Improving the Correctness of Some Database Research using ORA-Semantics

Tok Wang Ling,

Zhong Zeng, Mong Li Lee, Thuy Ngoc Le

National University of Singapore

ER 2016, Gifu, Japan

Outline

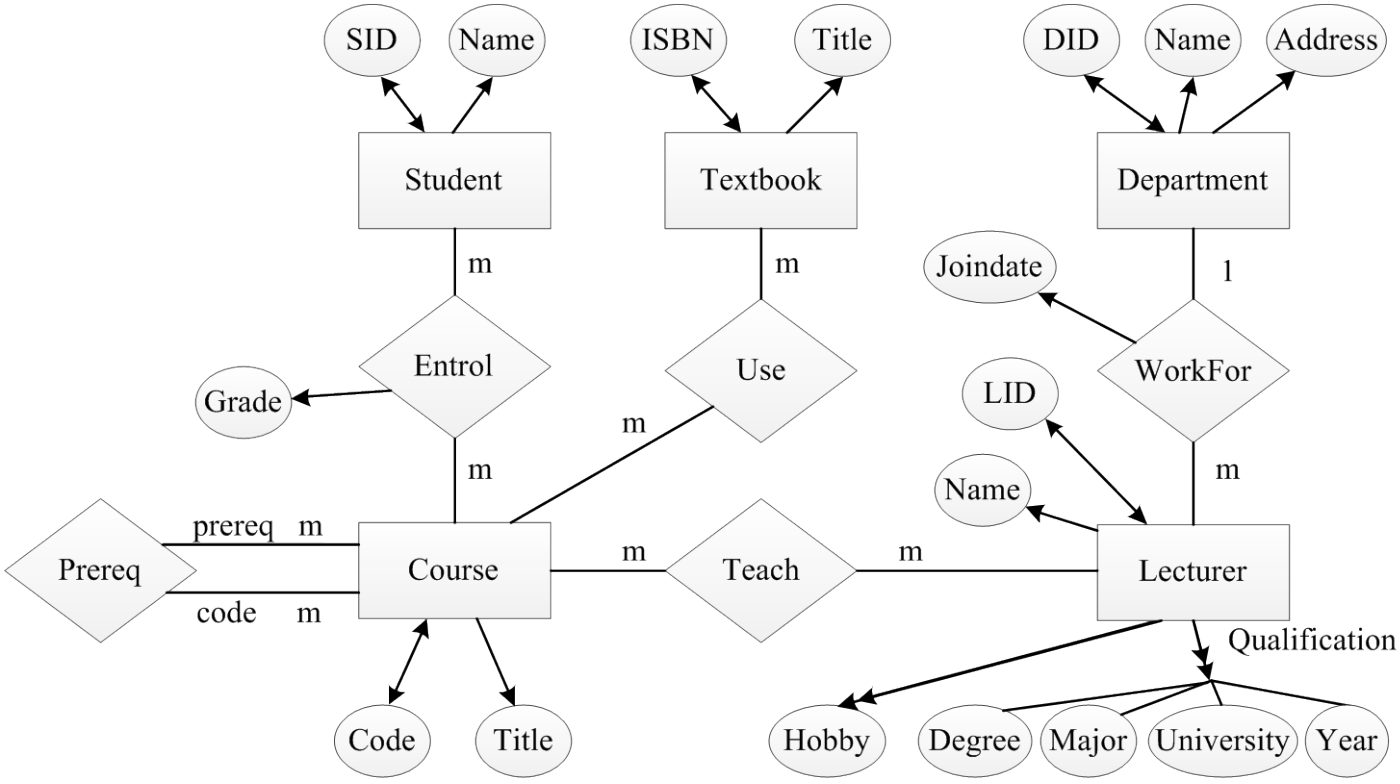
- Introduction
  - ➢ Object-Relationship-Attribute (ORA) Semantics **in ER Model**
- Limitations of Relational Model
- Limitations of XML Data Model
- ORA-semantics in Data and Schema Integration
- ORA-semantics in RDB Keyword Search
- ORA-semantics in XML Keyword Search
- Conclusion

Outline

- Introduction
- Limitations of Relational Model
- Limitations of XML Data Model
- ORA-semantics in Data and Schema Integration
- ORA-semantics in RDB Keyword Search
- ORA-semantics in XML Keyword Search
- Conclusion

ER Model and ORA-Semantics

- We call the concepts of object class, relationship type, and their attributes in the ER model as Object-Relationship-Attribute (ORA) semantics



(ER diagram for a university database)

ER Model and ORA-Semantics (cont.)

- A database designer must know the ORA-semantics in order to design a good schema
- A programmer must know the ORA-semantics in order to write SQL or XQuery programs correctly
- A user needs to know ORA-semantics in order to ask sensible queries

❖ The relational model and XML data model do not capture ORA-semantics, which leads to problems in RDB/XML database design, data/schema integration, and RDB/XML keyword query processing

Outline

- Introduction
- Limitations of Relational Model
- Limitations of XML Data Model
- ORA-semantics in Data and Schema Integration
- ORA-semantics in RDB Keyword Search
- ORA-semantics in XML Keyword Search
- Conclusion

Limitations of Relational Model

Outline

- Relational Model (RM) does ***not*** capture **ORA-semantics**, which leads to many problems in database schema design, data/schema integration, keyword query processing, etc.
- Relation in RM is **not** the same as relationship. Relation name has ***no*** real meaning.
- Functional Dependency (FD) and Multi-valued Dependency (MVD) are integrity constraints which are mainly imposed by organizations or database designers. They have ***no*** semantics.
- Universal Relation Assumption (URA) in Relational Model ***cannot*** handle complex relationship types such as recursive relationship type, ISA relationship, multiple relationship types between / among object classes.
- RM ***cannot*** differentiate between object attribute and relationship attribute.

Limitations of Relational Model

Outline (cont.)

- Normalization only uses FDs and MVDs to reduce data redundancy and obtain normal form relations but normal form relations ***cannot*** capture **ORA-semantics** in a RDB.
- Key in relation is ***not*** the same as OID of object class.
- Database schema design approaches based on URA such as **decomposition method** and **synthesizing** method ***cannot*** handle complex relationship types directly and so they have many limitations and problems.
- For data/schema integration, we ***need*** to have the concepts of **global** FD/MVD, **global** OID, **relationship identification** besides object identification, etc., as multiple databases may be from different organizations and locations, etc.
- More …

FDs and MVDs

- 2 classes of integrity constraints in relational model:
  - Functional Dependency (FD)
  - Multivalued Dependency (MVD)
- Most of FDs are imposed by database designers or organizations.
  - **E.g.** E# and SSN are unique with respect to the particular database.
    - Both E# and SSN can be used to identify an employee. But why do we need both?
    - E# is **local** to a company vs SSN is **global** in US
    - Both E# and SSN are artificially introduced by some designers
  - E.g. Each employee has one name.
    - Why? Some employee may have more than one name.
    - It is an imposed constraint for efficiency processing purpose.

FDs and MVDs (cont.)

- **Existence of MVDs** are mainly because of wrong designs:
  a) Singled valued attributes and multivalued attributes are wrongly put in one relation

$$Lecturer(LID, Name, Hobby)$$

  - Single valued attribute: LID, $Name$
  - Multivalued attribute: $Hobby$, a lecturer may have several hobbies
  - Key: {LID, $Hobby$}
  - MVD: $LID \twoheadrightarrow Hobby$

FDs and MVDs (cont.)

- **Existence of MVDs** are mainly because of wrong designs: (cont.)

  b) 2 independent multivalued attributes are wrongly put in one relation

$$Lecturer\_hobby\_qual\ (LID, Hobby, Degree, Major, Univ, Year)$$

  - Multivalued attributes:
    - $Hobby$ & $\{Degree, Major, Univ, Year\}$ i. e. $Qualification$
    - ❖ A lecturer may have several hobbies and several qualifications
  - Key: all attributes
  - MVDs: $LID \twoheadrightarrow Hobby$

    $LID \twoheadrightarrow \{Degree, Major, Univ, Year\}$

FDs and MVDs (cont.)

- **Existence of MVDs** are mainly because of wrong designs:

  c) 2 independent relationship types are wrongly put in one relation

$$CTL(Code, ISBN, LID)$$

  - Relationship types:
    - Many-to-many relationship between course and textbook
    - Many-to-many relationship between course and lecturer
  - Key: all attributes
  - MVDs: $Code \twoheadrightarrow ISBN$

    $Code \twoheadrightarrow LID$

FDs and MVDs (cont.)

- **MVDs are problematic** because they are relation sensitive [1]

  In previous slide:

  $$CTL\,(Code, ISBN, LID)$$

  with $\{Code \twoheadrightarrow ISBN,\quad Code \twoheadrightarrow LID\}$

  ❑ Suppose we add one more attribute percentage:

  $$CTL'(Code, ISBN, LID, percentage)$$

  A tuple (c, i, l, p) means lecturer l teaches course c and p percentage of his material is from textbook i

  FD: $\{Code, ISBN, LID\} \rightarrow percentage$

  However, $Code \twoheadrightarrow ISBN$ & $Code \twoheadrightarrow LID$ do not hold in CTL'

  ❑ This shows that MVDs are relation sensitive. They are difficult to discover before relations are known.

FDs and MVDs (cont.)

- FDs and MVDs **cannot be automatically discovered**

  $Student(SID, Name)$

  - Even if student names are unique in a database instance

    $Name \rightarrow SID$

    is incorrect in general

- FDs and MVDs do not capture ORA-semantics

  $Lecturer(LID, Name, DID, \mathbf{Joindate})$

  - $LID \rightarrow \boldsymbol{Joindate}$

    does not indicate whether $\boldsymbol{Joindate}$ is an attribute of objects lecturers or an attribute of relationship between lectures and departments [2]

FDs and MVDs (cont.)

Note that During **normalization** (i.e. database schema design)

- We must maintain / enforce the given set of FDs, i.e., the **closure of the set of FDs** remain unchanged.
- However, we want to remove all **MVDs**.

Relational Database Design Methods

- **3 common methods** for relational database schema design:

  1) Decomposition method
  2) Synthesis method [3]
  3) The ER approach

- Objectives:

  a) Remove redundancy
  b) Remove transitive dependencies but keep the closure of given set of FDs unchanged
  c) Remove MVD completely

  - E.g. $CTL(Code, ISBN, LID) \Longrightarrow CT(Code, ISBN)$ & $CL(Code, LID)$

Relational Database Design Methods (cont.)

- 3 common methods for relational database schema design:

1) Decomposition method

- Based on the assumption that a database can be represented by a universal relation (the **Universal Relation Assumption - URA**) which contains a set of attributes.
- This relation is then **decomposed** into smaller relations in order to remove redundant data using a given set of FDs and MVDs

Relational Database Design Methods (cont.)

1) Decomposition method (cont.)

- Disadvantages:

a) Almost impossible to obtain MVDs before decomposition as MVDs are relation sensitive

b) The process is non-deterministic, depending on the order of FDs and MVDs for decomposition.

c) Need to find / derive the MVDs in the decomposed relations.

d) Some schemas obtained may be very bad as some FDs may be lost, i.e. may not keep the closure of given set of FDs.

e) It cannot handle complex relationship types: recursive relationship, ISA relationship, multiple relationship types among object classes, multivalued attributes, many-to-many relationship type without attribute in ERD (because of the URA).

f) Meaningful relation names cannot be automatically generated without the knowledge of ORA-semantics from the database designer.

Relational Database Design Methods (cont.)

2) Synthesis method [3]

- Also based on URA and assume a database is represented by a set of attributes with a set of FDs
- Synthesize a set of 3NF relations and keep the closure of the given set of FDs remain unchanged

❖ **Disadvantages**:

a) The process is non-deterministic, depending on the non-redundant covering of FDs found to generate 3NF relations

b) Cannot handle complex relationship types, multivalued attributes, many-to-many relationship type without attribute in ER

c) Does not guarantee reconstructibility

d) Meaningful relation names cannot be automatically generated except manually changed by the database designer with ORA-semantics.

e) Global redundant attributes [4] may still exist

f) Does not consider MVDs

3) The ER approach

a) Based on relaxed URA

b) Construct an ERD including recursive relationship, ISA relationship, more than one relationship type among object classes

c) Normalize ERD to a normal form ERD [5]

d) Translate the normal form ERD to normal form relations with additional constraints (ISA, role name, inclusion dependency).

e) Meaningful relation names can be automatically generated based the object class names, relationship types names, etc. in the ERD and capture the ORA-semantics.

f) No need to consider MVDs

- The ER approach captures the ORA-semantics and avoids the problems of the decomposition method and synthesis method

Outline

- Introduction
- Limitations of Relational Model
- Limitations of XML Data Model
- ORA-semantics in Data and Schema Integration
- ORA-semantics in RDB Keyword Search
- ORA-semantics in XML Keyword Search
- Conclusion

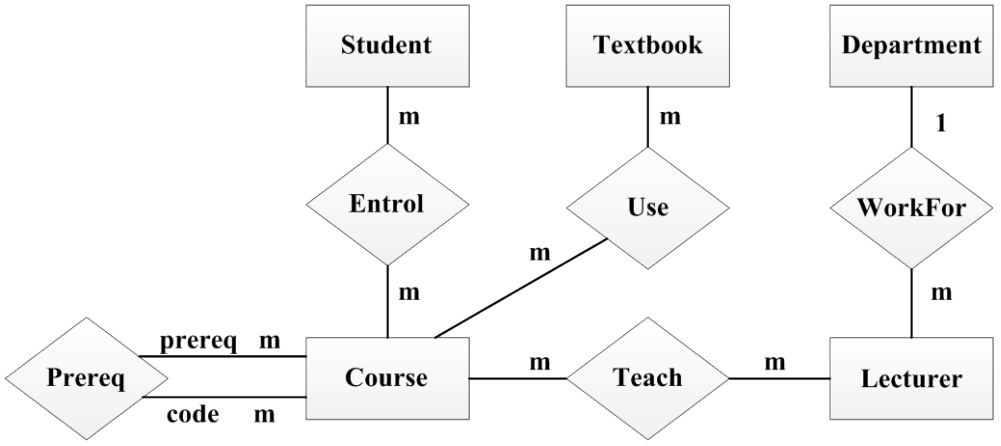
Limitations of XML Data Model

XML DTD and XML Schema

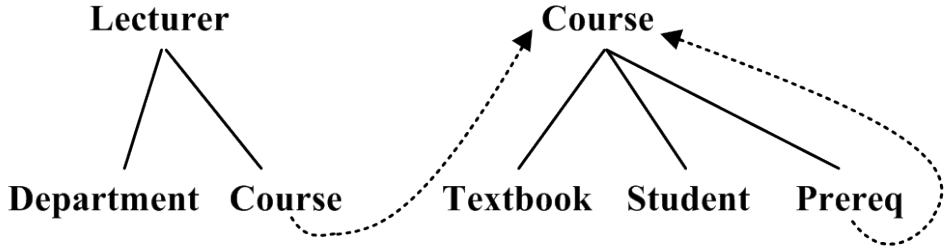
- The constraints on the structure and content of an XML can be described by DTD or XML Schema

```
<!DOCTYPE universitydb [
    <!ELEMENT db (Lecturer*, Course*)>
    <!ELEMENT Lecturer (Hobby*, Qualification*,
                        Department)>
    <!ATTLIST Lecturer LID   ID   #REQUIRED
                       Name   cdata
                       Course   IDREFS   #IMPLIED>
    ....
    <!ELEMENT Course (Textbook*, Student*)>
    <!ATTLIST   Course Code   ID   #REQUIRED
                       Title   cdata
                       Prereq   IDREFS   #IMPLIED>
    <!ELEMENT Student (Name, Grade)>
    <!ATTLIST    Student SID   cdata   #REQUIRED>
    ....
]>
```

(An XML DTD for the university database)



(An ER diagram)



(A possible XML schema tree)

XML DTD and XML Schema (cont,)

- DTD/XML Schema specifies the structural representation of XML with simple constraints, and has no concept of ORA-semantics

1) ID in DTD is not the same as object identifier
2) IDREF is not the same as foreign key to key reference in RDB. IDREF has no type.
   E.g. Prereq IDREFS #IMPLIED
3) Multivalued attribute cannot be defined as an attribute but as sub-element
4) Relationship type is implicit via *parent-child* relationship

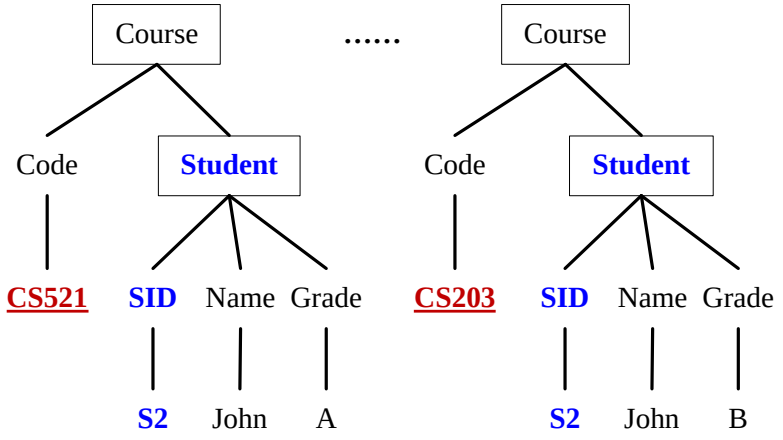
XML DTD and XML Schema (cont,)

- DTD/XML Schema specifies the structural representation of XML with simple constraints, and has no concept of ORA-semantics (cont.)

1) ID in DTD is object identifier (OID). However, OID may not be able to define as ID

```
<!ELEMENT Course (Textbook*, Student*)>
<!ATTLIST   Course Code   ID   #REQUIRED
                  Title   cdata
                  Prereq  IDREFS  #IMPLIED>
<!ELEMENT Student (Name, Grade)>
<!ATTLIST   Student  SID  cdata  #REQUIRED>
```

(Part of XML DTD for the university database)



(example XML fragment)

❖ We cannot define SID as ID of $Student$ elements because the same student element may occur multiple times as he may enroll more than one course

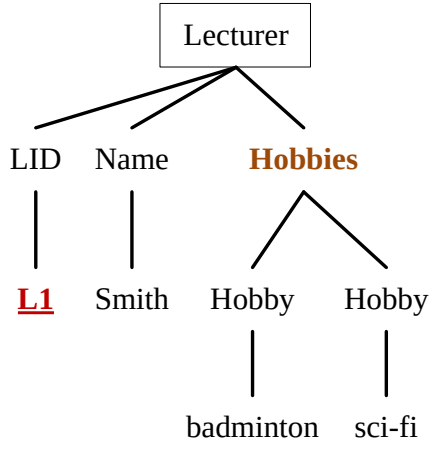
XML DTD and XML Schema (cont,)

- DTD/XML Schema specifies the structural representation of XML with simple constraints, and has no concept of ORA-semantics (cont.)

2) Multivalued attribute cannot be defined as an attribute

```
<!ELEMENT db (Lecturer*, Course*)>
<!ELEMENT Lecturer (Hobbies, Department)>
<!ATTLIST   Lecturer LID   ID   #REQUIRED
                     Name   cdata
                     Course   IDREFS   #IMPLIED>
<!ELEMENT Hobbies (Hobby*)>
<!ELEMENT Hobby  (#PCDATA) >
```

(Part of XML DTD for the university database)



(example XML fragment)

- We cannot define $Hobby$ as attributes of $Lecturer$ elements.
- They have to be declared as sub-elements of Lecturer.

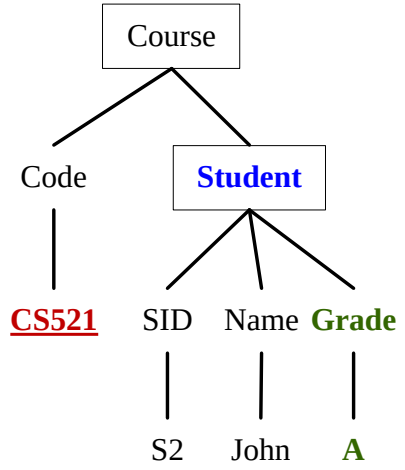
XML DTD and XML Schema (cont.)

- DTD/XML Schema specifies the structural representation of XML with simple constraints, and has no concept of ORA-semantics (cont.)

3) Relationship type is implicit via parent-child relationship

```
<!ELEMENT Course (Textbook*, Student*)>
<!ATTLIST   Course Code   ID   #REQUIRED
                   Title  cdata
                   Prereq IDREFS #IMPLIED>
<!ELEMENT Student (Name, Grade)>
<!ATTLIST   Student SID  cdata  #REQUIRED>
```

(Part of XML DTD for the university database)

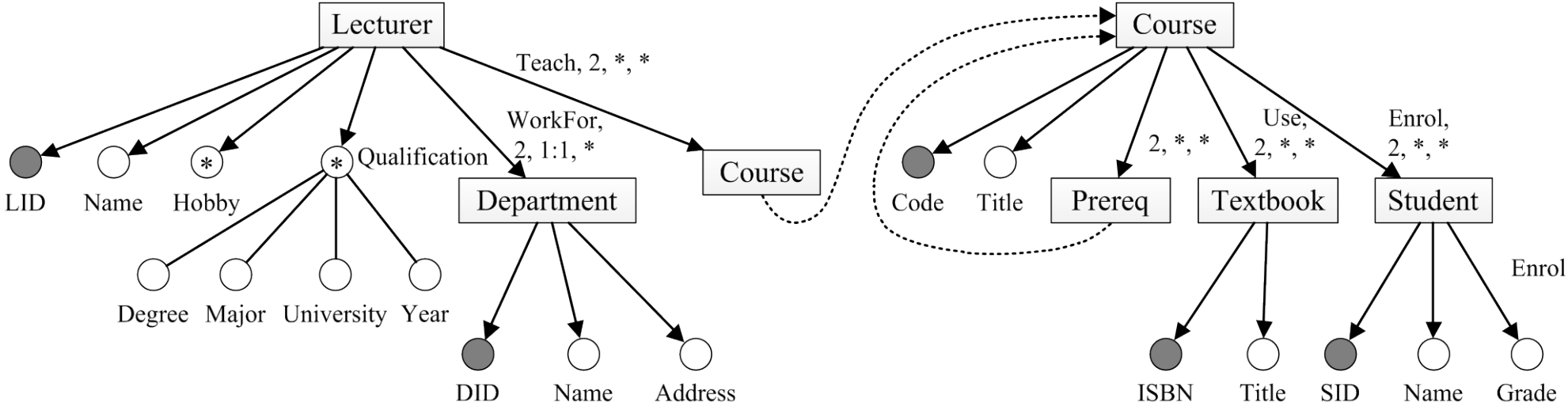


(example XML fragment)

- cannot distinguish between **object attribute** $(Name)$ vs **relationship attribute** $(Grade)$ as both $Name$ and $Grade$ are sub-elements of Student

ORA-SS Data Model [6]

- **ORA-SS data model** [6] is designed to capture **ORA-semantics** in XML data
  - ✓ Distinguish between objects, relationships, and attributes
  - ✓ Capture identifier of object class
  - ✓ Distinguish single valued attribute vs multivalued attribute
  - ✓ Explicit relationship type with name, degree and cardinality
  - ✓ Distinguish object attribute vs relationship attribute



(An ORA-SS schema diagram for the university database)

Outline

- Introduction
- Limitations of Relational Model
- Limitations of XML Data Model
- ORA-semantics in Data and Schema Integration
- ORA-semantics in RDB Keyword Search
- ORA-semantics in XML Keyword Search
- Conclusion

ORA-semantics in Data and Schema Integration

- Data and schema integration has been widely studied. However, the challenge to achieve a good quality integration remain
- Some important concepts and issues:
  1. Different data model
  2. Different relationship type
  3. Local / Global object identifier
  4. Local / Global FD
  5. Semantic dependency
  6. Schematic discrepancy

(1) Different data model

- Databases may have different data models: RDB, XML, NoSQL, etc.
- We need to transform the schemas of different data models into ERD's, and then integrate the databases
- Transformation are done **semi-automatically** with **ORA-semantics enrichment** manually
- ERD captures the ORA-semantics
  - ✓ So improve the correctness of the integrated data/schema

(2) Different relationship type

- **Entity resolution** (i.e., **object identification** and **record linking**) is not enough for data/schema integration
- Consider 2 databases about person and house:

  DB1: $PersonHouse(SSN, Address)$

  DB2: $PersonHouse(SSN, Address)$

  - Even if SSN and $Address$ uniquely identify a person and a house, we cannot integrate DB1 and DB2 directly by merging them because

    - **DB1 may capture relationship type** Own i. e. **person owns house**
    - **DB2 may capture relationship type** $Live$ i. e. **person lives in house**

  ❖ The 2 relationship types between person and house are different

  ❖ So, we also need **relationship resolution / identification**

(3) Local / Global object identifier

- We need to consider **local** object identifier vs **global** object identifier for correct data/schema integration
- Consider 2 databases with the same schema:

  DB1: $Enrol(SID, Code, Grade)$
  DB2: $Enrol(SID, Code, Grade)$

  - We cannot integrate DB1 and DB2 directly by merging them because they may come from 2 universities, and the same SID and $Code$ may refer to different students and courses

  ❖ SID and $Code$ are **local** identifiers.

  ❖ We need to know the **global identifiers** for data integration.

(4) Local / Global FD

- We need to consider local FD vs global FD for correct data/schema integration
- Consider 2 bookstore databases:

DB1: $Book(ISBN, Title, First\_Author, Price)$

DB2: $Book(ISBN, Title, First\_Author, Price)$

- ❖ We cannot integrate DB1 and DB2 directly because the same book may have different prices in different stores
- ❖ We have

global FD: $ISBN \rightarrow \{Title, First\_Author\}$

local FD: $ISBN \rightarrow Price$

- ❖ The integrated database should include 2 relations:

$Book\_infor\ (ISBN, Title, First\_Author)$

$Book\_price\ \ (ISBN, bookstore, Price)$

(5) Semantic dependency [2]

- Semantic dependency [2] is used to capture the **semantic relationship** between 2 sets of attributes
- Consider 2 relations about employees and departments

R1: $Emp(EID, Ename, Joindate, DID)$

R2: $Dept(DID, Dname)$

with FDs: $EID \rightarrow \{Ename, Joindate, DID\}$ & $DID \rightarrow Dname$

- It is unclear if $Joindate$ is
  - the **date** when an **employee joined the company** **or**
  - the **date** when an **employee started working for a department**

- However, if $\{EID, DID\} \xrightarrow{Sem} Joindate$ holds, then $Joindate$ indicates the date when an employee started working for a department

(6) Schematic discrepancy [7]

- Schematic discrepancy [7] occurs when the name of an attribute or a relation in one database corresponds to attribute values in the other databases

- Suppose we want to store the quantities of parts supplied by suppliers in each month of the year.
  - There are 3 equivalent designs:

DB1: $Supply(SID, PID, Month, Quantity)$

DB2: $Supply(SID, PID, Jan, Feb, \ldots, Dec)$

DB3: $Jan\_Supply(SID, PID, Quantity)$

$Feb\_Supply(SID, PID, Quantity)$

$\ldots$

$Dec\_Supply(SID, PID, Quantity)$

(6) Schematic discrepancy [7] (cont'd)

DB1: $Supply(SID, PID, Month, Quantity)$

DB2: $Supply(SID, PID, Jan, Feb, \ldots, Dec)$

DB3: $Jan\_Supply(SID, PID, Quantity)$

$Feb\_Supply(SID, PID, Quantity)$

$\ldots$

$Dec\_Supply(SID, PID, Quantity)$

- The value of $Month$ in DB1 corresponds to attribute names in DB2, and a relation name in DB3

- We remove the context of schema constructs by transforming attributes that cause schematic discrepancy into object classes, relationship types, and attributes [7].

NUS National University of Singapore

Summary

- Many issues must be considered during data and schema integration:

1. Different data model
2. Different relationship type
3. Local/Global object identifier
4. Local/Global FD
5. Semantic dependency
6. Schematic discrepancy

❖ All the above require ORA-semantics to achieve a good quality integration

Outline

- Introduction
- Limitations of Relational Model
- Limitations of XML Data Model
- ORA-semantics in Data and Schema Integration
- ORA-semantics in RDB Keyword Search
- ORA-semantics in XML Keyword Search
- Conclusion

Querying a database - RDB or XML - 2 ways

NUS National University of Singapore

Structured Search (e.g., SQL XPath, XQuery)

```
SELECT E.Grade
FROM Student S, Enrol E, Course C
WHERE S.SID=E.SID AND E.Code=C.Code
      AND S.Name LIKE '%John%' AND C.Title LIKE '%Java%'
```

- precise (+)
- expressive (+)
- learn complex query languages (-)
- need to know schema (-)

Current Keyword Search (keyword query)

- unsatisfactory answers (-)
- not expressive (-)
- user friendly (+)
- users do not know schema (+)

Unsatisfactory answers

- **Meaningless answers**
- **Missing answers**
- **Duplicated answers**
- **Incomplete answers**
- **Schema-dependent answers**

Show later

Querying a database - RDB or XML

NUS National University of Singapore

Structured Search
(e.g., SQL XPath, XQuery)

```
SELECT E.Grade
FROM Student S, Enrol E, Course C
WHERE S.SID=E.SID AND E.Code=C.Code
        AND S.Name LIKE '%John%' AND C.Title LIKE '%Java%'
```

- precise (+)
- expressive (+)
- learn complex query languages (-)
- need to know schema (-)

Current Keyword Search
(keyword query)

- unsatisfactory answers (-)
- not expressive (-)
- user friendly (+)
- users do not know schema (+)

How to have advantages of both structured search and KWS?

Querying a database - RDB or XML

Structured Search
(e.g., SQL XPath, XQuery)

```
SELECT E.Grade
FROM Student S, Enrol E, Course C
WHERE S.SID=E.SID AND E.Code=C.Code
        AND S.Name LIKE '%John%' AND C.Title LIKE '%Java%'
```

- precise (+)
- expressive (+)
- learn complex query languages (-)
- need to know schema (-)

Current Keyword Search
(keyword query)

- not satisfactory answers (-)
- not expressive (-)
- user friendly (+)
- users do not know schema(+)

SEARCH → **Keyword SEARCH**

- **More satisfactory answers**
- **More expressive queries**

ORA-semantics in

RDB Keyword Search - Background

- **RDB query processing** *Example: University database*

Student

SID	Name
S1	Bill
S2	John
S3	Mary

Course

Code	Title	LID
CS301	IR	L2
CS521	DB	L1
CS203	Java	L1

Department

DID	Name	Address
D1	Computing	Smith Street
D2	Business	John Street

Lecturer

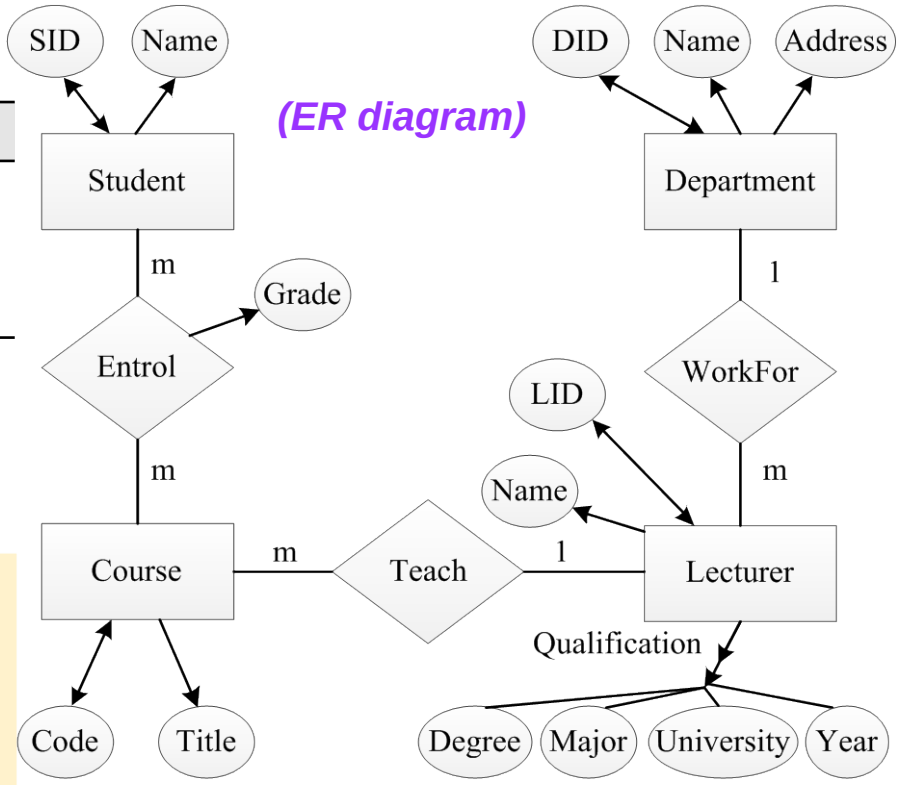
LID	Name	DID
L1	Smith	D1
L2	Smith	D2
L3	Steven	D1

Enrol

	SID	Code	Grade
E1	S1	CS521	A
E2	S2	CS203	B
E3	S2	CS521	A
E4	S3	CS203	A
E5	S3	CS301	B

Qualification

	DID	Degree	Major	University	Year
Q1	L1	PhD	CS	NUS	2016
Q2	L3	PhD	CS	SMU	2015
Q3	L3	Master	EE	NTU	2013



(ER diagram)

Query: find grade that student John obtains in Java course

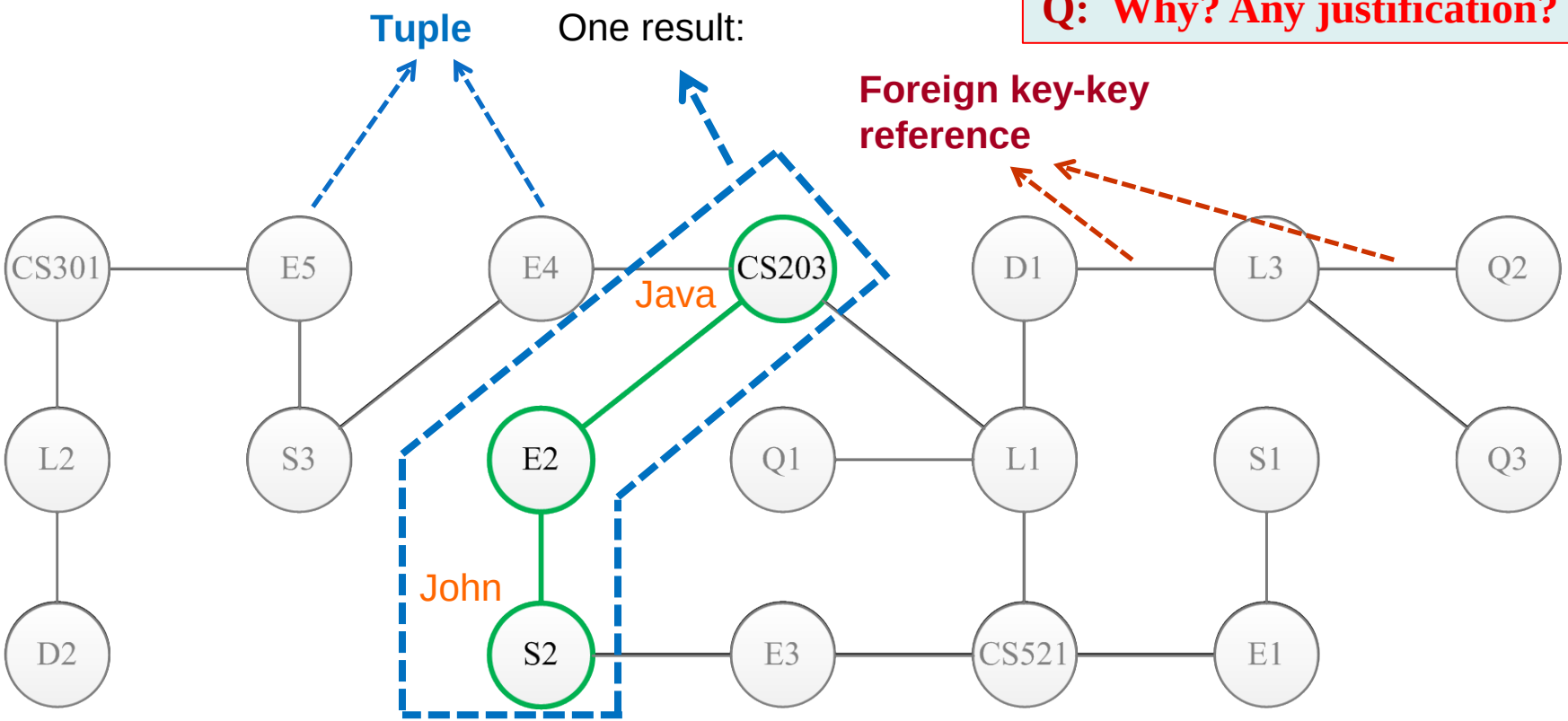
```
SELECT E.Grade
FROM Student S, Enrol E, Course C
WHERE S.SID=E.SID AND E.Code=C.Code
        AND S.Name LIKE '%John%' AND C.Title LIKE '%Java%'
```

– Current data graph approach [8]

KW Query result: Minimal connected subgraph which contains nodes that match keywords (Steiner Tree)

Q={John Java}

Q: Why? Any justification?



(*data graph of university database*)

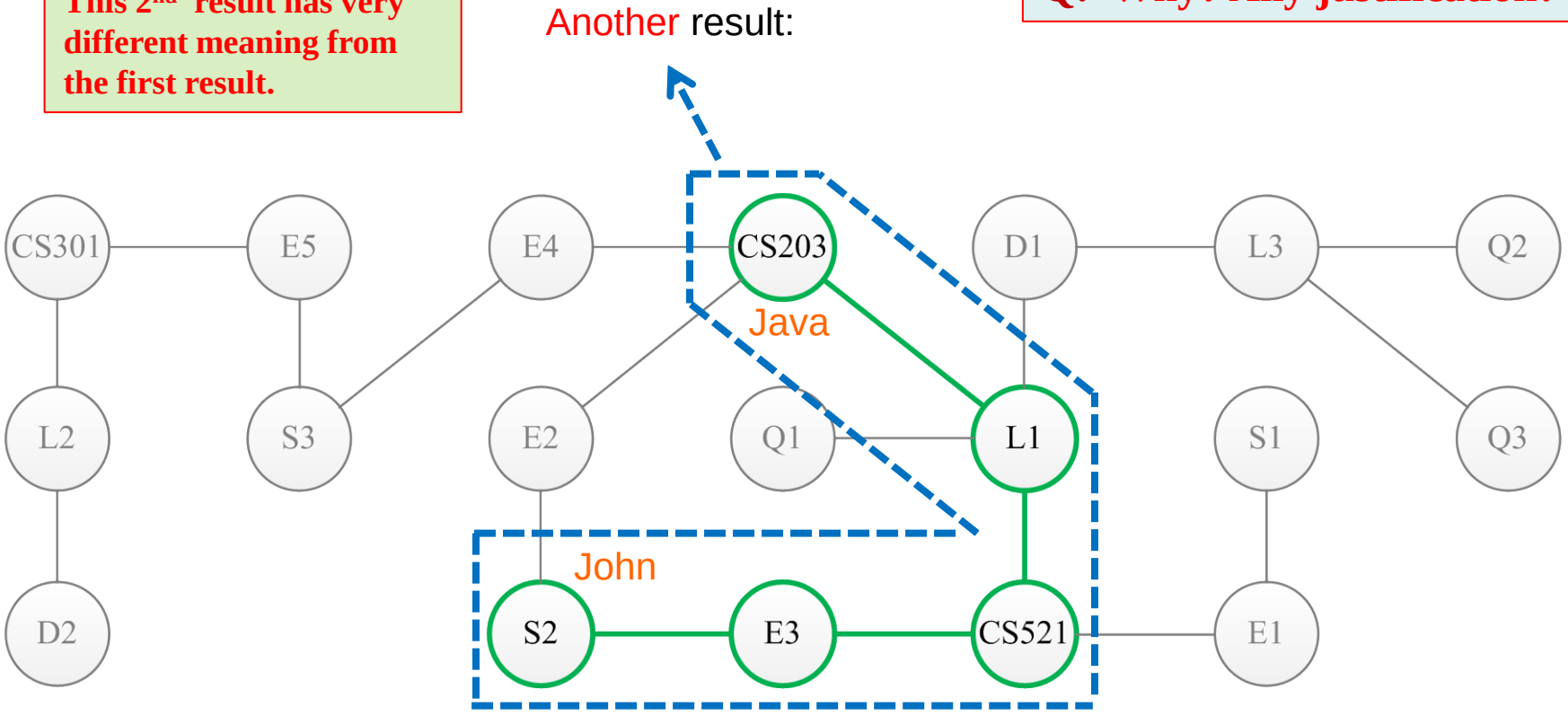
– Current data graph approach [8]

Q={John Java}

Query result: Minimal connected subgraph which contains nodes that match keywords (Steiner Tree)

Q: Why? Any justification?

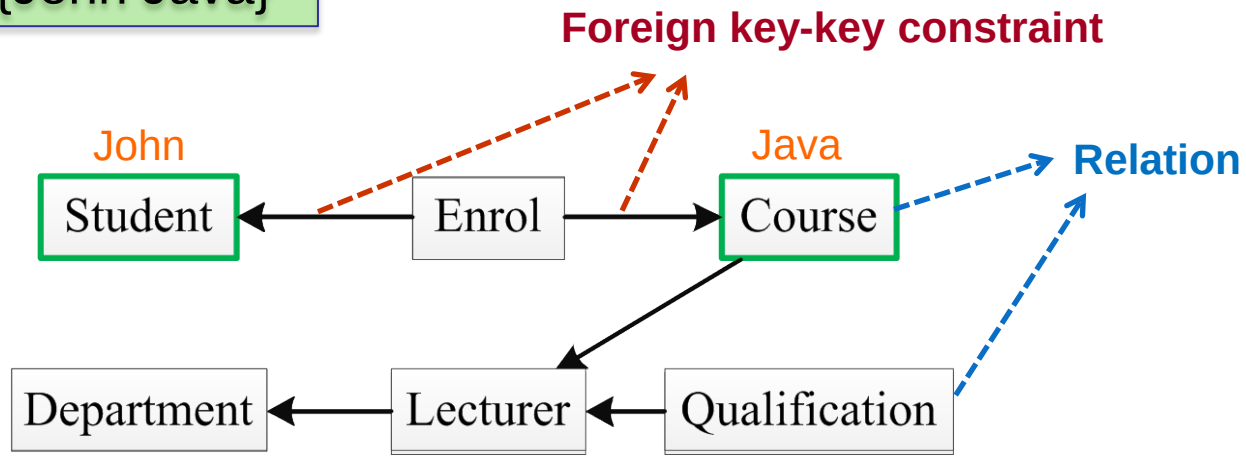
This 2nd result has very different meaning from the first result.



(data graph of university database)

– Current schema graph approach [9]

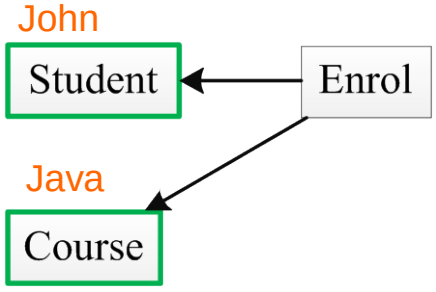
Q={John Java}



(*schema graph of university database*)

Traverse to obtain a **minimal connected subgraph** which covers keywords with tuples matching the keywords

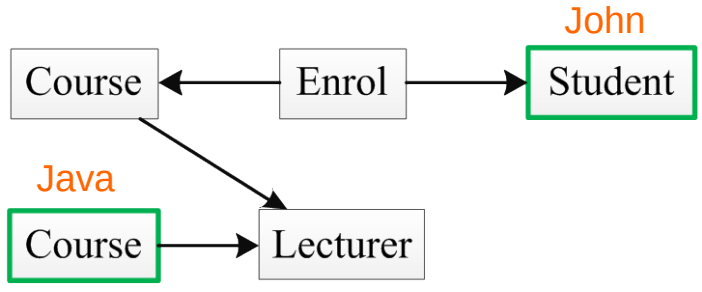
One graph:



Translate into SQL

```
SELECT *
FROM Student S, Enrol E, Course C
WHERE S.SID=E.SID AND E.Code=C.Code
      AND S.Name LIKE '%John%'
      AND C.Title LIKE '%Java%'
```

Another graph:



– Summary of both current approaches

NUS National University of Singapore

- Current keyword query processing methods
  - Based on **foreign key references**
  - 2 approaches:

i. Data Graph approach

1) ***Nodes*** *are tuples;* ***edges*** *are foreign key references between 2 tuples.*
2) *Find minimum Steiner trees of the matched tuples (nodes).*

ii. Schema Graph approach

1) ***Nodes*** *are relations;* ***edges*** *are foreign key references between relations.*
2) *Generate SQL statements for the keyword query.*

– Problems of current RDB keyword search

Both schema graph approach and data graph approach have following problems:

1) Incomplete object answer
2) Incomplete relationship answer
3) Meaningless answer
4) Complex answer
5) Inconsistent types of answers
6) Schema dependent answer

❖ Reason:

They are unaware of ORA-semantics, and thus cause problems

– Problems of current RDB keyword search

1) Incomplete object answer

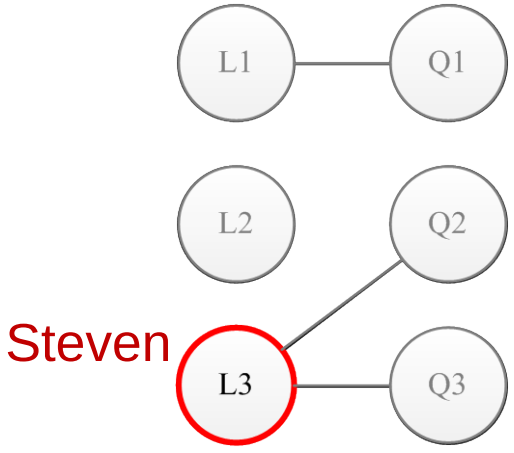
Q = {Steven}

Lecturer

LID	Name	DID
L1	Smith	D1
L2	Smith	D2
L3	Steven	D1

Qualification

	DID	Degree	Major	University	Year
Q1	L1	PhD	CS	NUS	2016
Q2	L3	PhD	CS	SMU	2015
Q3	L3	Master	EE	NTU	2013



Corresponding data graph

Only 1 answer:
L3

Additional information about qualifications of Steven is expected because they are properties of lecturers

– Problems of current RDB keyword search

2) Incomplete relationship answer

Student

SID	Name
S1	Bill
S2	John
S3	Mary

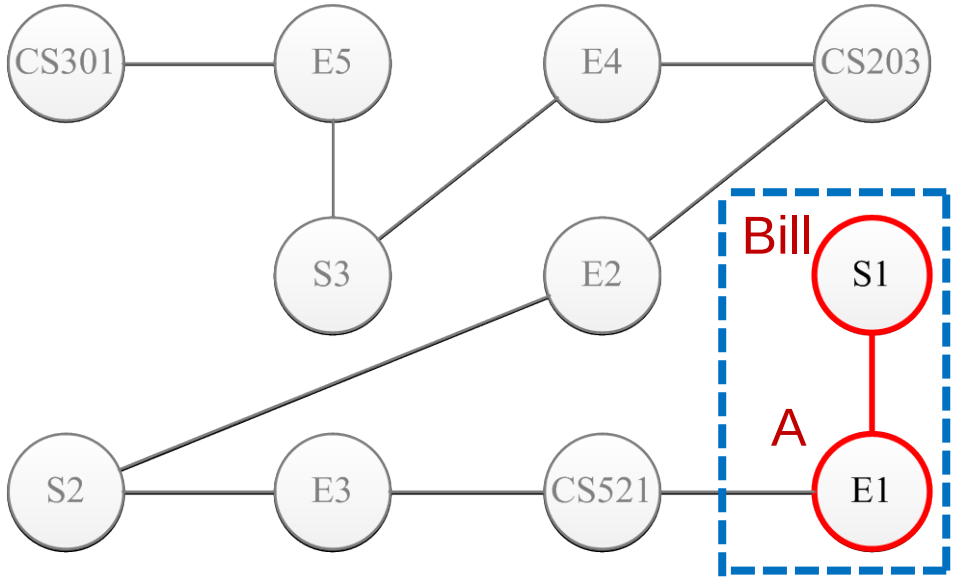
Enrol

	SID	Code	Grade
E1	S1	CS521	A
E2	S2	CS203	B
E3	S2	CS521	A
E4	S3	CS203	A
E5	S3	CS301	B

Course

Code	Title	LID
CS301	IR	L2
CS521	DB	L1
CS203	Java	L1

Q = {Bill A}



Corresponding data graph

One answer:
S1-E1

Expected:
Grade is a relationship attribute;
The details of other participating objects (i.e. course) of the relationship are expected

– Problems of current RDB keyword search

3) Meaningless answer

Student

SID	Name
S1	Bill
S2	John
S3	Mary

Course

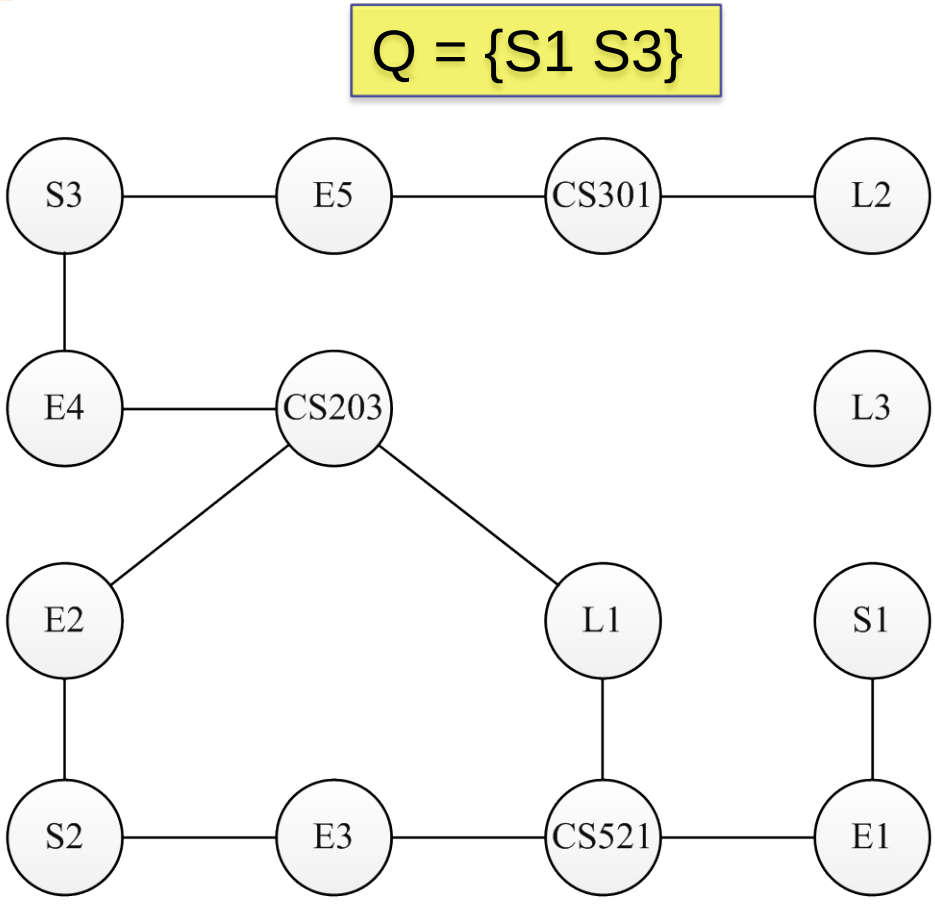
Code	Title	LID
CS301	IR	L2
CS521	DB	L1
CS203	Java	L1

Lecturer

LID	Name	DID
L1	Smith	D1
L2	Smith	D2
L3	Steven	D1

Enrol

	SID	Code	Grade
E1	S1	CS521	A
E2	S2	CS203	B
E3	S2	CS521	A
E4	S3	CS203	A
E5	S3	CS301	B



Corresponding data graph

– Problems of current RDB keyword search

3) Meaningless answer (cont.)

Student

SID	Name
S1	Bill
S2	John
S3	Mary

Course

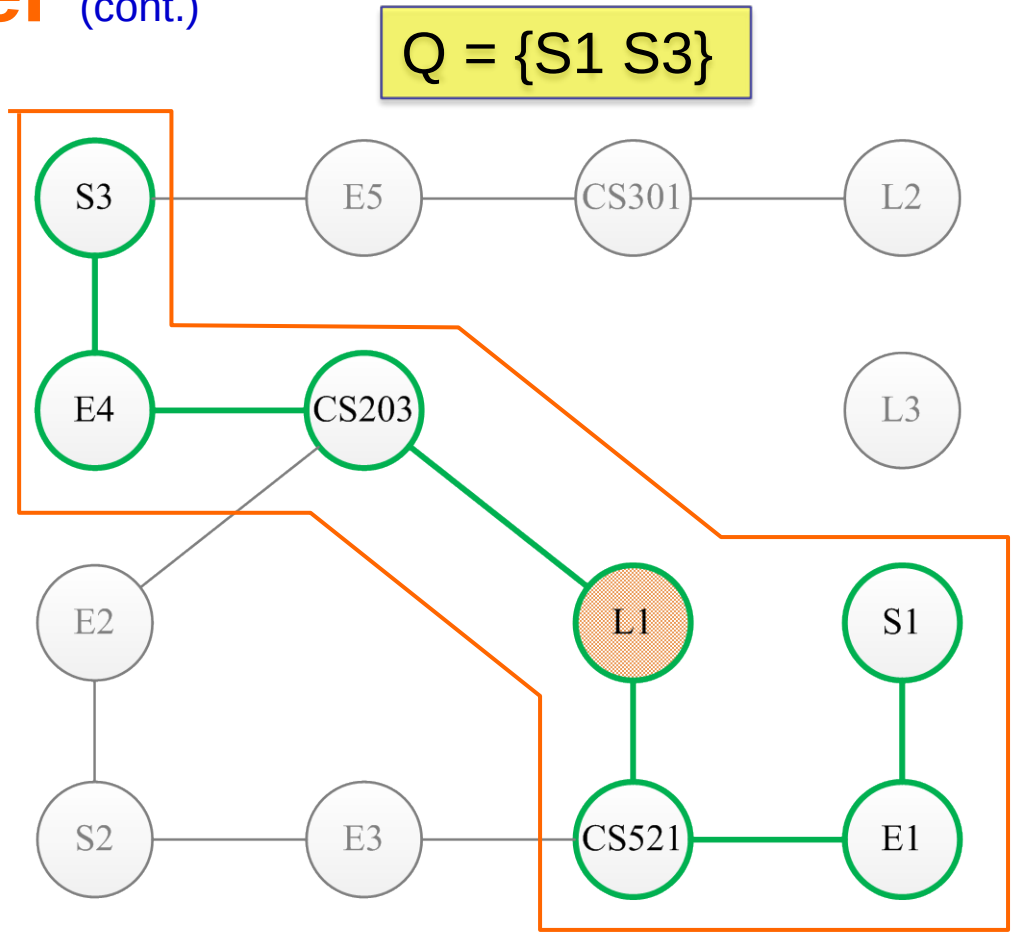
Code	Title	LID
CS301	IR	L2
CS521	DB	L1
CS203	Java	L1

Lecturer

LID	Name	DID
L1	Smith	D1
L2	Smith	D2
L3	Steven	D1

Enrol

	SID	Code	Grade
E1	S1	CS521	A
E2	S2	CS203	B
E3	S2	CS521	A
E4	S3	CS203	A
E5	S3	CS301	B



2 answers:

1st answer: S3-E4-CS203-L1-CS5201-E1-S1

Meaning? (difficult to know from the minimal connected subgraph):

the common lecturer of S1 & S3 (meaningful)

– Problems of current RDB keyword search

3) Meaningless answer (cont.)

Student

SID	Name
S1	Bill
S2	John
S3	Mary

Course

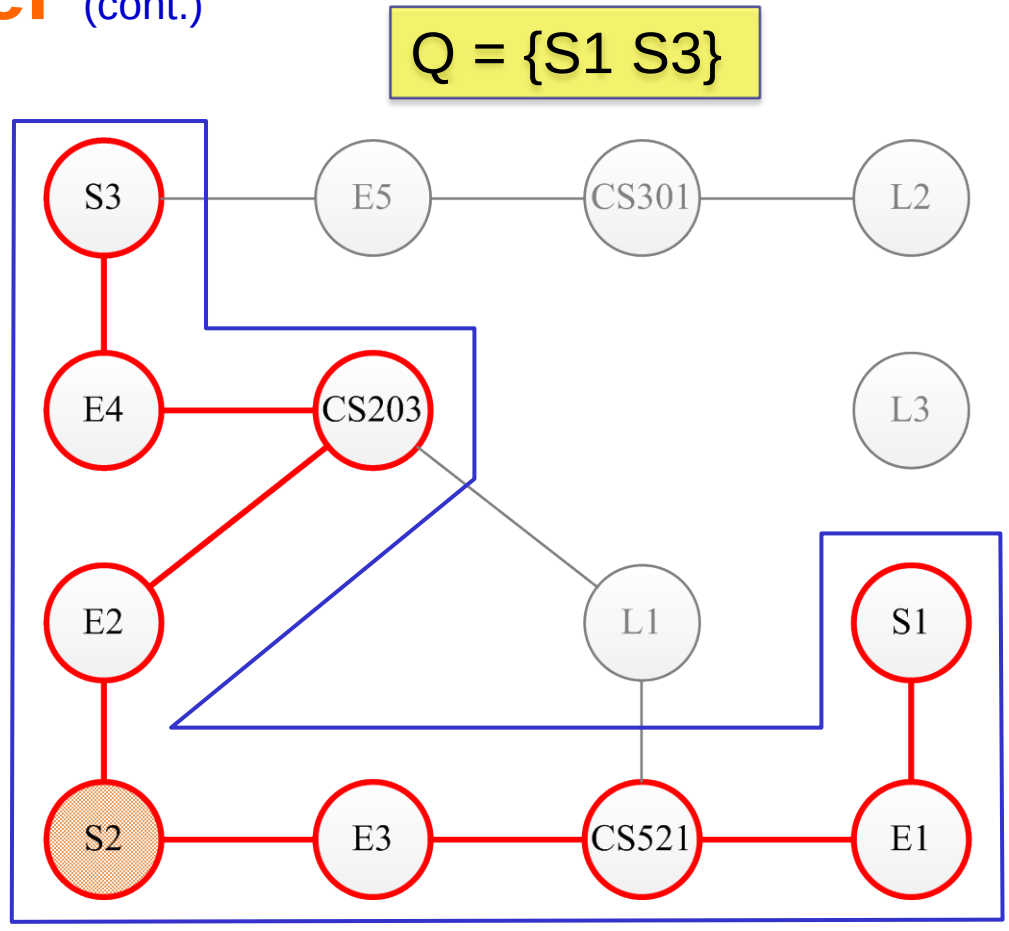
Code	Title	LID
CS301	IR	L2
CS521	DB	L1
CS203	Java	L1

Lecturer

LID	Name	DID
L1	Smith	D1
L2	Smith	D2
L3	Steven	D1

Enrol

	SID	Code	Grade
E1	S1	CS521	A
E2	S2	CS203	B
E3	S2	CS521	A
E4	S3	CS203	A
E5	S3	CS301	B



2nd answer:

S3-E4-CS203-E2-S2-E3-CS5201-E1-S1

Meaning? S2 enrolls some course with S1 and enrolls another course with S3.

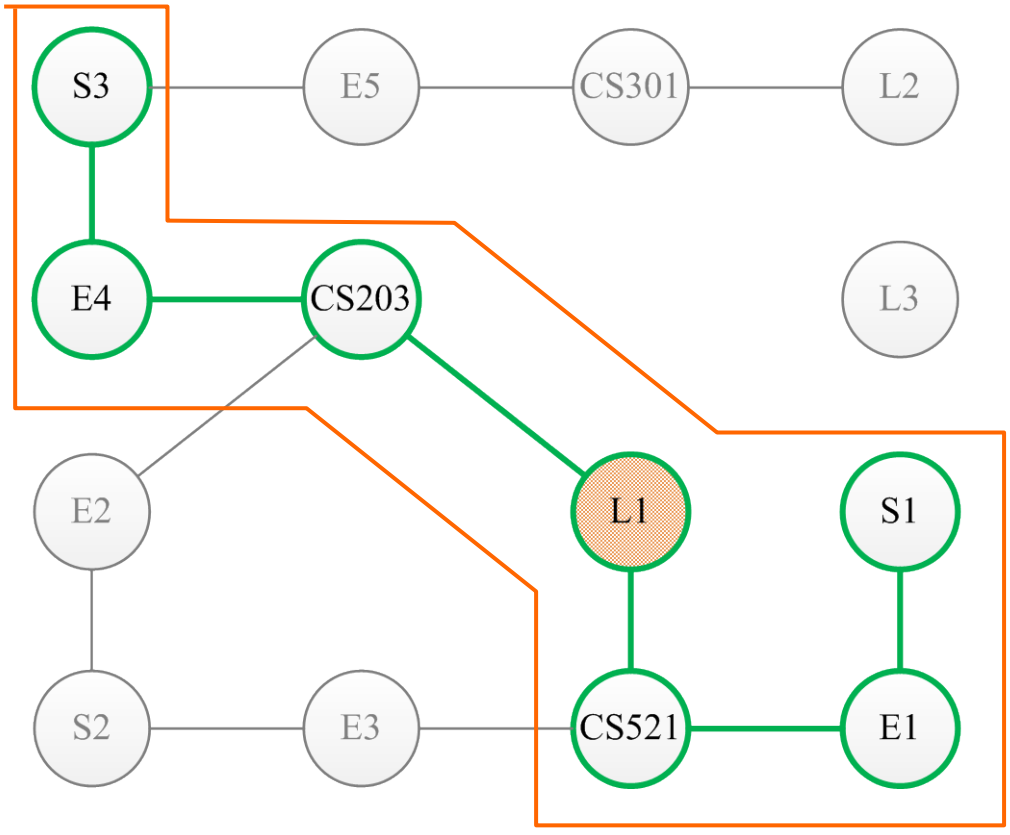
Probably not meaningful: not correspond to an LCA of any hierarchical structure XML doc representing the same database

– Problems of current RDB keyword search

4) Complex answer

- Difficult to understand the meaning

The 1st answer in previous example



Q = {S1 S3}

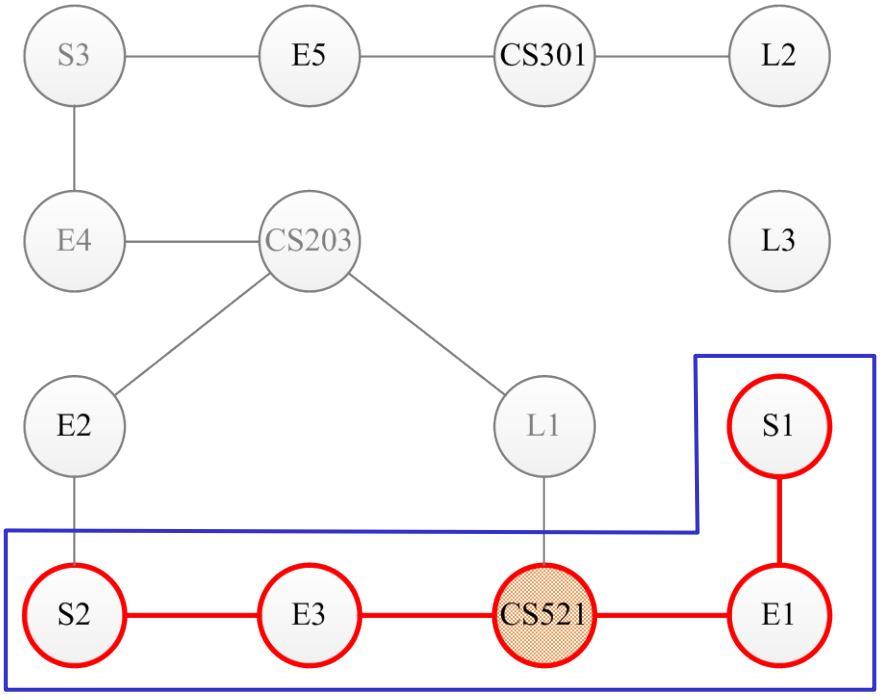
How to present the answer?

1) Structures are difficult to understand;
2) Some tuples are important while some others are not

– Problems of current RDB keyword search

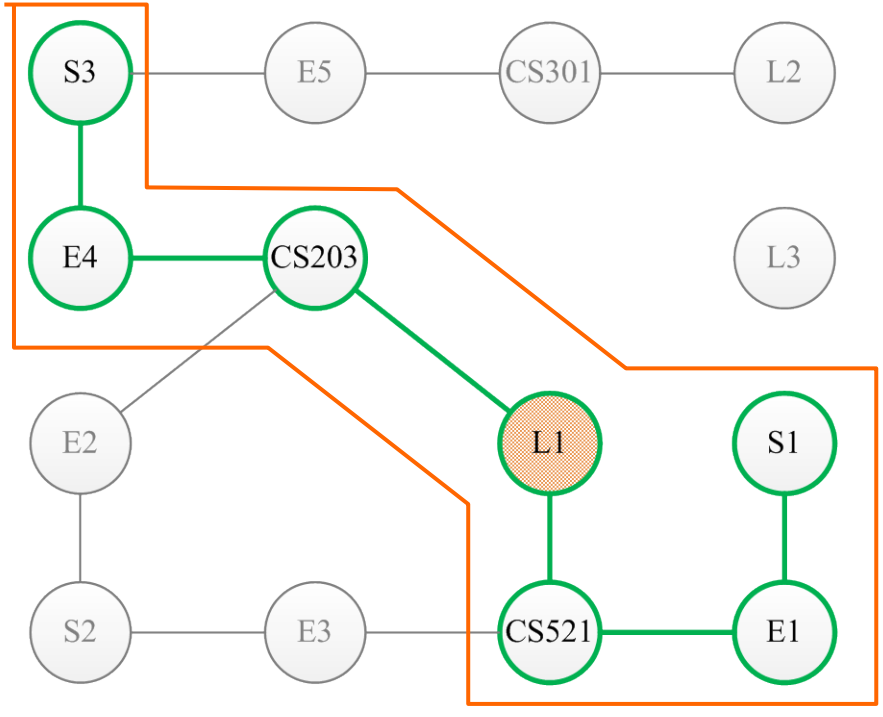
5) Inconsistent types of answers

Q1 = {S1 S2}



common course of S1 & S2

Q2 = {S1 S3}



common lecturer of S1 & S3

Two similar queries have very different answers and user will get confused

– Problems of current RDB keyword search

6) Schema dependent answer

Student

SID	Name
S1	Bill
S2	John
S3	Mary

Enrol

	SID	Code	Grade
E1	S1	CS521	A
E2	S2	CS203	B
E3	S2	CS521	A
E4	S3	CS203	A
E5	S3	CS301	B

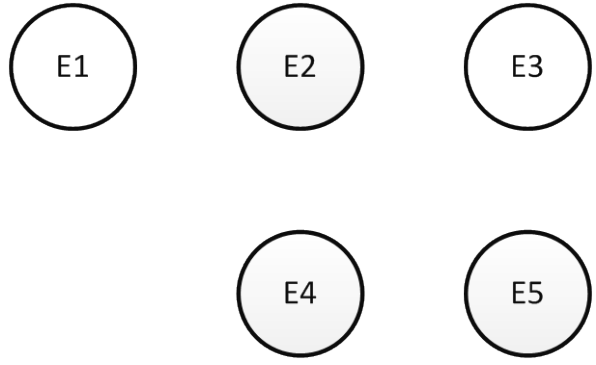
Course

Code	Title	LID
CS301	IR	L2
CS521	DB	L1
CS203	Java	L1

If we Denormalize →

Enrollment (1NF)

	SID	Name	Code	Title	LID	Grade
E1	S1	Bill	CS521	DB	L1	A
E2	S2	John	CS203	Java	L1	B
E3	S2	John	CS521	DB	L1	A
E4	S3	Mary	CS203	Java	L1	A
E5	S3	Mary	CS301	IR	L2	B



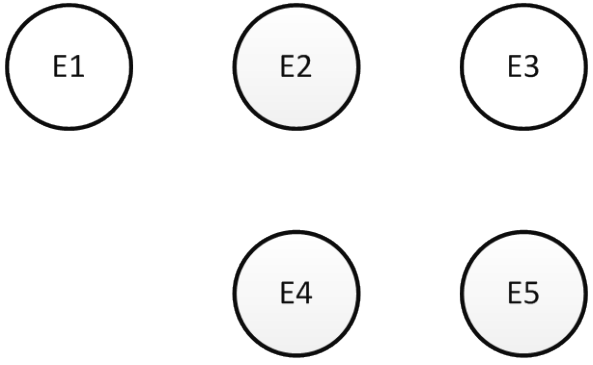
(Corresponding data graph which has only nodes and no edge)

– Problems of current RDB keyword search

6) Schema dependent answer (cont.)

Enrollment (1NF)

	SID	Name	Code	Title	LID	Grade
E1	S1	Bill	CS521	DB	L1	A
E2	S2	John	CS203	Java	L1	B
E3	S2	John	CS521	DB	L1	A
E4	S3	Mary	CS203	Java	L1	A
E5	S3	Mary	CS301	IR	L2	B



(Corresponding data graph which has only nodes and no edge)

Q = {S3}

2 answers:

1) E4
2) E5

The information of student S3 are duplicated.

- Should only output E4 or E5

Q = {S1 S3}

No answer returns because no connected subgraph contains all the keywords

Expected answers: common lecturer of S1 & S3

– Problems of current RDB keyword search

Summary.

Both schema graph approach and data graph approach have following problems:

1) Incomplete object answer
2) Incomplete relationship answer
3) Meaningless answer
4) Complex answer
5) Inconsistent types of answers
6) Schema dependent answer

❖ They are unaware of ORA-semantics, and thus cause problems

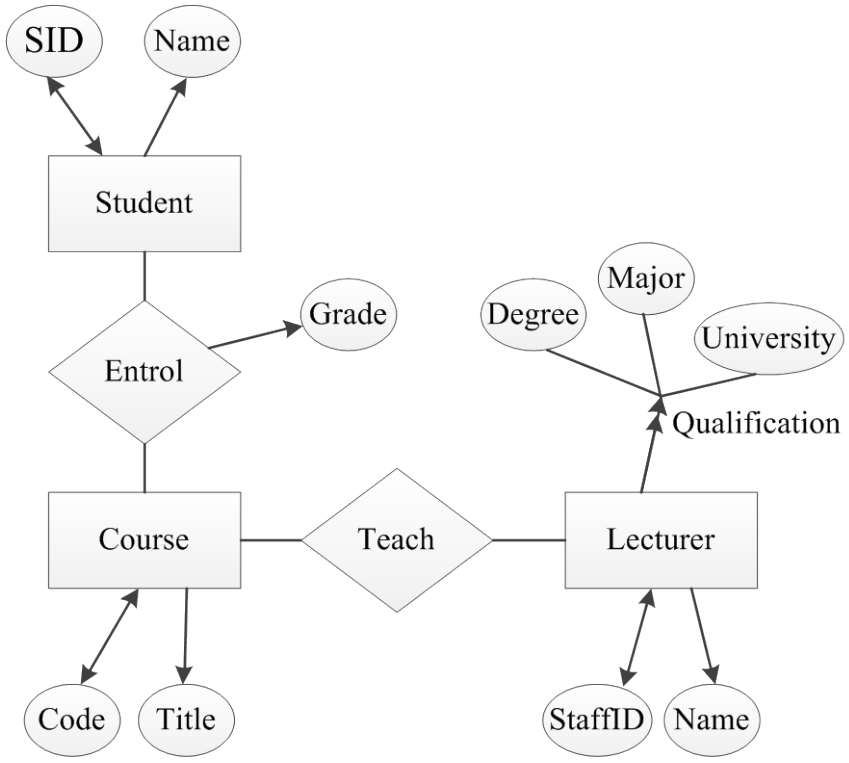
– our ORA-Semantics approach

- We use ORA semantics and classify relations in an RDB into object relations, relationship relations, component relations, and mixed relations
  - An **object relation** captures the information of objects
  - A **relationship relation** captures the information of relationships
  - A **mixed relation** contains information of both objects and relationships, which occurs when we have a many-to-one relationship
  - The information of multivalued attributes of objects and relationships are stored as **component relations** of the respective object or relationship

These different types of relations capture the ORA-semantics explicitly.

– our ORA-Semantics approach (Example)

(ER diagram of University database)



(schema)

Student(SID, Name)

Course(Code, Title, LID)

$\text{Course}[\text{LID}] \subseteq \text{Lecturer}[\text{StaffID}]$

Enrol(SID, Code, Grade)

$\text{Enrol}[SID] \subseteq \text{Student}[\text{SID}]$

$\text{Enrol}[\text{Code}] \subseteq \text{Course}[\text{Code}]$

Lecturer(LID, Name, DID)

$\text{Lecturer}[\text{DID}] \subseteq \text{Department}[\text{DID}]$

Department(DID, Name, Address)

Qualification(LID, Degree, Major, University)

$\text{Qulification}[\text{LID}] \subseteq \text{Lecturer}[\text{LID}]$

Types of Relations

- Object Relation
- Relationship Relation
- Mixed Relation
- Component Relation of object/relationship

– Object-Relationship-Mixed (ORM) graph

- **ORM data graph** $G_D(V, E)$ is an undirected graph
  - Each **node** $v \in V$ corresponds to a tuple of an object/relationship/mixed relation, including tuples of its **component relations**
  - $v.type \in \{object, relationship, mixed\}$
  - Each **edge** $e(u, v) \in E$ indicates a foreign key-key reference between tuples in u and v

- **ORM schema graph** $G_S(V, E)$ is an undirected graph
  - Each **node** $v \in V$ corresponds to an object/relationship/mixed relation, and its associated **component relations**
  - $v.type \in \{object, relationship, mixed\}$
  - Each **edge** $e(u, v) \in E$ indicates a foreign key-key reference between relations in u and v

– ORM data and schema graph (Example)

Student

SID	Name
S1	Bill
S2	John
S3	Mary

Course

Code	Title	LID
CS301	IR	L2
CS521	DB	L1
CS203	Java	L1

Department

DID	Name	Address
D1	Computing	Smith Street
D2	Business	John Street

Lecturer

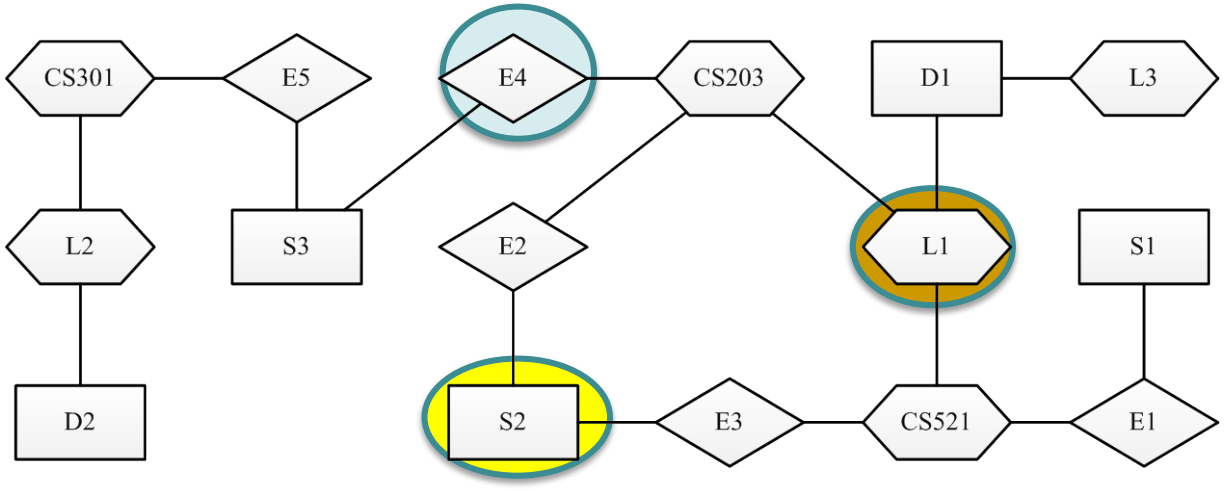
LID	Name	DID
L1	Smith	D1
L2	Smith	D2
L3	Steven	D1

Qualification

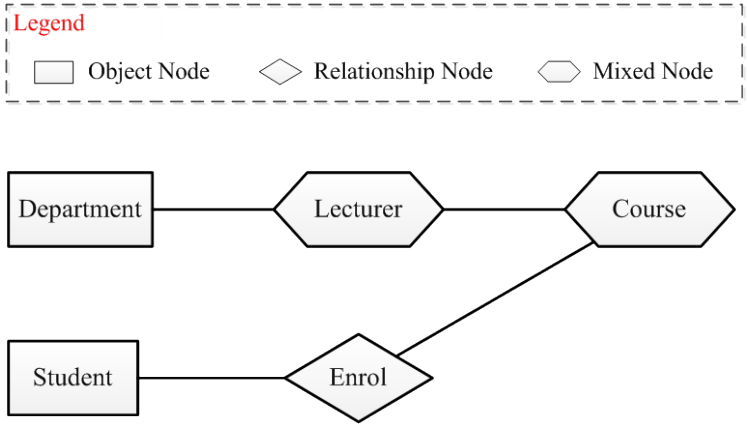
	DID	Degree	Major	University	Year
Q1	L1	PhD	CS	NUS	2016
Q2	L3	PhD	CS	SMU	2015
Q3	L3	Master	EE	NTU	2013

Enrol

	SID	Code	Grade
E1	S1	CS521	A
E2	S2	CS203	B
E3	S2	CS521	A
E4	S3	CS203	A
E5	S3	CS301	B



ORM data graph



ORM schema graph

ORA-Semantics in RDB Keyword Search

NUS National University of Singapore

Topics to be discussed

1) Search over the ORM data/schema graph and process queries based on the types of keyword match nodes [10]
   - Utilize ORA semantics to retrieve more complete and informative answers and solves the mentioned problems of current RDB keyword search

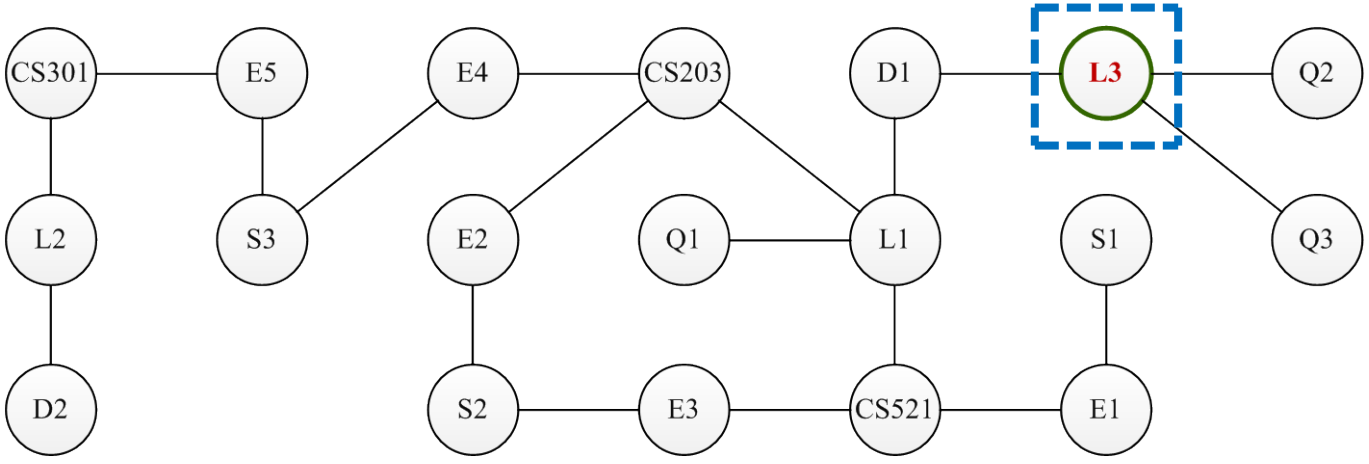
2) Extend keyword queries to include metadata keywords [11]
   - Utilize ORA semantics to identify keyword context and search target in order to infer user's search intention
   - This solves the problem of inherent ambiguity of keyword query

3) Answer aggregate functions in keyword queries [12]
   - Utilize ORA semantics to distinguish objects with the same attribute value and detect duplicate objects and relationships in order to compute aggregates correctly

1) Search over the **ORM data/schema graph** and process queries based on the types of keyword match nodes

Previous Approaches

Q = {Steven}



Lecturer

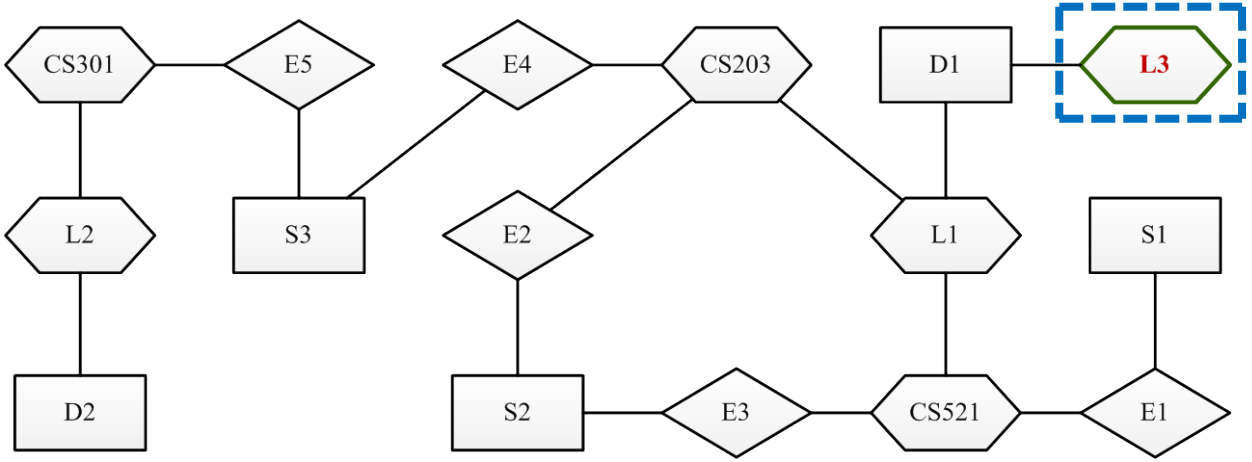
LID	Name	DID
L3	*Steven*	D1

- Return lecturer tuple **L3** only

1) Search over the **ORM data/schema graph** and process queries based on the types of keyword match nodes (cont.)

Our Approach

Q = {Steven}



Lecturer

LID	Name	DID
L3	*Steven*	D1

Qualification

DID	Degree	Major	University	Year
L3	**PhD**	**CS**	**SMU**	**2015**
L3	**Master**	**EE**	**NTU**	**2013**

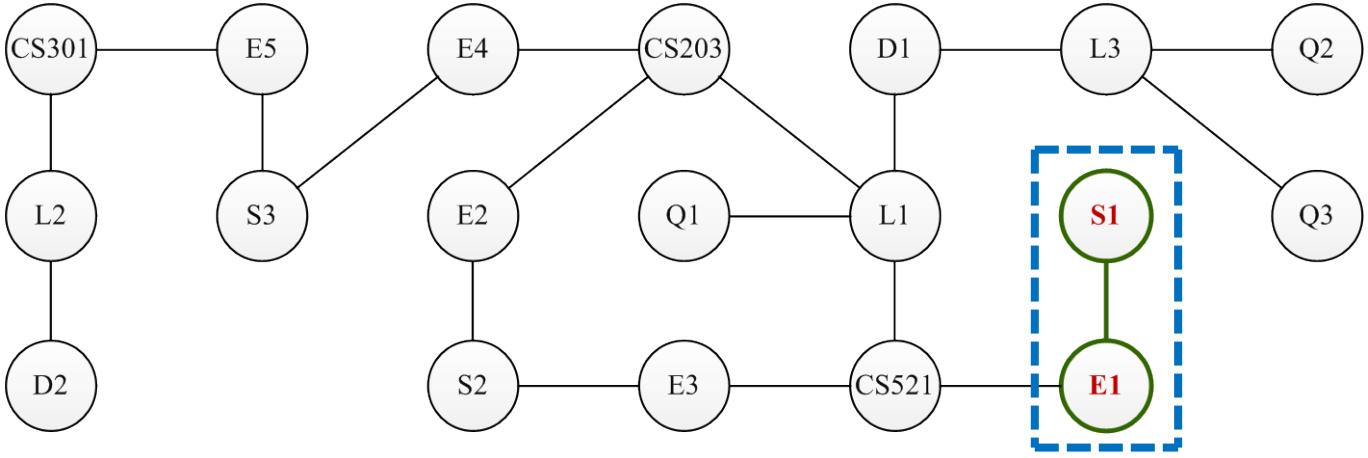
- Return lecturer tuple **L3** **together with his qualifications,** **all properties of the lecturer object.**

Avoid problem of incomplete object answer

1) Search over the **ORM data/schema graph** and process queries based on the types of keyword match nodes (cont.)

Previous Approaches

Q = {Bill A}



Student

SID	Name
S1	*Bill*

Enrol

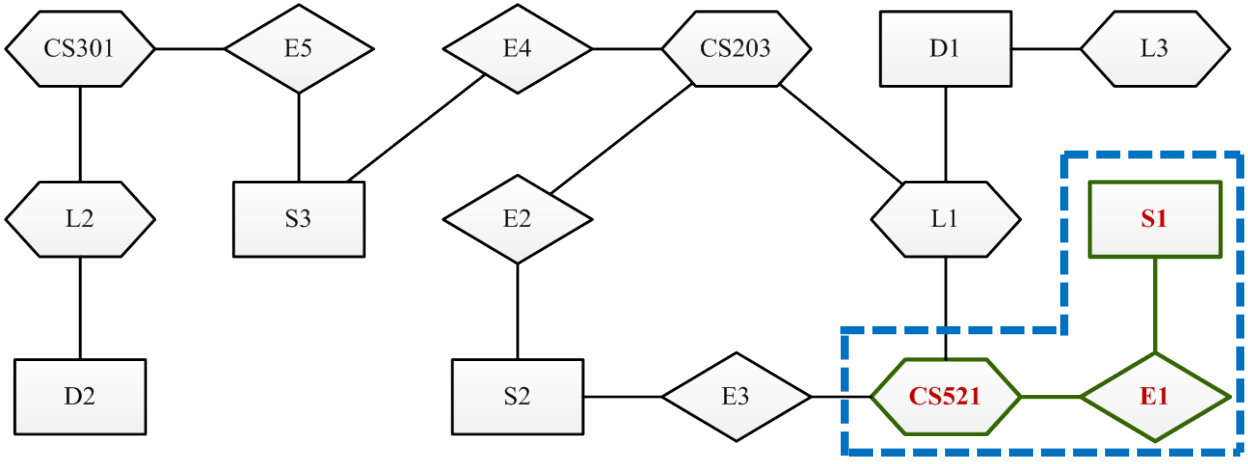
	SID	Code	Grade
E1	S1	CS521	*A*

- Return student tuple **S1** and enrol tuple **E1**

1) Search over the **ORM data/schema graph** and process queries based on the types of keyword match nodes (cont.)

Our Approach

Q = {Bill A}



Student

SID	Name
S1	*Bill*

E1

Enrol

SID	Code	Grade
S1	CS521	*A*

Course

Code	Title	LID
CS521	*DB*	*L1*

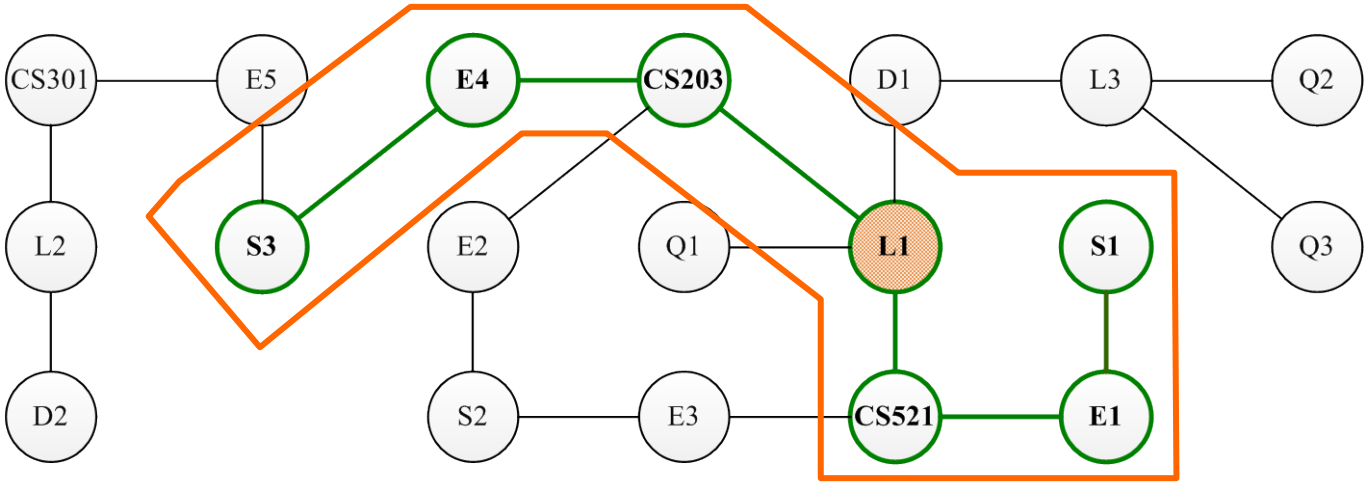
- Return student tuple **S1**, enrol tuple **E1** and **course tuple CS521** as participating object of enrol relationship

Avoid problem of incomplete relationship answer

1) Search over the **ORM data/schema graph** and process queries based on the types of keyword match nodes (cont.)

Previous Approaches

Q = {S1 S3}



2 answers:

1st answer: S3-E4-CS203-L1-CS5201-E1-S1

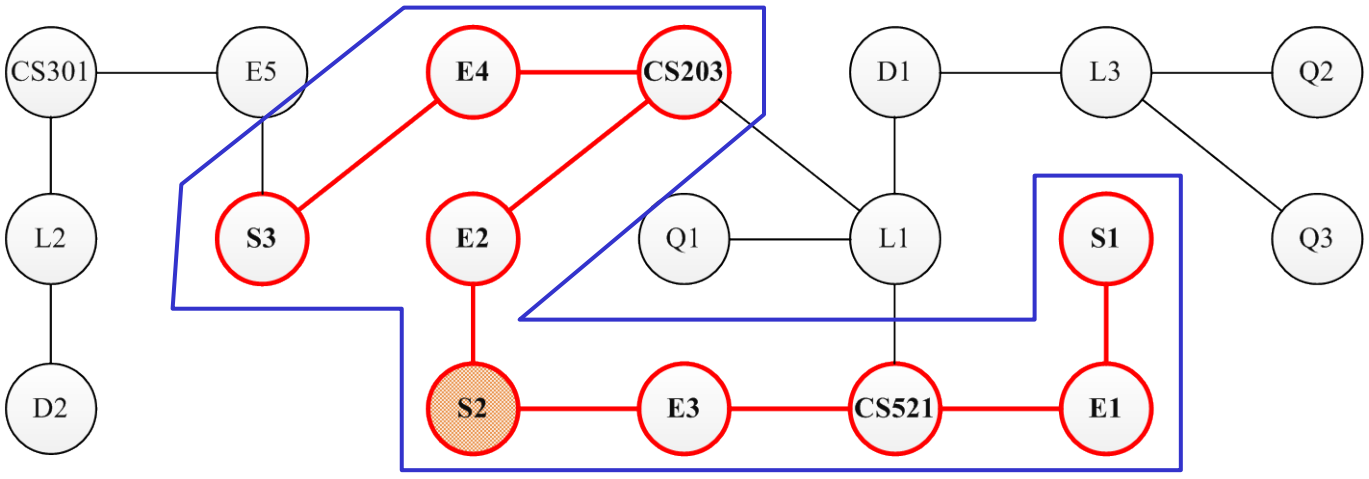
Meaning:

common lecturer of S1 & S3 (meaningful)

1) Search over the **ORM data/schema graph** and process queries based on the types of keyword match nodes (cont.)

Previous Approaches

Q = {S1 S3}



2nd answer:

S3-E4-CS203-E2-S2-E3-CS5201-E1-S1

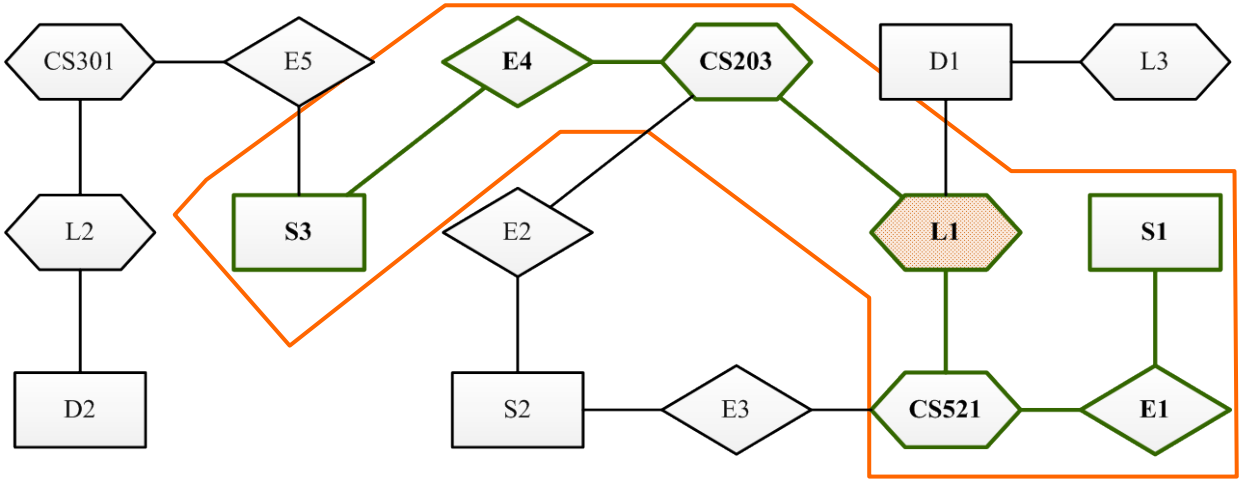
Meaning: S2 enrolls some course with S1 and enrolls another course with S3

(Probably not meaningful)

1) Search over the **ORM data/schema graph** and process queries based on the types of keyword match nodes (cont.)

Our Approach

Q = {S1 S3}



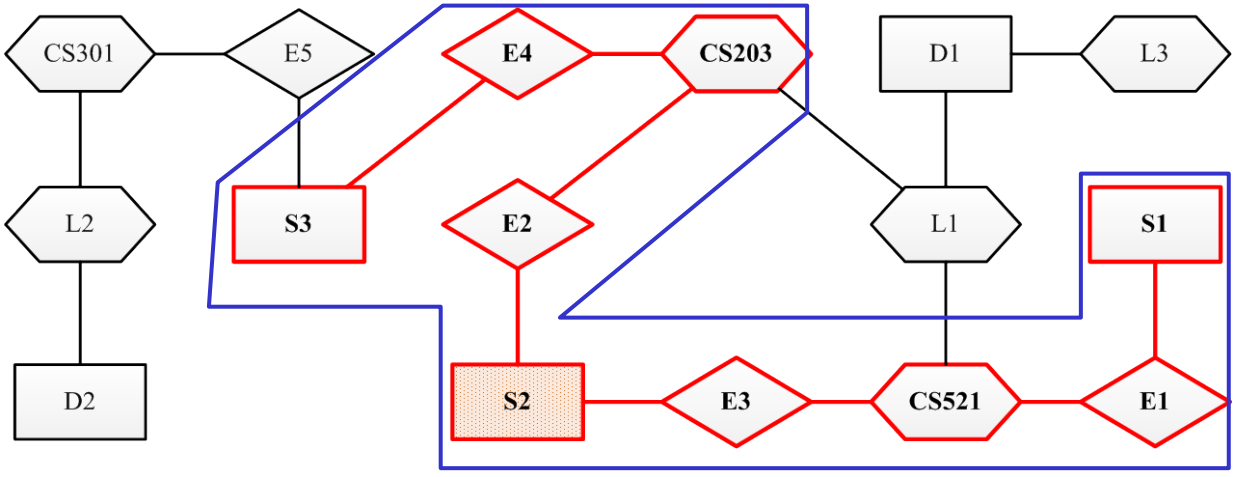
- Meaningful

Paths from **L1** to **S3** and **S1** consists of tuples from distinct relations, representing close relationships from L1 to S3 and S1

1) Search over the **ORM data/schema graph** and process queries based on the types of keyword match nodes (cont.)

Our Approach

Q = {S1 S3}



- Probably not meaningful
- Will not output or low ranking

Avoid problem of meaningless answer

Paths from **S2** to S3 and S1 consists of some tuples from the same relations (i.e. Student, Enrol), representing less close relationships from S2 to S3 and S1

1) Search over the **ORM data/schema graph** and process queries based on the types of keyword match nodes (cont.)

Summary

W e have solved all the problems in the current RDB keyword search except the problem of inconsistent types of answers for similar type of queries, i.e.

1) Incomplete object answer
2) Incomplete relationship answer
3) Meaningless answer
4) Complex answer
5) Schema dependent answer

NUS National University of Singapore

2) Extend keyword queries to include metadata keywords

- Our Observations
  - A keyword query is inherently **ambiguous**
  - However, when a user issues a query, he/she must have some particular search intention in mind
  - Idea: user can explicitly indicate his/her **search intention** whenever possible, to reduce keyword query ambiguity
    - **Augment query with metadata keywords that match relation names and attribute names**

Q = {John Mary} → Q' = {Course Student John Student Mary}

- Keyword Course indicates user is interested in course information (but not Department information)
- Keyword Student gives context that John refers to student name (but not Department at John street)

NUS National University of Singapore

2) Extend keyword queries to include metadata keywords (cont.)

Q = {Course Student John Student Mary}

o Determine objects and relationships referred to by keywords

- Course matches the name of $Course$ relation
- Student matches the name of $Student$ relation
- Mary matches the $Name$ attribute value of a tuple in $Student$ relation
- John has 2 matches:
  1. $Name$ attribute value of a tuple in $Student$ relation
  2. $Address$ attribute value of a tuple in $Department$ relation

2) Extend keyword queries to include metadata keywords (cont.)

Q = {Course Student John Student Mary}

o Determine objects and relationships referred to by keywords

- Course matches the name of *Course* relation
- Student matches the name of *Student* relation
- Mary matches the *Name* attribute value of a tuple in *Student* relation
- John has 2 matches:
  1. *Name* attribute value of a tuple in *Student* relation
  2. *Address* attribute value of a tuple in *Department* relation

Not likely because of the context of Student

➢ {Course} refers to some course object

➢ {Student, John} refers to a student name John

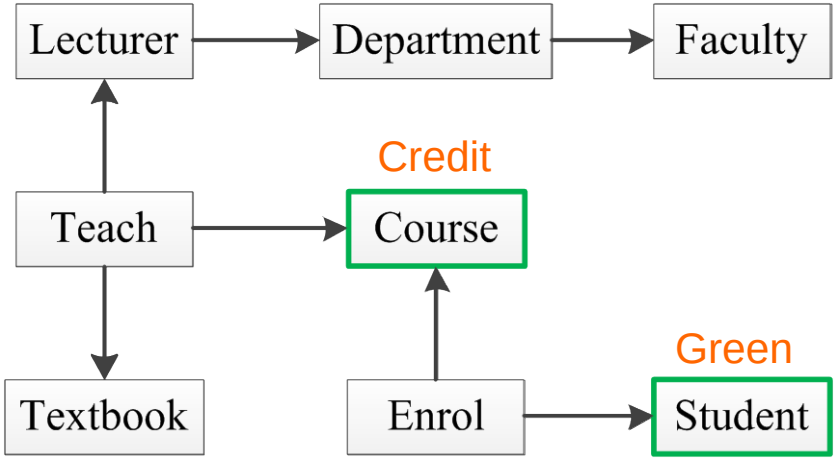
➢ {Student, Mary} refers to a student name Mary

NUS National University of Singapore

3) Answer **aggregate functions** in keyword queries

- SQAK [19] may return incorrect answers
- E.g., find total credits obtained by student Green

Q = {Green SUM Credit}



Student

Sid	Sname	Age
s1	George	22
s2	Green	24
s3	Green	21

Course

Code	Title	Credit
c1	Java	5.0
c2	Database	4.0
c3	Multimedia	3.0

Enrol

Sid	Code	Grade
s1	c1	A
s1	c2	B
s1	c3	B
s2	c1	A
s3	c1	A
s3	c3	B

```
SELECT S.Sname, SUM(C.Credit)
FROM Student S, Enrol E, Course C
WHERE E.Sid=S.Sid AND E.Code=C.Code
      AND S.Sname = 'Green'
GROUP BY S.Sname
```

Output answer: 13

Correct answer: s2 is 5, s3 is 8

Do not distinguish students with the same name and output a total credits of two different students, which is incorrect

[19] SQAK: Doing more with keywords. In SIGMOD, 2008

NUS National University of Singapore

3) Answer aggregate functions in keyword queries (cont.)

- SQAK does not consider Object-Relationship-Attribute (ORA) semantics in the database and thus suffers from the problems of returning incorrect answers
  - cannot distinguish objects with the same attribute value
  - cannot detect duplicates of objects and relationships

❖ So without ORA semantics, it is impossible to process aggregate queries correctly

- Idea: exploit ORA semantics and propose a semantic approach to answer aggregate queries correctly

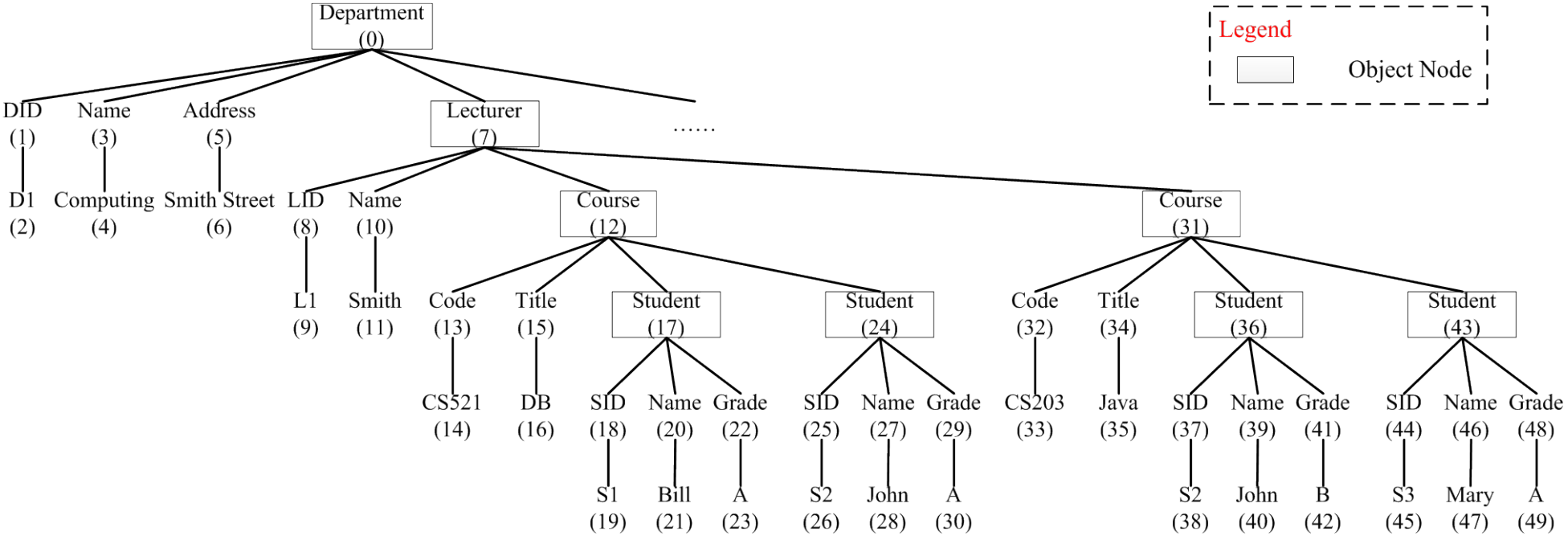
Outline

- Introduction
- Limitations of Relational Model
- Limitations of XML Data Model
- ORA-semantics in Data and Schema Integration
- ORA-semantics in RDB Keyword Search
- ORA-semantics in XML Keyword Search
- Conclusion

ORA-semantics in XML Keyword Search

– Background

- XML query processing

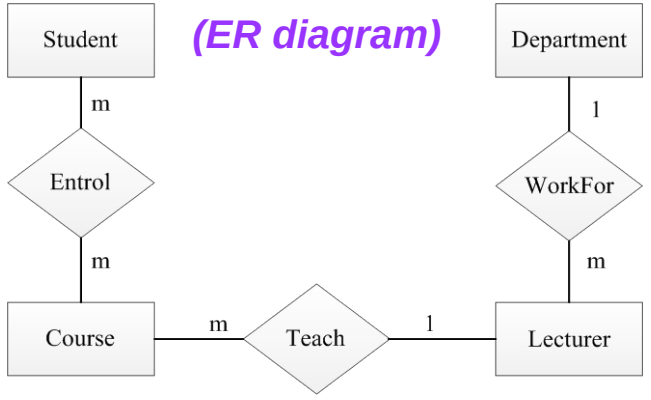


(Universit.xml)

Query: find grade that student John obtains in Java course

```
//Course[Title=Java][Student/Name=John]/Grade
```

(XPath)

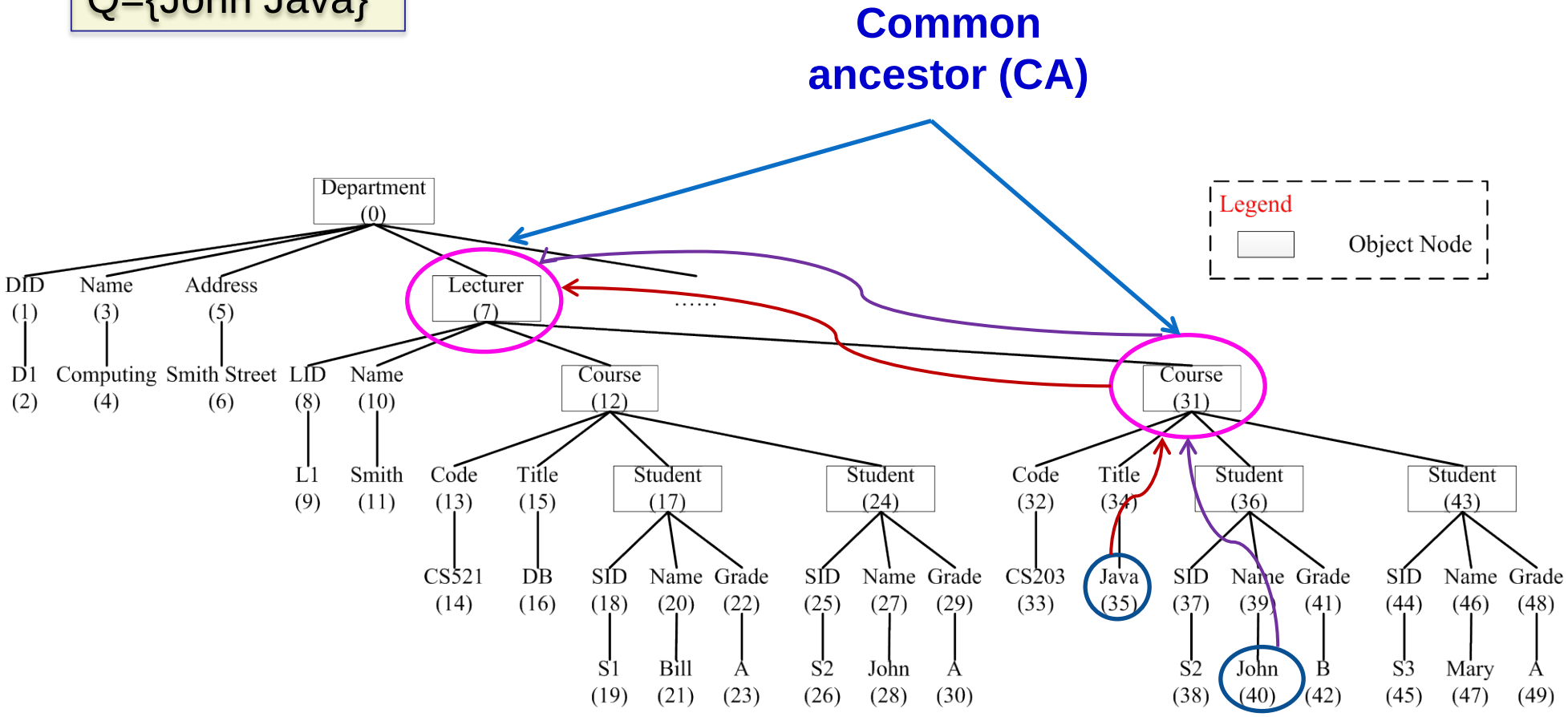


(ER diagram)

ORA-semantics in XML Keyword Search

– Current **XML keyword search** : **LCA approach**

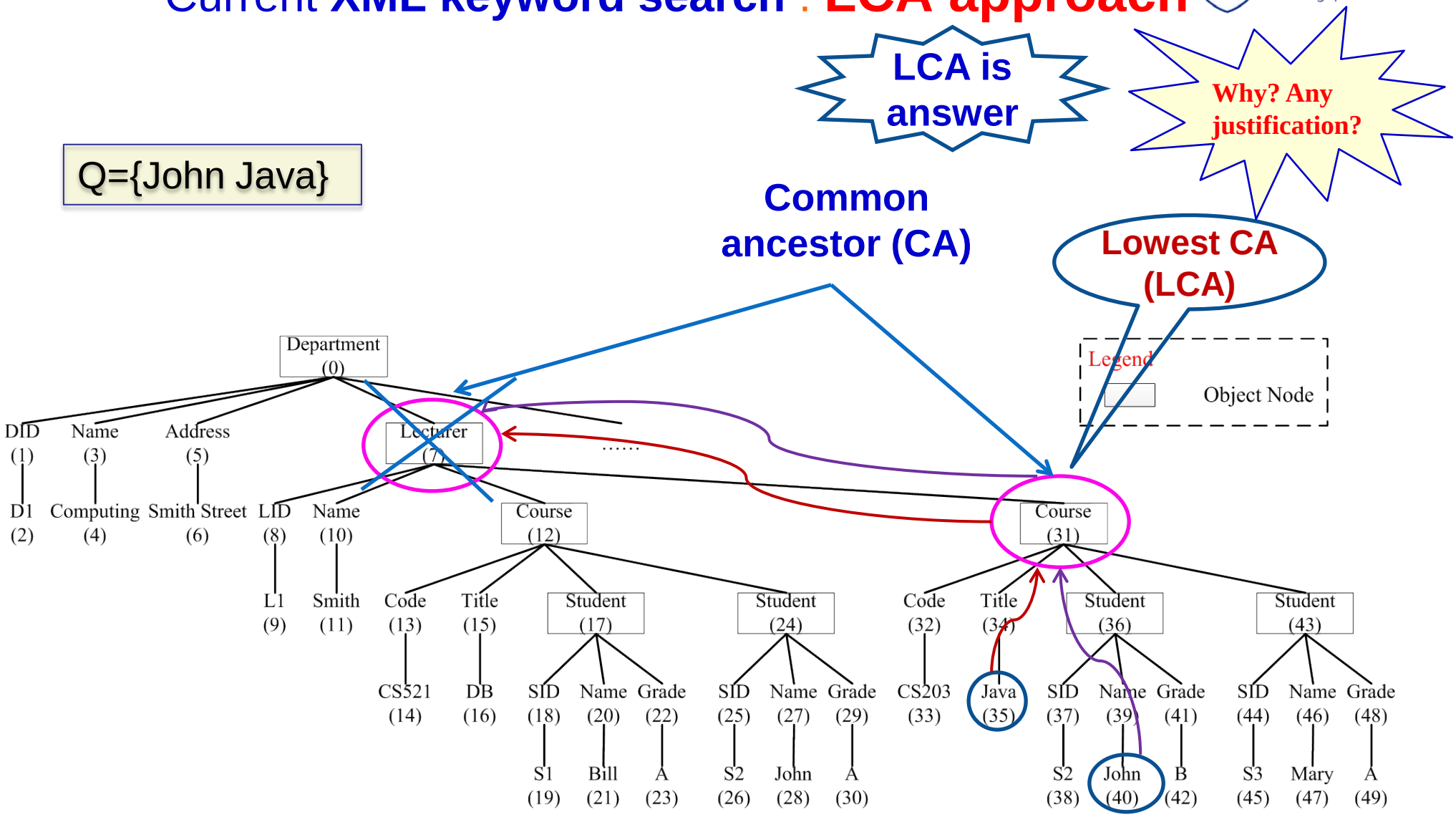
Q={John Java}



(Universit.xml)

ORA-semantics in XML Keyword Search

– Current **XML keyword search** : **LCA approach**



(Universit.xml)

– **Problems** of current XML keyword search

NUS National University of Singapore

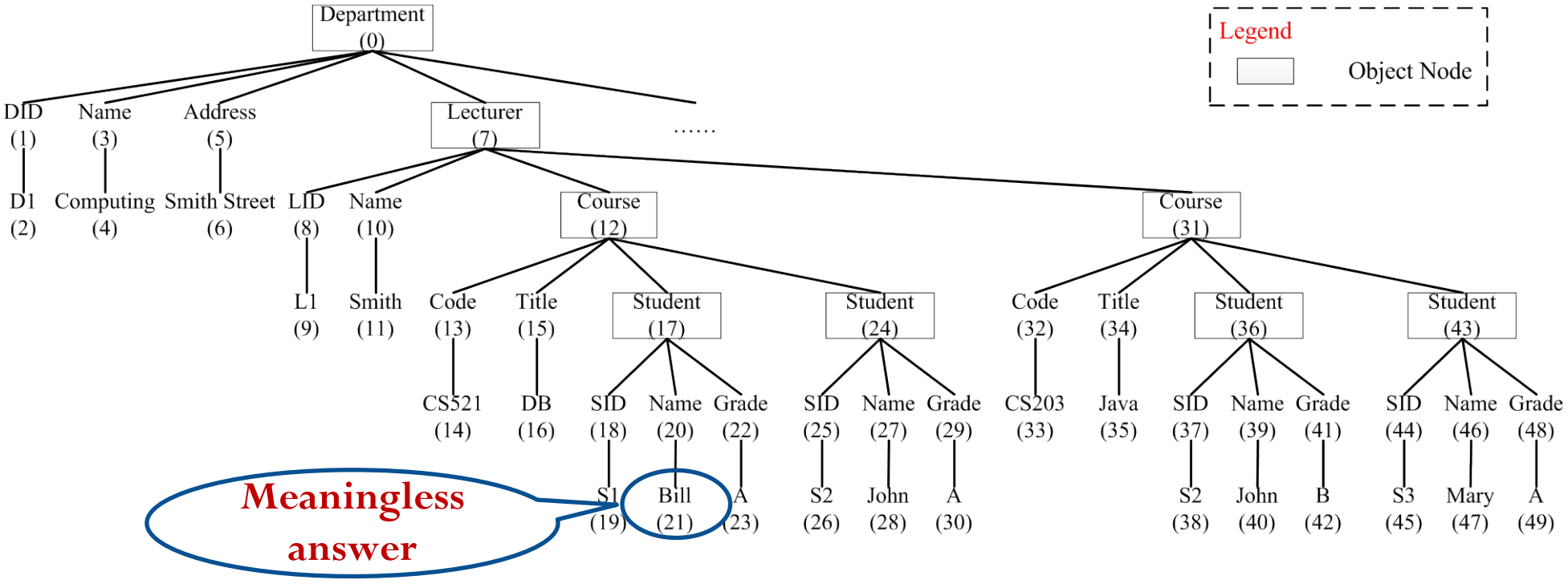
- LCA-based approach such as SLCA [13], ELCA [14], etc.
  - Rely only on the hierarchical structure of XML
  - Only consider LCA as possible answers
  - Do not consider ORA-semantics

- Problems:
  1) Meaningless answer
  2) Missing answer
  3) Duplicated answer
  4) Problems related to relationships
  5) Inconsistent types of answers
  6) Schema dependent answer

– Problems of current XML keyword search

1) Meaningless answer

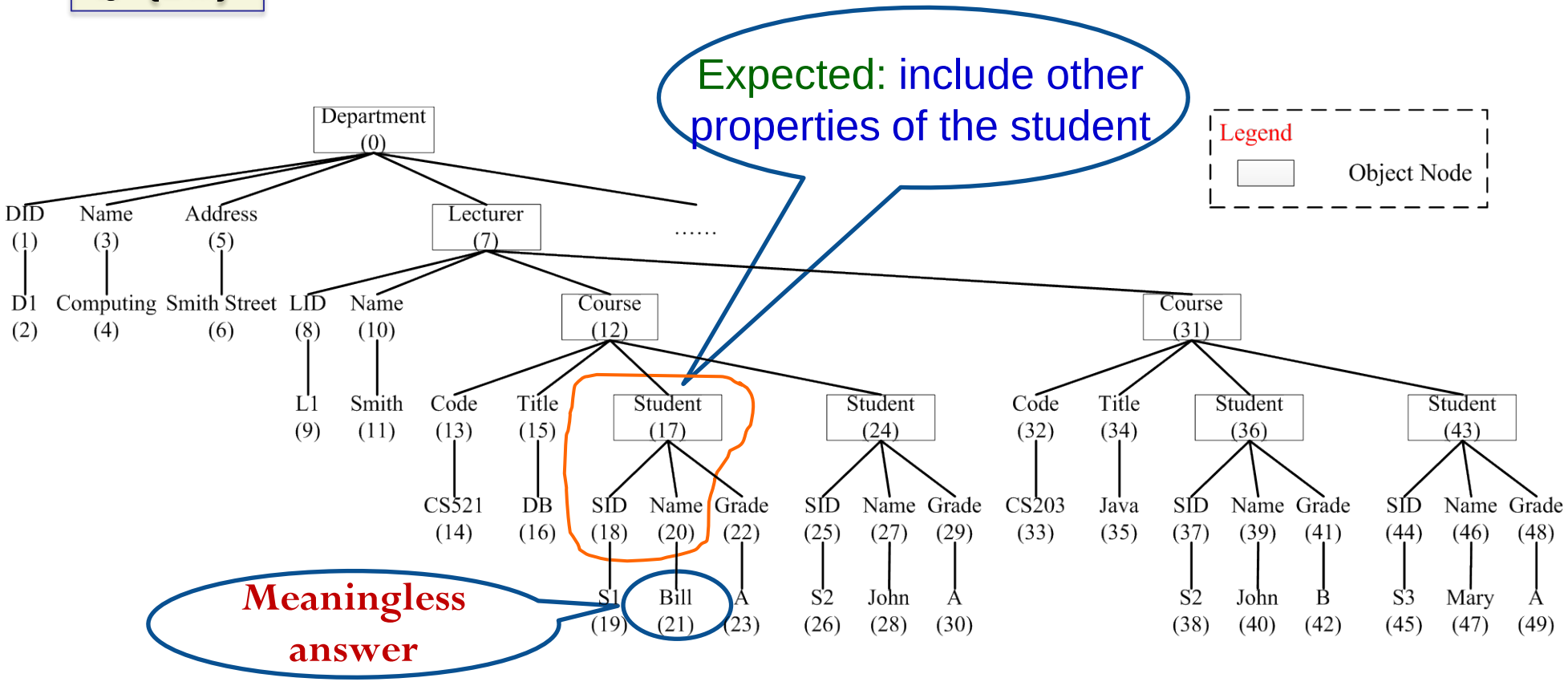
Q={Bill}



– Problems of current XML keyword search

1) Meaningless answer

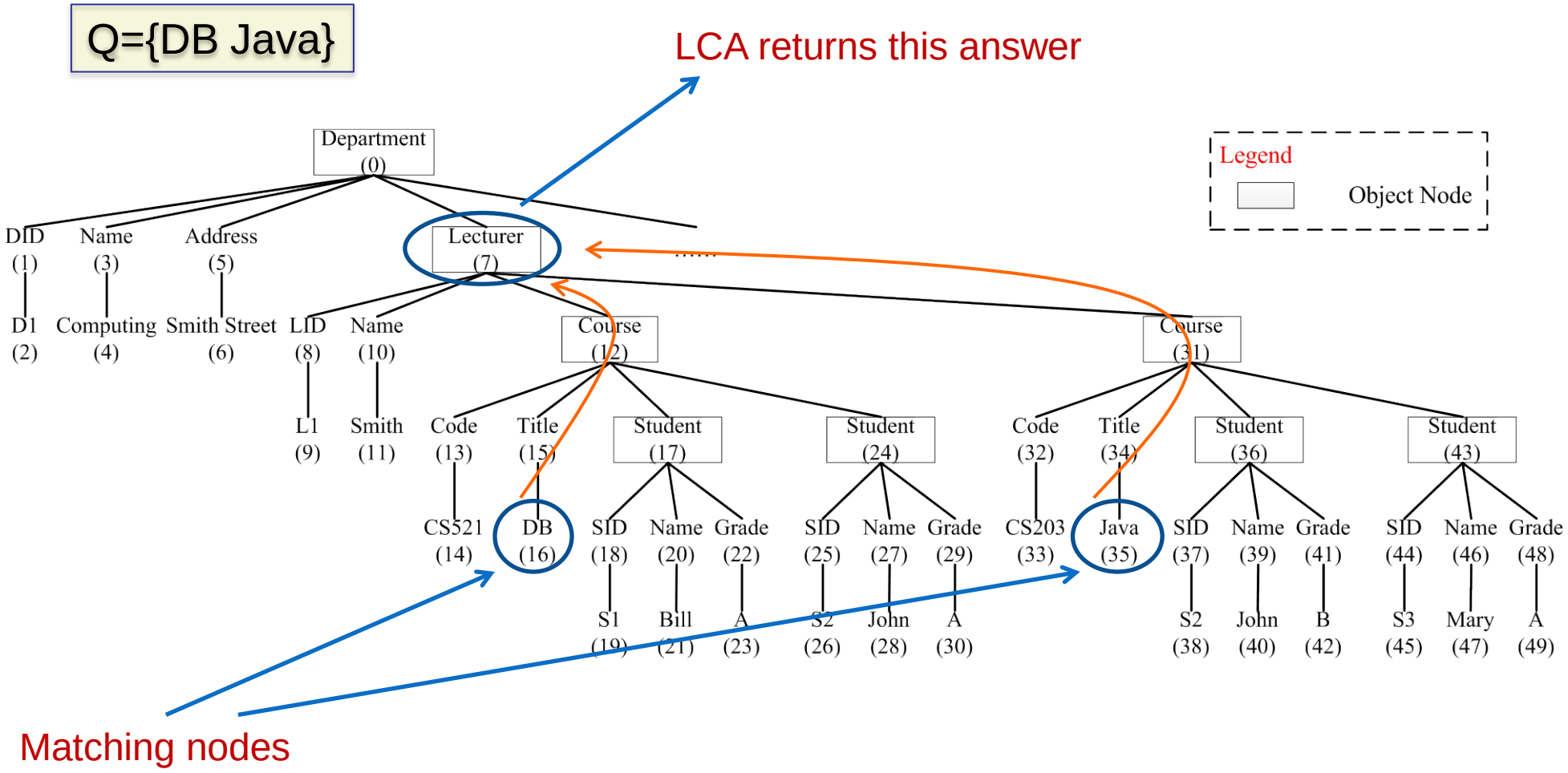
Q={Bill}



Reasons: do not have concept of object → cannot distinguish object node vs. non-object node

– Problems of current XML keyword search

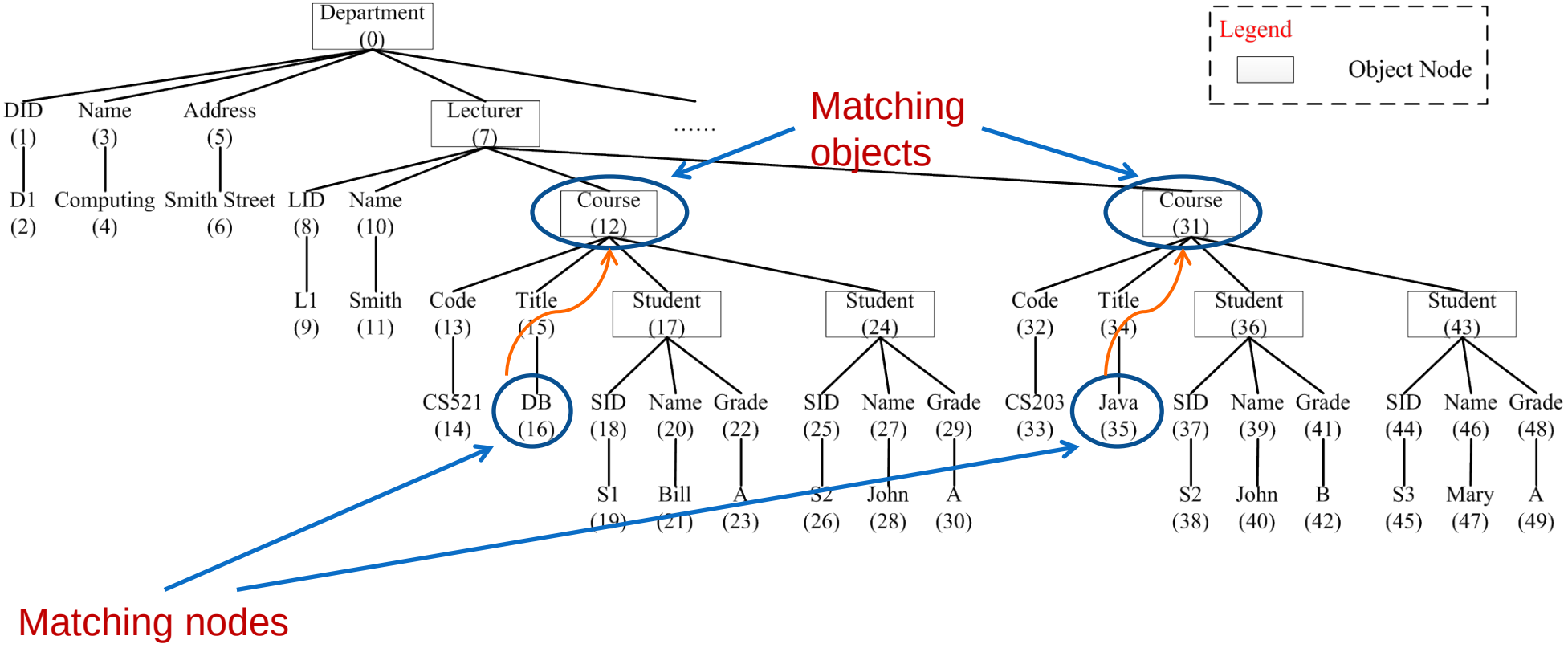
2) Missing answer



– Problems of current XML keyword search

2) Missing answer

Q={DB Java}

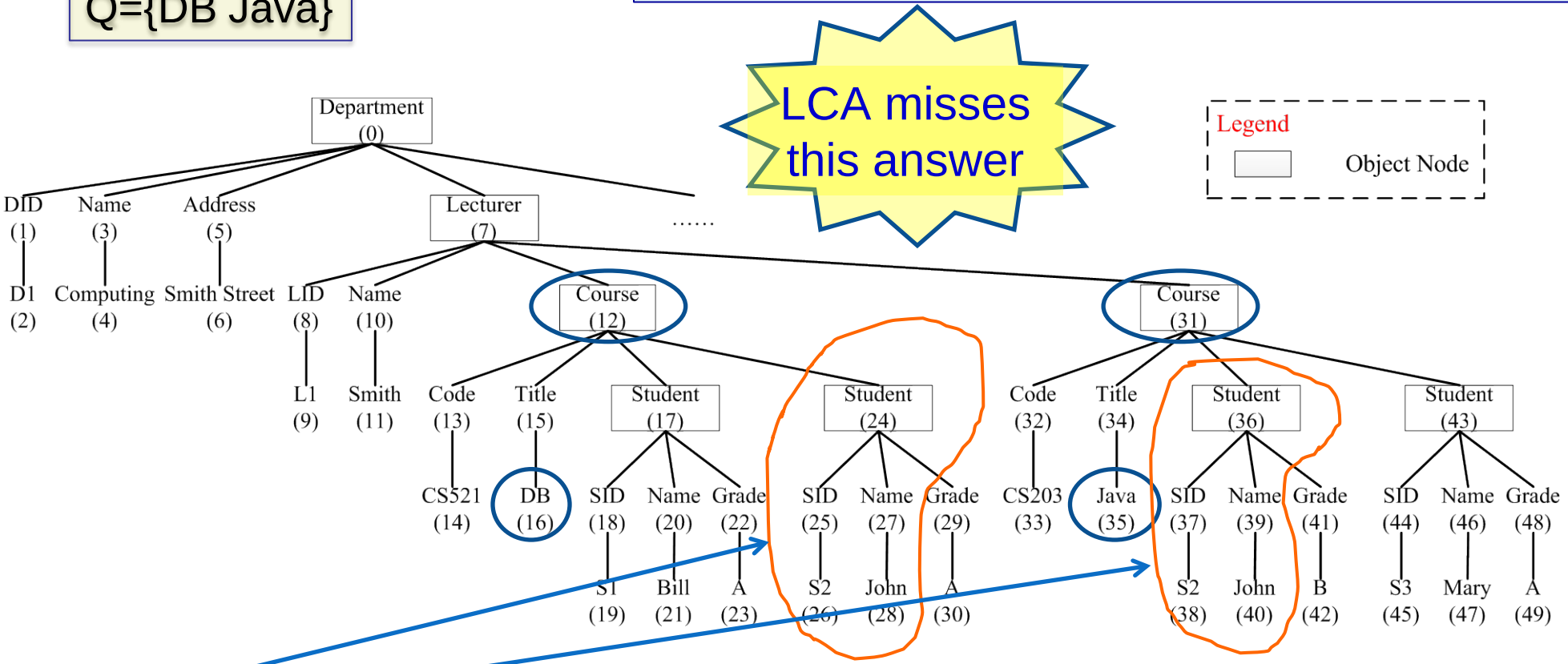


– Problems of current XML keyword search

2) Missing answer

Reasons:
(1) do not have the concepts of object & OID, so do not discover object duplication
(2) also need to search for common descendants

Q={DB Java}



Identical subtree → The same student → Student takes 2 courses → Should be returned: **common descendant** of 2 courses

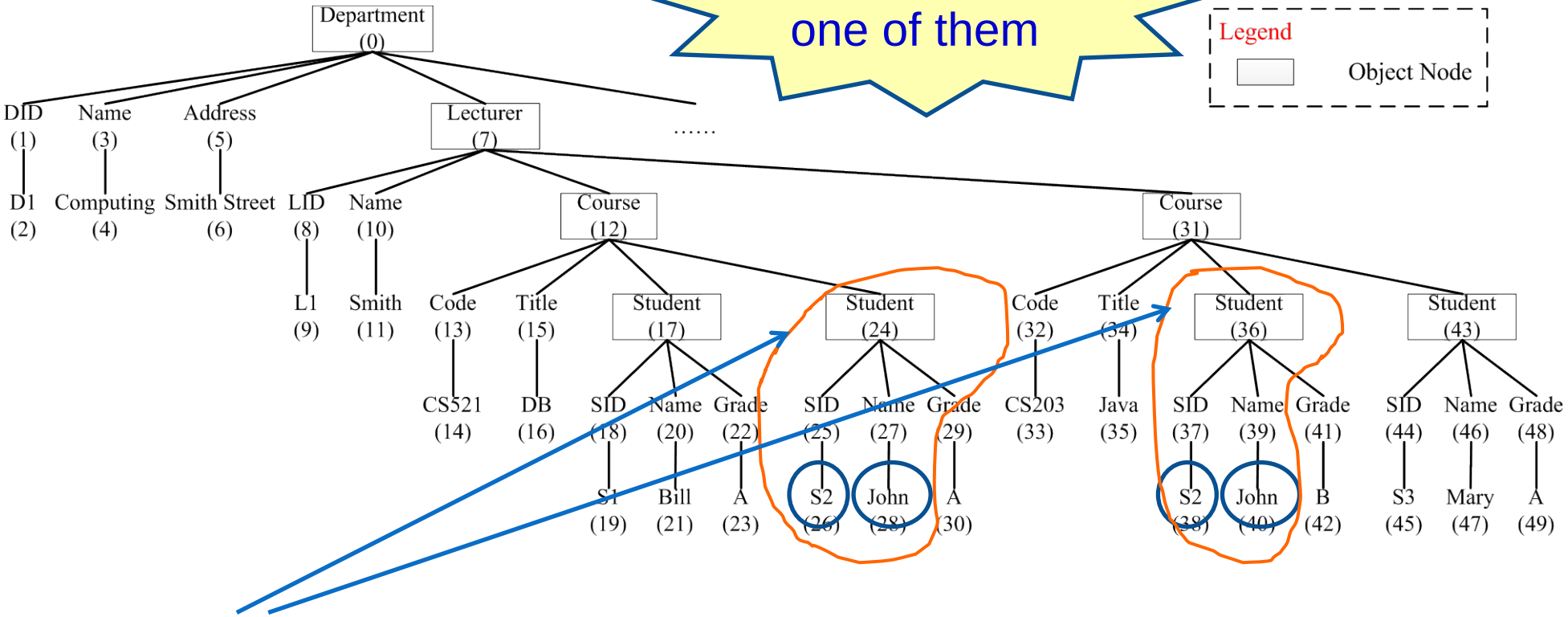
– Problems of current XML keyword search

3) Duplicated answer

Reasons: do not have concept of object, OID
→ do not discover object duplication

Q={S2 John}

Should return only one of them

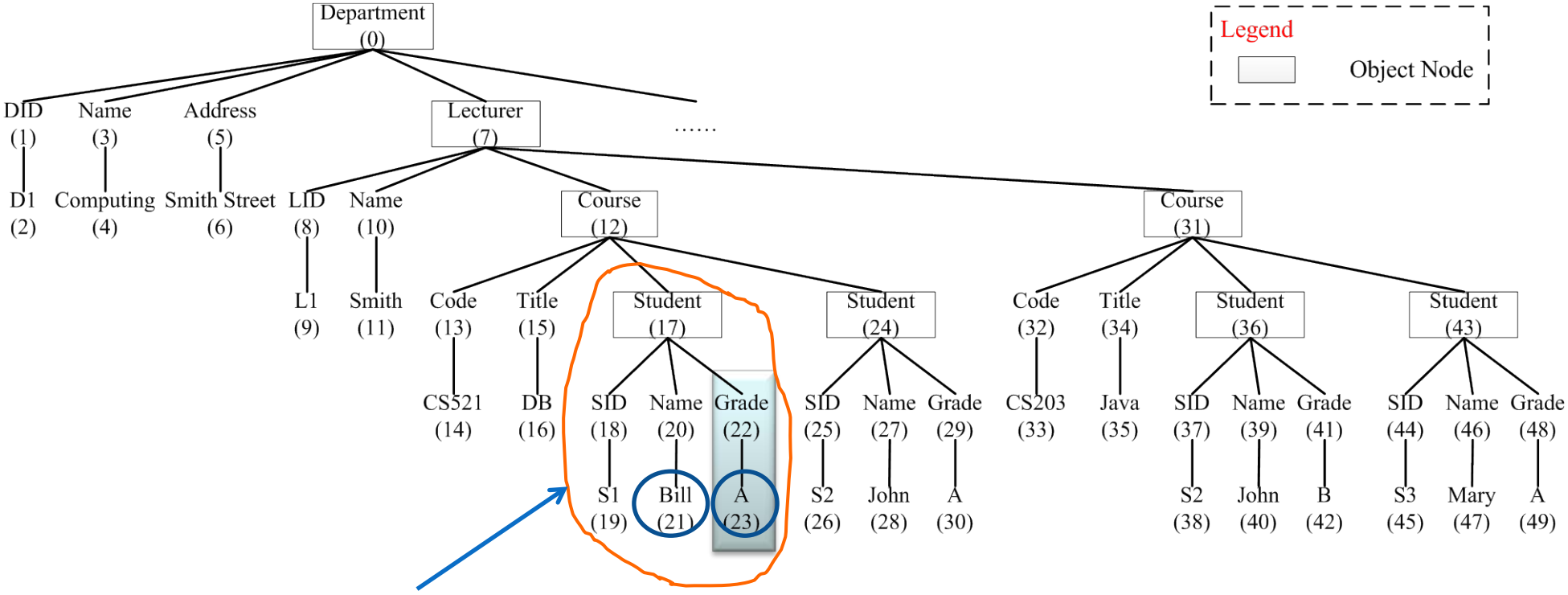


Identical subtrees → Duplicated answers

– Problems of current XML keyword search

4) Problems related to relationships

Q={Bill A}



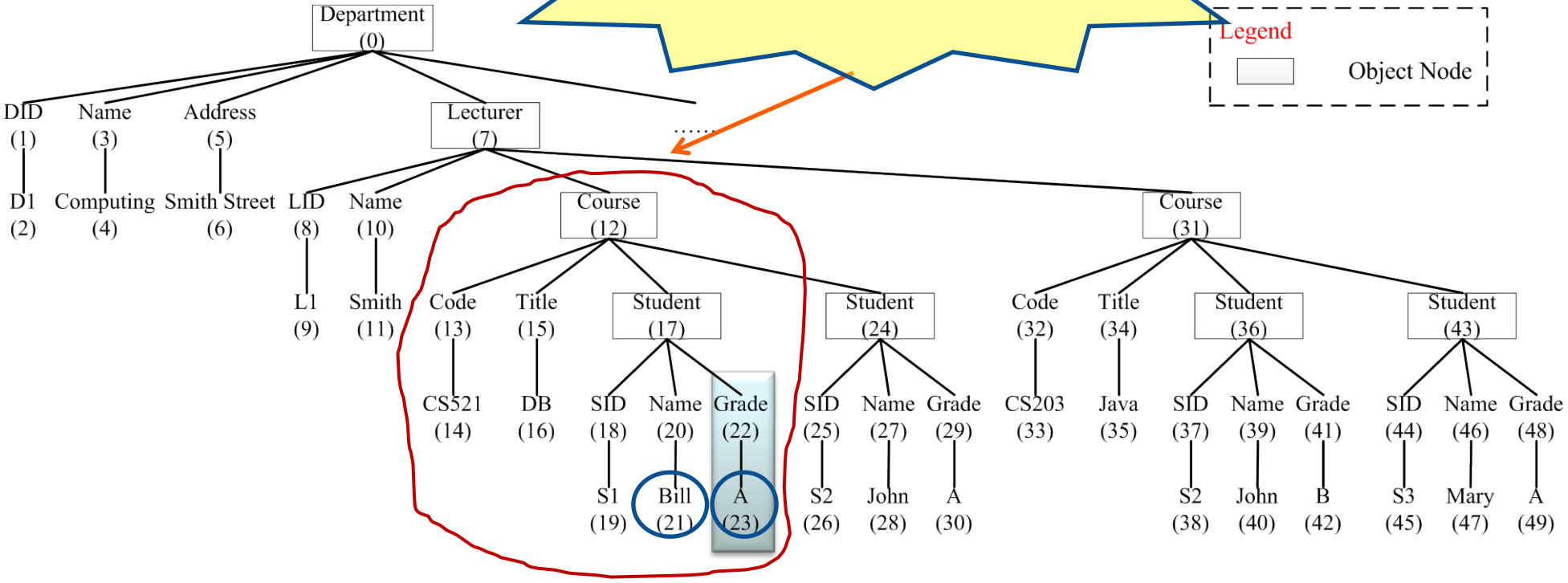
Grade is an attribute of the relationship between student and course, not an object attribute

– Problems of current XML keyword search

4) Problems related to relationships

Q={Bill A}

Include other object (course) involved in the relationship

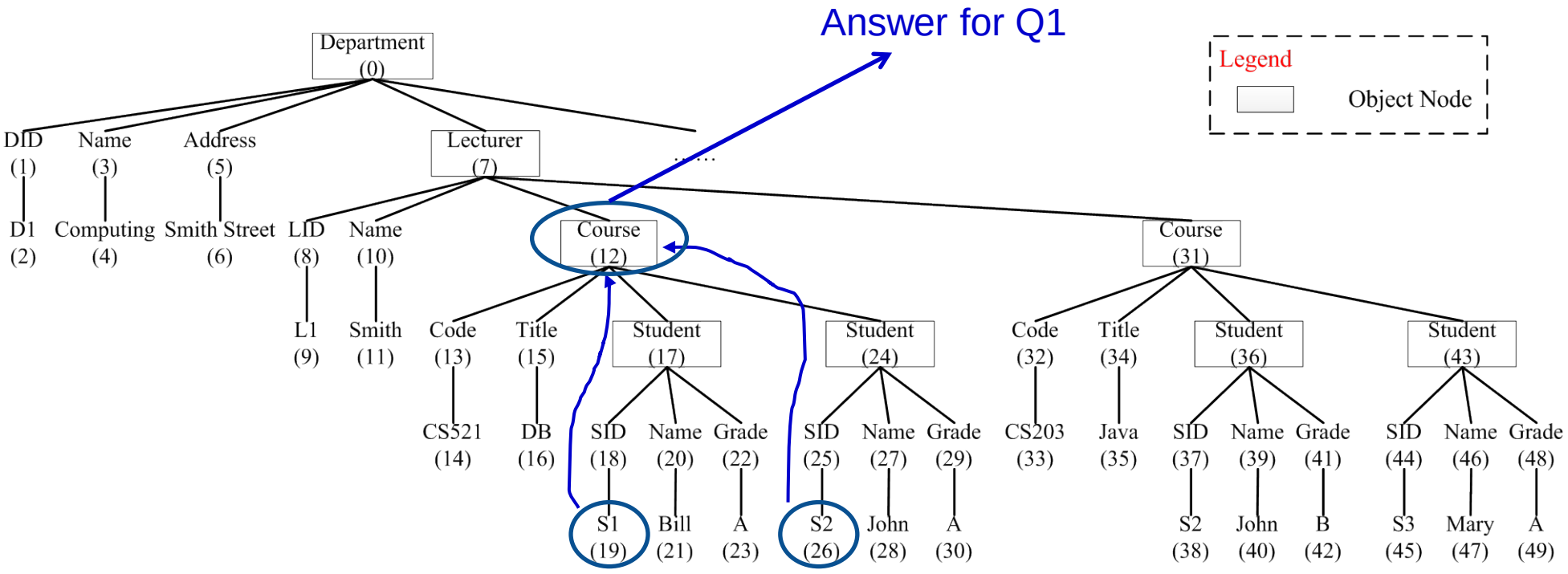


Reasons: do not have concept of relationship
→ cannot distinguish obj. attribute vs. rel. attribute

– Problems of current XML keyword search

5) Inconsistent types of answers

Q1 = {S1 S2}

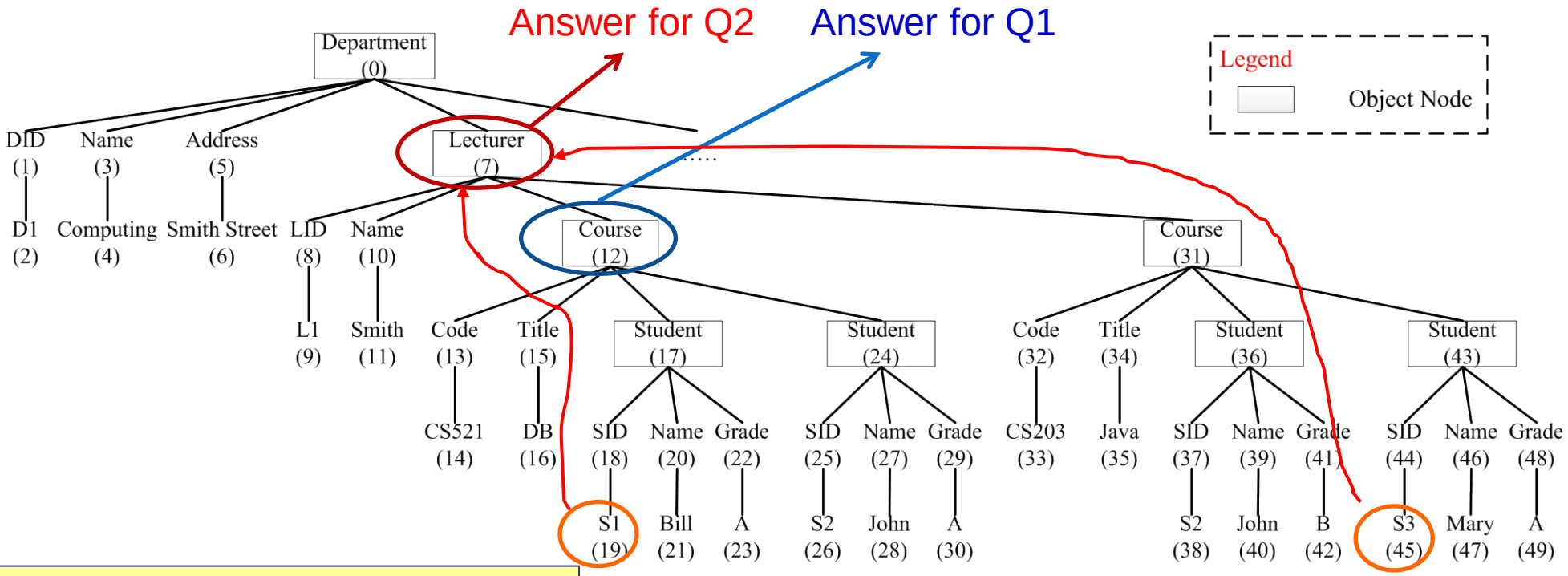


– Problems of current XML keyword search

5) Inconsistent types of answers

Q1 = {S1 S2}

Q2 = {S1 S3}



Two similar queries have very different answers and user will get confused

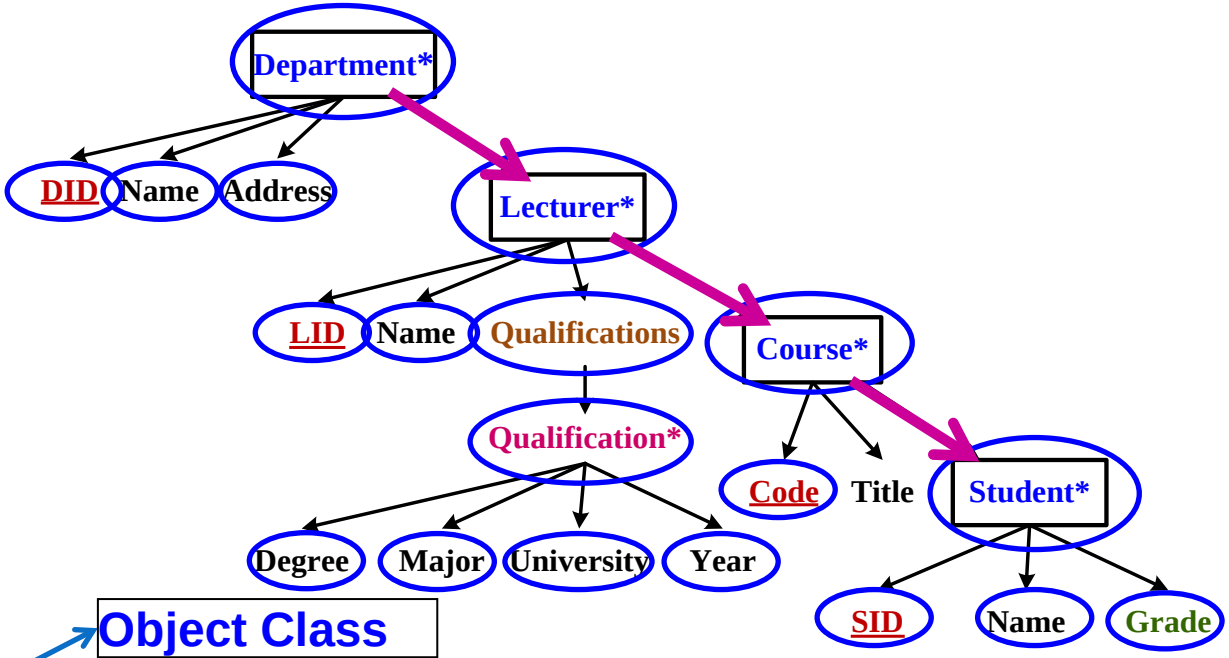
Reasons:
(1) do not have the concepts of object & relationship
(2) rely on hierarchical structure of XML data

– Problems of current XML keyword search

6) Schema dependent answer

- Will discuss it later.

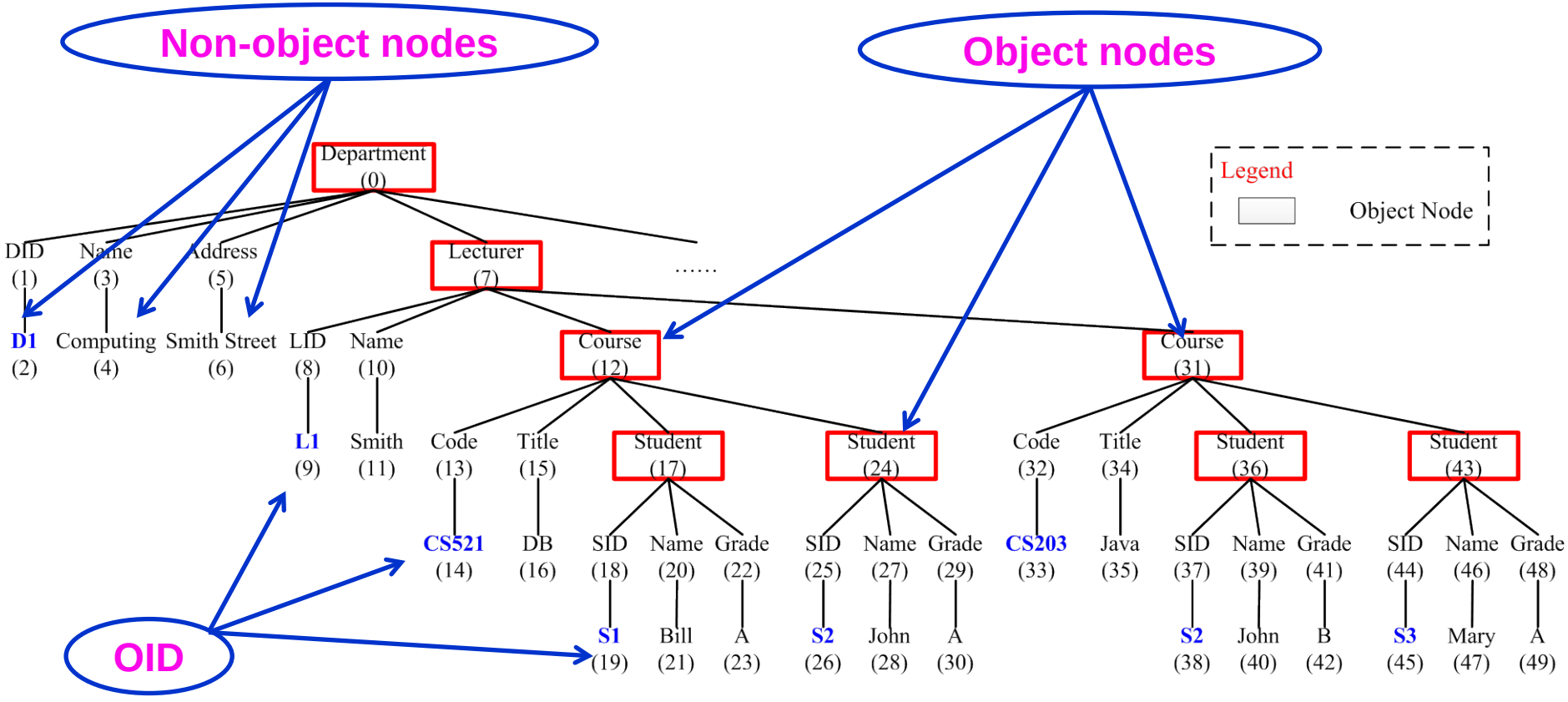
– Discovery of ORA semantics in XML[15]



XML schema

- Nodes
  - Internal Node
    - Object Class
    - Aggregation Node
    - Composite Attribute
    - Explicit Relationship type
  - Leaf Node
    - Object ID
    - Object attribute
    - Relationship attribute
- Edges
  - Implicit relationship
  - Edge joined object class (the rest edges)

– Object nodes vs. non-object nodes



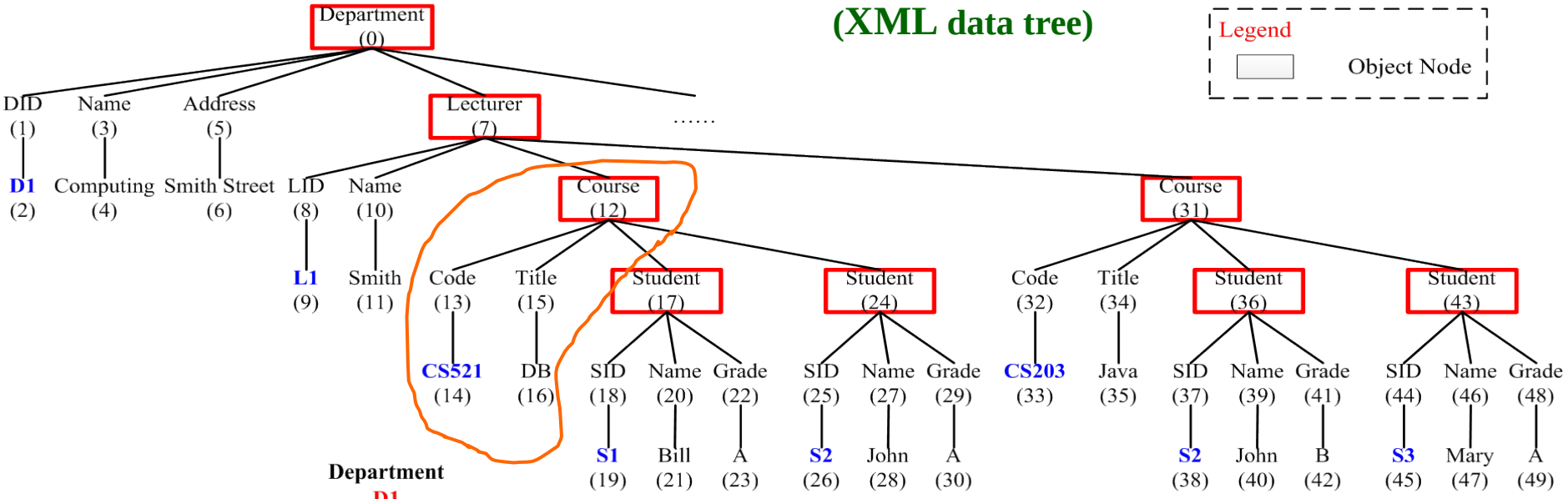
(XML data tree)

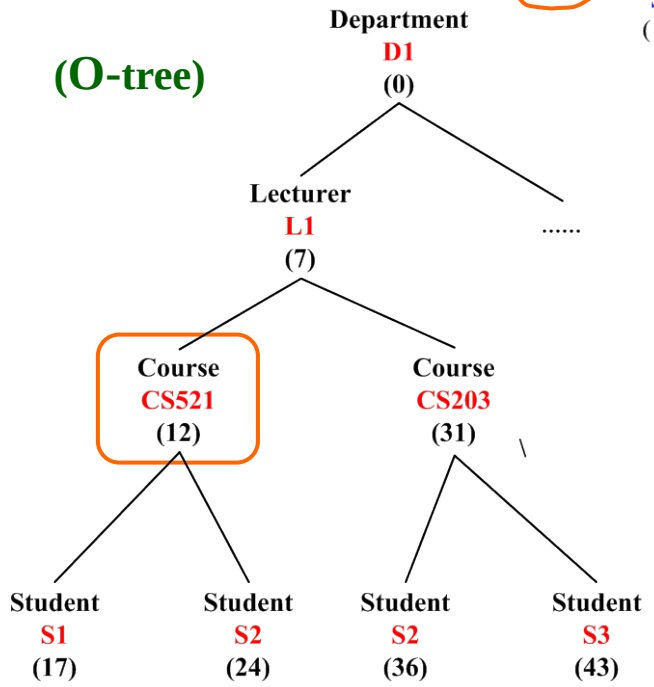
– XML Object Tree (O-tree)

- An O-tree is a tree extracted from an XML data tree
  - keeping only object nodes
    - Objects (and relationships) are what users want to find
    - Attribute value along without knowing its object/relationship is not very meaningful to user
  - associating non-object nodes to the corresponding object nodes

❖ Largely reduce size of XML data tree

– O-tree (Example)





ORA-semantics in XML Keyword Search

NUS National University of Singapore

Topics to be discussed

- Search over **O-tree** [16]
  - Find lowest common object ancestors (LOCAs) to avoid returning meaningless answers and duplicated answers
  - Search for highest common object descendants (HCODs) to avoid missing answers
- Search for **common relatives** (CRs) to perform a schema independent keyword search [17]
- Answer **aggregate functions** in keyword queries on XML [18]
  - Detect duplicate objects and relationships in order to compute aggregates correctly

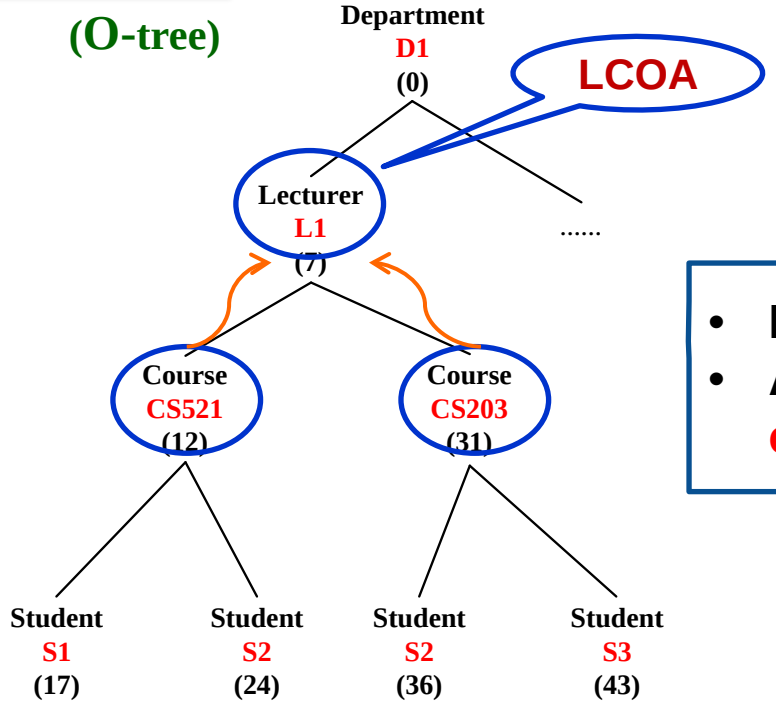
ORA-semantics in XML Keyword Search

- Search over **O-tree**
  - **LCOA (Lowest common object ancestor)**
    - similar to **existing LCA based approaches**, but
      - avoid returning meaningless answers and duplicated answers
  - **HCOA (Highest common object descendant)**
    - **more answers (**but **not all missing answers)**

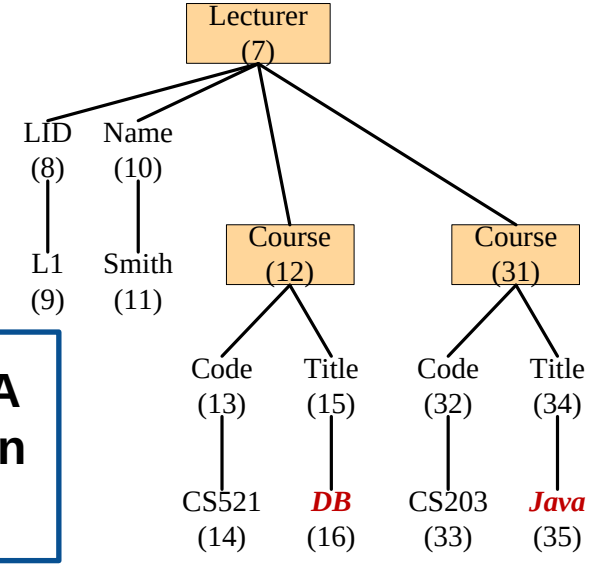
- Search over **O-tree**
  - **LCOA (Lowest common object ancestor)**
    - o similar to **existing LCA based approaches**, but
    - – avoid returning **meaningless answers** and **duplicated answers**

Q={DB Java}

(O-tree)



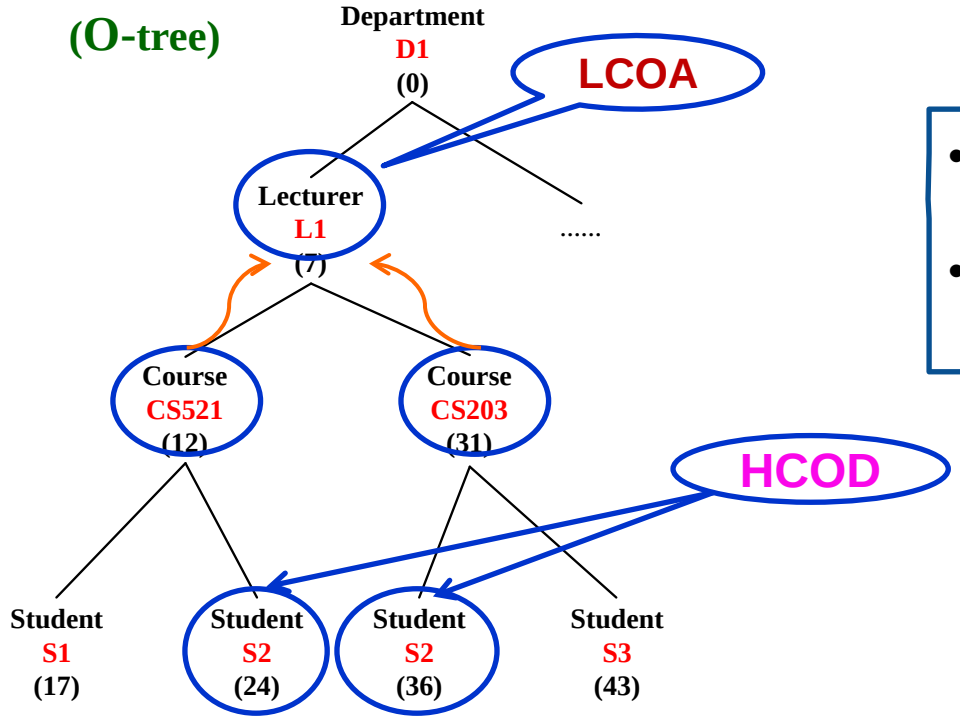
- **LCOA similar to LCA**
- **An LCOA must be an object node**



(XML fragment)

- Search over **O-tree**
  - **HCOA (Highest common object descendant)**
    - more answers (but not all missing answers)

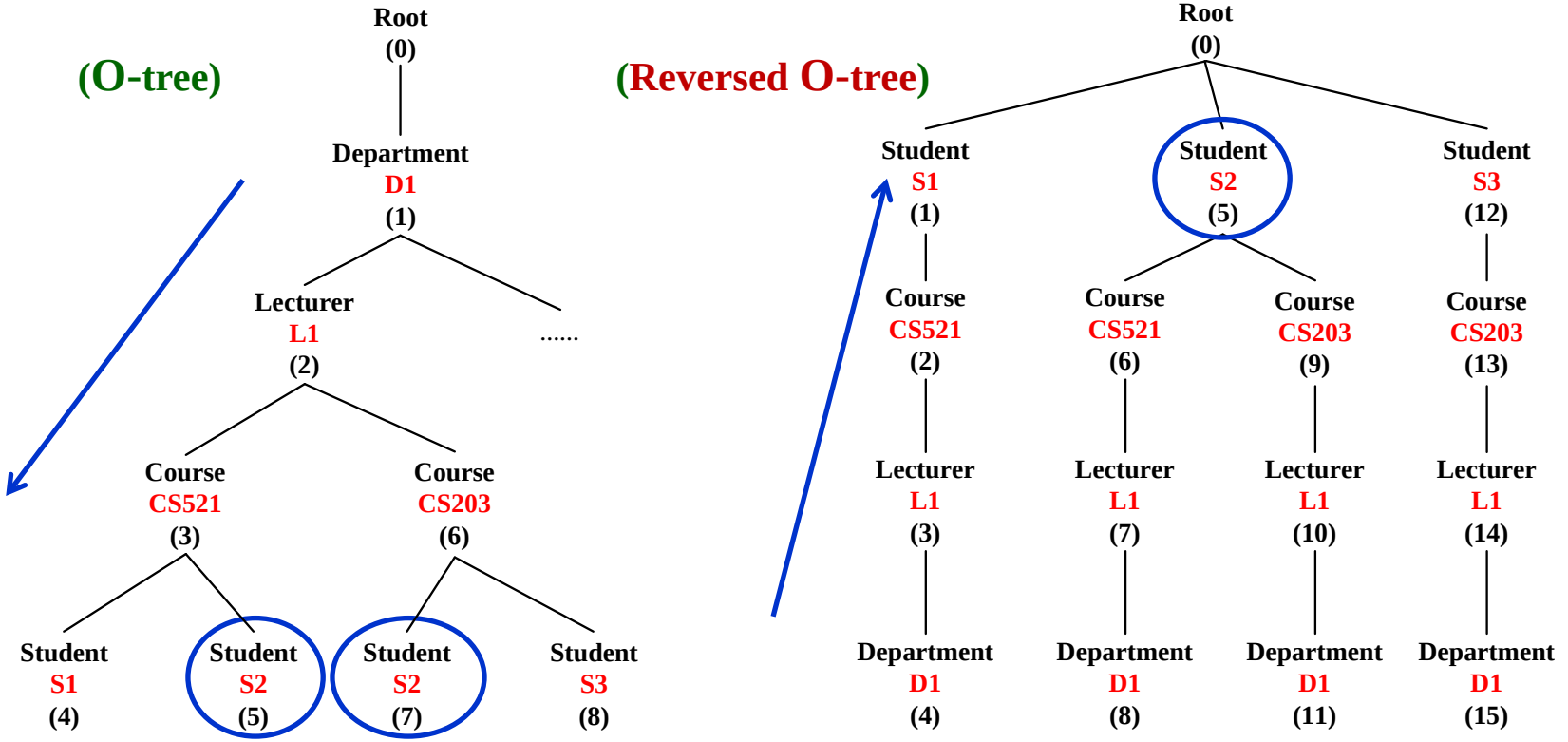
Q={DB Java}



- An **HCOD** is a **set** of object nodes referring to the **same object**
- Each object node is an **descendant** of an matching object node

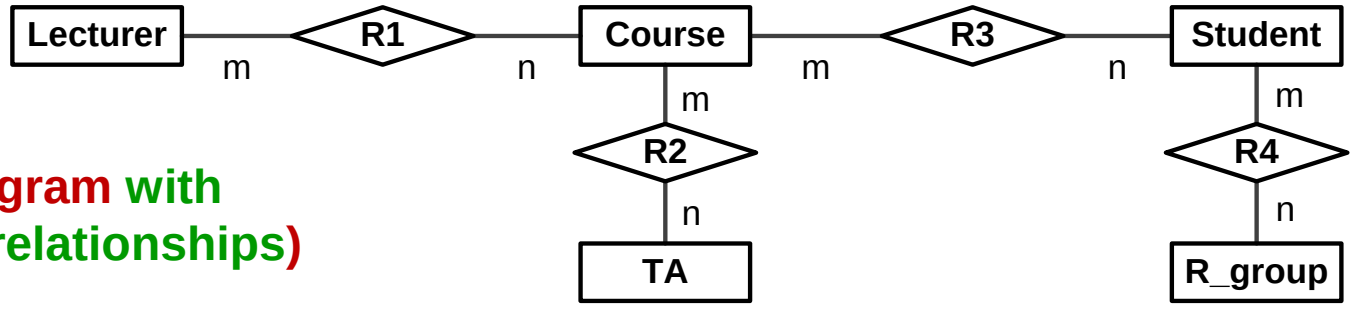
common ancestors of reversed O-tree are equivalent to common descendants of original O-tree

- Search over **O-tree**
  - Use **reserved O-tree** to find **HCOD**
    - Each path from root to leaf is **reversed**
    - Object nodes: refer to same object and have same ancestors → **merged**
    - **Relationship attributes** go with lower object



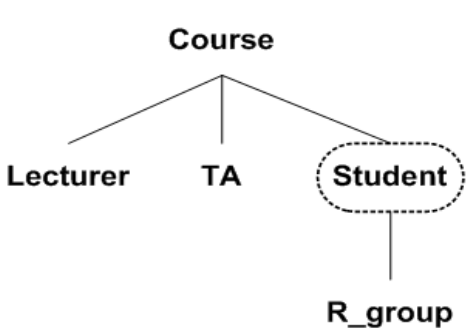
❑ Schema independent XML keyword search

➢ Motivation

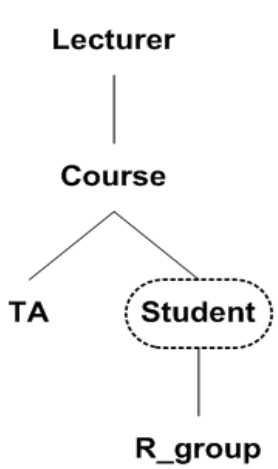


(ER diagram with binary relationships)

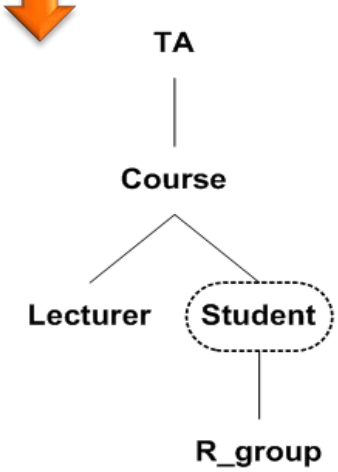
Many ways to represent the database in XML



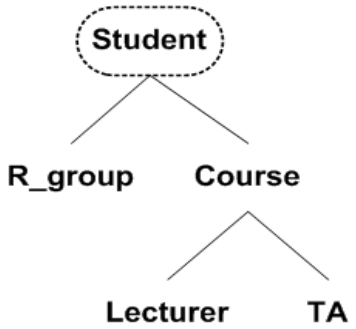
(Schema tree 1)



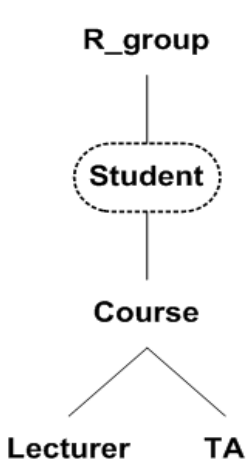
(Schema tree 2)



(Schema tree 3)



(Schema tree 4)

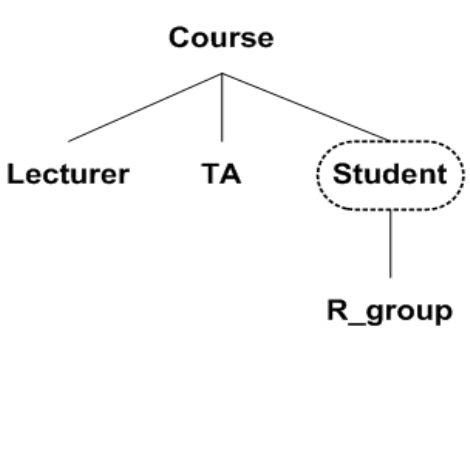


(Schema tree 5)

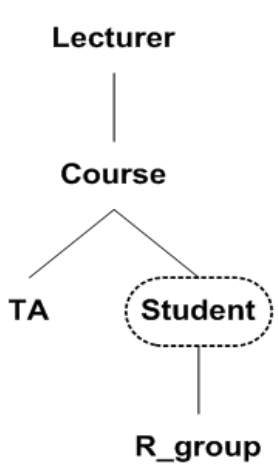
(Five Reasonable XML schema trees)

❑ Schema independent XML keyword search

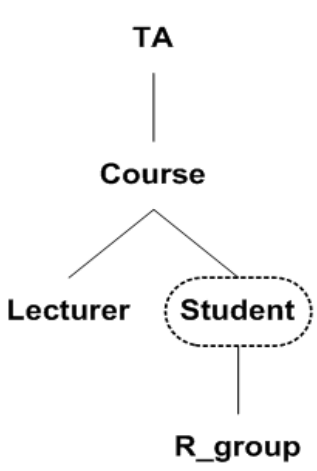
- Motivation
  - Users may know database is about courses, lecturers, TAs, students, research group (R_group)
  - But they may not know (and not necessary need to know) what schema looks like (and which schema?)



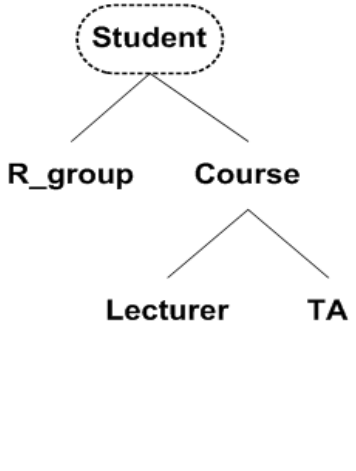
(Schema tree 1)



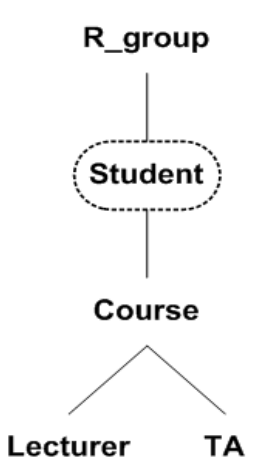
(Schema tree 2)



(Schema tree 3)



(Schema tree 4)



(Schema tree 5)

(Five Reasonable XML schema trees)

❑ Schema independent XML keyword search

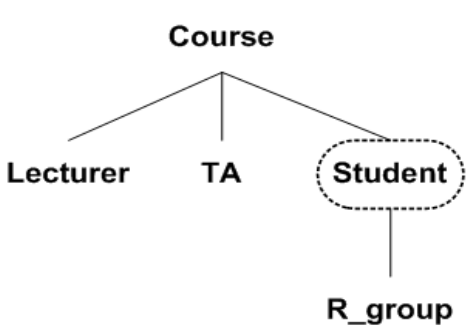
➢ Motivation

Q = {studentA studentB}

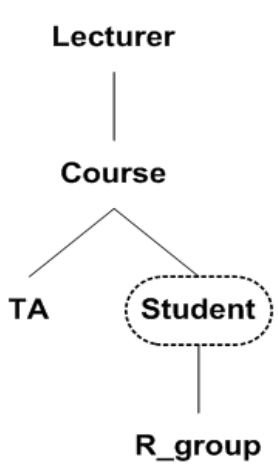
Expected answers

Ans1. Common courses
Ans2. Common R_groups
Ans3. Common lecturers
Ans4. Common TAs

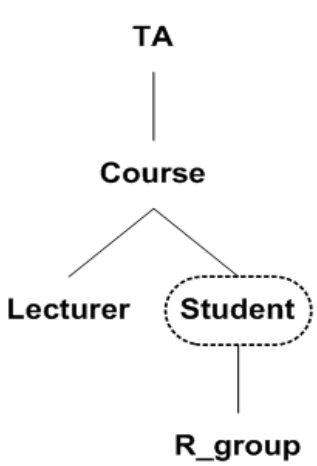
Common ancestors in some schema(s)



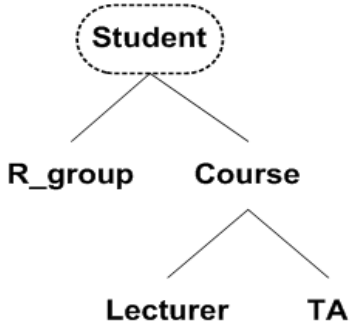
(Schema tree 1)



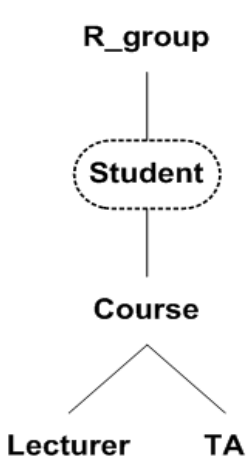
(Schema tree 2)



(Schema tree 3)



(Schema tree 4)



(Schema tree 5)

(Five Reasonable XML schema trees)

Schema independent XML keyword search

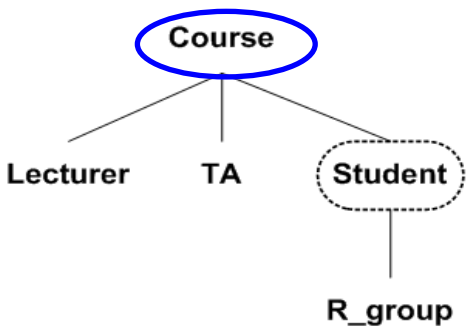
- Motivation

Q = {studentA studentB}

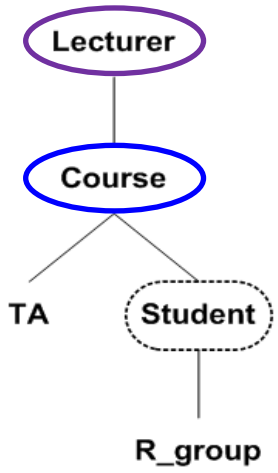
Expected answers

- **Ans1. Common courses** —LCA→ **Schema1, Schema2, Schema3**
- **Ans2. Common R_groups** —LCA→ **Schema5**
- **Ans3. Common lecturers** —LCA→ **Schema2**
- **Ans4. Common TAs** —LCA→ **Schema3**

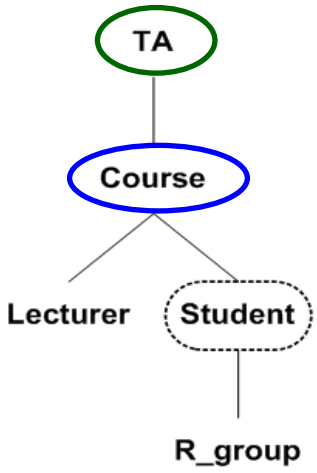
All meaningful answers



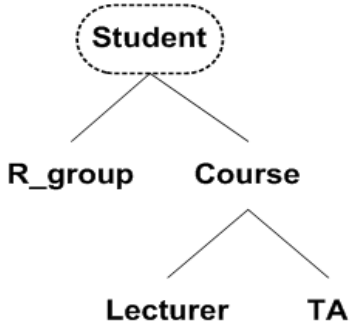
(Schema tree 1)



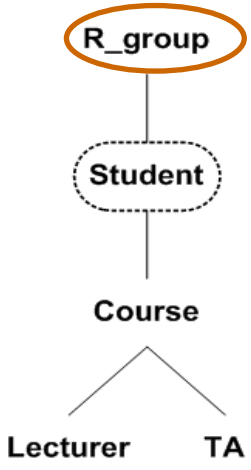
(Schema tree 2)



(Schema tree 3)



(Schema tree 4)



(Schema tree 5)

(Five Reasonable XML schema trees)

❑ Schema independent XML keyword search

- Motivation

Q = {studentA studentB}

Five different sets of answers:

Schema 1: **Ans1 (course)**

Schema 2: **Ans1** & **Ans3 (lecturer)**

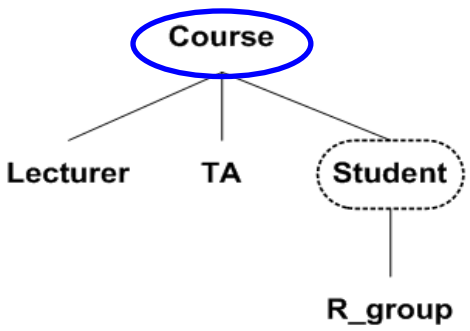
Schema 3: **Ans1** & **Ans4 (TA)**

Schema 4: **no answer**

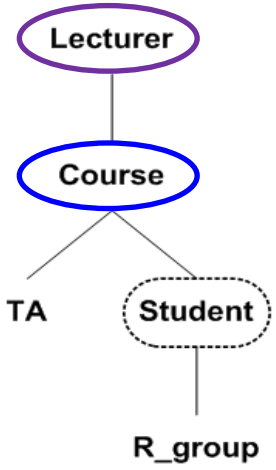
Schema 5: **Ans2 (R_group)**

Different answer sets

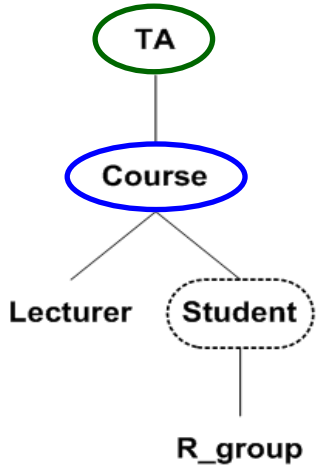
No schema provides all 4 answers



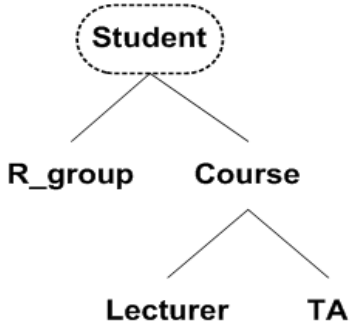
(Schema tree 1)



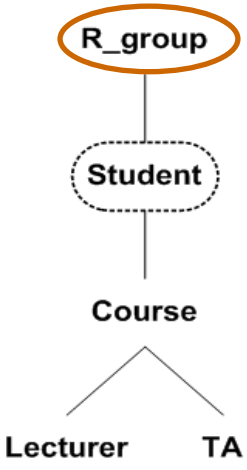
(Schema tree 2)



(Schema tree 3)



(Schema tree 4)



(Schema tree 5)

(Five Reasonable XML schema trees)

NUS National University of Singapore

❑ **Schema independent** XML keyword search

- ➢ Motivation
  - ❖ Different users may have different expectations
  - ❖ However, expectations of a user should be independent from schema designs because user does not know which schema is used
  - ❖ However, all five different schema designs provide five different sets of answers by LCA semantics

❑ Schema independent XML keyword search

➢ Intuition of our Common Relative (CR) semantics

Q = {studentA studentB}

Expected answers

- **Ans1. Common courses**
- **Ans2. Common R_groups**
- **Ans3. Common lecturers**
- **Ans4. Common TAs**

- **Schema 1: Ans1 (course)**
- **Schema 2: Ans1 & Ans3 (lecturer)**
- **Schema 3: Ans1 & Ans4 (TA)**
- **Schema 4: no answer**
- **Schema 5: Ans2 (R_group)**

How to find all types of answers with any one particular schema?

❑ Schema independent XML keyword search

➢ Intuition of our **Common Relative** (CR) semantics

Q = {studentA studentB}

Expected answers:

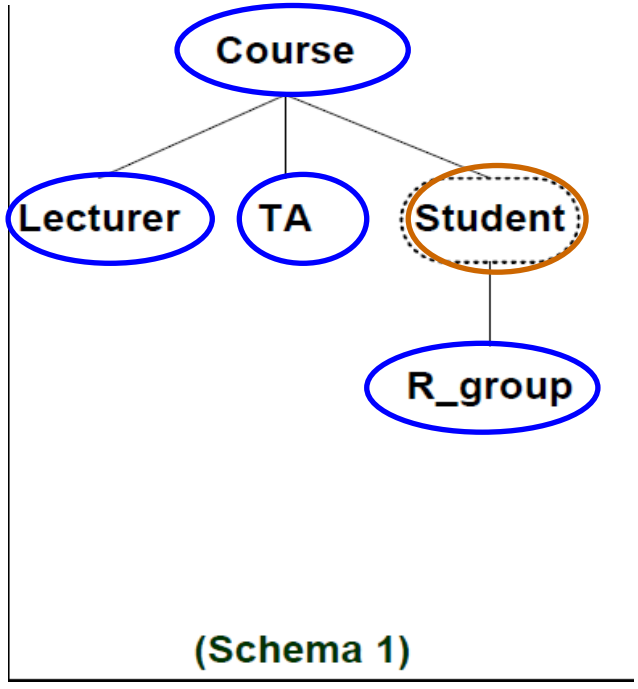
Ans1. Common courses —LCA→ **YES**

Ans2. Common R_groups —LCA→ **NO**

Ans3. Common lecturers —LCA→ **NO**

Ans4. Common TAs —LCA→ **NO**

How to find Ans2 with Schema1?

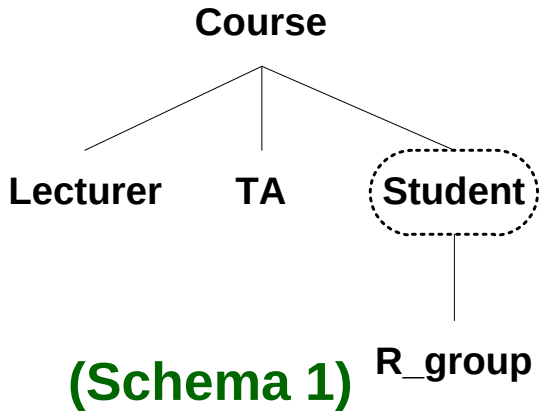


❑ Schema independent XML keyword search

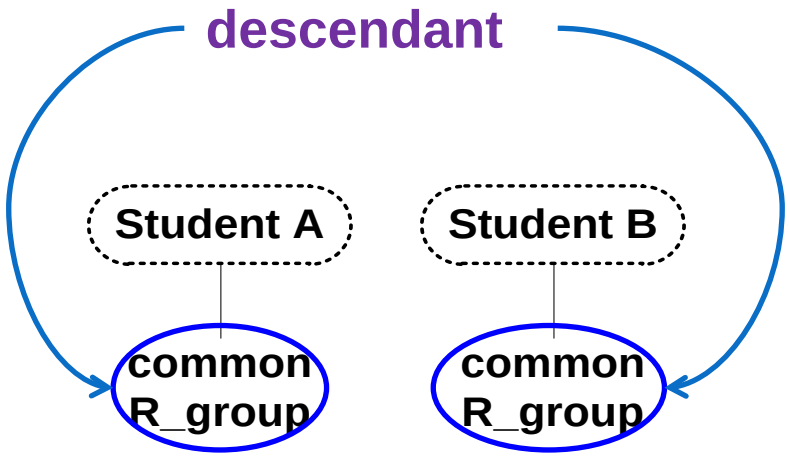
➢ Intuition of our Common Relative (CR) semantics

Ans2: Common R_groups

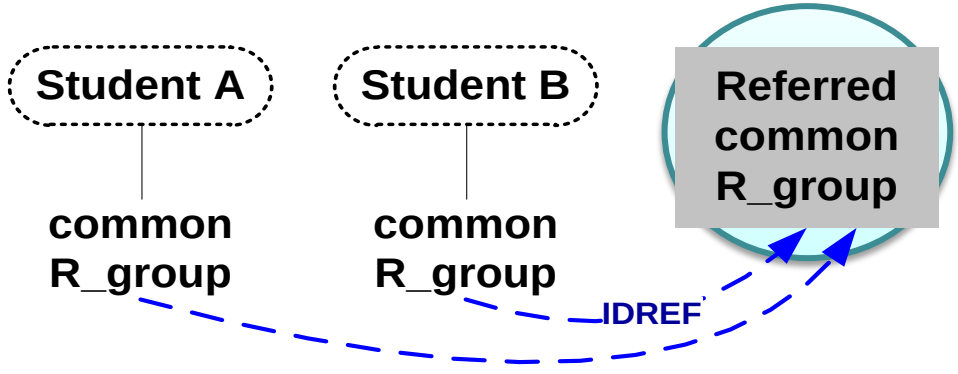
Q = {studentA studentB}



(Schema 1)



(a part of data w.r.t. Schema 1)



(a part of data with IDREFs w.r.t. Schema 1)

- Schema independent XML keyword search
  - Intuition of our Common Relative (CR) semantics

Q = {studentA studentB}

Expected answers:

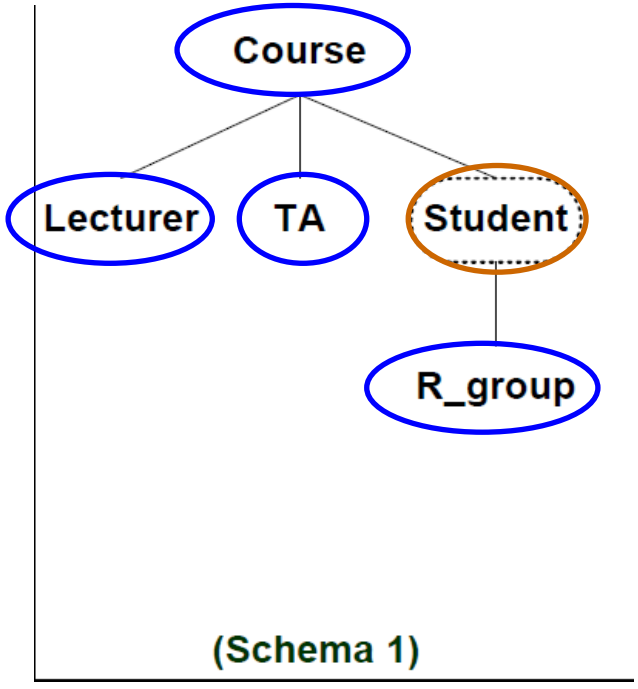
Ans1. Common courses —LCA→ **YES**

Ans2. Common R_groups —LCA→ **NO**

Ans3. Common lecturers —LCA→ **NO**

Ans4. Common TAs —LCA→ **NO**

How to find Ans3 with Schema1?

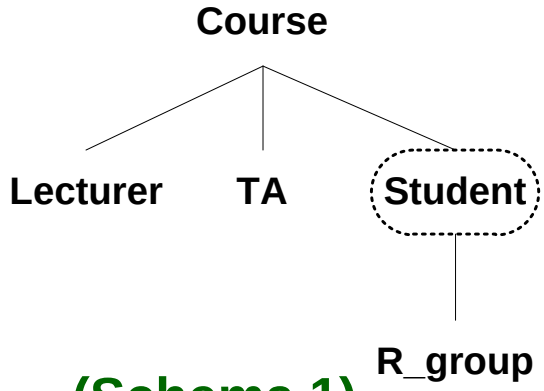


❑ Schema independent XML keyword search

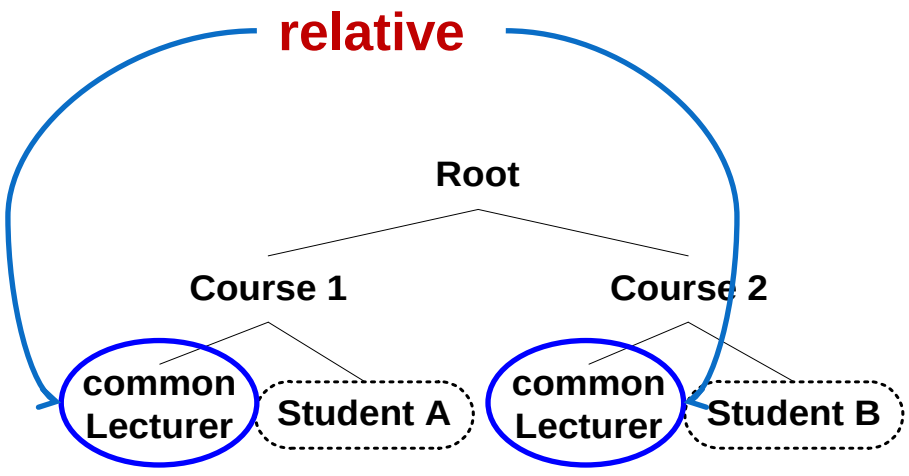
➢ Intuition of our Common Relative (CR) semantics

Ans3: Common lecturers

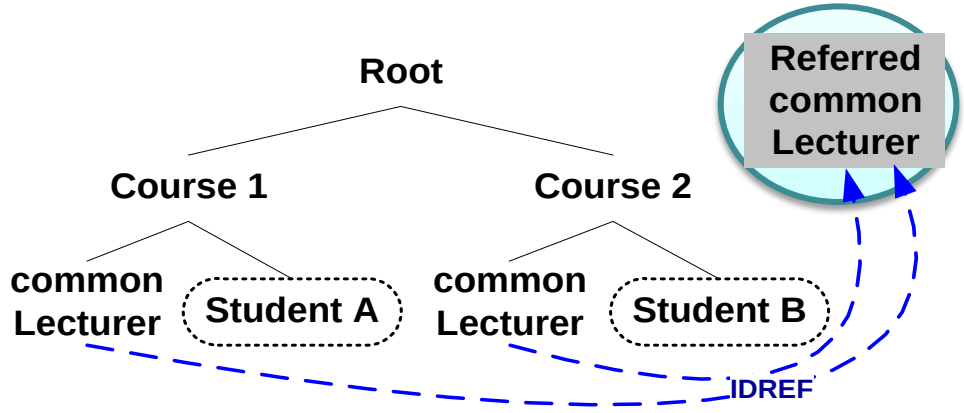
Q = {studentA studentB}



(Schema 1)



(a part of data w.r.t. Schema 1)



(a part of data with IDREFs w.r.t. Schema 1)

NUS National University of Singapore

❑ Schema independent XML keyword search

➢ Intuition of our Common Relative (CR) semantics

Q = {studentA studentB}

Expected answers:

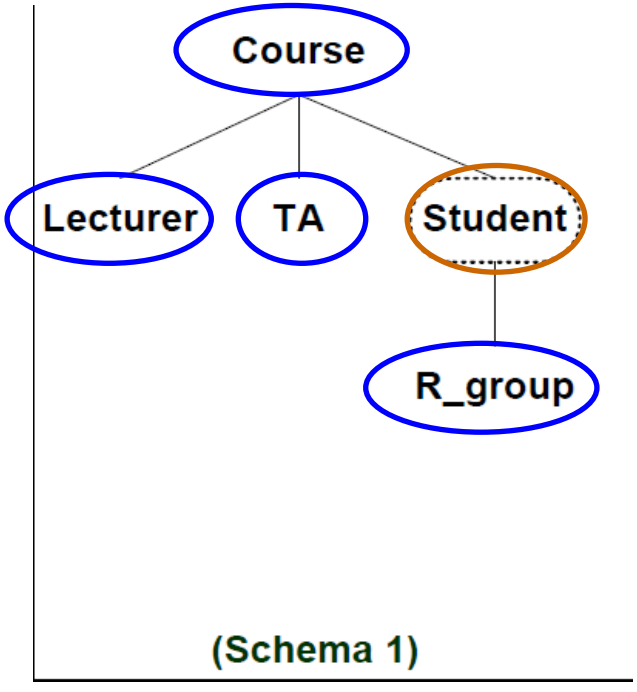
Ans1. Common courses —LCA→ **YES**

Ans2. Common R_groups —LCA→ **NO**

Ans3. Common lecturers —LCA→ **NO**

Ans4. Common TAs —LCA→ **NO**

Similar to Ans3



with Schema 1, we can find all answers:

- **common ancestors**
- **common descendants**
- ❖ **common relatives**

The Common Relative semantics Theory

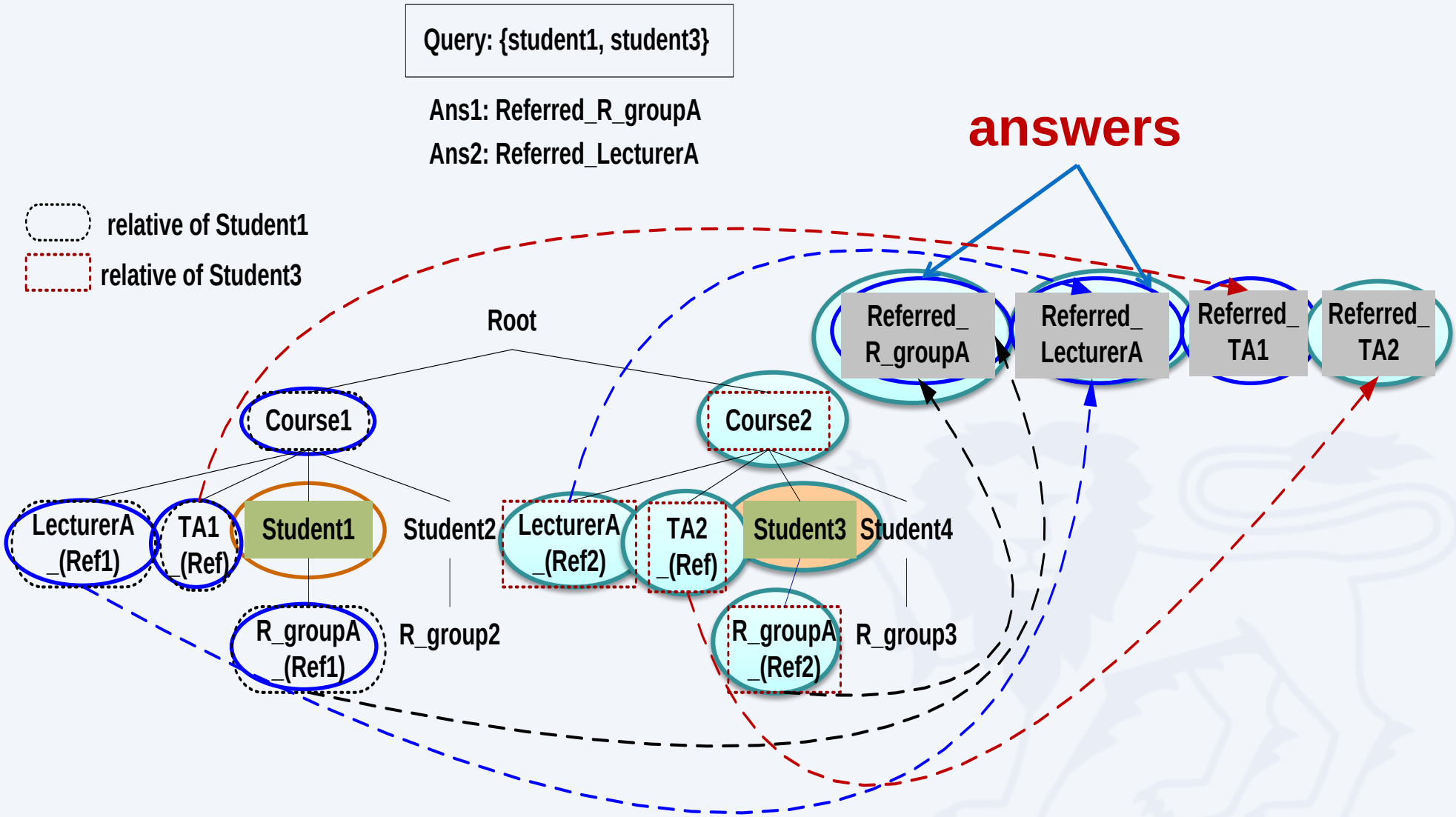
- **Definition**. Two nodes are relatives if all nodes on the path connecting them are of different object classes.

- **Property 1**. If u is a relative of v in an XML database D, then there exists some XML database D' equivalent to D such that u' is an ancestor of v', where u' and v' refer to the same object with u and v respectively.

- **Property 2.** If w is a common relative of u and v in an XML database D, then there exists some XML database D' equivalent to D such that w' is a common ancestor of u' and v', where w', u' and v' refer to the same object with w, u and v respectively.

[Ref] For more detail see paper by Thuy Ngoc Le, Zhifeng Bao, Tok Wang Ling, "Schema-independence in XML Keyword Search", ER, 2014.

The Common Relative semantics



(a part of data with IDREFs w.r.t. Schema 1)

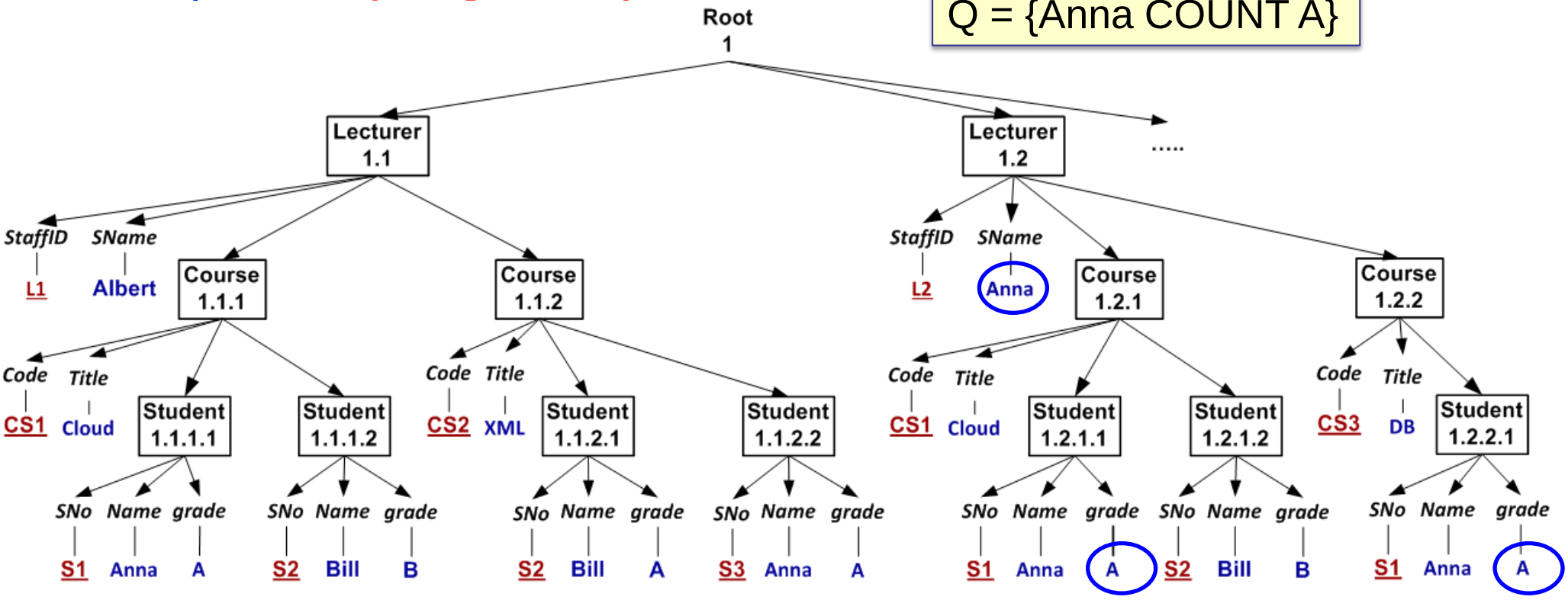
Summary on Schema-independent XML keyword search

- **We have shown that:**
  - **meaningful answers** can be found beyond common ancestors
  - when users issue a query, their expectations are independent from the schema designs.
- We proposed a novel semantics called CR **(Common Relative),** which corresponds to a **common ancestor in some equivalent document.**
  - provides more **meaningful answers** than common ancestors
  - also includes **common descendants** and **common relatives.**
  - The answers are **independent** from schema designs

- Answer **aggregate functions** in keyword queries on XML
- Challenges
  1. A query usually has different interpretations
     - if all answers from different interpretations are mixed altogether, results for group-by and aggregate functions will be incorrect
     - Need to generate all interpretations of a query and process them separately
  2. An object and a relationship can be duplicated
     - cause wrong results if not detected
     - Need to detect duplicated objects and relationships and do not count them multiple times

- Answer **aggregate functions** in keyword queries on XML
  - Impact of query interpretations

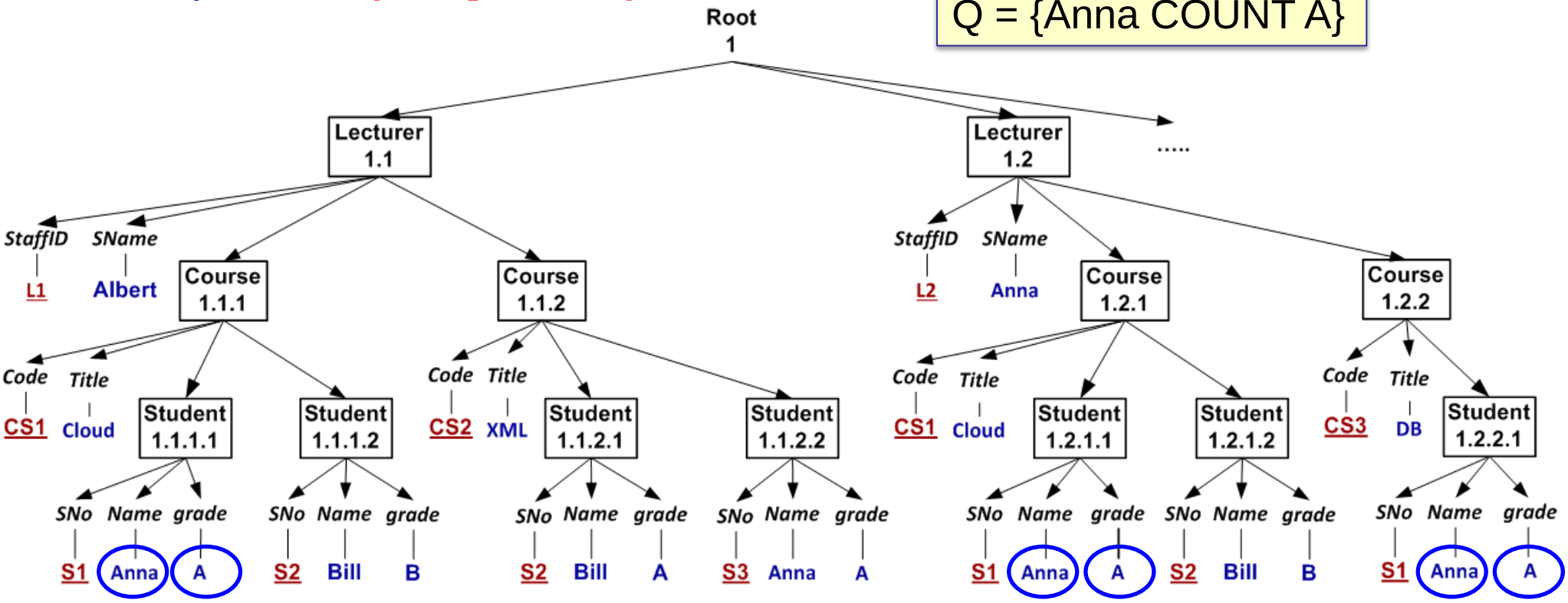
Q = {Anna COUNT A}



IQ_1 find number of grade A of students taking courses taught by Lecturer Anna → count(A) = 2

- Answer **aggregate functions** in keyword queries on XML
  - Impact of query interpretations

Q = {Anna COUNT A}



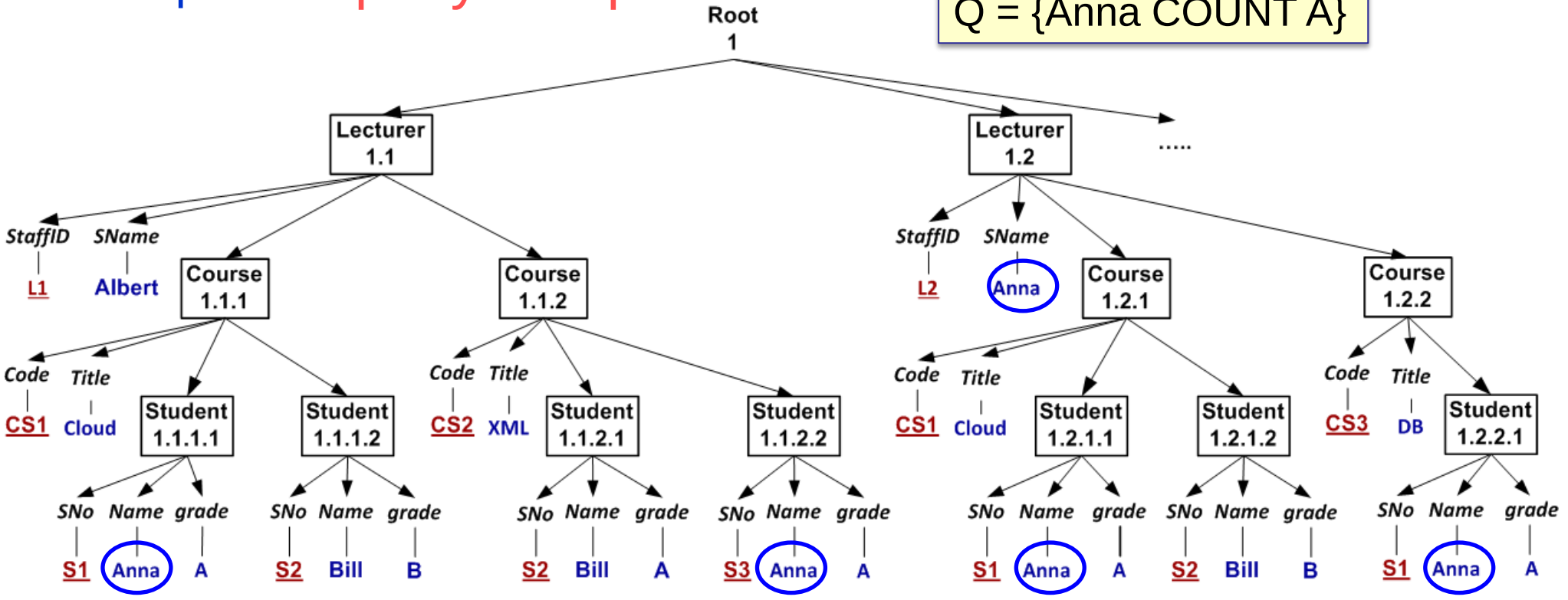
IQ_2 find number of grade A of Student Anna whose SNo is S1

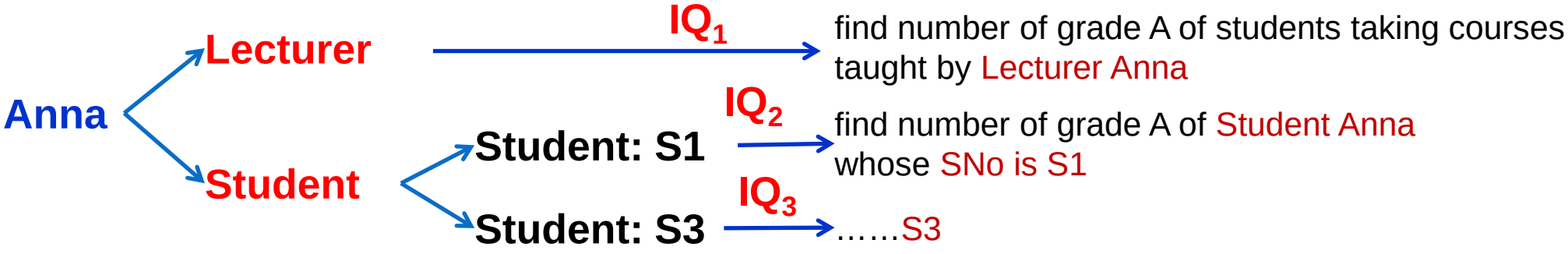
→ count(A) = 3 without considering duplicated relationships

→ count(A) = 2 considering duplicated relationships

- Answer aggregate functions in keyword queries on XML
  - Impact of query interpretations

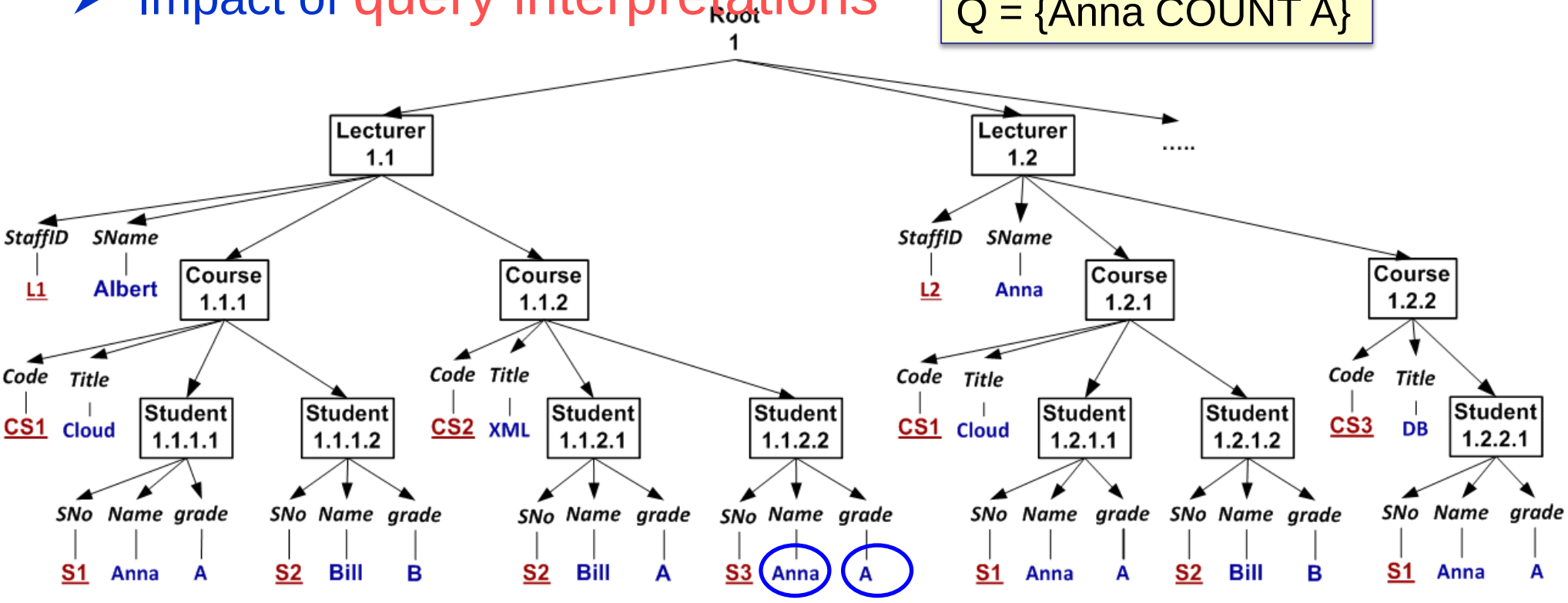
Q = {Anna COUNT A}





- Answer aggregate functions in keyword queries on XML
  - Impact of query interpretations

Q = {Anna COUNT A}

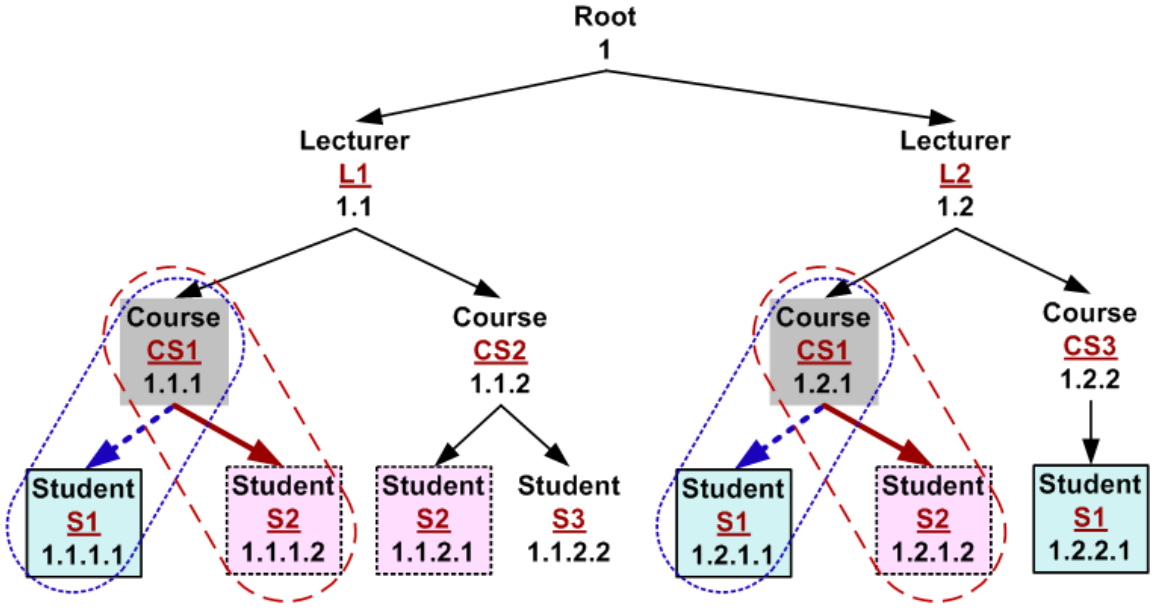


IQ_1 find number of grade A of students taking courses taught by Lecturer Anna → count(A) = 2

IQ_2 find number of grade A of Student Anna whose SNo is S1 → count(A) = 2 (not 3)

IQ_3 findnumber of grade A of Student Anna whose SNo is S3 → count(A) = 1

- Answer **aggregate functions** in keyword queries on XML
  - Impact of duplicated objects & relationships



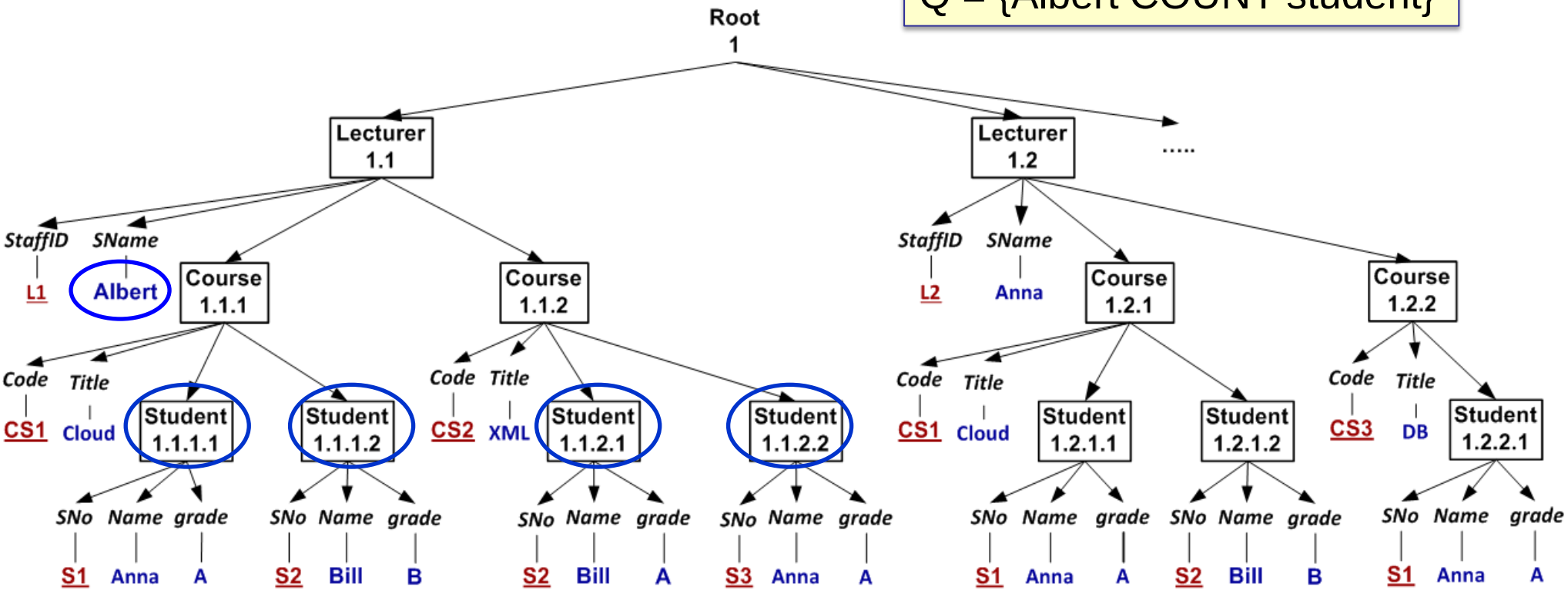
Reasons of duplication:

m : n or m : 1 relationships

Relationship	Duplication
{<Course:CS1>, <Student:S1>}	{Course (1.1.1), Student (1.1.1.1)}, {Course (1.2.1), Student (1.2.1.1)}
{<Course:CS1>, <Student:S2>}	{Course (1.1.1), Student (1.1.1.2)}, {Course (1.2.1), Student (1.2.1.2)}

- Answer **aggregate functions** in keyword queries on XML
  - Impact of duplicated objects & relationships

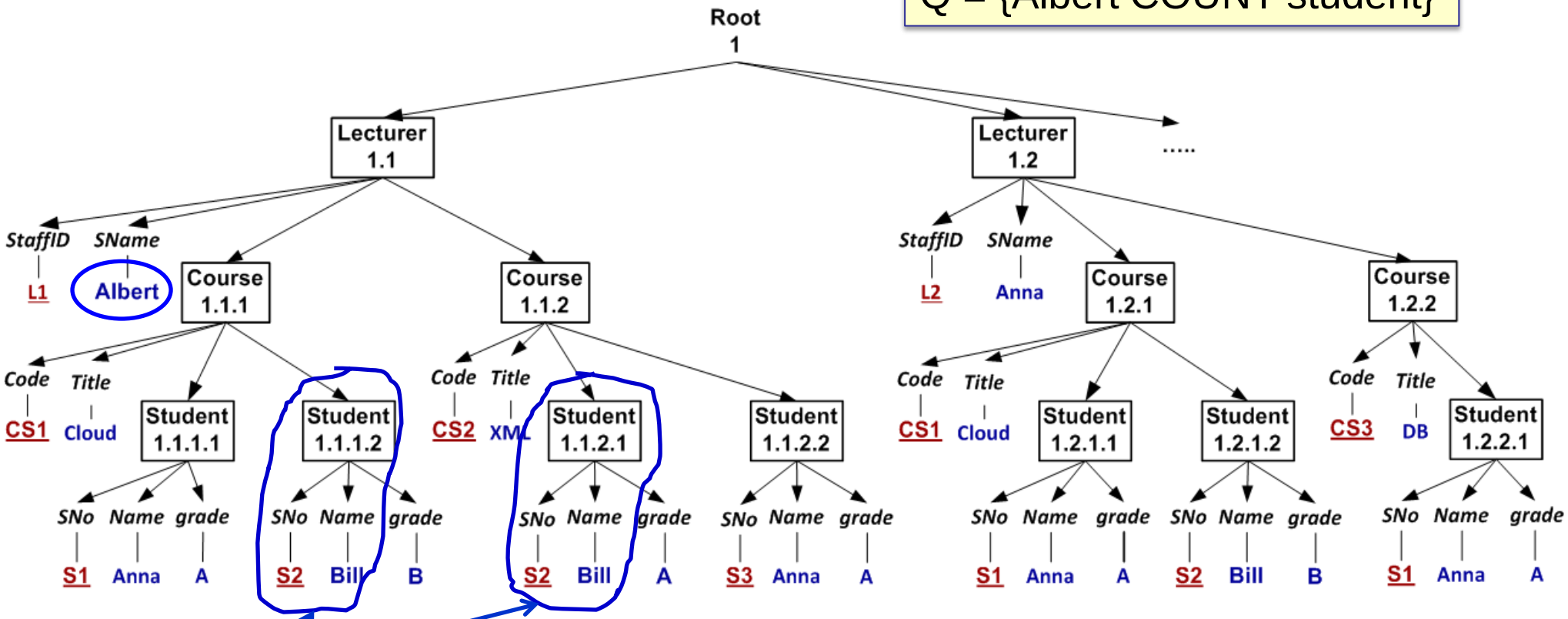
Q = {Albert COUNT student}



Without considering duplicated objects ⟶ count=4

- Answer **aggregate functions** in keyword queries on XML
  - Impact of duplicated objects & relationships

Q = {Albert COUNT student}



Without considering duplicated objects → count=4

Considering duplicated objects → count=3

Outline

- Introduction
- Limitations of Relational Model
- Limitations of XML Data Model
- ORA-semantics in Data and Schema Integration
- ORA-semantics in RDB Keyword Search
- ORA-semantics in XML Keyword Search
- Conclusion

Conclusion 1

- Common database models such as **relational model** and **XML data model** have **no** concepts of ORA-semantics, which leads to problematic schemas in database design
  - FDs are **artificially imposed** by database designers
  - Existence of MVDs is because of **wrong designs**
  - MVDs are relation sensitive
  - FD & MVD do not capture ORA-semantics
  - **Decomposition** and **Synthesis** method for RDB design
    - ❖ Process is non-deterministic
    - ❖ Cannot handle recursive relationship, ISA relationship, more than one relationship type among object classes in ER
    - ❖ Synthesis does not guarantee reconstructibility and does not consider MVD

➢ RDB design using ER approach is much better.

Conclusion 2

- Without ORA-semantics, **data and schema integration** suffers from many problems such as
  - different data models
  - different relationship types
  - local/global object identifier
  - local/global FD
  - **semantic dependency**
  - **schematic discrepancy**

Conclusion 3

- Existing **RDB / XML keyword search** do not consider ORA-semantics, and thus return
  - incomplete answers
  - duplicated answers
  - meaningless answers
  - inconsistent types of answers
  - schema dependent answers

❑ We exploit ORA semantics in RDB (ORM schema/data graph) and in XML (O-tree) to find solutions for the above problems

❑ We include metadata keywords, aggregate functions in keyword queries to enhance their expressive power and evaluation, and utilize ORA-semantics to process queries correctly

❖ ORA semantics can solve all the above problems and improve the correctness of database research in these areas!

References

1. **An analysis of multivalued and join dependencies based on the entity-relationship approach.**
   T. W. Ling.
   In Data & Knowledge Engineering, 1985.
2. **Resolving structural conflicts in the integration of entity relationship schemas.**
   M. L. Lee, and T. W. Ling.
   In OOER, 1995.
3. **Synthesizing third normal form relations from functional dependencies.**
   P. A. Bernstein.
   In ACM Trans. Database Syst., 1976.
4. **An improved third normal form for relational databases.**
   T. W. Ling, F. W. Tompa, and T. Kameda.
   In ACM Trans. Database Syst., 1981.
5. **A normal form for entity-relationship diagrams.**
   T. W. Ling.
   In ER, 1985.
6. **ORA-SS: an object-relationship-attribute model for semistructured data.**
   G. Dobbie, X. Wu, T. W. Ling, and M. L. Lee.
   Technical report, National University of Singapore, 2000.

References

7. **Extending and inferring functional dependencies in schema transformation.**
   Q. He and T. W. Ling.
   In CIKM, 2004.
8. **Keyword searching and browsing in databases using BANKS.**
   A. Hulgeri and C. Nakhe.
   In ICDE, 2002.
9. **Discover: keyword search in relational databases.**
   V. Hristidis and Y. Papakonstantinou.
   In VLDB, 2002.
10. **A Semantic Approach to Keyword Search over Relational Databases.**
    Z. Zeng, Z. Bao, M. L. Lee, and T. W. Ling.
    In ER, 2013.
11. **ExpressQ: Identifying Keyword Context and Search Target in Relational Keyword Queries.**
    Z. Zeng, Z. Bao, T. N. Le, M. L. Lee, and T. W. Ling.
    In CIKM, 2014.
12. **Answering Keyword Queries involving Aggregates and GROUPBY on Relational Databases.**
    Z. Zeng, M. L. Lee, and T. W. Ling.
    In EDBT, 2016.

References

13. **Efficient keyword search for smallest LCAs in XML databases.**
    Y. Xu and Y. Papakonstantinou.
    In SIGMOD, 2005.
14. **Fast ELCA computation for keyword queries on XML data.**
    R. Zhou, C. Liu, and J. Li.
    In EDBT, 2010.
15. **Discovering semantics from data-centric XML.**
    L. Li, T. N. Le, H. Wu, T. W. Ling, and S. Bressan.
    In DEXA, 2013.
16. **Object semantics for xml keyword search.**
    T. N. Le, W. T. Ling, H. V. Jagadish, and J. Lu.
    In DASFAA, 2014.
17. **Schema-independence in xml keyword search.**
    T. N. Le, Z. Bao, and W. T. Ling.
    In ER, 2014.
18. **Group-by and aggregate functions in xml keyword search.**
    T. N. Le, Z. Bao, W. T. Ling, and G. Dobbie.
    In DEXA, 2014.
19. **SQAK: Doing more with keywords.**
    Sandeep Tata and Guy M Lohman. In SIGMOD, 2008.

NUS
National University
of Singapore