

INFORMATIQUE 1: INTRODUCTION À L'INFORMATIQUE

LICENCE D'ÉDUCATION EN MATHÉMATIQUES (S1)
ESEF-J

Année universitaire : 2025-
2026

Pr : Fouad NAFIS

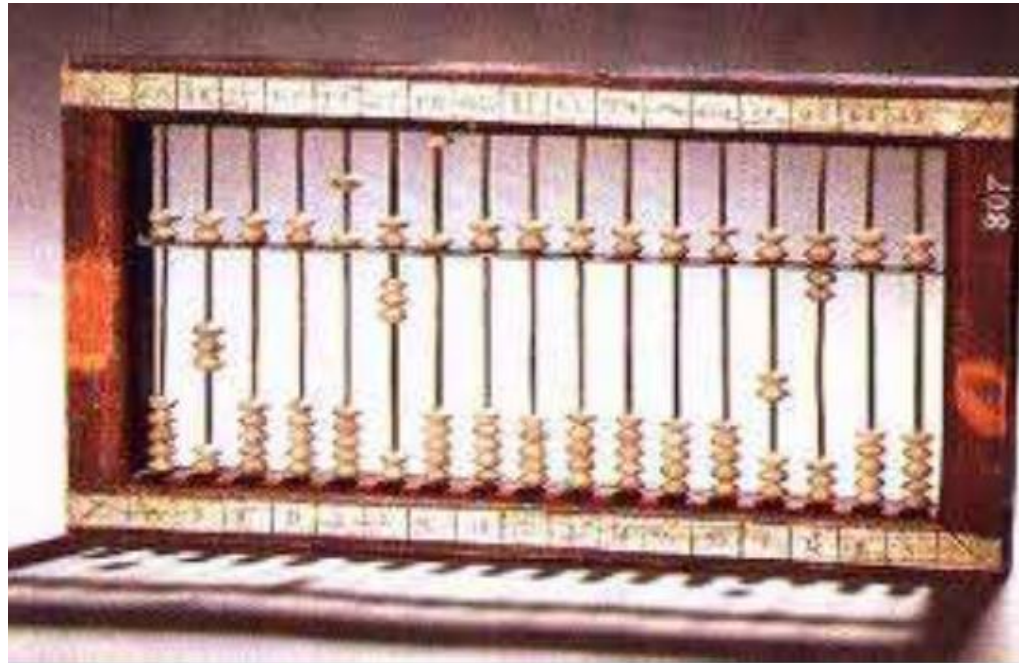
PLAN

- ❑ Historique
- ❑ Introduction
- ❑ **Partie 1: Systèmes de numération et Structure des ordinateurs**
- ❑ **Partie 2: Introduction à l'HTML**
- ❑ **Partie 3: Base de données relationnelles**

HISTORIQUE

- A l'origine de l'ordinateur, on retrouve une très longue série d'inventions dans les premiers siècles.
- Au fil des siècles, le but des hommes sera de **calculer de plus en plus vite** et en minimisant le plus possible la marge d'erreur. C'est de ce besoin de calculer que naissent les premières machines à calculer.

Le **boulier** ou **abaque** (7ème siècle avant J.C, Mésopotamie)



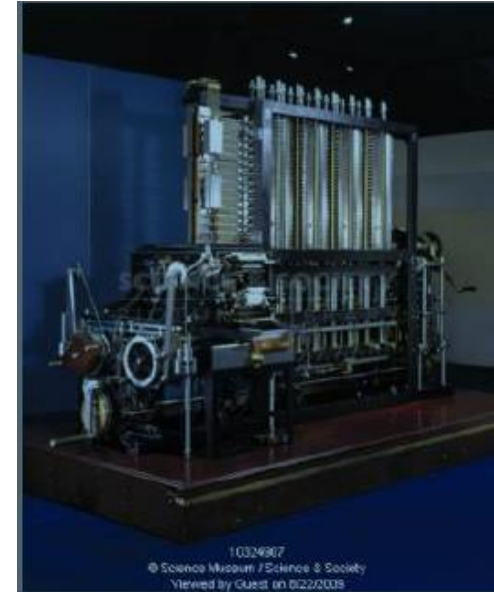
- ✓ En 1642, **Blaise Pascal** crée une machine capable d'additionner et de soustraire : la **Pascaline**.
- ✓ Elle a été créée spécifiquement pour le père de Blaise Pascal, qui était percepteur d'impôts.



- ✓ En 1673, **Leibniz** qui s'est inspiré de la **Pascaline** invente une machine capable de multiplier et de diviser.



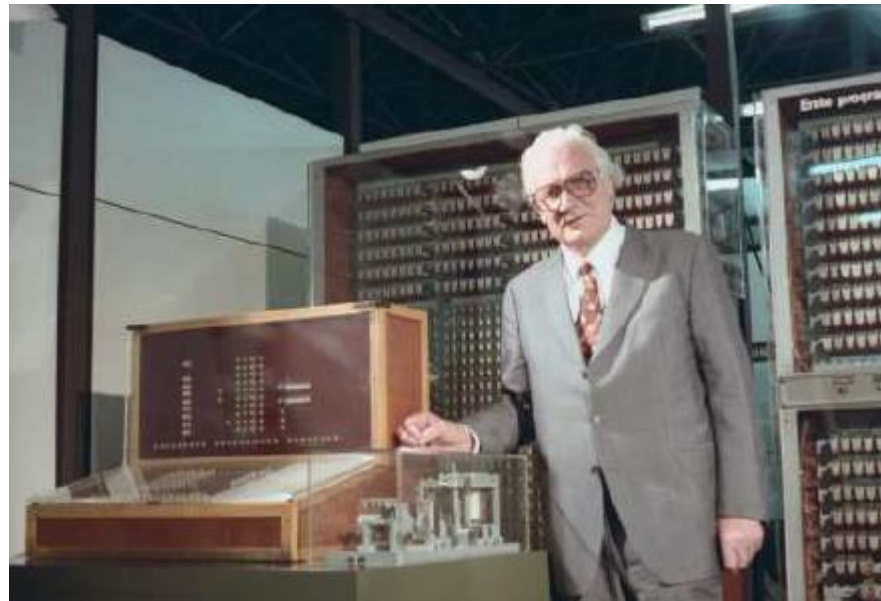
- ✓ En **1834**, **Charles Babbage** s'inspire du métier à tisser de Jacquard pour élaborer une machine qui, à l'aide de cartes perforées, évalue les différentes fonctions (addition, soustraction, multiplication, et division).



- ✓ A partir de **1885**, les **calculateurs** sont dotés de **clavier** et avec la découverte de l'électricité, des **moteurs électriques** qui remplacent les manivelles.



- ✓ Pendant la première partie du **20ème siècle**, l'avancée des mathématiques permet de donner un **nouveau souffle** à la recherche informatique.
- ✓ Celle-ci aboutit à partir de **1938** à la naissance du premier ordinateur programmable qui utilise le binaire : le **Z3**, créé par l'Allemand **Konrad Zuse**. Cet ordinateur réalise une multiplication en 4 secondes en moyenne.



Historique

- ✓ En **1943**, **Howard Aiken** met en place en collaboration avec **IBM** le premier calculateur électromécanique : **Mark I**. C'est une machine qui pèse 5 tonnes et mesure 17 m de longueur et 2,5 m en hauteur. Cette machine calcule **5 fois plus vite que l'homme**.

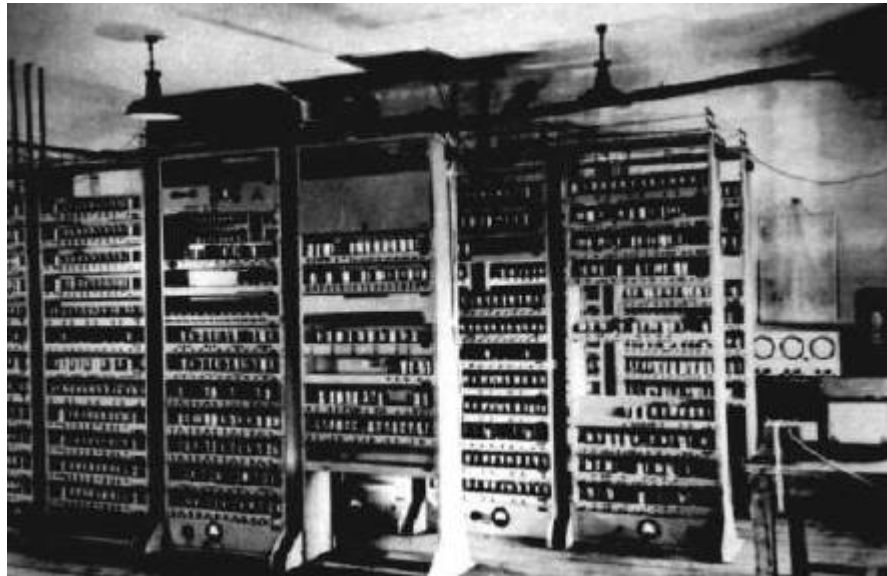


- ✓ En **1943**, l'**ENIAC** devient le premier ordinateur ne comportant **aucune pièce mécanique**.
- ✓ Il est composé de 18 000 lampes et s'étend sur plus 160 m².



Historique

- ✓ Alors que l'ENIAC n'était programmable que manuellement, l'**EDVAC** permet la **mémorisation**.
- ✓ C'est une réelle innovation en **1946**, puisqu'il permet de mémoriser 1024 mots en mémoire centrale et 20 000 en mémoire magnétique.
- ✓ L'apparition du **transistor** en **1948** révolutionne l'informatique, permettant ainsi de fabriquer des ordinateurs moins encombrants et qui consomment moins d'électricité.



Historique

- ✓ **Années 1950** : apparition de nouveaux **langages de programmation** et des premiers **circuits intégrés**. La taille et le coût de l'ordinateur sont réduits.
- ✓ **Années 1970** : création des premiers **systèmes d'exploitation**, dont l'inévitable **UNIX**.
- ✓ **1971** : création du 1er **microprocesseur** par Intel.



Steve Jobs et Steve Wozniak présentent le 1^{er} micro-ordinateur personnel d'Apple Computer.

Historique

- ✓ **IBM** crée en **1981** son **premier PC**.
- ✓ L'ordinateur rentre alors dans les foyers et on parle pour les années **1980** d'**informatique familiale**.
- ✓ Parallèlement, l'apparition des jeux sur ordinateur rend l'informatique **conviviale**.



Historique

- ✓ Les avancées technologiques s'accélérent considérablement dans les années 1990.
- ✓ **Aujourd'hui**, la micro-informatique se développe à un rythme très rapide. Avec la miniaturisation des composants matériels, on parle de **nanotechnologie**.





- Science du traitement automatique de l'information grâce à une machine appelée ordinateur. Ce mot est utilisé pour la première fois en 1962 par Philippe Dreyfus.
- Le terme « informatique » vient de la contraction des mots « information » et « automatique ».

Informatique = **information** + **automatique**

INTRODUCTION

Introduction du module

Ordinateur ?????

- Ensemble de circuits électroniques permettant de manipuler des données sous forme binaire , c'est-à-dire sous forme de bits (0 ou 1).
- Les trois éléments essentiels d'un ordinateur sont, **le processeur, les mémoires et les dispositifs de gestion des entrées-sorties.**



Introduction du module : Utilisation des ordinateurs

Systeme d'exploitation

Programme système qui gère les différentes ressources de la machine

Programme

Suite d'instructions dans un langage donnée, définissant un traitement exécutable par un ordinateur

- programmes systèmes
- programmes d'application

Programmation

A partir d'un problème donné, c'est la réalisation d'un programme (langage de programmation machine) dont l'exécution apporte une solution satisfaisante au problème posé

SYSTÈMES DE NUMÉRATION

Objectifs:

- Comprendre le principe des codages binaires.
- Représenter des nombres entiers dans diverses représentations classiques (M&S, Cà1, Cà2).
- Représenter des nombres réels en IEEE 754.
- Comprendre le lien entre représentation des nombres et calcul arithmétique.
- Représenter des données alphanumériques ASCII.

Partie 1.1 : Systèmes de numération

Plan

1. Le système binaire
2. Le système décimal
3. Le système octal
4. Le système hexadécimal
5. Passage de la base 2, 8, 16 à la base 10
6. Passage de la base 10 à la base 2, 8, 16
7. Passage de l'octal au binaire
8. Passage de l'hexadécimal au binaire
9. L'arithmétique binaire
10. Les nombres négatifs et les nombres réels
11. Différentes représentations relatives aux données

Systèmes de numération

- Données
 - Nombres (entiers, réels)
 - Images
 - Vidéos
 - Sons
 - etc.
- Toujours représentées sous forme binaire (0 ou 1) à l'aide de bits.



Systèmes de numération

Définition

- Un **système de numération** est un ensemble de symboles et de règles permettant de représenter les nombres
- Un système de numération se définit par deux éléments:
 - La base du système,
 - Les symboles du système.

En informatique, les systèmes les plus utilisés sont les suivants: **décimal, binaire, octal et hexadécimal.**

Système	Base	Symboles	Nbre de symbole
Décimal	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9.	10
Binaire	2	0, 1.	2
Octal	8	0, 1, 2, 3, 4, 5, 6, 7.	8
Hexadécimal	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.	16

Systèmes de numération

Notation

Soit N un nombre quelconque exprimé dans une base b . N sera noté comme suit:

$$N = (a_{n-1} a_{n-2} a_{n-3} \dots a_0)_b$$

Tel que:

b : base du système de numération.

a_i : symbole du système, $i = 0, \dots, n-1$. avec $a_i < b$

Systèmes de numération

Exemple:

- $N_1 = (19017)_{10}$

En Décimal, avec: $a_4 = 1$, $a_3 = 9$, $a_2 = 0$, $a_1 = 1$, $a_0 = 7$.

On remarque que les a_i sont tous inférieurs à la base 10 ($a_i < 10$).

- $N_2 = (1011101)_2$

En Binaire, avec: $a_6 = 1$, $a_5 = 0$, $a_4 = 1$, $a_3 = 1$, $a_2 = 1$, $a_1 = 0$, $a_0 = 1$ ($a_i < 2$).

- $N_3 = (1370)_8$

En Octal, avec: $a_3 = 1$, $a_2 = 3$, $a_1 = 7$, $a_0 = 0$ ($a_i < 8$).

- $N_4 = (A9120)_{16}$

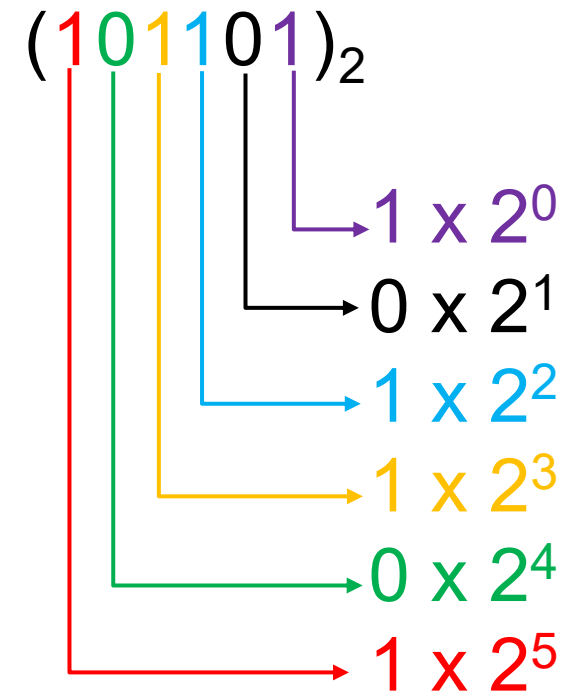
En Hexadécimal, avec: $a_4 = A$, $a_3 = 9$, $a_2 = 1$, $a_1 = 2$, $a_0 = 0$ ($a_i < 16$).

- $N_5 = (18095)_8$

La notation $N_5 = (18095)_8$ n'est pas correcte, car tous les chiffres doivent être inférieurs à 8 ($a_i < 8$) ce qui n'est pas le cas pour le 2^{ième} chiffre et aussi pour le 4^{ième} ($a_1 = 9 > 8$) et ($a_3 = 8$).

1. Le système binaire

- C'est le système de numération utilisant la **base 2**.
- On nomme **bit** les chiffres de la numération binaire positionnelle.
- Un **bit** peut prendre deux valeurs, notées par convention **0** et **1**.
- C'est le système utilisée en informatique pour la représentation des informations au niveau machine.
- Les deux états **0** et **1** sont les seuls que la machine peut assimiler.
- La position des 0 et des 1 indique respectivement l'absence ou la présence d'une puissance de 2.



2. Le système décimal

- C'est le système usuel dans la vie quotidienne. La base du système décimal est la base **10** et ses symboles sont les dix chiffres: **0, 1, 2, 3, 4, 5, 6, 7, 8, 9**.
- Les dix chiffres précédents sont suffisants pour exprimer n'importe quel nombre .
- La machine ne pouvant assimiler que les deux valeurs 0 et 1, il serait important de savoir comment exprimer les nombres décimaux en binaires et comment effectuer l'opération inverse et on parle de conversion de base.

The diagram illustrates the expansion of the decimal number $(319038)_{10}$ into its base-10 components. Each digit is connected by a colored line to its corresponding term in the expansion:

- Red line from '3' to 3×10^5
- Green line from '1' to 1×10^4
- Yellow line from '9' to 9×10^3
- Blue line from '0' to 0×10^2
- Black line from '3' to 3×10^1
- Purple line from '8' to 8×10^0

The expansion is shown as:

$$(319038)_{10} = 3 \times 10^5 + 1 \times 10^4 + 9 \times 10^3 + 0 \times 10^2 + 3 \times 10^1 + 8 \times 10^0$$

3. Le système octal

- La base du système octal est 8.
- En octal, les nombres sont représentés sous forme de combinaisons de chiffres parmi les suivants: 0, 1, 2, 3, 4, 5, 6, et 7.

The diagram illustrates the positional expansion of the octal number $(540371)_8$. Each digit is connected by a colored line to its corresponding power of 8:

- Red line from '5' to 5×8^5
- Green line from '4' to 4×8^4
- Yellow line from '0' to 0×8^3
- Blue line from '3' to 3×8^2
- Black line from '7' to 7×8^1
- Purple line from '1' to 1×8^0

The expansion is shown as:

$$(540371)_8 = 5 \times 8^5 + 4 \times 8^4 + 0 \times 8^3 + 3 \times 8^2 + 7 \times 8^1 + 1 \times 8^0$$

4. Le système hexadécimal

- Le système **hexadécimal** (**base 16**) utilise 16 chiffres pour la représentation des nombres, à savoir:
 - les chiffres du système décimal: **0, 1, 2, 3, 4, 5, 6, 7, 8, 9.**
 - les six (6) premières lettres de l'alphabet: **A, B, C, D, E, F.**

Hexadécimal	Décimal
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	10
B	11
C	12
D	13
E	14
F	15

$(140FC1)_H$

1×16^0

12×16^1

15×16^2

0×16^3

4×16^4

1×16^5

5. Passage de la base 2, 8, 16 à la base 10

- L'exemple suivant illustre la méthode de **conversion**, en **décimal**, d'un nombre exprimé dans une **base b quelconque**.

Soit $N = (a_{n-1} a_{n-2} a_{n-3} \dots a_0)_b$

- Pour avoir la représentation en décimal du nombre **N** exprimé dans une base **b quelconque**, il suffit d'effectuer le calcul suivant:

$$(N)_b = a_{n-1} * b^{n-1} + a_{n-2} * b^{n-2} + \dots + a_1 * b^1 + a_0 * b^0$$

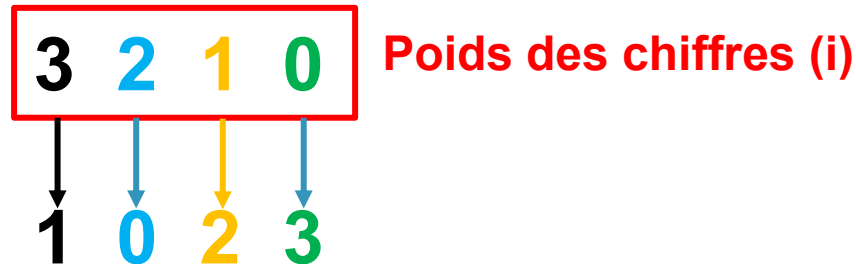
- La formule générale s'écrit comme suit:
$$(N)_{10} = \sum_{i=0}^{n-1} a_i * b^i$$

Avec : i étant le poids du chiffre a_i .

5. Passage de la base 2, 8, 16 à la base 10

- **Exemple 1** : On considère le nombre: $N = (1023)_4$

- ✓ On commence par définir le poids de chaque chiffre et cela en les numérotant de droite à gauche et on commençant la numérotation à partir de 0.



- ✓ Puis, on multiplie chaque chiffre a_i de poids i par la base b élevée à la puissance i . ($a_i * b^i$)

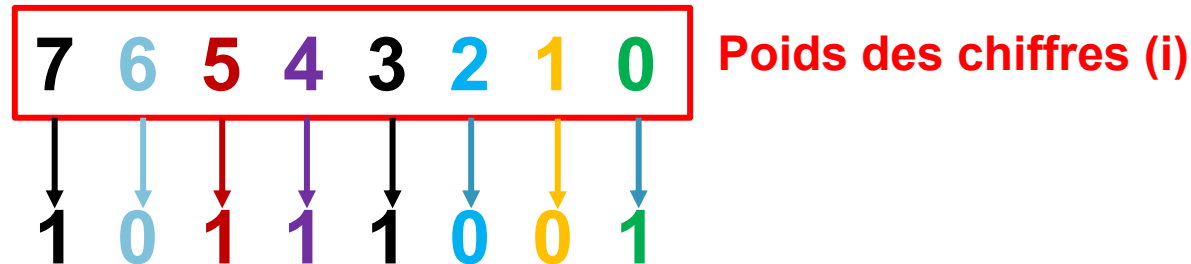
$$\begin{aligned} N &= 1 * 4^3 + 0 * 4^2 + 2 * 4^1 + 3 * 4^0 \\ &= 64 + 0 + 8 + 3 \\ &= 75. \end{aligned}$$

Ainsi, nous avons: $N = (1023)_4 = (75)_{10}$

5. Passage de la base 2, 8, 16 à la base 10

- **Exemple 2 :** Soit à convertir en décimal, le nombre N exprimé en binaire comme suit: $N = (10111001)_2$

✓ Les poids des chiffres:



✓ Ainsi, le nombre N en décimal est calculé comme suit:

$$\begin{aligned} N &= 1 * 2^7 + 0 * 2^6 + 1 * 2^5 + 1 * 2^4 + 1 * 2^3 + 0 * 2^2 + 0 * 2^1 + 1 * 2^0 \\ &= 128 + 0 + 32 + 16 + 8 + 0 + 0 + 1 \\ &= 185. \end{aligned}$$

$$\text{D'où: } N = (10110101)_2 = (185)_{10}$$

5. Passage de la base 2, 8, 16 à la base 10

- **Exemple 3** : Soit à convertir en décimal le nombre $X = (175)_8$.

$$\begin{aligned}(X)_{10} &= 1 \cdot 8^2 + 7 \cdot 8^1 + 5 \cdot 8^0 \\ &= 64 + 56 + 5 \\ &= 125.\end{aligned}$$

$$\text{Ainsi, } (175)_8 = (125)_{10}$$

Soit à convertir en décimal le Nombre $X = (A24)_H$.

$$\begin{aligned}(X)_{10} &= A \cdot 16^2 + 2 \cdot 16^1 + 4 \cdot 16^0 \\ &= 10 \cdot 16^2 + 2 \cdot 16^1 + 4 \cdot 16^0 \\ &= 2596.\end{aligned}$$

$$\text{Ainsi, } (A24)_H = (2596)_{10}$$

6. Passage de la base 10 à la base 2, 8, 16

Pour exprimer en binaire, un nombre exprimé dans une base b , on dispose d'une méthode par divisions successives.

Conversion par division successives

Soit X un nombre exprimé dans la base 10. Pour l'exprimer dans une autre base b , il suffit d'effectuer des divisions successives sur b jusqu'à l'obtention d'un résultat nul.

Les étapes à suivre sont les suivantes:

soit X_i : le résultat de la division.

r_i : le reste de la division n° i .

1) Effectuer la division $X / b = X_0$ et le reste r_0

si $X_0 = 0$ alors aller à 3)

sinon aller à 2)

2) Effectuer la division $X_i / b = X_{i+1}$ et le reste r_{i+1}

si $X_{i+1} = 0$ alors aller à 3)

sinon aller à 2)

3) Arrêter la division. Le résultat est $(X)_{10} = (r_m r_{m-1} \dots r_1 r_0)_b$

6. Passage de la base 10 à la base 2, 8, 16

Exemples:

$Y = (1110011)_2$

Diagram illustrating the conversion of the decimal number 115 to binary. The process uses repeated division by 2, with remainders recorded from bottom to top. The remainders are 1, 1, 0, 0, 1, 1, 1, which read from bottom to top give the binary result 1110011. A red arrow indicates the sequence of divisions. A blue arrow points to the final remainder 1, labeled r_6 . A text label states: "résultat = 0, donc arrêt."

1) Soit le nombre $Y = (125)_{10}$, convertir ce nombre en Octal:

$Y = (175)_8$

Diagram illustrating the conversion of the decimal number 125 to octal. The process uses repeated division by 8, with remainders recorded from bottom to top. The remainders are 5, 7, 1, which read from bottom to top give the octal result 175. A red arrow indicates the sequence of divisions.

$Y = (A24)_H$

Diagram illustrating the conversion of the decimal number 2596 to hexadecimal. The process uses repeated division by 16, with remainders recorded from bottom to top. The remainders are 4, 2, 10 (A), which read from bottom to top give the hexadecimal result A24. A red arrow indicates the sequence of divisions.

7. Passage de l'octal au binaire

La base 8 est une puissance de la base 2. Pour convertir un nombre octal en binaire, on procède comme suit:

- On a $8 = 2^3$ cela veut dire que pour représenter un seul chiffre octal en binaire, il faut utiliser 3 bits.
- Ainsi, la représentation des chiffres de la base 8 en binaire est la suivante:

Chiffre octal	Chiffre binaire équivalent
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

7. Passage de l'octal au binaire

Exemple: Soit le nombre $Y = (175)_8 = (???)_2$.

Pour trouver l'équivalent binaire de ce nombre octal, il suffit de trouver l'équivalent binaire de chaque chiffre octal.

Octal	1			7			5		
Binaire	0	0	1	1	1	1	1	0	1

$$Y = (175)_8 = (1111101)_2$$

8. Passage de l'hexadécimal au binaire

La base 16 est une puissance de la base 2.

Pour convertir un nombre hexadécimal en binaire, on procède comme suit:

On a $16 = 2^4$ cela veut dire que pour représenter un seul chiffre hexadécimal en binaire, il faut utiliser 4 bits.

Ainsi, la représentation des chiffres de la base 16 en binaire est la suivante:

Chiffre hexadécimal	Chiffre binaire équivalent
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

8. Passage de l'hexadécimal au binaire

Exemple: Soit le nombre $Y = (A24)_H = (???)_2$.

Pour trouver l'équivalent binaire de ce nombre hexadécimal, il suffit de trouver l'équivalent binaire de chaque chiffre hexadécimal.

Hexa	A				2				4			
Binaire	1	0	1	0	0	0	1	0	0	1	0	0

$$Y = (A24)_H = (101000100100)_2$$

ARITHMÉTIQUE BINAIRE

9. L'arithmétique binaire

Addition

- 60+17 en base 2

$$\begin{array}{r} 11\ 1\ 1\ 1\ 0\ 0 \\ +\ 0\ 1\ 0\ 0\ 0\ 1 \\ \hline 1\ 0\ 0\ 1\ 1\ 0\ 1 \end{array}$$

→ 77

- $2^5 + 2^4 + 2^3 + 2^2 = 32 + 16 + 8 + 4 = (60)_{10}$
- $2^4 + 2^0 = 16 + 1 = (17)_{10}$
- $2^6 + 2^3 + 2^2 + 2^0 = 64 + 8 + 4 + 1 = (77)_{10}$

9. L'arithmétique binaire

Soustraction

- 8-1 en base 2

$$\begin{array}{r} 1 \ 10 \ 10 \ 10 \\ - 0^1 \ 0^1 \ 0^1 \ 1 \\ \hline 0 \ 1 \ 1 \ 1 \end{array}$$

→ 7

- $2^3 = (8)_{10}$
- $2^0 = (1)_{10}$
- $2^2 + 2^1 + 2^0 = 4 + 2 + 1 = (7)_{10}$

9. L'arithmétique binaire

Multiplication

7*5 en base 2

$$\begin{array}{r} 111 \\ * 101 \\ \hline 111 \\ 000 \\ 111 \\ \hline 100011 \end{array}$$

→ 35

- $2^2 + 2^1 + 2^0 = 4 + 2 + 1 = (7)_{10}$
- $2^2 + 2^0 = 4 + 1 = (5)_{10}$
- $2^5 + 2^1 + 2^0 = 32 + 2 + 1 = (35)_{10}$

9. L'arithmétique binaire

Division

- 37/5 en base 2

$$\begin{array}{r} 100101 \\ / \quad 101 \\ \hline 111 \end{array}$$

→ 7

- $2^5 + 2^2 + 2^0 = 32 + 4 + 1 = (37)_{10}$
- $2^2 + 2^0 = 4 + 1 = (5)_{10}$
- $2^2 + 2^1 + 2^0 = 4 + 2 + 1 = (7)_{10}$

$$\begin{array}{r|l} 100101 & 101 \\ 1001 & 111 \\ - 101 & \\ \hline 1000 & \\ - 101 & \\ \hline 0111 & \\ - 101 & \\ \hline 010 & \end{array}$$

REPRÉSENTATION DES ENTIERS RELATIFS

10. Les nombres négatifs et les nombres réels

Décimal négatif → binaire

3 façons de représenter les nombres négatifs avec n bits:

- **Module et signe (noté M&S)**

- ✓ on utilise le bit le plus à gauche pour représenter le signe
- ✓ exemple mots de 4 bits $((3)_{10}=(11)_2)$ donc $(0011)_2=(3)_{10}$ et $(1011)_2=(-3)_{10}$

- **Complément à 1 (noté C_{à1})**

- ✓ pour un nombre négatif, on prend la représentation de la partie entière et on inverse tous les bits
- ✓ exemple mots de 4 bits: $((4)_{10}=(100)_2)$ donc $(0100)_2=(4)_{10}$ et $(1011)_2=(-4)_{10}$

- **Complément à 2 (noté C_{à2})**

- ✓ idem, mais avant d'inverser, on soustrait 1 (ou on inverse puis on ajoute 1)
- ✓ exemple mots de 4 bits : $((6)_{10}=(110)_2)$ donc $(0110)_2=(6)_{10}$ et $(1010)_2=(-6)_{10}$

10. Les nombres négatifs et les nombres réels

Addition (ADD) et soustraction (SUB)

- Le bit signe est traité comme tous les autres bits
- La soustraction est un cas particulier de l'addition
- Les nombres négatifs sont traités comme des nombres à additionner

Exemple mots de 3 bits:

- Addition par C à 1, (si retenue? on ajoute 1 au résultat)
 - ✓ $010_2 + 110_2 = 1000_2 \rightarrow$ une retenue +1 $\rightarrow 001_2$ équivalent en décimal $2 + (-1) = 1$
 - ✓ $001_2 + 101_2 = 110_2 \rightarrow$ pas de retenue $\rightarrow 110_2$ équivalent en décimal $1 + (-2) = -1$
- Addition par C à 2 (Directe)
 - ✓ $010_2 + 111_2 = 1\ 001_2 \rightarrow 001_2$ équivalent en décimal $2 + (-1) = 1$
 - ✓ $001_2 + 110_2 = 111_2 \rightarrow 111_2$ équivalent en décimal $1 + (-2) = -1$

REPRÉSENTATION DES NOMBRES RÉELS RELATIFS

10. Les nombres négatifs et les nombres réels

Conversion nombre réel binaire → décimal

- Virgule fixe: $d_{n-1} d_{n-2} d_{n-3} \dots d_2 d_1 d_0$, $d_{-1} d_{-2} d_{-m}$ avec n et m fixe
- Un nombre qui s'exprime en base B par 3 bits avec n=2 et m=1 le chiffres 101 s'analyse: $1*B^1 + 0*B^0 + 1*B^{-1}$
 - ✓ En binaire $(10,1)_2 = 1*2^1 + 0*2^0 + 1*2^{-1} = (2,5)_{10}$
 - ✓ En octale $(10,1)_8 = 1*8^1 + 0*8^0 + 1*8^{-1} = (8,125)_{10}$
 - ✓ En hexadécimal $(10,1)_{16} = 1*16^1 + 0*16^0 + 1*16^{-1} = (16,0625)_{10}$

10. Les nombres négatifs et les nombres réels

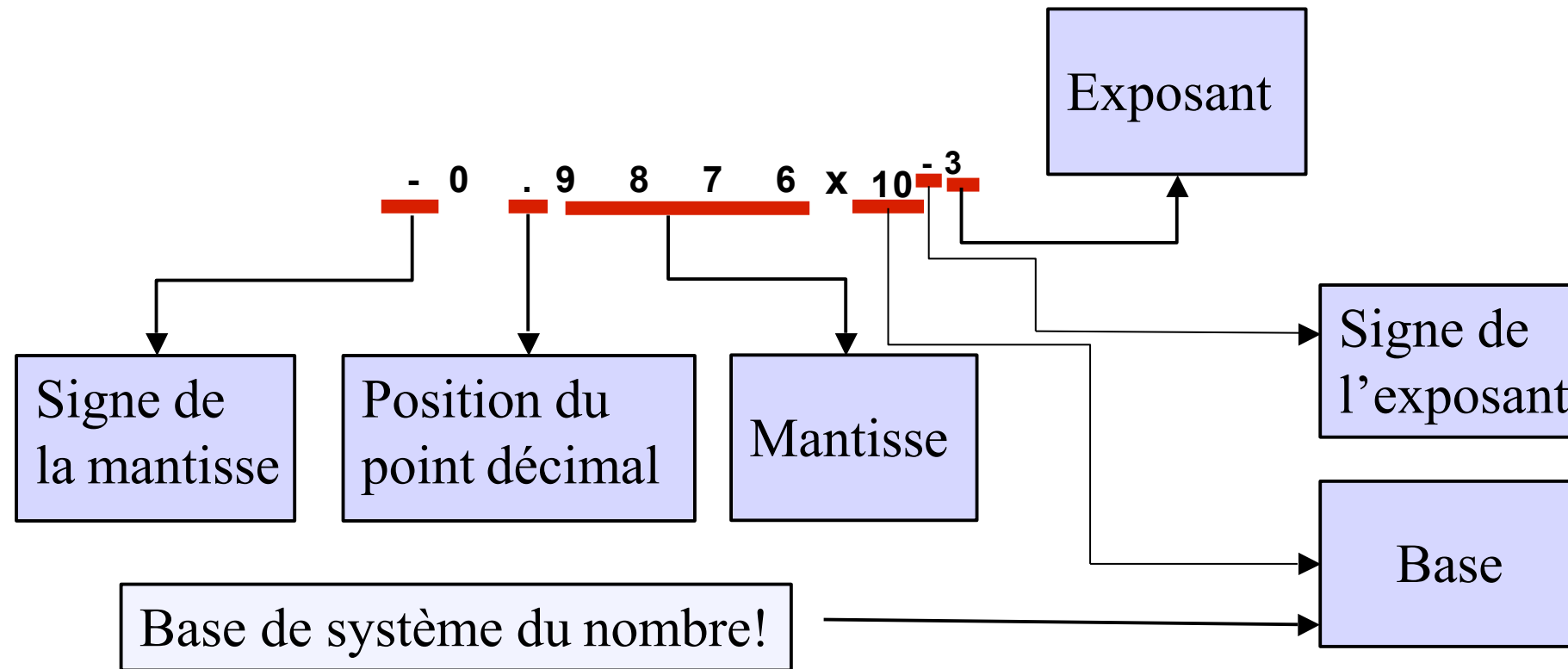
Conversion nombre réel décimal → Binaire

Exemple : $(77,75)_{10} = (??)_2$ avec $n=8$ et $m=2$

- Pour la partie entière → il faut faire une division successive des quotients sur 2 puis écrire les restes en commençant par le dernier: $77_{10} = 01001101$
- Pour la partie réelle → il faut faire une multiplication successive des parties réel par 2 puis écrire la suite de la partie entière en commençant par le premier.
- $0,75 \times 2 = 1,5$
- $0,5 \times 2 = 1$
- En binaire $0,11$ ($1 \times 2^{-1} + 1 \times 2^{-2} = 0,5 + 0,25 = 0,75$) Résultat $77,75_{10} = 01001101,11_2$

10. Les nombres négatifs et les nombres réels

Réel négatif (virgule flottante): **Notation exponentielle**



10. Les nombres négatifs et les nombres réels

Réel négatif (virgule flottante): Représentation normalisée

- Un nombre représenté en virgule flottante est normalisé s'il est sous la forme:
 - $\pm 0, M * X^{\pm c}$
 - M un nombre dont le premier chiffre est non nul
- Exemple:
 - $+ 59,4151 * 10^{-5} \Rightarrow$

Normalisé: $+0,594151 * 10^{-3}$

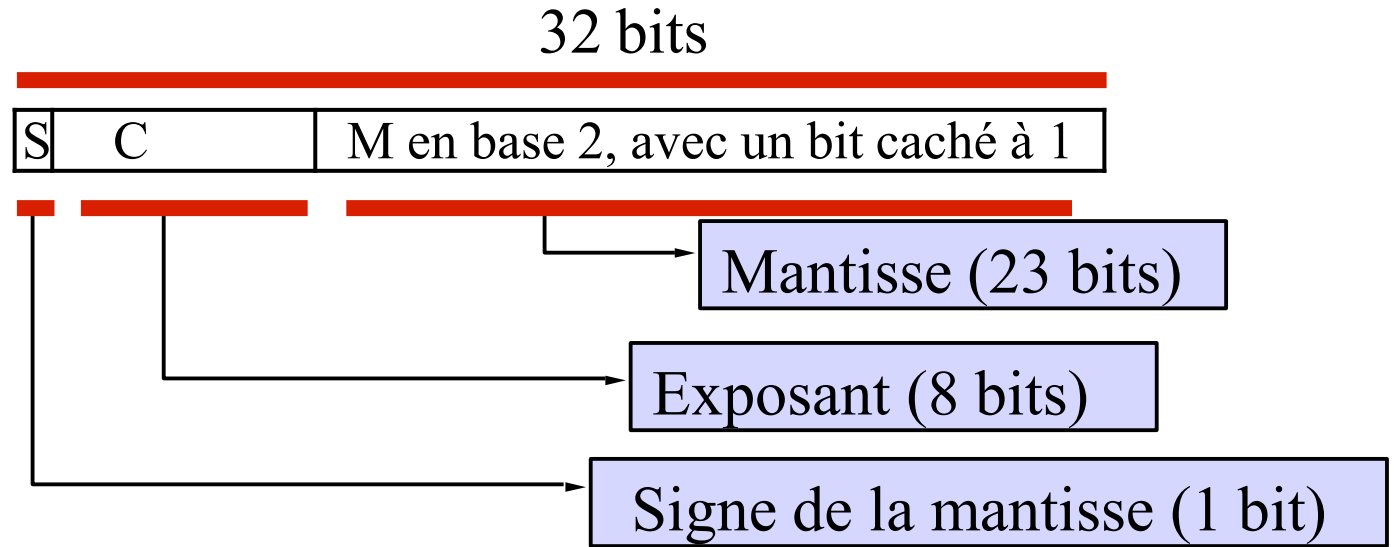
10. Les nombres négatifs et les nombres réels

Réel négatif (virgule flottante): **Norme IEEE 754**

Un format standardisé

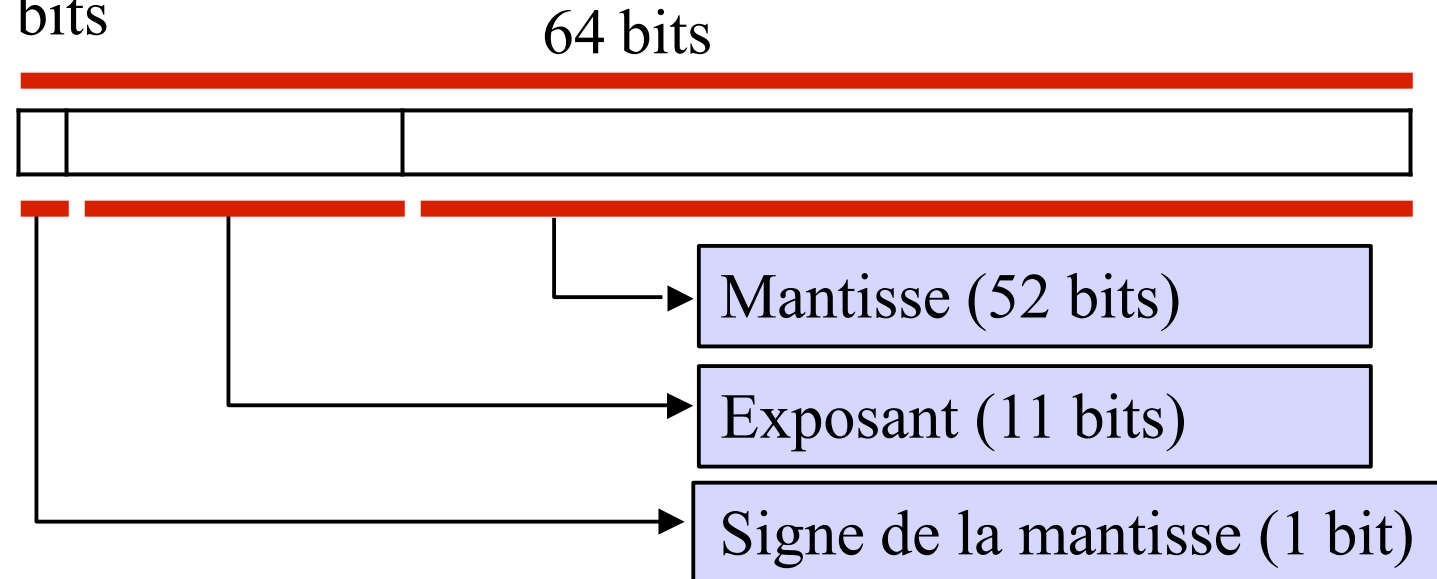
Format simple précision: 32 bits

- Bit du signe (1 bit)
- Exposant (8 bits)
- Mantisse (23 bits)



Format double précision: 64 bits

- Bit du signe (1 bit)
- Exposant (11 bits)
- Mantisse (52 bits)



10. Les nombres négatifs et les nombres réels

Réel négatif (virgule flottante): **Norme IEEE 754**

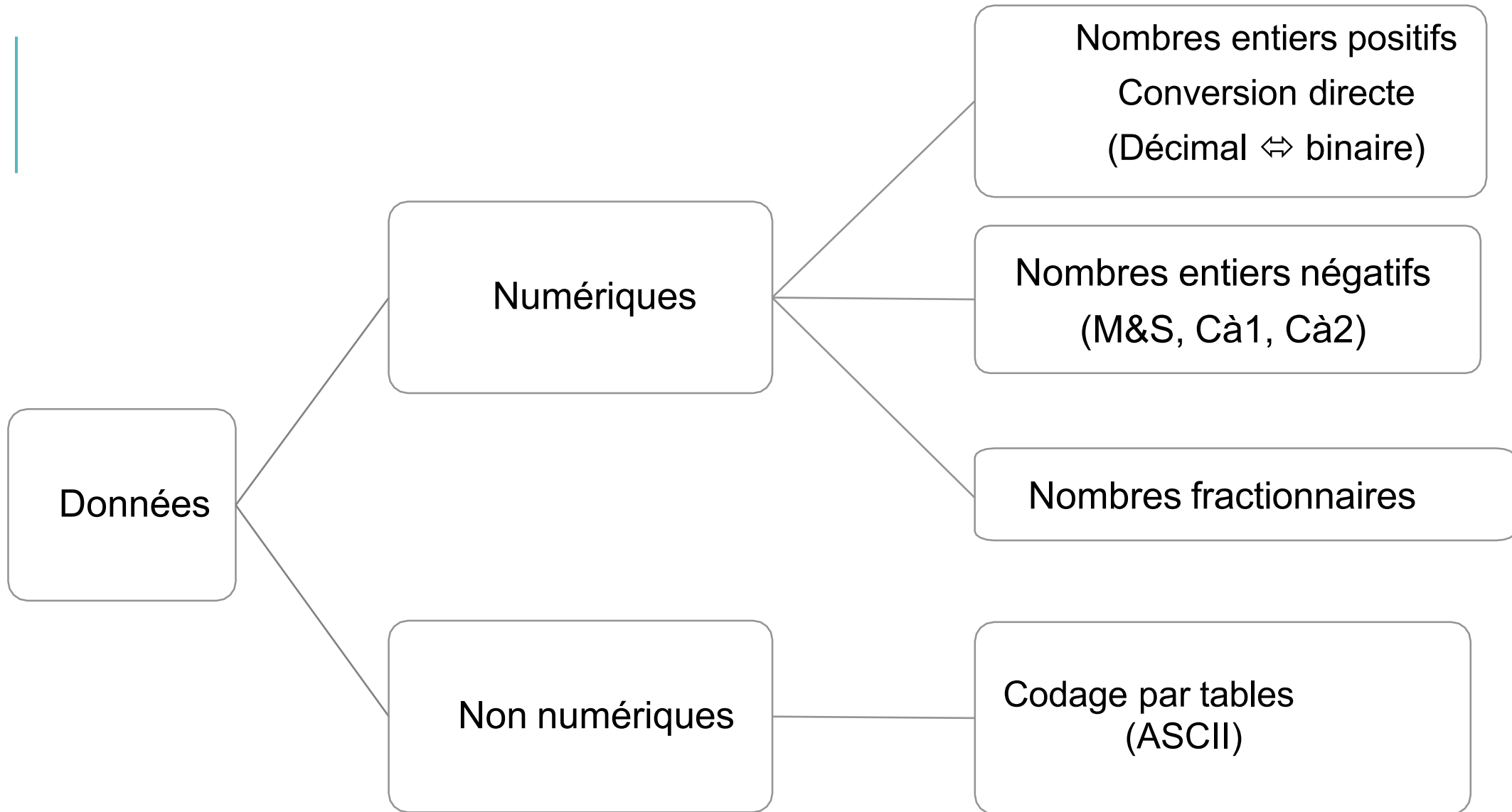
- La mantisse est **normalisée** sous la forme

$$\pm 1, M \cdot 2^{\pm E}$$

- Le 1 précédant la virgule n'est pas codé en machine et est appelé **bit caché**
- Exemple:
 - Simple précision: application, codons le nombre
 - 6, 625
 - Bit de signe (-) donc = 1
 - $(6, 625)_{10} = (110, 1010)_2$
 - $110, 1010 = 1, 101010 \times 2^2$
 - Mantisse = 101010000000000000000000
 - Exposant décalé = $2 + 127 = (129)_{10} = (10000001)_2$

$$\mathbf{-6, 625 = 1\ 10000001\ 101010000000000000000000}$$

11. Différentes représentations relatives aux données



REPRÉSENTATION DU TEXTE

11. Différentes représentations relatives aux données

Codage ASCII

Chaque caractère possède son équivalent en **code** numérique: c'est le **code ASCII** (**American Standard Code for Information Interchange**). Le **code ASCII** de base représentait les caractères sur 7 bits (c'est-à-dire 128 caractères possibles, de 0 à 127).

Binary	Octal	Décimal	Hexadécimal	Caractères
010 0001	041	33	21	!
010 0010	042	34	22	"
010 0011	043	35	23	#
011 0000	060	48	30	0
011 0001	061	49	31	1
011 0010	062	50	32	2
100 0001	101	65	41	A
100 0010	102	66	42	B
110 0001	141	97	61	a
110 0010	142	98	62	b
110 0011	143	99	63	c

11. Différentes représentations relatives aux données

Dec	Hx	Oct	Char		Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL	(null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH	(start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX	(start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX	(end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT	(end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ	(enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK	(acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL	(bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS	(backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB	(horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF	(NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT	(vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF	(NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR	(carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO	(shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI	(shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE	(data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1	(device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2	(device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3	(device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4	(device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK	(negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN	(synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB	(end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN	(cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM	(end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB	(substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC	(escape)	59	3B	073	;	:	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS	(file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS	(group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS	(record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US	(unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Source: www.LookupTables.com

11. Différentes représentations relatives aux données

UNITÉS DE MESURE EN INFORMATIQUE

Multiples d'octets tels que définis par IEC 60027-2					
Préfixe SI			Préfixe binaire		
Nom	Symbole	Valeur	Nom	Symbole	Valeur
kilooctet	ko	10^3	kibioctet	Kio	2^{10}
mégaoctet	Mo	10^6	mébioctet	Mio	2^{20}
gigaoctet	Go	10^9	gibioctet	Gio	2^{30}
téraoctet	To	10^{12}	tébioctet	Tio	2^{40}
pétaoctet	Po	10^{15}	pébioctet	Pio	2^{50}
exaoctet	Eo	10^{18}	exbioctet	Eio	2^{60}
zettaoctet	Zo	10^{21}	zébioctet	Zio	2^{70}
yottaoctet	Yo	10^{24}	yobioctet	Yio	2^{80}

REPRÉSENTATION D'UNE IMAGE NUMÉRIQUE

REPRÉSENTATION D'UNE IMAGE NUMÉRIQUE

Principe général

Une image numérique est composée de **pixels** (éléments d'image). Chaque pixel est codé par **un nombre binaire** représentant sa couleur ou sa luminosité.

Ainsi, une image est une **matrice de nombres binaires** stockée en mémoire.

REPRÉSENTATION D'UNE IMAGE NUMÉRIQUE

Image en niveaux de gris

Chaque pixel représente une **intensité lumineuse** entre le noir et le blanc.

Codage sur **1 octet (8 bits)** → **256 niveaux de gris** (de 0 à 255).

0 = noir

255 = blanc

Exemple :

Pixel = $11001010_2 = 202_{10} \rightarrow$ gris clair

REPRÉSENTATION D'UNE IMAGE NUMÉRIQUE

Image couleur (modèle RVB)

Chaque pixel est composé de **trois composantes** :

Rouge (Red)

Vert (Green)

Bleu (Blue)

Chaque composante est codée sur 8 bits →

$3 \times 8 = 24$ bits par pixel

→ 16 777 216 couleurs possibles

Exemple :

Rouge pur = (255, 0, 0)

Blanc = (255, 255, 255)

Noir = (0, 0, 0)

REPRÉSENTATION D'UNE IMAGE NUMÉRIQUE

Taille d'une image

Taille (en bits) = largeur \times hauteur \times profondeur de couleur

Exemple :

Image 800 \times 600 pixels, 24 bits par pixel \rightarrow

$800 \times 600 \times 24 = 11\,520\,000$ bits = 1,37 Mo

La taille **augmente avec la résolution et la profondeur de couleur.**

REPRÉSENTATION D'UNE IMAGE NUMÉRIQUE

Formats d'image

BMP : image brute, sans compression → très lourde.

JPEG : compression avec perte → photos.

PNG : compression sans perte → schémas, transparence.

GIF : 256 couleurs, adapté aux images animées.

REPRÉSENTATION D'UNE IMAGE NUMÉRIQUE

Résolution et affichage

La **résolution** (dpi ou ppi) indique le **nombre de pixels par unité de longueur**.

Plus la résolution est élevée :
meilleure qualité d'impression,
mais fichier plus lourd.

Exemple :

100 dpi → affichage écran

300 dpi → impression papier

REPRÉSENTATION DU SON

REPRÉSENTATION DU SON

Le son : nature et numérisation

Le son est une **onde mécanique** : variation de pression de l'air en fonction du temps.

Pour le représenter dans un ordinateur, il faut le **numériser** : convertir un signal analogique (continu) en signal numérique (discret).

Trois étapes :

Échantillonnage

Quantification

Codage binaire

REPRÉSENTATION DU SON

Le son : nature et numérisation

Le son est converti en binaire grâce à l'échantillonnage et la quantification.

Plus la fréquence et la profondeur sont grandes, meilleure est la qualité (mais fichiers plus lourds).

La compression réduit la taille mais peut altérer la fidélité.

Les formats principaux : **WAV, MP3, FLAC, AAC.**

REPRÉSENTATION DU SON

L'échantillonnage

Le signal est mesuré à intervalles de temps réguliers.

Fréquence d'échantillonnage : nombre d'échantillons par seconde (en hertz, Hz).

Exemple :

44 100 Hz \rightarrow 44 100 échantillons par seconde (format CD audio).

Théorème de Shannon–Nyquist :

Pour reproduire correctement un son, il faut échantillonner au moins **deux fois la fréquence maximale du signal**.

Exemple : pour un son de 20 000 Hz, il faut au minimum 40 000 Hz d'échantillonnage.

REPRÉSENTATION DU SON

La quantification

Chaque échantillon analogique (valeur réelle) est **arrondi** à la valeur la plus proche dans un ensemble fini.

Cet ensemble est déterminé par la **profondeur de codage** (ou résolution, en bits).

Exemple :

8 bits $\rightarrow 2^8 = 256$ valeurs possibles.

16 bits $\rightarrow 2^{16} = 65536$ valeurs possibles.

Plus la profondeur est grande, **plus le son est fidèle** (mais le fichier est plus lourd).

REPRÉSENTATION DU SON

Le codage binaire

Chaque valeur quantifiée est convertie en **nombre binaire**.

Exemple : un échantillon de valeur 45 $\rightarrow 45 = 00101101_2$.

Ces valeurs binaires sont enregistrées successivement dans un fichier audio.

On peut choisir :

Codage non signé : valeurs de 0 à $2^n - 1$.

Codage signé (complément à deux) : valeurs de -2^{n-1} à $2^{n-1} - 1$) pour les signaux centrés autour de 0)

REPRÉSENTATION DU SON

Taille d'un fichier sonore

Pour un son non compressé (format PCM, .wav) :

Taille

$= \text{Frequence d'échantillonnage} \times \text{Profondeur (en bits)} \times \text{Duree (en s)} \times \text{Nombre de canaux}$

Diviser ensuite par 8 pour obtenir des octets.

Exemple :

Stéréo (2 canaux), 44 100 Hz, 16 bits, 1 minute :

$$44\,100 \times 16 \times 60 \times 2 / 8 = 10\,584\,000 \text{ octets} \approx 10,1 \text{ Mo.}$$

REPRÉSENTATION DU SON

Compression du son

Le son brut (WAV) est volumineux.

On applique une **compression** :

Sans perte : FLAC, ALAC (même qualité que l'original).

Avec perte : MP3, AAC, OGG (suppression de fréquences peu perceptibles).

But : réduire la taille tout en gardant une qualité acceptable.

REPRÉSENTATION DU SON

Paramètre	Rôle	Exemple
Fréquence d'échantillonnage	Détermine la bande passante du son	44,1 kHz
Résolution (profondeur)	Détermine la précision des amplitudes	16 bits
Canaux	Mono = 1, Stéréo = 2	Stéréo CD
Compression	Réduit la taille du fichier	MP3, FLAC

PARTIE II : LE WEB ET SES LANGAGES

INTRODUCTION AU LANGAGE HTML

LE WEB

LE WEB

Le Web est un nom donné à l'ensemble hyperdocumentaire formé par la totalité des documents disponibles sur Internet à travers un protocole particulier : HTTP.

LE WEB

Définition 2

- Le Web est un assemblage de technologies diverses. Chaque technologie s'occupe d'un problème ou aspect particulier de la chaîne.
 - Normes de contenu : XHTML, HTML, CSS, XML
 - Automatisation de contenu : Javascript, VBScript
 - Processeurs : Perl, Php, Python, ASP.Net
 - Templating: CMS (WordPress, Joomla)

LE WEB

1.3. Principe

- Le principe central du Web, c'est la transaction HTTP :
- Un émetteur (client) envoie une REQUETE
- Un serveur y répond.
- C'est une généralisation du « client-serveur » à partir de technologies publiques.

LE WEB

La requête

- Un message court et formaté
- Sa construction suit une norme HTTP
- Demande un document à travers une URL (GET)
- Dépose des données sur le serveur (POST)

LE WEB

URL (Uniform Resource Locator).

- Un URL est le moyen de nommer un objet dans le monde WWW. La syntaxe d'un URL est la suivante :

type:serveur:port/chemin_d'accès/fichier#etiquette?parametres

Le type de serveur :

- ⊕ http : pour les URL provenant de serveurs WWW
- ⊕ ftp : pour les fichiers à transférer
- ⊕ telnet : pour ouvrir une session interactive

LE LANGUAGE HTML

LE LANGAGE HTML

⊕ Ce n'est pas uniquement écrire des pages en HTML, il faut penser en terme de **projet**:

- ⊕ Définir le contenu
- ⊕ Trouver une arborescence ergonomique
- ⊕ Appliquer / Respecter la charte graphique
- ⊕ Produire les pages
- ⊕ Installer le site sur le serveur
- ⊕ Maintenance, politique de mise à jour

LE LANGAGE HTML

HTML, L'ORIGINE

- ⊕ HTML **Hyper Text Markup Language** est né en 1989 sous l'impulsion de Tim Berners Lee, " inventeur " du Web.
- ⊕ HTML est basé sur **SGML (Structured Markup Language)**, qui est une vieille norme utilisée pour la description de documents. Elle est conçue pour les grosses documentations techniques.
- ⊕ HTML est une **instance** de SGML.

LE LANGAGE HTML

2.3. HTML, LES PRINCIPES

- ⊕ Il contient des **commandes**, implémentées par des **balises** pour marquer les différents types de texte (titres, paragraphe, listes ...) , pour inclure des images, des formulaires, des liens ...
- ⊕ C'est un **langage à balisage** qui décrit la structure logique d'un document **hypertexte**. Il a volontairement été conçu pour être simple.

LE LANGAGE HTML

2.4. L'HYPERTEXTE

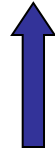
- ⊕ Le langage **HTML** permet de créer des documents interactifs grâce à des **liens hypertextes**, qui relient votre document à d'autres documents.
- ⊕ En **cliquant** sur une zone de texte (ou une image, un logo) mise en évidence, on peut accéder à un **nouveau document** situé sur un autre ordinateur en n'importe quel point du globe.

LE LANGAGE HTML

2.5. INTRODUCTION AU MARQUAGE - 1

- ✚ Pour décrire un fichier hypertexte, le langage HTML insère des **balises** dans le texte du document :

```
<marqueur> ici votre texte </marqueur>
```



Début de mise en forme



Fin de mise en forme

✚ Synonymes: marqueur, élément, tag, balise.

LE LANGAGE HTML

2.5. Introduction au marquage

⊕ Ces balises peuvent être insérées n'importe où dans le texte, entre 2 phrases, mots, lettres ...

```
<gras>Le <italique> cours </italique> HTML</gras>
```



Le *cours* HTML

LE LANGAGE HTML

2.5. Introduction au marquage - 3

⊕ Il faut respecter une logique d'imbrication:

Bon:

```
<gras><italique> Le cours HTML </italique> </gras>
```

Mauvais:

```
<gras><italique> Le cours HTML</gras></italique>
```

LE WEB

2.5. INTRODUCTION AU MARQUAGE - 4

⊕ Le langage HTML est sensible à la casse, toujours écrire en minuscules.

Bon:

```
<gras><italique> Le cours HTML </italique> </gras>
```

Mauvais:

```
<GRAS><italique> Le cours HTML </italique> </GRAS>
```

```
<Gras><ITALIQUE> Le cours HTML </ITALIQUE> </Gras>
```

```
<GRAS><ITALIQUE> Le cours HTML </italique> </GRAS>
```

LE WEB

2.6. LES ATTRIBUTS

⊕ Les balises peuvent posséder un ou plusieurs attributs qui permettent de spécifier l'action de la balise. Toujours mettre la valeur de l'attribut entre guillemets.

```
<marqueur attribut="argument">texte</marqueur>
```

```
<marqueur attribut1="argument" attribut2="argument">texte</marqueur>
```

LE WEB

2.7. LES COMMENTAIRES

```
<!-- Voici un commentaire HTML -->
```

```
<!--
```

```
Voici un commentaire HTML qui  
peut se placer sur plusieurs lignes
```

```
-->
```

LE WEB

2.8. QUE CHOISIR POUR ÉCRIRE DE L' HTML?

⊕ **A la main, avec un éditeur de texte**

⊕ SimpleText, Bbedit, Emacs , WordPad, Bloc-notes ...

⊕ **Avec un logiciel « assistant » au code HTML**

⊕ PageSpinner sur Macintosh, Amaya sur Unix, HTML-Kit sur Windows ...

⊕ **A l'aide d'un programme dit "WYSIWYG"**

⊕ Dreamweaver , Golive, Netscape Composer, FrontPage, Claris Home page

...

LE WEB

2.9. QUE CHOISIR POUR LIRE HTML?

⊕ Le client doit pouvoir interpréter HTML et afficher le résultat. Ils sont divers, tournant sur des systèmes différents:

- Netscape Navigator,
- Mozilla,
- Internet explorer,
- Safari,
- Opera,
- Google Chrome
-etc.

LE WEB

2.10. NOTE SUR LES CARACTÈRES ACCENTUÉS

⊕ Vous pouvez rencontrer le codage ASCII sur 7 bits spécifié par la norme pour afficher les caractères accentués ou spéciaux. Ceux ci devront faire l'objet d'un codage spécial au sein du fichier HTML.

- é s'écrit é
- être s'écrit être

⊕ Les serveurs Web acceptent les caractères accentués de la norme iso-8859-1. On spécifie l'encodage dans le fichier HTML.

```
<meta http-equiv="Content-Type" content="text/html;  
      charset=utf-8" />
```

Code texte	Les accents
à	à
â	â
è	è
é	é
ê	ê
î	î
ë	ë
ô	ô
ö	ö
ù	ù
û	û

LE WEB

2.11. STRUCTURE DE BASE D'UN FICHIER HTML

<!DOCTYPE>

Renseigne le navigateur sur le langage utilisé par votre page web.

<html>

<head>

Donne des informations à votre navigateur pour traiter correctement les informations se trouvant dans le «body»

</head>

<body>

La partie visible de votre page qui sera affichée par le navigateur

</body>

</html>

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="UTF-8" />
<title>Document sans nom</title>
</head>
<body>
</body>
</html>
```

- Une balise **<html>** contenant une seule balise **<head>** et une seule balise **<body>**, SAUF pour les pages comprenant des frames.

LE WEB

2.12. ANALYSE DES BALISES

`<html> .. </html>` Délimite le début et la fin du document

`<head> .. </head>` Entête du document, contient des méta-informations

Ex: `<title>.. </title>` titre de la fenêtre dans laquelle la page s'affiche

`<meta> . </meta>` méta-informations sur le document

`<script> . </script>` script ou référence

`<body> . . </body>` Corps du document

LE WEB

RAPPELS

- ⊕ Les balises sont sensibles à la casse et sont écrites en **minuscules**.
- ⊕ Les noms d'attributs sont sensibles à la casse, doivent être écrits en **minuscules** et encadrées par des **guillemets**. Tous les attributs doivent recevoir **une valeur**.
- ⊕ Les balises fermantes sont obligatoires.
- ⊕ Les balises vides sont signalées par une balise spéciale. (ex: `
`)

UN PEU PLUS LOIN DANS HTML

```
<html>
<head>
<title>Mon deuxieme fichier</title>
</head>
<body>
<h2>Hello World</h2>
<p>
L'emplacement de votre nouveau mat&eacute;riel est-il &agrave; proximit&eacute; (moins de 5m) d'une prise r&eacute;seau
libre ? Si oui, relevez le num&eacute;ro de la prise (il est &eacute;crit sur une plaque de c&eacute;ramique bleue, il faut
respecter la distinction entre lettres majuscules et minuscules). Sinon, il faut faire installer une nouvelle prise par le
Service Travaux, auquel vous devrez adresser un bon de session interne. La suite des op&eacute;rations aura lieu quand vous
aurez une prise disponible. Dans le doute, interrogez par courrier &eacute;lectronique &agrave; elhoussaini.m@gmail.com
</p>
<p>Votre demande doit nous parvenir par l'interm&eacute;diaire du correspondant informatique de votre Unit&eacute;. Elle se
composera de deux parties : un bon de session interne envoy&eacute; au Service Informatique Scientifique et un courrier
&eacute;lectronique adress&eacute; &agrave; monAdresse qui contiendra toutes les informations utiles (dont le num&eacute;ro
de la bonne session). </p>
<ol>
<li> le premier &eacute;l&eacute;ment
<li> le deuxi&egrave;me &eacute;l&eacute;ment
<li> le troisi&egrave;me &eacute;l&eacute;ment
</ol>
<p><a href="http://www.ucd.ac.ma">Vous trouverez ici la suite du texte !! </a></p>
</body>
</html>
```

LE WEB

RÉSULTAT



Hello World

L'emplacement de votre nouveau matériel est-il à proximité (moins de 5m) d'une prise réseau libre ? Si oui, relevez le numéro de la prise (il est écrit sur une plaque de céramique bleue, il faut respecter la distinction entre lettres majuscules et minuscules). Sinon, il faut faire installer une nouvelle prise par le Service Travaux, auquel vous devrez adresser un bon de session interne. La suite des opérations aura lieu quand vous aurez une prise disponible. Dans le doute, interrogez par courrier électronique à elhoussaini.m@gmail.com

Votre demande doit nous parvenir par l'intermédiaire du correspondant informatique de votre Unité. Elle se composera de deux parties : un bon de session interne envoyé au Service Informatique Scientifique et un courrier électronique adressé à elhoussaini.m@gmail.com qui contiendra toutes les informations utiles (dont le numéro du bon de session).

1. le premier élément
2. le deuxième élément
3. le troisième élément

[Vous trouverez ici la suite du texte !!](#)

LE WEB

2.13. ÉLÉMENTS DU LANGAGE HTML

1- LES DÉLIMITEURS

`<p> . . </p>`

Paragraphe

`
`

Saut à la ligne

`<hr />`

Règle horizontale

`<pre> . . </pre>`

Affiche le texte tel quel

`<blockquote> . . </blockquote>` Tabulation

`<address> . . </address>` Indique des informations de contact

2.13. ÉLÉMENTS DU LANGAGE HTML

2- LES STYLES

`<hn> . . </hn>` Titre de niveau n, de 1 à 6

`<H1>`Une très grande taille`</H1>`

Une très grande taille

`<H2>`Une grande taille`</H2>`

Une grande taille

`<H3>`Une taille moyenne`</H3>`

Une taille moyenne

`<H4>`Une petite taille`</H4>`

Une petite taille

`<H5>`Une très petite taille`</H5>`

Une très petite taille

`<H6>`Une taille vraiment minuscule`</H6>`

Une taille vraiment minuscule

2.13. ÉLÉMENTS DU LANGAGE HTML

3- L'ALIGNEMENT

Pour centrer le texte

```
<center> ... </center>
```

Ou alors

```
<P align="right">...</P>
```

```
<P align="center">...</P>
```

```
<P align="left">...</P>
```

```
<P align="justify">...</P>
```

Retrait de texte (décaler le texte vers la droite (Tabulation))

```
<blockquote>...</blockquote>
```

LE WEB

2.13. ÉLÉMENTS DU LANGAGE HTML

4- LES SÉPARATEURS

- * Les caractères "**espace**", "**tabulation**" et "**retour à la ligne**" sont considéré comme un seul espace.
- * Plusieurs espaces consécutifs : seul le premier est pris en compte.
- * Pour afficher plusieurs espaces il faut utiliser l'entité caractère: ** **

LE WEB

2.13. ÉLÉMENTS DU LANGAGE HTML

5- MISE EN FORME DES CARACTÈRES

- * Spécification d'une police locale

```
<font face="police" size="s|+s|-s" color="#rrvvbb">...  
</font>
```

- * face : la première police installée est utilisée. Ne proposer que des polices dont on est sûr qu'elles sont disponibles sur l'ordinateur de consultation :

- Arial/Helvetica
- Times New Roman/Times

- * size: taille de la police

- 1...7(taille absolue) (3 est la taille par défaut)
- +2,-3...(taille relative par rapport à la valeur par défaut)

- * color : détermine la couleur des caractères

LE WEB

2.13. Éléments du langage HTML

6- Éléments de listes

<code> . . </code>	Liste non triée, liste à puces
<code> . . </code>	Liste triée, liste à numéros
<code></code>	Élément de la liste

Principaux attributs:

`start` = valeur de départ (`n`) , Permet de démarrer la numérotation à une valeur spécifique autre que 1

`type` = type d'ordre

listes ordonnées (`A, a, I, i, 1`) , changer le type de numérotation

listes non ordonnées (`disc, circle, square`) , choisir la forme de la puce

Et aussi: `menu, dir, dl, dt, dd...`

LE WEB

2.13. Éléments du langage HTML

7- Balises de ligne

`texte gras`

texte gras

`<i>texte italique</i>`

texte italique

`<U>texte souligné</U>`

texte souligné

`<big>texte gros</big>`

texte gros

`<small>texte petit</small>`

texte petit

`
`

retour à la ligne

`_{..}`

texte en indice

`^{..}`

texte en exposant

`<strike>..</strike>`

texte barré

LE WEB

2.13. Éléments du langage HTML

8- Les liens - 1

`<a> . . `

Création d'un lien hypertexte

Principaux attributs:

href = `url`

name = `chaîne de caractères`

Exemple :

`UCD `

LE WEB

2.13. Éléments du langage HTML

8- Les liens - 2

ou vers un point d'ancrage du document

```
<a name = "ref" >référence</a>
```

```
<p>
```

.

```
<p>
```

.

.

.

```
<p>
```

.

```
<a href ="monfichier.html#ref">Vers la référence</a>
```


LE WEB

2.13. Éléments du langage HTML

8- Les liens - 3

Courrier électronique

L'adressage de courrier est possible en HTML de deux manières différentes selon que l'on utilise les formulaires ou non. La première façon est d'utiliser la commande suivante:

```
<a href="mailto:professeurAgmail.com">contactez moi</a>
```

LE WEB

2.13. Éléments du langage HTML

8- Les liens - 4

⊕ Les adresses du Web ou URL (Uniform Resource Locator) indiquées dans « href », sont du type:

`http://www.esef.ucd.ac.ma/3ISI/Cours/intro-html.html#Lien`

⊕ Le protocole: `http`

⊕ Le serveur: `esef.ucd.ac.ma`

⊕ Le fichier: `3IILM/Cours/intro-html.html`

⊕ Un ancrage: `#Lien`

LE WEB

2.13. Éléments du langage HTML

8- Les liens - 5

- ⊕ L'adresse indiquée dans le lien (*URL*) peut être **absolue**, à utiliser lorsque le document appelé n'est pas situé sur le même site que le document appelant, elle inclut tout le chemin en commençant par le protocole.
- ⊕ Ou **relative**, à utiliser lorsque le document appelé est situé sur le même site que le document appelant, elle n'inclut qu'une partie du chemin:

Mon_repertoire/monfichier.html

“.” : fichier en cours

“..” : fichier parent



2.13. Eléments du langage HTML

9- Les tableaux -1

<table> . . </table>

englobe tous les éléments du tableau.

Principaux attributs:

align = Contrôle l'alignement (left, center, right, justify)

bgcolor = détermine la couleur du tableau. ("#xxxxxx")

border = largeur de la bordure autour du tableau, en pixels (0 : invisible)(n)

cellpadding = espacement dans les cellule (n) (en pixels)

cellspacing = espacement entre les cellules (n)

width = largeur du tableau ("x%" | "x"),

height= Hauteur du tableau ("x%" | "x")

frame = définit les côtés visibles d'un tableau (void, above, below, hside, lhs, rhs, vside, box, border)

rules = définit la manière dont on va appliquer l'attribut frame (none, groups, rows, cols, all)

Autres attributs:

Background, bordercolor, bordercolordark, bordercolorlight,
class, dir, id, lang, style

LE WEB

2.13. Éléments du langage HTML

9- Les tableaux -2

Remarques :

- En principe, la taille du tableau et celle des cellules qui s'adaptent automatiquement à l'encombrement des données. Cependant, il est possible de contrôler la taille exacte du tableau en utilisant l'attribut **width** avec des **valeurs** en **pixels** ou en **pourcentage**.
- Il est possible d'imbriquer des tableaux (placer un tableau à l'intérieur d'une cellule).
- Il est possible d'insérer des images dans les cellules d'un tableau ou d'en colorer le fond.

Balises associées : caption, col, colgroup, tbody, td, tfoot, th, thead, tr

LE WEB

2.13. Eléments du langage HTML

9- Les tableaux -3

<caption > </caption>

La balise HTML **CAPTION** permet de mettre un titre au tableau.

L'interprétation de la balise HTML **CAPTION** est très différente suivant le navigateur.

Principaux attributs :

```
Align = (top, bottom, left, right)
class
dir
id
lang
style
```

LE WEB

2.13. Eléments du langage HTML

9- Les tableaux -4

<colgroup > </colgroup>

La balise HTML `colgroup` permet de faire un groupe de colonne d'un tableau pour leurs assigner les mêmes propriétés

La balise HTML `colgroup` est utilisée avec la balise HTML `col`.

Exemple :

```
<table border="1" >
<colgroup>
    <col width="50px"  aling="center" />
    <col width="100px" aling= "left" />
</colgroup>
<tr>
    ---
    ---
</tr>
</table>
```

LE WEB

2.13. Éléments du langage HTML

9- Les tableaux -5

<tbody > </tbody>

Permet de définir un ensemble de lignes comme le "corps" du tableau.

tbody est utilisée avec **tfoot** et **thead** respectivement "pied" et "entête" de tableau.

Principaux attributs :

align

class

dir

id

lang

style

title

Valign = (top, middle, bottom, baseline)

LE WEB

2.13. Éléments du langage HTML

9- Les tableaux -6

`<tr> . . </tr>`

(Table Row) renferme les éléments descriptifs d'une ligne de tableau.

Principaux attributs :

```
align =  
  (left, center, right,  
   justify)  
valign = (top, middle,  
          bottom)  
border = (n)  
background
```

bgcolor

class

dir

id

lang

style

LE WEB

2.13. Éléments du langage HTML

9- Les tableaux -7

`<td> . . </td>`

(Table Data) Définit une cellule de données

La balise HTML et XHTML **td** est contenu dans la balise HTML et XHTML **<tr>**.

Principaux attributs :

`align`

`valign`

`bgcolor`

`colspan, rowspan = (n)`

`height, width = (n)`

`background = (URL/Image.gif)`

`class`

`dir`

`id`

`lang`

`nowrap` = spécifie qu'il ne doit pas avoir de rupture de ligne dans la cellule.

`style`

`valign`

`width`

LE WEB

2.13. Éléments du langage HTML

9- Les tableaux -8

`<th> </th>`

(Table Header) définit une cellule titre du tableau, généralement située dans la première ligne ou colonne du tableau.

Principaux attributs :

`align`

`class`

`dir`

`id`

`lang`

`style`

`title`

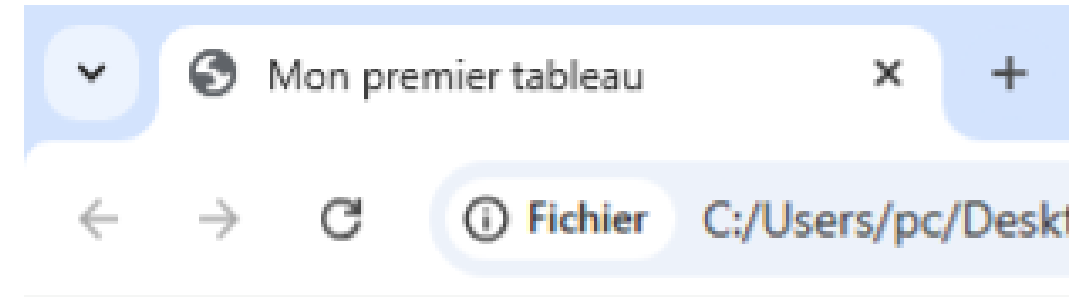
`Valign = (top, middle, bottom, baseline)`

LE WEB

2.13. Éléments du langage HTML

9- Exemple d'un tableau -9

```
<html>
<head> <title>Mon premier tableau</title> </head>
<table border="1" align="left" bgcolor="#33f9ff">
<caption align="bottom">
<b>C'est le titre du tableau</b>
</caption>
<tr>
<th>Joueur</th> <th>Ahmed</th> <th>Karim</th>
</tr>
<tr>
<th>Age</th> <td>24</td> <td>26</td>
</tr>
<tr>
<th>Poids</th> <td>75</td> <td>81</td>
</tr>
<tr>
<th>Distance</th> <td>45</td> <td>52</td>
</tr>
</table>
</html>
```



Joueur	Ahmed	Karim
Age	24	26
Poids	75	81
Distance	45	52

C'est le titre du tableau

LE WEB

2.13. Éléments du langage HTML

10- Les images - 1

` . . `

HTML permet d'insérer des images dans les documents. Pour cela il faut employer la balise vide ``

Les attributs sont :

`src` = **est obligatoire**, et contient le chemin vers l'image qu'on veut intégrer.

`alt` = description courte de l'image, texte apparaissant dans une info-bulle quand la souris est positionnée pendant 1 à 2 secondes sur l'image.

`align` = précise l'alignement vertical et horizontal (`bottom`, `middle`, `top`, `left`, `right`)

LE WEB

2.13. Eléments du langage HTML

10- Les images - 2

⊕ Attention au poids des images, il est important d'optimiser son fichier image.

⊕ Trois formats d'image peuvent être mis sur l'internet :- le format **GIF**, il est utilisé pour des images contenant peu de couleur. En effet le nombre de couleur dans le gif est de 256 dont une couleur transparente.- le format **JPG**, il est utilisé pour des images contenant beaucoup de couleur, une photo par exemple.- le format **PNG**, concurrent du gif il peut contenir plus de couleurs que le gif.

Cependant, il reste encore peu utilisé.

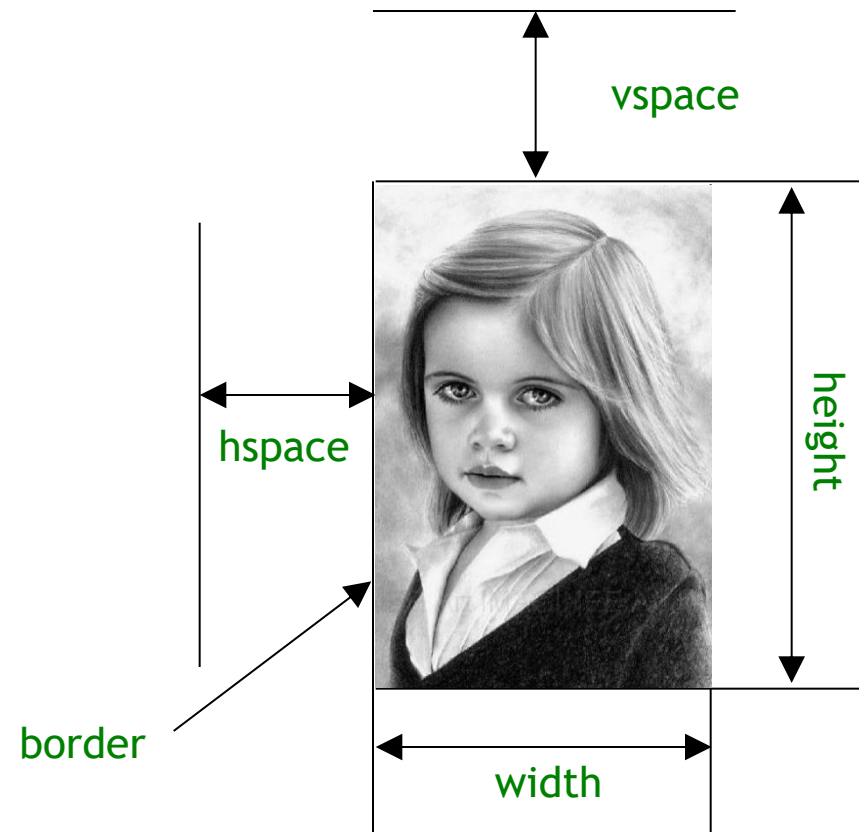
⊕ Attention aux droits sur les images. La seule image qui vous appartient est celle que vous avez prise avec votre matériel photo. Il faut aussi l'autorisation des personnes figurant sur la photo.

LE WEB

2.13. Éléments du langage HTML

10- Les images - 3

attributs de positionnement



Images réactives

- ⊕ Les images réactives permettent à l'internaute de naviguer de manière plus conviviale via des liens.
- ⊕ Une image réactive est décomposée en différentes parties permettant d'accéder à des pages différentes.
- ⊕ Etant donné que l'utilisateur doit savoir où cliquer, les différentes parties doivent être séparées correctement.

LE WEB

2.13. Éléments du langage HTML

10- Les images - 5

Une image réactive est définie par les balises `<map>` et `</map>`.

```
<map name="nomdelamap">
  <area shape="forme" coords="x,y,..." href="url" nohref>
</map>
```

La balise `<area>` permet de définir chaque région. Les attributs de la balise `<area>` sont les suivants :

- `shape` : permet de définir la forme d'une région (`rect`, `poly`, `circle`, `default`).
- `coords` : définit une liste de coordonnées séparées par une virgule de la région.
- `href` : définit l'URL de la ressource internet à laquelle la région est liée.
- `nohref` : indique que la région est une zone qui n'est liée à aucune ressource internet.

Pour référencer une image réactive une fois celle-ci construite, vous devez utiliser la balise suivante :


```

```

LE WEB

2.13. Éléments du langage HTML

10- Les images - 6

Exemple	Source	Résultat
	<pre><map name="mamap"> <area shape="rect" coords="0,0,50,50" href="news.htm"> <area shape="rect" coords="51,0,100,50" href="contact.htm"> <area shape="rect" coords="101,0,150,50" href="apropos.htm"> </map> </pre>	

Remarque

Si vous souhaitez prendre en compte les navigateurs qui ne supportent pas les images réactives, utilisez :

```
<a href="menutexte.htm">
```

```
</a>
```

Ainsi, si l'utilisateur clique sur l'image, il verra un menu texte lui permettant de choisir une URL.

LE WEB

UN FICHER DIFFÉRENT ...

```
<html>

<head>

<title>des cadres</title>

</head>

<frameset cols="25%,75%">

<frame name="menu" src="menu.html"
  scrolling="yes">

<frame name="cible" src="cible.html"
  scrolling="no">

<noframes>
```

```
<body>

message pour les
navigateurs ne supportant
pas les cadres

</body>

</noframes>

</frameset>

</html>
```

LE WEB

2.13. ÉLÉMENTS DU LANGAGE HTML

11- LES CADRES (LES FRAMES) -1

- ⊕ Ils permettent de diviser la fenêtre principale du navigateur en plusieurs sous fenêtres de taille réduite, chacune d'entre elles affichant un document différent.
- ⊕ Utilisation: Au moins une des divisions est généralement employée pour afficher un menu de parcours du site.
- ⊕ Pour créer une page contenant des frames:
 - Définir le découpage (frameset).
 - Définir le contenu des cadres (pages HTML classiques).
- ⊕ Certains navigateurs ne les supportent pas, il est possible de les désactiver, il faut donc prévoir un fichier de substitution inscrit entre les balises:

<noframes> .. </noframes>

LE WEB

2.13. ÉLÉMENTS DU LANGAGE HTML

11- LES CADRES (LES FRAMES) -2

⊕ `<frameset rows|cols="d1,d2,d3...,*">`

⊕ `rows|cols="d1,d2,d3...,*"`

- `rows` : division en **lignes** (horizontale)
- `cols` : pour une division en **colonnes** (verticale)
- `d1,d2, d3...` : indique la hauteur (largeur) en pixel (ou en pourcentage) de chaque ligne (colonne) ; le séparateur de valeurs est la virgule.
 - * permet d'attribuer à la dernière ligne (colonne) tout l'espace restant.

⊕ Contenu de `<frameset>...</frameset>`

- Chaque frameset contient autant d'éléments `<frame>` que d'arguments dans l'option `rows` ou `cols`.
- Chaque `<frame>` indique quelle page afficher.

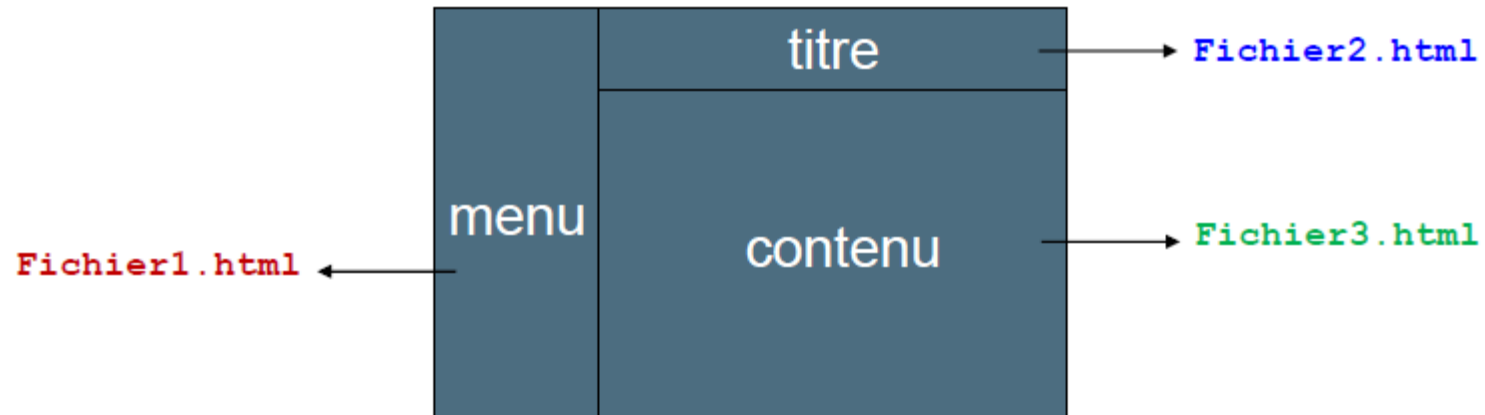
LE WEB

2.13. ÉLÉMENTS DU LANGAGE HTML

11- LES CADRES (LES FRAMES) -3

Fichier frame.html

```
<FRAMESET cols="70,424">  
  <FRAME name="menu" src="Fichier1.html">  
  <FRAMESET rows="69,487">  
    <FRAME name="titre" src="Fichier2.html">  
    <FRAME name="contenu" src="Fichier3.html">  
  </FRAMESET>  
</FRAMESET>
```



Un formulaire est une section d'un document qui contient des contrôles interactifs permettant à un utilisateur de fournir des informations.

- Permettent d'interagir avec l'utilisateur en lui proposant d'entrer des informations
- En HTML : uniquement l'interface de formulaire
- L'essentiel du travail sera fait par le **script** qui traitera la soumission du formulaire

LE WEB

2.13. Éléments du langage HTML

12- Les formulaires

12.1- Balise <form>

Un formulaire HTML est placé à l'intérieur d'une balise <form> .

Celle-ci prend les attributs suivants :

- **action** URL du script auquel sera soumis le formulaire.
- **method** Méthode HTTP, valant soit "get" soit "post"
- **enctype** qui spécifie le codage des données dans l'URL, valeur par défaut "application/x-www-form-urlencoded".

Exemple (Formulaire élémentaire)

```
<form action="action.php" method="get">  
<div><input type="submit"></div>  
</form>
```


LE WEB

2.13. Éléments du langage HTML

12- Les formulaires

12.2- Ensembles de champ

En HTML, il est interdit de mettre des champs de formulaire directement à l'intérieur d'un `<form>` . Il faut d'abord les regrouper :

- Dans des paragraphes `<p>` si les champs de formulaires sont à l'intérieur de paragraphes de textes (rare).
- Dans des ensembles de champ `<fieldset>` pour regrouper des champs de formulaire de sémantique proche. On pourra alors donner une légende à l'ensemble de champs avec la balise `<legend>` .
- Sinon dans des divisions `<div>` sans contenu sémantique.

Exemple (Ensemble de champ)

```
<fieldset>
  <legend>Mensurations</legend>
  <input type="text" name="taille">
  <input type="text" name="poids">
</fieldset>
```

LE WEB

2.13. Éléments du langage HTML

12- Les formulaires

12.3- Étiquettes

- La plupart des champs sont naturellement accompagnés d'une **étiquette** (`<label>`).
- On peut la placer où on veut, en général juste à gauche ou à droite du champ.
- Son attribut **for** référence l'attribut **id** du champ correspondant.
- Dans les navigateurs graphiques, un clic sur l'étiquette d'un champ permet en général de sélectionner le champ.

Exemple (Étiquette)

```
<label for="taille">Taille:</label>  
<input type="text" name="taille" id="taille">
```

LE WEB

2.13. Éléments du langage HTML

12- Les formulaires

12.4- Champs de saisie

- La balise `<input>` a une utilisation très vaste dans les formulaires. Elle représente un champ de saisie.
- L'attribut `type` détermine le type (texte, mot de passe, liste, etc.) du champ.
- L'attribut `name` (nom du paramètre de la requête HTTP) est **obligatoire** (sauf pour les types `"reset"` et `"submit"`) ; il permet de préciser au serveur à quelle saisie on fait référence.

Exemple (Zone de texte pour écrire un commentaire)

```
<input type="text" name="Commentaire">
```

LE WEB

2.13. Éléments du langage HTML

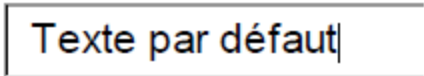
12- Les formulaires

12.5- Saisie d'une ligne de texte

- `type="text"` est utilisé pour la saisie d'un texte dont la taille est inférieure à une ligne.
- L'attribut `value` permet de préciser la valeur par défaut (facultatif).
- La taille maximale de la chaîne de caractères à saisir peut être spécifiée à l'aide de l'attribut `maxlength` (facultatif).

Exemple

```
<input type="text" name="prenom" value="Texte par défaut" maxlength="50">
```



LE WEB

2.13. Éléments du langage HTML

12- Les formulaires

12.6- Saisie d'un mot de passe

- `type="password"` est utilisé pour la saisie d'un texte dont les caractères sont remplacés par des astérisques : c'est généralement utilisé pour la saisie des mots de passe. Le mot de passe est quand même transmis en clair au serveur !
- L'attribut `value` permet de préciser la valeur par défaut (facultatif).
- La taille maximale de la chaîne de caractères à saisir peut être spécifiée à l'aide de l'attribut `maxlength` (facultatif).

Exemple

```
<input type="password" name="pwd" value="12345678">
```

LE WEB

2.13. Éléments du langage HTML

12- Les formulaires

12.7- Choix multiples parmi une liste

- `type="checkbox"` permet de choisir plusieurs éléments parmi une liste de possibilités.
- Cela se matérialise sous forme de cases à cocher.
- La valeur retournée est **obligatoirement** précisée à l'aide de l'attribut `value`.
- L'attribut `checked="checked"` permet de cocher la case par défaut.
- Certains langages côté serveurs imposent que les noms de champs de formulaire à choix multiples se terminent par `[]`.

Exemple

```
<input type="checkbox" name="pub[]" value="site" checked="checked" id="pub-site">  
<label for="pub-site">Recevoir des offres de notre site</label>  
<input type="checkbox" name="pub[]" checked="checked" value="externe" id="pub-externe">  
<label for="pub-externe">Recevoir des offres externes</label>
```

Choix 1	<input checked="" type="checkbox"/>
Choix 2	<input type="checkbox"/>
Choix 3	<input type="checkbox"/>
Choix 4	<input type="checkbox"/>

LE WEB

2.13. Éléments du langage HTML

12- Les formulaires

12.8- Choix unique parmi une liste

- `type="radio"` permet de choisir un seul élément parmi une liste de possibilités.
- Cela se matérialise sous forme de boutons radio.
- La valeur retournée est **obligatoirement** précisée à l'aide de l'attribut `value`.
- L'attribut `checked="checked"` permet de préciser la valeur par défaut.

Choix 1	<input checked="" type="radio"/>
Choix 2	<input type="radio"/>
Choix 3	<input type="radio"/>
Choix 4	<input type="radio"/>

Exemple Recevoir de la pub:

```
<input type="radio" name="pub" value="oui" id="pub-oui" checked="checked">
<label for="pub-oui">oui</label>

<input type="radio" name="pub" value="non" id="pub-non">
<label for="pub-non">non</label>
```

LE WEB

2.13. Éléments du langage HTML

12- Les formulaires

12.9- Fichiers joints

- `type="file"` permet de joindre au formulaire un fichier.
- À cause de la taille de la requête due au téléchargement (**upload**) du fichier, il faut impérativement utiliser la méthode POST et l'encodage `multipart/form-data` du conteneur `<form>` .

Exemple

```
<label for="fichier">Fichier:</label>  
<input type="file" name="fichier" id="fichier">
```


LE WEB

2.13. Éléments du langage HTML

12- Les formulaires

12.10 Champs cachés

- `type="hidden"` permet de cacher des champs au client mais leur contenu est envoyé avec le formulaire.
- Ceci permet de préciser des informations, en utilisant l'attribut `value` , concernant l'interaction client/serveur.
- C'est à utiliser **avec précaution** car cela peut être à l'origine de problèmes de sécurité assez graves : ne pas oublier que le client peut éditer la page à la main pour changer la valeur de ces champs!

Exemple

```
<input type="hidden" name="monnaie_utilisee" value="EUR">  
<input type="hidden" name="customerCB" value="c2415-345-8563">
```

LE WEB

2.13. Éléments du langage HTML

12- Les formulaires

12.11 Ré-initialisation d'un formulaire

- `type="reset"` permet de réinitialiser le formulaire en affectant aux différents champs leur valeur par défaut.
- L'attribut `value` permet de changer le texte du bouton correspondant.

Exemple

```
<input type="reset" value="Tout effacer">
```

LE WEB

2.13. Éléments du langage HTML

12- Les formulaires

12.12 Soumettre le formulaire

- `type="submit"` permet de soumettre le formulaire.
- Le client envoie le contenu du formulaire à l'adresse précisée par l'attribut `action` de la balise `<form>`.
- L'attribut `value` permet de changer le texte du bouton correspondant.

Exemple

```
<input type="submit" value="Envoyer">
```

LE WEB

2.13. Éléments du langage HTML

12- Les formulaires

12.13 Saisie de plusieurs lignes de texte

- Pour les saisies multiligne, on utilise la balise `<textarea>` .
- Le texte délimité par cette balise permet d'initialiser la valeur par défaut du champ.
- La balise fermante est **obligatoire** même si le champ est vide.
- Les attributs `rows` et `cols` (obligatoires) permettent de spécifier la taille en lignes et colonnes de la fenêtre de saisie.

Exemple

```
<textarea name="bio" cols="40" rows="5">
  Fille de Josiane Balasko, Marilou Berry fait ses premiers
  pas au cinéma à 8 ans...
</textarea>
```

LE WEB

2.13. Éléments du langage HTML

12- Les formulaires

12.14 Menus de sélection

- La balise `<select>` permet d'ajouter une liste de sélection :
- L'attribut facultatif `size` permet de préciser le nombre de choix apparaissant sur la page Web. Par défaut, vaut 1.
- L'attribut `multiple` `= "multiple"` permet d'autoriser des réponses multiples.
- Les choix du menu sont indiqués à l'aide de la balise `<option>` :
- L'attribut `selected` `= "selected"` permet de spécifier le(s) choix par défaut.
- L'attribut `value` permet de spécifier la valeur associée au choix.



Exemple

```
<select name="age">
  <option value="20">Moins de 20 ans</option>
  <option value="35" selected="selected">21 à 35 ans</option>
  <option value="50">36 à 50 ans</option>
  <option value="51">Plus de 51 ans</option>
</select>
```

LE WEB

2.13. Éléments du langage HTML

12- Les formulaires

12.15 Exemple: form1.html

Enregistrement d'un utilisateur

Nom

Prénom

Sexe Homme : ☐ Femme : ☐

Fonction

Commentaires

BASE DE DONNÉES RELATIONNELLES

SYSTÈME D'INFORMATION

Introduction générale

Enregistrement d'un utilisateur

Nom

Prénom

Sexe

Homme : ☐

Femme : ☐

Fonction

Enseignant ▼

Commentaires

Tapez ici vos
commentaires

Envoyer

Notions

- Une **information** est un renseignement qui accroît la connaissance concernant une personne, un objet ou un événement déterminé.
- L'information peut être :
 - **objective**, quand elle reflète un ensemble de données porteur de sens ;
 - **subjective**, quand elle résulte de l'interprétation d'un ensemble de données.



NOTIONS

■ Exemple

Les diplômes détenus par une personne, son lieu de naissance, la marque d'une voiture, la référence du produit commandé par un client, l'adresse du client, le solde d'un compte... sont des données qui apportent **objectivement** une information.

Toutefois, elles peuvent donner lieu à interprétation pour juger **subjectivement** de la valeur d'un diplôme, de ce que signifie la possession de telle marque de voiture, de l'intérêt d'un client habitant à tel endroit...

NOTIONS

■ Donnée

- Une **donnée** ne devient une information que quand elle trouve son sens par rapport à un référentiel (un contexte, un système de valeurs, un problème à résoudre...).
- Les **données** utilisées dans les organisations se situent dans un référentiel stable. Elles échappent en grande partie à l'interprétation et fournissent alors sans ambiguïté de l'information. Dans les systèmes informatiques, la donnée est par ailleurs la traduction codée d'une information. Les termes « donnée » et « information » sont donc souvent synonymes dans ce contexte.

■ Information

- Une **information** peut-être quantitative (quand elle intervient dans des calculs) ou à l'opposé qualitative, permanente (toujours disponible) ou au contraire temporaire, structurée ou non, formelle (quand elle respecte des règles de présentation ou de diffusion) ou informelle. On peut également distinguer les informations de fonctionnement, d'influence, d'anticipation...

DÉFINITION D'UN SYSTÈME

- Un **Système** est un ensemble d'éléments (matériels ou immatériels) en **interaction dynamique** et organisés en fonction d'un **but donné**.
- Ces éléments transforment des données (entrées) en résultats (sorties)
- **L'entreprise** est alors considérée comme un ensemble d'éléments (des moyens humains, matériels, financiers et techniques) en interrelations.
- Toute organisation humaine (l'État, une famille, ...) peut être perçue comme un système.

SYSTÈME D'INFORMATION

Le Système d'Information (SI) est l'ensemble de méthodes et moyens recueillant, contrôlant, mémorisant et distribuant les informations nécessaires à l'exercice de l'activité de tous les points de l'organisation.

Il joue le rôle de coordinateur entre les éléments de l'entreprise et/ou l'extérieur.

SYSTÈME D'INFORMATION

Les principales fonctions d'un SI sont :

- **Collecter** les informations des éléments du système ou de l'environnement extérieur au système.
- **Mémoriser** les données manipulées par le système.
- **Traiter** les données stockées
- **Transmettre** les informations vers les composantes du système, ainsi vers l'environnement extérieur du système

LES TROIS SYSTÈMES DE L'ENTREPRISE

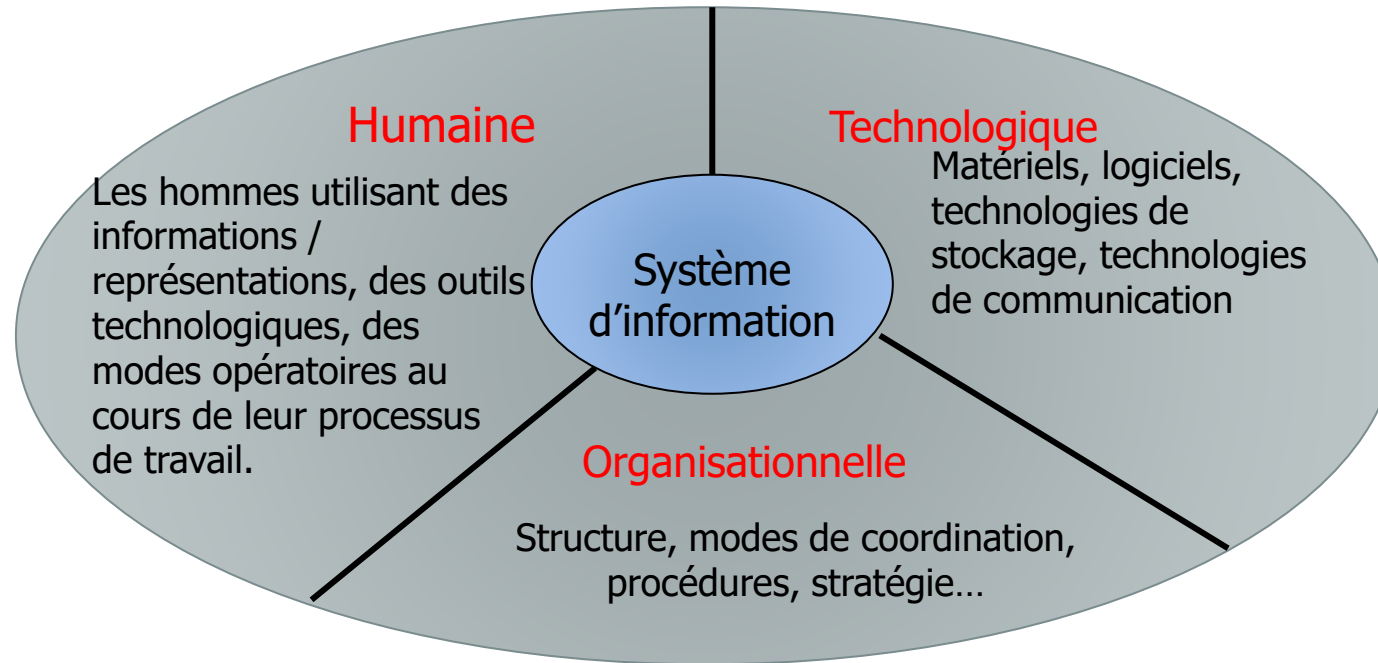
L'**entreprise** ou l'organisation peut être constituée de trois composantes:

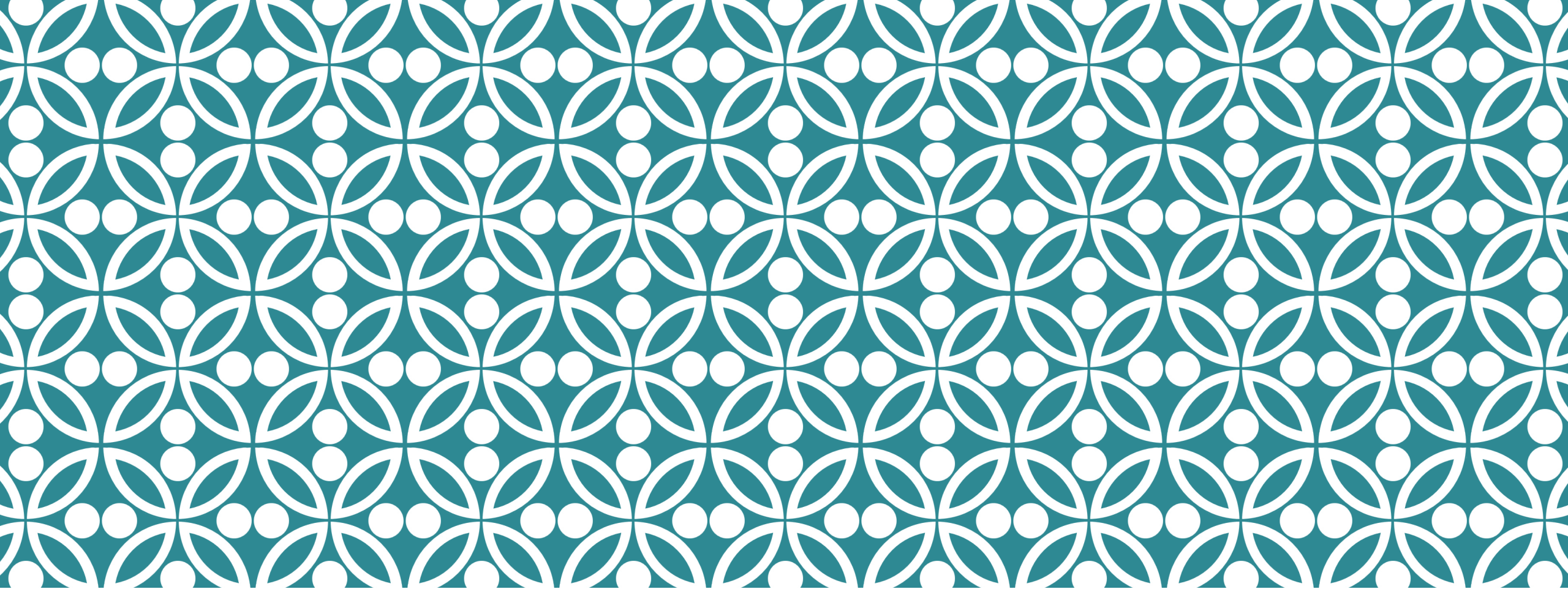
- **le système opérant**
- **le système de pilotage (Décision)**
- **le système d'information.**

➤ Ce sont trois sous-systèmes qui interagissent entre eux

SYSTÈME D'INFORMATION

DIMENSIONS D'UN SYSTÈME D'INFORMATION





CONCEPTION DES SYSTÈMES D'INFORMATION

LA METHODE MERISE

ANALYSE ET CONCEPTION DU SYSTÈME D'INFORMATION

Que faut-il pour analyser, concevoir ...?

On doit :

- Avoir une vision abstraite du fonctionnement,
- Garantir les délais, la pertinence, l'efficacité,
- Faciliter la maintenance
- Prolonger la durée de vie
- ...

Nous avons, donc, besoin :

- De Modèles,
- De Méthodologies

MODÈLE, C'EST QUOI ?

Un **modèle** est une représentation abstraite, d'une partie du monde réel, exprimée dans un langage de représentation.

Ce langage peut être :

- **Formel** : ayant une syntaxe et une sémantique bien définies
- **Semi-formel** : notation graphique formalisée.
- **Informel** : description en langage naturel.

MODÈLE, POURQUOI?

Les principales motivations sont :

- **Comprendre** et analyser la structure et le fonctionnement de l'entreprise;
- **Prévoir** de manière fiable le comportement et les performances des processus opérationnels avant leur implantation;
- **Choisir** les meilleures alternatives d'implantation
- **Identifier** les risques d'implantation à gérer
- **Justifier** les choix d'implantation sur les critères liés aux ressources et aux coûts
- **Bâtir** une vision commune du fonctionnement de l'entreprise et la communiquer facilement au plus grand ensemble possible du personnel.

MÉTHODOLOGIE, POURQUOI?

Formalisation claire et complète du problème informationnel.

Maîtrise de la résolution du problème par l'utilisation de critères objectifs pour évaluer les solutions.

Construction de SI pertinents, complets, cohérents, fiables flexibles et adaptatifs.

Évaluation du SI à tout moment de son cycle de vie.

Faciliter la coopération entre concepteurs, informaticiens gestionnaires, utilisateurs.

Rigueur dans l'élaboration de la solution.

Réduire les coûts et les délais.

MÉTHODOLOGIE, TYPOLOGIE?

Approche cartésienne.

- Orientée traitements.

Approche systémique.

- Orientée données.

Approche Objet.

- Orientée données et traitements.

MÉTHODOLOGIE, TYPOLOGIE?

MERISE : Méthode d'Etude et de Réalisation Informatique pour les Systèmes d'Entreprises (Tardieu, 79).

AXIAL : Analyse et Conception des Systèmes d'Informations Assistés par Logiciel (IBM, 86).

SSADM : Structured Systems Analysis and Design Method (CCTA, 80).

SADT : Structured Analysis and Design Technique (Softech, 73).

RUP : Rational Unified Process (IBM, 2003).

MÉTHODE MERISE

La conception d'un **système d'information** n'est pas évidente, car il faut réfléchir à l'ensemble de l'organisation que l'on doit mettre en place. La phase de conception nécessite **des méthodes** permettant de mettre en place un modèle sur lequel on va s'appuyer.

La **modélisation** consiste à créer une représentation virtuelle d'une réalité de telle façon à faire ressortir les points auxquels on s'intéresse. Ce type de **méthode** est appelé analyse. Il existe plusieurs méthodes **d'analyse**, la méthode la plus utilisée est la méthode **MERISE**.

Le **but** de cette méthode est d'arriver à **concevoir** un **système d'information**.

MÉTHODE MERISE

La méthode **MERISE** (Méthode d'Etude et de Réalisation Informatique pour les Systèmes d'Entreprises)

Apparue en 1979.

C'est une méthode et une démarche :

- Analyse
 - Conception,
 - Réalisation de projets informatiques.
- Basée sur une séparation nette entre les **données** et les **traitements**.

MÉTHODE MERISE

Merise est une **méthode** qui permet de construire un système d'information **automatisé** qui soit efficace, flexible et adapté à l'entreprise.

Merise est une **méthode d'analyse**, de **conception et de développement du système d'information**. Elle vise à recenser la totalité des informations dont un organisme a besoin pour assurer tout ou partie de ses activités fondamentales.

APPROCHE PAR NIVEAUX

Sc	Niveau	Préoccupation	Données	Traitements
	Conceptuel	Quoi?	Modèle Conceptuel de Données (MCD)	Modèle Conceptuel de Traitements (MCT)
	Logique ou Organisationnel	Qui fait quoi? Où?	Modèle Logique de Données (MLD)	Modèle Logique de Traitements (MLT)
	Physique ou Opérationnel	Quand? Avec quels moyens?	Modèle Physique de Données (MPD)	Modèle Opérationnel de Traitements (MOT)

APPROCHE PAR NIVEAUX

- Le niveau conceptuel qui décrit la statique et la dynamique du **système d'information** en se préoccupant uniquement du point de vue du gestionnaire.

Niveau	Préoccupation	Données	Traitements
Conceptuel	Quoi?	Modèle Conceptuel de Données (MCD)	Modèle Conceptuel de Traitements (MCT)

APPROCHE PAR NIVEAUX

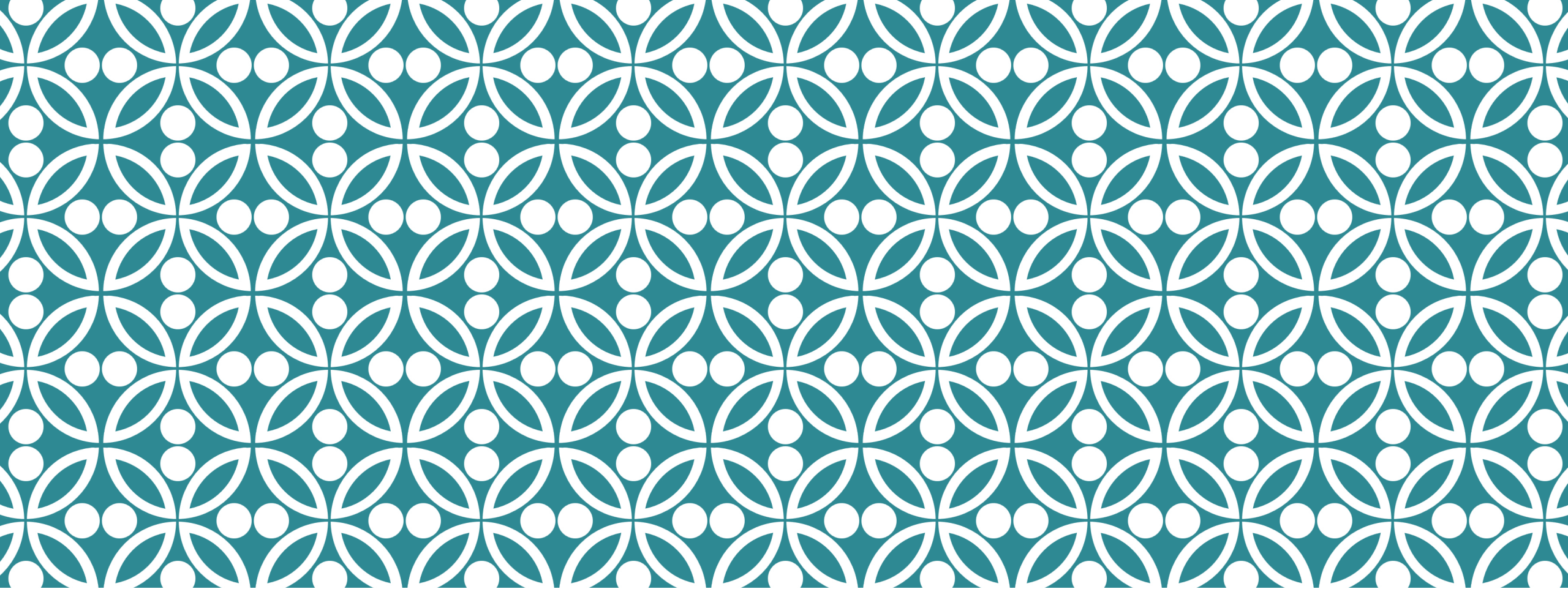
- Le niveau organisationnel décrit la nature des ressources qui sont utilisées pour supporter la description statique et dynamique du **système d'information**. Ces ressources peuvent être humaines et/ou matérielles et logicielles.

Niveau	Préoccupation	Données	Traitements
Logique ou Organisationnel	Qui fait quoi? Où?	Modèle Logique de Données (MLD)	Modèle Logique de Traitements (MLT)

APPROCHE PAR NIVEAUX

- Le **niveau opérationnel** dans lequel on choisit les techniques d'implantation du **système d'information** (données, traitements, ...)

Niveau	Préoccupation	Données	Traitements
Physique ou Opérationnel	Quand? Avec quels moyens?	Modèle Physique de Données (MPD)	Modèle Opérationnel de Traitements (MOT)



NIVEAU CONCEPTUEL

Modèle Conceptuel de Données

MODÈLE CONCEPTUEL DES DONNÉES

PRÉSENTATION

Le **Modèle Conceptuel des Données (MCD)** est une représentation statique du système d'information de l'entreprise qui met en évidence sa sémantique. Il a pour but d'écrire de façon formelle les données qui seront utilisées par le système d'information. Il s'agit donc d'une représentation des données, facilement compréhensible.

Autrement dit: Le **MCD** est une représentation de l'ensemble des données manipulées par l'entreprise ainsi que les relations entre ces données.

CONCEPTS MANIPULÉS

■ Les différents concepts manipulés

■ **Entité**

■ **Relation**

■ **Attribut**

■ **Notion d'occurrence**

■ **Occurrence d'une propriété**

■ **Occurrence d'une entité**

■ **Identifiant**

MODÈLE CONCEPTUEL DES DONNÉES

ENTITÉ

Une **entité** ou **individu** est un **objet** autonome (concret ou abstrait), pourvu d'une existence propre et ayant une signification dans l'univers de **l'entreprise**.

Exemple :

- *Client, Fournisseur, Service, ...*
- *Etudiant, Professeur, Cours, ...*
- *Produit, Magasin, Fournisseur,...*

MODÈLE CONCEPTUEL DES DONNÉES

RELATION

C'est une représentation d'une association entre entités.

La **relation** est dépourvue d'une existence propre. Elle n'existe qu'à travers les entités qu'elle relie.

Exemple :

Un Client possède un Compte Bancaire

Un Étudiant est inscrit à un Cours

Un Département a plusieurs Employés

MODÈLE CONCEPTUEL DES DONNÉES

ATTRIBUT

Une **donnée** élémentaire qui caractérise une **entité** ou une relation.

Exemple :

- L'entité **Client** est caractérisée par :
 - *Prenom_Client*
 - *Nom_Client*
 - *Adresse_Client*
- L'entité **Produit** est caractérisée par :
 - *N°Ref*
 - *Désignation*
 - *Prix*
- La relation **Commande** peut être caractérisée par :
 - *Qte*

MODÈLE CONCEPTUEL DES DONNÉES

Occurrence d'une propriété

- Ensemble des **valeurs** que peut prendre une **propriété**.
- **Exemple :**
 - *Mourad, Hind, Abdelali et Sami* sont des occurrences de la propriété *Prénom_Client*.

Occurrence d'une entité

- Un ensemble d'occurrences de chacune des propriétés de l'entité.
- **Exemple :**
 - L'ensemble *(Mohammed-Alamine, El Houssaini, El Jadida)* constitue une occurrence de l'individu *Client*.

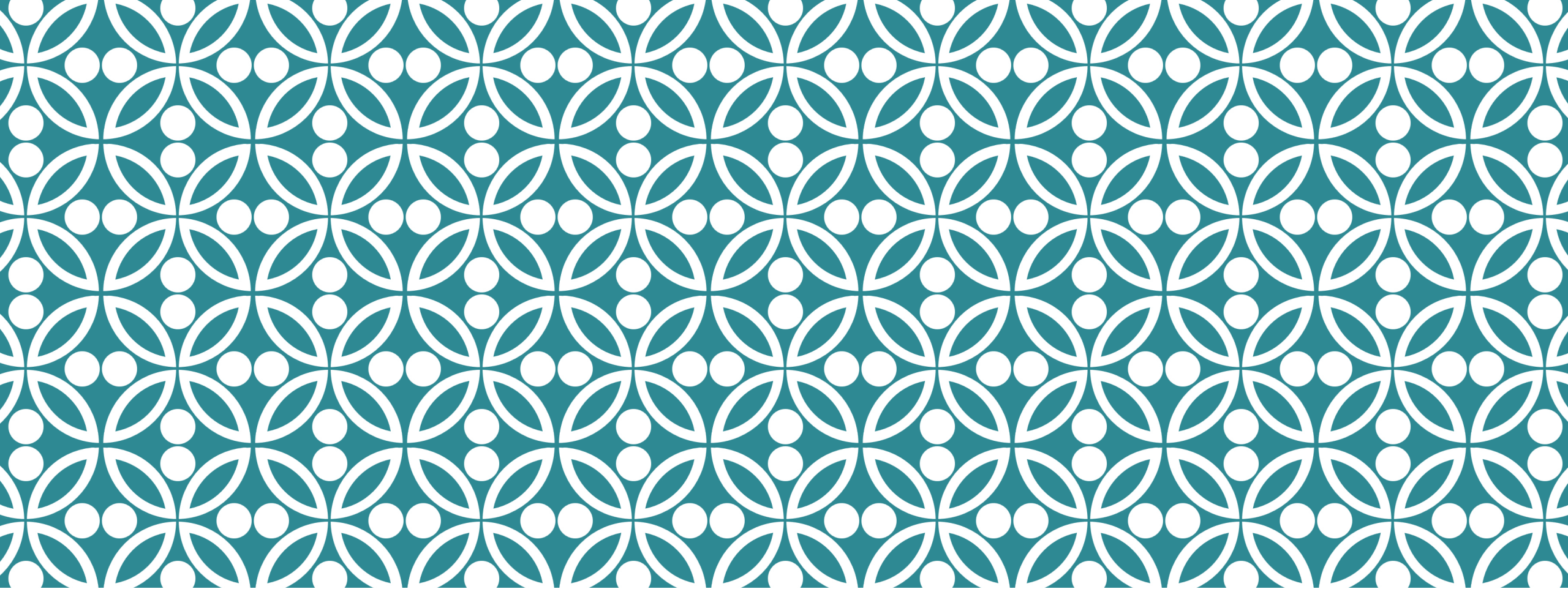
MODÈLE CONCEPTUEL DES DONNÉES

IDENTIFIANT

Un ou plusieurs attributs permettant de distinguer de manière unique et sans ambiguïté les différentes occurrences d'une entité.

Exemple :

- *N°Client* est l'identifiant de l'entité *Client*, car deux clients ne peuvent pas avoir le même *numéro*.
- *NBC* est l'identifiant de l'entité *Détails_commandes*.



MODÈLE CONCEPTUEL DE DONNÉES

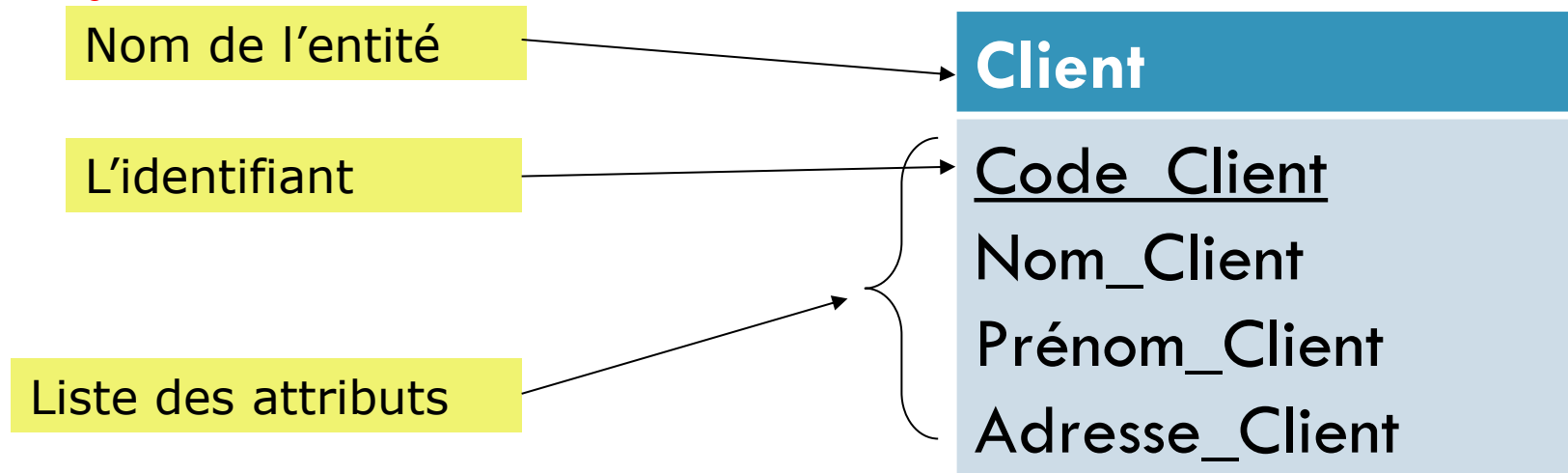
Formalisme graphique

MODÈLE CONCEPTUEL DES DONNÉES

REPRÉSENTATION DES ENTITÉS

Représentées par un rectangle divisé en deux parties.

- La partie **supérieure** contient le nom de l'entité, alors que **l'autre partie** contient la liste de ses attributs.
- L'identifiant** est **souligné**

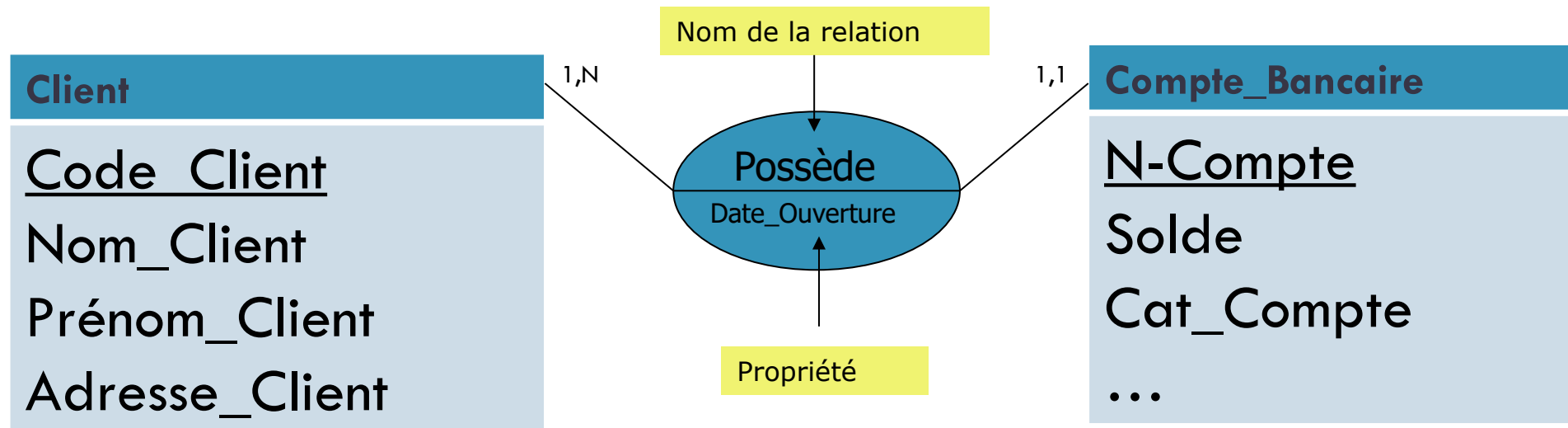


MODÈLE CONCEPTUEL DES DONNÉES

REPRÉSENTATION DES RELATIONS

Formalisée par un rectangle aux angles arrondis.

- Comme pour le cas d'une **entité**, une **relation** contient deux volets.
- Le volet **supérieur** contient le nom de la relation, alors que la **partie inférieure** contient les **attributs** de la relation.

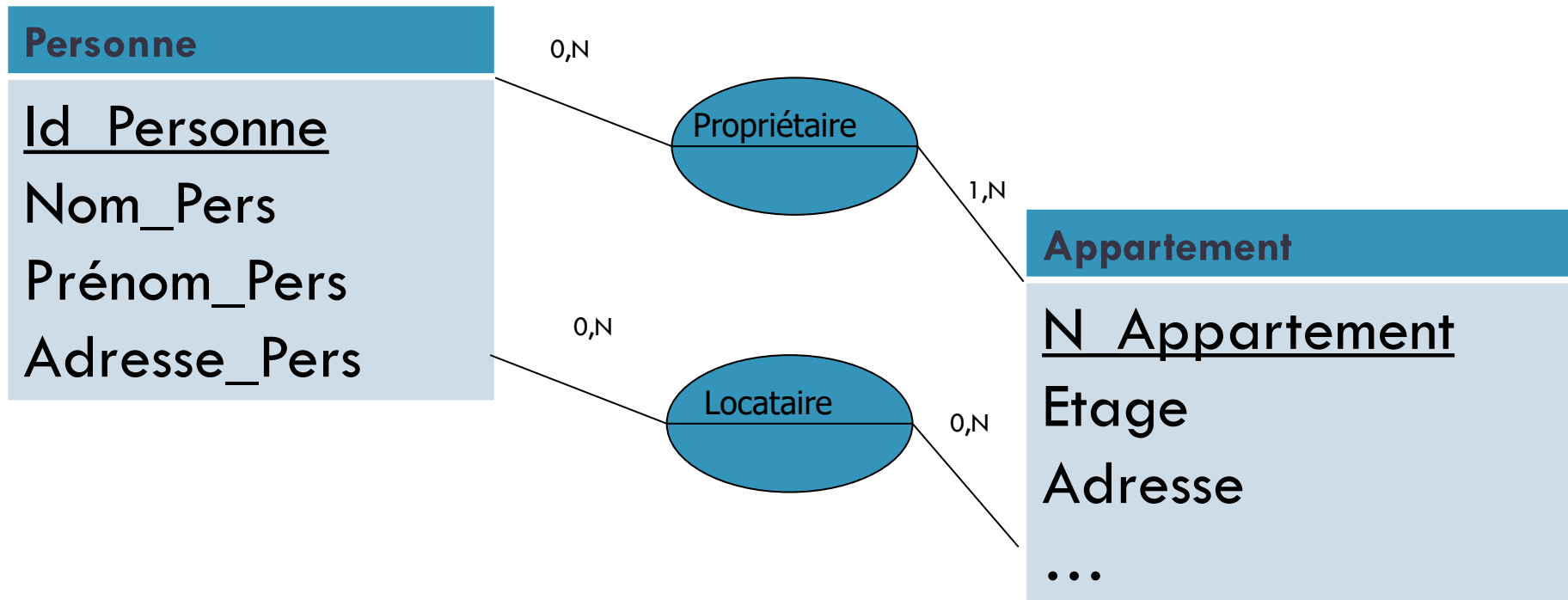


MODÈLE CONCEPTUEL DES DONNÉES

REPRÉSENTATION DES RELATIONS

Remarque :

- Il peut y avoir plusieurs relations entre deux entités



MODÈLE CONCEPTUEL DES DONNÉES

DIMENSION D'UNE RELATION

La **dimension** d'une **relation (association)** indique le **nombre** d'entités participant à l'association.

Une relation peut être

- Unaire
- Binaire
- Ternaire
- n-aire

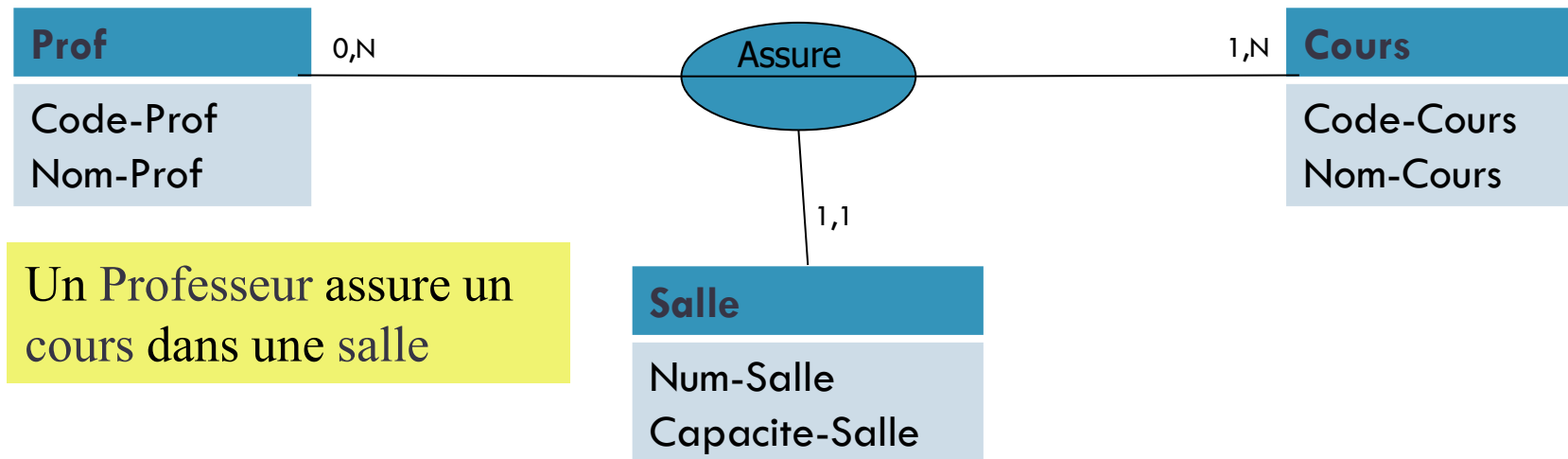
Les dimensions les plus courantes sont 2 (association **binaire**) et 3 (association **ternaire**)

MODÈLE CONCEPTUEL DES DONNÉES

DIMENSION D'UNE RELATION

L'association binaire exprime la présence de liens sémantiques entre les occurrences d'une entité A et les occurrences d'une entité B.

L'association ternaire exprime la présence de liens sémantiques entre les occurrences de 3 entités.



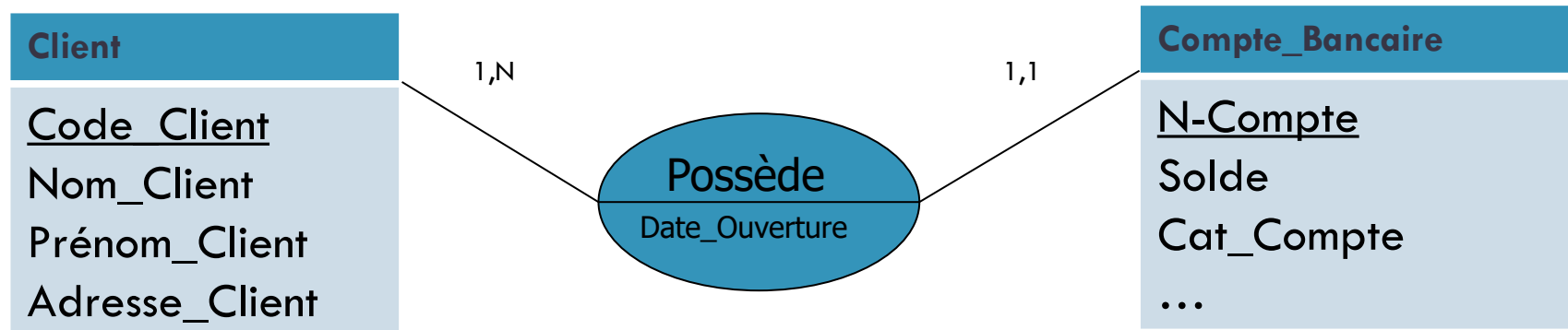
MODÈLE CONCEPTUEL DES DONNÉES

CARDINALITÉS

Mesurent la participation **minimale** et **maximale** de l'entité à la relation.

Les cardinalités sont notées sous la forme de (X, Y) où :

- **X** : cardinalité **minimale**
- **Y** : cardinalité **maximale**



- Un **client** possède au moins un (**1**) **compte** bancaire, comme il peut en posséder **plusieurs** (**N**). D'où les cardinalités (**1, N**).
- Un **compte** bancaire appartient à **un client et un seul**, d'où les cardinalités (**1, 1**).

MODÈLE CONCEPTUEL DES DONNÉES

DIMENSION D'UNE RELATION

Les différents types de fonctionnalités d'une **association** binaire définie entre deux **entités** sont les suivants :

- $(0,1)$: une occurrence de l'entité ne participe jamais plus d'une fois à la relation
- $(1,1)$: une occurrence de l'entité participe toujours une et une seule fois à la relation
- $(1, N)$: une occurrence de l'entité participe toujours au moins une fois à la relation
- $(0, N)$: une occurrence de l'entité peut participer plusieurs fois à la relation

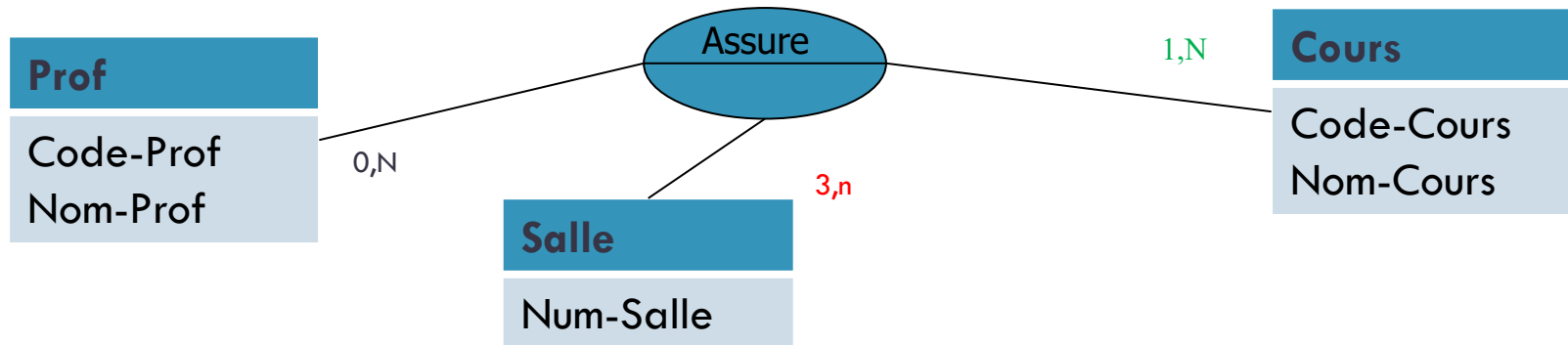
MODÈLE CONCEPTUEL DES DONNÉES

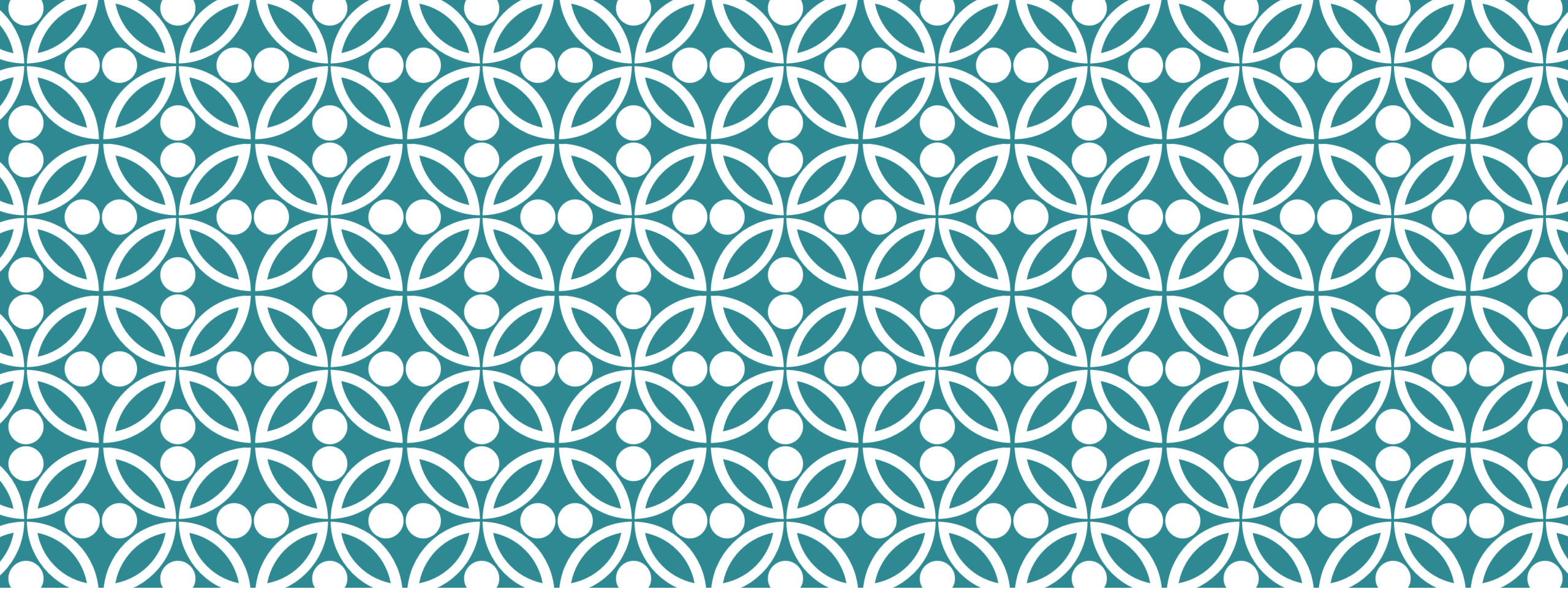
RÈGLES DE GESTION

Les règles de gestion (RG) du MCD traduisent les contraintes qui **doivent être respectées** par le modèle.

Exemple: Dans le MCD d'une école, les règles de gestion peuvent être les suivantes:

- **RG1** : **Tout professeur enseigne en pratique au moins une matière**, mais certains d'entre eux peuvent être dispensés d'enseignement en raison de leurs travaux de recherche.
- **RG2** : Toute matière est enseignée dans au moins une classe.
- **RG3** : Toute classe a au moins trois enseignements.





MODÈLE CONCEPTUEL DE DONNÉES

Dépendance fonctionnelle

MODÈLE CONCEPTUEL DES DONNÉES

DÉPENDANCE FONCTIONNELLE ENTRE PROPRIÉTÉS

Deux propriétés **a** et **b** sont reliées par une dépendance fonctionnelle, notée

a(source) – df → b(but), si la connaissance de la **valeur de a** détermine de manière unique **la valeur de b**.

■ **Exemple :**

Code_Client – df → Nom_Client

- La connaissance du code client détermine **une et une seule valeur** du nom client.
- La **réciproque est fausse**, le nom client ne permet pas de déterminer son code, **car** plusieurs clients peuvent avoir le même nom.

MODÈLE CONCEPTUEL DES DONNÉES

DÉPENDANCE FONCTIONNELLE ENTRE PROPRIÉTÉS

Remarque :

- La dépendance fonctionnelle peut porter sur la concaténation de plusieurs propriétés.

Exemple :

$A = (NBC + Ref) - df \rightarrow Qte \text{ commandée.}$

- La **référence seule** ne suffit pas à déterminer la quantité commandée.
- Le **n° de bon de commande** ne suffit pas non plus
- En revanche, la connaissance du **n° de bon de commande et de la référence du produit** détermine celle de la quantité commandée.

MODÈLE CONCEPTUEL DES DONNÉES

DÉPENDANCE FONCTIONNELLE ÉLÉMENTAIRE

On dit qu'il y a dépendance fonctionnelle élémentaire entre les propriétés a et b et on la note

$a \rightarrow b$ si : $a - df \rightarrow b$ aucune partie de a ne peut déterminer b .

Exemple :

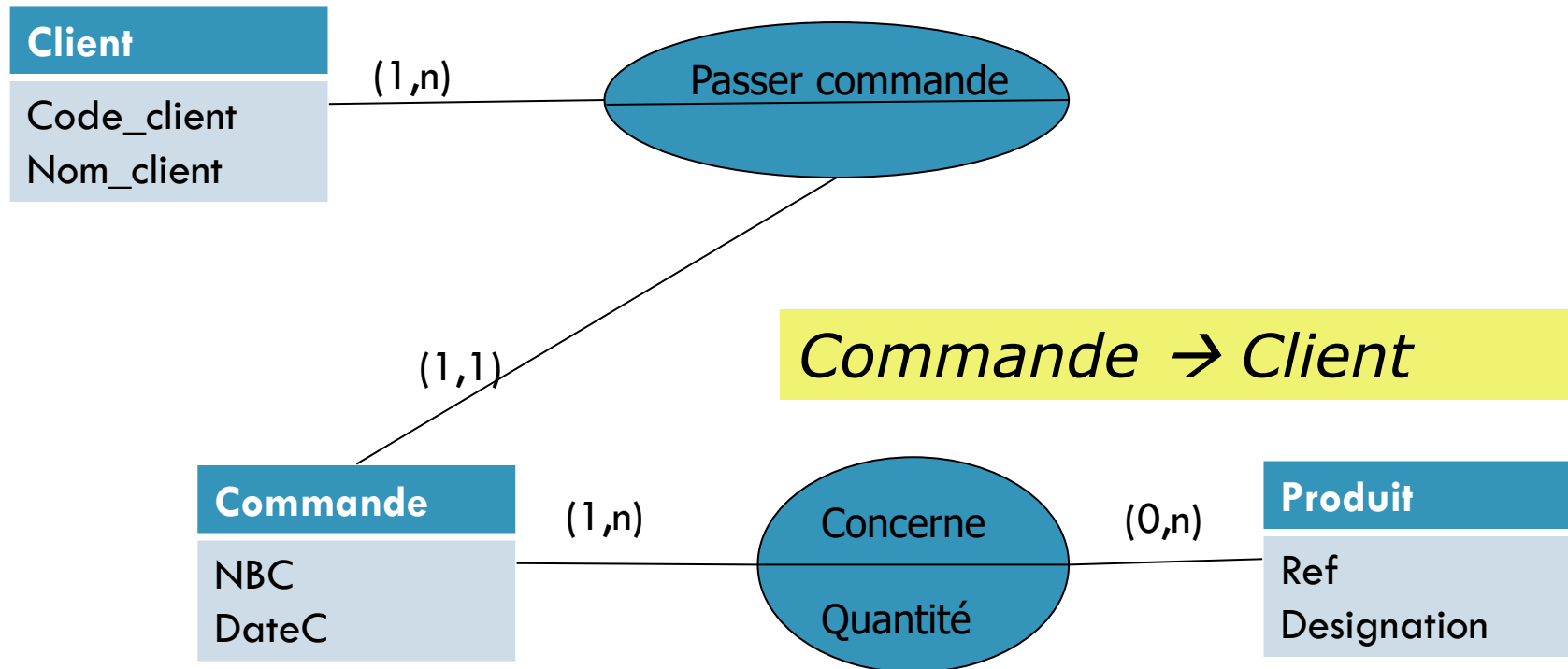
▪ **Code_client + Nom_Client – df → Adresse_Client**

mais n'est pas élémentaire puisque la connaissance du Code_Client suffit de déterminer l'adresse du client.

MODÈLE CONCEPTUEL DES DONNÉES

DÉPENDANCE FONCTIONNELLE ENTRE ENTITÉS

On dit qu'il existe une dépendance fonctionnelle entre deux entités E1 et E2 et on la note $E1 \rightarrow E2$, si toute occurrence de E1 détermine une et une seule occurrence de E2.



MODÈLE CONCEPTUEL DES DONNÉES

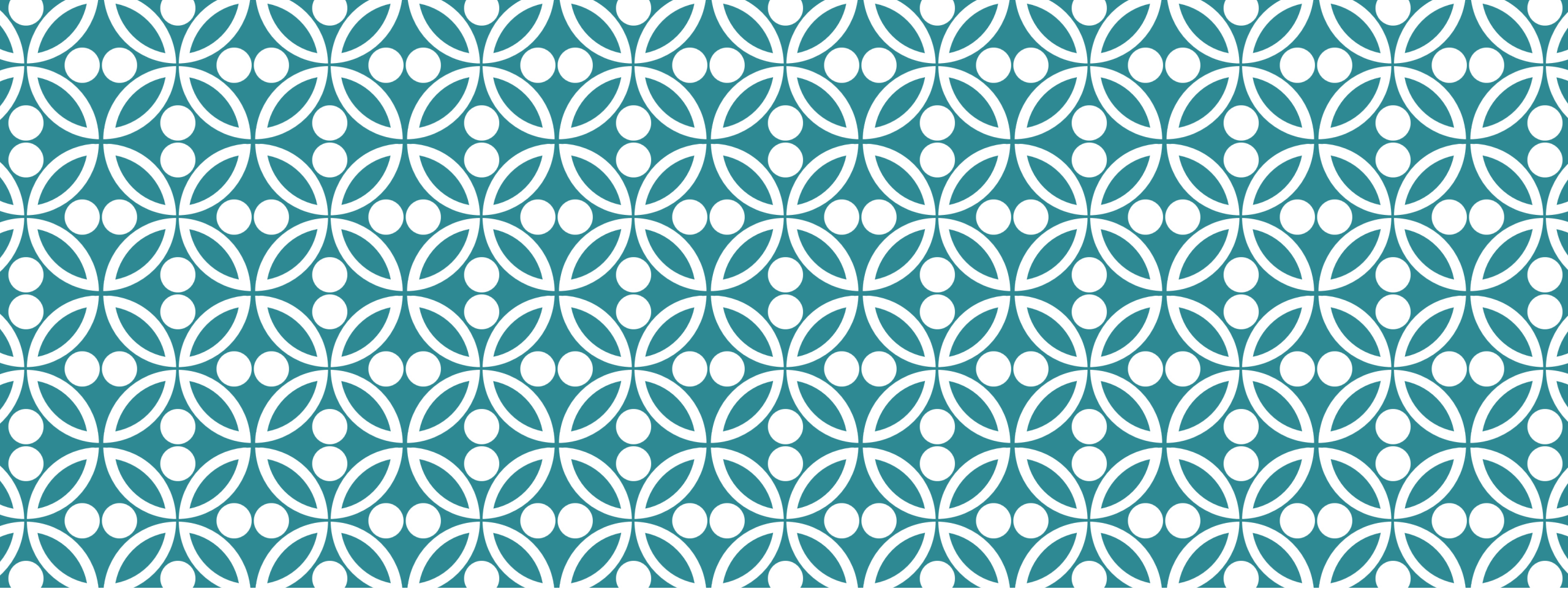
DÉPENDANCE FONCTIONNELLE ÉLÉMENTAIRE

Remarque :

- La cardinalité **maximale 1** correspond toujours à une dépendance fonctionnelle entre entités.
- On peut assimiler les dépendances fonctionnelles entre entités aux dépendances fonctionnelles entre les identifiants de ces entités.

Exemple :

- Commande \rightarrow Client
Est assimilable à
- NBC \rightarrow Code_Client



LA NORMALISATION DU MODÈLE RELATIONNEL

Les formes normales

NORMALISATION DES ENTITÉS D'UN MCD

L'**objectif** de la normalisation est de construire un schéma de base de données cohérent.

Un mauvais schéma logique peut conduire à un certain nombre d'anomalies pendant la phase d'exploitation de la base de donnée.

Pour qu'un modèle relationnel soit **normalisé**, il faut qu'il **respecte certaines contraintes** appelées les formes normales. Les formes normales s'appuient sur les dépendances fonctionnelles entre attributs.

Les entités d'un MCD doivent vérifier quelques règles, appelées **formes normales**.

NORMALISATION DES ENTITÉS

PREMIÈRE FORME NORMALE (1FN)

Une entité est en première forme normale si:

- toutes les propriétés sont **élémentaires**
- il existe au moins un **identifiant** caractérisant chaque **occurrence** de l'entité représentée.

Exemple

Client

Nom_Client

Prenom_Client

Adresse_Client

Client n'est pas en 1FN, car ...

Car :

- pas **d'identifiant** (plusieurs clients peuvent avoir le même nom).
- Adresse_Client est sans doute la **concaténation** de RUE et Ville

NORMALISATION DES ENTITÉS

DEUXIÈME FORME NORMALE (2FN)

Une entité est en deuxième forme normale si :

- elle est en 1FN
- toute propriété d'une entité doit dépendre de l'**identifiant** par une dépendance fonctionnelle **élémentaire**.

Exemple

Ligne_Commande

NBC
REFP
DESIGNATIONP
QTEC

L'**identifiant** est la concaténation de NBC + Ref
mais la dépendance fonctionnelle NBC + Ref – df → Désignation n'est pas élémentaire puisque Ref → Désignation.

NORMALISATION DES ENTITÉS

TROISIÈME FORME NORMALE (3FN)

Pour qu'une entité soit en troisième forme normale il faut que :

- elle soit en deuxième forme normale
- toute propriété doit dépendre de l'identifiant par une dépendance fonctionnelle élémentaire directe.

Exemple

Client
Code_Client
Nom_Client
Code_CategorieC
Nom_CategorieC

Cette entité n'est pas en 3FN car la dépendance fonctionnelle:

Code_Client → Nom_Catégorie n'est pas directe du fait de la transitivité :

Code_Client → Code_Catégorie → Nom_Catégorie.

NORMALISATION DES ENTITÉS

FORME NORMALE DE BOYCE-CODD (FNBC)

Une entité est en FNBC si :

- elle est en 3FN
- une propriété appartenant à l'**identifiant** ne doit pas dépendre d'une autre propriété.

Exemple : Si on considère qu'un moniteur entraîne un seul sport et que chaque étudiant n'a qu'un seul moniteur par sport

Entraînement

Code_Etudiant
code_Sport
Code_Moniteur

Code-Etud, Code_Sport → Code_Moniteur
Code_Moniteur → Code_Sport
n'est pas en FNBC.

NORMALISATION DES ENTITÉS - EXEMPLE

Employé: description d'un employé travaillant sur un projet d'un laboratoire.

Employé (NumEmp, NumLab, NumProj, NomEmp, NomProj, Rue, ville)

avec les dépendances fonctionnelles suivantes:

$(\text{NumEmp}, \text{NumLab}) \rightarrow \text{NumProj}, \text{NomProj}, \text{NomEmp}$

$\text{NumEmp} \rightarrow \text{NomEmp}$

$\text{NumEmp} \rightarrow \text{rue}$

$\text{NumEmp} \rightarrow \text{ville}$

$\text{NumProj} \rightarrow \text{NomProj}$

a- identifier les redondances éventuelles dans sa population,

b- établir le (un) graphe minimum de ses dépendances,

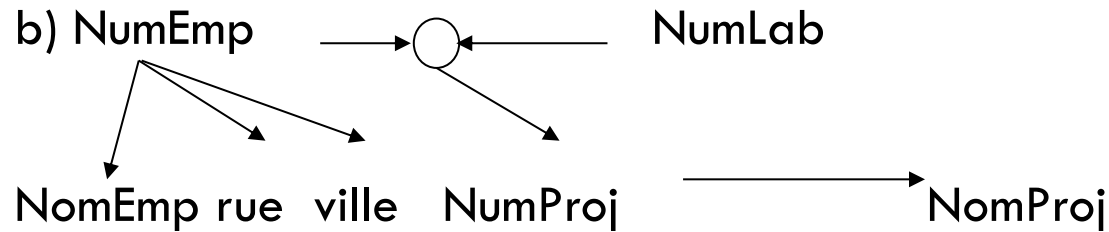
c- définir son (ses) identifiant(s),

d- définir sa forme normale et la justifier,

e- si nécessaire, proposer une décomposition optimale.

NORMALISATION DES ENTITÉS - EXEMPLE

a) Il y a des redondances pour les attributs NomEmp et NomProj.



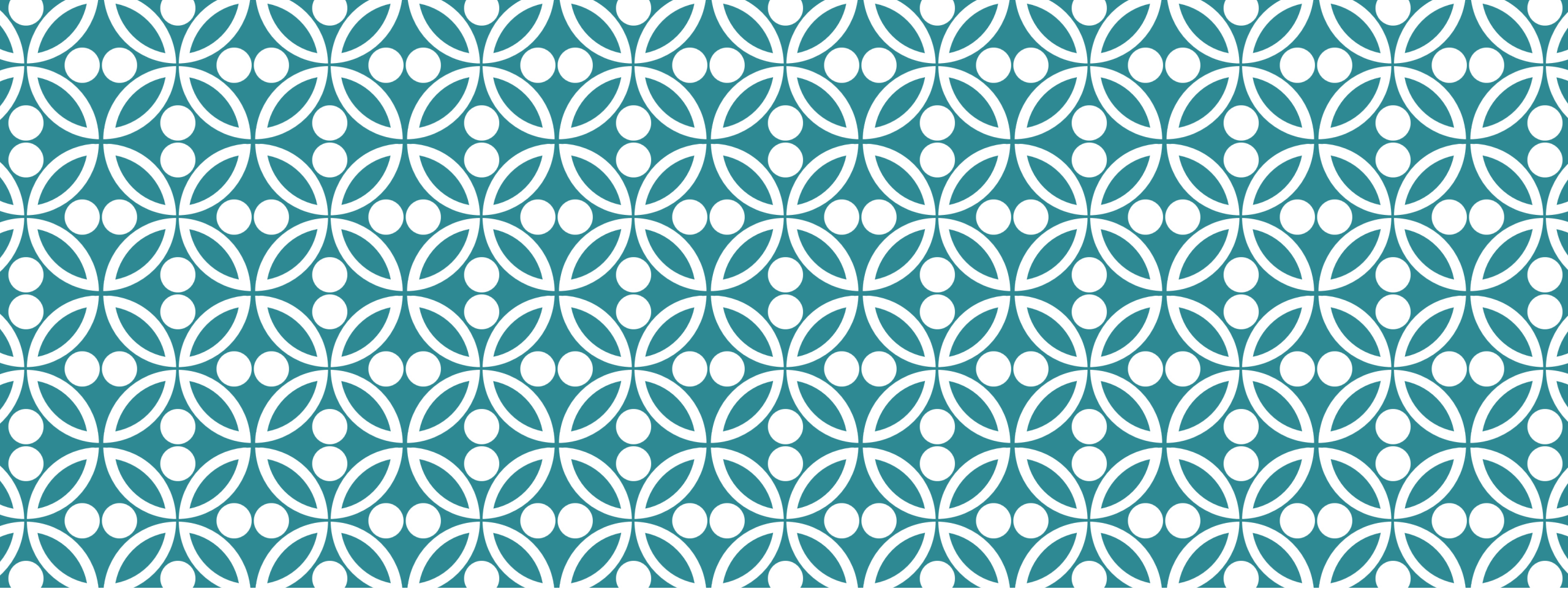
c) L'identifiant est composé de (NumEmp + NumLab).

d) La relation est en première forme normale, mais pas en seconde forme normale, car les attributs NomEmp, rue et ville dépendent d'une partie d'identifiant (et non pas de l'identifiant entier).

e) La relation peut être décomposée en:

- Employé (N°Emp, NomEmp, adresse)
- Projet (N°Proj, NomProj)
- ELP (N°Emp, N°Lab, N°Proj)

Ces relations sont en 1FN, 2FN et 3FN forme normale.



MODÈLE CONCEPTUEL DE DONNÉES

Construction du MCD

MODÈLE CONCEPTUEL DES DONNÉES

OBJECTIF

Nous allons comprendre la méthode de construction du **Modèle Conceptuel de Données** à l'aide d'un exemple (spécification données) :

Gestion des commandes et livraison d'une entreprise de ventes d'articles.

Exemple de document : Bon de commande

des commandes et livraison d'une entreprise de ventes d'articles.				
N° Bon	Date		
Nom Client	Code Client		
e de document : Bon de commande					
Représentant		
Référence		Désignation	Prix	Quantité	Montant
.....	
			THT	
			TVA	
			TTC	

MODÈLE CONCEPTUEL DES DONNÉES

OBJECTIF

La méthode adoptée :

- Recueil des informations
- Dictionnaire de données
- Dictionnaire de données épuré
- Graphe de dépendances fonctionnelles
- Modèle Conceptuel de Données
- Normalisation du MCD.

MODÈLE CONCEPTUEL DES DONNÉES

RECUEIL DES INFORMATIONS

Recueillir toutes les informations manipulées :

- Interviews des différents postes de travail
- Documents utilisés
- Connaissance du problème à étudier

MODÈLE CONCEPTUEL DES DONNÉES

RECUEIL DES INFORMATIONS

Expliciter clairement les règles de gestion:

- RG1 : un client peut passer **une** ou **plusieurs** commandes ou **aucune** commande.
- RG2 : une commande peut concerner **un** ou **plusieurs** produits.
- RG3 : le stock ne concerne que les produits conçus et fabriqués sur commande.
- RG4 : une commande est passée à un représentant qui n'est pas toujours le même pour un client donné.

MODÈLE CONCEPTUEL DES DONNÉES

DICTIONNAIRE DE DONNÉES

On établit le dictionnaire de données à partir des informations contenues dans les documents et fichiers recueillis.

Nom	Signification	Type	Longueur	Nature	Règles
NBL	N° de Bon de livraison	N ¹	4	E ⁴	
Date	Date commande	D ³	8	E	jj/mm/aa
NBL	N° de Bon de livraison	N	4	E	
Date	Date livraison	D	8	E	
Code	Code client	N	10	E	
N°Cli	N° client	N	10	E	
Adresse	Adresse client	AN ²	60	CO ⁶	
Montant	Montant	N	8	CA ⁵	PU*Qte
THT	Total hors taxes	N	12	CA	Somme
TVA	TVA	N	8	E	
TTC	Total TTC	N	12	CA	Somme

1. N: Numérique
2. AN : Alphanumérique
3. D : Date
4. E: Élémentaire
5. CA: Calculer
6. CO: concaténer

MODÈLE CONCEPTUEL DES DONNÉES

DICTIONNAIRE DE DONNÉES - EPURÉ

Éliminer toutes les informations inutiles.

Ces informations peuvent être :

- Informations calculées
- Informations concaténées
- Synonymes

▪ Polysèmes

Informations calculées

- Montants, THT, TVA et TTC

Informations concaténées

- Adresse = Rue + Ville

Exemple :

Synonymes

- Code client et N° client

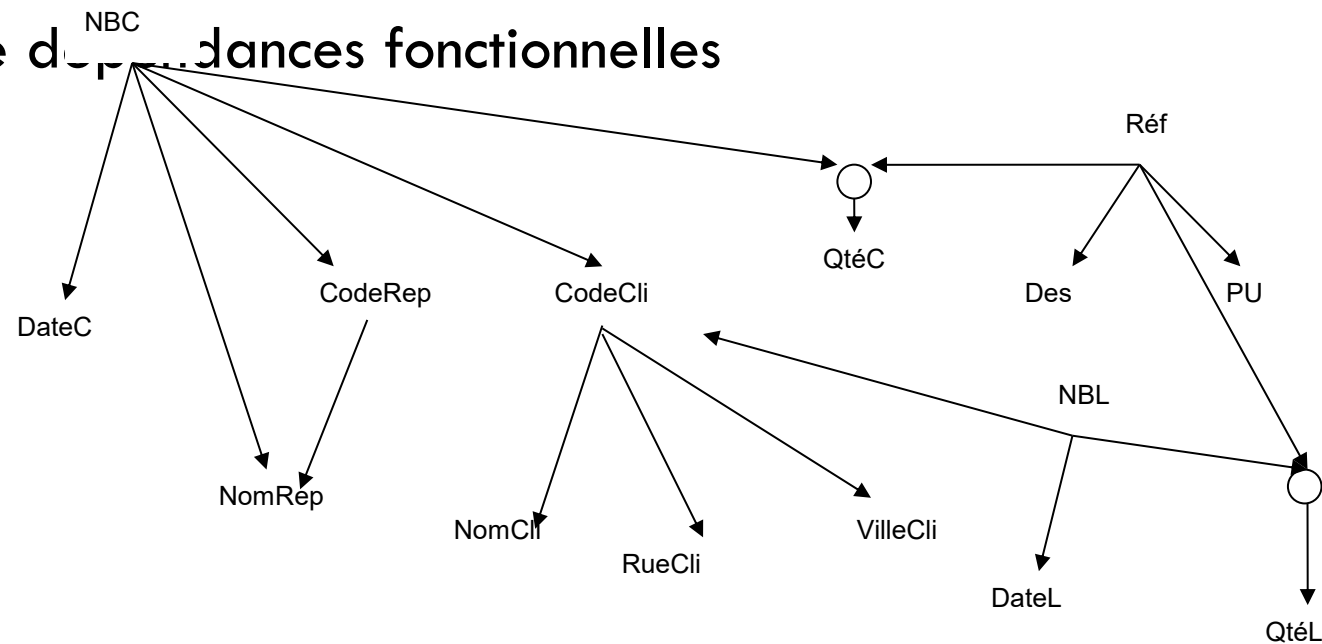
Polysèmes

- Qté = QtéC1 et QtéC2
- Date = DateC1 et DateC2

MODÈLE CONCEPTUEL DES DONNÉES

GRAPHE DE DÉPENDANCES FONCTIONNELLES

Graphe de dépendances fonctionnelles



MODÈLE CONCEPTUEL DES DONNÉES

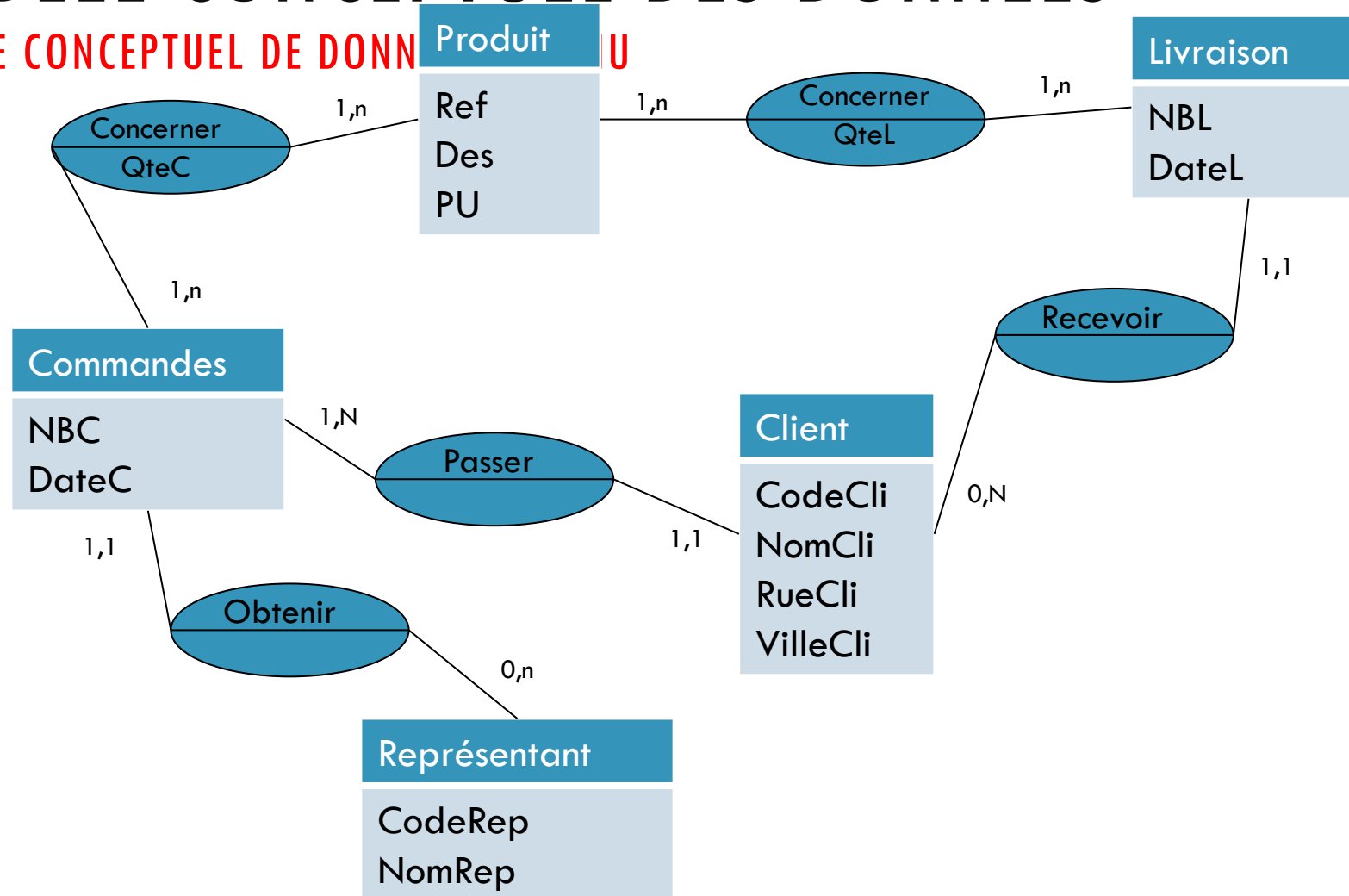
CONSTRUCTION DU MCD

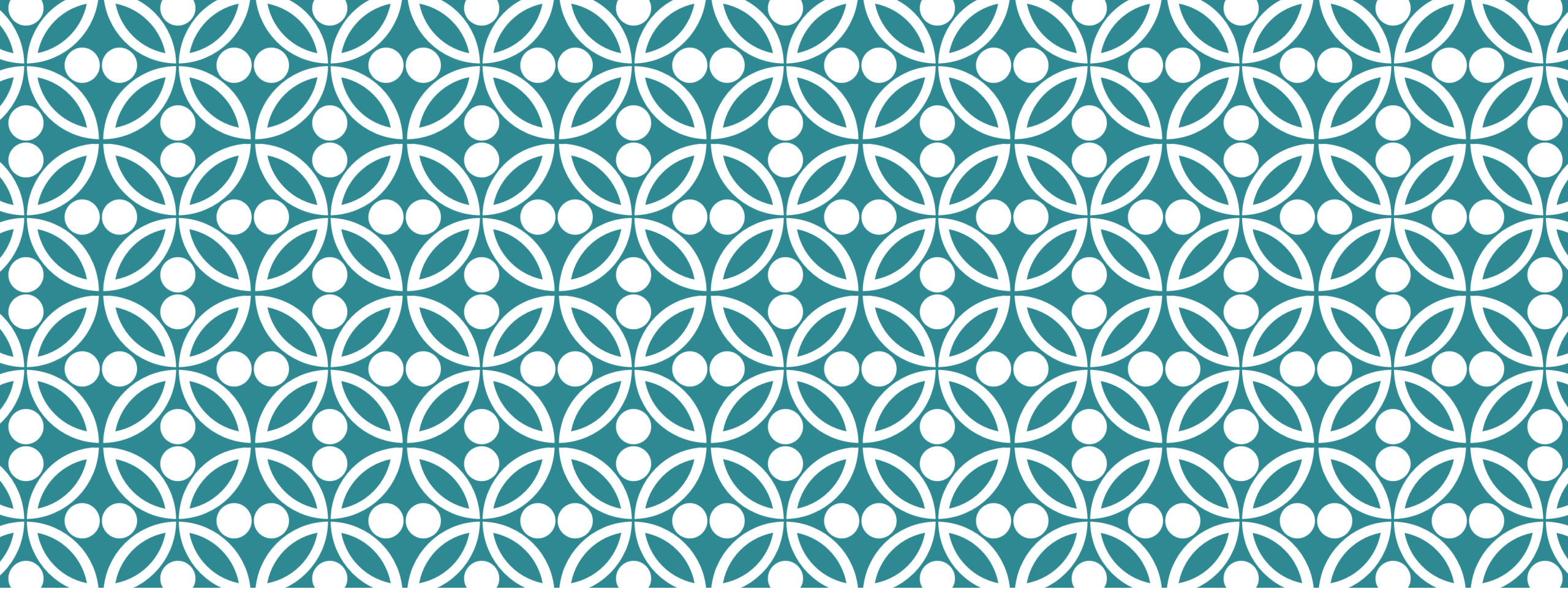
Dégager le MCD :

- Les feuilles issues d'un seul nœud constituent des entités. Les nœuds dont elles sont issues sont des identifiants
- Les arcs sont des relations entre entités
- Les feuilles issues de plusieurs nœuds sont des propriétés des relations
- Les règles de gestion doivent permettre de trouver des cardinalités des entités par rapport aux relations.

MODÈLE CONCEPTUEL DES DONNÉES

MODÈLE CONCEPTUEL DE DONN





Modèle Logique de Données (MLD)

NIVEAU LOGIQUE

MODÈLE LOGIQUE DES DONNÉES

PRÉSENTATION

Le **Modèle Conceptuel de Données (MCD)** a permis de représenter le plus fidèlement possible les réalités de l'univers à informatiser. Mais cette représentation ne peut pas être directement manipulée et acceptée par un système informatique. Il est donc **nécessaire de passer** du **niveau conceptuel** à un **second niveau** plus proche des capacités des systèmes informatiques.

Ce niveau, appelé **niveau logique**, consiste à choisir l'un des trois modèles suivants :

- modèle hiérarchique (années 80),
- modèle réseau,
- ou **modèle relationnel**

MODÈLE LOGIQUE DES DONNÉES

PRÉSENTATION

Le **MCD** décrit les données indépendamment du logiciel utilisé (concepts)

Le Modèle Logique des Données:

- représente les données du **MCD** en tenant compte du logiciel de gestion choisi.
- traduit le **MCD** en formalisme compréhensible par la machine.

Le logiciel de gestion utilisé peut être:

- Logiciel de gestion de fichiers (SGF)
- Logiciel de gestion de BD (**SGBD**) :
 - SGBD Réseau
 - SGBD Relationnel
 - ...

MODÈLE RELATIONNEL

RAPPELS

Inventé par Edgar Frank Codd en 1970

Utilise les **relations** pour représenter les **données**

Manipule plusieurs concepts :

- Tables ou Relation
- **Clé primaire**
- **Clé étrangère**
- ...

Relation ou table :

- Sous ensemble du produit cartésien de plusieurs ensembles (Domaines)
- Représentée par un tableau constitué:
 - d'un Nombre fixe de **colonnes** (champs)
 - d'un Nombre arbitraire de **lignes** (enregistrements)

MODÈLE RELATIONNEL

RAPPELS

Clé primaire (Primary Key)

- Chaque table doit avoir une **clé primaire**
- La clé **identifie** de manière unique et sans ambiguïté les **enregistrements** d'une table
- La clé peut être concaténée

Exemples

- Clients (Code, Nom, ...)
- Détails_Cdes (NBC, Ref, QtéC)

Clé étrangère

- Un simple champ qui joue le **rôle** de clé primaire dans une autre table
- Permet d'établir des relations entre tables (utilisée dans les jointures)

Exemple

- Clients (Code, Nom, ...)
- Commandes (NBC, DateC, Code#)

RÈGLES DE PASSAGE

PASSAGE DU MCD AU MLD RELATIONNEL

Ces **règles** sont de type algorithmique et peuvent être mises en œuvre par des outils de **génie logiciel**, Par exemple *PowerDesigner* (anciennement *PowerAMC*).

La traduction des concepts de base du modèle conceptuel est régie par les trois règles:

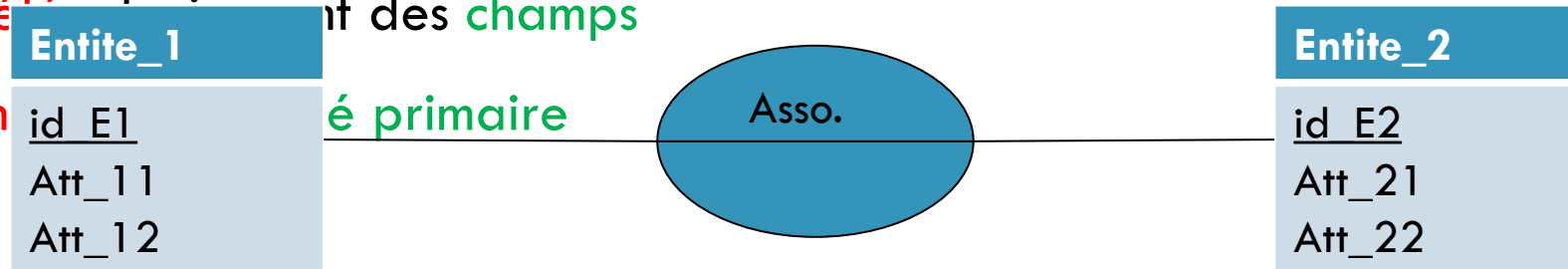
PASSAGE DU MCD AU MLD

RÈGLE 1

Chaque **entité** du MCD se transforme en une **table** du MLD

Les **propriétés** deviennent des **champs**

L'**identifiant** est **primaire**



Entite_1(id_E1, Att_11, Att_12)

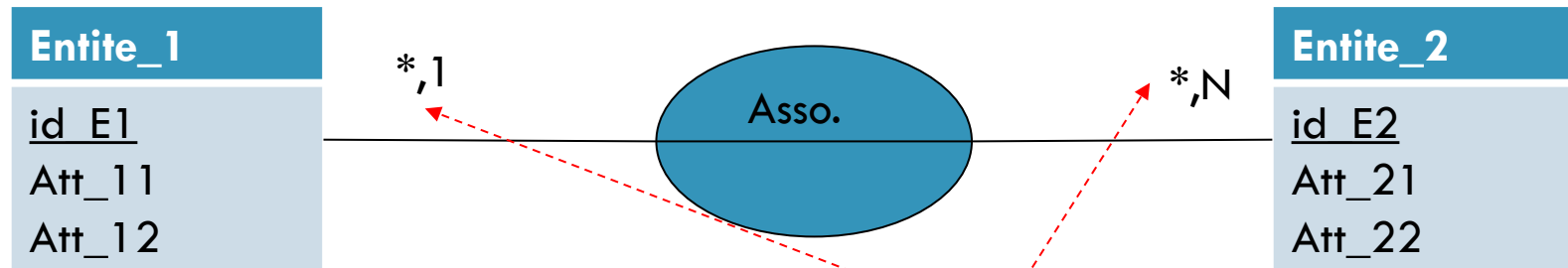
Entite_2(id_E2, Att_21, Att_22)

PASSAGE DU MCD AU MLD

RÈGLE 2

Si la **relation** est de type $(*,1)-(*,N)$, alors:

- La **relation disparaît**
- L'**identifiant** de l'entité de cardinalité $(*,N)$ devient **clé étrangère** de l'entité de cardinalité $(*,1)$
- Les **propriétés** de la relation deviennent des **champs** de l'entité de cardinalité $(*,1)$



Entite_1(id_E1, Att_11, Att_12, id_E2#)

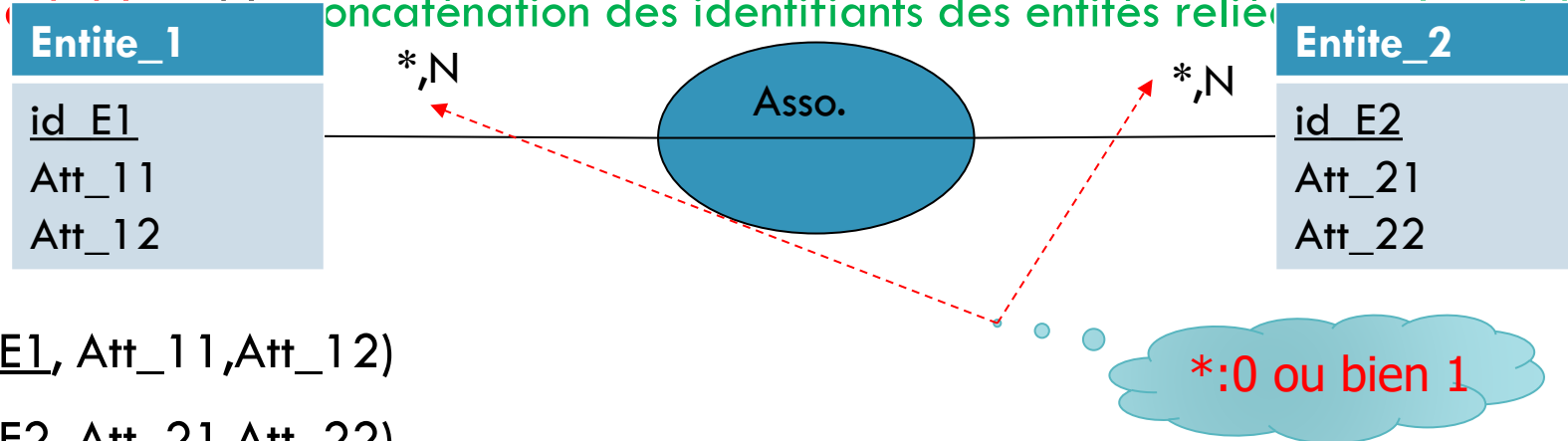
Entite_2(id_E2, Att_21, Att_22)

PASSAGE DU MCD AU MLD

RÈGLE 3

Si la relation est de type $(*,N)-(*,N)$, alors:

- La **relation** devient une **table**
- Les **propriétés** de la relation deviennent des **champs** de la table
- La **clé de la relation** est la **concaténation des identifiants des entités reliées**

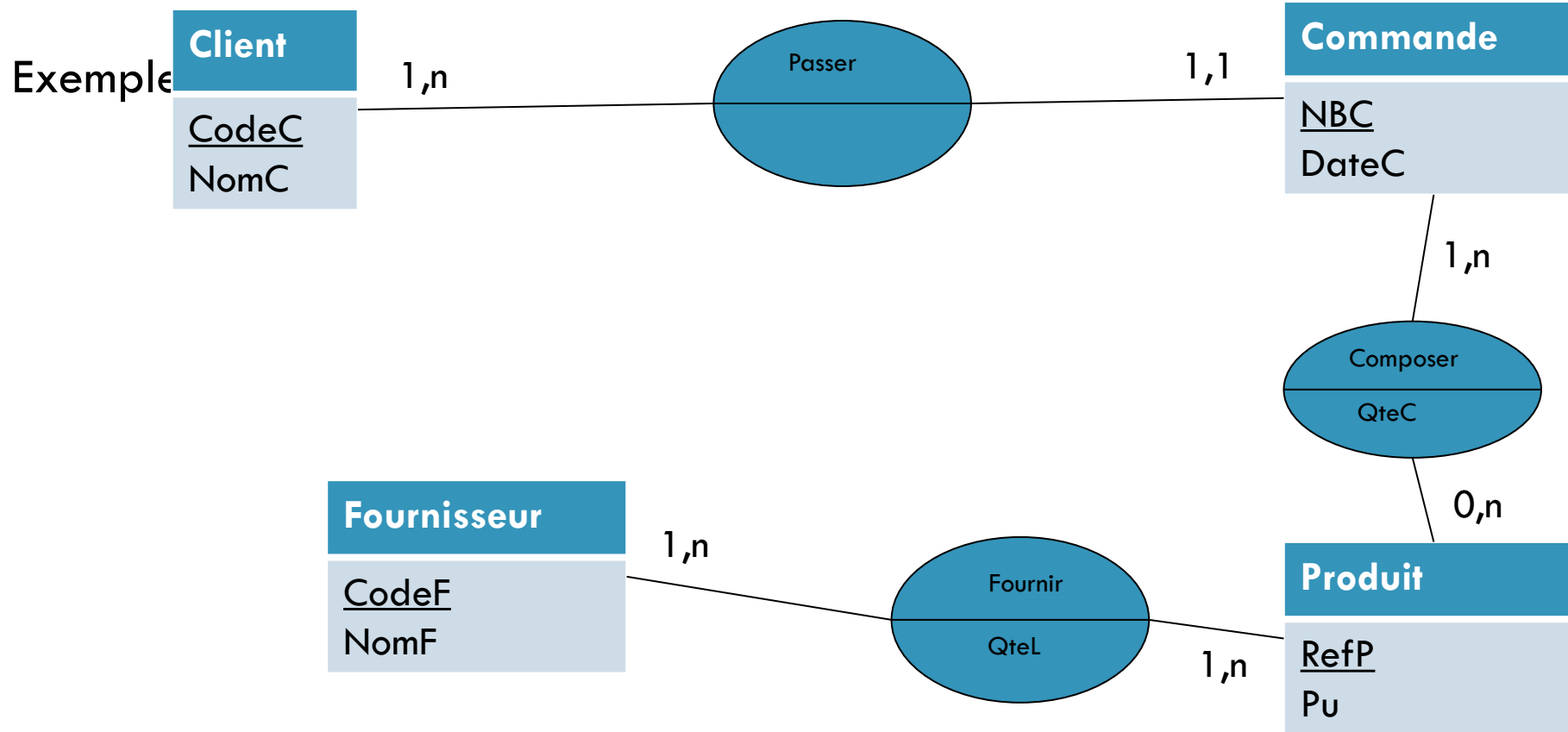


Entite_1(id_E1, Att_11, Att_12)

Entite_2(id_E2, Att_21, Att_22)

Asso.(id_E1#, id_E2#)

PASSAGE DU MCD AU MLD



PASSAGE DU MCD AU MLD

Exemple: **MLD**

- Client (CodeC, NomC)
- Commande(NBC, DateC, CodeC#)
- Produit(RefP, Pu)
- Fournisseur(CodeF, NomF)
- Composer(NBC#, RefP#, QteC)
- Fournir(CodeF#, RefP#, QteL)