Chapter 24: Distributed Databases
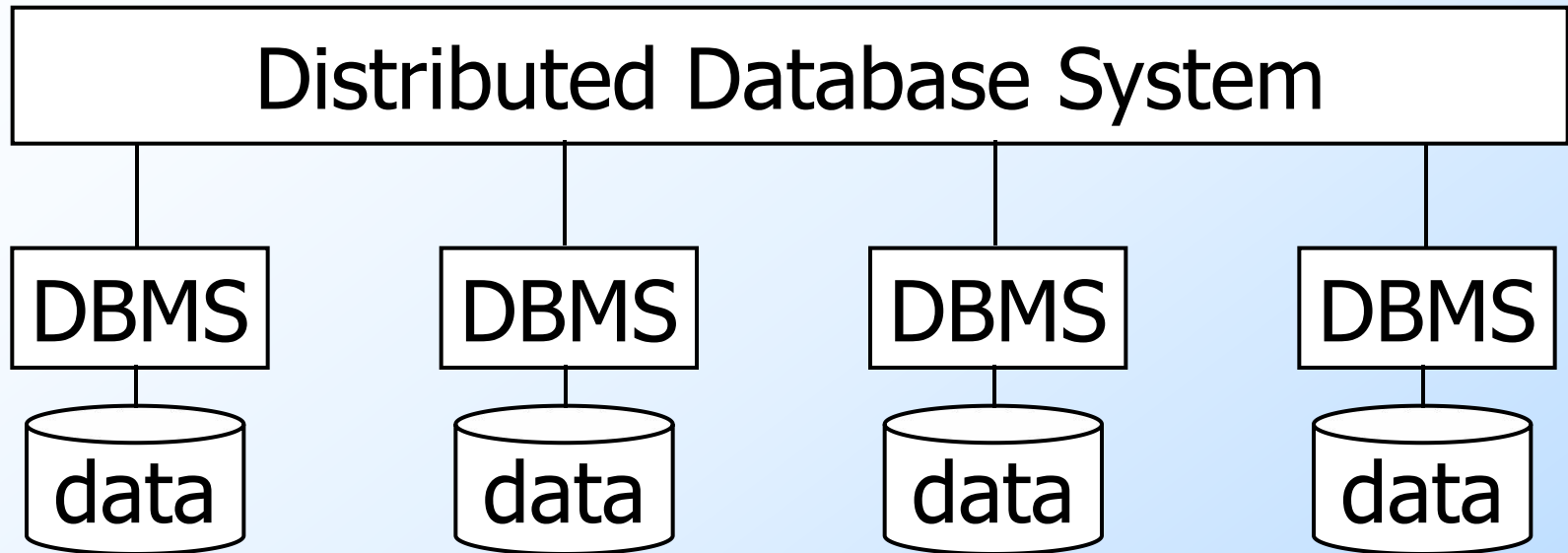
Distributed Databases

Advantages of a DDBS

◆ Modularity

◆ Fault Tolerance

◆ High Performance

◆ Data Sharing

◆ Low Cost Components

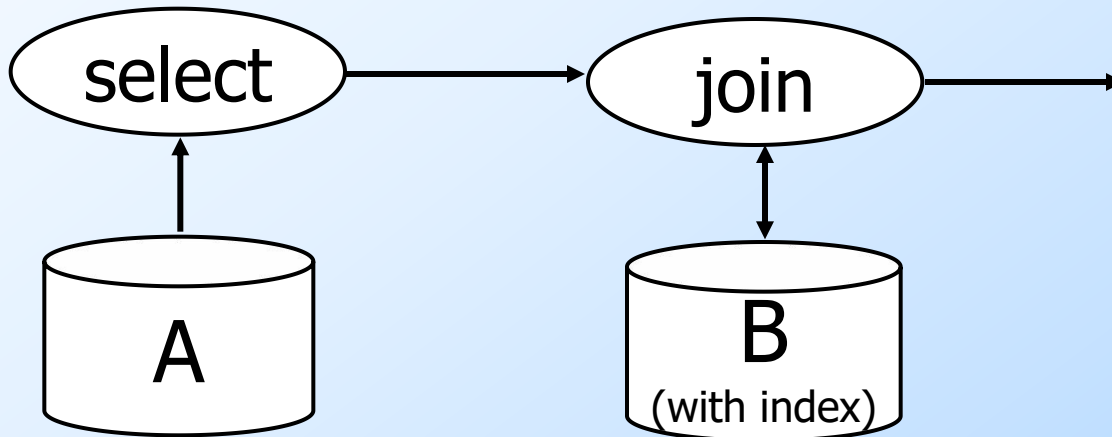# Issues

◆ Data Distribution

◆ Exploiting Parallelism

◆ Concurrency and Recovery

◆ Heterogeneity

Parallelism: Pipelining
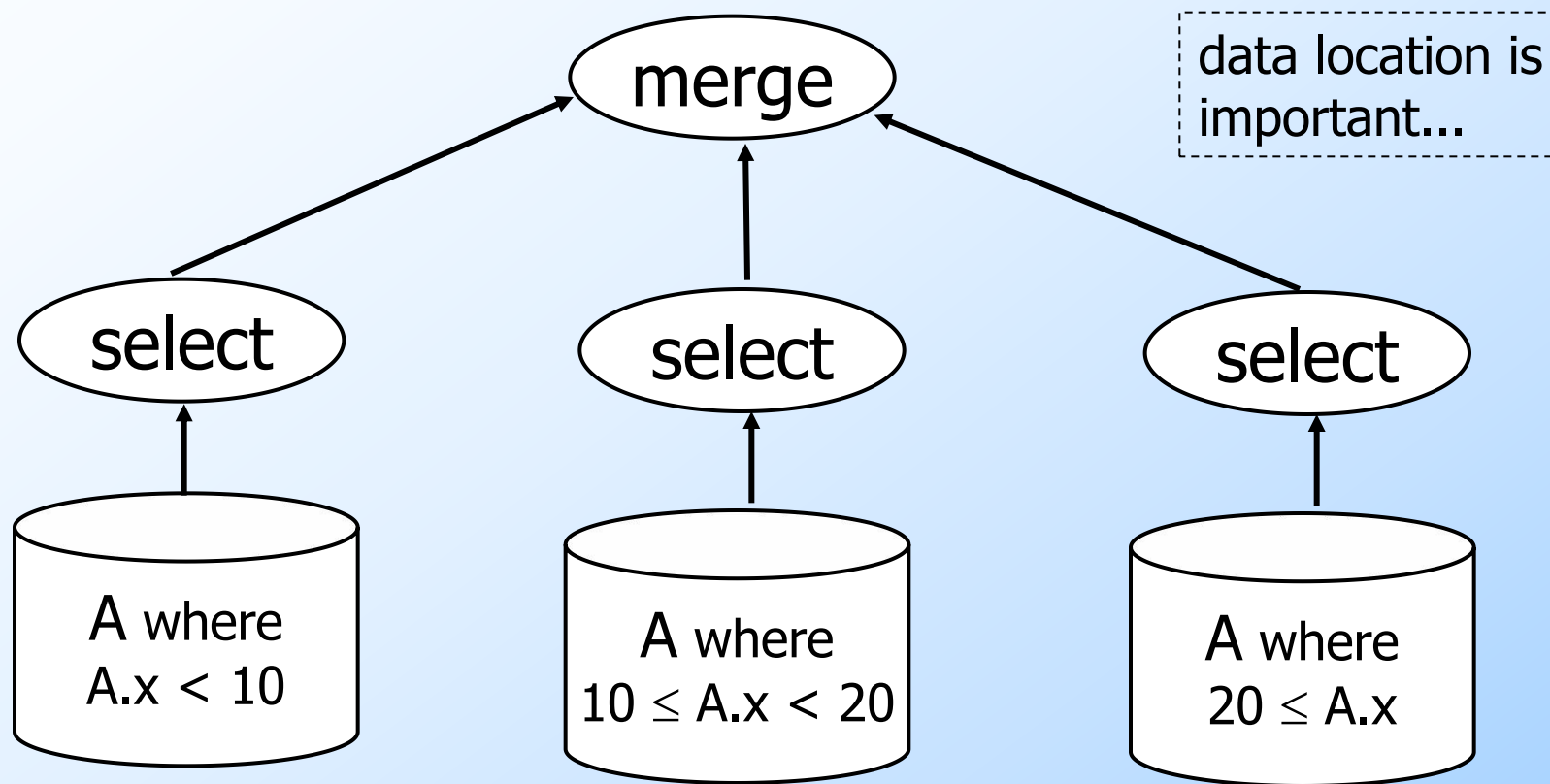
◆ Example:
- $T_1 \leftarrow$ SELECT *
  FROM A WHERE cond
- $T_2 \leftarrow$ JOIN T_1 and B

Parallelism: Concurrent Operations
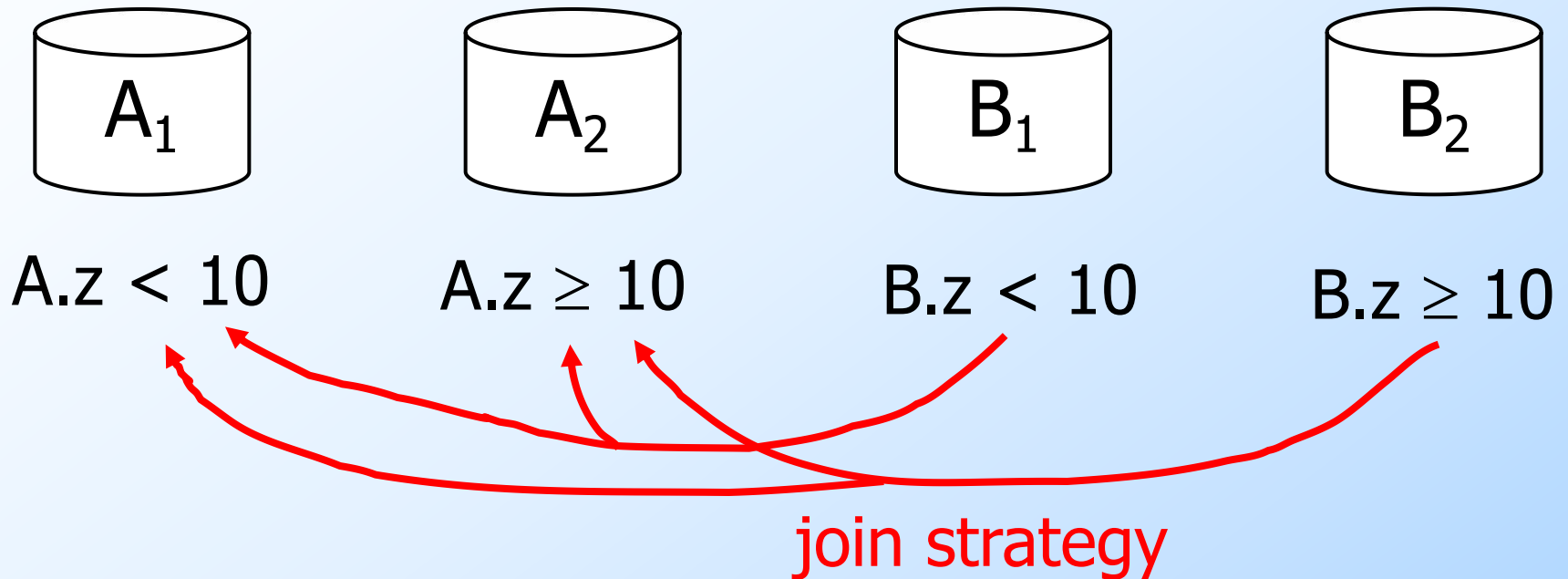
◆Example: SELECT * FROM A WHERE cond

```
                        merge
```

data location is
important...

```
      select          select          select
```

A where
A.x < 10

A where
$10 \leq A.x < 20$

A where
$20 \leq A.x$

Join Processing

◆ Example: JOIN A, B over attribute X

A_1 A_2 B_1 B_2

$A.x < 10$ $A.x \geq 10$ $B.x < 10$ $B.x \geq 10$

join strategy

Join Processing

◆ Example: JOIN A, B over attribute X

A_1 A_2 B_1 B_2

$A.z < 10$ $A.z \geq 10$ $B.z < 10$ $B.z \geq 10$

join strategy
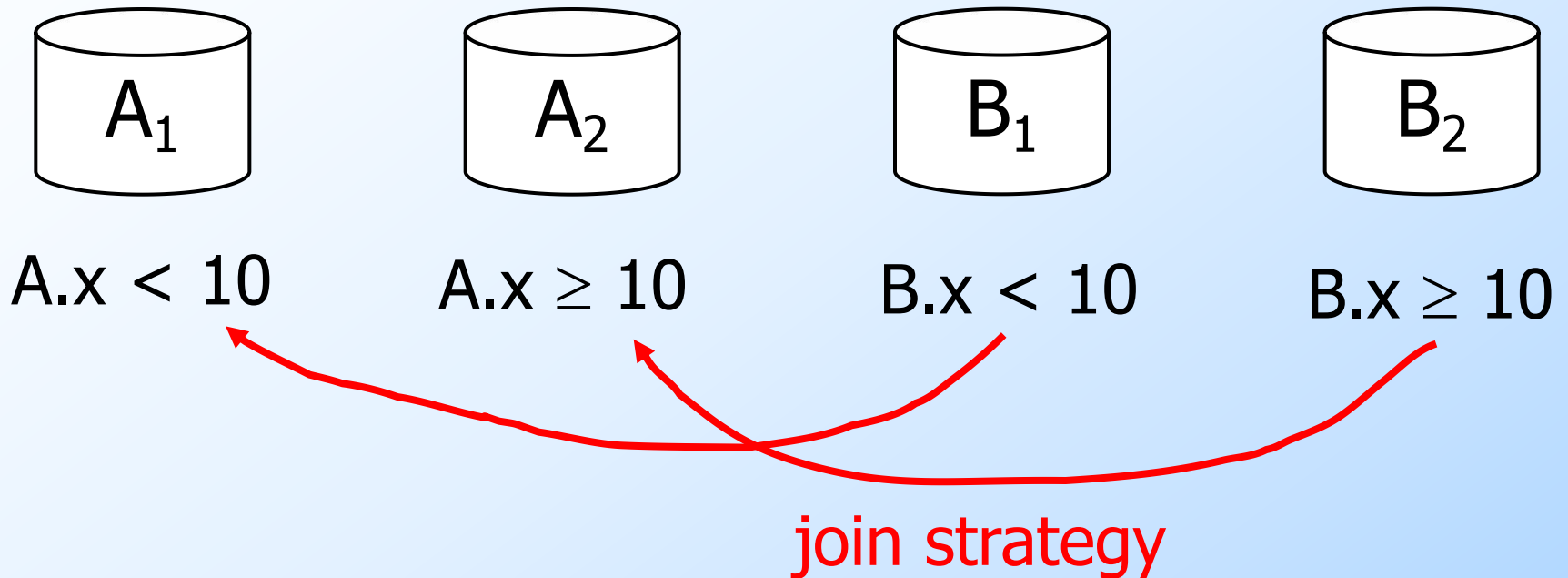
Concurrency & Recovery

◆Two Phase Commit
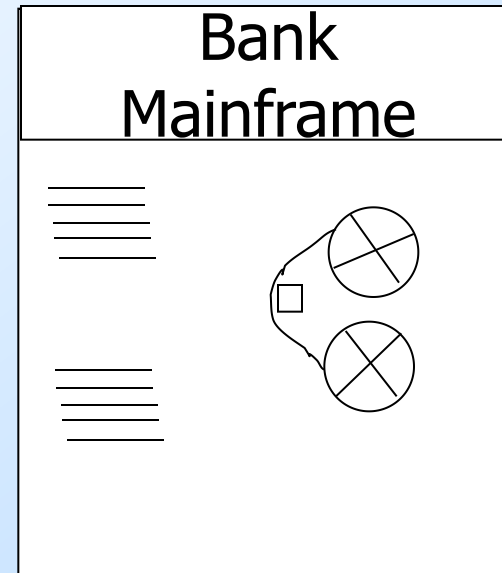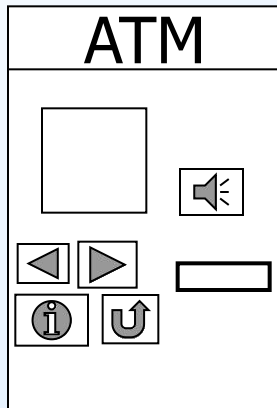
2PC: ATM Withdrawl

◆ Mainframe is coordinator

◆ Phase 1: ATM checks if money available; mainframe checks if account has funds (money and funds are "reserved")

◆ Phase 2: ATM releases funds; mainframe debits account

Replicated Data Mangement

◆Key to fault-tolerance, durability

◆Illustrates transaction processing issues

◆Various concurrency control/recovery algorithms available

Primary Copy Algorithm

◆ Updates run at primary site

◆ Backups repeat writes;
backups allow "out-of-date" reads

Primary Site	
A	5 3
B	9 8
C	7 6 4
D	25

Backup Site 1	
A	5 3
B	9 8
C	7 6 4
D	25

Backup Site 2	
A	5 3
B	8
C	6 4
D	25

T1: A:5; C:6
T2: B:9; C: 7

propagate in order