

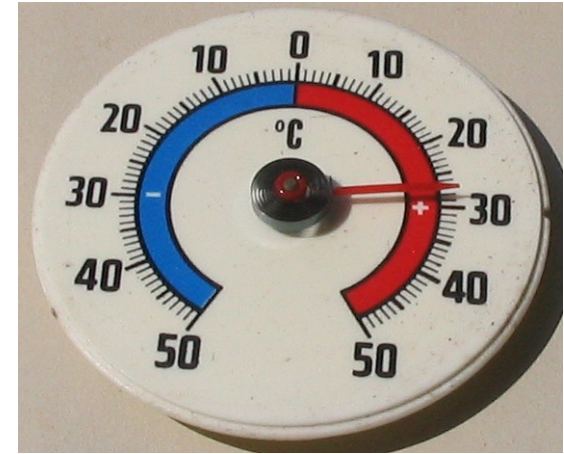
I. Informatique et ses fondements :

1ère partie : le codage binaire

ICN : 1. Le codage binaire

- 1. Représentation de l'information sous forme de bits**
- 2. Stocker les images ou les sons**
- 3. Compression**
- 4. Organisation des données**
- 5. Bases de données**

Pourquoi coder l'information ?



- Informations diverses, hétérogènes, plus ou moins structurées
 - Perçues par nos sens humains ou par des instruments
- Comment les représenter ?

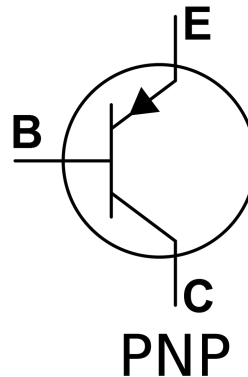
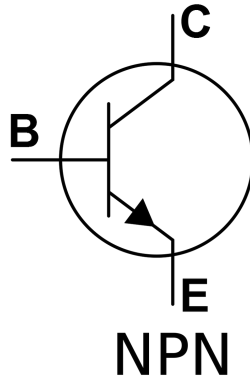
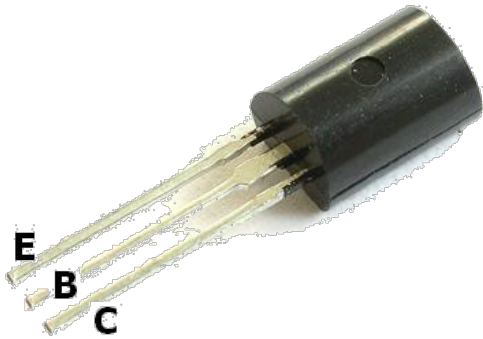
Numérisation et codage numérique

Au niveau électronique

- Soit le courant électrique passe, soit il ne passe pas
- **Codage binaire** avec des 0 ou des 1
 - **Bit** = *Binary Digit*
- Ordinateurs actuels basés sur l'électronique
 - Bientôt la photonique ou le quantique ?

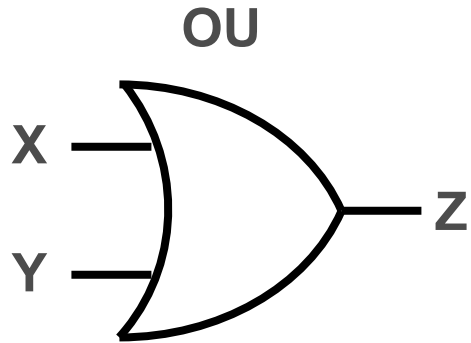
Opérations sur les bits

- Transistor = Télérupteur miniature
 - Robinet pour commander le passage du courant



Opérations sur les bits

- Avec les transistors, on construit des portes logiques
 - ET, OU, OU exclusif, NON, ...



X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

- Puis des additionneurs de bits, ...

Coder les nombres

- Bit limité à 2 valeurs
 - Assembler les bits pour coder des grands nombres
 - ✓ Un octet = 8 bits = 256 valeurs (de 0 à 255)
- Pour coder un âge, il suffit de 7 bits (de 0 à 127)
- Pour coder une température, il en faut plus
 - Et des nombres négatifs
 - Et des nombres à virgules

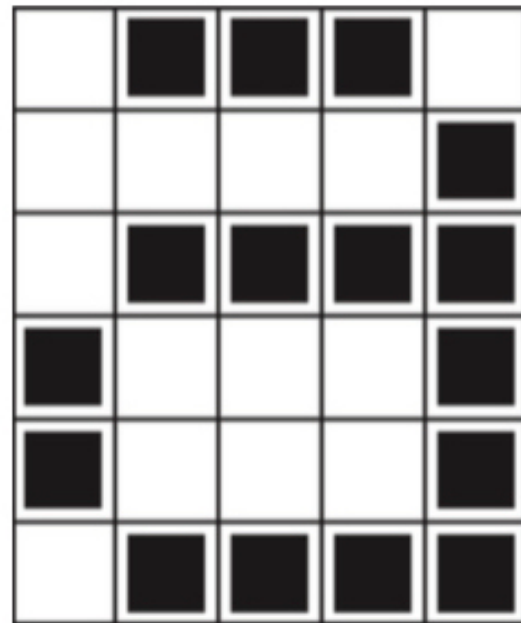
**L'ordinateur sait manipuler
les principaux types de nombres**

Peu importe ce qu'il y a dedans

- L'ordinateur sait manipuler les nombres
- Mais il se fiche de leur signification
- C'est le programmeur ou l'utilisateur qui doit lui donner ces nombres en entrée de manière adaptée
 - Quelle unité de mesure ?
- Et correctement interpréter les résultats

Coder du texte

- De nombreuses façons de stocker une lettre
 - Un bit par case à afficher ?
 - Quelle largeur et quelle hauteur ?
 - Et si on ajoute de la couleur ?
- Code ASCII
 - Numéro 8bits = 256 caractères possibles
 - ✓ Étendu par Unicode pour gérer toutes les langues



**Le dessin du caractère (la police)
n'est nécessaire qu'à l'affichage**

Ce qu'il faut retenir

- Le courant électrique code un bit 0 ou 1
- On assemble ces bits pour représenter des nombres, voire des lettres, etc.
- Des circuits électroniques effectuent des calculs avec ces nombres
 - Sans savoir ce qu'ils signifient

Illustrations

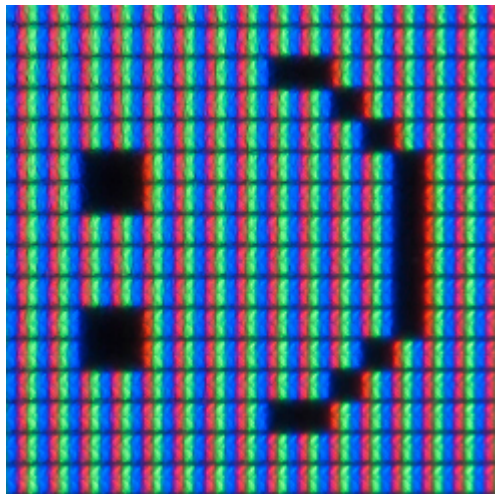
- p. 2 : [20050501_1315_2558-Bimetall-Zeigerthermometer.jpg](#) CC-BY-SA-3.0 (<http://creativecommons.org/licenses/by-sa/3.0/>) , via Wikimedia Common
- p.2 : Little visuals : <https://stocksnap.io/author/401>

ICN : 1. Le codage binaire

1. Représentation de l'information sous forme de bits
- 2. Stocker les images ou les sons**
3. Compression
4. Organisation des données
5. Bases de données

Les pixels

- Jeux vidéos, télévision numérique, réalité virtuelle, ...
 - Quelle technologie pour l'affichage ?



- Des **Pixels** (*Picture Element*)
 - Composés de 3 couleurs (Rouge Vert Bleu)
- Servent aussi pour la retouche, l'impression, etc.

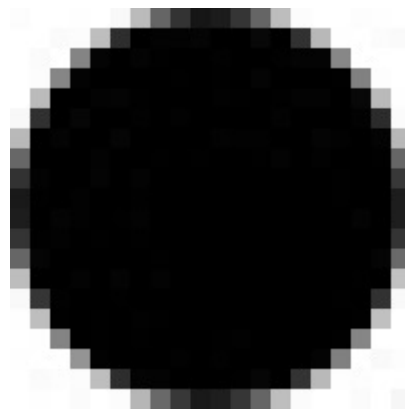
Codage des pixels

- Suite de bits codant chaque pixel
 - 1 bit par pixel en noir et blanc
 - 8 bits pour 256 niveaux de gris
 - 24 bits pour les couleurs (8 bits par composante)
- Comment savoir si une suite de bits correspond à 24 pixels noir et blanc ou à 1 seul pixel en couleur ?
- Comment savoir si l'image est rectangulaire ou carrée ?

**Il faut un en-tête précisant
le codage des pixels
et les dimensions de l'image**

Images bitmaps ou vectorielles ?

- L'ordinateur n'a aucune idée du contenu d'une image
 - Pour savoir qu'il y a un disque, il faut un algorithme de reconnaissance de forme
- Comment mieux coder de telles images ?
 - Décrire le disque serait plus facile
 - ✓ Et beaucoup plus concis !



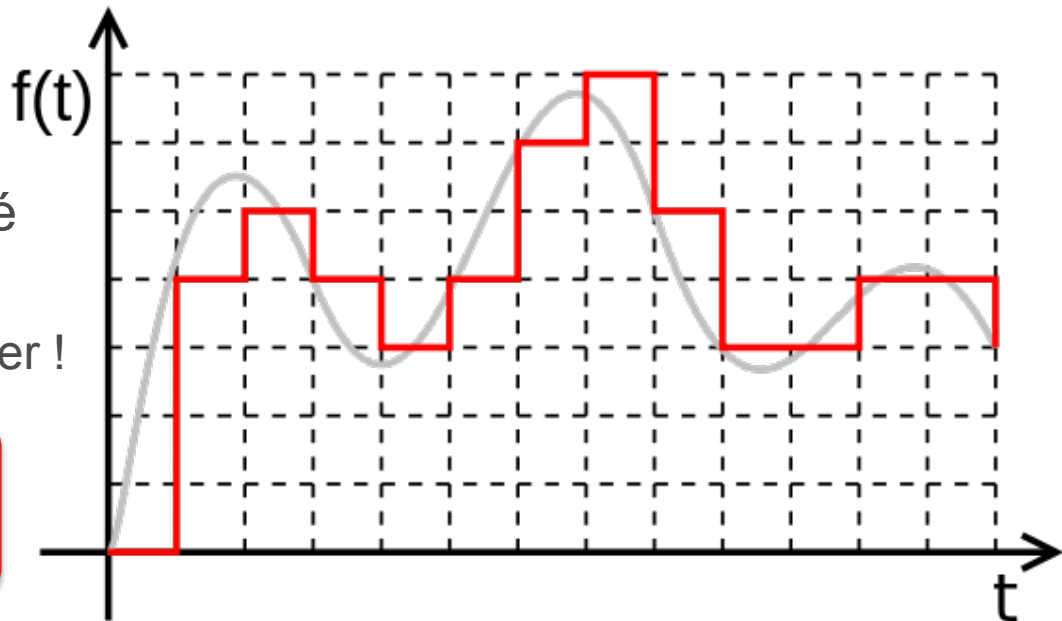
**Une image vectorielle est
un ensemble de formes mathématiques
avec leurs couleurs**

- Bien pour le dessin technique, pas pour les photos !

Codage du son

- Signal composé d'une infinité de valeurs intermédiaires
 - On ne peut pas toutes les coder !

Échantillonnage



- Quelques valeurs approximatives
 - Suffisamment proches pour que l'oreille humaine n'entende pas la différence
- Un paquet de bits pour chaque valeur
 - Avec un en-tête pour préciser la fréquence

Codage « vectoriel » du son

- Un morceau de musique, c'est aussi une partition
 - Facile à coder en bits
- Et des caractéristiques d'instruments
 - Type, timbre, ...
- On code la façon de générer la musique plutôt que le résultat
 - Similaire à l'idée des images vectorielles



Ce qu'il faut retenir

- Images et sons peuvent être codés par des suites de nombres, de bits
- Un programme permet à l'ordinateur d'interpréter ce codage pour afficher ou imprimer une image, jouer un son, ...
- ... et de coder des données lors d'un scan d'image, d'un enregistrement de son, ...
- Entre les deux (stockage sur le disque, transfert sur Internet, retouche photo ou audio, ...), les données peuvent rester codées

Illustrations

- p.2 : Six Fingers © *Inria* - *J.C. Moschetti*

ICN : 1. Le codage binaire

1. Représentation de l'information sous forme de bits
2. Stocker les images ou les sons
3. **Compression**
4. Organisation des données
5. Bases de données

Pourquoi de la compression ?

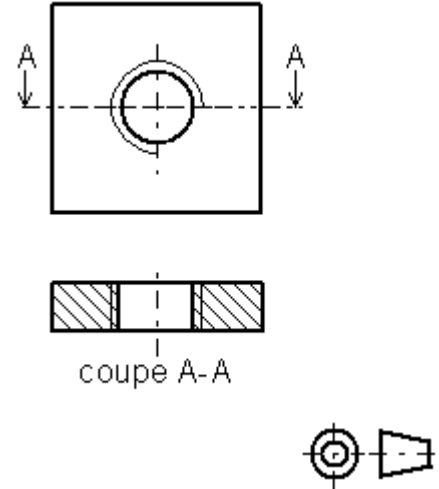
- Imaginons un film Full HD de 2h (sans le son)
- 7200 secondes x 24 images par seconde = **172 800 images**
- 1920x1080 pixels = **2 073 600 pixels par image**
- 24 bits pour les couleurs de chaque pixel
- Il faut 8 599 633 920 000 bits pour stocker la vidéo
 - Soit **1 Téraoctet !**
 - 100 heures de téléchargement sur ADSL
 - ✓ Pas idéal pour la vidéo à la demande...



**Comment fait-on tenir un tel film
sur un DVD de quelques gigaoctets ?**

Compresser les images

- Certaines images ont des zones répétitives
 - Couleurs uniformes, parties identiques, ...
 - Inutile de stocker plusieurs fois la même information
 - ✓ Permet de réduire significativement la taille des images « simples »
 - ✓ Format d'image GIF, PNG, ...
- On peut appliquer le même principe à du texte, du son, etc.



Compresser les photos

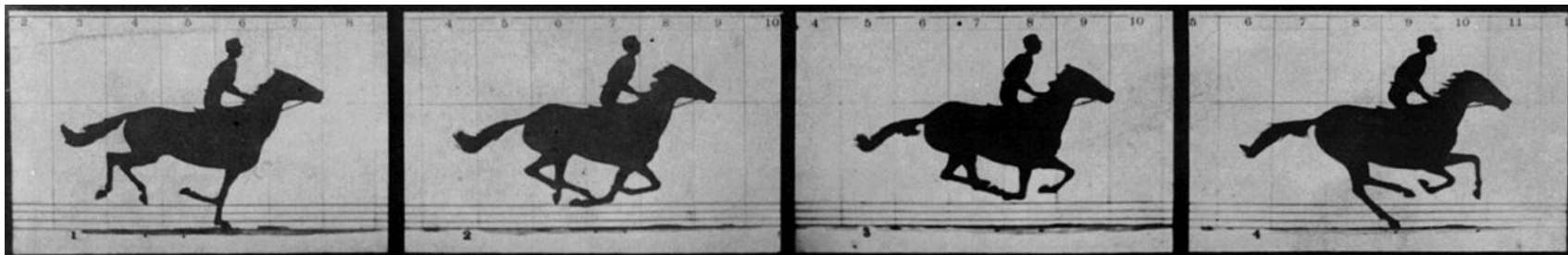
- Les photos sont souvent non répétitives
 - Mais a-t-on besoin de tous les détails ?
 - L'œil humain voit-il la différence entre 2 pixels similaires ?

**Compression avec perte par
approximation des blocs de pixels**

- Format d'image JPEG, ...
 - Résultat plus ou moins bon selon le taux de compression



Compresser les vidéos



Les images successives d'un film sont souvent très similaires

**Trouver les similitudes entre images
et stocker uniquement les différences**

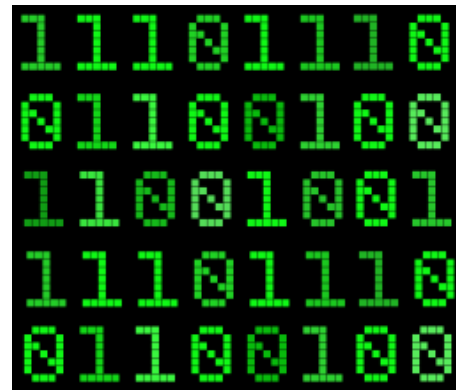
- Stocker une image complète de temps en temps
 - Au début de chaque plan
- Et ne stocker que les différences entre elles
- Format MPEG, ...

Compresser le son

- L'oreille humaine n'entend que les fréquences entre 20 Hz et 20 kHz
 - Inutile de coder les autres fréquences sonores
 - On peut aussi approximer les fréquences similaires
 - Format MP3
-
- Une fois codé, le signal peut être réécouté ou traité pour appliquer des effets (*reverb*, ...)

Ce qu'il faut retenir

- Différentes façons de (dé)coder ou (dé)compresser les données multimédia
 - Plus ou moins adaptées selon les cas
- L'ordinateur ne sait pas ce qu'il manipule
 - Des suites de bits sans aucun sens
- Les données sont accompagnées d'un en-tête expliquant comment elles sont codées (taille, durée, couleurs, compression, ...)

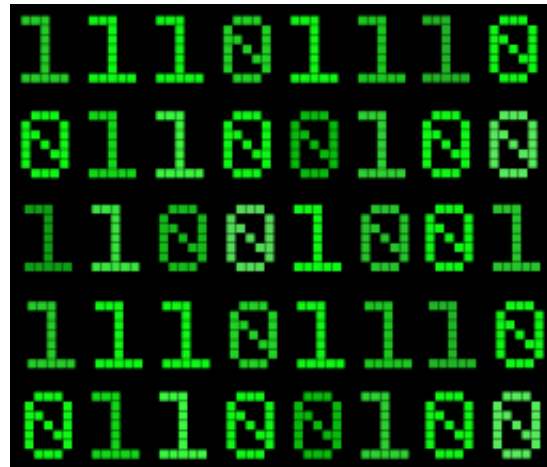


ICN : 1. Le codage binaire

1. Représentation de l'information sous forme de bits
2. Stocker les images ou les sons
3. Compression
4. **Organisation des données**
5. Bases de données

Comprendre le codage

- Les données sont stockées en mémoire, sur un disque, ou transférées sur le réseau
 - Peu importe le contenu
- L'ordinateur qui les lit ou les génère doit savoir comment elles sont codées
 - En-tête qui indique leur taille, durée, couleurs, type de compression, ...
- L'en-tête spécifie le codage des données
 - L'ordinateur doit d'abord comprendre l'en-tête



Types de données de base

- Entiers plus ou moins grands (8, 16, 32, 64bits)
 - Éventuellement négatifs
- Nombres *flottants* $1,45628963 \times 10^{23}$
 - Plus ou moins grands
- Le programmeur choisit selon les valeurs possibles de ce qu'il doit stocker
 - Âge, mois, température, niveau de gris, ...

Données plus complexes

- Comment stocker une date (22/11/2016) ou une couleur (RVB) ?
 - Combinaison de plusieurs nombres
- Des **Tuples** (*Structure, Enregistrement, ...*)
 - Une date est composée de 3 entiers (jour, mois, année)
 - ✓ Une époque est composée de 2 dates (début, fin)
 - Une image est composée de pixels, composés de 3 couleurs

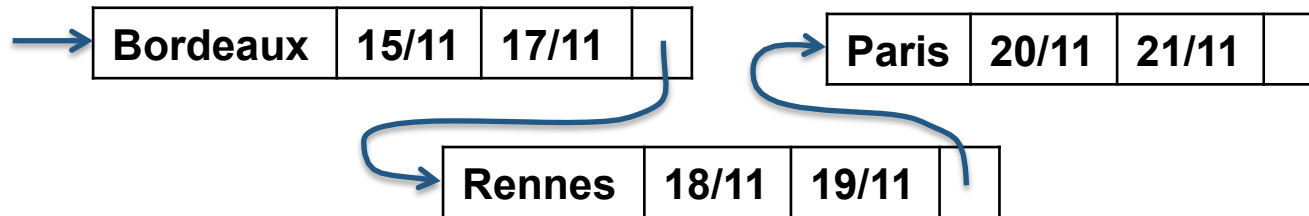
Types de données avancées

- Parfois, on ne sait pas à l'avance combien de données on va manipuler
 - Quels invités à une fête ?
 - Quelles étapes pendant un voyage ?
- **Tableau**
 - Éléments de structure similaire, numérotés

N°	Nom	Naissance	Taille (m)
1	Joe	21/02/1869	1,6
2	Jack	15/06/1868	1,7
3	William	02/03/1867	1,8
4	Averell	09/11/1866	1,9

Types de données avancées

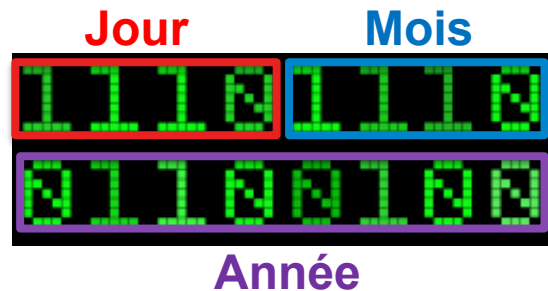
- Parfois, on ne sait pas à l'avance combien de données on va manipuler
 - Quels invités à une fête ?
 - Quelles étapes pendant un voyage ?
- **Liste**
 - Éléments de structure similaire, pointant vers leur successeur dans la liste



Représentation des données en mémoire

- Comment sont réparties les données concrètement ?
 - Quel ordre ? Quelle taille ?

**L'organisation précise
des différentes données
doit être clairement spécifiée**



- Décrite par le programmeur lors de l'écriture du programme
- Ou dans la spécification du format de fichier
 - Un fichier GIF commence par GIF89a puis la largeur et la hauteur de l'image sur 16bits chacun, etc.

Ce qu'il faut retenir

- Les données sont composées d'éléments de base
 - Entiers, nombres flottants, lettres, ...
- Qui peuvent être assemblés pour représenter des objets complexes
 - Tuples, tableaux, listes, ...
- Cet assemblage doit être bien spécifié pour pouvoir être utilisé

ICN : 1. Le codage binaire

1. Représentation de l'information sous forme de bits
2. Stocker les images ou les sons
3. Compression
4. Organisation des données
5. **Bases de données**

Gestion de données

- Gestion de stock, clients, ventes, bibliothèques, site web, ...
- Comment explorer et manipuler ces données de manière pratique pour un humain ?
 - Pas en lisant les structures de données en mémoire de l'ordinateur !

Système de gestion de base de données

- Médiateur entre l'humain et les données
- Exemples : MySQL, Oracle, ...

Le modèle relationnel – Les bases de données

- Des tableaux : Les *Relations*

Titre	Réalisateur	Acteur	<i>Films</i>
Imitation game	Morten Tyldum	Benedict Cumberbatch	
Snowden	Oliver Stone	Joseph Gordon-Levitt	

Titre	Cinéma	Heure	<i>Séances</i>
Imitation game	CGR le Français	19:45	
Snowden	Les Tourelles	20:00	
Imitation game	Les Tourelles	22:00	

Le modèle relationnel – Les requêtes

- Où puis-je voir un film avec B. Cumberbatch ?
- En SQL

```
select Cinéma
from Films, Séances
where Films.Titre = Séances.Titre
       and Acteur = "Benedict Cumberbatch"
```
- Langage standard compréhensible par un humain
- Traduit en langage machine
pour être exécuté rapidement par l'ordinateur

Protéger les données

- Les données peuvent avoir une valeur inestimable
 - Les comptes d'une entreprise
 - Vos photos de vacances
- Les systèmes informatiques ne sont pas infaillibles
 - Pannes matérielles, bugs logiciels, attaques, ...
- Garantir que les données ne seront pas perdues
 - Exemple : **Répliquer**

Partager les données

- Les données sont souvent manipulées par plusieurs utilisateurs
- Par exemple, Alice et Bob partagent leur musique
 - Alice lit un morceau de Leonard Cohen
 - Bob le lit à son tour
 - Alice ajoute un like à ce morceau
 - Bob fait de même
 - ✓ Le like de Alice a disparu ?!
- Des ***Transactions*** permettent de partager les données sans conflits lors des accès concurrents

Ce qu'il faut retenir

- Les structures de données manipulées par l'ordinateur ne sont pas adaptées aux humains
 - Assemblages complexes de bits, potentiellement compressés, etc.
- Les systèmes de gestion de base de données fournissent un moyen pratique et fiable de les manipuler