Data Models Revisited: Improving the Quality of Database Schema Design, Integration, and Keyword Search with ORA-Semantics

Tok Wang Ling
Zhong Zeng, Mong Li Lee, Thuy Ngoc Le
National University of Singapore
DEXA 2018

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
- Future Research
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
- Future Research

Introduction

**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]
Introduction

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.

- However, the relational model and XML data model do not capture ORA-semantics, which lead to problems in RDB/XML database design, data/schema integration, and RDB/XML keyword query processing (to be discussed).

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
- Future Research
Limitations of Relational Model

FDs and MVDs

2 classes of integrity constraints in relational model:
  o Functional Dependency (FD)
  o Multivalued Dependency (MVD)

Most of FDs are imposed by database designers or organizations.

- E.g. E# and SSN are unique with respect to a company database.
  - Both E# and SSN can be used to identify an employee.
  - But why do we need both of them?
  - E# is local to a company vs SSN is global in US.
  - Concepts: Local object identifier vs global object identifier
  - Both E# and SSN are artificially introduced by some designers

- E.g. Each employee only has one name.
  - Why? Some employee may have more than one name.
  - Reason: It is an imposed restriction by designer for efficiency processing purpose.

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

- A multivalued attribute and a multivalued/single valued attribute are wrongly grouped in one relation.
  
  E.g. 
  
  \[ \text{Lecturer\_hobby\_qual (LID, Hobby, Degree, Major, Univ, Year)} \]

  o 2 multivalued attributes:
    - Hobby
    - \{Degree, Major, Univ, Year\} i.e. Qualification
  
  - A lecturer may have several hobbies and several qualifications
  
  o Key: all attributes
  
  o MVDs: \( \text{LID} \rightarrow \text{Hobby} \)
           \[ \text{LID} \rightarrow \{\text{Degree, Major, Univ, Year}\} \]
  
  o The relation not in 4NF.
Limitations of Relational Model

FDs and MVDs (cont.)

- **Existence of MVDs** are mainly because of wrong designs (cont.)
  - 2 relationship types are wrongly grouped in one relation.
    
    E.g.
    
    \[ CTL \ (Code, ISBN, LID) \]
    
    - 2 independent relationship types:
      - Many-to-many relationship between course and textbook
      - Many-to-many relationship between course and lecturer
    
    - **Key**: all attributes.
    
    - **Relation CTL is in 3NF but not in 4NF.**
    
    - **MVDs**: \( Code \rightarrow ISBN \) and \( Code \rightarrow LID \)
    
    - The relation not in 4NF.

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

  In the previous example:
  
  \[ CTL \ (Code, ISBN, LID) \]
  
  with \( \{ Code \rightarrow ISBN, \ Code \rightarrow LID \} \)
  
  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, \( Code \rightarrow ISBN \) and \( Code \rightarrow LID \) do not hold in \( CTL' \)
  
  Note that CTL is not in 4NF but CTL’ is.
  
  - This shows that **MVDs are relation sensitive**. They are difficult to discover before relations are known.
Limitations of Relational Model
FDs and MVDs (cont.)

- FDs and MVDs cannot be automatically discovered / mined.
  E.g.
  \[
  \text{Student}(\text{SID}, \text{Name})
  \]
  - Even if student names are unique in a database instance
    \[\text{Name} \rightarrow \text{SID}\]
    is incorrect in general

- FDs and MVDs do not capture ORA-semantics.
  E.g.
  \[
  \text{Lecturer}(\text{LID}, \text{Name}, \text{DID}, \text{Joindate})
  \]
  - \text{FD}: \text{LID} \rightarrow \text{Name}, \text{DID}, \text{Joindate}
    - It does not indicate whether \text{Joindate} is an attribute of objects
      lecturers or an attribute of relationship between lectures and departments [2].

Limitations of Relational Model
FDs and MVDs (cont.)

- During normalization (i.e. database schema design)
  - Remove data redundancy in order to avoid updating anomalies. Why?
  - 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 to obtain 4NF. Why?
Limitations of Relational Model

Relational Database Design Methods

- **3 common methods** for relational database schema design:
  1) Decomposition method
  2) Synthesis method [3]
  3) The ER approach

- **Objectives:**
  - Remove redundancy
  - Remove transitive dependencies but keep the closure of given set of FDs unchanged
  - Remove MVDs completely (Why?)

---

Limitations of Relational Model

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
Limitations of Relational Model
Relational Database Design Methods (cont.)

1) Decomposition method (cont.)

❖ Disadvantages:

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

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

c) Difficult 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.

Limitations of Relational Model
Relational Database Design Methods (cont.)

2) Synthesis method [3]

❖ 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, etc. in ERD.

c) Does not guarantee reconstructibility.

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


f) Does not consider MVDs.
Limitations of Relational Model

Relational Database Design Methods (cont.)

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 on the object class names, relationship types names, etc. in the ERD and capture the ORA-semantics.
f) No need to consider MVDs.
   Why?
   ❖ The ER approach captures the ORA-semantics and avoids the problems of the decomposition method and synthesis method

Limitations of Relational Model

Summary
❖ Functional Dependency (FD) and Multi-valued Dependency (MVD) are integrity constraints which are mainly imposed by organizations or database designers. They have no ORA-semantics.
❖ Definitions of all normal forms are with respect to a single relation which are not correct. There may have global redundancies among relations in a DB.
❖ Universal Relation Assumption (URA) in Relational Model cannot handle complex relationship types such as recursive relationship type, ISA, etc.
❖ Normalization only uses FDs and MVDs to reduce data redundancy and obtain normal form relations. Keep FDs but remove MVDs. Why?
❖ Normal form databases may give bad performance (too many joins). Non Normal form databases may give good performance if information related to some FDs/MVDs will not be updated, i.e. strong FDs/MVDs. Physical database design theory behind.
❖ Relational Model cannot differentiate between object attribute and relationship attribute. (e.g. attribute Joindate)
Limitations of Relational Model

Summary (cont.)

- **Relation** in Relational Model is **not** the same as relationship. **Relation name** has **no** real meaning.
- **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.

- We **need** to know the concepts of **global FD/MVD**, **global OID**, **relationship identification** besides object identification, as multiple databases may be from different organizations. Very important concepts in **data/schema integration**.

- **Relational Model** does **not** capture **ORA-semantics**, which leads to many problems in database areas!

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
- Future Research
Limitations of XML Data Model

**XML DTD and XML Schema**

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

```xml
<!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)*

Limitations of XML Data Model

**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 object identifier (OID). However, OID may not be able to define as ID

```xml
<!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 (m:m relationships between students and courses)
Limitations of XML Data Model

XML DTD and XML Schema (cont.)

2) **IDREF** is not the same as foreign key to key reference in RDB. IDREF has no type.

   E.g.  Prereq  IDREFS  #IMPLIED

   IDREF cannot be constrained.

3) Multivalued attribute cannot be defined as an attribute

   ```xml
   <!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)

   - We cannot define **Hobby** as attributes of **Lecturer** elements.
   - **Hobby** has to be declared as sub-elements of Lecturer elements.
   - Can’t tell hobby is an multi-valued attribute of lecturers
4) Relationship type is implicit via parent-child relationship

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

(Example XML fragment)

- Cannot distinguish between object attribute (Name) vs relationship attribute (Grade) as both Name and Grade are sub-elements of Student.

Limitations of XML Data Model

**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
- Future Research

ORA-semantics needed 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) Entity resolution and different relationship type
  3) Local vs Global object identifier
  4) Local vs Global FD
  5) Semantic dependency
  6) Schematic discrepancy
(1) Different data models

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

(2) Different relationship types

- Entity resolution (i.e., object identification and record linking) is not enough for data/schema integration
- Consider 2 databases about person and house:
  
  $DB1: \text{PersonHouse}(SSN, Address)$
  $DB2: \text{PersonHouse}(SSN, Address)$

  o 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 $\text{Own}$ i.e. $person$ owns $house$
  - $DB2$ may capture relationship type $\text{Live}$ 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 vs Global object identifier

- We need to consider **local** object identifier vs **global** object identifier for correct data/schema integration.
- Consider 2 databases from 2 universities 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, because 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 FD vs Global FD

- We need to consider **local** FD vs **global** FD for correct data/schema integration.
- Consider 2 databases of two bookstores:

  **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 → {Title, First_Author}
    - **local** FD: ISBN → Price
  - The integrated database should include 2 relations:

    Book_infor (ISBN, Title, First_Author)
    Book_price (ISBN, bookstore, Price)
ORACLE semantics in Data and Schema Integration

(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:

  \[ \text{R1: } \text{Emp}(EID, Ename, \text{Jointdate}, DID) \]
  \[ \text{R2: } \text{Dept}(DID, Dname) \]

  with FDs: \( EID \to \{ Ename, \text{Jointdate}, DID \} \) & \( DID \to Dname \)

  - It is unclear if \( \text{Jointdate} \) is:
    a) the date when an employee joined the company or
    b) the date when an employee started working for a department
    i.e. whether Jointdate is an entity attribute or a relationship attribute.

    - If \( \{ EID, DID \} \xrightarrow{\text{Sem}} \text{Jointdate} \)
      i.e. \( \text{Jointdate} \) is the date when an employee started working for a department,
      then when an employee moves to another department, we need to update Jointdate.

ORACLE semantics in Data and Schema Integration

(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:

    | Design | Table Name          | Attributes            |
    |--------|---------------------|-----------------------|
    | DB1:   | Supply              | SID, PID, Month, Quantity |
    | DB2:   | Supply              | SID, PID, Jan, Feb, ..., Dec |
    | DB3:   | Jan_Supply          | SID, PID, Quantity    |
    |        | Feb_Supply          | SID, PID, Quantity    |
    |        | …                   |                      |
    |        | Dec_Supply          | SID, PID, Quantity    |
(6) Schematic discrepancy

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].

Summary

Many issues must be considered during data and schema integration:

1) Different data model
2) Relationship resolution / identification besides entity resolution
3) Local vs Global object identifier
4) Local vs Global FD
5) Semantic dependency
6) Schematic discrepancy

All the above require ORA-semantics to achieve good quality integrated databases / schemas.
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
- Future Research

Querying a database - RDB or XML - 2 ways

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)

John, Java

- 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

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)

- 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%'

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

How to have advantages of both structured search and KWS?

SEARCH → Keyword SEARCH

More satisfactory answers
More expressive queries
**ORA-semantics in RDB Keyword Search**

- **RDB query processing**

  **Example: University database**

<table>
<thead>
<tr>
<th>Student</th>
<th>Course</th>
<th>Department</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID</td>
<td>Name</td>
<td>Code</td>
</tr>
<tr>
<td>S1</td>
<td>Bill</td>
<td>CS301</td>
</tr>
<tr>
<td>S2</td>
<td>John</td>
<td>CS521</td>
</tr>
<tr>
<td>S3</td>
<td>Mary</td>
<td>CS203</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Enrol</th>
<th>Qualification</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID</td>
<td>Code</td>
</tr>
<tr>
<td>E1</td>
<td>S1</td>
</tr>
<tr>
<td>E2</td>
<td>S2</td>
</tr>
<tr>
<td>E3</td>
<td>S2</td>
</tr>
<tr>
<td>E4</td>
<td>S3</td>
</tr>
<tr>
<td>E5</td>
<td>S3</td>
</tr>
</tbody>
</table>

  **Query:** find grade that student *John* obtains in *Java* course

  ```sql
  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%'
  ```

**ORA-semantics in RDB Keyword Search**

- **Current data graph approach** [8]

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

  **Q:** Why? Any justification?

  (data graph of university database)
ORA-semantics in RDB Keyword Search
– Current data graph approach [8]

Q={John Java}

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

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

Another result:

Q: Why? Any justification?

(discussion of results and their implications)

ORA-semantics in RDB Keyword Search
– Current schema graph approach [9]

Q={John Java}

Foreign key-key constraint

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

Select a 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%'

One graph:

Another graph:

Translation into SQL
ORA-semantics in RDB Keyword Search

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

1) Incomplete object answer

Lecturer

<table>
<thead>
<tr>
<th>LID</th>
<th>Name</th>
<th>DID</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1</td>
<td>Smith</td>
<td>D1</td>
</tr>
<tr>
<td>L2</td>
<td>Smith</td>
<td>D2</td>
</tr>
<tr>
<td>L3</td>
<td>Steven</td>
<td>D1</td>
</tr>
</tbody>
</table>

Qualification

<table>
<thead>
<tr>
<th>DID</th>
<th>Degree</th>
<th>Major</th>
<th>University</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>L1</td>
<td>PhD</td>
<td>CS</td>
<td>NUS</td>
</tr>
<tr>
<td>Q2</td>
<td>L3</td>
<td>PhD</td>
<td>CS</td>
<td>SMU</td>
</tr>
<tr>
<td>Q3</td>
<td>L3</td>
<td>Master</td>
<td>EE</td>
<td>NTU</td>
</tr>
</tbody>
</table>

Q = \{Steven\}

Only 1 answer: L3

Additional information about qualifications of Steven is expected because they are properties of lecturers
2) Incomplete relationship answer

\[ Q = \{ \text{Bill A} \} \]

Student

<table>
<thead>
<tr>
<th>SID</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>Bill</td>
</tr>
<tr>
<td>S2</td>
<td>John</td>
</tr>
<tr>
<td>S3</td>
<td>Mary</td>
</tr>
</tbody>
</table>

Enrol

<table>
<thead>
<tr>
<th>SID</th>
<th>Code</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>E1</td>
<td>S1</td>
<td>CS521 A</td>
</tr>
<tr>
<td>E2</td>
<td>S2</td>
<td>CS203 B</td>
</tr>
<tr>
<td>E3</td>
<td>S2</td>
<td>CS521 A</td>
</tr>
<tr>
<td>E4</td>
<td>S3</td>
<td>CS203 A</td>
</tr>
<tr>
<td>E5</td>
<td>S3</td>
<td>CS301 B</td>
</tr>
</tbody>
</table>

Course

<table>
<thead>
<tr>
<th>Code</th>
<th>Title</th>
<th>LID</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS301</td>
<td>IR</td>
<td>L2</td>
</tr>
<tr>
<td>CS521</td>
<td>DB</td>
<td>L1</td>
</tr>
<tr>
<td>CS203</td>
<td>Java</td>
<td>L1</td>
</tr>
</tbody>
</table>

One answer: S1-E1

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

3) Meaningless answer

\[ Q = \{ S1 S3 \} \]

Student

<table>
<thead>
<tr>
<th>SID</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>Bill</td>
</tr>
<tr>
<td>S2</td>
<td>John</td>
</tr>
<tr>
<td>S3</td>
<td>Mary</td>
</tr>
</tbody>
</table>

Course

<table>
<thead>
<tr>
<th>Code</th>
<th>Title</th>
<th>LID</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS301</td>
<td>IR</td>
<td>L2</td>
</tr>
<tr>
<td>CS521</td>
<td>DB</td>
<td>L1</td>
</tr>
<tr>
<td>CS203</td>
<td>Java</td>
<td>L1</td>
</tr>
</tbody>
</table>

Enrol

<table>
<thead>
<tr>
<th>SID</th>
<th>Code</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>E1</td>
<td>S1</td>
<td>CS521 A</td>
</tr>
<tr>
<td>E2</td>
<td>S2</td>
<td>CS203 B</td>
</tr>
<tr>
<td>E3</td>
<td>S2</td>
<td>CS521 A</td>
</tr>
<tr>
<td>E4</td>
<td>S3</td>
<td>CS203 A</td>
</tr>
<tr>
<td>E5</td>
<td>S3</td>
<td>CS301 B</td>
</tr>
</tbody>
</table>

Lecturer

<table>
<thead>
<tr>
<th>LID</th>
<th>Name</th>
<th>DID</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1</td>
<td>Smith</td>
<td>D1</td>
</tr>
<tr>
<td>L2</td>
<td>Smith</td>
<td>D2</td>
</tr>
<tr>
<td>L3</td>
<td>Steven</td>
<td>D1</td>
</tr>
</tbody>
</table>

Enrol

<table>
<thead>
<tr>
<th>SID</th>
<th>Code</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>E1</td>
<td>S1</td>
<td>CS521 A</td>
</tr>
<tr>
<td>E2</td>
<td>S2</td>
<td>CS203 B</td>
</tr>
<tr>
<td>E3</td>
<td>S2</td>
<td>CS521 A</td>
</tr>
<tr>
<td>E4</td>
<td>S3</td>
<td>CS203 A</td>
</tr>
<tr>
<td>E5</td>
<td>S3</td>
<td>CS301 B</td>
</tr>
</tbody>
</table>

More information expected:
Grade is a relationship attribute;
The details of other participating objects (i.e. course) of the relationship are expected
ORA-semantics in RDB Keyword Search
– Problems of current RDB keyword search

3) Meaningless answer (cont.)

\[ Q = \{ S_1, S_3 \} \]

2 answers:
1st answer: \( S_3-E_4-CS203-L_1-CS5201-E_1-S_1 \)

**Meaning?** (difficult to know from the minimal connected subgraph):
the common lecturer of \( S_1 \) & \( S_3 \) (meaningful)

2nd answer: \( S_3-E_4-CS203-E_2-CS203-E_3-CS5201-E_1-S_1 \)

**Meaning?** \( S_2 \) enrolls a same course with \( S_1 \) and enrolls another same course with \( S_3 \).

Probably not meaningful: not correspond to an LCA of any hierarchical structure XML doc representing the same database
ORA-semantics in RDB Keyword Search

- 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 to user?

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

Q1 = \{S1, S2\}
Q2 = \{S1, S3\}

common course of S1 & S2

common lecturer of S1 & S3

Two similar queries have very different answers and user will get confused!
ORA-semantics in RDB Keyword Search

– Problems of current RDB keyword search

6) Schema dependent answer

<table>
<thead>
<tr>
<th>Student</th>
<th>Enrol</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID</td>
<td>Name</td>
</tr>
<tr>
<td>S1</td>
<td>Bill</td>
</tr>
<tr>
<td>S2</td>
<td>John</td>
</tr>
<tr>
<td>S3</td>
<td>Mary</td>
</tr>
<tr>
<td>E1</td>
<td>S1</td>
</tr>
<tr>
<td>E2</td>
<td>S2</td>
</tr>
<tr>
<td>E3</td>
<td>S2</td>
</tr>
<tr>
<td>E4</td>
<td>S3</td>
</tr>
<tr>
<td>E5</td>
<td>S3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Enrollment (1NF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID</td>
</tr>
<tr>
<td>E1</td>
</tr>
<tr>
<td>E2</td>
</tr>
<tr>
<td>E3</td>
</tr>
<tr>
<td>E4</td>
</tr>
<tr>
<td>E5</td>
</tr>
</tbody>
</table>

If we denormalize, we can observe:

Q = {S1, S3}

No answer returns because no connected subgraph contains all the keywords.

Expected answers: common lecturer of S1 & S3 from the 3 original normalized relations.

Another query:
Q = {S3}

2 answers:
1) E4
2) E5

The information of student S3 are duplicated.
- Should only output E4 or E5
- The 3 original normalized relations give correct answer.
Summary of Problems.

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

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

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.
ORA-semantics in RDB Keyword Search
– our ORA-Semantics approach

(ER diagram of University database)

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

Types of Relations

(Example)

(ER diagram of University database)

student(SID, Name)

Course(Code, Title, LID)

Course.[LID] ⊆ Lecturer.[StaffID]

enrol(SID, Code, Grade)

Enrol.[SID] ⊆ Student.[SID]

Enrol.[Code] ⊆ Course.[Code]

Lecturer(LID, Name, DID)

Lecturer.[DID] ⊆ Department.[DID]

Department(DID, Name, Address)

Qualification(LID, Degree, Major, University)

Qualification.[LID] ⊆ Lecturer.[LID]

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 $v$
ORA-semantics in RDB Keyword Search

– ORM data and schema graph (Example)

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
ORA-semantics in RDB Keyword Search

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

Previous Approaches

Q = {Steven}

![Data Graph](image1)

Return lecturer tuple L3 only

Our Approach

Q = {Steven}

![ORM Data Graph](image2)

Correctly return lecturer tuple L3 together with his qualifications, all properties of the lecturer object.

Avoid problem of incomplete object answer
ORA-semantics in RDB Keyword Search

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

**Previous Approaches**

\[ Q = \{ \text{Bill A} \} \]

![Data Graph](image)

- Only return student tuple \(S1\) and enrol tuple \(E1\)

**Our Approach**

\[ Q = \{ \text{Bill A} \} \]

![ORM Data Graph](image)

- Correctly return student tuple \(S1\), enrol tuple \(E1\) and course tuple \(CS521\) as participating object of enrol relationship

Avoid problem of **incomplete relationship answer**
ORA-semantics in RDB Keyword Search

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

**Summary**
We 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 (skipped)
4) **Complex** answer (skipped)
5) **Schema dependent** answer

Need **ORA-semantics** to solve these problems.

---

ORA-semantics in RDB Keyword Search

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

\[
Q = \{\text{John, Mary}\} \quad \text{or} \quad Q' = \{\text{Course Student John Student Mary}\}
\]
ORA-semantics in RDB Keyword Search

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

\[ Q = \{ \text{Course, Student, John, Student, Mary} \} \]

- (Course) refers to some course object – the search target
- (Student, John) refers to a student name John
- (Student, Mary) refers to a student name Mary

**Query Pattern:**

\[ \text{Name=John} \]
\[ \text{Enrol} \]
\[ \text{Course} \]
\[ \text{Name=Mary} \]
\[ \text{Enrol} \]
\[ \text{Student} \]

- **Search intention:** find course that is enrolled by both students John and Mary

**ORM schema graph**

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

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

\[ Q = \{ \text{Green, SUM, Credit} \} \]

**Student**

<table>
<thead>
<tr>
<th>Sid</th>
<th>Sname</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>s1</td>
<td>George</td>
<td>22</td>
</tr>
<tr>
<td>s2</td>
<td>Green</td>
<td>24</td>
</tr>
<tr>
<td>s3</td>
<td>Green</td>
<td>21</td>
</tr>
</tbody>
</table>

**Course**

<table>
<thead>
<tr>
<th>Code</th>
<th>Title</th>
<th>Credit</th>
</tr>
</thead>
<tbody>
<tr>
<td>c1</td>
<td>Java</td>
<td>5.0</td>
</tr>
<tr>
<td>c2</td>
<td>Database</td>
<td>4.0</td>
</tr>
<tr>
<td>c3</td>
<td>Multimedia</td>
<td>3.0</td>
</tr>
</tbody>
</table>

**Output answer:** 13

**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

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

Correct answer: s2 is 5, s3 is 8

ORA-semantics in RDB Keyword Search

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
- Future Research
ORA-semantics in XML Keyword Search
– Background
  - XML query processing

Query: find grade that student John obtains in Java course

XPath:
//Course[Title=Java][Student/Name=John]/Grade

ORA-semantics in XML Keyword Search
– Current XML keyword search: LCA approach

Q=(John Java)

Common ancestor (CA)

(User.png)
ORA-semantics in XML Keyword Search

– Current XML keyword search: LCA approach

Q=(John Java)

Common ancestor (CA)

LCA is an answer

Lowest CA (LCA)

Why? Any justification?

Oracle semantics in XML Keyword Search

– Problems of current XML keyword search

- 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
ORA-semantics in XML Keyword Search

1) Meaningless answer

\[ Q = \{ \text{Bill} \} \]

Expected: include other properties of the student

Reasons: do not have concept of object → cannot distinguish object node vs. non-object node
ORA-semantics in XML Keyword Search
– Problems of current XML keyword search

2) Missing answer

\[ Q = \{\text{DB, Java} \} \]

LCA returns this answer

Legend

Object Node

Matching nodes

ORA-semantics in XML Keyword Search
– Problems of current XML keyword search

2) Missing answer

\[ Q = \{\text{DB, Java} \} \]

Matching objects

Legend

Object Node

Matching nodes
Problems of current XML keyword search

2) Missing answer

Q=(DB Java)

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

LCA misses this answer

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

3) Duplicated answer

Q=(S2 John)

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

Should return only one of them

Identical subtrees → Duplicated answers
Problems of current XML keyword search

vs.

Problems related to relationships

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 = \{S_1, S_2\}

Q2 = \{S_1, S_3\}

Two similar queries but have very different answers and user will be confused.

Reasons:
(1) do not have the concepts of object & relationship
(2) rely on hierarchical structure of XML data
ORA-semantics in XML Keyword Search

– Problems of current XML keyword search

6) Schema dependent answer

• Will discuss it later.
ORA-semantics in XML Keyword Search
– Object nodes vs. non-object nodes

- **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

- **Non-object nodes**
  - Associated non-object nodes to the corresponding object nodes

- **Legend**
  - Object Node

---

**ORA-semantics in XML Keyword Search**
– **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
ORA-semantics in XML Keyword Search

**O-tree (Example)**

**(XML data tree)**

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 to avoid missing answers (Skip)

- **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

❑ **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? What is schema?)

---

(ER diagram with binary relationships)

Many ways to represent the database in XML

---

(Five Reasonable XML schema trees)
ORA-semantics in XML Keyword Search

Schema independent XML keyword search

Motivation

Expected answers

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

Q = \{studentA, studentB\}

Common ancestors in some schema(s)

(Five Reasonable XML schema trees)
ORA-semantics in XML Keyword Search

**Schema independent** XML keyword search

**Motivation**

Five different sets of answers for the 5 schemas:

- **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

(5 Reasonable XML schema trees)

**ORA-semantics in XML Keyword Search**

**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 and what is schema.
- However, all five different schema designs provide five different sets of answers by LCA semantics
ORA-semantics in XML Keyword Search

- **Schema independent** XML keyword search

  - Intuition of our Common Relative (CR) semantics

  \[ Q = \{\text{studentA, studentB}\} \]

  **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?

---

ORA-semantics in XML Keyword Search

- **Schema independent** XML keyword search

  - Intuition of our Common Relative (CR) semantics

  \[ Q = \{\text{studentA, studentB}\} \]

  **Expected answers:**

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

  - Schema 1: Ans1 (course) → LCA, YES
  - Schema 2: Ans2 & Ans3 (lecturer) → NO
  - Schema 3: Ans1 & Ans4 (TA) → NO
  - Schema 4: no answer → NO
  - Schema 5: Ans2 (R_group) → NO

  How to find Ans2 with Schema1?
ORA-semantics in XML Keyword Search

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

Find Ans2: Common R_groups

\[ Q = \{\text{studentA, studentB}\} \]

Common descendant

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

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

Expected answers:

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

How to find Ans3 with Schema1?
ORA-semantics in XML Keyword Search

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

Find Ans3: Common lecturers

\[ Q = \{ \text{studentA, studentB} \} \]

Course

Lecturer

TA

Student

Relative

Course 1

Course 2

Course 1

Course 2

common

Lecturer

Student A

common

Lecturer

Student B

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

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

ORAS- semantics in XML Keyword Search

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

Expected answers:

\[ Q = \{ \text{studentA, studentB} \} \]

Ans1. Common courses

Ans2. Common R_groups

Ans3. Common lecturers

Ans4. Common TAs

with Schema 1, we can find all answers:
- common ancestors
- common descendants
  - common relatives

LCA

YES

NO

NO

NO

Similar to Ans3

Course

Lecturer

TA

Student

R_group

(Schema 1)
The Common Relative semantics

Query: {student1, student3}

Ans1: Referred_R_groupA
Ans2: Referred_LecturerA

The Common Relative semantics

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
- We need ORA-semantics to solve the problems
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
     - Must detect duplicated objects and relationships and do not count them multiple times.
     - Skip some details.

**Impact of query interpretations**

\[ Q = \{ \text{Anna COUNT A} \} \]

- Find number of grade A of students taking courses taught by Lecturer Anna
- Find number of grade A of Student Anna whose SNo is S1
- ...
Answer aggregate functions in keyword queries on XML

Impact of query interpretations

\[ Q = \{\text{Anna COUNT A}\} \]

IQ₁: find number of grade A of students taking courses taught by Lecturer Anna
\[ \text{count}(A) = 2 \]

IQ₂: find number of grade A of Student Anna whose SNo is S₁
\[ \text{count}(A) = 2 \quad \text{(not 3, need ORA-semantics)} \]

IQ₃: find number of grade A of Student Anna whose SNo is S₃
\[ \text{count}(A) = 1 \]

Reasons of duplication: 
- m : n or m : 1 relationships

Need ORA-semantics to detect duplicates!
Answer aggregate functions in keyword queries on XML

Impact of duplicated objects & relationships

Q = \{S1 \text{ COUNT } A\}

Without considering duplicated relationships → 3
Considering duplicated relationships → 2

Need ORA-semantics to detect duplicates!

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
- Future Research
**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** (which captures ORA-semantics) 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

  - We need **ORA-semantics** to solve the problems
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 (bad!)

- 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!

Future Research

1. Data/Schema Integration.
   - **Relationship Resolution** in Data/schema integration
   - Handle recursive relationship, ISA relationship for object type and relationship type, and cycle in schema, etc.
   - Composition of relationships, etc.

2. Keyword query search in RDB and XML data
   - Handle recursive relationship, ISA relationship for object type and relationship type, and cycle in schema, etc.
   - Allow synonym and composition of relationships, etc., in KWQ (via deductive rules)
   - **Data model independent** keyword query search for data.
   - **Extract** ORA-semantics from web documents to achieve better quality of web search results.
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.

4. An improved third normal form for relational databases.

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.

   Q. He and T. W. Ling.

8. Keyword searching and browsing in databases using BANKS.
   A. Hulgeri and C. Nakhe.
   In ICDE, 2002.

   V. Hristidis and Y. Papakonstantinou.
   In VLDB, 2002.

10. A Semantic Approach to Keyword Search over Relational Databases.
    In ER, 2013.

11. ExpressQ: Identifying Keyword Context and Search Target in Relational Keyword Queries.
    In CIKM, 2014.

12. Answering Keyword Queries involving Aggregates and GROUPBY on Relational Databases.
    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.

   T. N. Le, W. T. Ling, H. V. Jagadish, and J. Lu.
   In DASFAA, 2014.

   T. N. Le, Z. Bao, and W. T. Ling.
   In ER, 2014.

18. Group-by and aggregate functions in XML keyword search.
   In DEXA, 2014.

19. SQAK: Doing more with keywords.