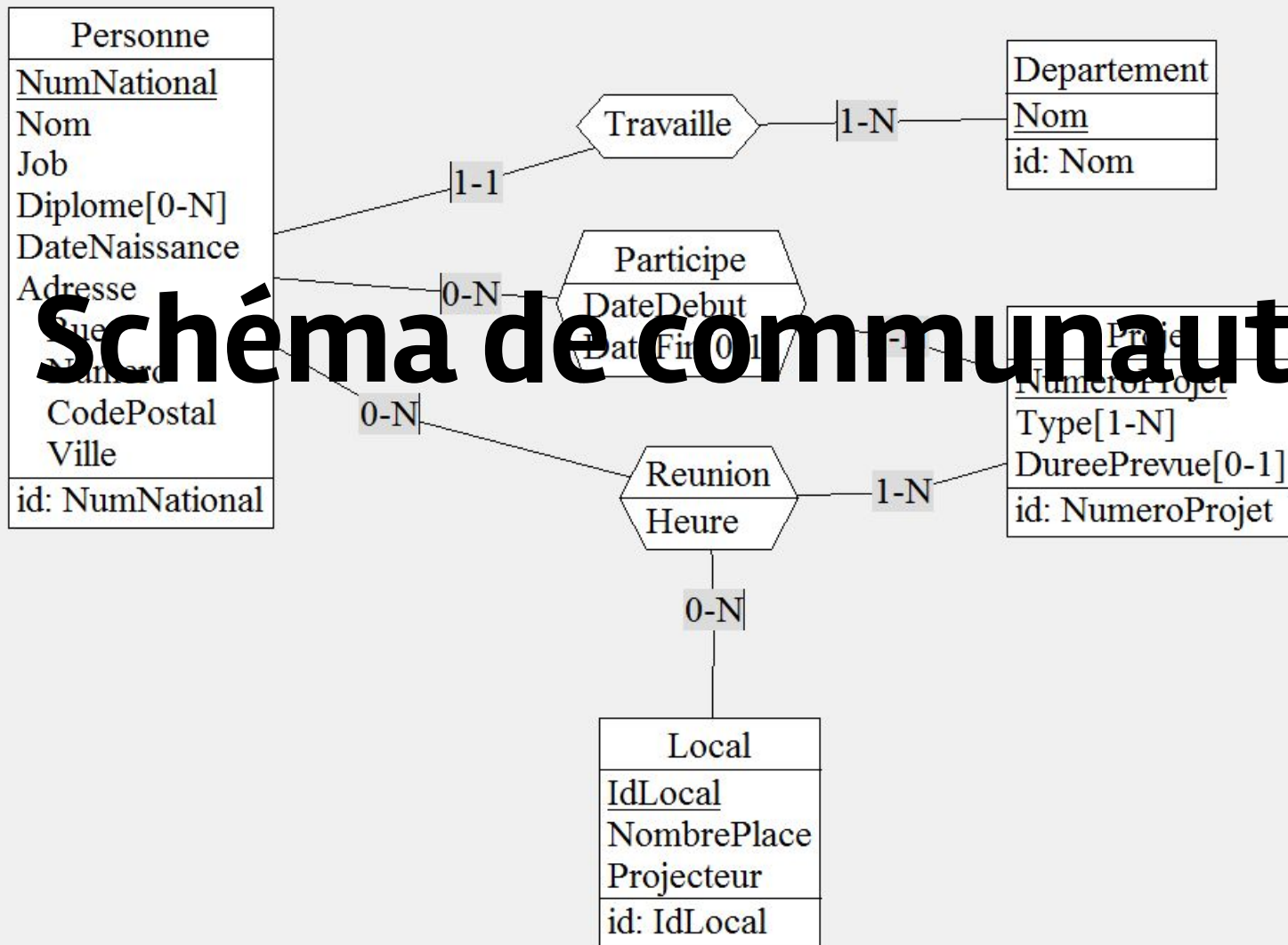
# La notion de vue

- ▶ Une vue est un **schéma externe** réalisé pour un groupe d'utilisateurs déterminé (rôle commun)  
=>différentes perspectives sur le "schéma général" "(appelé schéma de communauté)
- ▶ Une vue utilise une partie du schéma de communauté et/ou le réorganise par rapport aux *tâches* à réaliser
- ▶ Avantages:
  - Simplicité d'utilisation (répond mieux aux besoins des utilisateurs)
  - Sécurisation et protection des données (contrôle d'accès)
- ▶ Redondance!

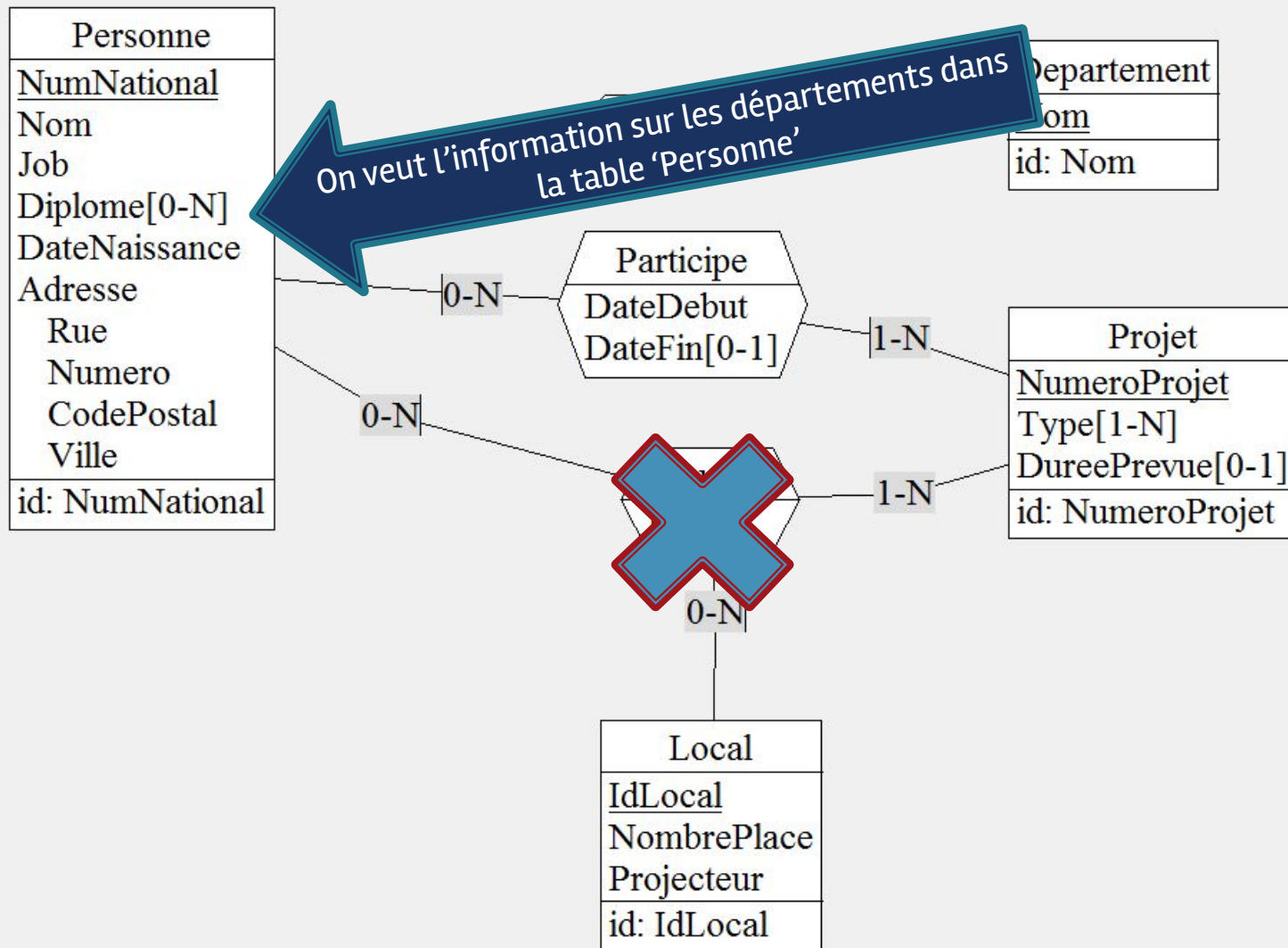
# Les types de schémas



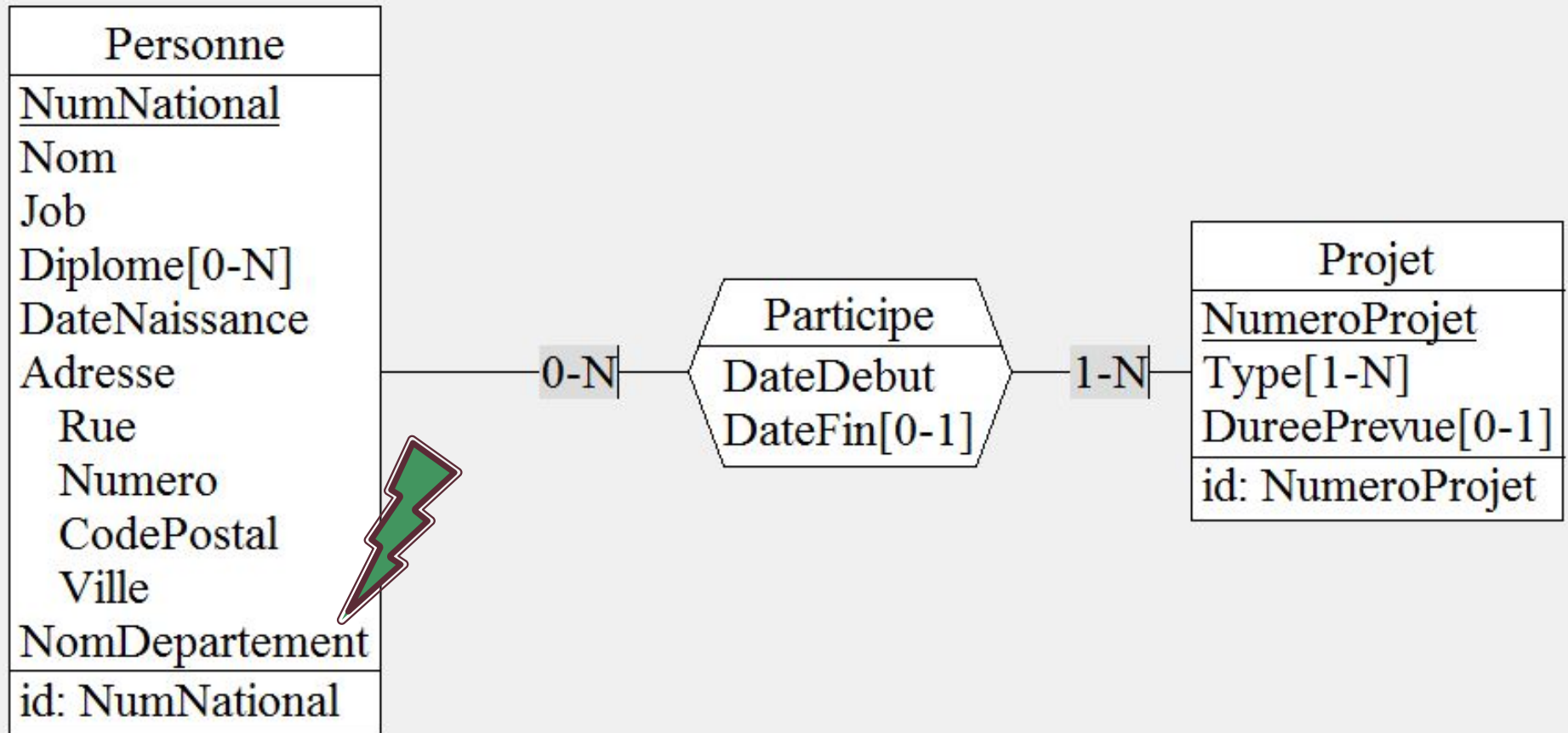
# Exemple de vue/schéma externe



# Exemple de vue/schéma externe



# Exemple de vue/schéma externe







# Chapitre 1: Introduction

## 7. Les contraintes

# Les contraintes

- ▶ Contrainte d'intégrité: toute prescription sur le schéma conceptuel dû à des particularités du domaine d'application
- ▶ Valable pour toute la DB et vérifié lors de toute modification de données
- ▶ Exemples
  - Les points des étudiants sont compris entre 0 et 20
  - Deux étudiants ne peuvent avoir le même numéro d'identification (*contrainte d'intégrité référentielle*)
  - Tout étudiant doit avoir plus de 18 ans
  - Un étudiant doit suivre au moins 10 cours par année académique
- ▶ Tout n'est pas exprimable par contrainte (ex: une faute d'orthographe dans un prénom)
- ▶ Applicable aux données, pas aux applications

# Les contraintes et le SGBD

- ▶ Le respect des contraintes est assuré/contrôlé par le SGBD lors de:
  - l'insertion de données
  - la suppression de données
  - la modification de données
- ▶ Décisions possibles:
  - Refus de l'action
  - Adaptation de l'action pour respecter les contraintes établies
- ▶ Utilisation/contrôle via les métadonnées (stockées et gérées par le SGBD dans le data catalog et data dictionary)



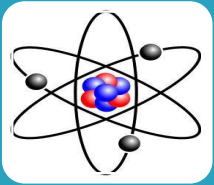
# Chapitre 1: Introduction

## 8. Les transactions

# Les transactions

- ▶ Gestion des accès concurrents aux données car plusieurs utilisations de données en simultanée
  - Lecture & Lecture: OK
  - Lecture & Écriture: ?
  - Écriture & Écriture: KO
- ▶ Assurer la cohérence et la consistance des données
  - **Cohérence**: les données sont en accord avec la réalité
  - **Consistance**: les données sont en concordance les unes avec les autres

# Propriétés ACID



## Atomicité

- La transaction est entière réalisée ou pas du tout (« All or nothing »)
- Si problème => Roll back



## Consistance

- La transaction doit faire passer la base de données d'un état consistant à une autre état consistant



## Isolation/indépendance

- Pas d'influence entre les données de la transaction et celles non incluses
- Pas d'effet partiel visible d'une transaction pendant son exécution



## Durabilité

- Une transaction réalisée avec succès garde ses effets sur les données (une transaction réussie est permanente)

*Pour conclure...*

# Structure conceptuelle d'un SGBD

