Principes d'utilisation des systèmes de gestion de bases de données

Contraintes

Rappel : les deux parties du cours

- création et gestion de la base
  - les problèmes BD
    - liés à la construction de la base :
      - modèle, conception, indépendance des niveaux, contraintes, confidentialité, mise à jour, persistance
    - liés à la dynamique de la base :
      - reprise sur panne, contrôle de concurrence
    - liés à l'interrogation de la base :
      - interrogation, grandes quantités (optimisation)
  - les traitements bas niveau en mode programme : PL/SQL
- accès à la base depuis un programme généraliste
  - pbs MP étudiés à travers (outre PL/SQL) : Java, PHP
  - XML, XSQL, XSLT, intégration avec relationnel

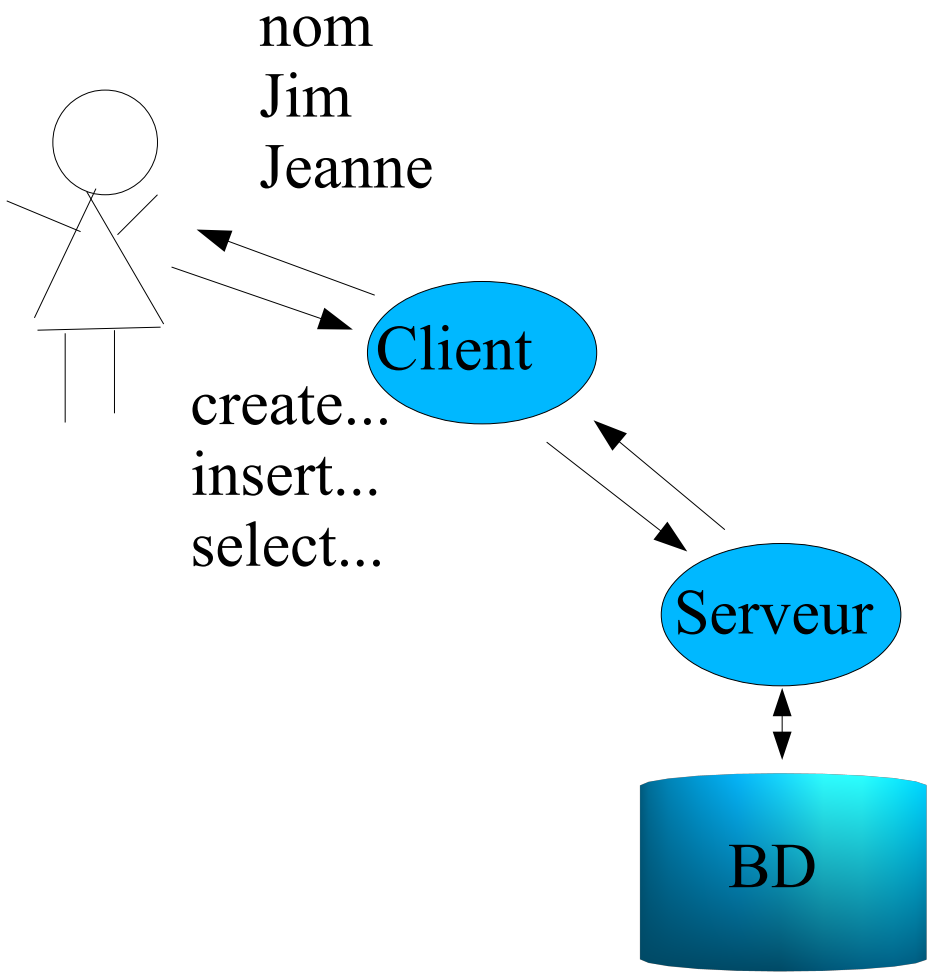
Rappel : principe

On va considérer les problèmes BD un par un

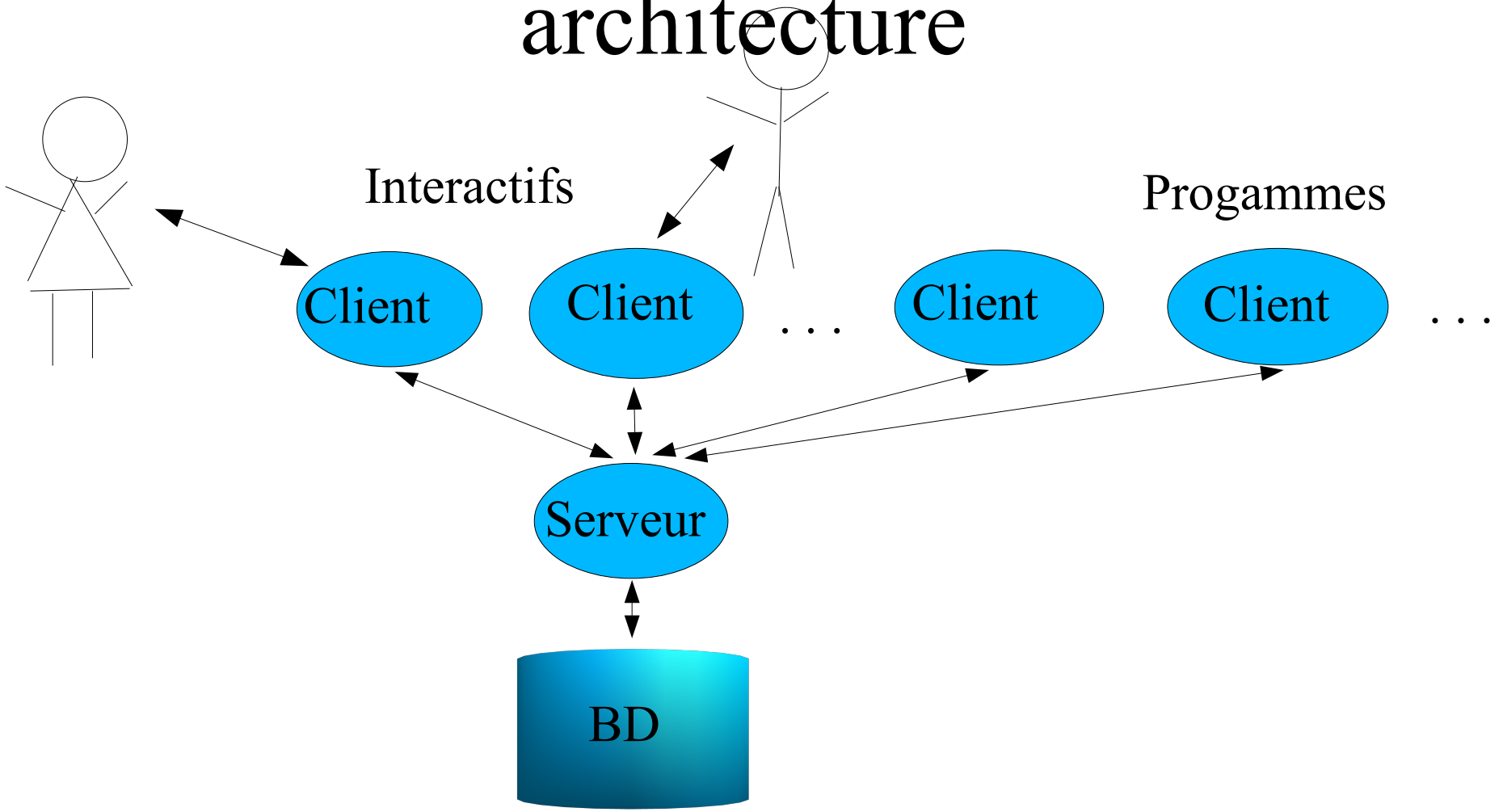
Pour chacun, on va :

- Comprendre où est le problème
- Comprendre et utiliser les outils que fournit le SGBD :
  - Les concepts pour approcher ce problème
  - Les ordres de base de données pour programmer une solution concrète à une application

Rappel : première application BD



Rappel : vue d'ensemble et architecture



Rappel : vue d'ensemble

- La bonne architecture pour gérer les problèmes BD (transparent précédent) :
  - La base de données : données
  - Le SGBD (programmes en train de s'exécuter) :
    - Le serveur
      - Unique
      - Tourne en permanence
    - Les clients (interactifs ou applications)
      - En nombre quelconque
      - Se (dé)connectent à volonté
      - Non coordonnés

vue d'ensemble : précisions

(Dessin tableau : écran, éditeur, unix waller, 3 clients sqlplus (dont autre compte SGBD), serveur SGBD, disque, espace Unix, espace SGBD, espaces utilisateurs (dont mêmes noms de table))

Compte Unix =/= compte SGBD

Contraintes

Rappel du problème

Situation :

- La manipulation des informations (humains, programmes) engendre des risque d'erreurs

Il faut :

- Date ou kilométrage ne doivent pas être négatifs
- Pour un train donné : nombre de réservations + nombre de places libres = nombre total de places
- Cas général : il faut empêcher les valeurs
  - Absurdes
  - incohérentes entre elles

définition

une *contrainte* est une propriété qui doit être vérifiée par la base de données

le *langage de contraintes* est le sous-ensemble de SQL (ordres BD) pour définir les contraintes

Analyse du problème et outils SGBD

Une mise à jour, humaine ou par programme, peut résulter en des données « incorrectes » (abîmées, non « intègres »)

Outils fournis par le SGBD :

- Pour le programmeur : le langage de contraintes (sous-ensemble de SQL) pour définir pour chaque application ce qu'il entend par « correctes »
- Comportement SGBD :
  - Détection automatique mises à jour violant contraintes déclarées
  - Refus de ces mises à jour (non effectuées)
  - attention : transaction en cours pas annulée

Le langage de contraintes

Obligation de renseigner une colonne (not null)

Unicité d'une valeur dans une colonne (unique)

Types énumérés

Clef primaire (primary key) (= unique + not null)

Clef étrangère (intégrité référentielle) (foreign key)

Inclusion entre valeurs de deux colonnes (references)

Vérification d'une formule arithmétique portant sur les lignes d'une seule table (chaque ligne indépendamment des autres) (check)

- limitations :
  - comparaison avec (par exemple) :
    - La logique du premier ordre
    - Fonctions Java
  - propriété (admise) : il existe des contraintes qu'il est impossible de définir avec SQL
- les pallier ?
  - énoncer la contrainte en français en commentaire
  - programmation : vu bientôt

ordres SQL

```
create table t (
a integer not null,
b integer unique,
c char(2),
primary key (a, c),
foreign key (b) references s(d)
)

alter table t
add (check (a + b < 12) constraint toto)
```

Séquences

Situation :

- Nécessité en pratique de clefs numériques

Outils fournis par le SGBD :

- un générateur de valeurs séquentielles (« numéros de séquence ») pour :
  - Génération d'une nouvelle valeur
  - Accès à la dernière valeur générée, pour coordonner les valeurs des clefs dans plusieurs
    - Tables
    - Lignes

Ordres SQL

Création d'un générateur (appelé une « séquence ») : create sequence nom

Utilisation d'une séquence :

- Dans select, insert, update
- nom_sequence.nextval génère la prochaine valeur (première si premier appel) (ex : insert clef primaire)
- nom_sequence.currval : renvoie la valeur courante (= celle générée par le dernier nextval de la session) (ex : insert de clef étrangère)
- Utilisable simultanément par plusieurs utilisateurs

exemple

people(num, nom, age) owner(num, voiture)

people(num) : clé primaire

owner(num) : clé étrangère

create sequence sgbd

insérer dans la base Toto, 21 ans, ayant une Ferrari

insert into people values (sgbd.nextval, 'toto', 21)

insert into owner (sgbd.currval, 'ferrari')

confidentialité

on peut donner le droit à un autre utilisateur d'utiliser une séquence

Compétences à acquérir

Analyse : savoir prédire le comportement d'une séquence d'ordres SQL en présence de contraintes (et de séquences)

Construction :

- Savoir écrire les ordres SQL adéquats pour modéliser les contraintes d'un cahier des charges
- Connaître les possibilités du langage (ex : comparaison avec les fonctions Java)

démonstration

contraintes et tests

séquences