

## La fragmentation

1

### Fragmentation

- La fragmentation est le processus de **décomposition** d'une base de donnée logique en un ensemble de "sous" bases de données.
- Cette décomposition doit être sans perte d'information.
- La fragmentation peut être coûteuse s'il existe des applications qui possèdent des besoins opposés.
- Le processus de fragmentation peut se faire sous différentes stratégies: horizontale, verticale ou hybride.
- Tout stratégie de fragmentation doit satisfaire les trois règles de correctitude : **complétude**, **reconstruction** et **disjonction**.

2

## Fragmentation

### Règle de correction

- ▷ **Règle de complétion:** pour toute donnée d'une relation R, il existe un fragment  $R_i$  de la relation R qui possède cette donnée → pas de perte d'information
- ▷ **Règle de reconstruction:** il est toujours possible de reconstruire la relation R par un opérateur algèbre relationnelle
- ▷ **Règle de disjoncteur:** si une donnée apparaît dans un fragment alors elle ne peut apparaître dans aucun autre fragment sauf on autorise la redondance de l'attribut jointure pour la fragmentation verticale qui fait le lien entre les fragment

3

## Fragmentation

### Il existe trois type de fragmentation

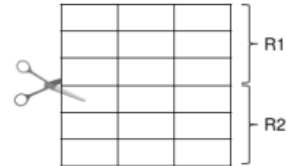
- ▷ **Fragmentation horizontale :** la table est fragmentée par rapport à ses instances en un ensemble de ligne
- ▷ **Fragmentation verticale:** la table est fragmentée selon ses attribut en un ensemble de colonne
- ▷ **Fragmentation mixte:** la table est fagmentée horizontalement et verticalement

4

## 1. Fragmentation horizontale FH

### Définition

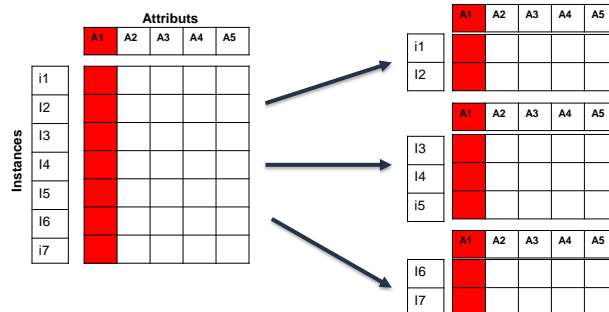
Ce type de fragmentation permet de découper une relation en sous relations contenant des sous-ensembles des tuples de la relation mère



- ▷ L'opérateur de fragmentation est **la restriction**
  - ▷ L'opérateur de reconstruction de R est **l'union**
- $R = R1 \text{ union } R2 \dots \text{ union } Rn$**

5

## 1. Fragmentation horizontale FH



6

## 1. Fragmentation horizontale FH

### Définition

- La fragmentation horizontale est basée sur l'**opération de sélection** qui utilise un **prédicat (condition)**.
- La reconstitution de la relation se fait par l'union des fragments.
- Nous avons besoin de définir deux notions: **prédicat simple**, et **prédicat minterm**.

### Prédicat simple:

- Soit une relation  $R(A_1, A_2, \dots, A_n)$ , tel que  $A_i$  est un attribut de  $R$  défini sur un domaine  $D_i$ . Un prédicat simple  $p_j$  est défini sur  $R$  comme suit:  **$p_j : A_i \theta \text{ val}$**   
 Tel que  $\theta \in \{=, <, >, \neq, \geq, \leq\}$  et  $\text{val} \in D_i$ .  
 On note  $Pr$  l'ensemble de tous les prédicats simples définis sur  $R$ .

7

## 1. Fragmentation horizontale FH

### Exemple de prédicat simple:

- Soit une entreprise qui a des annexes dans plusieurs villes: **Alger, Sétif, Blida**. Le gérant veut maintenir une base de données des employés et les projets sur lesquels ils travaillent.
- Le gérant utilise 4 tables comme suit:
  - ▷ **Employés** (idEmp, nomEmp, poste);
  - ▷ **Projetj** (idPrj, nomPrj, designation, budget, ville);
  - ▷ **Salaire** (poste, salaire);
  - ▷ **Affectation** (idEmp, idPrj, resp, dur) // Affectation d'un employé à un projet avec telle responsabilité et telle durée.
- Soit la relation **Projet**, on peut définir quelques prédicats simples:
  - ▷ nomPrj = "Construction "
  - ▷ budget  $\leq$  100000

8

## 1. Fragmentation horizontale FH

### Les types de fragmentation horizontale

Il existe deux types de fragmentation horizontale:

- ▷ primaire
- ▷ et dérivée.

9

## 1. Fragmentation horizontale FH

### Fragmentation horizontale primaire

FHP est la fragmentation horizontale qui est basée sur l'opération de sélection qui utilise un prédicat minterm:

$$R_i = \sigma_{m_i} (R)$$

Tel que :

- ▷  $R_i$  est un fragment de la relation  $R$ .
- ▷  $m_i$  est un minterm.
- ▷  $\sigma$  est l'opération de sélection

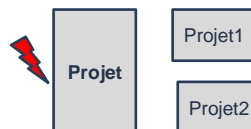
10

## 1. Fragmentation horizontale FH

### Exemple 1: Fragmentation horizontale primaire

Fragmentation de la relation Prj de l'exemple précédent en fragmentation horizontale peut être faite comme suit:

- ▷  $\text{Projet1} = \sigma_{\text{budget} \leq 100000} (\text{Prj})$
- ▷  $\text{Projet2} = \sigma_{\text{budget} > 100000} (\text{Prj})$



Cet exemple présente un des problèmes de la fragmentation horizontale. Si le domaine des attributs est continu et infini il sera difficile de définir le nombre maximal de fragmentations possibles. On ne peut pas partager le domaine en un nombre fini de sous-domaines

11

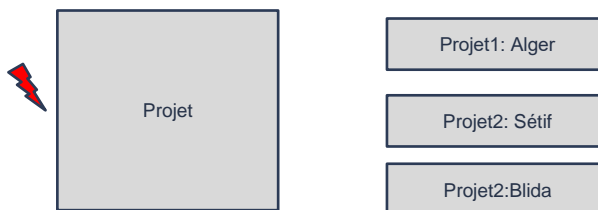
## 1. Fragmentation horizontale FH

### Exemple 2: Fragmentation horizontale primaire

On peut fragmenter maintenant la relation Projet en se basant sur la localisation comme suit:

- ▷  $\text{Projet 1} = \sigma_{\text{ville} = \text{"Alger"}} (\text{Projet})$
- ▷  $\text{Projet 2} = \sigma_{\text{ville} = \text{"Setif"}} (\text{Projet})$
- ▷  $\text{Projet 3} = \sigma_{\text{ville} = \text{"Blida"}} (\text{Projet})$

Dans ce cas, le nombre maximal de fragmentations possibles est fixe



12

## 1. Fragmentation horizontale FH

- Pour éviter les problèmes de fragmentation horizontale, nous avons besoin de bien **choisir les prédicats** simples qui seront à la base des prédicats minterm permettant de fragmenter une relation.
- D'une manière générale, cet ensemble doit **contenir seulement** les prédicats avec des **attributs** et **conditions** utilisés par les applications.

13

## 1. Fragmentation horizontale FH

### Fragmentation horizontale dérivée

- La FHD d'une relation(R) est basée sur des fragments horizontaux d'une autre relation(S) à condition qu'il y est un lien entre R et S.
- Les nouveaux fragments sont définis par semi-jointure.
- **Rappel:** La semi-jointure entre deux relations R et S ( $R \ltimes S$ ) permet de garder les tuples de R qui ont relation avec des tuples de S.

14

## 1. Fragmentation horizontale FH

### Exemple: Fragmentation horizontale dérivée

Dans l'exemple précédent, nous voulons grouper les employés de la relation **Employés** (**idEmp**, **nomEmp**, **poste**) en 2 groupes selon leurs salaires. Un groupe pour ceux qui ont un salaire **inférieur à 30000**, et les autres (**≥ 30000**) dans le deuxième groupe.

On fragmente d'abord la relation **Sal**(**poste**, **salaire**) en deux fragments comme suit:

▷ **Salaire1** =  $\sigma_{\text{salaire} < 30000}$  (**Salaire**)

▷ **Salaire2** =  $\sigma_{\text{salaire} \geq 30000}$  (**Salaire**)

Les deux fragments contenant les deux groupes de salariés sont définis ensuite par **semi-jointure** sur la colonne "poste" pour définir l'ensemble final des fragments comme suit:

▷ **Employés1** = **Employés**  $\bowtie_{\text{poste}}$  **Salaire1**

▷ **Employés2** = **Employés**  $\bowtie_{\text{poste}}$  **Salaire2**

▷ **Employés1** = **projection( les attributs employés )** (**Employés** Join<sub>poste</sub> **Salaire1**)

15

Poste	Salaire
Comptable	20000
Ingenieur	65555
admin	25000
technicien	45555

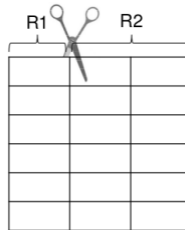
Poste	Salaire
Comptable	20000
admin	25000
Poste	Salaire
Ingenieur	65555
technicien	45555

16



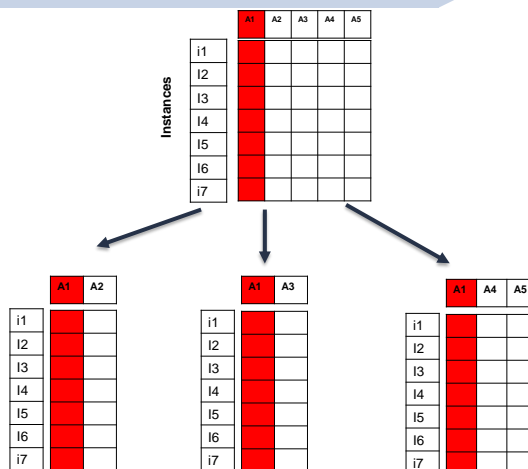
## 2. Fragmentation verticale

La **fragmentation verticale** permet de découper une relation R en plusieurs relations. Chaque fragment contient un sous-ensemble d'attributs ainsi que la clé primaire de R.



17

## 2. Fragmentation verticale



18

## 2. Fragmentation verticale

Une relation R **fragmentée verticalement** est décomposée en plusieurs sous ensemble  $R_1, R_2, \dots, R_n$  telque: chaque

$$R_i = \text{project} ( R / R_i )$$

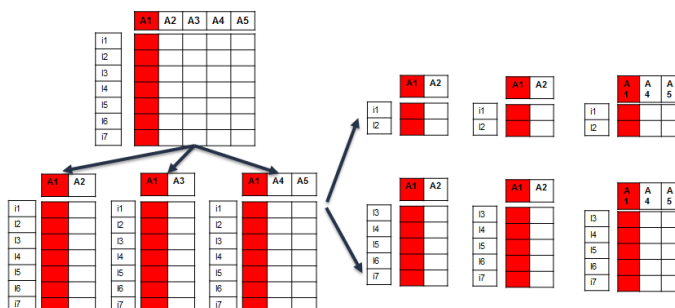
- ▷ L'opérateur de fragmentation est **la projection**
- ▷ L'opérateur de reconstruction de R est **la jointure**  
 $R = R_1 \text{ join } R_2 \dots \text{ join } R_n$

19

## 3. Fragmentation mixte

La Fragmentation Mixte(FM) (ou hybride) est une combinaison des deux fragmentations précédentes (FH et FV).

- ▷ **Verticale suivie d'une horizontale**

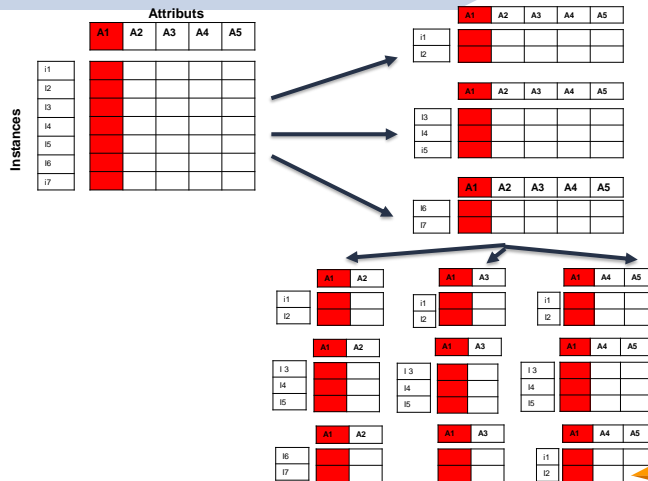


### 3. Fragmentation mixte

La Fragmentation Mixte(FM) (ou hybride) est une combinaison des deux fragmentations précédentes (FH etFV).

▷ **Horizontale** suivie d'une **verticale**

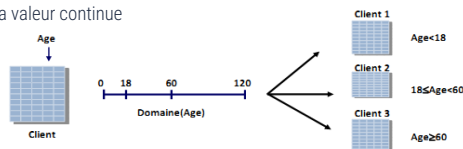
### 3. Fragmentation mixte



23

### Fragmentation par intervalle (domaine) « Range »

La fragmentation se fait sur un attribut à valeur continue

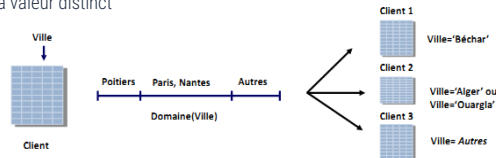


```
CREATE TABLE Client
( Client_id NUMBER(5),
  Nom Varchar2(20), Ville Varchar2(20),
  Age Number(3),
  Genre Varchar2(1))
PARTITION BY RANGE (Age) (
  PARTITION Client_Moins_18 VALUES LESS THAN (18) TABLESPACE TBSMoins27,
  PARTITION Client_18_59 VALUES LESS THAN (60) TABLESPACE TBS27-59,
  PARTITION Client_60_Et_Plus VALUES LESS THAN (MAXVALUES) TABLESPACE TBSPlus60
);
```

24

## Oracle: fragmentation par Liste

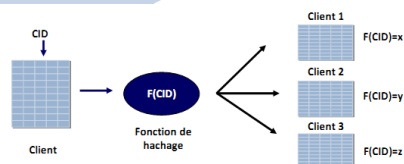
La fragmentation se fait sur un attribut à valeur distinct



```
CREATE TABLE Client
( Client_id NUMBER(5), Nom Varchar2(20), Ville Varchar2(20), Age Number(3),
  Genre Varchar2(1),
  PARTITION BY List (Ville) (
    PARTITION Client_Béchar VALUES ('Béchar') TABLESPACE TBSBECHAR,
    PARTITION Client_Alg_Oua VALUES ('Alger','Ouargla') TABLESPACE TBSALGOUA,
    PARTITION Client_Autres VALUES (DEFAULT) TABLESPACE TBSAUTRES
  );
```

25

## Oracle: fragmentation par hash



```
CREATE TABLE Client
(Client_id NUMBER(5),
 Nom Varchar2(20),
 Ville Varchar2(20),
 Age Number(3),
 Genre Varchar2(1),
 PARTITION BY Hash (CID)
(
  PARTITIONS 3 STORE IN (TBS1, TBS2, TBS3)
);
```

26

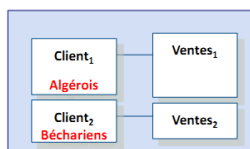
## Fragmentation composite

```
CREATE TABLE Client
(Client_id NUMBER(5),
Nom Varchar2(20), Ville Varchar2(20), Age Number(3),
Genre Varchar2(1))
PARTITION BY RANGE (Age),
SUBPARTITION BY LIST (Genre)
SUBPARTITION TEMPLATE
(
SUBPARTITION Client1 VALUES ('M') TABLESPACE TBSMasculin ,
SUBPARTITION Client2 VALUES ('F') TABLESPACE TBSFéminin )
(
PARTITION Client_Moins_18 VALUES LESS THAN (18),
PARTITION Client_18_59 VALUES LESS THAN (60),
PARTITION Client_60_Et_Plus VALUES LESS THAN (MAXVALUES)
);
```

27

## Fragmentation drivée par mode de référence

Fragmenter une table selon le schéma de fragmentation d'une autre table en utilisant le lien par clé étrangère



```
CREATE TABLE Ventes ( Client_id NUMBER(5), Time_id NUMBER(5),
Montant NUMBER(20),
CONSTRAINT order_items_fk FOREIGN KEY (Client_id) REFERENCES
Client (Client_id)
)
PARTITION BY REFERENCE (order_items_fk);
```

28

## Allocation des données

### Allocation centralisée

- Une seule base de données, une seule SGBD, sur un site des utilisateurs répartis sur le réseau.  
Tous les sites sauf le site central doivent passer par le réseau pour tous les accès aux données
- Coût de communication très élevé, fiabilité et disponibilité de données réduites

### Allocation fragmentée ou partitionnée

- partitionnement de la base de données en fragments disjoints distribués chacun sur un site
- Coût de stockage et de communication réduits mais fiabilité et disponibilité réduites

29

### Réplication complète

- Une copie de la totalité de la BD sur chaque site
- Fiabilité, disponibilité optimale mais coût de stockage et de communication élevé à cause des mises à jour

### Réplication sélective

- Combinaison de la centralisation, fragmentation et réplication
- Tous les avantages de chaque approche tout en évitant les inconvénients

30

## Recherche dans une table avec index

Soit la table **Etudiant** (matricule, nom, prenom, age, section )

Exemple: **SELECT \* FROM Etudiant WHERE Age = 24;**

- Un index sur Age permet d'obtenir un accès plus rapide sur les enregistrements des tuples des employés de 24 ans.

### Requêtes conjonctives

**SELECT \***

**FROM Etudiant**

**WHERE** age = 24 **AND** Section = 'M1';

- Supposons l'existence de deux index sur la relation Etudiant ; un sur **Age** et l'autre sur **Section**.
- L'utilisation du premier index va fournir les enregistrements satisfaisant la 1ère condition qui seront regroupés dans une table temporaire T1. Idem pour le deuxième index ~ T2
- Enfin l'intersection de T1 et T2 donnera la réponse