



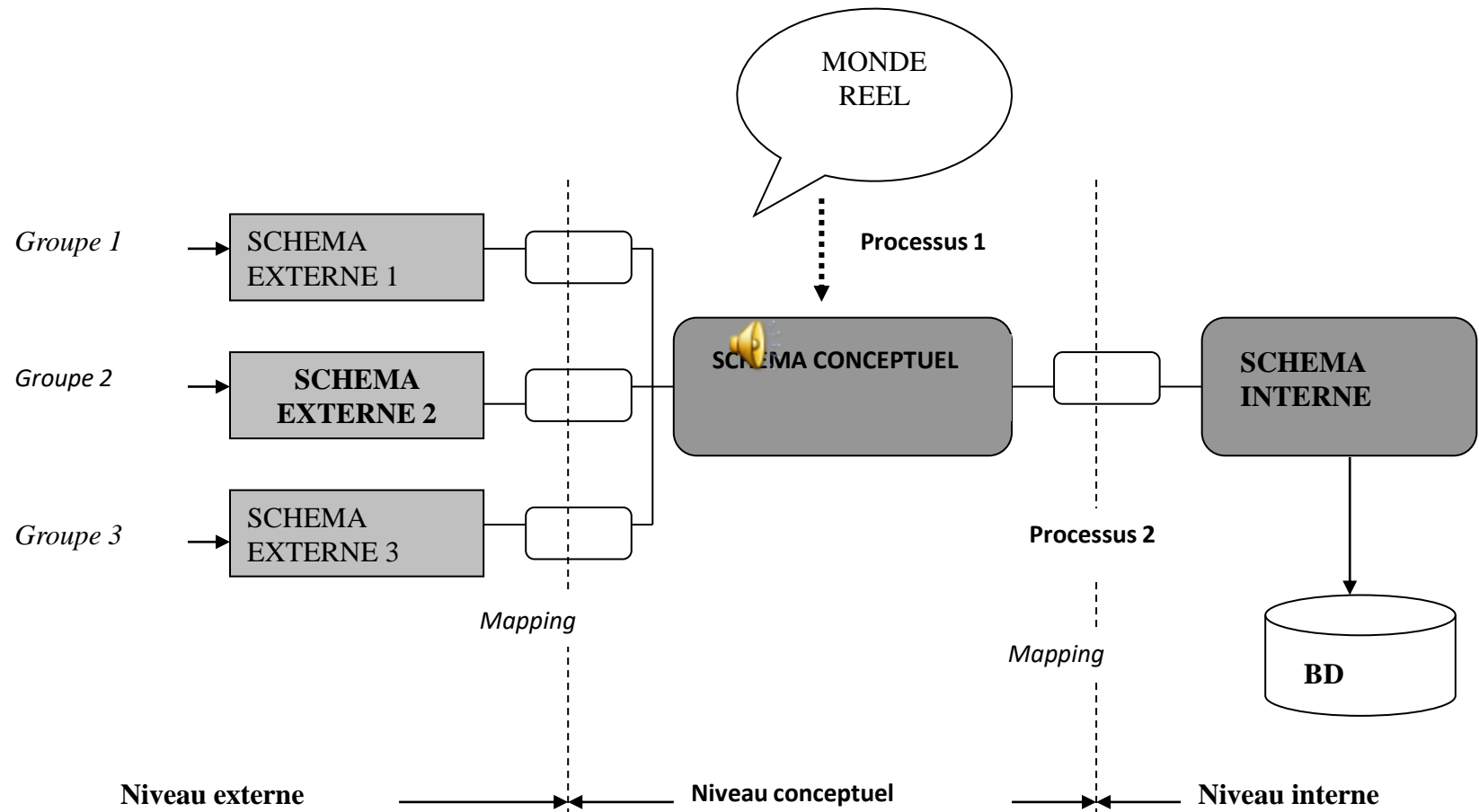
Architecture et fonctions d'un SGBD

Partie 3 : Vues externes &
Contraintes d'intégrité
Suite & Fin



Schémas Externes (Vues)

Niveaux de représentation des données



2.2. Fonctions des SGBD relationnels

2.2. 4. Vues Externes

La notion de vue va au-delà du simple sous-schéma comme sous-ensemble de schéma de la base.



Une vue est une relation virtuelle une fenêtre dynamique sur la base de données, c'est à dire qu'il n'existe pas de fichier qui représente directement une vue, par contre sa définition est stockée dans un catalogue appelé catalogue **Vue**.

2.2. Fonctions des SGBD relationnels

2.2. 4. Vues Externes


La définition de la vue montre comment la vue est dérivée à partir des relations indiquées (relations sources). La commande permettant de définir une vue est :



```
CREATE VIEW <nom de vue> [(liste d'attributs)]  
AS <bloc de qualification> ;
```

2.2. Fonctions des SGBD relationnels

2.2. 4. Vues Externes

La définition de la vue montre comment la vue est dérivée à partir des relations indiquées (relations sources). La commande permettant de définir une vue est : 

CREATE VIEW <nom de vue> [(liste d'attributs)]
AS <bloc de qualification> ;

Le **bloc de qualification** commence par un SELECT et peut comporter toutes les possibilités que nous avons décrit dans une requête d'interrogation de SQL. Cependant, il faut souligner qu'aucun n-uplet n'est extrait de la base par une telle commande.

2.2. Fonctions des SGBD relationnels

2.2. 4. Vues Externes

La définition de la vue montre comment la vue est dérivée à partir des relations indiquées (relations sources). La commande permettant de définir une vue est :

```
CREATE VIEW <nom de vue> [(liste d'attributs)]  
AS <bloc de qualification> ;
```

Le **bloc de qualification** commence par un SELECT et peut comporter toutes les possibilités que nous avons décrit dans une requête d'interrogation de SQL. **Cependant, il faut souligner qu'aucun n-uplet n'est extrait de la base par une telle commande.**

2.2. Fonctions des SGBD relationnels

2.2. 4. Vues Externes

La commande permettant de définir une vue est :

```
CREATE VIEW <nom de vue> [(liste d'attributs)]  
AS <bloc de qualification> ;
```



Exemple :

```
CREATE VIEW ALGER-PIECE AS SELECT NP, NOM, POIDS  
FROM PIECE  
WHERE VILLE = 'ALGER';
```

La nouvelle relation ALGER-PIECE a trois attributs **hérités** de la relation PIECE. De nouveaux noms **auraient pu** être donnés aux attributs de cette nouvelle "relation".

2.2. Fonctions des SGBD relationnels

2.2. 4. Vues Externes

La commande permettant de définir une vue est: **CREATE VIEW** <nom de vue> [(liste d'attributs)]

AS <bloc de qualification> ;

Les noms des attributs **doivent** être spécifiés dans le cas d'une ambiguïté ou si le résultat du SELECT est une fonction d'agrégat.




Exemple :

```
CREATE VIEW PQ (NP, SOMQTE) AS SELECT NP, SUM( QTE )  
FROM FOURNITURE  
GROUP BY NP.
```

2.2. Fonctions des SGBD relationnels

2.2. 4. Vues Externes

Une vue constitue **une fenêtre dynamique** sur la base en ce sens que toute modification sur une relation source est automatiquement reportée au travers de la vue. Ainsi lorsqu'une pièce n'est plus stockée à Alger, elle n'apparaît plus dans la vue ALGER-PIECE.

Cependant mise à part  le fait **qu'il est impossible de définir des chemins d'accès dessus**, une vue se comporte comme tout autre relation de la base. En particulier on peut l'interroger et elle peut servir à la construction d'autres vues.

2.2. Fonctions des SGBD relationnels

2.2. 4. Vues Externes

La requête suivante permet de connaître les numéros des pièces stockées à Alger dont le poids est entre 20 et 40 :

```
SELECT NP  
FROM ALGER-PIECE  
WHERE POIDS BETWEEN 20 AND 40;
```



2.2. Fonctions des SGBD relationnels

2.2. 4. Vues Externes

```
SELECT NP  
FROM ALGER-PIECE  
WHERE POIDS BETWEEN 20 AND 40;
```

Pour traiter une telle requête, les SGBD relationnels effectuent ce qu'il est convenu d'appeler l'opération de composition de vue.

Il s'agit avant tout accès **à la base de remplacer une vue par l'expression qui la définit** jusqu'à former une requête qui ne porte que sur des relations de base. La requête précédente devient :

```
SELECT NP  
FROM PIECE  
WHERE POIDS BETWEEN 20 AND 40 AND VILLE = 'ALGER';
```



2.2. Fonctions des SGBD relationnels

2.2. 4. Vues Externes

Il y a une contre partie au fait que les vues soient des fenêtres dynamiques sur une base de données, autorisant ainsi différents usagers à avoir leur propre vision de tout ou partie de cette base : il n'est pas toujours possible d'effectuer des mises à jour au travers des vues.



2.2. Fonctions des SGBD relationnels

2.2. 4. Vues Externes


Il y a une contre partie au fait que les vues soient des fenêtres dynamiques sur une base de données, autorisant ainsi différents usagers à avoir leur propre vision de tout ou partie de cette base : il n'est pas toujours possible d'effectuer des mises à jour au travers des vues.

En effet, pour faire **une mise à jour** au travers d'une vue, il doit être possible de **propager cette mise à jour sur les relations sources**. Si la vue est telle qu'il existe une correspondance biunivoque entre les n-uplets de la vue et ceux **d'une et une seule** relation source , **les mises à jours et les suppressions ne posent aucun problème et peuvent être propagées.**



2.2. Fonctions des SGBD relationnels

2.2. 4. Vues Externes


En ce qui concerne les **insertions**, si la vue est une **projection** d'une relation, il faut alors déterminer quelle sera la valeur des constituants n'apparaissant pas dans la vue. En particulier dans l'exemple **ALGER-PIECE**, l'insertion d'une nouvelle pièce au travers de la vue laisse indéfinie les valeurs **du matériau**. 

Au-delà de ces simples règles, la propagation des mises à jour au travers d'une vue quelconque est parfois dangereuse, voir impossible.

2.2. Fonctions des SGBD relationnels

2.2. 4. Vues Externes

Nous allons montrer sur un exemple que la mise à jour au travers d'une vue peut conduire à des incohérences. Considérons les deux relations :

EMP (NUM, NOM, NDEP*) 
DEPT (NDEP, DIR)

Qui ont pour signification respective qu'un employé a un numéro NUM, un nom NOM, qu'il travaille dans département de numéro NDEP et qu'un département a un numéro de département NDEP et un directeur DIR.

2.2. Fonctions des SGBD relationnels

2.2. 4. Vues Externes

EMP (NUM, NOM, NDEP*)

DEPT (NDEP, DIR)

L'extension de ces deux relations est :

EMP :

<u>NUM</u>	NOM	<u>NDEP</u>
E1	ALI	D1
E2	AMIN	D1
E3	SAMIR	D1
E4	SALIM	D2
E5	LYES	D2

DEP :

<u>NDEP</u>	<u>DIR</u>
D1	AMIN
D2	NADIR

Supposons une vue EMPDEP définie par la jointure de ces deux relations sur NDEP:

```
CREATE VIEW EMPDEP
AS SELECT NUM, NOM, EMP.NDEP, DIR
FROM EMP, DEP
WHERE EMP.NDEP = DEP.NDEP;
```

2.2. Fonctions des SGBD relationnels

2.2. 4. Vues Externes

EMP :

<u>NUM</u>	<u>NOM</u>	<u>NDEP</u>
E1	ALI	D1
E2	AMIN	D1
E3	SAMIR	D1
E4	SALIM	D2
E5	LYES	D2

DEP :

<u>NDEP</u>	<u>DIR</u>
D1	AMIN
D2	NADIR



```
CREATE VIEW EMPDEP  
AS SELECT NUM, NOM, EMP.NDEP, DIR  
FROM EMP, DEP  
WHERE EMP.NDEP = DEP.NDEP;
```



<u>NUM</u>	<u>NOM</u>	<u>NDEP</u>	<u>DIR</u>
E1	ALI	D1	AMIN
E2	AMIN	D1	AMIN
E3	SAMIR	D1	AMIN
E4	SALIM	D2	NADIR
E5	LYES	D2	NADIR

2.2. Fonctions des SGBD relationnels

2.2. 4. Vues Externes

NUM	NOM	NDEP	DIR
E1	ALI	D1	AMIN
E2	AMIN	D1	AMIN
E3	SAMIR	D1	AMIN
E4	SALIM	D2	NADIR
E5	LYES	D2	NADIR

```
CREATE VIEW EMPDEP  
AS SELECT NUM, NOM, EMP.NDEP, DIR  
FROM EMP, DEP  
WHERE EMP.NDEP = DEP.NDEP
```



Si nous autorisons l'utilisateur à modifier la vue EMPDEP de la manière suivante :

```
UPDATE EMPDEP  
SET DIR = 'MADJID'  
WHERE NUM = 'E1';
```



En d'autres termes cela revient à dire que nous changeons le directeur de E1 qui de AMIN devient MADJID. En considérant le fait qu'un directeur est lié à un département de manière fonctionnelle, nous créons ainsi une incohérence car qui est le directeur du département D1 AMIN ou MADJID ? Un problème qui rejoint la conception de la base de données

2.2. Fonctions des SGBD relationnels

2.2. 4. Vues Externes

En outre, une mise à jour au travers d'une vue est impossible si, par exemple, l'expression qui la définit comporte une fonction de calcul.

Plusieurs études ont été consacrées au problème de mise à jour au travers des vues. Mais dans la pratique, les SGBD sont plus draconiens dans l'attitude vis à vis de ce problème en ce sens qu'une mise à jour au travers d'une vue V n'est autorisée que si :

- V est bâtie sur une seule relation source, disons R ,
- Il existe une association une à une entre les n -uplets de V et ceux de R de telle sorte qu'une opération de modification (I, S, M) sur V se traduit en Une opération sur R .

Cela signifie que les insertions, suppressions, et modifications ne sont possibles au travers d'une vue que lorsque celle-ci ne comporte pas :

- D'opération de jointure
- L'option GROUP BY ou UNIQUE
- De fonction de calcul

