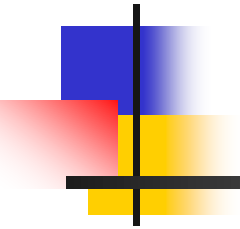


Mobile and Heterogeneous databases

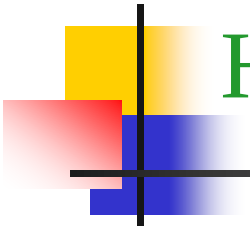
Heterogeneous Distributed Databases

Query Processing

A.R. Hurson

Computer Science

Missouri Science & Technology



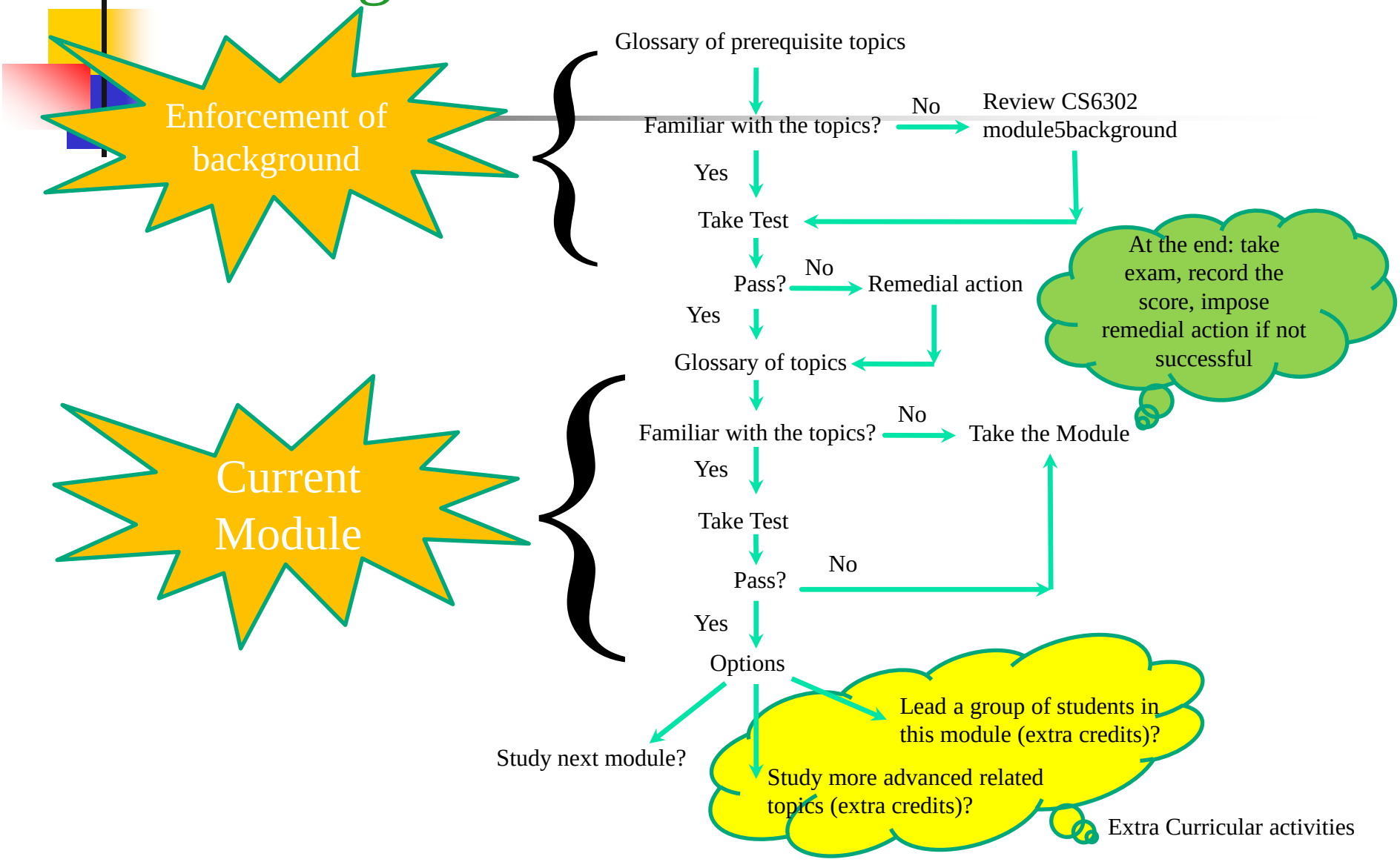
Heterogeneous Distributed Databases

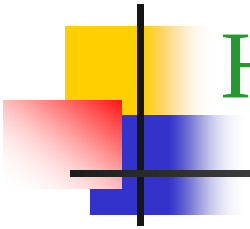
Note, this unit will be covered in two lectures. In case you finish it earlier, then you have the following options:

1) Take the early test and start CS6302.module6
2) Study the supplement module (supplement CS6302.module5)
3) Act as a helper to help other students in studying CS6302.module5

Note, options 2 and 3 have extra credits as noted in course outline.

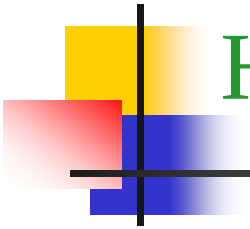
Heterogeneous Distributed Databases





Heterogeneous Distributed Databases

- You are expected to be familiar with:
  - Heterogeous Distributed Databases,
  - Query processing and Transaction processing in homogeneous distributed databases
- If not, you need to study CS6302.module4



Heterogeneous Distributed Databases

- At this point you should be fully familiar with the concept of Heterogeneous Distributed Databases (multidatabases) and parameters that distinguish multidatabases from the so called homogeneous distributed databases.
- As a reminder, autonomy and heterogeneity are two major factors that distinguish the aforementioned platforms from each other.

Heterogeneous Distributed Databases

- Within the scope of multidatabases you are also expected to be familiar with the “issues” of concern.
- If you recall, many of these “issues” are traced back to the “homogeneous distributed databases”. However, autonomy and heterogeneous nature of multidatabases make it much harder to deal with these “issues”.

Heterogeneous Distributed Databases

- Finally, You are excepted to be familiar with different approached to “global information sharing” process and different solutions for handling “multidatabases”.

Heterogeneous Distributed Databases

- In this module, we will concentrate on query processing in multidatabases. The goal is to identify aspects of query processing in "homogeneous distributed databases" that can be transported and difficulties that are due to the heterogeneous and autonomy characteristics of multidatabases.

Heterogeneous Distributed Databases

- MultiDatabase Systems – Query processing
  - Many of the distribution query processing and optimization techniques within the scope of distributed systems can be carried over to multidatabases. However, there are some important differences.
  - Let us review query processing in centralized and distributed databases.

Heterogeneous Distributed Databases

- MultiDatabase Systems – Query processing
  - Query processing in centralized databases involves three steps:
    - Query decomposition,
    - Query optimization, and
    - Query execution.
  - Query processing in distributed databases involves four steps:
    - Query decomposition/Data localization,
    - Global optimization,
    - Local optimization, and
    - Query execution

Heterogeneous Distributed Databases

- MultiDatabase Systems – Query processing
  - Assume the following query and the two relations involved:

    Find names of employees who are managing a project

EMP

ENO	ENAME	Title

ASG

ENO	PNO	RESP	ADDRESS

Heterogeneous Distributed Databases

- MultiDatabase Systems – Query processing
  - In SQL the aforementioned query is represented as:

```
SELECT   ENAME
FROM     EMP, ASG
WHERE    EMP.ENO = ASG.ENO
AND      RESP = ‘Manager’;
```

Heterogeneous Distributed Databases

- MultiDatabase Systems – Query processing
  - In relational algebra form the query can be represented in two forms as follows:

$$\Pi_{name}(\sigma_{RESP = \text{“Manager”} \wedge EMP.NO = ASG.NO}(EMP \times ASG))$$

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

Heterogeneous Distributed Databases

- MultiDatabase Systems – Query processing
  - In a centralized database environment, the choice is clear. Second strategy avoids Cartesian product and hence it is much less computing resource intensive than the first strategy.
  - In distributed environment, as we discussed before, other parameters need to be taken into considerations in order to define a suitable strategy, i.e., Data Transfer cost, site computational capability, …

Heterogeneous Distributed Databases

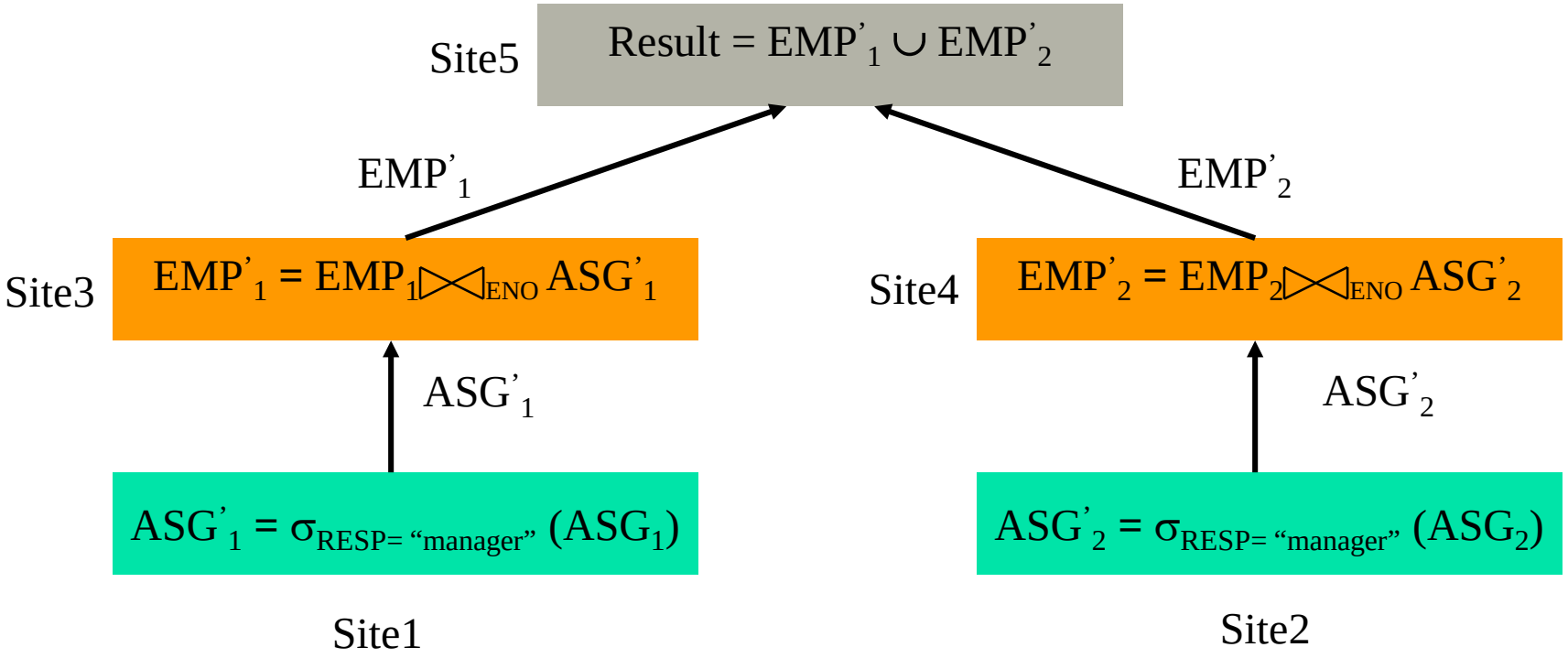
- MultiDatabase Systems – Query processing
  - Based on the query, location of data sets, size of the data sets, communication cost, processing capability, … a dynamic strategy should be laid out.
  - According to the strategy, then the query is decomposed into sub-queries.
  - Sub-queries are sent to the designated sites for execution.

Heterogeneous Distributed Databases

- MultiDatabase Systems – Query processing
  - Furthermore, assume that the relations are horizontally fragmented as follows:
    - EMP1: $\sigma_{No \leq \text{“E3”}}$ (EMP) $site_3$
    - EMP2: $\sigma_{No > \text{“E3”}}$ (EMP) $site_4$
    - ASG1: $\sigma_{No \leq \text{“E3”}}$ (ASG) $site_1$
    - ASG2: $\sigma_{No > \text{“E3”}}$ (ASG) $site_2$

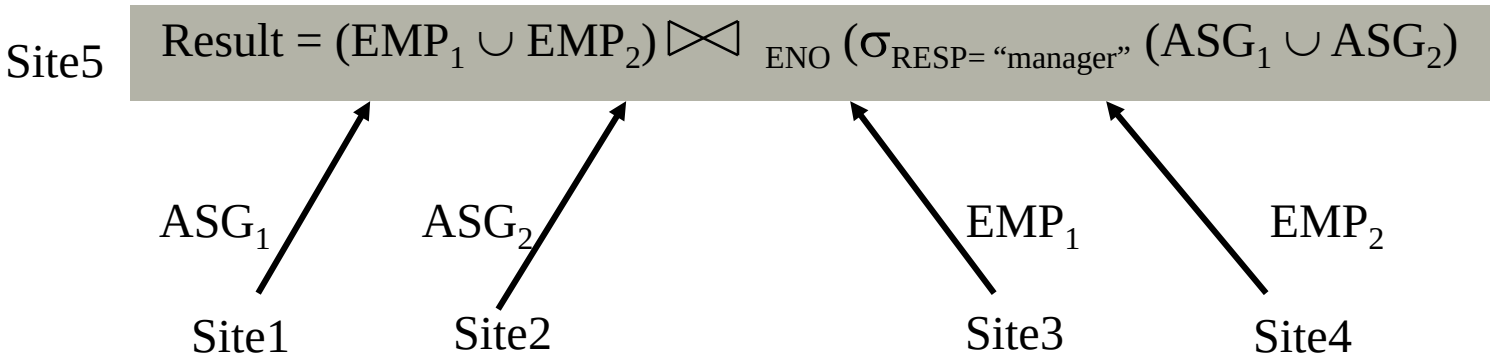
Heterogeneous Distributed Databases

- MultiDatabase Systems – Query processing
  - Now there are choices to execute this query:



Heterogeneous Distributed Databases

- MultiDatabase Systems – Query processing



Heterogeneous Distributed Databases

- MultiDatabase Systems – Query processing
  - Query processing in multidatabases is more different and complicated than the one we studied in traditional distributed databases. This complexity is due to the heterogeneity and local autonomy aspects of multidatabases.

Heterogeneous Distributed Databases

- MultiDatabase Systems – Query processing
  - Because of heterogeneity:
    - Data representation in different local databases may be different,
    - The capability of component databases may be different,
    - Cost of processing queries on different local databases may be different,
    - There may be difficulties in moving data between local databases,
    - The local optimization capability of local databases might be quite different.
    -
    -
    -

Heterogeneous Distributed Databases

- MultiDatabase Systems – Query processing
  - Local autonomy poses additional problems. For example:
    - As a result of communication autonomy and/or association autonomy the local database may terminate its services at any time. This requires query processing methods that are tolerant to system unavailability. Therefore, the challenge is to respond to a user query when the component database is unavailable, unwilling, and uncooperative.

Heterogeneous Distributed Databases

- MultiDatabase Systems – Query optimization
  - The design autonomy may restrict the availability and accuracy of statistical information needed in order to carry out the query optimization.
  - The execution autonomy may limit the application of some query processing and optimization strategies. For example, it may not be possible to perform semi-join operation.

Heterogeneous Distributed Databases

- MultiDatabase Systems – Query processing
  - Global query is resolved (split) with the help of the global schema (schema integration). Resolution of the global query results in a set of sub-queries to be executed at the local sites.

Heterogeneous Distributed Databases

- MultiDatabase Systems – Query processing
  - Query processing at the global level is a sequence of four step process:
    - Compilation and translation,
    - Unification decomposition,
    - Optimization, and
    - Translation and execution.

Heterogeneous Distributed Databases

- MultiDatabase Systems – Query processing
  - Compilation and translation: Query is compiled and transformed into an internal form.
  - Unification decomposition: Integrated data items are replaced from corresponding local data items along with inconsistency resolution functions (if any).
  - Optimization: The query tree is optimized and analyzed. At this stage, sub-trees to be resolved by local databases are identified.
  - Translation and execution: Executable sub-trees to be executed at local databases are constructed and passed to local systems for execution.

Heterogeneous Distributed Databases

- MultiDatabase Systems – Query processing
  - Steps one and four are similar to those in traditional data base systems (centralized/distributed) and hence will not be discussed further.

Heterogeneous Distributed Databases

- MultiDatabase Systems – Query processing
  - Unification decomposition: The issue is to determine how the integrated data can be constructed and which local data should be used for its construction.
  - The process could be complicated due to the equivalent data items at different local databases – A simple query that accesses a local data item may have to access an arbitrary number of data items at other sites because of direct or indirect equivalence relationships.

Heterogeneous Distributed Databases

- MultiDatabase Systems – Query processing
  - Similar to traditional distributed databases, two optimization techniques could be used:
    - Heuristic based optimization
    - Cost based optimization

Heterogeneous Distributed Databases

- MultiDatabase Systems – Query processing
  - Heuristic based optimization: Decompose the global query into the smallest possible sub-queries where each sub-query is executed at one local database (here multiple sub-queries may be sent to the same site).
    - Decomposition is relatively easier,
    - More chances to perform global optimization,
    - More work at the global optimizer,
    - More communication between global and local components.

Heterogeneous Distributed Databases

- MultiDatabase Systems – Query processing
  - Heuristic based optimization: Decompose the global query into the largest possible sub-queries where each sub-query can be executed at one local database.
    - Less work at global optimizer,
    - Fewer messages between global and local components,
    - More work at the local databases.

Heterogeneous Distributed Databases

- MultiDatabase Systems – Query processing
  - Cost based optimization: Given a query Q, its execution plans "execution space E_Q", and cost function C on E_Q, we want to find an execution plan $e_Q \in E_Q$ that has the minimum cost.
  - Local autonomy is the key factor that complicates the task beyond its complexity in traditional distributed databases for two reasons:

Heterogeneous Distributed Databases

- MultiDatabase Systems – Query processing
  - Cost based optimization
    - Global database management system may not have complete cost information about global sub-queries in order to perform the global optimization.
    - Global database management system interact with the local database management system at its application program interface level. As a result, it is unaware of internal data structure and functions of the local database management systems.

Heterogeneous Distributed Databases

- MultiDatabase Systems – Query processing
  - Cost based optimization: Three alternatives can be used to determine the cost of executing queries at the local nodes:
    - Treat local nodes as a black box, run some test queries on them, and from these determine the necessary cost information.
    - Use previous knowledge about local node and their external characteristics to determine the cost information,
    - Monitor the run-time behavior of the local node and dynamically collect the cost information.

Heterogeneous Distributed Databases

- MultiDatabase Systems – Query processing
  - Cost estimation of Global sub-queries
    - We can use a logical cost model to estimate cost of sub-queries:
      - Cost of a simple query (Q) on a relation is:
      - Cost (Q) = $C_0 + C_1 + C_2$
      - C_0 is the initialization cost
      - C_1 is the cost of finding qualifying tuples
      - C_2 is the cost of processing selected tuples.

Heterogeneous Distributed Databases

- MultiDatabase Systems – Query processing
  - C_0 is a function of local data base management system
  - C_1 is a function of the relation being accessed, and
  - C_2 is a function of the number of tuples being returned.
  - Cost (Q) = $c_0 + c_1 * |R| + c_2 * |R| * s$

Cardinality of the relation being accessed

Selectivity of the query

  - $|R|$ and *s* are unknown to the global database management system.

Heterogeneous Distributed Databases

- MultiDatabase Systems – Query processing
  - Cost coefficients c_0, c_1, and c_2 can be derived by a calibration process – run a set of suit of specially designed calibration queries, in isolation, on a specially designed calibration database (synthetic database) on the local site.
  - This strategy can be extended to the domain of more complicated queries and non relational database systems.

Heterogeneous Distributed Databases

- MultiDatabase Systems – Query processing
  - Cost based optimization: As an alternative to calibration queries and databases, one can use probing queries on component nodes to determine cost information. This approach can be extended to the domain of the so called "sample queries" where queries can be classified based on different criteria and sample queries for each class are issued to derive and measure cost information.