

Bases de données (1/4)

Introduction : SQL / bases de données relationnelles

Stéphane Gonnord, Laurent Jouhet

Lycée du parc - Lyon

Plan

Problématique

Des exemples : des données en pagaille

Un cahier des charges : une boîte noire

SGBDR, SQL : ???

SGBD : Système de Gestion de Base de Données

Ça ressemble à quoi en pratique ? cf TP

Le langage SQL : Structured Query Language

Modèle relationnel

Des maths pour formaliser un peu

Algèbre relationnelle

Les villes, départements et régions

- Vous disposez de plein d'infos sur toutes les villes de France...
- Ville(Lyon, Rhône, 484344) ; Département(Rhône, 69, Rhône-Alpes) ; Région(Rhône-Alpes).

Les villes, départements et régions

- Vous disposez de plein d'infos sur toutes les villes de France...
- Ville(Lyon, Rhône, 484344) ; Département(Rhône, 69, Rhône-Alpes) ; Région(Rhône-Alpes).
- Comment relier tout cela ?

Les villes, départements et régions

- Vous disposez de plein d'infos sur toutes les villes de France...
- Ville(Lyon, Rhône, 484344) ; Département(Rhône, 69, Rhône-Alpes) ; Région(Rhône-Alpes).
- Comment relier tout cela ? Et si un département **change** de nom ?

Les villes, départements et régions

- Vous disposez de plein d'infos sur toutes les villes de France...
- Ville(Lyon, Rhône, 484344) ; Département(Rhône, 69, Rhône-Alpes) ; Région(Rhône-Alpes).
- Comment relier tout cela ? Et si un département **change** de nom ?
- Quelques questions / **requêtes** (parfois **compliquées**) :
 - Quelles sont les villes de plus de 100000 habitants ?

Les villes, départements et régions

- Vous disposez de plein d'infos sur toutes les villes de France...
- Ville(Lyon, Rhône, 484344) ; Département(Rhône, 69, Rhône-Alpes) ; Région(Rhône-Alpes).
- Comment relier tout cela ? Et si un département **change** de nom ?
- Quelques questions / **requêtes** (parfois **compliquées**) :
 - Quelles sont les villes de plus de 100000 habitants ?
 - Combien y a-t-il de communes en France ?
 - Quelle est la population vivant dans les villes française ?
 - Quelles sont les villes de la Loire ?
 - Combien de départements/villes/habitants en région Poitou-Charentes ?
 - ...

Base de notes de colles

- Vous disposez de plein d'infos sur des Gens...
- Colleurs et collés : nom, prénom, identifiant.

Base de notes de colles

- Vous disposez de plein d'infos sur des Gens...
- Colleurs et collés : nom, prénom, identifiant.
- Comment relier tout cela ; qu'est-ce qu'une colle ?

Base de notes de colles

- Vous disposez de plein d'infos sur des Gens...
- Colleurs et collés : nom, prénom, identifiant.
- Comment relier tout cela ; qu'est-ce qu'une colle ?
- Quelques questions :
 - Quelles sont les notes de l'élève Dupont ? Et la moyenne ?

Base de notes de colles

- Vous disposez de plein d'infos sur des Gens...
- Colleurs et collés : nom, prénom, identifiant.
- Comment relier tout cela ; qu'est-ce qu'une colle ?
- Quelques questions :
 - Quelles sont les notes de l'élève Dupont ? Et la moyenne ?
 - Quelles ont été les colles avec une note ≤ 5 ?

Base de notes de colles

- Vous disposez de plein d'infos sur des Gens...
- Colleurs et collés : nom, prénom, identifiant.
- Comment relier tout cela ; qu'est-ce qu'une colle ?
- Quelques questions :
 - Quelles sont les notes de l'élève Dupont ? Et la moyenne ?
 - Quelles ont été les colles avec une note ≤ 5 ?
 - Quels colleurs ont donné des notes ≥ 17 ?
- Problèmes supplémentaires :
 - Doit pouvoir être **consultée** par différentes personnes.
 - Doit pouvoir être **modifiée** par différentes personnes ; pas les mêmes !



Un annuaire

- « Ensemble » de personnes...

Un annuaire

- « Ensemble » de personnes... nom, prénom, téléphone, numéro de sécu (**identifiant unique**).

Un annuaire

- « Ensemble » de personnes... nom, prénom, téléphone, numéro de sécu (**identifiant unique**).
- Comment stocker ces données (en mémoire et sur le disque dur) ?

Un annuaire

- « Ensemble » de personnes... nom, prénom, téléphone, numéro de sécu (**identifiant unique**).
- Comment stocker ces données (en mémoire et sur le disque dur) ?
- Quelques valeurs/lignes/enregistrements :

Gonnord	Stéphane	068655****	1711179****
Moynot	Olivier	06*****	1*****
Bel	Pierre	06*****	1*****
Jouhet	Laurent	06*****	1*****

...

Un annuaire

- « Ensemble » de personnes... nom, prénom, téléphone, numéro de sécu (**identifiant unique**).
- Comment stocker ces données (en mémoire et sur le disque dur) ?
- Quelques valeurs/lignes/enregistrements :

Gonnord	Stéphane	068655****	1711179****
Moynot	Olivier	06*****	1*****
Bel	Pierre	06*****	1*****
Jouhet	Laurent	06*****	1*****

...

- Quelques questions :
 - Quelles sont les personnes ayant comme prénom "Olivier" ?

Un annuaire

- « Ensemble » de personnes... nom, prénom, téléphone, numéro de sécu (**identifiant unique**).
- Comment stocker ces données (en mémoire et sur le disque dur) ?
- Quelques valeurs/lignes/enregistrements :

Gonnord	Stéphane	068655****	1711179****
Moynot	Olivier	06*****	1*****
Bel	Pierre	06*****	1*****
Jouhet	Laurent	06*****	1*****

...

- Quelques questions :
 - Quelles sont les personnes ayant comme prénom "Olivier" ?
 - Quel est le nom des personnes nées en 1971 ?

Un annuaire

- « Ensemble » de personnes... nom, prénom, téléphone, numéro de sécu (**identifiant unique**).
- Comment stocker ces données (en mémoire et sur le disque dur) ?
- Quelques valeurs/lignes/enregistrements :

Gonnord	Stéphane	068655****	1711179****
Moynot	Olivier	06*****	1*****
Bel	Pierre	06*****	1*****
Jouhet	Laurent	06*****	1*****

...

- Quelques questions :
 - Quelles sont les personnes ayant comme prénom "Olivier" ?
 - Quel est le nom des personnes nées en 1971 ?
 - Qui a un nom commençant par un 'G' ?

Et le rapport avec python ?

Dans ce cours, pas d'algorithmes, mais une boîte noire !

Cahier des charges

- Ne pas tripatouiller soi-même des fichiers.

Cahier des charges

- Ne pas tripatouiller soi-même des fichiers.
- Ne pas s'occuper de la façon dont sont stockées les informations...

Cahier des charges

- Ne pas tripatouiller soi-même des fichiers.
- Ne pas s'occuper de la façon dont sont stockées les informations... ni où.

Cahier des charges

- Ne pas tripatouiller soi-même des fichiers.
- Ne pas s'occuper de la façon dont sont stockées les informations... ni où.
- Pouvoir collecter/fournir des informations sans programmer. Avoir un langage de **description** de requêtes.

Cahier des charges

- Ne pas tripatouiller soi-même des fichiers.
- Ne pas s'occuper de la façon dont sont stockées les informations... ni où.
- Pouvoir collecter/fournir des informations sans programmer. Avoir un langage de **description** de requêtes.
- Avoir une bonne puissance d'expressivité, gérer des entités et des relations entre elles.

Cahier des charges

- Ne pas tripatouiller soi-même des fichiers.
- Ne pas s'occuper de la façon dont sont stockées les informations... ni où.
- Pouvoir collecter/fournir des informations sans programmer. Avoir un langage de **description** de requêtes.
- Avoir une bonne puissance d'expressivité, gérer des entités et des relations entre elles.
- Autoriser des accès de différents **clients**.

Cahier des charges

- Ne pas tripatouiller soi-même des fichiers.
- Ne pas s'occuper de la façon dont sont stockées les informations... ni où.
- Pouvoir collecter/fournir des informations sans programmer. Avoir un langage de **description** de requêtes.
- Avoir une bonne puissance d'expressivité, gérer des entités et des relations entre elles.
- Autoriser des accès de différents **clients**.
- Garantir une certaine pérennité des données.

Systèmes de Gestion de Base de Données

- Un utilisateur ; des données. Et entre les deux ?

Systèmes de Gestion de Base de Données

- Un utilisateur ; des données. Et entre les deux ?
- Quelques scénarios :
 - Alice va chercher un livre dans sa bibliothèque.

Systemes de Gestion de Base de Données

- Un utilisateur ; des données. Et entre les deux ?
- Quelques scénarios :
 - Alice va chercher un livre dans sa bibliothèque.
 - Alice demande à Bob d'aller chercher un livre dans **sa** bibliothèque à elle.

Systemes de Gestion de Base de Données

- Un utilisateur ; des données. Et entre les deux ?
- Quelques scénarios :
 - Alice va chercher un livre dans sa bibliothèque.
 - Alice demande à Bob d'aller chercher un livre dans sa bibliothèque à elle. (ici, Bob est le SGBD)
 - Au téléphone !

Systemes de Gestion de Base de Données

- Un utilisateur ; des données. Et entre les deux ?
- Quelques scénarios :
 - Alice va chercher un livre dans sa bibliothèque.
 - Alice demande à Bob d'aller chercher un livre dans sa bibliothèque à elle. (ici, Bob est le SGBD)
 - Au téléphone !
 - Alice va chercher sur google les pages contenant «coléoptère».

Systèmes de Gestion de Base de Données

- Un utilisateur ; des données. Et entre les deux ?
- Quelques scénarios :
 - Alice va chercher un livre dans sa bibliothèque.
 - Alice demande à Bob d'aller chercher un livre dans sa bibliothèque à elle. (ici, Bob est le SGBD)
 - Au téléphone !
 - Alice va chercher sur google les pages contenant «coléoptère».
 - Alice fait une requête SQLite depuis `sqliteman`

Systèmes de Gestion de Base de Données

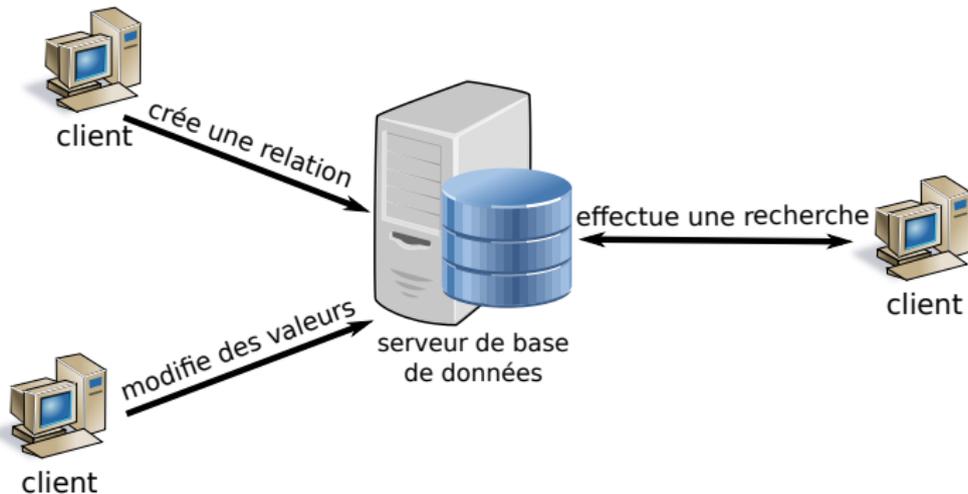
- Un utilisateur ; des données. Et entre les deux ?
- Quelques scénarios :
 - Alice va chercher un livre dans sa bibliothèque.
 - Alice demande à Bob d'aller chercher un livre dans sa bibliothèque à elle. (ici, Bob est le SGBD)
 - Au téléphone !
 - Alice va chercher sur google les pages contenant «coléoptère».
 - Alice fait une requête SQLite depuis `sqliteman`
 - Alice fait une requête SQL depuis Python + `sqlite3`

Systèmes de Gestion de Base de Données

- Un utilisateur ; des données. Et entre les deux ?
- Quelques scénarios :
 - Alice va chercher un livre dans sa bibliothèque.
 - Alice demande à Bob d'aller chercher un livre dans sa bibliothèque à elle. (ici, Bob est le SGBD)
 - Au téléphone !
 - Alice va chercher sur google les pages contenant «coléoptère».
 - Alice fait une requête SQLite depuis `sqliteman`
 - Alice fait une requête SQL depuis Python + `sqlite3`
- Le SGBD est toujours entre l'utilisateur et les données !

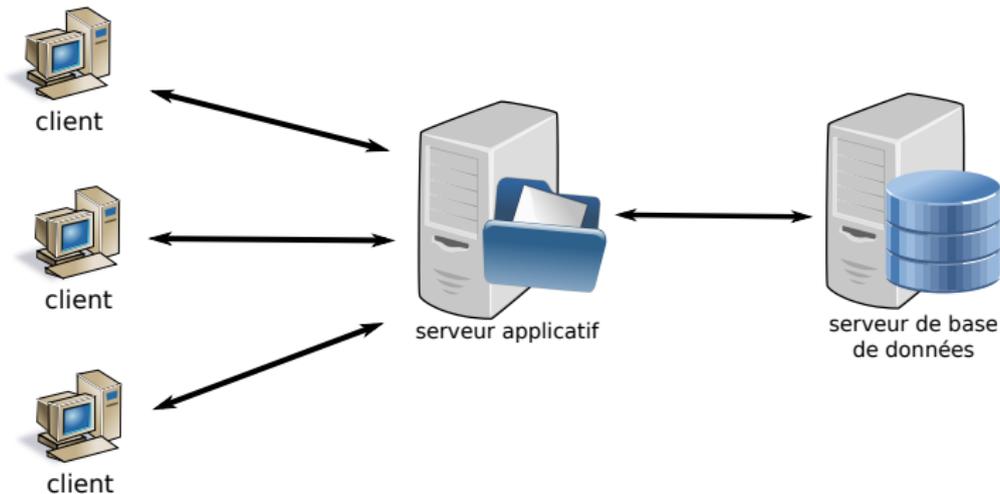
Architecture « client-serveur »

- Accès direct utilisateur/base de donnée.
- Architecture légère.



Architecture « trois tiers »

- Tiers applicatif : intermédiaire.
- Rôle de gendarme/filtre/traducteur.



Via sqliteman

cdr.s3db - Sqliteman

File Context Database System Help

Schema Pragmas

Database

- main
 - Tables (3)
 - communes
 - Columns (4)
 - Indexes (0)
 - System Indexes (0)
 - Triggers (0)
 - departements
 - Columns (3)
 - Indexes (0)
 - System Indexes (1)
 - Triggers (0)
 - regions**
 - Columns (2)
 - Indexes (0)
 - System Indexes (0)
 - Triggers (0)
 - Views (0)
 - System Catalogue (2)

```

1 SELECT departements.nom FROM
2 departements JOIN regions
3 ON departements.reg = regions.id
4 WHERE regions.nom = 'Poitou-Charentes'
```

Duration: 0.005 seconds Col: 39 Row: 4/4

Full View Item View Script Output

	nom
1	Vienne
2	Deux-Sèvres
3	Charente-Maritime
4	Charente

Query OK
Row(s) returned: 4
SELECT departements.nom FROM
departements JOIN regions
ON departements.reg = regions.id
WHERE regions.nom = 'Poitou-Charentes'

Sqlite: 3.7.17

Via la librairie sqlite3

Spyder (Python 2.7)

File Edit Search Source Run Interpreters Tools View ?

Editor - /home/stephane/documents/info-pour-tous/cours-sup/05-bdd/demo_bdd.py

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Thu Mar 13 09:55:17 2014
4
5 @author: stephane
6 """
7
8 import sqlite3
9
10 com = sqlite3.connect('annexes/cdr.db')
11
12 c = com.cursor()
13
14 resultats = list(c.execute("""SELECT departements.nom FROM
15 departements JOIN regions
16 ON departements.reg = regions.id
17 WHERE regions.nom = 'Centre'"""))
18
19 for x in resultats:
20     print x
21
22 for x in resultats:
23     print x[0]
24
25 resultats = c.execute("""SELECT departements.nom FROM
26 departements JOIN regions
27 ON departements.reg = regions.id
28 WHERE regions.nom = 'Poitou-Charentes'""")
29
30 for x in resultats:
31     print(x)
32
33
34 com.close()

```

Console - /home/stephane/documents/info-pour-tous/cours-sup/05-bdd/demo_bdd.py 00:04:06

```

(u'Eure-et-Loir',)
(u'Loiret',)
(u'Loire-et-Cher',)
(u'Indre-et-Loire',)
(u'Indre',)
(u'Cher',)
Eure-et-Loir
Loiret
Loire-et-Cher
Indre-et-Loire
Indre
Cher
(u'Vienne',)
Traceback (most recent call last):
  File "/home/stephane/documents/info-pour-tous/cours-sup/05-bdd/demo_bdd.py", line 30, in
    for x in resultats:
sqlite3.OperationalError: Could not decode to UTF-8 column 'nom' with text 'Deux-Svres'
>>> |

```

Object inspector

Source Editor Object list

list

Definition: list(...)

Variable explorer

Name /	Type	Size	Value
x	tuple	1	<tuple @ 0x8AC5C6C>

Permissions: RW End-of-lines: LF Encoding: UTF-8 Line: 17 Column: 64

Une base de données taupinale

Sujets de concours - Google Chrome

Google Agenda | Statistiques - CTRL + - G | Sujets de concours

localhost/html/Index.php

www.lamsade... | SQLZOO | L'encoding en ... | Hello, World! - ... | Cours SQL et A... | Other Bookmarks

Union des Professeurs de Spéciales Sujets de concours

année	2010-2014	chapitres à ...	aborder/éviter
concours	Mines-Ponts	Suites et fonctions	<input type="checkbox"/> <input type="checkbox"/>
filière	indifférent	Développements asymptotiques	<input type="checkbox"/> <input type="checkbox"/>
matière	indifférent	Intégration	<input type="checkbox"/> <input type="checkbox"/>
épreuve	indifférent	Séries numériques	<input type="checkbox"/> <input type="checkbox"/>
domaine	indifférent	Suites et séries de fonctions	<input type="checkbox"/> <input type="checkbox"/>
titre		Séries entières	<input type="checkbox"/> <input type="checkbox"/>
		Séries de Fourier	<input checked="" type="checkbox"/> <input type="checkbox"/>
		Équations différentielles	<input type="checkbox"/> <input type="checkbox"/>
		Fonctions de plusieurs variables	<input type="checkbox"/> <input type="checkbox"/>

aide | effacer | rechercher

m97xp2eb.pdf | m97xp2ea.pdf | Show all downloads...

Une base de données taupinale

Sujets de concours - Google Chrome

Google Agenda | Statistiques - CTRL + - G | Sujets de concours

localhost/html/index.php

www.lamsade... | SQLZOO | L'encoding en ... | Hello, World! - ... | Cours SQL et A... | Other Bookmarks

Union des Professeurs de Spéciales Sujets de concours

année	2010-2014	chapitres à ...	aborder/éviter
concours	Mines-Ponts	Suites et fonctions	<input type="checkbox"/> <input type="checkbox"/>
filière	indifférent	Développements asymptotiques	<input type="checkbox"/> <input type="checkbox"/>
matière	indifférent	Intégration	<input type="checkbox"/> <input type="checkbox"/>
épreuve	indifférent	Séries numériques	<input type="checkbox"/> <input type="checkbox"/>
domaine	indifférent	Suites et séries de fonctions	<input type="checkbox"/> <input type="checkbox"/>
titre		Séries entières	<input type="checkbox"/> <input type="checkbox"/>
		Séries de Fourier	<input checked="" type="checkbox"/> <input type="checkbox"/>
		Équations différentielles	<input type="checkbox"/> <input type="checkbox"/>
		Fonctions de plusieurs variables	<input type="checkbox"/> <input type="checkbox"/>

aide | effacer | rechercher

m97xp2eb.pdf | m97xp2ea.pdf | Show all downloads...

SQL : Villes, départements et régions

- Schéma relationnel :

```
communes(id integer, dep integer, nom text,  
         pop integer)
```

SQL : Villes, départements et régions

- Schéma relationnel :

```
communes(id integer, dep integer, nom text,  
         pop integer)
```

- *Nom des villes de plus de 100000 habitants ?*

```
SELECT nom FROM communes WHERE pop>=100000;
```

41, dont deux « Saint-Denis »

SQL : Villes, départements et régions

- Schéma relationnel :

```
communes(id integer, dep integer, nom text,  
         pop integer)
```

- *Nom des villes de plus de 100000 habitants ?*

```
SELECT nom FROM communes WHERE pop>=100000;
```

41, dont deux « Saint-Denis »

```
SELECT DISTINCT nom FROM communes WHERE pop>=100000;
```

SQL : Villes, départements et régions

- Schéma relationnel :

```
communes(id integer, dep integer, nom text,  
         pop integer)
```

- *Nom des villes de plus de 100000 habitants ?*

```
SELECT nom FROM communes WHERE pop>=100000;
```

41, dont deux « Saint-Denis »

```
SELECT DISTINCT nom FROM communes WHERE pop>=100000;
```

- *Population vivant dans des villes française ?*

SQL : Villes, départements et régions

- Schéma relationnel :

```
communes(id integer, dep integer, nom text,  
         pop integer)
```

- *Nom des villes de plus de 100000 habitants ?*

```
SELECT nom FROM communes WHERE pop>=100000;
```

41, dont deux « Saint-Denis »

```
SELECT DISTINCT nom FROM communes WHERE pop>=100000;
```

- *Population vivant dans des villes française ?*

```
SELECT COUNT(pop) FROM communes;
```

36705...

SQL : Villes, départements et régions

- Schéma relationnel :

```
communes(id integer, dep integer, nom text,  
         pop integer)
```

- *Nom des villes de plus de 100000 habitants ?*

```
SELECT nom FROM communes WHERE pop>=100000;
```

41, dont deux « Saint-Denis »

```
SELECT DISTINCT nom FROM communes WHERE pop>=100000;
```

- *Population vivant dans des villes française ?*

```
SELECT COUNT(pop) FROM communes;
```

36705... (étonnant ? non) !!

SQL : Villes, départements et régions

- Schéma relationnel :

```
communes(id integer, dep integer, nom text,
          pop integer)
```

- *Nom des villes de plus de 100000 habitants ?*

```
SELECT nom FROM communes WHERE pop>=100000;
```

41, dont deux « Saint-Denis »

```
SELECT DISTINCT nom FROM communes WHERE pop>=100000;
```

- *Population vivant dans des villes française ?*

```
SELECT COUNT(pop) FROM communes;
```

36705... (étonnant ? non) !!

```
SELECT SUM(pop) FROM communes;
```

64612967 : mieux !

Premières jointures : travailler avec plusieurs tables

- **Autres tables :**

`departements(id integer, reg integer, nom text)`

`regions(id integer, nom text)`

- *Quels sont les noms des villes de la Loire ?*

Premières jointures : travailler avec plusieurs tables

- **Autres tables :**

```
departements(id integer, reg integer, nom text)
```

```
regions(id integer, nom text)
```

- ***Quels sont les noms des villes de la Loire ?***

```
SELECT communes.nom FROM
```

```
communes JOIN departements
```

```
ON communes.dep = departements.id
```

```
WHERE departements.nom = 'Loire';
```

Premières jointures : travailler avec plusieurs tables

- **Autres tables :**

```
departements(id integer, reg integer, nom text)
```

```
regions(id integer, nom text)
```

- ***Quels sont les noms des villes de la Loire ?***

```
SELECT communes.nom FROM
```

```
communes JOIN departements
```

```
ON communes.dep = departements.id
```

```
WHERE departements.nom = 'Loire';
```

- ***Combien de départements/villes/habitants en région Poitou-Charentes ?***

Premières jointures : travailler avec plusieurs tables

- **Autres tables :**

```
departements(id integer, reg integer, nom text)
```

```
regions(id integer, nom text)
```

- ***Quels sont les noms des villes de la Loire ?***

```
SELECT communes.nom FROM
```

```
communes JOIN departements
```

```
ON communes.dep = departements.id
```

```
WHERE departements.nom = 'Loire';
```

- ***Combien de départements/villes/habitants en région Poitou-Charentes ?***

```
SELECT COUNT(*) FROM
```

```
regions JOIN departements
```

```
ON regions.id = departements.reg
```

```
WHERE regions.nom = 'Poitou-Charentes';
```

Premières jointures : travailler avec plusieurs tables

- **Autres tables :**

```
departements(id integer, reg integer, nom text)
```

```
regions(id integer, nom text)
```

- ***Quels sont les noms des villes de la Loire ?***

```
SELECT communes.nom FROM
```

```
communes JOIN departements
```

```
ON communes.dep = departements.id
```

```
WHERE departements.nom = 'Loire';
```

- ***Combien de départements/villes/habitants en région Poitou-Charentes ?***

```
SELECT COUNT(*) FROM
```

```
regions JOIN departements
```

```
ON regions.id = departements.reg
```

```
WHERE regions.nom = 'Poitou-Charentes';
```

```
SELECT COUNT(*) FROM
```

```
regions JOIN departements JOIN communes
```

```
ON regions.id = departements.reg AND communes.dep = departements.id
```

```
WHERE regions.nom = 'Poitou-Charentes';
```

```
SELECT SUM(pop) FROM
```

```
regions JOIN departements JOIN communes
```

```
ON regions.id = departements.reg AND communes.dep = departements.id
```

```
WHERE regions.nom = 'Poitou-Charentes';
```

Colleurs et collés

- Schéma relationnel :

```
colleurs(idc integer, nom tet, prenom text)
```

```
eleve(idc integer, nom tet, prenom text)
```

```
colle(idc integer, ide integer, date text, note integer)
```

Colleurs et collés

- Schéma relationnel :

```
colleurs(idc integer, nom tet, prenom text)
```

```
eleve(idc integer, nom tet, prenom text)
```

```
colle(idc integer, ide integer, date text, note integer)
```

- *Quelles sont les notes de l'élève Dupont ? Et sa moyenne ?*

Colleurs et collés

- Schéma relationnel :

```
colleurs(idc integer, nom tet, prenom text)
```

```
eleve(idc integer, nom tet, prenom text)
```

```
colle(idc integer, ide integer, date text, note integer)
```

- *Quelles sont les notes de l'élève Dupont ? Et sa moyenne ?*

```
SELECT note FROM
```

```
colles JOIN eleves
```

```
ON colles.ide = eleves.ide
```

```
WHERE nom='Dupont';
```

```
SELECT AVG(note) from ...
```

Colleurs et collés

- Schéma relationnel :

```
colleurs(idc integer, nom tet, prenom text)
```

```
eleve(idc integer, nom tet, prenom text)
```

```
colle(idc integer, ide integer, date text, note integer)
```

- *Quelles sont les notes de l'élève Dupont ? Et sa moyenne ?*

```
SELECT note FROM
```

```
colles JOIN eleves
```

```
ON colles.ide = eleves.ide
```

```
WHERE nom='Dupont';
```

```
SELECT AVG(note) from ...
```

- *Quels colleurs ont donné des notes ≥ 17 ?*

Colleurs et collés

- Schéma relationnel :

```
colleurs(idc integer, nom tet, prenom text)
```

```
eleve(idc integer, nom tet, prenom text)
```

```
colle(idc integer, ide integer, date text, note integer)
```

- *Quelles sont les notes de l'élève Dupont ? Et sa moyenne ?*

```
SELECT note FROM
```

```
colles JOIN eleves
```

```
ON colles.ide = eleves.ide
```

```
WHERE nom='Dupont';
```

```
SELECT AVG(note) from ...
```

- *Quels colleurs ont donné des notes ≥ 17 ?*

```
SELECT DISTINCT colleurs.nom from
```

```
colles JOIN colleurs
```

```
ON colles.idc = colleurs.idc
```

```
WHERE colles.note>=18;
```

Annuaire basique

- Schéma relationnel :

```
personnes(nom text, prenom text, tel text, num_secu text)
```

Annuaire basique

- Schéma relationnel :

```
personnes(nom text, prenom text, tel text, num_secu text)
```

- *Quelles sont les personnes ayant comme prénom "Olivier" ?*

Annuaire basique

- Schéma relationnel :

```
personnes(nom text, prenom text, tel text, num_secu text)
```

- *Quelles sont les personnes ayant comme prénom "Olivier" ?*

```
SELECT * FROM personnes  
WHERE prenom = 'Olivier'
```

Annuaire basique

- Schéma relationnel :

```
personnes(nom text, prenom text, tel text, num_secu text)
```

- *Quelles sont les personnes ayant comme prénom "Olivier" ?*

```
SELECT * FROM personnes  
WHERE prenom = 'Olivier'
```

- *Quel est le nom des personnes nées en 1971 ?*

Annuaire basique

- Schéma relationnel :

```
personnes(nom text, prenom text, tel text, num_secu text)
```

- *Quelles sont les personnes ayant comme prénom "Olivier" ?*

```
SELECT * FROM personnes  
WHERE prenom = 'Olivier'
```

- *Quel est le nom des personnes nées en 1971 ? Plus dur, n'est-ce pas ?*

Annuaire basique

- Schéma relationnel :

```
personnes(nom text, prenom text, tel text, num_secu text)
```

- *Quelles sont les personnes ayant comme prénom "Olivier" ?*

```
SELECT * FROM personnes
WHERE prenom = 'Olivier'
```

- *Quel est le nom des personnes nées en 1971 ? Plus dur, n'est-ce pas ?*

```
nom_secu = list(curseur.execute(
    """SELECT nom, secu FROM personnes"""))
res = [nom for (nom, secu) in nom_secu if secu[1:3]='71']
```

- Remarque : on triche un peu... merci Python
- *Qui a un nom commençant par un 'G' ?*

Annuaire basique

- Schéma relationnel :

```
personnes(nom text, prenom text, tel text, num_secu text)
```

- *Quelles sont les personnes ayant comme prénom "Olivier" ?*

```
SELECT * FROM personnes
WHERE prenom = 'Olivier'
```

- *Quel est le nom des personnes nées en 1971 ? Plus dur, n'est-ce pas ?*

```
nom_secu = list(curseur.execute(
    """SELECT nom, secu FROM personnes"""))
res = [nom for (nom, secu) in nom_secu if secu[1:3]='71']
```

- Remarque : on triche un peu... merci Python
- *Qui a un nom commençant par un 'G' ? Idem*

Algèbre relationnelle

- Cadre mathématique pour être (*faire ?*) plus sérieux/rigoureux.

Algèbre relationnelle

- Cadre mathématique pour être (*faire ?*) plus sérieux/rigoureux.
- Un vocabulaire spécifique de description de données (tuples).

Algèbre relationnelle

- Cadre mathématique pour être (*faire ?*) plus sérieux/rigoureux.
- Un vocabulaire spécifique de description de données (tuples).
- Une série d'opérateurs permettant de décrire formellement les requêtes.

Algèbre relationnelle

- Cadre mathématique pour être (*faire ?*) plus sérieux/rigoureux.
- Un vocabulaire spécifique de description de données (tuples).
- Une série d'opérateurs permettant de décrire formellement les requêtes.

À moins que ce ne soit l'inverse :-)

Algèbre relationnelle

- Cadre mathématique pour être (*faire ?*) plus sérieux/rigoureux.
- Un vocabulaire spécifique de description de données (tuples).
- Une série d'opérateurs permettant de décrire formellement les requêtes.

À moins que ce ne soit l'inverse :-)

- Attendu du programme : faire le lien entre :
 - une requête en langage courant ;
 - sa traduction en algèbre relationnelle ;
 - sa traduction en langage SQL.

Des maths fumeuses

- Un **schéma relationnel** :

$S = ((\text{nom}, \text{texte}), (\text{prenom}, \text{texte}), (\text{classe}, \text{entier}))$

- Trois **attributs**, avec leur **domaine** (*type*).
- Une **valeur** (*ligne/ un enregistrement*) :

$(\text{nom}:'\text{Gonnord}', \text{prenom}:'\text{Stéphane}', \text{classe}:841)$

- Une **relation** (*table*) : ensemble de valeurs

$R(S) =$

{

$(\text{nom}:'\text{Gonnord}', \text{prenom}:'\text{Stéphane}', \text{classe}:933)$

$(\text{nom}:'\text{Moynot}', \text{prenom}:'\text{Olivier}', \text{classe}:842)$

$(\text{nom}:'\text{Jouhet}', \text{prenom}:'\text{Laurent}', \text{classe}:842)$

$(\text{nom}:'\text{Bel}', \text{prenom}:'\text{Pierre}', \text{classe}:843)$

$(\text{nom}:'\text{Jouhet}', \text{prenom}:'\text{Laurent}', \text{classe}:843)$

}

Des maths fumeuses

- Un **schéma relationnel** :

$S = ((\text{nom}, \text{texte}), (\text{prenom}, \text{texte}), (\text{classe}, \text{entier}))$

- Trois **attributs**, avec leur **domaine** (*type*).
- Une **valeur** (*ligne/ un enregistrement*) :

$(\text{nom}:'\text{Gonnord}', \text{prenom}:'\text{Stéphane}', \text{classe}:841)$

- Une **relation** (*table*) : ensemble de valeurs

$R(S) =$

{

$(\text{nom}:'\text{Gonnord}', \text{prenom}:'\text{Stéphane}', \text{classe}:933)$

$(\text{nom}:'\text{Moynot}', \text{prenom}:'\text{Olivier}', \text{classe}:842)$

$(\text{nom}:'\text{Jouhet}', \text{prenom}:'\text{Laurent}', \text{classe}:842)$

$(\text{nom}:'\text{Bel}', \text{prenom}:'\text{Pierre}', \text{classe}:843)$

$(\text{nom}:'\text{Jouhet}', \text{prenom}:'\text{Laurent}', \text{classe}:843)$

}

(conception mauvaise ; pourquoi ?)

Opérateurs de l'algèbre relationnelle

- *Quel est le nom des département dont l'identifiant est 10 ou 11 ?*
- **Sélection** σ_{Cond} selon une condition (valeur d'un champ, non, et, ou).

Opérateurs de l'algèbre relationnelle

- *Quel est le nom des département dont l'identifiant est 10 ou 11 ?*
- **Sélection** σ_{Cond} selon une condition (valeur d'un champ, non, et, ou).
- **Union, intersection** de deux tables.

Opérateurs de l'algèbre relationnelle

- *Quel est le nom des département dont l'identifiant est 10 ou 11 ?*
- **Sélection** σ_{Cond} selon une condition (valeur d'un champ, non, et, ou).
- **Union, intersection** de deux tables. Inutile ?

Opérateurs de l'algèbre relationnelle

- *Quel est le nom des département dont l'identifiant est 10 ou 11 ?*
- **Sélection** σ_{Cond} selon une condition (valeur d'un champ, non, et, ou).
- **Union, intersection** de deux tables. Inutile ?
- **Projection** π_X d'une table selon un (des) attribut(s) X .

Opérateurs de l'algèbre relationnelle

- *Quel est le nom des département dont l'identifiant est 10 ou 11 ?*
- **Sélection** σ_{Cond} selon une condition (valeur d'un champ, non, et, ou).
- **Union, intersection** de deux tables. Inutile ?
- **Projection** π_X d'une table selon un (des) attribut(s) X .

$$\pi_{Nom}(\sigma_{Id=10 \text{ ou } Id=11}(\text{departements}))$$

- Soit, en SQL :

```
SELECT nom FROM departements WHERE id in (10, 11)
```

Opérateurs de l'algèbre relationnelle

- *Quel est le nom des département dont l'identifiant est 10 ou 11 ?*
- **Sélection** σ_{Cond} selon une condition (valeur d'un champ, non, et, ou).
- **Union, intersection** de deux tables. Inutile ?
- **Projection** π_X d'une table selon un (des) attribut(s) X .

$$\pi_{Nom} (\sigma_{Id=10 \text{ ou } Id=11} (departements))$$

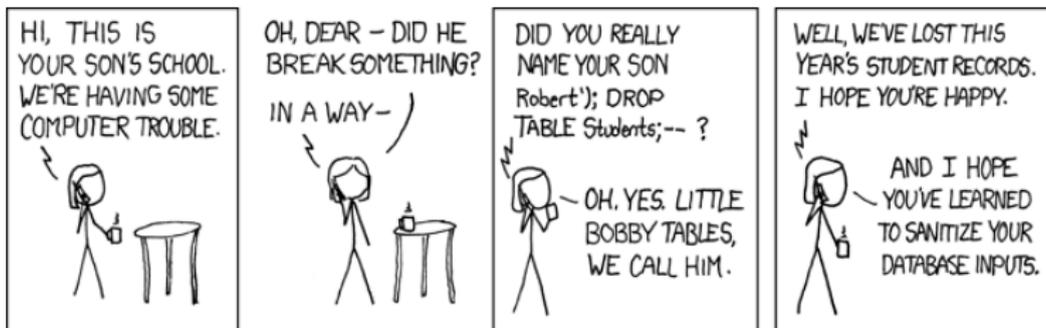
- Soit, en SQL :

```
SELECT nom FROM departements WHERE id in (10, 11)
```

Ou encore :

```
SELECT nom FROM departements WHERE id=10 OR id=11
```

C'est fini



Si un jour vous utilisez vraiment des bases de données

- pensez "sécurité" : <http://bobby-tables.com/>,
- cherchez 'bdd acid' sur internet...
- Il existe de nombreux SGBD et aussi des bases de données non relationnelles