

7 - Bases de données relationnelles



Pour leur conception, les bases de données sont ici vues comme des ensembles constitués de plusieurs populations d'objets *en interaction*, participant au bon fonctionnement d'un certain *système*. Établir un schéma de base de données consiste à décrire ces différentes populations d'objets, mais surtout et principalement à décrire les dépendances et les interactions entre ces populations.

7.1 Modèle ensembliste

Une base de donnée est constituée de plusieurs ensembles d'objets et d'opérateurs participant au bon fonctionnement d'un système:

Exemple 1 :

- Ensembles d'employés
- Ensembles de commandes
- Ensembles d'articles
- Ensembles de clients

Exemple 2 :

- Ensembles d'étudiants
- Ensembles de séances
- Ensembles de cours
- Ensembles de copies

On parle plus généralement d'**ensembles d'entités**.

Le modèle entité/association



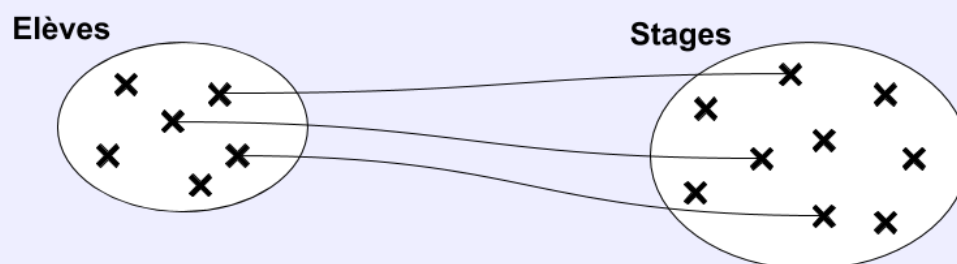
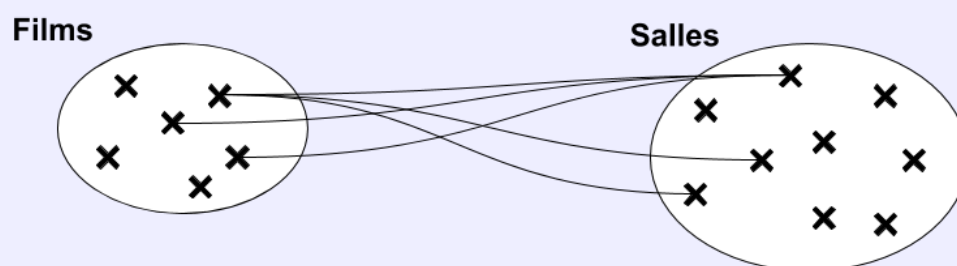
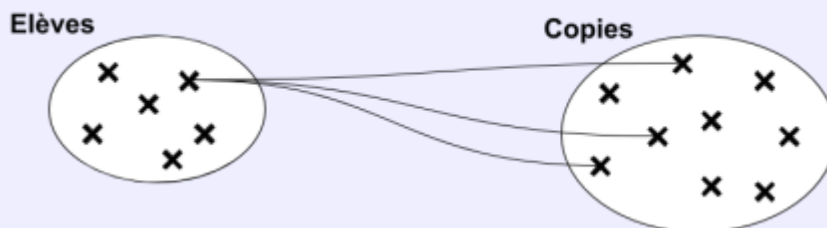
Le modèle entité/associations est une méthode de description des relations entre ensembles d'entités. Il s'appuie sur le prédicat selon lequel tous les éléments des ensembles d'entités sont discernables.

Le modèle entités/associations repose sur un langage graphique de description des données, indépendant du support et de la mise en œuvre informatique.

Définitions

Modéliser une base de données, c'est :

- Identifier les différents ensembles en interaction
- Identifier les liens de dépendance entre les différents ensembles



Les liens entre les différents ensembles sont appelés des **associations**

Association

Une association exprime des relations de dépendance entre deux ou plusieurs ensembles d'entités.



Définition : Une **association** entre les ensembles E_1 , ..., E_k est un sous-ensemble du produit $E_1 \times \dots \times E_k$.

Il s'agit donc d'un ensemble de k-uplets $\{ \dots, (t_1, \dots, t_k), \dots \}$ t.q. $t_1 \in E_1, \dots, t_k \in E_k$.

où k est le degré de l'association :



- $k=2$: association binaire
- $k=3$: association ternaire
- etc...

Rôles des associations

- **Attribution** : propriété, réservation, participation, supervision, auteur, rôle, pilote, ...
- **Événements** : achat, vente, séance, épreuve, appel, consultation, réunion, transaction, transport ...
- **Aggrégation/Composition** : tout/parties, contenant/contenu, supérieur/subordonné, pays/région, ...
- **Relations entre membres** : parenté, collaboration, cercle d'amis, ...
- ...

Contraintes de cardinalité

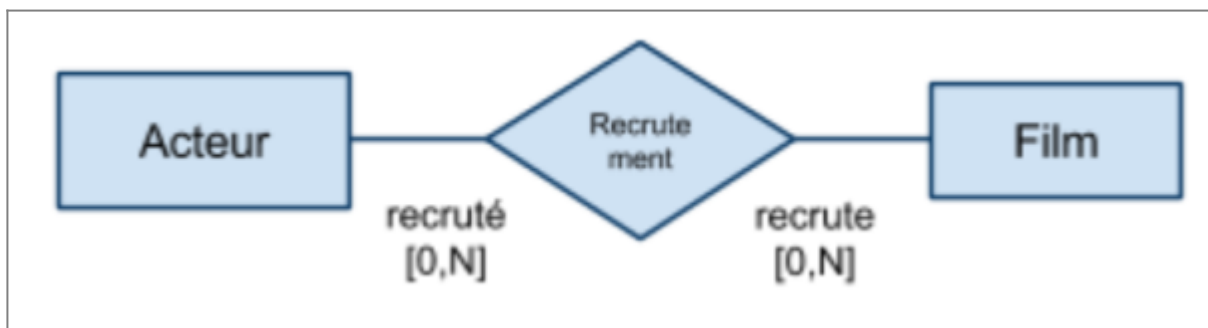


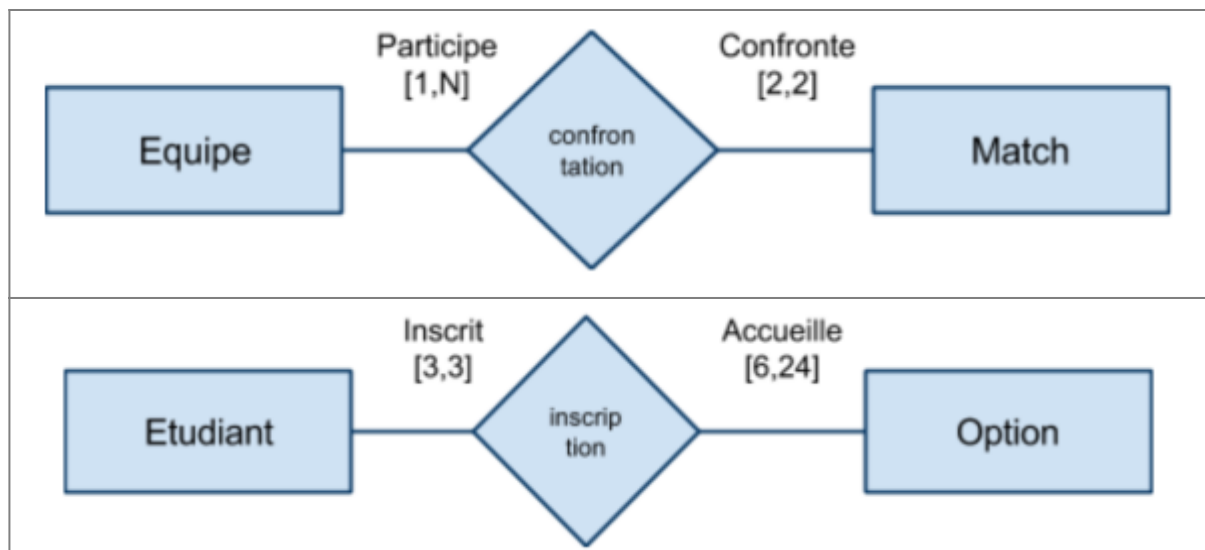
Pour chaque ensemble participant à une association, on précise dans combien d'instances de l'association chaque entité peut apparaître.

On donne en général un intervalle $[b_{\text{inf}}, b_{\text{sup}}]$ qui définit le nombre d'apparitions autorisées pour chaque rôle de l'association

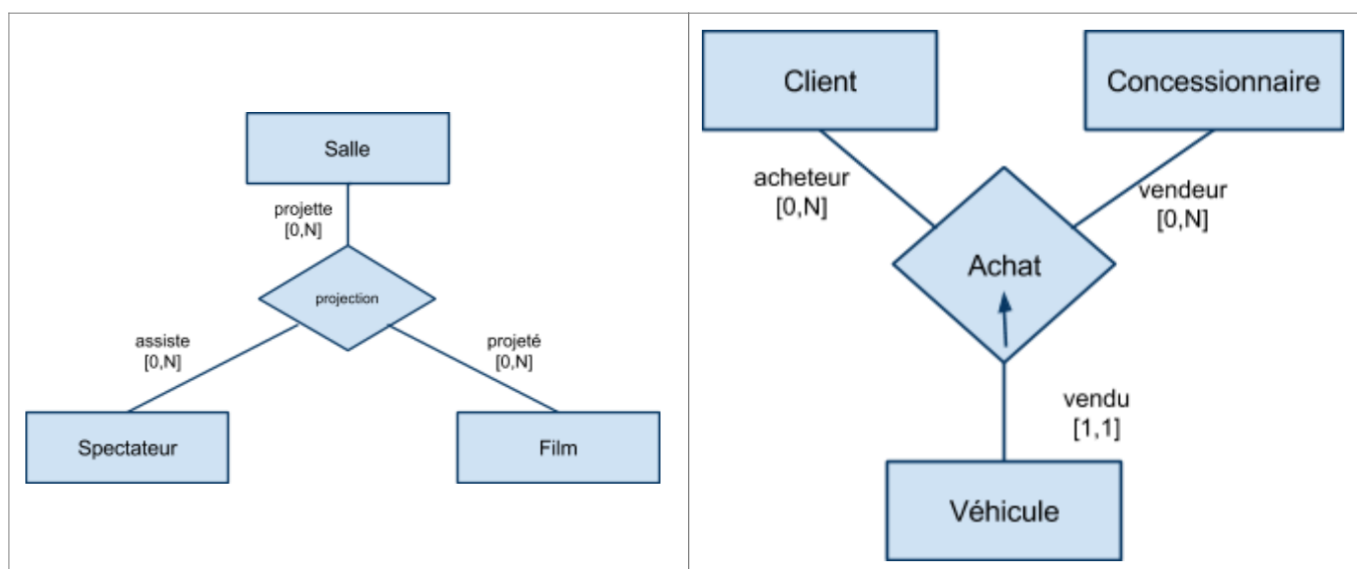
Représentation graphique

Associations binaires





Associations ternaires



Types d'associations

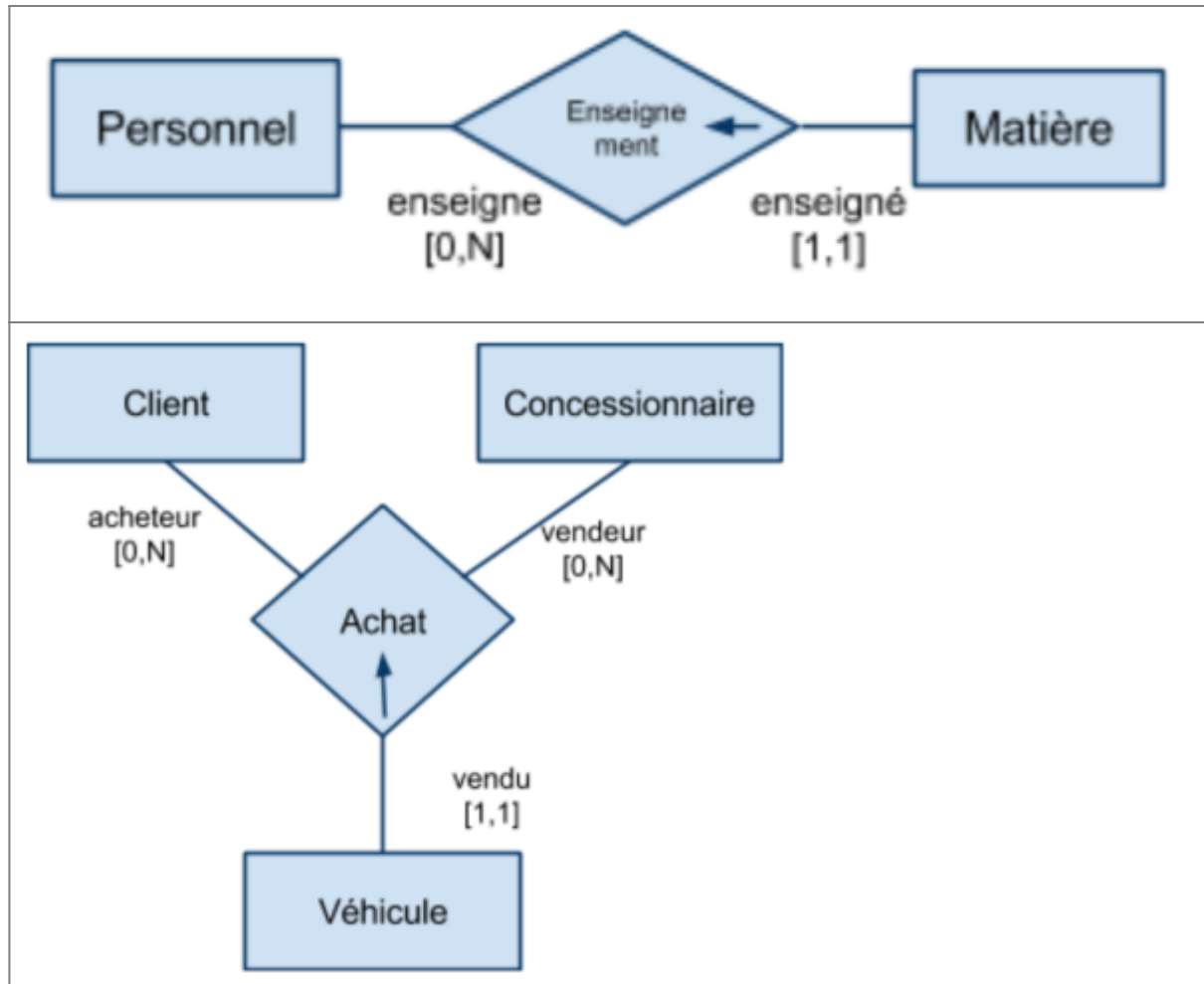
Associations de 1 à plusieurs (fonctionnelle)

Relation non symétrique entre les deux ensembles : [...,1] d'un côté, [...,N] de l'autre. Relation de type contenant/contenu, propriétaire/objet possédé, occupant/occupé, actif/passif etc... Il s'agit du type d'association le plus "courant".



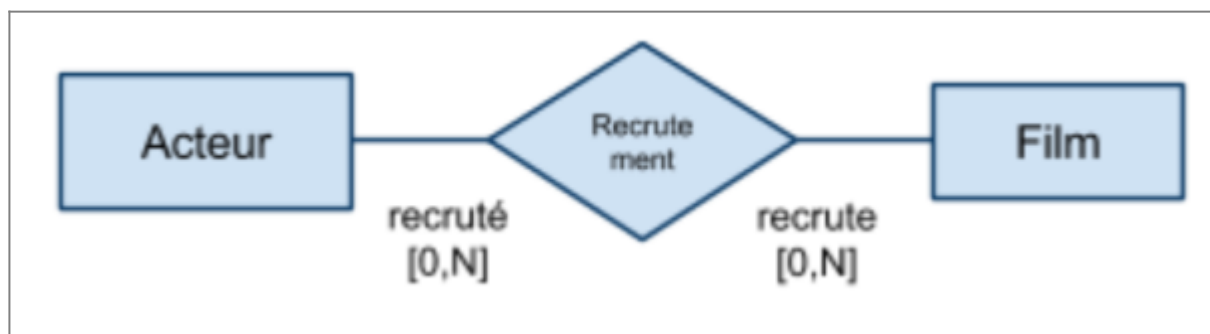
On dit parfois que l'ensemble dont la participation est unique est dit "à gauche" de l'association fonctionnelle, et celui dont la participation est multiple est "à droite", autrement dit la pointe de la flèche désigne l'ensemble de "droite":

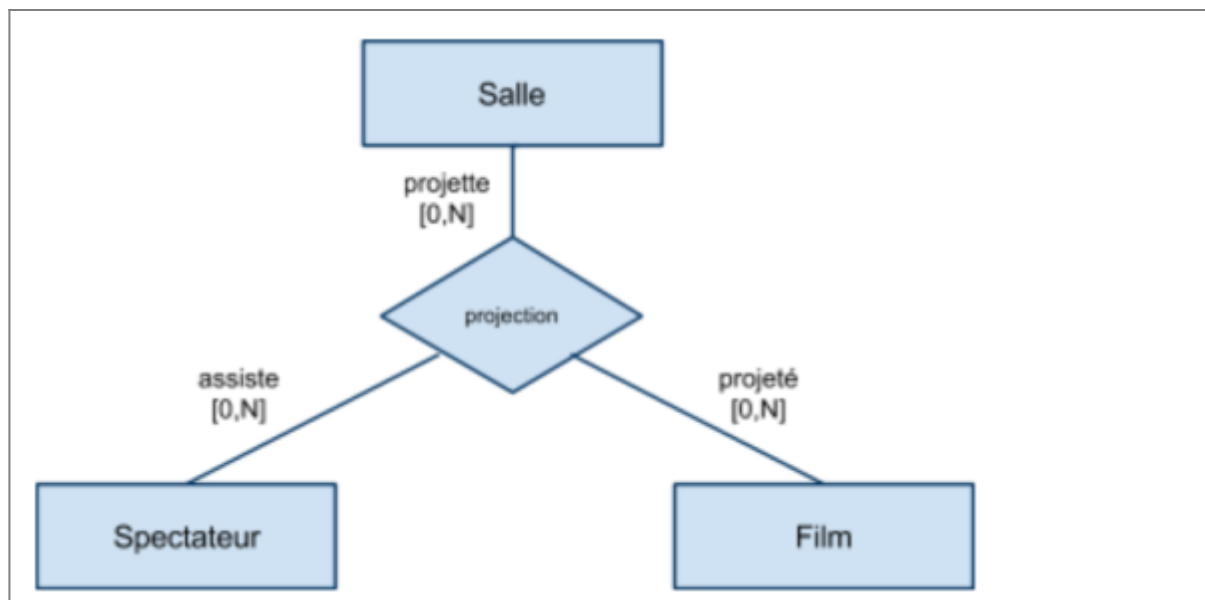
"à gauche" → "à droite"



Associations de plusieurs à plusieurs (croisée)

Dans une association "croisée", les tous les lien de l'association sont de cardinalité multiple [...N]





Modèles Entité Associations valués

Dans le cadre du modèle entité/association :

- les attributs des ensembles d'entités sont des *mesures*:
 - Soit A un attribut de l'ensemble d'entités \mathcal{E}



$A : \mathcal{E} \rightarrow D(A)$

- les attributs des associations sont des *opérateurs* :
 - Soit B un attribut de l'association sur $\mathcal{E} \times \mathcal{F}$

$B : \mathcal{E} \times \mathcal{F} \rightarrow D(B)$

Mesures

- Plusieurs mesures peuvent être opérées sur les objets d'un ensemble. Chaque mesure est un attribut
- Le schéma de l'ensemble est l'ensemble des attributs servant à le mesurer
- Les éléments de l'ensemble sont *discernables* ssi il existe un jeu de mesures différent pour chaque objet de l'ensemble
- Une *clé* est un jeu de mesures *minimal* (permettant de distinguer les objets) appartenant au schéma



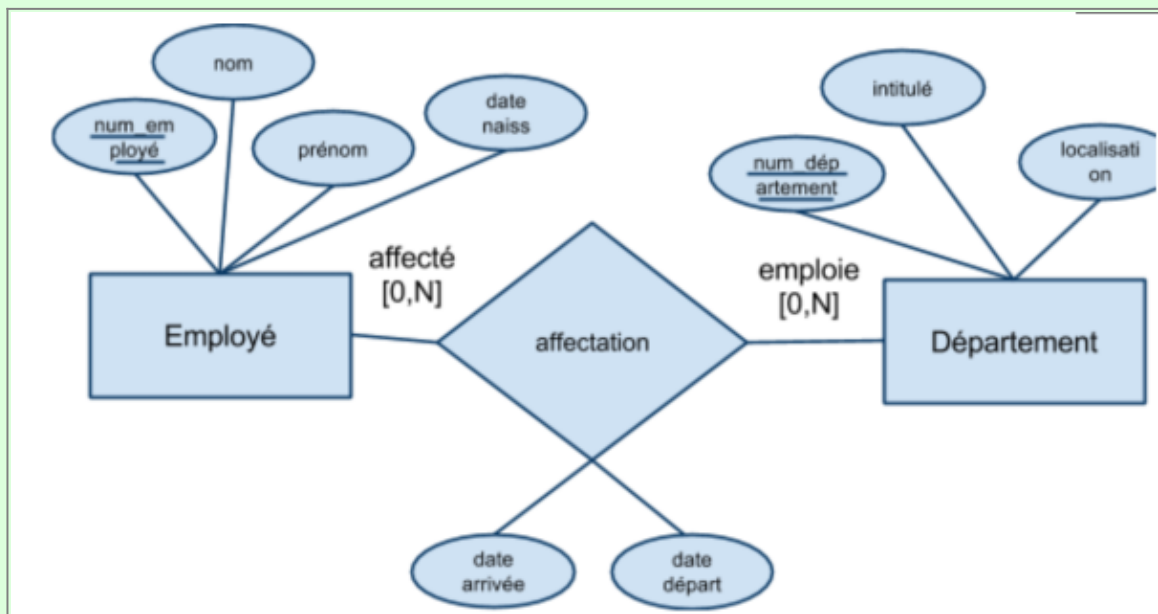
TODO

Ensembles discernables / non discernables

Opérateurs

- On s'intéresse ici aux associations qui représentent une "opération" (inscription, achat, embauche, affectation...).
- Lors d'une mise à jour de la base, certains événements tels que l'emprunt ou le retour d'un ouvrage, l'affectation d'un employé à un poste, ou la liste des anciens clients disparaissent.
- Il est possible de garder une trace des événements passés en mettant un (ou plusieurs) attributs sur une association.
- Ainsi, certaines associations peuvent être "datées", c'est à dire
 - avoir lieu à une date
 - ou prendre place sur une durée précise (prêt, accès temporaire, statut temporaire...)
- On peut ainsi mémoriser :
 - "Monsieur Dupont a été employé au département logistique de tant à tant."...
 - "L'étudiant X a été élève Centrale Méditerranée de telle année à telle année"...

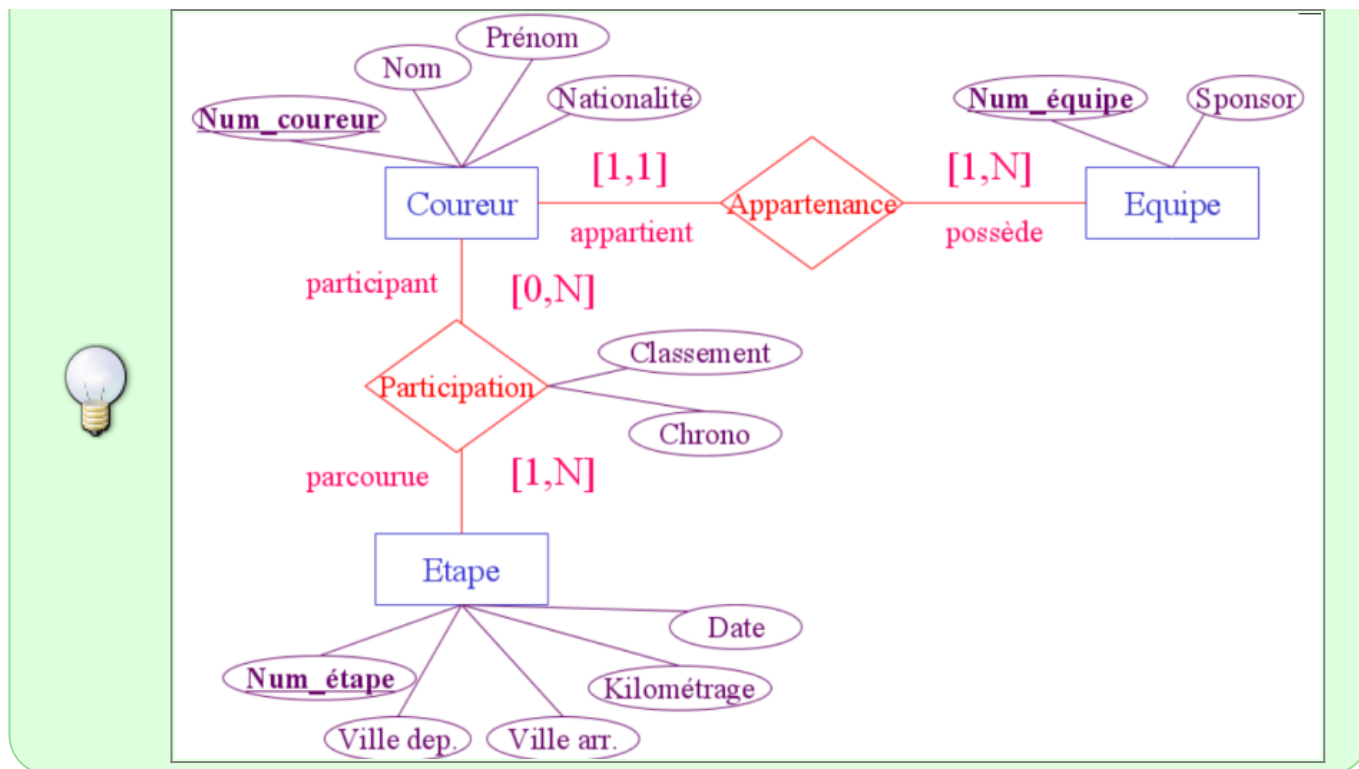
Exemple "Monsieur Dupont a été employé au département logistique de tant à tant."...



Exemple



- Chaque coureur est décrit par ses nom, prénom, nationalité et numéro de maillot.
- Chaque coureur appartient à une équipe qui possède un numéro, un sponsor associé.
- Chaque coureur participe à une ou plusieurs étapes. Une étape se caractérise par son numéro, son type (contre la montre/étape simple), ses points de départ et d'arrivée, sa date.
- A chaque étape est associée un classement d'arrivée pour chaque coureur, avec la durée totale de course.



7.2 Traduction vers le modèle relationnel

Il est possible de traduire un modèle entité/association vers un modèle relationnel (en perdant quelques propriétés).

Lors de la réalisation d'une base de données, on passe en général par les étapes suivantes:



1. Conception de la base sous forme d'un modèle entité/association.
2. Traduction sous la forme d'un modèle relationnel.
3. Normalisation (voir [Normalisation d'un schéma](#))
4. Mise en œuvre informatique.

Un petit nombre de règles permettent de traduire un modèle entité/association vers un modèle relationnel.

- Selon ces règles, à la fois les ensembles d'entités et les associations sont transformés en schémas relationnels.
- Les liaisons et dépendances entre schémas de relation sont assurés par la définition des **clés étrangères** (attributs communs à plusieurs tables).

Schéma de base et clé étrangère



- Un schéma (ou modèle) de bases de données est un ensemble fini de schémas



de relation.

- Une base de données est un ensemble fini de relations.
- Les liens et associations entre relations s'expriment sous la forme de **clés étrangères**

Définition



- Au sein d'un schéma relationnel R , Une clé étrangère est un attribut (ou un groupe d'attributs) qui constitue la clé primaire d'un schéma S distinct de R .
- La présence d'une clé étrangère au sein d'une relation r de schéma R introduit une contrainte d'intégrité sur les données :
 - la valeur des attributs de la clé étrangère d'un tuple de r doit être trouvée dans la table s correspondante.
- On indique la présence d'une clé étrangère à l'aide de pointillés : $\{..., \underline{\text{Clé étrangère}}, \dots\}$

Exemple

Schéma de base relationnelle :



- **Clients** (nom_client, adresse_client, solde)
- **Commandes** (num_Commande, nom_client, composant, quantité)
- **Fournisseurs** (nom_fournisseur, adresse_fournisseur)
- **Catalogue** (nom_fournisseur, composant, prix)

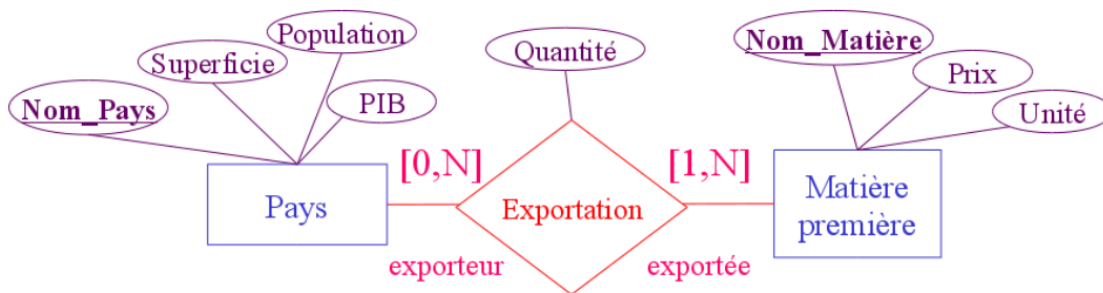
Traduction des associations de plusieurs à plusieurs

Une association croisée ne contient que des contraintes de cardinalité de type $[...,N]$. Soit R une telle association et E_1, \dots, E_k les ensembles participant à l'association.

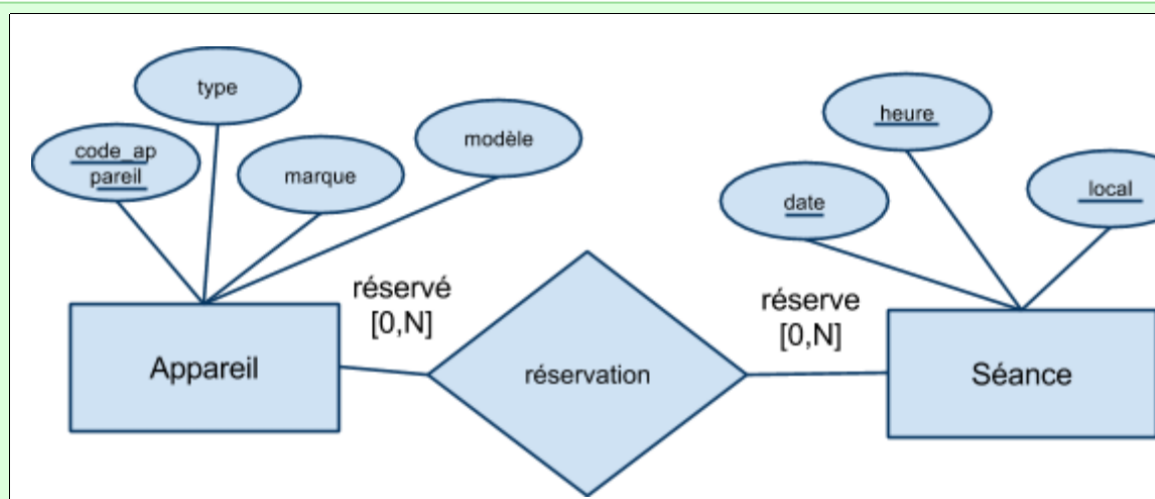
Règle de traduction :



- Chaque ensemble E_i est traduit par un schéma relationnel (contenant les mêmes attributs)
- L'association R est traduite sous la forme d'un schéma relationnel contenant:
 - les clés primaires des ensembles participant à l'association
 - (éventuellement) les attributs propres à l'association,

Exemple :**Traduction :**

- **Pays** (nom_pays, superficie, population, PIB)
- **Matière première** (nom_matière, unité, prix)
- **Exportation** (nom pays, nom matière, quantité)

**Traduction :**

- **Appareil** (code_appareil, type, marque, modèle)
- **Séance** (date, heure, local)
- **Réservation** (code appareil, date, heure, local)

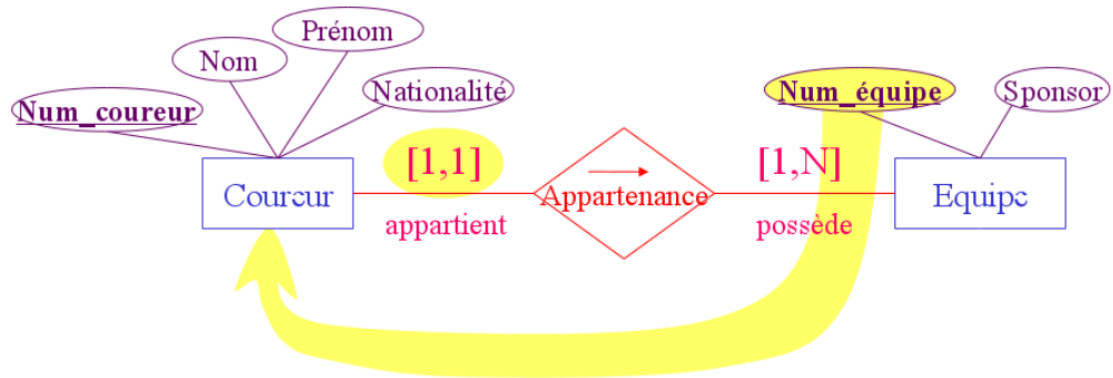
Traduction des associations de un à plusieurs

Soit une association fonctionnelle \$R\$. On suppose qu'il existe au moins un ensemble \$A\$ de cardinalité unique [1,1] participant l'association.

Règle de traduction

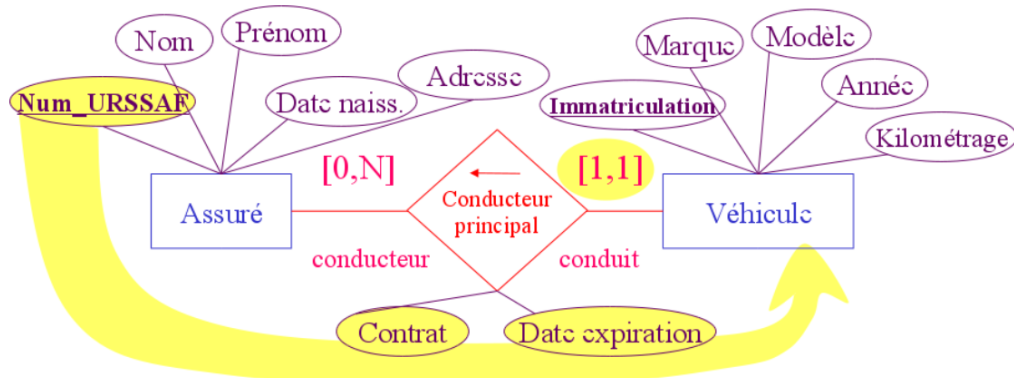
- Chaque ensemble participant est traduit sous forme de schéma relationnel
- L'association \$R\$ est traduite sous forme de **clé étrangère** : l'ensemble \$A\$ reçoit la clé primaire du (ou des) ensemble(s) dont la participation est multiple.

Exemple



→ **Coureur** (num_coureur, nom, prénom, nationalité, numéquipe)
Equipe (numéquipe, sponsor)

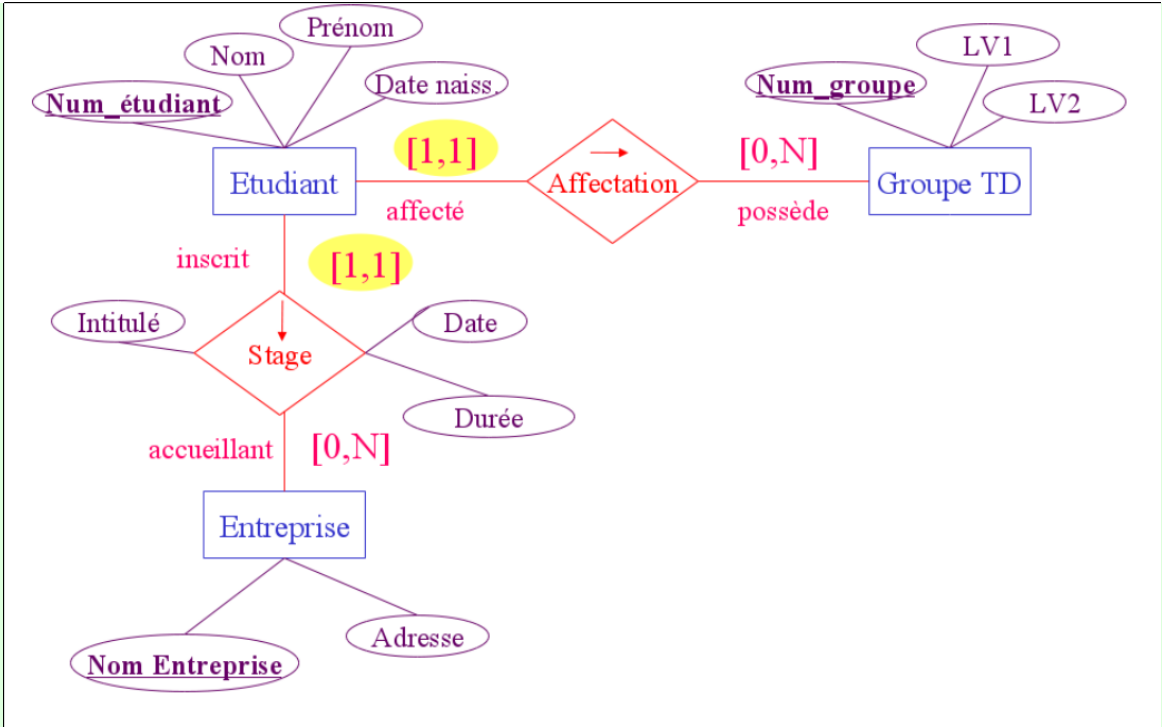
Remarque : lorsque l'association est évaluée, les attributs de l'association sont également injectés dans la table représentant l'ensemble de gauche.



→ **Assuré** (num_URSSAF, nom, prénom, date_naiss., adresse)
Véhicule (immatriculation, marque, modèle, année, kilométrage, num_URSSAF, contrat, date_exp.)

Exemple





Traduction :

- **Groupe_TD**(num_groupe, LV1, LV2)
- **Entreprise** (nom_entreprise, Adresse)
- **Etudiant** (num_étudiant, Nom, Prénom, Date_naiss, num_groupe, intitulé, date, durée, nom entreprise)

Exemple complet

Schéma de base relationnelle :



- **Clients** (nom_client, adresse_client, solde)
- **Commandes** (num_Commande, nom client, nom fournisseur, composant, quantité, montant)
- **Fournisseurs** (nom_fournisseur, adresse_fournisseur)
- **Catalogue** (nom_fournisseur, composant, prix)

Réalisation :

Clients :



nom_client	adresse_client	solde
Durand	7, rue des Lilas	335,00
Dubois	44, av. du Maréchal Louis	744,00
Duval	5, place du marché	33,00

Commandes :

num_Commande	nom_client	composant	quantité
6674	Dubois	micro controller	55
6637	Dubois	radio tuner	2
6524	Durand	transistor	4
6443	Duval	micro controller	7

Fournisseurs :



nom_fournisseur	adresse_fournisseur
Sage	33, College street, London
MoxCom	77 Ashley square, Mumbai

Catalogue :

nom_fournisseur	composant	prix
Sage	transistor	4,4
MoxCom	micro controller	3,7
MoxCom	radio tuner	7,0

SQL

Exemple de définition de table avec clé étrangère en SQL :



```
CREATE TABLE Commande (
  num_commande INTEGER NOT NULL,
  nom_client VARCHAR(30),
  nom_fournisseur VARCHAR(30),
  composant VARCHAR(30),
  quantité INTEGER,
  montant DECIMAL(12,2) NOT NULL,
  PRIMARY KEY (num_commande),
  FOREIGN KEY (nom_client) REFERENCES Client,
  FOREIGN KEY (nom_fournisseur, composant) REFERENCES Catalogue);
```

7.3 Interrogation des Bases de Données

Interroger une base de données , c'est sélectionner certaines données parmi l'ensemble des données proposés.

Exemples :

- Donner la liste des pays exportateurs de pétrole
- Liste des musiciens jouant à la fois du piano et du violon

- Liste des artistes français disque d'or en 1977
- Liste des suspects châtain taille moyenne présents à Poitiers la nuit du 12 au 13 février

C'est également croiser et recouper les informations présentes dans des bases

- pour faire ressortir des corrélations (exemple : type d'habitat/intentions de vote),
- pour des enquêtes de consommation, du marketing ciblé ...

C'est enfin personnaliser l'accès à l'information : ne retenir que les informations utiles à un instant et pour une personne donnée.

- Emploi du temps de l'élève X pour la semaine Y.
- Factures impayées du client Z

L'algèbre relationnelle



- propose un certain nombre d'opérations ensemblistes :
 - Intersection,
 - Union,
 - Projection,
 - Différence,
 - ...
- qui, à partir d'un ensemble de relations, permettent de construire de nouvelles relations.
- La relation nouvellement construite contient le résultat de la **requête**

7.4 Opérateurs mono-table

Extraction d'information à partir d'une table unique :

- projection π = extraction de colonnes
- sélection σ = extraction de lignes

7.4.1 Projection : π

Projection



- Soit r une relation de schéma R .
- Soit S un ensemble d'attributs, avec $S \subseteq R$

La **projection** $\pi_S(r)$ est une nouvelle relation de schéma S obtenue à partir des éléments de r restreints au schéma S : $\pi_S(r) = \{t(S) \mid t \in R\}$

(avec $t(S)$ la restriction de t au schéma S)

Exemple Catalogue :

nom_fournisseur	adresse_fournisseur	composant	prix
Sage	33, College street, London	transistor	4,4
MoxCom	77 Ashley square,Mumbay	micro controller	3,7
MoxCom	77 Ashley square,Mumbay	radio tuner	7,0



Requete : Donner la liste des fournisseurs (avec leur adresse): $\pi_{\text{nom_fournisseur, adresse_fournisseur}}(\text{Catalogue})$

\rightarrow u :

nom_fournisseur	adresse_fournisseur
Sage	33, College street, London
MoxCom	77 Ashley square,Mumbay

7.4.2 Sélection : σ **Condition sur R**

- On considère le schéma $R(A_1, \dots, A_n)$
- Une condition F sur R :
 - est un ensemble de contraintes sur les valeurs des attributs A_1, \dots, A_n
 - construites à l'aide d'opérateurs booléens classiques :
 - \wedge (et),
 - \vee (ou),
 - \neg (non),
 - $=, \neq, >, <, \geq, \leq, \dots$
 - et de valeurs numériques ou de texte.



Exemples : $F = (A_1 = 3) \wedge (A_1 > A_2) \wedge (A_3 \neq 4)$ $F = (A_1 = 2) \vee (A_2 = \text{"Dupont"})$

Sélection

- Soit r une relation de schéma R
- Soit F une condition sur R



La **sélection** $\sigma_F(r)$ est une nouvelle relation de schéma R , constituée de l'ensemble des enregistrements de r qui satisfont la condition F .

$$\sigma_F(r) = \{ t \in r \mid F(t) \text{ est vrai} \}$$

Exemple :

Requête : Donner la liste des fournisseurs qui vendent des micro-contrôleurs



$$u = \Pi_{\text{nom_fournisseur}}(\sigma_{\text{Composant} = \text{micro controller}}(\text{Fournisseur}))$$

nom_f
Moxcom

Exemple

Pays :



nom_pays	superficie	population	PIB/hab
Algérie	2.300.000	31.300.000	1630\$
Niger	1.200.000	11.400.000	890\$
Arabie Saoudite	2.150.000	24.300.000	8110\$

Requête : Donner la liste des pays dont le PIB/hab est > 1000\$

$$u = \Pi_{\text{nom_pays}}(\sigma_{\text{PIB/hab} > 1000}(\text{Pays}))$$

u :

nom_pays
Algérie
Arabie Saoudite

7.4.3 Structure d'une requête SQL

```
SELECT  A1,A2, ..., An    // liste d'attributs
FROM    R                // nom de la TABLE
WHERE   F                // condition sur les attributs
```

cette requête est semblable à :

- une sélection algébrique σ_F
- suivie par une projection algébrique $\Pi_{\{A1, \dots, An\}}$

soit : $\Pi_{\{A1, \dots, An\}}(\sigma_F(R))$

Exemples :

- Qui fournit des transistors ?

```
SELECT nom_fournisseur
FROM Fournisseur
WHERE composant = 'transistor';
```


- Liste de toutes les commandes de transistors :

```
SELECT *
FROM Commandes
WHERE composant = 'transistor'
```

- Qui fournit des micro-contrôleurs à moins de 5\$?

```
SELECT nom_fournisseur
FROM Catalogue
WHERE composant = 'micro controller' AND prix < 5
```

7.5 Opérateurs multi-tables

Principe : recoupement d'informations présentes dans plusieurs tables :

- Croisement des critères de sélection : **Jointure**
- Recherche ciblée : **Division**

7.5.1 La jointure : ⋈

Union de deux éléments :



- Soient les relations r et s de schémas R et S .
- On note $R \cap S$ la liste des attributs communs aux deux schémas et $R \cup S$ la liste des attributs appartenant à R ou à S .
- soit $t \in r$ et $q \in s$ tels que $t(R \cap S) = q(R \cap S)$

On note $t \cup q$ le tuple formé des valeurs de t et de q étendues au schéma $R \cup S$

Produit cartésien



- Soient r et s (de schémas R et S), avec $R \cap S = \emptyset$

Le produit cartésien $r \times s$ est une nouvelle table de schéma $R \cup S$ combinant les tuples de r et de s de toutes les façons possibles : $r \times s = \{t \cup q : t \in r, q \in s\}$

- La **jointure** est une opération qui consiste à effectuer un produit cartésien des tuples de deux relations pour lesquelles certaines valeurs correspondent.
- Le résultat de l'opération est une nouvelle relation.



Jointure

- Soient r et s (de schémas R et S), avec $R \cap S \neq \emptyset$



- La **jointure** $r \bowtie s$ est une nouvelle table de schéma $R \cup S$ combinant les tuples de r et de s ayant des valeurs communes pour les attributs communs.

$$r \bowtie s = \{t \cup q : t \in r, q \in s, t(R \cap S) = q(R \cap S)\}$$

Exemple

Matière_première :

nom_matière	unité	prix
pétrole	baril	45\$
gaz	GJ	3\$
uranium	lb	12\$

Exportations :



nom_pays	nom_matière	quantité
Algérie	pétrole	180.000
Algérie	gaz	20.000
Niger	uranium	30.000
Arabie Saoudite	pétrole	2.000.000
Arabie Saoudite	gaz	750.000

Matière_première \bowtie Exportations :

nom_pays	nom_matière	quantité	unité	prix
Algérie	pétrole	180.000	baril	45\$
Algérie	gaz	20.000	GJ	3\$
Niger	uranium	30.000	lb	12\$
Arabie Saoudite	pétrole	2.000.000	baril	45\$
Arabie Saoudite	gaz	750.000	GJ	3\$

Exemples de requêtes

- “Donner la liste des PIB/hab des pays exportateurs de pétrole” :

$\Pi_{\text{PIB/hab}}(\sigma_{\text{nom_matière} = \text{pétrole}}(\text{Pays} \bowtie \text{Exportations}))$

Schéma de base relationnelle :



- **Clients** (nom_client, adresse_client, solde)
- **Commandes** (num_Commande, nom_client, nom_fournisseur, composant, quantité, montant)
- **Fournisseurs** (nom_fournisseur, adresse_fournisseur)
- **Catalogue** (nom_fournisseur, composant, prix)

- “Donner le nom et l'adresse des clients qui ont commandé des micro controleurs” :

```
$$$ \text{nom_client, adresse_client} ( \sigma_{\text{composant} = 'micro-controller'} ( \text{Client} \bowtie \text{Commandes} ) ) $$$
```

7.5.2 Requêtes multi-tables en SQL

```
SELECT    A1,A2, ..., An      // liste d'attributs
FROM      R1, ..., Rm        // liste de TABLES
WHERE     F1 AND ... AND Fl   // liste de conditions sur les attributs
                                     // (en particulier conditions sur les attributs
                                     // sur lesquels s'effectue la jointure)
```

Pour exprimer la jointure sur l'attribut 'Aj' commun aux tables 'R1' et 'R2', on écrira : 'R1.Aj = R2.Aj'

Exemples :

```
SELECT PIB_par_hab
FROM Pays NATURAL JOIN Exportations
WHERE nom_matiere = 'petrole'
```

```
SELECT PIB_par_hab
FROM Pays, Exportations
WHERE nom_matiere = 'petrole'
AND Pays.nom_pays = Exportations.nom_pays
```

```
SELECT PIB_par_hab
FROM Pays
WHERE nom_pays IN (
    SELECT nom_pays
    FROM Exportations
    WHERE nom_matiere = 'petrole'
)
```

7.5.3 La division

Division

- Soient r (de schémas R) et s (de schémas S), avec $S \subseteq R$:



La division $r \div s$ est la relation (table) u de schéma $R-S$ maximale contenant des tuples tels que $u \times s \subseteq r$ (avec \times représentant le produit cartésien)

$$r \div s = \{ t \mid \forall q \in s, t \cup q \in r \}$$

→ on cherche les éléments de t qui “correspondent” à s

Exemples :

Nom_Pays	Nom_matière		Nom_matière	=	Nom_Pays
Algérie	Pétrole		Pétrole		Algérie
Algérie	Gaz		Gaz		
Niger	Uranium				
Arabie Saoudite	Pétrole				Arabie Saoudite
Arabie Saoudite	Gaz				

7.6 Recherches composées

- Certaines requêtes, peuvent être le résultat de la combinaison de plusieurs critères de recherche
- La combinaison de résultats est généralement réalisée à l'aide des opérations ensemblistes classiques (intersection, union...) pour exprimer «et», «ou», «non»...
- Pour alléger les formules, il est possible d'utiliser des tables intermédiaires.

Union



- Soient r1 et r2 deux tables de schéma R.

L'**union** $r1 \cup r2$ est une nouvelle table de schéma R constituée de l'ensemble des enregistrements qui appartiennent à r1 ou à r2: $r1 \cup r2 = \{ t \in r1 \} \cup \{ t \in r2 \}$

Intersection



- Soient r1 et r2 deux tables de schéma R.

L'**intersection** $r1 \cap r2$ est une nouvelle table de schéma R constituée de l'ensemble des enregistrements qui appartiennent à r1 et à r2: $r1 \cap r2 = \{ t \in r1 \} \cap \{ t \in r2 \}$

Différence



- Soient r1 et r2 deux tables de schéma R.

La **différence** $r1 - r2$ est une nouvelle table de schéma R constituée de l'ensemble des enregistrements qui appartiennent à r1 mais pas à r2: $r1 - r2 = \{ t \in r1 \} - \{ t \in r2 \}$

Exemples :

- Donner la liste des pays qui exportent à la fois du gaz et du pétrole :

$\pi_{\text{Pays}} \sigma_{\text{matière} = \text{gaz}} (\text{Exportations}) \cap \pi_{\text{Pays}} \sigma_{\text{matière} = \text{pétrole}} (\text{Exportations})$ en SQL :

```
SELECT pays FROM Exportations
WHERE matière = 'gaz'
INTERSECT (
    SELECT pays FROM EXPORTATIONS
    WHERE matière = 'pétrole');
```

- Donner la liste des pays qui exportent du gaz mais pas du pétrole :

$\pi_{\text{Pays}} \sigma_{\text{matière} = \text{gaz}} (\text{Exportations}) - \pi_{\text{Pays}} \sigma_{\text{matière} = \text{pétrole}} (\text{Exportations})$ en SQL :

```
SELECT pays FROM Exportations
WHERE matière = 'gaz'
EXCEPT (
    SELECT pays FROM EXPORTATIONS
    WHERE matière = 'pétrole');
```

* Donner la liste des clients qui commandent uniquement des produits 'Moxcom' :

$\pi_{\text{nom_client}} \text{Client} - \pi_{\text{nom_client}} \sigma_{\text{fournisseur} \neq \text{'Moxcom'}} \text{Client} \bowtie \text{Commande}$ en SQL :

```
SELECT nom_client FROM Client
EXCEPT (
    SELECT client FROM Client NATURAL JOIN Commande
    WHERE fournisseur <> 'Moxcom');
```

From:

<https://wiki.centrale-med.fr/informatique/> - **WiKi informatique**

Permanent link:

https://wiki.centrale-med.fr/informatique/tc_info:cm6

Last update: **2024/06/28 15:18**

