IBM Software Group

DB2 Grouper

A common component of DB2 product tools
DB2 and IMS Tools

Nathan Church
Jayashree Ramachandran
David Shough

DB2 Information Management Software

business on demand software

August, 2003

Problem Statement

- There are many relationships between the various DB2 objects, such as tables, in a business application. Some of these relationships can be discovered easily, while others cannot be easily found.
- As the complexity of data relationships increases, it becomes increasingly difficult to discover and keep track of this information.
- These relationships can be “dynamic” or “transient”, causing difficulties in managing business applications efficiently.
- Finding these relationships can be critical to treating portions of application data in a coherent manner.

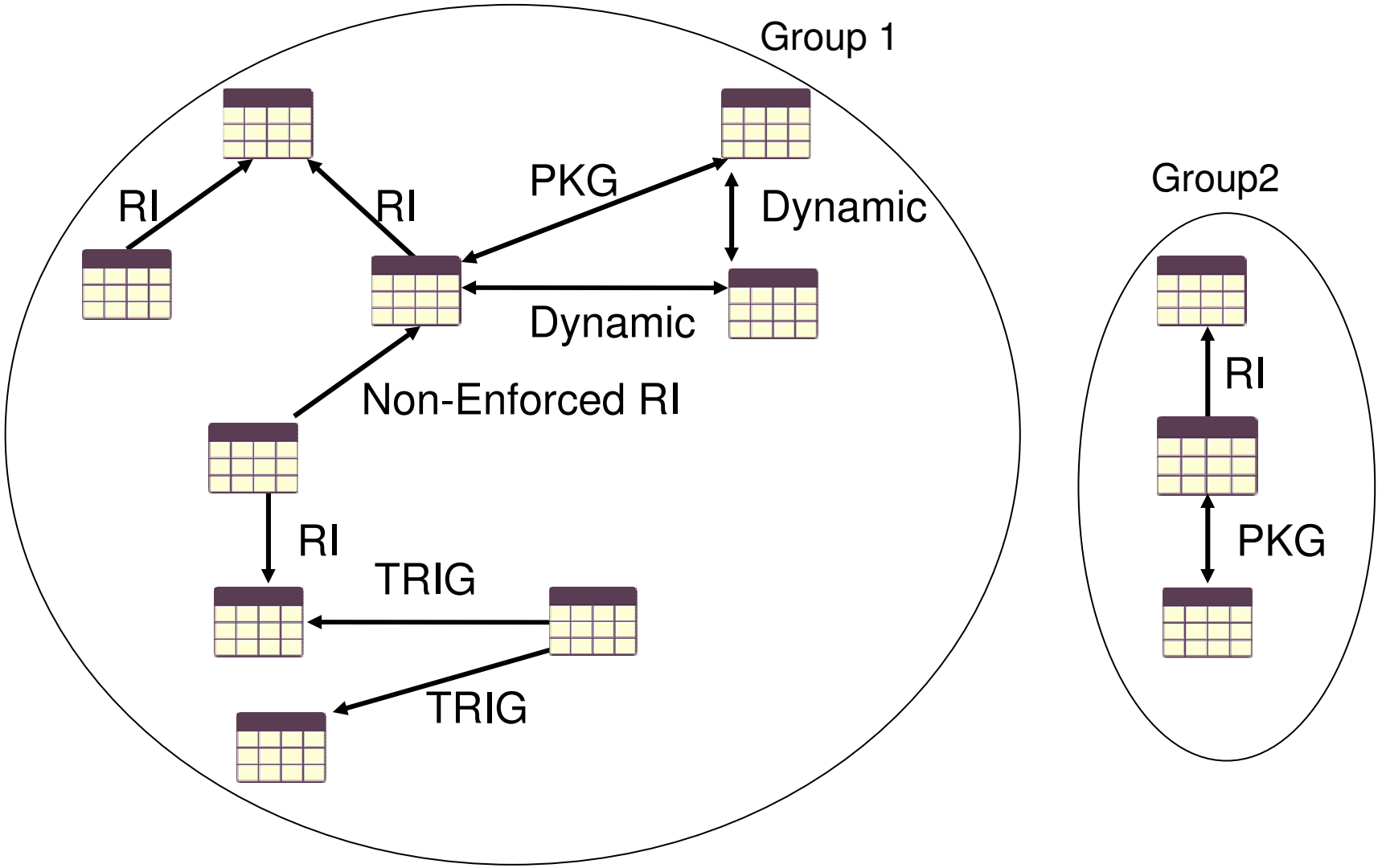
Problem Statement (Cont'd.)

- It can be difficult to understand all of these relationships, even though the information is useful to have.
- The information can be used for various kinds of administration activities, such as:
  - Planning effective backup and recovery strategies
  - Archiving and Relocation of data
  - Replicating data
  - Planning capacity and analyzing performance
- The objective of Grouper is to enable the location, augmentation, and management of this information as the basis for consistent data management activities.

The Solution - Grouper

- Grouper is a component for discovering, recording, and managing groups of related objects (tables) that comprise a business application.
- Grouper has the flexibility to augment and edit the information stored within it to support business needs.
- Grouper is autonomic at a predictive level; when users input a table to Grouper, it returns all of the tables that are related to it. Users can then choose either all or a subset of the tables returned for the operation being performed.
- Grouper is reusable:
  - Groups of related tables can be defined for multiple uses
  - The Grouper component can be integrated with various DB2 tools

Example of a data grouping problem

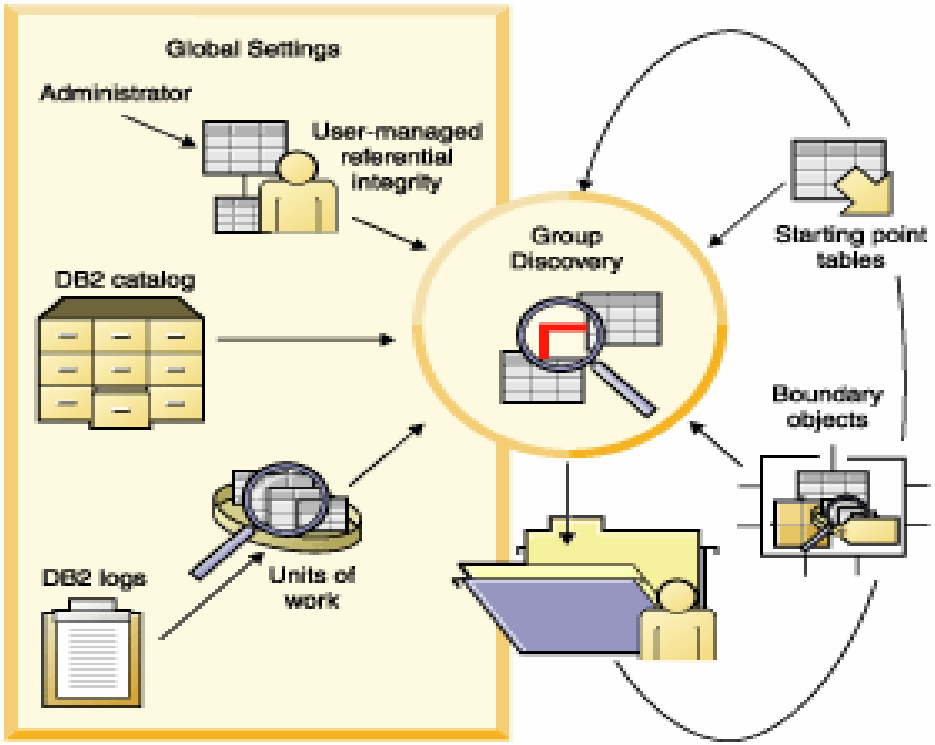


Types of Table Relationships

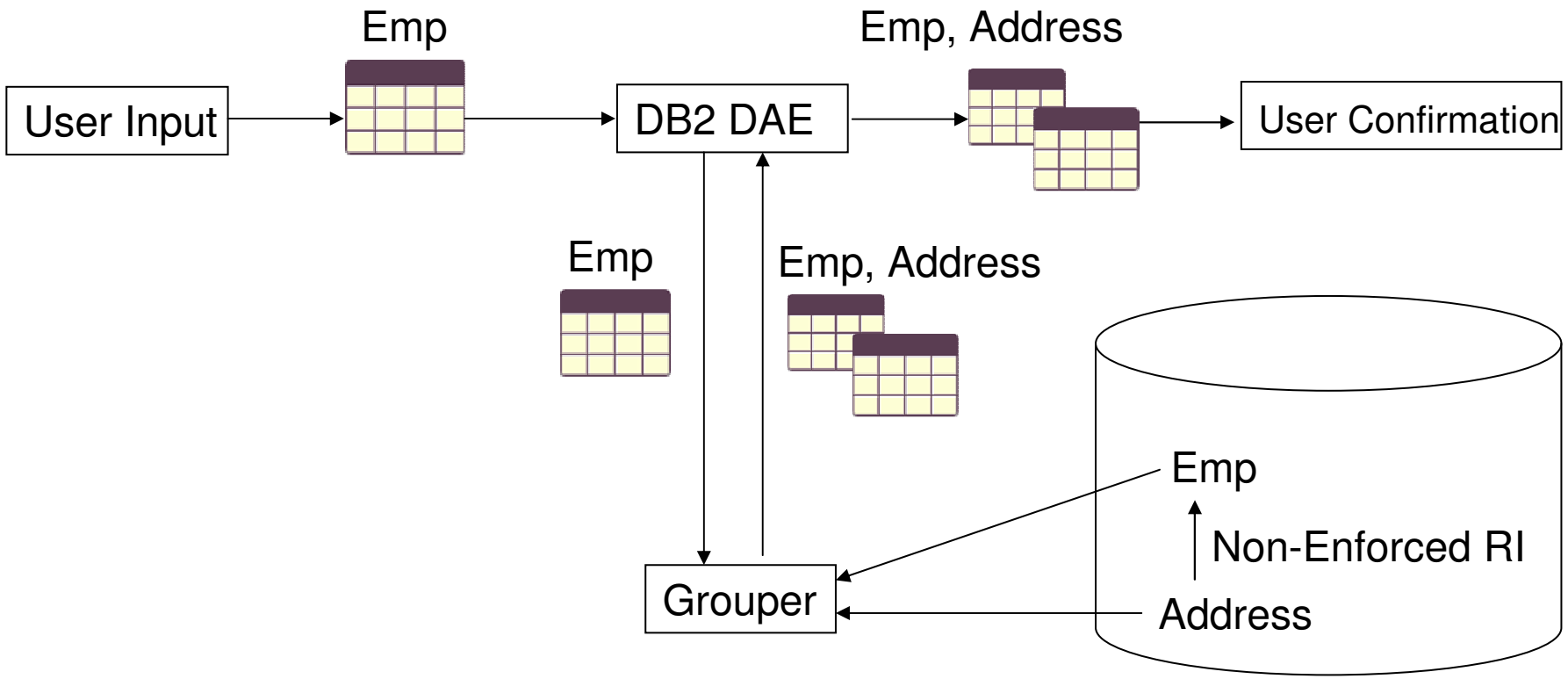
- Referential Constraints (RI)
  - DB2 System-enforced
  - Non-enforced
  - Can consider RI constraints only for grouping purposes (such as archiving)
- Trigger
- Packages (tables in an application that contains static SQL)
- LOBs (z/OS)
- Table Hierarchy and Reference Columns (Linux/UNIX/Windows)
- Tables within the same commit scope (dynamic SQL)
- Additional specified relationships based on knowledge about the system

Grouper Administration (GUI)

- Maintain sets of groupings
- Specify global relationships (non-enforced RI)
- Kick off dynamic (UOW) relationship discoveries
- Specify input values to the group discovery process
  - Group discovery scope (starting points)
  - Boundary objects
  - Additional relationships
- Launch the group discovery process
- Browse through the groupings
- Edit the groups



Example Use of Grouper by DB2 Data Archive Expert



Grouper usage examples

- **Data Archiving**:
  - As the volumes of data increases, performance deteriorates, and managing the data becomes difficult. To avoid this problem, inactive data is archived. However, there is still a need to archive referentially intact units.
  - Grouper component aids in determining the set of data to be archived simultaneously.
  - Specify a starting point table within DAE ISPF panels. DAE calls the Grouper component directly.
  - Using the Grouper client, define referential constraints not enforced by DB2.

Grouper usage examples (cont'd)

- **Data Recovery Expert:**
  - When recovering data to a specific point in time, related data needs to be recovered to that same point in time.
  - Grouper component aids in determining the set of related objects that need to be recovered simultaneously
- **General Data Administration/Understanding**
  - There is a need to understand relationships between various tables.
  - Much of the information is in the catalog, but there is a need to retrieve it and view it in a coherent manner.