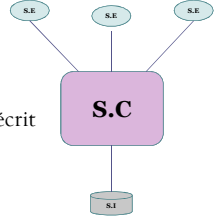


Introduction	Architectures	Objectifs	Fonctions	Historique	Principaux types
--------------	---------------	-----------	-----------	------------	------------------

SCHÉMA CONCEPTUEL

Description des données d'une entreprise ou d'une partie d'une entreprise en termes de types d'objets et de liens logiques indépendants de toute représentation en machine.



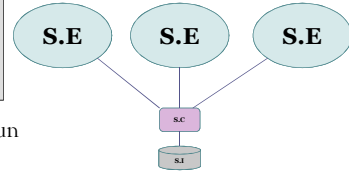
- ♦ Appelé aussi schéma logique, c'est un niveau intermédiaire qui décrit la structure générale de la base de données à tous les utilisateurs :
 - ✓ Quels sont les objets du monde réel à gérer ;
 - ✓ Les associations entre les objets et les règles de gestion ;
- ♦ C'est le résultat d'une analyse qui appartient au domaine des méthodes de conception qui fait partie du génie logiciel, par exemple: la méthode MERISE (que vous allez étudier dans l'E.M: Analyse et conception des systèmes d'information) ;
- ♦ Ce schéma décrit la structure de la base de données indépendamment de sa représentation physique et de la façon dont chaque groupe d'utilisateurs utilisent la BD.

25

Introduction	Architectures	Objectifs	Fonctions	Historique	Principaux types
--------------	---------------	-----------	-----------	------------	------------------

SCHÉMA EXTERNE

Description d'une partie de la base de données extraite ou calculée à partir de la base physique, correspondant à la vision d'un programme ou d'un utilisateur, donc à un arrangement particulier de certaines données.



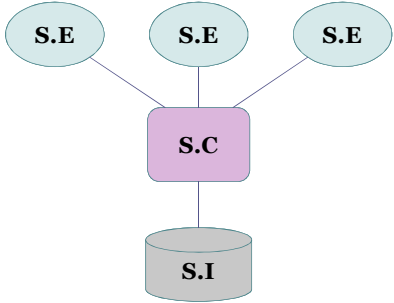
- ♦ Les schémas externes ou les Vues, décrivent comment un groupes d'utilisateurs voient les données:
 - ✓ Un étudiant ne consulte que les notes qui le concerne;
 - ✓ Un professeur ne consulte/MAJ que les notes de ses matières;
 - ✓ Un directeur peut consulter les notes de tous les étudiants de l'école;
 - ✓ Etc.
- ♦ C'est un sous-schéma d'une BD (une sous-base virtuelle ou partielle) à laquelle un groupe d'utilisateurs a le droit d'accéder (sécurité des données).

26

Introduction	Architectures	Objectifs	Fonctions	Historique	Principaux types
--------------	---------------	-----------	-----------	------------	------------------

CONCLUSION

L'architecture ANSI/SPARC offre aux SGBD la possibilité d'apporter des modifications à un niveau sans que cela interagisse (le moins possible) sur les autres niveaux.

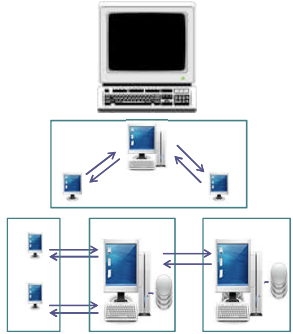


27

Introduction	Architectures	Objectifs	Fonctions	Historique	Principaux types
--------------	---------------	-----------	-----------	------------	------------------

ARCHITECTURES OPÉRATIONNELLES DES SGBD

- ♦ Les principales architectures opérationnelles:
 - ✓ Architecture centralisée (Mainframe 1960);
 - ✓ Architecture client-serveur (Depuis les années 80);
 - ✓ Architecture base de donnée répartie (Depuis les années 90).




28

Introduction | Architectures | Objectifs | Fonctions | Historique | Principaux types

ARCHITECTURE CENTRALISÉE

- Dans une architecture centralisée la même machine comporte:
 - ✓ La base de données;
 - ✓ Le SGBD gérant la BD;
 - ✓ Les programmes utilisateurs.
- La BD existe sur une seule machine, en plus elle est accessible seulement à partir de cette machine;



29

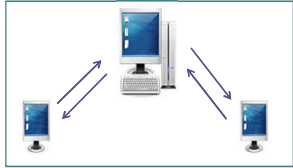
Introduction | Architectures | Objectifs | Fonctions | Historique | Principaux types

ARCHITECTURE CLIENT/SERVEUR

Définition:

Architecture hiérarchisée mettant en jeu d'une part un serveur de données gérant les données partagées en exécutant le SGBD, d'autre part des clients supportant les applications et la présentation, et dans laquelle les clients dialoguent avec les serveurs via un réseau en utilisant des requêtes de type question-réponse.


- L'architecture C/S permet une répartition hiérarchique des tâches:
 - ✓ La BD est stockée sur le serveur qui se charge de la partagée entre les postes clients en exécutant le SGBD;
 - ✓ Sur les postes clients sont installés les programmes utilisateurs permettant de manipuler la BD;
 - ✓ Les communications entre les postes clients et le serveur sont assurées par des protocoles standardisés.



30

Introduction | Architectures | Objectifs | Fonctions | Historique | Principaux types

FONCTIONNEMENT D'UN SYSTÈME CLIENT/SERVEUR



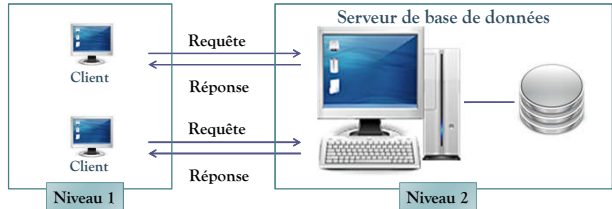
- Le client émet une requête au serveur via son adresse (IP) sur un Port (un service ou une application) particulier;
- Le serveur reçoit la requête et envoie une réponse au client via son adresse et son port.

Port	Application
80	Consultation d'une page web utilisant le protocole Hypertext Transfer Protocol (HTTP)
21	Transfert de fichier utilisant le protocole File Transfer Protocol (FTP)
25	envoi d'un mail utilisant le protocole Simple Mail Transfer Protocol (SMTP)

31

Introduction | Architectures | Objectifs | Fonctions | Historique | Principaux types

ARCHITECTURE CLIENT/SERVEUR À 2 NIVEAUX



- Dans une architecture à 2 niveaux (aussi appelée architecture 2-tier) les clients demandent des données et le serveur leur fournit directement ces données en utilisant sa propre BD (la BD existe sur le serveur).

32

Introduction | Architectures | Objectifs | Fonctions | Historique | Principaux types

ARCHITECTURE CLIENT/SERVEUR À 3 NIVEAUX

➤ Dans l'architecture à 3 niveaux (3-tier) nous avons:

- ✓ Des clients qui demandent des données;
- ✓ Un serveur d'application (middleware) qui ne contient pas la BD, chargé de fournir les données demandées par les clients mais faisant appel à un autre serveur;
- ✓ Le serveur de BD, fournissant au serveur d'application les données demandées par les utilisateurs.

33

Introduction | Architectures | Objectifs | Fonctions | Historique | Principaux types

ARCHITECTURE BD RÉPARTIE

➤ Dans l'architecture BD répartie nous avons:

- ✓ Des clients qui demandent des données;
- ✓ Plusieurs serveurs coopérant à la gestion la BD composées de plusieurs sous-bases, chacune de ces sous-bases est gérée par un serveur, ces sous-bases apparaissant comme une BD unique centralisée pour l'utilisateur;

➤ Une architecture BD répartie est une généralisation de l'architecture Client/Serveur.

34

Introduction | Architectures | Objectifs | Fonctions | Historique | Principaux types

AVANTAGES D'UNE ARCHITECTURE C/S

➤ L'architecture client/serveur est particulièrement recommandée pour des réseaux nécessitant un grand niveau de fiabilité, elle présente les avantages suivants :

- ✓ **Ressources centralisées:** étant donné que le serveur est au centre du réseau, alors la base de données est centralisée ce qui facilite son partage entre tous les utilisateurs et évite les problèmes de redondance;
- ✓ **Meilleure sécurité:** les données sont centralisées sur le serveur, ce qui demande de bien gérer la sécurité juste au niveau du serveur;
- ✓ **Administration au niveau serveur :** pas besoin d'administrer les clients;
- ✓ **Réseau évolutif:** possibilité de supprimer ou ajouter des clients dans le réseau sans perturber son fonctionnement.

35

Introduction | Architectures | Objectifs | Fonctions | Historique | Principaux types

INCONVÉNIENTS D'UNE ARCHITECTURE C/S

➤ L'architecture client/serveur a tout de même quelques faiblesses parmi lesquelles :

- ✓ **Coût élevé:** la mise en place d'un serveur demande un budget important;
- ✓ **Maillon faible:** le serveur est le point central d'un réseau client/serveur, si le serveur tombe en panne, tout le réseau est en panne.

36

Introduction	Architectures	Objectifs	Fonctions	Historique	Principaux types
--------------	---------------	-----------	-----------	------------	------------------

OBJECTIFS D'UN SGBD

♦ Les principaux objectifs d'un SGBD sont:

1. Indépendance physique;
2. Indépendance logique;
3. Manipulation des données par des langages non procéduraux;
4. Administration facilitée des données;
5. Efficacité des accès aux données;
6. Redondance contrôlée des données;
7. Cohérence des données;
8. Partage des données;
9. Sécurité des données.

37

Introduction	Architectures	Objectifs	Fonctions	Historique	Principaux types
--------------	---------------	-----------	-----------	------------	------------------

INDÉPENDANCE PHYSIQUE

Pas besoin de s'occuper directement de la gestion des fichiers physiques, la modification de l'organisation des données sur les supports de stockage (S.I) ne doit pas répercuter sur les données et leurs associations (S.C) et sur les vues utilisateurs (S.E).

♦ Il s'agit de pouvoir modifier le S.I en fonction des performances et de l'évolution des technologies informatiques sans toucher aux autres schémas;

♦ Exemples:

- ✓ Créer des index pour un fichier, trier un fichier, éclater un fichier, ou fusionner des fichiers;
- ✓ Changer ou mettre à jour le OS, changer le SF(Système de Fichiers).

38

Introduction	Architectures	Objectifs	Fonctions	Historique	Principaux types
--------------	---------------	-----------	-----------	------------	------------------

INDÉPENDANCE LOGIQUE

Permettre aux différentes applications et utilisateurs d'avoir des vues différentes des mêmes données(indépendance entre les utilisateurs, chacun voyant une partie de la BD via son S.E, selon une structuration particulière).
Permettre au DBA de modifier le S.C sans toucher les vues non concernés.

♦ Les utilisateurs ont des besoins différents, alors:

- ✓ Chaque utilisateur doit avoir des vues propres à ses besoins d'informations;
- ✓ Il doit être possible d'ajouter ou de supprimer des données dans les S.E et le S.C sans modifier les S.E non concernés.

39

Introduction	Architectures	Objectifs	Fonctions	Historique	Principaux types
--------------	---------------	-----------	-----------	------------	------------------

MANIPULATION DES DONNÉES PAR DES LANGAGES NON PROCÉDURAUX

Si les objectifs d'indépendance sont atteints, alors les utilisateurs voient les données indépendamment de leur représentation physique sur le support de stockage. Ils doivent pouvoir gérer les données au moyen des langages non procéduraux c'est-à-dire décrire les données qu'ils souhaitent retrouver ou mettre à jour sans dire la manière de les retrouver ou les mettre à jour qui est propre à la machine.

♦ Il s'agit d'avoir la possibilité d'interroger et de mettre à jour les données par des langages de haut niveau spécifiant les données que l'on veut manipuler (de quoi) et non pas comment y accéder (sans préciser d'algorithme d'accès);

♦ Exemples:

- ✓ Un utilisateur non expérimenté utilise des langages interactifs de haut niveau que l'on appelle classiquement langages de requêtes (SQL: Structured Query Language);
- ✓ Un programmeur utilise des langages de programmation(C, C++, JAVA, PHP...) exprimant les commandes fournis par le SGBD nécessaires pour pouvoir accéder à la BD.

40

Introduction	Architectures	Objectifs	Fonctions	Historique	Principaux types
--------------	---------------	-----------	-----------	------------	------------------

ADMINISTRATION FACILITÉE DES DONNÉES

Un SGBD doit fournir des outils pour décrire les données, à la fois leurs structures de stockage (type, longueur...) et leurs présentations externes. Les fonctions qui permettent de définir les données et de changer leur définition sont appelées outils d'administration des données. L'administration facilitée des données consiste à centraliser ces fonctions entre les mains d'un petit nombre d'utilisateurs qui sont les DBA.

- ♦ Il s'agit de permettre :
 - ✓ Un contrôle efficace des données;
 - ✓ Une résolution des conflits entre divers point de vue d'utilisateurs;
 - ✓ Une optimisation des accès aux données;
 - ✓ Une optimisation des moyens informatiques.

41

Introduction	Architectures	Objectifs	Fonctions	Historique	Principaux types
--------------	---------------	-----------	-----------	------------	------------------

EFFICACITÉ DES ACCÈS AUX DONNÉES

Un SGBD doit partager les ressources (CPU, unités d'entrées-sorties) entre les utilisateurs en optimisant l'utilisation globale de ces ressources.

- ♦ Un SGBD doit être capable de gérer un volume important de données et d'offrir un temps d'accès raisonnable aux utilisateurs:
 - ✓ Les entrée/sorties disques sont optimisés par l'utilisation des mémoires tampon en CPU, et par l'utilisation des mémoires caches des disques;
 - ✓ Une requête courte d'un utilisateur n'attend pas une requête longue d'un autre utilisateur ce qui donne un bons temps de réponse .

42

Introduction	Architectures	Objectifs	Fonctions	Historique	Principaux types
--------------	---------------	-----------	-----------	------------	------------------

REDONDANCE CONTRÔLÉE DES DONNÉES

Afin d'éviter les problèmes lors des mises à jour, un SGBD doit veiller à ce que chaque donnée ne doit être présente qu'une seule fois dans la BD.

- ♦ La redondance de données provoque le gaspillage de l'espace mémoire (support de stockage) et diminue les performances du système (opérations de mise à jour multiples), l'administration centralisée des données conduisait donc naturellement à la non-redondance physique des données afin d'éviter les mises à jour multiples.

43

Introduction	Architectures	Objectifs	Fonctions	Historique	Principaux types
--------------	---------------	-----------	-----------	------------	------------------

COHÉRENCE DES DONNÉES

Les données sont soumises à un certain nombre de règles qui définissent un état cohérent de la BD, Le SGBD doit connaître ces règles et les vérifier automatiquement lors de l'insertion, la modification ou la suppression des données, pour assurer la cohérence de la BD.

- ♦ Les règles que doivent explicitement ou implicitement suivre les données au cours de leur évolution sont appelées contraintes d'intégrité;
- ♦ Exemples:
 - ✓ La note d'un étudiant doit être entre 0 et 20;
 - ✓ Pour pouvoir valider un module il faut avoir une moyenne supérieure ou égale à 12;

44

Introduction	Architectures	Objectifs	Fonctions	Historique	Principaux types
--------------	---------------	-----------	-----------	------------	------------------

PARTAGE DES DONNÉES

L'objectif de partage est de permettre à plusieurs utilisateurs d'accéder simultanément aux données de la BD comme s'ils étaient seuls à les utiliser sans attendre mais aussi sans savoir qu'un autre utilisateur peut les modifier concurremment.

- ♦ Il s'agit de permettre à plusieurs utilisateurs d'accéder aux mêmes données au même temps ;
- ♦ Cela simple à gérer lorsqu'il s'agit uniquement de consultation (visualisation), mais ce n'est pas le cas lorsqu'il s'agit d'une mise à jour multiutilisateur par exemple :
 - ✓ La réservation sur le même avion le même siège pour plusieurs passagers différents;
- ♦ La mise à jour d'une donnée par plusieurs utilisateurs ne doit pas provoquer des collisions. Le SGBD doit gérer ces conflits liés au partage des données.

45

Introduction	Architectures	Objectifs	Fonctions	Historique	Principaux types
--------------	---------------	-----------	-----------	------------	------------------

SÉCURITÉ DES DONNÉES

Les données doivent être protégées contre les accès non autorisés. La sécurité des données doit aussi être assurée en cas de panne d'un programme ou du système, voire de la machine.

- ♦ Il s'agit de pouvoir associer à chaque utilisateur des droits d'accès aux données par exemple:
 - ✓ Un étudiant pourra consulter ses notes mais pas les notes des autres étudiants;
- ♦ Un SGBD doit être capable de restaurer les données après panne d'ordinateur à partir de sauvegardes;
- ♦ Si un groupes de commandes de mises à jour qui doit être exécuté totalement est interrompu par une panne, le SGBD doit assurer l'intégrité de la BD est donc soit exécuter le groupe soit annuler totalement (notion de transaction). Exemple :
 - ✓ Que faire si une panne est survenue lors d'une opération de transfert d'argent d'un compte vers un autre.

46

Introduction	Architectures	Objectifs	Fonctions	Historique	Principaux types
--------------	---------------	-----------	-----------	------------	------------------

FONCTIONS D'UN SGBD

♦ En résumé, un SGBD a pour fonctions:

1. Description des données;
2. Recherche de données;
3. Mise à jour des données;
4. Transformation des données;
5. Contrôle de l'intégrité des données;
6. Sécurité des données.

47

Introduction	Architectures	Objectifs	Fonctions	Historique	Principaux types
--------------	---------------	-----------	-----------	------------	------------------

DESCRIPTION DES DONNÉES

Un SGBD doit offrir des interfaces pour définir et décrire les données. La définition des différents schémas est effectuée par le DBA ou par les personnes jouant le rôle d'administrateur.

- ♦ Un SGBD fournit un langage de définition de données LDD (en anglais Data Definition Language DDL) permettant de définir les schémas d'une BD;
- ♦ Exemple de commandes LDD :
 - ✓ CREATE: création d'une structure de données;
 - ✓ ALTER: modification d'une structure de données;
 - ✓ DROP: suppression d'une structure de données;
 - ✓ RENAME: renommer une structure de données.

48

Introduction	Architectures	Objectifs	Fonctions	Historique	Principaux types
--------------	---------------	-----------	-----------	------------	------------------

RECHERCHE DE DONNÉES

Un SGBD doit offrir des commandes de recherche de données permettant de retrouver les données par le contenu sans préciser la procédure d'accès.

- Un SGBD fournit un Langage d'Interrogation des Données LID (en anglais Data Query Language DQL) permettant d'exprimer toutes les questions que l'on peut poser à une base de données;
- Exemple:
 - ✓ Trouver les étudiants dont le nom de famille commence par "M":

```
SELECT CIN, Nom, Prenom FROM Etudiant WHERE Nom LIKE "M%";
```

49

Introduction	Architectures	Objectifs	Fonctions	Historique	Principaux types
--------------	---------------	-----------	-----------	------------	------------------

MISE À JOUR DES DONNÉES

Un SGBD doit offrir des commandes de mise à jour de données permettant l'insertion de données dans la BD, la modification de données et la suppression de données.

- Un SGBD fournit un Langage de Manipulation de Données LMD (en anglais Data Manipulation Language DML) permettant de mettre à jour les données d'une BD;
- Exemple de commandes LMD :
 - ✓ INSERT: insertion des données;
 - ✓ UPDATE: modification des données;
 - ✓ DELETE: suppression des données.

50

Introduction	Architectures	Objectifs	Fonctions	Historique	Principaux types
--------------	---------------	-----------	-----------	------------	------------------

TRANSFORMATION DES DONNÉES

Un SGBD doit pouvoir assurer le passage des données depuis le format d'un niveau (S.I, S.C ou S.E) dans le format d'un autre niveau. On appelle transformation de données la fonction effectuant la restructuration de données conforme à un schéma en données conforme à un autre schéma.

- Un SGBD transforme les données en créant des objets virtuels nommés vues (views) qui agissent comme des fenêtres sur la BD, ces objets correspondent au S.E (autrement dit les vues) qui présente la partie visible de la BD pour chaque utilisateur. En théorie, les utilisateurs doivent accéder aux données en interrogeant les vues, ces dernières masquant le schéma de la BD;
- Un SGBD fournit des commandes permettant la gestion de la BD en utilisant les vues.

51

Introduction	Architectures	Objectifs	Fonctions	Historique	Principaux types
--------------	---------------	-----------	-----------	------------	------------------

CONTRÔLE DE L'INTÉGRITÉ DES DONNÉES

Un SGBD doit assurer la cohérence des données c'est-à-dire que les données doivent respecter leur type et des règles d'intégrités, en plus un SGBD doit éviter la redondance des données.

- Un SGBD fournit des commandes appartenant aux LDD permettant d'appliquer des règles sur les données de la BD;
- Exemple :
 - ✓ CHECK (note BETWEEN 0 AND 20) : application d'une contrainte pour s'assurer que les notes saisies par les utilisateurs comprises entre 0 et 20;
 - ✓ Cin CHAR(10) UNIQUE: application d'une contrainte pour s'assurer la non redondance de la cin dans la BD.

52

Introduction	Architectures	Objectifs	Fonctions	Historique	Principaux types
--------------	---------------	-----------	-----------	------------	------------------

SÉCURITÉ DES DONNÉES

Un SGBD doit offrir des commandes permettant d'attribuer les droits d'accès aux utilisateurs de la BD et des commandes permettant d'assurer l'exécution totale d'un groupe de commandes de mises à jour (Transaction) ou l'annulation de son exécution en cas de panne.

- ♦ Un SGBD fournit un Langage de Contrôle de Données LCD (en anglais Data Control Language DCL) pour gérer les utilisateurs (création des utilisateurs et affectation de leurs droits) et de contrôler l'exécution des transactions;
- ♦ Exemple de commandes LCD :
 - ✓ GRANT : autorisation d'un utilisateur à effectuer une action ;
 - ✓ DENY : interdiction à un utilisateur d'effectuer une action ;
 - ✓ REVOKE : annulation d'une commande de contrôle de données précédente ;
 - ✓ COMMIT : validation d'une transaction en cours ;
 - ✓ ROLLBACK : annulation d'une transaction en cours ;
 - ✓ LOCK : verrouillage sur une structure de données.

53


Introduction	Architectures	Objectifs	Fonctions	Historique	Principaux types
--------------	---------------	-----------	-----------	------------	------------------

HISTORIQUE DES SGBD

- ♦ Jusqu'aux années 60 (Avant l'apparition des SGBD): organisation classique des données en fichiers gérés par des Systèmes de Gestion de Fichiers (SGF);
 - ✓ Exemple: le langage COBOL (COmmon Business Oriented Language).
- ♦ Les années 60-70 (1^{ère} Génération des SGBD) : apparition des premiers SGBD qui permettent la séparation de la description des données de leur manipulation par les programmes d'application ;

Exemples :

 - ✓ IMS (Information Managment System) SGBD basé sur le modèle Hiérarchique lancé par la société IBM en 1966, pour le programme spatial Apollo;
 - ✓ IDS (Integrated Data Store) SGBD basé sur le modèle réseau développé chez la société General Electric par Charles William Bachman.



54

Introduction	Architectures	Objectifs	Fonctions	Historique	Principaux types
--------------	---------------	-----------	-----------	------------	------------------

HISTORIQUE DES SGBD

- ♦ Les années 70-80 (2^{ème} Génération des SGBD) : SGBD basées sur le modèle relationnel, les données sont manipulées par des langages non procéduraux (SEQUEL: Structured English QUERy Language , SQL Structured Query Language).

Exemples :

 - ✓ Ingres : SGBD basé sur le modèle Relationnel développé au sein de l'université de Berkeley à partir de 1973;
 - ✓ System R : SGBD basé sur le modèle Relationnel développé par la société IBM à partir de 1974;
- ♦ Les années 90 (3^{ème} Génération des SGBD): modèles de données extensibles intègre le relationnel et l'objet;

Exemples :

 - ✓ INGRES, SYBASE, INFORMIX, DB2, OBJECTSTORE, O2...
- ♦ L'objet n'a pas remplacé le relationnel, Le marché est dominé par les SGBD relationnel.

55

Introduction	Architectures	Objectifs	Fonctions	Historique	Principaux types
--------------	---------------	-----------	-----------	------------	------------------

HISTORIQUE DES SGBD

- ♦ Aujourd'hui: selon le site (www.planetoscope.com) chaque seconde 29 000 Go de données sont publiés dans le monde, soit 2,5 Exaotets (1Eb=1 000 000 000 Go) par jour:
 - ✓ Variété de données (textes, sons, images, vidéo, ...);
 - ✓ Augmentation considérable de volumes de données manipulées (ceux en rapport avec Internet): données scientifiques, réseaux sociaux, indicateurs économiques et sociaux, etc.;
 - ✓ La généralisation des TIC a favorisée une multiplication exponentielle de données (Big Data);
- ♦ La gestion et le traitement de ces volumes énorme de données présente un nouveau défi pour l'informatique.

56