

Advanced Management of Data

Concepts of Distributed Databases (1)

Distributed Databases

Distributed Computing System

- number of processing sites or nodes (computers)
- nodes are interconnected by a computer network
- nodes cooperate in performing certain tasks

Goal

- decompose big, unmanageable problems into smaller tasks
- solve the parts in a coordinated way

Distributed Databases

Big Data Technologies

Distributed databases combine concepts of distributed systems and database technologies to deal with the vast amounts of data that are being produced and collected, which includes

- storage
- retrieval
- analysis & mining
 - data mining and machine learning algorithms are often used to extract the needed knowledge

Distributed Database Concepts

Distributed Database (DDB)

- collection of database nodes that must be
 - [logically related](#)
- connected over a computer network to transmit data and commands among sites

Distributed database management system (DDBMS)

- software system that manages a distributed database
- in most cases the distribution is made [transparent](#) to the user

Distributed Database Concepts

Heterogeneous DDBMS

Different sites may run different hardware, different DBMS products, and may even be based on different underlying data models. Translations are required to allow communication between different DBMSs.

Heterogeneous systems usually result when individual sites have implemented their own databases and integration is considered at a later stage.

Homogeneous DDBMS

All sites use the same DBMS product.

Homogeneous systems are much easier to design and manage. They provide incremental growth, making the addition of a new site to the DDBMS easy, and allow increased performance by exploiting the parallel processing capability of multiple sites.

Distributed Database Concepts

Transparency

- data organization / distribution / network transparency
 - location transparency
 - naming transparency
- replication transparency
- fragmentation transparency
 - horizontal fragmentation
 - vertical fragmentation
- design transparency
- execution transparency

PROJECT

| Pname | <u>Pnumber</u> | Plocation | Dnum |
|-------|----------------|-----------|------|
|-------|----------------|-----------|------|

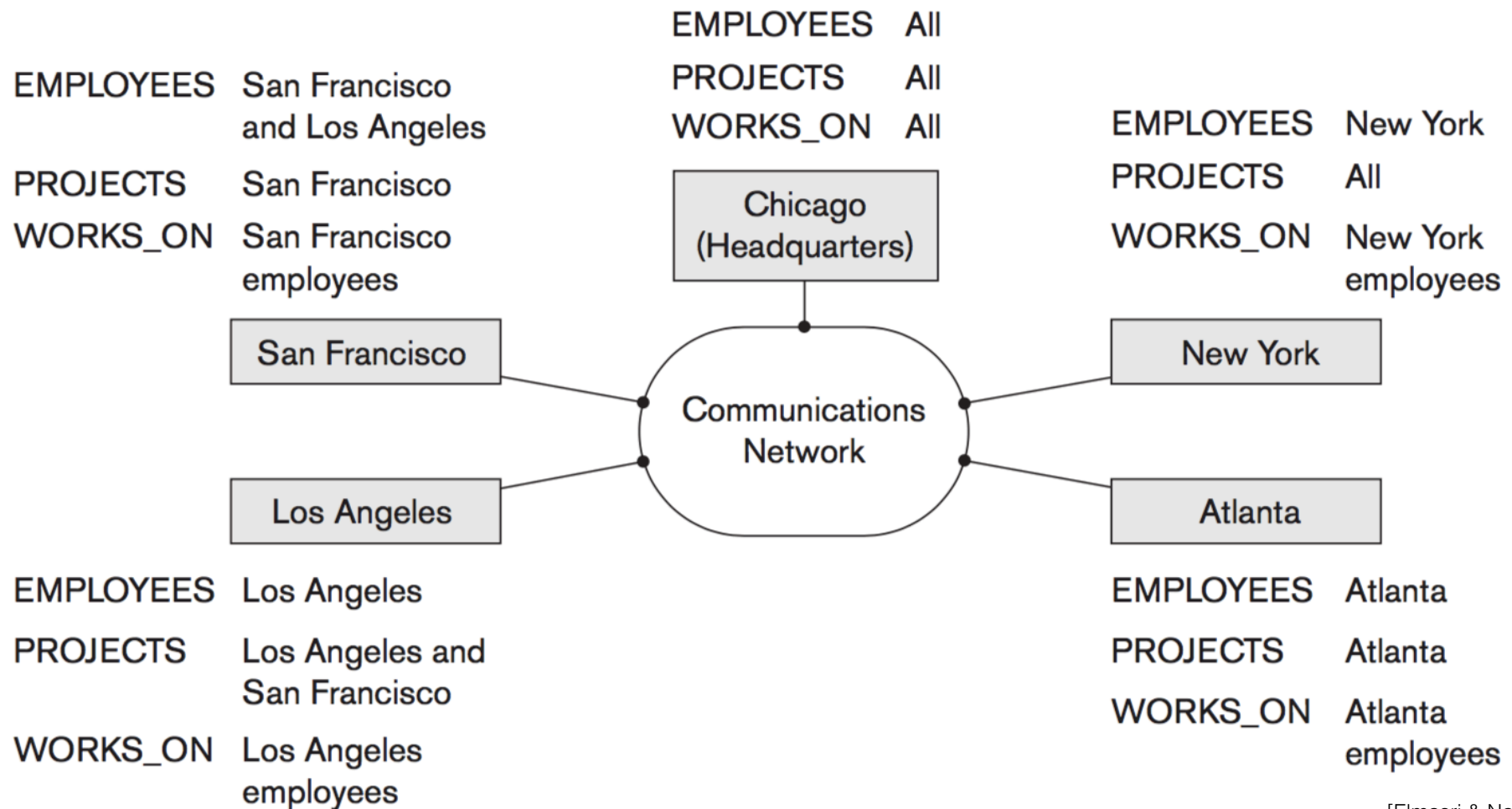
WORKS_ON

| <u>Essn</u> | <u>Pno</u> | Hours |
|-------------|------------|-------|
|-------------|------------|-------|

EMPLOYEE

| Fname | Minit | Lname | <u>Ssn</u> | Bdate | Address | Sex | Salary | Super_ssn | Dno |
|-------|-------|-------|------------|-------|---------|-----|--------|-----------|-----|
|-------|-------|-------|------------|-------|---------|-----|--------|-----------|-----|

Example



Distributed Database Concepts

Reliability

- probability that a system is running at a certain time point.

Availability

- probability that the system is continuously available during a time interval

Failure

- deviation of a system's behaviour from that which is specified in order to ensure correct execution of operations

Errors

- subset of system states that causes the failure

Fault

- cause of an error

Distributed Database Concepts

Cause of DDBMS failures

- transactions
- hardware
- communication networks
 - occur due to errors associated with messages and line failures
 - message errors can include loss, corruption, or out-of-order arrival at destination

Reliable DDBMS

- tolerates failures of underlying components
- processes user requests as long as database consistency is not violated

Distributed Database Concepts

Scalability

Scalability determines the extent to expand the capacity of a distributed system while continuing to operate without interruption:

- **Horizontal scalability**

The number of nodes in the distributed system can be expanded to distribute some of the data and processing loads from existing nodes to the new nodes.

- **Vertical scalability**

The capacity of individual nodes in the system can be expanded.

Distributed Database Concepts

Network Partitioning

As the number of nodes of a distributed system expands, it is possible that the connecting network may have faults.

This can cause the nodes to be partitioned into groups of nodes.

The nodes within each partition are still connected by a subnetwork, but communication among the partitions is lost.

Partition tolerance

A distributed system should have the capacity to continue operating while the network is partitioned.

Distributed Database Concepts

Autonomy

determines the extent to which individual nodes in a DDBMS can operate independently.

- design autonomy Data model usage and transaction management techniques among nodes are independent.
- communication autonomy To which extent each node can decide on sharing of information with other nodes?
- execution autonomy Users are independent to act as they wish.

Data Fragmentation

Horizontal Fragmentation (Sharding)

Horizontal fragmentation divides a relation R horizontally by grouping rows to create subsets of tuples (shards). Each subset has a certain logical meaning and can be specified in the relational algebra by a $\sigma_{C_i}(R)$ operation.

Complete horizontal fragmentation

Is a set of horizontal fragments whose conditions C_1, C_2, \dots, C_n include all the tuples in R , that is, every tuple in R satisfies $(C_1 \text{ OR } C_2 \text{ OR } \dots \text{ OR } C_n)$

By applying a UNION operation, the relation R can be reconstructed from a complete horizontal fragmentation.

Derived horizontal fragmentation

The partitioning of a primary relation can be applied to other (secondary) relations, which are related to the primary relation via a foreign key.

Example

EMPLOYEE

| Fname | Minit | Lname | <u>Ssn</u> | Bdate | Address | Sex | Salary | Super_ssn | Dno |
|----------|-------|---------|------------|------------|--------------------------|-----|--------|-----------|-----|
| John | B | Smith | 123456789 | 1965-01-09 | 731 Fondren, Houston, TX | M | 30000 | 333445555 | 5 |
| Franklin | T | Wong | 333445555 | 1955-12-08 | 638 Voss, Houston, TX | M | 40000 | 888665555 | 5 |
| Alicia | J | Zelaya | 999887777 | 1968-01-19 | 3321 Castle, Spring, TX | F | 25000 | 987654321 | 4 |
| Jennifer | S | Wallace | 987654321 | 1941-06-20 | 291 Berry, Bellaire, TX | F | 43000 | 888665555 | 4 |
| Ramesh | K | Narayan | 666884444 | 1962-09-15 | 975 Fire Oak, Humble, TX | M | 38000 | 333445555 | 5 |
| Joyce | A | English | 453453453 | 1972-07-31 | 5631 Rice, Houston, TX | F | 25000 | 333445555 | 5 |
| Ahmad | V | Jabbar | 987987987 | 1969-03-29 | 980 Dallas, Houston, TX | M | 25000 | 987654321 | 4 |
| James | E | Borg | 888665555 | 1937-11-10 | 450 Stone, Houston, TX | M | 55000 | NULL | 1 |

[Elmasri & Navathe]

Sharding based on Department (Dno): The following set C_n is a complete horizontal fragmentation:

C_1 : select * from EMPLOYEE where Dno = 1

C_2 : select * from EMPLOYEE where Dno = 4

C_3 : select * from EMPLOYEE where Dno = 5

Data Fragmentation

Vertical Fragmentation

A vertical fragment of a relation keeps only certain attributes of a relation R and can be specified by a $\pi_{L_i}(R)$ operation in the relational algebra.

Complete Vertical Fragmentation

A set of vertical fragments whose projection lists L_1, L_2, \dots, L_n include all the attributes in R but share only the primary key attribute of R . The projection lists satisfy the following conditions:

- $L_1 \cup L_2 \cup \dots \cup L_n = \text{ATTRS}(R)$, where $\text{ATTRS}(R)$ is the set of attributes of R
- $L_i \cap L_j = \text{PK}(R)$ for any $i \neq j$, where $\text{PK}(R)$ is the primary key of R

R can be reconstructed from a complete vertical fragmentation by applying a Full-Outer-Join operation.

Example

PROJECT

| Pname | <u>Pnumber</u> | Plocation | Dnum |
|-----------------|----------------|-----------|------|
| ProductX | 1 | Bellaire | 5 |
| ProductY | 2 | Sugarland | 5 |
| ProductZ | 3 | Houston | 5 |
| Computerization | 10 | Stafford | 4 |
| Reorganization | 20 | Houston | 1 |
| Newbenefits | 30 | Stafford | 4 |

Complete vertical fragmentation of PROJECT into

- PROJ_DATA:
SELECT Pname, Pnumber, Plocation
FROM PROJECT
- PROJ_DEPT:
SELECT Pnumber, Dnum
FROM PROJECT

PROJ_DATA

| Pname | <u>Pnumber</u> | Plocation |
|-----------------|----------------|-----------|
| ProductX | 1 | Bellaire |
| ProductY | 2 | Sugarland |
| ProductZ | 3 | Houston |
| Computerization | 10 | Stafford |
| Reorganization | 20 | Houston |
| Newbenefits | 30 | Stafford |

PROJ_DEPT

| <u>Pnumber</u> | Dnum |
|----------------|------|
| 1 | 5 |
| 2 | 5 |
| 3 | 5 |
| 10 | 4 |
| 20 | 1 |
| 30 | 4 |

Data Fragmentation

Mixed Fragmentation

Horizontal and vertical fragmentation can be intermixed.

A fragment of a relation R can be specified by a selection / projection combination of operations $\pi_L(\sigma_C(R))$:

- vertical fragment: $C = \text{TRUE}$ and $L \neq \text{ATTRS}(R)$
- horizontal fragment: $C \neq \text{TRUE}$ and $L = \text{ATTRS}(R)$
- mixed fragment: $C \neq \text{TRUE}$ and $L \neq \text{ATTRS}(R)$
- relation itself: $C = \text{TRUE}$ and $L = \text{ATTRS}(R)$

Data Fragmentation

Fragmentation schema

- defines a set of fragments that includes all attributes and tuples in the database

By applying some sequence of OUTER JOIN and UNION operations the whole database must be reconstructable from these fragments.

Allocation schema

- describes the allocation of fragments to nodes

Replication

A fragment is said to be **replicated**, if it is stored at more than one site.

Data Replication

Full Replication

The **whole database** is replicated at **every node** in the distributed system.

Advantages: As long as at least one site is up, the system can continue to operate, which maximizes availability.

It also improves performance of retrieval for global queries because the results of such queries can be obtained locally from any one site.

Disadvantages: Update operations can be slowed down drastically, since **every copy** must be updated similarly to keep consistency.

Concurrency control and recovery techniques are becoming more expensive.

Data Replication

No Replication

Each fragment is stored at only one site.

In this case, all fragments must be disjoint, except for the repetition of primary keys among vertical or mixed fragments.

Partial Replication

Some fragments of the database may be replicated whereas others may not.

The number of copies of each fragment can range from one up to the total number of sites in the distributed system.

Replication Schema

A description of the replication of fragments.

Data Allocation

Criteria for data distribution

Each (copy of a) fragment must be assigned to a particular site in the distributed system.

The choice of sites and the degree of replication depend on

- performance and availability goals of the system
- types and frequencies of transactions submitted at each site

Data that is frequently accessed at multiple sites should be replicated at those sites.

If many updates are performed, it may be useful to [limit replication](#).

Finding an optimal or even a good solution to distributed data allocation is a [complex optimization problem](#).

Example (1)

We want to fragment and distribute the following company database:

EMPLOYEE

| | | | | | | | | | |
|-------|-------|-------|------------|-------|---------|-----|--------|-----------|-----|
| Fname | Minit | Lname | <u>Ssn</u> | Bdate | Address | Sex | Salary | Super_ssn | Dno |
|-------|-------|-------|------------|-------|---------|-----|--------|-----------|-----|

DEPARTMENT

| | | | |
|-------|----------------|---------|----------------|
| Dname | <u>Dnumber</u> | Mgr_ssn | Mgr_start_date |
|-------|----------------|---------|----------------|

DEPT_LOCATIONS

| | |
|----------------|------------------|
| <u>Dnumber</u> | <u>Dlocation</u> |
|----------------|------------------|

PROJECT

| | | | |
|-------|----------------|-----------|------|
| Pname | <u>Pnumber</u> | Plocation | Dnum |
|-------|----------------|-----------|------|

WORKS_ON

| | | |
|-------------|------------|-------|
| <u>Essn</u> | <u>Pno</u> | Hours |
|-------------|------------|-------|

Example (2)

Description / Requirements

The company has **three** computer sites - one for each department:

- Site 1 is used by company headquarters and accesses all employee and project information regularly.
- Site 2 is used for department 5 only
- Site 3 is used for department 4 only

At all sites, we expect frequent access to the EMPLOYEE and PROJECT information for the employees who work in that department and the projects controlled by that department.

These sites mainly access the Name, Ssn, Salary, and Super_ssn attributes of EMPLOYEE.

Example (3)

According to these requirements, the whole database can be stored at site 1.

EMPLOYEE

| Fname | Minit | Lname | <u>Ssn</u> | Bdate | Address | Sex | Salary | Super_ssn | Dno |
|----------|-------|---------|------------|------------|--------------------------|-----|--------|-----------|-----|
| John | B | Smith | 123456789 | 1965-01-09 | 731 Fondren, Houston, TX | M | 30000 | 333445555 | 5 |
| Franklin | T | Wong | 333445555 | 1955-12-08 | 638 Voss, Houston, TX | M | 40000 | 888665555 | 5 |
| Alicia | J | Zelaya | 999887777 | 1968-01-19 | 3321 Castle, Spring, TX | F | 25000 | 987654321 | 4 |
| Jennifer | S | Wallace | 987654321 | 1941-06-20 | 291 Berry, Bellaire, TX | F | 43000 | 888665555 | 4 |
| Ramesh | K | Narayan | 666884444 | 1962-09-15 | 975 Fire Oak, Humble, TX | M | 38000 | 333445555 | 5 |
| Joyce | A | English | 453453453 | 1972-07-31 | 5631 Rice, Houston, TX | F | 25000 | 333445555 | 5 |
| Ahmad | V | Jabbar | 987987987 | 1969-03-29 | 980 Dallas, Houston, TX | M | 25000 | 987654321 | 4 |
| James | E | Borg | 888665555 | 1937-11-10 | 450 Stone, Houston, TX | M | 55000 | NULL | 1 |

DEPARTMENT

| Dname | <u>Dnumber</u> | Mgr_ssn | Mgr_start_date |
|----------------|----------------|-----------|----------------|
| Research | 5 | 333445555 | 1988-05-22 |
| Administration | 4 | 987654321 | 1995-01-01 |
| Headquarters | 1 | 888665555 | 1981-06-19 |

DEPT_LOCATIONS

| <u>Dnumber</u> | <u>Dlocation</u> |
|----------------|------------------|
| 1 | Houston |
| 4 | Stafford |
| 5 | Bellaire |
| 5 | Sugarland |
| 5 | Houston |

Example (4)

According to these requirements, the whole database can be stored at site 1.

WORKS_ON

| <u>Essn</u> | <u>Pno</u> | Hours |
|-------------|------------|-------|
| 123456789 | 1 | 32.5 |
| 123456789 | 2 | 7.5 |
| 666884444 | 3 | 40.0 |
| 453453453 | 1 | 20.0 |
| 453453453 | 2 | 20.0 |
| 333445555 | 2 | 10.0 |
| 333445555 | 3 | 10.0 |
| 333445555 | 10 | 10.0 |
| 333445555 | 20 | 10.0 |
| 999887777 | 30 | 30.0 |
| 999887777 | 10 | 10.0 |
| 987987987 | 10 | 35.0 |
| 987987987 | 30 | 5.0 |
| 987654321 | 30 | 20.0 |
| 987654321 | 20 | 15.0 |
| 888665555 | 20 | NULL |

PROJECT

| <u>Pname</u> | <u>Pnumber</u> | Plocation | Dnum |
|-----------------|----------------|-----------|------|
| ProductX | 1 | Bellaire | 5 |
| ProductY | 2 | Sugarland | 5 |
| ProductZ | 3 | Houston | 5 |
| Computerization | 10 | Stafford | 4 |
| Reorganization | 20 | Houston | 1 |
| Newbenefits | 30 | Stafford | 4 |

Example (5)

To determine the fragments to be replicated at sites 2 and 3, first we **horizontally fragment** DEPARTMENT by its key (Dnumber).

DEPARTMENT

| Dname | <u>Dnumber</u> | Mgr_ssn | Mgr_start_date |
|----------------|----------------|-----------|----------------|
| Research | 5 | 333445555 | 1988-05-22 |
| Administration | 4 | 987654321 | 1995-01-01 |
| Headquarters | 1 | 888665555 | 1981-06-19 |

DEP_5

| Dname | <u>Dnumber</u> | Mgr_ssn | Mgr_start_date |
|----------|----------------|-----------|----------------|
| Research | 5 | 333445555 | 1988-05-22 |

DEP_4

| Dname | <u>Dnumber</u> | Mgr_ssn | Mgr_start_date |
|----------------|----------------|-----------|----------------|
| Administration | 4 | 987654321 | 1995-01-01 |

Example (6)

We apply [derived fragmentation](#) to the PROJECT and DEPT_LOCATIONS relations based on their foreign keys for department number (Dnum, Dnumber).

PROJECT

| Pname | <u>Pnumber</u> | Plocation | Dnum |
|-----------------|----------------|-----------|------|
| ProductX | 1 | Bellaire | 5 |
| ProductY | 2 | Sugarland | 5 |
| ProductZ | 3 | Houston | 5 |
| Computerization | 10 | Stafford | 4 |
| Reorganization | 20 | Houston | 1 |
| Newbenefits | 30 | Stafford | 4 |

DEPT_LOCATIONS

| <u>Dnumber</u> | <u>Dlocation</u> |
|----------------|------------------|
| 1 | Houston |
| 4 | Stafford |
| 5 | Bellaire |
| 5 | Sugarland |
| 5 | Houston |

PROJS_5

| Pname | <u>Pnumber</u> | Plocation | Dnum |
|----------|----------------|-----------|------|
| ProductX | 1 | Bellaire | 5 |
| ProductY | 2 | Sugarland | 5 |
| ProductZ | 3 | Houston | 5 |

DEP_5_LOCS

| <u>Dnumber</u> | <u>Dlocation</u> |
|----------------|------------------|
| 5 | Bellaire |
| 5 | Sugarland |
| 5 | Houston |

PROJS_4

| Pname | <u>Pnumber</u> | Plocation | Dnum |
|-----------------|----------------|-----------|------|
| Computerization | 10 | Stafford | 4 |
| Newbenefits | 30 | Stafford | 4 |

DEP_4_LOCS

| <u>Dnumber</u> | <u>Dlocation</u> |
|----------------|------------------|
| 4 | Stafford |

Example (7)

We apply [derived fragmentation](#) to EMPLOYEE based on its foreign keys for department number (Dno).

EMPLOYEE

| Fname | Minit | Lname | <u>Ssn</u> | Bdate | Address | Sex | Salary | Super_ssn | Dno |
|----------|-------|---------|------------|------------|--------------------------|-----|--------|-----------|-----|
| John | B | Smith | 123456789 | 1965-01-09 | 731 Fondren, Houston, TX | M | 30000 | 333445555 | 5 |
| Franklin | T | Wong | 333445555 | 1955-12-08 | 638 Voss, Houston, TX | M | 40000 | 888665555 | 5 |
| Alicia | J | Zelaya | 999887777 | 1968-01-19 | 3321 Castle, Spring, TX | F | 25000 | 987654321 | 4 |
| Jennifer | S | Wallace | 987654321 | 1941-06-20 | 291 Berry, Bellaire, TX | F | 43000 | 888665555 | 4 |
| Ramesh | K | Narayan | 666884444 | 1962-09-15 | 975 Fire Oak, Humble, TX | M | 38000 | 333445555 | 5 |
| Joyce | A | English | 453453453 | 1972-07-31 | 5631 Rice, Houston, TX | F | 25000 | 333445555 | 5 |
| Ahmad | V | Jabbar | 987987987 | 1969-03-29 | 980 Dallas, Houston, TX | M | 25000 | 987654321 | 4 |
| James | E | Borg | 888665555 | 1937-11-10 | 450 Stone, Houston, TX | M | 55000 | NULL | 1 |

| Fname | Minit | Lname | <u>Ssn</u> | Bdate | Address | Sex | Salary | Super_ssn | Dno |
|----------|-------|---------|------------|------------|--------------------------|-----|--------|-----------|-----|
| John | B | Smith | 123456789 | 1965-01-09 | 731 Fondren, Houston, TX | M | 30000 | 333445555 | 5 |
| Franklin | T | Wong | 333445555 | 1955-12-08 | 638 Voss, Houston, TX | M | 40000 | 888665555 | 5 |
| Ramesh | K | Narayan | 666884444 | 1962-09-15 | 975 Fire Oak, Humble, TX | M | 38000 | 333445555 | 5 |
| Joyce | A | English | 453453453 | 1972-07-31 | 5631 Rice, Houston, TX | F | 25000 | 333445555 | 5 |

| Fname | Minit | Lname | <u>Ssn</u> | Bdate | Address | Sex | Salary | Super_ssn | Dno |
|----------|-------|---------|------------|------------|-------------------------|-----|--------|-----------|-----|
| Alicia | J | Zelaya | 999887777 | 1968-01-19 | 3321 Castle, Spring, TX | F | 25000 | 987654321 | 4 |
| Jennifer | S | Wallace | 987654321 | 1941-06-20 | 291 Berry, Bellaire, TX | F | 43000 | 888665555 | 4 |
| Ahmad | V | Jabbar | 987987987 | 1969-03-29 | 980 Dallas, Houston, TX | M | 25000 | 987654321 | 4 |

Example (8)

We **vertically fragment** the resulting EMPLOYEE fragments to include only the most frequently accessed attributes.

| Fname | Minit | Lname | <u>Ssn</u> | Bdate | Address | Sex | Salary | Super_ssn | Dno |
|----------|-------|---------|------------|------------|--------------------------|-----|--------|-----------|-----|
| John | B | Smith | 123456789 | 1965-01-09 | 731 Fondren, Houston, TX | M | 30000 | 333445555 | 5 |
| Franklin | T | Wong | 333445555 | 1955-12-08 | 638 Voss, Houston, TX | M | 40000 | 888665555 | 5 |
| Ramesh | K | Narayan | 666884444 | 1962-09-15 | 975 Fire Oak, Humble, TX | M | 38000 | 333445555 | 5 |
| Joyce | A | English | 453453453 | 1972-07-31 | 5631 Rice, Houston, TX | F | 25000 | 333445555 | 5 |

EMPD_5

| Fname | Minit | Lname | <u>Ssn</u> | Salary | Super_ssn | Dno |
|----------|-------|---------|------------|--------|-----------|-----|
| John | B | Smith | 123456789 | 30000 | 333445555 | 5 |
| Franklin | T | Wong | 333445555 | 40000 | 888665555 | 5 |
| Ramesh | K | Narayan | 666884444 | 38000 | 333445555 | 5 |
| Joyce | A | English | 453453453 | 25000 | 333445555 | 5 |

| Fname | Minit | Lname | <u>Ssn</u> | Bdate | Address | Sex | Salary | Super_ssn | Dno |
|----------|-------|---------|------------|------------|-------------------------|-----|--------|-----------|-----|
| Alicia | J | Zelaya | 999887777 | 1968-01-19 | 3321 Castle, Spring, TX | F | 25000 | 987654321 | 4 |
| Jennifer | S | Wallace | 987654321 | 1941-06-20 | 291 Berry, Bellaire, TX | F | 43000 | 888665555 | 4 |
| Ahmad | V | Jabbar | 987987987 | 1969-03-29 | 980 Dallas, Houston, TX | M | 25000 | 987654321 | 4 |

EMPD_4

| Fname | Minit | Lname | <u>Ssn</u> | Salary | Super_ssn | Dno |
|----------|-------|---------|------------|--------|-----------|-----|
| Alicia | J | Zelaya | 999887777 | 25000 | 987654321 | 4 |
| Jennifer | S | Wallace | 987654321 | 43000 | 888665555 | 4 |
| Ahmad | V | Jabbar | 987987987 | 25000 | 987654321 | 4 |

Example (9)

EMPD_5

| Fname | Minit | Lname | <u>Ssn</u> | Salary | Super_ssn | Dno |
|----------|-------|---------|------------|--------|-----------|-----|
| John | B | Smith | 123456789 | 30000 | 333445555 | 5 |
| Franklin | T | Wong | 333445555 | 40000 | 888665555 | 5 |
| Ramesh | K | Narayan | 666884444 | 38000 | 333445555 | 5 |
| Joyce | A | English | 453453453 | 25000 | 333445555 | 5 |

DEP_5

| Dname | <u>Dnumber</u> | Mgr_ssn | Mgr_start_date |
|----------|----------------|-----------|----------------|
| Research | 5 | 333445555 | 1988-05-22 |

DEP_5_LOCS

| <u>Dnumber</u> | <u>Location</u> |
|----------------|-----------------|
| 5 | Bellaire |
| 5 | Sugarland |
| 5 | Houston |

PROJS_5

| Pname | <u>Pnumber</u> | Plocation | Dnum |
|-----------|----------------|-----------|------|
| Product X | 1 | Bellaire | 5 |
| Product Y | 2 | Sugarland | 5 |
| Product Z | 3 | Houston | 5 |

Allocation of fragments to sites

- relation fragments at site 2 correspond to department 5
- EMPD_5 shows a mixed fragment which includes the EMPLOYEE tuples satisfying the condition Dno = 5
- the horizontal fragments of PROJECT, DEPARTMENT, and DEPT_LOCATIONS are similarly fragmented by department number 5

Example (10)

EMPD_4

| Fname | Minit | Lname | <u>Ssn</u> | Salary | Super_ssn | Dno |
|----------|-------|---------|------------|--------|-----------|-----|
| Alicia | J | Zelaya | 999887777 | 25000 | 987654321 | 4 |
| Jennifer | S | Wallace | 987654321 | 43000 | 888665555 | 4 |
| Ahmad | V | Jabbar | 987987987 | 25000 | 987654321 | 4 |

DEP_4

| Dname | <u>Dnumber</u> | Mgr_ssn | Mgr_start_date |
|----------------|----------------|-----------|----------------|
| Administration | 4 | 987654321 | 1995-01-01 |

DEP_4_LOCS

| <u>Dnumber</u> | <u>Location</u> |
|----------------|-----------------|
| 4 | Stafford |

PROJS_4

| Pname | <u>Pnumber</u> | Plocation | Dnum |
|-----------------|----------------|-----------|------|
| Computerization | 10 | Stafford | 4 |
| New_benefits | 30 | Stafford | 4 |

Allocation of fragments to sites

- relation fragments at site 3 correspond to department 4
- EMPD_4 shows a mixed fragment which includes the EMPLOYEE tuples satisfying the condition $Dno = 4$
- the horizontal fragments of PROJECT, DEPARTMENT, and DEPT_LOCATIONS are similarly fragmented by department number 4

Example (11)

WORKS_ON

| <u>Essn</u> | <u>Pno</u> | Hours |
|-------------|------------|-------|
| 123456789 | 1 | 32.5 |
| 123456789 | 2 | 7.5 |
| 666884444 | 3 | 40.0 |
| 453453453 | 1 | 20.0 |
| 453453453 | 2 | 20.0 |
| 333445555 | 2 | 10.0 |
| 333445555 | 3 | 10.0 |
| 333445555 | 10 | 10.0 |
| 333445555 | 20 | 10.0 |
| 999887777 | 30 | 30.0 |
| 999887777 | 10 | 10.0 |
| 987987987 | 10 | 35.0 |
| 987987987 | 30 | 5.0 |
| 987654321 | 30 | 20.0 |
| 987654321 | 20 | 15.0 |
| 888665555 | 20 | NULL |

We must now fragment the WORKS_ON relation and decide which fragments of WORKS_ON to store at sites 2 and 3.

Since each tuple in WORKS_ON relates an employee to a project, the problem arises that no attribute of WORKS_ON directly indicates the department to which each tuple belongs.

There may be projects, which are controlled by a different department than the related employee.

Hence, we should fragment WORKS_ON based on the department in which the employee works and then fragment further based on the department that controls the projects that employee is working on

Employees in Department 5

G1

| <u>Essn</u> | <u>Pno</u> | Hours |
|-------------|------------|-------|
| 123456789 | 1 | 32.5 |
| 123456789 | 2 | 7.5 |
| 666884444 | 3 | 40.0 |
| 453453453 | 1 | 20.0 |
| 453453453 | 2 | 20.0 |
| 333445555 | 2 | 10.0 |
| 333445555 | 3 | 10.0 |

C1 = C and (Pno in (SELECT Pnumber FROM PROJECT WHERE Dnum = 5))

G2

| <u>Essn</u> | <u>Pno</u> | Hours |
|-------------|------------|-------|
| 333445555 | 10 | 10.0 |

C2 = C and (Pno in (SELECT Pnumber FROM PROJECT WHERE Dnum = 4))

G3

| <u>Essn</u> | <u>Pno</u> | Hours |
|-------------|------------|-------|
| 333445555 | 20 | 10.0 |

C3 = C and (Pno in (SELECT Pnumber FROM PROJECT WHERE Dnum = 1))

Example (12)

PROJECT

| Pname | <u>Pnumber</u> | Plocation | Dnum |
|-----------------|----------------|-----------|------|
| ProductX | 1 | Bellaire | 5 |
| ProductY | 2 | Sugarland | 5 |
| ProductZ | 3 | Houston | 5 |
| Computerization | 10 | Stafford | 4 |
| Reorganization | 20 | Houston | 1 |
| Newbenefits | 30 | Stafford | 4 |

WORKS_ON

| <u>Essn</u> | <u>Pno</u> | Hours |
|-------------|------------|-------|
| 123456789 | 1 | 32.5 |
| 123456789 | 2 | 7.5 |
| 666884444 | 3 | 40.0 |
| 453453453 | 1 | 20.0 |
| 453453453 | 2 | 20.0 |
| 333445555 | 2 | 10.0 |
| 333445555 | 3 | 10.0 |
| 333445555 | 10 | 10.0 |
| 333445555 | 20 | 10.0 |
| 999887777 | 30 | 30.0 |
| 999887777 | 10 | 10.0 |
| 987987987 | 10 | 35.0 |
| 987987987 | 30 | 5.0 |
| 987654321 | 30 | 20.0 |
| 987654321 | 20 | 15.0 |
| 888665555 | 20 | NULL |

Fragments of WORKS_ON for employees working in department 5

C = Essn in (SELECT Ssn
FROM EMPLOYEE
WHERE Dno = 5)

The union of fragments G1, G2, and G3 gives all WORKS_ON tuples for employees who work for department 5.

Employees in Department 4

G4

| <u>Essn</u> | <u>Pno</u> | Hours |
|-------------|------------|-------|
|-------------|------------|-------|

C4 = C and (Pno in (SELECT Pnumber FROM PROJECT WHERE Dnum = 5))

G5

| <u>Essn</u> | <u>Pno</u> | Hours |
|-------------|------------|-------|
| 999887777 | 30 | 30.0 |
| 999887777 | 10 | 10.0 |
| 987987987 | 10 | 35.0 |
| 987987987 | 30 | 5.0 |
| 987654321 | 30 | 20.0 |

C5 = C and (Pno in (SELECT Pnumber FROM PROJECT WHERE Dnum = 4))

G6

| <u>Essn</u> | <u>Pno</u> | Hours |
|-------------|------------|-------|
| 987654321 | 20 | 15.0 |

C6 = C and (Pno in (SELECT Pnumber FROM PROJECT WHERE Dnum = 1))

Example (13)

PROJECT

| Pname | <u>Pnumber</u> | Plocation | Dnum |
|-----------------|----------------|-----------|------|
| ProductX | 1 | Bellaire | 5 |
| ProductY | 2 | Sugarland | 5 |
| ProductZ | 3 | Houston | 5 |
| Computerization | 10 | Stafford | 4 |
| Reorganization | 20 | Houston | 1 |
| Newbenefits | 30 | Stafford | 4 |

Fragments of WORKS_ON for employees working in department 4

C = Essn in (SELECT Ssn
FROM EMPLOYEE
WHERE Dno = 4)

The union of fragments G4, G5, and G6 gives all WORKS_ON tuples for employees who work for department 4.

WORKS_ON

| <u>Essn</u> | <u>Pno</u> | Hours |
|-------------|------------|-------|
| 123456789 | 1 | 32.5 |
| 123456789 | 2 | 7.5 |
| 666884444 | 3 | 40.0 |
| 453453453 | 1 | 20.0 |
| 453453453 | 2 | 20.0 |
| 333445555 | 2 | 10.0 |
| 333445555 | 3 | 10.0 |
| 333445555 | 10 | 10.0 |
| 333445555 | 20 | 10.0 |
| 999887777 | 30 | 30.0 |
| 999887777 | 10 | 10.0 |
| 987987987 | 10 | 35.0 |
| 987987987 | 30 | 5.0 |
| 987654321 | 30 | 20.0 |
| 987654321 | 20 | 15.0 |
| 888665555 | 20 | NULL |

Example (14)

Employees in Department 1

G7

| <u>Essn</u> | <u>Pno</u> | Hours |
|-------------|------------|-------|
|-------------|------------|-------|

C7 = C and (Pno in (SELECT Pnumber FROM PROJECT WHERE Dnum = 5))

G8

| <u>Essn</u> | <u>Pno</u> | Hours |
|-------------|------------|-------|
|-------------|------------|-------|

C8 = C and (Pno in (SELECT Pnumber FROM PROJECT WHERE Dnum = 4))

G9

| <u>Essn</u> | <u>Pno</u> | Hours |
|-------------|------------|-------|
| 888665555 | 20 | Null |

C9 = C and (Pno in (SELECT Pnumber FROM PROJECT WHERE Dnum = 1))

PROJECT

| Pname | <u>Pnumber</u> | Plocation | Dnum |
|-----------------|----------------|-----------|------|
| ProductX | 1 | Bellaire | 5 |
| ProductY | 2 | Sugarland | 5 |
| ProductZ | 3 | Houston | 5 |
| Computerization | 10 | Stafford | 4 |
| Reorganization | 20 | Houston | 1 |
| Newbenefits | 30 | Stafford | 4 |

Fragments of WORKS_ON for employees working in department 1

C = Essn in (SELECT Ssn
FROM EMPLOYEE
WHERE Dno = 1)

The union of fragments G7, G8, and G9 gives all WORKS_ON tuples for employees who work for department 1.

WORKS_ON

| <u>Essn</u> | <u>Pno</u> | Hours |
|-------------|------------|-------|
| 123456789 | 1 | 32.5 |
| 123456789 | 2 | 7.5 |
| 666884444 | 3 | 40.0 |
| 453453453 | 1 | 20.0 |
| 453453453 | 2 | 20.0 |
| 333445555 | 2 | 10.0 |
| 333445555 | 3 | 10.0 |
| 333445555 | 10 | 10.0 |
| 333445555 | 20 | 10.0 |
| 999887777 | 30 | 30.0 |
| 999887777 | 10 | 10.0 |
| 987987987 | 10 | 35.0 |
| 987987987 | 30 | 5.0 |
| 987654321 | 30 | 20.0 |
| 987654321 | 20 | 15.0 |
| 888665555 | 20 | NULL |

G1

| <u>Essn</u> | <u>Pno</u> | Hours |
|-------------|------------|-------|
| 123456789 | 1 | 32.5 |
| 123456789 | 2 | 7.5 |
| 666884444 | 3 | 40.0 |
| 453453453 | 1 | 20.0 |
| 453453453 | 2 | 20.0 |
| 333445555 | 2 | 10.0 |
| 333445555 | 3 | 10.0 |

C1 = C and (Pno in (SELECT Pnumber FROM PROJECT WHERE Dnum = 5))

G2

| <u>Essn</u> | <u>Pno</u> | Hours |
|-------------|------------|-------|
| 333445555 | 10 | 10.0 |

C2 = C and (Pno in (SELECT Pnumber FROM PROJECT WHERE Dnum = 4))

G3

| <u>Essn</u> | <u>Pno</u> | Hours |
|-------------|------------|-------|
| 333445555 | 20 | 10.0 |

C3 = C and (Pno in (SELECT Pnumber FROM PROJECT WHERE Dnum = 1))

Example (15)

The union of fragments G1, G4, and G7 gives all WORKS_ON tuples for projects controlled by department 5.

Additionally, we union fragments G2 and G3 at site 2 (WORKS_ON_5)

This allocation strategy permits the join between the local EMPLOYEE or PROJECT fragments at site 2 and the local WORKS_ON fragment to be performed completely locally.

G4

| <u>Essn</u> | <u>Pno</u> | Hours |
|-------------|------------|-------|
|-------------|------------|-------|

C4 = C and (Pno in (SELECT Pnumber FROM PROJECT WHERE Dnum = 5))

G7

| <u>Essn</u> | <u>Pno</u> | Hours |
|-------------|------------|-------|
|-------------|------------|-------|

C7 = C and (Pno in (SELECT Pnumber FROM PROJECT WHERE Dnum = 5))

WORKS_ON_5

| <u>Essn</u> | <u>Pno</u> | Hours |
|-------------|------------|-------|
| 123456789 | 1 | 32.5 |
| 123456789 | 2 | 7.5 |
| 666884444 | 3 | 40.0 |
| 453453453 | 1 | 20.0 |
| 453453453 | 2 | 20.0 |
| 333445555 | 2 | 10.0 |
| 333445555 | 3 | 10.0 |
| 333445555 | 10 | 10.0 |
| 333445555 | 20 | 10.0 |

G4

| <u>Essn</u> | <u>Pno</u> | Hours |
|-------------|------------|-------|
|-------------|------------|-------|

C4 = C and (Pno in (SELECT
Pnumber FROM PROJECT
WHERE Dnum = 5))

G5

| <u>Essn</u> | <u>Pno</u> | Hours |
|-------------|------------|-------|
| 999887777 | 30 | 30.0 |
| 999887777 | 10 | 10.0 |
| 987987987 | 10 | 35.0 |
| 987987987 | 30 | 5.0 |
| 987654321 | 30 | 20.0 |

C5 = C and (Pno in (SELECT
Pnumber FROM PROJECT
WHERE Dnum = 4))

G6

| <u>Essn</u> | <u>Pno</u> | Hours |
|-------------|------------|-------|
| 987654321 | 20 | 15.0 |

C6 = C and (Pno in (SELECT
Pnumber FROM PROJECT
WHERE Dnum = 1))

Example (16)

The union of fragments G2, G5, and G8 gives all
WORKS_ON tuples for projects controlled by
department 4.

Additionally, we union fragments G4, and G6 at site
3 (WORKS_ON_4)

This allocation strategy permits the join between the
local EMPLOYEE or PROJECT fragments at site 3
and the local WORKS_ON fragment to be
performed completely locally.

G2

| <u>Essn</u> | <u>Pno</u> | Hours |
|-------------|------------|-------|
| 333445555 | 10 | 10.0 |

C2 = C and (Pno in (SELECT
Pnumber FROM PROJECT
WHERE Dnum = 4))

G8

| <u>Essn</u> | <u>Pno</u> | Hours |
|-------------|------------|-------|
|-------------|------------|-------|

C8 = C and (Pno in (SELECT
Pnumber FROM PROJECT
WHERE Dnum = 4))

WORKS_ON_4

| <u>Essn</u> | <u>Pno</u> | Hours |
|-------------|------------|-------|
| 333445555 | 10 | 10.0 |
| 999887777 | 30 | 30.0 |
| 999887777 | 10 | 10.0 |
| 987987987 | 10 | 35.0 |
| 987987987 | 30 | 5.0 |
| 987654321 | 30 | 20.0 |
| 987654321 | 20 | 15.0 |