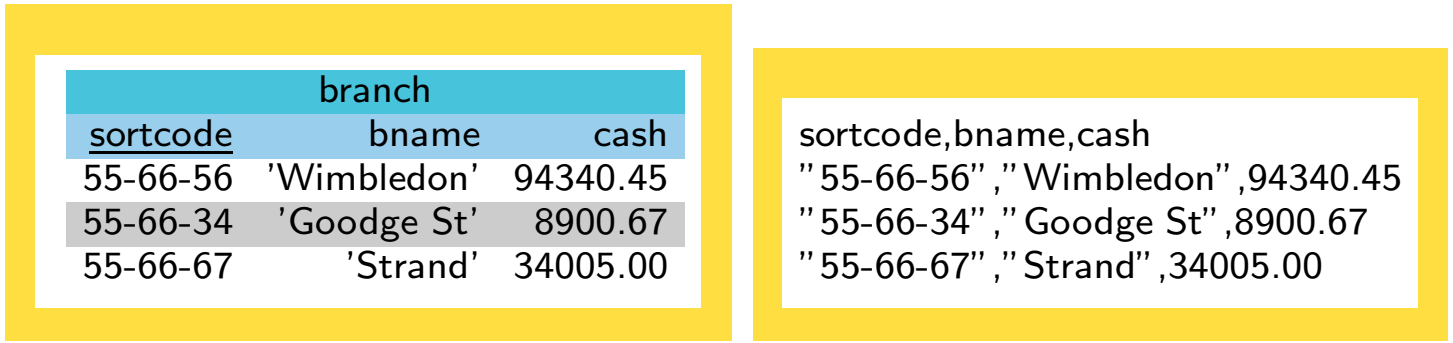
Distributed Databases

P.J. McBrien

Imperial College London

Distributed Heterogeneous Databases: Physical Differences

Database 1	Database 2



branch		
sortcode	bname	cash
55-66-56	'Wimbledon'	94340.45
55-66-34	'Goodge St'	8900.67
55-66-67	'Strand'	34005.00

```
sortcode,bname,cash
"55-66-56","Wimbledon",94340.45
"55-66-34","Goodge St",8900.67
"55-66-67","Strand",34005.00
```

RDBMS	CSV file
Query using SQL	Query by reading file

- Need **common data model** (**CDM**) *e.g.* Relational, ER

Distributed Heterogeneous Databases: Semantic Differences

Database 1	Database 2

branch

sortcode	bname	cash
56	'Wimbledon'	94340.45
34	'Goodge St'	8900.67
67	'Strand'	34005.00

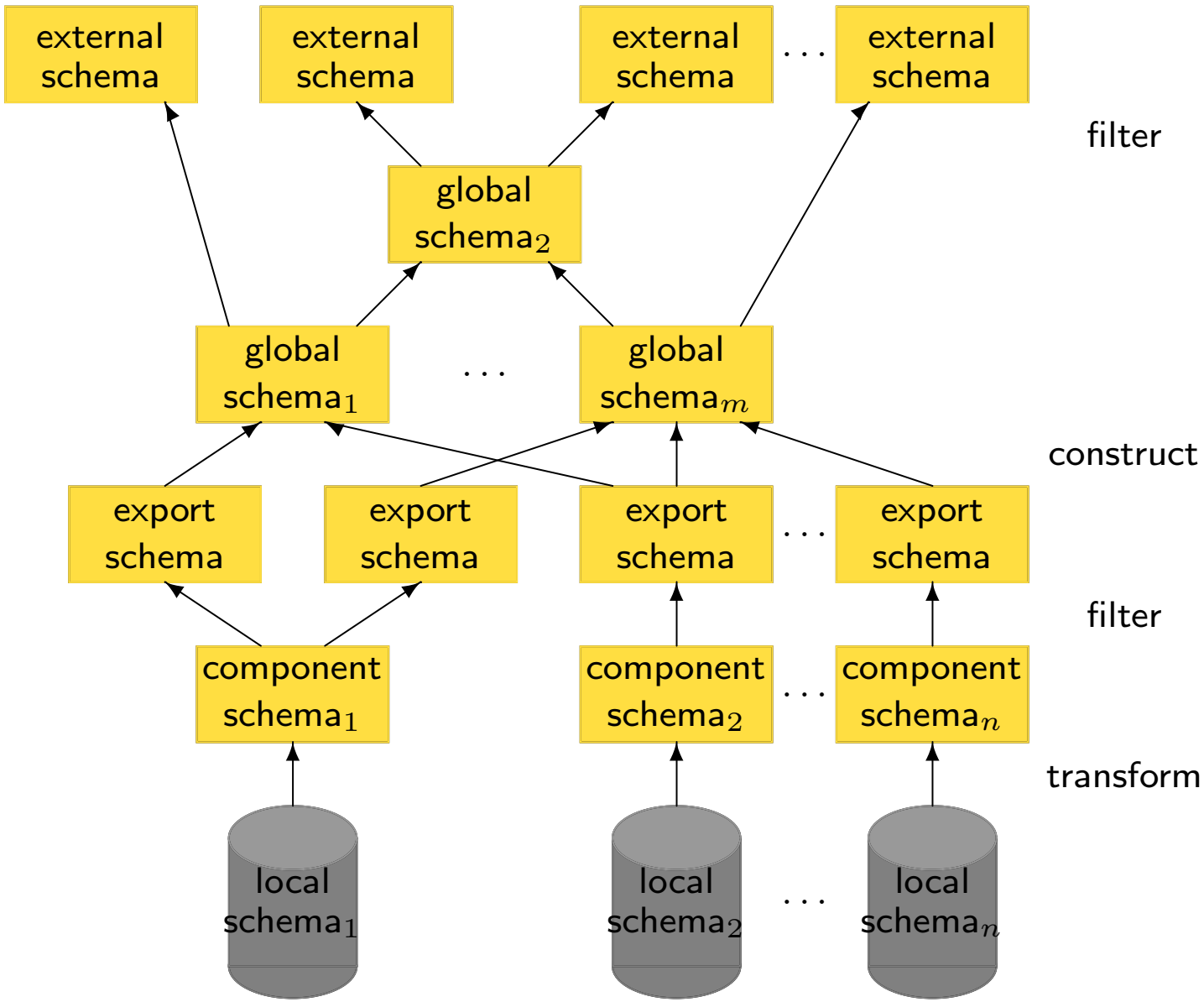
branch

sortcode	branchname	cash
55-66-56	'Wimbledon'	94340.45
55-66-34	'Goodge St'	8900.67
55-66-67	'Strand'	34005.00

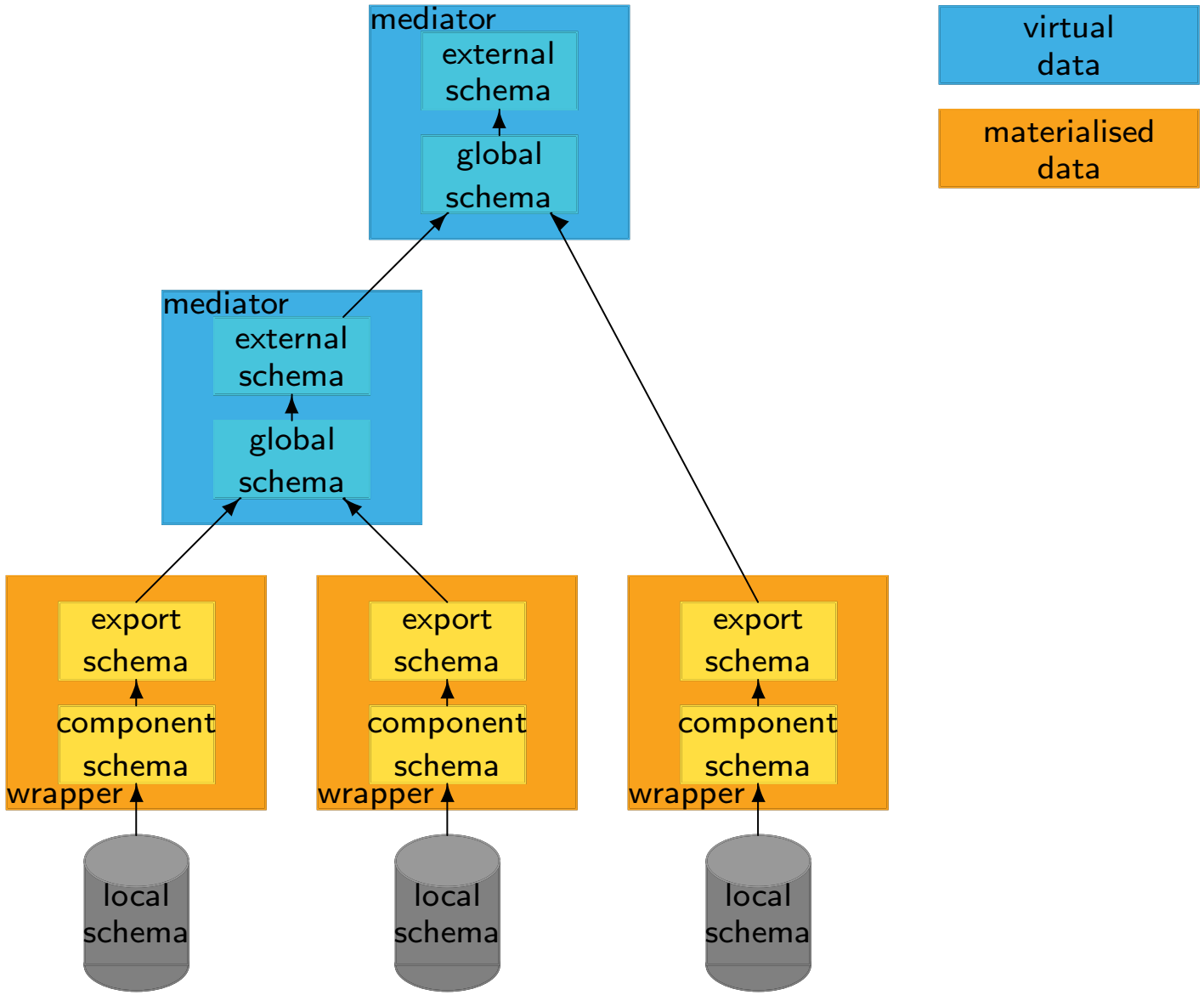
assume 55-66- sortcode	full sortcode
call branches bname	call branches branchname

- Need to perform **schema integration**

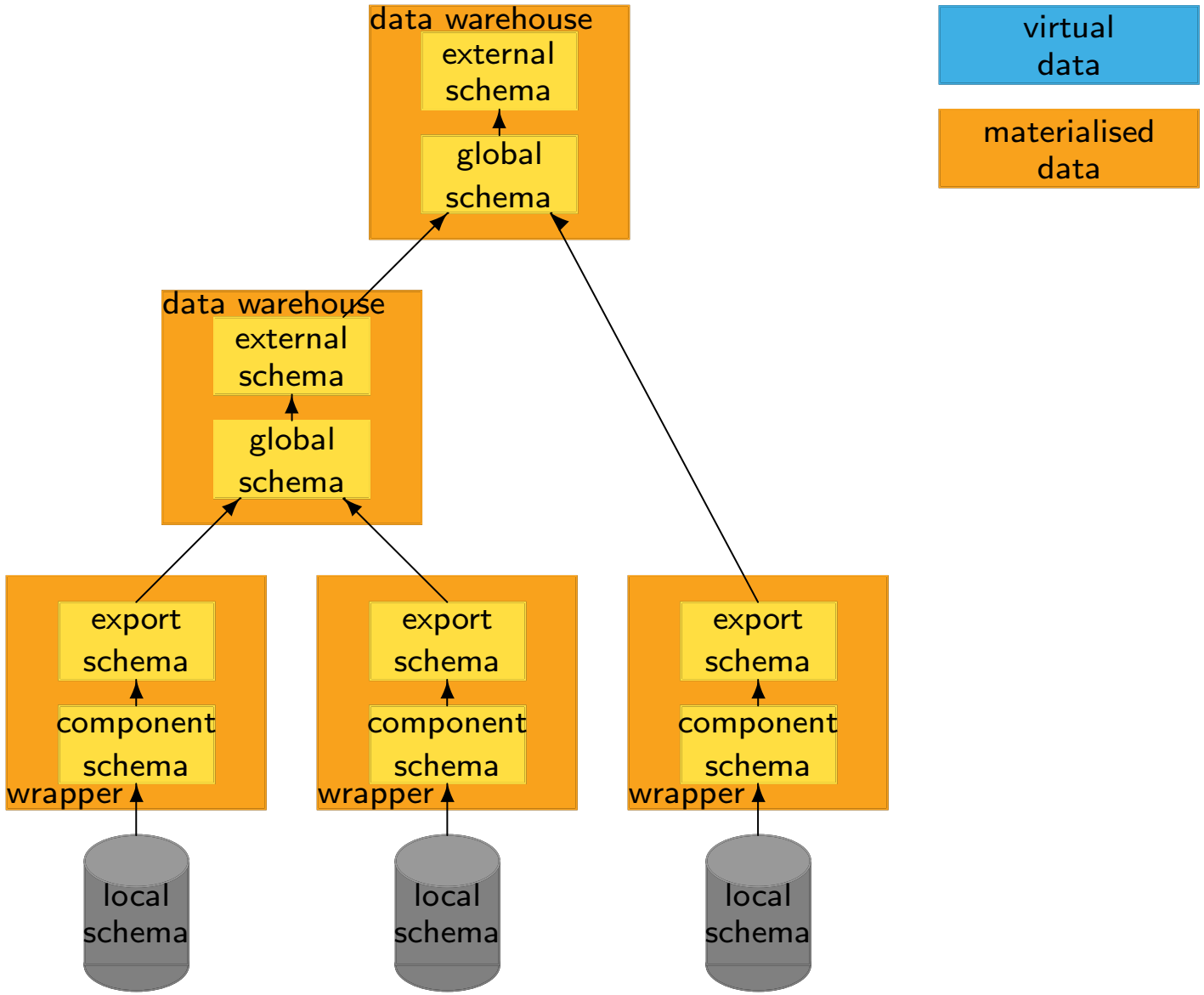
Logical Model for Heterogeneous Databases



Operational Model: Mediator Architecture



Operational Model: Data Warehouses



Topics to Cover in DDB

Operational Aspects

- How might data and hence queries be distributed between local schemas?
- How might a query on the global schema be optimised to run over the local schemas?
- How are ACID properties of transactions maintained in DDB?

Logical Aspects

- How are export schemas integrated to form a global schema?

Data Distribution

branch		
sortcode	bname	cash
56	'Wimbledon'	94340.45
34	'Goodge St'	8900.67
67	'Strand'	34005.00

movement			
mid	no	amount	tdate
1000	100	2300.00	5/1/1999
1001	101	4000.00	5/1/1999
1002	100	-223.45	8/1/1999
1004	107	-100.00	11/1/1999
1005	103	145.50	12/1/1999
1006	100	10.23	15/1/1999
1007	107	345.56	15/1/1999
1008	101	1230.00	15/1/1999
1009	119	5600.00	18/1/1999

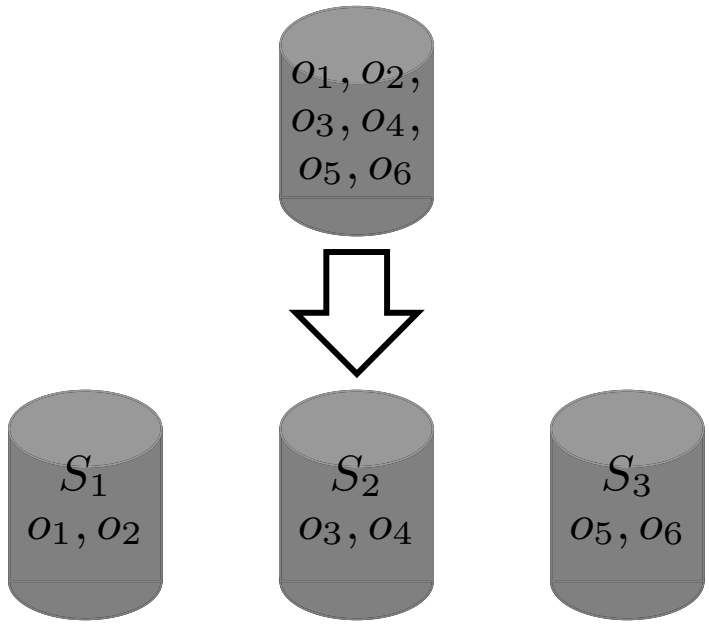
account				
no	type	cname	rate	sortcode
100	'current'	'McBrien, P.'	NULL	67
101	'deposit'	'McBrien, P.'	5.25	67
103	'current'	'Boyd, M.'	NULL	34
107	'current'	'Poulovassilis, A.'	NULL	56
119	'deposit'	'Poulovassilis, A.'	5.50	56
125	'current'	'Bailey, J.'	NULL	56

key branch(sortcode)
key branch(bname)
key movement(mid)
key account(no)

movement(no) $\stackrel{fk}{\Rightarrow}$ account(no)

account(sortcode) $\stackrel{fk}{\Rightarrow}$ branch(sortcode)

Data Distribution: Fragmentation



fragmentation

- split data between sites
- queries and updates must also be distributed
  $r[o_1], r[o_2], r[o_3] \rightarrow r[o_1], r[o_2]$ on S_1 and $r[o_3]$ on S_2

RDB: Horizontal Fragmentation

account

no	type	cname	rate	sortcode
100	'current'	'McBrien, P.'	NULL	67
103	'current'	'Boyd, M.'	NULL	34
107	'current'	'Poulovassilis, A.'	NULL	56
125	'current'	'Bailey, J.'	NULL	56

S_1

account

no	type	cname	rate	sortcode
101	'deposit'	'McBrien, P.'	5.25	67
119	'deposit'	'Poulovassilis, A.'	5.50	56

S_2

- $\text{account}_1 = \sigma_{\text{type}=\text{current}}\text{account}$
- $\text{account}_2 = \sigma_{\text{type}\neq\text{current}}\text{account}$

Quiz 1: Speed and Reliability of Horizontal Fragmentation

account				
no	type	cname	rate	sortcode
100	'current'	'McBrien, P.'	NULL	67
101	'deposit'	'McBrien, P.'	5.25	67
103	'current'	'Boyd, M.'	NULL	34
107	'current'	'Poulovassilis, A.'	NULL	56
119	'deposit'	'Poulovassilis, A.'	5.50	56
125	'current'	'Bailey, J.'	NULL	56

```
SELECT  no
FROM    account
WHERE   cname LIKE 'McBrien%'
```

If account is horizontally fragmented on no, compared with a single site version, is the DDB

A

Faster
Less Reliable

B

Faster
More Reliable

C

Slower
Less Reliable

D

Slower
More Reliable

Quiz 2: Horizontal fragmentation attributes

account				
no	type	cname	rate	sortcode
100	'current'	'McBrien, P.'	NULL	67
101	'deposit'	'McBrien, P.'	5.25	67
103	'current'	'Boyd, M.'	NULL	34
107	'current'	'Poulovassilis, A.'	NULL	56
119	'deposit'	'Poulovassilis, A.'	5.50	56
125	'current'	'Bailey, J.'	NULL	56

Which is the worst attribute to fragment on?

A
no

B
type

C
cname

D
rate

RDB: Derived Horizontal Fragmentation

movement

mid	no	amount	tdate
1000	100	2300.00	5/1/1999
1002	100	-223.45	8/1/1999
1005	103	145.50	12/1/1999
1006	100	10.23	15/1/1999
1004	107	-100.00	11/1/1999
1007	107	345.56	15/1/1999

S_1

movement

mid	no	amount	tdate
1001	101	4000.00	5/1/1999
1008	101	1230.00	15/1/1999
1009	119	5600.00	18/1/1999

S_2

- $\text{movement}_i = \text{movement} \ltimes \text{account}_i$

RDB: Vertical Fragmentation

account

no	type	rate	sortcode
100	'current'	NULL	67
101	'deposit'	5.25	67
103	'current'	NULL	34
107	'current'	NULL	56
119	'deposit'	5.50	56
125	'current'	NULL	56

S_1

account

no	cname
100	'McBrien, P.'
101	'McBrien, P.'
103	'Boyd, M.'
107	'Poulovassilis, A.'
119	'Poulovassilis, A.'
125	'Bailey, J.'

S_2

- $\text{account}_1 = \pi_{\text{no,type,rate,sortcode}}\,\text{account}$
- $\text{account}_2 = \pi_{\text{no,cname}}\,\text{account}$

Quiz 3: Vertical Fragmention

branch		
sortcode	bname	cash
56	'Wimbledon'	94340.45
34	'Goodge St'	8900.67
67	'Strand'	34005.00

key(bname)
key(sortcode)

Which of the following is a correct vertical fragmentation of branch?

A

$\pi_{\text{bname,cash}}$ branch
$\pi_{\text{sortcode,cash}}$ branch

B

$\pi_{\text{bname,sortcode}}$ branch
π_{cash} branch

C

$\pi_{\text{bname,cash}}$ branch
π_{sortcode} branch

D

$\pi_{\text{bname,cash}}$ branch
$\pi_{\text{sortcode,bname}}$ branch

RA Definitions

Horizonal Fragmentation

$$R = R_1 \cup \ldots \cup R_n$$

- 'plain' horizontal fragmentation splits rows using σ
  $R_1 = \sigma_{P_1} R, \quad \ldots, \quad R_n = \sigma_{P_n} R$
- derived horizontal fragmentation splits rows using $\ltimes$
  $R_1 = R \ltimes S_1, \quad \ldots, \quad R_n = R \ltimes S_n$
  $R \ltimes S = R \bowtie \pi_{R \cap S}(S)$

Vertical Fragmentation

$$R = R_1 \bowtie \ldots \bowtie R_n$$

- vertical fragmentation splits rows using π
  $R_1 = \pi_{attrs_1} R_1, \quad \ldots, \quad R_n = \pi_{attrs_n} R$
- A **loss-less join** decomposition

Worksheet: Fragmentation

account				
no	type	cname	rate	sortcode
100	'current'	'McBrien, P.'	NULL	67
101	'deposit'	'McBrien, P.'	5.25	67
103	'current'	'Boyd, M.'	NULL	34

S_1

account				
no	type	cname	rate	sortcode
107	'current'	'Poulovassilis, A.'	NULL	56
119	'deposit'	'Poulovassilis, A.'	5.50	56
125	'current'	'Bailey, J.'	NULL	56

S_2

Worksheet: Fragmentation

account		
no	type	sortcode
100	'current'	67
101	'deposit'	67
103	'current'	34
107	'current'	56
119	'deposit'	56
125	'current'	56

S_1

account	
sortcode	cname
67	'McBrien, P.'
67	'McBrien, P.'
34	'Boyd, M.'
56	'Poulovassilis, A.'
56	'Poulovassilis, A.'
56	'Bailey, J.'

S_2

RDB: Hybrid Fragmentation

S_1

account			
no	type	rate	sortcode
100	'current'	NULL	67
103	'current'	NULL	34
107	'current'	NULL	56
125	'current'	NULL	56

S_3

account	
no	cname
100	'McBrien, P.'
103	'Boyd, M.'
107	'Poulovassilis, A.'
125	'Bailey, J.'

S_2

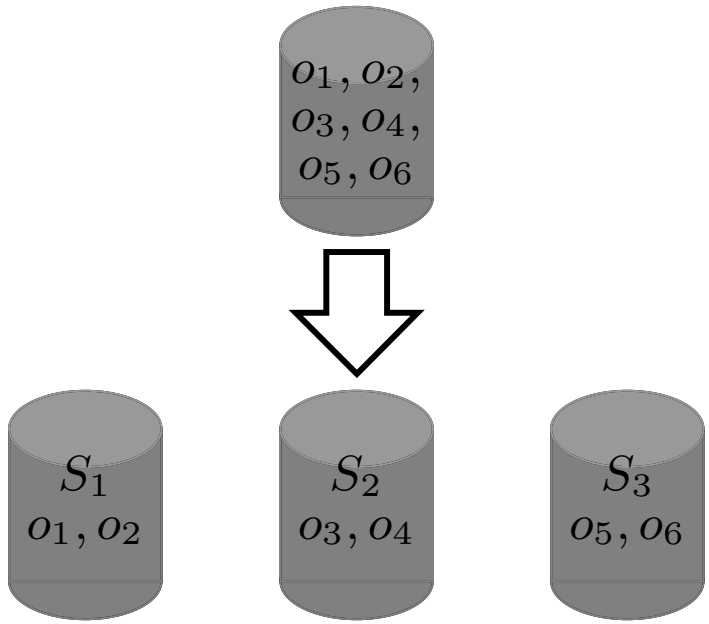
account			
no	type	rate	sortcode
101	'deposit'	5.25	67
119	'deposit'	5.50	56

S_4

account	
no	cname
101	'McBrien, P.'
119	'Poulovassilis, A.'

- $\text{account}_1 = \pi_{\text{no,type,rate,sortcode}}(\sigma_{\text{type=current}}\text{account}), \ldots$

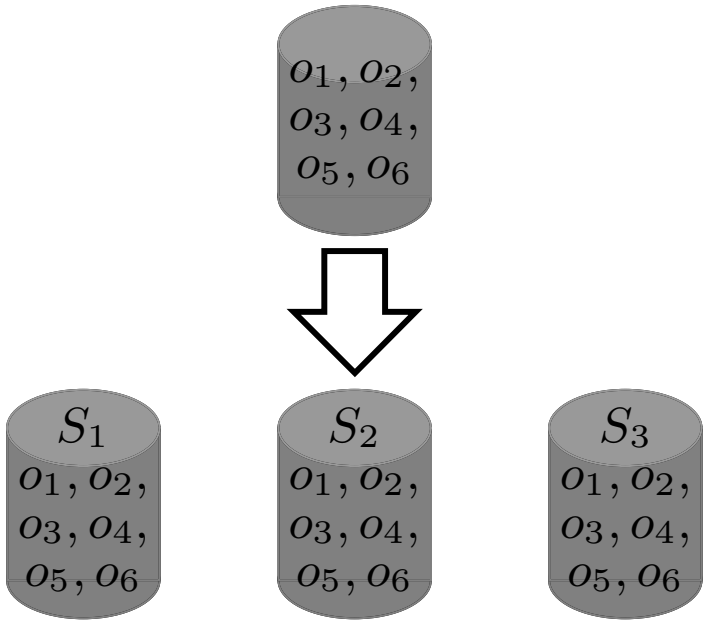
Data Distribution: Fragmentation



fragmentation

- split data between sites
- queries and updates must also be distributed
  $r[o_1], r[o_2], r[o_3] \rightarrow r[o_1], r[o_2]$ on S_1 and $r[o_3]$ on S_2

Data Distribution: Replication



replication

- copy data between sites
- queries may run on any site
  $r[o_1], r[o_2], r[o_3] \rightarrow r[o_1], r[o_2], r[o_3]$ on S_1 (or on S_2 or on S_3)
- updates write on all sites
  $r[o_1], w[o_1] \rightarrow r[o_1], w[o_1]$ on S_1 and $w[o_1]$ on S_2 and S_3

Quiz 4: Speed and Reliability of Replication: Reads

account				
no	type	cname	rate	sortcode
100	'current'	'McBrien, P.'	NULL	67
101	'deposit'	'McBrien, P.'	5.25	67
103	'current'	'Boyd, M.'	NULL	34
107	'current'	'Poulovassilis, A.'	NULL	56
119	'deposit'	'Poulovassilis, A.'	5.50	56
125	'current'	'Bailey, J.'	NULL	56

```
SELECT  no
FROM    account
WHERE   cname LIKE 'McBrien%'
```

Considering queries like those above, if account is replicated on several sites, compared with a single site version, is the DDB

A

Faster
Less Reliable

B

Faster
More Reliable

C

Slower
Less Reliable

D

Slower
More Reliable

Quiz 5: Speed and Reliability of Replication: Writes

account				
no	type	cname	rate	sortcode
100	'current'	'McBrien, P.'	NULL	67
101	'deposit'	'McBrien, P.'	5.25	67
103	'current'	'Boyd, M.'	NULL	34
107	'current'	'Poulovassilis, A.'	NULL	56
119	'deposit'	'Poulovassilis, A.'	5.50	56
125	'current'	'Bailey, J.'	NULL	56

```
UPDATE  account
SET     rate=2.0
WHERE   type='deposit'
```

Considering queries like those above, if account is replicated on several sites, compared with a single site version, is the DDB

A

Faster
Less Reliable

B

Faster
More Reliable

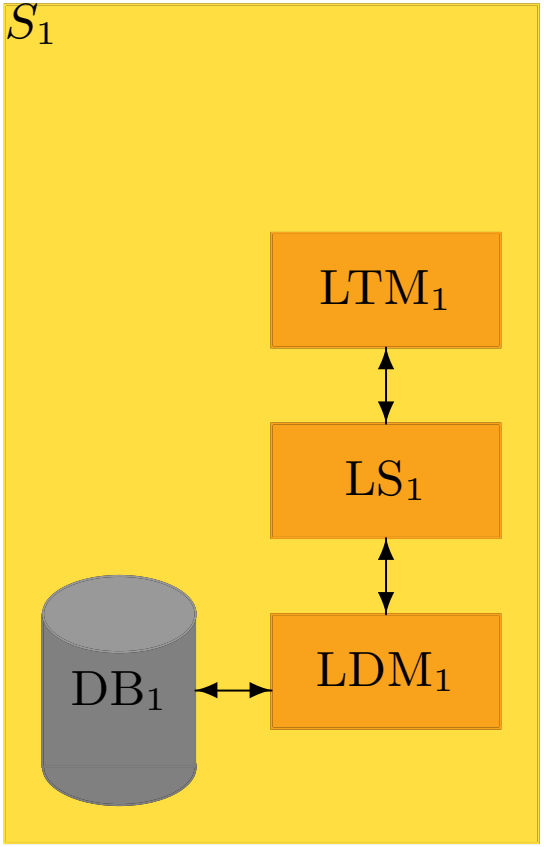
C

Slower
Less Reliable

D

Slower
More Reliable

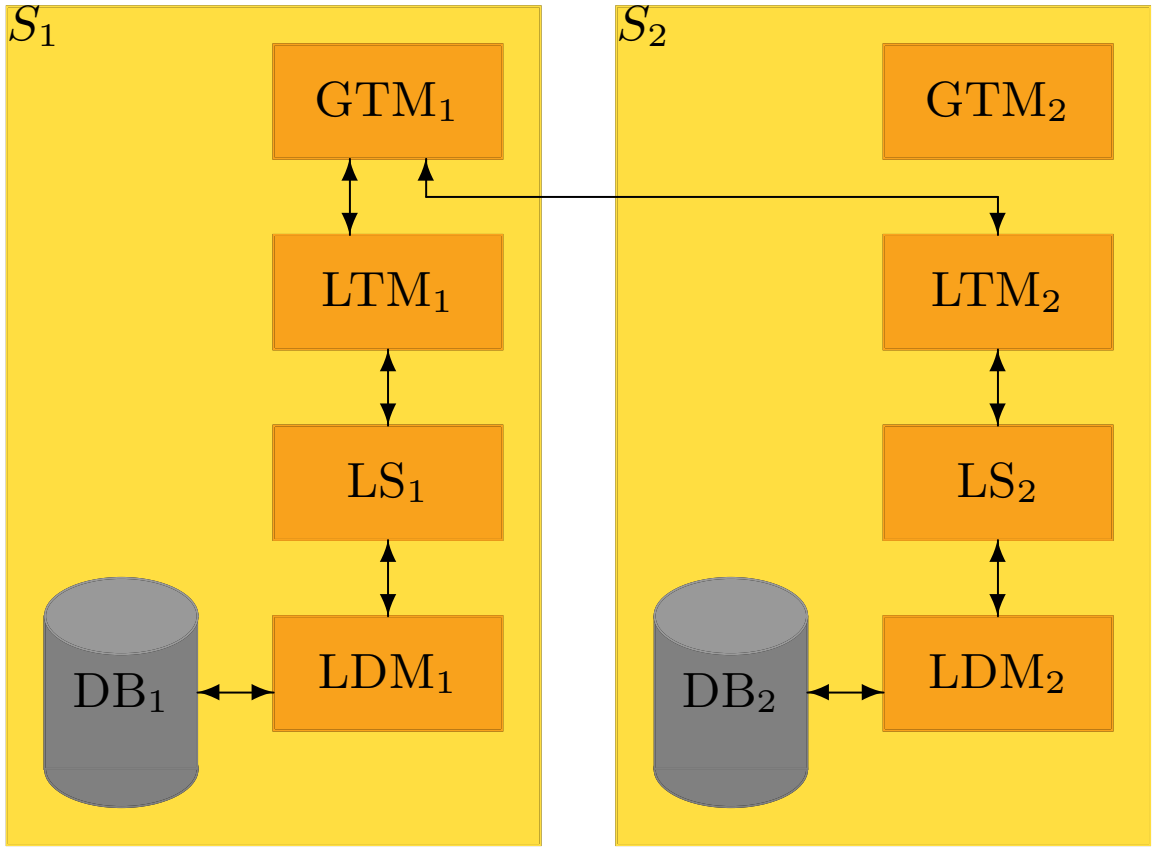
Distributed Query Processing



At each site:

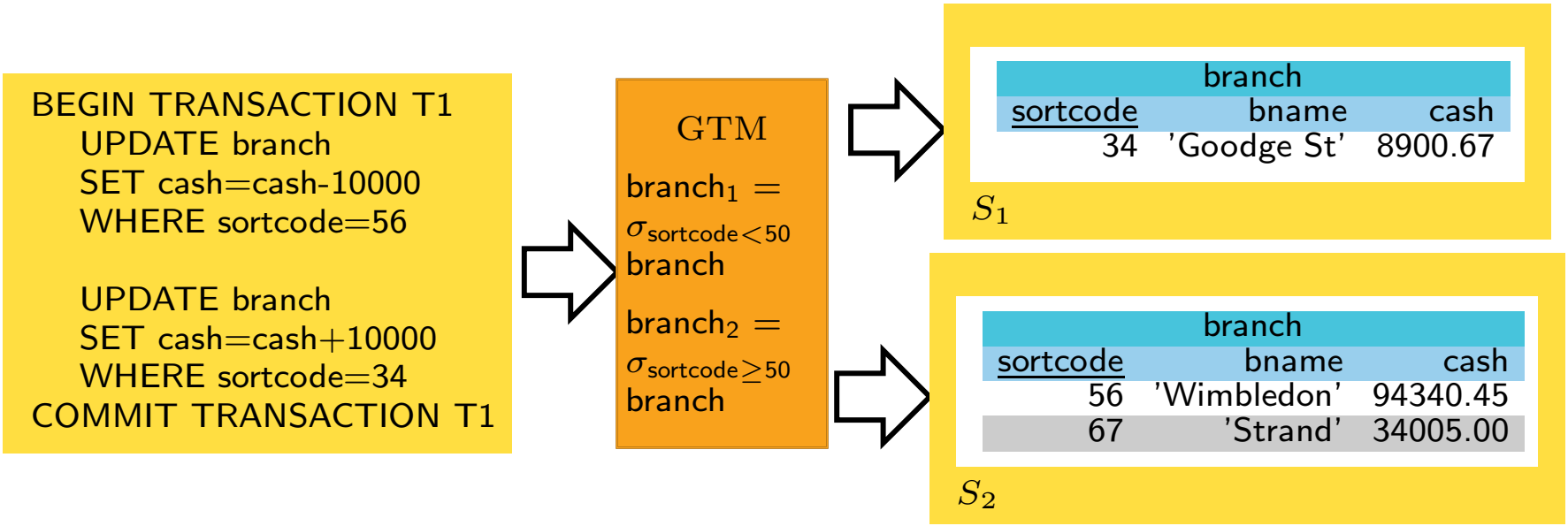
- Transaction Manager: query processor plans and translates queries to set of primitive operations
- Schedular: schedules the primitive operators to obey ACID properties
- Data Manager: efficient use of memory, maintain durability of transactions

Distributed Query Processing



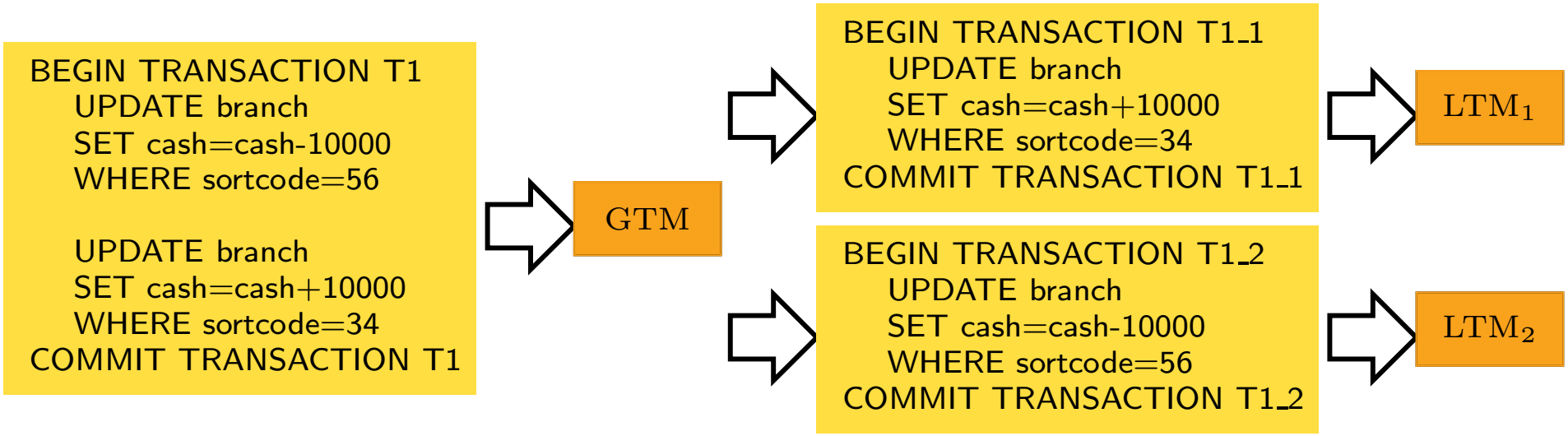
- TM contains **query processor** (**QP**)
- If DDB, QP must distribute query over fragments

Distribution of TP: GTM Plans Execution

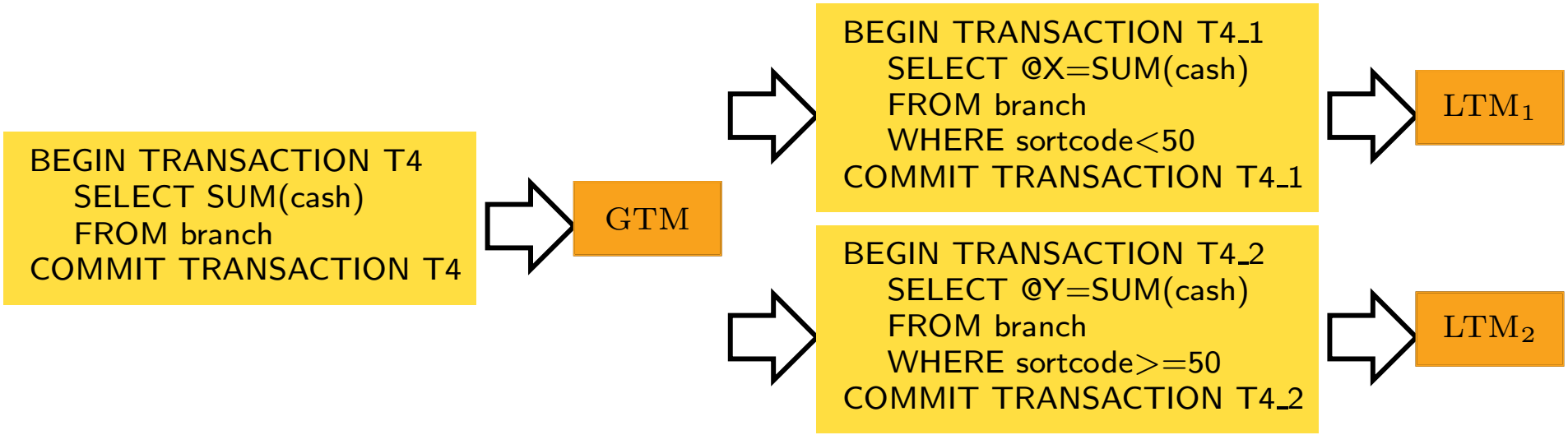


- GTM must transform transaction into sub transactions for each site
  - fragmentation → breakup transaction into fragments
  - replication → read only transactions goto one site, read/write transactions to all sites

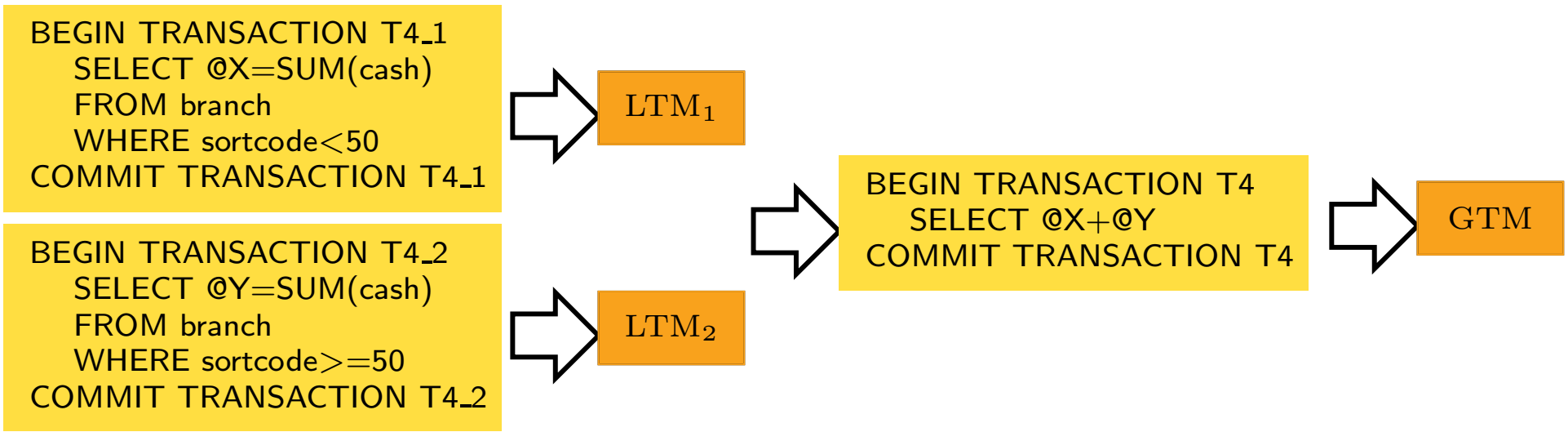
Distribution of TP: Example 1



Distribution of TP: Example 2



Distribution of TP: Example 2



Horizontal Fragmentation

Distributing a query over Horizontal Fragmentation

- Horizontal fragmentation decide using σ
- WHERE clause (σ) may determine only certain horizontal fragments are used for a query

account$_1$				
no	type	cname	rate	sortcode
100	'current'	'McBrien, P.'	NULL	67
103	'current'	'Boyd, M.'	NULL	34
107	'current'	'Poulovassilis, A.'	NULL	56
125	'current'	'Bailey, J.'	NULL	56

account$_2$				
no	type	cname	rate	sortcode
101	'deposit'	'McBrien, P.'	5.25	67
119	'deposit'	'Poulovassilis, A.'	5.50	56

- account$_1$ $= \sigma_{\text{type}=\text{'current'}}$,account
- account$_2$ $= \sigma_{\text{type}\neq\text{'current'}}$,account

Query Distribution

```
SELECT  account.rate
        branch.bname
FROM    account
        JOIN branch
        USING(sortcode)
WHERE   account.type='deposit'
AND     account.no=101
```

- Only send to S_2

Vertical Fragmentation

Distributing a query over Vertical Fragmentation

- Vertical Fragmentation decided using π
- SELECT clause (π) may determine only certain vertical fragments are used, but must ensure JOIN and WHERE clauses (σ) can be processed.

account$_1$			
no	type	rate	sortcode
100	'current'	NULL	67
101	'deposit'	5.25	67
103	'current'	NULL	34
107	'current'	NULL	56
119	'deposit'	5.50	56
125	'current'	NULL	56

account$_2$	
no	cname
100	'McBrien, P.'
101	'McBrien, P.'
103	'Boyd, M.'
107	'Poulovassilis, A.'
119	'Poulovassilis, A.'
125	'Bailey, J.'

Query Distribution

```
SELECT  account.rate
        branch.bname
FROM    account
        JOIN branch
        USING(sortcode)
WHERE   account.type='deposit'
AND     account.no=101
```

- Only send to S_1

- account$_1$ $= \pi_{\text{no,type,rate,sortcode}}$ account
- account$_2$ $= \pi_{\text{no,cname}}$ account

Worksheet: Distribution of Transactions

account

no	type	rate	sortcode
100	'current'	NULL	67
103	'current'	NULL	34
107	'current'	NULL	56
125	'current'	NULL	56

S_1

account

no	cname
100	'McBrien, P.'
103	'Boyd, M.'
107	'Poulovassilis, A.'
125	'Bailey, J.'

S_3

account

no	type	rate	sortcode
101	'deposit'	5.25	67
119	'deposit'	5.50	56

S_2

account

no	cname
101	'McBrien, P.'
119	'Poulovassilis, A.'

S_4

$$\text{account}_1 = \pi_{\text{no,type,rate,sortcode}}(\sigma_{\text{type}=\text{current}}\text{account})$$
$$\text{account}_2 = \pi_{\text{no,type,rate,sortcode}}(\sigma_{\text{type}\neq\text{current}}\text{account})$$
$$\text{account}_3 = \pi_{\text{no,cname}}(\sigma_{\text{type}=\text{current}}\text{account})$$
$$\text{account}_4 = \pi_{\text{no,cname}}(\sigma_{\text{type}\neq\text{current}}\text{account})$$

Worksheet: Distribution of Transactions (1)

LTM2:

```
BEGIN TRANSACTION TA_2
    UPDATE account
    SET    rate=5.5
    WHERE  type='deposit'
COMMIT TRANSACTION TA_2
```

GTM:

```
SELECT *
FROM   LTM2
```

Worksheet: Distribution of Transactions (2)

LTM3:

```
BEGIN TRANSACTION TB_3
    SELECT no
    FROM   account
    WHERE  cname='McBrien, P.'
COMMIT TRANSACTION TB_3
```

LTM4:

```
BEGIN TRANSACTION TB_4
    SELECT no
    FROM   account
    WHERE  cname='McBrien, P.'
COMMIT TRANSACTION TB_4
```

GTM:

```
SELECT *
FROM   LTM3
UNION ALL
SELECT *
FROM   LTM4
```

Worksheet: Distribution of Transactions (3)

LTM2

```
BEGIN TRANSACTION TC_2
    SELECT no,rate
    FROM   account
    WHERE  type='deposit'
COMMIT TRANSACTION TC_2
```

LTM4

```
BEGIN TRANSACTION TC_4
    SELECT no,cname
    FROM   account
COMMIT TRANSACTION TC_4
```

GTM

```
SELECT *
FROM   LTM2 NATURAL JOIN LTM4
```

Worksheet: Distribution of Transactions (4)

LTM1

```
BEGIN TRANSACTION TD_1
    SELECT COALESCE(SUM(rate),0.0) AS total_rate,
           COUNT(rate) AS no_rate
    FROM   account
COMMIT TRANSACTION TD_1
```

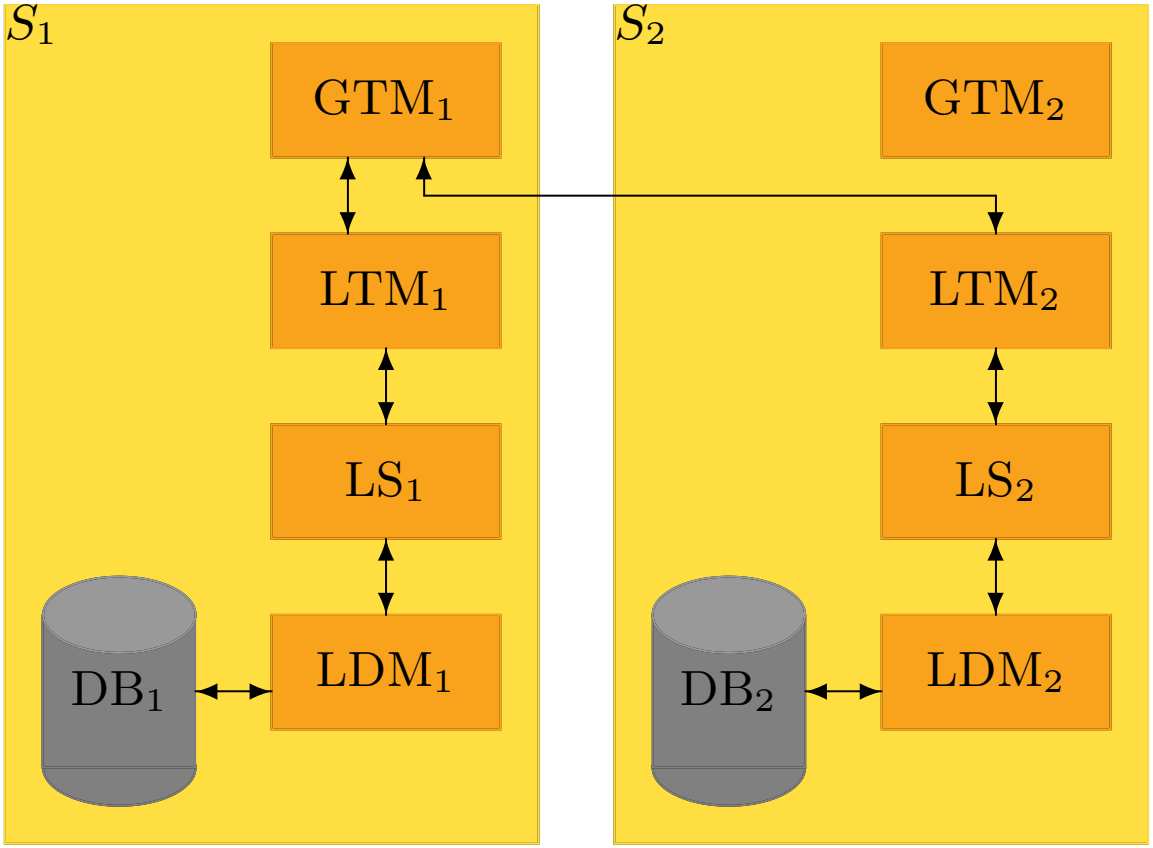
LTM2

```
BEGIN TRANSACTION TD_2
    SELECT COALESCE(SUM(rate),0.0) AS total_rate,
           COUNT(rate) AS no_rate
    FROM   account
COMMIT TRANSACTION TD_2
```

GTM:

```
SELECT (LTM1.total_rate+LTM2.total_rate)/
              (LTM1.no_rate+LTM2.no_rate)
FROM   LTM1,LTM2
```

Distributed Query Processing



Optimising a Distributed Query

Slow part of distributed query processing is network communication

Executing Distributed Joins in a DDB

S_1

account

no	type	...	sortcode
100	'current'		67
101	'deposit'		67
103	'current'		34
107	'current'		56
119	'deposit'		56
125	'current'		56

S_2

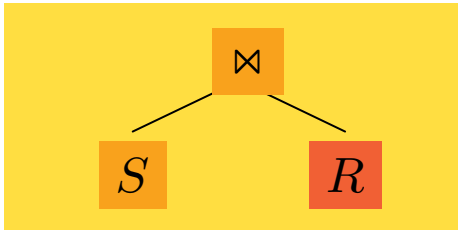
movement

mid	no	amount	tdate
1000	100	2300.00	5/1/1999
1001	101	4000.00	5/1/1999
1002	100	-223.45	8/1/1999
1004	107	-100.00	11/1/1999
1005	103	145.50	12/1/1999
1006	100	10.23	15/1/1999
1007	107	345.56	15/1/1999
1008	101	1230.00	15/1/1999
1009	119	5600.00	18/1/1999

CREATE INDEX ON movement(no)

Suppose we want to execute at S_1
account $\bowtie$ movement

Cost of Index Nested Loop Joins in DDB



Index Nested Loop Join in a DDB

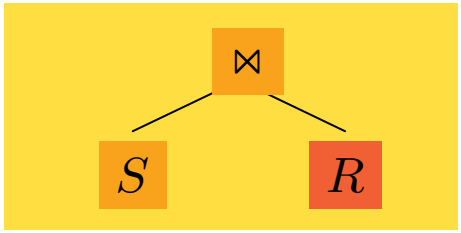
Iterate over rows of one table, and make a request for the matching row(s) from the other site

If S at S_1, and R at S_2, cost of $S \bowtie R$ at site S_1 is

$$c_0 \times |S| + c \times (RowSize(R \cap S) \times |S| + RowSize(R) \times |R \ltimes S|)$$

- c_0 reflects the cost of the overheads of a reply-answer communication
- $|S|$ counts the number of rows in S
- $RowSize(S)$ gives the size in bytes of one row of S
- c is the cost of transmitting one byte of data

Cost of Direct implementation of Joins



Direct Implementation of Join in a DDB

Fetch the entire table from the remote site, and perform join locally

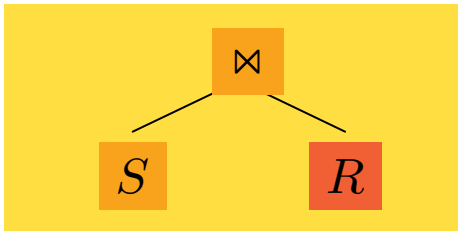
1. $S_1 \rightarrow S_2$: Request all rows in R
2. $S_2 \rightarrow S_1$: Reply with all rows of R
3. S_1: Compute $S \bowtie S$

If S at S_1, and R at S_2, cost of $S \bowtie R$ at site S_1 is

$$c_0 + c \times RowSize(R) \times |R|$$

- Only one request-reply needed
- Efficient if all (or most) rows from R required

Cost of Semi Join implementation of Joins



Semi Join implementation of Joins in a DDB

Send to remote site set of values to join onto, and preform join remotely, and then locally

1. $S_1 \rightarrow S_2$: Send $\pi_{Attr(S) \cap Atts(R)} S$
2. $S_2 \rightarrow S_1$: Reply with $R \ltimes \pi_{Attr(S) \cap Atts(R)} S$
3. S_1: Compute $S \bowtie R \ltimes \pi_{Attr(S) \cap Atts(R)} S \equiv S \bowtie R$

If S at S_1, and R at S_2, cost of $S \bowtie R$ at site S_1 is

$$c_0 + c \times (RowSize(R \cap S) \times |S| + RowSize(R) \times |R \ltimes S|)$$

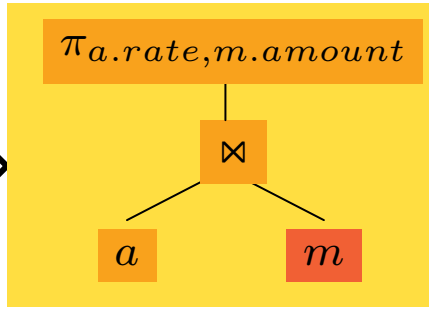
- Only one request-reply needed
- Works well if only some rows required from R

Example Comparison (1)

Assume rate, amount and no columns all occupy four bytes

S_1

```
SELECT account.rate,movement.amount
FROM   account
       JOIN movement
       USING (no)
```



Direct Join

$$
\begin{aligned}
S_2 \; to \; S_1 &= RowSize(\pi_{\text{no,amount}}\,\text{movement}) \times |\pi_{\text{no,amount}}\,\text{movement}| \\
&= (4+4) \times 9 \\
&= 72
\end{aligned}
$$

Semi-Join

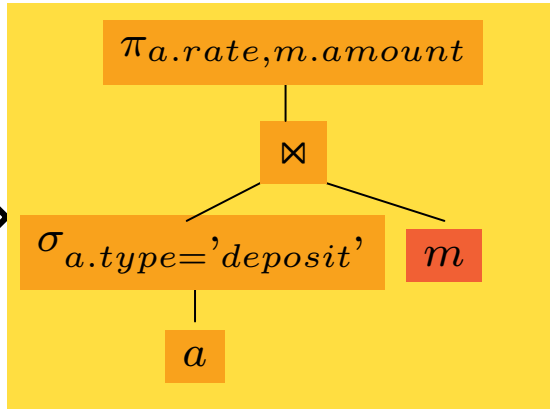
$$
\begin{aligned}
S_1 \; to \; S_2 &= RowSize(\pi_{\text{no}}\,\text{account}) \times |\pi_{\text{no}}\,\text{account}| \\
&= 4 \times 6 \\
&= 24 \\
S_2 \; to \; S_1 &= RowSize(\pi_{\text{no,amount}}\,\text{movement}) \times |\pi_{\text{no,amount}}\,\text{movement} \ltimes \text{account}| \\
&= (4+4) \times 9 \\
&= 72 \\
\textbf{total} &= 96
\end{aligned}
$$

Example Comparison (2)

Assume rate, amount and no columns all occupy four bytes

S_1

```
SELECT  account.rate,movement.amount
FROM    account
        JOIN movement
        USING (no)
WHERE   account.type='deposit'
```



Direct Join

$$
\begin{aligned}
S_2 \text{ to } S_1 &= RowSize(\pi_{\text{no,amount}}\,\text{movement}) \times |\pi_{\text{no,amount}}\,\text{movement}| \\
&= (4+4) \times 9 \\
&= 72
\end{aligned}
$$

Semi Join

$$
\begin{aligned}
S_1 \text{ to } S_2 &= RowSize(\pi_{\text{no}}\text{account}) \times |\pi_{\text{no}}\sigma_{\text{type='deposit'}}\text{account}| \\
&= 4 \times 2 \\
&= 8 \\
S_2 \text{ to } S_1 &= RowSize(\pi_{\text{no,amount}}\,\text{movement}) \times |\pi_{\text{no,amount}}\,\text{movement} \ltimes \sigma_{\text{type='deposit'}}\,\text{account}| \\
&= (4+4) \times 3 \\
&= 24 \\
\textbf{total} &= 32
\end{aligned}
$$

Worksheet: Distributed Joins

directory

telephone	name	charge
1000	Adams	10.00
1001	Jones	120.25
1002	Black	344.00
1003	Khan	243.50
1004	Smith	44.00
1005	Brown	-100.00
1006	Patel	200.00
1007	White	222.00

S_1

business

telephone	rate
1002	A
1004	B
1006	B

S_2

column	bytes
telephone	5
name	6
charge	4
rate	1