



# Analyste Programmeur en Automatisme, Robotique et Informatique Industrielle TS ARII

***Module MF 1.4***

***Analyser un projet informatique***

***Système de Bases de Données – SQL - MERISE***

**Patrick MONASSIER** – année 2019-2020

## Module 1.4 – Analyser un projet informatique

### Compétences

Modéliser selon les modalités d'analyse orientées objet.

### Objectifs

Décrire un traitement séquentiel

### **Définir un système de données**

### Contenu

- Algorithme
  - Type de données
  - Structure de contrôle
  - Procédures et fonctions
- Méthode d'analyse UML
- **Analyse MERISE.**
  - *Outil d'interface graphique*
  - *Langage SQL*

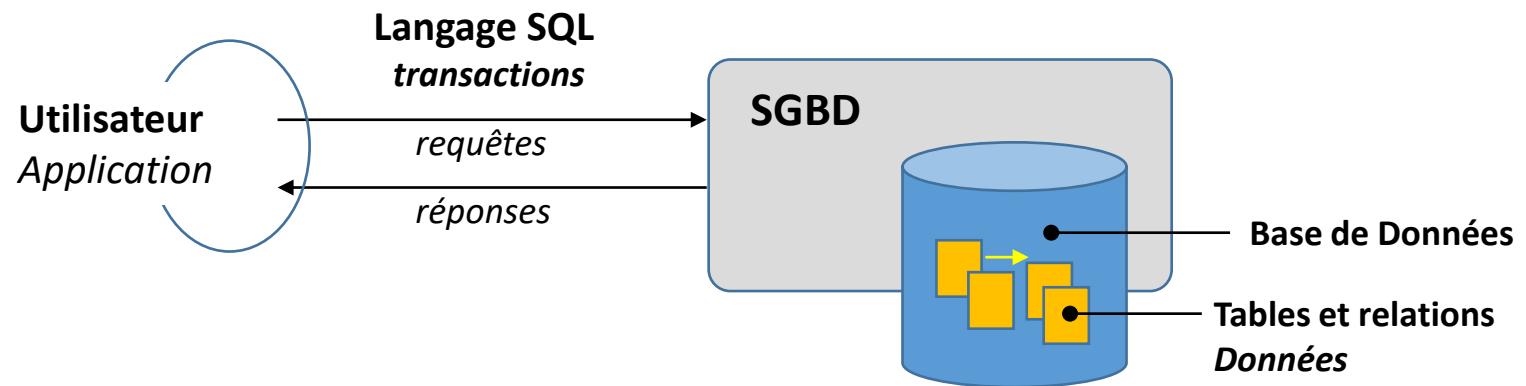
***Les principaux chapitres qui seront développés pendant le cours :***

- Introduction générale
- Types de programmation
- Les algorithmes
- La Programmation Orientée Objet – POO
- Les Bases de Données - SQL**
- La méthode d'analyse UML
- L'analyse MERISE
- Le langage SQL

de nombreux travaux pratiques et utilisations de logiciels

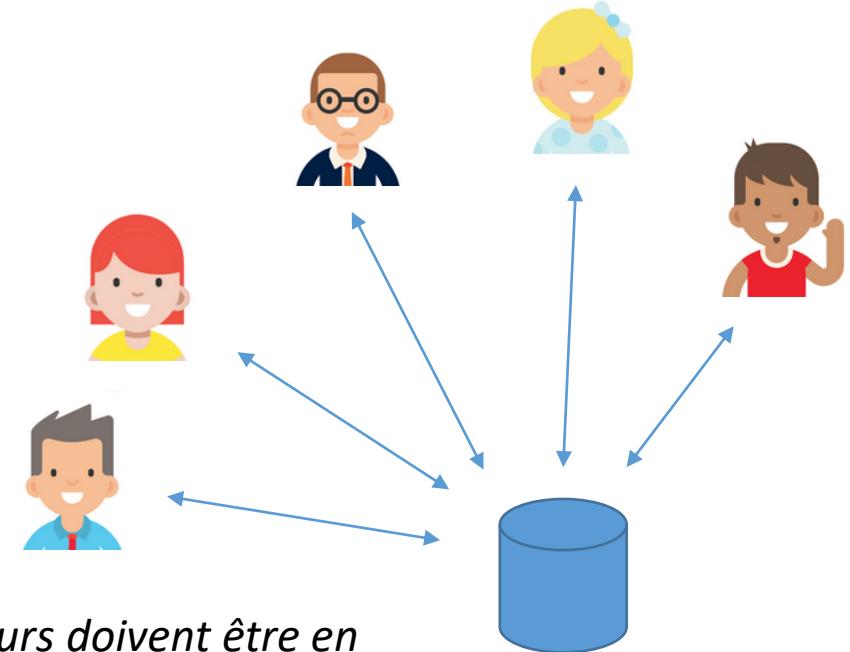
Un Système de **Gestion de Base de Données (SGBD)** - en anglais *DBMS* pour *database management system*) est un ensemble logiciel/matériel qui permet de stocker des informations dans une base de données. Un tel système permet de lire, écrire, modifier, trier, transformer ou même imprimer les données qui sont contenus dans la base de données . La majorité des SGBD du marché manipulent des **bases de données relationnelles**

Le **langage SQL** est reconnu par la grande majorité des systèmes de gestion de bases de données relationnelles du marché.



### Objectif des bases de données :

- ✓ la centralisation
- ✓ l'indépendance entre les données et les traitements
- ✓ la structuration de données complexes
- ✓ L'unicité des données
- ✓ le partage des données
- ✓ l'intégrité et la cohérence
- ✓ la confidentialité
- ✓ la sécurité

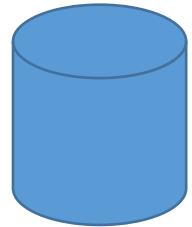


*Les multiples utilisateurs doivent être en mesure de manipuler les informations que la base contient, rapidement et n'importe quand.*

## Base de Données

Une **base de données** (son abréviation est BD, en anglais DB « database ») est une entité dans laquelle il est possible de stocker des **données** de façon structurée et avec le moins de redondance possible. Ces **données** doivent pouvoir être utilisées par des programmes, par des utilisateurs différents.

Les **objectifs** que l'on assigne généralement aux bases de données et aux systèmes qui les supportent sont les suivants : la centralisation, l'indépendance entre les données et les traitements, la structuration de données complexes, le partage des données, l'intégrité et la cohérence, la confidentialité, la sécurité.



### Exemples de Systèmes de Bases de Données

- MySQL. Véritable référence, MySQL est de loin le plus populaire des logiciels de base de données pour applications web
- PostgreSQL. C'est « l'ancêtre » des logiciels de base de données.
- Microsoft SQL Server....etc.

## Base de données distribuée

Une BDD distribuée est une database dont **certaines portions sont stockées à plusieurs endroits physiques**. Le traitement est réparti ou répliqué entre différents points d'un réseau.

Les bases de données distribuées peuvent être **homogènes ou hétérogènes**.

- ✓ Dans le cas d'un système de base de données **distribuée homogène**, tous les emplacements physiques fonctionnent avec le même hardware et tournent sous le même système d'exploitation et les mêmes applications de bases de données.
- ✓ Dans le cas d'une database **distribuée hétérogène**, le hardware, les systèmes d'exploitation et les applications de bases de données peuvent varier entre les différents endroits physiques.

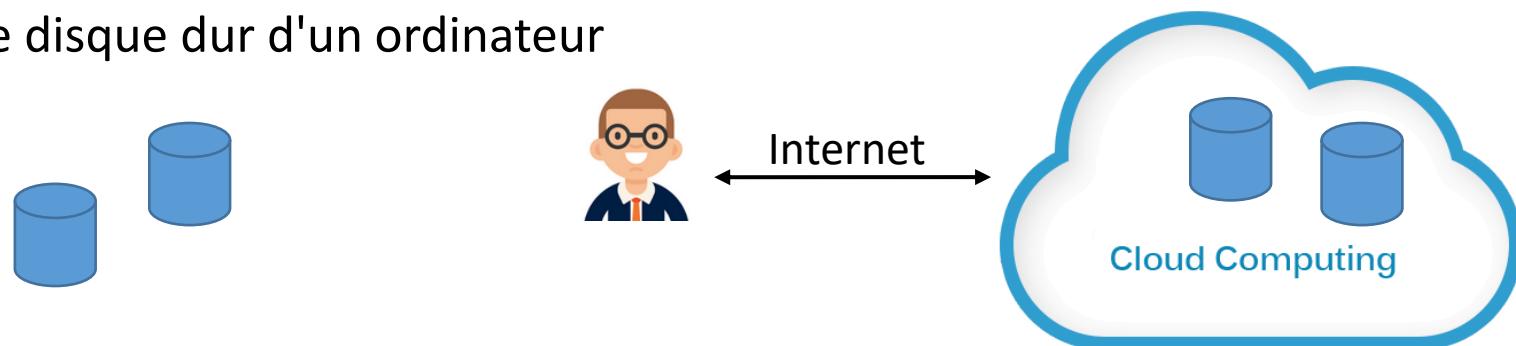


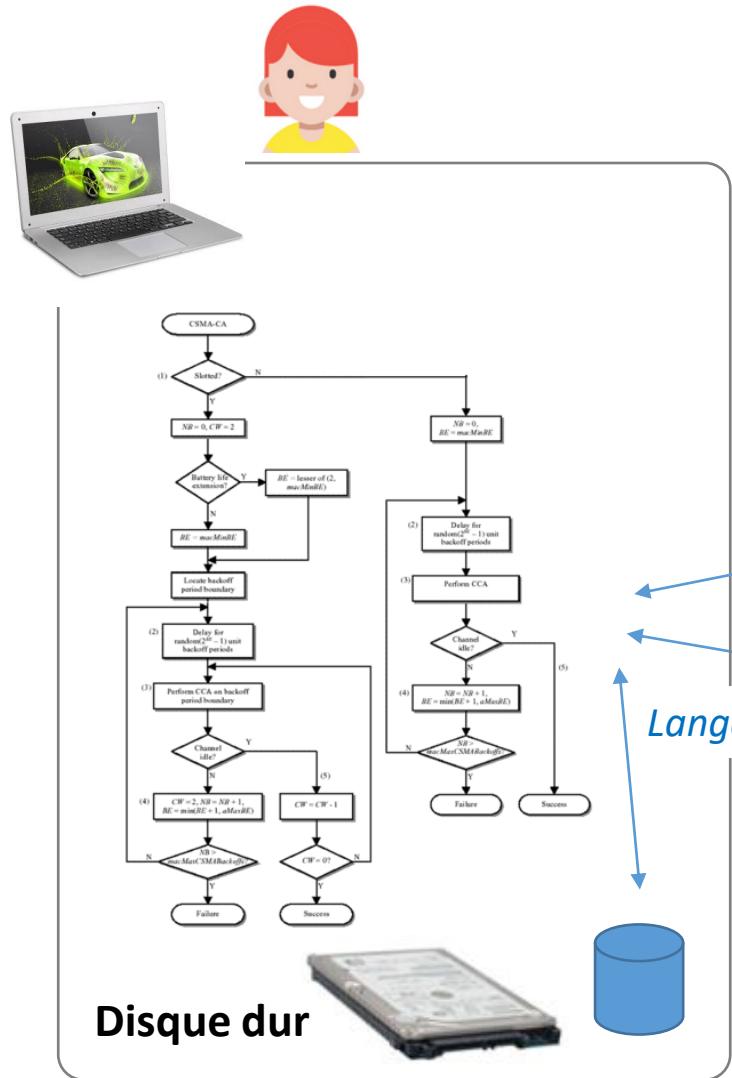
## Base de données cloud

La Base de Données Cloud est optimisée et directement créée pour les **environnements virtualisés**. Il peut s'agir d'un cloud privé, d'un cloud public ou d'un cloud hybride.

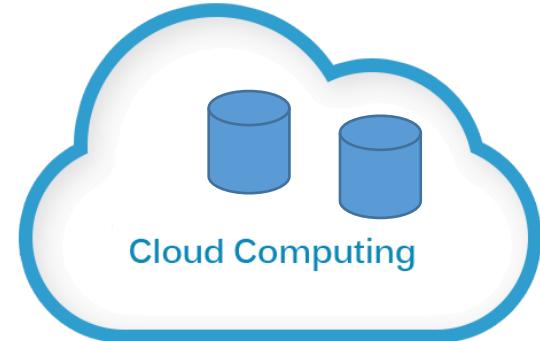
Les **bases de données cloud offrent plusieurs avantages** comme la possibilité de payer pour la capacité de stockage et la bande passante **en fonction de l'usage**. Par ailleurs, il est possible de **changer l'échelle** sur demande. Ces bases de données offrent aussi une **disponibilité** plus élevée.

Le **Cloud Computing** est un terme général employé pour désigner la livraison de ressources et de services à la demande par internet. Il désigne le stockage et l'accès aux données par l'intermédiaire **d'internet** plutôt que via le disque dur d'un ordinateur





Les données peuvent être stockées en différents endroits : sur l'ordinateur en local, sur des serveurs proches, sur des serveurs dans le cloud, en cloud **privé** ou **public**



Data center



*La protection des données est un enjeu majeur*



## *La protection des données - Cybersécurité*



Le mot **cybersécurité** désigne le rôle de l'ensemble des lois, politiques, outils, dispositifs, concepts et mécanismes de sécurité, méthodes de gestion des risques, actions, formations, bonnes pratiques et technologies qui peuvent être utilisés pour protéger les personnes et les actifs informatiques



### *Protection des données personnelles*



### RGPD

Règlement Général sur la Protection des Données

	IaaS	PaaS	SaaS
Modèle classique	Infrastructure (as a Service)	Platform (as a Service)	Software (as a Service)
Applications	Applications	Applications	Applications
Data	Data	Data	Data
Runtime	Runtime	Runtime	Runtime
Middleware	Middleware	Middleware	Middleware
O/S	O/S	O/S	O/S
Virtualization	Virtualization	Virtualization	Virtualization
Servers	Servers	Servers	Servers
Storage	Storage	Storage	Storage
Networking	Networking	Networking	Networking

*Dans l'entreprise*                            *Externalisé*

IaaS : *on héberge*  
PaaS : *on développe*  
SaaS : *on consomme*

Le **Cloud Computing** c'est pouvoir utiliser des ressources informatiques sans les posséder

## *Base de données relationnelle (SQL)*

Les bases de données relationnelles sont constituées d'un **ensemble de tables**. Au sein de ces tables, les **données sont classées par catégorie**. Chaque table comporte au moins une colonne correspondant à une catégorie. Chaque colonne comporte un certain nombre de données correspondant à cette catégorie.

L'API standard pour les bases de données relationnelles est le **Structured Query Language (SQL)**. Les bases de données relationnelles sont facilement extensibles, et de nouvelles catégories de données peuvent être ajoutées après la création de la database originale sans avoir besoin de modifier toutes les applications existantes.

## Tables

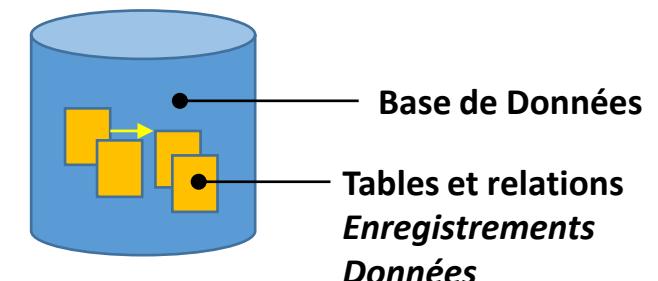
Dans les bases de données relationnelles, une **table** est un ensemble de **données** organisées sous forme d'un **tableau** où les **colonnes** correspondent à des catégories d'information (*une colonne peut stocker des numéros de téléphone, une autre des noms...*) et les **lignes** à des **enregistrements**, également appelés **entrées**.

Chaque **colonne** d'une **table** doit contenir des **données** d'un même type et porter un nom, de préférence significatif par rapport aux données contenues dans la colonne. Il y a certaines règles à respecter notamment le fait qu'il faille mettre un **identifiant** pour chaque **enregistrement** dans la table.

Diagram illustrating the components of a database table:

- Table**: Points to the overall structure of the table.
- identifiant**: Points to the **IDIdentité** column, which is the primary key.
- colonne**: Points to the **Nom** column.
- Données (champ)**: Points to the **Vacciné** field.
- enregistrement**: Points to the fourth row (José De Casta).

N° Enr.	IDIdentité	Nom	Prénom	DateNaissance	NIR	Taille	Poids	Vacciné
1	1	MARTIN	Pierre	01/02/1995	19502356874	1,75	72	<input type="checkbox"/>
2	2	VALFOND	Alain	21/05/2000	10005544123	1,84	81	<input checked="" type="checkbox"/>
3	3	LANDOIS	Aline	03/12/1988	28812665598	1,56	55	<input checked="" type="checkbox"/>
4	4	DE CASTA	José	15/01/2002	10201448965	1,78	83	<input type="checkbox"/>



## Constitution d'une table

Table Identité

Identité	
<b>ID</b>	<b>IDIdentité</b>
<b>Nom</b>	
<b>Prénom</b>	
<b>DateNaissance</b>	
<b>NIR</b>	
<b>Taille</b>	
<b>Poids</b>	
<b>Vacciné</b>	

Déclaration  
Contenu de la table

Identité				
Clé	Nom	Libellé	Type	Taille
	<b>IDIdentité</b>	Identifiant de Identité	Id. automatique	8
	Nom	Nom	Texte	50
	Prénom	Prénom	Texte	50
	DateNaissance	Date de naissance	Date	8
	NIR	N° Sécurité Sociale	Numérique	8
	Taille	Taille	Numérique	8
	Poids	Poids	Numérique	4
	Vacciné	Vacciné	Booléen	1

```
CREATE TABLE `Identité` (
  `IDIdentité` int(11),
  `Nom` varchar(50),
  `Prenom` varchar(50),
  `DateNaissance` date,
  `NIR` int(11),
  `Taille` float,
  `Poids` float,
  `Vaccine` tinyint(1)
)
```

Colonnes Champs

N° Enr.	IDIdentité	Nom	Prénom	DateNaissance	NIR	Taille	Poids	Vacciné
1	1	MARTIN	Pierre	01/02/1995	19502356874	1,75	72	<input type="checkbox"/>
2	2	VALFOND	Alain	21/05/2000	10005544123	1,84	81	<input checked="" type="checkbox"/>
3	3	LANDOIS	Aline	03/12/1988	28812665598	1,56	55	<input checked="" type="checkbox"/>
4	4	DE CASTA	José	15/01/2002	10201448965	1,78	83	<input type="checkbox"/>

Lignes  
Enregistrements

## Introduction au SQL

Quelques requêtes de base

- SELECT
- INSERT
- UPDATE
- DELETE
- UNION

### Création d'une requête

#### Type de requête

	<b>Sélection (SELECT)</b> Requête permettant de sélectionner des enregistrement
	<b>Ajout (INSERT)</b> Requête permettant d'insérer un enregistrement
	<b>Modification (UPDATE)</b> Requête permettant de modifier des enregistrements
	<b>Suppression (DELETE)</b> Requête permettant de supprimer des enregistrements
	<b>Union (UNION)</b> Requête permettant de réunir le résultat de plusieurs requêtes

## Introduction au SQL

### Des exemples

```
INSERT INTO table_name (column1, column2, column3, ...)  
VALUES (value1, value2, value3, ...);
```

```
INSERT INTO Customers (CustomerName, ContactName, Address, City, PostalCode, Country)  
VALUES ('Cardinal', 'Tom B. Erichsen', 'Skagen 21', 'Stavanger', '4006', 'Norway');
```

```
UPDATE table_name
```

```
SET column1 = value1, column2 = value2, ...  
WHERE condition;
```

```
UPDATE Customers  
SET ContactName='Juan'  
WHERE Country='Mexico';
```

```
DELETE FROM table_name WHERE condition;
```

```
DELETE FROM Customers  
WHERE CustomerName='Alfreds Futterkiste';
```

```
SELECT column_name(s) FROM table1
```

```
UNION
```

```
SELECT column_name(s) FROM table2
```

```
SELECT City FROM Customers
```

```
UNION
```

```
SELECT City FROM Suppliers
```

```
ORDER BY City;
```

## Introduction au SQL

```
SELECT * FROM table_name;
```

```
SELECT column1, column2, ...  
FROM table_name;
```

```
SELECT CustomerName, city  
FROM Customers;
```

```
SELECT * FROM Customers  
WHERE Country='Germany' AND (City='Berlin' OR City='München');
```

```
SELECT * FROM Customers  
WHERE NOT Country='Germany'  
AND NOT Country='USA';
```

```
SELECT column1, column2, ...  
FROM table_name  
WHERE condition;
```

```
SELECT * FROM Customers  
WHERE Country='Mexico';
```

```
SELECT * FROM Customers  
WHERE CustomerName LIKE 'a%';
```

```
SELECT * FROM Products  
WHERE Price BETWEEN 10 AND 20;
```

```
SELECT * FROM Customers  
WHERE Country  
IN ('Germany', 'France', 'UK');
```

```
SELECT column1, column2, ...  
FROM table_name  
ORDER BY column1, column2, ... ASC|DESC;
```

```
SELECT * FROM Customers  
ORDER BY Country;
```

```
SELECT * FROM Customers  
ORDER BY Country DESC;
```

```
SELECT * FROM Customers  
ORDER BY Country ASC, CustomerName DESC;
```

```
SELECT * FROM Customers  
WHERE Country='Mexico';  
ORDER BY Country ASC, CustomerName DESC;
```

**Introduction  
au SQL****SQL Tutorial**

SQL HOME	SQL Inner Join
SQL Intro	SQL Left Join
SQL Syntax	SQL Right Join
SQL Select	SQL Full Join
SQL Select Distinct	SQL Self Join
SQL Where	SQL Union
SQL And, Or, Not	SQL Group By
SQL Order By	SQL Having
SQL Insert Into	SQL Exists
SQL Null Values	SQL Any, All
SQL Update	SQL Select Into
SQL Delete	SQL Insert Into Select
SQL Select Top	SQL Case
SQL Min and Max	SQL Null Functions
SQL Count, Avg, Sum	SQL Stored Procedures
SQL Like	SQL Comments
SQL Wildcards	
SQL In	
SQL Between	
SQL Aliases	
SQL Joins	

**SQL Database**

SQL Create DB
SQL Drop DB
SQL Backup DB
SQL Create Table
SQL Drop Table
SQL Alter Table
SQL Constraints
SQL Not Null
SQL Unique
SQL Primary Key
SQL Foreign Key
SQL Check
SQL Default
SQL Index
SQL Auto Increment
SQL Dates
SQL Views
SQL Injection
SQL Hosting

**SQL References**

SQL Keywords
MySQL Functions
SQL Server Functions
MS Access Functions
SQL Operators
SQL Data Types
SQL Quick Ref

<https://www.w3schools.com/sql/default.asp>

## Module 1.4 – Analyser un projet informatique

### Compétences

Modéliser selon les modalités d'analyse orientées objet.

### Objectifs

Décrire un traitement séquentiel

Définir un système de données.

### Contenu

- Algorithme
  - Type de données
  - Structure de contrôle
  - Procédures et fonctions
- Méthode d'analyse UML
- Analyse MERISE.
  - Outil d'interface graphique
  - Langage SQL

### *Suite de ce diaporama*

- Analyse MERISE.
  - Outil d'interface graphique
  - Langage SQL

**MERISE** est une méthode de conception, de développement et de réalisation de projets informatiques. Le but de cette méthode est d'arriver à concevoir un système d'information. La méthode MERISE est basée sur la séparation des données et des traitements à effectuer en plusieurs modèles conceptuels et physiques. La séparation des données et des traitements assure une longévité au modèle. En effet, l'agencement des données n'a pas à être souvent remanié, tandis que les traitements le sont plus fréquemment.

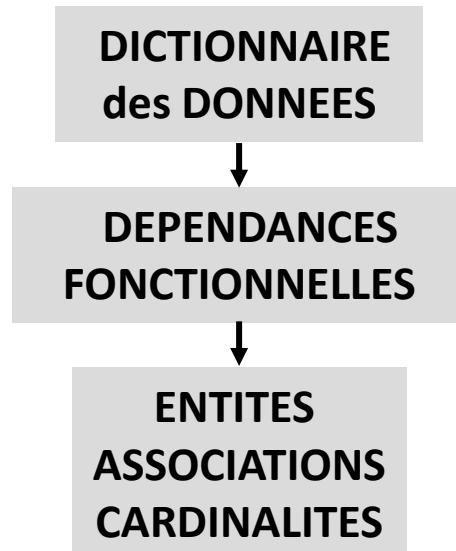
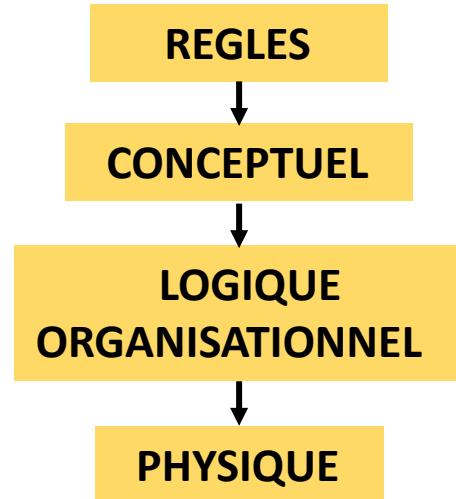
Le **langage SQL** (*Structured Query Language*) peut être considéré comme le langage d'accès normalisé aux bases de données. Le succès du langage SQL est dû essentiellement à sa simplicité et au fait qu'il s'appuie sur le schéma conceptuel pour énoncer des requêtes en laissant le SGBD responsable de la stratégie d'exécution.

MERISE est donc une méthode d'analyse et de conception des SI basée sur le principe de la séparation des données et des traitements. Elle possède un certain nombre de **modèles** (ou **schémas**) qui sont répartis sur 3 niveaux :

- La mise en place de **règles de gestion**,
- Le niveau **conceptuel**,
- Le niveau **logique ou organisationnel**,
- Le niveau **physique**.

Le **Modèle Conceptuel des Données MCD** est une représentation graphique et structurée des informations mémorisées par un Système d'Information. Le MCD est basé sur deux notions principales : les **entités** et les **associations**, d'où sa seconde appellation : le **schéma Entité/Association**. L'élaboration du MCD passe par les étapes suivantes :

- L'élaboration du **dictionnaire des données**,
- La recherche des **dépendances fonctionnelles** entre ces données,
- L'élaboration du MCD (création des **entités** puis des **associations** puis ajout des **cardinalités**).



## Les règles de gestion métiers

Recueillir les **besoins** des futurs utilisateurs de l'application. A partir de ces besoins, établir les **règles de gestion** des données à conserver. Si les futurs utilisateurs ne sont pas en mesure de fournir ces règles avec suffisamment de précision, il faut les **interroger** afin d'établir ces règles.

Par exemple : **informatisation d'une bibliothèque**, les règles de gestion sont les suivantes :

- Pour chaque livre, on doit connaître le titre, l'année de parution, un résumé et le type (roman, poésie, science fiction, ...).
- Un livre peut être rédigé par aucun (dans le cas d'une œuvre anonyme), un ou plusieurs auteurs dont on connaît le nom, le prénom, la date de naissance et le pays d'origine.
- Chaque exemplaire d'un livre est identifié par une référence composée de lettres et de chiffres et ne peut être paru que dans une et une seule édition.
- Un inscrit est identifié par un numéro et on doit mémoriser son nom, prénom, adresse, téléphone et adresse e-mail.
- Un inscrit peut faire zéro, un ou plusieurs emprunts qui concernent chacun un et un seul exemplaire. Pour chaque emprunt, on connaît la date et le délai accordé (en nombre de jours).

## Le dictionnaire des données

C'est un document qui regroupe toutes les données conservées dans la base.

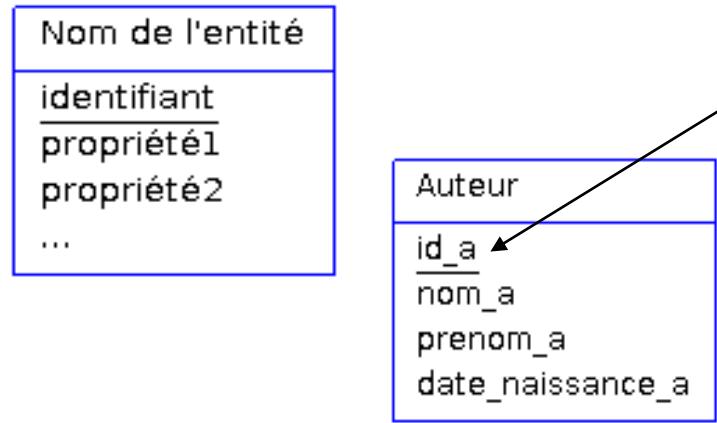
Pour chaque donnée, il indique :

- Le **code mnémonique** : libellé désignant la donnée
- La **désignation** : mention décrivant ce à quoi la donnée correspond
- Le **type de donnée** : alphabétique, numérique, date, booléen...etc.
- La **taille** de la donnée
- Des **remarques** éventuelles

Les données qui figurent dans dictionnaire des données, donc dans le MCD, sont **élémentaires**. Les données ne sont **ni calculées, ni composées**.

## Les entités

Chaque **entité** est unique et est décrite par un ensemble de propriétés encore appelées **attributs** ou **caractéristiques**. Une des propriétés de l'entité est **l'identifiant** : on utilise une donnée de type entier qui s'incrémente pour chaque occurrence (*enregistrement*).



À partir de cette entité, on peut retrouver la règle de gestion suivante : un auteur est identifié par un **numéro unique** (id\_a) et est caractérisé par un nom, un prénom et une date de naissance

Une entité peut n'avoir aucune, une ou plusieurs occurrences. Pour illustrer ce terme d'«occurrence», voici un exemple de table d'occurrences de l'entité Auteur, cette table est composée de trois occurrences (enregistrements) de l'entité Auteur :

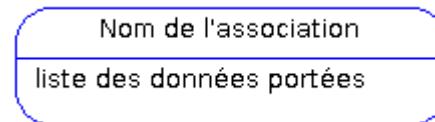
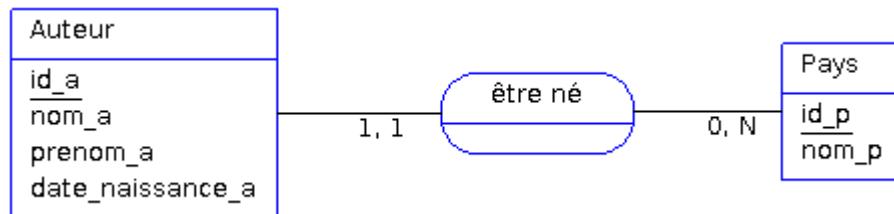
<b>id_a</b>	<b>nom_a</b>	<b>prenom_a</b>	<b>date_naissance_a</b>
1	Hugo	Victor	1802-02-26
2	Rimbaud	Arthur	1854-10-20
3	de Maupassant	Guy	1850-08-05

## Les associations

Une association définit un lien sémantique entre une ou plusieurs entités. En effet, la définition de liens entre entités permet de traduire une partie des règles de gestion qui n'ont pas été satisfaites par la simple définition des entités. Le formalisme d'une association est le suivant :

Généralement le nom de l'association est un verbe définissant le lien entre les entités qui sont reliées par cette dernière.

Par exemple :



Ici l'association «être né» traduit les deux règles de gestion suivantes :

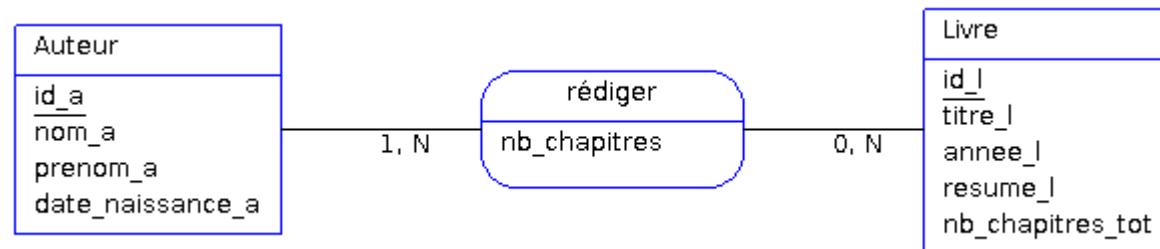
- Un auteur est né dans **un et un seul** pays,
- Dans un pays, sont nés **aucun, un ou plusieurs** auteurs.

On remarque que cette association est caractérisée par ces annotations **1,1** et **0,N** qui permettent de définir les règles de gestions précédentes. Ces annotations sont appelées les **cardinalités**.

## Les cardinalités

Une cardinalité est définie par un ‘minimum’ et un ‘maximum’

Les cardinalités les plus répandues sont les suivantes : **0,N ; 1,N ; 0,1 ; 1,1**. L'identifiant d'une association ayant des cardinalités 0,N/1,N de part et d'autre, est obtenu par la concaténation des entités qui participent à l'association. Imaginons l'association suivante :



Ici un auteur rédige au moins un ou plusieurs livres et pour chaque livre, on connaît le nombre de chapitres rédigés par l'auteur (on connaît aussi le nombre total de chapitres pour chaque livre).

L'association «rédiger» peut donc être identifiée par la concaténation des propriétés id\_a et id\_l. Ainsi, le couple **id\_a, id\_l** doit être unique pour chaque occurrence de l'association.

Analyste Programmeur en Automatisme, Robotique et  
Informatique Industrielle  
TS ARII

***Module MF 1.4***  
***Analyser un projet informatique***  
***Système de Bases de Données – SQL - MERISE***

**Fin de Présentation**