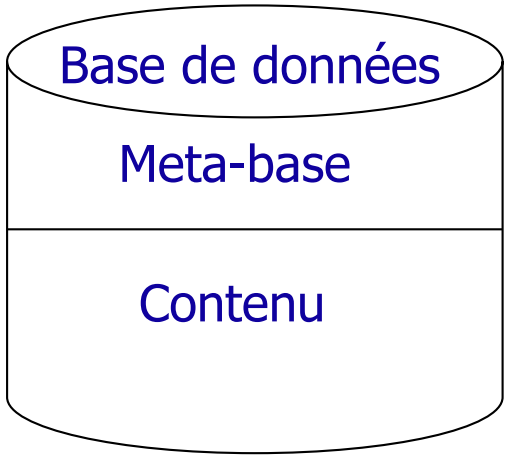
Catalogues, et Vues Virtuelles

Ladjel BELLATRECHE

bellatreche@ensma.fr
05 49 49 80 77

SQL catalogues

- Catalogues SQL sont des **tables** gérées par le SGBD, dans la metabase
  - **SYSTABLES** : une ligne pour chaque table
  - **SYSCOLUMNS :** une ligne pour chaque colonne
  - **SYSINDEXES :** une ligne pour chaque index



SYSTABLES: exemple

NAME	CREATOR	COLCOUNT	REMARKS	...
S	Lee	4	fournisseur	...
SP	Jamal	5	fourniture	...
P	David	5	piece	...

SYSCOLUMNS: exemple

NAME	TBNAME	CREATOR	COLTYPE	REMARKS	…
S#	S	Lee	CHAR	**oid**	…
SNAME	S	Lee	CHAR	**nom**	…
STATUS	S	Lee	SMALLINT		…
…	…		…	…	…
P#	SP	Lee	CHAR	cle etrangere	…
…	…		…	…	…

SYSINDEXES: exemple

NAME	TBNAME	CREATOR	...
XS	S	Lee	...
XSC	S	Lee	...
XSP	SP	Lee	...

- **Pas de champ REMARKS**

Requêtes aux catalogues

```
SELECT TBNAME
FROM SYSCOLUMNS
WHERE NAME = 'S#'

SELECT COUNT(*)
FROM SYSTABLES
WHERE CREATOR = 'Jamal'
```

- **Modification**
  - **Par le SGBD seulement.**
    - pourquoi ?

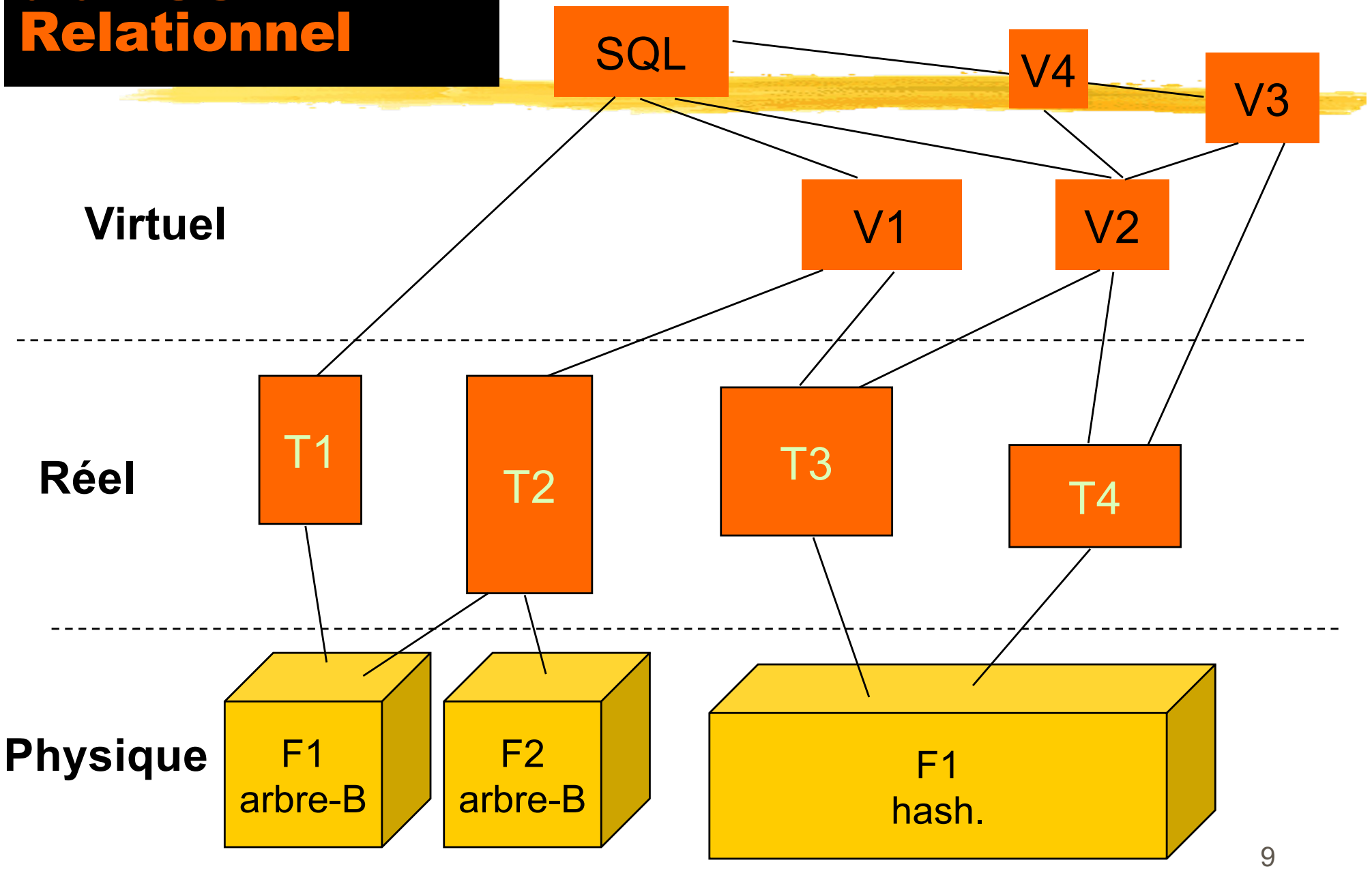
Gestion des Vues

- Objectifs
- Vues externes
- Interrogation des vues
- Mises à jour des vues
- Vues matérialisées
- Sécurité et autorisation
- Conclusion

Vues externes

- Objectif:
  - Indépendance logique des applications par rapport à la base
- Moyen
  - Les vues sont des relations virtuelles dont la définition est stockée dans la méta-base
- Problème
  - Interrogation efficace
  - Mise à jour au travers de vues

Architecture d'un SGBD Relationnel



Avantages des vues

- Vues personnalisées de la base de données à chaque utilisateur
  - Accès personnalisé aux données
- Fournir des possibilités d'abréviation :
  - Requêtes plus compactes
- Sécurité :
  - Vues sont des éléments de protection du SQL.
- Transparente pour l'utilisateur
  - Comme des tables de la base

Définition d'une vue virtuelle

- Une vue $V(a_1,a_2,\ldots,a_n)$ est une relation avec n attributs contenant le résultat d'une requête $Q(a_1,a_2,\ldots,a_n)$ évalue sur une base de données
- Remarques:
  - V possède un schéma relationnel avec les attributs $a_1, a_2,\ldots, a_n$
  - V peut être interrogée et il est possible de définir des vues à partir d'autres vues

Vues dans SQL

- Syntaxe:

  CREATE VIEW nom_vue [(col1, col2,…)]
  AS requête_SQL [WITH CHECK OPTION]

- Nom_vue : nom de la vue
- Col1, … (optionnel): permet de nommer les attributs de a vue (attributs de la requête par défaut)
- Requête_SQL : désigne une requête SQL définissant le contenu de la vue (définition de la vue)

Exemple: vue de sélection

Emp(Eno, Ename,Title,City) Project(Pno, Pname, Budget, City)
Pay(Title, Salary) Works (Eno,Pno, Resp, Dur)

Définition de la vue:
Employés parisiens

```
CREATE VIEW EmpParisien
AS
SELECT *
FROM Emp
WHERE City='Paris'
```

Exemple de vue de jointure

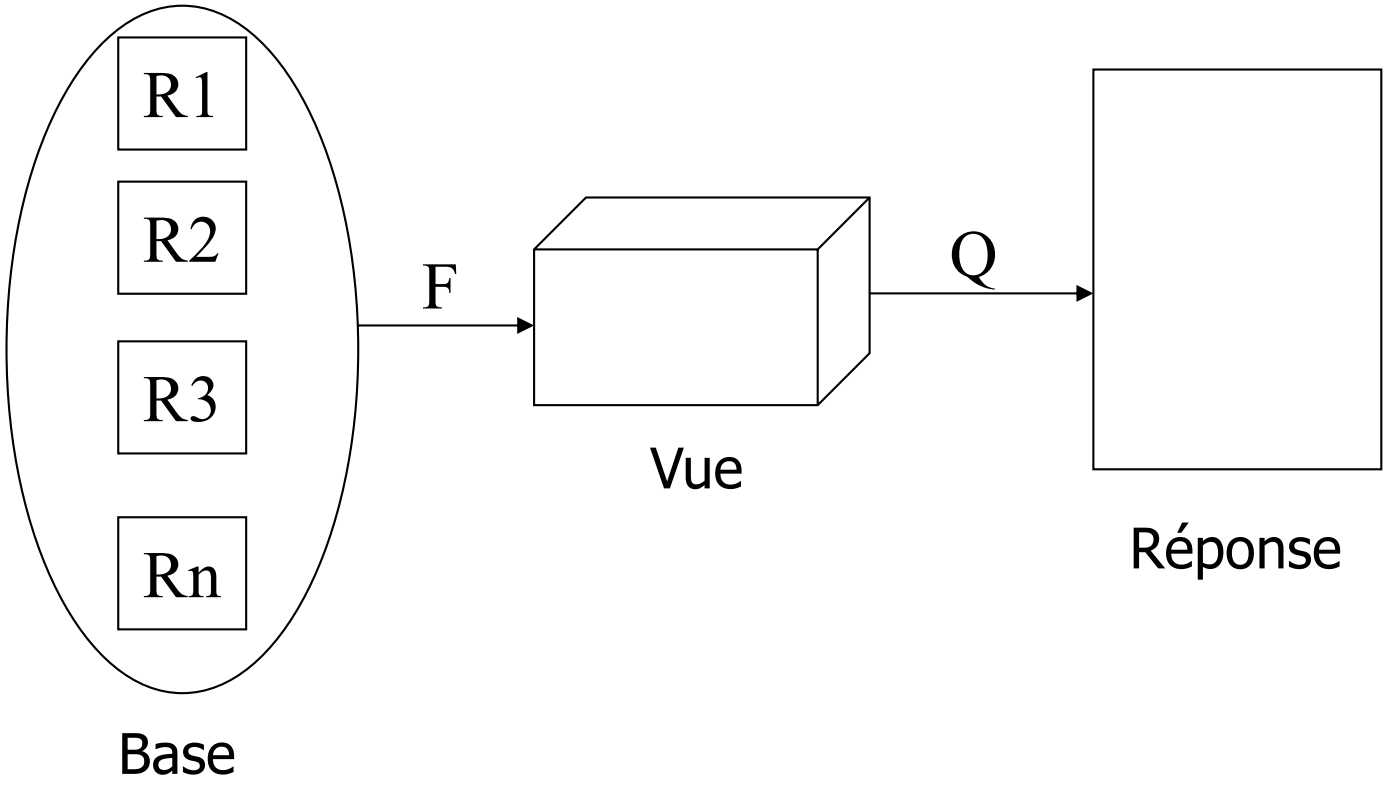
Définition de la vue:

Employés travaillant sur des projets parisiens

```
CREATE VIEW EmpProjParisien (NumE, NomE, NumP, NomP, Dur)
AS
SELECT Emp.Eno, Ename, Works.Pno, Pname, Dur
FROM Emp, Works, Project
WHERE Project.City='Paris'
AND Emp.Eno = Works.Eno
AND Works.Pno = Project.Pno
```

Interrogation de vues

Schéma fonctionnel



Modification de requêtes: Réécriture de requête

- Mécanisme consistant à modifier une requête initiale définie sur les tables de base:
  - En remplaçant certaines tables du FROM par des vues et
  - En enrichissant les conditions de la clause WHERE pour obtenir le résultat de la question initiale.
- Processus obtenu par la concaténation d'arbres
- Concaténation d'arbre:
  - Mécanisme consistant à remplacer un nœud pendant dans un arbre relationnel par un arbre calculant le nœud remplacé
- Trois matchings sont possibles
  - Full matching
  - Partial matching
  - No matching

Full matching

Accès seulement aux vues

Requête: Lister les noms des employés de projets parisiens:

SANS VUE

```
SELECT Ename
FROM Emp, Works, Project
WHERE Project.City='Paris'
AND Emp.Eno = Works.Eno
AND Works.Pno = Project.Pno
```

Réécriture

AVEC VUE

```
SELECT Ename
FROM EmpProjParisien
```

Partial matching

Accès aux vues et tables de base

Définition de la vue:

Employés travaillant sur des projets parisiens

```
CREATE VIEW EmpProjParisien (NumE, NomE, NumP, NomP, Dur)
AS
SELECT Emp.Eno, Ename, Works.Pno, Pname, Dur
FROM Emp, Works, Project
WHERE Project.City='Paris'
AND Emp.Eno = Works.Eno
AND Works.Pno = Project.Pno
```

Requête:

Employés travaillant sur des projets parisiens dont la durée < 2 ans

Question:

Tracer l'arbre algébrique correspondant à cette requête

Mise à jour de vues

- Problème: une vue est une relation virtuelle et toutes les modifications de cette relation doivent être répercutées sur les tables de base
- **Rarement possibles en SQL**
- **Impossibles quand la vue :**
  - **est une projection sur attributs autres que la clé**
  - **contient une jointure**
  - **contient une fonction agrégat**

CHECK OPTION

- La clause Check Option signifie que les Inserts et les updates au niveau des vues vont être rejetés s'ils violent la condition de définition de la vue
- WITH CHECK OPTION est optional en SQL mais très fortement recommandée pour éviter les surprises

Exemple

```
CREATE VIEW ProjetParis
WITH LOCAL CHECK OPTION
AS SELECT Pno, Pname, Budget, City
FROM Project
WHERE CITY ='Paris'

UPDATE ProjetParis
SET City ='Niort'                    Mise à jour rejetée
WHERE Pno = 123;
```

Vues vs. Tables

- Similarités:
  - Interrogation SQL
  - UPDATE, INSERT et DELETE sur vues modifiables
  - Autorisation d'accès
- Différences:
  - On ne peut indexer les vues
  - On peut pas de définir des contraintes sur des vues (clés)
  - Une vue est recalculée à chaque fois qu'on l'interroge

GRANT

- Le système d'autorisation de SQL

GRANT SELECT ON TABLE S TO Jacky ;

GRANT SELECT, UPDATE (STATUS, CITY) ON TABLE S TO Witold, Edwin ;

- **Pourrait être aussi DELETE, INSERT**

GRANT ALL ON TABLE S, P TO Me, You, Him ;

GRANT SELECT ON TABLE S TO PUBLIC ;

GRANT INDEX ON TABLE S TO Me ;

GRANT SELECT ON TABLE S TO You WITH GRANT OPTION ;

REVOKE

- Révoque l'autorisation de GRANT

REVOKE SELECT ON TABLE S FROM you ;

- La révocation se propage à travers les autorisations de **GRANT OPTION**

- **Peut être impossible en pratique dans le SGBD répartie**