SQL et les bases de données relationnelles

Introduction

Guillaume Raschia — Nantes Université

originaux de Philippe Rigaux, CNAM

Dernière mise-à-jour : 29 novembre 2023

Plan du cours

Tout ce qu'il faut comprendre pour **utiliser** efficacement une base de données relationnelle.

- Architecture (un peu)
- Structuration d'une base relationnelle
- Interrogation : approches déclarative et procédurale
- Conception de schémas

Pas (ou très peu) d'information sur le fonctionnement interne du système

Organisation du cours

Le cours s'appuie sur le document « Cours de bases de données : modèles et langages » de Philippe Rigaux du CNAM.

Il est organisé en 20 sessions : 8 CM • 5 TD • 6 TP • 1 Exam.

L'évaluation comporte la note d'examen et un compte-rendu de TP.

Plan de la session

Données et SGBD (S1.3)

Modèle relationnel (S2.1)

Fondements logiques (S3.1)

Données et SGBD (S1.3)

Les données

Donnée = valeur numérisée décrivant de manière élémentaire un fait, une mesure, une réalité

Exemple

le nom de l'auteur, l'âge du capitaine, le titre du livre ...

Les données décrivent des entités du monde réel, elles-mêmes associées les unes aux autres.

Exemple

Nicolas Bouvier est un écrivain suisse auteur du récit de voyage culte « l'usage du monde » paru en 1963 : deux entités, liées par la notion d'auteur.

Base de données

Une base de données a donc une structure, sinon c'est autre chose (une collection, un tas de documents, textes ou images).

Définition (Base de données)

Une base de données est un ensemble d'informations structurées mémorisées sur un support persistant.

Des fichiers structurés (tableur, CSV) sont des bases de données.

Exemple de fichiers structurés

Format CSV : une ligne par entité ; champs séparés par des ';' (fr)

```
"Bouvier" ; "Nicolas"; "L'usage du monde" ; 1963
```

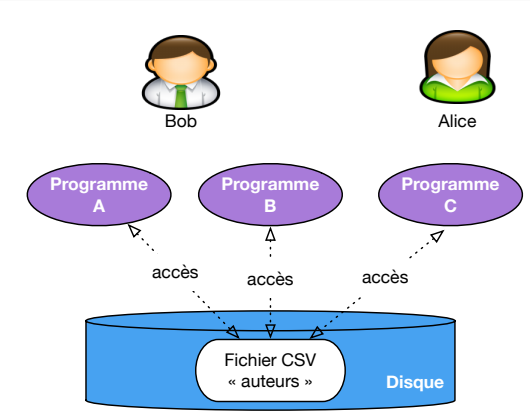
Base de données = 2, 10 ou 1 million de lignes sur le même format.

```
"Bouvier"   ; "Nicolas"; "L'usage du monde" ; 1963
"Stevenson" ; "Robert-Louis"  ; "Voyage dans les Cévennes avec un âne"
...
```

Suffisant ?

Fichiers = base de données ?

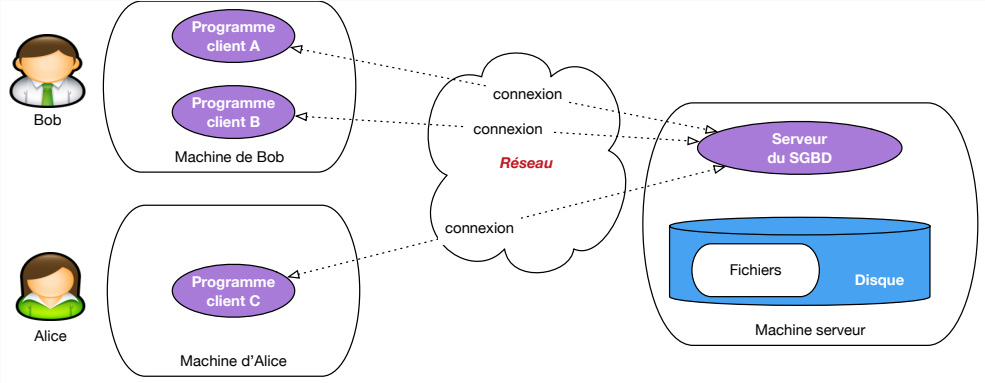
Peut-on construire des applications directement sur des fichiers ?



Insurmontables problèmes de productivité, de fiabilité, d'efficacité

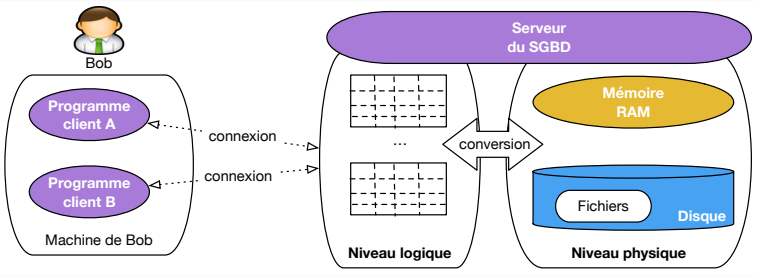
Le SGBD

Système informatique qui assure la Gestion de l'ensemble des informations stockées dans une Base de Données.



Niveaux d'abstraction et modèle de données

Le serveur peut présenter une représentation logique des données très éloignée de la représentation physique.

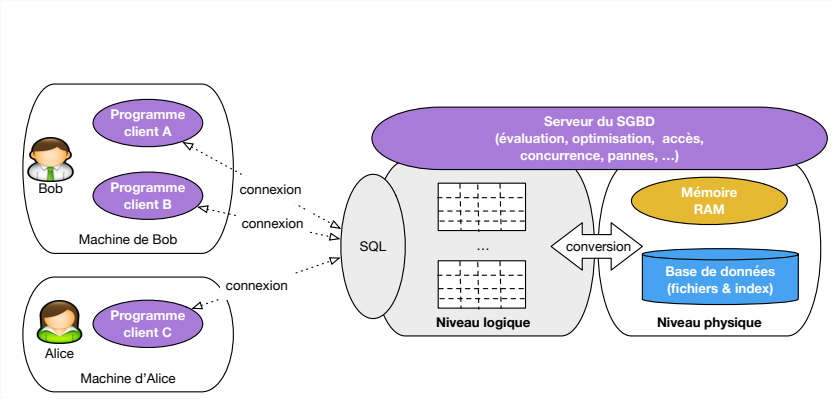


Le niveau logique définit la modélisation des données.

Langages - SQL

Pour manipuler les données, le serveur propose un langage d'interrogation.

Le plus répandu est SQL : interrogation, mise à jour, contraintes sur la base, droits d'accès...



À retenir

Les bases de données sont des ensembles structurés stockés dans des fichiers.

Les systèmes de gestion de base de données (SGBD) sont des systèmes qui prennent en charge toute la complexité de gestion des fichiers.

Un SGBD propose aux applications une vue logique indépendante du niveau physique (stockage).

SQL est le langage qui permet d'interagir avec la vue logique d'une base de données (relationnelle)

Modèle relationnel (S2.1)

Le modèle relationnel

Le modèle relationnel, c'est

- Une structure unique, la relation (ou table)
- Des contraintes qui définissent des **formes normales**, évitant les défauts de conception
- Des langages, concrétisés en pratique par SQL

Dans cette section, on parle de la structure.

Qu'est-ce qu'une relation ?

Notion mathématique : Etant donné un ensemble d'objets E, une relation (binaire) sur E est un sous-ensemble du produit cartésien $E \times E$.

Dans notre contexte, les « objets » sont des valeurs élémentaires (ou atomiques), comme les entiers $\mathbb{N}$, les réels $\mathbb{R}$, les chaînes de caractères $\mathcal{S}$.

L'ensemble des paires constituées des noms de département et de leur numéro de code est une relation sur $\mathcal{S} \times \mathbb{N}$.

Définition (Relation)

Une relation R de degré n sur les domaines $A_1, A_2, \cdots, A_n$ est un sous-ensemble fini du produit cartésien $R \subseteq A_1 \times A_2 \times \cdots \times A_n$

Représentation

Comment représente-t-on une relation ? Sous forme de table, le plus pratique.

nom	code
Ardèche	07
Gard	30
Manche	50
Paris	75

Attention, ce n'est pas n'importe quelle table. Se souvenir de la définition.

Les nuplets

Un élément d'une relation de dimension n est un nuplet $(a_1, a_2, \cdots, a_n)$.

Exemple : `(Manche, 50)`

Dans la représentation par table, un nuplet est une ligne.

On assimile nuplet et ligne, mais attention, ce n'est pas n'importe quelle ligne. Se souvenir de la définition.

Schéma de relation

On peut décrire une relation par

- Le nom de la relation
- Un nom (distinct) pour chaque dimension, dit nom d'attribut
- Le domaine de valeur de chaque dimension.

C'est le schéma de la relation, de la forme $R(A_1 : D_1, A_2 : D_2, \cdots, A_n : D_n)$

Exemple

`Département (nom: string, code: string)`, ou plus simplement
`Département (nom, code)`

Première forme normale

On ne peut pas avoir une valeur d'attribut qui soit « construite », comme par exemple une liste, ou une sous-relation.

Les valeurs dans une base de données sont dites atomiques

Définition (Première forme normale)

Une relation est en première forme normale si toutes les valeurs d'attribut sont connues et atomiques et si elle ne contient aucun doublon.

À retenir

Notions et vocabulaire

Terme du modèle	Terme de la représentation par table
Relation	Table
Nuplet	Ligne
Nom d'attribut	Nom de colonne
Valeur d'attribut	Cellule
Domaine	Type

Très simple ! (parfois trop). Favorise la rigueur et la clarté.

Fondements logiques (S3.1)

Notions élémentaires de logique formelle

- Le calcul propositionnel : valeurs de vérité et formules
- Les prédicats
- Collections et quantificateurs
- Importance pour les bases de données et SQL

Calcul propositionnel

Proposition : énoncé auquel on peut affecter deux valeurs de vérité, Vrai ou Faux.

Formule : expression basée sur des propositions et leur combinaison par des connecteurs logiques.

- la conjonction, notée $\wedge$
- la disjonction, notée $\vee$
- la négation, notée $\neg$

Exemple

$(p \wedge q) \vee r$

Valeur de vérité d'une formule

Induite à partir de valeurs de vérité des propositions, et des règles suivantes :

p	q	$p \wedge q$	$p \vee q$	$\neg p$
Vrai	Vrai	Vrai	Vrai	Faux
Vrai	Faux	Faux	Vrai	Faux
Faux	Vrai	Faux	Vrai	Vrai
Faux	Faux	Faux	Faux	Vrai

Exemple : $(p \wedge q) \vee r$ est Vrai pour les valeurs V, V, F de p, q, r

Equivalences

Un informaticien averti sait manier avec agilité les équivalences. Quelques exemples.

- $\neg(\neg F)$ est équivalente à F
- $F \vee (F_1 \wedge F_2)$ est équivalente à $(F \vee F_1) \wedge (F \vee F_2)$ (distribution)
- $F \wedge (F_1 \vee F_2)$ est équivalente à $(F \wedge F_1) \vee (F \wedge F_2)$ (distribution)
- $\neg(F_1 \wedge F_2)$ est équivalente à $(\neg F_1) \vee (\neg F_2)$ (loi DeMorgan)
- $\neg(F_1 \vee F_2)$ est équivalente à $(\neg F_1) \wedge (\neg F_2)$ (loi DeMorgan)

Donc $p \vee \neg(p \wedge \neg q)$ est une tautologie.

Prédicats

Extension puissante des propositions : construire des énoncés sur des « objets ».

Le prédicat `Compose`(X, Y) permet de construire des énoncés de la forme :

- `Compose`('Mozart', 'Don Giovanni')
- `Compose`('Debussy', 'La mer')
- Etc.

Ce sont des nuplets (ou des atomes, ou des faits).

- Il en existe une infinité
- Un contexte (ou « interprétation ») définit ceux qui sont vrais / faux.

Base de données ?

C'est un contexte donnant un ensemble fini de faits vrais. Tous les autres sont considérés comme faux.

Nuplets ouverts et fermés

Un nuplet énoncé avec des constantes est un nuplet fermé.

$$\texttt{Compose}(\text{'Mozart'}, \text{'Don Giovanni'})$$

Un nuplet énoncé avec au moins une variable est un nuplet ouvert.

$$\texttt{Compose}(X, \text{'Don Giovanni'})$$

Un nuplet ouvert désigne une infinité de faits possibles.

Interêt ?

En général on s'intéresse aux valeurs de X pour lesquelles les faits sont vrais. On a effectué une requête.

Collections et quantificateurs

Les nuplets ouverts expriment des contraintes sur un fait.

On peut exprimer des contraintes sur des collections de faits avec les quantificateurs.

- $\exists x P(x)$ est vraie s'il existe au moins une affectation de x pour laquelle $P(x)$ est vraie.
- $\forall x P(x)$ est vraie si $P(x)$ est vraie pour toutes les valeurs de x.

Variables libres et liées : un variable quantifiée est liée ; sinon elle est libre.

SQL = formules logique

Requête SQL = une formule avec des variables libres.

Résultat d'une requête = les valeurs des variables libres qui satisfont la formule.

La formule `Compose`$(X,$ 'Don Giovanni'$)$ s'écrit en SQL

```
select compositeur
  from Compose
  where oeuvre='Don Giovanni'
```

SQL c'est une syntaxe pour écrire des formules.

Déclarativité

On ne dit pas comment on calcule !

Exemples

Deux relations/prédicats :

```
Expert (id_expert, nom)
Manuscrit (id_manuscrit, auteur, titre, id_expert, commentaire)
```

$Q(t) = \texttt{Manuscrit}(\_, \text{'Proust'}, t, \_, \_)$

```
select titre from Manuscrit
where auteur = 'Proust'
```

$Q(n) = \exists x\ \texttt{Expert}(x, n) \wedge \texttt{Manuscrit}(\_, \text{'Proust'}, \_, x, \_)$

```
select nom from Expert as e
where exists (select *  from Manuscrit as m
              where e.id_expert = m.id_expert
              and auteur = 'Proust')
```

À retenir

SQL est un langage déclaratif qui exprime par une formule logique les propriétés du résultat à construire.

Avantages prouvés et éprouvés depuis les années 1970.

- signification précise, non ambigüe
- algorithmes efficaces
- langage robuste, universellement connu et adopté, normalisé
- déclarativité : SQL ne donne aucune indication sur la manière dont le système doit trouver le résultat.