

Bases de Données Relationnelles

Chapitre 1 Introduction générale

- I. Notions intuitives**
- II. Objectifs et avantages des SGBD**
- III. L'architecture ANSI/SPARC**
- IV. Notion de modélisation des données**
- V. Survol des différents modèles de données**
- VI. Bref historique, principaux SGBD commercialisés**

I Notions intuitives

• Base de données

ensemble structuré de données apparentées qui modélisent un univers réel

Une BD est faite pour enregistrer des faits, des opérations au sein d'un organisme
(administration, banque, université, hôpital, ...)

Les BD ont une place essentielle dans l'informatique

• Système de Gestion de Base de Données (SGBD)

DATA BASE MANAGEMENT SYSTEM (DBMS)

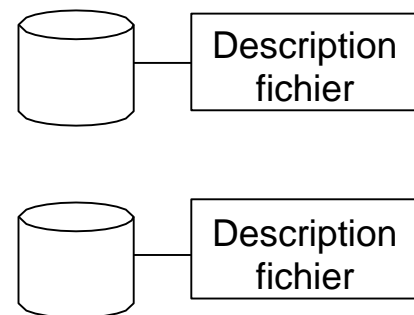
système qui permet de gérer une BD partagée par plusieurs utilisateurs simultanément

. Des fichiers aux Base de Données

Séparation des données et des programmes

FICHIER

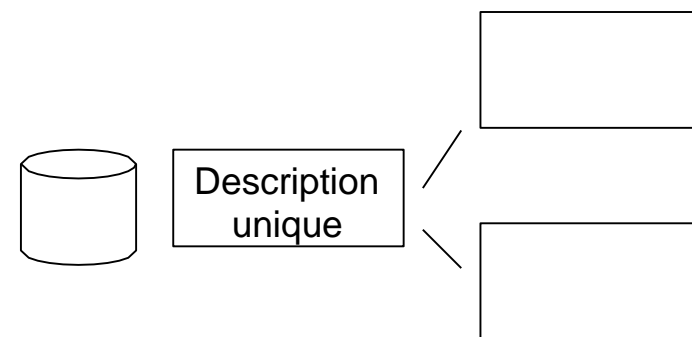
Les données des fichiers sont décrites dans les programmes



Programmes

BASE DE DONNEES

Les données de la BD sont décrites hors des programmes dans la base elle-même



Programmes

La multiplication des fichiers entraînait la *redondance* des données, ce qui rendait difficile les mises à jour.

D'où l'idée *d'intégration* et de *partage* des données

II Objectifs et avantages des SGBD

Que doit permettre un SGBD ?

□ Décrire les données

indépendamment des applications (de manière intrinsèque)

⇒ *langage de définition* des données

DATA DEFINITION LANGUAGE (DDL)

□ Manipuler les données

Interroger et mettre à jour les données sans préciser d'algorithme d'accès

dire QUOI sans dire COMMENT

langage de *requêtes* déclaratif

ex.:

quels sont les noms des produits de prix < 200DH ?

⇒ *langage de manipulation* des données

DATA MANIPULATION LANGUAGE (DML)

▣ Contrôler les données

intégrité

vérification de contraintes d'intégrité
ex.: le salaire doit être compris entre
3000DH et 20000DH

confidentialité

contrôle des droits d'accès, autorisation

⇒ *langage de contrôle des données*

DATA CONTROL LANGUAGE (DCL)

▣ Partage

une BD est partagée entre plusieurs utilisateurs en même temps
⇒ contrôle des accès concurrents

notion de **transaction**

L'exécution d'une transaction doit préserver la cohérence de la BD

▣ Sécurité

reprise après panne, journalisation

▣ Performances d'accès

index (hashage, arbres balancés ...)

▣ **Indépendance physique**

Pouvoir modifier les structures de stockage ou les index sans que cela ait de répercussion au niveau des applications

Les disques, les méthodes d'accès, les modes de placement, le codage des données ne sont pas apparents

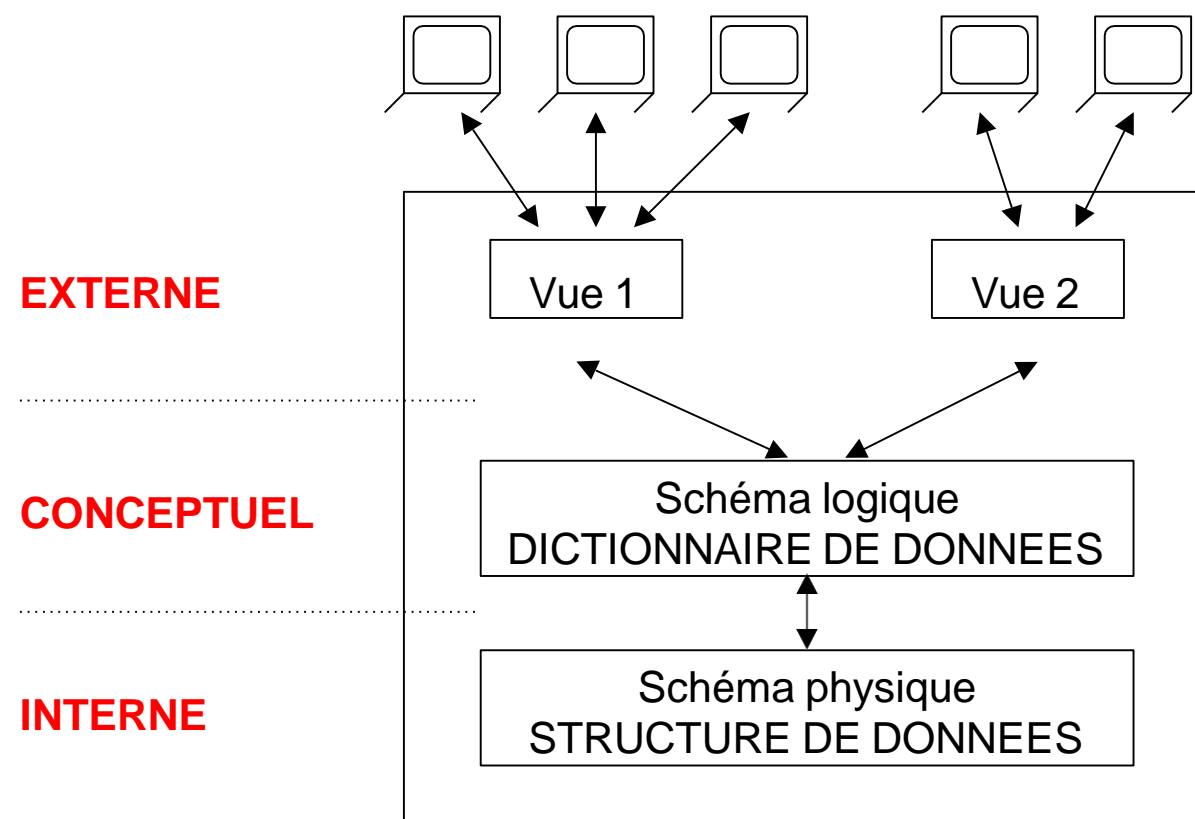
▣ **Indépendance logique**

Permettre aux différentes applications d'avoir des vues différentes des mêmes données

Permettre au DBA de modifier le schéma logique sans que cela ait de répercussion au niveau des applications

III L'architecture ANSI/SPARC

- proposition en 75 de l' ANSI/SPARC
(Standard Planning And Requirement Comitte)
- 3 niveaux de représentation des données



SGBD
Niveaux de représentation des données

▣ **Le niveau externe**

Le concept de **vue** permet d'obtenir l'indépendance logique

La modification du schéma logique n'entraîne pas la modification des applications
(une modification des vues est cependant nécessaire)

Chaque vue correspond à la perception d'une partie des données, mais aussi des données qui peuvent être synthétisées à partir des informations représentées dans la BD (par ex. statistiques)

▣ **Le niveau conceptuel**

il contient la description des données et des contraintes d'intégrité (Dictionnaire de Données)

le schéma logique découle d'une activité de modélisation

▣ **Le niveau interne**

il correspond aux structures de stockage et aux moyens d'accès (index)

Pour résumer :

Les fonctions des SGBD

• DEFINITION DES DONNEES

⇒ **Langage de définition des données (DDL)**
(conforme à un modèle de données)

• MANIPULATION DES DONNEES

Interrogation

Mise à jour

insertion, suppression, modification

⇒ **Langage de manipulation des données (DML)**
(langage de requête déclaratif)

• CONTRÔLE DES DONNEES

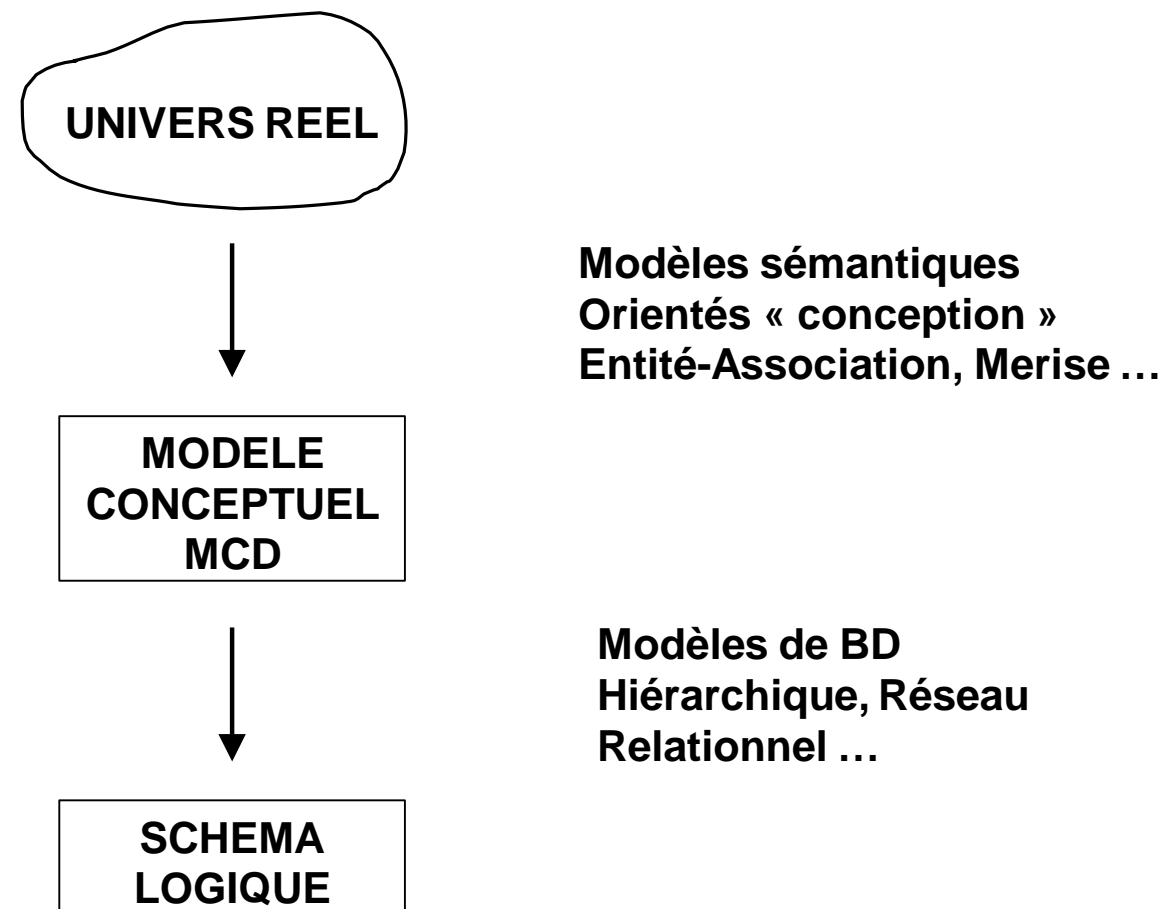
Contraintes d'intégrité

Contrôle des droits d'accès

Gestion de transactions

⇒ **Langage de contrôle des données (DCL)**

IV Notion de modélisation des données



- Les modèles de BD sont souvent trop limités pour pouvoir représenter directement le monde réel
- Méthodologies de conception présentées en ACSI, SGBD2

LE MODÈLE RELATIONNEL

- En 1970, CODD présente le modèle relationnel
- Schéma logique représenté par des **RELATIONS**

LE SCHÉMA RELATIONNEL

Le schéma relationnel est l'ensemble des RELATIONS qui modélisent le monde réel

- Les relations représentent les **entités** du monde réel (comme des personnes, des objets, etc.) ou les **associations** entre ces entités
- Passage d'un schéma conceptuel E-A à un schéma relationnel
 - une entité est représentée par la relation :
nom_de_l'entité (liste des attributs de l'entité)
 - une association M:N est représentée par la relation
: nom_de_l'association (
liste des identifiants des entités participantes,
liste des attributs de l'association)

CLIENT (IdCli, nom, ville)

PRODUIT (IdPro, nom, marque, prix, qstock)

VENTE (IdCli#, IdPro#, date, qte)

Représentation des données sous forme de **tables** :

CLIENT	IdCli	Nom	Ville
	X	Saïd	Fès
	Y	Jamal	Fès
	Z	Badr	Nador

PRODUIT	IdPro	Nom	Prix	Marque	Qstock
	P	PS1	100	IBM	10
	Q	PS2	100	Apple	10
	R	PS3	100	IBM	10
	S	PS4	100	Microsoft	10

VENTE	IdCli	IdPro	Date	Qte
	X	P		1
	X	Q		2
	X	R		3
	Y	P		4
	Y	Q		5
	Z	Q		6

LES AVANTAGES DU MODÈLE RELATIONNEL

- **SIMPLICITE DE PRÉSENTATION**

- représentation sous forme de **tables**

- **OPÉRATIONS RELATIONNELLES**

- algèbre relationnelle
- langages assertionnels

- **INDEPENDANCE PHYSIQUE**

- optimisation des accès
- stratégie d'accès déterminée par le système

- **INDEPENDANCE LOGIQUE**

- concept de **VUES**

- **MAINTIEN DE L'INTEGRITÉ**

- contraintes d'intégrité définies au niveau du schéma

VI Bref historique, principaux systèmes

Années 60 Premiers développements des BD

- fichiers reliés par des pointeurs
- systèmes IDS 1 et IMS 1 précurseurs des SGBD modernes

Années 70 Première génération de SGBD

- apparition des premiers SGBD
- séparation de la description des données de la manipulation de celles-ci par les applications
- modèles hiérarchique et réseau CODASYL
- langages d'accès navigationnels
- SGBD IDMS, IDS 2 et IMS 2

Années 80 Deuxième génération

- modèle relationnel
- les SGBDR représentent l'essentiel du marché BD (aujourd'hui)
- architecture répartie client-serveur

Années 90 Troisième génération

- modèles de données plus riches
- systèmes à objets
OBJECTSTORE, O2

Principaux systèmes

- Oracle
- DB2 (IBM)
- Ingres
- Informix
- Sybase
- SQL Server (Microsoft)
- O2
- Gemstone

Sur micro :

- Access
- Paradox
- FoxPro
- 4D
- Windev

Sharewares :

- MySQL
- MSQL
- Postgres
- InstantDB

I. LES CONCEPTS

**II. LES DÉPENDANCES
FONCTIONNELLES**

III. LES RÈGLES D'INTÉGRITÉ

IV. LES FORMES NORMALES

I LES CONCEPTS

- LE DOMAINE
- LA RELATION
- LES N-UPLETS
- LES ATTRIBUTS
- LE SCHÉMA D'UNE RELATION
- LE SCHÉMA D'UNE BDR
- LA REPRÉSENTATION

▣ LE DOMAINE

ensemble de valeurs atomiques d'un certain type
sémantique

Ex. :

$NOM_VILLE = \{ Nador, Fès, Rabat \}$

- les domaines sont les ensembles de valeurs possibles dans lesquels sont puisées les données
- deux ensembles peuvent avoir les mêmes valeurs bien que sémantiquement distincts

Ex. :

$NUM_ELV = \{ 1, 2, \dots, 2000 \}$

$NUM_ANNEE = \{ 1, 2, \dots, 2000 \}$

▣ LA RELATION

sous ensemble du produit cartésien de plusieurs domaines

$$R \subset D1 \times D2 \times \dots \times Dn$$

$D1, D2, \dots, Dn$ sont les domaines de R
 n est le degré ou l'arité de R

Ex.:

Les domaines :

$NOM_ELV = \{ \text{alaoui, slaoui} \}$

$PREN_ELV = \{ \text{said, jamal, badr} \}$ $DATE_NAISS =$

$\{ \text{Date entre 1/1/1990 et 31/12/2020} \}$ NOM_SPORT

$= \{ \text{judo, tennis, foot} \}$

La relation ELEVE

$ELEVE \subset NOM_ELV \times PREN_ELV \times DATE_NAISS$

$ELEVE = \{ (\text{alaoui, said, 1/1/1992}),$
 $(\text{slaoui, badr, 2/2/1994}) \}$

La relation INSCRIPT

$INSCRIPT \subset NOM_ELV \times NOM_SPORT$

$INSCRIPT = \{ (\text{alaoui, judo}), (\text{alaoui, foot}),$
 $(\text{slaoui, judo}) \}$

▣ LES N-UPLETS

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

- un **n-uplet** représente un fait

Ex.:

« alaoui said est un élève né le 1 janvier1992 »

« alaoui est inscrit au judo »

▣ LES ATTRIBUTS

Chaque composante d'une relation est un attribut

- Le nom donné à un attribut est porteur de sens
- Il est en général différent du nom de domaine
- Plusieurs attributs peuvent avoir le même domaine

Ex. :

La relation **TRAJET** :

$$\text{TRAJET} \subset \text{NOM_VILLE} \times \text{NOM_VILLE}$$

Dans laquelle la première composante représente la ville de départ **VD**, la deuxième composante la ville d'arrivée **VA** d'un trajet.

▣ LE SCHÉMA D'UNE RELATION

Le schéma d'une relation est défini par :

- le nom de la relation
- la liste de ses attributs

on note : $R (A_1, A_2, \dots, A_n)$

Ex.:

ELEVE (NOM, PRENOM, NAISS)
INSCRIPT (NOM_ELIV, SPORT)
TRAJET (VD, VA)

▣ LE SCHÉMA D'UNE BDR

Le schéma d'une base de données est défini par :

- l'ensemble des schémas des relations qui la composent

▣ LA REPRÉSENTATION

1 RELATION = 1 TABLE

U1	V1	W1	X1	Y1
U2	V2	W2	X2	Y2
U3	V3	W3	X3	Y3

1 ÉLÉMENT ou n-uplet = 1 LIGNE

LIGNE →

1 élément

U1	V1	W1	X1	Y1

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

1 ATTRIBUT = 1 COLONNE

U1				
U2				
U3				

↑

COLONNE

1 attribut ou propriété

Exemples :

- La relation ELEVE

ELEVE :

	NOM	PRENOM	NAISS
élément →	alaoui	said	1/1/1992
	slaoui	badr	2/2/1994
	filali	ahmad	3/03/91

- La relation INSCRIPT

INSCRIPT :

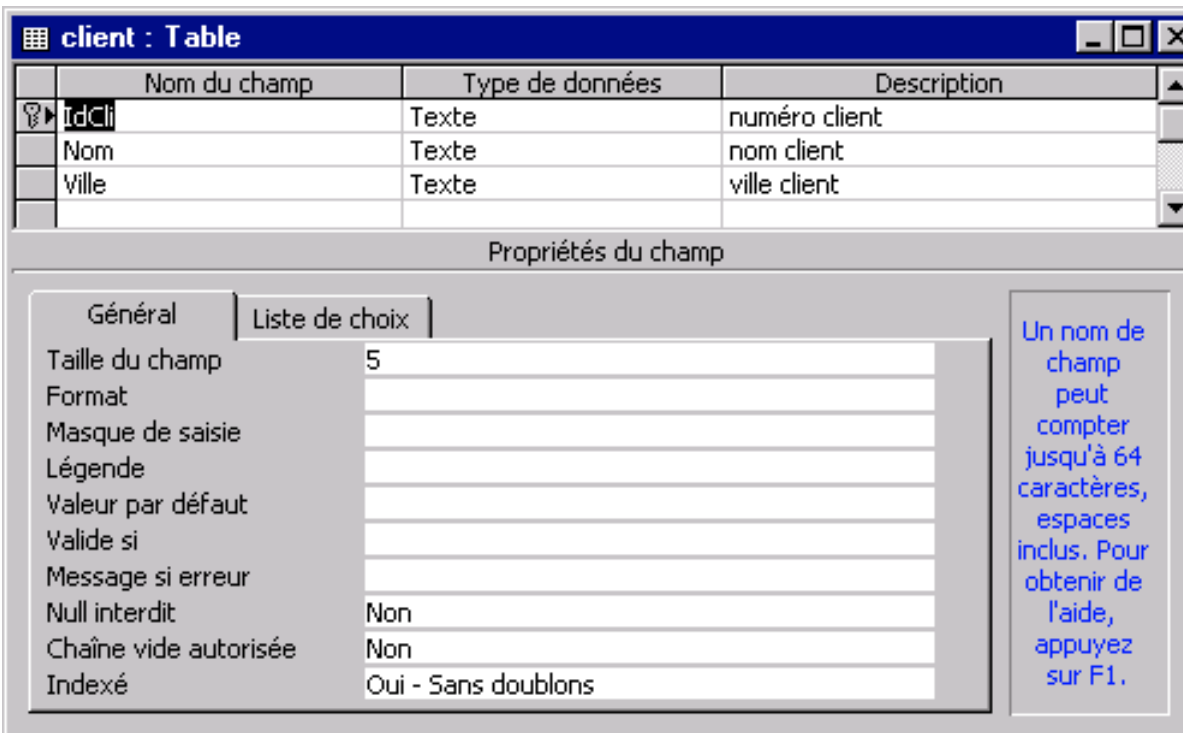
	NOM_ELV	SPORT
élément →	alaoui	judo
	alaoui	foot
	filali	judo

- La relation TRAJET

TRAJET :

	VD	VA
élément →	Nador	Fès
	Fès	Rabat
	Rabat	Nador

Fenêtre Création de Table d'Access



Nom du champ	Type de données	Description
IdCli	Texte	numéro client
Nom	Texte	nom client
Ville	Texte	ville client

Propriétés du champ

Général | Liste de choix

Taille du champ: 5

Format:

Masque de saisie:

Légende:

Valeur par défaut:

Valide si:

Message si erreur:

Null interdit: Non

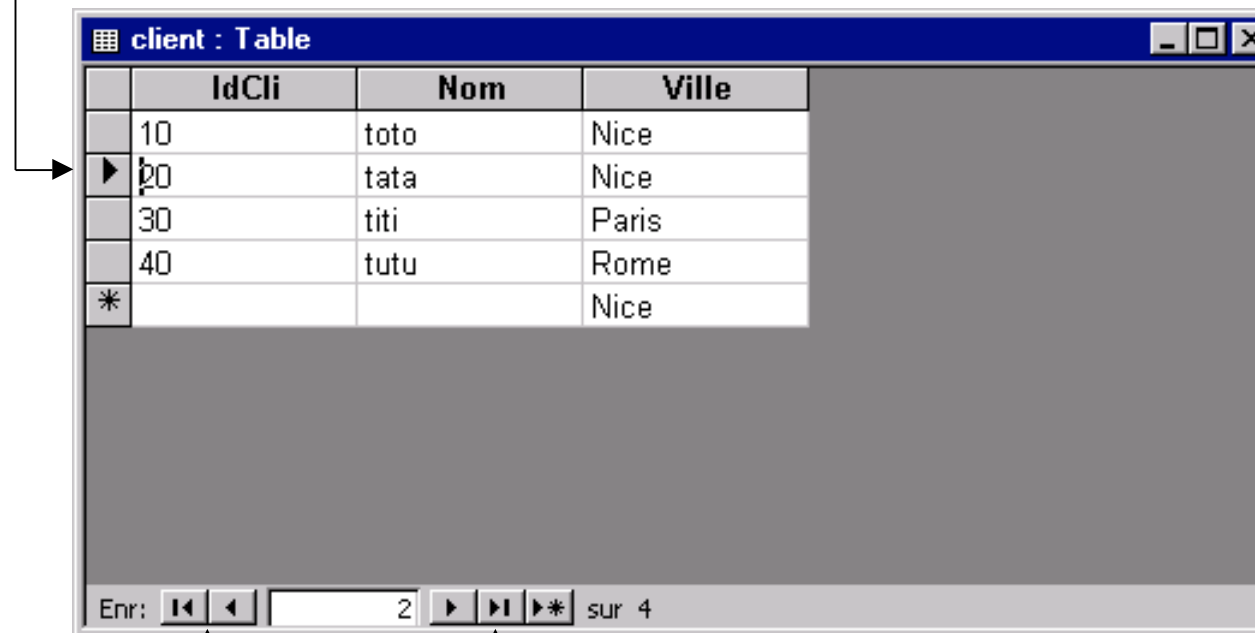
Chaîne vide autorisée: Non

Indexé: Oui - Sans doublons

Un nom de champ peut compter jusqu'à 64 caractères, espaces inclus. Pour obtenir de l'aide, appuyez sur F1.

Affichage d'une table dans Access

Sélecteur d'enregistrement



	IdCli	Nom	Ville
	10	toto	Nice
▶	20	tata	Nice
	30	titi	Paris
	40	tutu	Rome
*			Nice

Enr: 2 sur 4

Boutons de déplacement

III LES RÈGLES D'INTÉGRITÉ

Les règles d'intégrité sont les assertions qui doivent être vérifiées par les données contenues dans une base

Le modèle relationnel impose les contraintes structurelles suivantes :

- **INTÉGRITÉ DE DOMAINE**

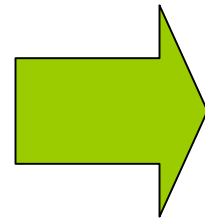
 - **INTÉGRITÉ DE CLÉ**

 - **INTÉGRITÉ RÉFÉRENCIELLE**
- La gestion automatique des contraintes d'intégrité est l'un des outils les plus importants d'une base de données.

 - Elle justifie à elle seule l'usage d'un SGBD.

▣ INTÉGRITÉ DE DOMAINE

Les valeurs d'une colonne de relation doivent appartenir au domaine correspondant

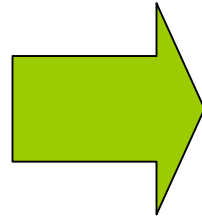


- contrôle des valeurs des attributs
- contrôle entre valeurs des attributs

▣ INTÉGRITÉ DE CLÉ

Les valeurs de clés primaires doivent être :

- uniques
- non NULL



- Unicité de clé
- Unicité des n-uplets

- **Valeur NULL**

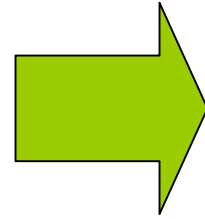
valeur conventionnelle pour représenter une information **inconnue**

- dans toute extension possible d'une relation, il ne peut exister 2 n-uplets ayant même valeur pour les attributs clés

sinon 2 clés identiques détermineraient 2 lignes identiques (d'après la définition d'une clé), ce qui est absurde

▣ INTÉGRITÉ RÉFÉRENCIELLE

Les valeurs de clés étrangères sont 'NULL' ou sont des valeurs de la clé primaire auxquelles elles font référence



- Relations dépendantes

- LES DÉPENDANCES :

Liaisons de un à plusieurs exprimées par des attributs particuliers: **clés étrangères** ou **clés secondaires**

Les contraintes de référence ont un impact important pour les opérations de mises à jour, elles permettent d'éviter les **anomalies** de mises à jour

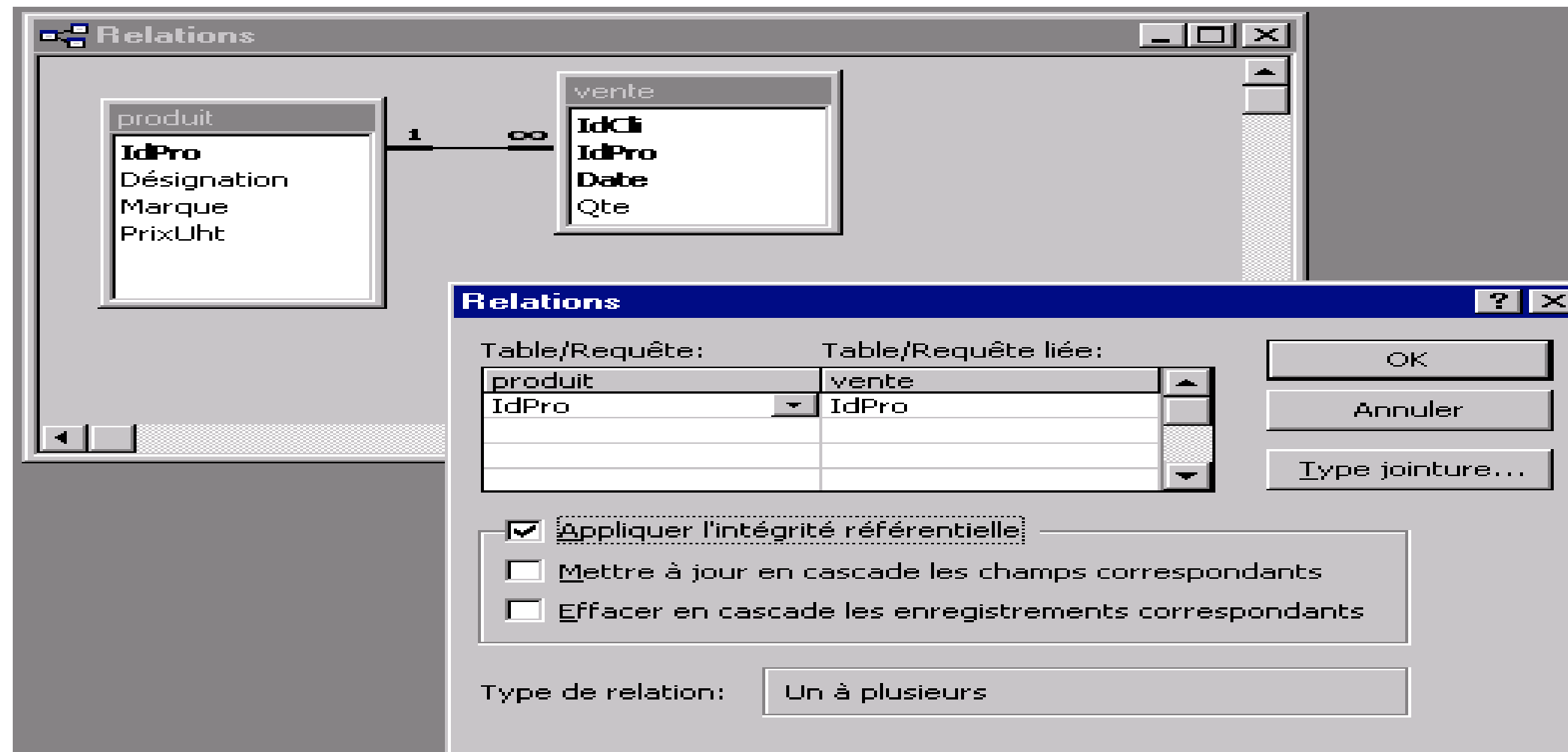
Exemple :

CLIENT (**no_client**, nom, adresse) ACHAT (**no_produit**, no_client, date, qte)

Clé étrangère **no_client** dans ACHAT

- **insertion** tuple **no_client = X** dans ACHAT
 - ∅ vérification si **X** existe dans CLIENT
- **suppression** tuple **no_client = X** dans CLIENT
 - ∅ soit interdire si **X** existe dans ACHAT
 - ∅ soit supprimer en cascade tuple **X** dans ACHAT
 - ∅ soit modifier en cascade **X = NULL** dans ~~ACHAT~~
- **modification** tuple **no_client = X en X'** dans CLIENT
 - ∅ soit interdire si **X** existe dans ACHAT
 - ∅ soit modifier en cascade **X en X'** dans ACHAT

Paramétrage des Relations dans Access



- IdPro de Vente est une clé étrangère qui fait référence à la clé primaire de Produit
- Appliquer l'intégrité référentielle signifie que l'on ne pourra pas avoir, à aucun moment, une ligne de Vente avec un code produit IdPro inexistant dans la table Produit.
- Une valeur de clé étrangère peut être Null

Chapitre 3 Présentation des données

Une fois la base et les tables créées, il faut pouvoir les exploiter.

L'utilisateur final aura besoin de visualiser et saisir des données, d'effectuer des calculs et d'imprimer des résultats.

La réponse à ces problèmes de présentation des données est fournie par :

- **les formulaires**

destinés à être affichés à l'écran

- **les états**

destinés à être imprimés.

I Les formulaires

2 types de formulaires :

- **de présentation des données**

Ils permettent de saisir, ou modifier les données d'une ou plusieurs tables sous une forme visuellement agréable

- **de distribution**

ils ne sont attachés à aucune table, et servent uniquement de page de menu pour orienter l'utilisateur vers d'autres formulaires ou états

Formulaire rudimentaire

A screenshot of a simple form window titled "produit". The form contains four fields: "IdPro" with the value "100", "Désignation" with the value "ps", "Marque" with the value "ibm", and "PrixUht" with the value "5 000,00 F". At the bottom, there is a navigation bar with the text "Enr: 1 sur 4" and several navigation icons.

Fenêtre Conception de Formulaire d'Access

A screenshot of the Access Form Design window, titled "Formulaire2 : Formulaire". The window shows a design grid with three sections: "En-tête de formulaire" (Header), "Détail" (Detail), and "Pied de formulaire" (Footer). The grid is currently empty, and the design view is active. A toolbar on the left side of the window contains various design tools for creating and formatting the form.

Formulaire avec sous-formulaire

The screenshot shows a software application window titled "categorie". The main form contains the following fields:

- IdCat:
- Désignation:
- TauxTva:

Below these fields is a sub-form titled "produit" which contains a table with the following data:

	IdPro	IdC	Désignation	Marque	PrixUht	Qstoc
▶	10	C1	ps	ibm	5 000,00 F	20
	20	C1	mac	apple	7 500,00 F	20
	30	C1	aptiva	ibm	12 000,00 F	10
	40	C1	power mac	apple	20 000,00 F	10
*		C1			0,00 F	0

At the bottom of the sub-form, there are navigation controls: "Enr: [back] [left] [1] [right] [next] [end] sur 4".

At the bottom of the main form, there are navigation controls: "Enr: [back] [left] [1] [right] [next] [end] sur 3".

Permet d'afficher les données de deux tables qui sont en relation l'une avec l'autre.

- Le formulaire principal affiche les données de la table principale
- Le sous formulaire affiche les données de la table liée

Si l'utilisateur change d'enregistrement principal, le sous formulaire est automatiquement mis à jour.

Création d'un formulaire de présentation

1) Définir la propriété Source de données (table ou requête)

Cliquer ici avec le bouton droit, puis sélectionner Propriétés

Boîte des propriétés du formulaire

Format	Données	Événement	Autres	Toutes
Source	produit			
Filtre				
Tri par				
Filtrage autorisé	Oui			
Modif autorisée	Oui			
Suppr autorisée	Oui			
Ajout autorisé	Oui			
Entrée données	Non			
Type Recordset	Feuille de réponse dynamique			
Verrouillage	Aucun			

2) Insérer dans le formulaire les Zones de texte liées aux champs de la Source de données

a) Sélectionner l'outil Zone de texte

b) Insérer la Zone de texte avec son Etiquette associée

c) Définir la propriété Source contrôle de la Zone de texte

The image illustrates the steps to add a text zone to a form. It shows a toolbar with various tools, a form grid with labels 'Texte0' and 'Indépendant', and a properties window for 'Texte0' with tabs for Format, Données, Événement, Autres, and Toutes. The 'Données' tab is active, showing the 'Source contrôle' property set to 'Désignation'.

Propriété	Valeur
Source contrôle	Désignation
Masque de saisie	
Valeur par défaut	
Valide si	
Message si erreur	
Activé	Oui
Verrouillé	Non
Rechercher filtre	Paramètres par défaut

Pour afficher la fenêtre des propriétés d'un contrôle, cliquer dessus avec le bouton droit de la souris

II Les états

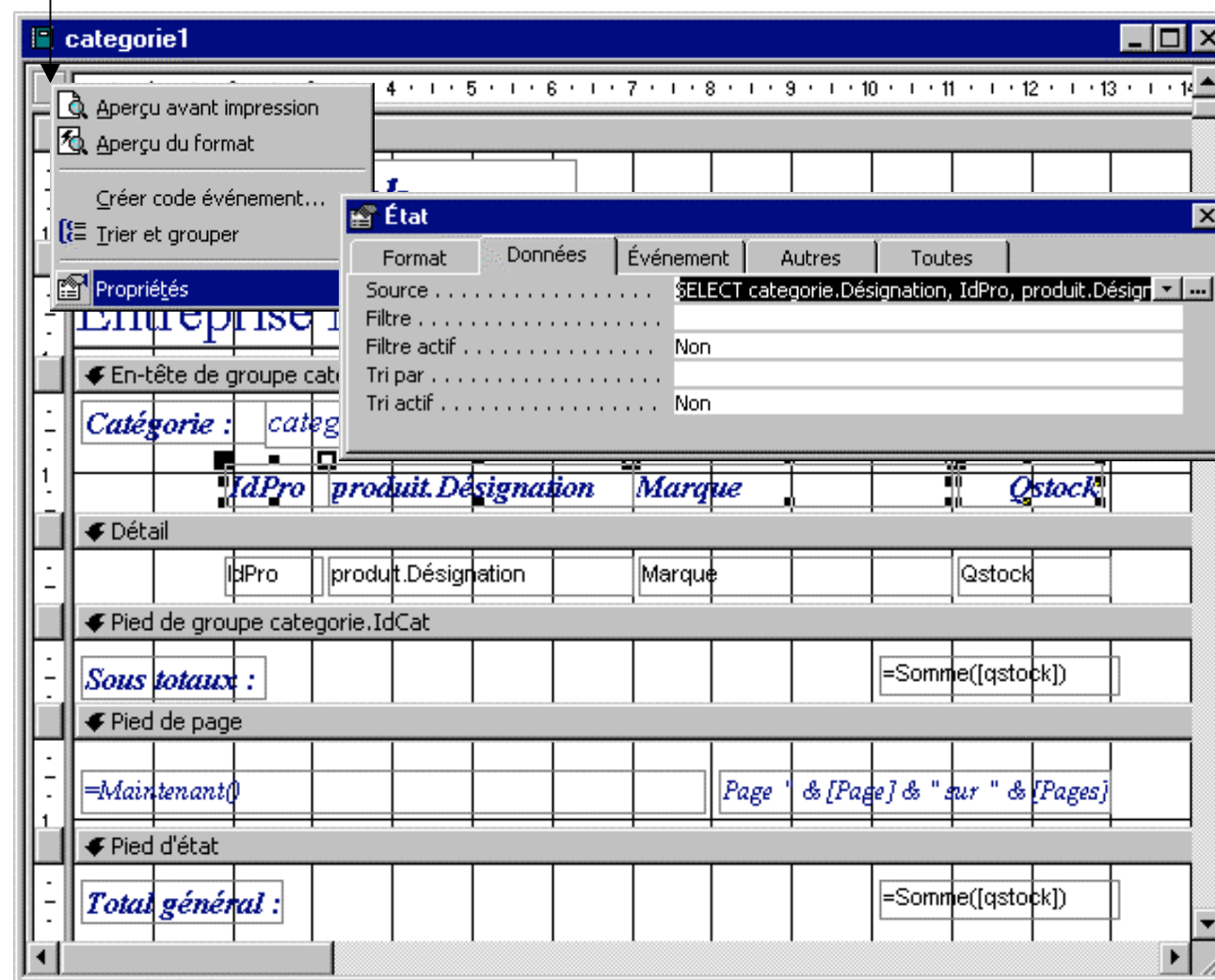
Un état permet d'imprimer des enregistrements, en les groupant et en effectuant des totaux et des sous totaux.

En-tête d'état	→	<i>Etat du Stock</i>			
En-tête de page	→	Entreprise MICRO			
En-tête de groupe	→	Catégorie :		0	
Détail	→	IdPro	Désignation	Marque	Qstock
		10	Ps	Ibm	10
		20	Imac	Apple	20
		30	Aptiva	Ibm	10
Pied de groupe	→	Sous totaux :			40
		...			
Pied de page	→	<i>Jeudi 12 février 1998</i>			<i>Page 1 sur 1</i>
Pied d'état	→	Total général :			200

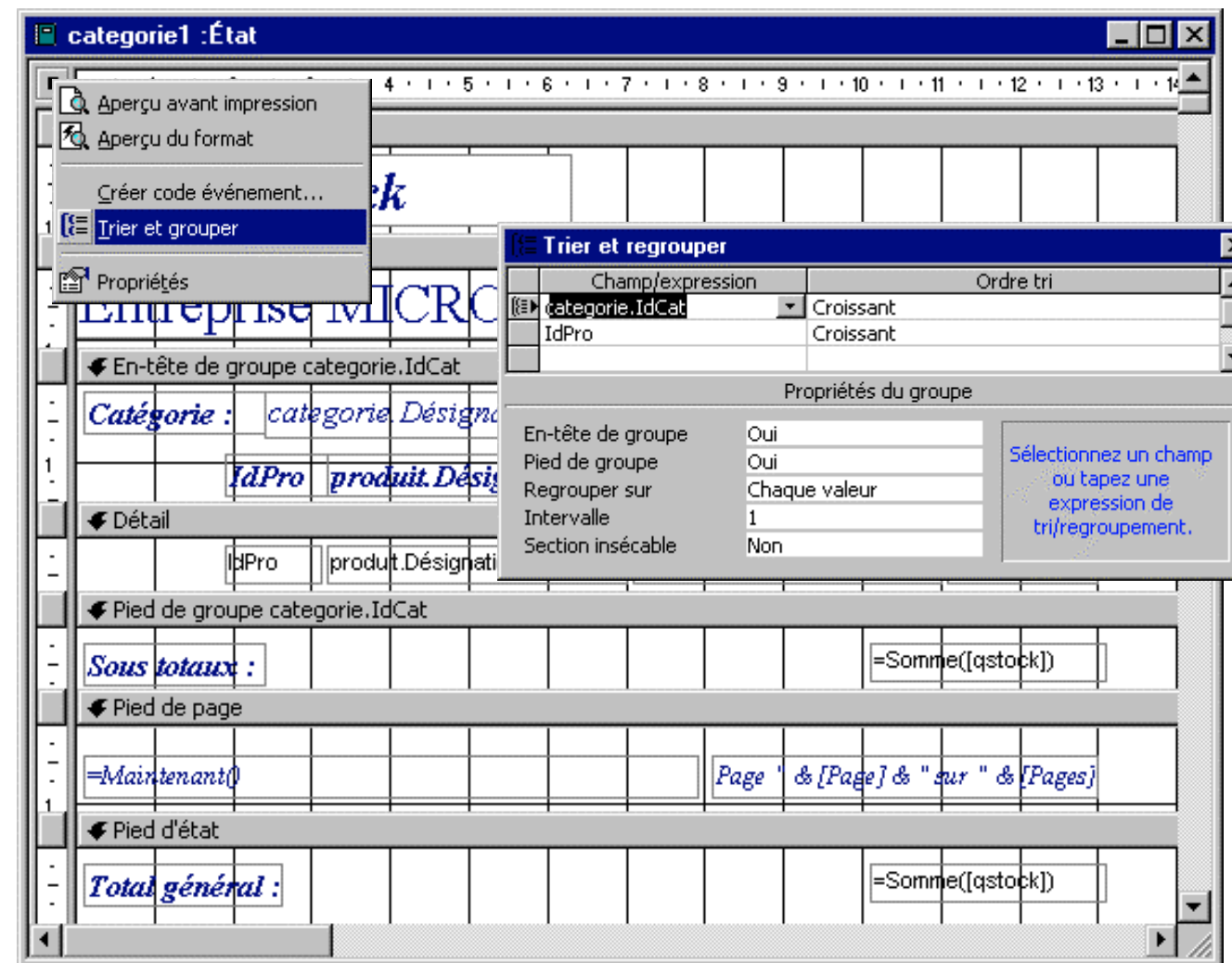
Création d'un état

- 1) Définir la propriété Source de données (table ou requête)

Cliquer ici avec le bouton droit, puis sélectionner Propriétés



2) Définir Trier et grouper



3) Placer les champs dans les différentes section de l'état

category1 :État

1 2 3 4 5 6 7 8 9 10 11 12 13 14

En-tête d'état

Etat du Stock

En-tête de page

Entreprise MICRO

En-tête de groupe categorie.IdCat

Catégorie : *categorie* Désignation

IdPro *produit* Désignation *Marque* *Qstock*

Détail

IdPro produit.Désignation Marque Qstock

Pied de groupe categorie.IdCat

Sous totaux : =Somme([qstock])

Pied de page

=Maintenant() Page ' & [Page] & " sur " & [Pages]

Pied d'état

Total général : =Somme([qstock])

Chapitre 4 L'algèbre relationnelle

I. Les opérations

II. Le langage algébrique

I Les opérations

L'Algèbre relationnelle est une collection d'opérations

▣ OPÉRATIONS

- opérandes : 1 ou 2 relations
- résultat : une relation

▣ DEUX TYPES D'OPÉRATIONS

① OPÉRATIONS ENSEMBLISTES

UNION
INTERSECTION
DIFFÉRENCE

① OPÉRATIONS SPECIFIQUES

PROJECTION
RESTRICTION
JOINTURE
DIVISION

UNION

L'union de deux relations R1 et R2 de même schéma est une relation R3 de schéma identique qui a pour n-uplets les n-uplets de R1 et/ou R2

On notera :

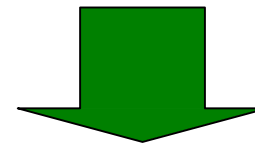
$$R3 = R1 \cup R2$$

R1	
A	B
0	1
2	3

 \cup

R2	
A	B
0	1
4	5

$$R3 = R1 \cup R2$$



R3	
A	B
0	1
2	3
4	5

▣ INTERSECTION

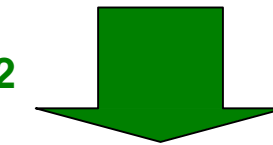
L'intersection entre deux relations R1 et R2 de même schéma est une relation R3 de schéma identique ayant pour n-uplets les n-uplets communs à R1 et R2

On notera :

$$R3 = R1 \cap R2$$

R1			R2	
A	B	\cap	A	B
0	1		0	1
2	3		4	5

$$R3 = R1 \cap R2$$



R3	
A	B
0	1

□ DIFFÉRENCE

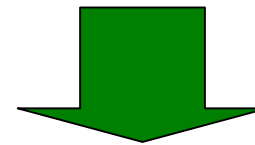
La différence entre deux relations R1 et R2 de même schéma est une relation R3 de schéma identique ayant pour n-uplets les n-uplets de R1 n'appartenant pas à R2

On notera :

$$R3 = R1 - R2$$

R1			R2	
A	B	-	A	B
0	1		0	1
2	3		4	5

$$R3 = R1 - R2$$



R3	
A	B
2	3

□ PROJECTION

La projection d'une relation R1 est la relation R2 obtenue en supprimant les attributs de R1 non mentionnés puis en éliminant éventuellement les n-uplets identiques

On notera :

$$R2 = \pi R1 (A_i, A_j, \dots, A_m)$$

la projection d'une relation R1 sur les attributs A_i, A_j, \dots, A_m

La projection permet d'éliminer des attributs d'une relation



Elle correspond à un découpage vertical :

A1	A2	A3	A4

Requête 1 :

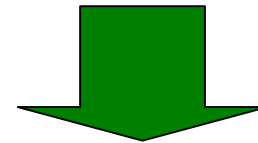
« Quels sont les références et les prix des produits ? »

PRODUIT (IdPro, Nom, Marque, Prix)



IdPro	Nom	Marque	Prix
P	PS1	IBM	1000
Q	Mac	Apple	2000
R	PS2	IBM	3000
S	Word	Microsoft	4000

π PRODUIT (IdPro, Prix)



IdPro	Prix
P	1000
Q	2000
R	3000
S	4000

Requête 2 :

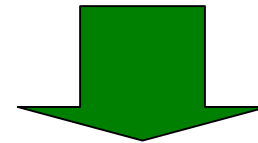
« Quelles sont les marques des produits ? »

PRODUIT (IdPro, Nom, Marque, Prix)



IdPro	Nom	Marque	Prix
P	PS1	IBM	1000
Q	Mac	Apple	2000
R	PS2	IBM	3000
S	Word	Microsoft	4000

π **PRODUIT (Marque)**



Marque
IBM
Apple
Microsoft

Notez l'élimination des doublons..

▣ RESTRICTION

La restriction d'une relation R1 est une relation R2 de même schéma n'ayant que les n-uplets de R1 répondant à la condition énoncée

On notera :

$$R2 = \sigma R1 (\text{condition})$$

la restriction d'une relation R1 suivant le critère "condition"

où "condition" est une relation d'égalité ou d'inégalité entre 2 attributs ou entre un attribut et une valeur

La restriction permet d'extraire les n-uplets qui satisfont une condition

Elle correspond à un découpage horizontal :

	A1	A2	A3	A4
→				
→				

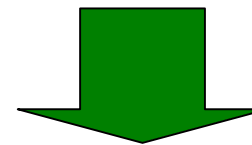
Requête 3 :

« Quelles sont les produits de marque 'IBM' ? »

PRODUIT (IdPro, Nom, Marque, Prix)

IdPro	Nom	Marque	Prix
P	PS1	IBM	1000
Q	Mac	Apple	2000
R	PS2	IBM	3000
S	Word	Microsoft	4000

σ PRODUIT (Marque = 'IBM')



IdPro	Nom	Marque	Prix
P	PS1	IBM	1000
R	PS2	IBM	3000

□ JOINTURE

La jointure de deux relations R1 et R2 est une relation R3 dont les n-uplets sont obtenus en concaténant les n-uplets de R1 avec ceux de R2 et en ne gardant que ceux qui vérifient la condition de liaison

On notera :

$$R3 = R1 \times R2 \text{ (condition)}$$

la jointure de R1 avec R2 suivant le critère condition

- Le schéma de la relation résultat de la jointure est la concaténation des schémas des opérandes (s'il y a des attributs de même nom, il faut les renommer)
- Les n-uplets de $R1 \times R2 \text{ (condition)}$ sont tous les couples $(u1, u2)$ d'un n-uplet de R1 avec un n-uplet de R2 qui satisfont "condition"
- La jointure de deux relations R1 et R2 est le produit cartésien des deux relations suivi d'une restriction
- La condition de liaison doit être du type :

$$\langle \text{attribut1} \rangle :: \langle \text{attribut2} \rangle$$

où : attribut1 \in 1ère relation et attribut2 \in 2ème relation
:: est un opérateur de comparaison (égalité ou inégalité)

La jointure permet de composer 2 relations à l'aide d'un critère de liaison

R1(A, B, C)

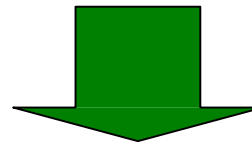
A	B	C
A1	B1	10
A2	B2	10
A3	B3	20
A4	B4	30



R2(U, V)

U	V
10	V1
20	V2
30	V3

R1 × R2 (R1.C = R2.U)



A	B	C	U	V
A1	B1	10	10	V1
A2	B2	10	10	V1
A3	B3	20	20	V2
A4	B4	30	30	V3

Jointure où l'opérateur de comparaison est l'égalité dans le résultat on fusionne les 2 colonnes dont les valeurs sont égales

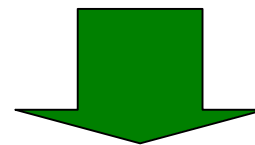
① La jointure permet d'enrichir une relation

Requête 5 :

« Donnez pour chaque vente la référence du produit, sa désignation, son prix, le numéro de client, la date et la quantité vendue »

VENTE As V				PRODUIT As P		
IdCli	IdPro	Date	Qte	IdPro	Désignation	Prix
X	P	1/1/98	1	P	PS	100
Y	Q	2/1/98	1	Q	Mac	100
Z	P	3/1/98	1			

VENTE × PRODUIT (V.IdPro=P.IdPro)



Idcli	IdPro	Date	Qte	Désignation	Prix
X	P	1/1/98	1	PS	100
Y	Q	2/1/98	1	Mac	100
Z	P	3/1/98	1	PS	100

- La normalisation conduit à décomposer ; la jointure permet de recomposer

□ DIVISION

Soit deux relations

$R1 (A1, A2, \dots, An, B1, B2, \dots, Bm)$

$R2 (B1, B2, \dots, Bm)$

Si le schéma de R2 est un sous-schéma de R1.

La division de R1 par R2 est une relation R3 dont :

- le schéma est le sous-schéma complémentaire de R2 par rapport à R1
- un n-uplet (a_1, a_2, \dots, a_n) appartient à R3 si $(a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_m)$ appartient à R1 pour tous $(b_1, b_2, \dots, b_m) \in R2$.

On notera :

$$R3 = R1 \div R2$$

la division de R1 par R2

- ① la division permet de rechercher dans une relation les sous n-uplets qui sont complétés par tous ceux d'une autre relation

Elle permet de répondre à des questions qui sont formulées avec le quantificateur universel :
"pour tout ..."

Requête 6 :

« Quels sont les élèves qui sont inscrits à tous les sports ? »

INSCRIPT

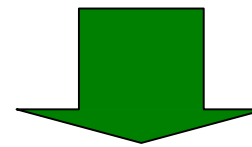
Elève	Sport
toto	judo
tata	danse
toto	foot
toto	danse

SPORT

Sport
judo
foot
danse

÷

RES = INSCRIPT ÷ SPORT



RES

Elève
toto

II Le langage algébrique

Le langage algébrique permet de formuler une question par une suite d'opérations de l'algèbre relationnelle

Requêtes sur le schéma CLIENT, PRODUIT, VENTE

CLIENT (**IdCli**, nom, ville)

PRODUIT (**IdPro**, désignation, marque, prix)

VENTE (**IdCli**, **IdPro**, **date**, qte)

Requête 8 :

« Donner les no des produits de marque Apple et de prix < 5000 DH »

$R1 = \sigma_{\text{PRODUIT}} (\text{marque} = \text{Apple})$

$R2 = \sigma_{\text{PRODUIT}} (\text{prix} < 5000)$

$R3 = R1 \cap R2$

$\text{RESUL} = \pi_{R3} (\text{IdPro})$

Requête 9 :

« Donner les no des clients ayant acheté un produit de marque Apple »

$R1 = \sigma_{\text{PRODUIT}} (\text{marque} = \text{'Apple'})$

$R2 = R1 \times \text{VENTE} (R1.\text{IdPro} = \text{VENTE}.\text{IdPro})$

$\text{RESUL} = \pi R2 (\text{IdCli})$

Requête 10 :

« Donner les no des clients n'ayant acheté que des produits de marque Apple »

$R1 = \text{VENTE} \times \text{PRODUIT} (\text{VENTE.IdPro} = \text{PRODUIT.IdPro})$

$R2 = \sigma R1 (\text{marque} = \text{'Apple'})$

$R3 = \pi R2 (\text{IdCli})$

$R4 = \sigma R1 (\text{marque} \neq \text{'Apple'})$

$R5 = \pi R4 (\text{IdCli})$

$\text{RESUL} = R3 - R5$

Requête 11 :

« Donner les no des clients ayant acheté tous les produits de marque Apple »

$$R1 = \sigma_{\text{PRODUIT}} (\text{marque} = \text{'Apple'})$$

$$R2 = \pi_{R1} (\text{IdPro})$$

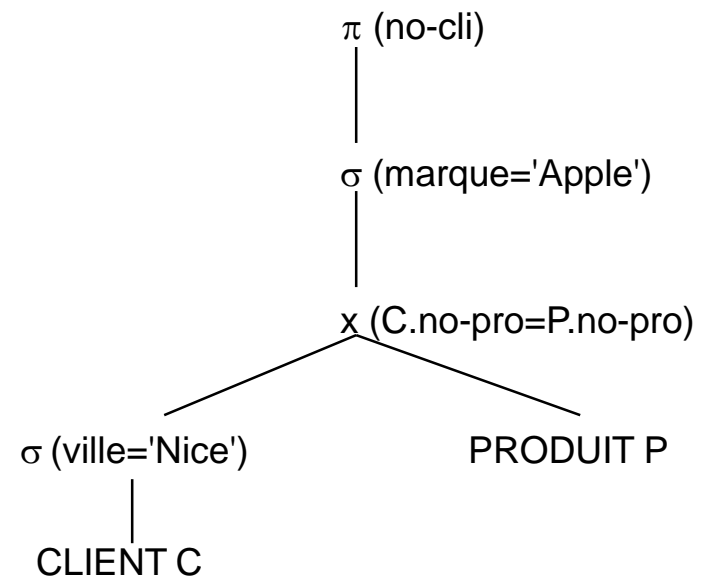
$$R3 = \pi_{\text{VENTE}} (\text{IdCli}, \text{IdPro})$$

$$R4 = R3 \div R2$$

▣ Arbre algébrique

une question peut être représentée par un arbre

« Quels sont les clients de Nice ayant acheté un produit de marque 'Apple' ? »



▣ Optimisation de requêtes

Plusieurs arbres équivalents peuvent être déduits d'un arbre donné à l'aide de règles de transformation simples, telles que permutation des jointures et restrictions, permutation des projections et des jointures, etc.

Ces transformations sont à la base des techniques **d'optimisation de requêtes**