CSE 4125: Distributed Database Systems
Chapter – 4

Distributed Database Design.

(part – B)

Outline

- Design of Derived Horizontal Fragmentation.
- Design of Vertical Fragmentation.
- Design of Mixed Fragmentation.
- Design of Allocation of the Fragmentation.

The Design of Derived Horizontal Fragmentation

Derived Horizontal Fragmentation

- The horizontal fragmentation of a relation cannot be based on a property of its own attributes, but is derived from the horizontal fragmentation of another relation.

- Derived fragmentation is used to facilitate the join between fragments.

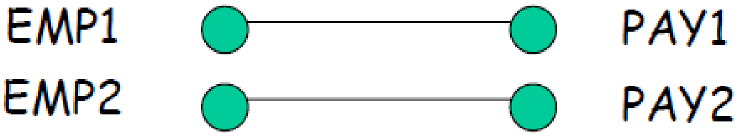
Distributed Join

- A **distributed join** is a join between horizontally fragmented relations.

- To join between two global relations *R* and *S*, all the tuples of *R* and *S* need to be compared.
  - So, it is necessary to compare all the fragments R_i of *R* with all the fragments S_j of *S*.

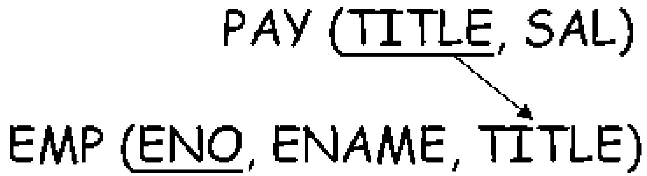
Join Graph

- A distributed join is represented efficiently using **join graphs**.
- The join graph *G* of the distributed join *R JN S* is a graph *(N, E)*, where
  - nodes *N*: fragments of *R* and *S*
  - nondirected edges *E*: Join between fragments which are not intrinsically empty.

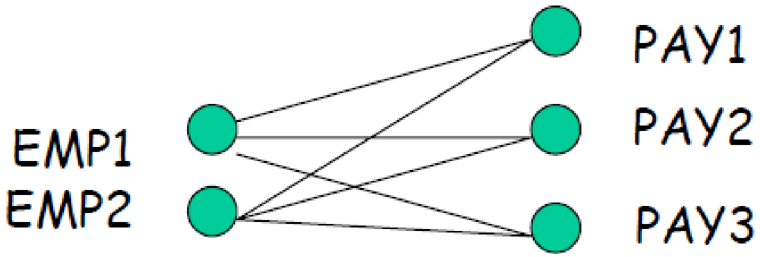


Join Graph (cont.)

- Example:



  - Divide ***EMP*** into *EMP1* and *EMP2* based on *TITLE*
  - Divide ***PAY*** into *PAY1*, *PAY2*, *PAY3* based on *SAL*.
- To join ***EMP*** and ***PAY***, we have the following scenario.



The Design of Vertical Fragmentation

Vertical Fragmentation

Partitioning the attributes of a relation into a set of smaller relations.

- So that many of the applications will run on only one fragment.

Types:

- *Vertical Partitioning:* sets must be disjoint.
- *Vertical clustering:* sets can be overlapped.

Vertical Partitioning

Approaches:

- *Grouping:* Progressively assigning each attribute to constitute fragment.
    - Top-down/ Bottom-up approach?
- *Splitting:* Progressively splitting global relations into fragment.
    - Top-down/ Bottom-up approach?

Vertical Clustering

Introduces *replication*.

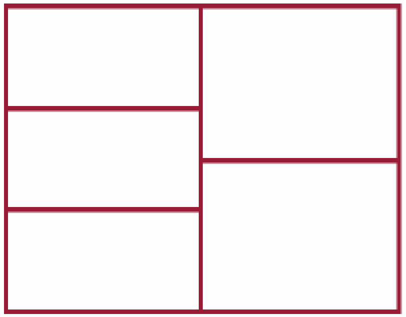
- Tuple identifier.

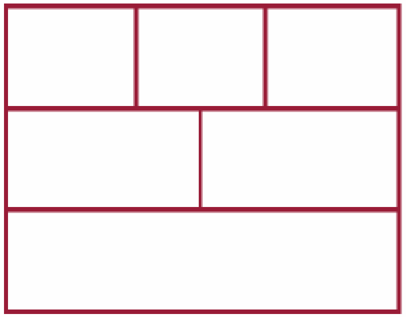
- *Convenient* for read-only application.
  - Why?
- *Not convenient* for update application.
  - Why?

The Design of Mixed Fragmentation

Approaches

1. Applying horizontal fragmentation to vertical fragmentation.
2. Applying vertical fragmentation to horizontal fragmentation.





The Design of Allocation of Fragmentation

Types of Allocation of the Fragments

1. Nonredundant allocation.
   - One copy → one site.

2. Redundant allocation.
   - One copy → many sites.
   - Replication.

Method for Nonredundant Allocation

Best fit:

- A measure is associated with each possible sites.
  - i.e. measuring the access frequency by local application.
- Site with best measure is selected.

Method for Redundant Allocation

a) All beneficial sites:

- Determining the set of sites where *benefit is higher.*
- For example, determining sites where–
  - *cost for update operation < cost for read-only application*.
- Allocate fragments to each one of those sites.

Method for Redundant Allocation (cont.)

b) Additional replication:

- Firstly, do the nonreplication.
- Then progressively add replicated fragments.
  - From most beneficial to less beneficial.
- *Degree of redundancy* is considered.
  - Because benefit does not grow proportionally to the degree of redundancy.

Additional Reading

- Different types of join graphs.
- The purpose of vertical fragmentation.
- Disadvantages of –
  - *All beneficial sites.*
  - *Additional replication.*

Practice Problems/ Questions

- Text book: Exercise 4.3