

Le Modèle Relationnel

- Le modèle relationnel est un *modèle logique* associé aux SGBD relationnels (ex. Oracle, DB2, SQLServer, Access, Paradox, dBase...).
- **Objectifs du modèle relationnel :**
 - indépendance physique
 - traitement du problème de redondance des données
 - LMD non procéduraux (faciles à utiliser)
 - devenir un standard

1- Conception de Base de Données relationnelle (BDR):

Difficultés de conception d'une BDR:

- Conception non intuitive
- mais points communs dans les cas à traiter (souvent similaires)

Objectifs :

- Eviter la redondance de données.
- Assurer la cohérence des données.
- Permettre de manipuler des requêtes.

2- Le Schema relationnel

- Des données rangées dans des tables
- Des cases monovaluées
- Des contraintes sur les colonnes des tables.
- La description de la structure des tables et des contraintes. Exemple de contraintes :
 - Clé primaire
 - Clé étrangère
 - ...



table ETUDIANT					table DEPARTEMENT		
numE	nom	prenom	licence	numDep	numDep	libDep	NbHab
1	Martin	Véra	2	77	01	Ain	...
2	Martin	Annie	1	75
3	Dupont	Sylvie	2	77	75	Paris	...
4	Martin	Annie	1	93
5	Dupond	Laurent	2	92	77	Seine-et-Marne	...
6	Lefèvre	Laurent	3		92	Hauts-de-Seine	...
					93	Seine-Saint-Denis	...

clé primaire

clé étrangère

clé primaire

- ETUDIANT (numE, nom, prenom, licence, #numDep)
- DEPARTEMENT (numDep, libDep, nbHab)

Clé primaire : permet d'accéder aux données de façon **claire** et **unique** (une ou plusieurs colonnes)

Clé étrangère : **Lien** avec l'identifiant d'un autre schéma

Attention : pas de clé étrangère sur un schéma E/A

Table (ou relation)

Ensemble d'enregistrements structurés par champs (=colonnes).

-> L'ordre des colonnes et des lignes de la table n'a pas importance.

Exemple : Table ETUDIANT

Colonne = Champ = Attribut = Propriété = Caractéristique

Ligne
=
Objet
=
N-uplet
=
Tuple
=
Enregistrement
=
Occurrence

nom	prenom	date_n
Dupont	René	
Durand	Paul	1990-12-31
Martin	Paul	1990-12-30
...

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

Domaine = ensemble de valeurs possibles d'un champ

SCHEMA DE LA RELATION

(Description en intention)

R (**A1**, **A2**,, **An**)

R : Nom de la relation

A1, **A2**, ..**An** : Nom des attributs de la relation

(**a_{1i}**, **a_{2j}**, ..., **a_{nk}**) : **n-uplet**

n : Degré de la relation (Relation n-aires)

- Exprimé de manière générale sous la forme :

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

- exemple :

Etudiant (num, nom, prenom, date_n)

Schéma relationnel d'une base de données

- Ensemble des schémas des relations de la base de données

- Exemple très simplifié 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)

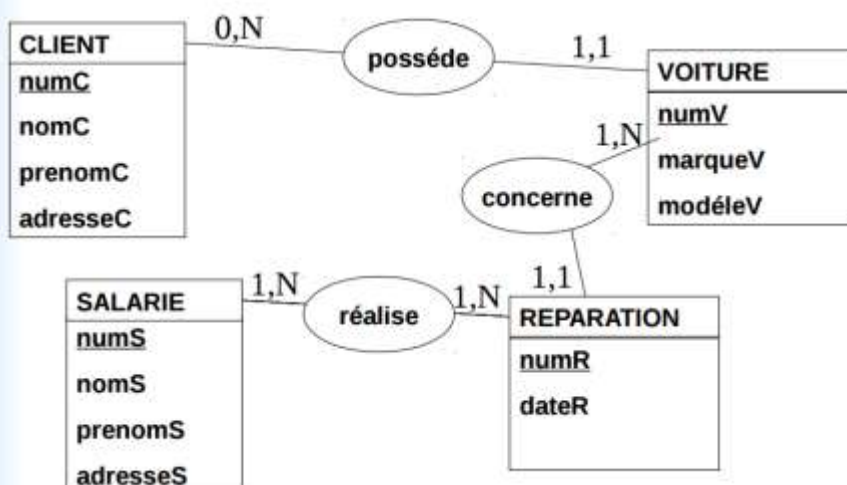
REPARER (#numS, #numR, nbHeuresTravail)

numS référence numS(SALARIE)

numR référence numR(REPARATION)

Schéma Entité Association

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



Relationnel : Clé primaire

- Identificateur d'un enregistrement de la table
 - permet de différencier 2 enregistrements d'une table, de façon claire, unique, et la plus concise possible
 - 2 enregistrements doivent avoir des valeurs de clé différentes
- Naturel ou artificiel (code, numéro incrémenté automatiquement)
- Choix parfois entre plusieurs clés (**clés candidates**)
- **Clé simple** (un seul attribut) ou **clé composée/multiple/concaténée** (plusieurs attributs)
- doit être stable dans le temps (*age ou date de naissance ?*)

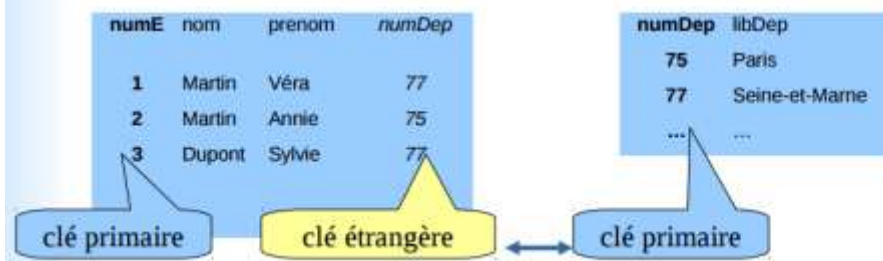
num	NIR	Login	nom	prenom	date_n
00001	1630706054781	rdupont	Dupont	René	
00002	1801277013111	pdurand	Durand	Paul	1990-12-31
00003	1801275012141	pmarti01	Martin	Paul	1990-12-30
00004	1801275012142	pmarti02	Martin	Pierre	1990-12-30

Clé primaire (Primary Key = PK)

- attribut ou ensemble d'attributs d'une table, permettant d'identifier et de distinguer ses occurrences
- par convention, est soulignée dans le schéma relationnel

Clé étrangère (Foreign Key = FK)

- fait la liaison entre 2 tables d'une base de données
- clé étrangère de la table T1, clé primaire de la table T2.
- doit être d'un type de données compatible à la table de référence
- peut ou non être tout ou partie de clé primaire de sa table



3- Les Contraintes d'intégrité

- **Définition :** Les contraintes d'intégrité sont des contrôles effectués lors de la saisie, modification ou suppression des données. On distingue 3 contrôles différents :

■ CONTRAINTES D'INTEGRITE

- CLE PRIMAIRE
- VALEUR NULLE
- CLE ETRANGERE

CONTRAINTES D'INTEGRITE :

Contrôles effectués lors de la saisie, modification ou suppression des données

CLE PRIMAIRE : Unicité de la valeur dans la(es) colonne(s) constituant la clé.

F012	Mozart	Wolfgang	
S234	Beethoven	Ludvig	
G087	Berlioz	Hector	
B555	Haydn		
X007	Ravel	Maurice	
A631	Chopin	Frédéric	
G087	Halliday	Johnny	

Impossible : Valeur refusée lors de la saisie

VALEUR NULLE :

Dans cette colonne certaines lignes peuvent ne pas contenir de valeurs

F012	Mozart	Wolfgang	
S234	Beethoven	Ludvig	
G087	Berlioz	Hector	
B555	Haydn		
X007	Ravel	Maurice	
A631	Chopin	Frédéric	
F132			

VALEUR PAR DEFAULT :

Valeur attribuée à toutes les lignes de la colonne, cette valeur peut être modifiée lors de la saisie.

ENSEMBLE DE VALEURS : Contrôle de l'appartenance de la valeur saisie à un intervalle ou bien à une liste de valeurs.

CLE ETRANGERE :

Attribut clé primaire dans une autre table. Vérification de la contrainte d'intégrité référentielle

F012	Mozart	Wolfgang	
S234	Beethoven	Ludvig	
G087	Berlioz	Hector	
B555	Haydn		
X007	Ravel	Maurice	
A631	Chopin	Frédéric	

	S234
	F012
	G087
	A631
	G087
	B555
	S234
	G087
	A831

Valeur inexistante dans la table

4- Du Modèle Entité/ Association au Modèle Relationnel

Modèle Entité Association – Modèle relationnel

- **Schéma EA : Description de la problématique**
 - Entités,
 - Associations,
 - Cardinalités
- **Schéma relationnel : Implémentation d'une solution**
 - Relations,
 - Contraintes (PK, FK, et d'autres...)
- **Comment faire le lien?**

Règles de passage Schéma E/A->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 primaires est l'ensemble des identifiants des entités concernées
- Tout attribut de l'association devient attribut de la nouvelle table

Schéma Entité/Association

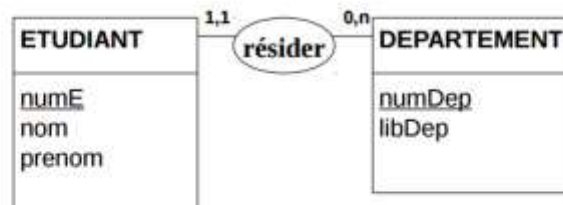


Schéma relationnel

- ETUDIANT (numE, nom, prenom, #numDep)
numDep référence numDep(DEPARTEMENT)
- DEPARTEMENT (numDep, libDep)

Représentation d'une relation de type N-M

Schéma Entité/Association

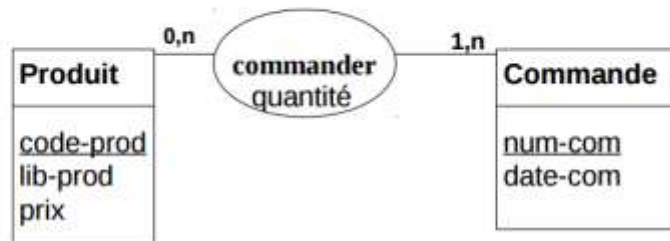


Schéma relationnel

- PRODUIT (code-prod, lib-prod, prix)
- COMMANDE (num-com, date-com)
- LIGNE-DE-COMMANDE (#code-prod, #num-com, quantité)
code-prod référence code-prod(PRODUIT)
num-com référence numCom(COMMANDE)

Représentation d'une relation de type N-M

Schéma Entité/Association

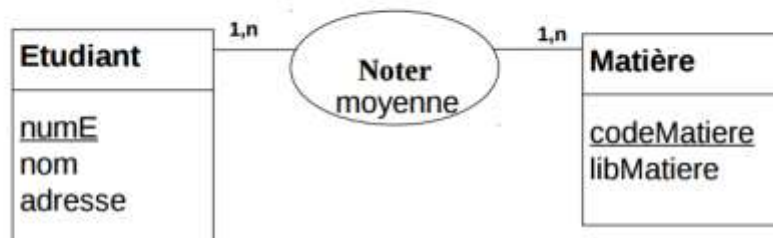


Schéma relationnel

- ETUDIANT (numE, nom, adresse)
- MATIERE (codeMatiere, libMatiere)
- NOTER(#numE, #codeMatiere, moyenne)
numE référence numE(Etudiant)
codeMatiere référence codeMatiere(Matiere)

Représentation d'une relation de type N-M (vide)

Schéma Entité/Association

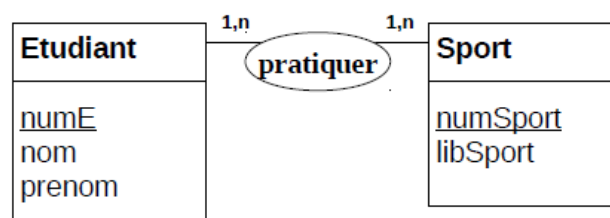


Schéma relationnel

- ETUDIANT (numE, nom, prenom)
- SPORT (numSport, libSport)
- PRATIQUER (#numE, #numSport)
numE référence numE(Etudiant)
numSport référence numSport(Sport)

Représentation d'une relation ternaire

Schéma Entité/Association

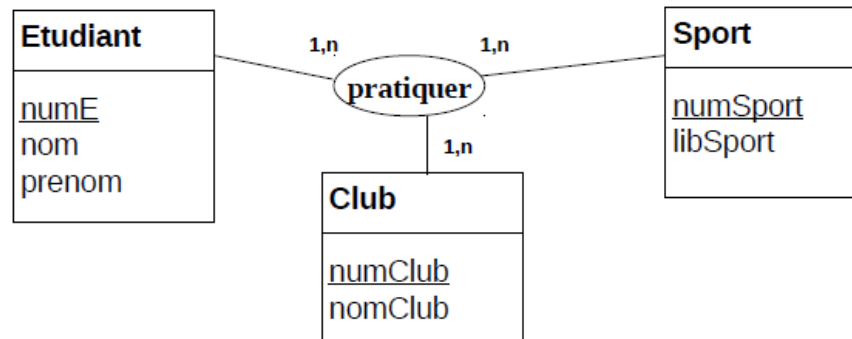
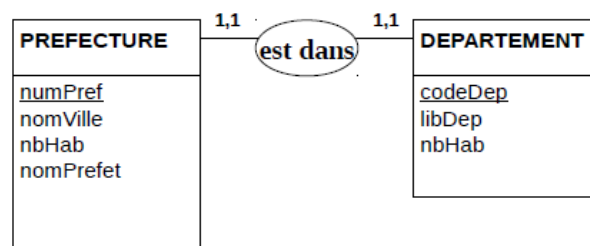


Schéma relationnel

- ETUDIANT (numE, nom, prenom)
- SPORT (numSport, libSport)
- CLUB(numClub, nomClub)
- PRATIQUER (#numE, #numSport, #numClub)
numE référence numE(Etudiant)
numSport référence numSport(Sport)
numClub référence numClub(Club)

Représentation d'une relation de type 1-1

Schéma Entité/Association



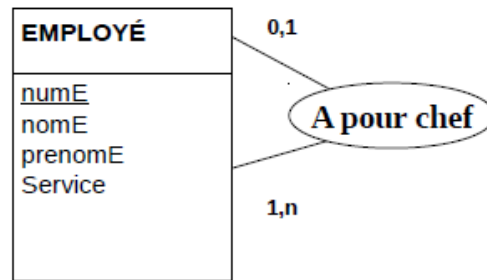
Attention : pas de clé étrangère sur un schéma E/A

Schéma relationnel

- PREFECTURE (numPref, nomVille, nbHab, nomPrefet, #codeDep)
codeDep référence codeDep(DEPARTEMENT)
- DEPARTEMENT (codeDep, libDep)

Représentation d'une relation reflexive

Schéma Entité/Association



Attention : pas de clé étrangère sur un schéma E/A

Schéma relationnel

- EMPLOYE (numE, nomE, prenomE, Service, #chef)
chef référence numE(EMPLOYE)

Types d'associations (selon cardinalités maximales)

- Association 1 - 1 (un à un)



- Association 1 - N (un à plusieurs)



- Association N - M (plusieurs à plusieurs)



Le Langage SQL

- SQL = *Structured Query Language*
- SQL permet la définition, la manipulation et le contrôle d'une base de données relationnelle. Il se base sur l'algèbre relationnelle.
- SQL est un standard depuis 1986.

SQL

Structured Query Language(SQL):

- Langage de base dans les SGBD
- Langage de Définition des Données (**LDD**):
Il permet la définition des structures des tables et des autres objets de la BD (index, ...)
- Langage de Manipulation des Données (**LMD**):
Il permet de faire des recherches (sélection) et des mises à jour (ajout, suppression, modification) des données de la base.

• SQL: LDD

CREATION DE LA BASE

✓ LANGAGE SQL

Création d'une table sans contraintes autres que NULL

```
CREATE TABLE Nom_table  
( Nom_Attr1 TYPE [NOT NULL],  
  Nom_Attr2 TYPE [NOT NULL],  
  .....  
  Nom_Attrn TYPE, [NOT NULL]  
);
```

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

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

CREATION DE LA BASE

✓ LANGUAGE SQL

Création d'une table avec contraintes de clé et contraintes plus générales

```
CREATE TABLE Nom_table  
( Nom_Attr1 TYPE [NOT NULL],  
  Nom_Attr2 TYPE [NOT NULL],  
  .....  
  Nom_Attrn TYPE, [NOT NULL],  
  CONTRAINTES );
```

```
CREATE TABLE Personnes  
(Nom CHAR(20) PRIMARY KEY,  
Prénom CHAR(20));
```

```
CREATE TABLE Personnes  
(Nom CHAR(20) NOT NULL,  
Prénom CHAR(20),  
CONSTRAINT clé_primaire PRIMARY KEY(Nom));
```

Nom_de_la_contrainte :

pk_nomTable

fk_nomTable_nomTableRéféréncée

ck_nomAttribut

Expression_de_la_contrainte :

PRIMARY KEY (liste_des_attributs_de_la_cle_primaire)

FOREIGN KEY (liste_des_attributs_de_la_cle_etrangere)

REFERENCES

NomTable

CHECK Nom_attribut Opérateur expression

Exemple de création de table avec les contraintes:

```
ex. CREATE TABLE CLIENT ( NumCli NUMBER(3),  
                           Nom CHAR(30),  
                           DateNaiss DATE,  
                           Salaire NUMBER(8,2),  
                           NumEmp NUMBER(3),  
  
                           CONSTRAINT cle_pri PRIMARY KEY (NumCli),  
                           CONSTRAINT cle_etr FOREIGN KEY (NumEmp)  
                             REFERENCES EMPLOYEUR(NumEmp),  
                           CONSTRAINT date_ok CHECK (DateNaiss < SYSDATE));
```

Définition des données

Contraintes d'intégrité

- Mot clé CONSTRAINT
- Identification par un nom de contrainte
- Clé primaire ⇒ PRIMARY KEY (clé)
- Clé étrangère ⇒ (clé) REFERENCES table(attribut)
- Contrainte de domaine ⇒ CHECK (condition)

CREATION DE LA BASE

■ SAISIE DES DONNEES

✓ LANGUAGE SQL

```
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);
```