

Chapitre 2. Représentation, Codification et Contrôle de l'information

- **Objectifs du cours :**
 - Connaître les **caractéristiques** d'une **information**,
 - Comprendre l'**intérêt** et **types** de la **codification**,
 - Savoir comment **assurer** des **contrôles** sur les **informations**.

1. Caractéristiques de l'information

Une information possède un certain nombre de caractéristiques :

- a. **La forme** : elle peut être écrite, orale, visuelle, tactile, etc.
 - Lorsque les informations sont stockées dans leurs formes (état) **brutes sans format prédéfini**, on les appelle **informations non structurées** (destinées aux humains, difficile de les traiter par Ordinateur).
 - Les **informations structurées** sont très précises et stockées dans un *format prédéfini* (facile à collecter, à analyser et à traiter par Ordinateur).
- b. **Contenu sémantique (sens)** : il s'agit de l'importance d'une information (est-elle sélective ? apporte-t-elle de nouvelles connaissances ? est-elle précise ?).
- c. **Coût et Valeur** :
 - Le coût d'une information correspond à l'ensemble des sommes engagées par une entreprise pour sa recherche, son traitement, son stockage et sa destruction.
 - La valeur d'une information est l'apport et gain acquis, ou bien les erreurs et pertes évitées grâce à l'information.

Important : Le coût d'une information doit toujours être inférieur à sa valeur
- d. **Qualité d'une information** : pour connaître la qualité d'une information, 3 critères s'imposent : **Fiabilité** (est-elle à jour), **Disponibilité** (parvient-elle à temps), **Pertinence** (réelle, autorisée ou légale, ni redondante, ni calculable).

2. Codification de l'information

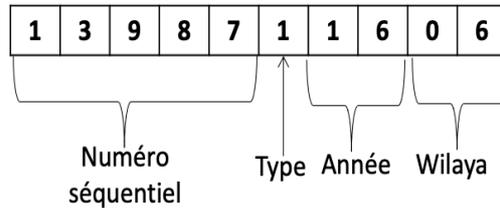
- **Objectifs (Pourquoi codifier)** :
 - Désigner une information d'une manière claire et unique
 - Pour que les informations soient traitées automatiquement par ordinateur
- **Mécanisme (Comment codifier)** :
 - Association des codes aux différentes informations manipulées par le SI
 - Les informations doivent être structurées (format prédéfini)

2.1. Définitions

- La codification est l'opération qui consiste à remplacer une information sous sa forme naturelle par un code clair et unique qui serait mieux adapté aux objectifs (besoins) de l'utilisateur et au traitement automatique (ordinateur).

- Un code est une représentation abrégée d'une information. Les codes peuvent être alphabétiques, numériques ou alphanumériques.

Exemple 1 : Matricule d'une voiture



Exemple 2 :

Soit le bon de commande suivant :

Bon de Commande

Numéro de commande :
Date de commande :

Numéro du Client:
Nom et Prénom du Client:
Adresse du Client :

Référence Produit	Désignation Produit	Quantité commandée

- Codification des noms d'attributs : Les désignations des informations (Numéro de commande, Numéro du Client...) sont trop longues et lourdes à manipuler => On doit les abréger (Codes uniques) sans perdre leurs significations. Ainsi, on aura :

Numéro de commande → *NumCmd*

Date de commande → *DateCmd*

Numéro du Client → *NumCL*

...

- Codification des valeurs : c'est selon le contexte, par exemple le *NumCL* peut prendre plusieurs valeurs selon le *NumCmd* comme suit :

La 1^{ère} commande est passée par le 3^{ème} Client se trouvant dans la région Centre

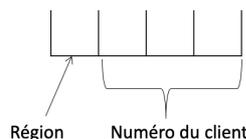
La 2^{ème} commande est passée par le 15^{ème} Client se trouvant dans la région Est

Les valeurs de *NumCL* sont : Valeur 1 : 3^{ème} Client se trouvant dans la région Centre

Valeur 2 : 15^{ème} Client se trouvant dans la région Est

On remarque que les valeurs d'informations sont longues et lourdes à manipulées => On doit structurer (créer un format prédéfini) pour stocker l'information.

Ainsi le format de *NumCL* doit être :



en supposant qu'une région ne possède pas plus de 999 clients

Avec ce nouveau format (Code), les valeurs de *NumCL* seront : Valeur 1 : C003

Valeur 2 : E015

On remarque que les valeurs et noms d'attributs sont beaucoup plus courts et faciles à stocker et à traiter par un ordinateur.

2.2. Caractéristiques d'une codification

- a. **Non ambiguïté (unicité)** : chaque code doit désigner un et un seul objet. Chaque objet doit être associé à un et un seul code. (Bijection entre l'ensemble des codes et l'ensemble des objets).

Exemple : Si on avait choisi ces codes comme nom d'attributs :

Numéro de commande \rightarrow <i>NumC</i>	}	<i>Ambiguïté</i> , <i>NumC</i> désigne Client ou Commande ?
Numéro du client \rightarrow <i>NumC</i>		

Solution : il faut choisir des codes différents.

- b. **Adapté aux besoins des utilisateurs** : simplicité de la fonction de codification et de décodification.

Exemple :

- Pour la décodification, *NumCL* (Numéro Client) = E087 désigne le 87^{ème} client de la région Est.
- Pour la codification d'un nouveau client : si on a un nouveau client à la région Ouest, comment le codifier ?
C'est simple : Région = O, Numéro = numéro du dernier client de la région Ouest + 1

c. Possibilité d'extension et d'insertion

- **Extension** : exprime le fait que l'ensemble des objets codifiés puisse s'accroître => Prévoir des codes qui tiennent compte de l'évolution des objets à codifier sans remettre en cause la codification choisie.

Exemple : *NumCL* : le numéro séquentiel est sur 3 positions => Maximum de clients dans une région est de 999.

- **Insertion** : exprime le fait qu'un nouveau code puisse s'insérer entre deux codes déjà existants sans remettre en cause la codification choisie => Prévoir des plages (intervalles) de valeurs, surtout lorsque les objets sont classés par catégorie.

Exemple : *NumCL* : numéro séquentiel :

De 001 à 700 : clients de catégorie *Particulier*

De 900 à 999 : clients de catégorie *Professionnel* (entreprise)

On a prévu une plage de 701 à 899 pour une éventuelle nouvelle catégorie.

- d. **Concision** : exprime le fait qu'un code doit être clair et court sans pour autant négliger la possibilité de l'évolution de l'ensemble des informations à coder.

Exemple : *NumCL* :

- Si la région d'un client n'est pas primordiale dans le code, il est inutile de la préciser

- => NumCL sera codifié avec un numéro séquentiel (3 positions) seulement.
- Si le nombre de clients ne dépasse pas 99 => 2 positions suffisent.

e. Significantive : la codification est une convention d'écriture entre la personne chargée du codage de l'information et celle chargée de son décodage. La lecture du code doit faire penser (rappeler) l'objet codifié => **Mnémonique** pour les noms d'attributs, **Table de correspondance** pour les valeurs.

Exemples :

- o Mnémonique :
 NumCL → Numéro du Client
 NumEtd → Numéro d'étudiant
- o Table de correspondance :
 Code Wilaya → entre 01 et 58. 06= Béjaia, 16= Alger...
 Etat Civil : C : célibataire, M : marié...

Conseil : Lors du choix d'un **code mnémonique**, évitez l'utilisation des :
 Accents (‘, é, à, è...), Tirets (-, _), Symboles (+, /, #, @...) et Espaces vides.

2.3. Types de codification

A. Codification élémentaire

A1. Codification séquentielle : consiste à affecter des numéros consécutifs aux objets à codifier.

Exemple : clé (0101, 0102...), les commandes NumCmd (001, 002, 003...)

Avantages	Inconvénients
+ Simplicité	- Non significatif
+ Non ambiguë	- Impossibilité d'insertion
+ Possibilité d'extension	- Pas de groupement possible

A2. Codification séquentielle par tranche : consiste à attribuer une tranche de codes à chaque catégorie d'objets à codifier (les valeurs sont séquentielles dans une tranche).

Exemple : Ouvrages dans une bibliothèque

001 à 100 : Technologie
 101 à 200 : Littérature
 201 à 300 : Médecine

...

Remarque : les tranches peuvent être décomposées ou non.

Exemple : Pour Médecine : 201 à 250 : Médecine générale
 251 à 270 : Pédiatrie
 271 à 299 : Ophtalmologie

Avantages	Inconvénients
+ Simplicité + Non ambiguë (si les tranches sont bien distinctes) + insertion et extension	- Non significatif (Nécessité d'une Table de correspondance) - Nombre des codes dans une tranche difficile à fixer - La répartition des codes en catégorie n'est pas toujours évidente.

A3. Codification Mnémonique : On reconnaît directement la propriété représentée à la lecture du code.

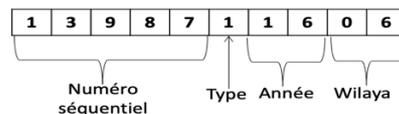
Exemples : Numéro d'étudiant → NumEtd,
 Numéro de poste de travail → NumPosTr

Avantages	Inconvénients
+ Significative + Très pratique + Facile à comprendre	- Porte sur les noms d'attributs et non les valeurs - Risque d'ambiguïté

B. Codification complexe

B1. Codification articulée : c'est la codification la plus utilisée en Informatique. Chaque code est décomposé en zones appelées *descripteurs*, chacune de ces zones a une signification.

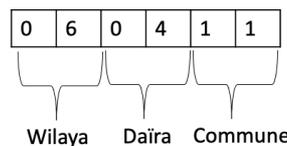
Exemple : Immatriculation d'une voiture



Avantages	Inconvénients
+ Non ambiguë + Facile à comprendre + Possibilité de regroupement selon un critère + Possibilité de contrôle sur certaines propriétés + Très utilisée	- Codage long et lourd à manipulé - Risque de saturation - Instabilité : le changement d'une caractéristique peut remettre en question toute la codification

B2. Codification hiérarchique à niveau : c'est un cas particulier de la codification articulée. Les descripteurs (zones) sont des niveaux hiérarchiques.

Exemple : codification d'une commune dans une Wilaya.



Avantages	Inconvénients
+ Mêmes avantages qu'Articulée + Facilité de recherche en arborescence	- Mêmes inconvénients qu'Articulée

2.4. Critères de choix d'une codification

Lors du choix d'une codification, Il faut tenir compte de :

- ◇ Utilisation future des codes à définir (population, propriétés stables, valeurs...),
- ◇ Nombre d'objets à codifier et sa croissance,
- ◇ Codification déjà existante,
- ◇ Avis des utilisateurs (personnes qui vont utiliser le code),
- ◇ Tester sérieusement la codification avant son adoption (utilisation).

3. Contrôles sur l'information

- **Objectif :** Le but d'un SI est de délivrer des informations correctes et **fiables** aux dirigeants d'une organisation pour assurer un bon fonctionnement et une aide à la décision. L'intrusion d'une **erreur** dans ces informations peut être **fatale** à l'organisation. Pour éviter les erreurs => Prévoir (ou subir) des **contrôles**.
- **Définition :** Contrôler une information c'est vérifier sa justesse et sa conformité à la réalité de l'organisation.
Exemple : Matricule d'une voiture

Contrôle d'un SI = Contrôles des Données + Contrôle des Traitements + Contrôles des Résultats

Remarque : Les contrôles peuvent être manuels (usuels, observations...) ou automatiques (programmés, machine...).

3.1. Principaux types de contrôles : on distingue 2 types :

- A. **Contrôles directs :** contrôles se rapportant à l'information elle-même indépendamment de toute autre information.

Exemple : Note d'un étudiant ≤ 20 .

A1. Contrôle de présence (ou de non présence) : consiste à vérifier que l'information figure bien (est présente) à l'emplacement où elle devrait se trouver sur le support qui peut être un document ou un fichier.

Il porte sur les informations dites **Obligatoires**.

Exemples : Fichier Étudiant : si nom n'existe pas => erreur

Prêt d'ouvrage : vérifier la présence de l'abonné

Inscription d'un nouvel abonné : vérifier la non présence (il ne doit pas exister déjà dans le fichier des abonnés)

A2. Contrôle de Type : consiste à vérifier que l'information est écrite dans la classe ou type approprié (numérique, alphabétique, alphanumérique...)

Exemples : N° Téléphone : 25L32E => erreur, il devrait être en numérique

N° Téléphone : 20202021 => pas d'erreur, le type est correct mais il faut ajouter un contrôle sur la longueur (10) et le format (0.....)

A3. Contrôle de cadrage : consiste à vérifier la position d'une information dans sa zone de saisie. Par convention : Numérique → Cadrage à droite

Alphabétique ou Alphanumérique → Cadrage à gauche

Exemple : N° et Nom d'un employé

N°

		4	1	1
--	--	---	---	---

 Nom

B	A	C	H	A		
---	---	---	---	---	--	--

Remarque : ce contrôle est généralement assuré d'une manière automatique par le logiciel.

A4. Contrôle de vraisemblance (plage de valeurs): consiste à vérifier que le valeur d'une information appartient au domaine (plage ou intervalle) des valeurs qui lui sont associés. i.e. s'assurer que la valeur d'une information est très semblable (possible, réalisable) en fonction de son sens.

Exemples : Date de naissance : 24/14/1995 => Erreur, car Mois ∈ {1, 2, ...,12}

Note d'un étudiant = -2 => Erreur, car Note ∈ [0, 20]

B. Contrôles indirects : contrôles déterminant la justesse d'une information par comparaison à d'autres informations.

Exemple : Quantité en Stock = Σ Quantités entrées – Σ Quantités sorties

B1. Contrôle de cohérence interne : c'est la vérification d'une justesse de la valeur d'une partie d'une donnée par rapport aux autres parties de la même donnée. Il est utilisé souvent dans le cas d'une codification articulée.

Exemples :

NumEtd

L	4	2	1	1
---	---	---	---	---

 => Erreur, car il n'y a pas de 4^{ème} année dans le cycle Licence.
↑ Cycle ↑ Année étude └───┘ Numéro séquentiel

Date : 30/02/2012 => Erreur, car le mois de Février ne peut pas avoir 30 jours.

B2. Contrôle de cohérence externe : consiste à vérifier la justesse de la valeur d'une donnée par rapport à d'autres données, soit dans le même fichier soit dans un autre fichier.

Exemple : Enregistrement d'un étudiant

N° L1265	Nom BenAli	Prénom Ahmed	DateNaiss 20/01/1997	Date Inscription 04/09/1995
-------------	---------------	-----------------	-------------------------	--------------------------------

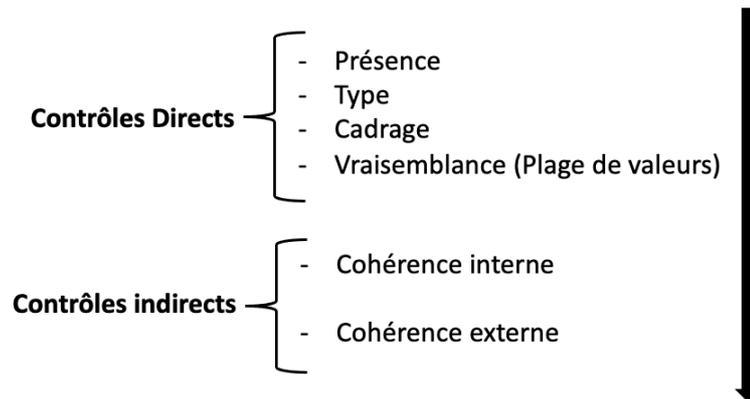
 => Erreur, car Date Naissance > Date Inscription

Remarque : Les contrôles de cohérence peuvent être représentées soit par des relations simples (<, >, =, ≤...) soit par des calculs.

Exemple : Contrôle de cohérence avec calcul : information d'un employé en 2021, année de recrutement : 2010, Ancienneté : 15 ans => Erreur, car l'ancienneté = 2021-2010 = 11 ans.

3.2. Ordre d'exécution des contrôles

Les contrôles directs doivent être effectués avant les contrôles indirects, ceci pour des raisons évidentes, car comment rapprocher une information d'une autre si elle est elle-même erronée.



3.3. Protection des informations

Lorsque l'information est trop importante (compte bancaire, N° sécurité sociale...) => On rajoute une clé de contrôle.

Exemple : Compte CCP N° : 25123438 Clé : 77

- ◇ S'il y a un risque de perte ou de détournement de l'information (Vol, escroquerie sur internet, attaque réseau...) => il faut assurer une **protection** des informations sensibles en les **Cryptant**.

Exemple :

- Tous les **mots de passe** (de comptes) dans une organisation **doivent être cryptés** pour qu'aucun employé ne puisse utiliser le compte d'un autre, ce qui permettra de bien gérer les droits d'accès (qui a le droit de voir quoi et de faire quoi) et ça permettra également de bien définir les responsabilités sur tout ce qui se passe dans le système informatique (Applications) dans l'organisation.

Mot de passe = ahmed20#bej ^{En cryptant} =====> 8747e564eb53cb2f1dcb9aae0779c2aa