

Gestion des données

Modèle Relationnel

HEI 2021 / 2022

Introduction

- Modèle Relationnel

- 1 seule structure : **la relation**
 - Représentée sous forme de *table*
 - Une table possède un nom et des colonnes représentées par un nom **d'attribut**
 - Chaque colonne contient des valeurs appartenant à un **domaine**
 - Une ligne de la table (ou tuple) correspond à une entité
 - Une schéma relationnel (MLD) est un ensemble de schéma de relations

Définitions

- Domaine
 - Ensemble de valeurs possible pour un attribut
- Attributs
 - Décrit le contenu d'une colonne d'une table
- Schéma de relation
 - Nom d'une table suivie de ses attributs et de leur domaine
- Clé primaire
 - Plus petit sous-ensemble d'attributs qui permet d'identifier une ligne de manière unique

Exemple

- Soit la relation FILM suivante:

FILM		
Titre	Année	Réalisateur
Pi	1998	Aronofsky
Camping	2006	Onteniente
Le Parrain	1972	Copola

- Le schéma de relation associé est :

FILM (Titre: varchar, Année: int, Realisateur: varchar)

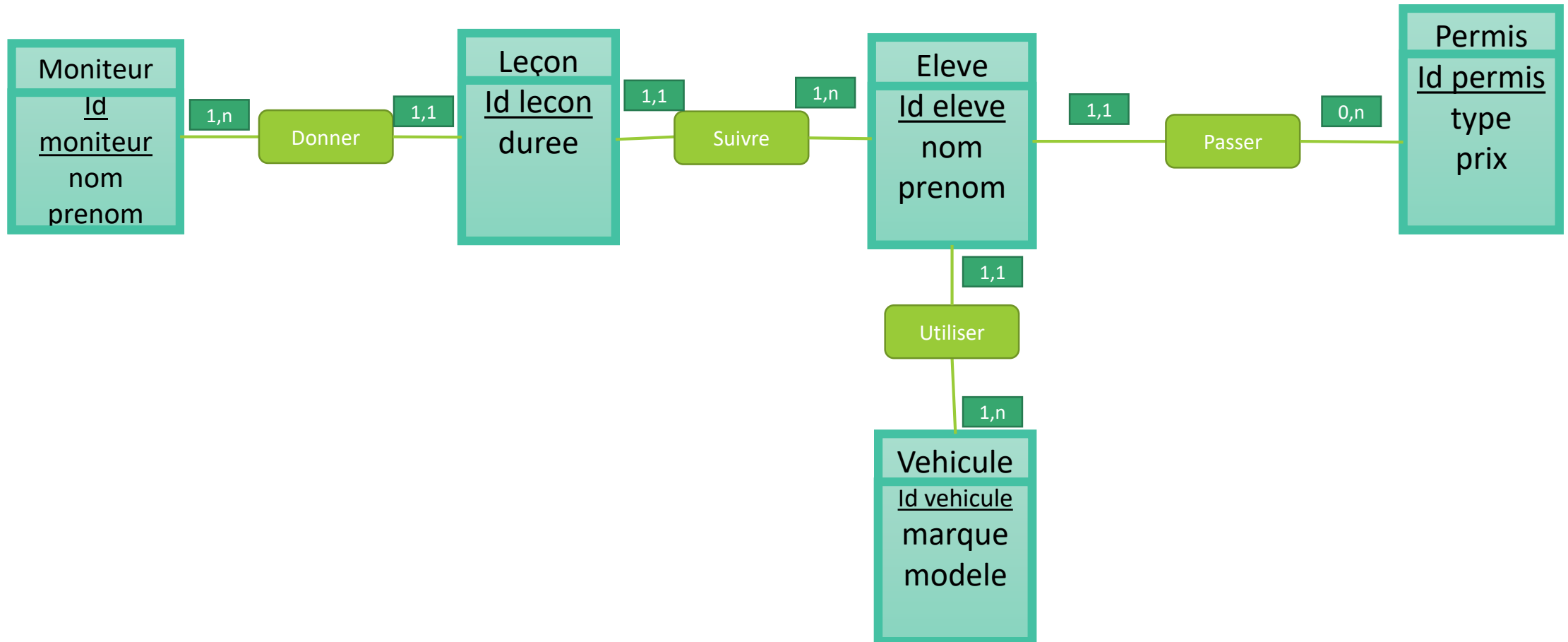
Passage d'un MCD à un MLD

- Objectif : Passer d'un modèle à 2 structures (entité et associations) vers un modèle à une structure (relation)
- Application de différentes règles.

Passage d'un MCD à un MLD

- Règle de passage des entités
 - Pour chaque entité du MCD on crée une relation :
 - Du même nom que l'entité
 - Chaque attribut de l'entité devient attribut de la relation
 - L'identifiant de l'entité devient la clé de la relation

Rappel : MCD – Auto-école



Exemple Auto-école

- A partir du MCD auto-école on obtient les relations suivantes :
 - Moniteur(id moniteur, nom, prenom)
 - Lecon(id lecon, duree)
 - Eleve(id eleve, nom, prenom)
 - Permis(id permis, type, prix)
 - Vehicule(id vehicule, marque, modele)

Passage d'un MCD à un MLD

- Règle de passage des associations
 - Il existe 3 types d'associations qui se basent sur la cardinalité maximale de chaque côté de l'association.
 - On a donc les 3 associations suivantes :
 - De type 1-1
 - De type 1-n
 - De type n-n

Passage d'un MCD à un MLD

- Association 1-N / 1-1
 - Pour une association 1-N entre 2 entités A et B la règle de passage au modèle logique est la suivante :
 - On crée les relations A et B
 - L'identifiant de B devient un attribut de la relation A
 - L'attribut « identifiant de B » dans la relation A est appelé **clé étrangère**
- Clé étrangère : Une clé étrangère dans une relation est formée d'un ou plusieurs attributs formant une clé primaire dans une autre relation
- Dans le cas d'une association 1-1 la clé étrangère peut être ajoutée dans l'une des 2 relations au choix

Exemple Auto-école

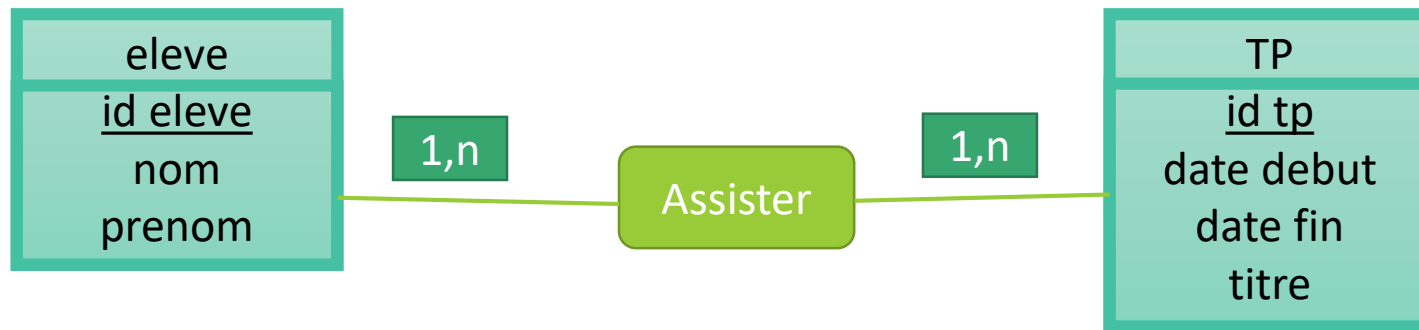
- A partir du MCD auto-école on obtient les relations suivantes :
 - Moniteur(id moniteur, nom, prenom)
 - Lecon(id lecon, duree, *id moniteur*, *id eleve*)
 - Eleve(id eleve, nom, prenom, *id vehicule*, *id permis*)
 - Permis(id permis, type, prix)
 - Vehicule(id vehicule, marque, modele)

Passage d'un MCD à un MLD

- Association N-N
 - Pour une association N-N entre 2 entités A et B la règle de passage au modèle logique est la suivante :
 - On crée les relation A et B
 - On crée une relation A-B pour l'association
 - On ajoute les clés primaires des relations A et B en tant qu'attributs de la relation A-B
 - Les attributs de l'associations deviennent attributs de cette relation

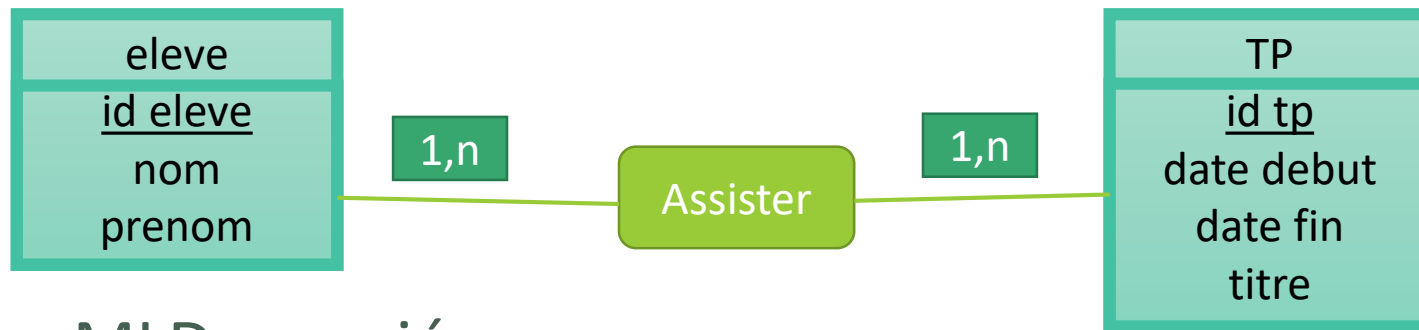
Exemple de passage pour une association N-N

- Soit le MCD suivant :



Exemple de passage pour une association N-N

- Soit le MCD suivant :



- Le MLD associé :

Eleve(id eleve, nom, prenom)

TP(id tp, date debut, date fin, titre)

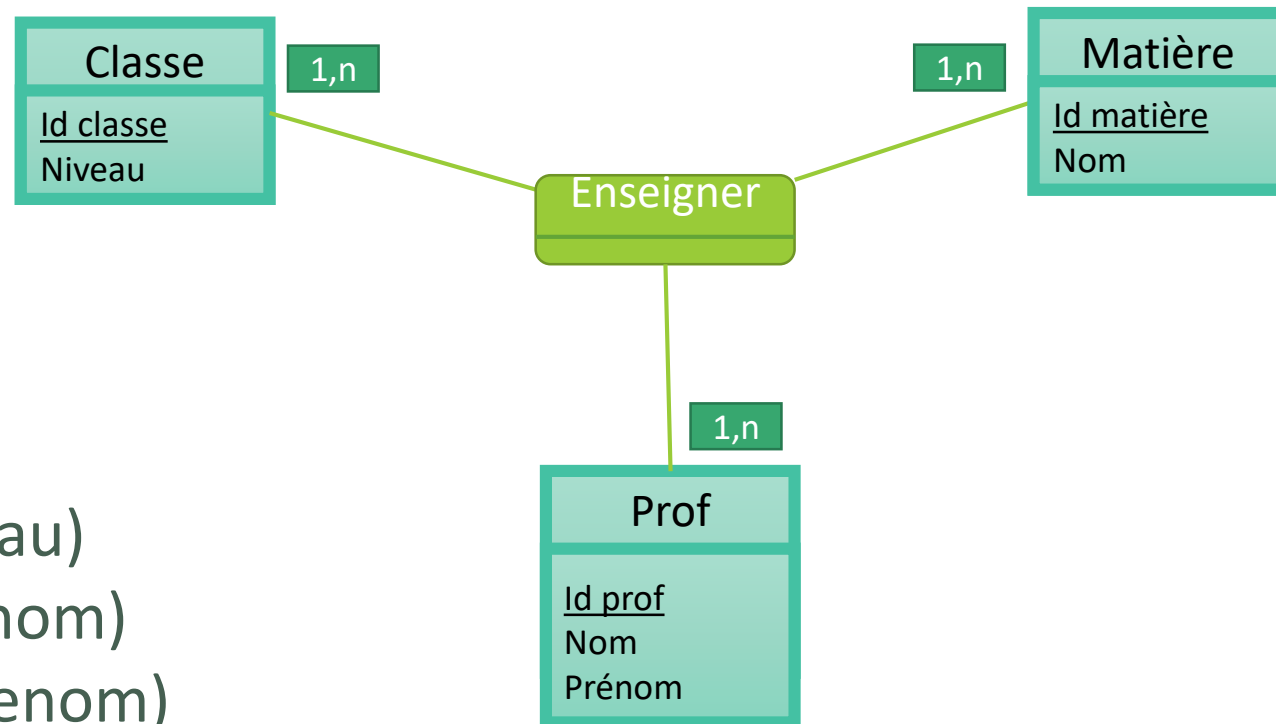
SuiviTp(*id eleve*, *id tp*)

Passage d'un MCD à un MLD

- Relation n-aire avec $n > 2$
 - Une association qui associe plus de 2 entités est représentée par une relation
 - Cette relation contient les identifiants des entités impliquées

Exemple de passage pour une association ternaire

- Soit le MCD suivant :



- Le MLD associé :
 - Classe(id classe, niveau)
 - Matière(id matière, nom)
 - Prof(id prof, nom, prenom)
 - Enseignement(*id classe*, *id matière*, *id prof*)

Langage de définition des données LDD

- SQL : Structured Query Language
 - Langage qui permet la communication avec une BDD relationnelle
 - 2 domaines :
 - LDD : langage de définition des données, permet de décrire la structure des objets et leurs relations.
 - LMD : langage de manipulation des données, permet d'ajouter, lire, modifier, et supprimer des objets.

Création d'une table

- Structure de données composée de :
 - Colonnes : les attributs d'une relation
 - Lignes : occurrences d'une relation
- Chaque colonne doit avoir :
 - Un nom
 - Un type

Création d'une table - Syntaxe

- **CREATE TABLE** permet d'ajouter une nouvelle table dans la base de données

```
CREATE TABLE [nom] (  
    [colonne 1],  
    [colonne 2],  
    [colonne 3]  
);
```

Définition des colonnes

- Pour chaque colonne, il est nécessaire de spécifier son nom ainsi que son type. Les autres caractéristiques sont facultatives.

```
CREATE TABLE person (  
    name          VARCHAR (50) ,  
    firstname     VARCHAR (50) ,  
    birthdate     DATE ,  
    admin         TINYINT  
);
```

Types de données (non exhaustif)

Type	Description	Remarque
varchar(n)	Chaine de caractères de taille variable (max n caractères)	
int	Nombre entier de 32 bits	De 0 à 4 294 967 295 en non signé De -2 147 483 648 to 2 147 483 647 en signé
tinyint	Nombre entier de 8 bits	De 0 à 255 en non signé De -128 à 127 en signé
float	Nombre à virgule flottante	
decimal(n,d)	Nombre à virgule de n chiffres dont d après la virgule	
date	Date	
datetime	Date et heure	
blob	Données binaires (4Go max)	

Les contraintes

- Contrainte de colonne
 - Associée à une colonne
- Contrainte de table
 - Associée à plusieurs colonnes ou la table entière

Les contraintes de colonnes

- Nullité de colonne : **NULL / NOT NULL**
- Unicité de valeur : **UNIQUE**
- Clé primaire : **PRIMARY KEY**
- Clé étrangère : **REFERENCES**
- Valeur par défaut d'une colonne : **DEFAULT**

Les contraintes de colonne - exemple

```
CREATE TABLE person (  
    id          INT PRIMARY KEY,  
    name        VARCHAR(50) NOT NULL,  
    firstname   VARCHAR(50) NOT NULL,  
    birthdate   DATE NULL,  
    admin       TINYINT NULL DEFAULT 0,  
    type_id     INT REFERENCES type(id)  
);
```


Auto-incrément

- Très fréquemment, la clé primaire des objets va être un champ technique géré par la base de données elle-même.
- Si aucune valeur n'est spécifiée pour cette colonne, la valeur sera un incrément de la valeur existante maximale.

Les contraintes de colonne - exemple

```
CREATE TABLE person (  
    id          INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    name        VARCHAR(50) NOT NULL,  
    firstname   VARCHAR(50) NOT NULL,  
    birthdate   DATE NULL,  
    admin       TINYINT NULL DEFAULT 0,  
    type_id     INT REFERENCES type(id)  
);
```

Les contraintes de tables

- **PRIMARY KEY** (col1, col2, ...)
 - Désigne les colonnes formant la clé primaire
- **UNIQUE** (col1, col2, ...)
 - Force le couple (col1, col2, ...) à être unique
- **FOREIGN KEY** (col1, col2, ...) **REFERENCES** table (col1, col2, ...)

Les contraintes de tables - exemple

```
CREATE TABLE person (  
    name          VARCHAR(50) NOT NULL,  
    firstname     VARCHAR(50) NOT NULL,  
    birthdate     DATE NULL,  
    admin         TINYINT NULL DEFAULT 0,  
    type_id       INT,  
    CONSTRAINT pk_person PRIMARY KEY(name, firstname),  
    CONSTRAINT fk_type  
        FOREIGN KEY (type_id) REFERENCES type(id)  
);
```

Autres opérations

- Modification de table
 - L'instruction **ALTER TABLE** permet de modifier une table. Elle consiste en un ensemble d'action sur les colonnes ou clés d'une table.
 - Syntaxe :
 - ALTER TABLE nomTable [Add][Modify][Drop]

Autres opérations

- Modification de table

```
ALTER TABLE rendu_tp  
  DROP COLUMN id_auteur,  
  ADD COLUMN rendu_tpcol VARCHAR(45) NULL,  
  DROP INDEX fk_rendu_auteur ;
```

```
ALTER TABLE rendu_tp  
  ADD CONSTRAINT fk_prof  
  FOREIGN KEY (id_prof) REFERENCES prof(id) ;
```

Autres opérations

- Destruction de table
 - L'instruction **DROP TABLE** permet de supprimer une table.
 - Syntaxe :
 - DROP TABLE nomTable;
 - Il n'est pas possible de supprimer une table si elle est la cible d'une clé étrangère

Auto école

- LDD:
 - Moniteur(id moniteur, nom, prenom)
 - Lecon(id lecon, duree, *id moniteur*, *id eleve*)
 - Eleve(id eleve, nom, prenom, *id vehicule*, *id permis*)
 - Permis(id permis, type, prix)
 - Vehicule(id vehicule, marque, modele)

Auto école

- MLD :

```
CREATE TABLE Moniteur(  
    id_moniteur INT PRIMARY KEY AUTO_INCREMENT,  
    nom VARCHAR(80),  
    prenom VARCHAR(80));  
CREATE TABLE Lecon(  
    id_lecon INT PRIMARY KEY AUTO_INCREMENT,  
    duree INT,  
    id_moniteur INT REFERENCES Moniteur(id_moniteur),  
    id_eleve INT,  
    CONSTRAINT fk_eleve (id_eleve) REFERENCES  
    eleve(id_eleve));
```

Auto école

- MLD :

```
CREATE TABLE Eleve (  
  id_eleve INT PRIMARY KEY AUTO_INCREMENT,  
  nom VARCHAR(80),  
  prenom VARCHAR(80),  
  id_ve REFERENCES vehicule(id_vehicule),  
  id_permis REFERENCES permis(id_permis));
```

```
CREATE TABLE Permis (  
  id_permis INT PRIMARY KEY AUTO_INCREMENT,  
  type INT,  
  prix INT);
```

Auto école

- MLD :

```
CREATE TABLE Vehicule(  
    id_vehicule INT PRIMARY KEY AUTO_INCREMENT,  
    marque VARCHAR(60) ,  
    modele VARCHAR(30) ) ;
```

Exercice

- Créer le MLD à partir du MCD bibliothèque
- Ecrire le LDD correspondant