

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

- 1 Les concepts généraux
- 2 Notions essentielles pour le modèle relationnel
- 3 Algèbre Relationnelle et SQL
- 4 Le modèle conceptuel Entités-Associations
- 5 Normalisation d'une relation

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

BD publiques  
Intro aux SGBD  
Cycle de vie  
Architecture

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

le modèle  
Entités-  
Associations

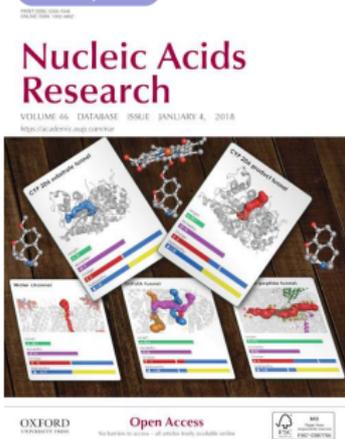
Normalisation  
d'une relation

- 1 Les concepts généraux
  - Bases de données publiques
  - Introduction aux SGBD
  - Cycle de vie d'une BD
  - Architecture d'une BD
- 2 Notions essentielles pour le modèle relationnel
- 3 Algèbre Relationnelle et SQL
- 4 Le modèle conceptuel Entités-Associations
- 5 Normalisation d'une relation

- Lieux de stockages d'informations biologiques.
- *Nucleic Acids Research* 2018 : liste de 180 bases de données !

- Le site wikipédia classe ces bases de données :  
[https://en.wikipedia.org/wiki/List\\_of\\_biological\\_databases](https://en.wikipedia.org/wiki/List_of_biological_databases)

► wikipedia



Main page  
Contents  
Current events  
Random article  
About Wikipedia  
Contact us  
Donate

Contribute  
Help  
Learn to edit  
Community portal  
Recent changes  
Upload file

Tools  
What links here  
Related changes  
Special pages  
Permanent link

Category   Talk

## Category:Biological databases

From Wikipedia, the free encyclopedia

The main article for this category is ***Biological database***.

### Contents

Top · 0-9 · A · B · C · D · E · F · G · H · I · J · K · L · M · N · O · P · Q · R · S · T · U · V · W · X · Y · Z

### Subcategories

This category has the following 13 subcategories, out of 13 total.

- |          |  |          |  |
|----------|--|----------|--|
| <b>B</b> | <ul style="list-style-type: none"> <li>↳ Biochemistry databases (2 P)</li> <li>↳ Biodiversity databases (3 C, 44 P)</li> <li>↳ Biometric databases (1 C, 16 P)</li> <li>↳ Biotechnology databases (3 P)</li> </ul> | <b>F</b> | <ul style="list-style-type: none"> <li>↳ Fisheries databases (13 P)</li> <li>↳ Food databases (6 P)</li> </ul> |
| <b>G</b> | <ul style="list-style-type: none"> <li>↳ Gene banks (23 P)</li> <li>↳ Genetics databases (2 C, 24 P)</li> </ul>  | <b>E</b> | <ul style="list-style-type: none"> <li>↳ Ecological databases (4 P)</li> </ul>                                 |

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxBD publiques  
Intro aux SGBD  
Cycle de vie  
ArchitectureNotions  
essentielsAlgèbre  
Relationnelle  
+ SQLle modèle  
Entités-  
AssociationsNormalisation  
d'une relation

## ● séquences nucléiques

- Genbank (NCBI, National Center for Biotechnology Information)
- EMBL (European Molecular Biology Laboratory)
- DDBJ (DNA Data Bank of Japan)

Plus de 300000 organismes, 241 millions de séquences,  
1635 milliards de paire de bases (décembre 2022) [▶ GenBank](#)

- **GOLD** : Genomes OnLine Database, 55 000 études, 183 000 échantillons biologiques, [▶ GOLD](#)
- 1000 genomes project... 4978 génomes humains  
[▶ 1000 genomes](#)

## ● séquences protéiques

- **UniProtKB/TrEMBL** (translated EMBL, annoté automatiquement, 229.000.000 séquences) [▶ TrEMBL](#)  
**UniProtKB/SwissProt** (annoté par des personnes, 568.000 entrées), [▶ SwissProt](#)  
**Protein Information Resource (PIR)**,...  
**UniProt** : TrEMBL  $\cup$  SwissProt  $\cup$  PIR
- **HIV-1 Protein Interactions** : interactions des protéines de HIV-1 [▶ HIV-1 protein interactions](#)
- **GISAID Initiative** : COVID-19 + influenza viruses (phylogénie des virus) [▶ GISAID](#)

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxBD publiques  
Intro aux SGBD  
Cycle de vie  
ArchitectureNotions  
essentielsAlgèbre  
Relationnelle  
+ SQLle modèle  
Entités-  
AssociationsNormalisation  
d'une relation

- Structures tridimensionnelles :
  - **PDB** : structure 3D Protein Data Bank, 199.000 entrées  
▶ PDB
  - **MMDB** : molecular Modeling Database (chirallographie par rayon X + spectroscopie NMR). 169000 structures  
▶ MMDB
- Transcriptome (variétés de méthodes – puces à oligo, puces à ADN classique, SAGE ...)
- Polymorphisme : **dbSNP** (900.000.000 Single Nucleotide Polymorphisms) ▶ dbSNP
- Structure chimique : **PubChem** : structures chimiques des petites molécules organiques et des informations sur leurs activités biologiques (113 millions de composés chimiques). ▶ PubChem
- Bases de données de noms :
  - Plants of the world (1,416,000 global plant names) ▶ PotW
  - WoRMS : marine names (602000 noms) ▶ WoRMS
- Données bibliographiques : **Medline**, **PubMed** (une base de données de citations et d'abstracts de la littérature biomédicale – plus large que medline). 34 millions de citations, de 1781 à nos jours. ▶ PubMed **PubMed Central**

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

BD publiques  
Intro aux SGBD  
Cycle de vie  
Architecture

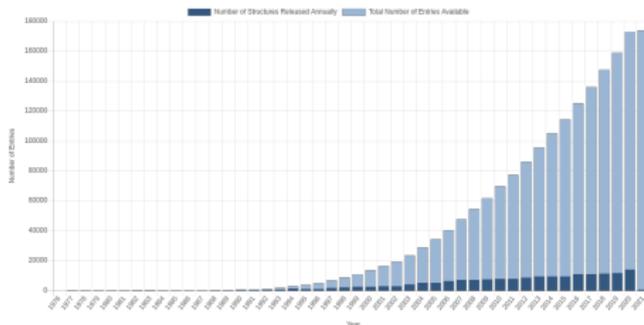
Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

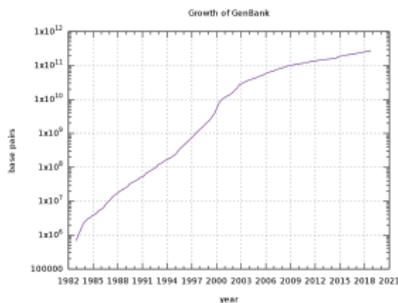
le modèle  
Entités-  
Associations

Normalisation  
d'une relation

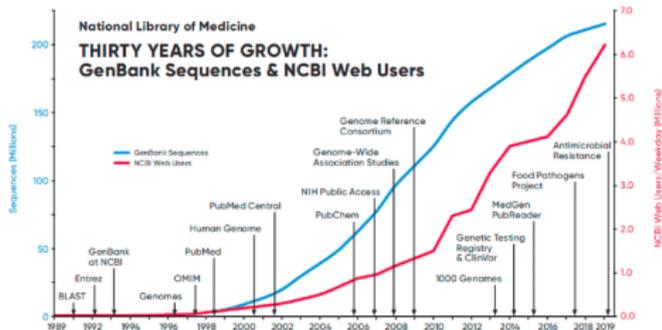
PDB :



GenBank :



As of release 228, retrieved 2018-11-30



Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxBD publiques  
Intro aux SGBD  
Cycle de vie  
ArchitectureNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQLle modèle  
Entités-  
AssociationsNormalisation  
d'une relation

- 1 **Banque de données** : un ensemble de données, propres à une certaine application.
  - collectées par les producteurs de données
  - puis commercialisées pour un usage plus large.
  - Pas de contraintes fortes de cohérence, de non-redondance.⇒ *Gros amas de données accessibles en ligne, produites par les institutions.*
- 2 **Base de données** : ensemble cohérent, intégré et partagé d'informations nécessaires au fonctionnement d'une entreprise,
  - gestion assurée par un logiciel "**Système de gestion de bases de données (SGBD)**" (Data Base Management System en anglais (DBMS)).
  - principaux SGBD : Oracle (44 %), DB2 (IBM, 22 %), SQL server, Access et Foxpro (Microsoft, 22 %), Sybase (3 %).

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxBD publiques  
Intro aux SGBD  
Cycle de vie  
ArchitectureNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQLle modèle  
Entités-  
AssociationsNormalisation  
d'une relation

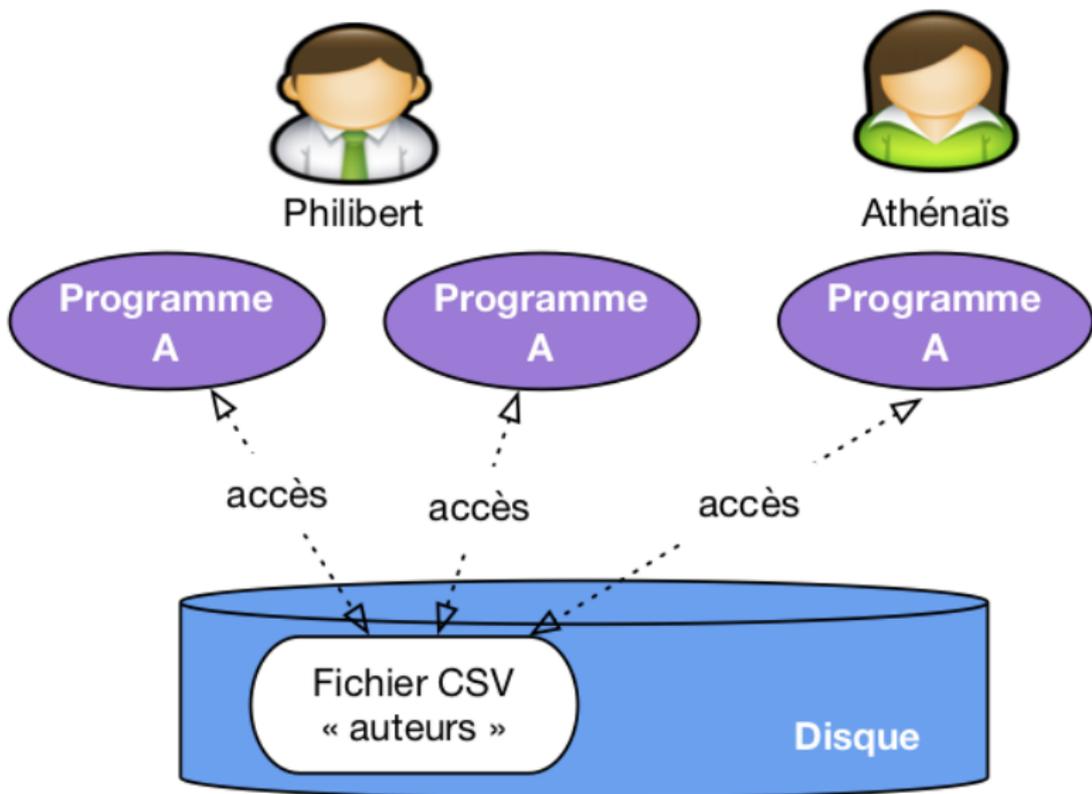
- ③ **Fichier** : Lorsque l'ensemble des données est homogène (ensemble des abonnés à une revue, le personnel d'une entreprise, les produits vendus en ligne...)
- **fichier** des abonnés, du personnel, des produits.
  - Dans ce cas, on utilisera un **système de gestion de fichiers (SGF)**, beaucoup moins complexe qu'une base de données.

Lorsque les données à gérer sont de diverses natures (étudiants, cours enseignements, filières, ECTS, salles) :

- la seule notion de fichier ne suffit pas, et  
⇒ utiliser l'approche bases de données.

Remarques :

- le stockage des données d'une BD se fait *via* des fichiers,
- ces fichiers sont totalement invisibles à l'utilisateur (relève de la compétence du SGBD)

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxBD publiques  
Intro aux SGBD  
Cycle de vie  
ArchitectureNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQLle modèle  
Entités-  
AssociationsNormalisation  
d'une relation

SGBD = Système de Gestion d'une Base de Données

DBMS = DataBase Manager System

Quelles sont les spécificités d'un SGBD ?

- 1 **Très grande quantité de données à gérer**, qui doivent être stockées dans plusieurs fichiers, voire plusieurs sites...
- 2 **Besoin d'interroger, mettre à jour souvent, rapidement et efficacement ces données**
- 3 **Contrôler la redondance d'informations**

- coût élevé en temps, en volume
- risque d'incohérence entre les différentes copies

Objectif : contrôler cette redondance, voire de la supprimer

- 4 **Partage des données / Accès concurrents.**
  - accès simultanés à la même information par plusieurs utilisateurs
  - contrôle de la concurrence : verrouillage des données (évite la lecture d'une information pendant sa mise à jour).
  - vue utilisateur : définir pour chaque classe d'utilisateurs la portion de la base de données qui l'intéresse (et dans la forme qui l'intéresse).

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxBD publiques  
Intro aux SGBD  
Cycle de vie  
ArchitectureNotions  
essentielsAlgèbre  
Relationnelle  
+ SQLle modèle  
Entités-  
AssociationsNormalisation  
d'une relation**5 Gérer les autorisations d'accès / Sécurité des données**

- confidentialité des données (multi-utilisateurs).
- droits sur les données (lecture, mise à jour, création, ... )
- notion de vue utilisateur.

**6 Offrir des interfaces d'accès multiples**

- plusieurs interfaces, pour les différents types d'utilisateurs
- interfaces orientées utilisateur final (langage de requête SQL / interface de type formulaire, ...)
- interfaces orientées programmeurs (interface avec des langages de programmation classiques : "embedded SQL").

**7 Représenter des relations complexes entre les données à travers un modèle de données.**

- modèles hiérarchique/réseau/relationnel/orienté-objet.
- chacun : ses propres concepts de relation entre les données

**8 Vérifier les contraintes d'intégrité :**

- contraintes structurelles (un employé a un chef unique)
- contraintes dynamiques (un salaire ne peut diminuer).

Souvent les contraintes dyn. doivent être codées dans les pg d'application.

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

BD publiques  
Intro aux SGBD  
Cycle de vie  
Architecture

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

## 9 Assurer la reprise après panne

- Une panne ne doit pas pouvoir bloquer le fonctionnement de l'entreprise de manière durable.

- Mécanismes pour assurer cette sécurité :  
**notion de transaction :**

- une séquence d'opérations élémentaires
- elle s'effectue complètement avec succès ou elle est annulée
- fait passer la BD d'un état cohérent à un nouvel état cohérent.

### **mécanisme de journalisation :**

- risques liés aux pannes disques
- régénérer une BD automatiquement à partir d'une version de sauvegarde et du journal des mouvements.

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

BD publiques

Intro aux SGBD

Cycle de vie

Architecture

Notions  
essentiels

Algèbre  
Relationnelle  
+ SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

- 1 BD relationnelles (Edgar Franck Codd, 1970).
  - Tables/Relations pour stocker des données reliées
  - Algèbre relationnelle ( $\cup$ ,  $\cap$ , projection... : théorie des ens.)
  - Vraie indépendance logique/physique
  - Premier SGBD Relationnel : IBM vers 1980  
1987 : SQL = Structured Query Language
- 2 BD Déductives : utilisation d'un langage logique (prolog).  
**Exemple** : Pascale ; Laurent ; Louise ;  
apourpere(Pascale, Louis) ;  
apourpère(Louis, Romuald) ;  
  
Quelles sont les ancêtres de Pascale ?
- 3 BD orientées Objets.
  - le modèle rel. n'est pas très adapté pour les données structurées comme les données multimédia, graphiques, géographiques...
  - le modèle objets essaie de répondre à ces nouveaux besoins.

Dans ce cours, nous ne verrons que le **modèle relationnel**.

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxBD publiques  
Intro aux SGBD  
Cycle de vie  
ArchitectureNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQLle modèle  
Entités-  
AssociationsNormalisation  
d'une relation

Niveau logique : vision conceptuelle des données

Niveau physique : implémentation

L'indépendance entre ces 2 niveaux est **TRES** importante :

- l'utilisateur n'est pas un spécialiste de l'implémentation.  
Il doit comprendre comment les données sont *logiquement* organisées
- l'implémentation peut changer (performances).  
Cela doit pas influencer l'organisation logique des données.  
(sinon : tous les pg d'interrogation seraient à refaire à chaque changement)
- Le modèle logique est plus clair, car indépendant des questions d'implémentation.
  - possibilité d'un langage de requête clair
  - si le modèle physique change, on n'est pas obligé de recoder les programmes de requêtes / mise à jour

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxBD publiques  
Intro aux SGBD  
Cycle de vie  
ArchitectureNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQLle modèle  
Entités-  
AssociationsNormalisation  
d'une relation**1 Conception d'une base de données.**

- c'est la phase d'analyse qui aboutit à déterminer le futur contenu de la base de données.
- Cette phase doit répondre aux questions suivantes :
  - Quelles sont les informations à mettre dans la BD ?
  - Sous quel format ?
- Il faut se mettre d'accord sur la nature et les caractéristiques des informations que la BD contiendra.
- travail à faire avec les futurs utilisateurs de la BD, pour savoir quels types d'interrogations seront à implémenter.

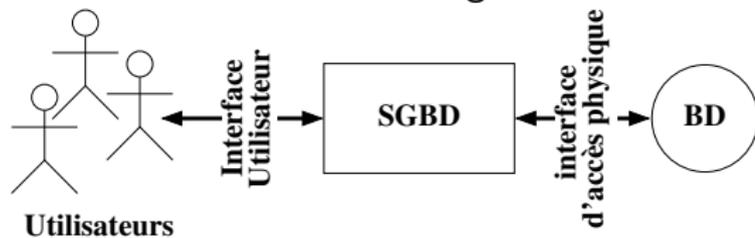
**2 Implantation de la base de données.**

- implémentation du résultat de la 1<sup>ème</sup> phase dans le SGBD
- **Langage de description de données (LDD).**
- Une fois que la description de la structure est donnée au SGBD, version initiale de la BD en entrant les premières données.

**3 Utilisation de la base de données.**

- **Langage de manipulation des données (LMD) :**
  - requêtes d'interrogation
  - requêtes de mise à jour (ajout, suppression, modification).

Un SGBD  $\equiv$  boîte noire assurant la gestion de la BD.



- Les utilisateurs interrogent la BD *via* l'**interface utilisateur**
    - définir le contenu de la BD (LDD),
    - interroger / modifier son contenu (LMD)
    - Cette interface doit gérer les questions suivantes : compréhension, analyse et vérification des requêtes.
  - le SGBD accède aux disques *via* l'**interface d'accès physique**.
    - optimisation du stockage des données (volume)
    - gestion des accès aux données (temps)
- car L'espace doit être minimal ainsi que le temps de réponse.

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

BD publiques  
Intro aux SGBD  
Cycle de vie  
Architecture

Notions  
essentielles

Algèbre  
Relationnelle  
+ SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

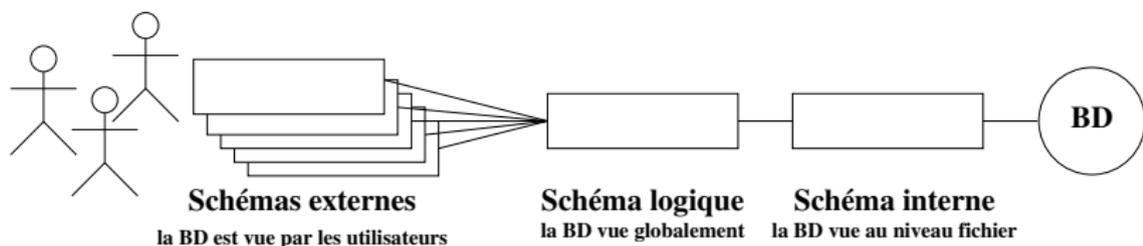
- Rappel : les différentes classes d'utilisateurs ne voient pas la BD de la même manière (vues).

Cette architecture est fondamentale pour pouvoir proposer l'**indépendance programme/ données**. Il faut pouvoir

- proposer aux utilisateurs de modifier leurs vues de la BD et leurs traitements sans se soucier des choix faits au niveau interne (fichiers).
- proposer à l'administrateur de la BD de pouvoir modifier ces choix, afin d'améliorer les performances.

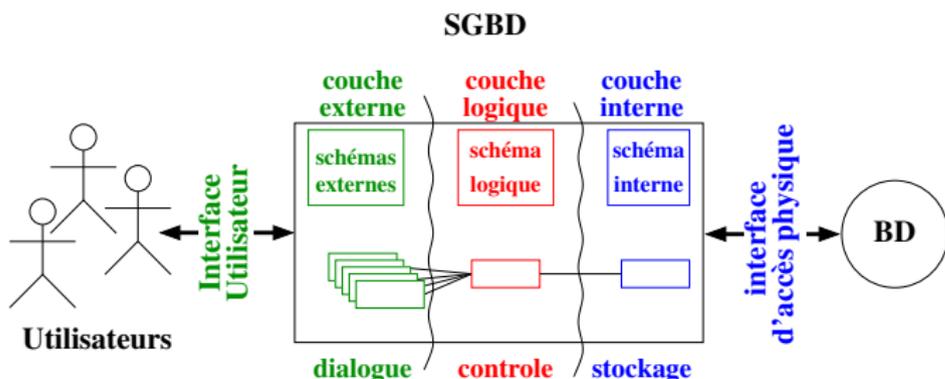
Accès multiples : le SGBD doit gérer des pb de coordination, de cohérence et de contrôle de ses actions

⇒ l'architecture doit être affinée : 3 couches



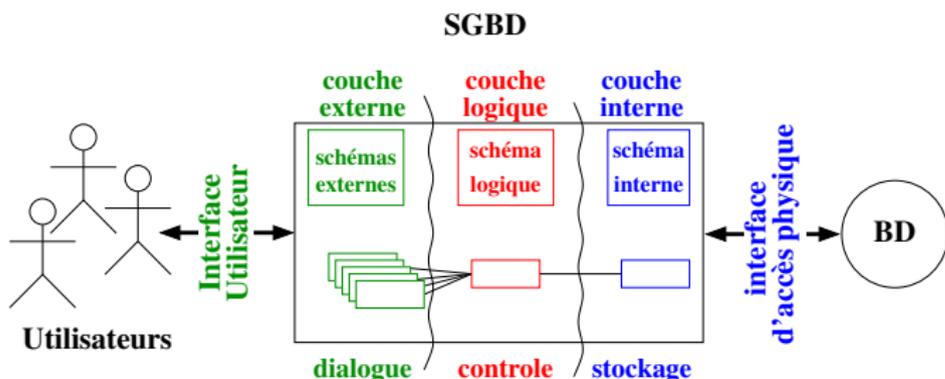
- 1 **Niveau externe** : dialogue avec les utilisateurs (analyse des demandes, contrôle des droits d'accès, présentation des résultats)
- 2 **Niveau interne** : stockage sur les supports physiques, gestion des structures de mémorisation (fichiers), et accès (gestion des index, des clés...)
- 3 **Niveau intermédiaire** : contrôle global
  - optimisation des requêtes,
  - gestion des conflits d'accès simultanés,
  - contrôle général de la cohérence globale,
  - coordination et suivi des processus en cours,
  - garantie du bon déroulement des actions entreprises, ...

**Niveau logique** : *ne dépendre ni des exigences des utilisateurs, ni des structures physiques choisies.*

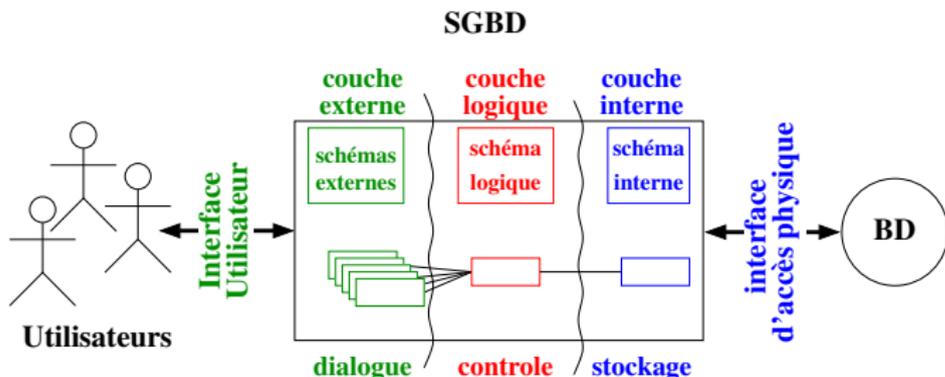


## 1 Une requête arrive :

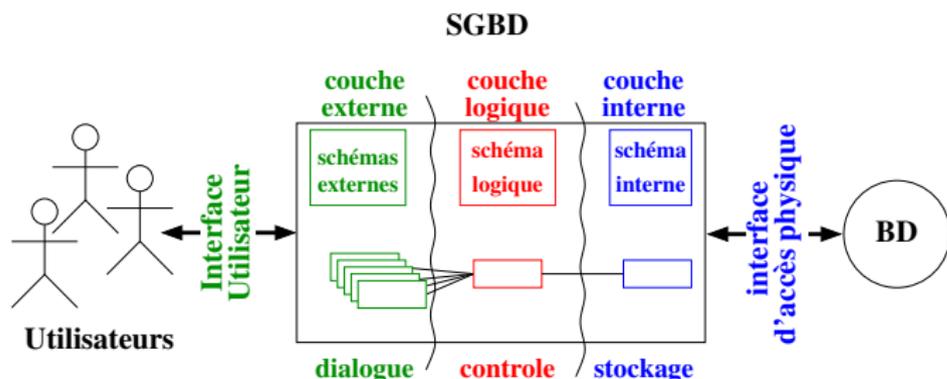
- exprimée dans un langage accepté par le système (LMD)
- analysée du point de vue syntaxique (conformité à la grammaire),
- puis du point de vue sémantique (les objets cités doivent être connus dans le schéma externe de l'utilisateur).
- validation au niveau externe



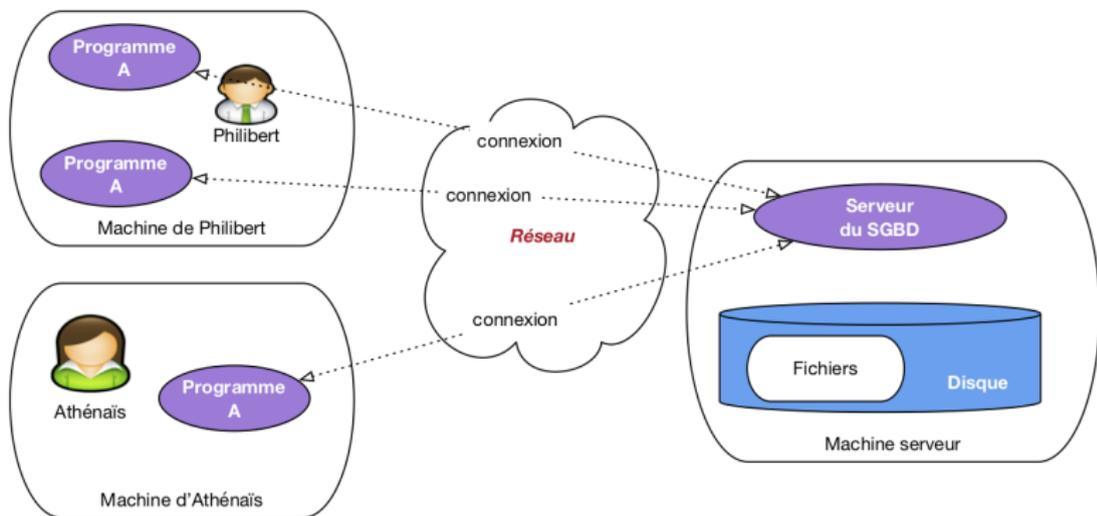
- 2 Traduction et passage de la requête à la couche logique
- les références aux objets du schéma externe sont remplacés par des références aux objets correspondants dans le schéma logique.
- Utilisation de règles de correspondance entre schémas logique et externe (règles définies au moment de la définition des schémas externes).

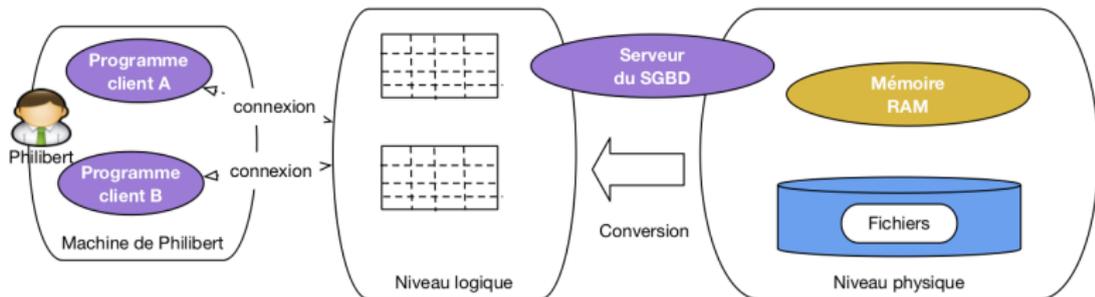


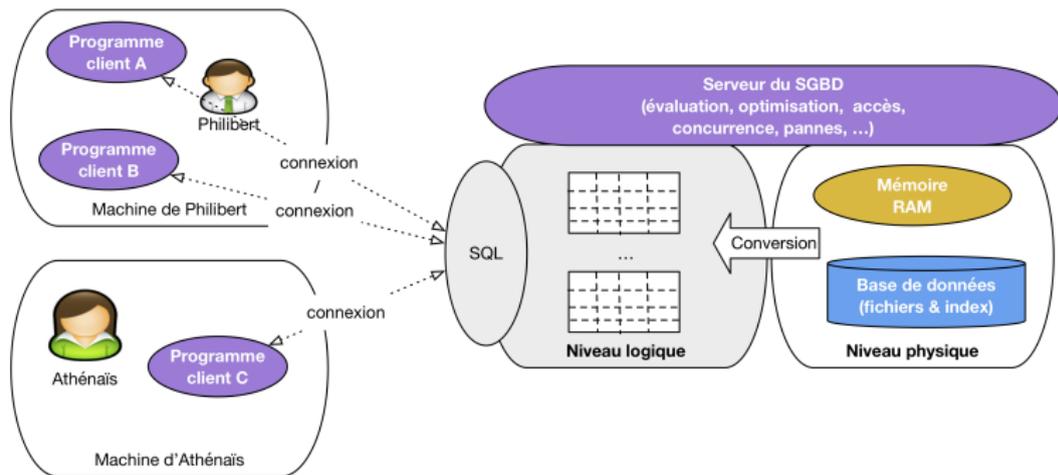
- Traduction et passage de la requête à la couche logique**
  - les références aux objets du schéma externe sont remplacés par des références aux objets correspondants dans le schéma logique.  
Utilisation de règles de correspondance entre schémas logique et externe (règles définies au moment de la définition des schémas externes).
- Au niveau logique : contrôles de confidentialité, sur la concurrence,...**
  - Si la requête est acceptée, elle est optimisée et découpée en sous-requêtes qui sont transférées au niveau interne,
  - sinon elle est soit refusée soit mise en attente.



- 4 Au niveau interne : chaque sous-requête est traduite en une ou plusieurs requêtes physiques correspondantes (en fonction du schéma interne).
- S'il s'agit d'une requête d'interrogation, les données extraites sont passées à la couche logique, puis à la couche externe (elles sont réorganisées selon le schéma externe de l'utilisateur).

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxBD publiques  
Intro aux SGBD  
Cycle de vie  
ArchitectureNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQLle modèle  
Entités-  
AssociationsNormalisation  
d'une relation

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxBD publiques  
Intro aux SGBD  
Cycle de vie  
ArchitectureNotions  
essentiellesAlgèbre  
Relationnelle  
+ SQLle modèle  
Entités-  
AssociationsNormalisation  
d'une relation

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxBD publiques  
Intro aux SGBD  
Cycle de vie  
ArchitectureNotions  
essentielsAlgèbre  
Relationnelle  
+ SQLle modèle  
Entités-  
AssociationsNormalisation  
d'une relation

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

- BD publiques
- Intro aux SGBD
- Cycle de vie
- Architecture

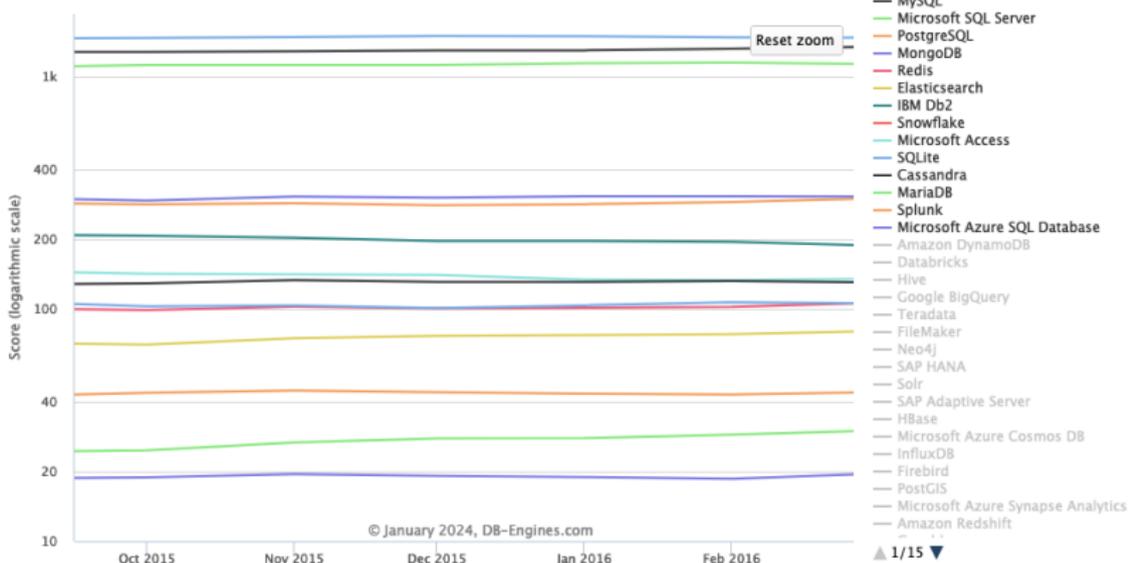
Notions  
essentiels

Algèbre  
Relationnelle  
+ SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

DB-Engines Ranking



Un VPN est un réseau privé qui utilise un réseau publique comme backbone

- Seuls les utilisateurs ou les groupes qui sont enregistrés dans ce vpn peuvent y accéder.
- Les données transitent dans un tunnel après avoir été chiffrées.
- Tout se passe comme si la connexion se faisait en dehors d'infrastructure d'accès partagé comme Internet
- Étanchéité du trafic entre les différents réseaux privés virtuels



Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

BD publiques  
Intro aux SGBD  
Cycle de vie  
Architecture

Notions  
essentielles

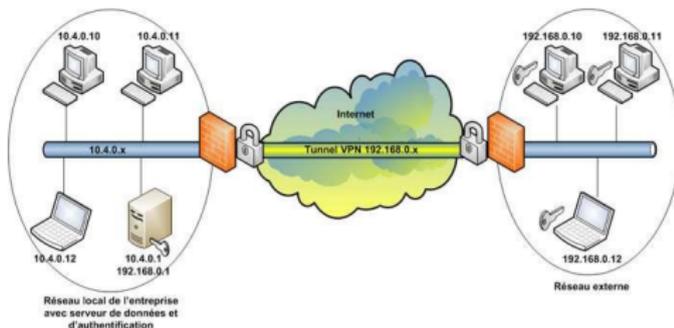
Algèbre  
Relationnelle  
+ SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

Un VPN est un réseau privé qui utilise un réseau public comme backbone

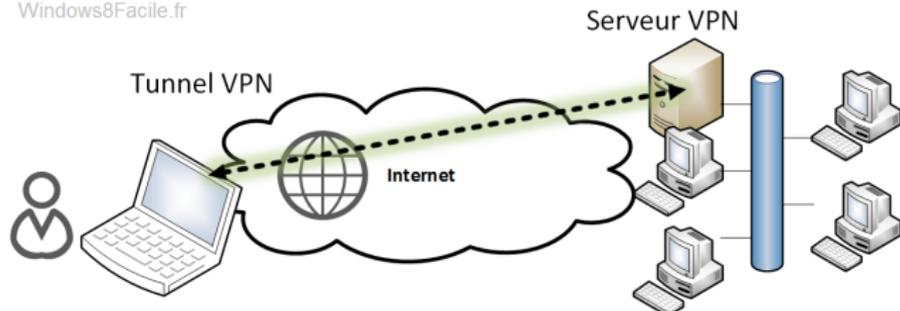
- Seuls les utilisateurs ou les groupes qui sont enregistrés dans ce vpn peuvent y accéder.
- Les données transitent dans un tunnel après avoir été chiffrées.
- Tout se passe comme si la connexion se faisait en dehors d'infrastructure d'accès partagé comme Internet
- Étanchéité du trafic entre les différents réseaux privés virtuels



Un VPN est un réseau privé qui utilise un réseau public comme backbone

- Seuls les utilisateurs ou les groupes qui sont enregistrés dans ce vpn peuvent y accéder.
- Les données transitent dans un tunnel après avoir été chiffrées.
- Tout se passe comme si la connexion se faisait en dehors d'infrastructure d'accès partagé comme Internet
- Étanchéité du trafic entre les différents réseaux privés virtuels

Windows8Facile.fr



Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxBD publiques  
Intro aux SGBD  
Cycle de vie  
ArchitectureNotions  
essentielsAlgèbre  
Relationnelle  
+ SQLle modèle  
Entités-  
AssociationsNormalisation  
d'une relation

- 1 Engagement étudiant : **est-ce fait ?**

<https://glpi.univ-cotedazur.fr/...=153>

- 2 Téléchargement de *anyconnect* :

*anyconnect-win-4.10.05085-predeploy-k9.zip* sur

<https://wiki.univ-cotedazur.fr/display/SUP/VPN>

windows / MacOS / Linux

- 3 Installation : suivant votre système d'exploitation

- 4 Connexion :

Connect to :	<b>open.unice.fr</b>
Group :	<b>UNS</b>
Username :	<b>login@etu.polytech</b>
Password :	

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Schéma d'une rel.

Schéma de BD

Identifiants

LDD de SQL

Algèbre  
Relationnelle  
+ SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

- 1 Les concepts généraux
- 2 Notions essentielles pour le modèle relationnel
  - Schéma d'une relation dont le nom est R.
  - Un schéma de base de données
  - Les identifiants d'une relation
  - Langage de Description de Données de SQL
- 3 Algèbre Relationnelle et SQL
- 4 Le modèle conceptuel Entités-Associations
- 5 Normalisation d'une relation

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentielsSchéma d'une rel.  
Schéma de BD  
Identifiants  
LDD de SQLAlgèbre  
Relationnelle  
+ SQLle modèle  
Entités-  
AssociationsNormalisation  
d'une relation

- Le modèle relationnel est basé sur une organisation des données sous forme de tables (tableaux de valeurs).
- concept mathématique de **relation** de la théorie des ensembles
- **L'algèbre relationnelle** : 1970 par Edgar Franck Codd (IBM) : ensemble d'opérations formelles sur les relations.  $T_1 \square T_2 \rightarrow T_3$
- une relation dans un ensemble (« sur un ensemble ») est une proposition qui lie un certain nombre d'éléments.
  - La relation « est\_marié\_avec » relie deux personnes
  - La relation entre commande/client est une relation binaire (dim 2).
  - Une location de voiture représente une relation entre un véhicule, une personne et une date.

Une relation peut être caractérisée par  $n$  dimensions (arité  $n$  ou relation  $n$ -aire).

Un **univers** est un ensemble fini et non vide de noms, appelés attributs.

**Exemple** : horaires des projections des films actuellement à l'affiche

- $U = \{ \text{titreFilm}, \text{realisateur}, \text{acteur}, \text{producteur}, \text{nomCinema}, \text{horaire} \}$
- $\text{Dom}(\text{titreFilm}) = \text{chaînes de caractères}$
- $\text{Dom}(\text{realisateur}) = \text{chaînes de caractères}$
- $\text{Dom}(\text{acteur}) = \text{chaînes de caractères}$
- $\text{Dom}(\text{producteur}) = \text{chaînes de caractères}$
- $\text{Dom}(\text{nomCinema}) = \text{chaînes de caractères}$
- $\text{Dom}(\text{horaire}) = \{ h : m \mid h \in [1, \dots, 23], m \in [0, \dots, 59] \}$

Un **schéma d'une relation** dont le nom est R est un sous ensemble non vide de l'univers U.

### Exemple :

- Schéma de la relation

$$\text{Film} = \{ \text{titreFilm}, \text{realisateur}, \text{acteur}, \text{producteur} \}$$

- Schéma de la relation

$$\text{Projection} = \{ \text{titreFilm}, \text{nomCinema}, \text{horaire} \}$$

**Intuition** : on décrit deux tables dont les colonnes vont décrire les attributs de certains objets, et les lignes ces objets.

*Film* =

titreFilm	realisateur	acteur	producteur
⋮	⋮	⋮	⋮

*Projection* =

titreFilm	nomCinema	horaire
⋮	⋮	⋮

Soit  $E = \{A_1, A_2, \dots, A_n\}$  le schéma d'une relation.

Un  **$n$ -uplet** est une fonction

$$n : E \rightarrow \text{Dom}(A_1) \cup \dots \cup \text{Dom}(A_n)$$

telle que  $\forall A_i \in E, n(A_i) \in \text{Dom}(A_i)$

Si  $E' \subset E$ , la restriction de  $n$  à  $E'$  se note  $n(E')$ .

**Exemple.**

- un  $n$ -uplet sur la relation *Film* :

titreFilm	→	"La syndicaliste"		realisateur	→	"Jean-Paul Salomé"
acteur	→	"Isabelle Huppert"		producteur	→	"Bertrand Faivre"

("La syndicaliste", "Jean-Paul Salomé", "Isabelle Huppert", "Bertrand Faivre")
- un  $n$ -uplet sur la relation *Projection* :

titreFilm	→	"La syndicaliste"		nomCinema	→	"Cap Cinema"
horaire	→	"22 :15"				

("La syndicaliste", "Cap Cinema", "22 :15")
- sa restriction à (nomCinema, titreFilm) est :

titreFilm	→	"La syndicaliste"		nomCinema	→	"Cap Cinema"
-----------	---	-------------------	--	-----------	---	--------------

("La syndicaliste", "Cap Cinema")

Pourquoi avoir défini un  $n$ -uplet comme une fonction?

Une **relation**  $r$  sur un schéma de relation  $S$  est un ensemble de  $n$ -uplets sur  $S$ . On dit aussi que  $S$  est le schéma de  $r$ .

## Exemple.

*Film* =

titreFilm	realisateur	acteur	producteur
"La syndicaliste"	"Jean-Paul Salomé"	"Isabelle Huppert"	"Bertrand Faivre"
"La syndicaliste"	"Jean-Paul Salomé"	"Grégory Gadebois"	"Bertrand Faivre"
"La syndicaliste"	"Jean-Paul Salomé"	"François-Xavier Demaison"	"Bertrand Faivre"
"The Fabelmans"	"Steven Spielberg"	"Gabriel LaBelle"	"Kristie Macosko Krieger"
"The Son"	"Florian Zeller"	"Hugh Jackman"	"Joanna Laurie"

*Projection* =

titreFilm	nomCinema	horaire
"La syndicaliste"	"Cap Cinema"	"13 :15"
"La syndicaliste"	"Cap Cinema"	"16 :30"
"La syndicaliste"	"Cap Cinema"	"21 :05"
"La syndicaliste"	"Cinéma du centre"	"18 :55"
"The Fabelmans"	"Cap Cinema"	"15 :50"
"The Son"	"Cinéma du centre"	"21 :00"

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Schéma d'une rel.

Schéma de BD

Identifiants

LDD de SQL

Algèbre  
Relationnelle  
+ SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

Un **schéma d'une base de données** sur un univers  $U$  est un ensemble non vide d'expressions de la forme  $N(S)$  où

- $S$  est un schéma de relation et
- $N$  est un nom de relation.

### Exemple :

- $U = \{ \text{titreFilm}, \text{realisateur}, \text{acteur}, \text{producteur}, \text{nomCinema}, \text{horaire} \}$
- $S = \left\{ \begin{array}{l} \text{Film}(\text{titreFilm}, \text{realisateur}, \text{acteur}, \text{producteur}), \\ \text{Projection}(\text{titreFilm}, \text{nomCinema}, \text{horaire}) \end{array} \right\}$

**Intuition :** Le schéma de la base est la donnée du format des schémas des différentes tables de la base.

Une **base de données**  $B$  sur un schéma de base  $\mathcal{S}$  (avec univers  $U$ ) est un ensemble de relations finies  $r_1, \dots, r_n$  où

- $r_i$  est associée à un nom de relation  $N_i$  et
- $r_i$  est telle que si  $N_i(S) \in \mathcal{S}$ , alors  $r_i$  a  $S$  comme schéma.

- On peut aussi imposer des contraintes sur les données.
  - 1 l'un des attributs d'une table doit être une **clef** de cette table (il n'existe pas deux  $n$ -uplets de cette relation qui ont la même valeur pour cet attribut).
  - 2 pour une même classe de personnel, le salaire le plus haut ne doit pas dépasser de plus de 20% le salaire le plus bas.

Ces contraintes sont appelées **contraintes d'intégrité**, elles font aussi partie de la spécification du format des données de la base.

Un **identifiant** (ou **clé**, **clé primaire**) d'une relation est un ensemble minimum d'attributs de la relation tel qu'il ne peut pas exister deux  $n$ -uplets (tuples en anglais) ayant même valeur pour cet identifiant.

- Une relation peut avoir plusieurs identifiants.
- Un identifiant peut être composé de plusieurs attributs.
- On souligne les identifiants (attention :  $\{\underline{a}, b\} \neq \{a, \underline{b}\}$ ).

**Exemple** : Nom, prénom, dateN, LieuN, NumSS, NumTel, adresse

**Règle** : Les champs associés aux attributs des identifiants sont toujours remplis. On dit que les attributs des identifiants sont obligatoires.  $\Rightarrow$  Pourquoi ?

Pour favoriser la compréhension du schéma, on définit une **relation** par :

- son nom,
- une liste de couples (nom d'attr, dom),
- ses identifiants,
- sa définition en français.

Les **identifiants externes** ou **clés externes** ou **clés étrangères** sont des attributs ou des ensembles d'attributs dont la valeur est toujours égale à un identifiant d'une autre relation (voire de la même).

### Exemple.

- Etudiants (nom, prénom, numEtudiant, age)
- Cours (nomC, horaire, enseignant)
- suit (numEtudiant, nomC)

Dans la relation suit :

- nomC référence un cours
- numEtudiant référence un étudiant.

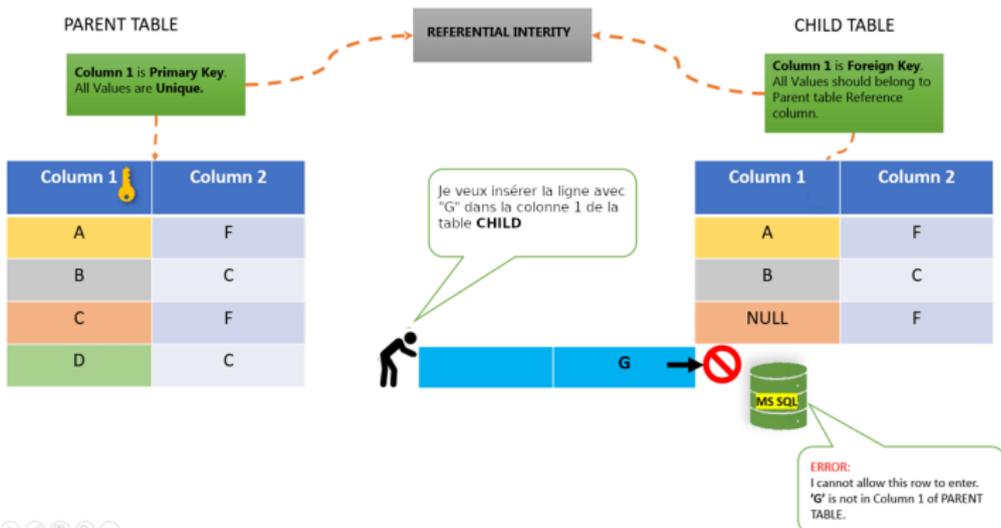
Définition d'un schéma d'une relation :

Suit(numEtudiant: Integer, nomC: chaîne de caractères)  
identifiants externes : numEtudiant référence un Etudiant  
nomC référence un Cours

Si la relation référencée a plusieurs identifiants,

- il faut alors préciser lequel est utilisé comme clef externe : `nomC` référence un `Cours.nomC`.

A tout instant, les valeurs d'une clef externe doivent appartenir à l'ensemble des valeurs pour la clef primaire de la relation référencée.



Si la relation référencée a plusieurs identifiants,

- il faut alors préciser lequel est utilisé comme clef externe :  
`nomC` référence un `Cours.nomC`.

A tout instant, les valeurs d'une clef externe doivent appartenir à l'ensemble des valeurs pour la clef primaire de la relation référencée.

- Cette **intégrité de référence** est assurée par le SGBD
- une insertion/modification d'un  $n$ -uplet avec clef externe doit être compatible avec un  $n$ -uplet de la relation référencée,
- une suppression d'un  $n$ -uplet de la table référencée ne doit pas entraîner des incohérences dans certains  $n$ -uplets de la table avec clef externe.

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Schéma d'une rel.

Schéma de BD

Identifiants

LDD de SQL

Algèbre  
Relationnelle  
+ SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

Comment représenter un attribut multi-valué, c'est à dire une colonne de la table dans laquelle on aimerait mettre plusieurs valeurs ?

Par exemple, les étudiants peuvent avoir plusieurs prénoms, mais le nombre de prénoms n'est pas connu à l'avance.

Comment représenter un attribut multi-valué, c'est à dire une colonne de la table dans laquelle on aimerait mettre plusieurs valeurs ?

Par exemple, les étudiants peuvent avoir plusieurs prénoms, mais le nombre de prénoms n'est pas connu à l'avance.

- 1 Considerer 3 attributs (prenom1, prenom2, prenom3).
  - Combien d'attributs a-t-on besoin ?
  - Comment poser une question sur l'un des prénoms ?
  - Inconvénients : on réserve de la place en trop pour les étudiants qui n'ont qu'un seul prénom.

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Schéma d'une rel.

Schéma de BD

Identifiants

LDD de SQL

Algèbre  
Relationnelle  
+ SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

Comment représenter un attribut multi-valué, c'est à dire une colonne de la table dans laquelle on aimerait mettre plusieurs valeurs ?

Par exemple, les étudiants peuvent avoir plusieurs prénoms, mais le nombre de prénoms n'est pas connu à l'avance.

- 1 Considerer 3 attributs (prenom1, prenom2, prenom3).
  - Combien d'attributs a-t-on besoin ?
  - Comment poser une question sur l'un des prénoms ?
  - Inconvénients : on réserve de la place en trop pour les étudiants qui n'ont qu'un seul prénom.

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Schéma d'une rel.

Schéma de BD

Identifiants

LDD de SQL

Algèbre  
Relationnelle  
+ SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

Comment représenter un attribut multi-valué, c'est à dire une colonne de la table dans laquelle on aimerait mettre plusieurs valeurs ?

Par exemple, les étudiants peuvent avoir plusieurs prénoms, mais le nombre de prénoms n'est pas connu à l'avance.

- 1 Considerer 3 attributs (prenom1, prenom2, prenom3).
  - Combien d'attributs a-t-on besoin ?
  - Comment poser une question sur l'un des prénoms ?
  - Inconvénients : on réserve de la place en trop pour les étudiants qui n'ont qu'un seul prénom.
- 2 Dupliquer les lignes autant de fois qu'il y a de prénoms.
  - inconvénients : duplication des données (problématique si changement de l'adresse par exemple), occupation mémoire.

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Schéma d'une rel.

Schéma de BD

Identifiants

LDD de SQL

Algèbre  
Relationnelle  
+ SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

Comment représenter un attribut multi-valué, c'est à dire une colonne de la table dans laquelle on aimerait mettre plusieurs valeurs ?

Par exemple, les étudiants peuvent avoir plusieurs prénoms, mais le nombre de prénoms n'est pas connu à l'avance.

- 1 Considerer 3 attributs (prenom1, prenom2, prenom3).
  - Combien d'attributs a-t-on besoin ?
  - Comment poser une question sur l'un des prénoms ?
  - Inconvénients : on réserve de la place en trop pour les étudiants qui n'ont qu'un seul prénom.
- 2 Dupliquer les lignes autant de fois qu'il y a de prénoms.
  - inconvénients : duplication des données (problématique si changement de l'adresse par exemple), occupation mémoire.

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Schéma d'une rel.

Schéma de BD

Identifiants

LDD de SQL

Algèbre  
Relationnelle  
+ SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Schéma d'une rel.

Schéma de BD

Identifiants

LDD de SQL

Algèbre  
Relationnelle  
+ SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

Comment représenter un attribut multi-valué, c'est à dire une colonne de la table dans laquelle on aimerait mettre plusieurs valeurs ?

Par exemple, les étudiants peuvent avoir plusieurs prénoms, mais le nombre de prénoms n'est pas connu à l'avance.

- 1 Considerer 3 attributs (prenom1, prenom2, prenom3).
  - Combien d'attributs a-t-on besoin ?
  - Comment poser une question sur l'un des prénoms ?
  - Inconvénients : on réserve de la place en trop pour les étudiants qui n'ont qu'un seul prénom.
- 2 Dupliquer les lignes autant de fois qu'il y a de prénoms.
  - inconvénients : duplication des données (problématique si changement de l'adresse par exemple), occupation mémoire.
- 3 On supprime de la relation Etudiants l'attribut prénom et on crée une autre relation :
  - EtudPrenoms(numEtudiant, prenom)
  - Par contre on ne connaît plus l'ordre des prénoms !

⇒ EtudPrenoms(numEtudiant, prenom, numeroPrenom).

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentielles

Schéma d'une rel.

Schéma de BD

Identifiants

LDD de SQL

Algèbre  
Relationnelle  
+ SQLle modèle  
Entités-  
AssociationsNormalisation  
d'une relation

- le langage de description de données (LDD) permet de décrire la structure logique de la base de données
- Pour le modèle relationnel, SQL (Structured Query Language) est devenu la norme.  
SQL = LDD + LMD
- Ici : uniquement LDD

## ① CREATE TABLE nom\_table

```
(  
  nom_col1 type_col1 [DEFAULT valeur1] [contrainte_col1],  
  nom_col2 type_col2 [DEFAULT valeur2] [contrainte_col2],  
  ...  
  [CONSTRAINT [contrainte_table1]],  
  [CONSTRAINT [contrainte_table2]],  
  ...  
)
```

## Remarques importantes :

- ① Le *nom* de la table doit être choisi avec soin (+ pas de é,à,...)
- ② Chaque colonne porte un *nom*, *unique* dans la table.
  - 2 tables différentes peuvent avoir des colonnes de même nom.
  - à proscrire lorsque la sémantique des colonnes est différente,
  - à conseiller lorsque la sémantique des 2 colonnes est la même.

- 3 `type_col` représente les types de données :
  - types numériques :
    - types numériques exacts (`NUMERIC`  $\equiv$  `DECIMAL`  $\equiv$  `DEC` , `INTEGER`  $\equiv$  `INT` , et `SMALLINT` ), `NUMERIC(5,2)`  $\equiv$  `DEC(5,2)`
    - types approchés (`REAL` , `FLOAT` et `DOUBLE PRECISION` ) : binaire
  - chaînes de caractères : `CHAR(30)` , `VARCHAR(50)` .  $\sim$  `CHAR` complétée à droite avec " ".
  - `TINYBLOB` , `BLOB` , `MEDIUMBLOB` , `LONGBLOB` : Binary Long Object stockage des objets binaires de grande taille. Type pas normalisé.
  - `TINYTEXT` , `TEXT` , `MEDIUMTEXT` , et `LONGTEXT` : du texte
  - Énumération `ENUM` : chaîne dont la valeur est choisie parmi une liste de valeurs autorisées lors de la création de la table. `ENUM("un", "deux", "trois")`
  - `DATE` , `TIME` , `TIMESTAMP` (ou `DATETIME` = `DATE` + `TIME` ) et `YEAR` . (mysql ne vérifie pas qu'il s'agit d'une bonne date, on peut écrire le 31/04/2024 par exemple).

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Schéma d'une rel.

Schéma de BD

Identifiants

LDD de SQL

Algèbre  
Relationnelle  
+ SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

④ **DEFAULT** : permet de spécifier une valeur par défaut. Cette valeur par défaut peut être numérique, une chaîne de caractères, ou bien un mot clef comme **NULL** , **USER** , **CURRENT\_DATE** , **CURRENT\_TIME** ...

⑤ contraintes de colonne :

- contrainte d'obligation : **NOT NULL**
- contrainte d'unicité : **UNIQUE** / **NOT UNIQUE**
- contrainte de clé primaire : **PRIMARY KEY** (identique à **UNIQUE** mais ne peut apparaître qu'une seule fois par table). C'est la colonne privilégiée pour les contraintes référentielles.
- contrainte d'intégrité référentielle : **REFERENCES nom\_table(col)**
- contrainte dite sémantique : **CHECK** suivi d'une contrainte. Exemple :  
**Age INTEGER CHECK (Age < 100)**

⑥ contraintes de tables :

- contrainte d'unicité : **UNIQUE** . Exemple : **UNIQUE (code\_rayon,date\_emprunt)**
- contrainte dite sémantique : **CHECK** . Exemple : **CHECK ((salaire <= 20000 AND age < 30) OR (salaire>30000))**

Il est possible de donner un nom à une contrainte grâce au mot clé **CONSTRAINT** suivi du nom que l'on donne à la contrainte, de telle manière à ce que le nom donné s'affiche en cas de non respect de l'intégrité, c'est-à-dire lorsque la clause que l'on a spécifiée n'est pas validée.

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Schéma d'une rel.

Schéma de BD

Identifiants

LDD de SQL

Algèbre  
Relationnelle  
+ SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

- faire attention aux définitions des contraintes d'intégrité.
- il est possible de définir les comportements du SGBD en cas de modification des données de la table référencée.  
« Que fait-on lorsque l'identifiant d'un  $n$ -uplet est mis à jour ou même lorsqu'il est supprimé ? »

**reference\_definition:**

```
REFERENCES tbl_name [(index_col_name,...)]  
                [ON DELETE reference_option]  
                [ON UPDATE reference_option]
```

**reference\_option:**

```
CASCADE | RESTRICT ou NO ACTION | SET NULL | SET DEFAULT
```

- **CASCADE** indique la suppression/modification en cascade des lignes de la table étrangère dont les clés étrangères correspondent aux clés primaires des lignes effacées.

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Schéma d'une rel.

Schéma de BD

Identifiants

LDD de SQL

Algèbre  
Relationnelle  
+ SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

- faire attention aux définitions des contraintes d'intégrité.
- il est possible de définir les comportements du SGBD en cas de modification des données de la table référencée.  
« Que fait-on lorsque l'identifiant d'un  $n$ -uplet est mis à jour ou même lorsqu'il est supprimé ? »

**reference\_definition:**

```
REFERENCES tbl_name [(index_col_name,...)]  
                [ON DELETE reference_option]  
                [ON UPDATE reference_option]
```

**reference\_option:**

```
CASCADE | RESTRICT ou NO ACTION | SET NULL | SET DEFAULT
```

- **RESTRICT/NO ACTION** indique une erreur en cas d'effacement d'une valeur référencée. **RESTRICT** permet d'interdire la suppression d'un enregistrement référencé par un enregistrement d'une autre relation.

Introduction  
aux BDRJean-Paul  
Comet,  
Nadia  
Abchiche-  
MimouniLes concepts  
générauxNotions  
essentielles

Schéma d'une rel.

Schéma de BD

Identifiants

LDD de SQL

Algèbre  
Relationnelle  
+ SQLle modèle  
Entités-  
AssociationsNormalisation  
d'une relation

- faire attention aux définitions des contraintes d'intégrité.
- il est possible de définir les comportements du SGBD en cas de modification des données de la table référencée.  
« Que fait-on lorsque l'identifiant d'un  $n$ -uplet est mis à jour ou même lorsqu'il est supprimé ? »

**reference\_definition:**

```
REFERENCES tbl_name [(index_col_name,...)]  
                [ON DELETE reference_option]  
                [ON UPDATE reference_option]
```

**reference\_option:**

```
CASCADE | RESTRICT ou NO ACTION | SET NULL | SET DEFAULT
```

- **SET NULL** place la valeur **NULL** dans la ligne de la table étrangère en cas d'effacement d'une valeur correspondant à la clé.

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Schéma d'une rel.

Schéma de BD

Identifiants

LDD de SQL

Algèbre  
Relationnelle  
+ SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

- faire attention aux définitions des contraintes d'intégrité.
- il est possible de définir les comportements du SGBD en cas de modification des données de la table référencée.  
« Que fait-on lorsque l'identifiant d'un  $n$ -uplet est mis à jour ou même lorsqu'il est supprimé ? »

**reference\_definition:**

```
REFERENCES tbl_name [(index_col_name,...)]  
                [ON DELETE reference_option]  
                [ON UPDATE reference_option]
```

**reference\_option:**

```
CASCADE | RESTRICT ou NO ACTION | SET NULL | SET DEFAULT
```

- **SET DEFAULT** place la valeur par défaut (qui suit ce paramètre) dans la ligne de la table étrangère en cas d'effacement d'une valeur correspondant à la clé.

Introduction  
aux BDR

Jean-Paul  
Comet,  
Nadia  
Abchiche-  
Mimouni

Les concepts  
généraux

Notions  
essentielles

Schéma d'une rel.

Schéma de BD

Identifiants

LDD de SQL

Algèbre  
Relationnelle  
+ SQL

le modèle  
Entités-  
Associations

Normalisation  
d'une relation

- 1 Lorsque l'identifiant est composé de plusieurs attributs :
  - définition de la clé primaire : **PRIMARY KEY** (Cf. infra)
  - Ces colonnes ne peuvent plus prendre la valeur **NULL** et doivent être telles que deux lignes ne puissent avoir simultanément la même combinaison de valeurs pour ces colonnes.
  - Exemple : **PRIMARY KEY (colonne1, colonne2, ...)**
- 2 Définition d'une clé étrangère de plusieurs attributs :
  - **FOREIGN KEY + REFERENCES**
  - Exemple :  
**FOREIGN KEY (col1, col2, ...)**  
**REFERENCES Nom\_de\_la\_table\_ref(col1,col2,...)**