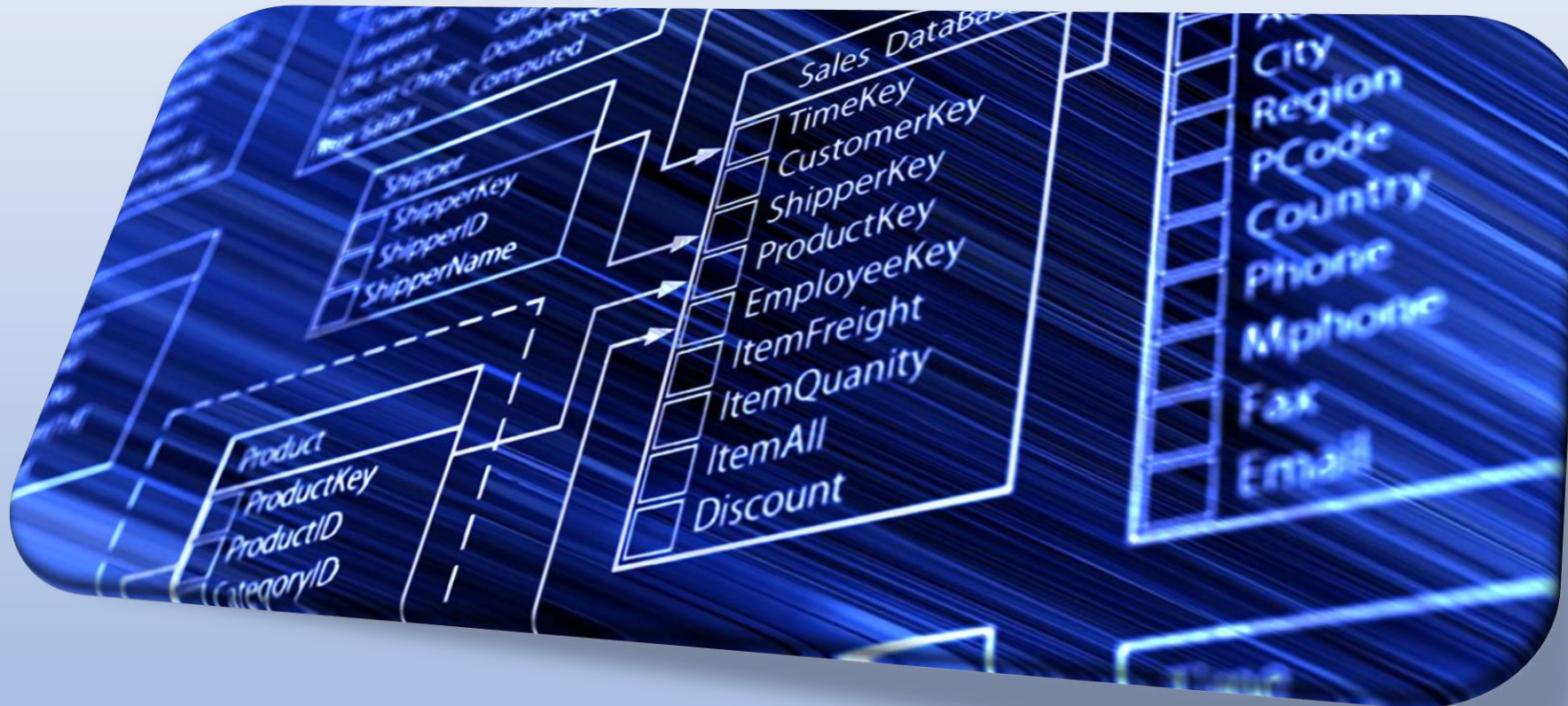


# TIn311 – Initiation aux bases de données



# Introduction

## Contact

### Infos

Alexandre CONDETTE



- ALUMNI IPSA Toulouse – Promo 2020 – Major ELSS
- Ingénieur développement logiciel chez Spacebel
- Gestion Opérationnelle des Simulateurs Numériques de satellites en orbite basse au CST (CNES) de Toulouse
- Astrophysique, Electronique, Python, C / C++, Linux, Shell, TCL, SQL, Bases de données

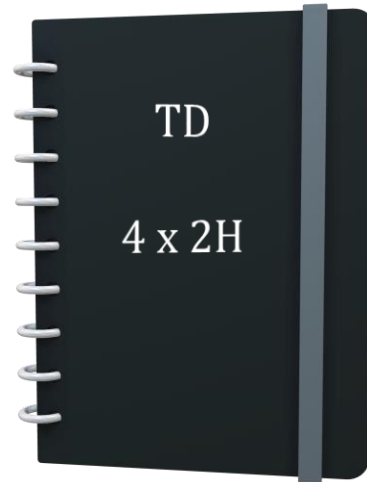
### Contact

- [alexandre.condette@spacebel.fr](mailto:alexandre.condette@spacebel.fr)
- [alexandre.condette.external@cnes.fr](mailto:alexandre.condette.external@cnes.fr)
- [alexandre2.condette@ipsa.fr](mailto:alexandre2.condette@ipsa.fr)



## Introduction

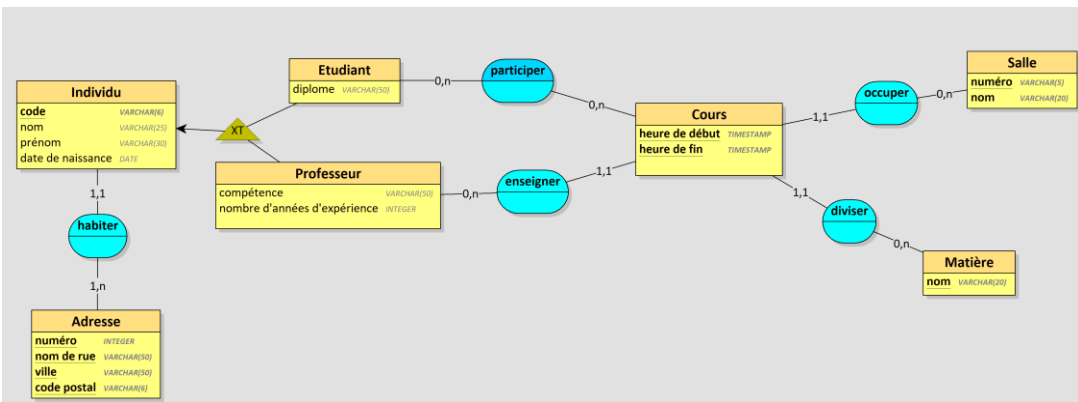
### Organisation de la matière



Notation : 1 Partiel de 2H (format à définir)  
1 x 4h de BE noté

# Objectifs

- Comprendre et maîtriser les concepts de bases de données relationnelles
- Savoir modéliser des données au niveau « **conceptuel** » et « **logique** » (MCD / MLD) en respectant les conventions graphiques et les règles de transformation d'un modèle MCD en MLD
- Aborder le langage **SQL** pour concevoir et rédiger des interrogations pertinentes de bases de données
- Concevoir un outil de production individuel automatisé





# Définitions

## Données

- En informatique, une donnée est une information spécifique ou une valeur stockée de manière structurée
- Représente une unité d'information élémentaire : entité, interaction, transaction, événements
- Peut être traitée, manipulée, analysée
- Différentes formes : nombres, texte, image, vidéo
- Défis :
  - Accès rapides, faciles, concurrent
  - Organisation des données
  - Fiabilité
  - Intégrité
  - Durabilité
  - Sécurité
  - Gestion de grande quantité de données



## Définitions

### Base de données

- Une base de données est un ensemble d'informations, organisées de manière à être facilement accessibles, gérées et mise à jour.
- Permet donc de stocker, de gérer, de récupérer et manipuler les données de manière efficace et cohérente.
- Facilite l'accès, la recherche et l'analyse des données en fonction des besoins spécifique de l'utilisateur.
- Généralement, l'administrateur de la base de données régule les accès des utilisateurs afin de contrôler leurs actions et d'analyser les usages. Pour garantir la cohérence des données et l'intégralité des transactions, toutes les transactions réalisées sur une base de données doivent répondre aux exigences de la conformité ACID :
  - Le principe d'Atomicité garantit la bonne exécution de la transaction. Les transactions de base de données, peuvent être décomposées en plus petites parties. Si une partie d'une transaction échoue, toute la transaction sera annulée.
  - La propriété de Cohérence signifie que seules les données qui suivent des règles prédéfinies peuvent être écrites dans la base de données.
  - L'isolement : capacité de traiter simultanément plusieurs transactions de manière indépendante.
  - La durabilité



## Définitions

### Système de Gestion de Base de Données

- Un SGBD est un logiciel spécialisé pour :
  - Définir le modèle de données
  - Créer et gérer efficacement les bases de données (administration de la base)
  - Fonctionnalités pour le stockage, l'organisation, récupérer et mettre à jour les données de manière structurée.
  - Simplifié la manipulation des données et garantir leur intégrité
  - Sécurité et performance
  - Concurrence des accès, droits d'accès

# SGBD et Interfaces

- Un SGBD est un moteur qui fournit un travail avec lequel on interagit au travers de requêtes SQL.
- Plusieurs interfaces possibles : GUI pour rendre les interactions plus faciles, distincts du SGBD, mais qui facilitent la saisie des requêtes et la visualisation des résultats
- DBBrowser for SQLite, phpMyAdmin, pgAdmin, SQL Server Studio, etc...

**phpMyAdmin**

Récents Préférées

- 'test' applications
  - Nouvelle base de données
    - information\_schema
    - mysql
    - performance\_schema
    - sys
    - test
      - Nouvelle table
        - applications
          - Colonnes
            - Nouvelle colonne
            - code (char)
            - cost (int, nullable)
            - description (text, nullable)
            - name (text)

Parcourir Structure SQL Rechercher Insérer Exporter Importer Privileges Opérations

⚠ La sélection courante ne contient pas de colonne unique. Les grilles d'édition, les cases à cocher ainsi que les liens Éditer, Copier et Supprimer ne sont pas disponibles.

✓ Affichage des lignes 0 - 2 (total de 3, traitement en 0.0006 seconde(s).)

```
SELECT * FROM `applications`
```

☐ Profilage [ Éditer en ligne ] [ Éditer ] [ Expliquer SQL ] [ Créer le code source PHP ] [ Actualiser ]

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table

+ Options

code	name	description	cost
1V73	Splunk Monitoring	NULL	NULL
2G59	Zabbix	NULL	42
C500	BSM	Scenario Monitoring	NULL

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table

Opérations sur les résultats de la requête

Imprimer Copier dans le presse-papiers Exporter Afficher le graphique Créer une vue

DB Browser for SQLite - Y:\cours\ln30\ln30.db

Fichier Édition Vue Outils Aide

Nouvelle base de données Ouvrir une base de données Enregistrer les modifications Annuler les modifications Ouvrir un Projet Attacher une Base de Données

Structure de la Base de Données Parcourir les données Éditer les Pragma Exécuter le SQL

SQL 1 x

```
1 SELECT * FROM Eleve;
```

	numero_license	nom	prenom	age	nom_cx
1	29	CHENAULT	Tom	14	Poussin
2	117	RONDEY	Julien	NULL	Poussin
3	90	MOYON	Vincent	NULL	Minimes
4	27	CHAPUT	Thomas	NULL	Minimes
5	2	ALLEMANG	Owen	14	Poussin
6	80	MADELBOS	Clemence	14	Poussin

Result: 134 enregistrements ramenés en 118ms  
At line 1:  
SELECT \* FROM Eleve;

Éditer le contenu d'une Cellule

Mode : Texte Importer Exporter Définir comme NULL

NULL

Type actuel des données de la cellule : NULL  
0 octet Appliquer

Serveur distant

Identité

Nom Commit Dernière modification Taille

Journal SQL Graphique DB Schema Serveur distant



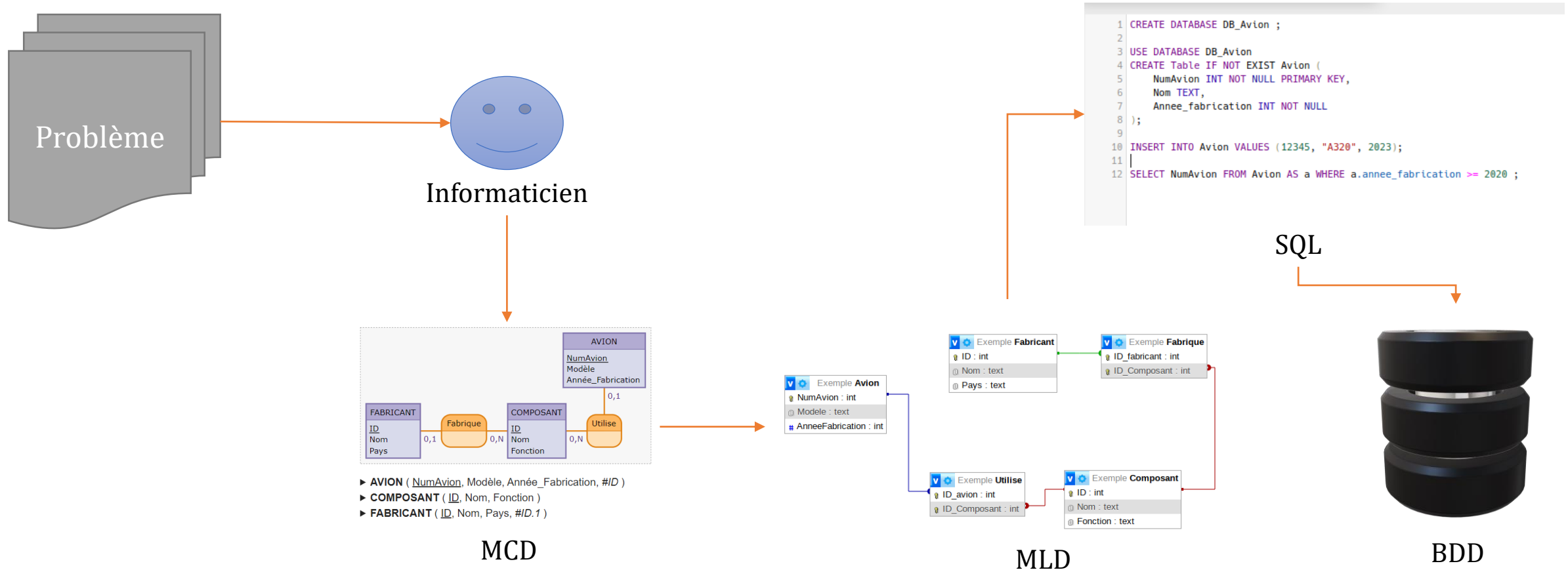
# Définitions

## DB Browser for SQLite

- SQLite : Un SGBD-R sans serveur, compact (<1 Mio), une base = un fichier
- Multi OS : MacOS, Windows, Linux
- Très répandu, même sur Android et iOS, chaque portable utilise des centaines de bases
  - Plus de 1 trilion de bases SQLite actuellement utilisée environ
- Limites :
  - Pas de relation client / serveur, la base est sur la même machine que l'application
  - Très souple et conciliant
  - Certaines fonctionnalités plus avancées ne sont pas disponibles nativement
- Mais bon pour les sites web avec trafic moyen, l'analyse de données etc. si la base n'est pas trop volumineuse



## La méthode



# Le Modèle relationnel

## Genèse

- Inventé par Edgar Codd, informaticien britannique en 1970,
- Manière de modéliser les relations existantes entre plusieurs informations
- Permet de décrire les relations de façon logique et mathématique (Algèbre relationnelle).
- Données organisées sous formes de tables (appelées « relations ») associées entre elles,
  - Dans ces tables : les lignes (ou « n-uplet ») correspondent à un enregistrement
  - Une colonne correspond à un attribut / une propriété cohérente, intègre avec certaines contraintes
- Principe de clés :
  - Clé primaire : Colonne ou ensemble de colonne dont la valeur permet d'identifier de manière univoque chaque enregistrement de la table
  - Clés étrangères : Etablisent les relations entre les tables
- Opérations de manipulation
- Normalisation

# Le Modèle relationnel

## Exemple

### Livres

Identifiant	Titre	Auteur	Année
1	Une brève histoire du temps	S. Hawking	1988
2	L'origine des espèces	C. Darwin	1859
3	La sixième extinction ou comment l'Homme tue la planète	E. Kolbert	2015

## Le Modèle relationnel

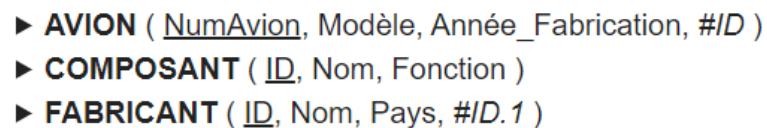
### Avantages

- Simplicité conceptuelle : facile à comprendre et à concevoir
- Possibilité de changer la structure sans modifier les applications
- Flexibilité : Interconnexions de tables pour représenter des relations complexes, combinaison de données d'une manière non prévue à l'origine
- Normalisation : Minimise la redondance, optimise la structure, le stockage et la cohérences des données et de leur modification
- Langage de haut niveau SQL
- Adapté à la gestion dans des environnements multi-utilisateurs → Essentiel pour les systèmes critiques



# Modèle Conceptuel de Données

- Indépendant du modèle informatique



## Le MCD

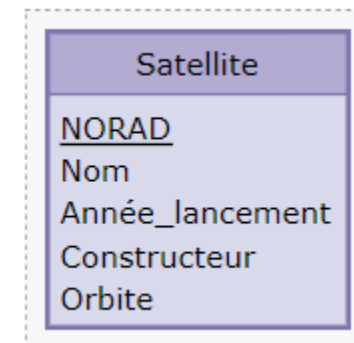
### Les Entités

#### Entité

- Représente un ensemble « **d'objets** » (concrets ou abstrait) homogènes, dotés d'une existence propre et autonome dans l'organisation
- Une entité a un **nom unique** afin de la manipuler facilement (ex. Patient, Personne, Client, Avion, Satellite, etc.)

#### Attribut (ou propriété)

- « **Champ** » de l'entité.
- Donnée qui sera effectivement stockée dans la base.
- **Atomique** : Donnée la plus fine que l'on trouve dans la base et qui a un sens lui même



#### ▼ SATELLITE ( NORAD, Nom, Année\_lancement, Constructeur, Orbite )

- Le champ *NORAD* constitue la clé primaire de la table. C'était déjà un identifiant de l'entité *Satellite*.
- Les champs *Nom*, *Année\_lancement*, *Constructeur* et *Orbite* étaient déjà de simples attributs de l'entité *Satellite*.

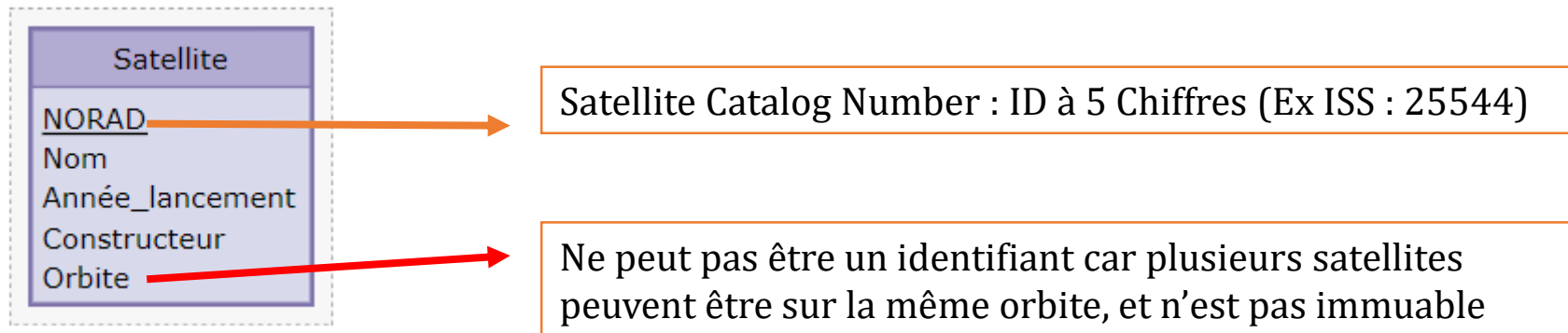
## Le MCD

### Identifiant de l'entité

#### Identifiant

- L'identifiant d'une entité est un attribut particulier, qui permet de caractériser de manière unique un élément de cette entité
- Immuable et minimal, peut être artificiel si aucun attribut de l'entité ne l'identifie naturellement
- Présenté de manière soulignée dans la représentation

#### Exemple :



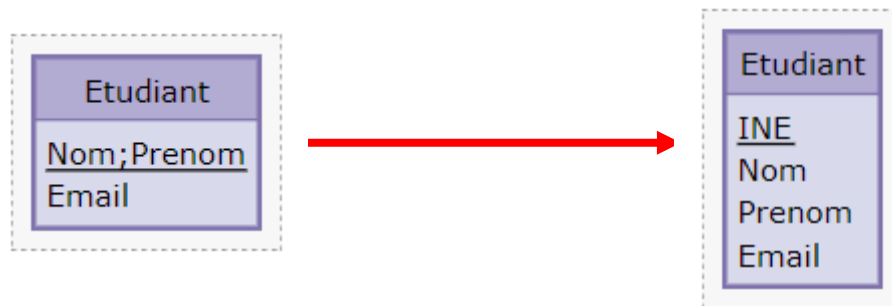
## Le MCD

### Identifiant de l'entité

#### Identifiant

- L'identifiant peut être composé de plusieurs attributs, mais alors autant préférer un identifiant artificiel pour des raisons pratique
- Plus lourd à gérer, comparaisons de 2 éléments plutôt que d'un seul.
- De préférence, utiliser des numéros / matricules / ID

#### Exemple :

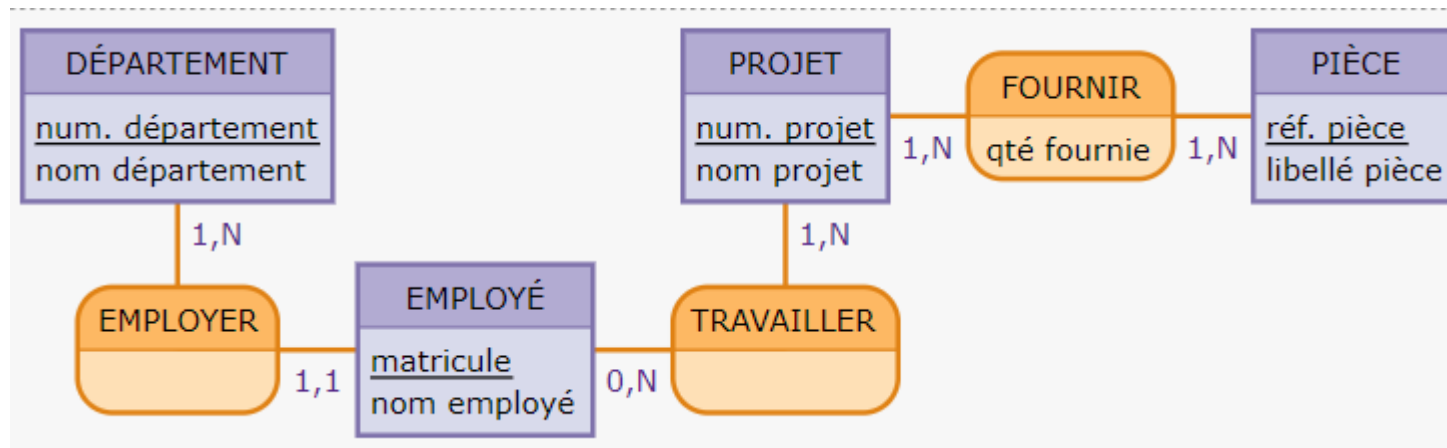


- De plus dans l'exemple de gauche, il peut s'avérer que l'identifiant ne puisse pas se résumer à la combinaison de 2 valeurs
- (Ex : 2 étudiants qui ont exactement le même nom et prénom)
- Dans ce cas, il faut revoir tout le modèle et les requêtes !

## Le MCD

### Les Associations

- Une association correspond à la relation, correspondance entre deux entités
- Chaque entité joue un rôle propre dans la relation
- Une association peut également être porteuse d'informations supplémentaires (avoir ses propres attributs)
- De préférence représentée par un verbe

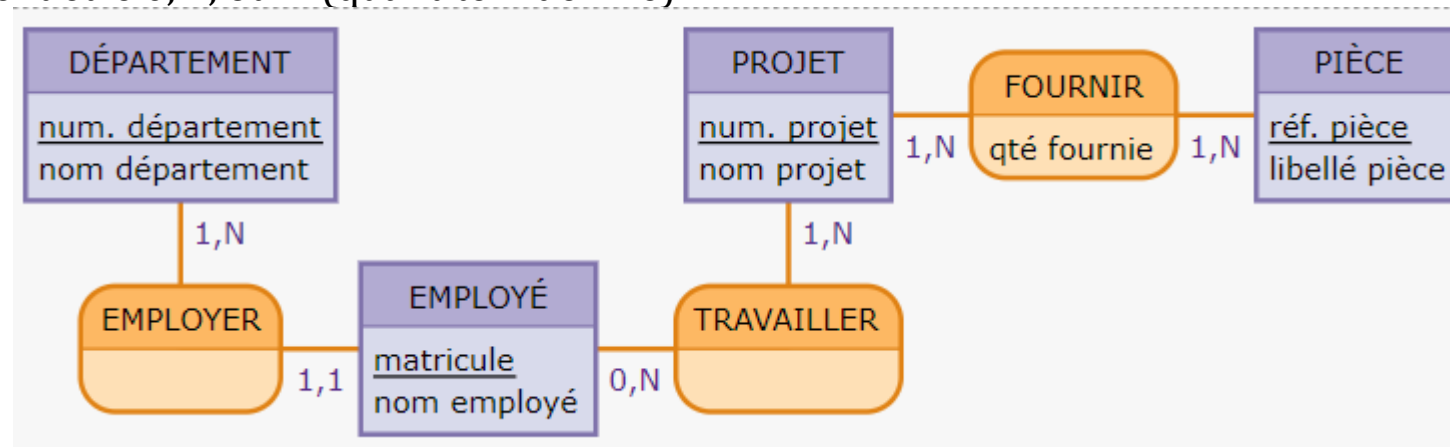




## Le MCD

### Les Cardinalités

- Nombre minimum et maximum de fois qu'une entité peut-être engagée dans une relation
- Le minimum (0 ou 1) est souvent discutable :
  - 0 indique une possibilité : ex « Il se peut qu'aucun employé ne travail sur un projet »
  - 1 indique une contrainte : ex « Pour qu'un département existe, il faut qu'il y ait au moins 1 employé dans ce département »
- Le maximum a un impact sur la manière de concevoir le MLD
- Les cardinalités peuvent être 0, 1, ou N (quantité indéfinie)



## Le MCD

### Les types d'associations

Cardinalité 1	Cardinalité 2	Type d'association
0, 1	1, 1	Un-à-un
1, 1	1, 1	Erreur les entités peuvent n'en faire qu'une, le modèle est à revoi
0, 1 1, 1	0, N 1, N	Un à plusieurs
0, N 1, N	0, N 1, N	Plusieurs à plusieurs

## Le MCD

### Méthode

1. Identifier les entités (pertinentes) en présence
  - Trouver l'identifiant ou en créer un
  - Trouver les attributs
2. Identifier les relations (pertinentes) entre ces entités
  - Trouver d'abord les liens sous forme de verbe
  - Puis appliquer les cardinalités



#### Exemple :

Un organisme départemental souhaite mettre en place une base de données pour le suivi des films projetés dans les salles de cinéma du département. Pour simplifier on considère qu'une salle de cinéma ne projette qu'un seul film à une heure donnée. Toutefois, un même film peut-être projeté simultanément dans plusieurs salles.

Pour des raisons d'organisation et d'espace, une salle de cinéma ne projette chaque film qu'une seule fois par jour et toujours à la même heure. On ne souhaite pas archiver l'historique des projections des films par salle,

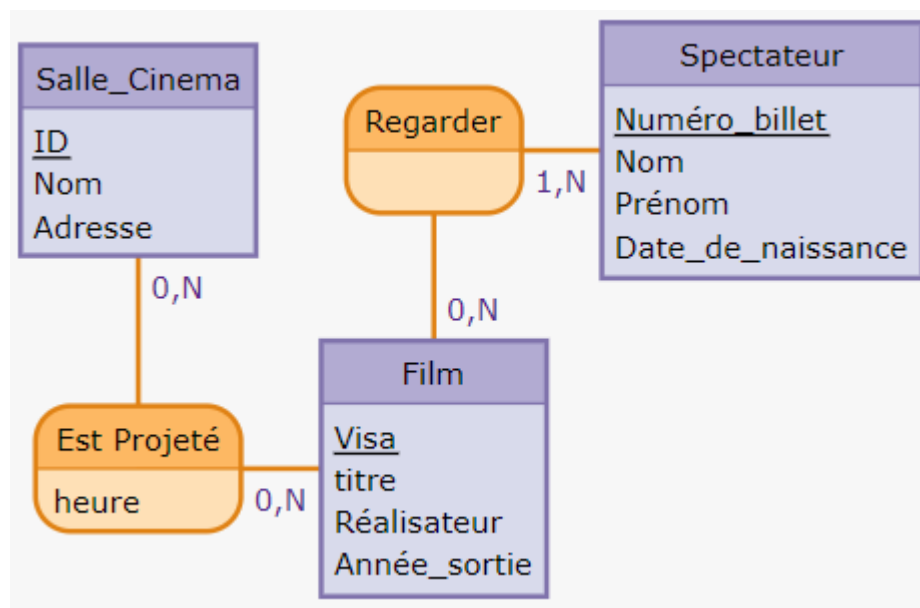
On dispose pour chaque salle des données suivantes : Nom, Adresse, liste des films projetés

Chaque spectateur est identifié par un numéro de billet, son nom, son prénom, son adresse et sa date de naissance

Pour chaque film, on souhaite stocker son visa d'exploitation, son titre, le nom du réalisateur et son année de sortie

## Le MCD

### Exemples



Un organisme départemental souhaite mettre en place une base de données pour le suivi des **films projetés** dans les **salles de cinéma** du département. Pour simplifier on considère qu'une salle de cinéma ne **projette** qu'un seul film à une heure donnée. Toutefois, **un même film peut-être projeté simultanément dans plusieurs salles**.

Pour des raisons d'organisation et d'espace, une salle de cinéma ne projette chaque film qu'une seule fois par jour et toujours à la même heure. On ne souhaite pas archiver l'historique des projections des films par salle,

On dispose pour chaque salle des données suivantes : **Nom**, **Adresse**, liste des films projetés

Chaque **spectateur** est identifié par un numéro de billet, son nom, son prénom, son adresse et sa date de naissance (Donc implicitement, **un spectateur doit regarder** un film !)

Pour chaque film, on souhaite stocker son visa d'exploitation, son titre, le nom du réalisateur et son année de sortie

## Le MLD

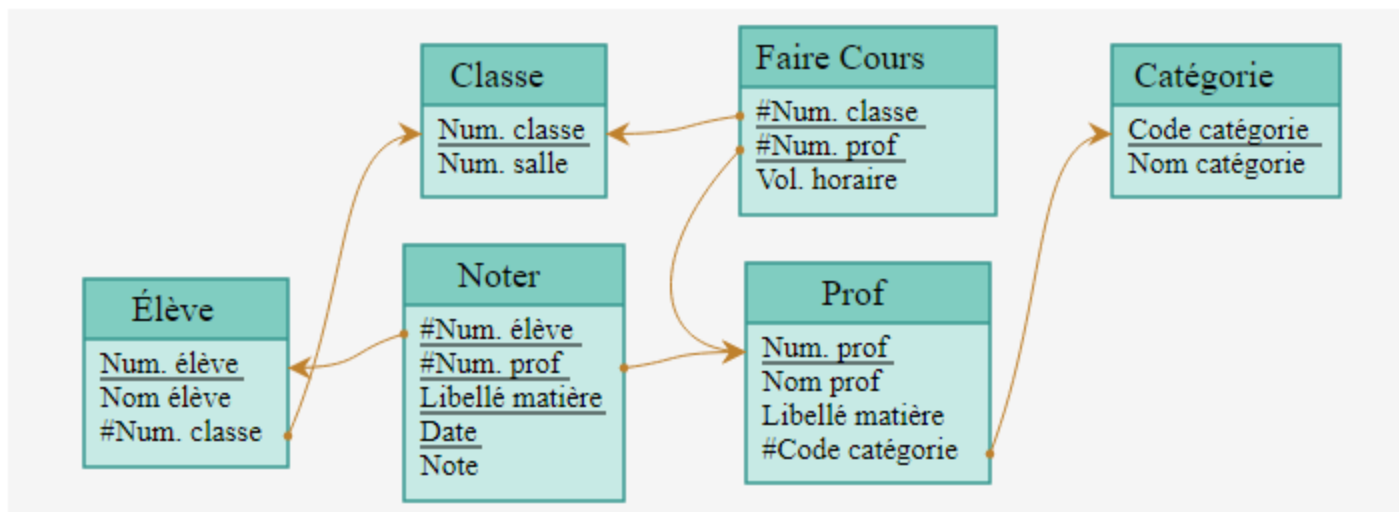
### Modèle Logique de Données

- Spécifique au modèle relationnel
- Uniquement constitué de table :
  - Ensemble structuré de données organisées en tableau
  - Colonne = Données ayant la même structure et représentant la même grandeur (date de naissance, quantité, etc.)
  - Ligne = Ensemble des propriétés associées entre elle pour former une entité concrète de gestion = enregistrement ou tuple
- Etape essentielle : On définit la structure de la base et des données de manière indépendante du SGBD
- Représentation abstraite mais précise de la manière dont les données sont organisées et interconnectées,
- Une fois le MLD défini, il peut être converti en un schéma de base de données spécifique au SGBD



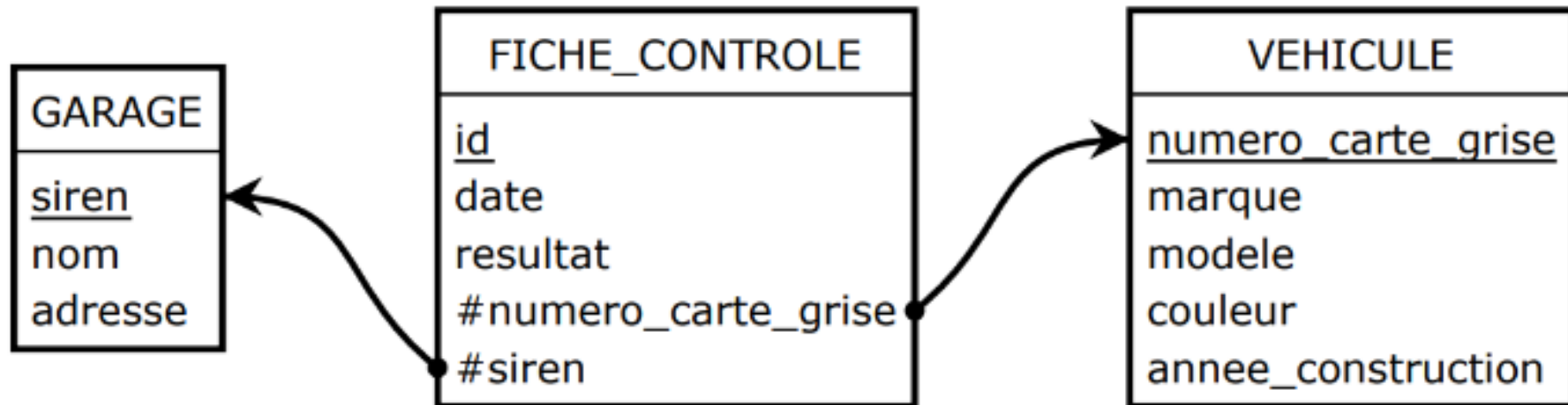
# Représentation

- Convention d'écriture :
  - - Clé primaire
  - #Clé\_étrangère
- Représentation graphique :
- Représentation textuelle :
  - Etudiant (INE, nom, #Num\_classe
- Pour les attributs, on essaie d'éviter les accents, les espaces (donc pas comme dans l'exemple présenté ici)



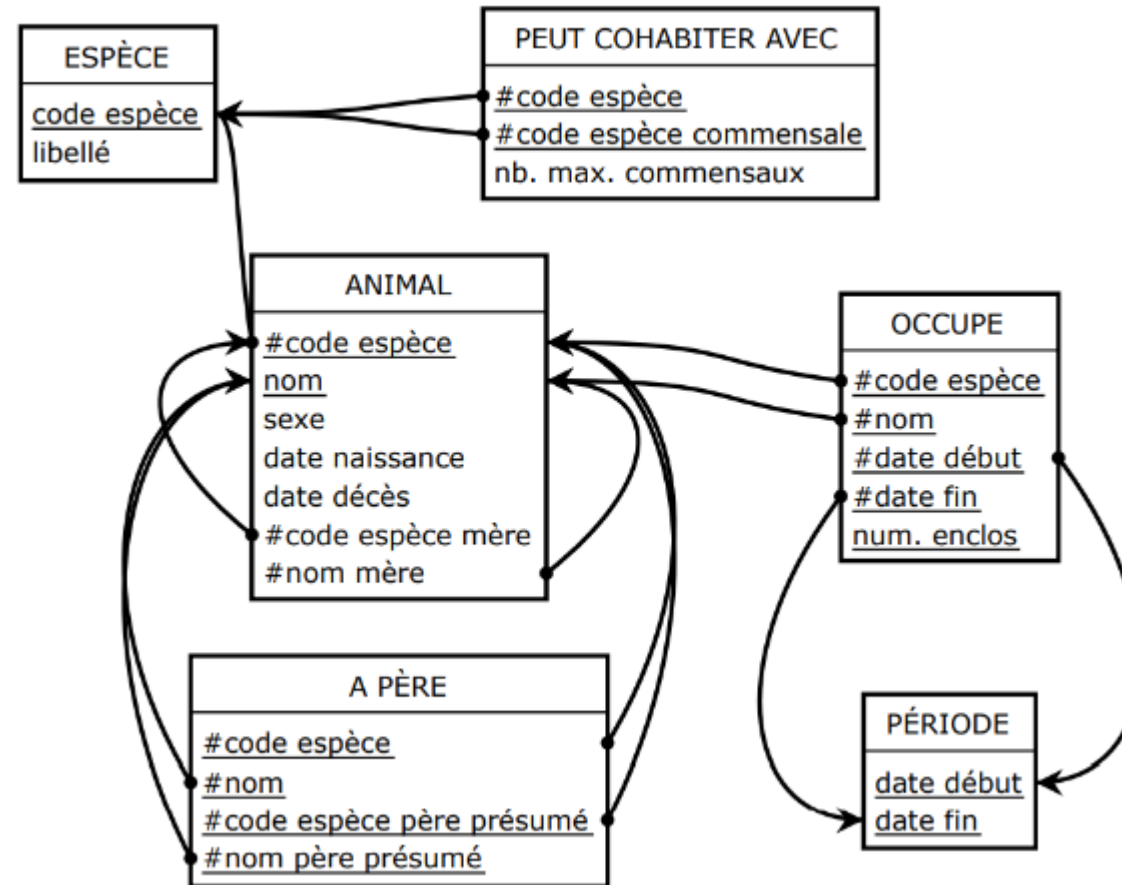
## Le MLD

### Exemples



## Le MLD

### Exemples



## Le MLD

### Passage du MCD au MLD - Propriétés

Element du MCD	Element du MLD
Entité	Table
Attribut	Colonnes
Identifiant	Clé primaire
Association (1,1), (1, 1) Association (1,1) , (1, N)	Relation clé primaire / clé étrangère
Association (1, N), (1, N)	Table supplémentaire

- Concernant les tables supplémentaires :
  - La clé primaire est composée des deux clés étrangère référençant les entités de l'association
  - Les attributs de l'association deviennent des colonnes de cette nouvelle table

## Le MLD

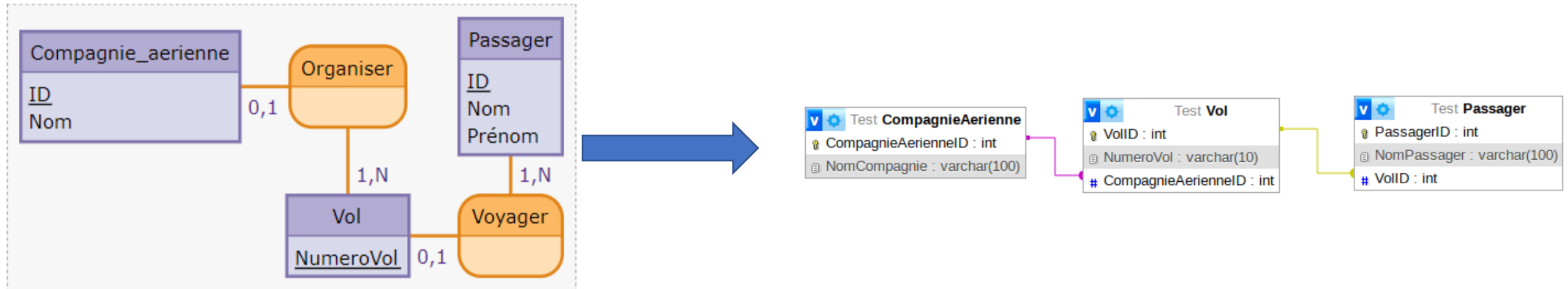
### Passage du MCD au MLD - Exemples





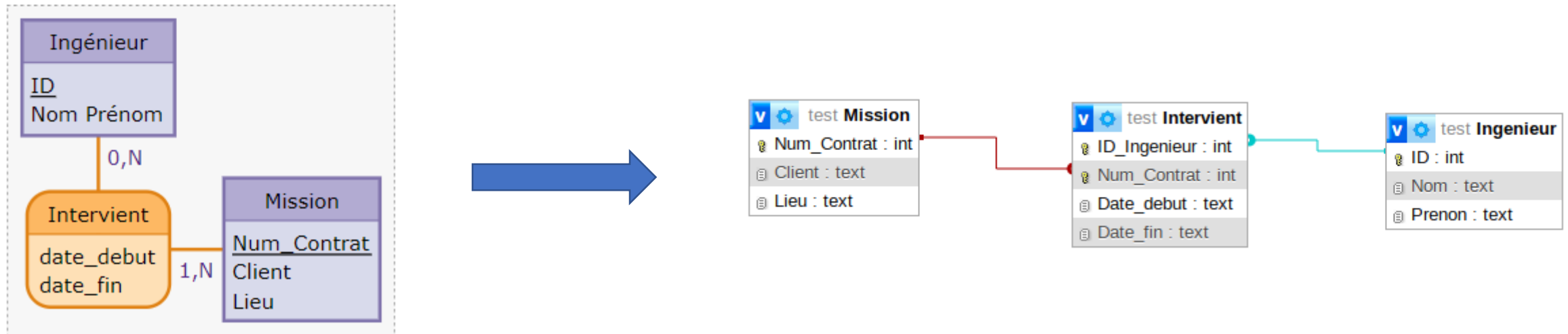
## Le MLD

### Passage du MCD au MLD - Exemples



## Le MLD

### Passage du MCD au MLD - Exemples



# Le SQL

- Langage standard des SGDB-R, créée en 1974, normalisé en 1986

- Fonctionnalités :
  - Définir les données : Langage de Définition des Données
  - Manipuler les données : Langage de Manipulation des données
  - Protéger les données : Langage de contrôle des données

- C'est un langage de haut niveau (verbeux) et déclaratif :

## On décrit le « quoi » et pas le « Comment »

- Différents dialectes en fonction des SGDB-R : sqlite, postgre, Microsoft SQL

- Simple et lisible, intuitif

- Requêtes simples mais également plus complexes pour créer, modifier, gérer efficacement la base de données

- En SQL on n'écrit pas des scripts / programme mais des **requêtes** !7

- Une requête agit sur la base, son effet est permanent

- Langage ni compilé, ni interprété

```
1  -- Créer une nouvelle base de données
2  CREATE DATABASE MaBaseDeDonnees;
3
4  -- Sélectionner la base de données nouvellement créée
5  USE MaBaseDeDonnees;
6
7  -- Créer une table "Utilisateurs" avec quelques colonnes
8  CREATE TABLE Utilisateurs (
9      ID INT AUTO_INCREMENT PRIMARY KEY,
10     Nom VARCHAR(50),
11     Prenom VARCHAR(50),
12     Age INT
13 );
14
15 -- Insérer quelques enregistrements dans la table "Utilisateurs"
16 INSERT INTO Utilisateurs (Nom, Prenom, Age) VALUES
17     ('Doe', 'John', 30),
18     ('Smith', 'Alice', 25),
19     ('Johnson', 'Bob', 35);
```

# Le SQL

## Fonctions

- **Langage de définition des données :**
    - Décrire l'organisation des données
    - Créer / supprimer / modifier des tables, vues, index de la base
  - **Langage de Manipulation des données :**
    - Insérer des données dans une table
    - Modifier tout ou partie des données d'une table
    - Supprimer des données, avec ou sans condition
  - **Administration de la base :**
    - Contrôler l'accès
    - Optimiser
    - Sauvegarder
    - Exporter
    - Importer
1. Insérer des données dans une table :


```
sql

-- Insérer un nouvel enregistrement dans la table "Clients"
INSERT INTO Clients (ID, Nom, Email, Telephone)
VALUES (1, 'Doe', 'doe@example.com', '123-456-7890');
```

2. Modifier des données d'une table :

### 1. Décrire l'organisation des données :


```
sql
```

 Copy code

```
-- Décrire la structure d'une table existante  
DESC NomDeLaTable;
```

## 2. Créer une table :

```
sql
```

 Copy code

```
-- Créer une nouvelle table "Clients" avec des colonnes  
CREATE TABLE Clients (  
    ID INT PRIMARY KEY,  
    Nom VARCHAR(50),  
    Email VARCHAR(100)  
);
```

### 1. Insérer des données dans une table :

```
-- Insérer un nouvel enregistrement dans la table "Clients"
INSERT INTO Clients (ID, Nom, Email, Telephone)
VALUES (1, 'Doe', 'doe@example.com', '123-456-7890');
```

## 2. Modifier des données dans une table :

```
-- Mettre à jour l'email du client avec ID 1
UPDATE Clients
SET Email = 'john.doe@example.com'
WHERE ID = 1;
```

# Le SQL

## Syntaxe

- Une requête se termine toujours par un ;
- Langage non sensible à la casse, mais certaines conventions :
  - Mots clés en MAJUSCULE
  - Eviter les noms de tables ou de colonne contenant des accents ou des espaces
  - Les chaînes de caractères sont entre simple quote
  - Indentation non nécessaire, mais fortement recommandée pour la lisibilité !

```
-- Créer une table "Produits"
CREATE TABLE Produits (
    ID INT PRIMARY KEY,
    Nom VARCHAR(100),
    Prix DECIMAL(10, 2),
    Stock INT
);

-- Insérer quelques enregistrements dans la table "Produits"
INSERT INTO Produits (ID, Nom, Prix, Stock)
VALUES (1, 'Ordinateur portable', 899.99, 10),
       (2, 'Smartphone', 499.99, 20),
       (3, 'Écran LCD 27 pouces', 299.99, 15);

-- Mettre à jour le prix du produit "Smartphone"
UPDATE Produits
SET Prix = 449.99
WHERE Nom = 'Smartphone';
```

# Le SQL

## Types de données

- **Pour chaque colonne on devra choisir un type de données adéquat**
  - Contrainte d'intégrité : Si la données est numérique, on ne pourra saisir que des nombres !
  - Souvent dépendant du SGBD utilisé
- Dans le cadre de ce module on se contentera des types suivants :

```

SQL 1
1  INT (INTEGER) -- Nombre entier, ex 42
2  REAL -- Nombre flottant, ex 3.14
3  TEXT (VARCHAR) -- Chaine de caractère
4  NULL -- Absence de valeur
5

```

- NULL représente l'absence de valeur, c'est-à-dire que pour une ligne d'une table, il est possible que certaines colonnes n'aient pas de valeur. Pour certaines colonnes en revanche c'est obligatoire, on déclare alors la colonne comme NOT NULL

```

SQL 1
1  CREATE TABLE Etudiant (
2      INE INT NOT NULL PRIMARY KEY,
3      Nom TEXT NOT NULL,
4      Prenom TEXT NOT NULL,
5      Email TEXT);

```

# Le SQL

## Opérateur et fonctions

Opérateur / Fonctions	Utilisation
+, -, *, /, %	Addition, soustraction, multiplication, division, modulo
=, <>, <, <=, >, >=	Egal, différent, inférieur, ect
BETWEEN X and Y	Compris dans [X, Y]
IN (a, b, c) NOT IN(...)	Parmi a, b, c ou l'inverse
OR, AND, NOT	Opérateur booléen
LIKE	Texte similaire
IS NULL, IS NOT NULL	Comparaison avec NULL
MIN, MAX, AVG, SUM	Min, Max, Moyenne, Somme

# Le SQL

## Exemples

```

1  -- Créer une table pour les étoiles
2  CREATE TABLE Etoiles (
3      ID INT PRIMARY KEY,
4      Nom VARCHAR(100),
5      Type VARCHAR(50),
6      Masse FLOAT,
7      Luminosite FLOAT
8  );
9
10 -- Insérer des données sur les étoiles
11 INSERT INTO Etoiles (ID, Nom, Type, Masse, Luminosite)
12 VALUES
13     (1, 'Sol', 'Naine jaune', 1.0, 1.0),
14     (2, 'Sirius', 'Étoile principale de séquence', 2.0, 25.0),
15     (3, 'Vega', 'Étoile blanche de la séquence principale', 2.1, 50.0);
16
17 -- Créer une table pour les planètes
18 CREATE TABLE Planetes (
19     ID INT PRIMARY KEY,
20     Nom VARCHAR(100),
21     Type VARCHAR(50),
22     DistanceDuSoleil FLOAT,
23     Diametre FLOAT
24 );
25

```

```

49 -- Sélectionner des étoiles de type "Naine jaune"
50 SELECT * FROM Etoiles WHERE Type = 'Naine jaune';
51
52 -- Sélectionner des planètes de type "Tellurique"
53 SELECT * FROM Planetes WHERE Type = 'Tellurique';
54

```