Distributed database design

Design Strategy and Issue

- **In general setting**

  Making decision about the placement of data and programs across the sites of the network as well as designing network itself.

- **In DDBMS**

  - Placement of distributed DBMS software

  - Placement of applications that run on a database.

Distribution Design

Two approaches

1. **Top down approach**

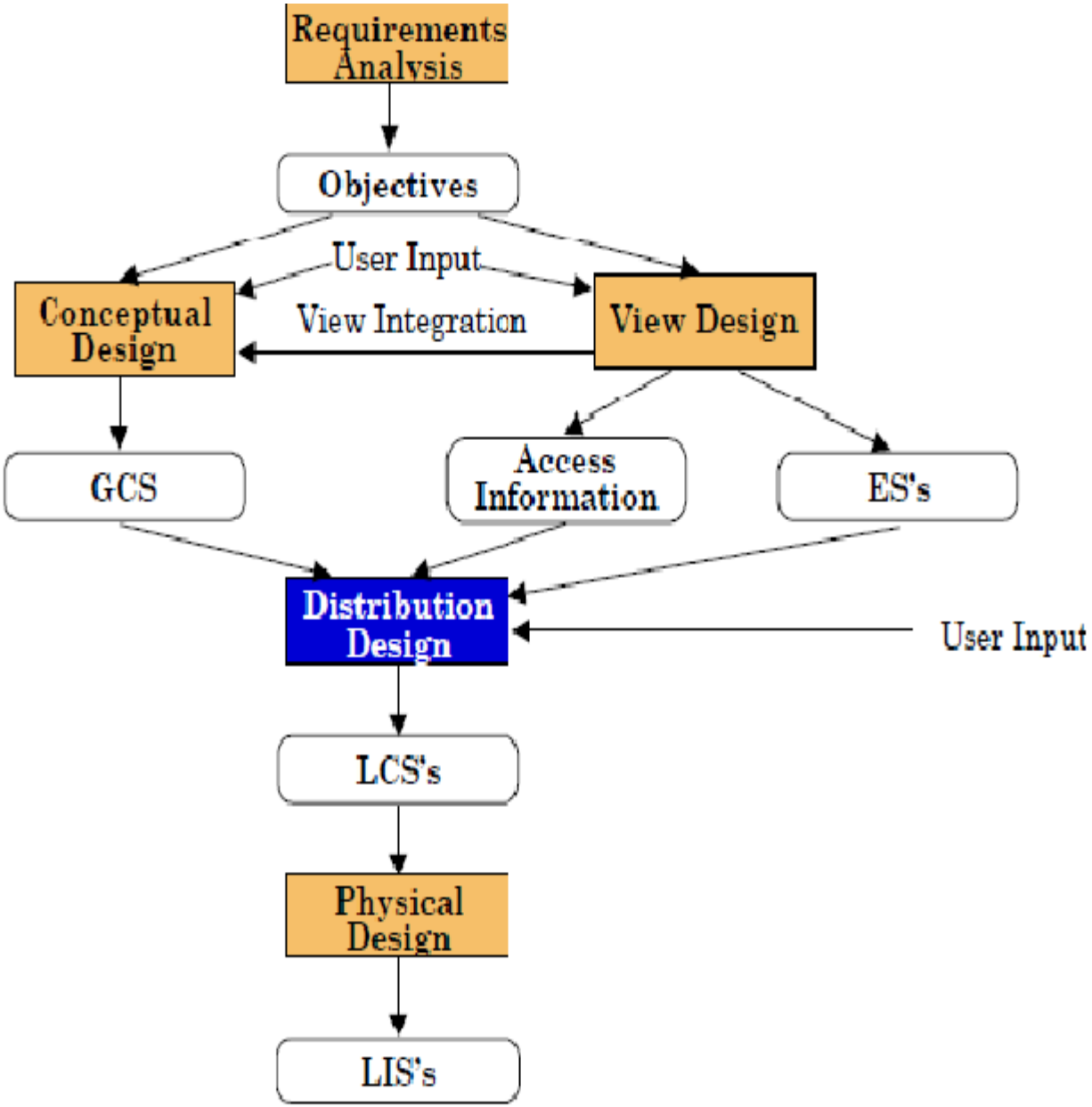
- Mostly while designing system from scratch.
- In homogeneous system.

2. **Bottom up approach**

   When the database exists already at number of sites.

Distribution Design

Top down approach



Distribution Design Issues

1. Why to fragment?
2. How to fragment?
3. How much to fragment?
4. How to test correctness?
5. How to allocate?
6. Information requirements

Data Replication

Database replication is the frequent electronic copying data from a database in one computer or server to a database in another so that all users share the same level of information. The result is a distributed database in which users can access data relevant to their tasks without interfering with the work of others. The implementation of database replication for the purpose of eliminating data ambiguity or inconsistency among users is known as normalization.

Data Replication

- *Replication* is the process of copying and maintaining database objects in multiple databases that make up a distributed database system.
- Changes applied at one site are captured and stored locally before being forwarded and applied at each of the remote locations.
- Replication provides user with fast, local access to shared data, and protects availability of applications because alternate data access options exist.
- Even if one site becomes unavailable, users can continue to query or even update the remaining locations.

Replication Objects,groups and sites

Replication object:A *replication object* is a database object existing on multiple servers in a distributed database system.

Replication groups: you create and use a replication group to organize the schema objects necessary to support a particular database application.

Replication Sites: A replication group can exist at multiple *replication sites*. Replication environments support two basic types of sites: **master sites** and **snapshot sites**.

Data Fragmentation

Fragmentation involves subsetting (splitting) tables and storing the parts of the tables on different nodes.

Types of fragmentation

1. **Horizontal Fragmentation**
2. **Vertical Fragmentation**
3. **Hybrid(Mixed) Fragmentation.**

Horizontal Fragmentation

$PROJ_1$: projects with budgets less than \$200,000

$PROJ_2$: projects with budgets greater than or equal to \$200,000

PROJ

PNO	PNAME	BUDGET	LOC
P1	Instrumentation	150000	Montreal
P2	Database Develop.	135000	New York
P3	CAD/CAM	250000	New York
P4	Maintenance	310000	Paris
P5	CAD/CAM	500000	Boston

$PROJ_1$

PNO	PNAME	BUDGET	LOC
P1	Instrumentation	150000	Montreal
P2	Database Develop.	135000	New York

$PROJ_2$

PNO	PNAME	BUDGET	LOC
P3	CAD/CAM	250000	New York
P4	Maintenance	310000	Paris
P5	CAD/CAM	500000	Boston

Horizontal Fragmentation

- Horizontal Fragmentation are subsets of tuples (rows).
- In the Horizontal Fragmentation Table remain same as previous.Only the rows are splitted.
- vertical Fragmentation splits table column wise.

Fragment 1

customer_id | Name | Area | Payment Type | Sex 1

1 | Bob | London | Credit card | Male 2

2 | Mike | Manchester | Cash | Male

Fragment 2

customer_id | Name | Area | Payment Type | Sex 3

1 | Ruby | London | Cash | Female

Vertical Fragmentation

$PROJ_1$: information about project budgets

$PROJ_2$: information about project names and locations

PROJ

PNO	PNAME	BUDGET	LOC
P1	Instrumentation	150000	Montreal
P2	Database Develop.	135000	New York
P3	CAD/CAM	250000	New York
P4	Maintenance	310000	Paris
P5	CAD/CAM	500000	Boston

$PROJ_1$

PNO	BUDGET
P1	150000
P2	135000
P3	250000
P4	310000
P5	500000

$PROJ_2$

PNO	PNAME	LOC
P1	Instrumentation	Montreal
P2	Database Develop.	New York
P3	CAD/CAM	New York
P4	Maintenance	Paris
P5	CAD/CAM	Boston

Vertical Fragmentation

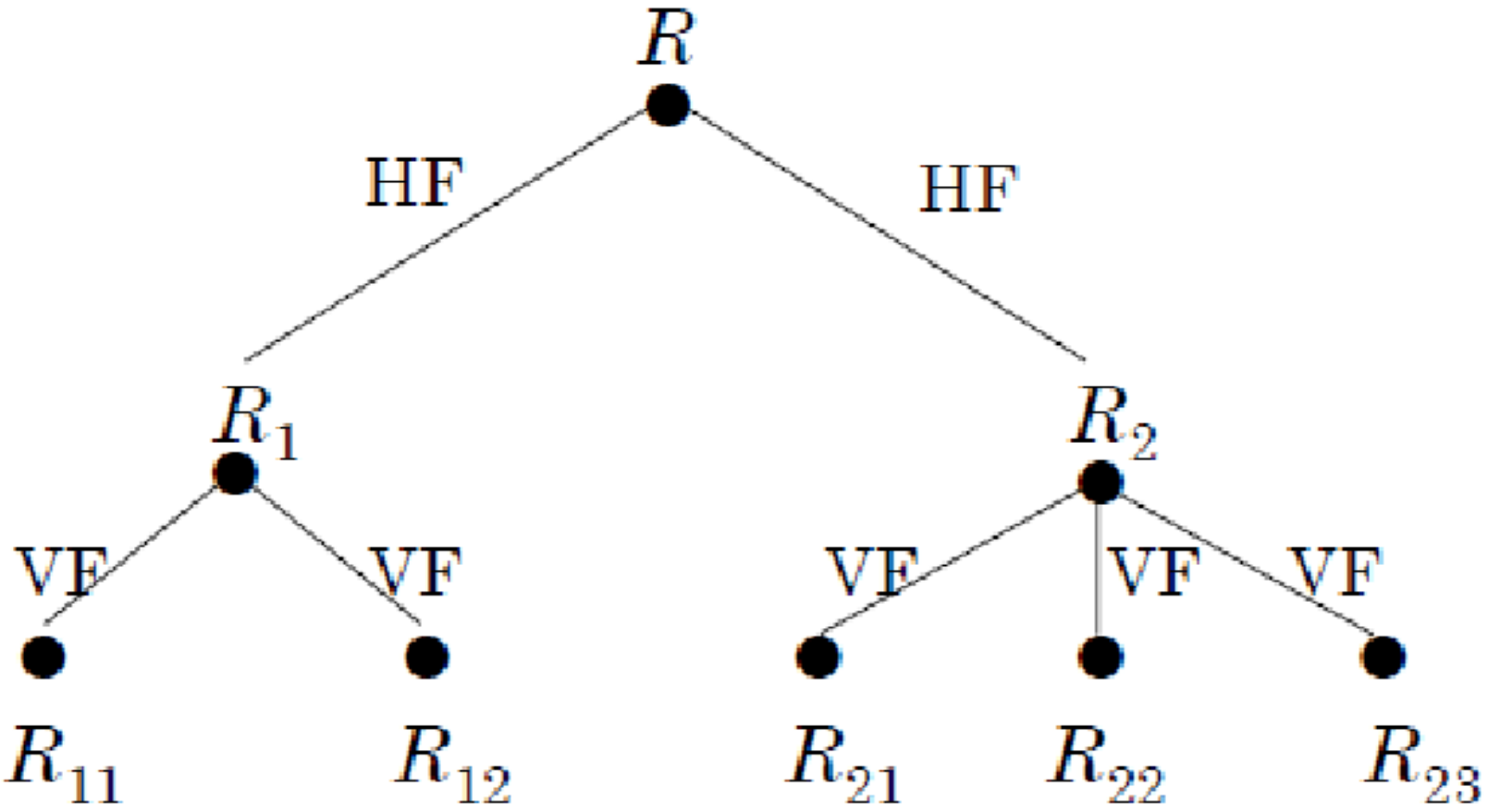
- Vertical fragmentation are subset of attributes.
- vertical Fragmentation splits table column wise.
- In Vertical Fragmentation one table splits into two or more tables.
- Fragment 1

customer_id | Name | Area | Sex
1 | Bob | London | Male
2 | Mike | Manchester | Male
3 | Ruby | London Female

- Fragment 2

customer_id | Payment Type
1 | Credit card
2 | Cash
3 | Cash

Hybrid(Mixed) Fragmentation



Hybrid(Mixed) Fragmentation

Mixed fragmentation is a process of simultaneously applying the horizontal and vertical fragmentation on a relation. It can be achieved in one of two ways: by performing horizontal fragmentation followed by vertical fragmentation.

Resource Allocation

Allocating resource(DBMS) depends on

1. Problem(resource) statement

   Given $F=\{F_1,F_2,F_3....F_n\}$ fragments

   $S=\{S_1,S_2,S_3...S_n\}$ no. of sites

   $Q=\{q_1,q_2,q_3...q_n\}$ no. of applications

- Resource allocation is distribution of fragments(f) into sites(S).
- Distribution should be optimal.

Minimal cost

Performance and

Constraints

Semantic Data Control

Semantic data control involves

1. View management
2. Security Control
3. Integrity Control

Objective :

- Ensures that authorized users perform correct operations on the database, contributing to the maintenance of the database integrity.

Semantic Data Control

1. View management

- Views might be derived from fragments.
- View definition storage should be treated as database storage.
- Query modification results in a distributed query.
- View evaluations might be costly if base relations are distributed.

Semantic Data Control

2. Security Control

i. Data protection

a.) Prevent the physical content of data to be understood by unauthorized users

b.) Encryption/decryption

- Data Encryption Standard
- Public-key encryption

ii. Authorization control

Only authorized users perform operations they are allowed to do on the database.

Semantic Data Control

3. Integrity Conrol

- Maintain database consistency by enforcing a set of constraints defined on the database.

a.) Structural constraints

- basic semantic properties inherent to a data model

e.g. unique key constraint in relational model.

b.) Behavioral constraints

- regulate application behavior

e.g., dependencies in the relational model(Fkey)

c.)Two components

- Integrity constraint specification
- Integrity constraint enforcement