Parallel and Distributed Databases
Rule 5:
Fragmentation Independence

Prepared For: CS527

Prepared By: Josh Levine, Pranav Shivkumar, Pratik Mistry, Shounak Rangwala, Swapnil Kamate, Vikhyat Dhamija

Data Fragmentation: Introduction(1)

- Data Fragmentation is breaking down data over the network and over the distributed databases
- It provides distribution transparency of the data over the DBs to the end user in DDBMS
- It further divides the normalized data in a distributed system
- Main goal of DDBMS is to provide the data to the user from the nearest location to them and as fast as possible
- Hence the data in a table are divided according to users locations or requirements

Data Fragmentation: Introduction(2)

- Example:
  Employee table is fragmented into three fragments and stored in NYC, LA, and Chicago sites. Assume a user query that demands for data from NYC site Emp table. The name of the Emp table at NYC site may be NYC.Employee, at LA site may be LA.Employee, and at Chicago site may be Chicago.Employee

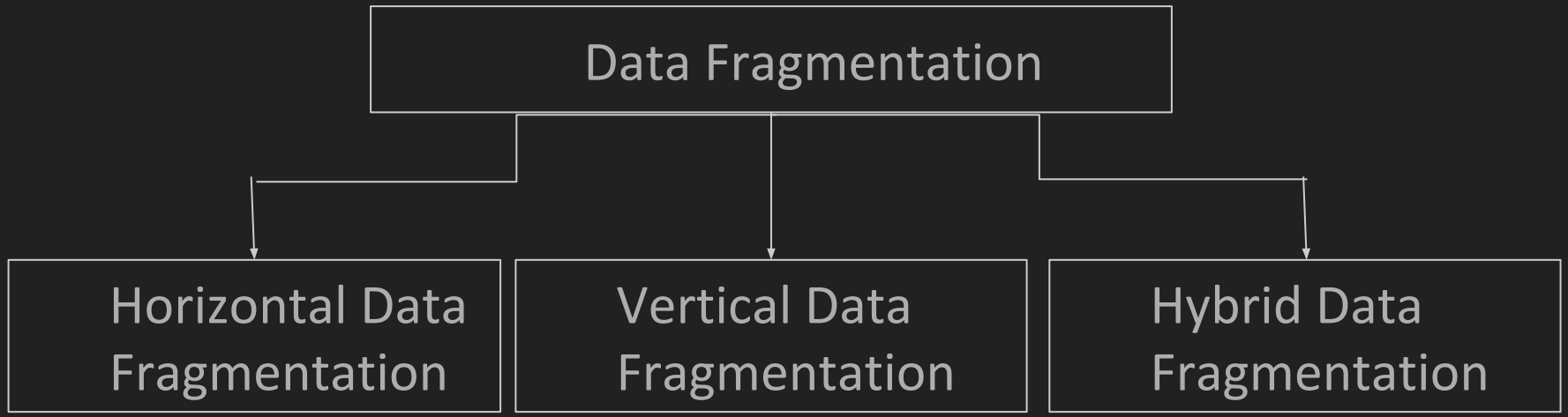
- Things to keep in mind while fragmenting the data:
  - Completeness - partial records in the table should not be considered while fragmenting
  - Reconstructions - should give whole table’s data while combining
  - Disjointedness - no overlapping of data in the fragments

Data Fragmentation: Pros and Cons

- Advantages:
  - Easy usage of Data
  - Efficiency
  - Security
  - Parallelism
  - Reliability
  - Balanced Storage

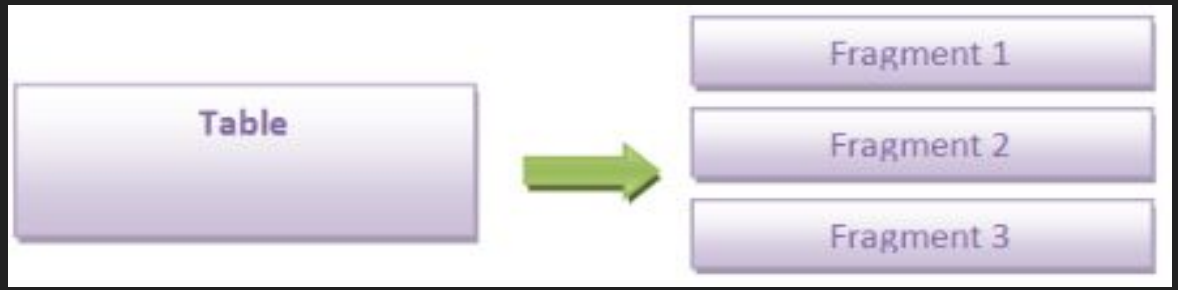
- Disadvantages:
  - When data from a different fragment is required, the access may be inefficient and may take very long
  - Lack of back-up copies in different data sites may render the database ineffective in case of site failure
  - If you are using recursive fragmentation, the reconstruction will need an expensive, complex technique

Data Fragmentation: Types



Horizontal Data Fragmentation

- Horizontal subset of table data are created and are stored in different database in DDB
- Data is fragmented in tuples/entities
- The union of the horizontal fragments should be the original table
- Data in the nearest DB will be accessed quickly



Horizontal Data Fragmentation: Example

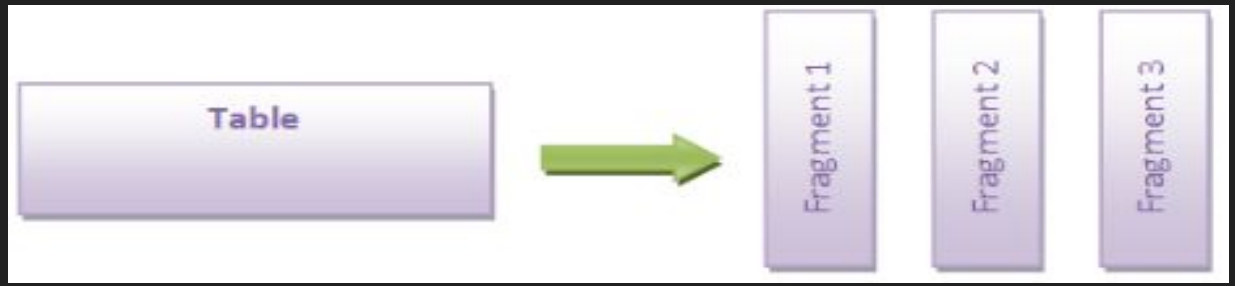
- Consider an organization having presence in India, UK and USA. Total employees across all locations will be huge and to retrieve employees record based on one of the locations, the whole table needs to be accessed.

- Hence, dividing table based on locations and to access records from any of the locations its size will be small and the retrieval is efficient
  Example:
  *SELECT * FROM EMPLOYEE WHERE EMP_LOCATION = 'INDIA;*
  *SELECT * FROM EMPLOYEE WHERE EMP_LOCATION = 'USA';*
  *SELECT * FROM EMPLOYEE WHERE EMP_LOCATION = 'UK;*

- These queries will give the subset of records from EMPLOYEE table depending on the location of the employees

Vertical Data Fragmentation

- Vertical subset of a relation i.e. a relation / table is fragmented by considering the columns of relation
- Efficient if there is consistent access to different attributes groups
- It may divide the table into different tables
- The join of the vertical fragments should be the original table



Vertical Data Fragmentation: Example

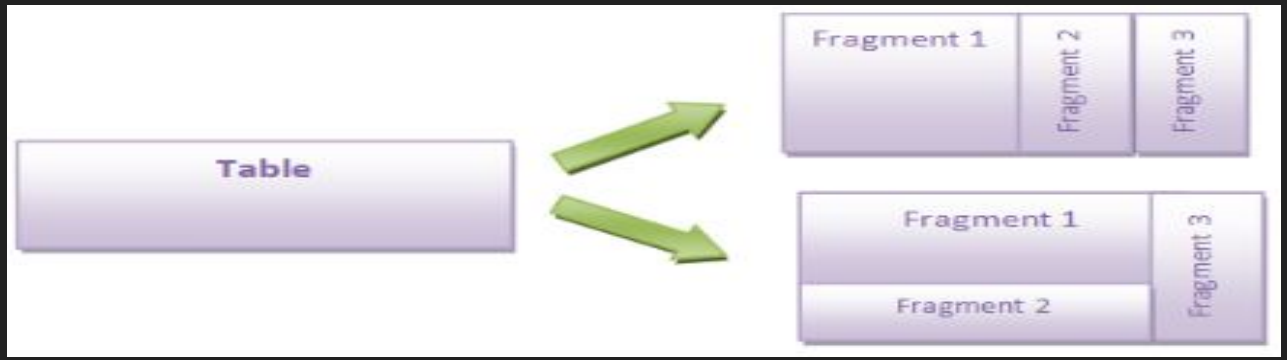
- Consider the earlier table as an example having ID, Name, Address, Age, location, DeptID, ProjID as attributes. To retrieve only specific columns without fragmentation, “Select *” needs to be used. So, creating vertical fragmentation based on need will be efficient.

- If we want to find the address of an employee, query only the second fragment
  Example:
  SELECT EMP_ID, EMP _FIRST_NAME, EMP_LAST_NAME, AGE FROM EMPLOYEE;
  SELECT EMP_ID, STREETNUM, TOWN, STATE, COUNTRY, PIN FROM EMPLOYEE;
  SELECT EMP_ID, DEPTID, ProjID FROM EMPLOYEE;

- This will avoid performing ‘SELECT *’ operation which will need lot of memory to query the whole table – to traverse whole data as well as to hold all the columns

Hybrid Data Fragmentation

- It will have horizontal fragmentation to have subset of data to be distributed over the DB, and vertical fragmentation to have subset of columns of the table
- Fragmentation can be done in any order
- The union on horizontal fragments and full outer join of the vertical fragments should result in the original table



Hybrid Data Fragmentation: Example

- Consider the earlier table as an example where we had fragmented it horizontally and vertically. Suppose we want to retrieve records of employees of a single location and only specific attributes, querying the entire table would be inefficient

- If we want to find the details of an employee of “India” location, query only the first fragment
  Example:
  SELECT EMP_ID, EMP _FIRST_NAME, EMP_LAST_NAME, AGE FROM EMPLOYEE WHERE EMP_LOCATION = ‘INDIA;
  SELECT EMP_ID, DEPTID FROM EMPLOYEE WHERE EMP_LOCATION = ‘INDIA;
  SELECT EMP_ID, EMP _FIRST_NAME, EMP_LAST_NAME, AGE FROM EMPLOYEE WHERE EMP_LOCATION = ‘US;
  SELECT EMP_ID, PROJID FROM EMPLOYEE WHERE EMP_LOCATION = ‘US;

- This query will be efficient in memory and speed

Thank You