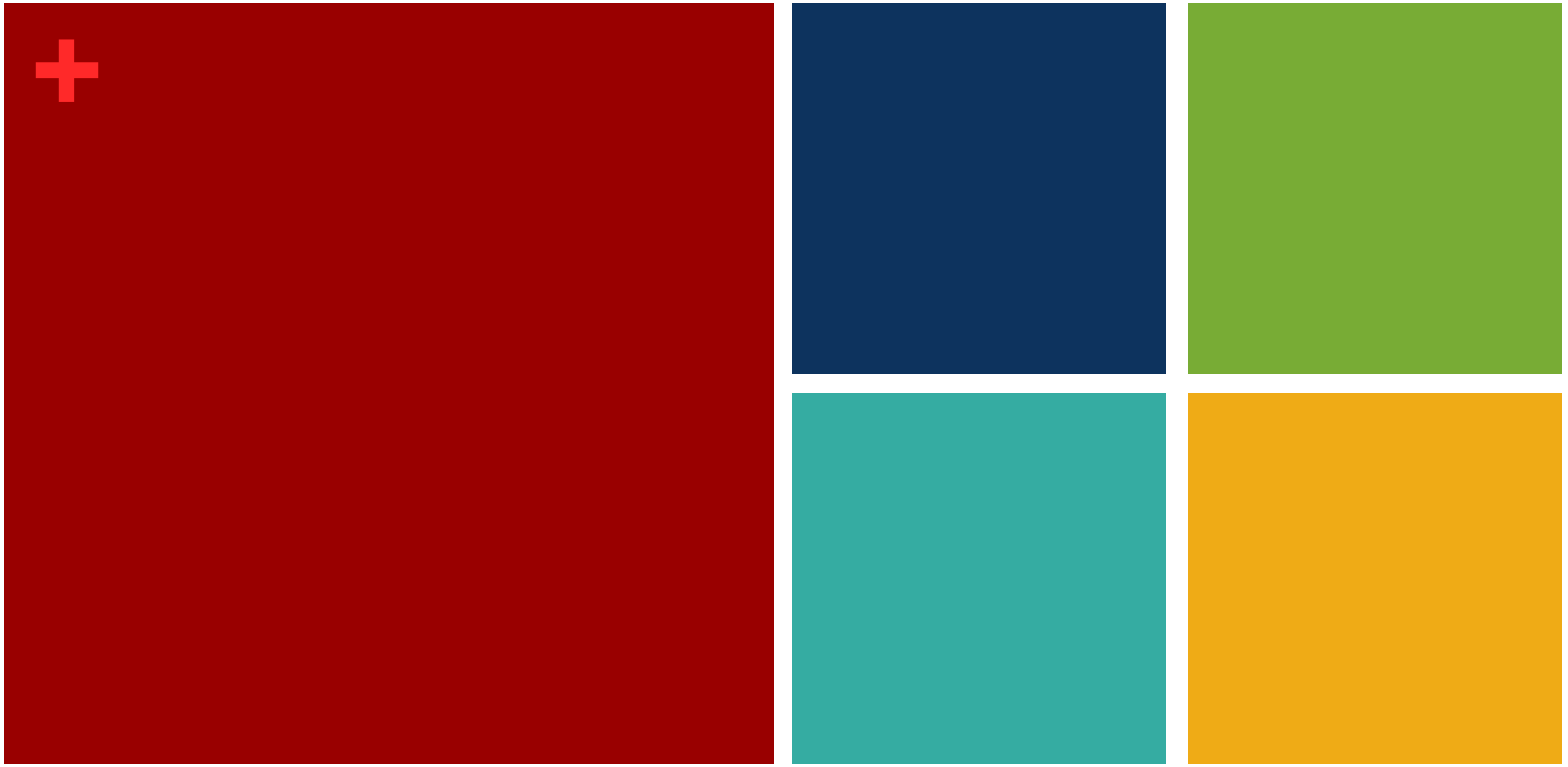
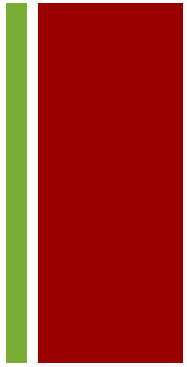


Tutorial 2:
Distributed Query Processing

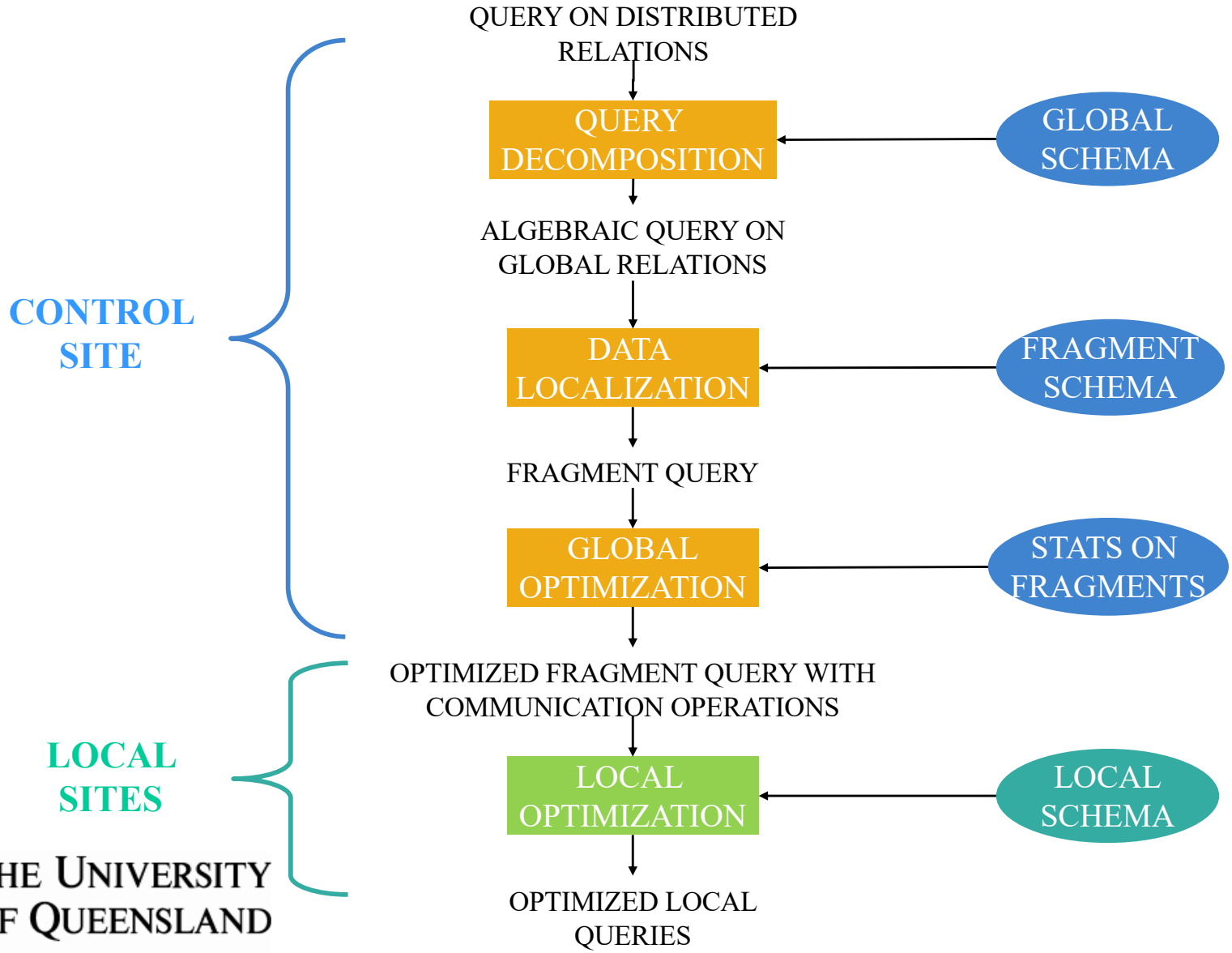
THE UNIVERSITY OF QUEENSLAND
AUSTRALIA

+ Q1 Distributed Query Processing



- At the global level, distributed query processing consists of three main steps:
  1. Query Decomposition
  2. Data Localization
  3. Global Optimization
- Discuss these three steps, focusing on the input, output, objectives for each step.

Q1 Layers of Query Processing



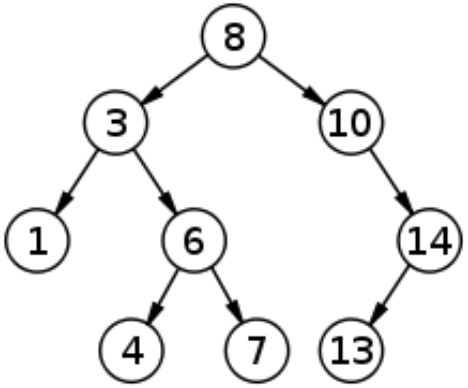
+ Distributed Query Processor

- From high level query
  - Relational Calculus:$\{t|F(t)\}$
    - $\{t|\ EMPLOYEE(t)\ \ and\ t.SALARY > 10000\}$
  - SQL Query:
    - SELECT … FROM … WHERE …
- To a sequence of database operators on fragments
  - Relational Algebra
    - Selection $\sigma_F(R)$ F is a formula: $SALARY > 10000$
    - Projection $\Pi_{A,B}(R)$ A, B are two attributes of R
    - Union $R \cup S$
    - Difference $R - S$
    - Cartesian Product $R \times S$

Relational Algebra Operation Complexity

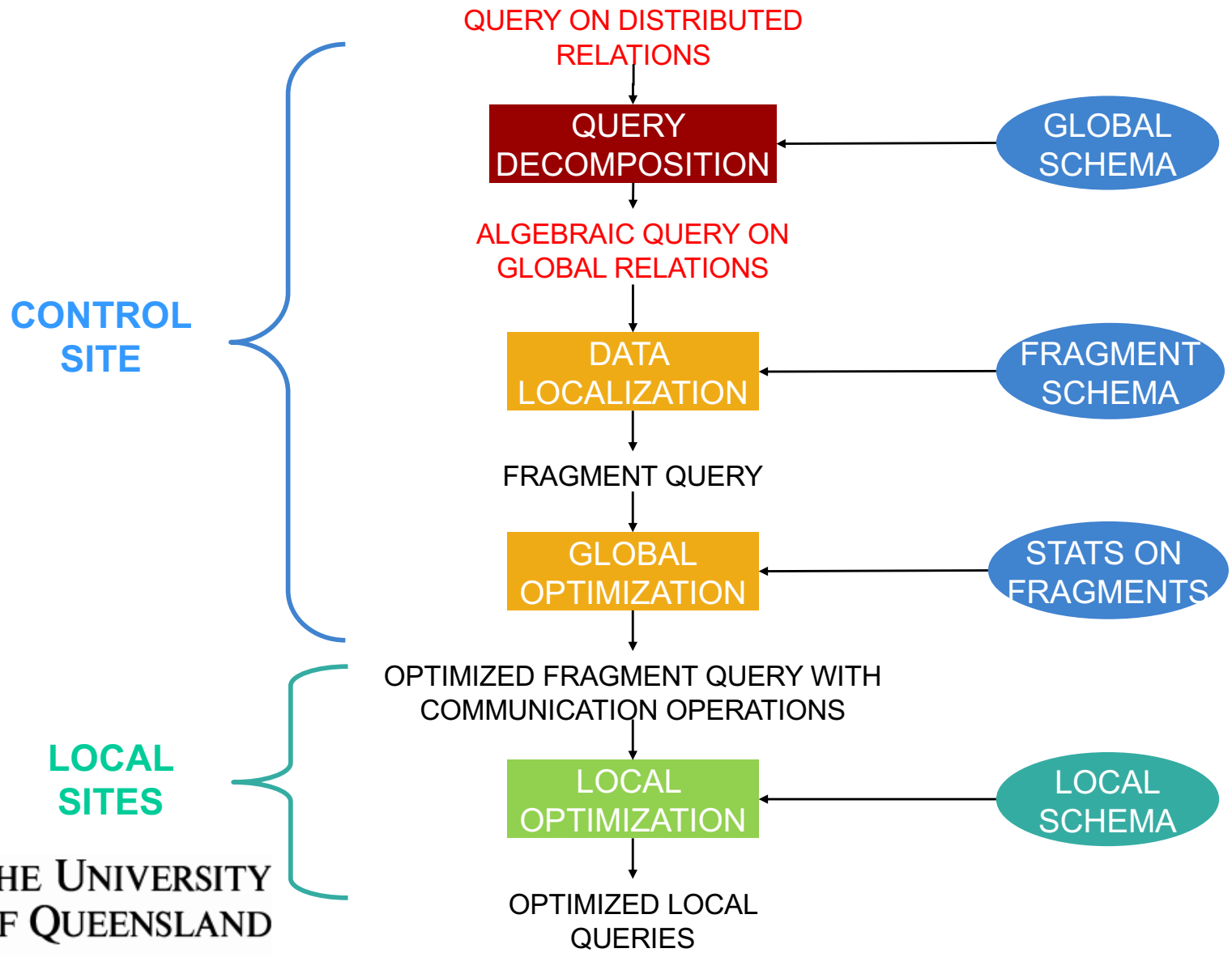
Operation	Complexity
Selection Project (without duplicate elimination)	$O(n)$
Project (with duplicate elimination) Group by	$O(n \log n)$
Join Semi-Join Division Set Operators	$O(n \log n)$
Cartesian Product	$O(n^2)$

Find a value from a tree takes $O(\log n)$ time



- Optimization Principle
  - Reduce number of operations
  - Reduce size n

Q1 Layers of Query Processing



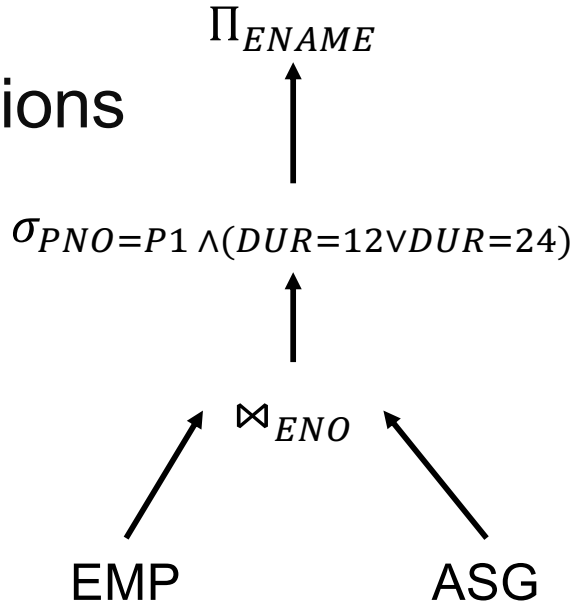
THE UNIVERSITY OF QUEENSLAND
AUSTRALIA

Q1-1 Query Decomposition

- Input: User query on global data expressed in relational calculus.

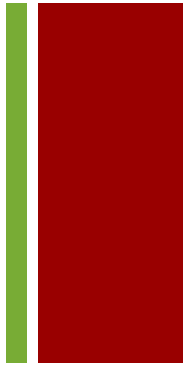
```
SELECT    ENAME
FROM      EMP, ASG
WHERE     EMP.ENO = ASG.ENO AND ASG.PNO= "P1"
AND （DUR=12 OR DUR=24）
```

- Output: An algebraic query on global relations



- Nothing to do with the distribution of data
  - Same for centralized and distributed system

Q1-1 Query Decomposition



- Objective:
  - Same as centralized DBMS
    - Semantically correct
    - Redundant work is avoided
    - Better query performance
- Steps
  1. Normalization
  2. Analysis
  3. Redundancy Removal
  4. Rewrite

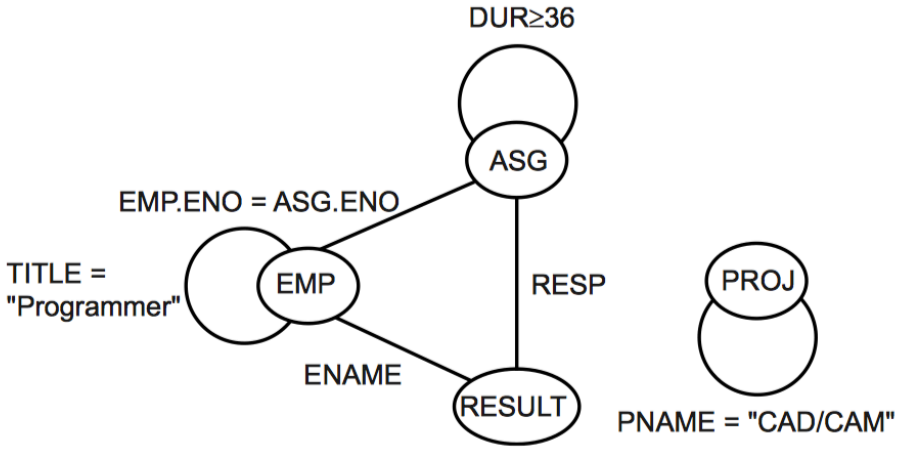
+ Q1-1 Query Decomposition Steps

1. Normalization
   - Transform the WHERE clause to normal form
   - Conjunctive normal form
     - $(p_{11} \vee p_{12} \vee \ldots \vee p_{1n}) \wedge \ldots \wedge (p_{11} \vee p_{12} \vee \ldots \vee p_{1n})$
       - EMP.ENO = ASG.ENO ∧ ASG.PNO = ”P1” ∧ (DUR=12 ∨ DUR=24)
   - Disjunctive normal form
     - $(p_{11} \wedge p_{12} \wedge \ldots \wedge p_{1n}) \vee \ldots \vee (p_{11} \wedge p_{12} \wedge \ldots \wedge p_{1n})$
       - (EMP.ENO = ASG.ENO ∧ ASG.PNO = ”P1” ∧ DUR=12) ∨
       - (EMP.ENO = ASG.ENO ∧ ASG.PNO = ”P1” ∧ DUR=24)
     - Replicated join and select
   - One form can be converted to another through transformation rules.

Q1-1 Query Decomposition Steps

2. Analysis
   - Remove impossible or unnecessary normalized queries
   1. Type Incorrect
      - Attribute or relation name not defined in global schema
      - Operation applied on wrong type
        - Birthdate > 20
   2. Semantically Incorrect

```
SELECT ENAME, RESP
FROM   EMP, ASG, PROJ
WHERE  EMP.ENO = ASG.ENO
AND    PNAME = "CAD/CAM"
AND    DUR ≥ 36
AND    TITLE = "Programmer"
```



Q1-1 Query Decomposition Steps

3. Elimination of Redundancy

```
SELECT TITLE
FROM   EMP
WHERE  (NOT (TITLE = "Programmer")
AND    (TITLE = "Programmer"
OR     TITLE = "Elect. Eng.")
AND    NOT (TITLE = "Elect. Eng."))
OR     ENAME = "J. Doe"
```

p_1 : TITLE = “Programmer”
p_2 : TITLE=“Elect.Eng”
p_3 : ENAME=“J. Doe”

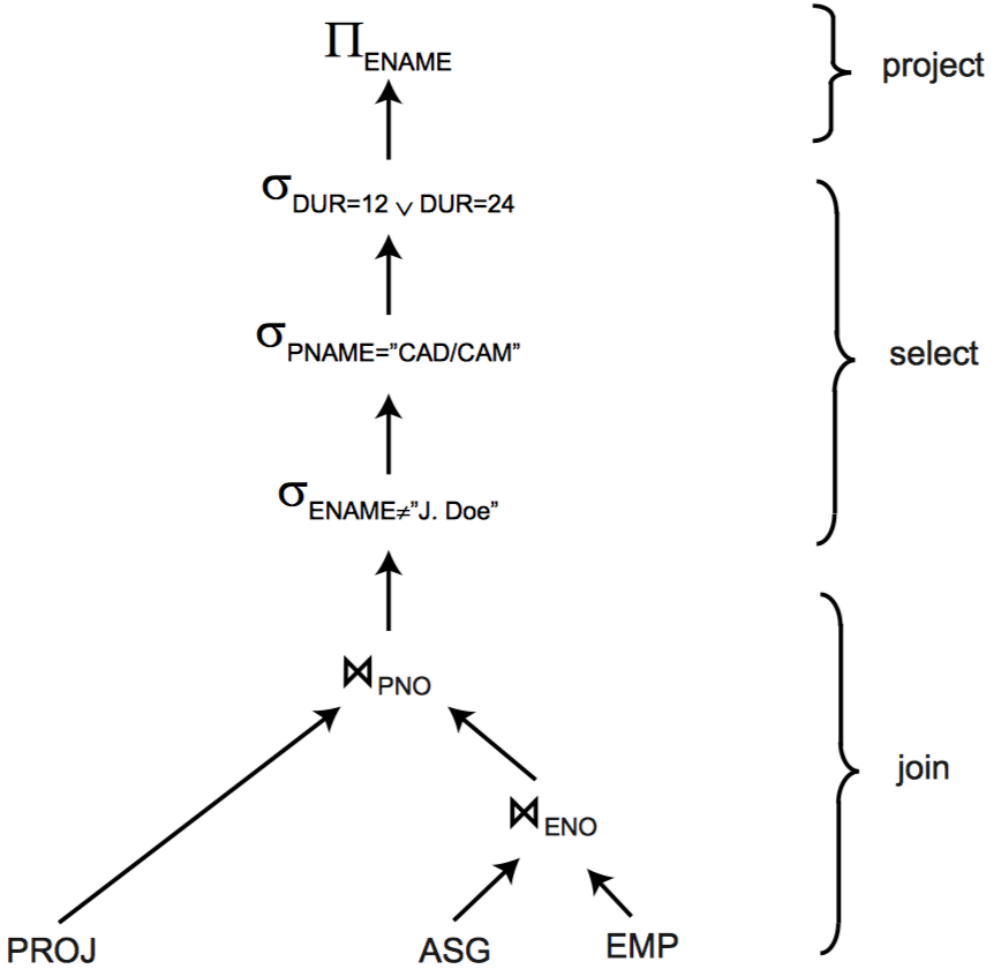
$$(\neg p_1 \wedge (p_1 \vee p_2) \wedge \neg p_2) \vee p_3$$
$$= (\neg p_1 \wedge ((p_1 \wedge \neg p_2) \vee (p_2 \wedge \neg p_2)) \vee p_3$$
$$= (\neg p_1 \wedge p_1 \wedge \neg p_2) \vee (\neg p_1 \wedge p_2 \wedge \neg p_2)) \vee p_3$$
$$= (false \wedge \neg p_2) \vee (\neg p_1 \wedge false)) \vee p_3$$
$$= false \vee false \vee p_3$$
$$= p_3$$

```
SELECT TITLE
FROM   EMP
WHERE  ENAME = "J. Doe"
```

+ Q1-1 Query Decomposition Steps

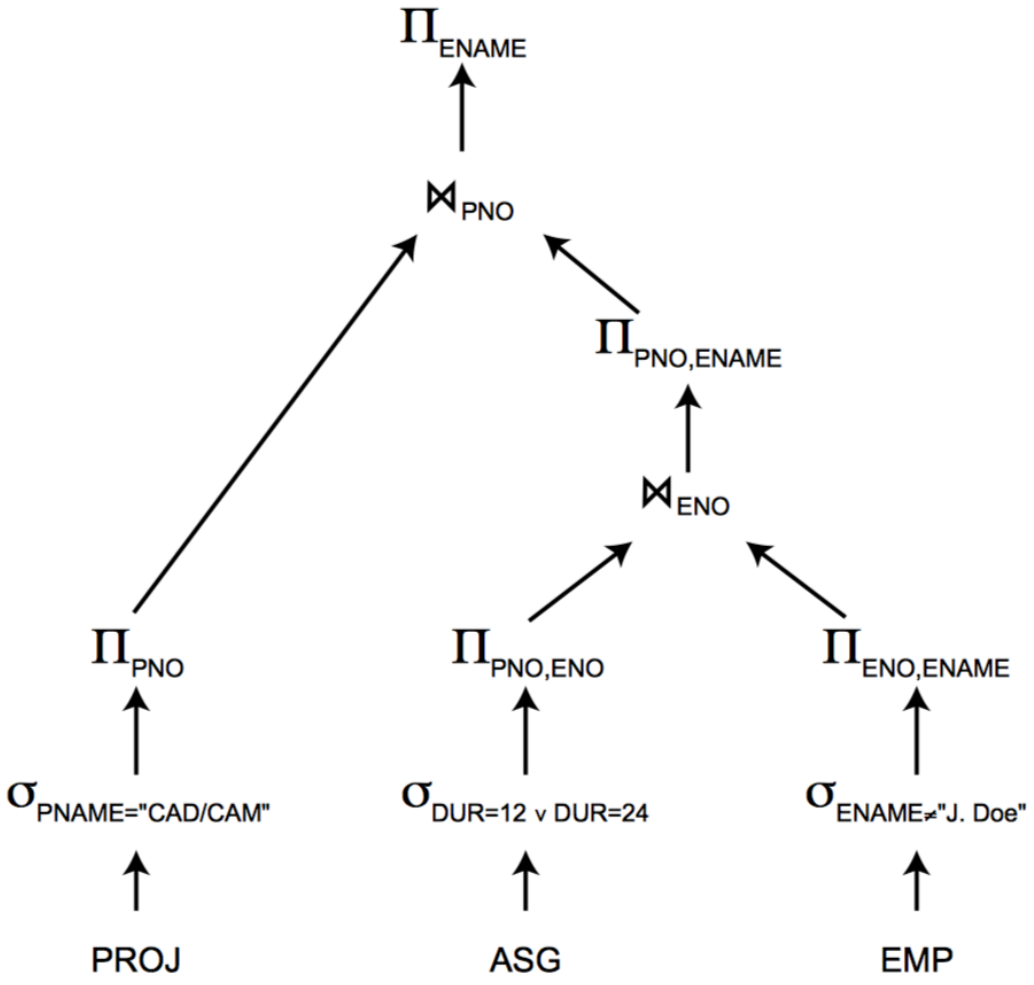
4. Rewriting
   - Rewrite to relational algebra
   - Operation Tree

```
SELECT ENAME
FROM   PROJ, ASG, EMP
WHERE  ASG.ENO = EMP.ENO
AND    ASG.PNO = PROJ.PNO
AND    ENAME != "J. Doe"
AND    PROJ.PNAME = "CAD/CAM"
AND    (DUR = 12 OR DUR = 24)
```

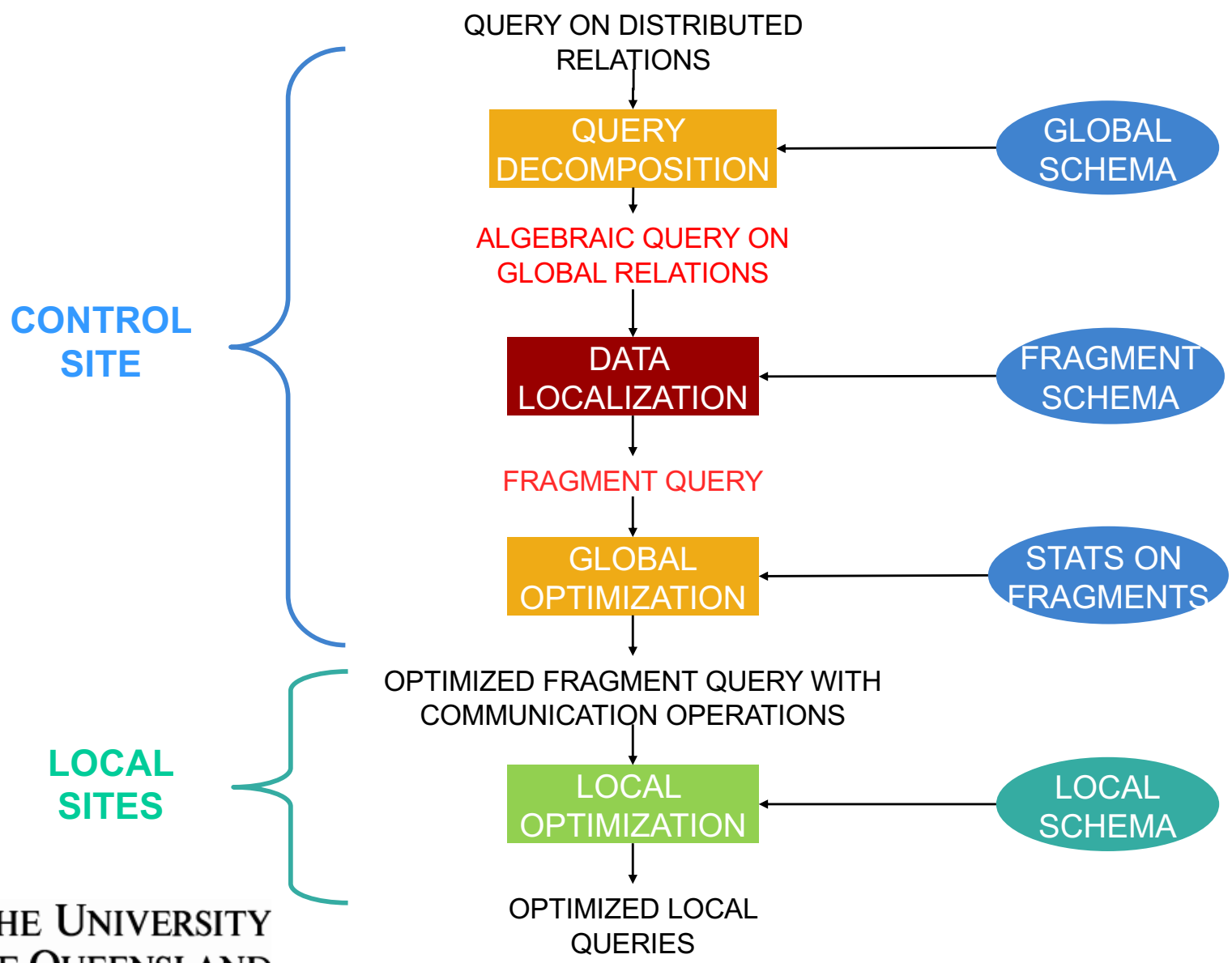


+ Q1-1 Query Decomposition Steps

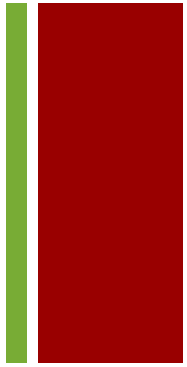
- Rewriting
  - Separate the unary operations
    - Simplify query expression
  - Group unary operation on same relation together
  - Commute binary with unary
    - Selection/Projection first
  - Reorder binary operations



Q1 Layers of Query Processing



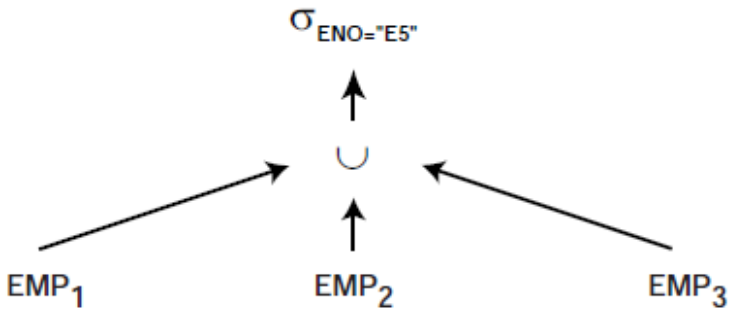
+ Q1-2 Data Localization



- Consider the distribution of data
  - Fragments
- Input: Relational algebra query from data decomposition
- Output: An algebraic query expressed on physical fragments
- Objective:
  - Consider the data distribution
  - Localize the query's data using data distribution information in the fragment schema

+ Q1-2 Data Localization

- Localized Query
  - Replace the leaves of the operator tree with subtrees corresponding to the localization programs
  - Not efficient
- Reduced Query
  - Push unary operation down （Selection and Projection）
  - Reduce HF with selection and join
  - Reduce VF with Join

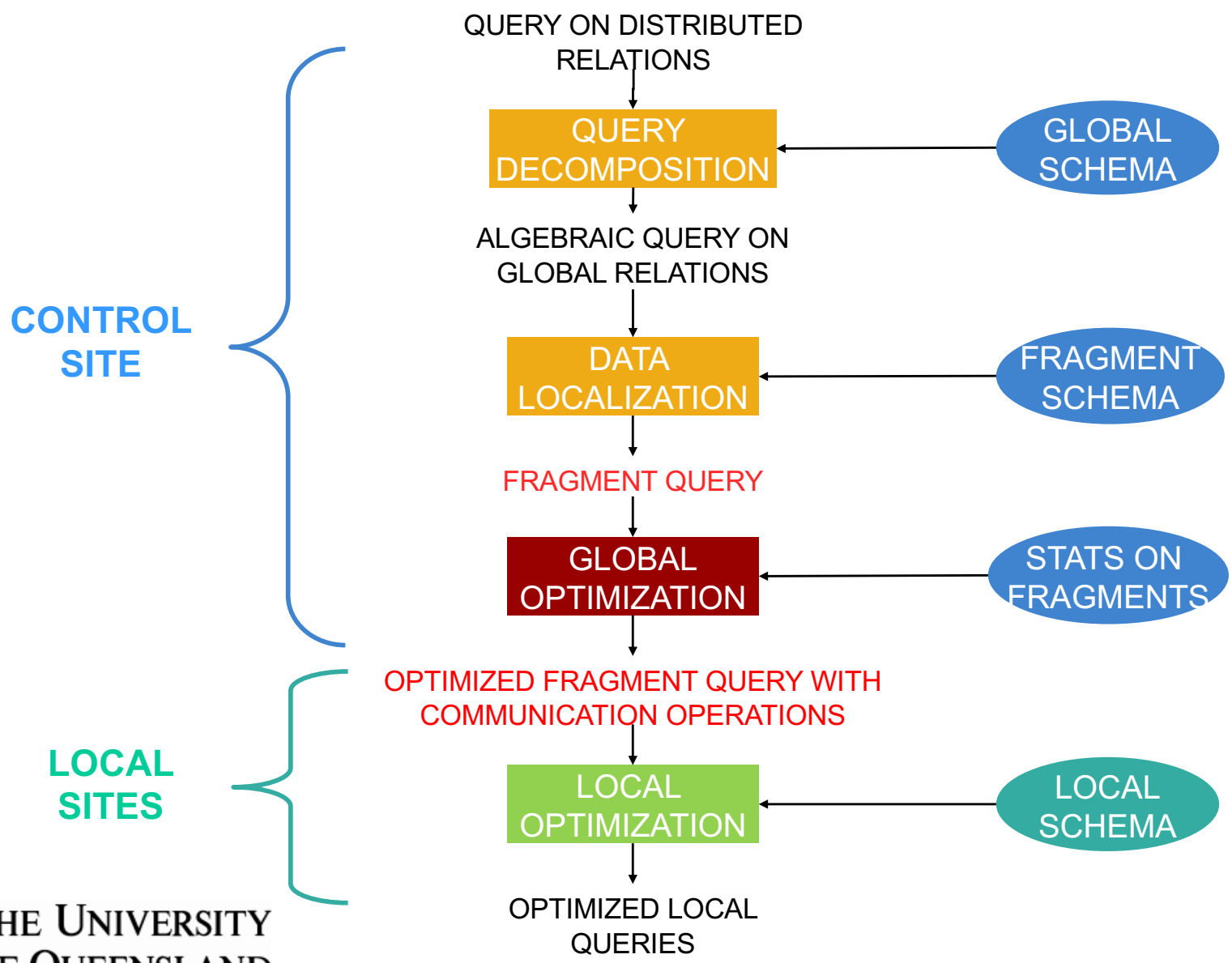


(a) Localized query

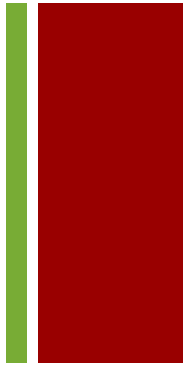
(b) Reduced query

THE UNIVERSITY OF QUEENSLAND
AUSTRALIA

Q1 Layers of Query Processing



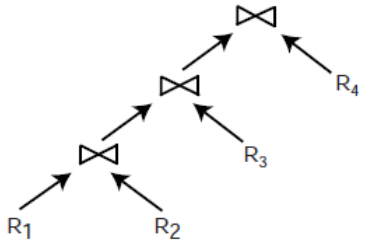
+ Q1-3 Global Optimization



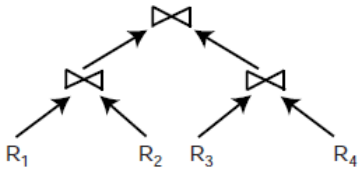
- Input: Reduced Queries on fragments
- Output:
  - Query execution plan minimizes the objective cost function
  - Communication operations

+ Q1-3 Global Optimization

- Component
  1. Search Space / Solution Space
     - Set of alternative execution plans
  2. Cost Model/Function
     - I/O cost + CPU cost + Communication cost
     - Join order
     - Semi-Join
  3. Search Strategy/Algorithm
     - Dynamic programming, Greedy, Randomized



(a) linear join tree



(b) bushy join tree

- Objective:
  - Find the best (not necessarily optimal) global execution schedule/query plan
  - Minimize a cost function

Q2 Localization

- PROJ1 = $\sigma_{PNO \leq "P2"}$ PROJ
- PROJ2 = $\sigma_{PNO > "P2"}$ PROJ

EMP

ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng
E2	M. Smith	Syst. Anal.
E3	A. Lee	Mech. Eng.
E4	J. Miller	Programmer
E5	B. Casey	Syst. Anal.
E6	L. Chu	Elect. Eng.
E7	R. Davis	Mech. Eng.
E8	J. Jones	Syst. Anal.

ASG

ENO	PNO	RESP	DUR
E1	P1	Manager	12
E2	P1	Analyst	24
E2	P2	Analyst	6
E3	P3	Consultant	10
E3	P4	Engineer	48
E4	P2	Programmer	18
E5	P2	Manager	24
E6	P4	Manager	48
E7	P3	Engineer	36
E8	P3	Manager	40

PROJ

PNO	PNAME	BUDGET
P1	Instrumentation	150000
P2	Database Develop.	135000
P3	CAD/CAM	250000
P4	Maintenance	310000

PAY

TITLE	SAL
Elect. Eng.	40000
Syst. Anal.	34000
Mech. Eng.	27000
Programmer	24000

+ Q2-1

- Transform the following query into a reduced query on fragments:

  - SELECT ENO, PNAME
  - FROM PROJ, ASG
  - WHERE

    PROJ.PNO = ASG.PNO

    AND PNO = "P4"

EMP

ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng
E2	M. Smith	Syst. Anal.
E3	A. Lee	Mech. Eng.
E4	J. Miller	Programmer
E5	B. Casey	Syst. Anal.
E6	L. Chu	Elect. Eng.
E7	R. Davis	Mech. Eng.
E8	J. Jones	Syst. Anal.

ASG

ENO	PNO	RESP	DUR
E1	P1	Manager	12
E2	P1	Analyst	24
E2	P2	Analyst	6
E3	P3	Consultant	10
E3	P4	Engineer	48
E4	P2	Programmer	18
E5	P2	Manager	24
E6	P4	Manager	48
E7	P3	Engineer	36
E8	P3	Manager	40

PROJ

PNO	PNAME	BUDGET
P1	Instrumentation	150000
P2	Database Develop.	135000
P3	CAD/CAM	250000
P4	Maintenance	310000

PAY

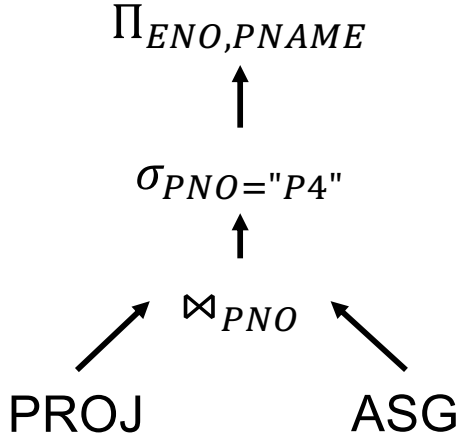
TITLE	SAL
Elect. Eng.	40000
Syst. Anal.	34000
Mech. Eng.	27000
Programmer	24000

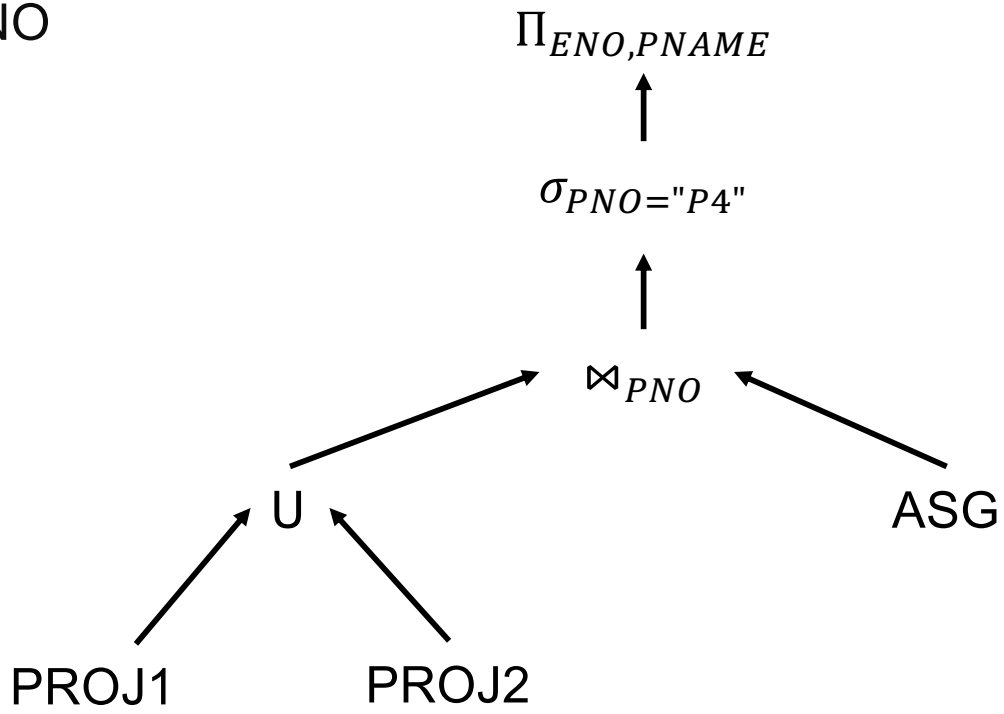
+ Q2-1

- Localized query
  - SELECT ENO, PNAME
  - FROM PROJ1 U PROJ2, ASG
  - WHERE

    PROJ.PNO = ASG.PNO

    AND PNO = "P4"





Q2-1 Reduced Query

- Reduce with selection
  - SELECT ENO, PNAME
  - FROM PROJ, ASG
  - WHERE

    PROJ.PNO = ASG.PNO

    AND PNO = "P4"

(PROJ1 = $\sigma_{PNO \leq "P2"}$ PROJ) $\wedge$ (PNO = "P4") $=\Phi$

PROJ2 = $\sigma_{PNO > "P2"}$ PROJ

~~PROJ1, ASG~~

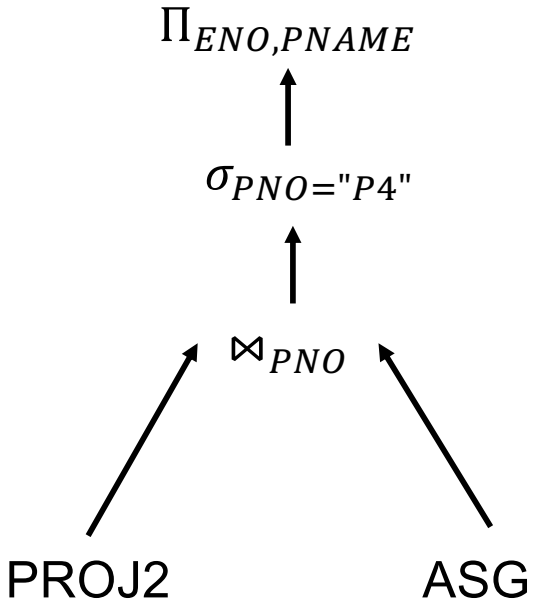
PROJ2, ASG

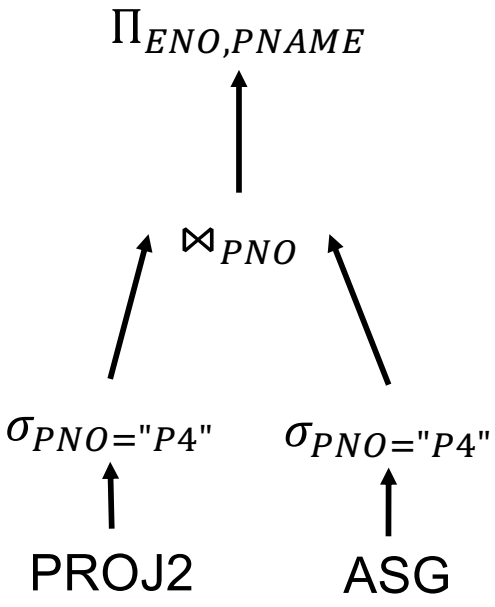
+ Q2-1 Reduced Query

- SELECT ENO, PNAME
- FROM PROJ2, ASG
- WHERE

  PROJ2.PNO = ASG.PNO

  AND PNO = "P4"





Reduced query

+ Q2.2

- PROJ1 = $\sigma_{PNO \leq "P2"}$ PROJ
- PROJ2 = $\sigma_{PNO > "P2"}$ PROJ
- ASG1 = $\sigma_{PNO \leq "P2"}$ ASG
- ASG2 = $\sigma_{"P2" < PNO \leq "P3"}$ ASG
- ASG3 = $\sigma_{PNO > "P3"}$ ASG

EMP

ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng
E2	M. Smith	Syst. Anal.
E3	A. Lee	Mech. Eng.
E4	J. Miller	Programmer
E5	B. Casey	Syst. Anal.
E6	L. Chu	Elect. Eng.
E7	R. Davis	Mech. Eng.
E8	J. Jones	Syst. Anal.

ASG

ENO	PNO	RESP	DUR
E1	P1	Manager	12
E2	P1	Analyst	24
E2	P2	Analyst	6
E3	P3	Consultant	10
E3	P4	Engineer	48
E4	P2	Programmer	18
E5	P2	Manager	24
E6	P4	Manager	48
E7	P3	Engineer	36
E8	P3	Manager	40

PROJ

PNO	PNAME	BUDGET
P1	Instrumentation	150000
P2	Database Develop.	135000
P3	CAD/CAM	250000
P4	Maintenance	310000

PAY

TITLE	SAL
Elect. Eng.	40000
Syst. Anal.	34000
Mech. Eng.	27000
Programmer	24000

+ Q2-2

- Transform the following query into a reduced query on fragments, and determine whether it is better than the localized query:

  - SELECT RESP, BUDGET
  - FROM ASG, PROJ
  - WHERE

    ASG.PNO = PROJ.PNO

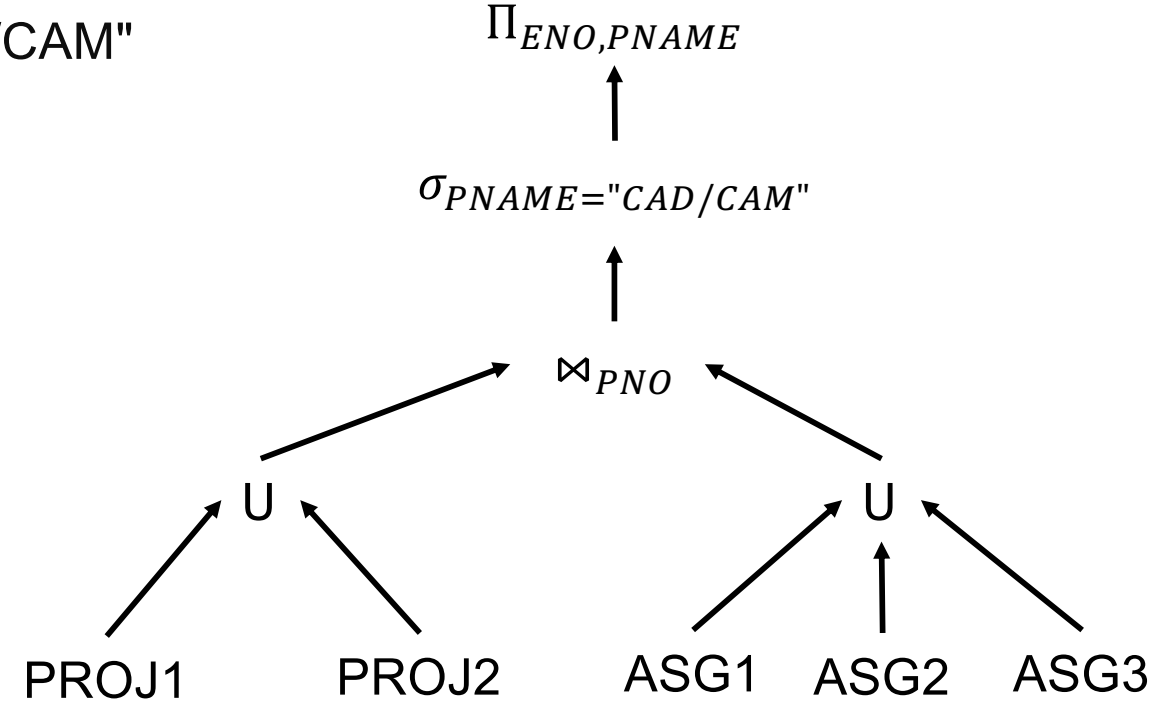
    AND PNAME = "CAD/CAM"

+ Q2-2

- Localized query:
  - SELECT RESP, BUDGET
  - FROM PROJ1 U PROJ2, ASG1 U ASG2 U ASG3
  - WHERE

    ASG.PNO = PROJ.PNO

    AND PNAME = "CAD/CAM"



Q2-2

- Reduce with join:
  - SELECT RESP, BUDGET
  - FROM PROJ, ASG
  - WHERE

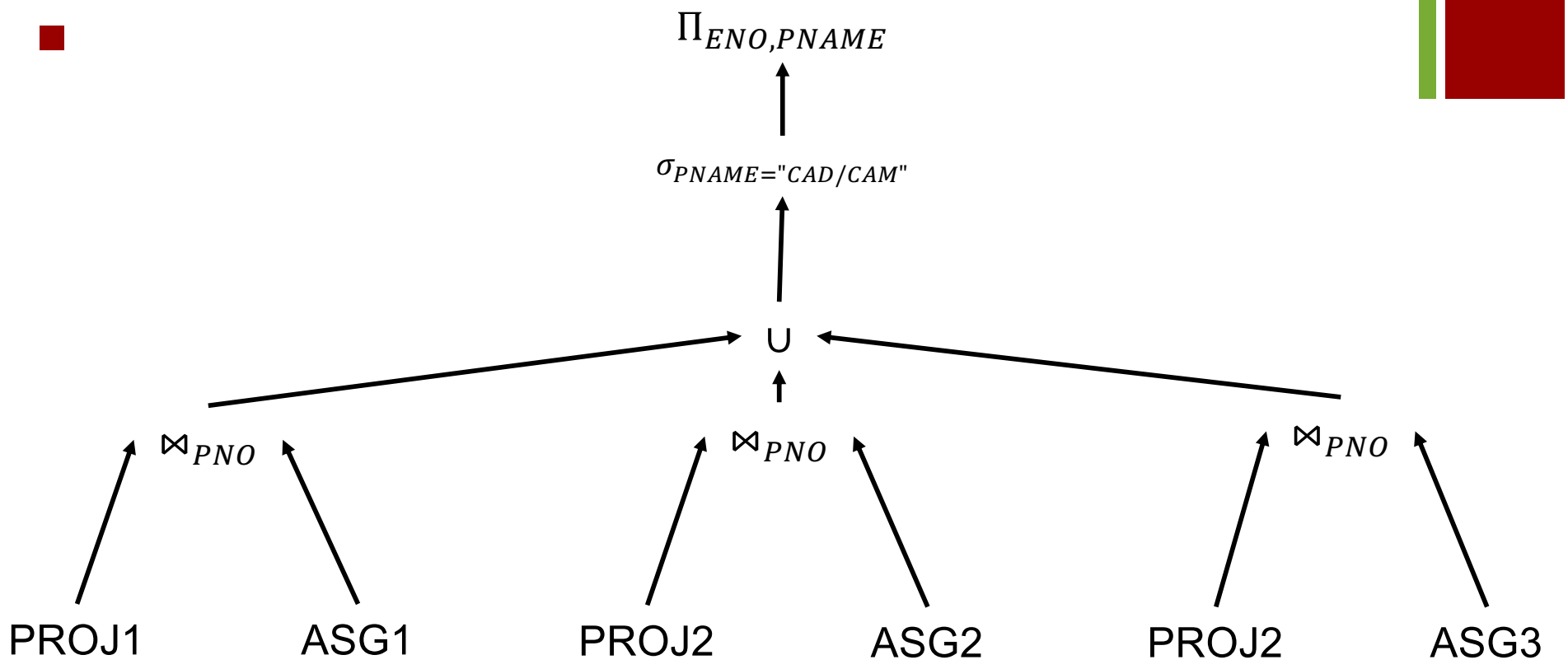
    ASG.PNO = PROJ.PNO

    AND PNAME = "CAD/CAM"

PROJ1, ASG1
PROJ1, ASG2
PROJ1, ASG3
PROJ2, ASG1
PROJ2, ASG2
PROJ2, ASG3

(PROJ1 = $\sigma_{PNO \leq "P2"}$ PROJ) ∧ (ASG1 = $\sigma_{PNO \leq "P2"}$ ASG)
(PROJ2 = $\sigma_{PNO > "P2"}$ PROJ) ∧ (ASG1 = $\sigma_{PNO \leq "P2"}$ ASG)
(PROJ1 = $\sigma_{PNO \leq "P2"}$ PROJ) ∧ (ASG2 = $\sigma_{"P2" < PNO \leq "P3"}$ ASG)
(PROJ2 = $\sigma_{PNO > "P2"}$ PROJ) ∧ (ASG2 = $\sigma_{"P2" < PNO \leq "P3"}$ ASG)
(PROJ1 = $\sigma_{PNO \leq "P2"}$ PROJ) ∧ (ASG3 = $\sigma_{PNO > "P3"}$ ASG)
(PROJ2 = $\sigma_{PNO > "P2"}$ PROJ) ∧ (ASG3 = $\sigma_{PNO > "P3"}$ ASG)

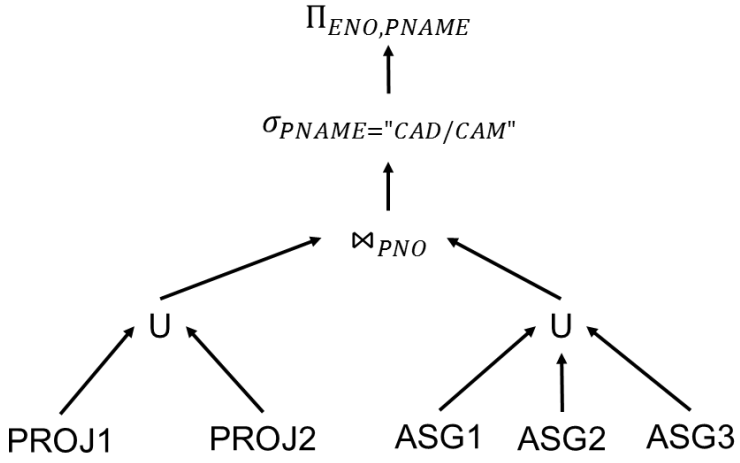
+ Q2-2 Reduced Query



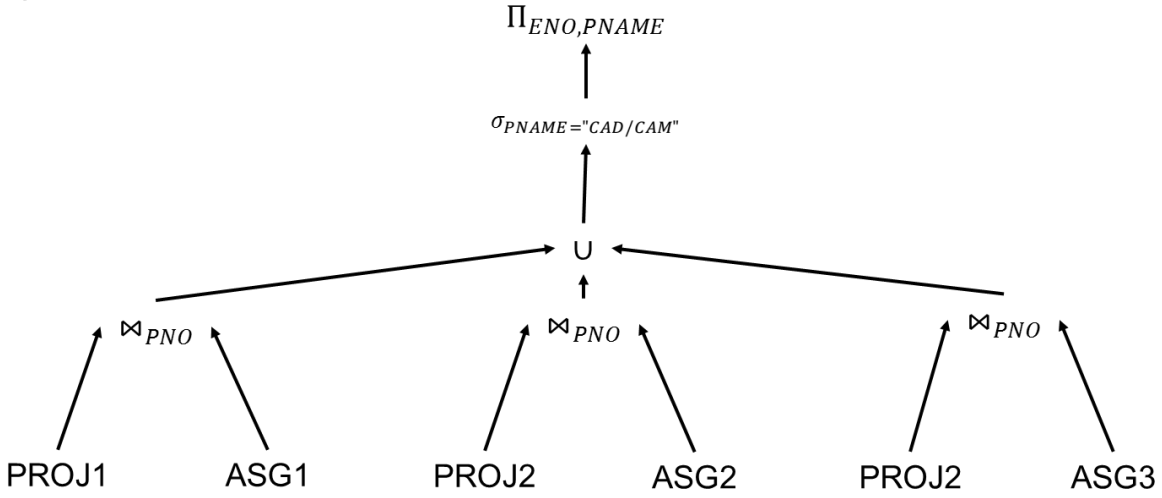
(PROJ1 = $\sigma_{PNO\leq"P2"}$ PROJ) ∧ (ASG1 = $\sigma_{PNO\leq"P2"}$ ASG)
(PROJ2 = $\sigma_{PNO>"P2"}$ PROJ) ∧ (ASG2 = $\sigma_{"P2"<PNO\leq"P3"}$ ASG)
(PROJ2 = $\sigma_{PNO>"P2"}$ PROJ) ∧ (ASG3 = $\sigma_{PNO>"P3"}$ ASG)

Q2-2 Reduced Query

- Better?
  - Centralized Computation & Network is slow
    - Same Communication
  - Parallel Computation
    - Compute in parallel
    - Eliminate unnecessary work



Localized query

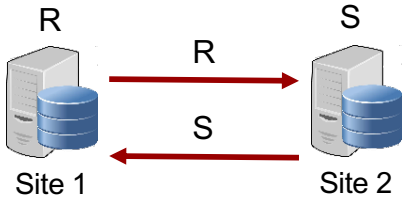


Reduced query

+ Q3 Semi-Join

- Join
  - R and S on the same server
    - From disk to memory
    - Communication Cost: 0

- R and S on different servers
  - Same join result
  - Send data over network
  - Network is much slower than disk and memory
    - Network Cost dominate
  - Site 1 to Site 2:
    - Send R
    - Cost: 10
  - Or Site 2 to Site 1:
    - Send S
    - Cost: 12



R(A,B)

A	B
1	4
1	5
2	4
2	6
3	7

S(B,C,D)

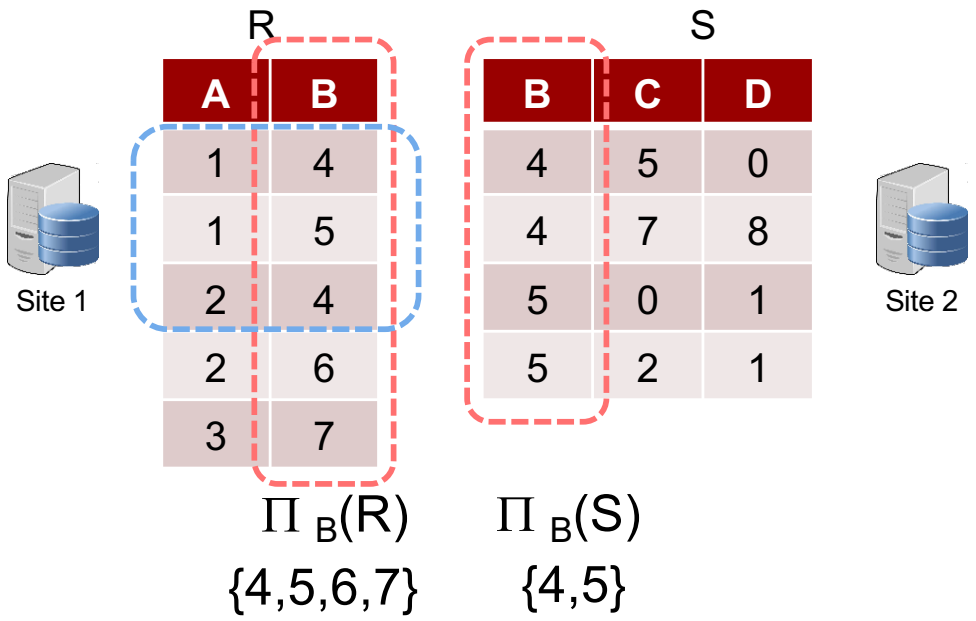
B	C	D
4	5	0
4	7	8
5	0	1
5	2	1

$R \bowtie S$

A	B	C	D
1	4	5	0
1	4	7	8
1	5	0	1
1	5	2	1
2	4	5	0
2	4	7	8

+ Q3

- Network Transfer cost dominates the total cost
- How to reduce the data size sent over network?
- For R
  - Only {(1,4) (1,5) (2,4)} is useful
  - But Site 1 does not know it
- The common attribute B acts as a filter
  - If Site 1 knows S.B has {4,5}
  - No need to send {(2,6)(3,7)} to Site 2
    - Site 2 to Site 1
      - {4,5}
      - Cost: 2
    - Site 1 to Site 2
      - {(1,4) (1,5) (2,4)}
      - Cost: 6
    - Total Cost: 8



THE UNIVERSITY OF QUEENSLAND
AUSTRALIA

+ Q3-1

1. R ∝ S
   - = $\Pi_{R(A,B)}(R \bowtie S)$
   1. Select the values of S.B
      - {4,5}
   2. Send S.B from Site 2 to Site 1
   3. Join R ⋈ S.B

   - Only keeps the left table's attributes

R

A	B
1	4
1	5
2	4
2	6
3	7

S

B	C	D
4	5	0
4	7	8
5	0	1
5	2	1

$\Pi_B(S)$

{4,5}

Site 1 ← Site 2

R ∝ S

A	B
1	4
1	5
2	4

+ Q3-2

2. $S \propto R$
   - $= \Pi_{B,C,D}(S \bowtie R)$
   - Select the values of R.B
     - {4,5,6,7}
   - Send R.B from Site 1 to Site 2
   - Join S ⋈ R.B

R

A	B
1	4
1	5
2	4
2	6
3	7

$\Pi_B(R)$

S

B	C	D
4	5	0
4	7	8
5	0	1
5	2	1

{4,5,6,7}

Site 1 → Site 2

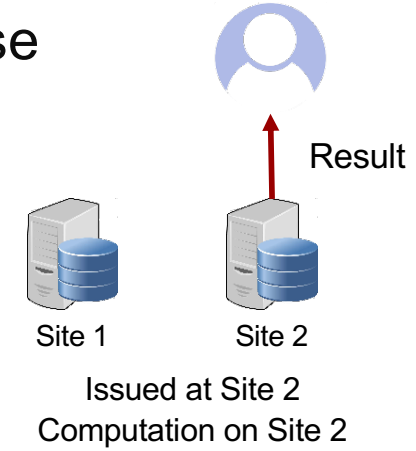
S ∝ R

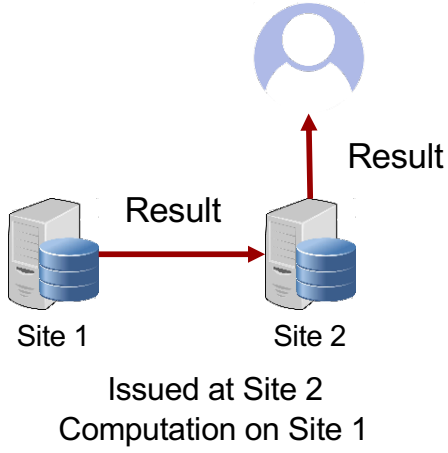
B	C	D
4	5	0
4	7	8
5	0	1
5	2	1

+ Q3-3 Semi-Join for Distributed DB

3. Assume R is at site 1 and S is at site 2, and **a query R ⋈ S has been issued at site 2**.
   1. Give steps for a query processing strategy using semi-join,
   2. Check if the semi-join is a beneficial option in this case (ignore local processing cost).

- Issue at Site 2
  - Site 2 is responsible to return the query result
  - Join Computation should be on Site 2
    - Otherwise, send results from Site 1 to Site 2
    - Additional useless cost
- Use semi-join to reduce the network cost





THE UNIVERSITY OF QUEENSLAND
AUSTRALIA

+ Q3-3

R

A	B
1	4
1	5
2	4
2	6
3	7

S

B	C	D
4	5	0
4	7	8
5	0	1
5	2	1

$\Pi_B(S)$


Site 1

{4,5}

Cost:2


Site 2

$R \bowtie S$

$R \propto S.B$

A	B
1	4
1	5
2	4

{(1,4),(1,5),(2,4)}

Cost:6

Total Cost:8 < 10


Site 2

Steps:

1. Site 2 sends t1 = $\Pi_B(S)$ to Site 1;
2. Site 1 sends t2 = $R \propto t1$ to Site 2;
3. Site 2 returns $t2 \bowtie S$ to the user.

$R \propto S.B \bowtie$

B	C	D
4	5	0
4	7	8
5	0	1
5	2	1

A	B	C	D
1	4	5	0
1	4	7	8
1	5	0	1
1	5	2	1
2	4	5	0
2	4	7	8


THE UNIVERSITY OF QUEENSLAND
AUSTRALIA