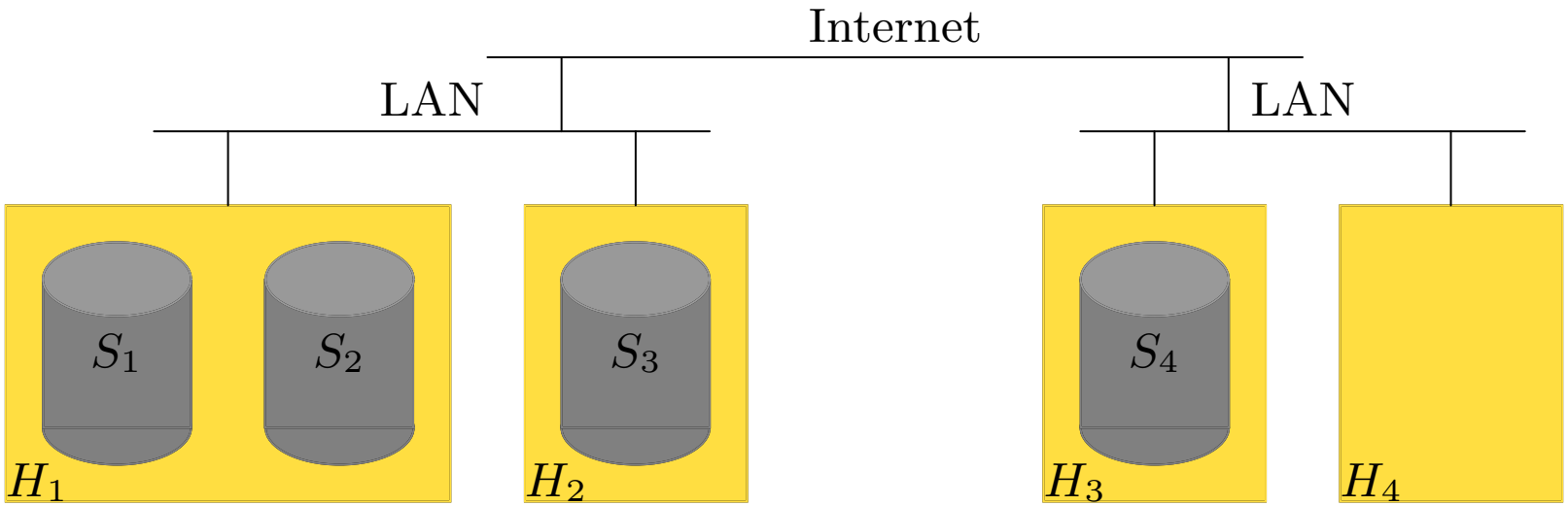
Distributed Query Processing

P.J. McBrien

Imperial College London

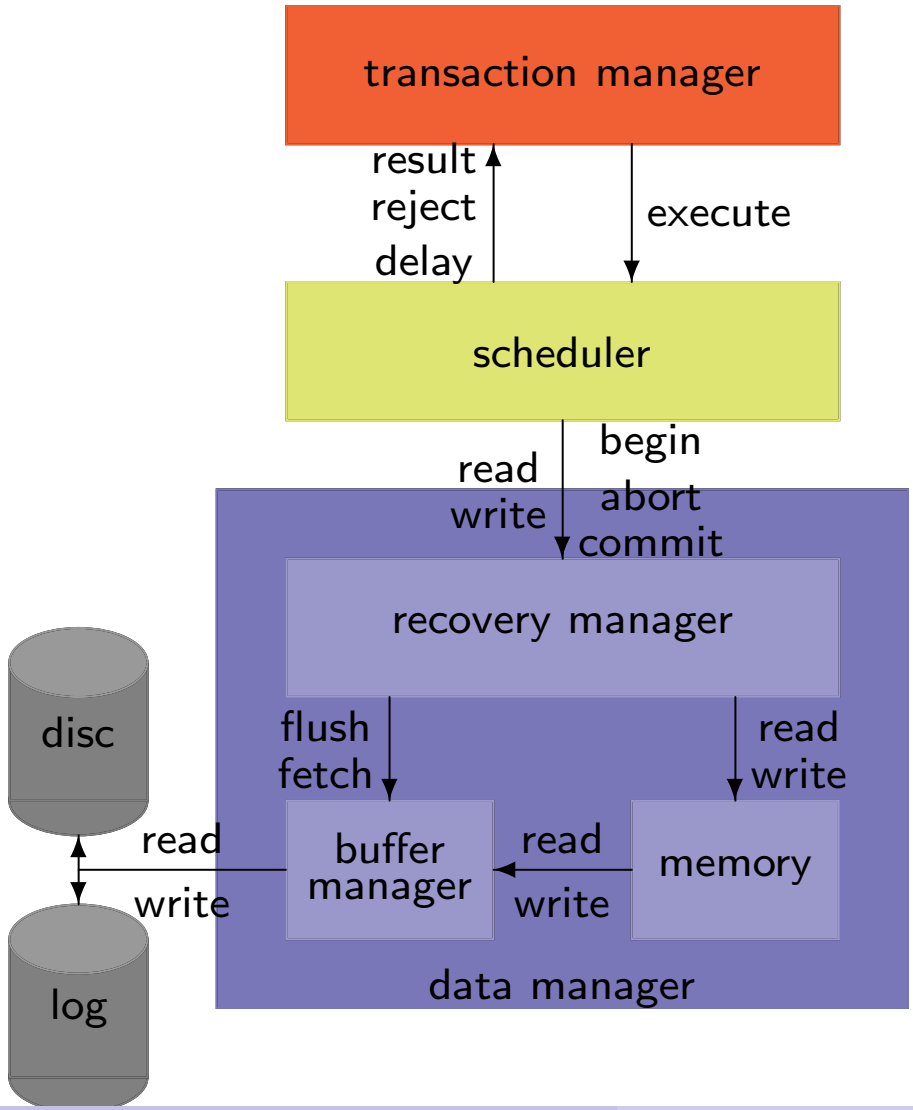
Distributed Databases



Need to distribute transactions of databases that might have

- Remote Tables
- Fragmentation
- Replication
- Migration

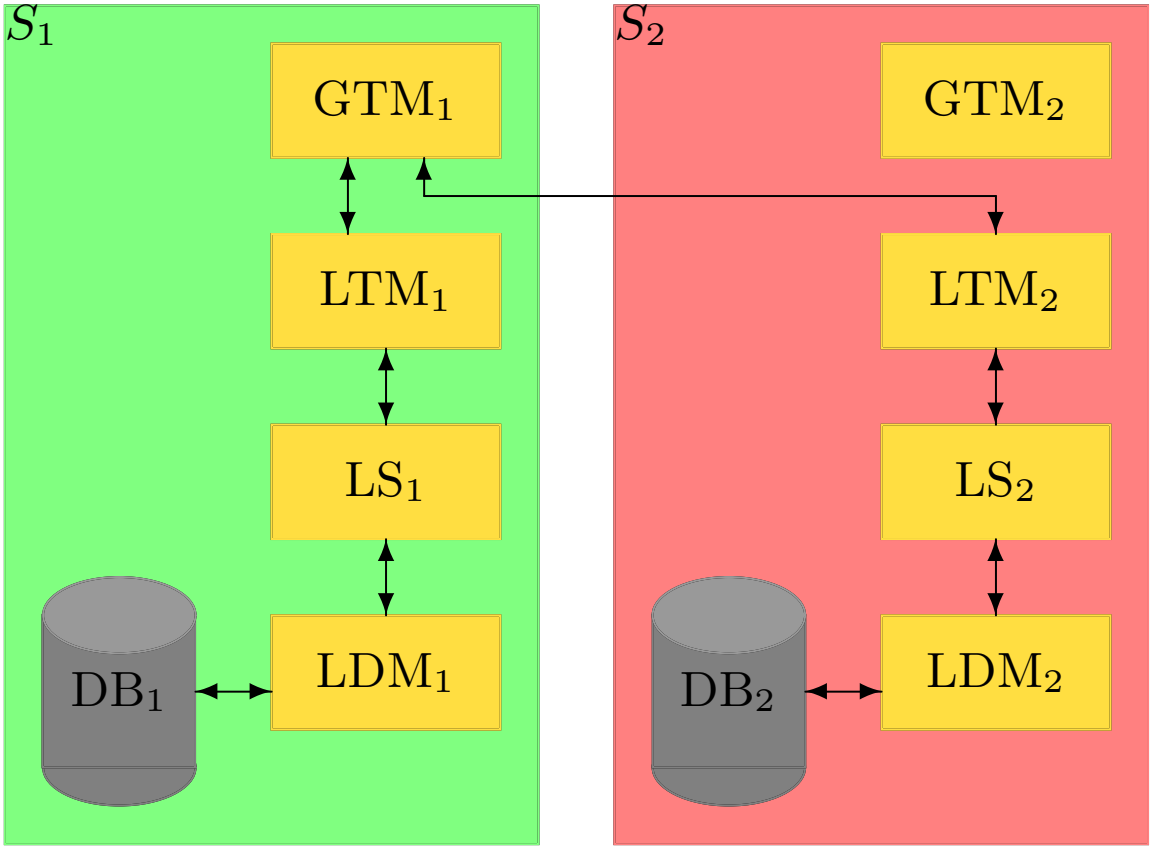
Single-Site Query Processing



Three Main Components of a DBMS

- 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

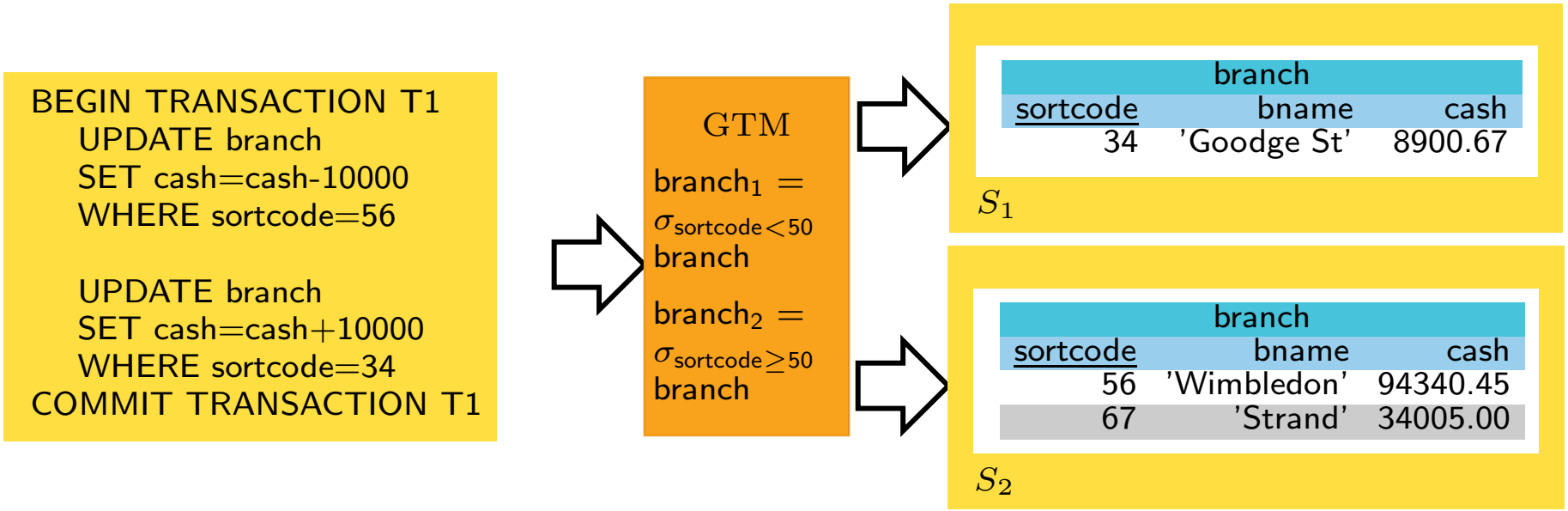
Distributed Query Processing



DDB Architecture

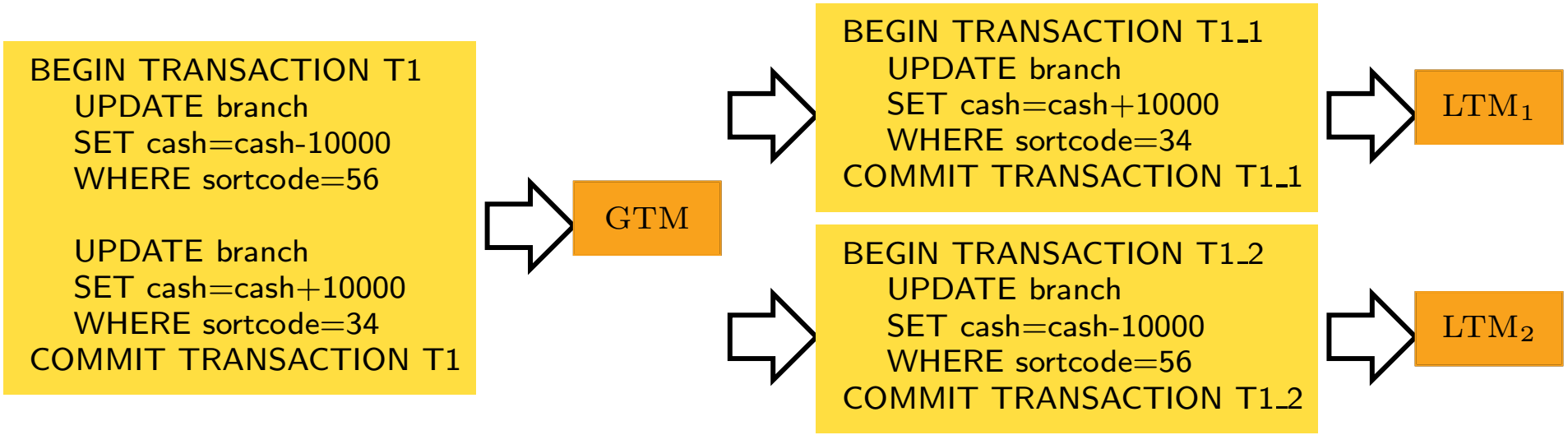
- Each site runs **local transaction manager** (**LTM**), scheduler, and data manager
- One or more sites run a **global transaction manager** (**GTM**)
- Slow part of distributed query processing is network communication

Distribution of TP: GTM Plans Execution

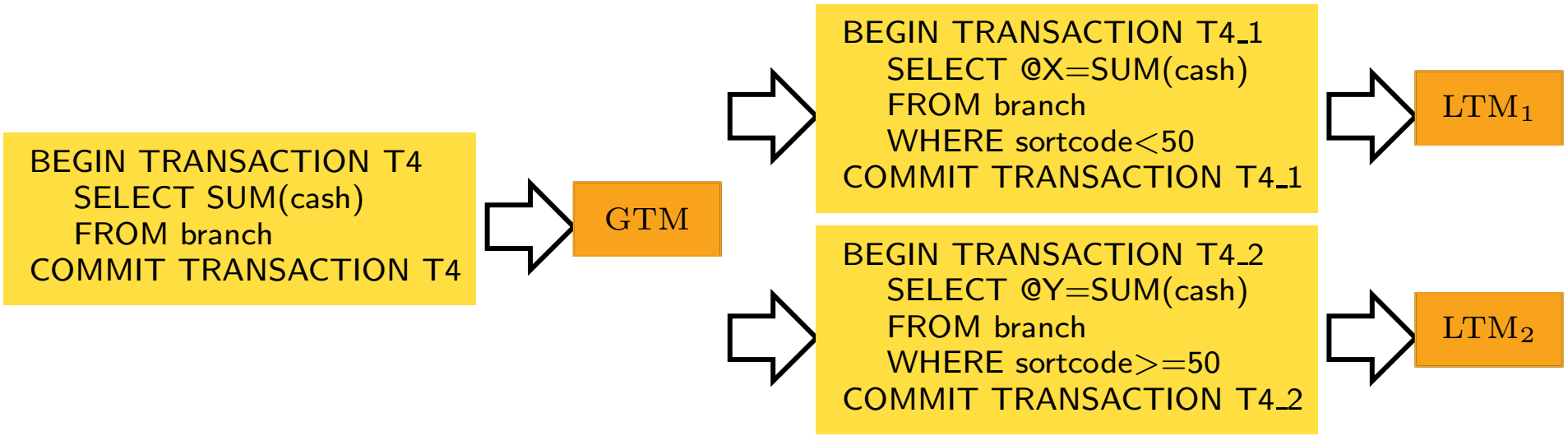


- GTM must transform transaction into sub transactions for each site
  - remote tables/fragmentation → breakup transaction into fragments
  - replication → read only transactions goto one site, read/write transactions to all sites
  - migration → change query plans with migration

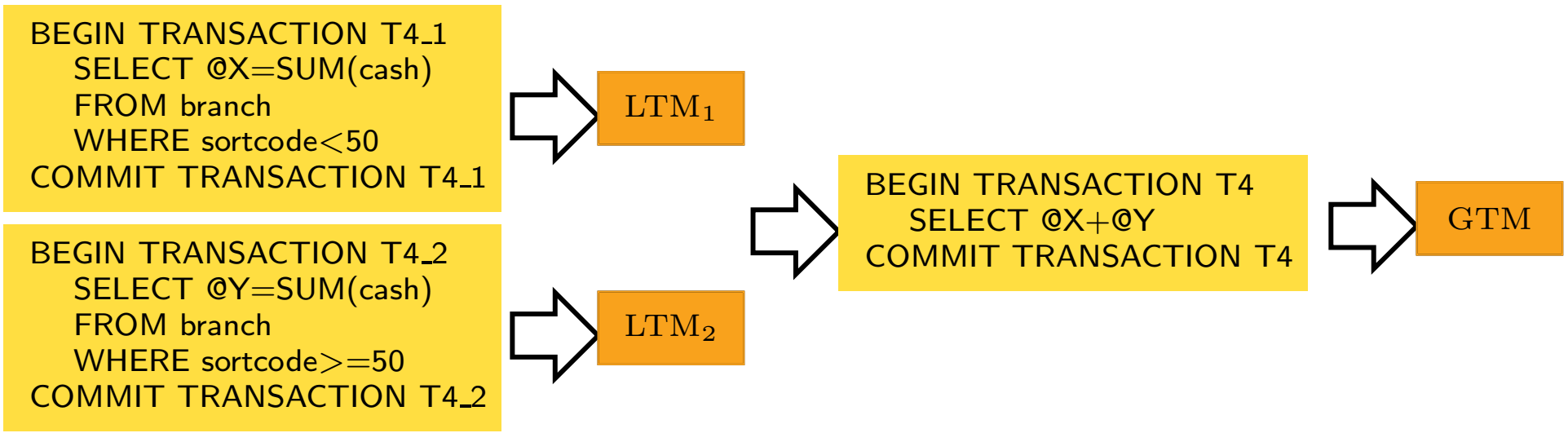
Distribution of TP: Example 1



Distribution of TP: Example 2

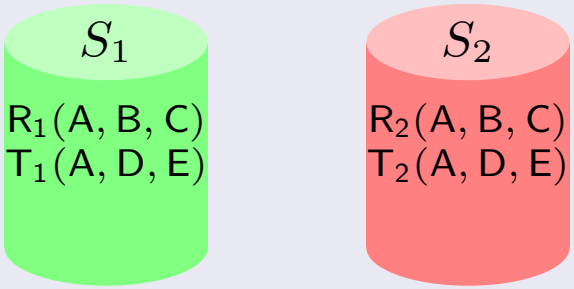


Distribution of TP: Example 2



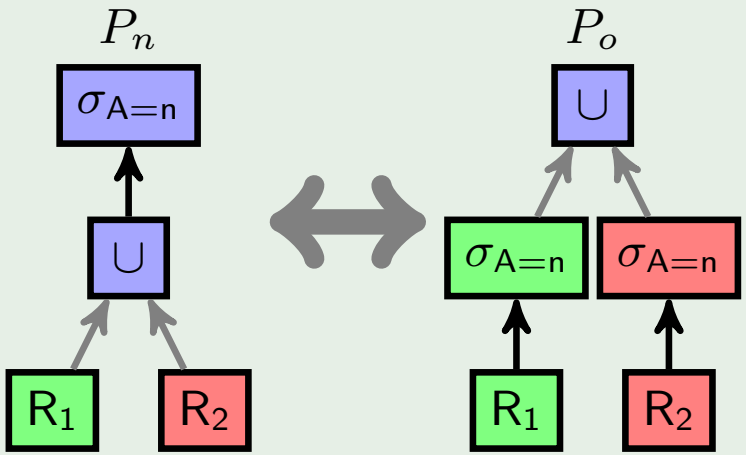
RA Equivalences: Selects over Horizontal Fragmentation (1)

Query Pattern



S_3: $\sigma_{A=n}\,R$

Query Plan



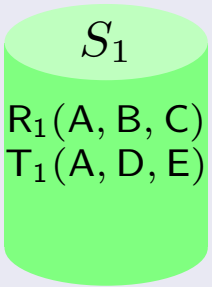
Query Cost

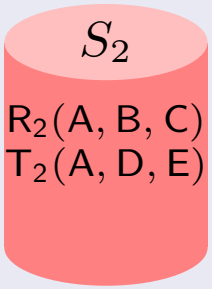
$NDB(P_n) = |R|.RowSize(R)$

$NDB(P_o) = |\sigma_{A=n}\,R|.RowSize(R)$

RA Equivalences: Selects over Horizontal Fragmentation (2)

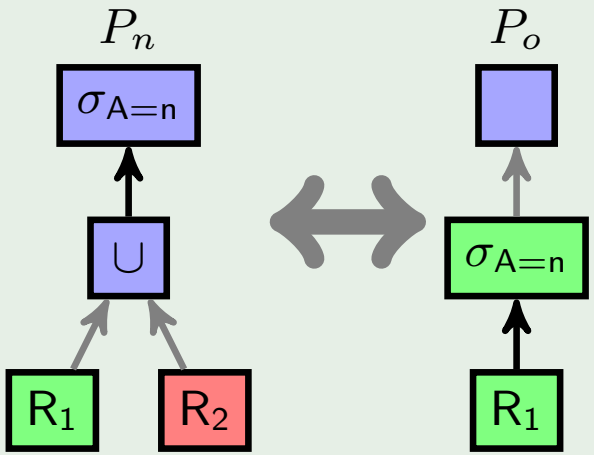
Query Pattern





S_3: $\sigma_{A=n}\, R$
$n \in K_1$
$R_1 = \sigma_{A \in K_1}\, R$
$R_2 = \sigma_{A \in K_2}\, R$

Query Plan



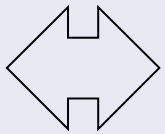
Query Cost

$\text{NDB}(P_n) = |R|.\text{RowSize}(R)$
$\text{NDB}(P_o) = |\sigma_{A=n}\, R|.\text{RowSize}(R)$

SQL DML: An Implementation of the RA

SQL SELECT statements: Rough Equivalence to RA

SELECT $A_1, ..., A_n$
FROM $R_1, ..., R_m$
WHERE P_1
AND ...
AND P_k



$$\pi_{A_1,...,A_n} \sigma_{P_1 \wedge ... \wedge P_k} R_1 \times ... \times R_m$$

SQL SELECT implements RA π, σ and $\times$

$\pi_{\text{bname,no}}\, \sigma_{\text{branch.sortcode=account.sortcode} \wedge \text{account.type='current'}}$ (branch $\times$ account)

```
SELECT  branch.bname,
        account.no
FROM    account, branch
WHERE   account.sortcode=branch.sortcode
AND     account.type='current'
```

Horizontal Fragmentation In SQL

Distributing a query over Horizontal Fragmentation

- Selection of horizontal fragment decided 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$_1$ = $\sigma_{\text{type}=\text{`}current\text{'}}$account
- account$_2$ = $\sigma_{\text{type}\neq\text{`}current\text{'}}$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 in SQL

Distributing a query over Vertical Fragmentation

- Selection of 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

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

Worksheet: SQL Queries over Fragmented Data

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:

–

Worksheet: Distribution of Transactions (2)

LTM3:

```
BEGIN TRANSACTION TB_3
    SELECT no
    INTO   ltm3
    FROM   account
    WHERE  cname='McBrien, P.'
COMMIT TRANSACTION TB_3
```

LTM4:

```
BEGIN TRANSACTION TB_4
    SELECT no
    INTO   ltm4
    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
    INTO   ltm2
    FROM   account
    WHERE  type='deposit'
COMMIT TRANSACTION TC_2
```

LTM4

```
BEGIN TRANSACTION TC_4
    SELECT no,cname
    INTO   ltm4
    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
    INTO   ltm1
    FROM   account
COMMIT TRANSACTION TD_1
```

LTM2

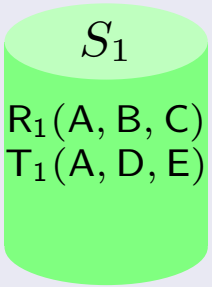
```
BEGIN TRANSACTION TD_2
    SELECT COALESCE(SUM(rate),0.0) AS total_rate,
           COUNT(rate) AS no_rate
    INTO   ltm2
    FROM   account
COMMIT TRANSACTION TD_2
```

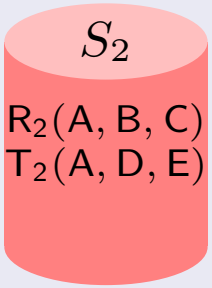
GTM:

```
SELECT (ltm1.total_rate+ltm2.total_rate)/
           (ltm1.no_rate+ltm2.no_rate)
FROM   ltm1, ltm2
```

RA Equivalences: Joins over Derived Horizontal Fragmentation

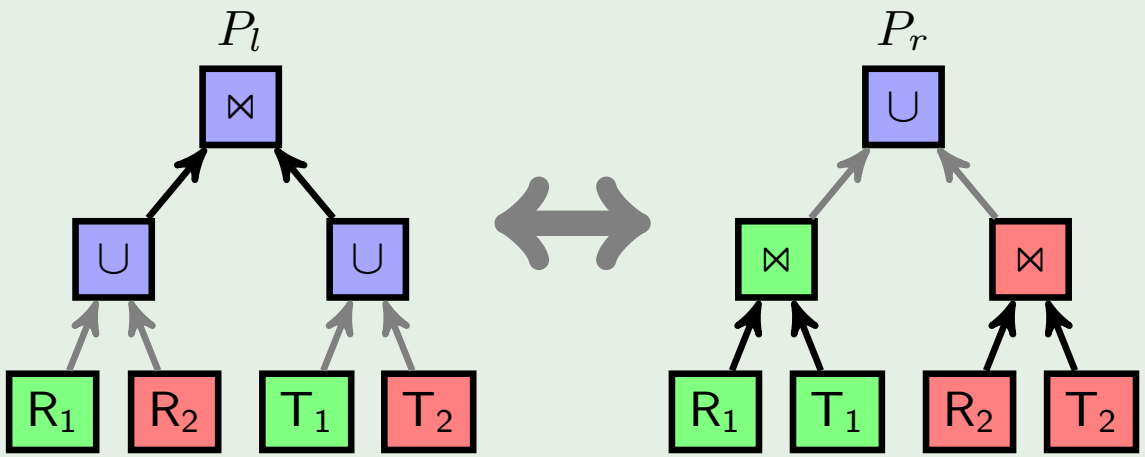
Query Pattern





S_3: $R \ltimes T$
$T_n = T \ltimes R_n$

Query Plan



Query Cost

$NDB(P_l) =$
$|R|.RowSize(R) + |T|.RowSize(T)$
$NDB(P_r) =$
$|R \ltimes T|.RowSize(R \ltimes T)$

Executing Distributed Joins over Remote Tables

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

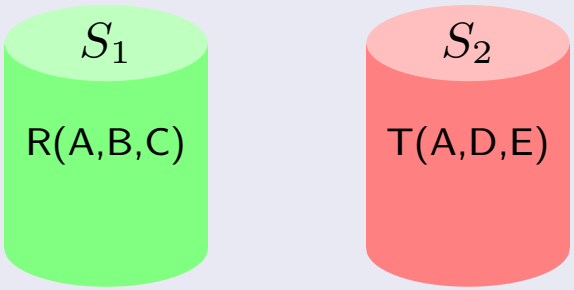
mid	no	amount	tdate
1000	100	2300.00	5/1/1999
1001	101	4000.00	5/1/1999
1002	100	-223.45	8/1/1999
1004	107	-100.00	11/1/1999
1005	103	145.50	12/1/1999
1006	100	10.23	15/1/1999
1007	107	345.56	15/1/1999
1008	101	1230.00	15/1/1999
1009	119	5600.00	18/1/1999

Suppose we want to execute

$$S_1 : \text{account} \bowtie \text{movement}$$

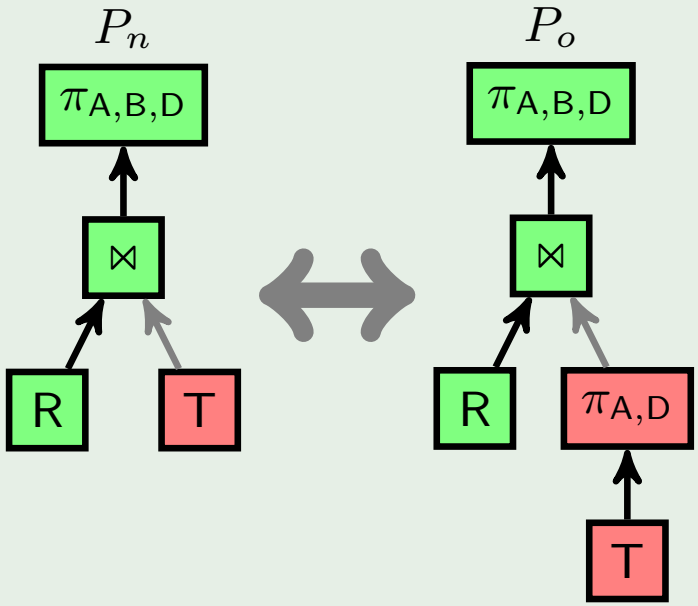
RA Equivalences: Push project inside join

Query Pattern



S_1: $\pi_{A,B,D}(R \bowtie T)$

Query Plan



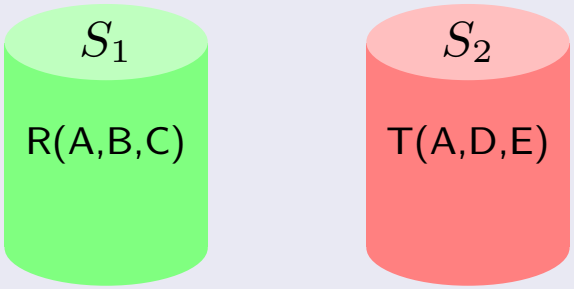
Query Cost

$\text{NDB}(P_n) = |T|.\text{RowSize}(T)$

$\text{NDB}(P_o) = |\pi_{A,D}\ T|.\text{RowSize}(\pi_{A,D}\ T)$

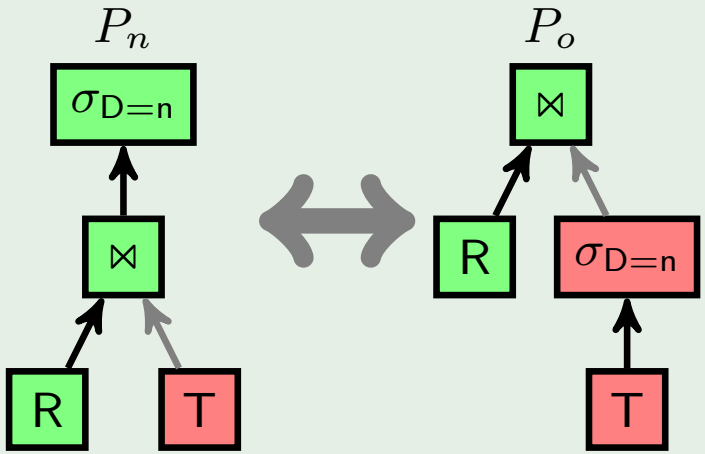
RA Equivalences: Push select inside join

Query Pattern



S_1: $\sigma_{\mathsf{D=n}}(\mathsf{R} \bowtie \mathsf{T})$

Query Plan



Query Cost

$\mathsf{NDB}(P_n) = |\mathsf{T}|.\mathsf{RowSize}(\mathsf{T})$
$\mathsf{NDB}(P_o) = |\sigma_{\mathsf{D=n}}\ \mathsf{T}|.\mathsf{RowSize}(\mathsf{T})$

Quiz 1: Calculating the Network Data Bytes Transferred

S_1

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

S_2

movement

mid	no	amount	tdate
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

S_3

```
SELECT   mid,
         no
FROM     movement
WHERE    tdate='15/1/1999'
```

If each column occupies 4 bytes, what is the number of **network data bytes (NDB)** required to execute the query at S_3?

A	B	C	D
144	72	24	8

Data Metadata

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)

Metadata

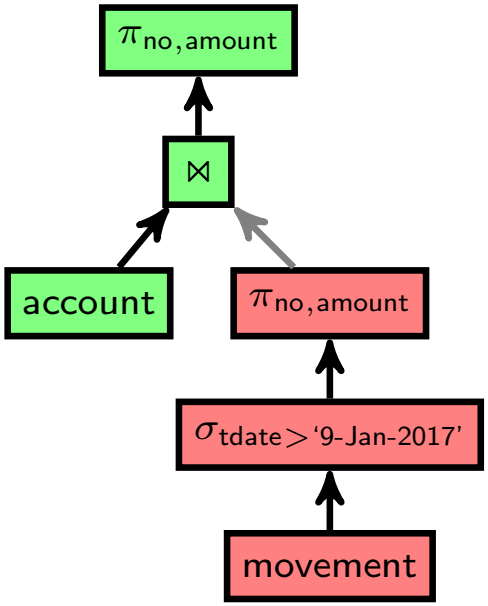
column	
name	size
account	6
amount	4
bname	32
branch	3

column	
name	size
cname	32
mid	4
movement	9
no	4

column	
name	size
rate	4
sortcode	1
tdate	4
type	8

$|\text{account}| = 6$
$|\text{branch}| = 3$
$|\text{movement}| = 9$

Quiz 2: Number of Network Data Bytes Required for a Query



Metadata

column	
name	size
account	6
amount	4
bname	32
branch	3

column	
name	size
cname	32
mid	4
movement	9
no	4

column	
name	size
rate	4
sortcode	1
tdate	4
type	8

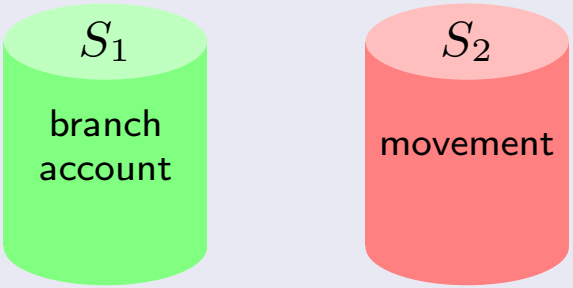
$|account| = 6$
$|branch| = 3$
$|movement| = 9$

How many NDB are required for the above distributed query plan, if the selectivity of the query $\sigma_{tdate>'9\text{-}Jan\text{-}2017'}$ is $\frac{2}{3}$?

A	B	C	D
144	48	16	8

Example Comparison (1)

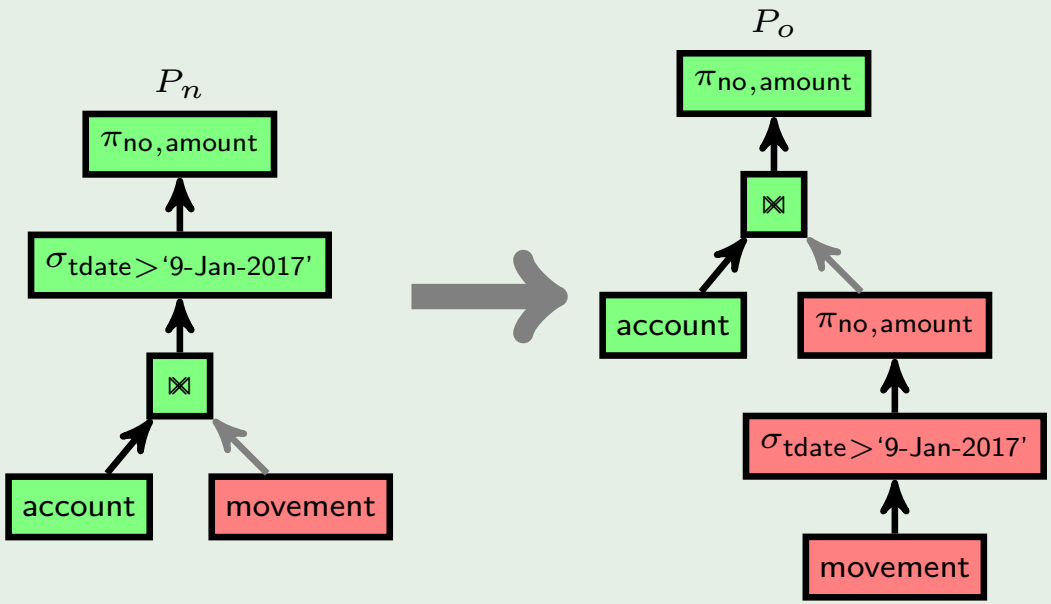
Query Pattern



S_1:

```
SELECT  account.no,
        movement.amount
FROM    account
        JOIN movement
        USING (no)
WHERE   tdate>'9-Jan-2017'
```

Query Plan



Example Comparison (1): Distributed Query Plans

Direct Execution of Query

$$\begin{aligned} S_2 \, to \, S_1 &= \mathsf{RowSize}(\mathsf{movement}) \times |\mathsf{movement}| \\ &= (4+4+4+4) \times 9 \\ &= 144 \end{aligned}$$

Project and Selects Pushed inside Join

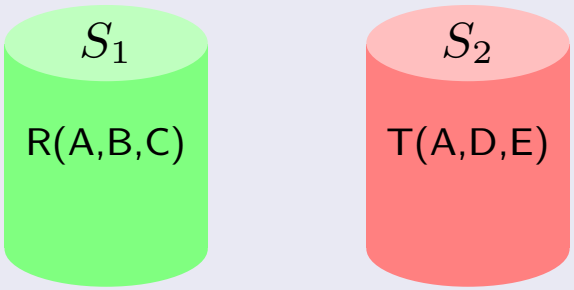
$$\begin{aligned} S_2 \, to \, S_1 &= \mathsf{RowSize}(\pi_{\mathsf{no,amount}} \, \mathsf{movement}) \times |\pi_{\mathsf{no,amount}} \, \sigma_{\mathsf{tdate} > \text{'9-Jan-2017'}} \, \mathsf{movement}| \\ &= (4+4) \times 6 \\ &= 48 \end{aligned}$$

Optimisation Rule

Pushing selects and projections inside a join will always make the query more efficient (unless the selections or projections where 'trivial', in which case it will make no difference).

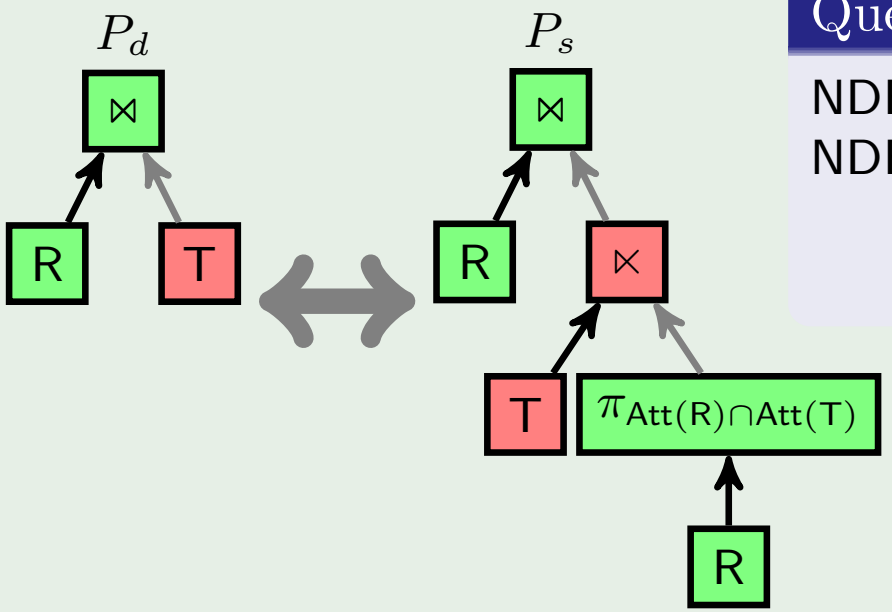
RA Equivalences: Direct and Semi Joins

Query Pattern



S_1: R ⋈ T

Query Plan



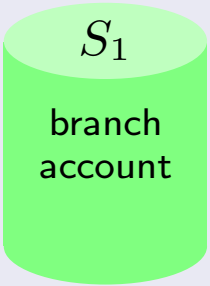
Query Cost

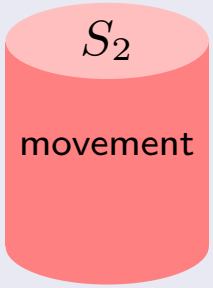
$$NDB(P_d) = |T|.RowSize(T)$$

$$NDB(P_s) = |\pi_{Att(R)\cap Att(T)} R|.RowSize(\pi_{Att(R)\cap Att(T)} R) + |T \ltimes R|.RowSize(T)$$

Example Comparison (2)

Query Pattern

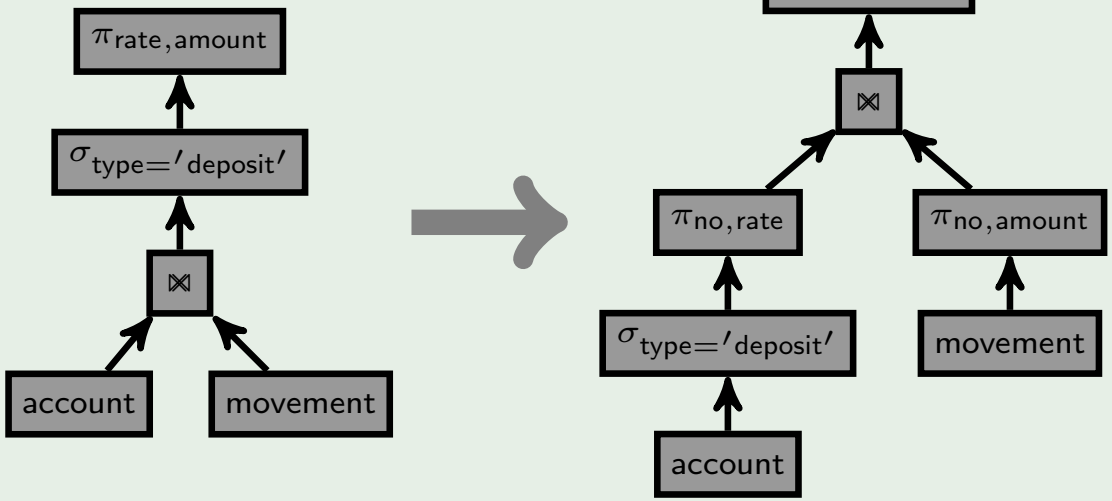




S_1:

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

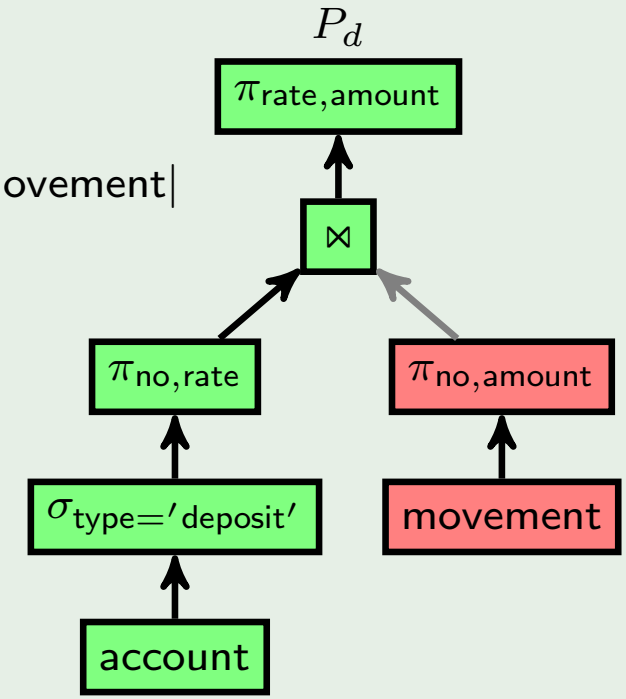
Query Plan



Example Comparison (2): Distributed Query Plans

Direct Join

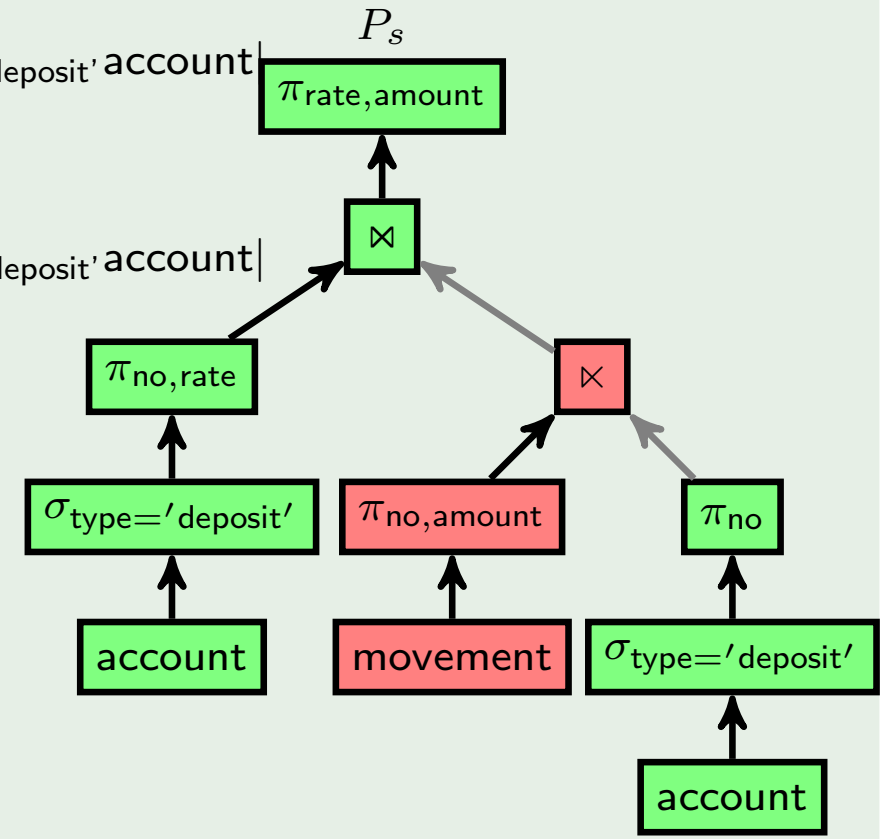
$$
\begin{aligned}
S_2 \; to \; S_1 &= \mathsf{RowSize}(\pi_{\mathsf{no,amount}}\,\mathsf{movement}) \times |\pi_{\mathsf{no,amount}}\,\mathsf{movement}| \\
&= (4+4) \times 9 \\
&= 72
\end{aligned}
$$



Example Comparison (2): Distributed Query Plans

Semi-Join

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



Quiz 3: General Efficiency of Direct Join compared to Semi-Join

Comparing the Direct Join plan P_d with the Semi-Join plan P_s, for any query Q is it the case that

A

Always

$\mathsf{NDB}(P_d) \leq \mathsf{NDB}(P_s)$

B

Always

$\mathsf{NDB}(P_d) \geq \mathsf{NDB}(P_s)$

C

Sometimes

$\mathsf{NDB}(P_d) \leq \mathsf{NDB}(P_s)$

Sometimes

$\mathsf{NDB}(P_d) \geq \mathsf{NDB}(P_s)$

Worksheet: Joins over Remote Tables

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

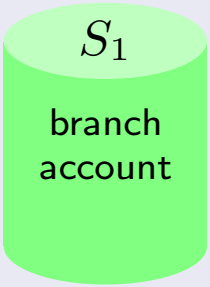
S_2

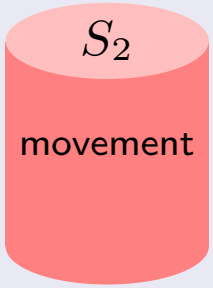
column	bytes
telephone	5
name	6
charge	4
rate	1

$$\text{business}(\text{telephone}) \overset{\text{fk}}{\Rightarrow} \text{directory}(\text{telephone})$$

Example Comparison (3)

Query Pattern

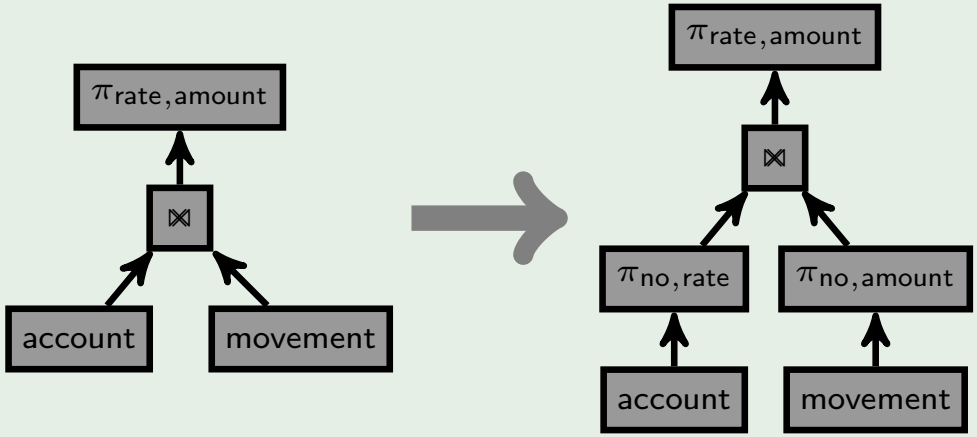




S_1:

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

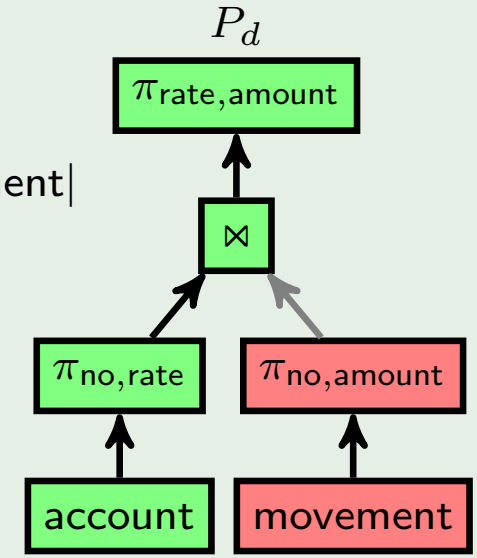
Query Plan



Example Comparison (3): Distributed Query Plans

Direct Join

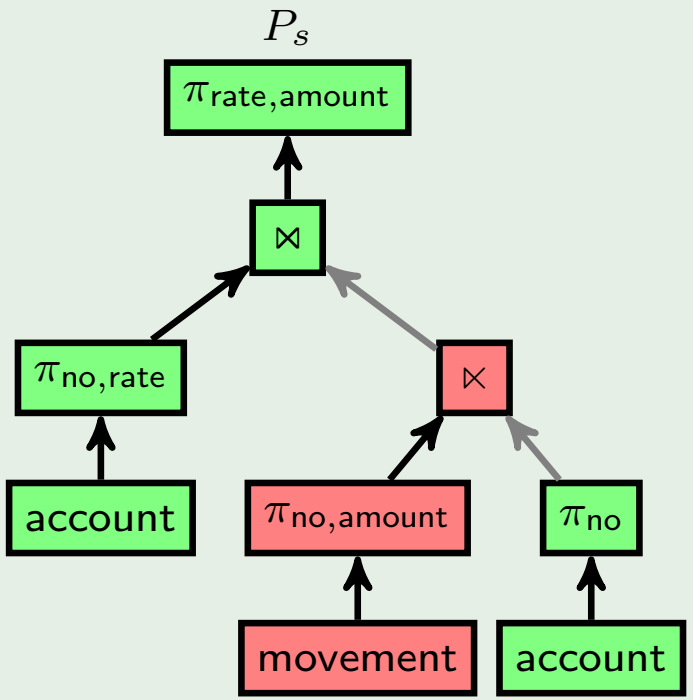
$$
\begin{aligned}
S_2 \; to \; S_1 &= \mathsf{RowSize}(\pi_{\mathsf{no,amount}}\,\mathsf{movement}) \times |\pi_{\mathsf{no,amount}}\,\mathsf{movement}| \\
&= (4+4) \times 9 \\
&= 72
\end{aligned}
$$



Example Comparison (3): Distributed Query Plans

Semi-Join

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



Quiz 4: Selectivity of Queries

customer			
cname	current	deposit	sortcode
McBrien, P.	100	101	67
Poulovassilis, A.	107	119	56
Boyd, M.	103		34
Bailey, J.	125		56

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

```
SELECT   cname,
         sortcode,
         mid
FROM     movement
         JOIN customer
         ON   no IN (deposit,current)
WHERE    tdate='15/1/1999'
```

What is the selectivity over customer of the above SQL query?

A	B	C	D
0.25	0.5	0.75	1.0

Worksheet: PSJ over Remote Tables

S_1

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_2

business	
telephone	rate
1002	A
1004	B
1006	B

column	bytes
telephone	5
name	6
charge	4
rate	1

$$\text{business}(\text{telephone}) \overset{\text{fk}}{\Rightarrow} \text{directory}(\text{telephone})$$

Worksheet: $\pi_{\mathsf{name,rate}}\sigma_{\mathsf{charge}>300.00}\mathsf{business} \bowtie \mathsf{directory}$

(2) **Direct Join**

introduce $f_1 = \frac{|\,\sigma_{\mathsf{charge}>300.00}\,\mathsf{directory}|}{|\mathsf{directory}|} = \frac{1}{8}$

$$\begin{aligned} S_1 \rightarrow S_2 &= \mathsf{RowSize}(\pi_{\mathsf{telephone,name}}\,\mathsf{directory}) \times |\,\sigma_{\mathsf{charge}>300.00}\,\mathsf{directory}| \\ &= (5+6) \times f_1 d \\ &= 11 f_1 d \text{ Bytes} \end{aligned}$$

(4) **Semi Join**

introduce $f_2 = \frac{|\,\sigma_{\mathsf{charge}>300.00}\,\mathsf{directory} \ltimes \mathsf{business}|}{|\mathsf{business}|} = \frac{1}{3}$

$$\begin{aligned} S_2 \rightarrow S_1 &= \mathsf{RowSize}(\pi_{\mathsf{telephone}}\,\mathsf{directory}) \times |\mathsf{business}| = 5b \\ S_1 \rightarrow S_2 &= \mathsf{RowSize}(\pi_{\mathsf{telephone,name}}\,\mathsf{directory}) \times |\,\sigma_{\mathsf{charge}>300.00}\,\mathsf{directory} \ltimes \mathsf{business}| \\ &= (5+6) \times f_2 b = 11 f_2 b \end{aligned}$$

Total data transfer$=5b + 11 f_2 b$ Bytes

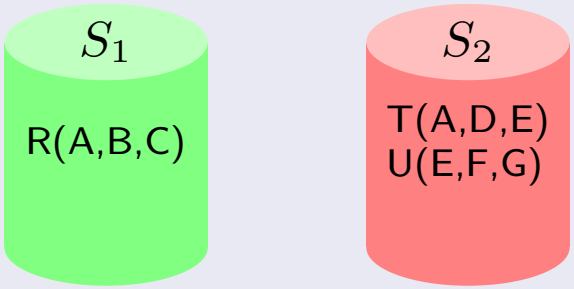
For current data, total data transfer $= 26$ Bytes

(5) $11 f_1 d = 5b + 11 f_2 b$

$\frac{b}{d} = \frac{11 f_1}{(5+11 f_2)} = 15.9\%$

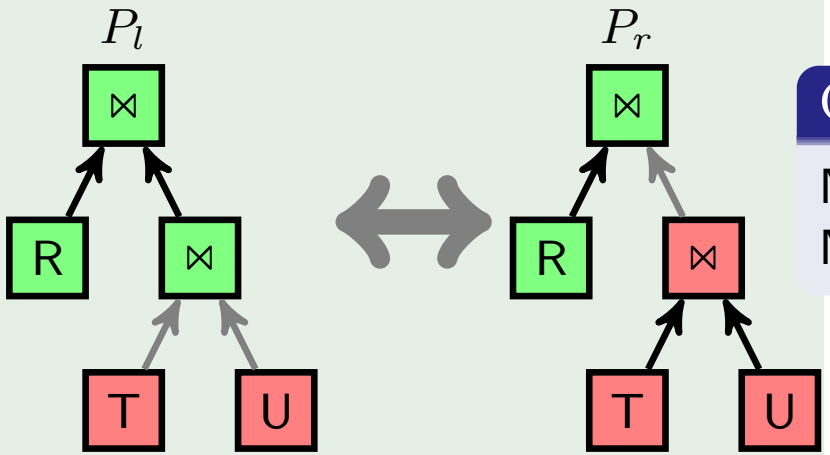
RA Equivalences: Local and Remote joins

Query Pattern



S_1: R ⋈ T ⋈ U

Query Plan



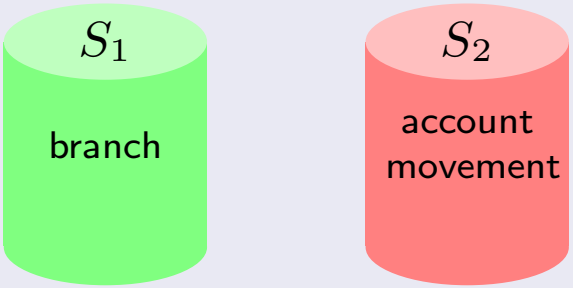
Query Cost

$\text{NDB}(P_l) = |\text{T}|.\text{RowSize}(\text{T}) + |\text{U}|.\text{RowSize}(\text{U})$

$\text{NDB}(P_r) = |\text{T} \bowtie \text{U}|.\text{RowSize}(\text{T} \bowtie \text{U})$

Example Comparison (4)

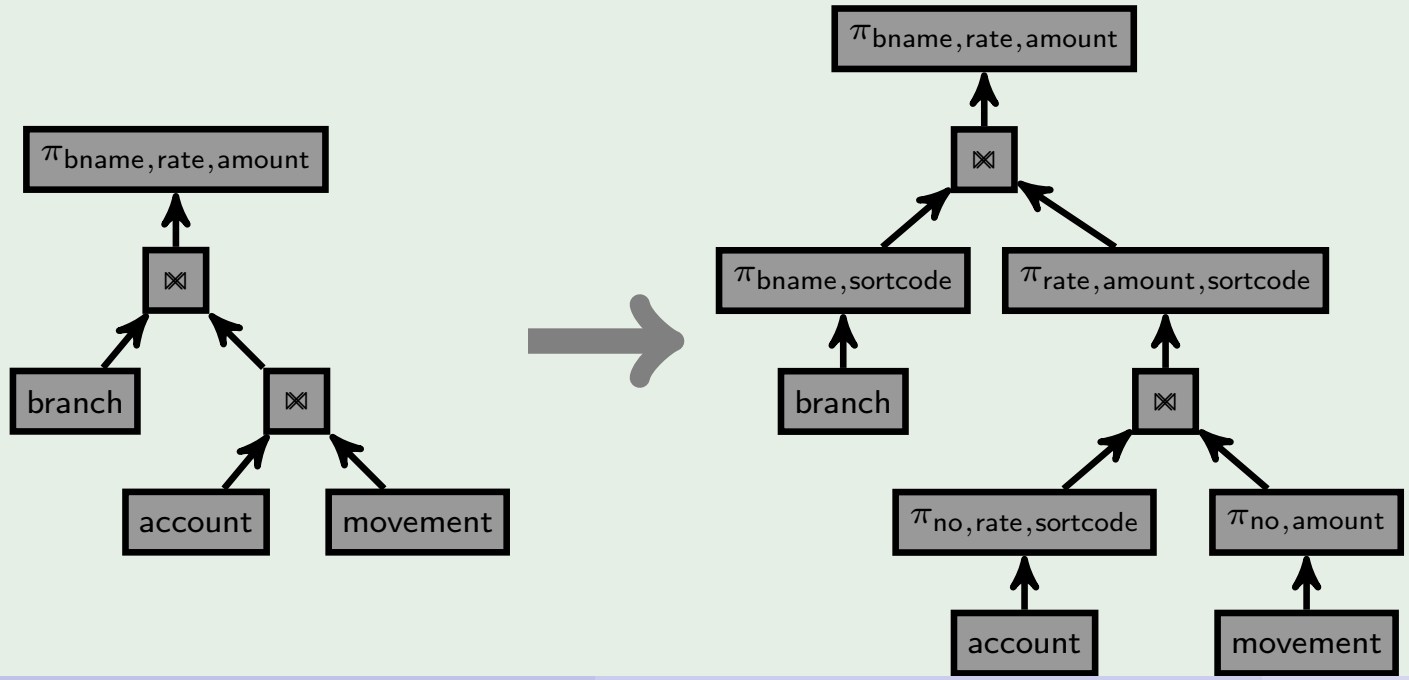
Query Pattern



S_1:

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

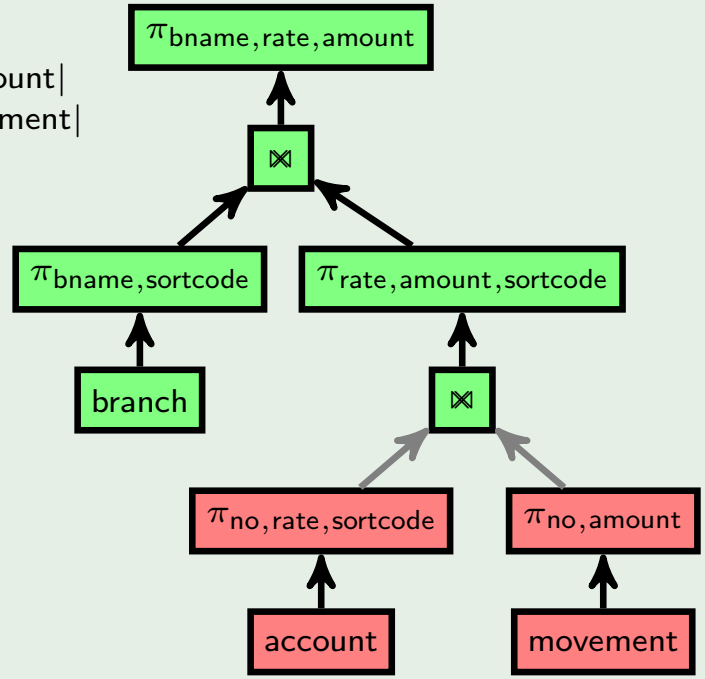
Query Plan



Example Comparison (4): Distributed Query Plans

Local Join

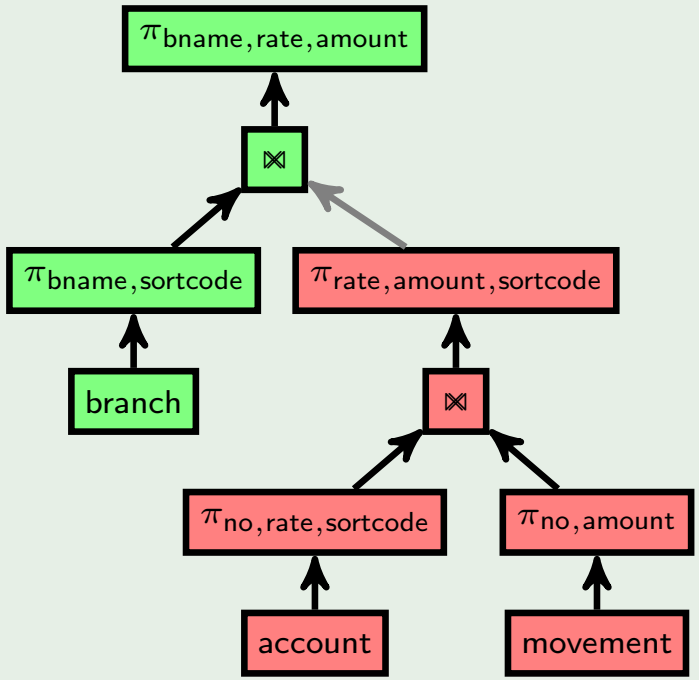
$$
\begin{array}{lcl}
S_2 \; to \; S_1 & = & \text{RowSize}(\pi_{\text{no,rate,sortcode}}\text{account}) \times |\pi_{\text{no,rate,sortcode}}\text{account}| \\
& = & \quad \text{RowSize}(\pi_{\text{no,amount}}\text{movement}) \times |\pi_{\text{no,amount}}\text{movement}| \\
& = & (4 + 4 + 1) \times 6 + (4 + 4) \times 9 \\
& = & 54 + 72 \\
& = & 126
\end{array}
$$



Example Comparison (4): Distributed Query Plans

Remote Join

$$
\begin{aligned}
S_2 \; to \; S_1 &= \text{RowSize}(\pi_{\text{rate,amount,sortcode}}\text{account} \bowtie \text{movement}) \times \\
&= \quad |\pi_{\text{rate,amount,sortcode}}\text{account} \bowtie \text{movement}| \\
&= (4 + 4 + 1) \times 9 \\
&= 81
\end{aligned}
$$



Quiz 5: General Efficiency of Local Join compared to Remote Join

Comparing the Local Join plan P_l with the Remote Join plan P_r, for any query Q is it the case that

A

Always

$\mathsf{NDB}(P_l) \leq \mathsf{NDB}(P_r)$

B

Always

$\mathsf{NDB}(P_l) \geq \mathsf{NDB}(P_r)$

C

Sometimes

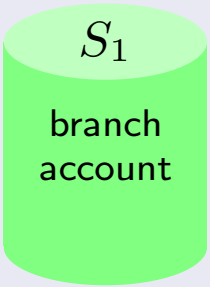
$\mathsf{NDB}(P_l) \leq \mathsf{NDB}(P_r)$

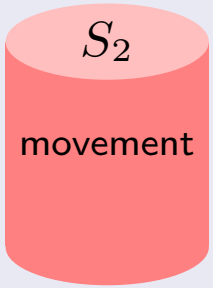
Sometimes

$\mathsf{NDB}(P_l) \geq \mathsf{NDB}(P_r)$

Example Comparison (5)

Query Pattern

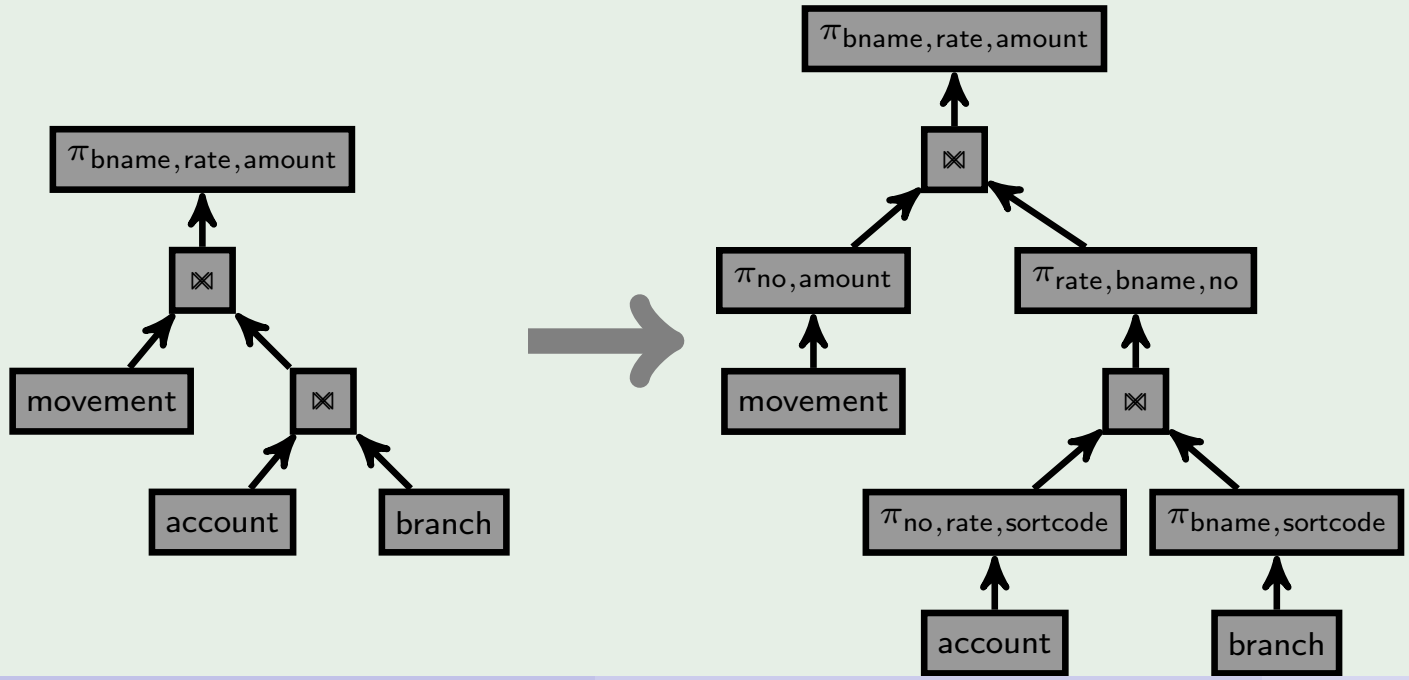




S_2:

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

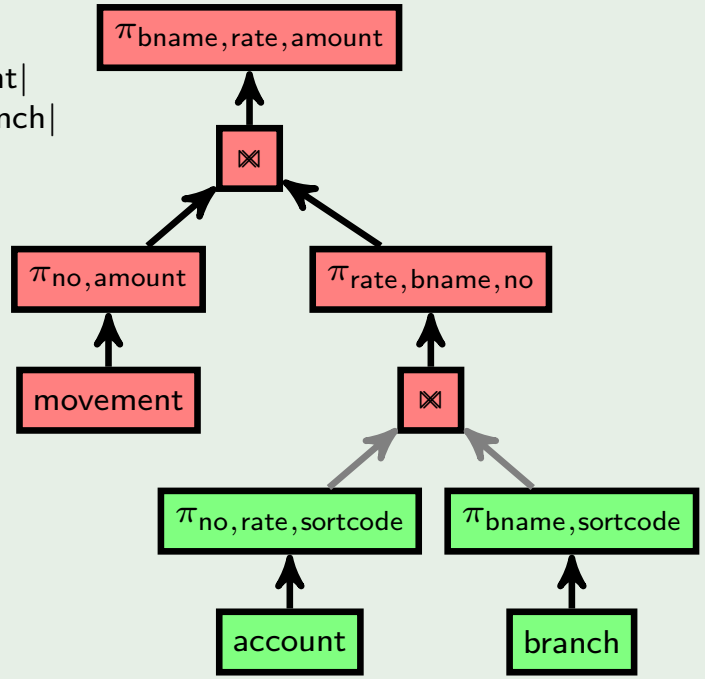
Query Plan



Example Comparison (5): Distributed Query Plans

Local Join

$$
\begin{array}{lcl}
S_1 \; to \; S_2 & = & \text{RowSize}(\pi_{\text{no,rate,sortcode}}\text{account}) \times |\pi_{\text{no,rate,sortcode}}\text{account}| \\
& = & \quad \text{RowSize}(\pi_{\text{bname,sortcode}}\text{branch}) \times |\pi_{\text{bname,sortcode}}\text{branch}| \\
& = & (4 + 4 + 1) \times 6 + (32 + 1) \times 3 \\
& = & 54 + 99 \\
& = & 153
\end{array}
$$



Example Comparison (5): Distributed Query Plans

Remote Join

$$
\begin{aligned}
S_1 \; to \; S_2 &= \text{RowSize}(\pi_{\text{rate,bname,no}}\text{account} \bowtie \text{branch}) \times \\
&= \quad |\pi_{\text{rate,bname,no}}\text{account} \bowtie \text{branch}| \\
&= (4 + 32 + 4) \times \times 6 \\
&= 240
\end{aligned}
$$