

I. Bases de données

- Les besoins
- Qu'est ce qu'un SGBD, une BD
- Architecture d'un SGBD
- Cycle de vie
- Plan du cours

1

Exemples classiques d'applications BD

- Gestion des personnels, étudiants, cours, inscriptions, ... d'une université
- Système de réservation
- Gestion des comptes clients d'une banque
- Gestion des commandes à Amazon.com
- Gestion des jeux olympiques
- ...

2

Gestion des réservations de billets de trains

Billet =	nom client numéro train date classe no wagon numéros place départ: - gare - heure gare d'arrivée	Train =	numéro train gare départ heure départ destination finale heure d'arrivée jours
		Arrêt =	numéro train no arret gare heure départ heure d'arrivée

3

Besoins de description

- 1 - Décrire le données de l'application (trains, trajets et réservations) sans faire référence à une solution informatique particulière
👉 *modélisation conceptuelle*
- 2 - Élaborer une description équivalente pour le stockage des données dans le SGBD choisi
👉 *modélisation logique*
👉 *langage de description de données (LDD)*

4

Besoins de manipulation

3a - Créer la base de données initiale avec les données représentant le réseau SNCF

☞ *langage permettant l'insertion de données*

3b - Créer au fur et à mesure les données sur les réservations. Modifier si besoin et éventuellement supprimer toute donnée déjà rentré

☞ *langage de manipulation de données (LMD)*

(insertion, modification, suppression)

5

Besoins d'interrogation

4 - Répondre à toute demande d'information portant sur les données contenues dans la base. Par exemple:

a) Durand Julien a-t-il une réservation pour aujourd'hui ?
Si oui, donner les informations connues sur cette réservation.

b) Quels sont les horaires des trains de Bordeaux à Paris entre 9h et 10h le dimanche ?

c) Donner les destinations au départ de Bordeaux sans arrêts intermédiaires.

☞ *langage de requête (langage d'interrogation)*

6

Besoins d'exactitude / cohérence

5 - Il faut pouvoir exprimer toutes les règles qui contraignent les valeurs pouvant être enregistrées de façon à éviter toute erreur qui peut être détectée. Par exemple:

- Il ne faut jamais donner la même place dans le même train à 2 clients
- Les arrêts d'un train sont numérotés de façon continue (il ne peut y avoir pour un train donné un arrêt no 3 s'il n'y a pas un arrêt no 2 et un arrêt no 1)
- La date de réservation pour un train doit correspondre à un jour de circulation de ce train
- Le numéro de train dans une réservation / arrêt doit correspondre à un train existant
- L'heure de départ d'une gare doit être postérieure à l'heure d'arrivée dans cette gare
- L'heure d'arrivée à un arrêt doit être postérieure à l'heure de départ de l'arrêt précédent

☞ *langage d'expression de contraintes d'intégrité*

7

Besoins de garanties

6 - Il ne faut pas que les informations (par exemple, les réservations) soient perdues à cause d'un dysfonctionnement quelconque: erreur de programmation, panne système, panne de l'ordinateur, coupure de courant, ...

☞ *garantie de fiabilité*

7 - Il ne faut pas qu'une action faite pour un utilisateur (par exemple, l'enregistrement d'une réservation) soit perdue du fait d'une autre action faite simultanément pour un autre utilisateur (réservation de la même place).

☞ *garantie de contrôle de concurrence*

8

Besoins de confidentialité

- 8 - Toute information doit pouvoir être protégée contre l'accès par des utilisateurs non autorisés
- en lecture
 - en écriture

Exemple : Interdire par exemple aux clients de modifier les numéros des trains ou les horaires ou leur réservation.

👉 garantie de confidentialité

9

Besoin d'efficacité

- 9-10 Le temps de réponse du système doit être conforme aux besoins:
- en interactif: pas plus de 3 secondes
 - en programmation: assez rapide pour assumer la charge de travail attendue (nombre de transactions par jour)

👉 mécanismes d'optimisation

👉 éventuellement, répartition / duplication des données

10

Moyens

Base de données (BD)

ensemble cohérent, intégré, partagé de données structurées défini pour les besoins d'une application

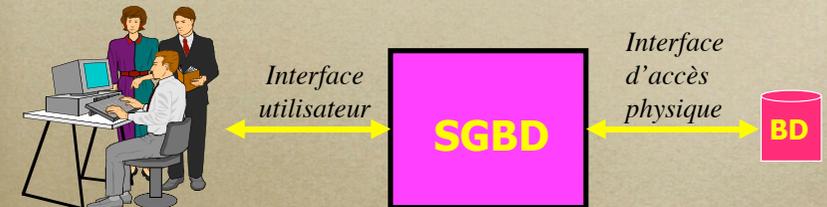
Système de gestion de base de données (SGBD)

logiciel permettant de couvrir les besoins:

- définir une représentation des informations apte à **stocker**, **interroger** et **manipuler** (insérer, supprimer, mettre à jour) de **grandes quantités de données** (plus que la mémoire vive) dont il faut garantir la **longévité** et l'**accessibilité** de manière **concurrente** (plusieurs utilisateurs simultanés) et **sûre**.

11

Architecture d'un SGBD



Analyse/vérification des requêtes
Convivialité de l'interface
Puissance des langages

Stockage / accès aux données
Optimisation des performances

12

Trois couches

Couche externe

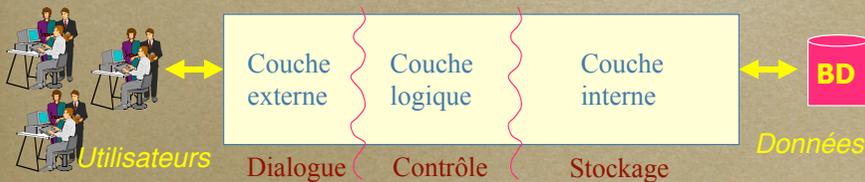
dialogue avec les utilisateurs
vues associées à chaque groupe d'utilisateurs

Couche interne

stockage des données sur des supports physiques,
gestion des structures de mémorisation (fichiers) et d'accès (gestion des index, des clés, ...)

Couche logique

contrôle global et structure globale des données



13

Principe de fonctionnement du SGBD un exemple avec le "parcours" d'une requête

- 1- Analyse syntaxique et sémantique d'une requête
- 2- Traduction au niveau logique
- 3- Contrôles de confidentialité, concurrence...
- 4- Si la requête est acceptée, optimisation et découpage en sous-requêtes élémentaires transférées au niveau interne
- 5- Au niveau interne, traduction des sous-requêtes en requêtes physiques correspondantes.



14

Cycle de vie d'une base de données

Quatre phases

- 1- Conception de la base (schéma conceptuel)
- 2- Implantation des données (schéma logique)
- 3- Utilisation (interrogation, mises à jour)
- 4- Maintenance (correction, évolution)

15

Le niveau conceptuel

Description des besoins => modèle conceptuel

- Support du dialogue concepteurs / utilisateurs
- Indépendant de la solution informatique
- Deux parties couvertes par le modèle statique (la structure des données) dynamique (règles et opérations)

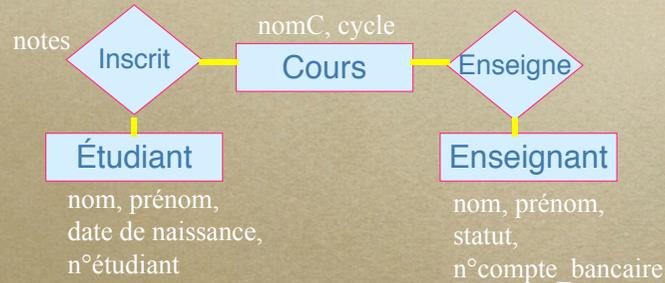
Contraintes d'intégrité

inhérentes aux données ou traduisant les règles des applications
e.g. "il ne doit pas y avoir plus de 20 % d'écart entre les salaires des employés d'un même service et d'une même catégorie"

16

Schéma conceptuel (SC) entité association

Un exemple de schéma conceptuel



Exemple de contraintes

- un cours est par un et un enseignant
- un étudiant ne peut s'inscrire qu'à au plus 6 cours

17

Implantation

Traduction du schéma conceptuel en un **schéma logique** dans les concepts du modèle utilisé par le SGBD choisi

Exemple de traduction dans le Schéma logique (SL) relationnel

- Étudiant : nom, prénom, date de naissance, n°étudiant
- Enseignant : nom, prénom, statut, n°compte_bancaire
- Cours : nomC, cycle, nom_enseignant
- Inscription : n°étudiant, nom_cours, note1, note2

18

Implantation des données

Choix des structures de stockage des données par les administrateurs système

- ✓ **Schéma interne** : description des choix d'enregistrement des données dans les fichiers.
- ✓ Fait appel à un nouveau modèle, le **modèle interne**, où les concepts sont ceux de fichier, organisation de fichier, index, chemin d'accès, clé, ...

19

Description du cours

Bases de données relationnelles

- II. Modèle conceptuel : le modèle entité-association
- III. Modèle logique : le modèle relationnel et l'algèbre relationnelle
- IV. La théorie des dépendances et la normalisation des relations
- V. Les bases de données relationnelles en pratique :
Structured Query Language (SQL)

20

Organisation

Support du cours

- Voir cours du Prof. Stefano Spaccapietra

<http://lbdwww.epfl.ch/f/teaching/courses/>

- Page web en construction



Evaluation

- **Un projet (par groupe de 4 étudiants)**
 - * semaine du 11 octobre : groupe + sujet
 - * semaine du 8 novembre : rapport de conception
 - * semaine du 13 décembre : rapport final + démonstration
- **Un examen**