

# LE MODELE RELATIONNEL

---

INVENTE PAR T. CODD (IBM SAN-JOSE)

PUBLICATION ACM 1970

1. CONCEPTS POUR LA DESCRIPTION
2. CONCEPTS POUR LA MANIPULATION
3. CONCEPTS ADDITIONNELS

# 1. CONCEPTS DESCRIPTIFS

---

- ENSEMBLE DE CONCEPTS POUR FORMALISER LA DESCRIPTION D'ARTICLES DE FICHIERS PLATS
  
- MODELE NORMALISE MAIS EXTENSIBLE
  - Introduction de types de données variés (SQL2)
  - Introduction de la dynamique (Produits, SQL3)
  - Introduction des objets (SQL3)

# Domaine

---

## ➤ ENSEMBLE DE VALEURS

## ➤ EXEMPLES

- ENTIER
- REEL
- CHAINES DE CARACTERES
  
- FRANC
- SALAIRE = {4 000..100 000}
- COULEUR= {BLEU, BLANC, ROUGE}
  
- POINT = {(X:REEL, Y:REEL)}
- TRIANGLE = {(P1:POINT, P2:POINT, P3:POINT)}

# Produit cartésien

- LE PRODUIT CARTESIEN  $D_1 \times D_2 \times \dots \times D_n$  EST L'ENSEMBLE DES TUPLES (N-UPLETS) :

$\langle V_1, V_2, \dots, V_n \rangle$  TEL QUE  $V_i \in D_i$

➤ EXEMPLE

- $D_1 = \{\text{Bleu}, \text{Blanc}, \text{Rouge}\}$
- $D_2 = \{\text{Vrai}, \text{Faux}\}$

Bleu	Vrai
Bleu	Faux
Blanc	Vrai
Blanc	Faux
Rouge	Vrai
Rouge	Faux

# Relation

---

- SOUS-ENSEMBLE DU PRODUIT CARTESIEN D'UNE LISTE DE DOMAINES
- UNE RELATION EST CARACTERISEE PAR UN NOM

- **EXEMPLE**

- D1 = COULEUR
- D2 = BOOLEEN

CoulVins	Coul	Choix
	Bleu	Faux
	Blanc	Vrai
	Rouge	Vrai

# Attribut

---

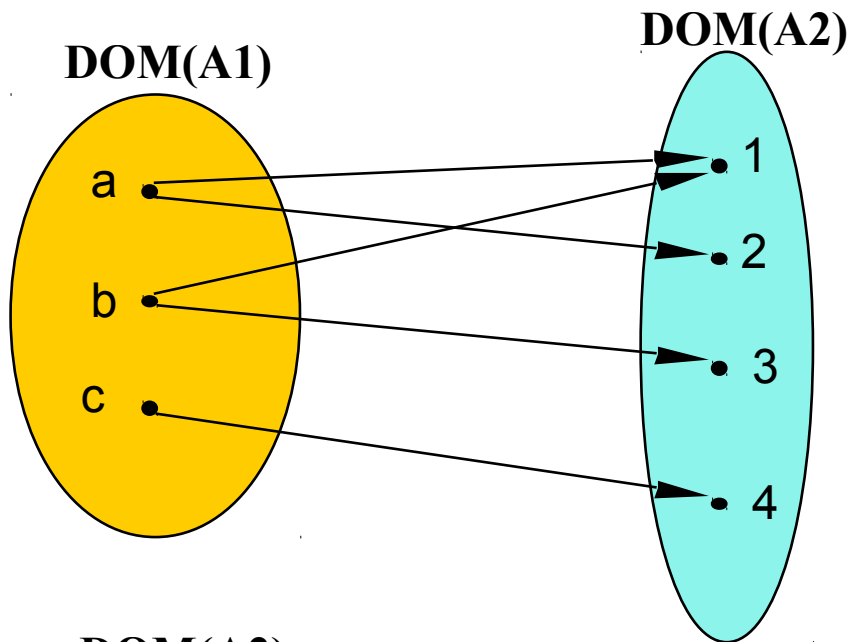
## ➤ VISION TABULAIRE DU RELATIONNEL

- Une relation est une table à deux dimensions
- Une ligne est un tuple
- Un nom est associé à chaque colonne afin de la repérer indépendamment de son numéro d'ordre

## ➤ ATTRIBUT

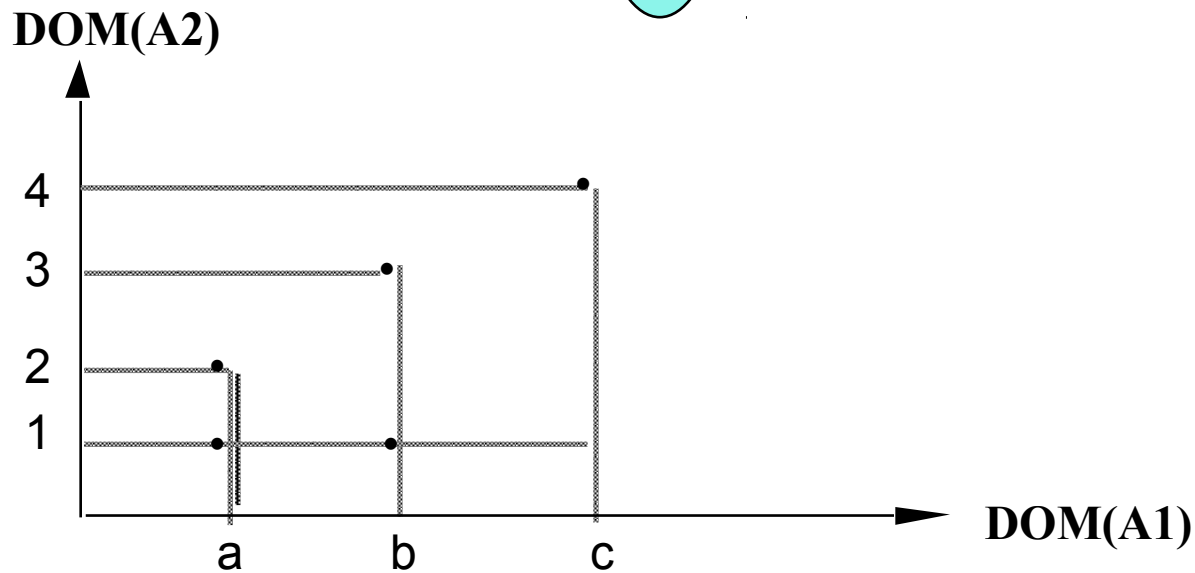
- nom donné à une colonne d'une relation
- prend ses valeurs dans un domaine

# Graphe d'une relation



➤ **RELATION BINAIRE  $R(A1, A2)$**

➤ **UNE RELATION N-AIRE EST UNE GENERALISATION A N DIMENSIONS**



# Exemple de relation

---

VINS	CRU	MILL	REGION	COULEUR
	CHENAS	1983	BEAUJOLAIS	ROUGE
	TOKAY	1980	ALSACE	BLANC
	TAVEL	1986	RHONE	ROSE
	CHABLIS	1986	BOURGOGNE	BLANC
	ST-EMILION	1987	BORDELAIS	ROUGE





# Clé

---

➤ GROUPE D'ATTRIBUTS MINIMUM QUI DETERMINE UN TUPLE UNIQUE DANS UNE RELATION

➤ EXEMPLES

- {CRU,MILLESIME} DANS VINS ==> NV
- NSS DANS PERSONNE

➤ CONTRAINTE D'ENTITE

- Toute relation doit posséder au moins une clé documentée

# Schéma

---

- **NOM DE LA RELATION, LISTE DES ATTRIBUTS AVEC DOMAINES, ET LISTE DES CLES D'UNE RELATION**
- **EXEMPLE**
  - VINS(NV: Int, CRU:texte, MILL:entier, DEGRE: Réel, REGION:texte)
  - Par convention, la clé primaire est soulignée
- **INTENTION ET EXTENSION**
  - Un schéma de relation définit l'intention de la relation
  - Une instance de table représente une extension de la relation
- **SCHEMA D'UNE BD RELATIONNELLE**
  - C'est l'ensemble des schémas des relations composantes

# Clé Etrangère

---

- GROUPE D'ATTRIBUTS DEVANT APPARAÎTRE COMME CLE DANS UNE AUTRE RELATION
- LES CLES ETRANGERES DEFINISSENT LES CONTRAINTES D'INTEGRITE REFERENTIELLES
  - Lors d'une insertion, la valeur des attributs doit exister dans la relation référencée
  - Lors d'une suppression dans la relation référencée les tuples référençant doivent disparaître
  - Elles correspondent aux liens entité-association obligatoires

# Exemple de Schéma

---

## ➤ EXEMPLE

**BUVEURS (NB, NOM, PRENOM, TYPE)**

**VINS (NV, CRU, MILL, DEGRE)**

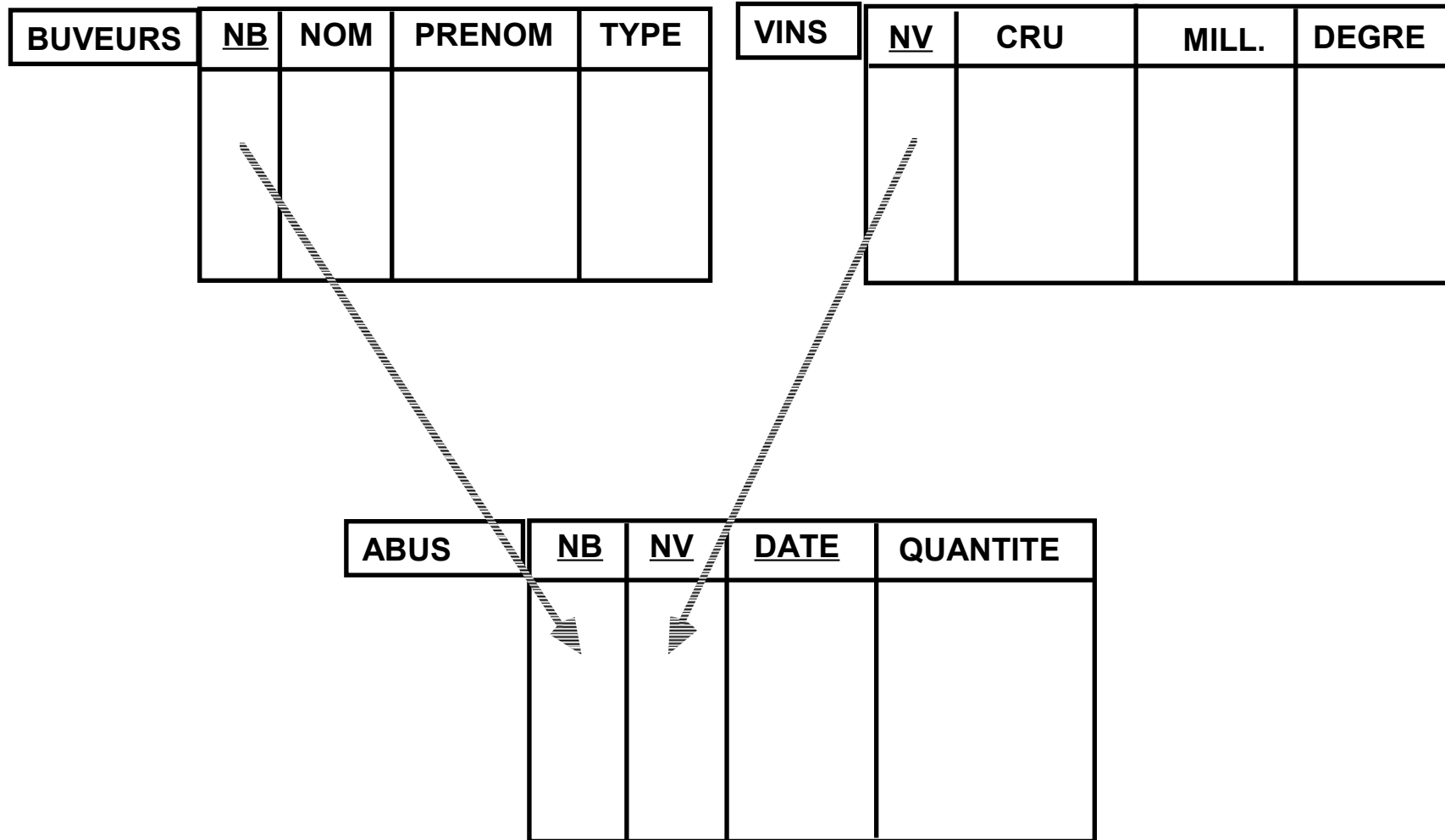
**ABUS (NB, NV, DATE, QUANTITE)**

## ➤ CLES ETRANGERES

**ABUS.NV REFERENCE VINS.NV**

**ABUS.NB REFERENCE BUVEURS.NB**

# Diagramme des Liens



# Concepts Descriptifs : Bilan

---

- **RELATION** ou **TABLE**
- **ATTRIBUT** ou **COLONNE**
- **DOMAINE** ou **TYPE**
- **CLE**
- **CLE ETRANGERE**



# Synthese : Create Table

---

## ➤ CREATION DES TABLES EN SQL

CREATE TABLE <relation name>

(<attribute definition>+)

[{PRIMARY KEY | UNIQUE} (<attribute name>+)]

## ➤ avec :

<attribute definition> ::= <attribute name> <data type>

[NOT NULL [{UNIQUE | PRIMARY KEY}] ]

## ➤ Exemple :

CREATE TABLE VINS

( NV INTEGER PRIMARY KEY

CRU CHAR VARYING

MILL INTEGER NOT NULL,

DEGRE FIXED 5.2 )

# 2. CONCEPTS MANIPULATOIRES

---

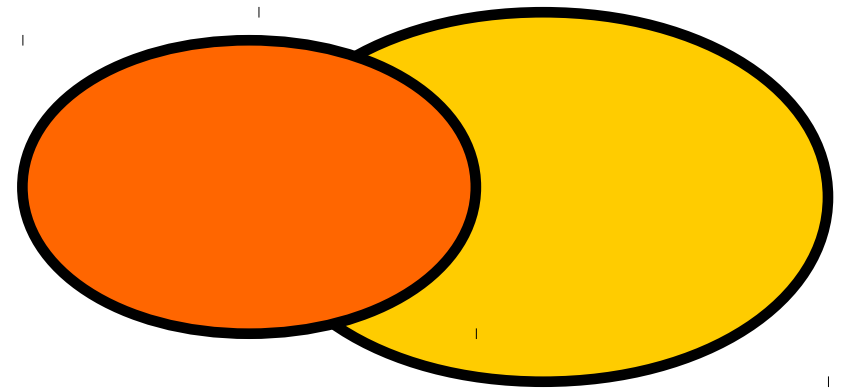
- UN ENSEMBLE D'OPERATIONS FORMELLES
  - Algèbre relationnelle
- CES OPERATIONS PERMETTENT D'EXPRIMER TOUTES LES REQUETES SOUS FORME D'EXPRESSIONS ALGEBRIQUES
- ELLES SONT LA BASE DU LANGAGE SQL
  - Paraphrasage en anglais des expressions relationnelles
  - Origine SEQUEL
- CES OPERATIONS SE GENERALISENT A L'OBJET
  - Algèbre d'objets complexes



# Opérations Ensemblistes

## ➤ OPERATION ENSEMBLISTE POUR DES RELATIONS DE MEME SCHEMA

- UNION notée  $\cup$
- INTERSECTION notée  $\cap$
- DIFFERENCE notée  $-$



## ➤ OPERATIONS BINAIRES

- Relation X Relation  $\rightarrow$  Relation
- 

## ➤ EXTENSION

- Union externe pour des relations de schémas différents
- Ramener au même schéma avec des valeurs nulles

# Projection

- Elimination des attributs non désirés et suppression des tuples en double
- Relation  $\rightarrow$  Relation notée  $\pi_{A1,A2,\dots,Ap} (R)$

VINS	Cru	Mill	Région	Qualité
VOLNAY	1983	BOURGOGNE	A	
VOLNAY	1979	BOURGOGNE	B	
CHENAS	1983	BEAUJOLAIS	A	
JULIENAS	1986	BEAUJOLAIS	C	

$\pi_{\text{Cru,Région}}$

$\pi(\text{VINS})$	Cru	Région
	VOLNAY	BOURGOGNE
	CHENAS	BEAUJOLAIS
	JULIENAS	BEAUJOLAIS

# Restriction

---

- Obtention des tuples de R satisfaisant un critère Q
- Relation  $\rightarrow$  Relation, notée  $\sigma_Q(R)$
- Q est le critère de qualification de la forme :
  - $A_i \theta \text{ Valeur}$
$$\theta = \{ =, <, >=, >, <=, \# \}$$
- Il est possible de réaliser des "ou" (union) et des "et" (intersection) de critères simples

# Exemple de Restriction

VINS	Cru	Mill	Région	Qualité
	VOLNAY	1983	BOURGOGNE	A
	VOLNAY	1979	BOURGOGNE	B
	CHENAS	1983	BEAUJOLAIS	A
	JULIENAS	1986	BEAUJOLAIS	C

$\sigma_{\text{MILL}} > 1983$

VINS	Cru	Mill	Région	Qualité
	JULIENAS	1986	BEAUJOLAIS	C

$\sigma_{\text{CRU}} = \text{"VOLNAY"}$

"

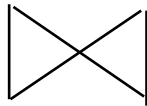
# Jointure

---

- Composition des deux relations sur un domaine commun
- Relation X Relation  $\rightarrow$  Relation
  - notée  $\bowtie$
- CRITERE DE JOINTURE
  - Attributs de même nom égaux :
    - Attribut = Attribut
    - JOINTURE NATURELLE
  - Comparaison d'attributs :
    - Attribut1  $\theta$  Attribut2
    - THETA-JOINTURE

# Exemple de Jointure

VINS	Cru	Mill	Qualité
	VOLNAY	1983	A
	VOLNAY	1979	B
	CHABLIS	1983	A
	JULIENAS	1986	C



LOCALISATION	Cru	Région	QualMoy
	VOLNAY	Bourgogne	A
	CHABLIS	Bourgogne	A
	CHABLIS	Californie	B



VINSREG	Cru	Mill	Qualité	Région	QualMoy
	VOLNAY	1983	A	Bourgogne	A
	VOLNAY	1979	B	Bourgogne	A
	CHABLIS	1983	A	Bourgogne	A
	CHABLIS	1983	A	Californie	B

# Complétude

---

## ➤ L'ALGEBRE RELATIONNELLE EST COMPLETE

- Les cinq (sept) opérations de base permettent de formaliser sous forme d'expressions toutes les questions que l'on peut poser avec la logique du premier ordre (sans fonction).

## ➤ EXEMPLE

- NOM ET PRENOM DES BUVEURS DE VOLNAY 1988  
?

```
PROJECT (NOM, PRENOM,  
        RESTRICT(CRU="VOLNAY" et MILL =1988,  
        JOIN(VINS, ABUS, BUVEURS)))
```

# SQL

---

➤ Une requête SQL est un paraphrasage d'une expression de l'algèbre relationnelle en anglais

➤ **FORMAT SIMPLIFIE :**

SELECT A1, A2, ...Ap

FROM R1, R2, ...Rk

WHERE Q [ {UNION | INTERSECT | EXCEPT } ... ]

➤ **SEMANTIQUE DU BLOC SELECT :**

PROJECT A1,A2,...Ap (

RESTRICT Q (

PRODUIT ( R1, R2, ..., Rk) ) )



# 3. CONCEPTS ADDITIONNELS

---

## ➤ ENSEMBLE DE CONCEPTS POUR :

- Etendre les fonctionnalités de manipulation
- Décrire les règles d'évolution des données
- Supporter des objets complexes (SQL3)

## ➤ INTRODUIITS PROGRESSIVEMENT DANS LE MODELE

- Complique le modèle
- Manque parfois de standard (SQL3)
- Des extensions à l'infini ...

# Fonction et Agrégat

---

## ➤ FONCTION

- Fonction de calcul en ligne appliquée sur un ou plusieurs attributs
- Exemple :  $\text{DEGRE} * \text{QUANTITE} / 100$

## ➤ AGREGAT

- Partitionnement horizontal d'une relation selon les valeurs d'un groupe d'attributs, suivi d'un regroupement par une fonction de calcul en colonne (Sum, Min, Max, Avg, Count, ...)

# Exemples d'agrégats

VINS	CRU	MILL	DEGRE	QUANTITE
	CHABLIS	1977	10.9	100
	CHABLIS	1987	11.9	250
	VOLNAY	1977	10.8	400
	VOLNAY	1986	11.2	300
	MEDOC	1985	11.2	200

**SELECT MOY(DEGRE)  
FROM VINS;**

AVG	DEGRE
	11.2

**SELECT CRU, SUM(QUANTITE)  
FROM VINS  
GROUP BY CRU;**

SUM	CRU	QUANTITE
	CHABLIS	350
	VOLNAY	700
	MEDOC	200

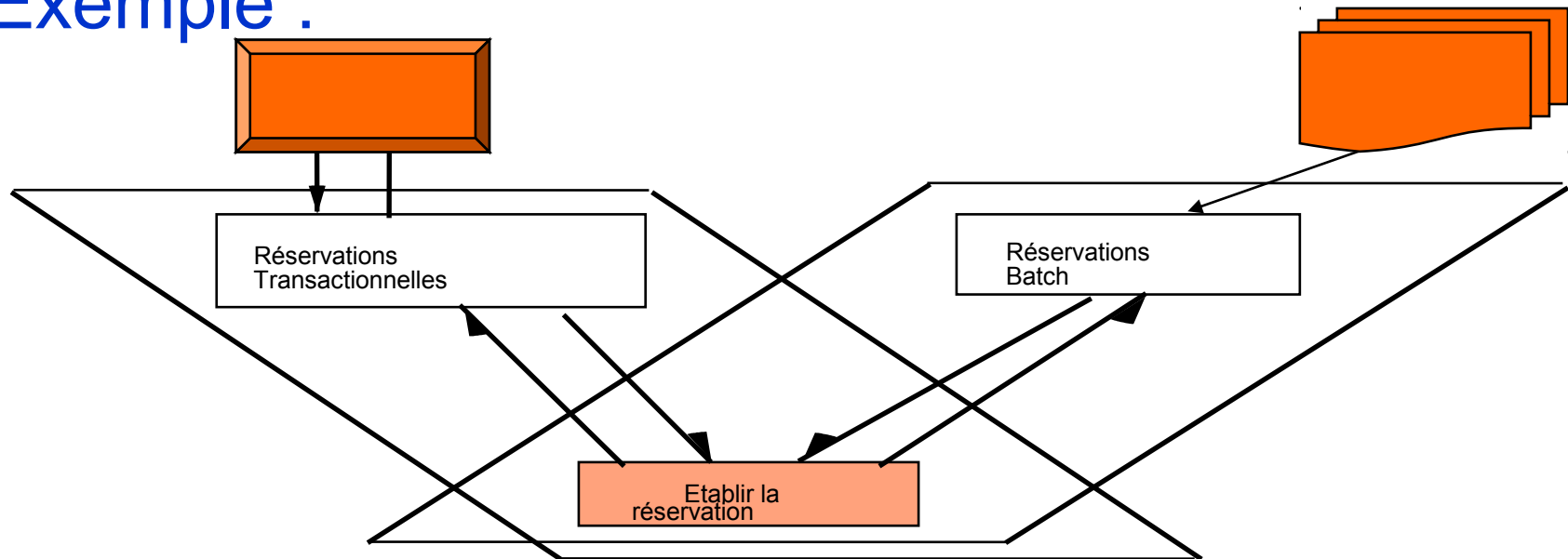
# Vue

---

- Relation d'un schéma externe déduite des relations de la base par une question
- Exemple : GrosBuveurs
  - CREATE VIEW GrosBuveurs AS
  - SELECT NB, Nom, Prénom,
  - FROM Buveurs, Abus
  - WHERE Buveurs.NB = Abus.NB and Abus.Quantité > 100
- Calcul de la vue
  - Une vue est une fenêtre dynamique sur la BD et est recalculée à chaque accès.
  - Une vue peut être matérialisée (vue concrète).

# Procédure stockée

- Procédure écrite en L3G/SQL ou L4G/SQL définie au niveau du schéma de la base et stockée avec
- Avantages :
  - partitionner les traitements entre client et serveur
  - limiter le trafic sur le réseau, partager des procédures
- Exemple :



# Déclencheur (Trigger)

---

- Action base de données déclenchée suite à l'apparition d'un événement particulier
- Forme :
  - {BEFORE | AFTER} <événement> THEN <action>
  - Un événement peut être :
    - une opération sur une table (début ou fin)
    - un événement externe (heure, appel, etc.)
  - Une action peut être :
    - une requête BD (mise à jour)
    - Une annulation (abort) de transaction
    - l'appel à une procédure cataloguée

# Déclencheur avec condition (Règle)

---

- Il est possible d'ajouter une condition afin de déclencher l'action seulement quand la condition est vérifiée
  - Une condition est une qualification portant sur la base.

## ➤ Exemples :

```
BEFORE      DELETE FROM Vins
THEN        DELETE FROM Abus
WHERE       Vins.Nv = Abus.Nv
```

```
BEFORE      UPDATE FROM EMPLOYE
IF          SALAIRE > 100.000
THEN        ABORT TRANSACTION
```

# 4. CONCLUSION

---

- Un ensemble de concepts bien compris et bien formalisés
- Un modèle unique, de plus en plus riche et standardisé
  - intégration des BD actives
  - intégration des BD objets
  - évolution vers un L4G standardisé
- Un formalisme qui s'étend plutôt bien
  - algèbre d'objets
- Un langage associé défini à plusieurs niveaux
  - SQL1, 2, 3