

Objectifs

- Savoir ce qu'est une base de données, un SGBD, un SGBDR, comment y sont représentées les données
- Connaître les principaux SGBD du marché
- Etre capable de modéliser un problème en utilisant le formalisme UML
- Savoir ce qu'est le langage SQL et à quoi il sert

Les bases de données

Définitions (1/3)

- Qu'est ce qu'une base de données (BDD) ?

Définitions (2/3)

- BDD = boîte qui contient des données ?
- Problèmes :
 - Un fichier texte est une BDD ?
 - Comment faire si le fichier contient énormément de données (plusieurs To) ?
 - Besoin d'outils pour de recherche, insertion, modification...

Définitions (3/3)

Une **base de données** est un ensemble de données qui sont stockées sur un support informatique, et structurées de manière à pouvoir facilement consulter et modifier leur contenu.

SGBD

- **SGBD** : Système de Gestion de Base de Données

Le SGBD est le logiciel destiné au stockage et à la manipulation de bases de données.

Pourquoi un SGBD ?

- Accès disques fréquents et *lents*
- Actions aboutissant à des incohérences

- SGBD => chaque utilisateur a l'impression d'être le seul à travailler sur le système

Acteurs des SGBD

- Différents profils d'utilisateurs
 - Administrateur
 - Programmeur
 - Utilisateur avec outil informatique
 - Utilisateur sans outil informatique

Fonctionnalités d'un SGBD

- Fonctionnalités attendues :
 - indépendances données / applications
 - consultation et modification des données
 - définition des schémas de données
 - stockage sur un support physique
 - sécurisation des accès
 - résolution des problèmes d'accès multiples
 - reprises sur incident

Notion de transaction

Une **transaction** est une séquence d'actions sur une base de données.

- Etat initial cohérent
- Etat final cohérent
- Contraintes d'intégrité

Propriétés ACID

- 4 propriétés assurant la fiabilité des transactions :
 - Atomicité : une transaction est un ensemble d'instructions indivisibles
 - Cohérence : une transaction ne peut amener à violer une des contraintes de la base
 - Isolation : deux transactions concurrentes ne peuvent interagir entre elles
 - Durabilité : si une panne survient pendant une transaction, l'ensemble de la transaction doit être annulée



Figure 1: Exécutions concurrentes

Ordonnancement des transactions

Utilisation des SGBD

Historique

- 1956 : premier disque dur
- 1964 : apparition du concept de base de données
- 1964 : moteur de base de données réseau IDS de General Electric
- 1966 : moteur de base de données hiérarchique IMS d'IBM
- 1970 : thèse de E. Codd introduisant le modèle relationnel
- 1974 : création du langage SQL
- 1974 : première version de INGRES
- 1975 : modèle entité-association
- 1977 : fondation de la société Oracle

Aujourd'hui

- Multiplication des besoins et des données
 - base de données réparties (1989)
 - base de données orientées objets (1990)
 - NoSQL (1998)

Utilisateurs de SGBD

- Banques
- Sites web marchands
- Réseaux sociaux
- Moteurs de recherche
- Producteurs de données
- ...

Les marché des SGBD



Figure 2: Logos de SGBD du marché

Modélisation des données

Cycle de vie d'un SGBD

1. Conception
 - description du besoin => *modèle conceptuel*
2. Implémentation
 - traduction du modèle dans le SGBD => *modèle logique*
3. Utilisation
 - interrogations, mises à jours

- 4. Maintenance
 - corrections, évolutions

Les différents types de modèles

- Hiérarchique : les données sont classées hiérarchiquement, selon une arborescence descendante. Ce modèle utilise des pointeurs entre les différents enregistrements. Il s'agit du premier modèle de SGBD.

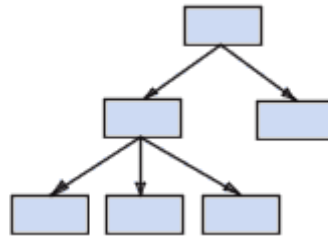


Figure 3: SGBD hiérarchique

- Réseau : comme le modèle hiérarchique ce modèle utilise des pointeurs vers des enregistrements. Toutefois la structure n'est plus forcément arborescente dans le sens descendant.

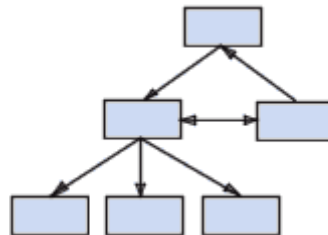


Figure 4: SGBD réseau

- Relationnel : Les données sont enregistrées dans des tableaux à deux dimensions (lignes et colonnes). La manipulation de ces données se fait selon la théorie mathématique des relations.
- Orienté colonne
- Orientée document
- Graphe

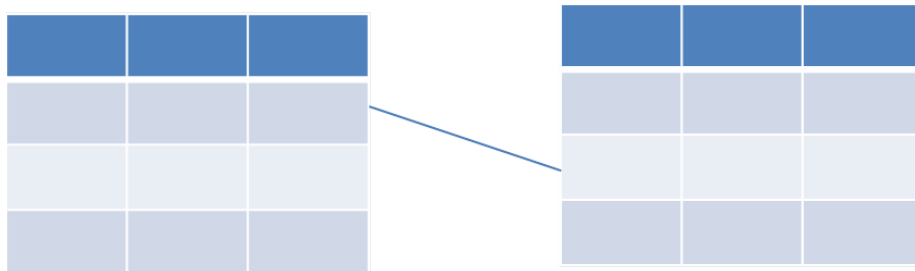


Figure 5: SGBD relationnel

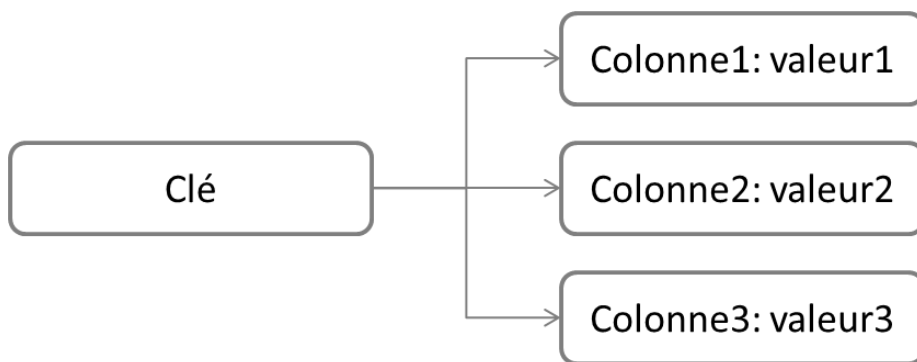


Figure 6: SGBD orienté colonne

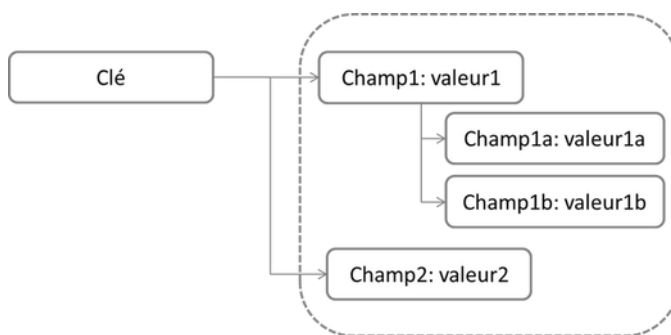


Figure 7: SGBD orienté document

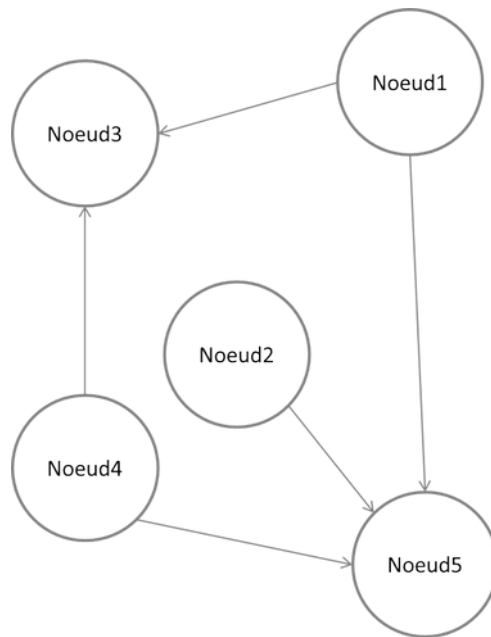


Figure 8: SGBD graphe

Le modèle relationnel

Généralités

- Créé par Codd en 1970
- Repose sur la théorie mathématique des ensembles
- Modèle le plus utilisé actuellement
- **SGBDR** = SGBD Relationnel
- Données de la base de données contenues dans des *relations*

Exemples de SGBDR

Les relations

Une **relation**, ou *table*, est un ensemble de données relatives au même sujet.

- Sorte de contrat sur la forme attendue des données
- Relations représentées sous forme de tables : en-tête + lignes



Figure 9: Logos des principaux SGBD relationnels

Les colonnes

La **colonne** est le constituant de la relation.

- 1 colonne = 1 *nom* + 1 *domaine*
- **Domaine** : contrainte sur le contenu de la colonne
 - type (texte, nombre, etc.)
 - longueur (20 caractères, 32 bits, etc.)
 - règles (non nul, inférieur à 10, etc.)

Exemple

- La relation VILLE contient trois colonnes nommées :
 - NOM
 - CODE_POSTAL
 - POPULATION
- La représentation classique de la relation est :
 - VILLE(NOM, CODE_POSTAL, POPULATION)

Exemple

- Le domaine de la colonne NOM est le suivant :
 - Type : texte
 - Longueur : 100 caractères
 - Les caractères autorisés sont les suivants : 0-9, a-z, A-Z, caractères accentués, espace, “- / ’ &”
 - Valeur nulle non autorisée

- Pas de valeur par défaut

Exemple

- Le domaine de la colonne CODE_POSTAL est le suivant :
 - Type : texte
 - Longueur : 5 caractères
 - Les deux premiers chiffres doivent correspondre à un numéro de département
 - Valeur nulle non autorisée
 - Pas de valeur par défaut

Exemple

- Le domaine de la colonne POPULATION est le suivant :
 - Type : nombre entier
 - Longueur : 32 bits (*valeur max = 4 294 967 295*)
 - Valeur nulle par défaut

Exemple

Table VILLE

NOM	CODE_POSTAL	POPULATION
Paris	75000	2229621
Champs-sur-Marne	77420	24913
Ajaccio	2A004	67507

Les types de données dans PostgreSQL

<http://docs.postgresql.fr/9.2/datatype.html>

- Numériques (smallint, integer, bigint, decimal, numeric, real, double precision, serial)
- Monétaires
- Caractères : varchar, char, text
- Binaire
- Date/heure : date, heure, date-heure, intervalle (avec/sans fuseau horaire)
- Booléen : valeurs possibles TRUE, t, true, y, yes, on, 1 / FALSE, f, false, n, no, off, 0
- Géométriques : point (16 octets), line (32o), lseg (32o), box (32o), path (16+16n o), polygon (40+16n o), circle (24o)

- Adresse réseau : IPv4, IPv6 et MAC. types cidr, inet (7 ou 9 octets) et macaddr (6 octets)
- Types composites : listes de types simples (*peu recommandés*)

Les liens entre relations

- Opérations possibles entre relations :
 - Union
 - Intersection
 - Différence
 - Produit cartésien
 - Sélection
 - Projection
 - Renommage
 - Jointure

L'union

- Réunit dans une même relation les éléments communs des deux relations initiales

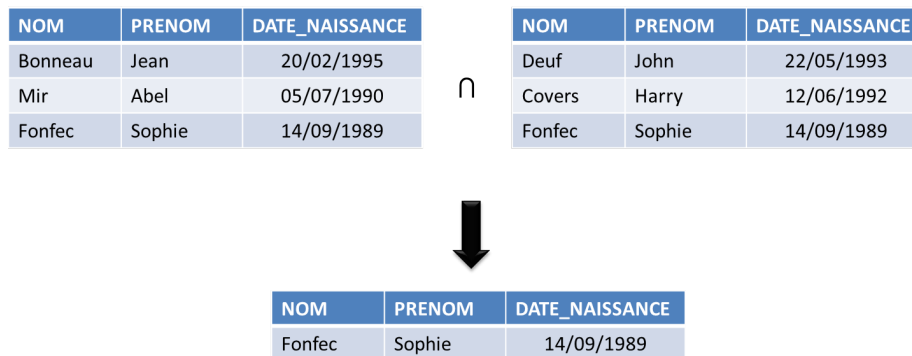


Figure 10: Union de deux relations

L'intersection

- Réunit dans une relation les éléments qui sont dans les deux relations initiales

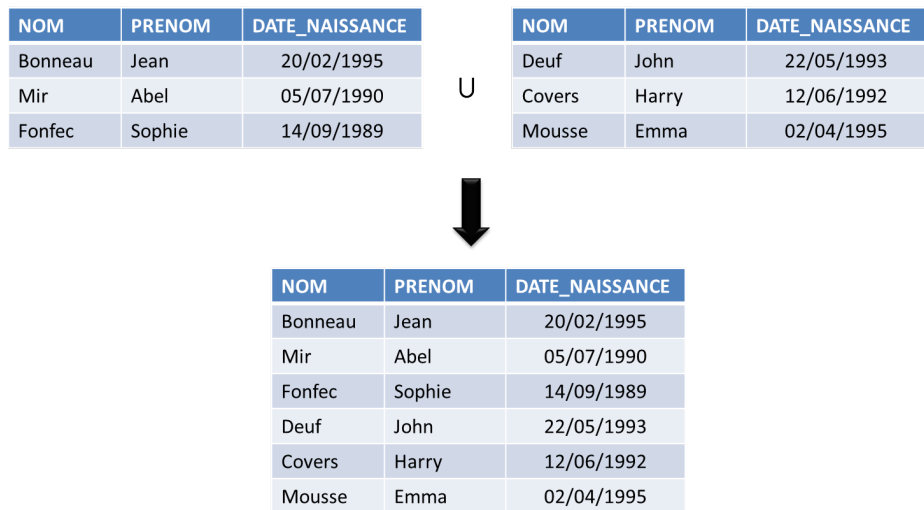


Figure 11: Intersection de deux relations

La différence

- Réunit dans une relation les éléments d'une relation donnée qui ne se trouvent pas dans une autre

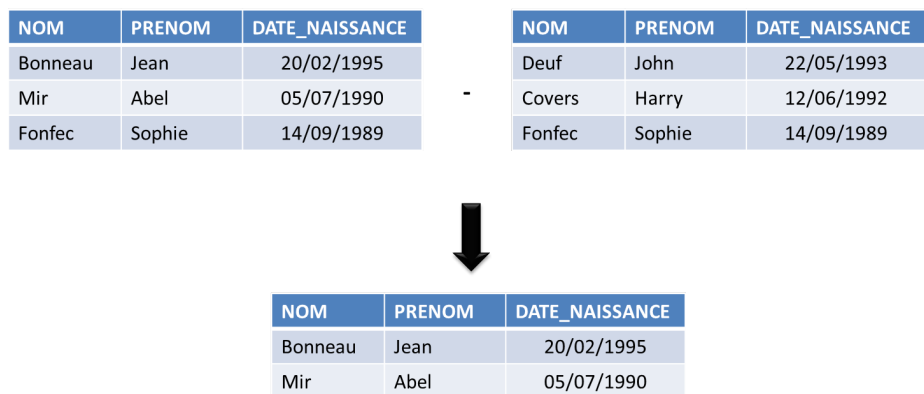


Figure 12: Différence entre deux relations

Le produit cartésien

- Combine dans une relation les éléments de relations en entrée

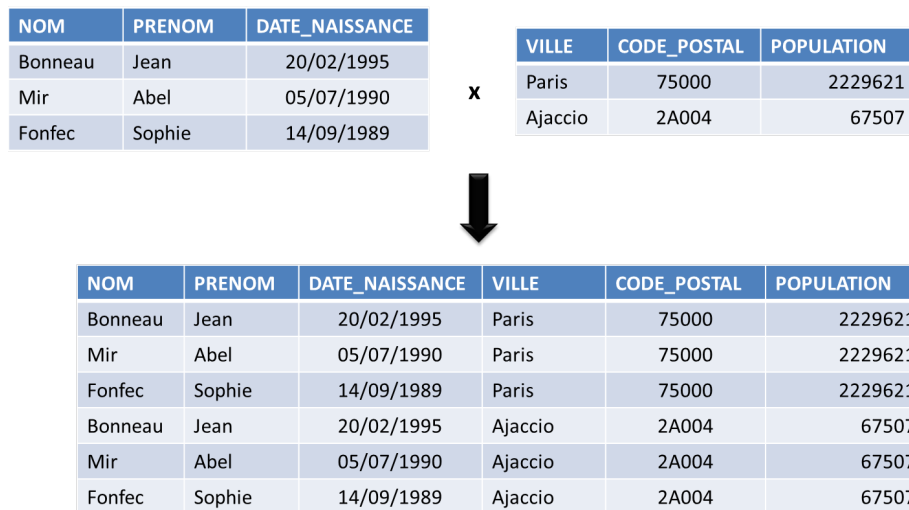


Figure 13: Produit cartésien de deux relations

La sélection

- Sélectionne les éléments d'une relation répondant à une condition

La projection

- Crée une nouvelle relation contenant l'ensemble des lignes d'une relation initiale, mais en ne conservant que certaines colonnes

Le renommage

- Renomme une colonne dans une relation

La jointure

- Composition des éléments de deux relations répondant à un critère de jointure

Clés et liens entre relations (1/2)

Une **clé primaire** est une colonne, ou un groupe de colonnes, permettant d'identifier de manière unique une ligne.

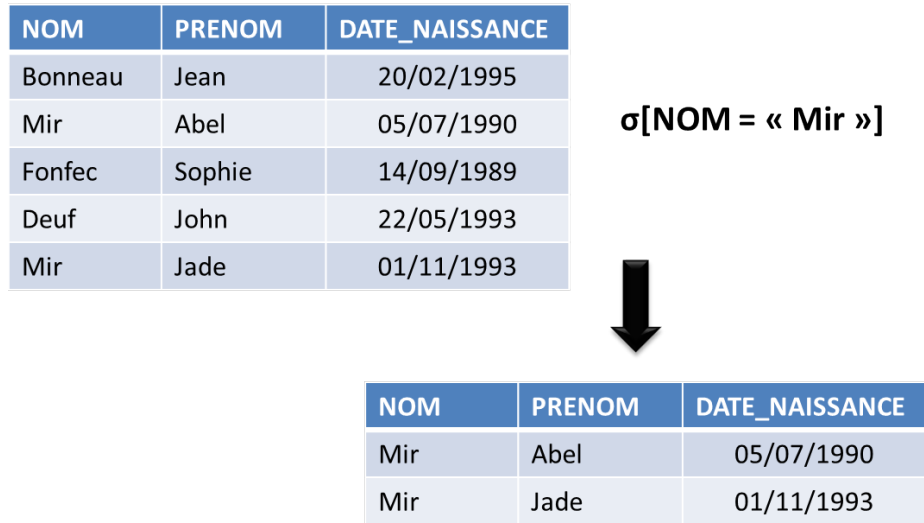


Figure 14: Sélection dans une relation

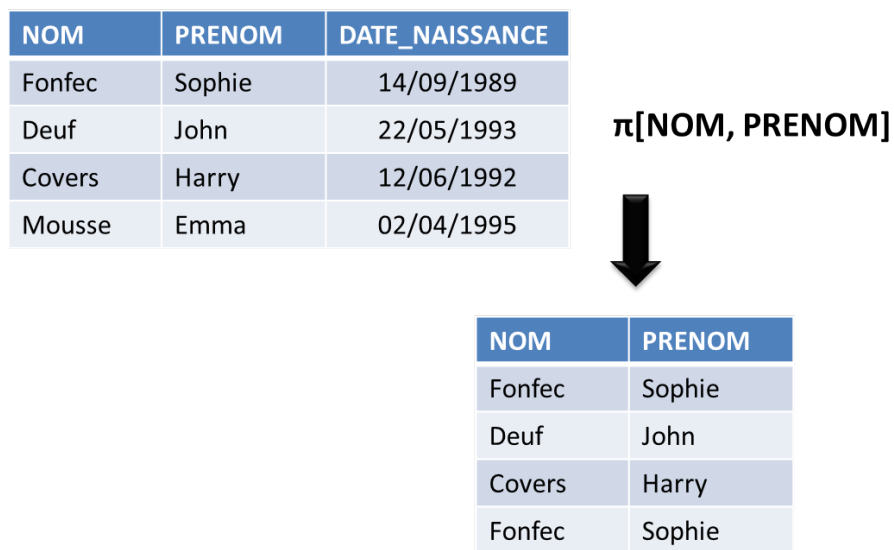


Figure 15: Projection dans une relation

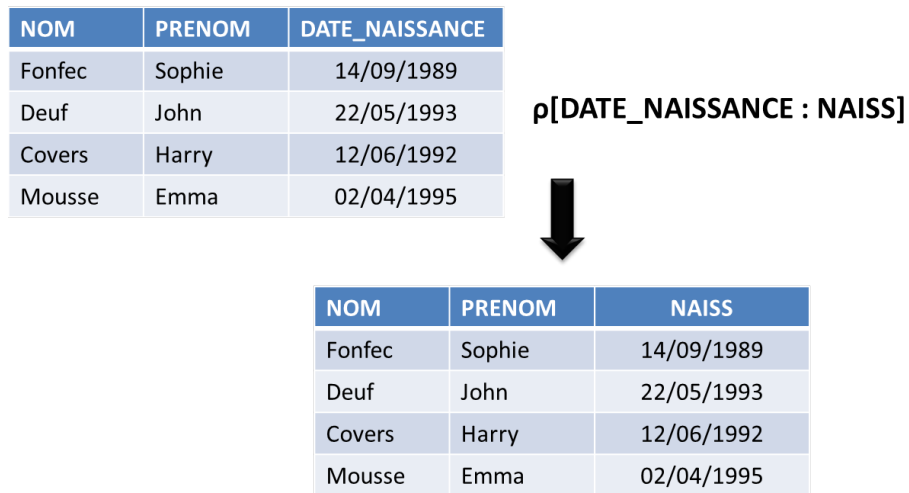


Figure 16: Renommage dans une relation

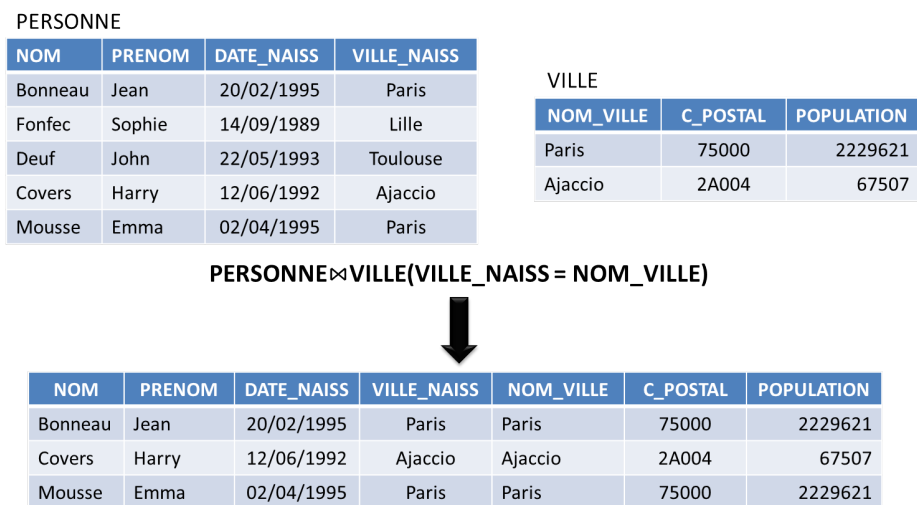


Figure 17: Jointure entre deux relations

Une **clé étrangère** est utilisée pour lier des relations entre elles. Elle fait référence à une clé primaire d'une autre relation.

Clés et liens entre relations (2/2)

- Différents types de liens possibles :
 - **1-1** : *un pays a une capitale, et une capitale est dans un seul pays*
 - **1-N** : *un pays possède plusieurs villes, mais une ville n'est que dans un seul pays*
 - **N-M** : *un pays possède des frontières et ces frontières concernent plusieurs pays*
- **Cardinalité** d'un lien

Exercice

Exercice - clés primaire et étrangères

UML

Problématique

- Comment établir le modèle conceptuel d'une BDD relationnelle ?
 - outils ?
 - méthodologie ?
 - ... ?

Problématique

- ☒ décrire une relation (*Relation(Colonne1, Colonne2...)*)
- ☒ décrire un domaine de manière textuelle (*Colonne1 est de type entier, la valeur doit être multiple de 3...*)
- ☐ le faire pour un ensemble de tables de manière concise

Présentation d'UML (1/2)

- UML = Unified Modeling Language
- Utilisé pour tous les aspects de la conception de projets informatiques
- Langage graphique, normalisé, interprétable par un ordinateur

Présentation d'UML (2/2)

- Actuellement en version 2
- 14 diagrammes :

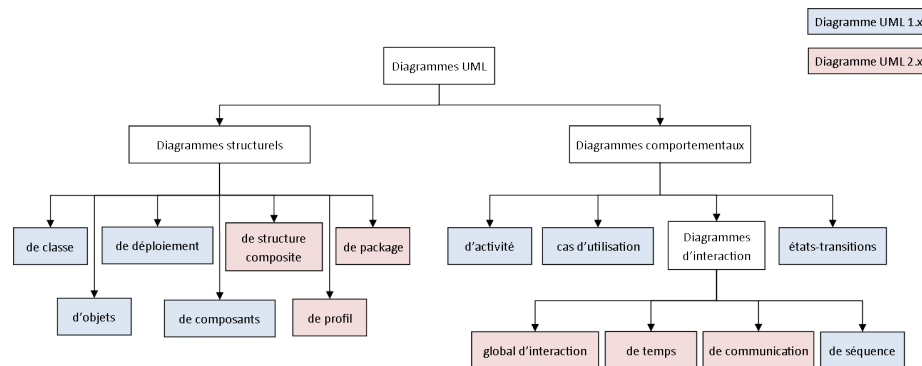


Figure 18: Diagrammes UML

Diagramme de classe

- *Diagramme de base de données*
- Relation représentée sous forme d'un rectangle constitué de deux cases
- Nom de la relation dans la case du haut
- Liste des colonnes dans la case du bas
- Syntaxe pour une colonne : `nom_colonne: type(longueur)`
- Contraintes particulières sur une colonne ajoutées sous forme de notes

Exemple

Différents types de liens

- Lien simple
- Agrégation
- Composition

Exercice

- Compagnie aériennes
- Association de randonneurs

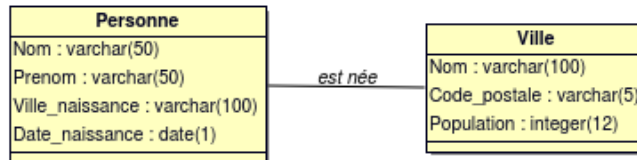


Figure 19: Diagramme UML de BDD

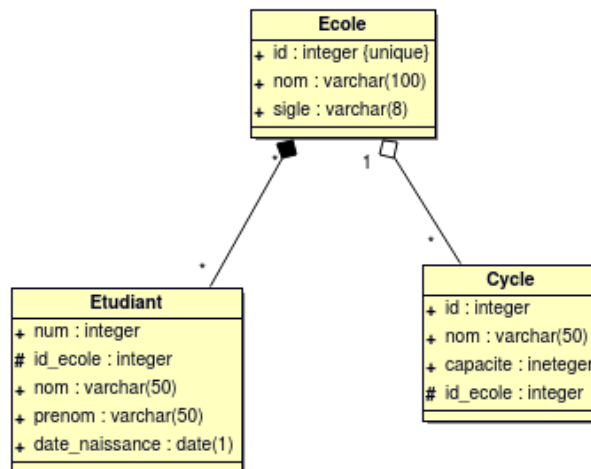


Figure 20: Agrégation et composition

Le langage SQL

Motivations / Problématique