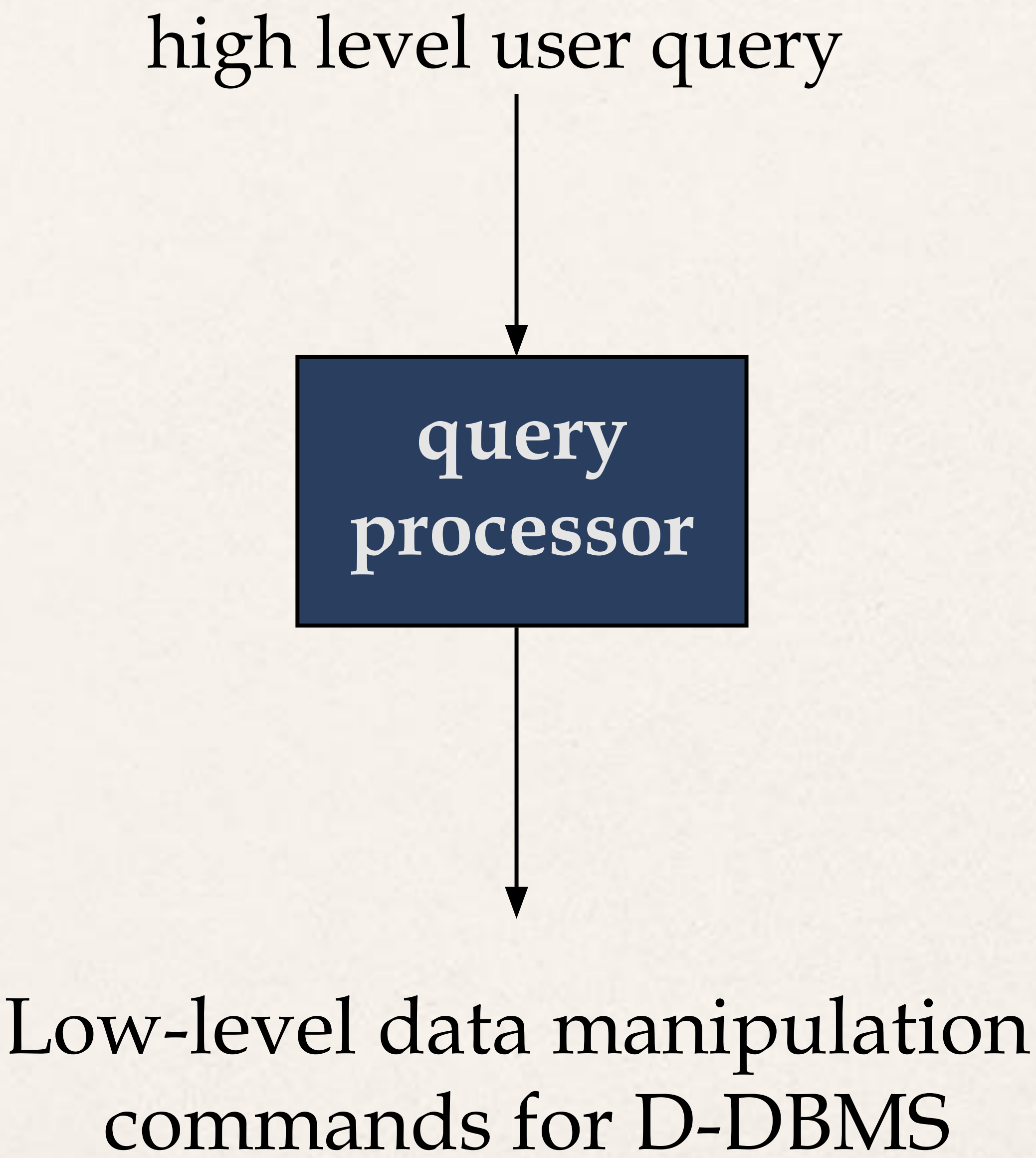
Outline

- Introduction
- Background
- Distributed Database Design
- Database Integration
- Semantic Data Control
- Distributed Query Processing
  - Overview
  - Query decomposition and localization
  - Distributed query optimization
- Multidatabase Query Processing
- Distributed Transaction Management
- Data Replication
- Parallel Database Systems
- Distributed Object DBMS
- Peer-to-Peer Data Management
- Web Data Management
- Current Issues

Query Processing in a DDBMS



Query Processing Components

- Query language that is used
  - SQL: "intergalactic dataspeak "
- Query execution methodology
  - The steps that one goes through in executing high-level (declarative) user queries.
- Query optimization
  - How do we determine the "best" execution plan?
- We assume a homogeneous D-DBMS

Selecting Alternatives

```
SELECT  ENAME
FROM    EMP,ASG
WHERE   EMP.ENO = ASG.ENO
AND   RESP = "Manager"
```

Strategy 1

- $\Pi_{ENAME}(EMP \bowtie_{ENO} (\sigma_{RESP=\text{“Manager”}} (ASG))$

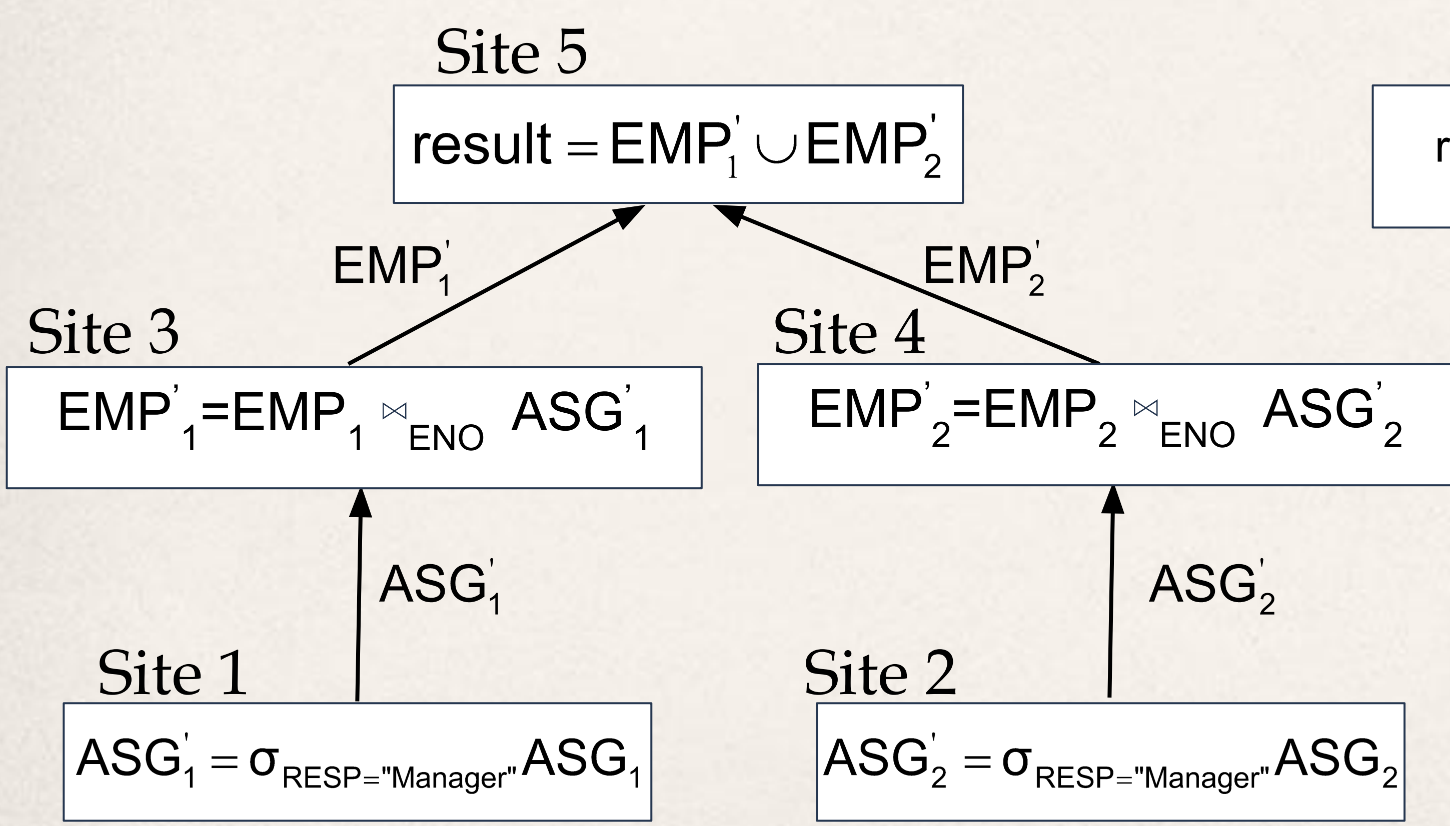
Strategy 2

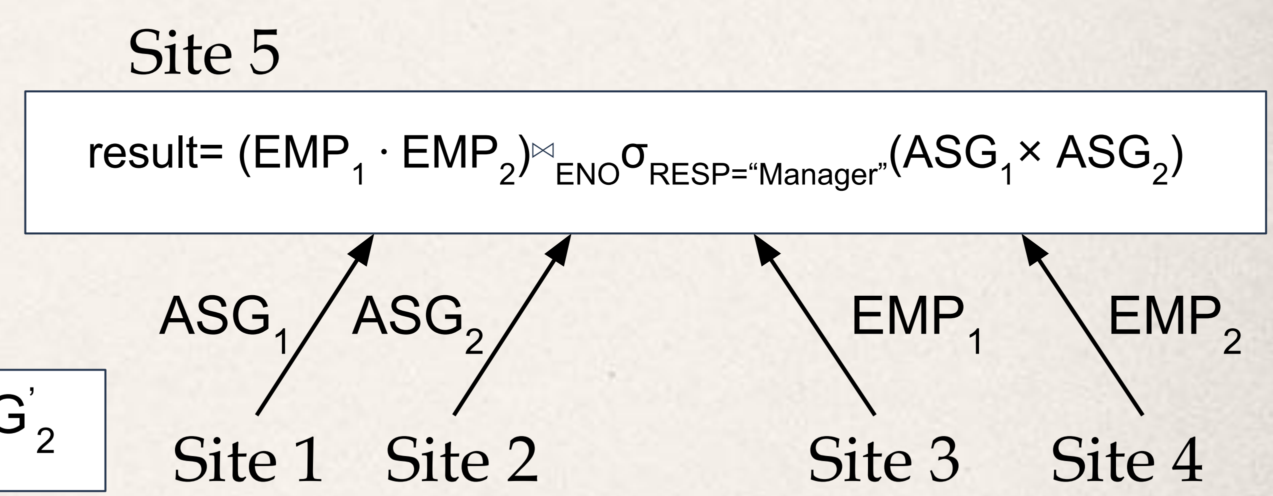
$\Pi_{ENAME}(\sigma_{RESP=\text{“Manager”} \wedge EMP.ENO=ASG.ENO}(EMP \times ASG))$

Strategy 1 avoids Cartesian product, so may be “better”

What is the Problem?

Site 1	Site 2	Site 3	Site 4	Site 5
$ASG_1 = \sigma_{ENO \leq \text{“E3”}}(ASG)$	$ASG_2 = \sigma_{ENO > \text{“E3”}}(ASG)$	$EMP_1 = \sigma_{ENO \leq \text{“E3”}}(EMP)$	$EMP_2 = \sigma_{ENO > \text{“E3”}}(EMP)$	Result





Cost of Alternatives

- Assume
  - *size*(EMP) = 400, *size*(ASG) = 1000
  - tuple access cost = 1 unit; tuple transfer cost = 10 units
- Strategy 1
  - produce ASG': (10+10) * tuple access cost 20
  - transfer ASG' to the sites of EMP: (10+10) * tuple transfer cost 200
  - produce EMP': (10+10) * tuple access cost * 2 40
  - transfer EMP' to result site: (10+10) * tuple transfer cost   200

  Total Cost 460
- Strategy 2
  - transfer EMP to site 5: 400 * tuple transfer cost   4,000
  - transfer ASG to site 5: 1000 * tuple transfer cost   10,000
  - produce ASG': 1000 * tuple access cost 1,000
  - join EMP and ASG': 400 * 20 * tuple access cost   8,000

  Total Cost 23,000

Query Optimization Objectives

- Minimize a cost function

  I/O cost + CPU cost + communication cost

These might have different weights in different distributed environments

- Wide area networks
  - communication cost may dominate or vary much
    - bandwidth
    - speed
    - high protocol overhead
- Local area networks
  - communication cost not that dominant
  - total cost function should be considered
- Can also maximize throughput

Complexity of Relational Operations

- Assume
  - relations of cardinality n
  - sequential scan

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

Query Optimization Issues – Types Of Optimizers

- Exhaustive search
  - Cost-based
  - Optimal
  - Combinatorial complexity in the number of relations
- Heuristics
  - Not optimal
  - Regroup common sub-expressions
  - Perform selection, projection first
  - Replace a join by a series of semijoins
  - Reorder operations to reduce intermediate relation size
  - Optimize individual operations

Query Optimization Issues – Optimization Granularity

- Single query at a time
  - Cannot use common intermediate results
- Multiple queries at a time
  - Efficient if many similar queries
  - Decision space is much larger

Query Optimization Issues – Optimization Timing

- Static
  - Compilation ⇒ optimize prior to the execution
  - Difficult to estimate the size of the intermediate results ⇒ error propagation
  - Can amortize over many executions
  - R*
- Dynamic
  - Run time optimization
  - Exact information on the intermediate relation sizes
  - Have to reoptimize for multiple executions
  - Distributed INGRES
- Hybrid
  - Compile using a static algorithm
  - If the error in estimate sizes > threshold, reoptimize at run time
  - Mermaid

Query Optimization Issues – Statistics

- Relation
  - Cardinality
  - Size of a tuple
  - Fraction of tuples participating in a join with another relation
- Attribute
  - Cardinality of domain
  - Actual number of distinct values
- Common assumptions
  - Independence between different attribute values
  - Uniform distribution of attribute values within their domain

Query Optimization Issues – Decision Sites

- Centralized
  - Single site determines the "best" schedule
  - Simple
  - Need knowledge about the entire distributed database
- Distributed
  - Cooperation among sites to determine the schedule
  - Need only local information
  - Cost of cooperation
- Hybrid
  - One site determines the global schedule
  - Each site optimizes the local subqueries

Query Optimization Issues – Network Topology

- Wide area networks (WAN) – point-to-point
  - Characteristics
    - Low bandwidth
    - Low speed
    - High protocol overhead
  - Communication cost will dominate; ignore all other cost factors
  - Global schedule to minimize communication cost
  - Local schedules according to centralized query optimization
- Local area networks (LAN)
  - Communication cost not that dominant
  - Total cost function should be considered
  - Broadcasting can be exploited (joins)
  - Special algorithms exist for star networks

Distributed Query Processing Methodology