

Introduction
aux BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts
généraux

Notions
essentielles

Algèbre
Relationnelle
+ SQL

le modèle
Entités-
Associations

NoSQL

Normalisation
d'une relation

Introduction aux Bases de données Relationnelles

Département Génie Biologique
GB4 – année 2025–2026



Jean-Paul Comet¹
Nadia Abchiche-Mimouni¹

¹Université Côte d'Azur

11 février 2026

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQLle modèle
Entités-
Associations

NoSQL

Normalisation
d'une relation

- Cours : 3 séances de 2h + 5 séances de 1h30
- TDs : 7 séances de 1h30 + 3 séances de 1h (2 groupes)
- Intervenants :

Ingrid Grenet	<code>ingrid.grenet@univ-cotedazur.fr</code>
Gilles Bernot	<code>Gilles.Bernot@univ-cotedazur.fr</code>
Guillaume Grataloup	<code>guillaume.grataloup@etu.unice.fr</code>
- Evaluation : Un examen final de 3h le 24 mai 9h–12h + un projet
Note finale = 1/3 projet + 2/3 examen
- Demande accès VPN : <https://formvpn.polytech.unice.fr/>
- [Lien vers la page du Cours](#)

Introduction
aux BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts
généraux

Notions
essentielles

Algèbre
Relationnelle
+ SQL

le modèle
Entités-
Associations

NoSQL

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 NoSQL
- 6 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

NoSQL

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 NoSQL
- 6 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

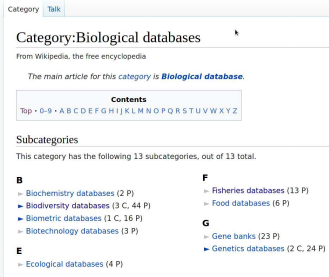
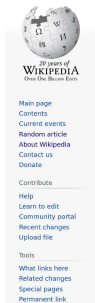
le modèle Entités-Associations

NoSQL

Normalisation d'une relation

- Lieux de stockages d'informations biologiques.
- *Nucleic Acids Research* 2018 : liste de 180 bases de données !
- Le site wikipédia classifie ces bases de données :
https://en.wikipedia.org/wiki/List_of_biological_databases

► wikipedia



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

NoSQL

Normalisation
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 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

NoSQL

Normalisation
d'une relation

- Structures tridimensionnelles :
 - **PDB** : structure 3D Protein Data Bank, 199.000 entrées
▶ [PDB](#)
 - **MMDB** : molecular Modeling Database (christallographie 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 Nucleotid 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

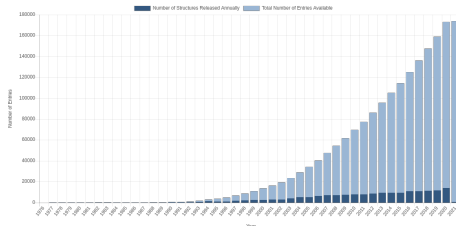
Algèbre
Relationnelle
+ SQL

le modèle
Entités-
Associations

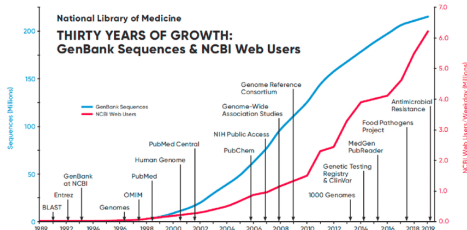
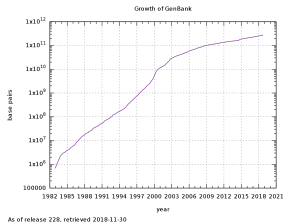
NoSQL

Normalisation
d'une relation

PDB :



GenBank :



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

NoSQL

Normalisation
d'une relation

- ❶ **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.*
- ❷ **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 Managment 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-
Associations

NoSQL

Normalisation
d'une relation

- ③ **Fichier** : Lorsque l'ensemble des données est homogène (ensemble des abonnées à une revue, le personnel d'une entreprise, les produits vendus en ligne...)
- **fichier** des abonnées, 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 BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts
généraux

BD publiques
Intro aux SGBD
Cycle de vie
Architecture

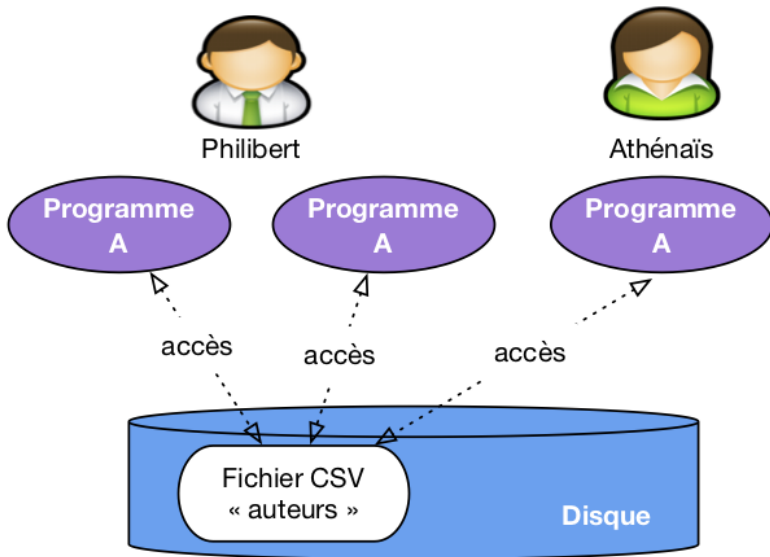
Notions
essentielles

Algèbre
Relationnelle
+ SQL

le modèle
Entités-
Associations

NoSQL

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

NoSQL

Normalisation
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
essentiellesAlgèbre
Relationnelle
+ SQLle modèle
Entités-
Associations

NoSQL

Normalisation
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 BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxBD publiques
Intro aux SGBD
Cycle de vie
ArchitectureNotions
essentiellesAlgèbre
Relationnelle
+ SQLle modèle
Entités-
Associations

NoSQL

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 BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxBD publiques
Intro aux SGBD
Cycle de vie
ArchitectureNotions
essentiellesAlgèbre
Relationnelle
+ SQLle modèle
Entités-
Associations

NoSQL

Normalisation
d'une relation

- ❶ 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
- ❷ BD Déductives : utilisation d'un langage logique (prolog).
Exemple : Pascale ; Laurent ; Louise ;
apourpere(Pascale, Louis) ;
apourpere(Louis, Romuald) ;

Quelles sont les ancêtres de Pascale ?
- ❸ 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-
Associations

NoSQL

Normalisation
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 ne 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 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

NoSQL

Normalisation
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^{ère} 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.

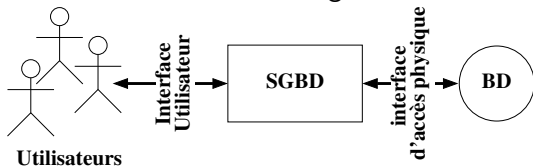
- **Langage de manipulation des données (LMD) :**
 - requêtes d'interrogation
 - requêtes de mise à jour (ajout, suppression, modification).

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

NoSQL

Normalisation
d'une relation

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 BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxBD publiques
Intro aux SGBD
Cycle de vie
ArchitectureNotions
essentiellesAlgèbre
Relationnelle
+ SQLle modèle
Entités-
Associations

NoSQL

Normalisation
d'une relation

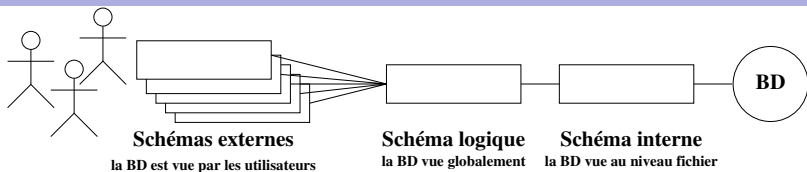
- Rappel : les différentes classes d'utilisateurs ne voient pas la BD de la même manière (vues).

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

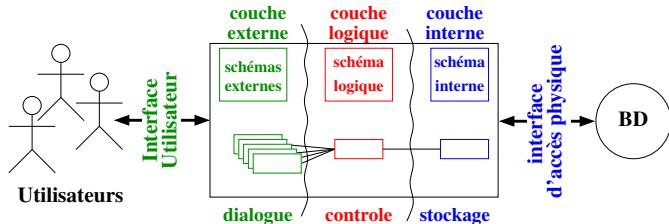
⇒ l'architecture doit être affinée : 3 couches



- 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, ...

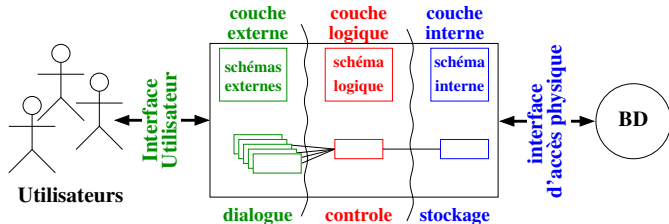
Niveau logique : *ne dépendre ni des exigences des utilisateurs, ni des structures physiques choisies.*

SGBD

**1** Une requête arrive :

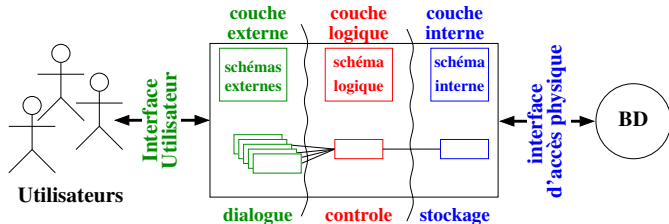
- exprimée dans un langage accepté par le système (LMD)
- analysée du point de vue syntaxique (conformité à la grammaire),
- puis du point de vue sémantique (les objets cités doivent être connus dans le schéma externe de l'utilisateur).
- validation au niveau externe

SGBD



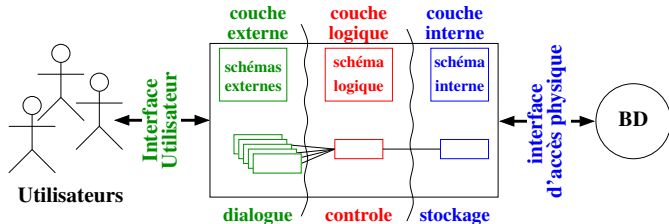
- ② Traduction et passage de la requête à la couche logique
- les références aux objets du schéma externe sont remplacés par des références aux objets correspondants dans le schéma logique.
- Utilisation de règles de correspondance entre schémas logique et externe (règles définies au moment de la définition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

SGBD



- 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).
- ③ **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.

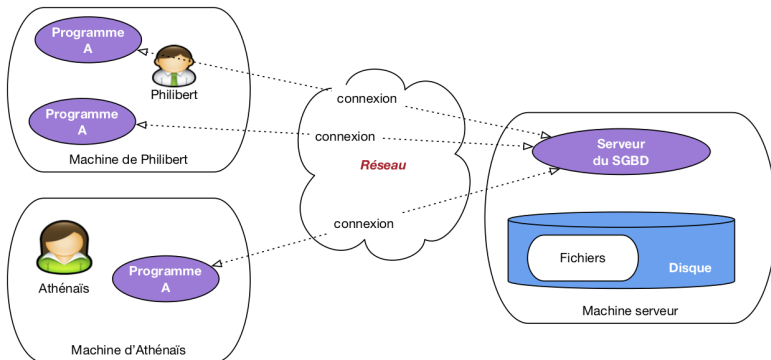
SGBD



- 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-
Associations

NoSQL

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

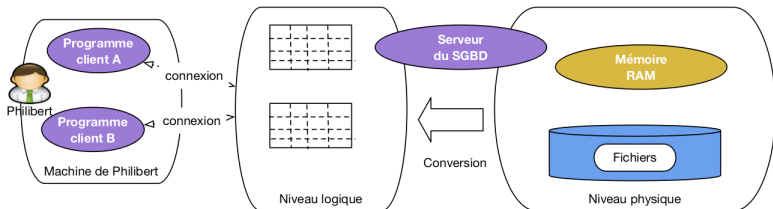
Notions
essentielles

Algèbre
Relationnelle
+ SQL

le modèle
Entités-
Associations

NoSQL

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

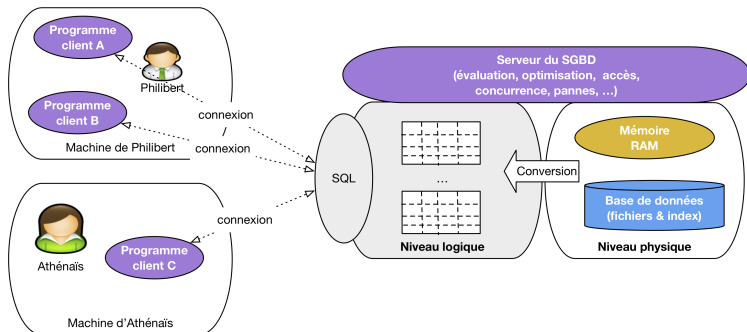
Notions
essentielles

Algèbre
Relationnelle
+ SQL

le modèle
Entités-
Associations

NoSQL

Normalisation
d'une relation



Quel SGBD ?

January 20

Introduction aux BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts généraux

BD publiques
Intro aux SGBD
Cycle de vie
Architecture

Notions essentielles

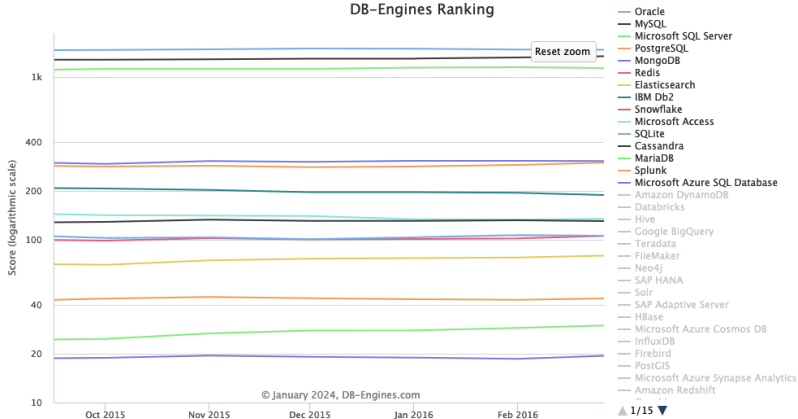
Algèbre
Relationnelle
+ SQL

le modèle
Entités-
Associations

NoSQL

Normalisation
d'une relation

DB-Engines Ranking



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

NoSQL

Normalisation
d'une relation

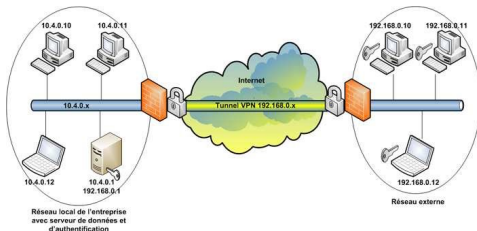
Un VPN est un réseau privé qui utilise un réseau public comme backbone

- 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

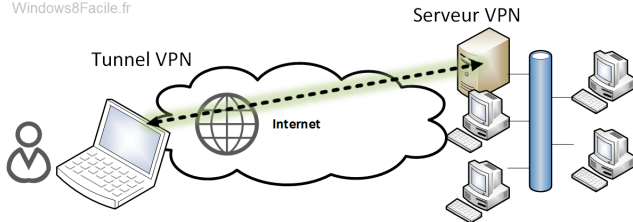
- Seuls les utilisateurs ou les groupes qui sont enregistrés dans ce vpn peuvent y accéder.
- Les données transitent dans un tunnel après avoir été chiffrées.
- Tout se passe comme si la connexion se faisait en dehors d'infrastructure d'accès partagé comme Internet
- Étanchéité du trafic entre les différents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



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-
Associations

NoSQL

Normalisation
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 :	open.unice.fr
Group :	UNS
Username :	login@etu.polytech
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

NoSQL

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 NoSQL
- 6 Normalisation d'une relation

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

Schéma d'une rel.

Schéma de BD

Identifiants

LDD de SQL

Algèbre
Relationnelle
+ SQLle modèle
Entités-
Associations

NoSQL

Normalisation
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. $T_1 \sqcap T_2 \longrightarrow T_3$
- 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).

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-
Associations

NoSQL

Normalisation
d'une relation

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

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

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

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
 $Film = \{titreFilm, realiseur, acteur, producteur\}$
- Schéma de la relation
 $Projection = \{titreFilm, nomCinema, horaire\}$

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
:	:	:

n -uplet sur un ensemble E d'attributs.

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"

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 : $\{\underline{a}, \underline{b}\} \neq \{\underline{a}, \underline{b}\}$).

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.

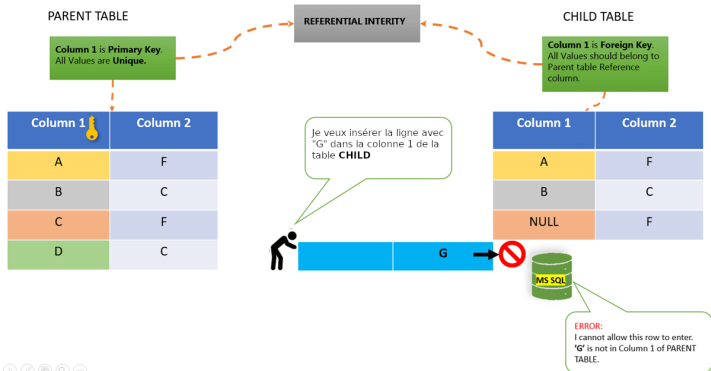
Définition d'un schéma d'une relation :

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

NoSQL

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.

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

NoSQL

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

NoSQL

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

NoSQL

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

NoSQL

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

NoSQL

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-
Associations

NoSQL

Normalisation
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.
$$\text{SQL} = \text{LDD} + \text{LMD}$$
- Ici : uniquement LDD

❶ 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 :

- ❶ Le *nom* de la table doit être choisi avec soin (+ pas de é,à,...)
- ❷ 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.

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

Schéma d'une rel.

Schéma de BD

Identifiants

LDD de SQL

Algèbre
Relationnelle
+ SQLle modèle
Entités-
Associations

NoSQL

Normalisation
d'une relation

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 31/04/2026 par exemple).

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-
Associations

NoSQL

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 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-
Associations

NoSQL

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

NoSQL

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-
Associations

NoSQL

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 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

NoSQL

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

NoSQL

Normalisation
d'une relation

- ❶ 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, ...)**
- ❷ Définition d'une clé étrangère de plusieurs attributs :
 - **FOREIGN KEY + REFERENCES**
 - Exemple :
FOREIGN KEY (col1, col2, ...)
REFERENCES Nom_de_la_table_ref(col1,col2,...)

Introduction
aux BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts
généraux

Notions
essentiels

Algèbre
Relationnelle
+ SQL

Opérateurs unaires
Traduction en SQL

Jointures

Opérateurs
ensemblistes

Opérateurs hors de
l'Alg.

Autres ordres SQL

le modèle
Entités-
Associations

NoSQL

Normalisation
d'une relation

- 1 Les concepts généraux
- 2 Notions essentielles pour le modèle relationnel
- 3 Algèbre Relationnelle et SQL
 - Les opérateurs de base unaires (sur une seule opérande)
 - Traduction en SQL d'une requête simple
 - Les opérateurs de jointure
 - Opérateurs ensemblistes : $\cup$, $\cap$, $\setminus$
 - Opérateurs de SQL qui ne sont pas dans l'algèbre
 - Autres ordres SQL
- 4 Le modèle conceptuel Entités-Associations
- 5 NoSQL
- 6 Normalisation d'une relation

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQLOpérateurs unaires
Traduction en SQL
JointuresOpérateurs
ensemblistesOpérateurs hors de
l'Alg.

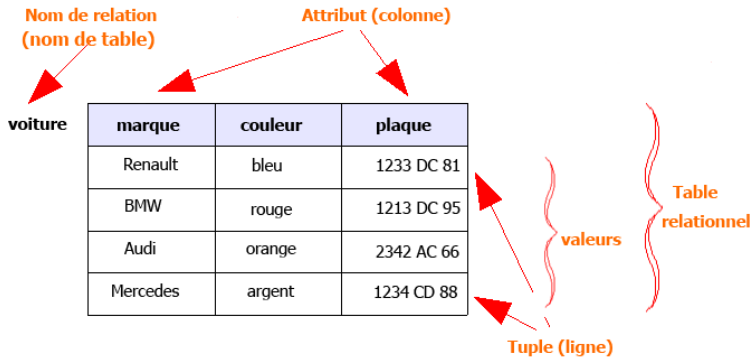
Autres ordres SQL

le modèle
Entités-
Associations

NoSQL

Normalisation
d'une relation

- Le modèle relationnel repose sur la notion de relation assimilable à un tableau.



- les langages classiques (python) manipulent des tableaux. Ils pourraient servir à rechercher les informations dans ce genre de structures de données.
- ⇒ Pourquoi alors avoir introduit des langages de manipulation de données relationnelles ?

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQL

Opérateurs unaires

Traduction en SQL

Jointures

Opérateurs
ensemblistesOpérateurs hors de
l'Alg.

Autres ordres SQL

le modèle

Entités-
Associations

NoSQL

Normalisation
d'une relation

⇒ Pourquoi alors avoir introduit des langages de manipulation de données relationnelles ?

- 1 Pour qu'un programmeur puisse effectuer une recherche à l'aide d'un langage classique impératif, il est nécessaire qu'il connaisse la structure de la relation.

- La couleur est-elle en 6ème colonne ou en 7ème ?

contraire au principe de *séparation des programmes et des données*

- 2 Les utilisateurs doivent connaître l'organisation des données (tri, existence d'index, ...) pour que leurs requêtes soient efficaces.

- comment aller rapidement au véhicule d'immatriculation FD-446-QM ?

si la structure change, les programmes doivent aussi changer...

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQL

Opérateurs unaires

Traduction en SQL

Jointures

Opérateurs
ensemblistesOpérateurs hors de
l'Alg.

Autres ordres SQL

le modèle
Entités-
Associations

NoSQL

Normalisation
d'une relation

- Les **Langages de Manipulation de données** doivent donc être le plus proche du modèle conceptuel :
 - ne porte que sur les concepts du schéma : relation, attributs, dom.
 - doivent ignorer complètement l'organisation interne des relations.
 - Le modèle relationnel a été proposé avec deux LMD de base :
 - l'**algèbre relationnel** et
 - le **calcul relationnel** équivalents en puissance
- SQL implémente les relations non pas comme des ensembles mais comme des **multi-ensembles** (il peut y avoir redondance). Ce choix a été fait pour des questions d'efficacité.
- L'**algèbre Relationnelle** repose sur un ens d'opérateurs $R_1 \square R_2 \rightarrow R_{\text{temp}}$
 - 5 opérateurs de base : *sélection, projection, union, différence, produit*
1 opérateur syntaxique : renommer (modifie le schéma/pas les n -uplets).
 - A partir de ces opérateurs, d'autres opérateurs ont été introduits :
 - ils ne sont que la composée de plusieurs opérateurs de base (raccourcis).
 - les plus fréquents : *intersection, jointure naturelle, théta jointure, division*.
 - Autre classification des opérateurs :
 - les opérateurs basés sur la théorie des ensembles : $\cup, \cap, \setminus, \times$
 - les opérateurs construits spécifiquement pour les BD relationnelles :
sélection, projection, jointure, division et renommage

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQLOpérateurs unaires
Traduction en SQL

Jointures

Opérateurs
ensemblistesOpérateurs hors de
l'Alg.

Autres ordres SQL

le modèle
Entités-
Associations

NoSQL

Normalisation
d'une relation

Cet opérateur construit une relation dans laquelle n'apparaissent que certains attributs de la relation opérande.

Intuition : on extrait uniquement certaines colonnes du tableau

Soit $R(A_1, A_2, \dots, A_n)$ une relation et soient $\{A_{i_1}, A_{i_2}, \dots, A_{i_j}\}$ un sous ensemble de ses attributs. La projection de R sur $\{A_{i_1}, A_{i_2}, \dots, A_{i_j}\}$, notée $\pi[A_{i_1}, A_{i_2}, \dots, A_{i_j}]R$ (π comme projection) est une nouvelle relation, temporaire, de schéma $(A_{i_1}, A_{i_2}, \dots, A_{i_j})$ et de population égale à

$$\{ t \mid \exists r : (r \in R) \wedge (t.A_{i_1} = r.A_{i_1}) \wedge \dots (t.A_{i_j} = r.A_{i_j}) \}$$

Remarque. Il s'agit d'une opération *ensembliste*. Donc si l'opération crée des doublons, ils sont automatiquement éliminés.

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQL

Opérateurs unaires

Traduction en SQL

Jointures

Opérateurs
ensemblistesOpérateurs hors de
l'Alg.

Autres ordres SQL

le modèle

Entités-
Associations

NoSQL

Normalisation
d'une relation**Exemple.** Soit la relation *Personne* suivante :

nom	prénom	jour	mois	année	genre
Jean	Thierry	29	02	1960	M
Talon	Achille	20	11	1978	M
Rochat	Dominique	07	12	1956	M
Martin	Joëlle	01	05	1968	F
Martin	Jules	03	03	1984	M
Rochat	Dominique	17	02	1976	F

 $\pi[\text{nom}, \text{prénom}] \textit{Personne}$ $\pi[\text{nom}, \text{année}] \textit{Personne}$

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQL

Opérateurs unaires

Traduction en SQL

Jointures

Opérateurs
ensemblistesOpérateurs hors de
l'Alg.

Autres ordres SQL

le modèle

Entités-
Associations

NoSQL

Normalisation
d'une relation**Exemple.** Soit la relation *Personne* suivante :

nom	prénom	jour	mois	année	genre
Jean	Thierry	29	02	1960	M
Talon	Achille	20	11	1978	M
Rochat	Dominique	07	12	1956	M
Martin	Joëlle	01	05	1968	F
Martin	Jules	03	03	1984	M
Rochat	Dominique	17	02	1976	F

 $\pi[\text{nom}, \text{prénom}] \textit{Personne}$ $\pi[\text{nom}, \text{année}] \textit{Personne}$

Introduction aux BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts généraux

Notions essentielles

Algèbre Relationnelle + SQL

Opérateurs unaires Traduction en SQL

Jointures

Opérateurs ensemblistes

Opérateurs hors de l'Alg.

Autres ordres SQL

le modèle

Entités- Associations

NoSQL

Normalisation d'une relation

Exemple. Soit la relation *Personne* suivante :

nom	prénom	jour	mois	année	genre
Jean	Thierry	29	02	1960	M
Talon	Achille	20	11	1978	M
Rochat	Dominique	07	12	1956	M
Martin	Joëlle	01	05	1968	F
Martin	Jules	03	03	1984	M
Rochat	Dominique	17	02	1976	F

$\pi[\text{nom}, \text{prénom}] \textit{Personne}$

nom	prénom
Jean	Thierry
Talon	Achille
Rochat	Dominique
Martin	Joëlle
Martin	Jules

$\pi[\text{nom}, \text{année}] \textit{Personne}$

Introduction aux BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts généraux

Notions essentielles

Algèbre Relationnelle + SQL

Opérateurs unaires Traduction en SQL

Jointures

Opérateurs ensemblistes

Opérateurs hors de l'Alg.

Autres ordres SQL

le modèle

Entités- Associations

NoSQL

Normalisation d'une relation

Exemple. Soit la relation *Personne* suivante :

nom	prénom	jour	mois	année	genre
Jean	Thierry	29	02	1960	M
Talon	Achille	20	11	1978	M
Rochat	Dominique	07	12	1956	M
Martin	Joëlle	01	05	1968	F
Martin	Jules	03	03	1984	M
Rochat	Dominique	17	02	1976	F

$\pi[\text{nom}, \text{prénom}] \textit{Personne}$

nom	prénom
Jean	Thierry
Talon	Achille
Rochat	Dominique
Martin	Joëlle
Martin	Jules

$\pi[\text{nom}, \text{année}] \textit{Personne}$

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQLOpérateurs unaires
Traduction en SQL

Jointures

Opérateurs
ensemblistesOpérateurs hors de
l'Alg.

Autres ordres SQL

le modèle
Entités-
Associations

NoSQL

Normalisation
d'une relation**Exemple.** Soit la relation *Personne* suivante :

nom	prénom	jour	mois	année	genre
Jean	Thierry	29	02	1960	M
Talon	Achille	20	11	1978	M
Rochat	Dominique	07	12	1956	M
Martin	Joëlle	01	05	1968	F
Martin	Jules	03	03	1984	M
Rochat	Dominique	17	02	1976	F

 $\pi[\text{nom}, \text{prénom}] \textit{Personne}$

nom	prénom
Jean	Thierry
Talon	Achille
Rochat	Dominique
Martin	Joëlle
Martin	Jules

 $\pi[\text{nom}, \text{année}] \textit{Personne}$

nom	année
Jean	1960
Talon	1978
Rochat	1956
Martin	1968
Martin	1984
Rochat	1976

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQL

Opérateurs unaires

Traduction en SQL

Jointures

Opérateurs
ensemblistesOpérateurs hors de
l'Alg.

Autres ordres SQL

le modèle

Entités-
Associations

NoSQL

Normalisation
d'une relation

Cet opérateur construit une relation dans laquelle seuls certains n -uplets apparaissent.

Intuition : On sélectionne des lignes (n -uplets) qui satisfont une certaine propriété, appelé **prédicat de sélection**

Soit $R(A_1, A_2, \dots, A_n)$ une relation la sélection selon un prédicat p , notée $\sigma[p]R$ (σ comme sélection) est une nouvelle relation, temporaire, de schéma identique à celui de R et de population égale à l'ensemble des n -uplets de R pour lesquels le prédicat p est vrai.

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQLOpérateurs unaires
Traduction en SQL

Jointures

Opérateurs
ensemblistesOpérateurs hors de
l'Alg.

Autres ordres SQL

le modèle

Entités-
Associations

NoSQL

Normalisation
d'une relation

nom	prénom	jour	mois	année	genre
Jean	Thierry	29	02	1960	M
Talon	Achille	20	11	1978	M
Rochat	Dominique	07	12	1956	M
Martin	Joëlle	01	05	1968	F
Martin	Jules	03	03	1984	M
Rochat	Dominique	17	02	1976	F

Exemple. A partir de la relation *Personne* on peut créer la relation

Femme $:= \sigma[\text{genre} = \text{'F'}] \text{Personne}$

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQL

Opérateurs unaires

Traduction en SQL

Jointures

Opérateurs
ensemblistesOpérateurs hors de
l'Alg.

Autres ordres SQL

le modèle

Entités-
Associations

NoSQL

Normalisation
d'une relation

nom	prénom	jour	mois	année	genre
Jean	Thierry	29	02	1960	M
Talon	Achille	20	11	1978	M
Rochat	Dominique	07	12	1956	M
Martin	Joëlle	01	05	1968	F
Martin	Jules	03	03	1984	M
Rochat	Dominique	17	02	1976	F

Exemple. A partir de la relation *Personne* on peut créer la relation

Femme $:= \sigma[\text{genre} = \text{'F'}] \text{Personne}$

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellAlgèbre
Relationnelle
+ SQLOpérateurs unaires
Traduction en SQL

Jointures

Opérateurs
ensemblistesOpérateurs hors de
l'Alg.

Autres ordres SQL

le modèle
Entités-
Associations

NoSQL

Normalisation
d'une relation

nom	prénom	jour	mois	année	genre
Jean	Thierry	29	02	1960	M
Talon	Achille	20	11	1978	M
Rochat	Dominique	07	12	1956	M
Martin	Joëlle	01	05	1968	F
Martin	Jules	03	03	1984	M
Rochat	Dominique	17	02	1976	F

Exemple. A partir de la relation *Personne* on peut créer la relation

Femme $:= \sigma[\text{genre} = 'F']\text{Personne}$

nom	prénom	jour	mois	année	genre
Martin	Joëlle	01	05	1968	F
Rochat	Dominique	17	02	1976	F

Syntaxe du prédicat.

$\langle p \rangle$	::=	$\langle \text{condition} \rangle$	
		$\langle p \rangle \langle \text{opérateur logique} \rangle \langle p \rangle$	
		$\neg \langle p \rangle$	négation
		$(\langle p \rangle \vee \langle p \rangle)$	disjonction
$\langle \text{opérateur logique} \rangle$	::=	$\vee$	
		$\wedge$	conjonction
		$\text{Attr } \langle \text{opérateur comp} \rangle \text{ valeur}$	
		$\text{Attr } \langle \text{opérateur comp} \rangle \text{ Attr}$	
$\langle \text{condition} \rangle$	::=	$\text{Attr } \langle \text{opérateur comp} \rangle \text{ valeur}$	
		$\text{Attr } \langle \text{opérateur comp} \rangle \text{ Attr}$	
$\langle \text{opérateur comp} \rangle$	::=	$= \mid \neq \mid \leq \mid < \mid > \mid \geq$	

Remarques.

- Les opérateurs de comparaisons $\leq, <, >, \geq$ ne peuvent s'appliquer que sur des attributs dont le domaine est muni d'un ordre :
 - valeurs numériques,
 - dates,
 - chaînes de caractères (ordre alphanumérique $\sim$ alphabétique généralisé)

Pour les domaines sans ordre, on ne peut utiliser que les opérateurs $=$ et $\neq$.
- Lorsque l'on compare deux attributs, il faut s'assurer que les deux attributs aient même domaine.

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQLOpérateurs unaires
Traduction en SQL

Jointures

Opérateurs
ensemblistesOpérateurs hors de
l'Alg.

Autres ordres SQL

le modèle

Entités-
Associations

NoSQL

Normalisation
d'une relation

nom	prénom	jour	mois	année	genre
Jean	Thierry	29	02	1960	M
Talon	Achille	20	11	1978	M
Rochat	Dominique	07	12	1956	M
Martin	Joëlle	01	05	1968	F
Martin	Jules	03	03	1984	M
Rochat	Dominique	17	02	1976	F

Trouver la liste des noms et prénoms de tous les hommes nés avant 1975.

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQL

Opérateurs unaires

Traduction en SQL

Jointures

Opérateurs
ensemblistesOpérateurs hors de
l'Alg.

Autres ordres SQL

le modèle

Entités-
Associations

NoSQL

Normalisation
d'une relation

nom	prénom	jour	mois	année	genre
Jean	Thierry	29	02	1960	M
Talon	Achille	20	11	1978	M
Rochat	Dominique	07	12	1956	M
Martin	Joëlle	01	05	1968	F
Martin	Jules	03	03	1984	M
Rochat	Dominique	17	02	1976	F

Trouver la liste des noms et prénoms de tous les hommes nés avant 1975.

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQLOpérateurs unaires
Traduction en SQL

Jointures

Opérateurs
ensemblistesOpérateurs hors de
l'Alg.

Autres ordres SQL

le modèle

Entités-
Associations

NoSQL

Normalisation
d'une relation

nom	prénom	jour	mois	année	genre
Jean	Thierry	29	02	1960	M
Talon	Achille	20	11	1978	M
Rochat	Dominique	07	12	1956	M
Martin	Joëlle	01	05	1968	F
Martin	Jules	03	03	1984	M
Rochat	Dominique	17	02	1976	F

Trouver la liste des noms et prénoms de tous les hommes nés avant 1975.

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQLOpérateurs unaires
Traduction en SQL

Jointures

Opérateurs
ensemblistesOpérateurs hors de
l'Alg.

Autres ordres SQL

le modèle

Entités-
Associations

NoSQL

Normalisation
d'une relation

nom	prénom	jour	mois	année	genre
Jean	Thierry	29	02	1960	M
Talon	Achille	20	11	1978	M
Rochat	Dominique	07	12	1956	M
Martin	Joëlle	01	05	1968	F
Martin	Jules	03	03	1984	M
Rochat	Dominique	17	02	1976	F

Trouver la liste des noms et prénoms de tous les hommes nés avant 1975.

nom	prénom
Jean	Thierry
Rochat	Dominique

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQLOpérateurs unaires
Traduction en SQL

Jointures

Opérateurs
ensemblistesOpérateurs hors de
l'Alg.

Autres ordres SQL

le modèle

Entités-
Associations

NoSQL

Normalisation
d'une relation

nom	prénom	jour	mois	année	genre
Jean	Thierry	29	02	1960	M
Talon	Achille	20	11	1978	M
Rochat	Dominique	07	12	1956	M
Martin	Joëlle	01	05	1968	F
Martin	Jules	03	03	1984	M
Rochat	Dominique	17	02	1976	F

Trouver la liste des noms et prénoms de tous les hommes nés avant 1975.

$$H := \pi \left[\begin{array}{c} \text{nom} \\ \text{prénom} \end{array} \right] \left(\sigma \left[\begin{array}{c} \text{genre} = 'M' \wedge \\ \text{année} < 1975 \end{array} \right] \text{Personne} \right)$$

nom	prénom
Jean	Thierry
Rochat	Dominique

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQL

Opérateurs unaires

Traduction en SQL

Jointures

Opérateurs
ensemblistesOpérateurs hors de
l'Alg.

Autres ordres SQL

le modèle

Entités-
Associations

NoSQL

Normalisation
d'une relation

L'opérateur de renommage α permet de changer le nom d'un ou plusieurs attribut(s) d'une relation R .

$\alpha[Attr1 : newNameForAttr1, ...]R$

Remarque. Cet opérateur est utile lorsqu'il y a un problème d'homonymie ou de synonymie, ou alors avant des opérations ensemblistes (union, différence, intersection) qui nécessite que les attributs correspondants aient le même nom.

Exemple.

$$\alpha \left[\begin{array}{l} \text{nom:n} \\ \text{prénom:p} \end{array} \right] \pi \left[\begin{array}{l} \text{nom} \\ \text{prénom} \end{array} \right] \left(\sigma \left[\begin{array}{l} \text{genre='M'} \wedge \\ \text{année} < 1975 \end{array} \right] \text{Personne} \right)$$

nom	prénom
Jean	Thierry
Rochat	Dominique

$\Rightarrow$

n	p
Jean	Thierry
Rochat	Dominique

Introduction
aux BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts
généraux

Notions
essentielles

Algèbre
Relationnelle
+ SQL

Opérateurs unaires
Traduction en SQL

Jointures

Opérateurs
ensemblistes

Opérateurs hors de
l'Alg.

Autres ordres SQL

le modèle

Entités-
Associations

NoSQL

Normalisation
d'une relation

ordre **SELECT**

SELECT $A_1, \dots, A_N$ FROM R WHERE C ;

- $A_1, \dots, A_N$: attributs
- R : nom d'une relation
- C : condition de sélection sur les n -uplets de R , par exemple comme la sélection de l'algèbre relationnelle.

Le **SELECT** fait une **projection** (opération algébrique) du résultat de la **sélection** sur les attributs $A_1, \dots, A_N$.

Introduction
aux BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts
généraux

Notions
essentielles

Algèbre
Relationnelle
+ SQL

Opérateurs unaires

Traduction en SQL

Jointures

Opérateurs
ensemblistes

Opérateurs hors de
l'Alg.

Autres ordres SQL

le modèle

Entités-
Associations

NoSQL

Normalisation
d'une relation

Exemple. Considérons le schéma de base de données suivant.

$$S = \left\{ \begin{array}{l} \text{Film}(\text{Titre}, \text{date}, \text{longueur}, \text{couleur}, \text{nomStudio}, \text{IdProd}), \\ \text{Joue}(\text{TitreFilm}, \text{FilmDate}, \text{nomActeur}, \text{Paye}), \\ \text{Acteur}(\text{nom}, \text{adresse}, \text{genre}, \text{dateNaissance}), \\ \text{Studio}(\text{nom}, \text{Adresse}), \\ \text{Producteur}(\text{nom}, \text{adresse}, \text{Id}) \end{array} \right\}$$

- 1 Quels sont les titre et longueur des films produits par le studio Disney en 1999 ?
- 2 On peut remplacer la suite exhaustive des attributs d'une table par le symbole $*$:

Introduction
aux BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts
généraux

Notions
essentielles

Algèbre
Relationnelle
+ SQL

Opérateurs unaires
Traduction en SQL

Jointures

Opérateurs
ensemblistes

Opérateurs hors de
l'Alg.

Autres ordres SQL

le modèle

Entités-
Associations

NoSQL

Normalisation
d'une relation

Exemple. Considérons le schéma de base de données suivant.

$$S = \left\{ \begin{array}{l} \text{Film}(\text{Titre}, \text{date}, \text{longueur}, \text{couleur}, \text{nomStudio}, \text{IdProd}), \\ \text{Joue}(\text{TitreFilm}, \text{FilmDate}, \text{nomActeur}, \text{Paye}), \\ \text{Acteur}(\text{nom}, \text{adresse}, \text{genre}, \text{dateNaissance}), \\ \text{Studio}(\text{nom}, \text{Adresse}), \\ \text{Producteur}(\text{nom}, \text{adresse}, \text{Id}) \end{array} \right\}$$

- ❶ Quels sont les titre et longueur des films produits par le studio Disney en 1999 ?

```
SELECT Titre, longueur FROM Film  
WHERE nomStudio='Disney' and Date=1999;
```

- ❷ On peut remplacer la suite exhaustive des attributs d'une table par le symbole * :

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentielsAlgèbre
Relationnelle
+ SQL

Opérateurs unaires

Traduction en SQL

Jointures

Opérateurs
ensemblistesOpérateurs hors de
l'Alg.

Autres ordres SQL

le modèle

Entités-
Associations

NoSQL

Normalisation
d'une relation**Exemple.** Considérons le schéma de base de données suivant.

$$S = \left\{ \begin{array}{l} \text{Film}(\text{Titre}, \text{date}, \text{longueur}, \text{couleur}, \text{nomStudio}, \text{IdProd}), \\ \text{Joue}(\text{TitreFilm}, \text{FilmDate}, \text{nomActeur}, \text{Paye}), \\ \text{Acteur}(\text{nom}, \text{adresse}, \text{genre}, \text{dateNaissance}), \\ \text{Studio}(\text{nom}, \text{Adresse}), \\ \text{Producteur}(\text{nom}, \text{adresse}, \text{Id}) \end{array} \right\}$$

- ❶ Quels sont les titre et longueur des films produits par le studio Disney en 1999 ?

SELECT Titre, longueur **FROM** Film
WHERE nomStudio='Disney' and Date=1999;

$$\pi \left[\begin{array}{c} \text{Titre} \\ \text{Long} \end{array} \right] \sigma \left[\begin{array}{c} \text{nomStudio} = ' \text{Disney}' \wedge \\ \text{Date} = 1999; \end{array} \right] \text{Film}$$

- ❷ On peut remplacer la suite exhaustive des attributs d'une table par le symbole * :

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentielsAlgèbre
Relationnelle
+ SQL

Opérateurs unaires

Traduction en SQL

Jointures

Opérateurs
ensemblistesOpérateurs hors de
l'Alg.

Autres ordres SQL

le modèle

Entités-
Associations

NoSQL

Normalisation
d'une relation**Exemple.** Considérons le schéma de base de données suivant.

$$S = \left\{ \begin{array}{l} \text{Film}(\text{Titre}, \text{date}, \text{longueur}, \text{couleur}, \text{nomStudio}, \text{IdProd}), \\ \text{Joue}(\text{TitreFilm}, \text{FilmDate}, \text{nomActeur}, \text{Paye}), \\ \text{Acteur}(\text{nom}, \text{adresse}, \text{genre}, \text{dateNaissance}), \\ \text{Studio}(\text{nom}, \text{Adresse}), \\ \text{Producteur}(\text{nom}, \text{adresse}, \text{Id}) \end{array} \right\}$$

- ❶ Quels sont les titre et longueur des films produits par le studio Disney en 1999 ?

```
SELECT Titre, longueur FROM Film
WHERE nomStudio='Disney' and Date=1999;
```

$$\pi \left[\begin{array}{c} \text{Titre} \\ \text{Long} \end{array} \right] \sigma \left[\begin{array}{c} \text{nomStudio} = ' \text{Disney}' \wedge \\ \text{Date} = 1999; \end{array} \right] \text{Film}$$

- ❷ On peut remplacer la suite exhaustive des attributs d'une table par le symbole * :

```
SELECT * FROM Film
WHERE nomStudio='Disney' and Date=1999;
```

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQL

Opérateurs unaires

Traduction en SQL

Jointures

Opérateurs
ensemblistesOpérateurs hors de
l'Alg.

Autres ordres SQL

le modèle

Entités-
Associations

NoSQL

Normalisation
d'une relation**Exemple.** Considérons le schéma de base de données suivant.

$$S = \left\{ \begin{array}{l} \text{Film}(\text{Titre}, \text{date}, \text{longueur}, \text{couleur}, \text{nomStudio}, \text{IdProd}), \\ \text{Joue}(\text{TitreFilm}, \text{FilmDate}, \text{nomActeur}, \text{Paye}), \\ \text{Acteur}(\text{nom}, \text{adresse}, \text{genre}, \text{dateNaissance}), \\ \text{Studio}(\text{nom}, \text{Adresse}), \\ \text{Producteur}(\text{nom}, \text{adresse}, \text{Id}) \end{array} \right\}$$

- ❶ Quels sont les titre et longueur des films produits par le studio Disney en 1999 ?

```
SELECT Titre, longueur FROM Film
WHERE nomStudio='Disney' and Date=1999;
```

$$\pi \left[\begin{array}{c} \text{Titre} \\ \text{Long} \end{array} \right] \sigma \left[\begin{array}{c} \text{nomStudio} = ' \text{Disney}' \wedge \\ \text{Date} = 1999; \end{array} \right] \text{Film}$$

- ❷ On peut remplacer la suite exhaustive des attributs d'une table par le symbole * :

```
SELECT * FROM Film
WHERE nomStudio='Disney' and Date=1999;
```

$$\sigma \left[\begin{array}{c} \text{nomStudio} = ' \text{Disney}' \wedge \\ \text{Date} = 1999; \end{array} \right] \text{Film}$$

Introduction
aux BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts
généraux

Notions
essentielles

Algèbre
Relationnelle
+ SQL

Opérateurs unaires

Traduction en SQL

Jointures

Opérateurs
ensemblistes

Opérateurs hors de
l'Alg.

Autres ordres SQL

le modèle

Entités-
Associations

NoSQL

Normalisation
d'une relation

Exemple. Considérons le schéma de base de données suivant.

$$S = \left\{ \begin{array}{l} \text{Film}(\text{Titre}, \text{date}, \text{longueur}, \text{couleur}, \text{nomStudio}, \text{IdProd}), \\ \text{Joue}(\text{TitreFilm}, \text{FilmDate}, \text{nomActeur}, \text{Paye}), \\ \text{Acteur}(\text{nom}, \text{adresse}, \text{genre}, \text{dateNaissance}), \\ \text{Studio}(\text{nom}, \text{Adresse}), \\ \text{Producteur}(\text{nom}, \text{adresse}, \text{Id}) \end{array} \right\}$$

③ Utilisation des ALIAS :

④ on peut un peu compliquer la sélection :

Introduction
aux BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts
généraux

Notions
essentielles

Algèbre
Relationnelle
+ SQL

Opérateurs unaires
Traduction en SQL

Jointures

Opérateurs
ensemblistes

Opérateurs hors de
l'Alg.

Autres ordres SQL

le modèle
Entités-
Associations

NoSQL

Normalisation
d'une relation

Exemple. Considérons le schéma de base de données suivant.

$$S = \left\{ \begin{array}{l} \text{Film}(\text{Titre}, \text{date}, \text{longueur}, \text{couleur}, \text{nomStudio}, \text{IdProd}), \\ \text{Joue}(\text{TitreFilm}, \text{FilmDate}, \text{nomActeur}, \text{Paye}), \\ \text{Acteur}(\text{nom}, \text{adresse}, \text{genre}, \text{dateNaissance}), \\ \text{Studio}(\text{nom}, \text{Adresse}), \\ \text{Producteur}(\text{nom}, \text{adresse}, \text{Id}) \end{array} \right\}$$

③ Utilisation des ALIAS :

```
SELECT Titre AS nom, long AS Durée FROM Film  
WHERE nomStudio='Disney' and Date=1999;
```

④ on peut un peu compliquer la sélection :

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQL

Opérateurs unaires

Traduction en SQL

Jointures

Opérateurs
ensemblistesOpérateurs hors de
l'Alg.

Autres ordres SQL

le modèle

Entités-
Associations

NoSQL

Normalisation
d'une relation**Exemple.** Considérons le schéma de base de données suivant.

$$S = \left\{ \begin{array}{l} \text{Film}(\text{Titre}, \text{date}, \text{longueur}, \text{couleur}, \text{nomStudio}, \text{IdProd}), \\ \text{Joue}(\text{TitreFilm}, \text{FilmDate}, \text{nomActeur}, \text{Paye}), \\ \text{Acteur}(\text{nom}, \text{adresse}, \text{genre}, \text{dateNaissance}), \\ \text{Studio}(\text{nom}, \text{Adresse}), \\ \text{Producteur}(\text{nom}, \text{adresse}, \text{Id}) \end{array} \right\}$$

3 Utilisation des ALIAS :

SELECT Titre AS nom, long AS Durée FROM Film
WHERE nomStudio='Disney' and Date=1999;

$$\alpha \left[\begin{array}{l} \text{Titre : nom} \\ \text{long : Duree} \end{array} \right] \pi \left[\begin{array}{l} \text{Titre} \\ \text{long} \end{array} \right] \sigma \left[\begin{array}{l} \text{nomStudio} = ' \text{Disney}' \\ \wedge \text{Date} = 1999; \end{array} \right] \text{Film}$$

Attention, les colonnes ont changé de noms.

4 on peut un peu compliquer la sélection :

Exemple. Considérons le schéma de base de données suivant.

$$S = \left\{ \begin{array}{l} \text{Film}(\text{Titre}, \text{date}, \text{longueur}, \text{couleur}, \text{nomStudio}, \text{IdProd}), \\ \text{Joue}(\text{TitreFilm}, \text{FilmDate}, \text{nomActeur}, \text{Paye}), \\ \text{Acteur}(\text{nom}, \text{adresse}, \text{genre}, \text{dateNaissance}), \\ \text{Studio}(\text{nom}, \text{Adresse}), \\ \text{Producteur}(\text{nom}, \text{adresse}, \text{Id}) \end{array} \right\}$$

3 Utilisation des ALIAS :

```
SELECT Titre AS nom, long AS Durée FROM Film
WHERE nomStudio='Disney' and Date=1999;
```

$$\alpha \left[\begin{array}{l} \text{Titre : nom} \\ \text{long : Duree} \end{array} \right] \pi \left[\begin{array}{l} \text{Titre} \\ \text{long} \end{array} \right] \sigma \left[\begin{array}{l} \text{nomStudio} = \text{'Disney'} \\ \wedge \text{Date} = 1999; \end{array} \right] \text{Film}$$

Attention, les colonnes ont changé de noms.

4 on peut un peu compliquer la sélection :

```
SELECT Titre FROM Film
WHERE nomStudio='Disney' and
(Date>1999 OR longueur <90) AND NOT couleur;
```

Introduction
aux BDRJean-Paul
Comet,
Nadia
Abchiche-
MimouniLes concepts
générauxNotions
essentiellesAlgèbre
Relationnelle
+ SQLOpérateurs unaires
Traduction en SQL

Jointures

Opérateurs
ensemblistesOpérateurs hors de
l'Alg.

Autres ordres SQL

le modèle
Entités-
Associations

NoSQL

Normalisation
d'une relation

- 1 la valeur NULL code l'absence de valeur
- 2 `NULL op v` est évalué à NULL si `op` est un opérateur arithmétique et `v` une valeur.
- 3 `NULL = v` est évalué à UNKNOWN (idem pour `NULL > v`).
En particulier, cette expression n'est pas évaluée à vraie.
- 4 Faire donc attention lorsqu'on manipule l'absence de valeur NULL.

`SELECT * FROM Film WHERE Date >= 1970 OR Date < 1970;`
renvoie la table Film **privée des n -uplets pour lesquels la valeur Date est null.**

Recherche des n -uplets pour lesquels une colonne n'a pas de valeur :

- `SELECT * FROM my_table WHERE phone = NULL;`
- `SELECT * FROM my_table WHERE phone IS NULL;`
- `SELECT * FROM my_table WHERE phone = '';`

Introduction
aux BDR

Jean-Paul
Comet,
Nadia
Abchiche-
Mimouni

Les concepts
généraux

Notions
essentielles

Algèbre
Relationnelle
+ SQL

Opérateurs unaires
Traduction en SQL

Jointures

Opérateurs
ensemblistes

Opérateurs hors de
l'Alg.

Autres ordres SQL

le modèle

Entités-
Associations

NoSQL

Normalisation
d'une relation

Soient $\{B_1, \dots, B_k\} \subset \{A_1, \dots, A_n\}$.

```
SELECT A1, ..., An FROM R WHERE C  
ORDER BY B1, ..., Bk;
```

Supposons que $k = 2$

Si $n(B_1) < n'(B_1)$ alors n avant n' .

Si $n(B_1) = n'(B_1)$ alors n avant n' ssi $n(B_2) < n'(B_2)$

```
SELECT * FROM Film  
WHERE nomStudio='Disney' and Date='1999'  
ORDER BY Longueur, Titre;
```