

CSE 4125: Distributed Database Systems

Chapter – 3 : Part D

Levels of Distributed Transparency

Distribution Transparency

The property of DDB by which the internal details of the distribution are hidden from the users (i.e. application programmer).

- A transparent system “*hides*” the implementation details from users.
- The advantage of a *fully transparent DDB* is the high level of support that it provides for the development of complex applications.

Levels of Distribution Transparency:

–Levels at which an application programmer view the DDB, depending on how much distribution transparency is provided by the DDBMS.

Important levels are –

- i. Level-1: Fragmentation transparency.*
- ii. Level-2: Location transparency.*
- iii. Level-3: Local mapping transparency.*

Level-1: Fragmentation Transparency:

- Programmer works on global relation.
- Fragmentation information is hidden.

Enables Programmer to query upon any relation as if it *were not fragmented*.

Availability to programmer	Global relation	Fragmentation	Location	Local mapping
	yes	n/a	n/a	n/a

Example

Global schema:

SUPPLIER (SNUM, NAME, CITY)

Level-2: Location Transparency:

- Fragmentation information is provided. Programmer works on fragments.
- Location (i.e. site name) information is hidden.

Enables Programmer to query upon fragments as if they were stored *locally in the user's site*.

Availability to programmer	Global relation	Fragmentation	Location	Local mapping
	yes	yes	n/a	n/a

Example

Global schema:

SUPPLIER (SNUM, NAME, CITY)

Fragmentation schema:

SUPPLIER₁ = SL_{CITY = “DHK”}(SUPPLIER)

SUPPLIER₂ = SL_{CITY = “CTG”}(SUPPLIER)

Level-3: Local Mapping Transparency:

- Location information is provided. Programmer works on fragmentation at specific location (site).
- Local DBMS information is hidden.

Enables Programmer to query upon fragments at a site as if the local DBMS is “known” (i.e. Oracle or MySQL) .

Availability to programmer	Global relation	Fragmentation	Location	Local mapping
	yes	yes	yes	n/a

Example

Global schema:

SUPPLIER (SNUM, NAME, CITY)

Fragmentation schema:

$SUPPLIER_1 = SL_{CITY = \text{“DHK”}}(SUPPLIER)$

$SUPPLIER_2 = SL_{CITY = \text{“CTG”}}(SUPPLIER)$

Allocation schema:

$SUPPLIER_1$ @ site 1.

$SUPPLIER_2$ @ site 2, 3.

Distribution transparency for read-only application.

Objective

We analyze with an example the different levels of distribution transparency:

- Level 1: Fragmentation transparency.
- Level 2: Location transparency.
- Level 3: Local mapping transparency.

For a *read-only* application.

Scenario

Global schema:

SUPPLIER (SNUM, NAME, CITY)

Fragmentation schema:

SUPPLIER₁ = SL_{CITY = "DHK"}(SUPPLIER)

SUPPLIER₂ = SL_{CITY = "CTG"}(SUPPLIER)

Allocation schema:

SUPPLIER₁ @ site 1.

SUPPLIER₂ @ site 2, 3.

Assume, a SUPINQUIRY application –

Print NAME of a given SNUM.

Example

SUPPLIER

SNUM	NAME	CITY
1	A	DHK
2	B	CTG
3	C	DHK
4	D	CTG

SUPPLIER₁

SNUM	NAME	CITY
1	A	DHK
3	C	DHK

1

SUPPLIER₂

SNUM	NAME	CITY
2	B	CTG
4	D	CTG

2

SUPPLIER₂

SNUM	NAME	CITY
2	B	CTG
4	D	CTG

3

Basic Structure

Reading a value from terminal and assigning it to a variable:

```
read(terminal, $ SNUM);
```

Query: Get *NAME* for a given *SNUM*. Example –

```
Select NAME into $ NAME  
from SUPPLIER [ @siteNumber ]  
where SNUM = $ SNUM;
```

Writing a value of a variable to terminal:

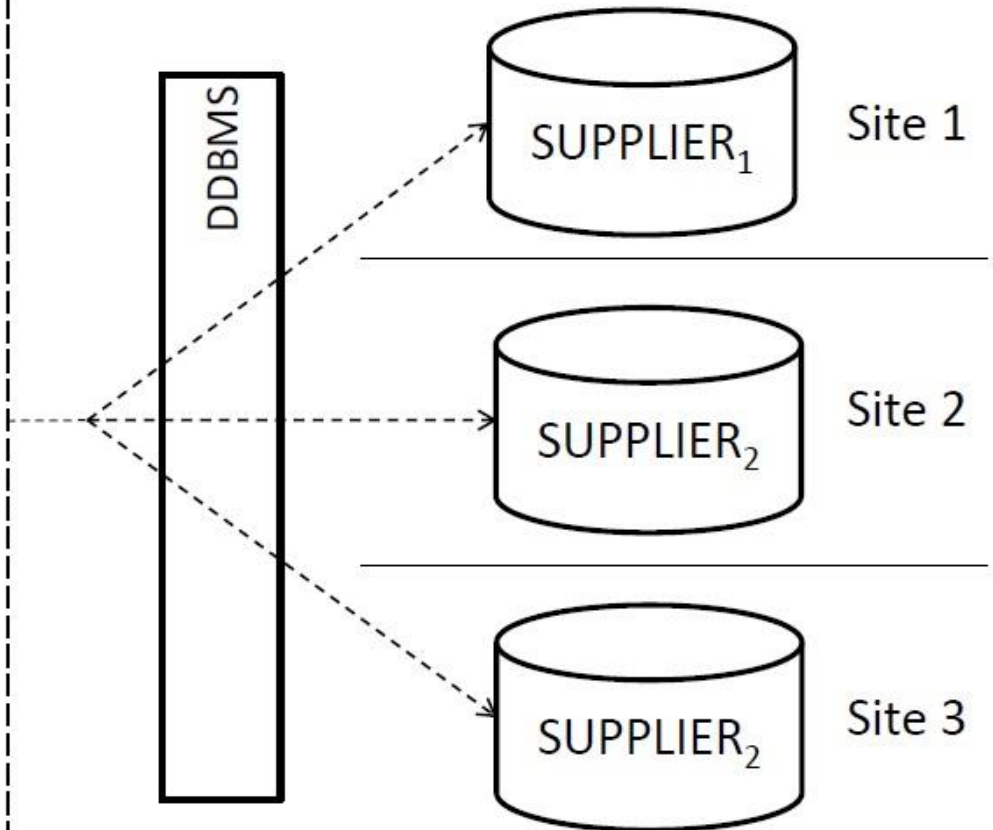
```
write(terminal, $ NAME);
```

Analyzing Level – 1 transparency

SUPINQUIRY

Hint:

Use global relation only.
Because fragmentation
information is hidden.



Level – 1 Transparency

Reading a value from terminal and assigning it to a variable:

```
read(terminal, $ SNUM);
```

Query: Get *NAME* for a given *SNUM*. Example –

```
Select NAME into $ NAME  
from SUPPLIER  
where SNUM = $ SNUM;
```

Writing a value of a variable to terminal:

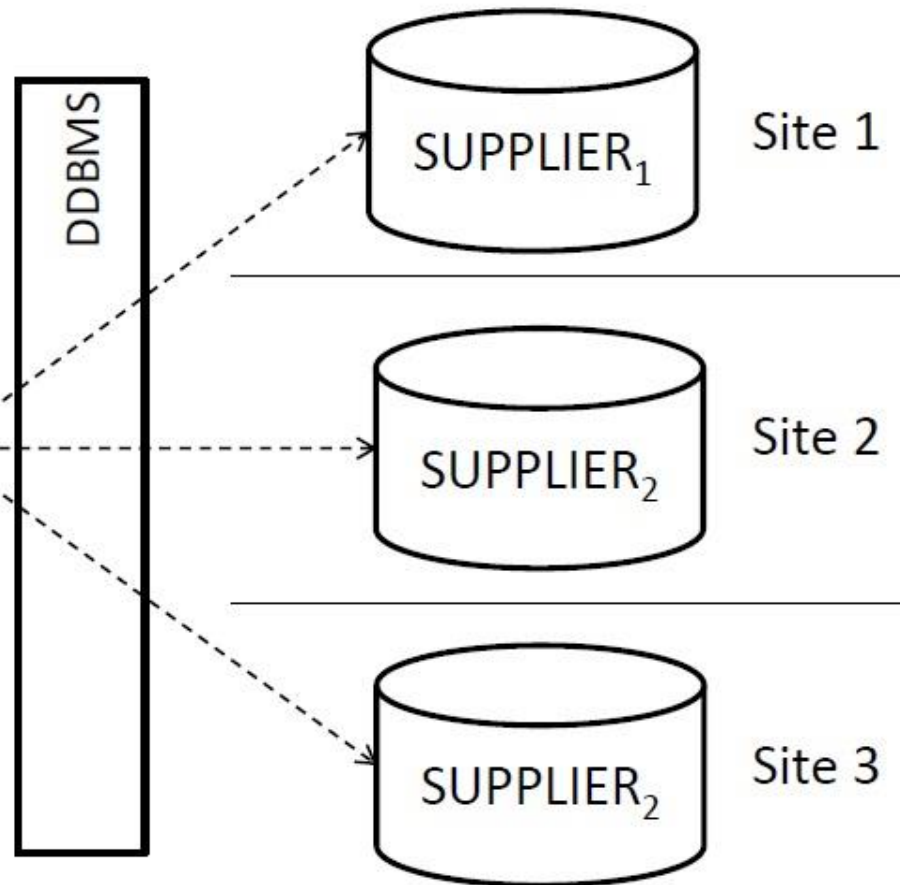
```
write(terminal, $ NAME);
```


SUPINQUIRY

read(*terminal*, \$ *SNUM*);

Select *NAME* into \$ *NAME*
from *SUPPLIER* where
SNUM = \$ *SNUM*;

write(*terminal*, \$ *NAME*).



Example

read(*terminal*, \$ *SNUM*);

Select *NAME* into \$ *NAME*
from *SUPPLIER* where
SNUM = \$ *SNUM*;

write(*terminal*, \$ *NAME*).

*SUPPLIER*₁

SNUM	NAME	CITY
1	A	DHK
3	C	DHK

1

*SUPPLIER*₂

SNUM	NAME	CITY
2	B	CTG
4	D	CTG

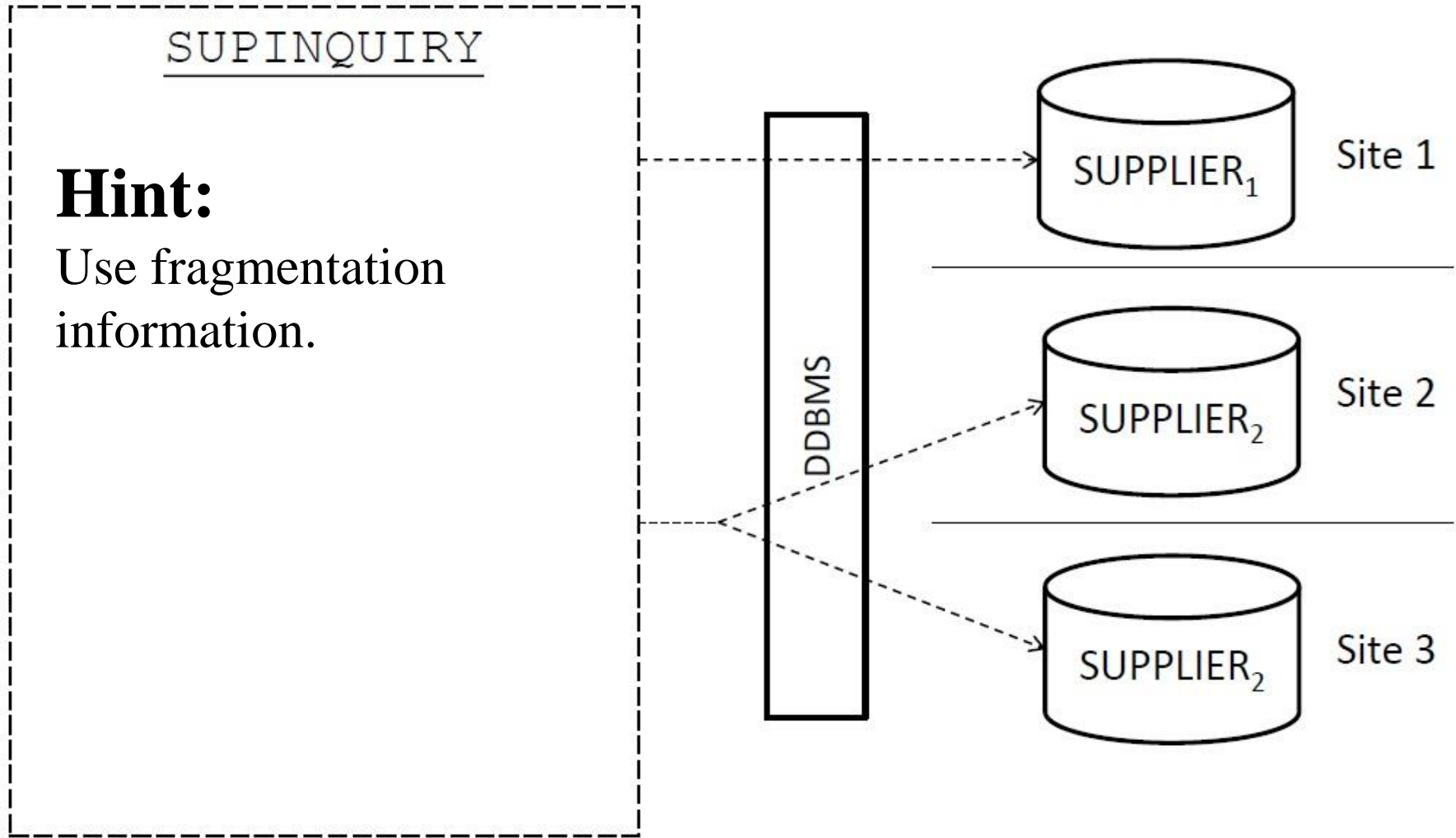
2

*SUPPLIER*₂

SNUM	NAME	CITY
2	B	CTG
4	D	CTG

3

Analyzing Level – 2 transparency



Level – 2 Transparency

Reading a value from terminal and assigning it to a variable:
read(terminal, \$ *SNUM*);

Query: Get *NAME* for a given *SNUM*. Example –

```
Select NAME into $ NAME
from SUPPLIER1
where SNUM = $ SNUM;
if not #FOUND then
  Select NAME into $ NAME
  from SUPPLIER2
  where SNUM = $ SNUM;
```

Writing a value of a variable to terminal:
write(terminal, \$ *NAME*);

SUPINQUIRY

read(*terminal*, \$ *SNUM*);

 Select *NAME* into \$ *NAME*
 from *SUPPLIER*₁
 where *SNUM* = \$ *SNUM*;
if not #FOUND then
 Select *NAME* into \$ *NAME*
 from *SUPPLIER*₂
 where *SNUM* = \$ *SNUM*;

write(*terminal*, \$ *NAME*).



Example

```
read(terminal, $ SNUM);
```

```
    Select NAME into $ NAME  
    from SUPPLIER1  
    where SNUM = $ SNUM;  
if not #FOUND then  
    Select NAME into $ NAME  
    from SUPPLIER2  
    where SNUM = $ SNUM;
```

```
write(terminal, $ NAME).
```

SUPPLIER₁

SNUM	NAME	CITY
1	A	DHK
3	C	DHK

1

SUPPLIER₂

SNUM	NAME	CITY
2	B	CTG
4	D	CTG

2

SUPPLIER₂

SNUM	NAME	CITY
2	B	CTG
4	D	CTG

3

Analyzing Level – 3 transparency

SUPINQUIRY

Hint:

Use fragmentation information + location information (i.e. site numbers) .



Site 1



Site 2



Site 3

Level – 3 Transparency

Reading a value from terminal and assigning it to a variable:
read(terminal, \$ *SNUM*);

Query: Get *NAME* for a given *SNUM*. Example –

```
Select NAME into $ NAME
from SUPPLIER1 AT SITE 1
where SNUM = $ SNUM;
if not #FOUND then
  Select NAME into $ NAME
  from SUPPLIER2 AT SITE 3
  where SNUM = $ SNUM;
```

Writing a value of a variable to terminal:
write(terminal, \$ *NAME*);

SUPINQUIRY

read(*terminal*, \$ *SNUM*);

 Select *NAME* into \$ *NAME*
 from *SUPPLIER*₁ AT *SITE 1*
 where *SNUM* = \$ *SNUM*;

if not #FOUND then

 Select *NAME* into \$ *NAME*
 from *SUPPLIER*₂ AT *SITE 3*
 where *SNUM* = \$ *SNUM*;

write(*terminal*, \$ *NAME*).

DDBMS

SUPPLIER₁

Site 1

SUPPLIER₂

Site 2

SUPPLIER₂

Site 3

Example

```
read(terminal, $ SNUM);
```

```
    Select NAME into $ NAME  
    from SUPPLIER1 AT SITE 1  
    where SNUM = $ SNUM;  
if not #FOUND then  
    Select NAME into $ NAME  
    from SUPPLIER2 AT SITE 3  
    where SNUM = $ SNUM;
```

```
write(terminal, $ NAME).
```

SUPPLIER₁

SNUM	NAME	CITY
1	A	DHK
3	C	DHK

1

SUPPLIER₂

SNUM	NAME	CITY
2	B	CTG
4	D	CTG

2

SUPPLIER₂

SNUM	NAME	CITY
2	B	CTG
4	D	CTG

3

Practice

See the provided practice pdf.