Bases de données
Introduction

Silviu Maniu

2022/2023

Polytech App3

Contenu

Les questions des SGBD

On s'intéresse aux SGBD : **systèmes de gestion bases de données**

Utilisation :

- **qui** peut l'utiliser ?
- dans **quelles** situations
- **pourquoi** (pour quel problème)
- **comment** l'utiliser
- quels modes (*interactif*, *programme*)

Les questions des SGBD

Nous gérons les **principes** ; indépendant d'un SGBD particulier

- nous utilisons PostgreSQL, mais applicable aux autres (Oracle, SQLite, MySQL, etc.)

Contenu

Les problèmes de bases de données

Contexte : situations de la vraie vie (réservations, inventaire, etc.)

Situation :

- programmeur écrit une application
- il faut gérer les problèmes liés a l'implémentation (algorithmique, réseau, interface, etc.)
- *ici* : sous-ensemble des problèmes (données)

Exemple
Application SNCF : clients effectuent des réservations (données client, train), employés gèrent les trains (horaire, type, places)

Persistance

Situation :

- client effectue une réservation sur le Web
- il se déconnecte

Persistance :

- sa réservation doit être gardée dans le SGBD
- son numéro CB ne doit pas être gardé

Grandes quantités de données

Situation :

- quantité d'information supérieure par rapport a la mémoire vive
- stockage disque – mais *lent*

Gestionnaire disque :

- il faut temps de réponse rapide
- gercer le transfert entre le disque et la mémoire vive
- son numéro CB ne doit pas être gardé

Reprise sur panne

Situation :

- client en train de payer une réservation
- panne internet après paiement

Reprise sur panne :

- faut gérer les cas ou le paiement a abouti (garder réservation) ou n'a pas abouti (supprimer réservation)

Concurrence

Situation :

- il reste une seule place dans un train
- 2 clients connectés simultanément

Gestion concurrence :

- il faut donner la place à un seul client
- gestion file d'attente

Confidentialité

Situation :

- client (« hacker ») essaie de modifier réservations, horaires, paiements,...

Confidentialité/Securité :

- une client ne doit pas avoir ce pouvoir, mais un employé a besoin des droits de changement (SAV)

Contraintes d'intégrité

Situation :

- un employé introduit des données corrompues (prix négatif, horaire impossible)

Integrité :

- vérifier les données avant leur stockage (valeurs permises et par rapport aux autres données)

Répartition

Situation :

- réservation à l'international (Italie, Espagne)

Répartition :

- capable de gérer les données reparties entre plusieurs applications

Indépendance des niveaux

Situation :

- decision de modifier l'organisation des données sur le disque (passage à l'échelle, normalisation)

Indépendance :

- *modularité* : ex., l'interface ne doit pas avoir besoin des modifications aussi

Modèle de données, conception, mises à jour, interrogation

Situation :

- application doit savoir manipuler les information des clients (reservation, paiements, données perso)

SGBD :

- *représentation* des données
- *organiser* les données (tables)
- *création, modification, destruction*
- *interrogation*
- simple pour les utilisateurs/developpeurs

Contenu

But du cours

Domaine des bases de données : outils théoriques et pratiques pour résoudre ces problèmes

SGBD : fournit un ensemble d'outils conceptuels et pratiques qui permettent au programmeur de gérer ces problèmes

Application bases de données : ordres de base de données (SGBD) – en général, le langage SQL

1. **mode interactif** : directement en ligne de commande (administrateur BD, développeur)
2. **mode programme** : programme autonome (Java, Python), interface Web

But du cours

Résoudre les problèmes d'une application BD décrite par un cahier des charges, en utilisant les outils SGBD en mode interactif ou programme

Acteurs d'un application BD (*qui ?*)

3 catégories dans une application :

- programmeur : écrit l'application du début à la fin
- client/utilisateur (non-informaticien) : ne connait pas les détails de l'implémentation, désire une utilisation simple
- admistrateur (DBA) : installe logiciels SGBD, langage etc. ; création comptes, droits, etc.

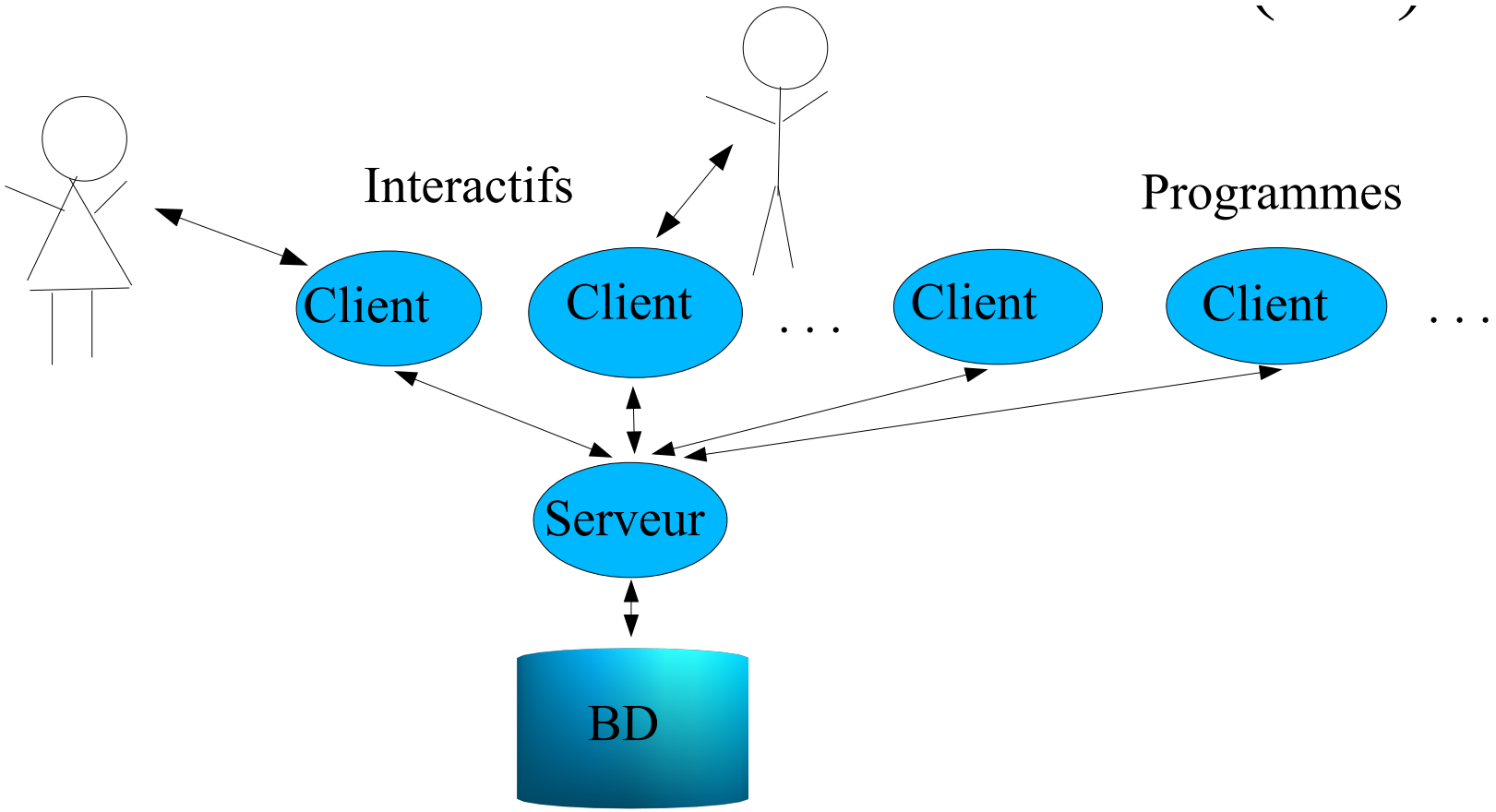
  Important – *le programmeur est le seul qui utilise le mode programme : DBA n'a pas besoin pour gérer les installations, trop complexe pour clients*

Vue d'ensemble et architecture

L'architecutre d'une BD :

- la **base de données** (BD) – gère les **données** (réservations, transactions, utilisateurs)
- le **SGBD** – accueil de la BD, gère programmes en train de s'exécuter
  - le **serveur** : unique, tourne en permanence
  - les (programmes) **clients** : peuvent être plusieurs *en même temps*, connexions/déconnexions non coordonnées

Vue d'ensemble et architecture



Vue d'ensemble et architecture

Mode interactif : directement en ligne de commande ; pour gérer la partie BD de l'application en écrivant les ordres BD nécessaires, **session** = ordres SQL entre connexion et déconnexion

Mode programme : programme qui s'exécute, même gestion ordres BD qu'en interactif, ordres lancés depuis un programme (Java, Web)

Structure du cours

Nous allons considérer quelques **problèmes** un par un

- comprendre ou est le problème
- comprendre et utiliser les outils nécessaires : les concepts, les ordres correspondants

Nous allons considérer le **mode interactif** (ordres SQL et PL/pgSQL) et **mode programme** (Java JDBC)

Compétences à acquérir

- comprendre l'intérêt de chaque problème BD (**pourquoi ?**)
- pour une application : considérer les problèmes un par un, les occurrences dans l'application (**quand ?**)
- programmer la meilleure solution pour chaque occurrence (**comment ?**)

Remerciements

Le contenu de ce cours est grandement inspiré des cours d'Emmanuel Waller

- `https://www.lri.fr/~waller/`