

BASES DE DONNÉES

COURS:

**LE MODÈLE RELATIONNEL
& INTRODUCTION À “SQL”**

LE MODÈLE RELATIONNEL

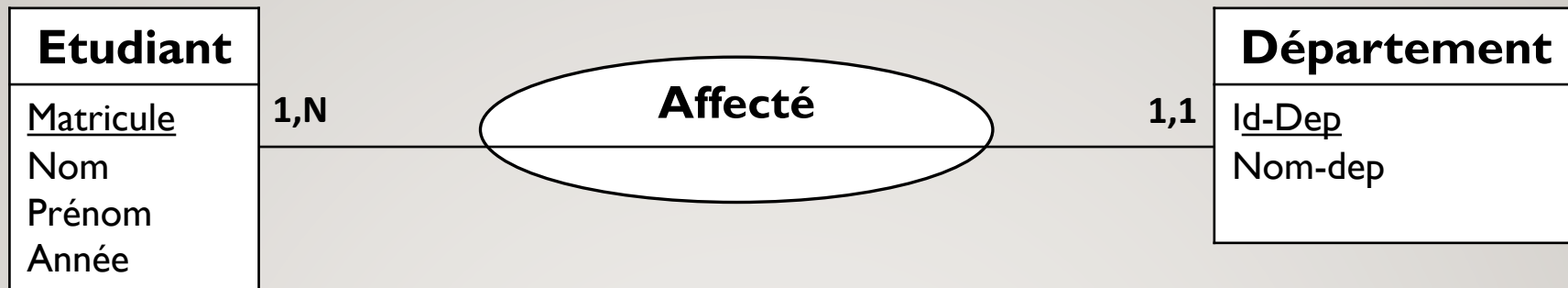
- Le modèle relationnel est un modèle logique associé aux SGBD relationnels
- **Exemples de SGBD-R:** Oracle, BD2, SQLServer, Access, Paradox, dBase,
- **Objectifs du modèle relationnel:**
 - Indépendance physique
 - Traitement du problème de redondance des données
 - LMD non procéduraux (facile à utiliser)
 - Devenir un standard.

CONCEPTION DE BD RELATIONNELLE

- La difficulté de conception de bases de données :
- conception non intuitive
- **Objectif de BD Relationnelle:**
- Éviter la redondance de données
- Assurer la cohérence des données
- Permettre des requêtes

LE SCHEMA RELATIONNEL

- ❖ Des données rangées dans des **tables**
- ❖ Cases **monovaluées**
- ❖ Des **contraintes** sur les colonnes des tables.
- ❖ **Exemple de contraintes :**
 - Clé primaire
 - Clé étrangère



Clé
Primaire

Clé
Etrangère

Clé
Primaire

Matricule	Nom	Prénom	Année	Dep
I234	L2	02
I238	L3	03
I236	M2	02
I239	L2	03
I233	L1	02

Id-Dep	Nom-Dep
02	MI
03	SNV
05	SM

Table Département

Table Etudiant

Modèle Entité-Association

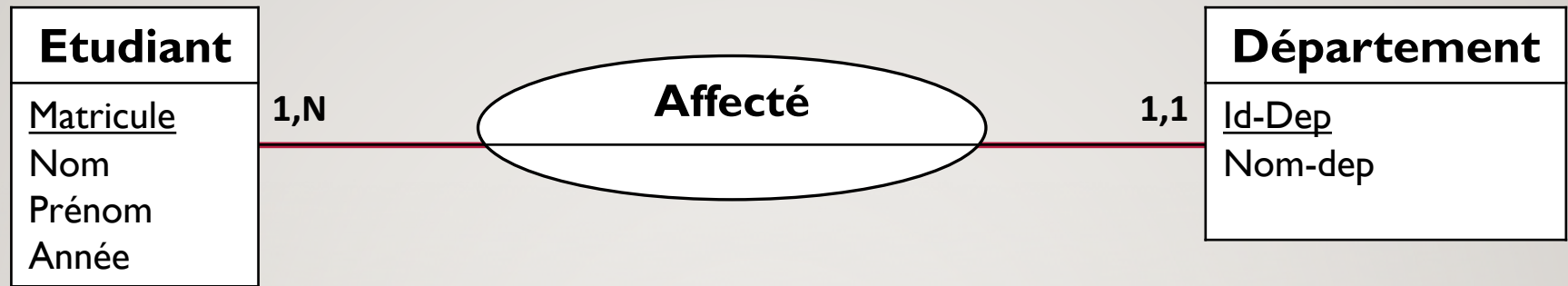


Schéma Relationnel

- **Etudiant** (Matricule, Nom, Prénom, Année, #Dep)
- **Département** (Id-Dep, Nom-dep)

Schéma Relationnel

- **Clé Primaire:** Permet d'accéder aux données de façon claire et unique (*une ou plusieurs colonnes*) soulignée (convention usuelle)
- **Clé étrangère:** Lien avec l'identifiant d'un autre schéma, i.e fait **Référence** à une autre table. **Précédée d'un #**
- **Attention:** Pas de clé étrangère dans un modèle **E/A**

Table ou Relation

- Ensemble d'occurrence structurées par colonnes.
- L'ordre des lignes et des colonnes n'est pas important
- Exemple: **Table Etudiant**

Ligne =

Enregistrement =

Objet =

Tuple =

Occurrence =

Colonne = Champ = Attribut = Propriété = Caractéristique

Matricule	Nom	Prénom	Année	Dep
I234	L2	02
I238	L3	03
I236	M2	02
I239	L2	03
I233	L1	02

Donnée = Valeur de l'attribut prise dans un domaine de valeur

Domaine = Ensemble de valeurs possibles d'un champ

Schéma de la Relation

• **R** (**A1, A2, A3,, An**)

- **R** : Nom de la relation
- **A1, A2, A3, ...An**: Noms des attributs de la relation
- **(a_{1i}, a_{2j},, a_{nk})** : n-uplet
- **n**: degré de la relation

• Exprimé de manière générale sous la forme :

NomTable (**nomChamp1, nomChamp2, nomChamp3,...**)

– Exemple : **Etudiant** (**Num**, **Nom**, **Prenom**, **Date_n**)

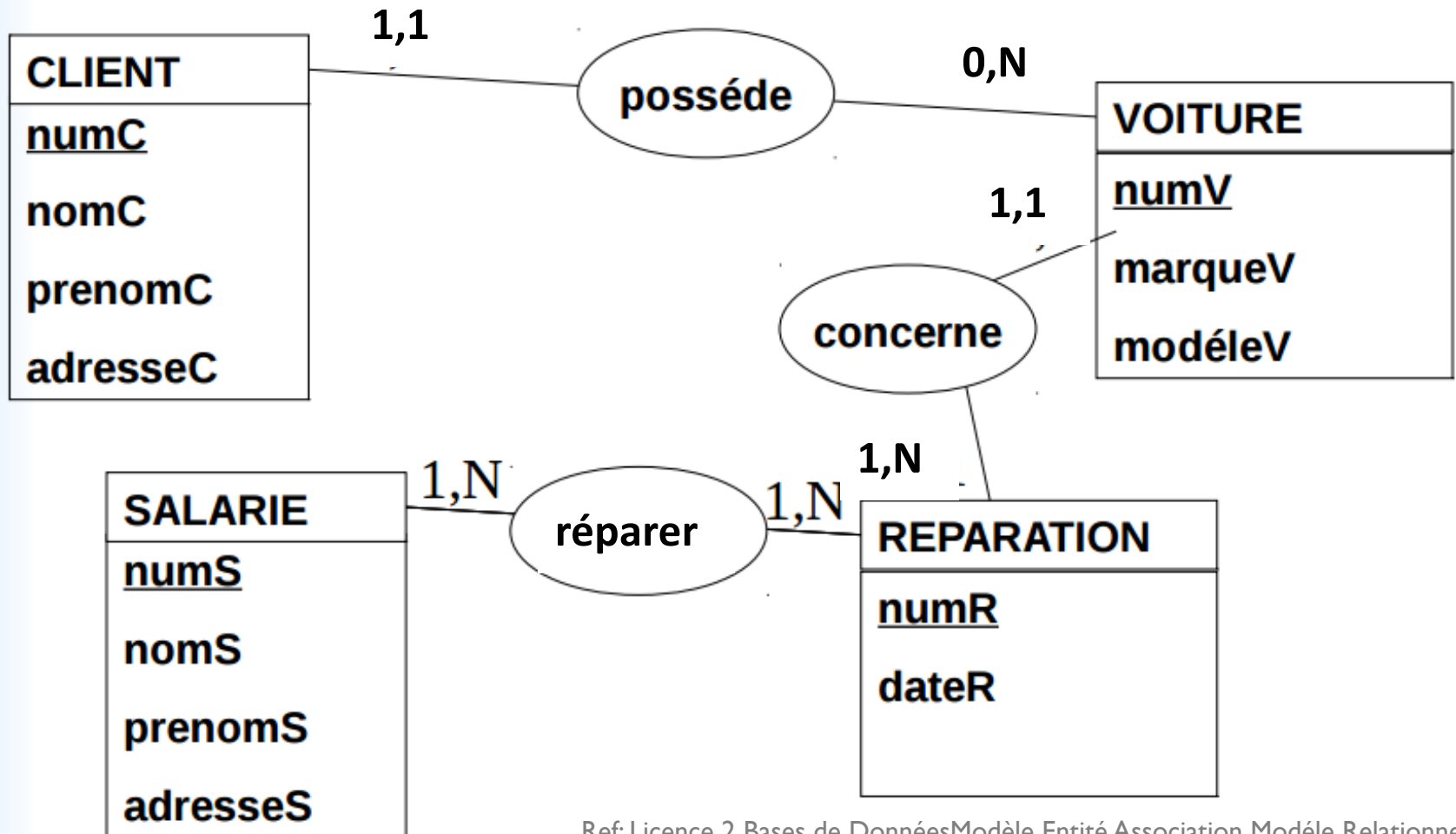
Schéma Relationnel d'une BD

Exemple de gestion de garage:

- **CLIENTS** (numC, nomC, prenomC, adresseC)
- **SALARIE** (numS, nomS, prenomS, adresseS, dateNaissS)
- **VOITURE** (numV, marqueV, modeleV, #numC)
 - numC référence numC (CLIENTS)
- **REPARATION** (numR, dateArriveeR, #numV)
 - numV référence numV (Voiture)
- **REPARER** (#numS, #numR, nbHeuresTravail)
 - numS référence numS (SALARIE)
 - numR référence numR (REPARATION)

Schéma Entité Association

- Le schéma Entité Association qui a permis de créer le schéma relationnel précédent



Clé Primaire. --- Primary Key (PK)

- ❖ La clé primaire est un identificateur qui permet de différencier 2 enregistrements dans la même table d'une façon **unique** et **claire**.
- ❖ La clé peut être:
 - **Naturelle** : Exemple : Matricule d'un étudiant, ...
 - **Artificielle** : Exemple: numéro incrémenté automatiquement.
- ❖ Parfois un choix est effectué parmi plusieurs clés possibles (**clés candidates**).
- ❖ Type de clé :
 - **Clé simple**: Un seul attribut.
 - **Clé Composée**: Appelée aussi **Multiple**, **Concaténée**, elle est composée de plusieurs attributs.
- ❖ La clé doit être **stable** dans le temps, Exemple: **âge**, **poids**, ... ne peuvent pas être une clé.
- ❖ Par convention est **soulignée** dans un schéma relationnel

Clé Étrangère --- Foreign Key (FK)

- ❖ La clé étrangère fait la liaison entre **2 tables** d'une base de données.
- ❖ La **clé étrangère** de la **Table T1** est une **clé primaire** de la **Table T2**
- ❖ La clé **étrangère** dans la table **T1** doit être de **type compatible** à la table de **référence T2**

Table Etudiant

Matricule	Nom	Prénom	Année	Dep
I234	L2	02
I238	L3	03
I236	M2	02
I239	L2	03
I233	L1	02

Table Département

Id-Dep	Nom-Dep
02	MI
03	SNV
05	SM

Clé
Etrangère

Compatible

Clé
Primaire

CONTRAINTES D'INTEGRITÉ

❖ Les **contraintes d'intégrité** sont des **contrôles** effectués lors de la **saisie**, la **modification** ou la **suppression** des données.

❖ **Contraintes d'intégrité:**

➤ **Clé Primaire**

➤ **Valeur Nulle**

➤ **Clé Étrangère**

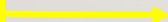
Contrainte 1:

Clé Primaire --- Primary Key (PK)


- Unicité de la valeur dans la(es) colonne(s) constituant la clé

Table Etudiant

Clé
Primaire



Matricule	Nom	Prénom	Année	Dep
1234	L2	02
1238	L3	03
1236	M2	02
1239	L2	03
1238	L1	02



Impossible: Valeur refusée lors de la saisie

Contrainte 2: Valeur Nulle --- Null Value

- Certaines lignes dans certaines colonnes peuvent ne pas contenir de valeurs.
- **Valeur par Défaut** : une valeur qui est attribuée à toutes les lignes de la colonne par défaut. Cette valeur peut être modifiée
- **Ensemble de valeurs** : Contrôle de l'appartenance de la valeur saisie à un intervalle ou bien à une liste de valeurs.

Matricule	Nom	Prénom	Année	Dep
I234	L2	02
I238			L3	03
I236		02
I239		L2	03
I238	L1	02

Contrainte 3: Clé Étrangère --- Foreign Key (FK)

- Vérification de la contrainte d'intégrité référentielle

Table Etudiant

Matricule	Nom	Prénom	Année	Dep
I234	L2	02
I238	L3	03
I236	M2	02
I239	L2	04
I233	L1	02

Table Département

Id-Dep	Nom-Dep
02	MI
03	SNV
05	SM

?

Valeur
inexistante

MODÈLE E/A

➤ Description de la Problématique :

- **Entités**
- **Associations**
- **Cardinalités**

MODÈLE RELATIONNEL

➤ Implémentation d'une Solution :

- **Relations**
- **Contraintes (Pk, FK, et d'autres ...)**

RÈGLES DE PASSAGE

SCHÉMA E/A → MODÈLE RELATIONNEL

❖ Règle 1 : Entité

- Chaque entité donne une table
- Son identifiant est la clé de la table

❖ Règle 2 : Association de type 1–N ou 1–1

- L'identifiant de l'entité **côté N** est ajoutée du **côté 1** où elle devient **clé étrangère**

❖ Règle 3 : Association de type N–M

- Création d'une nouvelle table dont la clé primaire est l'ensemble des identifiants des entités concernées
- Tout attribut de l'association devient attribut de la nouvelle table

Schéma Entité-Association: type 1-N / N-1

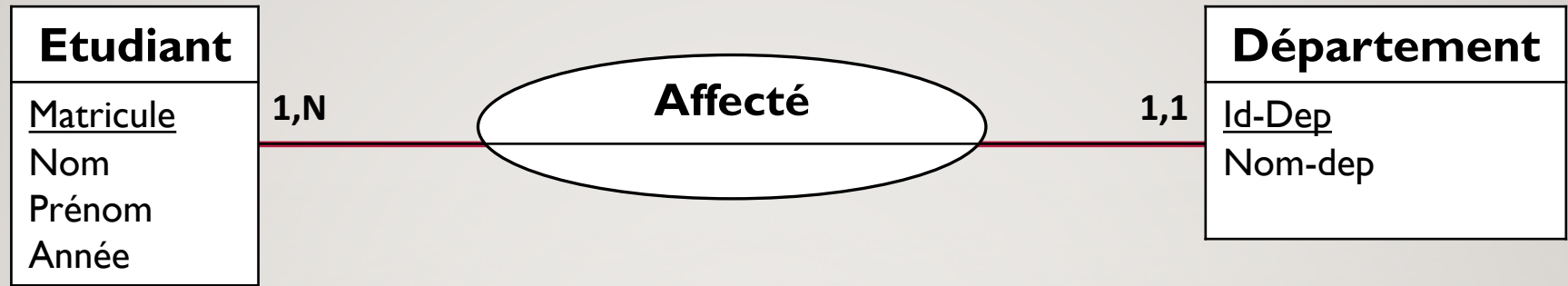


Schéma Relationnel

➤ **Etudiant** (Matricule, Nom, Prénom, Année, #Dep)

Dep référence Id-Dep (DEPARTEMENT)

➤ **Département** (Id-Dep, Nom-dep)

Schéma Entité-Association: type N-M / N-N

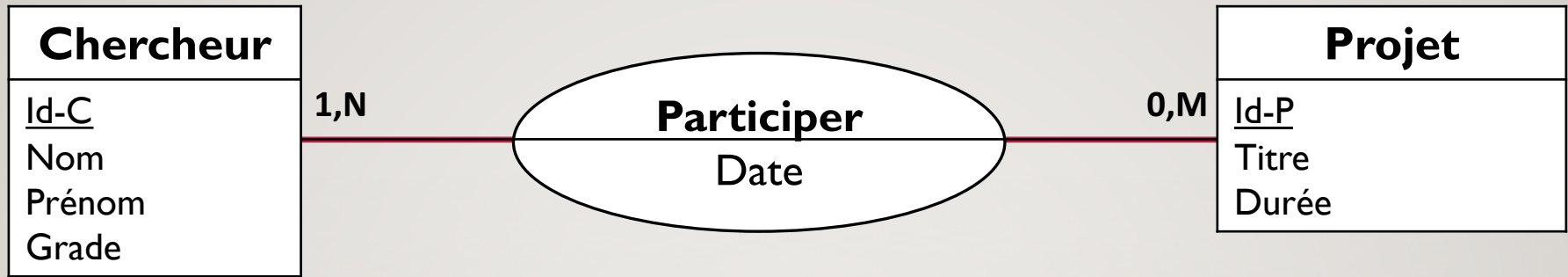


Schéma Relationnel

- **Chercheur** (Id-C, Nom, Prénom, Grade)
- **Projet** (Id-P, Titre, Durée)
- **Participer** (#Id-C, #Id-P, Date)

Id-C référence Id-C (Chercheur)

Id-P référence Id-P (Projet)

Schéma Entité-Association: type 1-1

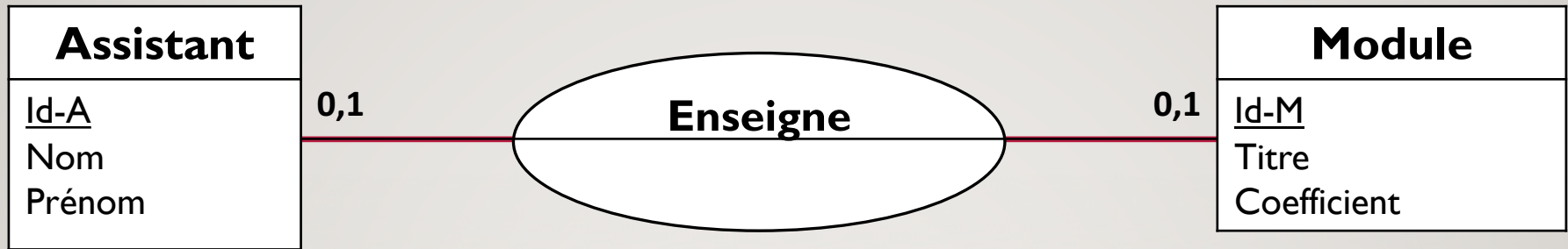


Schéma Relationnel

➤ **Assistant** (Id-A, Nom, Prénom, #Id-M)
Id-M référence Id-M (Module)

➤ **Module** (Id-M, Titre, Coefficient)

❖ Ou Bien:

➤ **Assistant** (Id-A, Nom, Prénom)

➤ **Module** (Id-M, Titre, Coefficient, #Id-A)

Id-A référence Id-A (Assistant)

Schéma Entité-Association: Relation Reflexive

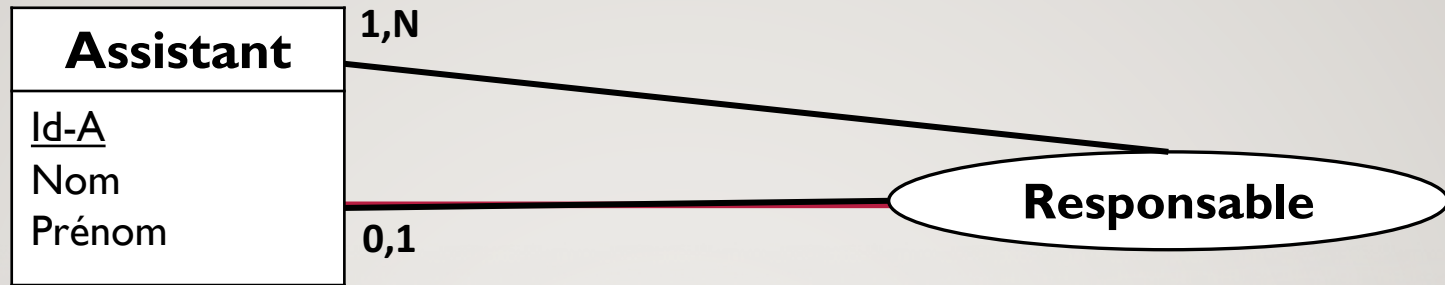


Schéma Relationnel

➤ **Assistant** (**Id-A**, Nom, Prénom, **#Chef**)

Chef référence **Id-A** (**Assistant**)

EXERCICE DE MODÉLISATION

- Une association gère des adhérents (nom, prénom, date de naissance) qui s'inscrivent à un choix d'activités (Bridge, Tricot, Judo...). Un tarif de cotisation annuelle est fixé pour chaque activité. Chaque activité a un seul responsable (nom, prénom) et un nombre de participants maximum. Pour chaque activité, on indique la date d'adhésion du participant et s'il a payé sa cotisation.
- **Donner le modèle E/A.**
- **Traduire le modèle E/A à un schéma relationnel**

LE LANGAGE SQL

- **SQL** = **S**tructured **Q**uery **L**anguage
- **SQL** **L**angage de **D**éfinition des **D**onnées (**LDD**)
 - SQL permet la **définition** des structures des tables et des autres objets de la base de données
- **SQL** **L**angage de **M**anipulation des **D**onnées (**LMD**)
 - SQL permet de faire la **manipulation** d'une BD : *Recherche (selection)*, et des *misés à jouret (Ajout, Suppression, Modification)* et le **contrôle** d'une base de données relationnelle..
- **SQL** se base sur l'*Algèbre Relationnelle*
- **SQL** est un standard de puis 1986

Création de la Base

- Création d'une table avec une seule **contrainte** : **NULL**

CREATE TABLE Nom_table

```
( Nom_Attr_1 TYPE [NOT NULL],  
  Nom_Attr_2 TYPE [NOT NULL], .....  
  Nom_Attr_n TYPE [NOT NULL] );
```

TYPE : Type de données
à choisir dans l'ensemble
des types proposés

Exemple: Création de
la Table “ **Personne**”

```
CREATE TABLE Personne  
(Nom CHAR(20) NOT NULL,  
  Prénom CHAR (20) NOT NULL);
```

Création de la Base

- Création d'une table avec **contraintes** de **Clé** et **NULL**

CREATE TABLE Nom_table

```
( Nom_Attr_1 TYPE [NOT NULL],  
  Nom_Attr_2 TYPE [NOT NULL], .....  
  Nom_Attr_n TYPE [NOT NULL],
```

```
CONSTRAINT Expression(s)_de_la/les_ Contrainte(s));
```

```
CREATE TABLE Personnes  
(Id NUMBER (10) PRIMARY KEY,  
  Nom CHAR (20) NOT NULL,  
  Prénom CHAR(20) NOT NULL);
```

Exemple:

Création de la
Table “
Personne”

```
CREATE TABLE Personne  
( Id NUMBER (10) NOT NULL,  
  Nom CHAR(20) NOT NULL,  
  Prénom CHAR (20) NOT NULL,  
CONSTRAINT clé_Prim PRIMARY KEY ( Id));
```

Nom de la contrainte (facultatif)

➤ Expressions de la Contrainte :

Clé Primaire

➤ **PRIMARY KEY** (liste_des_attributs_de_la_cle_primaire)

Clé Étrangère

➤ **FOREIGN KEY** (liste_des_attributs_de_la_cle_etrangere)
REFERENCES NomTable

➤ **CHECK** Nom_attribut Opérateur expression

Contrainte de domaine

➤ Exemple :

```
CREATE TABLE CLIENT ( NumCli NUMBER(10),  
NomCli CHAR (50),  
DateNaiss DATE,  
Salaire NUMBER (10,2),  
NumEmp NUMBER (10),  
CONSTRAINT  
PRIMARY KEY (NumCli),  
FOREING KEY (NumEmp) REFERENCES EMPLOYEUR (NumEmp),  
CHECK (DateNaiss < SYSDATE));
```

Création de la Base

Saisie des Données

- **INSERT INTO** Nom_de_table **VALUES** (Val_1, Val_2,Val_n);
- **INSERT INTO** Nom_de_table (Nom_col_i, Nom_col_j,Nom_col_m) **VALUES** (Val_i, Val_j,Val_m);
- **INSERT INTO** Nom_de_table **VALUES** (Val_1, Val_2, ..., **NULL**, ..Val_n);

BIBLIOGRAPHIE

- GARDARIN, Georges. *Bases de données*. Editions Eyrolles, 2003.
- Florence Petit, Sylvain Cherrier, Claire David, Licence 2 : Bases de Données , Modèle Entité Association, Modèle Relationnel. *Institut d'électronique et d'informatique Gaspard-Monge (IGM), Université Gustave Eiffel*