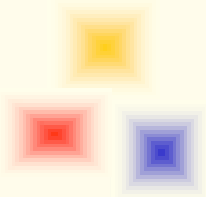
Database Management Systems and Their Implementation

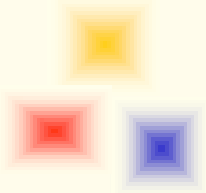


Course Goal and its Preliminary Courses

-
-
-

build the foundation of further research in database field

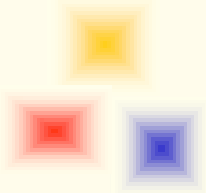
use database system better



Main Contents

Main Contents

relational distributed database management system



Course History

-
-
-

References

Stefano Ceri, –Distributed Databases‖

Wang Nengbin, –Principles of Database Systems‖

Raghu Ramakrishnan, Johannes Gehrke, –Database Management Systems‖ , 3rd Edition, McGraw

Molina, Jeffrey.D.Ullman, –Database Systems: the Complete Book‖

S.Bing Yao et al, –Query Optimization in DDBS‖

Table of Contents

Table of Contents

1. Introduction

1.1 The History of Database Technology and its Classification

- before 1960‘
-
-
  -
  -
  -
  -

Database Landscape M

1.2 Distributed Database Systems

Features of DDBS :

-
-
-

The advantages of DDBS:

-
-
-
-
-
-

The disadvantages of DDBS:

-
-

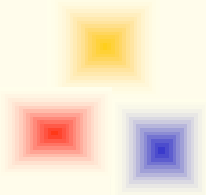
The main problems in DDBS:

-
-
-

Another problem specially for DDBS:

-

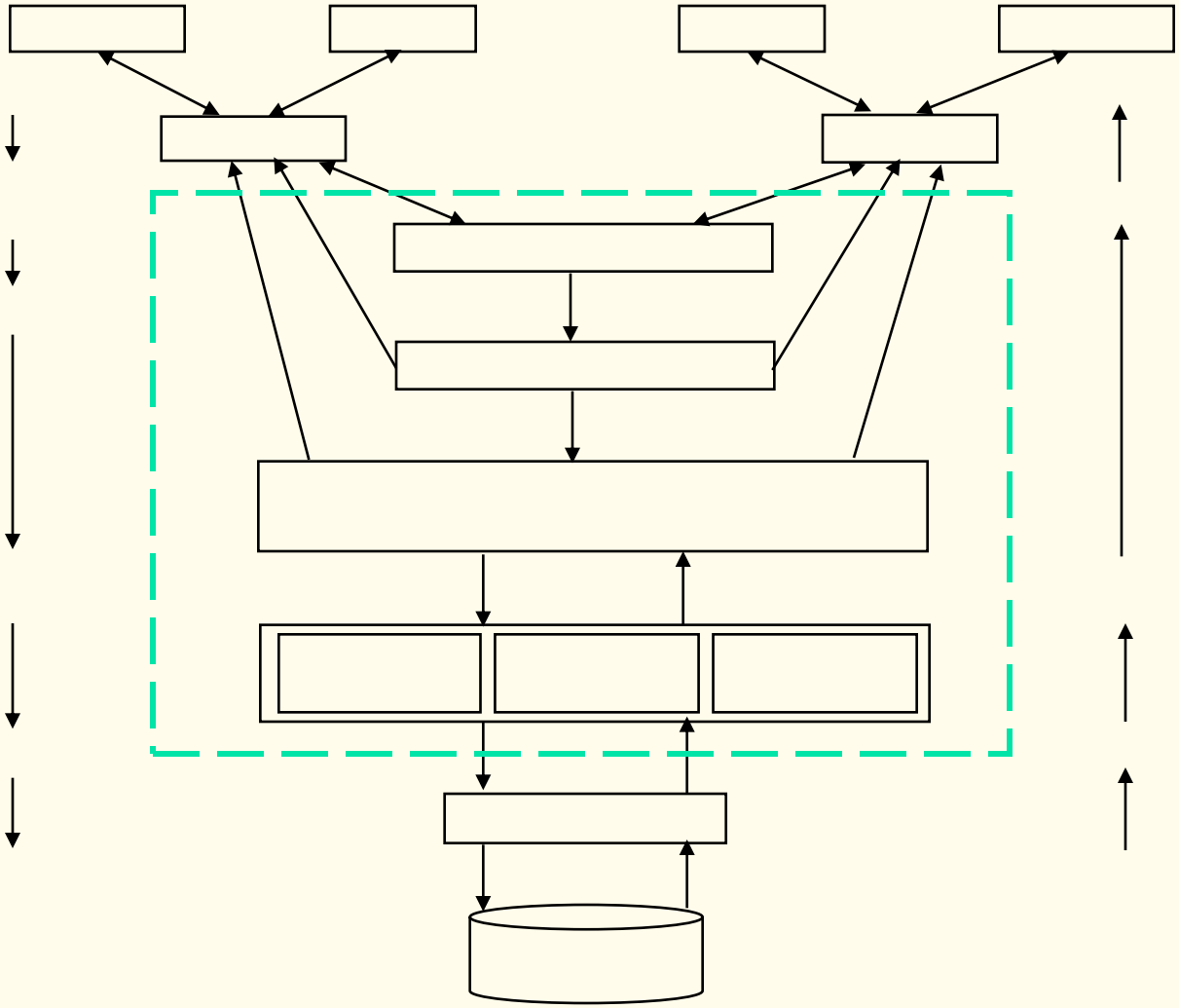
2. The Architecture of DBMS



Main Contains

-
-
-

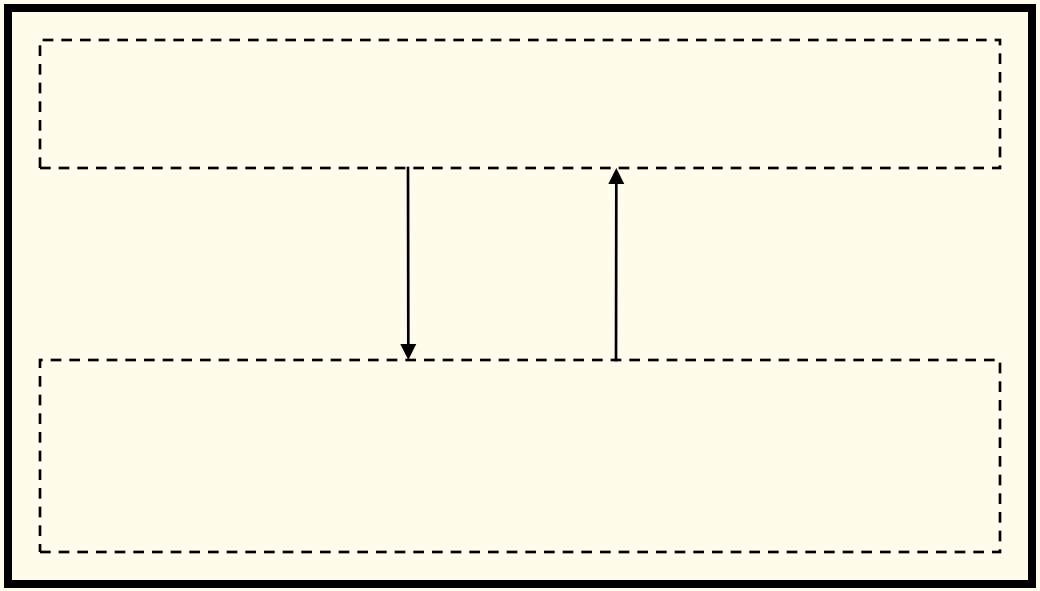
2.1 The Components of DBMS Core



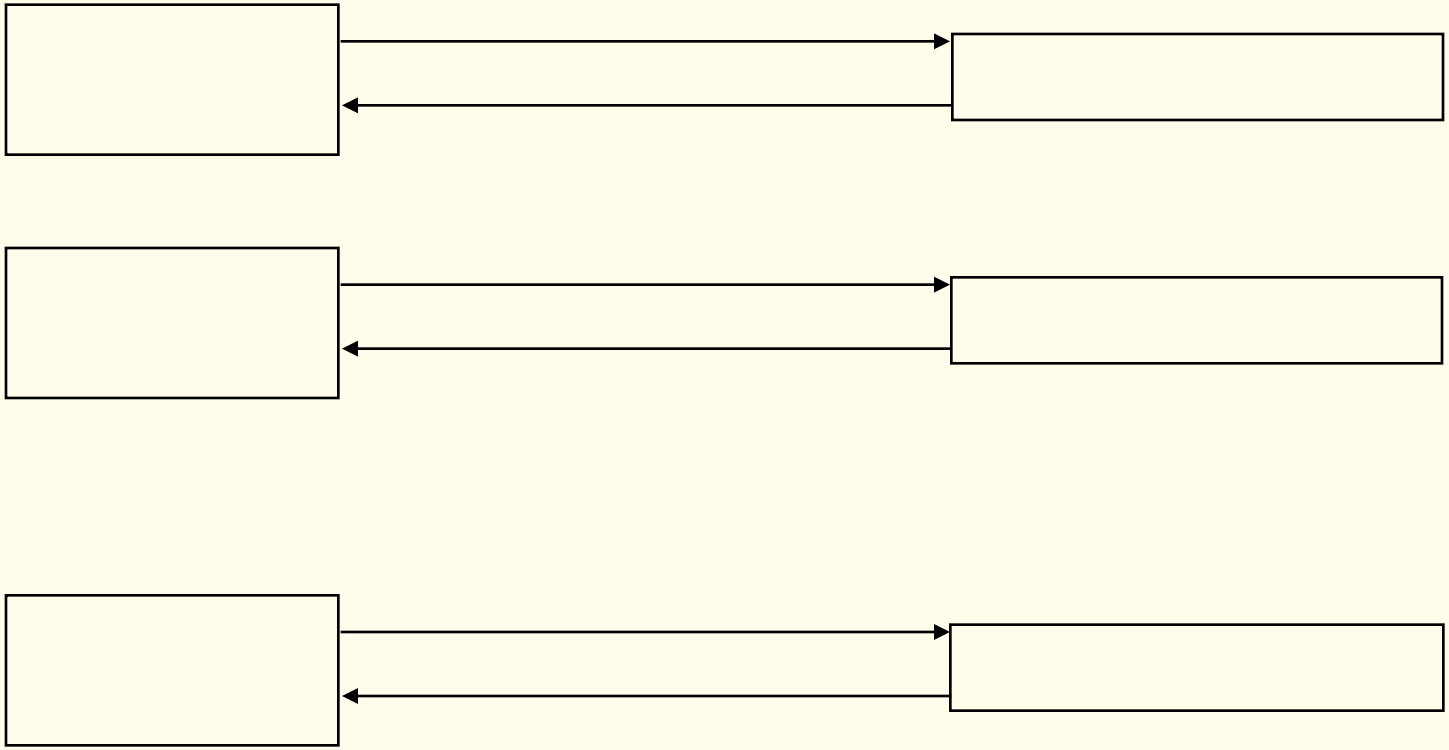
2.2 The Process Structure of DBMS

-
-
-
-

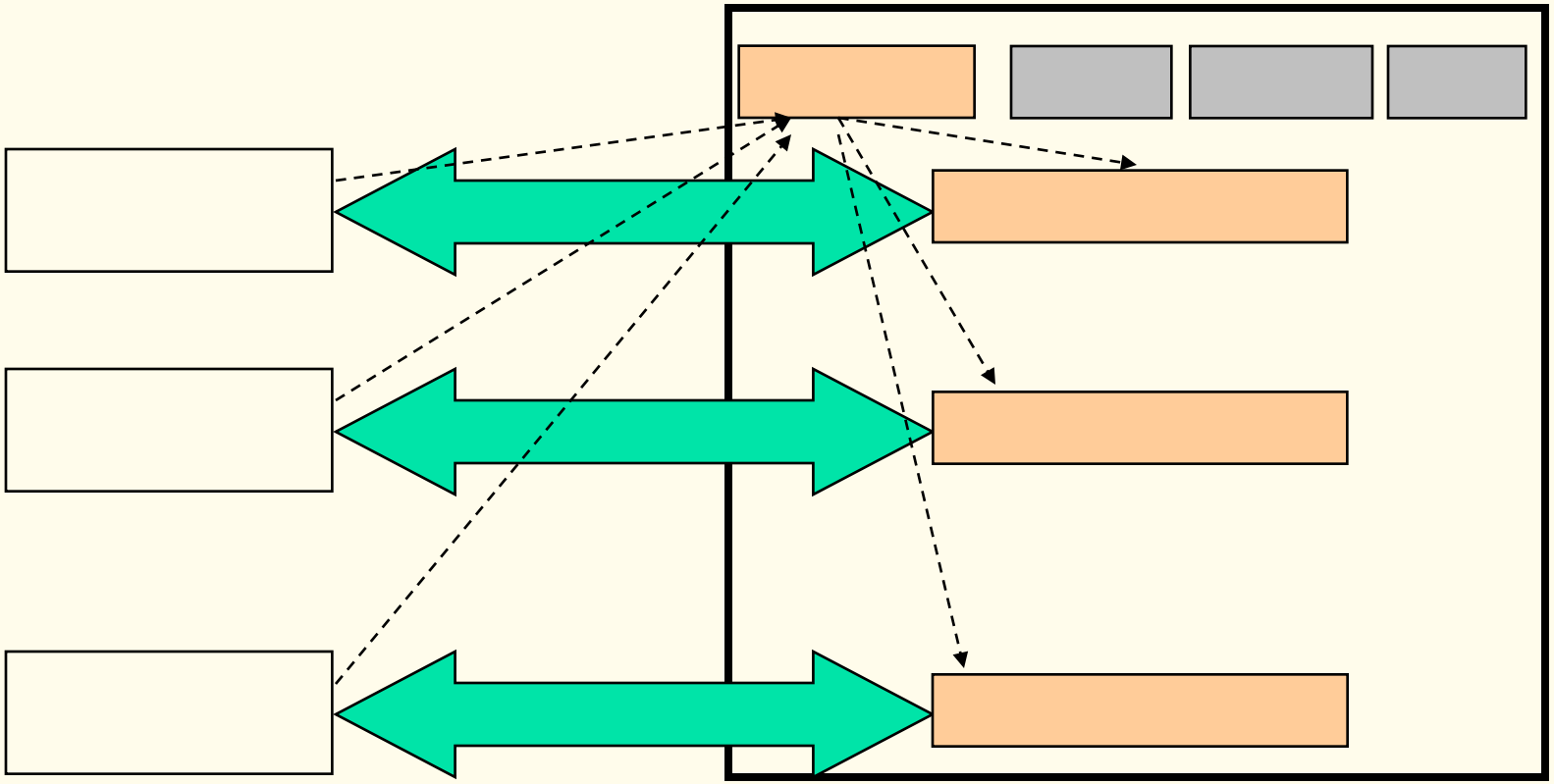
Single process structure



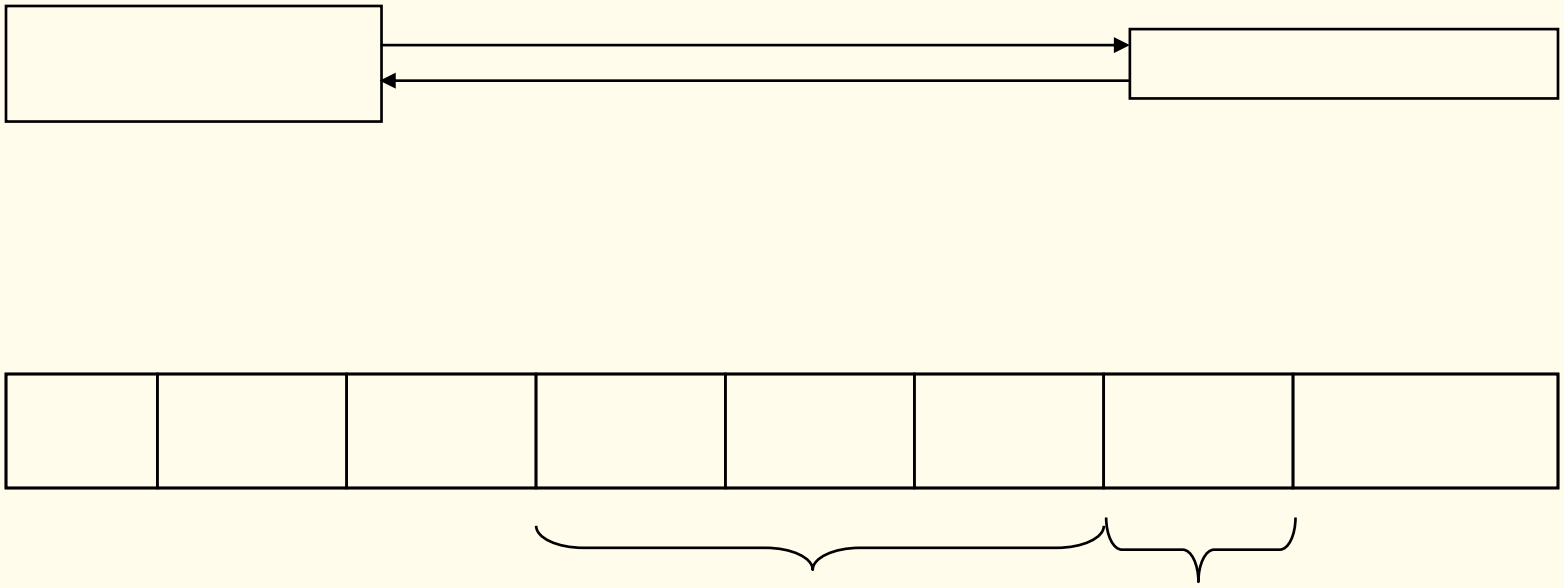
Multi processes structure



Multi threads structure



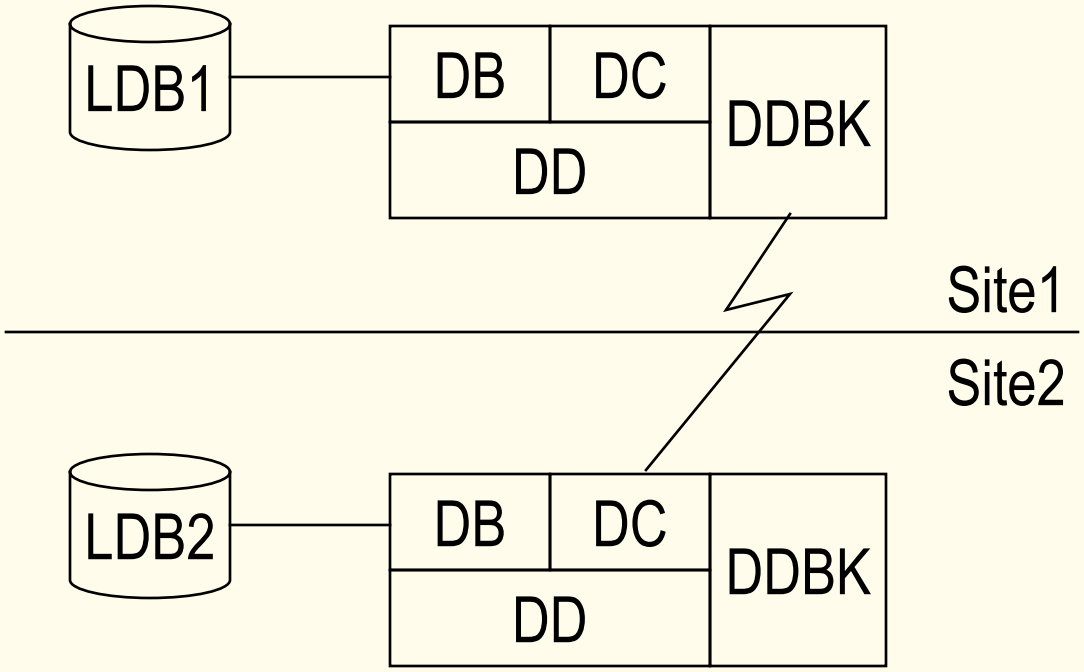
Communication protocols between processes / threads



Communication protocols between processes / threads

-
-
-
-
-
-
-

2.3 The Components of DDBMS Core



DB: database management

DC: communication control

DD: catalog management

DDBK: core, responsible for parsing, distributed transaction management, concurrency control, recovery and global query optimization.

An example of global query optimization

R1

Site1

R2

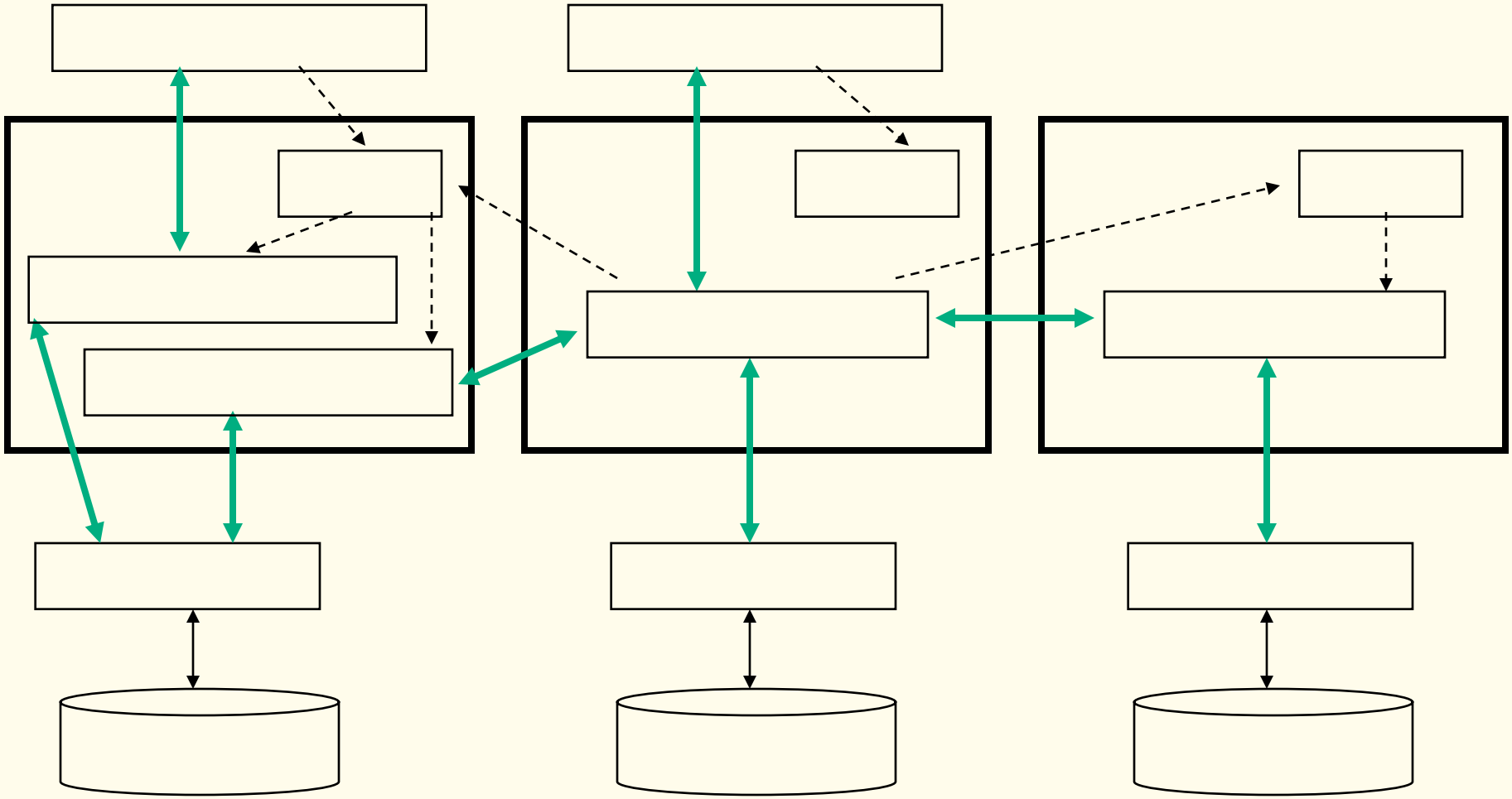
Site2

to site1, R‘

From R1, R‘

Where R1.a = R‘.b;

2.4 The Process Structure of DDBMS



3. Database Access Management

Main Contains

-
-
-
-

Access Types

-
-
-
-
-

File Organization

-
-
  hash function according to some attribute‘s value.
-
-
-

Index Technique

- √ √
- √
-
-
-
-
-

Bitmap index index itself is data

Date	Store	State	Class	Sales
3/1/96	32	NY	A	6
3/1/96	36	MA	A	9
3/1/96	38	NY	B	5
3/1/96	41	CT	A	11
3/1/96	43	NY	A	9
3/1/96	46	RI	B	3
3/1/96	47	CT	B	7
3/1/96	49	NY	A	12

Bitmap Index for Sales

8bit	4bit	2bit	1bit
0	1	1	0
1	0	0	1
0	1	0	1
1	0	1	1
1	0	0	1
0	0	1	1
0	1	1	1
1	1	0	0

Bitmap Index for State

0	0	0	0	0	0	1	0
0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0
0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	1
0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0

for Class

A	B	C
1	0	0
1	0	0
0	1	0
1	0	0
1	0	0
0	1	0
0	1	0
1	0	0

Access Primitives (examples)

- **Function**
- **Function**
- **Function** *rid*
- **Function** *tname*
- **Function** *rid* *rid*
- **Function**

Access Primitives (examples)

- **Function** *offset* *rid*
- **Function** *offset* *rid* *newtuple*
- **Function** *rid* *buf*
- **Function** *rid* *iid*

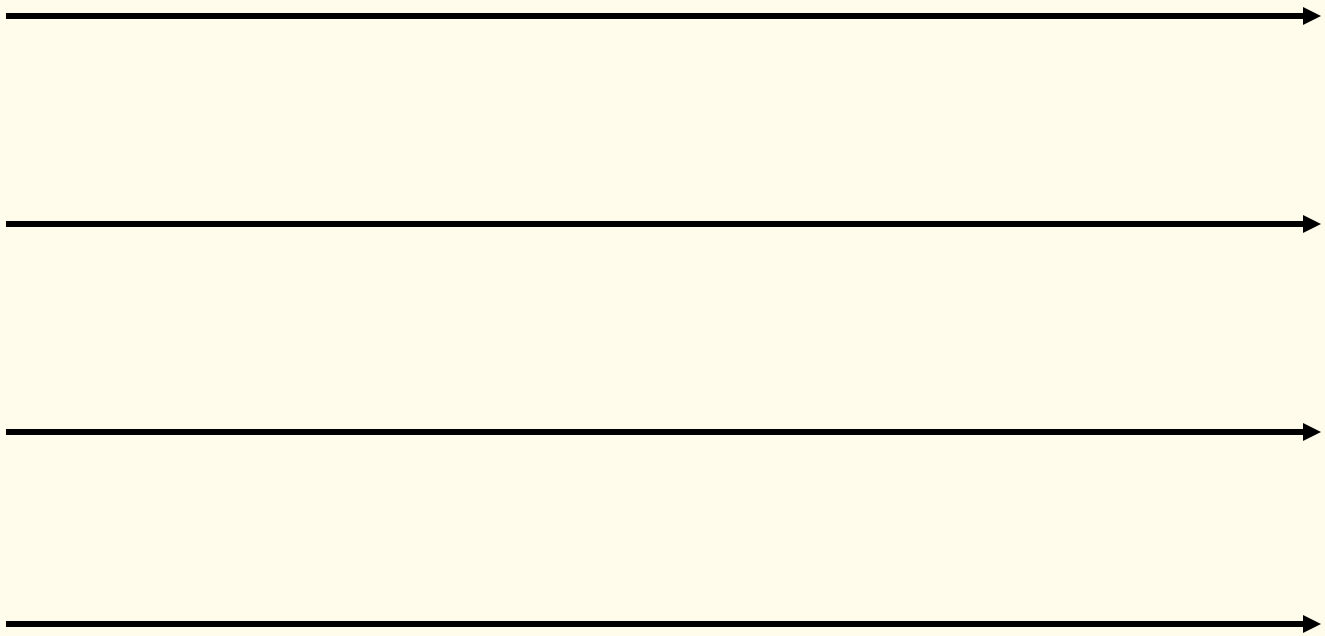
Access Primitives (examples)

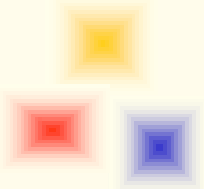
- **Function** *iid*
- **Function** *offset*
  rid *buf*
- **Function**
  attribute has the –*flag*ll relation with *pvalue*
  offsetbuf iid
- **Function**

4. Data Distribution

4.1 Strategies of Data Distribution

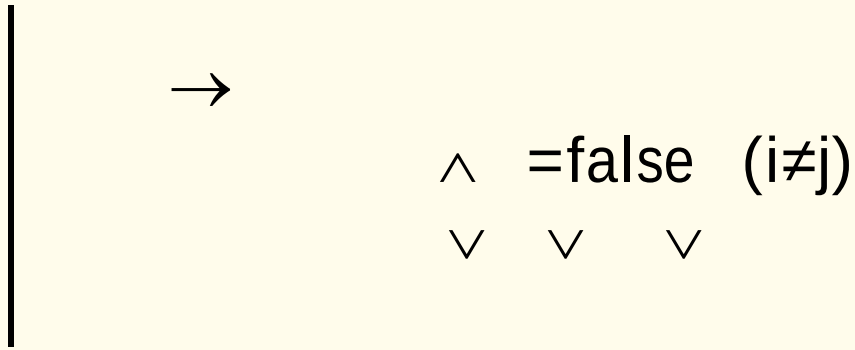
Comparison of four strategies





The criteria of fragmentation:

Fragmentation Methods



according to itself‘s attribute, but to another relation‘s

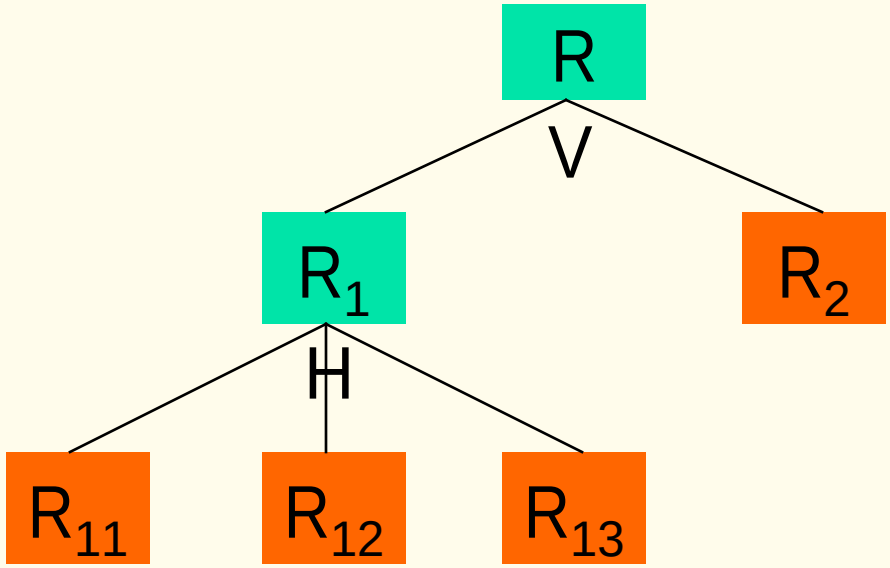
An example of Derived Fragmentation

derived from TEACHER‘s fragmentation.

(2) Vertical Fragmentation

-
-

(3) Mixed Fragmentation



4.3 Different Transparency Level

–information hiding‖ method

- don‘t have to know if they are fragmentized

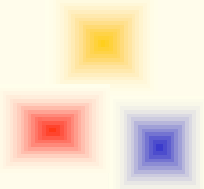
- fragmentized, but he don‘t have to worry the
- but he don‘t have to worry every local
-

4.4 Problems Caused by Data Distribution

Multi copies‘ consistency

Mainly the change of tuples‘ store location

→ → →



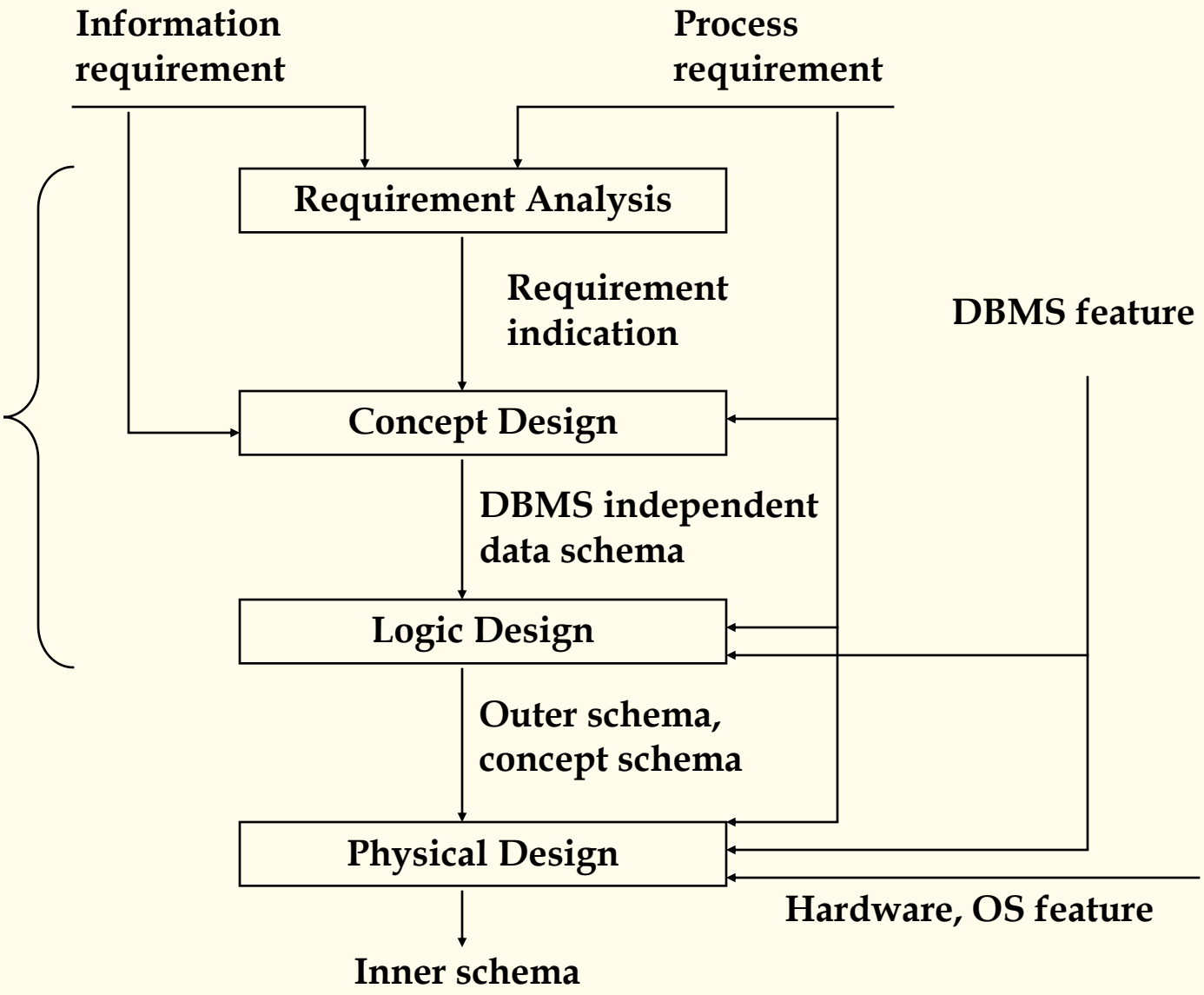
4.5 Distributed Database Design

-
-

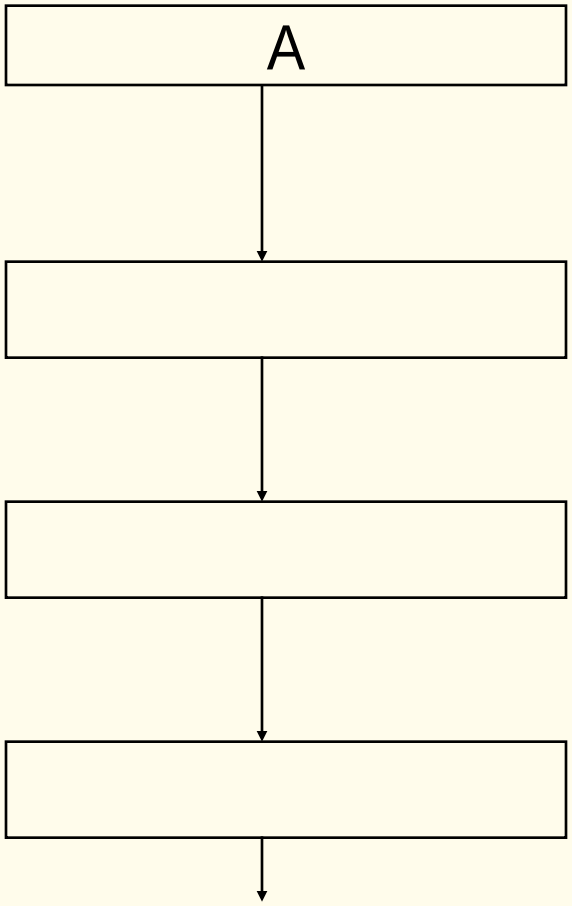
should ask the following questions to every application while requirement analysis:

-
-
-

Design flow of centralized database



Design flow of distributed database

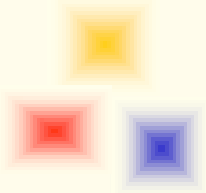


(1) Fragmentation design

WHERE DEPT=‗9 ‘ AND AGE>20;

WHERE SEX=‗Male‘;

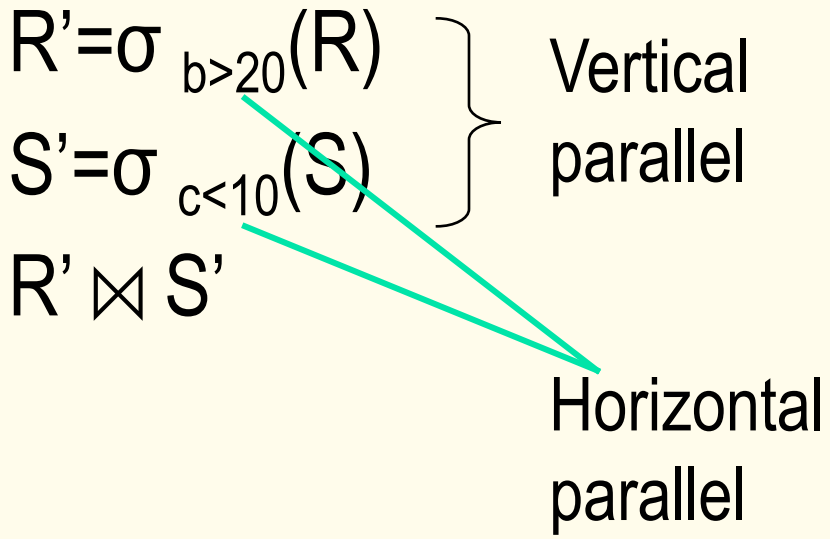
Problem:



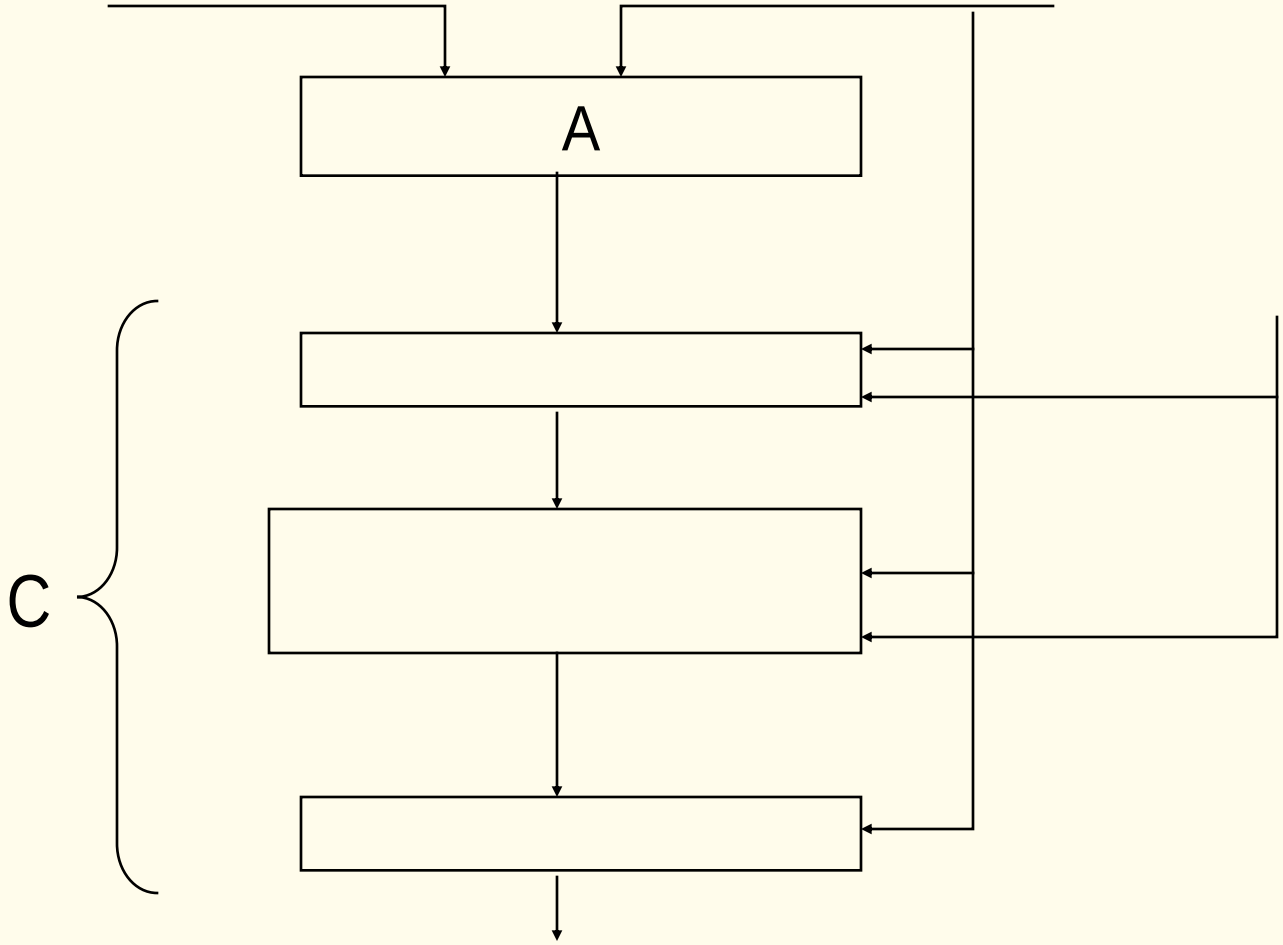
General rules:

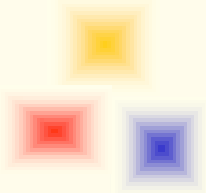
-

Example:



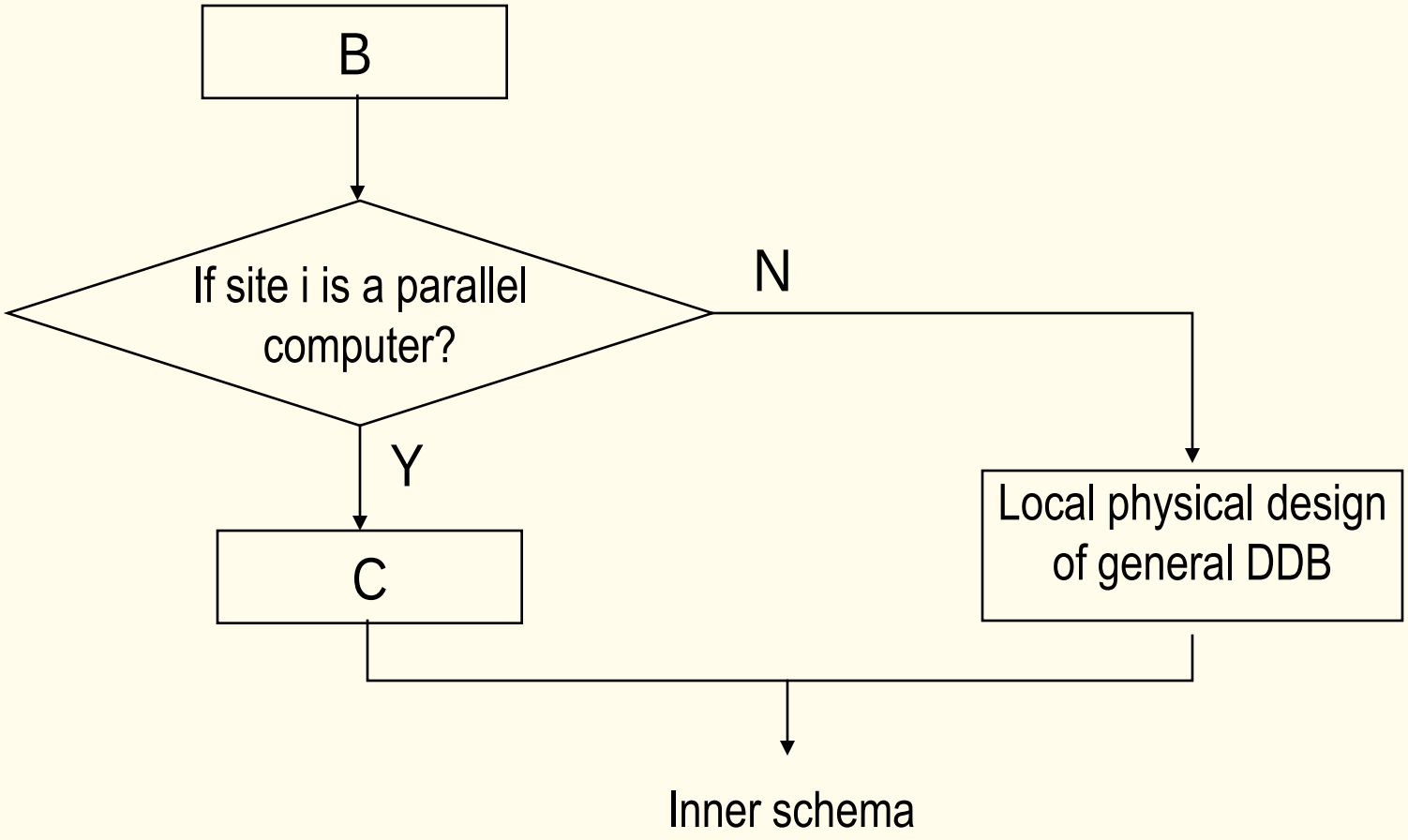
Design flow of parallel database





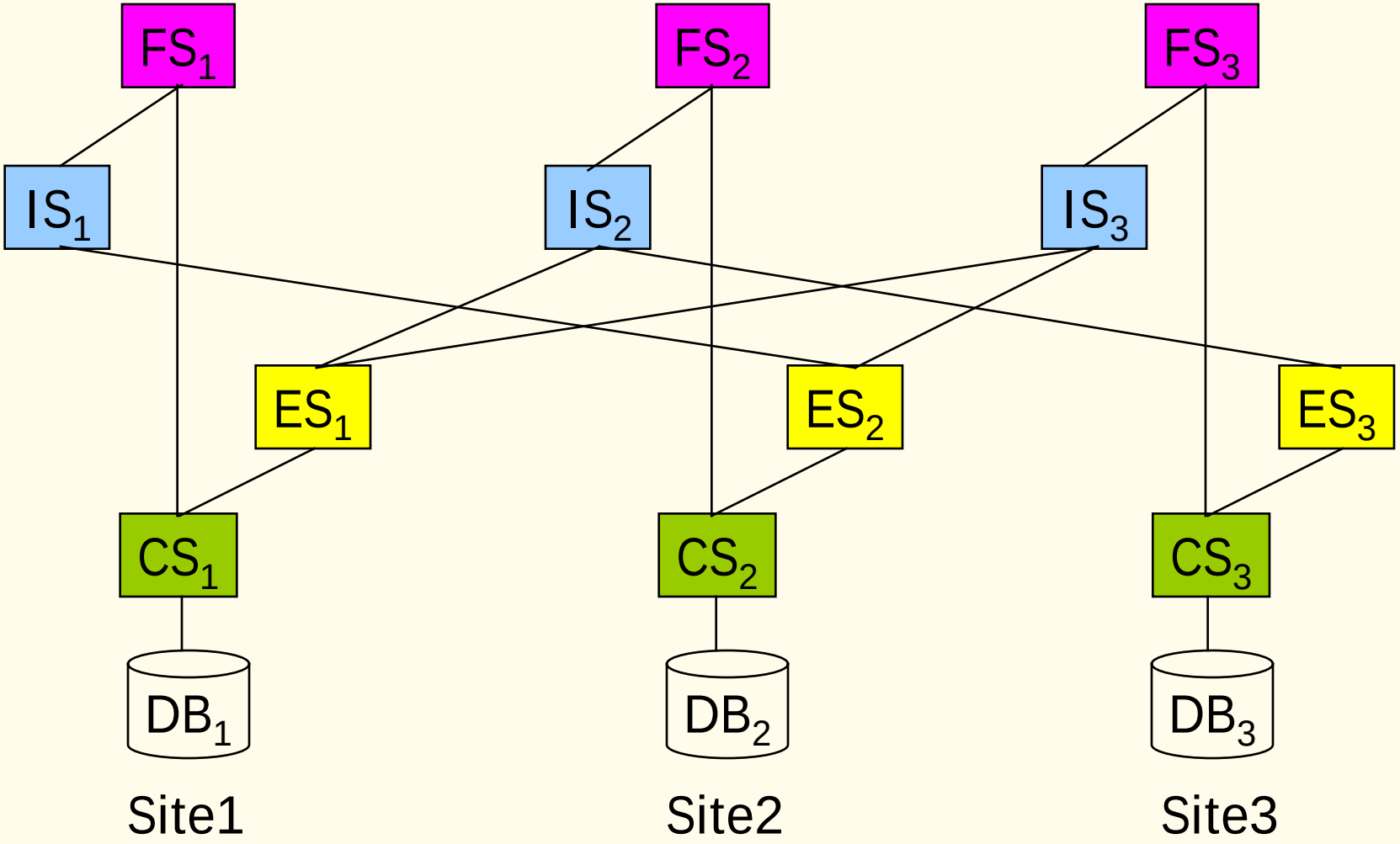
Data fragmentation mode in PDB

	PDB	DDB
The goal of fragmentation and distribution	computer‘s ability as	
Fragmentation in accordance with		
Distribution mode		


B
If site i is a parallel computer?
N
Y
C
Local physical design of general DDB
Inner schema

3) Federated database

-
-
-
-


FS1
FS2
FS3
IS1
IS2
IS3
ES1
ES2
ES3
CS1
CS2
CS3
DB1
DB2
DB3
Site1
Site2
Site3

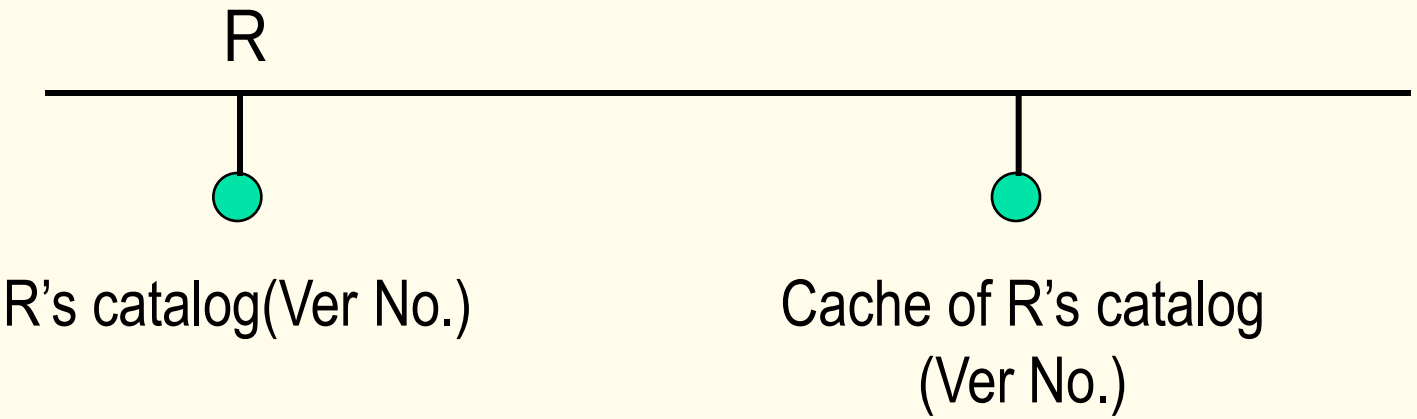
-
-
-
  ≠
- User‘s query on FS ⇒
  ⇒
- ⇒

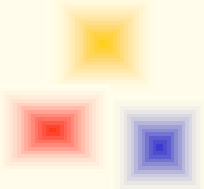
4.6 The Distribution of Catalog

Its main function is to transfer the user‘s operating

4.6.3 Distribution strategies of catalog

-
-


R
R's catalog(Ver No.)
Cache of R's catalog
(Ver No.)



4.6.4 Catalog management in R* --- Site autonomy

-
  -
  -
  -
-

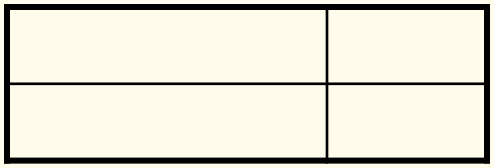
  <SWN>::=User@UserSite.ObjectName@BirthSite

  -

  - User: user‘s name. With this, different users can

Name Resolution --- Mapping PN to SWN

user using –Define Synonym



Only have ObjectName: search –ObjectNameǁ in

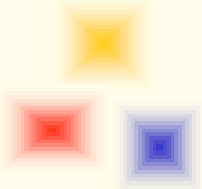
user –Userǁonǁcurrentǁsite.

Name Resolution --- Mapping PN to SWN

Name completion rule:

-
-

5. Query Optimization



Introduction

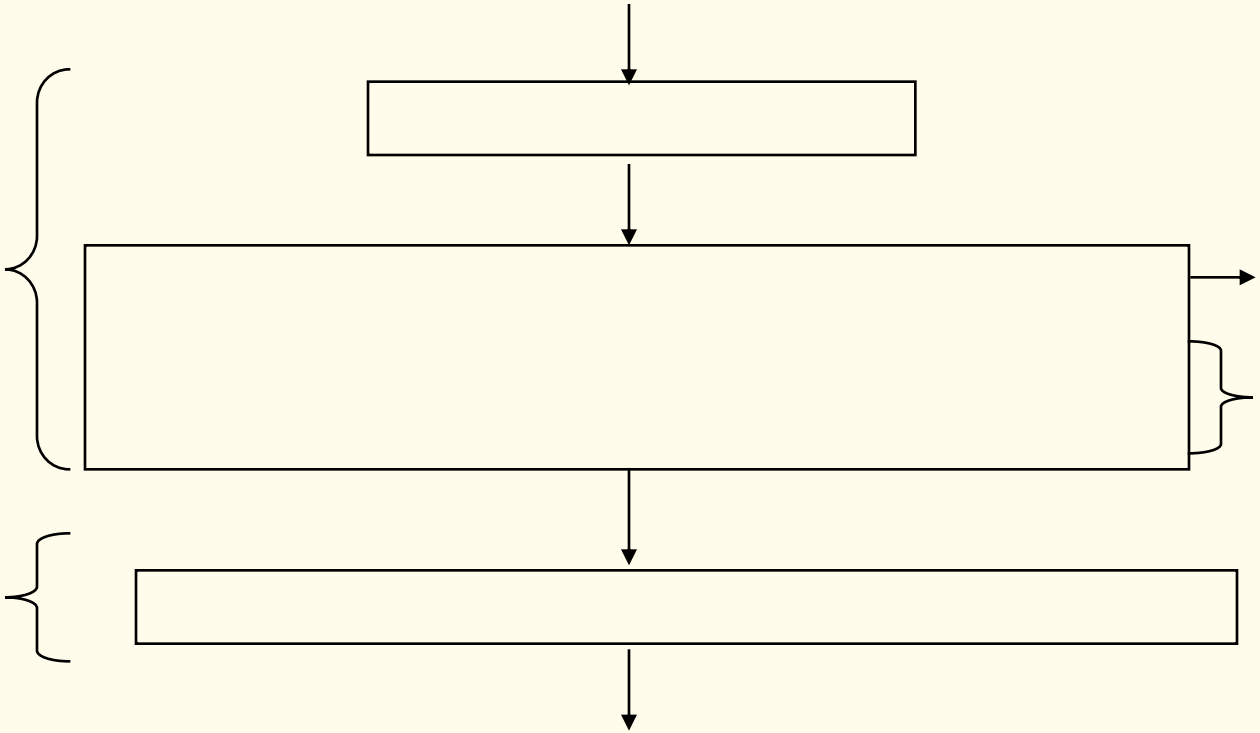
- Rewritell the query statements

- The goal is to gain the result of user‘s

5.1 Summary of Query Optimization in DDBMS

- Global Query
- Fragment Query

General flow



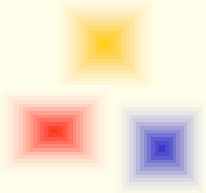
Example

$S1 = \sigma_{CITY=\text{`}Nanjing\text{'}}$

$S2 = \sigma_{CITY=\text{`}Shanghai\text{'}}$

$\ltimes$

$\ltimes$

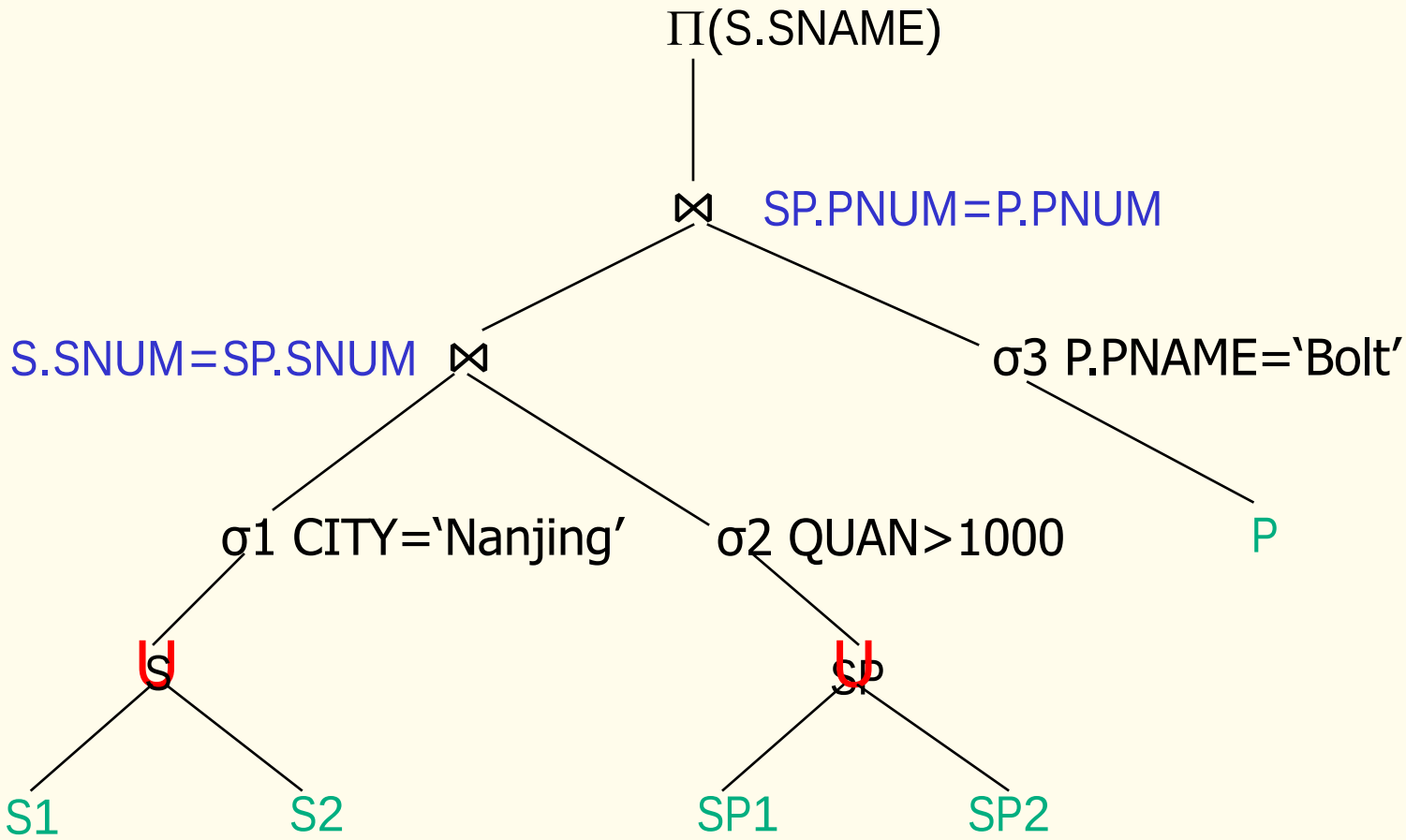


Global Query:

CITY=‗Nanjing‘

PNAME=‗Bolt‘

Query tree

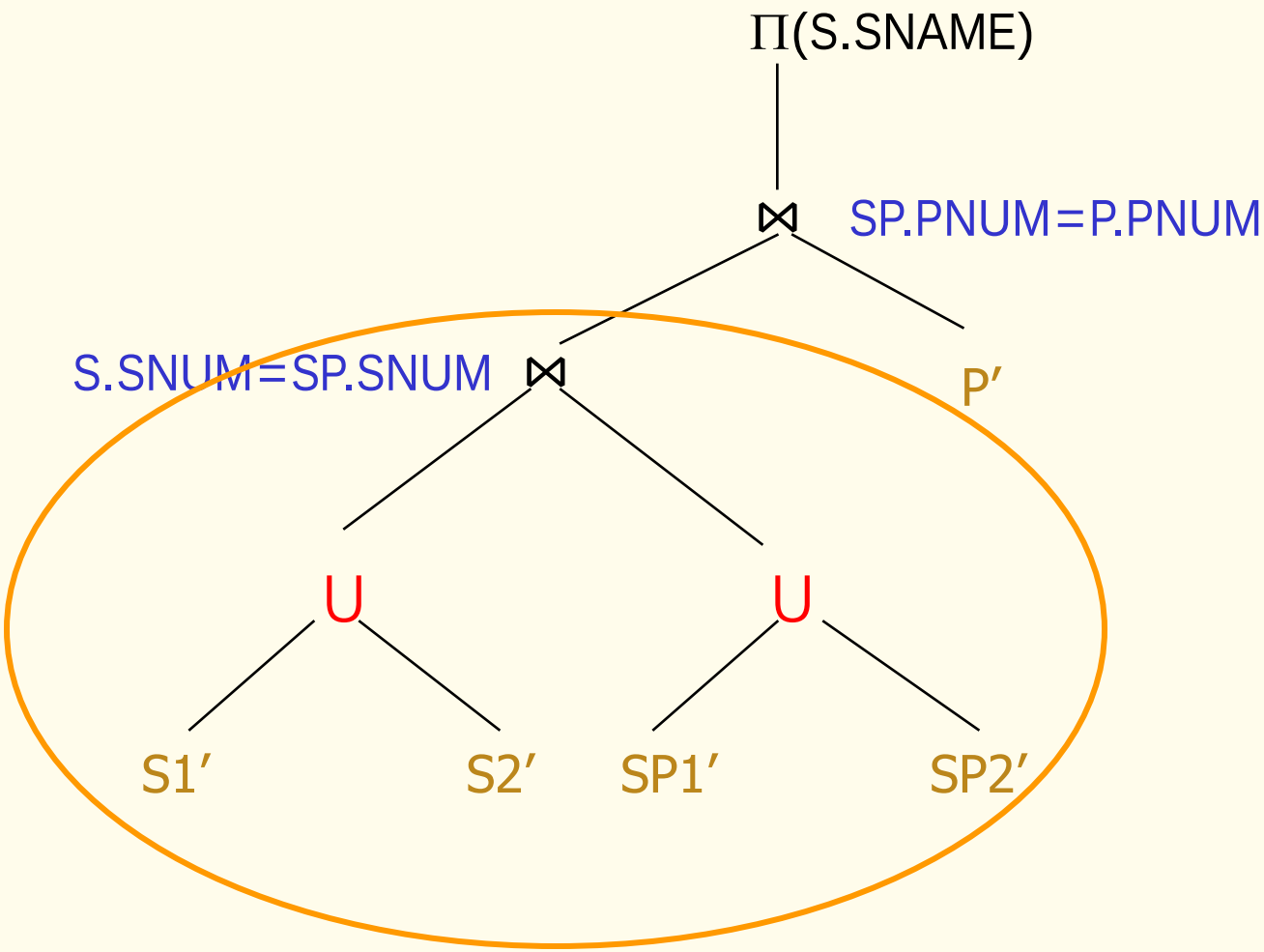

Π(S.SNAME)
⋈ SP.PNUM=P.PNUM
S.SNUM=SP.SNUM ⋈
σ3 P.PNAME='Bolt'
σ1 CITY='Nanjing'
σ2 QUAN>1000
P
S
SP
S1
S2
SP1
SP2

After equivalent transform (Algebra optimization) :

U

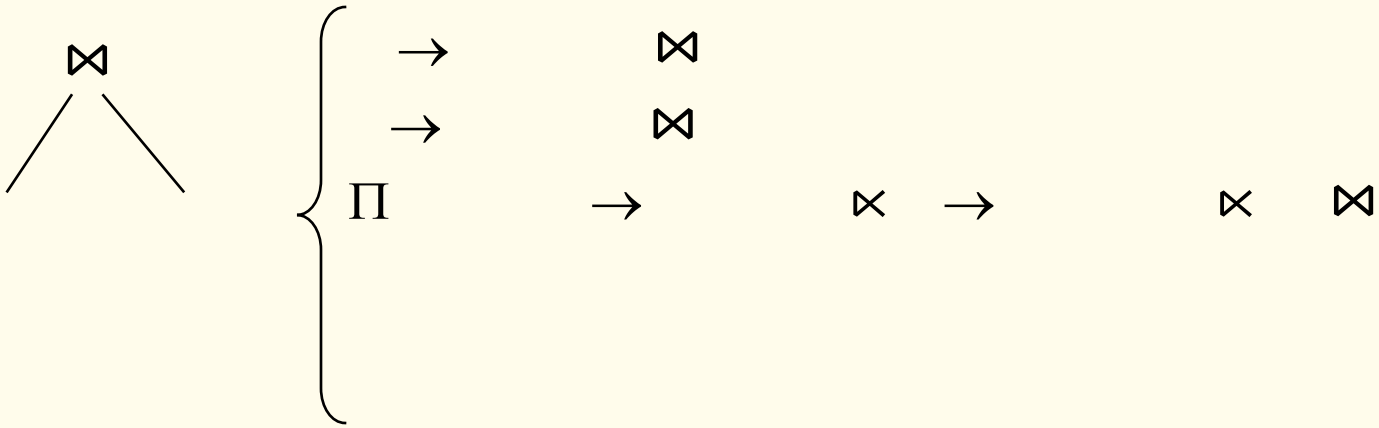
S1 S2

The result of equivalent transform



The operation optimization of the sub tree (yellow) :

- (1) (S1‘ U S2‘) ⋈ (SP1‘ U SP2‘)
- Then consider –Site Selection‖, may produce many combination
-

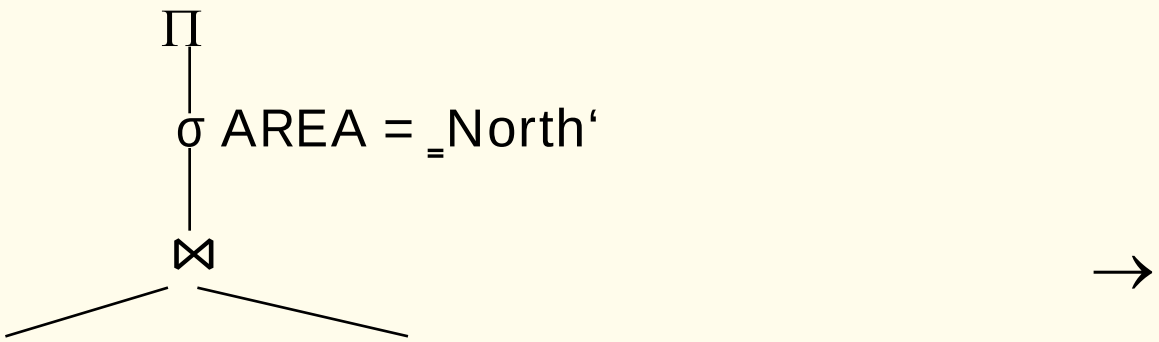


–good‖

5.2 The Equivalent Transform of a Query

Π σ (EMP)$\equiv\sigma$ Π

Π $\sigma_{AREA=\_NORTH'}$ $\bowtie$



$\rightarrow$

$\bowtie$ × × ≡ E2 ×

$\bowtie$ × × × ≡ (× ×

Π $\Pi_{A1...An}$ $\Pi_{B1...Bm}$(E)) ≡ $\Pi_{A1...An}$

…A …B

σ σ σ (E)) ≡ σ

σ Π σ $\Pi_{A1...An}$(E)) ≡ $\Pi_{A1...An}$ σ

…B which don‘t belong to …A

$\Pi_{A1...An}$ σ (E)) ≡ $\Pi_{A1...An}$ σ $\Pi_{A1...An\ ,\ B1...Bm}$

σ ×E2) ≡ σ ×

F2, and there are only E1‘s attributes in F1, and there are only E2‘s attributes in

σ ×E2) ≡ σ ×σ

F2, and there are only E1‘s

σ ×E2) ≡ σ σ ×

σ U E2) ≡ σ U σ

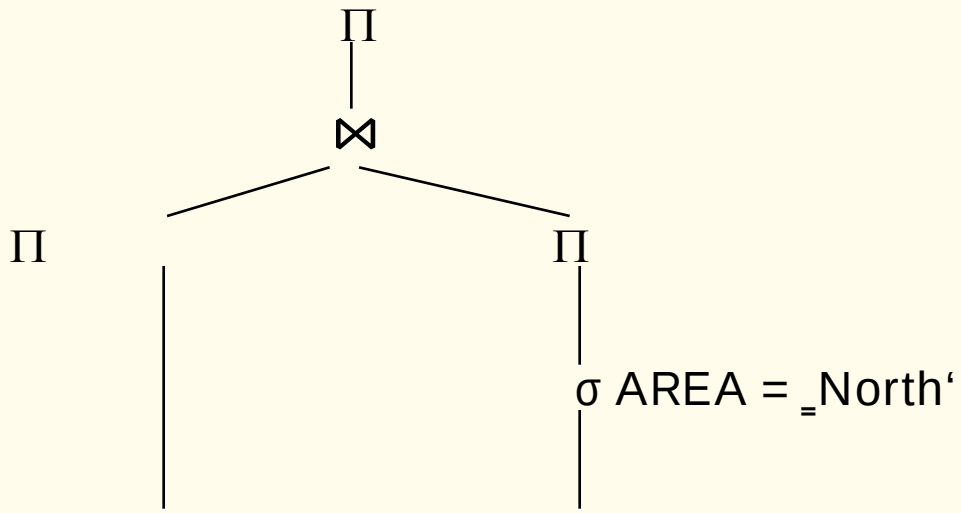
σ E2) ≡ σ σ

…A

…B are E1‘s attributes, and C …C are E2‘s

$\Pi_{A1...An}$ ×E2) ≡ $\Pi_{B1...Bm}$ ×$\Pi_{C1...Ck}$

$\Pi_{A1...An}$ U E2) ≡ $\Pi_{A1...An}$ U $\Pi_{A1...An}$



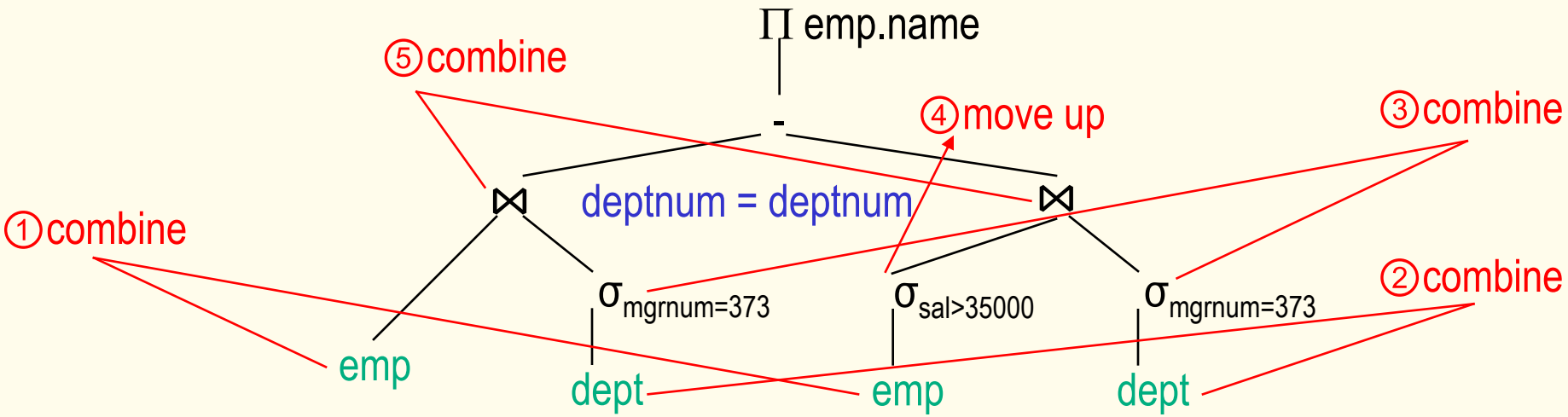
5.3 Transform Global Queries into Fragment Queries

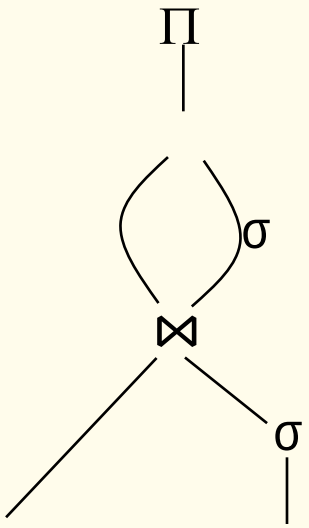
-
  - U U … U
  - ⋈ ⋈ … ⋈

✓

✓ Π ⋈ σ −

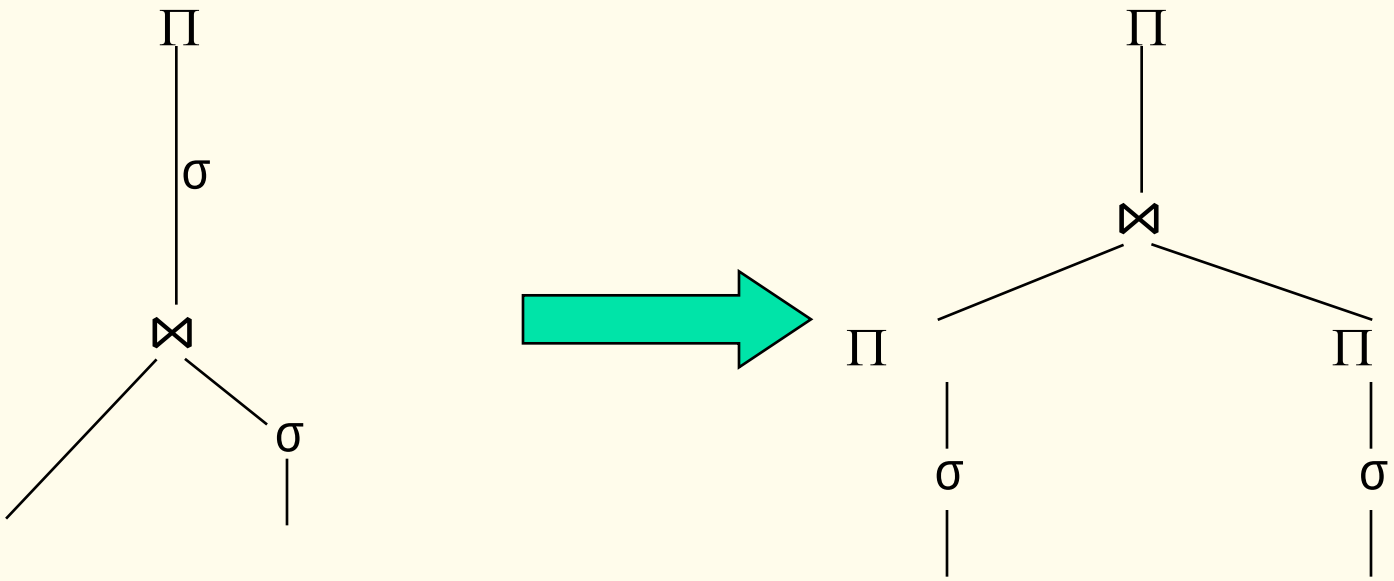
σ ⋈ σ

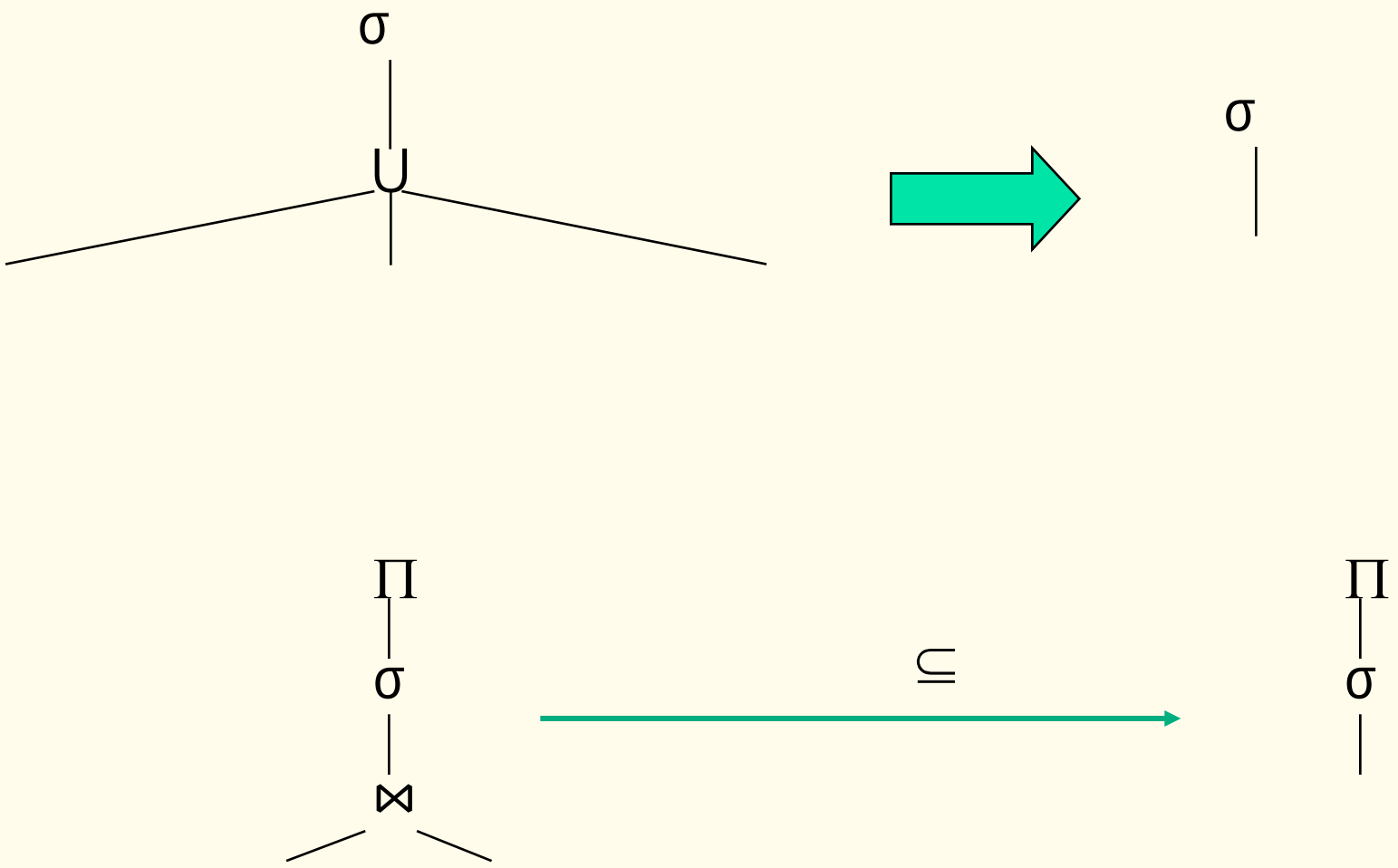

Π emp.name
⑤combine
-
④move up
③combine
①combine
⋈
deptnum = deptnum
⋈
②combine
σmgrnum=373
σsal>35000
σmgrnum=373
emp
dept
emp
dept



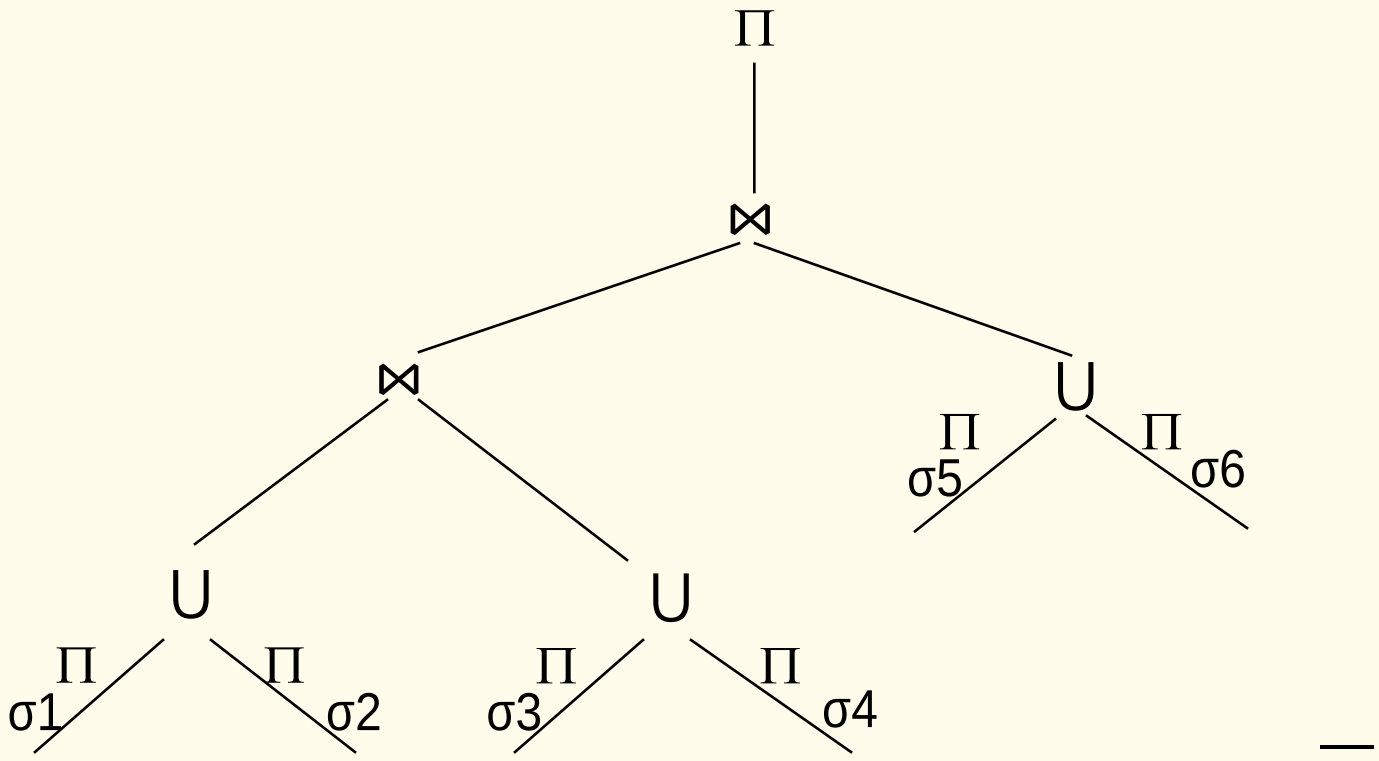
- ✓ ⋈ ≡ R
- ✓ U R ≡ R
- ✓ - R ≡ ϕ
- ✓ ⋈ σ (R) ≡ σ
- ✓ U σ (R) ≡ R
- ✓ - σ ≡ σ
- ✓ σ ⋈ σ (R) ≡ σ
- ✓ σ U σ (R) ≡ σ
- ✓ σ - σ (R) ≡ σ

emp ⋈ ($\sigma_{mgrnum=373}$ dept)

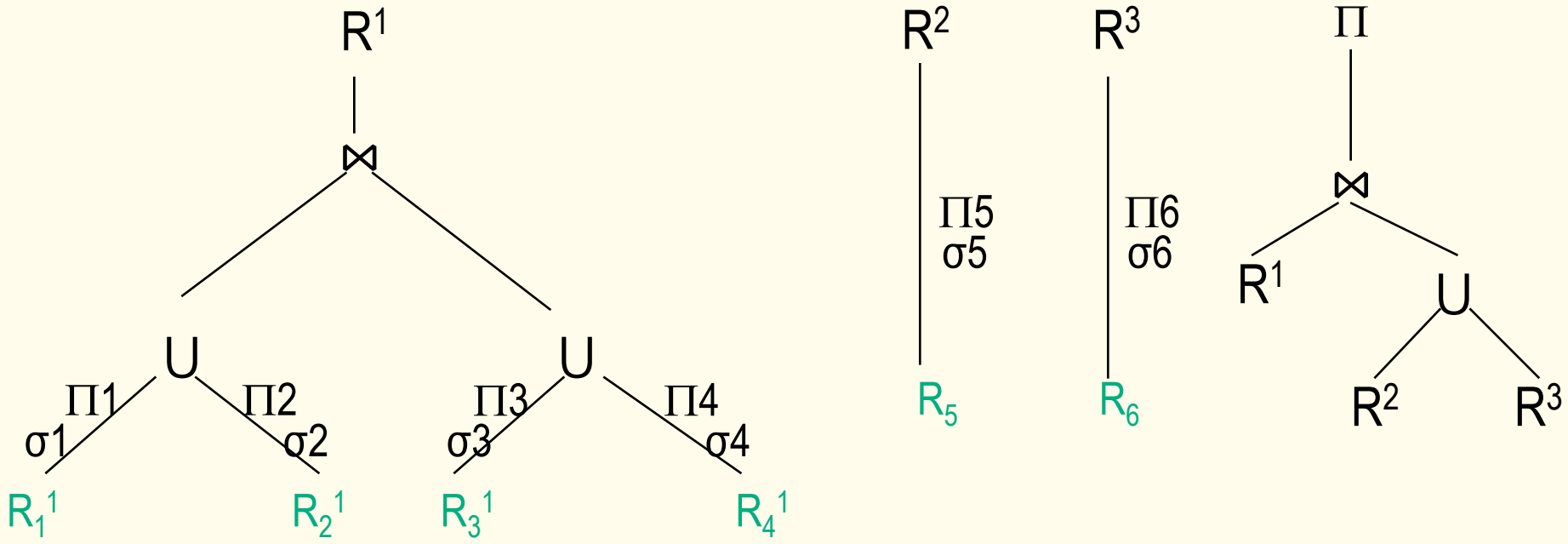

Π
σ
⋈
σ
Π
⋈
Π
Π
σ
σ


σ
∪
σ
Π
σ
⋈
⊆
Π
σ

5.4 Query Decomposition



Decomposition method :

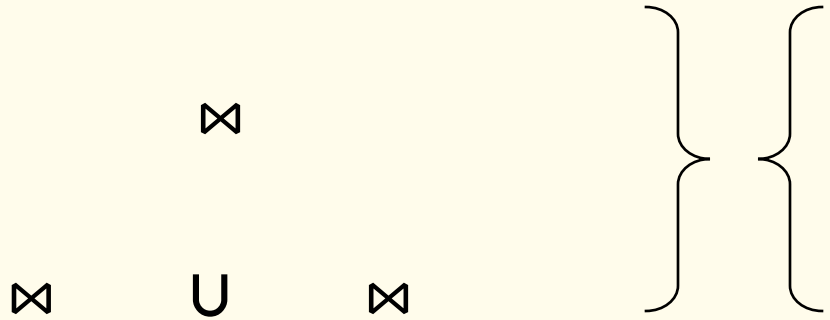


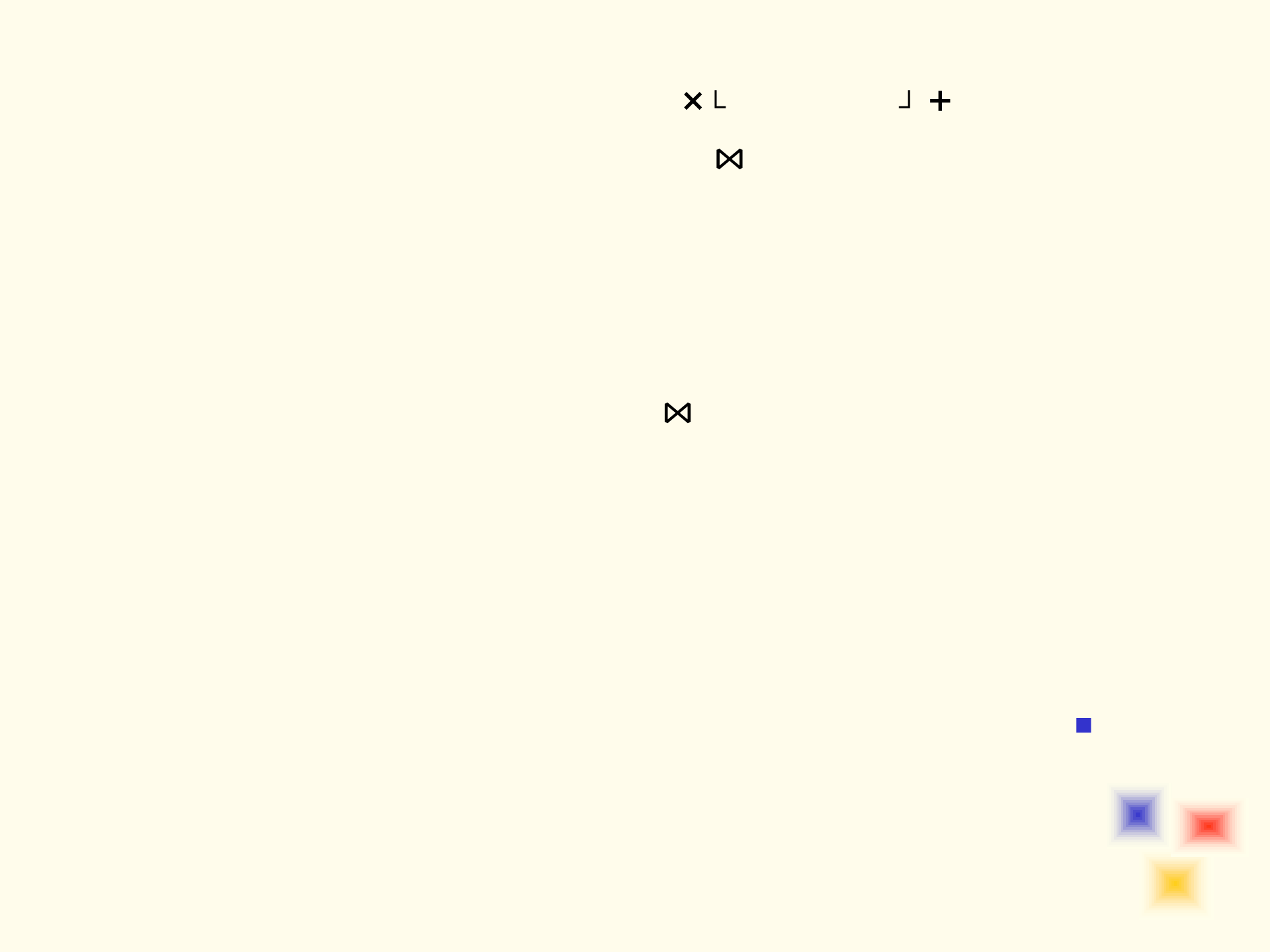
Site1 Site2 Site3 Assembling tree

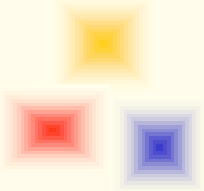
5.5 The Optimization of Binary Operations

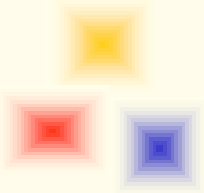
How to find a –goodǁ access strategy to compute the

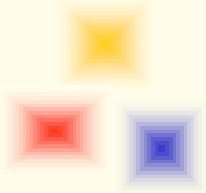
I. Main problems in global optimization











-
-

Processing cost

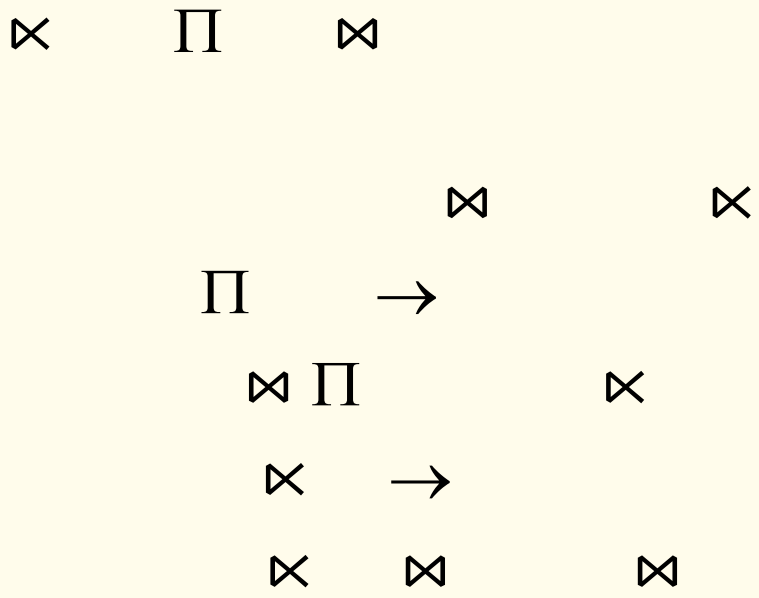
2) Processing cost

μ

×

❖

5.6 Implement Join Operation With Semi_join



Cost comparison

-
-

s‘ + C r‖, 2C r‘ + C s‖) =

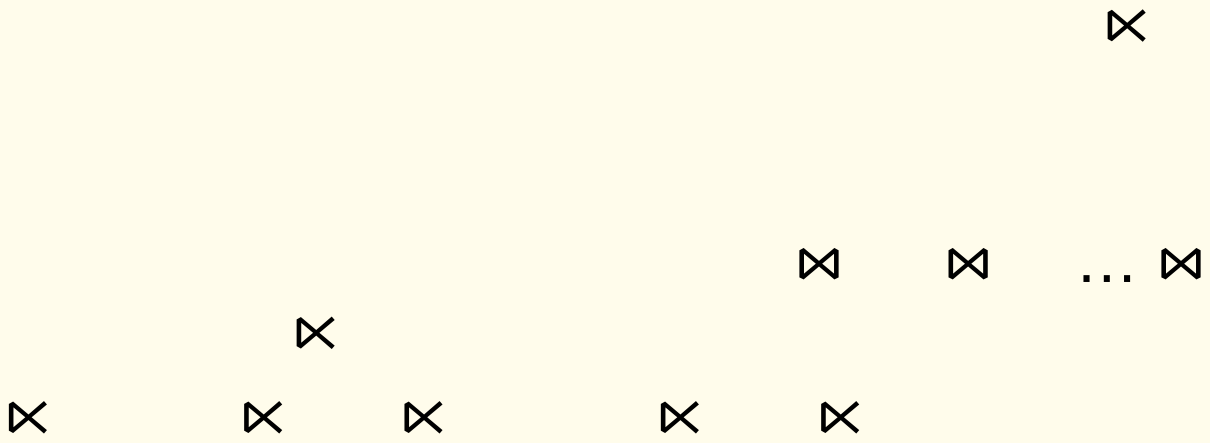
min(s‘ + r‖, r‘ + s‖)

s‘, r‘ Π Π

s‖, r‖ ∝ ∝

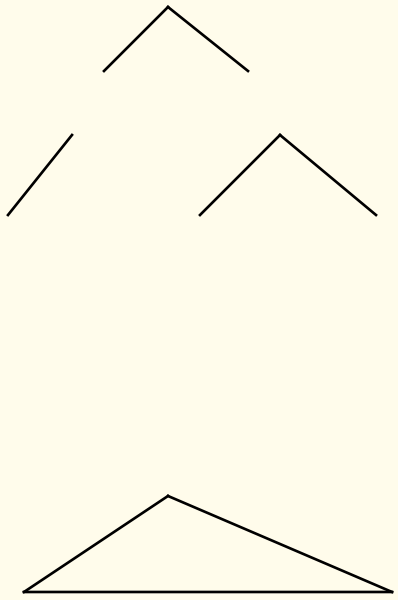
-

II. Comments on semi_join

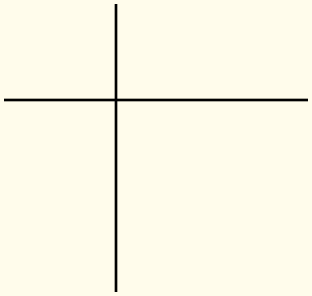


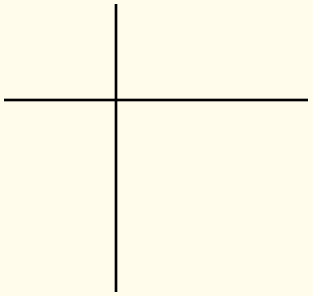
-
-
  ∈
-

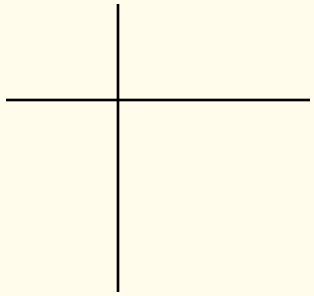
Query graph:



Can be proved:

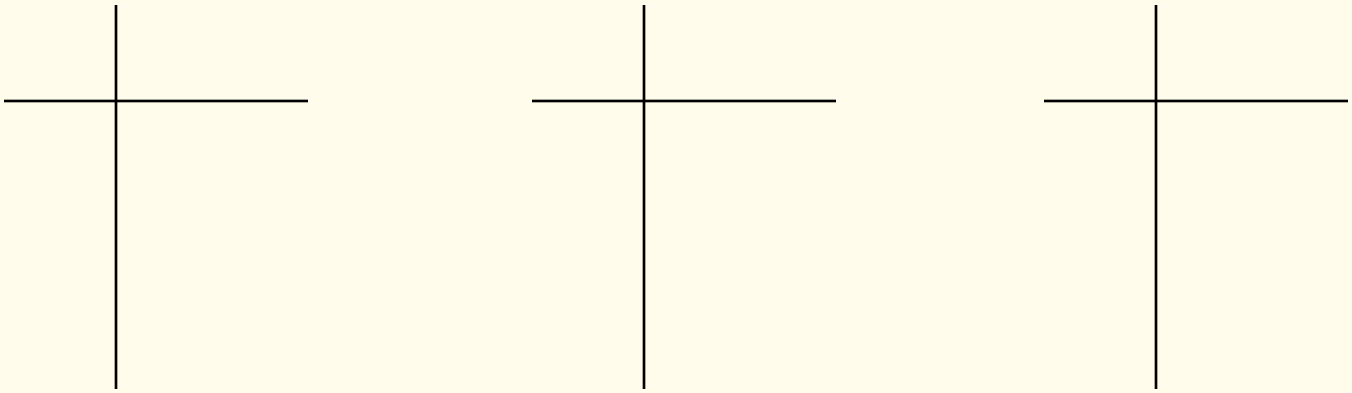






$\ltimes$

Another example about full reducer of CQ



R‘=R ⋉ ———— S‘=S ⋉ R‘= ———— T‘=T ⋉ S‘= ————

R‖=R‘ ⋉ T‘= ———— S‖=S‘ ⋉ R‖= ———— T‖=T‘ ⋉ S‖= ————

R‖‘=R‖ ⋉ T‖=Φ, So the full reducer of relation R in

R‘=R ⋉ S‘=S ⋉ R‘ T‘=T ⋉ S‘ R‖=R‘ ⋉ T‘

S‖=S‘ ⋉ R‖ T‖=T‘ ⋉ S‖ R‖‘=R‖ ⋉ T‖= Φ

-
-
-

⋉

SDD-1 Algorithm: heuristic rule (p189~190)

5.7 Direct Join

-

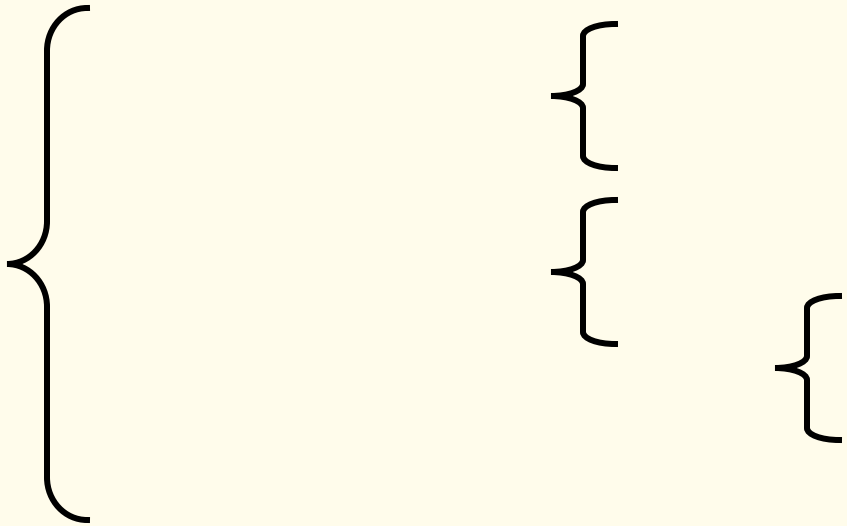
-

1)

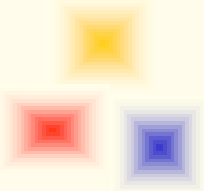
- For O, the relation doesn‘t need storing.

2)

III. Six implementation strategies of join in R*



- In NL, if I is shipped whole, index can‘t be shipped

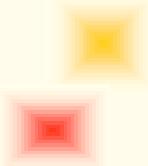


-
- because R* doesn‘t

$$\bigcup \subseteq \bigcap \bigcup \Phi \text{ for all } i, j$$

$$\bigcup \qquad \bigcup$$

- If SP is derived fragmented according to the supplier‘s city:
- If SP is derived fragmented according to the part‘s type: don‘t



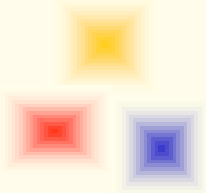
, …, S

), …, SUM(S

), … COUNT(S

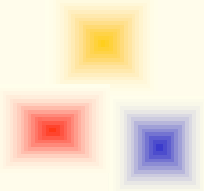
), …, MIN(S

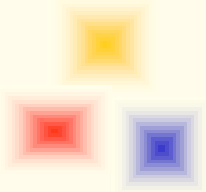
), …, MAX(S

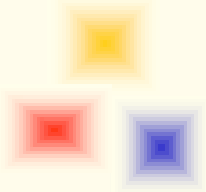


5.9 Update Strategies

$$\lim_{n \to \infty} p^n = 0$$







-
-
-

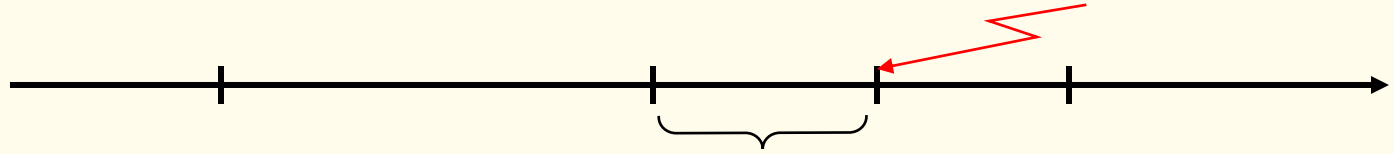
6. Recovery

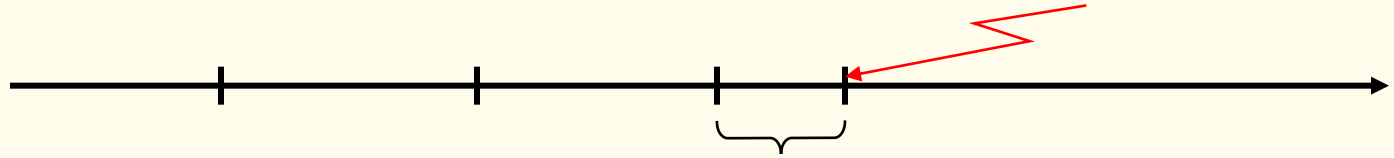
6.1 Introduction

Restore DB to a consistent state after some failures.

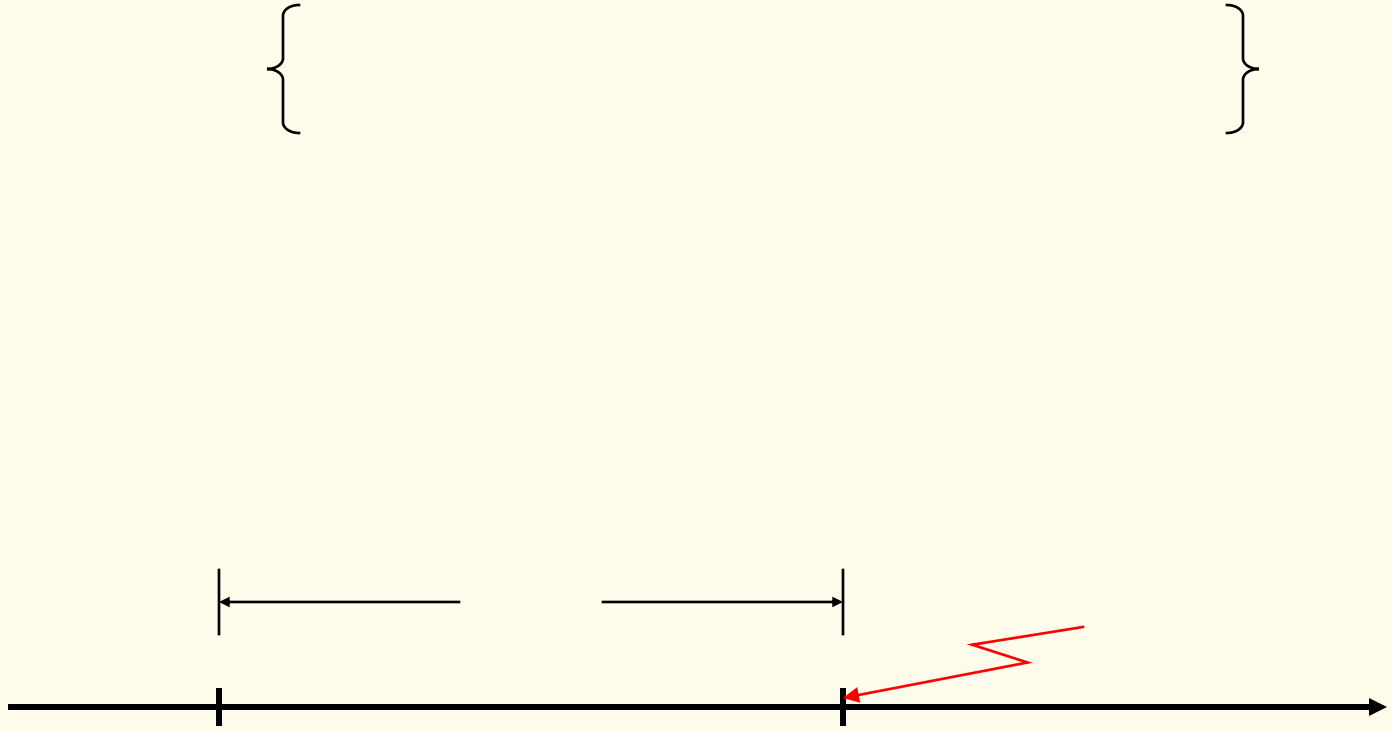
-
-
-

1) Periodical dumping





2) Backup + Log

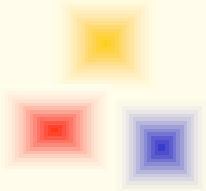


3) Multiple Copies

cannot be collapsed because of some copy‘s failure.

6.2 Transaction

- ✓
- ✓
  →
- ✓
- ✓



Begin transaction

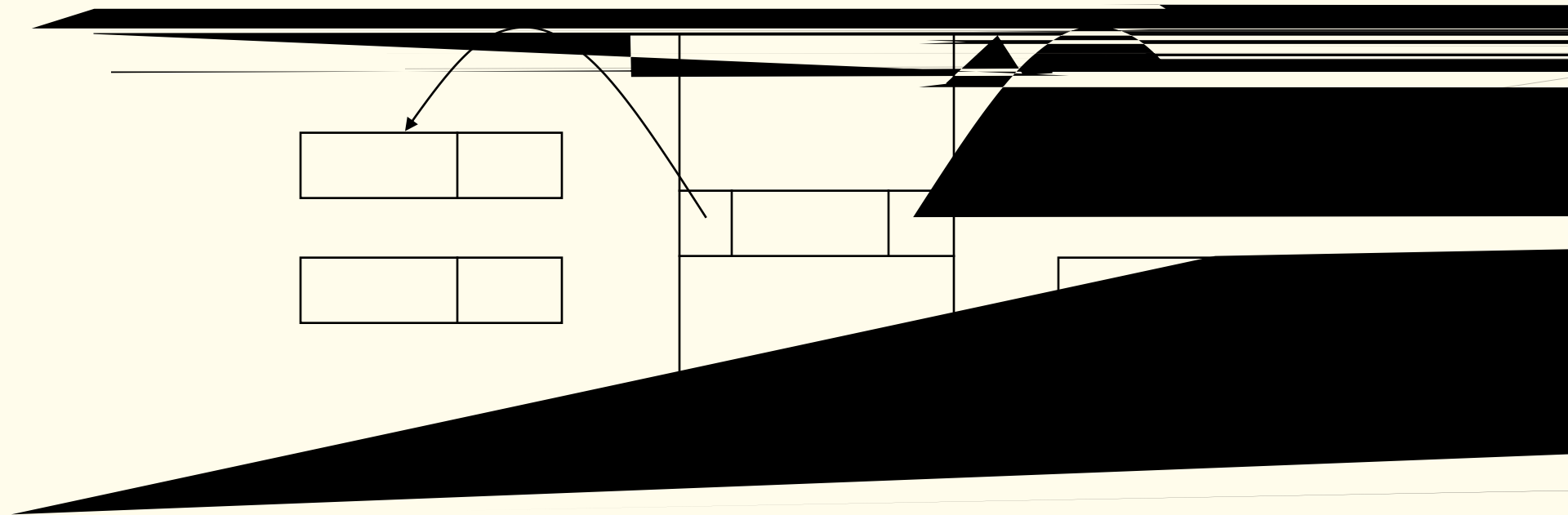
if A<0 then Display –insufficient fund‖

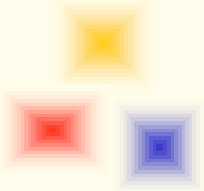
Rollback

Display –transfer complete‖

Commit

6.3 Some Structures to Support Recovery

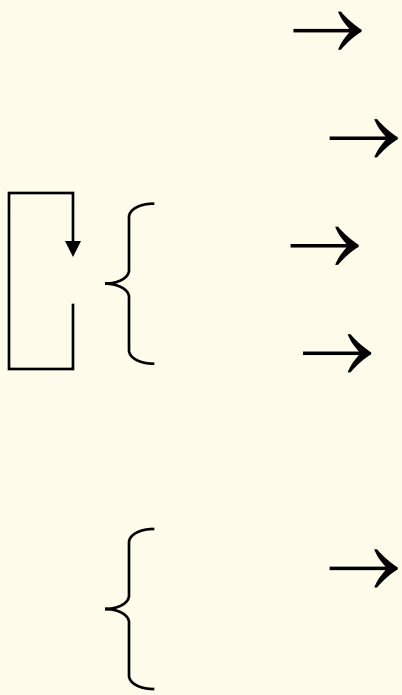




- ✓ Don‘t have to store Log information for aborted
- ✓
- ✓

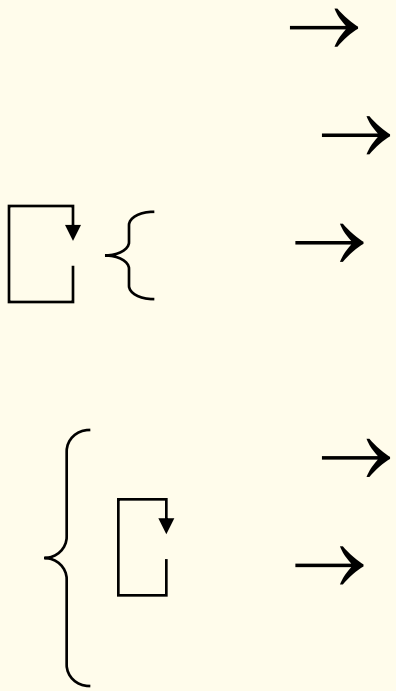
6.4 Commit Rule and Log Ahead Rule

(2) Three kinds of update strategy



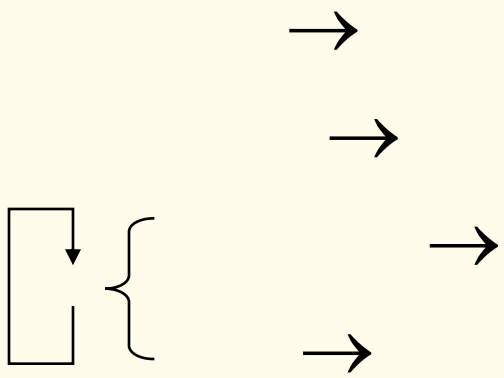
The recovery after failure in this situation

	✓	
✓	✓	
✓		



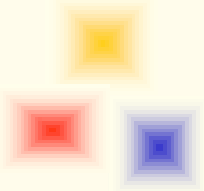
The recovery after failure in this situation

	✓	
✓	✓	
✓		



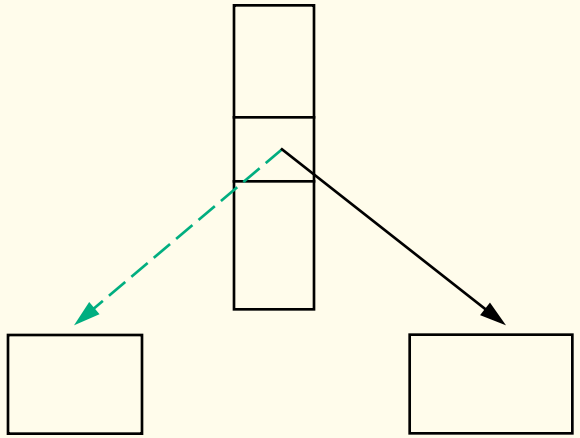
The recovery after failure in this situation

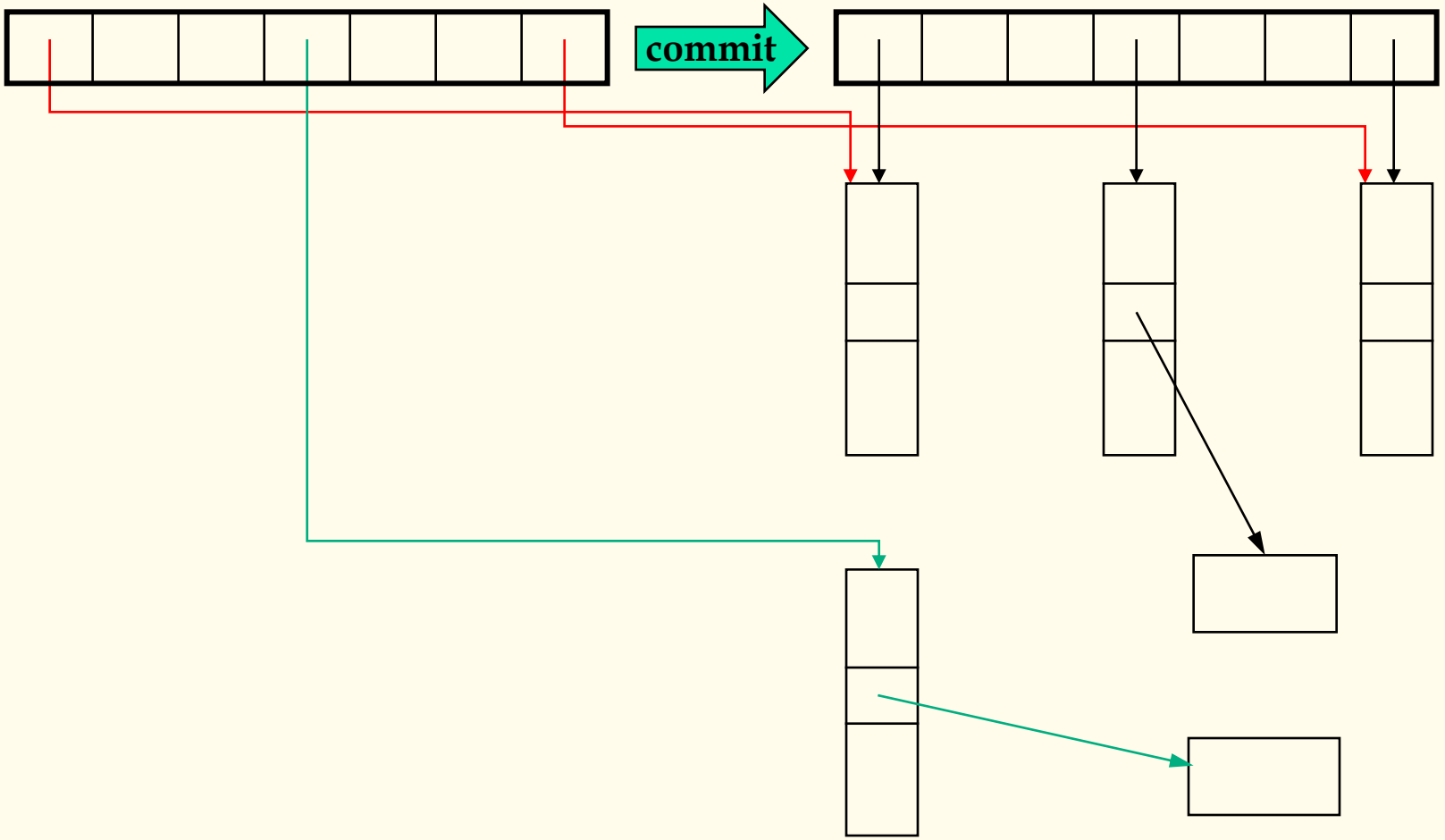
	✓	
✓	✓	
✓		



6.5 Update Out of Place

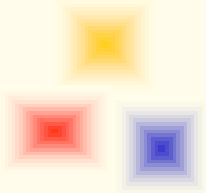
-
-
-




commit

6.6 Recovery Procedures

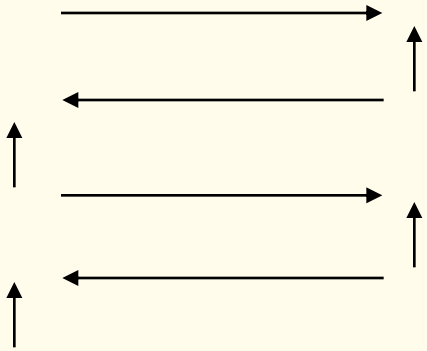
-
-



6.7 System Start Up

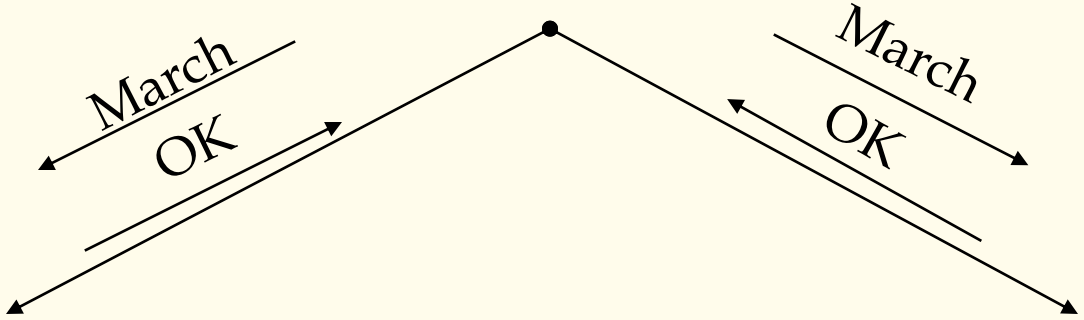
6.8 Two Phase Commit

-
- transactions‘ harmony with each other relies on
-
-

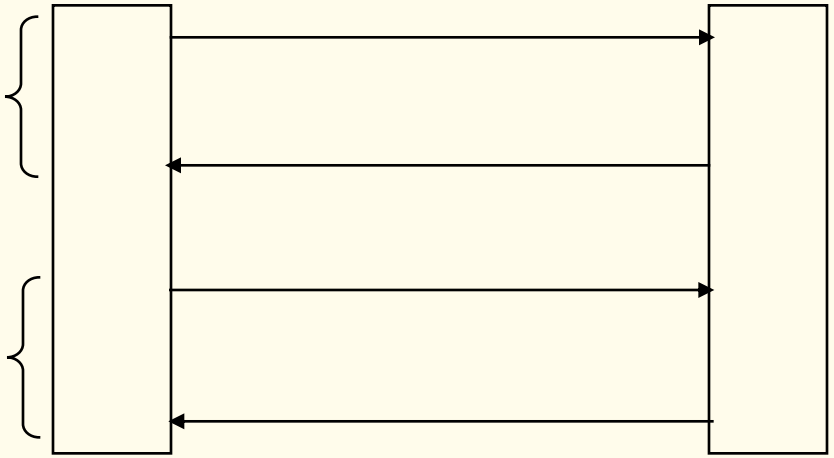


If A has not received OK i+1
1) B had not received March i+1
2) OK i+1 has been lost

A send March i+1 again; for B :
1) If has received, send OK i+1 again
2) or, ↑, send OK i+1


March
OK
March
OK

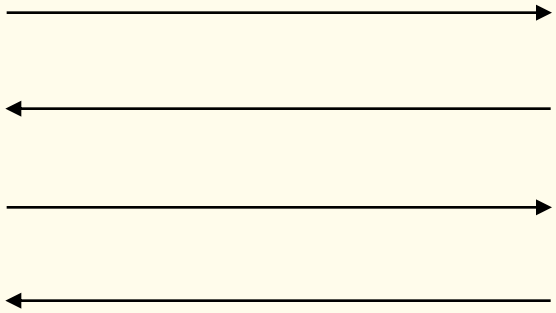
Two Phase Commit



-

-

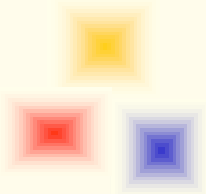
Three Phase Commit



-
-

7. Concurrency Control

7.1 Introduction



7.1.3 Serialization --- the criterion for concurrency consistency

,…T

,…T

→

→

→ →

→ →

7.1.4 View equivalent and conflict equivalent

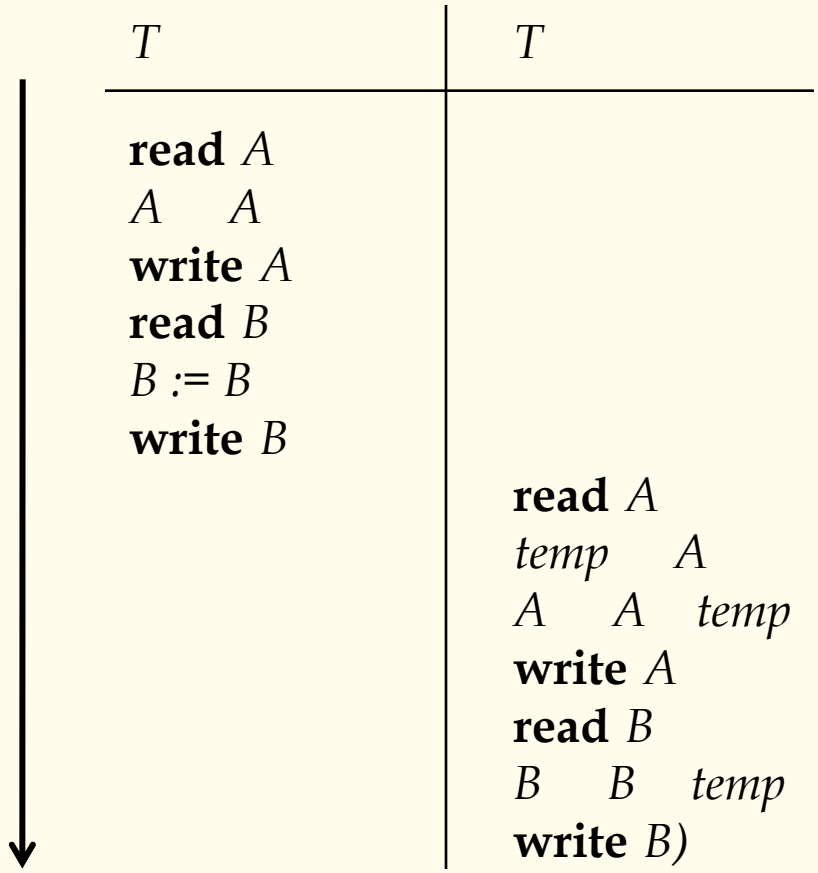
- *Schedules*

✓

✓

Example Schedules

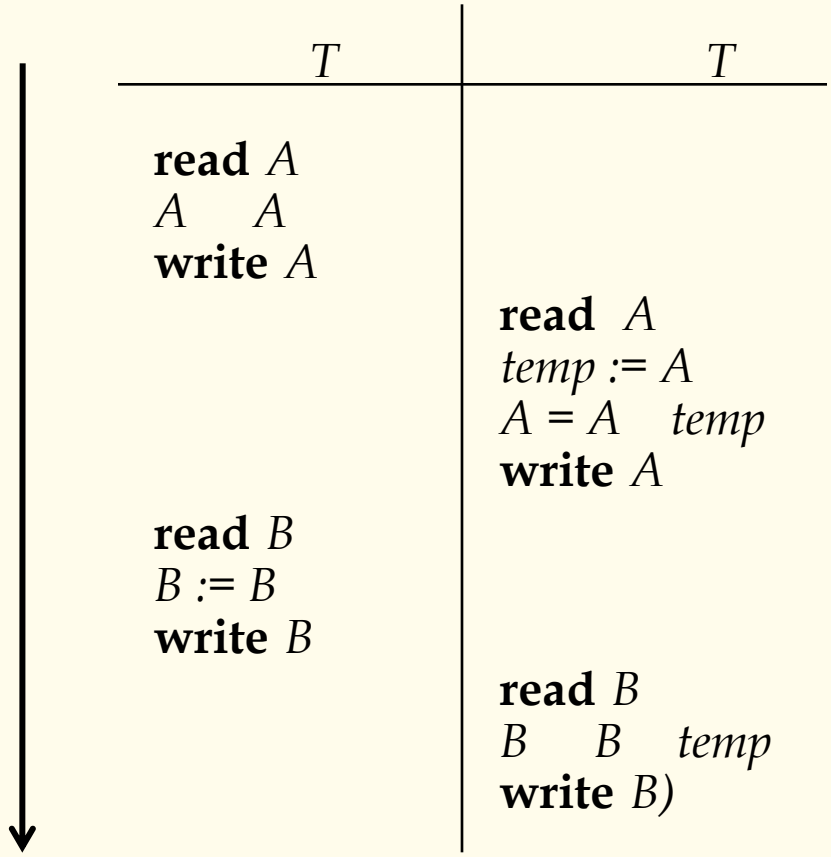
- *T* *A* *B* *T*
  A *B*. *T* *T*



Example Schedules

- *T* *T* .

equivalent



View equivalent and *Conflict equivalent*

- *S*

  S *view equivalent*

- 、

- ① ②

- *S*

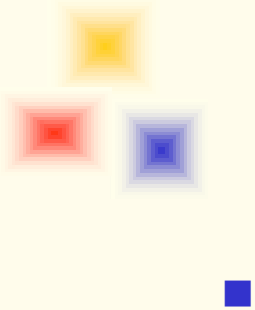
  S *conflict equivalent.*

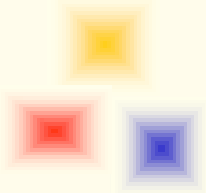
- S
- **view serialization** **conflict serialization**
- (y) → R (x) = s‘

  s is conflict serialization because s‘ is a serial execution.
- schedule s‘

  s‘ = R

  It is view equivalent with s, and s‘ is a serial execution, so s is

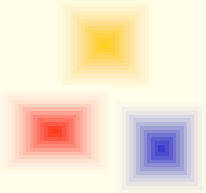




7.1.5 Preceding graph

$\rightarrow \mathsf{T}$

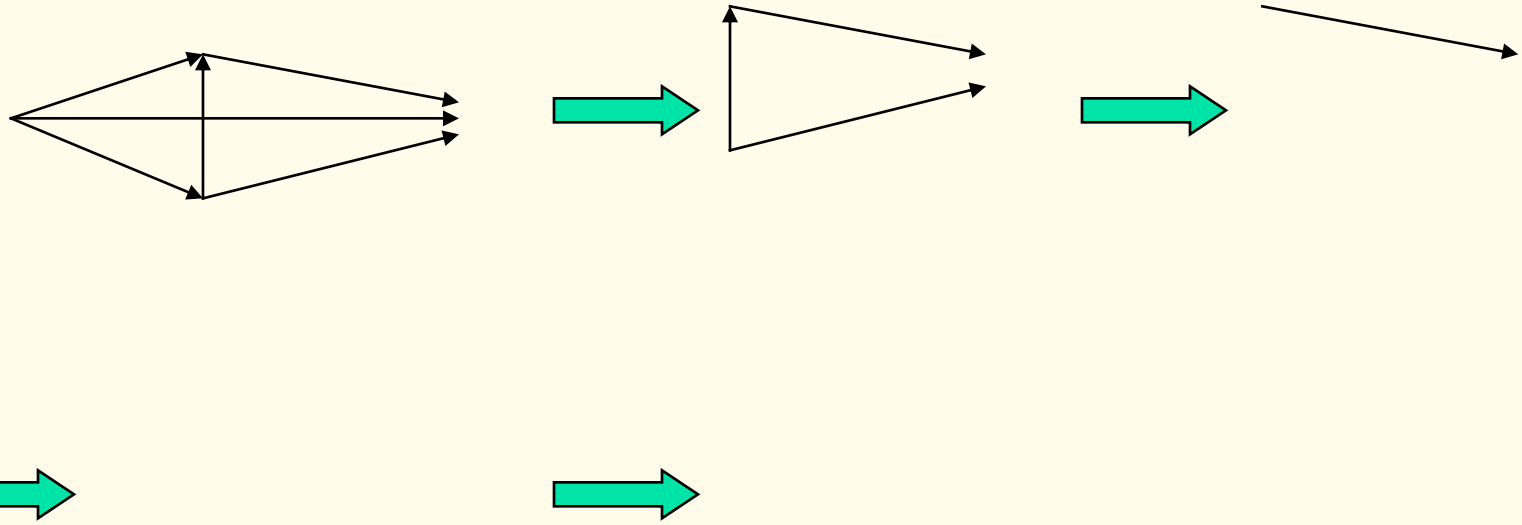
-
-
-



Find equivalent serial execution while serialization

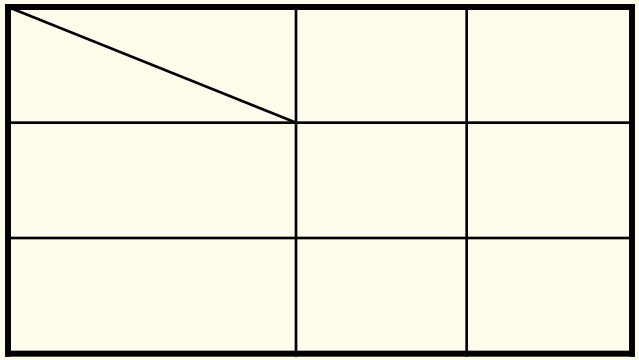
Example:

$$s = W_3(y)R_1(x)R_2(y)W_3(x)W_2(x)W_3(z)R_4(z)W_4(x)$$



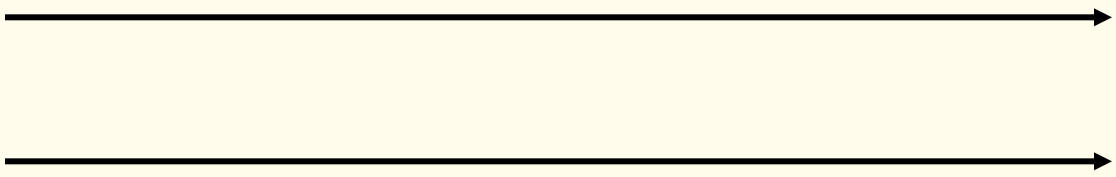
→T →T →T

7.2 Locking Protocol

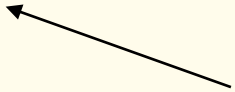


Conclusions :

7.2.3 (S,U,X) locks



7.3 Deadlock & Live Lock



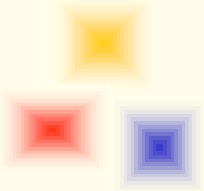
-
- don‘t let it occur

7.3.1 Deadlock Detection

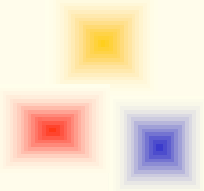
(i=1,2,…n)}

(i ≠ j)}

-
-



- Pick a victim (youngest, minimum abort cost, …)



7.4 Lock Granularities

- - -

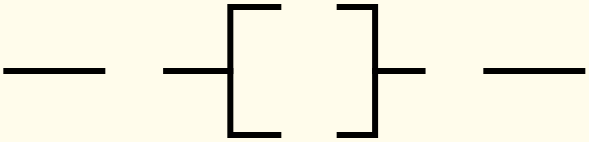
-
-

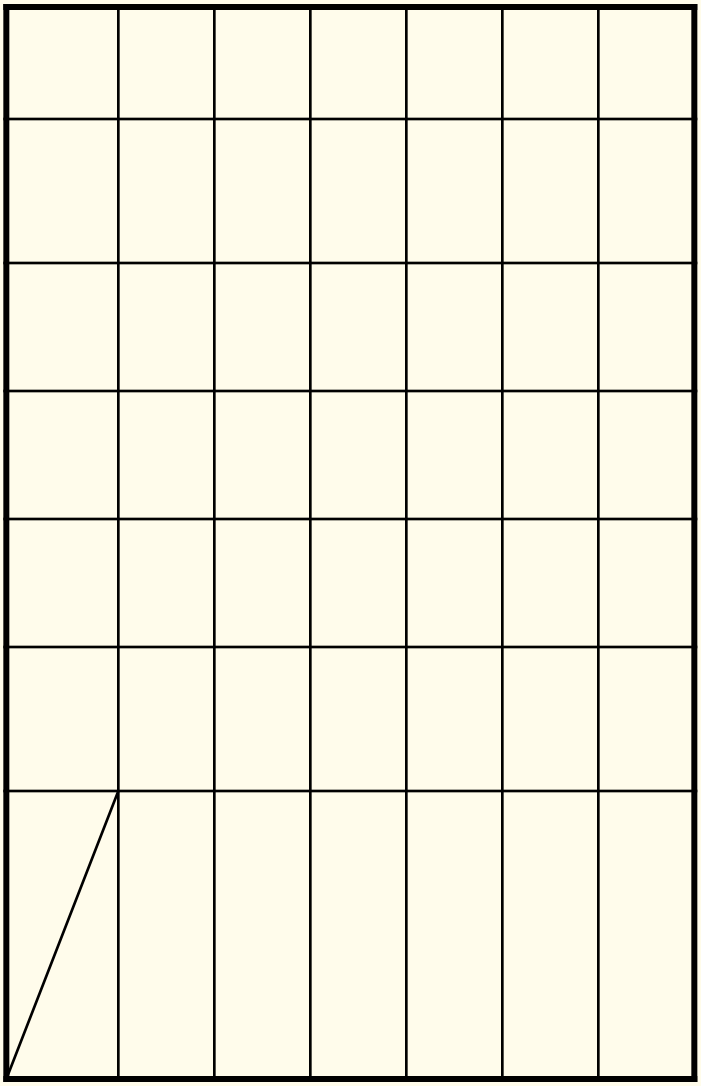
7.4.2 Intention lock

-
-

-
-
- +

Exclusion degree





Locking Rules:

-

7.5 Locking on Index (B+ Tree)

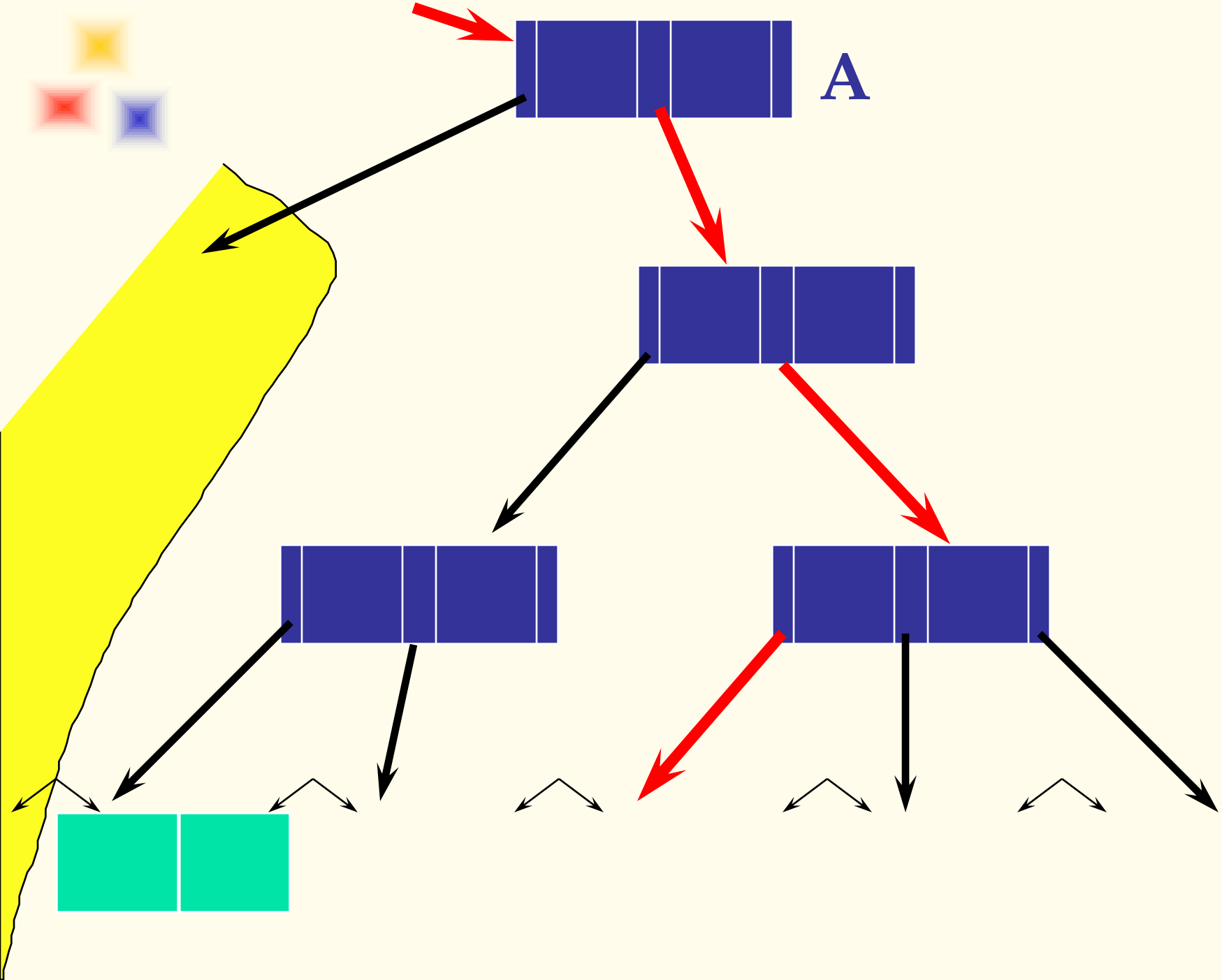
-
-
  - Btw, don‘t confuse this with multiple granularity locking!
-
-
  -

Some Useful Observations

-
- unit of B+ tree is page. Don‘t need multi granularity
-
-

Some Useful Observations

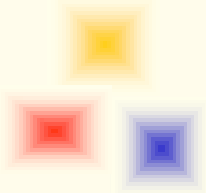
- this sense, the locks on index don‘t need keeping to mapping from attribute value to tuples‘ addresses.


A

Tree Locking Algorithm

on parent can be released, because traversing can‘t go back.

-
-
-
-



Tree Locking Algorithm

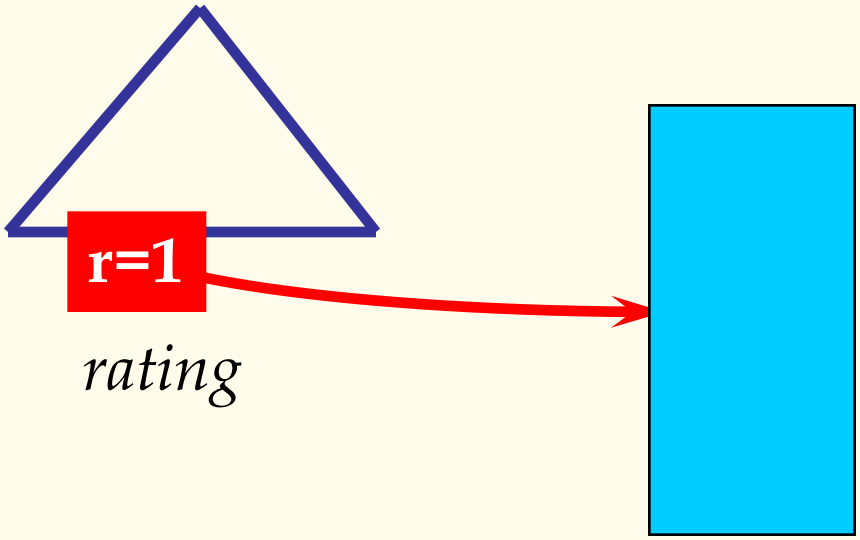
7.6 Phantom and Its Prevention

- **assumption**
  - *rating* _____ *age*
  - *rating* *age*
  - *age*
  - *rating* _____ *age*
- No consistent DB state where T1 is –correctǁ!

The Problem

- *rating*
  -
  -
-
- If the system don‘t support multi granularity locking,

Index Locking



-
  rating
  - *rating*
    would
- *rating*
  age = 96), he can‘t get the X lock on the index node
  rating
  can‘t insert the new index item to realize the insert
-

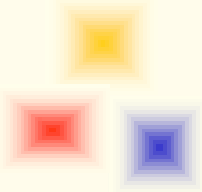
Predicate Locking

-
-
  -
-

*age > 2*salary*

7.7 Isolation Level of Transaction

-
-



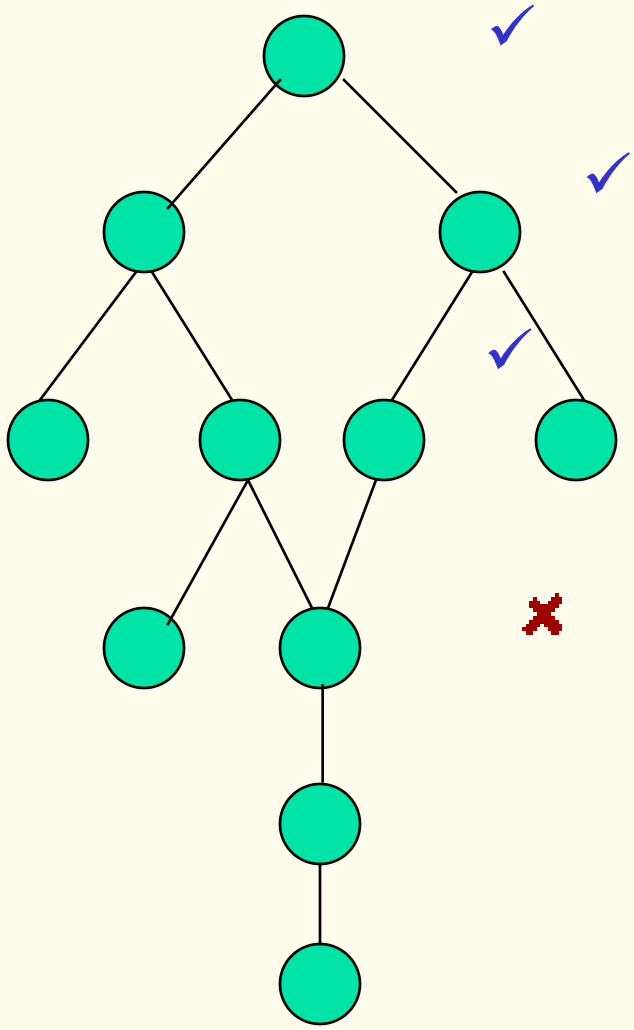
Example :

```
SET TRANSACTION ISOLATION LEVEL
{  READ COMMITTED |
   READ UNCOMMITTED |
   REPEATABLE READ |
   SERIALIZABLE
}
```

7.8 Lock Mechanism in OODBMS

- -

Locking steps of RL(WL) locks:



7.9 The Time Stamp Method

A number generated by computer‘s internal

tr

tw

Read/Write operations under T.S method

Remarks:

-
-

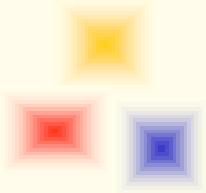
7.10 Optimistic Concurrency Control Method

above are called –pessimistic method‖.

Three phases of transaction execution:

1.

Information must be reserved:

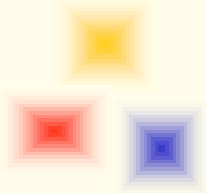


Checking method while transaction ends:

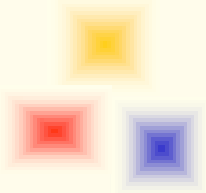
→ T

7.11 Locking in DDBMS

-
-
-
-
-



-
-
-

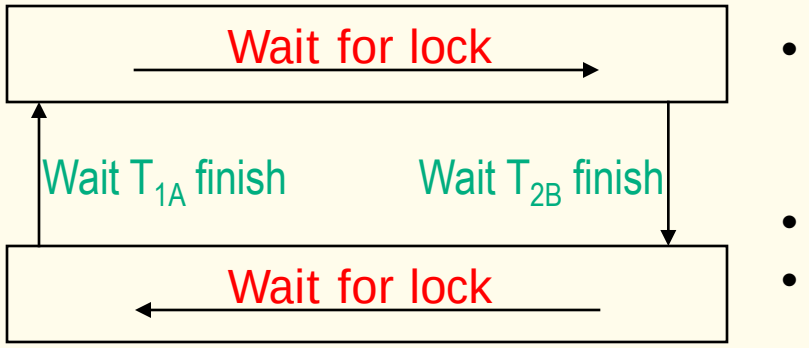


n

7.11.3 k-out-of-n locking

-
-
-
- ✓
- ✓
- ✓
- ✓

7.11.5 Global Deadlock



-
-
-

Global wait-for graph:

EXT → T into the graph; if T is the tail of wait current site, add T → EXT into the graph.

Processing method of global wait-for graph:

If on some site has: EXT→T →T → →T →EXT

Check other sites if has: EXT→T →T → →T →EXT

≠T

EXT→T →T → →T →T → →T →EXT

7.12 Time Stamp Technique in DDBMS

-
-
-