

BDA / TP N° 4: Base de Données Réparties (Fragmentation Horizontale)

1) Introduction : La Fragmentation Horizontale (FH) consiste à découper les relations en sous-relations appelées fragments. La répartition se fait donc en deux étapes : *la fragmentation* et *l'allocation* de ces fragments aux sites intéressés.

○ **Pourquoi fragmenter ?**

- Généralement les applications utilisent des sous-ensembles de relations.
- Une relation entière peut représenter une unité de distribution très grande
- Utilisation de petits fragments permet de faire tourner plus d'un processus simultanément.

○ **Comment fragmenter ?**

On distingue trois possibilité de fragmentation (partitionnement): (i) Fragmentation Horizontale, (ii) Fragmentation Verticale et (iii) Fragmentation Hybride.

Remarque: Trois types de partitionnement disponibles avec MySQL 5.1.4¹: **RANGE, LIST, HASH.**

2) Problème de sélection de schéma de fragmentation

Le processus de la fragmentation horizontale est illustré dans la figure suivante.

3) Formalisation de problème :

○ **Entrées :**

- Schéma de base de données SH (T_1, T_2, \dots, T_k)
- Ensemble de requêtes fréquentes Q
- Contrainte de maintenance : nombre maximal de fragments des tables

○ **Sortie :**

- Ensemble de sous-schémas minimisant le coût d'exécution de requêtes et satisfaisant la contrainte de maintenance

○ **Démarche à suivre :** La figure suivante illustre les étapes de fragmentation horizontale.

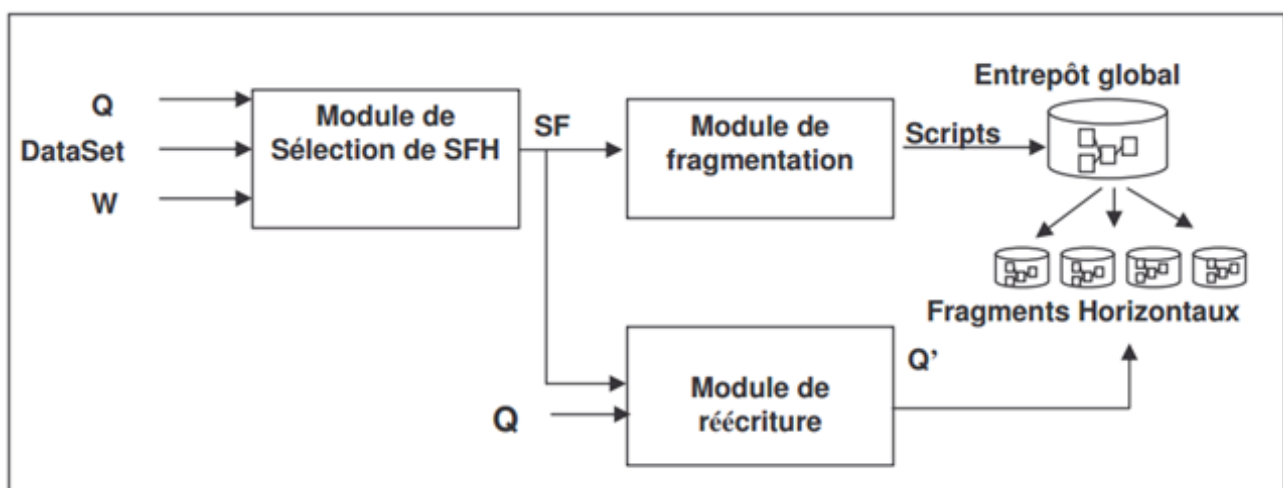


Fig. 1 Le processus de la fragmentation

¹ <http://www.sqlpac.com/referentiel/docs/mysql-5.1-partitionnement.htm>

1) Exemple de Création d'un fragment sous MySQL

```
MySQL> CREATE TABLE tr (id INT, name VARCHAR(50), purchased DATE)
->     PARTITION BY RANGE( YEAR(purchased) ) (
->         PARTITION p0 VALUES LESS THAN (1990),
->         PARTITION p1 VALUES LESS THAN (1995),
->         PARTITION p2 VALUES LESS THAN (2000),
->         PARTITION p3 VALUES LESS THAN (2005)
->     );
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> CREATE TABLE partition_date_test (dt DATE PRIMARY KEY)
-> PARTITION BY RANGE COLUMNS(dt) (
-> PARTITION day_20100226 VALUES LESS THAN ('2010-02-27'),
-> PARTITION day_20100227 VALUES LESS THAN ('2010-02-28'),

-> PARTITION max VALUES LESS THAN MAXVALUE);
Query OK, 0 rows affected (0.06 sec)
```

```
mysql> ALTER TABLE tr DROP PARTITION p2;

ALTER TABLE members ADD PARTITION (PARTITION p3 VALUES LESS THAN (2000));
ALTER TABLE members
    REORGANIZE PARTITION p0 INTO (
        PARTITION n0 VALUES LESS THAN (1960),
        PARTITION n1 VALUES LESS THAN (1970)
    );
```

Partition par list:

```
CREATE TABLE tt (
    id INT,
    data INT
)
PARTITION BY LIST(data) (
    PARTITION p0 VALUES IN (5, 10, 15),
    PARTITION p1 VALUES IN (6, 12, 18)
);
ALTER TABLE tt ADD PARTITION (PARTITION p2 VALUES IN (7, 14, 21));
```

Partitionnement par hachage (HASH) : Partitionnement quasiment identique avec celui par clé (KEY) :

```
CREATE TABLE orders_key2 (
    id INT AUTO_INCREMENT,
    customer_surname VARCHAR(30),
    store_id INT,
    salesperson_id INT,
    order_date DATE,
    note VARCHAR(500),
    INDEX idx (id)
) ENGINE = MYISAM
PARTITION BY KEY(salesperson_id, customer_surname)
```

Récriture des requêtes sur le schema fragmenté:

```
SQL> SELECT ename,sal,yearly_sal FROM emp_year_sal PARTITION (low_sal);
CREATE TABLE emp_year_sal
(ename VARCHAR2(20),
sal NUMBER,
yearly_sal AS (sal*12) VIRTUAL)
```

PARTITION BY RANGE (**yearly_sal**)
(PARTITION **low_sal** VALUES LESS THAN (20000),
PARTITION **mid_sal** VALUES LESS THAN (40000),
PARTITION **high_sal** VALUES LESS THAN (60000),
PARTITION **others** VALUES LESS THAN (MAXVALUE));

Travail Demandé:

- Créer la base de données SSB
- Charger un jeu de données de 1 GO.
- Extraire les prédicats de sélection des requêtes de la charge,
- Proposer un schéma de fragmentation,
- Réécrire des requêtes sur le schema ragmenté,
- Comparer les coûts de la charge de requêtes executes sur le schéma gobal et le schema fragmenté
- Commenter les résultats ??

Requête	Schema global	Schéma fragementé	Requêtes	Schema global	Schema fragementé
Q1.1			Q 3		
Q1.2			Q 3.1		
Q1.3			Q 3.2		
Q 2.1			Q 3.3		
Q 2.2			Q 3.4		
Q 4.1			Q 4.2		
Q 4.3					