

22.- Stockage et manipulation de données, des fichiers aux bases de données

Niveau : L3

Prérequis : Cours de modèle relationnel
+ aisance avec Linux

Introduction: On s'intéresse ici aux données informatiques à savoir des représentations d'informations pouvant prendre plusieurs formes, et plus précisément à leur stockage et manipulations. Plusieurs méthodes existent en fonction des besoins et de l'organisation de ces données.

I) Systèmes de fichiers

1.) Introduction

Problème 1: Comment stocker de grandes quantités de données non structurées entre elles (images, vidéos, textes,...) de manière permanente ?

Solution 2: Utiliser des disques de stockage de masse (HDD, SSD,...) et organiser les données dans des systèmes de fichiers.

Objectifs 3: On suppose que les disques considérés sont découpés en blocs, et que les opérations possibles sur ces disques sont : lire le bloc k et écrire le bloc k . On souhaite alors :

- Trouver, lire et écrire des données;
- Accéder aux mêmes données simultanément;
- Empêcher l'accès aux données d'un utilisateur aux autres utilisateurs;
- Ne pas réécrire par dessus une donnée involontairement.

Définition 4: Un **fichier** est l'unité de stockage persistante de l'information créée par un processus.

Exemples 5: fichier image par un appareil photo

- Fichier texte par un logiciel de traitement de texte (bloc-notes, libre office, word, ...)
- Fichier exécutable par gcc

Définition 6: Un **système de fichiers** est la partie du système d'exploitation gérant les fichiers. Cela désigne par métonymie l'organisation des fichiers sur un volume physique ou logique (FAT32, ext4, ...).

2.) Fichiers

Proposition 7: Un fichier est identifié, entre autres, par un nom. Celui-ci est généralement découpé en deux parties séparées par un point: son nom et son **extension**. Cette dernière est une convention de nommage pouvant être utilisée par un logiciel.

Exemples 8: .jpg → fichier image / .mp3 → fichier audio
• c → fichier contenant un programme C.

Proposition 9: Un fichier n'a pas de structure prédéfinie: son contenu n'a pas d'importance pour le système de fichiers, qui ne voit qu'une suite d'octets. La structure d'un fichier peut alors changer selon le programme utilisé: le système d'exploitation ne fournit pas d'aide mais n'impose de restriction sur leur contenu.

Proposition 10: La norme POSIX définit 7 types de fichiers, parmi lesquels les **fichiers réguliers**, les **dossiers**, **liens symboliques**...

Activité 11: Présentation des différents types de fichiers sous Linux avec des exemples et les commandes "ls -l" et "stat".

Définition 12: On appelle **mode** d'un fichier les permissions d'accès à ce fichier. Dans POSIX, les droits sont ceux de lecture, écriture et exécutions, autorisés ou non pour le propriétaire, le groupe et les autres utilisateurs.

3.) Système de fichiers

Proposition 13: Un système de fichiers a une structure arborescente, où les dossiers sont les nœuds internes, et tous les types de fichiers les feuilles.

Exemple 14: Arborescence à la racine avec "ls -R" ou "tree":

```

/
├── bin
├── etc
├── home
├── usr
├── alicia
│   ├── travail
│   │   ├── dm.txt
│   │   └── exam.tex
│   └── images
│       └── vacances.jpg
└── bob
    ...
  
```

Définition 15: Un **chemin d'accès** est la spécification du nom d'un fichier. Celui-ci peut être **absolu** en considérant le chemin de la racine jusqu'au fichier, ou **relatif** en considérant le chemin à partir du dossier courant.

Remarque 16: "." désigne le dossier courant, et ".." le dossier parent du dossier courant. Dans le cas de la racine, on prend pour convention que son parent est elle-même.

Remarque 17: Chaque composante du chemin d'accès est séparée par un **séparateur** dépendant de l'OS. Celui-ci est "/" sur les systèmes UNIX-like, "\" sur Windows ou ">" sur MULTICS.

Exemples 18: /home/alice/travail/dm.txt
 ../images/vacances.jpg.

DEV: Organisation logique du système de fichiers et 2.

II] Bases de données

Remarque 19: Les systèmes de fichiers sont très généraux: ils sont peu efficaces à organiser efficacement des données très structurées. C'est le rôle des bases de données.

Activité 20: Comparaison entre la manipulation d'une base de données par un SGBD et par le système de fichiers (fichiers & k).

1.) Rappels:

Définitions 21: Une **base de données** est un ensemble d'une ou plusieurs tables. Une **table** est une collection de lignes stockées sur un support externe (usuellement un disque de stockage).

Une **ligne** est une suite de valeurs dont le nombre et le type sont déterminés.

Exemple 22:

ID	Nom
1	Alice
2	Bob
3	Charlie

Personnes
Table

ID 1	ID 2
1	2
1	3

Amitiés
Colonne

Remarque 23: Contrairement aux fichiers, les données ici sont très ordonnées.

Ne rien écrire hors du cadre.

2.) SGBD

Définition 24: Un **système de gestion de bases de données**, raccourci en **SGBD**, est un logiciel permettant de gérer différentes bases de données.

Proposition 25: Les SGBD sont généralement à usage des utilisateurs novices comme expérimentés. Ils permettent de définir les bases de données ainsi que d'ajouter, de modifier, de supprimer et de consulter les tables, les colonnes et les lignes des bases de données gérées. Les manipulations se font par des commandes SQL.

Proposition 26: Les SGBD permettent d'organiser et de gérer les données: en plus des détails rappelés à partie 1, les SGBD utilisent des clés primaires et étrangères, des contraintes et des déclencheurs pour garantir des propriétés sur les données. Ces propriétés pourront alors être vérifiées après modification des données par le SGBD.

Exemples 27: Contraintes: Notes entre 0 et 20, âge plus grand que 8.
Déclencheurs: Calcul de moyenne, compteur, ...

Proposition 28: Dans un SGBD, les utilisateurs ne peuvent accéder ou modifier une donnée que s'ils y sont autorisés. Ces droits appelés **privileges**, peuvent être donnés aux utilisateurs, généralement par la ou les personnes gérant le SGBD.

Remarque 29: Il existe d'autres types de privileges, comme pouvoir donner des privileges.

Proposition : les SGBD ont des mesures protégeant contre les corruptions de données. Par ailleurs, ils ont également des politiques de gestion de concurrence, permettant à plusieurs utilisateurs de consulter et modifier les données en parallèle.

DEV: Gestion de la concurrence dans un SGBD.

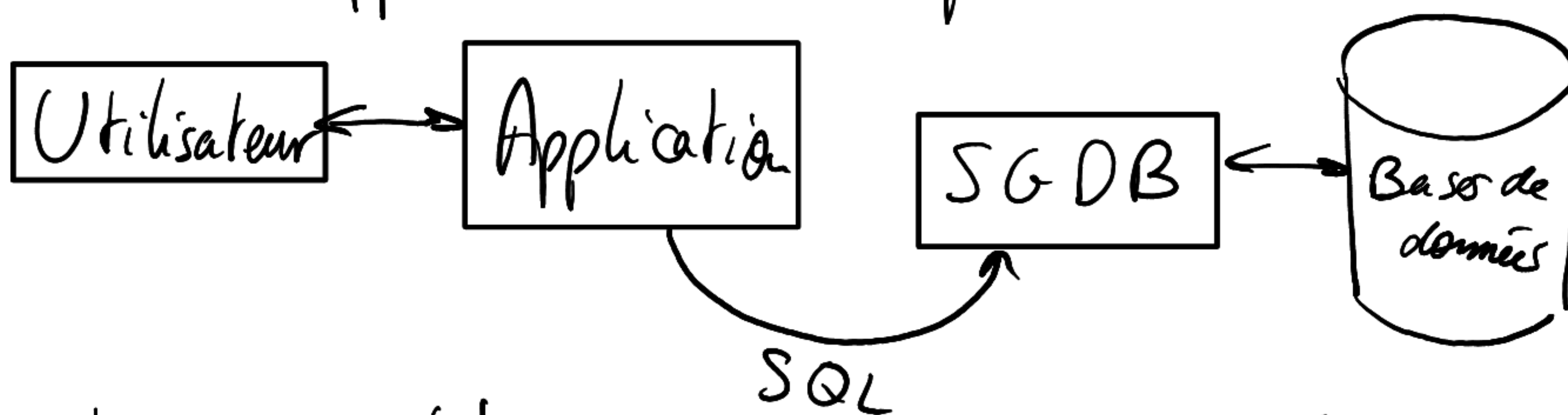
3.) Architecture des SGBD

Proposition 30. Usuellement, les SGBD sont des serveurs, au sens du protocole client-serveur. Dans ce cadre, d'autres applications agissent en temps que client du SGBD en demandant la gestion de leurs données.

Exemple 31: PostgreSQL (SGBD) comme serveur

Nextcloud / Matrix (Synapse) comme clients.

Applications consultables par des utilisateurs.



Proposition 32 : Un SGBD va traiter les requêtes SQL de la manière suivante :

- Réception de la requête SQL.
- Analyse de la requête : syntaxe et cohérence avec la structure.
- Traduction de la requête en un plan d'accès puis est efficace et optimisée.
- Interprétation et exécution du plan d'accès.
- Retour au programme ou à l'utilisateur.