KOREA UNIVERSITY

군산대학교 KUNSAN NATIONAL UNIVERSITY

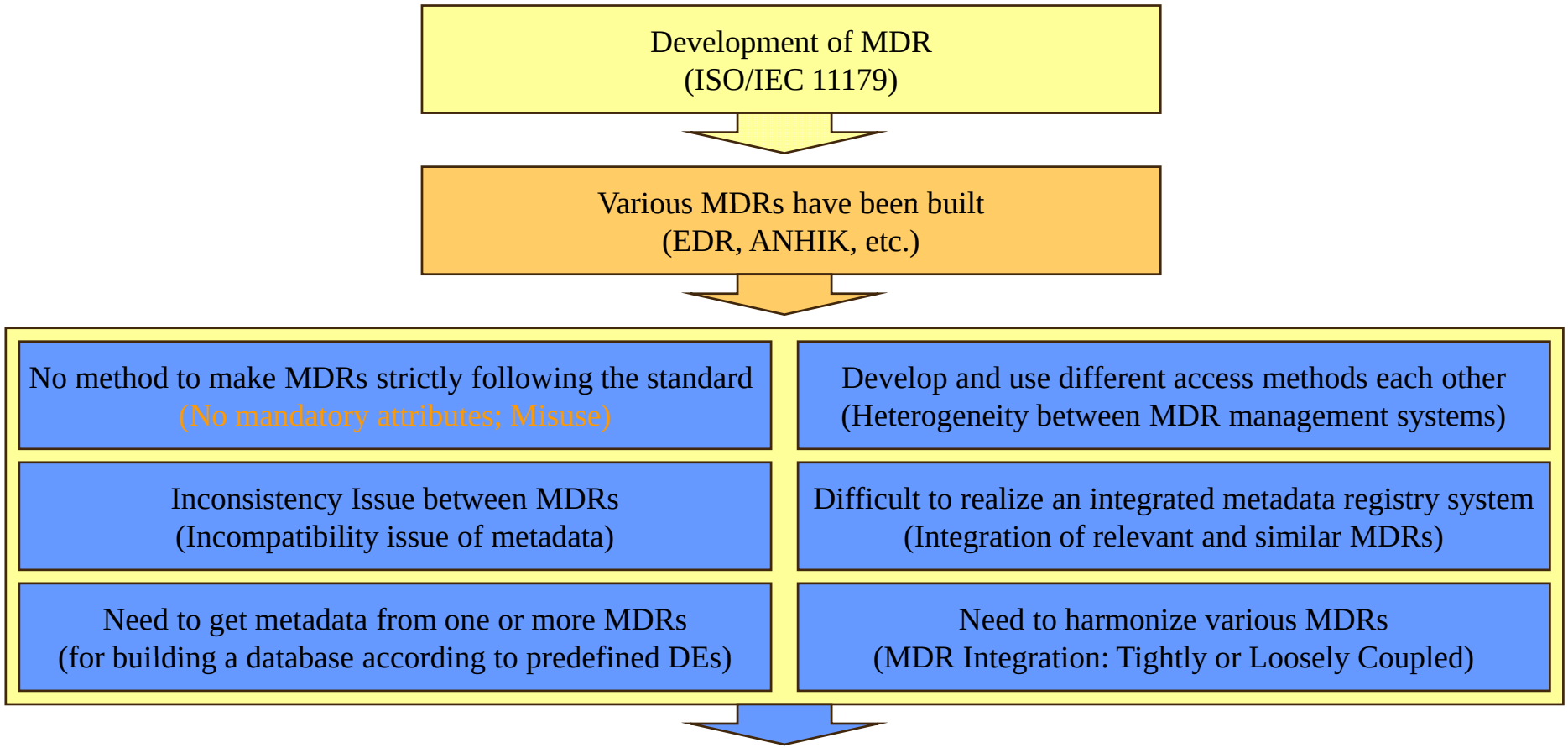
SQLMM MDR: Query Language for Sharing and Exchanging of Metadata between MDRs

Sept. 6, 2008
Wuhan, China

Dongwon Jeong (djeong@kunsan.ac.kr), Kunsan National University
Yixin Jing (yixin_jing@korea.ac.kr)
Doo-Kwon Baik (baikdk@korea.ac.kr), Korea University

Contents

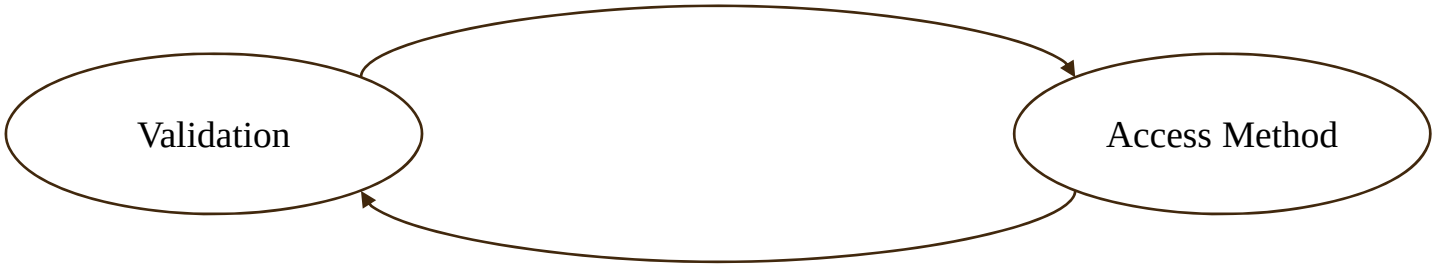
Background and Need



How to resolve the issues?

Background and Need (cnt.)

- **Key Points to Resolve the Issues**
  - ✓ How to make valid metadata registries (following the standard, ISO/IEC 11179)
    - How to guarantee the validation of metadata registries
  - ✓ How to consistently share and exchange metadata between MDRs
    - A consistent and standardized access method

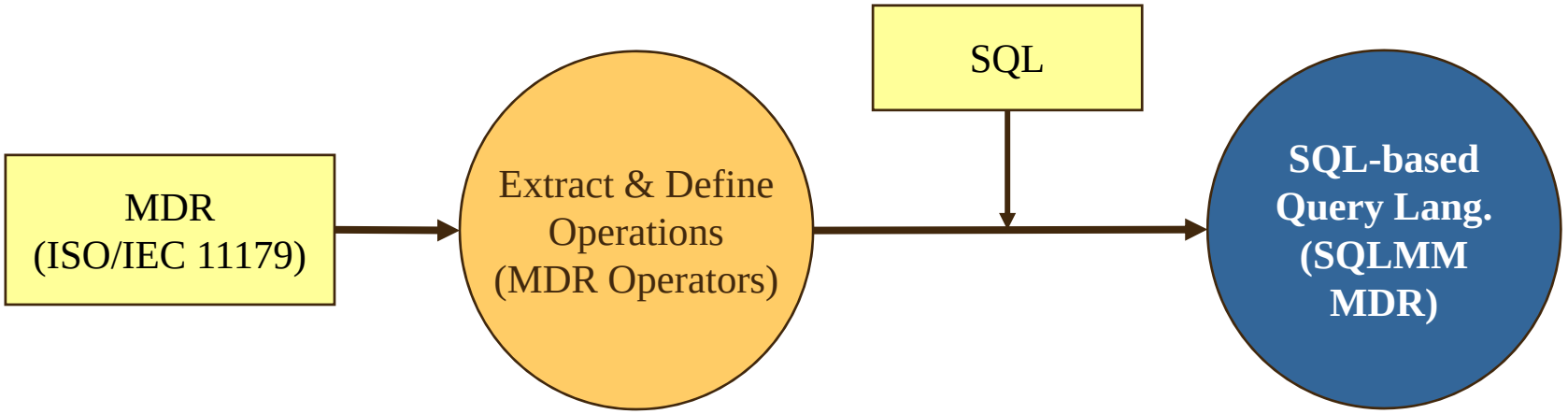


Possible Solutions

- **Interface Mapping between MDR Management Systems**
  - ✓ Required for implementation of one-to-one adaptor
  - ✓ Do not guarantee metadata registries to be valid
- **Standardized Binding API-based Approach**
  - ✓ Every system should implement the binding API
  - ✓ For example, ISO/IEC 20944
  - ✓ Do not guarantee metadata registries to be valid
- **Direct DB Access**
  - ✓ Users should know the schema structures of the corresponding MDRs
  - ✓ Do not guarantee metadata registries to be valid
- **Standardized Query Language-based Approach**
  - ✓ DBMSs should support this language (Require an extension)
  - ✓ No need to implement any functions such as binding interfaces
    - DBMS developers are responsible for the implementation
  - ✓ Guarantee metadata registries to be valid
  - ✓ Our Approach

Purpose of Our Proposal

- **Goal of the Proposal**
  - ✓ To develop a SQL-based query language for consistent access (share and exchange) to metadata registries
  - ✓ This query language is named SQLMM MDR
- **How to achieve the goal**
  - ✓ Analyze metadata registry access patterns,
  - ✓ Define MDR operators, and
  - ✓ Integrate the operators into SQL (i.e., extend SQL)

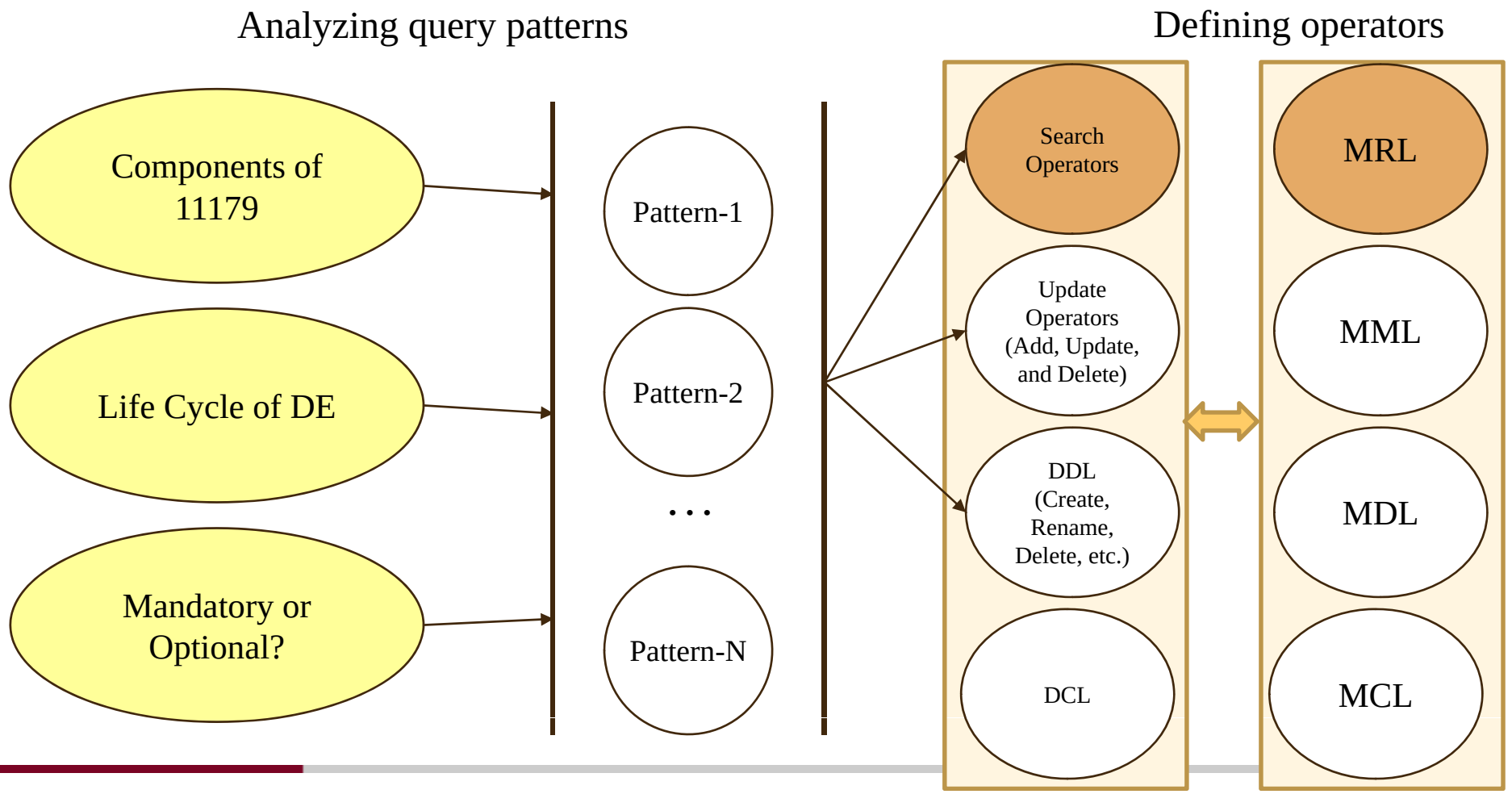


How to Define MDR Operators

- **Considerations to analyze operation patterns**
  - ✓ Search Target: Data element vs. the others (i.e., data element concept, conceptual domain, value domain, object class, etc.)
  - ✓ Mandatory vs. Optional attributes
  - ✓ Life cycle of data elements: Submitted, Registered, Qualified, Standard, etc.
  - ✓ and .....

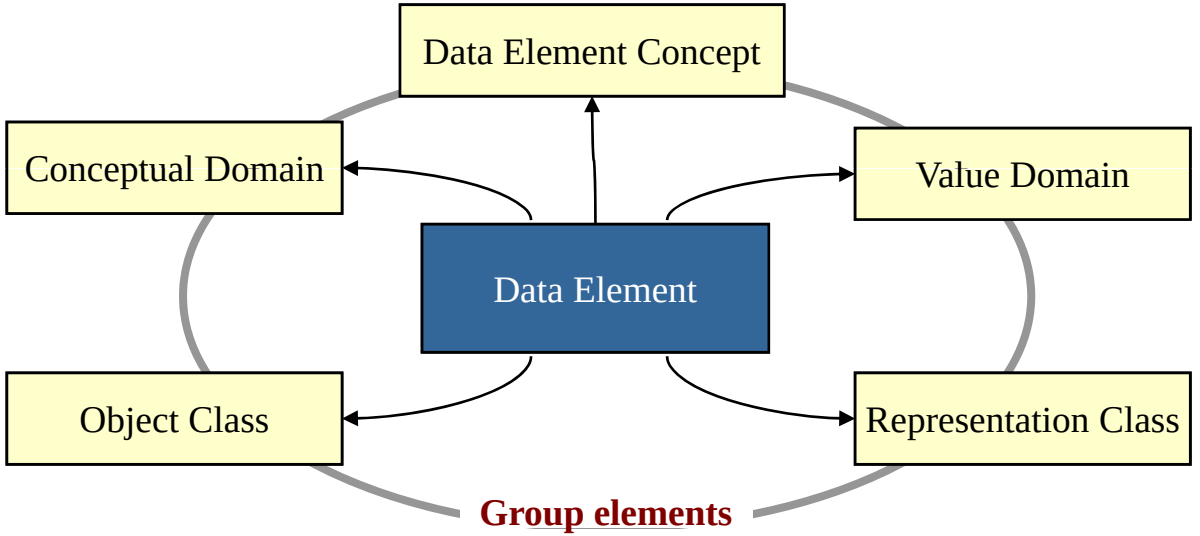
How to Define MDR Operators (cnt.)

- **A Methodology to achieve the goal: Definition Processes**



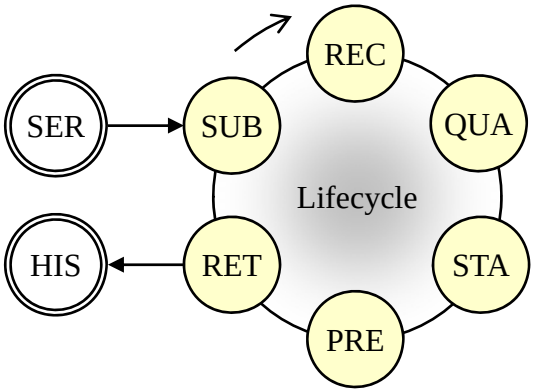
Query Patterns of Metadata Registry

- **Key components of metadata registry**
- **Data element vs. Group element**
  - ✓ Most components are used as a logical grouping unit of data elements.
  - ✓ Group elements (GEs)
    - – can logically cluster data elements
    - – For example, conceptual domain, data element concept, class object, and so on.
  - ✓ Search targets are classified into two classes: DE and GEs



Query Patterns of Metadata Registry (cont.)

- **Lifecycle of a data element**
  - ✓ Submitted, Recorded, Qualified, Standard, Preferred standard, and Retired



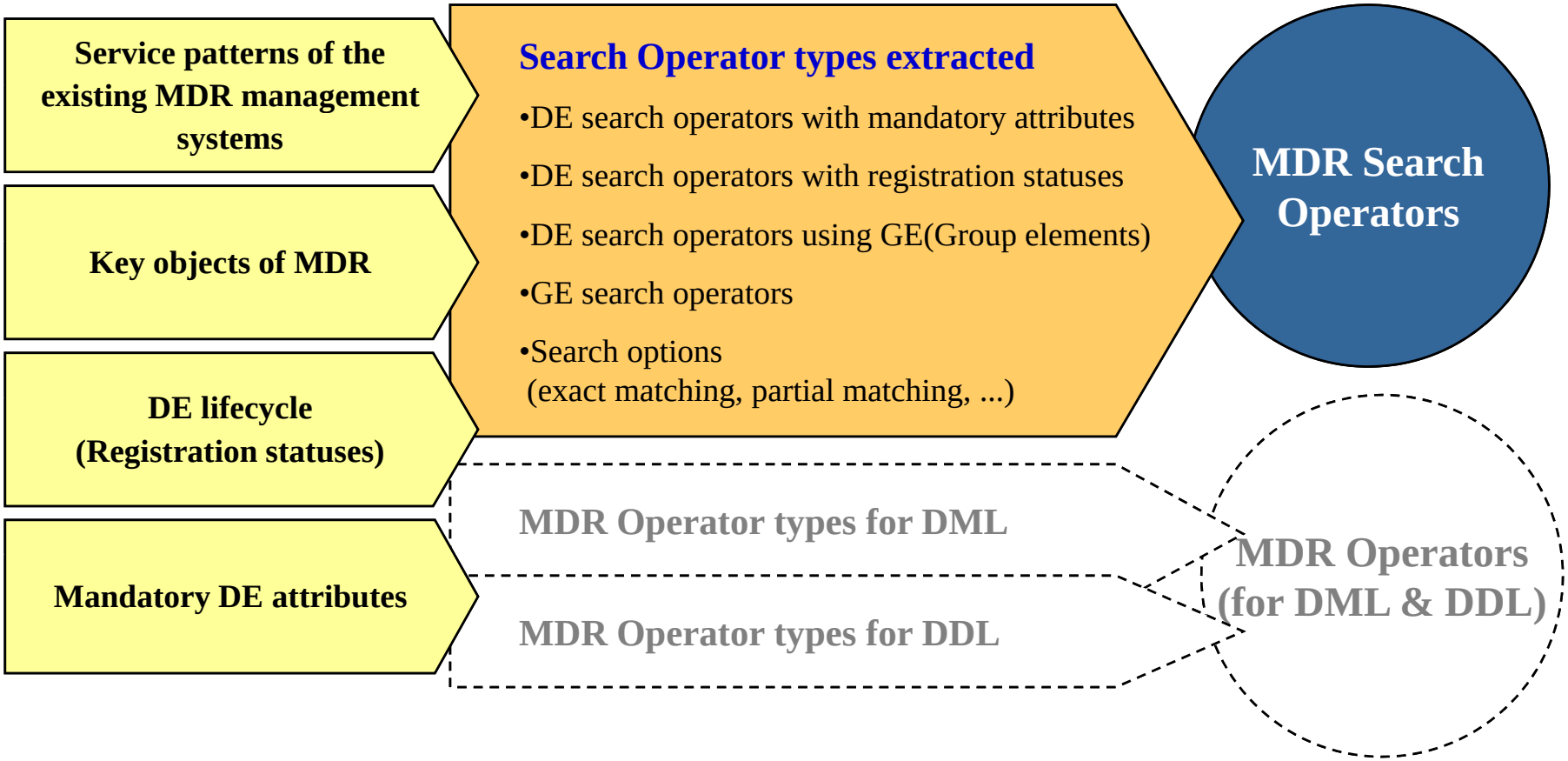
<Notation>

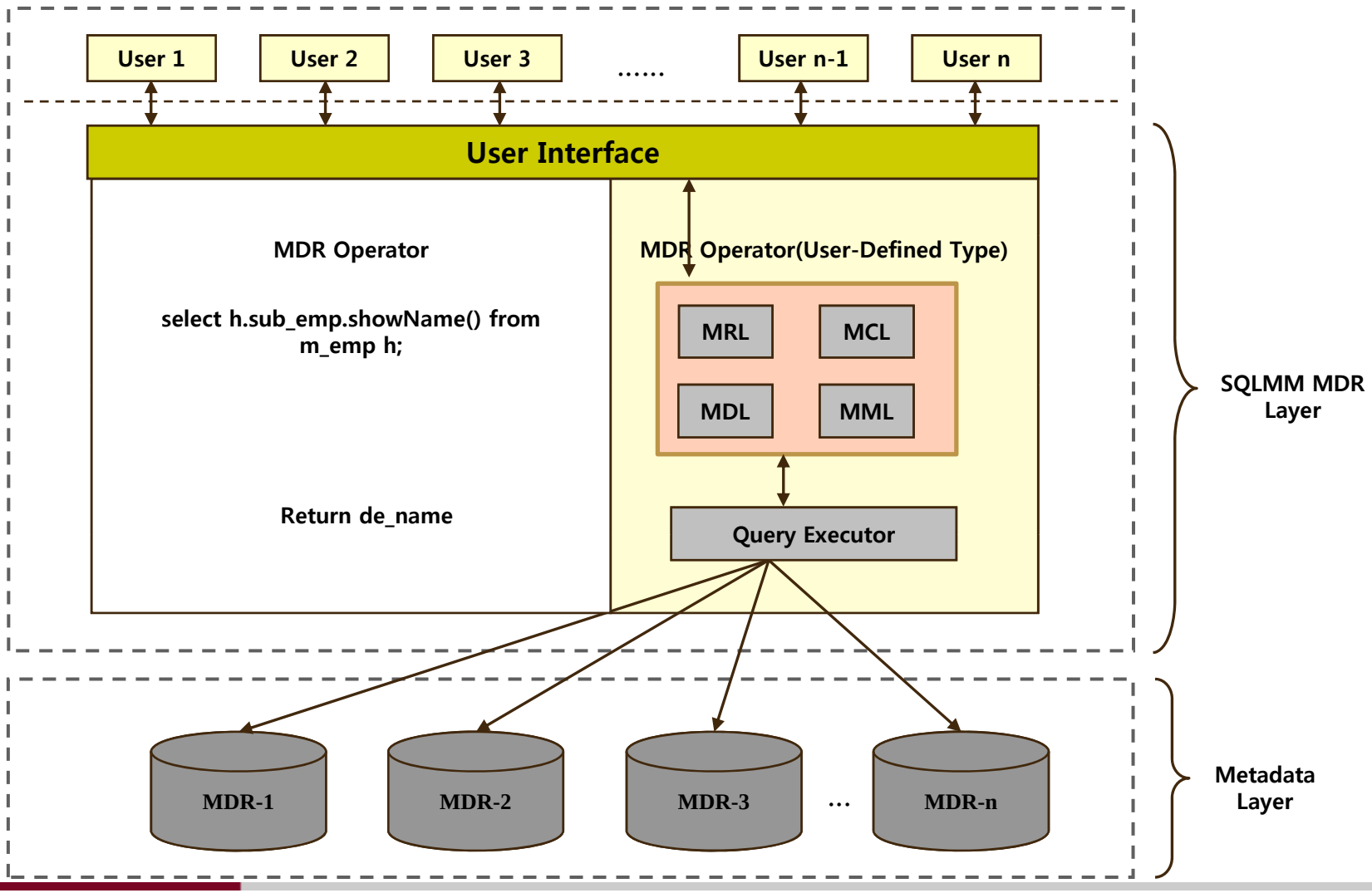
- SER: Submitter
- SUB: Submitted
- REC: Recorded
- QUA: Qualified
- STA: Standard
- PRE: Preferred standard
- RET: Retired
- HIS:Retired data element set

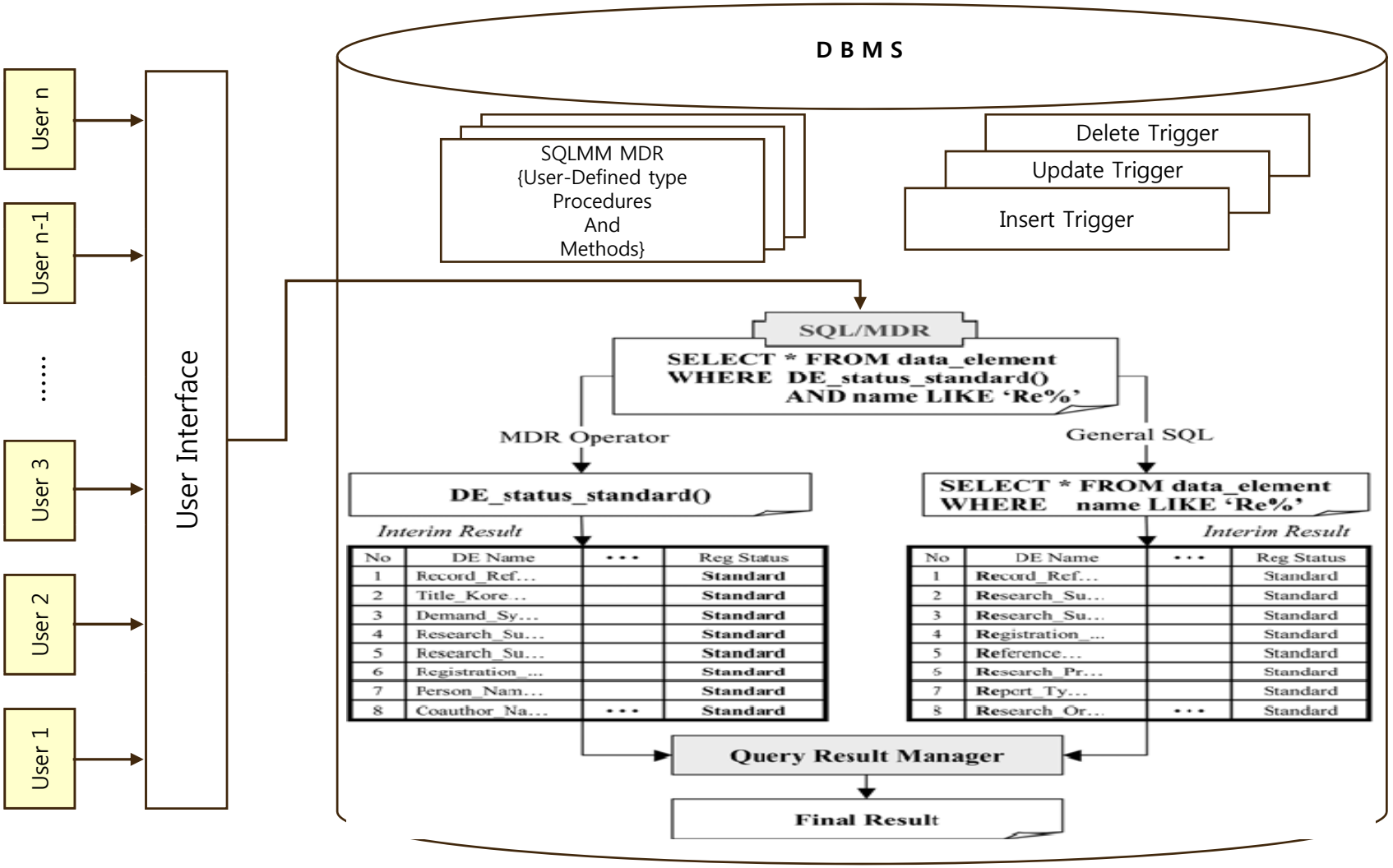
- **A Data element has mandatory attributes and optional attributes.**
  - ✓ Mandatory attributes
    - – Must be captured to achieve the goal of this paper
    - – For example, name, definition, and context of data elements should be described necessarily.
  - ✓ Optional attributes
    - – There is no need to be captured indispensably, but ???

Summary of MDR (Search) Query Patterns

- **MDR query operators for searching are defined by using the following factors:**




User 1
User 2
User 3
.....
User n-1
User n
User Interface
MDR Operator
select h.sub_emp.showName() from m_emp h;
Return de_name
MDR Operator(User-Defined Type)
MRL
MCL
MDL
MML
Query Executor
SQLMM MDR Layer
MDR-1
MDR-2
MDR-3
...
MDR-n
Metadata Layer


User n
User n-1
.....
User 3
User 2
User 1
User Interface
D B M S
SQLMM MDR
{User-Defined type
Procedures
And
Methods}
Delete Trigger
Update Trigger
Insert Trigger
SQL/MDR
SELECT * FROM data_element
WHERE DE_status_standard()
AND name LIKE 'Re%'
MDR Operator
General SQL
DE_status_standard()
SELECT * FROM data_element
WHERE name LIKE 'Re%'
Interim Result
Interim Result
No DE Name ... Reg Status
1 Record_Ref... Standard
2 Title_Kore... Standard
3 Demand_Sy... Standard
4 Research_Su... Standard
5 Research_Su... Standard
6 Registration_... Standard
7 Person_Nam... Standard
8 Coauthor_Na... ... Standard
No DE Name ... Reg Status
1 Record_Ref... Standard
2 Research_Su... Standard
3 Research_Su... Standard
4 Registration_... Standard
5 Reference... Standard
6 Research_Pr... Standard
7 Report_Ty... Standard
8 Research_Or... ... Standard
Query Result Manager
Final Result

An Example of Search Operators

❖ MDR Query Operators for Retrieval: Search Operators

Search Operators	Description	Notations
DE_name(KW, OPT) DE_definition((KW, OPT) DE_context(KW, OPT) DE_identifier(KW, OPT) DE_reg_organization(KW, OPT), . . .	Query operators to retrieve DEs with mandatory attributes	KW : Keyword given by users OPT : Search options (Partial, exact, and starting with matching) RA : Registration attribute DA : Mandatory attributes of data elements such as name, definition, context, and so on.
DE_status(RA, DA, KW, OPT) DE_status(RA) DE_status_submitted(DA, KW, OPT) DE_status_retired(DA, KW, OPT), . . .	Query operators to retrieve DEs with registration statuses	
DE_object_class(KW, OPT) DE_conceptual_domain(KW, OPT) DE_concept(KW, OPT)	Query operators to retrieve DEs using the group elements	
object_class(KW, PT) conceptual_domain(KW, OPT) element_concept(KW, OPT), . . .	Query operators to retrieve GE(Group Elements)	

Examples of SQLMM MDR

- ***Query 1***. Retrieve all data elements where registration status is ‘RECORDED’.

SQLMM MDR> SELECT DE_status.RECORDED()
FROM data_element .....(1-a)

- ***Query 2***. List all data elements where registration status is 'RECORDED' and name is ‘KOREA’.

SQLMM MDR> SELECT * FROM data_element
WHERE DE_status.RECORDED()
AND name=‘KOREA’ …….…….....…….(2-a)

Advantages

- **Advantages**
  - ✓ Ease of use
  - ✓ Familiarity to users
  - ✓ Consistent and Standardized access method
    - – Independent query description on physical structures (table structures)
    - – Low modeling cost (decreases query description time)
    - – Simplicity of query statement for accessing distributed registries
    - – Low complexity for distributed query composition

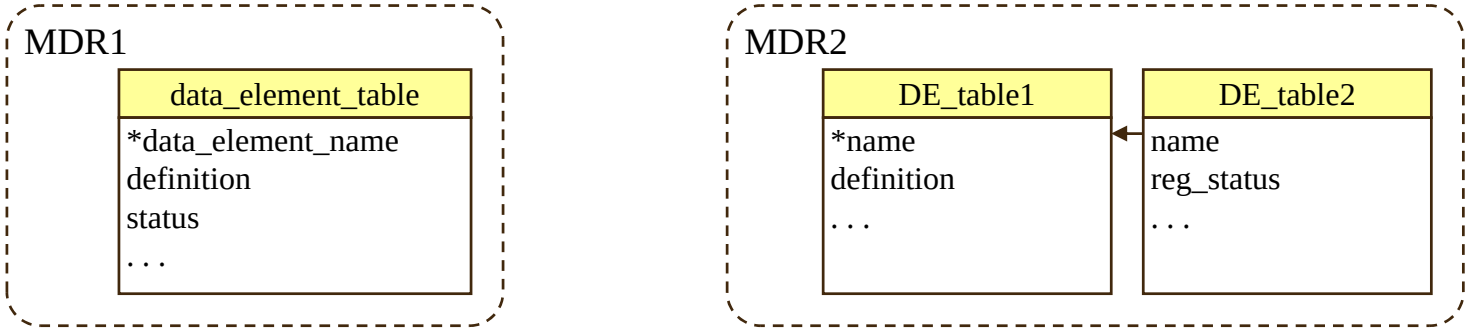
- **In its application aspect, SQLMM MDR**
  - ✓ can be used Unified Access Method to Various Registries
  - ✓ provides Unified query and Unified query result form
  - ✓ can be used for developing distributed registries management system
  - ✓ can be used as a communication protocol (method) for developing semi-automatic tools: Registries Mapping and Integrating tools
  - ✓ Data Integration (among databases following their own registry)

Advantages: with a Scenario

- **The below query shows advantages such as efficiency and simplicity of SQLMM MDR.**
- **Let's assume the following situation to show contributions of SQLMM MDR.**
  - ✓ Assumption (Situation): There are two registries with their own physical structure
    - There exist two metadata registries.
    - Each metadata registry has different MDR structure each other.
    - The first metadata registry, MDR1 is designed as follows:
      (1) data_element_table includes all of the data elements,
      (2) data_element_name is a field name of data_element_table and means name of data elements, and
      (3) Status is a field name of data_element_table and means registration status of data elements
    - The second metadata registry, MDR2 consists of two tables as follows:
      (1) table1 and table2 include all of the data elements together. Most attributes are included in table1 and some attributes including registration status are in table2,
      (2) table1 and table2 use name as their join key, and
      (3) Registration status is involved in reg_status, a field of table2.

Advantages: with a Scenario (cont.)

- **Table Structures of Two MDRs, MDR1 and MDR2**



- **Query 3. (Access to two metadata registries) Retrieve all data elements, where registration status is 'RECORDED' from metadata registries, MDR1 and MDR2.**

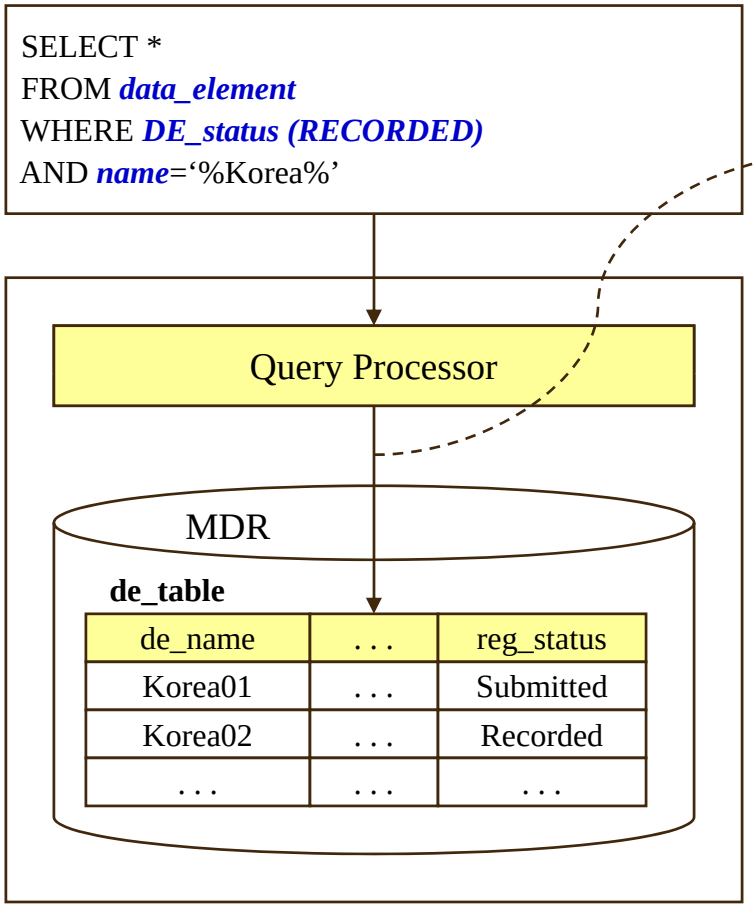
```
SQLMM MDR>  SELECT   DE_status (RECORDED)
                FROM     data_element .....(3-a)
```

In case of SQLMM MDR based approach

```
SQL>  SELECT   data_element_name
          FROM     data_element_table
          WHERE   status = 'RECORDED' .....(3-b)
SQL>  SELECT   table1.name
          FROM     table1, table2
          WHERE   table1.name=table2.name
          AND  table2.reg_status = 'RECORDED' .....(3-c)
```

In case of use of SQL
//query rewriting is required, i.e., requires several queries
//high design time, complicated processing, etc.

Restriction (Disadvantage? Barrier?)



Query processor should know the followings:
(1) data_element ⇔ de_table
(2) name ⇔ de_name
(3) reg_status holds registration information

In a word, a **preprocessing** (mapping) is required for realization of the proposal.

1st approach to define SQLMM MDR
: Loose Coupling

vs.

Another approach to define SQLMM MDR
: Using a Predefined MDR Schema (Tightly Coupled)

If a relational schema is given, this problem can be easily solved ?!

Conclusion

- **SQL MM MDR**
  - ✓ SQL-based query language for consistent access to metadata registries (ISO/IEC 11179)
  - ✓ Advantages
    - – Unified access method: One described, All returned (i.e., same query, same formatted results)
    - – Strengthen standardization of registries