

Introduction aux bases de données

Modèle relationnel

Programme

2. **Modèle relationnel de données**

2.1 Éléments du modèle relationnel

2.2 Algèbre relationnelle

3. Langage SQL

3.1 Manipulation de données (interrogation, mise à jour)

3.2 Définition de schémas et de contraintes d'intégrité

2. Éléments du modèle relationnel

Introduction intuitive

Une Base de Données Relationnelle (BDR) peut être vue comme un ensemble de **tables** ou de **relations**

- Une table est un ensemble de lignes ou de colonnes

Exemple de BDR composée de 3 tables:

- **fournisseur** est une table contenant le numéro (nof), le nom (nomf) et la ville (ville) de chaque fournisseur
- **pièce** est une table contenant le numéro (nop), le nom (nomp) et le prix (prix) de chaque pièce
- **vente** est une table indiquant qu'une pièce (nop) est vendue par un fournisseur (nof)

2. Éléments du modèle relationnel

Table fournisseur

nof	nomf	ville
1	Girard	Lyon
2	Blanc	Paris
3	Merlin	Nancy

Table pièce

nop	nomp	prix
1	vis	1.5
2	écrou	2
3	boulon	2.5

Table vente

nop	nof
1	1
1	2
2	2
2	3
3	1
3	2
3	3

Une base de données relationnelle constituée de 3 tables (relations)

Formalisation du modèle relationnel : DOMAINE

- Un **domaine** est un ensemble de valeurs atomiques d'un certain type sémantique

Exemple

NOM_VILLE = {Berlin, Paris, Barcelone}

- Les domaines sont les ensembles de valeurs possibles dans lesquels sont puisées les données
- Deux ensembles peuvent avoir les mêmes valeurs même si sémantiquement distincts

Exemple

NUM_PRODUIT = {1, 2, ... 5000}

NUM_ETUDIANT = {1, 2, ..., 5000}

Formalisation du modèle relationnel : DOMAINE

- Le **produit cartésien** des domaines D_1, D_2, \dots, D_n noté $D_1 \times D_2 \times \dots \times D_n$ est l'ensemble des tuples (v_1, v_2, \dots, v_n) tels que $v_i \in D_i, 1 \leq i \leq n$

Exemple

Les domaines :

NOM = { DUPONT, DURAND }

PRENOM = { Pierre, Paul, Jacques }

$NOM \times PRENOM = \{(DUPONT, Pierre), (DUPONT, Paul), (DUPONT, Jacques), (DURAND, Pierre), (DURAND, Paul), (DURAND, Jacques)\}$

Formalisation du modèle relationnel : RELATION

- Une **RELATION** est un sous ensemble du produit cartésien de plusieurs domaines

$$R \subset D_1 \times D_2 \times \dots \times D_n$$

D_1, D_2, \dots, D_n sont les domaines de R et n est le degré ou l'arité de R

Exemple

Les domaines :

NOM = { DUPONT, DURAND }

PRENOM = { Pierre, Paul, Jacques }

Relation : ETUDIANT \subset NOM \times PRENOM

ETUDIANT = {(DUPONT , Pierre), (DURAND,Paul)}

Formalisation du modèle relationnel : LES N-UPLETS

- Un élément d'une relation est un n-uplet de valeurs (tuple en anglais)
- Un **n-uplet** représente un fait

Exemple

« Dupond Marc est un étudiant de l'université de lorraine »

« Dupond est inscrit en informatique »

Formalisation du modèle relationnel : LES ATTRIBUTS

- Chaque composante d'une relation est un **attribut**
- Chaque attribut a un nom (qui a un sens) (nom de la colonne)
- Plusieurs attributs peuvent avoir le même domaine

Exemple

TRAJET (Num, Ville_Depart, Ville-Arrivee)

Les attributs « Ville_Depart » et « Ville_arrivee » prennent leurs valeurs dans le même ensemble

Formalisation du modèle relationnel : LES ATTRIBUTS

- Attribut **atomique** / **composé** (subdivisé en attributs)
ex. les attribut *nom* et *prix* sont atomiques
ex. l'attribut *adresse* est composé de trois attributs : *rue*, *ville*, *CP*
- Attribut **dérivé** : calculé à partir d'autre(s) attribut(s)
ex. attribut *âge* calculé à partir de l'attribut *date_de_naissance*
- Attribut **monovalué** / **multivalué** (plusieurs valeurs par tuple) ex. *nom* versus *prénoms_des_enfants*

Formalisation du modèle relationnel : LES ATTRIBUTS

La valeur d'un attribut dans un tuple peut être absente

- Cet attribut est à valeur facultative
- Cette absence de valeur est notée **NULL**
 - Ex : l'attribut *téléphone* pour une personne n'ayant pas de téléphone
 - Ex : l'attribut *prix* pour une pièce dont le prix n'est pas encore fixé

Pièce

nop	nomp	prix
1	Vis10	1.5
2	écrou	2
3	boulon	2.5
4	vis15	NULL

Formalisation du modèle relationnel : SCHEMA D'UNE RELATION

- Le schéma d'une relation est défini :
 - Le nom de la relation
 - La liste de ses attributs
 - Un ensemble de contraintes d'intégrité
- On la note : **$R(A_1, A_2, \dots, A_n)$**
- L'extension d'une relation correspond à l'ensemble de ses éléments (n-uplets) → le terme RELATION désigne une extension
- L'intention d'une relation correspond à sa signification → le terme SCHÉMA DE RELATION désigne l'intention d'une relation

Exemple

ETUDIANT (NumE, Nom, Prenom, Dnaiss)

COURS (NumC, Nom_cours, NbH, Credits)

INSCRIT (NumC, NumE, note)

Formalisation du modèle relationnel : REPRESENTATION

- 1 relation = 1 table

Nom de la relation

Vin

Nom d'attribut

Cru	Millésime	Région	Couleur
Chenas	1983	Beaujolais	Rouge
Tokay	1980	Alsace	Blanc
Tavel	1986	Rhône	Rosé
Chablis	1986	Bourgogne	Blanc
St-émilion	1987	Bordelais	Rouge

tuple

une relation est un ensemble \Rightarrow on ne peut pas avoir 2 lignes identiques

Formalisation du modèle relationnel : SCHEMA D'UNE BDR

- **Le schéma d'une base de données** est défini par :
 - l'ensemble des schémas des relations qui la composent
- Attention à la différence entre :
 - le schéma de la BDR qui dit comment les données sont organisées dans la base
 - l'ensemble des n-uplets de chaque relation, qui représentent les données stockées dans la base

Exemple

BUVEUR(nob, nom, prénom)

VIN(nov, cru, millésime, degré, couleur)

CONSOMMATION(nob, nov, date, quantité)

Formalisation du modèle relationnel : Dépendance fonctionnelle

- Soit $R(A_1, A_2, \dots, A_n)$ une relation. Soit X et Y des sous ensembles de $\{A_1, \dots, A_n\}$. On dit que Y dépend fonctionnellement de X ($X \rightarrow Y$) si à chaque valeur de X correspond une valeur unique de Y .

On dit que : X détermine Y

Exemple

ETUDIANT (NumE, Nom_Etudiant, Prenom_Etudiant)

$\text{NumE} \rightarrow (\text{Nom_Etudiant}, \text{Prenom_Etudiant})$

Formalisation du modèle relationnel :

Clé d'une relation

- **Clé d'une relation** : Ensemble minimal d'attributs qui ont une valeur unique et définie pour tout tuple de la relation (à tout instant) qui détermine tous les autres attributs

Exemple

ETUDIANT (NumE, Nom_Etudiant, Prenom_Etudiant)

NumE \rightarrow (Nom_Etudiant, Prenom_Etudiant)

NumE est une clé

- Toute relation doit avoir au moins une clé, la clé primaire.
- Notation : clé primaire soulignée dans le schéma.
ETUDIANT (**NumE**, Nom_Etudiant, Prenom_Etudiant)

Formalisation du modèle relationnel :

Clé étrangère

- **Clé étrangère d'une relation** : attribut(s) qui forment la clé primaire d'une autre relation (ou dont la valeur est unique dans l'autre relation)
- Notation : clé étrangère en italique dans le schéma (ou soulignée en pointillés)

Exemple

DIPLÔME(numD, spécialité, nbAnnee, totalCredit)

ETUDIANT(numE, nom, prenom, *numD*)

numD clé étrangère dans ETUDIANT qui référence numD clé primaire dans la relation DIPLOME

Formalisation du modèle relationnel :

Les règles d'intégrité

- Les règles d'intégrité sont les assertions qui doivent être vérifiées par les données contenues dans une base
 - Intégrité du domaine
 - Intégrité de Clé
 - Intégrité référentielle

Formalisation du modèle relationnel : Intégrité du domaine

- **Contraintes liées au domaine** : les données doivent vérifier certaines conditions pour être cohérentes.
- ✓ contrôle des valeurs des attributs
- ✓ contrôle entre valeurs des attributs

Exemples :

superficie > 0

millésime > 1900

annéeDébutProjet < annéeFinProjet

Formalisation du modèle relationnel : Intégrité de Clé

- Les **valeurs des clés primaires** doivent être :
 - Uniques
 - Non NULL
- La valeur NULL est utilisée pour représenter une information inconnue

Formalisation du modèle relationnel : Intégrité référentielle (1/2)

- Les valeurs de clés étrangères sont NULL ou sont des valeurs de la clé primaire auxquelles elles font référence
 - Relation dépendantes
- Dépendances : Liaisons de un à plusieurs exprimées par des attributs particuliers : clés étrangères ou clés primaire
 - Permettent d'éviter les anomalies de mises à jour

Formalisation du modèle relationnel : Intégrité référentielle (2/2)

Exemple

DIPLOME(noDiplome, nomD, nbAnnee, totalCredit)

ETUDIANT(noEtudiant, nomE, prenomE, adresseE)

INSCRIT(noDiplome, noEtudiant, année)

- Insertion tuple avec noEtudiant = X dans INSCRIT nécessite
→ vérification si X existe dans ETUDIANT
- Suppression tuple noEtudiant = X dans ETUDIANT nécessite
 - Soit interdire la suppression si X existe dans INSCRIT
 - Soit supprimer en cascade tuple X dans INSCRIT
 - Soit modifier en cascade X=NULL dans INSCRIT

Formalisation du modèle relationnel :

Exemple BDR

Exemple

Client(noClient, nom, prénom, adresse)

Produit(noProduit, libellé, prixUnitaire)

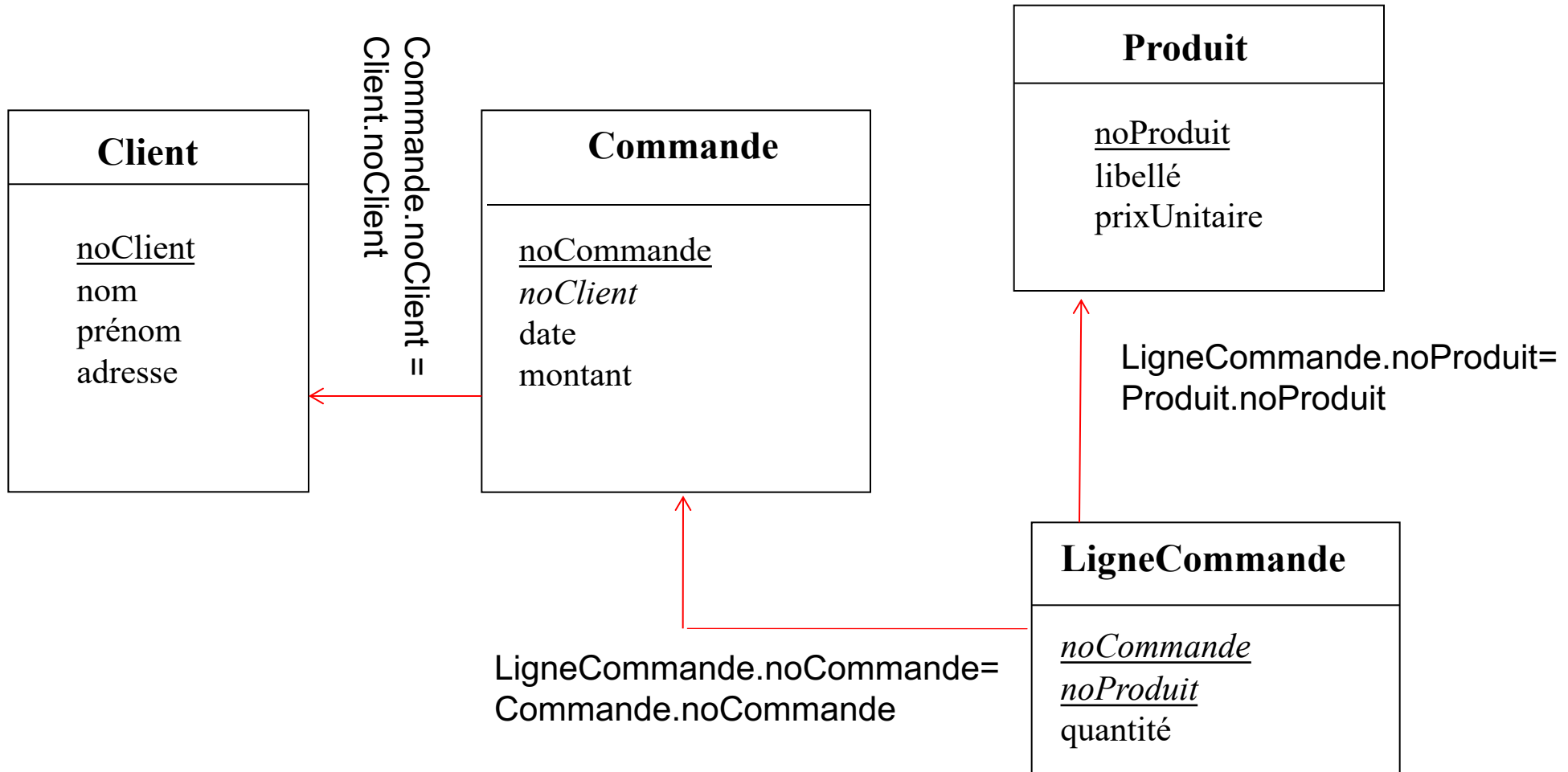
Commande(noCommande, noClient, date, montant)

LigneCommande(noCommande, noProduit, quantité)

- Quelle est la clé (primaire) de chaque relation ?
- Quelles sont les clés étrangères ?
- Représentation graphique du schéma des relations

Formalisation du modèle relationnel :

Représentation graphique de la BDR



La condition sur la flèche n'est pas forcément mentionnée lorsque les attributs formant les clés étrangères ont le même nom que les attributs cibles

Représentation graphique des schémas des tables HR

Dans l'instance Oracle accessible pour les TPs, il existe un utilisateur fictif nommé HR (Human Resources) propriétaire d'un ensemble de tables accessibles à tous les utilisateurs et dont les schémas sont les suivants (clés en gras, clés étrangères en italique) :

```
EMPLOYEES (EMPLOYEE_ID, FIRST_NAME, LAST_NAME, EMAIL, PHONE_NUMBER, HIRE_DATE, JOB_ID,
           SALARY, COMMISSION_PCT, MANAGER_ID, DEPARTMENT_ID)
COUNTRIES (COUNTRY_ID, COUNTRY_NAME, REGION_ID)
DEPARTMENTS (DEPARTMENT_ID, DEPARTMENT_NAME, MANAGER_ID, LOCATION_ID)
JOBS (JOB_ID, JOB_TITLE, MIN_SALARY, MAX_SALARY)
JOB_HISTORY (EMPLOYEE_ID, START_DATE, END_DATE, JOB_ID, DEPARTMENT_ID)
LOCATIONS (LOCATION_ID, STREET_ADDRESS, POSTAL_CODE, CITY, STATE_PROVINCE, COUNTRY_ID)
REGIONS (REGION_ID, REGION_NAME)
```

Quelques précisions sur ces tables

La base de données "HR" (Human Ressources) décrit les ressources humaines d'une importante société implantée dans plusieurs pays à travers le monde.

La table **REGIONS** décrit différentes régions du monde (Europe, Amérique...) dans lesquels se situent les pays décrits dans la table **COUNTRIES** (France, Brazil...).

Chaque service ou département de la société (table **DEPARTMENTS**) est localisé à une adresse (table **LOCATIONS**) dans un pays (table **COUNTRIES**).

La table **JOBS** décrit les différents emplois (ou postes) et précise pour chacun l'intervalle de salaire associé.

La table **EMPLOYEES** décrit les employés et l'emploi qu'ils occupent actuellement ainsi que le département auquel ils sont rattachés. Un employé peut être sous les ordres d'un autre employé (défini par *manager_id*). Lorsqu'un employé change de poste, l'information sur le poste précédent (ainsi que la période associée) et le département précédent est gardée dans la table **JOB_HISTORY**. Un département a un employé à sa tête (*manager_id*)

JOBS J
job_id job_title min_salary max_salary

JOB_HISTORY JH
employee_id start_date end_date job_id department_id

EMPLOYEES E
employee_id first_name last_name email phone_number hire_date job_id salary commission_pct manager_id department_id

DEPARTMENTS D
department_id department_name manager_id location_id

REGIONS R
region_id region_name

COUNTRIES C
country_id country_name region_id

LOCATIONS L
location_id street_address postal_code city state_province country_id

Plan du cours

1. Introduction aux Bases de Données (BD)
2. Modèle relationnel de données
 - 2.1 Éléments du modèle relationnel
 - 2.2 **Algèbre relationnelle**

2.2. Algèbre relationnelle (A.R.)

- L'AR est un langage de manipulation de données relationnelles
(E. CODD, 1970)
- L'AR comporte 8 **opérations** (ou opérateurs) qui s'appliquent des **relations** (opérandes) et donnent une **relation** comme résultat
- **Opération ensembliste**
 - Union - Intersection - Différence
- **Opérations spécifiques**
 - Restriction - Projection - Produit cartésien
 - Jointure - Division

Opérations ensembliste : UNION

- L'union de deux relations R_1 et R_2 de même schéma est une relation R_3 de schéma identique qui a pour n-uplets les n-uplets de R_1 et/ou R_2 (réunir les tuples de deux tables sans doublon)
- On note : $R_3 = R_1 \cup R_2 \rightarrow \text{UNION}(R_1, R_2)$

R_1

Nom	Prenom
Dupond	marc
Durand	jacques

R_2

Nom	Prenom
Dupond	marc
Dupont	jean

$R_3 = R_1 \cup R_2$



Nom	Prenom
Dupond	marc
Durand	jacques
Dupont	jean

Exemple d'union de relations

VINS-1

Numéro	Cru	Millésime	Degré
100	Chablis	1974	12
110	Mecurey	1978	13
120	Mâcon	1977	12

VINS-2

Numéro	Cru	Millésime	Degré
100	Chablis	1974	12
200	Sancerre	1979	11

VINS-3 = UNION (VINS-1,VINS-2)


Numéro	Cru	Millésime	Degré
100	Chablis	1974	12
110	Mecurey	1978	13
120	Mâcon	1977	12
200	Sancerre	1979	11

Opérations ensembliste : INTERSECTION

- L'intersection entre deux relations R_1 et R_2 de même schéma est une relation R_3 de schéma identique qui a pour n-uplets les n-uplets communs à R_1 et R_2 (réunir les tuples présents dans les deux tables)
- On note : $R_3 = R_1 \cap R_2 \rightarrow \text{INTERSECTION}(R_1, R_2)$

R1	Nom	Prenom
	Dupond	marc
	Durand	jacques

R2	Nom	Prenom
	Dupond	marc
	Dupont	jean

$R_3 = R_1 \cap R_2$ 

Nom	Prenom
Dupond	marc

Exemple d'intersection de relations

VINS-1

Numéro	Cru	Millésime	Degré
100	Chablis	1974	12
110	Mecurey	1978	13
120	Mâcon	1977	12

VINS-2

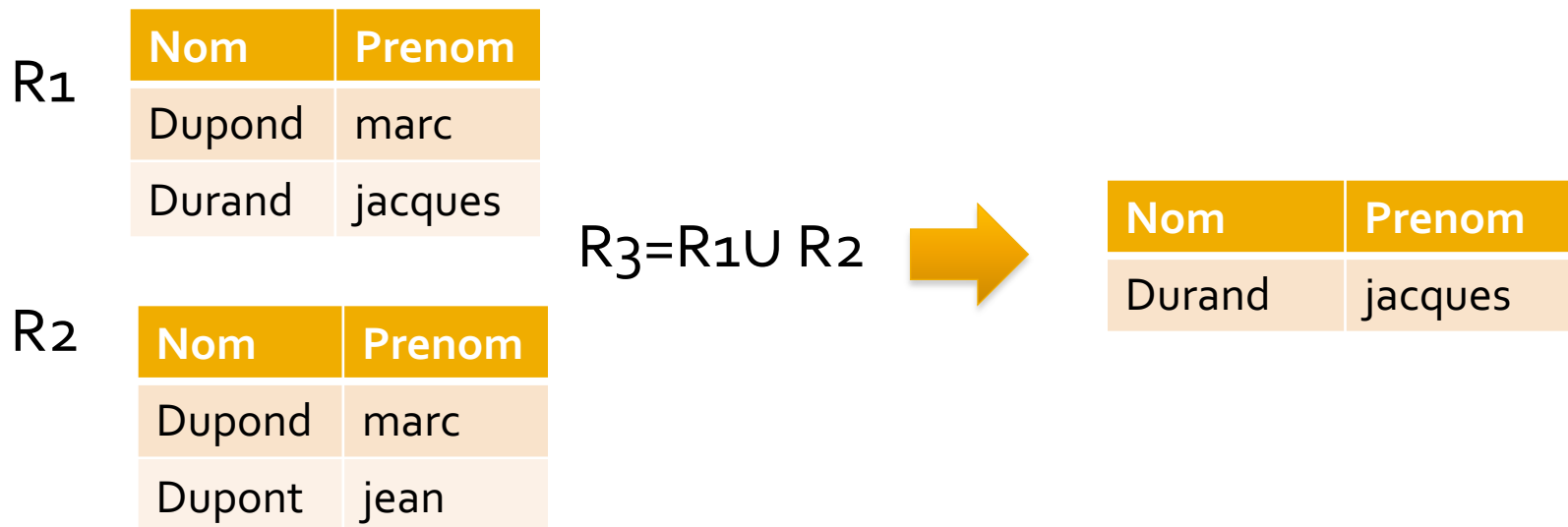
Numéro	Cru	Millésime	Degré
100	Chablis	1974	12
200	Sancerre	1979	11

VINS-4 = INTERSECT(VINS-1,VINS-2)

Numéro	Cru	Millésime	Degré
100	Chablis	1974	12

Opérations ensembliste : DIFFERENCE

- La différence entre deux relations R_1 et R_2 de même schéma est une relation R_3 de schéma identique qui a pour n-uplets les n-uplets de R_1 n'appartenant pas à R_2 (tuples de la première table qui ne sont pas dans la seconde table)
- On note : $R_3 = R_1 - R_2 \rightarrow \text{DIFFERENCE}(R_1, R_2)$



Exemple de différence de relations

VINS-1

Numéro	Cru	Millésime	Degré
100	Chablis	1974	12
110	Mecurey	1978	13
120	Mâcon	1977	12

VINS-2

Numéro	Cru	Millésime	Degré
100	Chablis	1974	12
200	Sancerre	1979	11

VINS-5 = DIFFERENCE (VINS-1, VINS-2)

Numéro	Cru	Millésime	Degré
110	Mecurey	1978	13
120	Mâcon	1977	12

Opérations spéciales : RESTRICTION

- La restriction d'une relation R_1 est une relation R_2 de même schéma n'ayant que les n -uplets de R_1 répondant à la condition énoncée (Supprime les tuples qui ne vérifient pas une condition)
- On note : $R_2 = \text{RESTRICT}(R_1, \langle \text{condition} \rangle)$
- La condition C d'une restriction est une formule logique quelconque avec des connecteurs **ET**, **OU**, **NON** entre conditions simples de la forme $A_i \theta a$ où
 - A_i est un nom d'attribut
 - a est un élément du domaine de A_i (constante)
 - θ est un opérateur de comparaison ($=, <, >, \neq, \geq, \leq$)

ex. $(\text{Cru} = \text{"Chablis"} \text{ OU } \text{Cru} = \text{"Mâcon"}) \text{ ET } \text{Millésime} < 1988$

Opérations spéciales : RESTRICTION

Ex.

PRODUIT

idPro	nomP	Qtité	prix
1	vis	1200	0.12
2	clous	4600	0.09
3	boulons	6500	0.16

Q. Liste des produits dont la quantité >3000

R2=RESTRICT(PRODUIT, Qtité>3000)

idPro	nomP	Qtité	prix
2	clous	4600	0.09
3	boulons	6500	0.16

Exemple de restriction de relation

VINS

Numéro	Cru	Millésime	Degré
100	Chablis	1974	12
110	Mecurey	1978	13
120	Mâcon	1977	12
200	Sancerre	1977	12

VINS-8 = RESTRICT (VINS ; Degré = 12)

Numéro	Cru	Millésime	Degré
100	Chablis	1974	12
120	Mâcon	1977	12
200	Sancerre	1977	12

Opérations spéciales : RESTRICTION

- La projection d'une relation R_1 est la relation R_2 obtenue en supprimant les attributs de R_1 non mentionnés puis en éliminant éventuellement les tuples (Supprime des attributs non souhaités et les doublons !)
- On note : $R_2 = \text{PROJECT}(R_1, A_2, A_3, \dots, A_k)$

Ex.

PRODUIT

idPro	nomP	Qtité	prix
1	vis	1200	0.12
2	clous	4600	0.09
3	boulons	6500	0.16

Q. Liste des références et prix des produits

$R_2 = \text{PROJECT}(\text{PRODUIT}, \text{idPro}, \text{prix})$

idPro	prix
1	0.12
2	0.09
3	0.16

Exemple de projection de relation

VINS

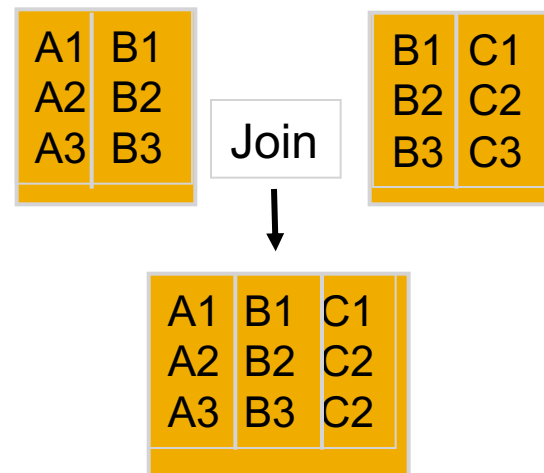
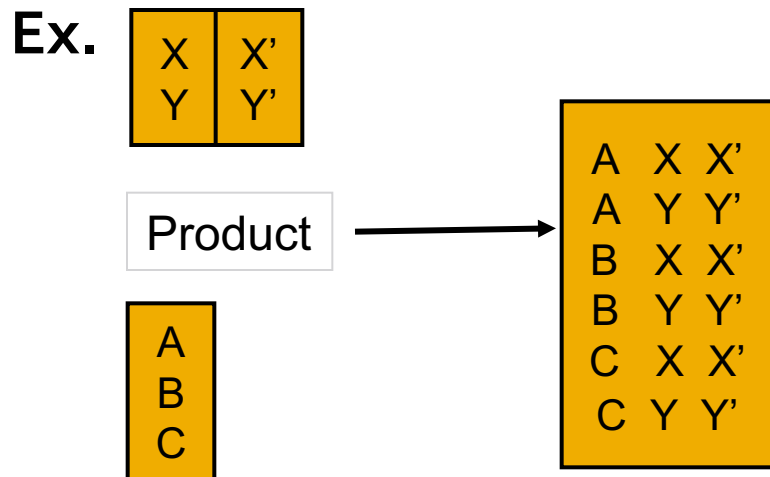
Numéro	Cru	Millésime	Degré
100	Chablis	1974	12
110	Mecurey	1978	13
120	Mâcon	1977	12
200	Sancerre	1977	12

VINS-7 = PROJECT (VINS ; Millésime, Degré)

Millésime	Degré
1974	12
1978	13
1977	12

Opérations spéciales : JOINTURE

- La jointure de deux relations R_1 et R_2 est une relation R_3 dont les n-uplets sont obtenus en concaténant les nuplets de R_1 avec ceux de R_2 (et en ne gardant que ceux qui vérifient la condition de liaison)
- On note : $R_3 = R_1 \times R_2 \rightarrow R_3 = \text{PRODUCT}(R_1, R_2)$
- On note : $R_3 = R_1 \times R_2 (<\text{condition}>) \rightarrow R_3 = \text{JOIN}(R_1, R_2, <\text{condition}>)$



Opérations spéciales : JOINTURE

Exemple

VENTE

numCl	numPro	date	qte
1	V	1/1/18	20
2	C	2/5/18	45
3	B	6/4/18	50

PRODUIT

numPro	nomP	dispo	prix
V	vis	1200	0.12
C	clous	4600	0.09
B	boulons	6500	0.16

Q. Donner pour chaque vente la référence du produit, son nom, son prix, le num du client, la date et la quantité vendue

R = PRODUCT(VENTE, PRODUIT, VENTE.numPro = PRODUIT.numPro)

numCl	numPro	Date	Qte	nomP	prix
1	V	1/1/18	20	vis	0.12
2	C	2/5/18	45	clous	0.09
3	B	6/4/18	50	boulons	0.16

Exemple de produit cartésien de relations

VINS

Numéro	Cru	Millésime	Degré
100	Chablis	1974	12
200	Sancerre	1979	11

VITICULTEURS

Nom	Ville	Région
Nicolas	Pouilly	Bourgogne
Martin	Bordeaux	Bordelais

VIGNOBLE = PRODUCT (VINS, VITICULTEURS)

Numéro	Cru	Millésime	Degré	Nom	Ville	Région
100	Chablis	1974	12	Nicolas	Pouilly	Bourgogne
100	Chablis	1974	12	Martin	Bordeaux	Bordelais
200	Sancerre	1979	11	Nicolas	Pouilly	Bourgogne
200	Sancerre	1979	11	Martin	Bordeaux	Bordelais

Exemple de jointure de relations

VINS

Numéro	Cru	Millésime	Degré
100	Chablis	1974	12
110	Mecurey	1978	13
120	Mâcon	1977	12

VITICULTEURS

Nom	Ville	Région
Nicolas	Pouilly	Bourgogne
Félix	Mâcon	Bourgogne

V-BIS = JOIN (VINS, VITICULTEURS / Cru = Ville)

Numéro	Cru	Millésime	Degré	Nom	Ville	Région
120	Mâcon	1977	12	Félix	Mâcon	Bourgogne

Opérations spéciales : AUTO-JOINTURE

- Une auto-jointure est une jointure d'une relation avec elle-même

Exemple

Q. Quels sont les noms des clients qui habitent la même ville que Dupond
CLIENT

numCl	nomC	ville
C1	Dupond	Grenoble
C2	Durand	Paris
C3	Dupont	Grenoble

$R_1 = \text{JOIN}(\text{CLIENT As CL}_1, \text{CLIENT As CL}_2, \text{CL}_1.\text{Ville} = \text{CL}_2.\text{ville})$

$R_2 = \text{RESTRICT}(R_1, \text{nomC} = \text{'Dupont'})$

$R_3 = \text{PROJECT}(R_2, \text{nomC})$

nomC
Dupond
Dupont

Opérations spéciales : JOINTURE externe

- **Jointure externe** : On garde les tuples d'une des tables qui n'ont pas de correspondant dans l'autre table

Exemple :

1. Liste des clients avec leurs produits en incluant les clients qui n'ont rien commandé

1. Inclure les tuples « célibataires » dans la jointure

- Jointure externe à gauche

LEFT-JOIN (R, S ; C) (jointure de R et S + «célibataires» de R)

- Jointure externe à droite

RIGHT-JOIN(R,S ; C) (jointure de R et S + «célibataires» de S)

Exemple de jointure naturelle de relations

Chercher les informations sur les vins et leur viticulteur

VINS-1

Numéro	Cru	Millésime	Degré
150	Riesling	1984	11
110	Mecurey	1978	13
120	Mâcon	1977	12

VITIC

Nom	Numéro	Région
Nicolas	150	Alsace
Félix	120	Bourgogne

VINS-C = JOIN (VINS-1, VITIC)

Numéro	Cru	Millésime	Degré	Nom	Région
150	Riesling	1984	11	Nicolas	Alsace
120	Mâcon	1977	12	Félix	Bourgogne

Exemple de jointure externe à gauche

Chercher les informations sur les vins et leur viticulteur en incluant les vins qui n'ont pas de viticulteur

VINS-1

Numéro	Cru	Millésime	Degré
150	Riesling	1984	11
110	Mecurey	1978	13
120	Mâcon	1977	12

VITIC

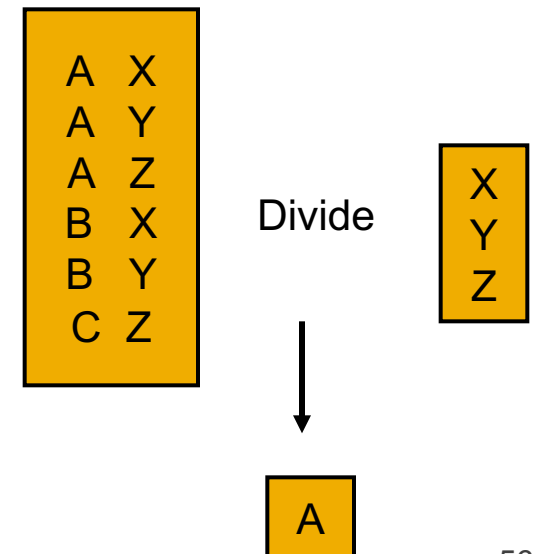
Nom	Numéro	Région
Nicolas	150	Alsace
Félix	120	Bourgogne

VINS-C = LEFT-JOIN (VINS-1,VITIC)

Numéro	Cru	Millésime	Degré	Nom	Région
150	Riesling	1984	11	Nicolas	Alsace
110	Mecurey	1978	13	null	null
120	Mâcon	1977	12	Félix	Bourgogne

Opérations spéciales : DIVISION

- Soit deux relations $R_1 (A_1, \dots, A_n, B_1, \dots, B_m)$ et $R_2 (B_1, \dots, B_m)$. Si le schéma de **R_2 est un sous-schéma de R_1** . La **division de R_1 par R_2** est une relation R_3 dont :
 - R_3 est le sous-schéma complémentaire de R_2 par rapport à R_1
 - un n-uplet $(a_1, a_2, \dots, a_n) \in R_3$ si $(a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_m)$ appartient à R_1 pour tous $(b_1, b_2, \dots, b_m) \in R_2$.
- On note **$R_3 = \text{DIVIDE}(R_1, R_2)$**
- **DIVIDE** : Sélectionner les parties de tuples d'une table qui apparaissent avec tous les tuples de la seconde table



Exemple de division de relations

Produit

numProd
P1
P2
P3

Stock

numProd	numDep
P1	D1
P1	D2
P1	D4
P2	D1
P2	D2
P2	D4
P3	D1
P3	D4

Numéro des dépôts stockant tous les produits :

$\text{DIVIDE}(\text{Stock}, \text{Produit})$

numDep
D1
D4

Opérations spéciales : DIVISION

Exemple

Q. Donner la liste des clients qui ont acheté **tous** les produits de plus de 0.12€.

DIVIDE (C, P)

C = **PROJECT** (Commande; numCl, numPro)

P' = **RESTRICT** (Produit; prix > 0.12)

P = **PROJECT** (P'; numPro)

- La division permet de répondre à des questions qui sont formulées avec le quantificateur universel : "**pour tout** ..."

Exemple d'une BDR

Base de données de Vente par Correspondance

Client (noClient, nom, prénom, adresse)

Produit (noProduit, libellé, prixUnitaire)

Commande (noCommande, *noClient*, date, montant)

LigneCommande (*NoCommande*, *noProduit*, quantité)

- 1) Libellé et prix unitaire de chaque produit.
- 2) Libellé des produits dont le prix est inférieur à 50€.
- 3) Trouver les produits commandés en quantité supérieure à 10 (en une commande) et dont le prix dépasse 100€. Afficher le numéro de chaque produit, son libellé ainsi que le numéro de la commande.**
- 4) Numéro des produits qui n'ont pas été commandés.
- 5) Nom et prénom des clients ayant commandé le produit numéro 56.

Requête et arbre algébrique

Question 3 :

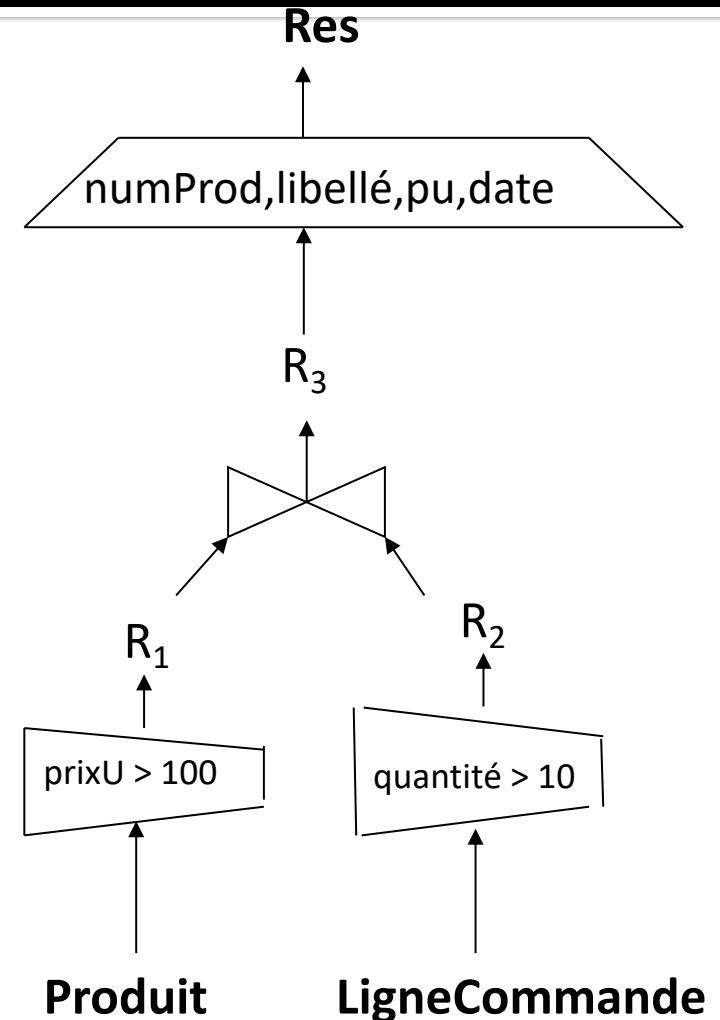
Notons **Res** la relation résultat
et R_1 , R_2 , R_3 des relations intermédiaires

$R_1 = \text{RESTRICT}(\text{Produit}; \text{prixUnitaire} > 100)$

$R_2 = \text{RESTRICT}(\text{LigneCommande}; \text{quantité} > 10)$

$R_3 = \text{JOIN}(R_1, R_2)$

$\text{Res} = \text{PROJECT}(R_3; \text{noProduit}, \text{libellé}, \text{noCommande})$



Algèbre relationnelle et évaluation de requêtes dans les SGBDR

- Les requêtes (SQL) dans les SGBDR sont transformées en requêtes algébriques
- 5 opérateurs sont suffisants : project, restrict, product, union, différence
 - Les autres peuvent se définir en fonction de ces cinq opérateurs (INTERSECTION, JOIN, DIVISION)
- Quelques propriétés utiles pour l'**optimisation** de requêtes

$$\text{RESTRICT}(R; c1 \text{ ET } c2) = \text{RESTRICT}(\text{RESTRICT}(R; c2); c1))$$

$$\text{RESTRICT}(\text{RESTRICT}(R; c2); c1)) = \text{RESTRICT}(\text{RESTRICT}(R; c1); c2))$$

$$\text{PROJECT}(\text{PROJECT}(R; \text{liste1}); \text{liste2}) = \text{PROJECT}(R, \text{liste2}) \text{ si } \text{liste2} \subseteq \text{liste1}$$

$$\text{JOIN}(R, S; c) = \text{JOIN}(S, R; c)$$

$$\text{JOIN}(\text{JOIN}(R, S; c1), T; c2) = \text{JOIN}(R, \text{JOIN}(S, T; c2); c1)$$