Applications of ORA-SS Model
Topics

- **Normal form ORA-SS schema diagram**
  - remove redundant data
  - resolve class hierarchy conflicts

- **Storage schema for ORA-SS/XML databases**
  - use Object Relational Model

- **ORA-SS/XML Views**
  - derived information from references and class hierarchy
  - defining views
  - materialized view maintenance
  - view updates

- **Evaluating XML queries on ORA-SS databases**
  - XML schema to ORA-SS Schema
  - XML document to ORA-SS database

- **Translating** relational schema into ORA-SS schema

- **Integration** of XML documents

- **ORA-Semantics based XML Keyword Search**
Applications of ORA-SS Model

**Normal Form (NF) ORA-SS Schema Diagram**

- The two binary relationship types are many-to-many
- Schema may have a lot of redundant data. Why? Where?
- Update anomalies. Why?
- Normal Form schema is needed. What?
Applications of ORA-SS Model

**NF ORA-SS Schema Diagram (cont.)**

- Two better solutions:
  - Redundancies are removed, in normal form

(Solution 1)

(Solution 2)

**Q:** What are the problems of these two schemas/designs?

- Symmetric queries cannot be processed equally efficiently.
**Applications of ORA-SS Model**

**NF ORA-SS Schema Diagram (cont.)**

**Example:** The ORA-SS schema attempts to show that lecturers teach courses and use all the textbooks as described on the curriculum, i.e. there is a MVD constraint:

\[
\text{course-code} \rightarrow\rightarrow \text{isbn} \mid \text{staff#}
\]

The relation for the relationship type **ctl** is:

\[
\text{ctl (course-code, isbn, staff#)}
\]

It is **not in 4NF** because of the above MVD, hence the relationship type **ctl** is not in R-NF.

- **A better design:** MVD is removed.

The relations for the relationship types **ct** and **cl** are:

\[
\text{ct (course-code, isbn)}
\]

\[
\text{cl (course-code, staff#)}
\]

Both relations are in 4NF.

**Q:** Are there any redundancy? Yes! What? How to remove them if necessary?
Store ORA-SS in **nested relations**

- Problems in existing storage approaches
  - stored in **flat files** -- it is long and difficult to query or update
  - Relational DBMS -- **join** needs much time

- ORA-SS reflects the nested structure of semi-structured data and store data in nested relations.
  - **less join** in nested relations
Applications of ORA-SS Model

Storage Schema for ORA-SS/XML Databases

- **Main Rules**
  - Each object class together with its attributes form a nested relation (object relation)
  - Each relationship type together with its attributes form a nested relation (relationship relation)

- Nested relations can be handled by Object Relational model, e.g. ORACLE 8i and newer versions.
Object Relations

- Supplier \((S\#, \text{Name}, (\text{City})^*)\)
  - Notation: \((\ldots)^*\) indicates a repeating group in a nested relation; suppliers may have many cities.

- Part \((P\#, \text{Name}, \text{color})\)

- Project \((J\#, \text{Name}, \text{Loc})\)

Relationship relations

- SP \((S\#, P\#, \text{price})\)
- SPJ \((S\#, P\#, J\#, \text{Qty})\)

Constraints:

- \(\text{SPJ}[S\#, P\#] \subseteq \text{SP}[S\#, P\#]\)
- and other referential constraints.
Another example:

Object relations:
- professor (staff#, name, (area)*)
- course (cno, title)
- text (isbn, (author)*, title)

Relationship relations:
- pc (staff#, cno)
- cx (cno, isbn)

Constraint:
- cx [cno] ⊆ pc [cno]

and other referential constraints
Applications of ORA-SS Model

Views

- What information can be directly derived from references and class hierarchy?

Fig. Referencing an object class in an ORA-SS schema diagram

Applications of ORA-SS Model

Views (cont.)
- Valid views of an ORA-SS schema
- Operations: selection, projection, join, swap

❖ The positions of relationship attributes may change

E.g. The positions of price in the first 2 views.

Source schema

View 1
Swap Supplier and Part

View 2
Swap Supplier and Project from source schema

View 3
Drop Supplier from View 2 and with aggregate function sum
**Views (cont.)**

- Views from ORA-SS schema

**Main Rules**

- The hierarchical order of the object classes in a relationship relation can be changed (using **swap** operator).

- Object classes can be **dropped** from a relationship relation.

In this case, the attributes of the relationships will have different cardinalities or change to some **aggregate functions** such as sum, max/min, average, etc.
Main related work

- In all these works, the original data are in RDB
  
  * SilkRoute
    - Two declarative language RXL and XML-QL to define and query the views over relational data
  
  * XPERANTO
    - uses a canonical mapping to create a default XML view from relational data
  
  * Oracle, IBM DB2, and SQL Server
    - provide the ability to export relational data to materialized XML views
Disadvantages of the main related work

- Ignore **semantic** information in source data
  - For example, ignore the difference between object class, attribute and relationship in schema
- **Cannot** check the **validity** of designed views
- **Difficult** to use **query languages** to define views
  - Proprietary language or XQuery
  - It is difficult to write an XQuery program to swap two object classes (elements). E.g. View 1 and View 2 in slide 11.
Our approach for XML views

- **Design valid XML Views**
  - Based on a semantically rich model: ORA-SS
  - Use query operators, such as selection, drop, swap, join, etc.
  - Support more flexible views than related work, such as swapping views

- **Generating XQuery View Definitions**
  - XML data are stored as XML documents

- **Generating SQLX View Definition**
  - XML data are stored in an object-relational database
Views: **Selection operator**

- A *selection* operator filters data by using predicates.
- For example, we design a view that depicts projects for which there exist suppliers for which there exist parts with a price > 80.
Views: **Drop (projection) operator**

- **Drop** operator selects or drops object classes or attributes in the source schema. The source semantics may be affected.
- For example, the following view drops the object class supplier and its attributes.

```
Drop operator

\[
\text{View schema}
\]
```

**Source schema**

```
\[
\text{Note: We can have total_qty derived attribute below part also.}
\]
```
Views: Join operator

- Join operator joins two object classes and their attributes together by key-foreign key reference (i.e. IDREF and ID in XML data).
- For example, the following view joins supplier’ and supplier together.

Source schema

View schema
Views: **Swap operator**

- **Swap** operator exchanges the positions of any two object classes in an arbitrary path.
- For example, the following view swaps *project* and *supplier*.

```
<table>
<thead>
<tr>
<th>project</th>
<th>js,2,1:n,1:n</th>
</tr>
</thead>
<tbody>
<tr>
<td>part</td>
<td>sp,2,1:n,1:n</td>
</tr>
<tr>
<td>supplier</td>
<td>spj, 3, 1:n, 1:n</td>
</tr>
</tbody>
</table>

*Source schema*

```

```
<table>
<thead>
<tr>
<th>supplier</th>
<th>sp,2,1:n,1:n</th>
</tr>
</thead>
<tbody>
<tr>
<td>part</td>
<td>spj, 3, 1:n, 1:n</td>
</tr>
<tr>
<td>project</td>
<td>sp</td>
</tr>
</tbody>
</table>

*View schema*

**Note:** Position of price changed
Views: **Swap operator**

- **Swap** operator exchanges the positions of any two object classes in an arbitrary path.
- For example, the following view swaps *project* and *supplier*.

**Source schema**

**View schema**

*Note: Position of price changed*
Other Topics on Views

- materialized views and maintenance
- view update problem
Applications of ORA-SS Model

Evaluating XML queries on ORA-SS Databases

Related issues:

- Map XML schema to ORA-SS schema
  - Extra semantic information is needed
- Map ORA-SS schema to storage schema (OR model)
- Map XML documents to ORA-SS databases
  - using object relational (OR) model DBMS
- Map XML queries to queries on the ORA-SS databases
  then to SQLX queries on the OR DBMS
  - Need to handle recursions and wildcard such as * (any), | (or), ! (not), etc.
- Result construction, i.e. to XML documents

Note: SQLX queries are SQL queries with XML extension which can be directly evaluated in the object-relational database to produce XML documents.

Note: The semantics in ORA-SS schema can be used to optimize Twig Pattern Query and XML Keyword Query processing and remove redundant answers.
Applications of ORA-SS Model

Translating Relational Schema into ORA-SS Schema

Translation from relational schema to ORA-SS schema diagram is divided into the following two steps:

Step 1. Identify various inherent semantics and implicit structure in the relational schema. This step is known as **semantic enrichment**.

Step 2. Translate semantically enriched relational schema to ORA-SS schema diagram according to a set of **translation rules**.
Applications of ORA-SS Model

Translating Relational Schema into ORA-SS Schema (cont.)

Step 1. Semantic Enrichment

Extra information needed for Semantic Enrichment:

- FDs and keys
- Inclusion dependencies
- Semantic dependencies

Example:

EMPLOYEE(E#, ENAME, JOINDATE, D#)

JOINDATE is functionally dependent on only E#.

Assuming JOINDATE refers to the date on which an employee assumes duty with the department. We say that

JOINDATE is semantically dependent on \{E#, D\}
Applications of ORA-SS Model

Translating Relational Schema into ORA-SS Schema (cont.)

Semantic Enrichment using SD together with FD and IND FDs and keys

To identify:

- **Object relations** and **object attributes** that represent regular and weak entity types, and their properties.

- **Relationship relations** and **relationship attributes** that represent various relationship types such as binary, n-ary, recursive and ISA (inheritance), and their properties.

- **Mix-type relations**: We need to split mix-type relations into object relations and relationship relations.

- **Fragments** of object relations or relationship relations that represent multi-valued attributes of object types or relationship types.

- **Cardinality** constraints
Applications of ORA-SS Model

Translating Relational Schema into ORA-SS Schema (cont.)

Example: An original relational schema

<table>
<thead>
<tr>
<th>TABLE</th>
<th>COLUMNS</th>
</tr>
</thead>
<tbody>
<tr>
<td>COURSE</td>
<td>CODE, TITLE</td>
</tr>
<tr>
<td>DEPT</td>
<td>D#, DNAME</td>
</tr>
<tr>
<td>STUDENT</td>
<td>S#, DEGREE</td>
</tr>
<tr>
<td>TUTORIAL</td>
<td>T#, DAY, TIME, ROOM</td>
</tr>
<tr>
<td>HOBBIES</td>
<td>S#, HOBBY</td>
</tr>
<tr>
<td>STUDENTDEPT</td>
<td>S#, D#</td>
</tr>
<tr>
<td>C_S</td>
<td>CODE, S#, GRADE</td>
</tr>
<tr>
<td>ATTEND</td>
<td>CODE, T#, S#</td>
</tr>
<tr>
<td>CONSULTATION</td>
<td>CODE, S#, RECORD</td>
</tr>
</tbody>
</table>
Applications of ORA-SS Model

Translating Relational Schema into ORA-SS Schema (cont.)

The Semantically Enriched Schema

Object Relations:
COURSE (CODE, TITLE)
DEPT (D#, DNAME)
STUDENT (S#, DEGREE)
TUTORIAL (T#, DAY, TIME, ROOM)

Fragment of Object Relations
HOBBIES (S#, HOBBY)
HOBBIES[S#] ⊆ STUDENT[S#]

(Hobby is a multivalued attribute of Student object class).

Relationship Relations:
STUDENTDEPT (S#, D#)
C_S (CODE, S#, GRADE)
ATTEND (CODE, T#, S#)
ATTEND[CODE, S#] = C_S[CODE, S#]

Fragment of Relationship Relations
CONSULTATION (CODE, S#, RECORD)
CONSULTATION [CODE, S#] ⊆ C_S[CODE, S#]

(Record is a multivalued attribute of the C_S relationship type)
Applications of ORA-SS Model

Translating Relational Schema into ORA-SS Schema (cont.)

Step 2. Enriched Relational Schema to ORA-SS Schema Translation

Objectives:

- Identify object classes and their attributes from object relations
- Identify relationship types and their attributes from relationship relations
- Identify hierarchical structure
- Generate ORA-SS schema
Applications of ORA-SS Model

Translating Relational Schema into ORA-SS Schema (cont.)

Translation Rules

1. **Object relation rule**: to translate object relations

2. **Relationship relation rule**: to translate relationship relations

3. **Combination rule**: to be applied to the result obtained from the application of object and relationship relation rules, and generate the final ORA-SS schema.
A possible derived ORA-SS schema diagram of the given relational schema.

Note: There are many other possible ORA-SS schemas with different hierarchical structures.
Applications of ORA-SS Model

ORA-Semantics based XML Keyword Search

Note: ORA-Semantics means Object-Relationship-Attribute Semantics
Current XML keyword search approaches:

**Lowest Common Ancestor (LCA) - based**

\[ Q = \{DB, John\} \]

(Common ancestor (CA))

(LCA is answer)

(Lowest CA (LCA))

(Matching node)
Applications of ORA-SS Model

ORA-Semantics based XML Keyword Search

Comparing structured query and keyword query

<table>
<thead>
<tr>
<th>Structured Search (e.g., XPath, XQuery)</th>
<th>Keyword Search (KWS) (keyword query)</th>
</tr>
</thead>
<tbody>
<tr>
<td>For $s1=\text{doc(SC-XMLDB.xml)//Student[Name/First=Bill]}$</td>
<td>Bill, John, course</td>
</tr>
<tr>
<td>For $s2=\text{doc(SC-XMLDB.xml)//Student[Name/First=John]}$</td>
<td></td>
</tr>
<tr>
<td>Where $s1/\text{Course/Code}=s2/\text{Course/Code}$</td>
<td></td>
</tr>
<tr>
<td>Return $s1/\text{Course}$</td>
<td></td>
</tr>
</tbody>
</table>

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

Unsatisfactory answers

- Meaningless answers e.g. $Q=\{\text{Jane, Kate}\}$
- Missing answers e.g. $Q=\{\text{Bill, John}\}$
- Duplicated answers e.g. $Q=\{\text{CS5201, DB}\}$
- Incomplete answers e.g. $Q=\{\text{DB, A}\}$
- Schema-dependent answers e.g. $Q=\{\text{Bill, John}\}$
Applications of ORA-SS Model

ORA-Semantics based **XML Keyword Search**

**Comparing structured query and keyword query**

<table>
<thead>
<tr>
<th>Structured Search (e.g., XPath, XQuery)</th>
<th>Keyword Search (KWS) (keyword query)</th>
</tr>
</thead>
<tbody>
<tr>
<td>• precise (+)</td>
<td>• unsatisfactory answers (-)</td>
</tr>
<tr>
<td>• expressive (+)</td>
<td>• not expressive (-)</td>
</tr>
<tr>
<td>• learn complex query languages (-)</td>
<td>• user friendly (+)</td>
</tr>
<tr>
<td>• need to know schema (-)</td>
<td>• users do not know schema (+)</td>
</tr>
</tbody>
</table>

For $s1=\text{doc(SC-XMLDB.xml)}/\text{Student}[\text{Name/First}=\text{Bill}]$
For $s2=\text{doc(SC-XMLDB.xml)}/\text{Student}[\text{Name/First}=\text{John}]$
Where $s1/\text{Course/Code}=s2/\text{Course/Code}$
Return $s1/\text{Course}$

How to have advantages of both structured search and KWS?
Applications of ORA-SS Model

ORA-Semantics based XML Keyword Search

Comparing structured query and keyword query

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

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

Keyword Search (KWS) (keyword query)

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

More satisfactory answers
More expressive queries
Reasons of the problems of LCA-based approaches

- LCA-based approaches do not have the concepts of object, object ID (OID)
  - Cannot distinguish object nodes and other nodes
    - Meaningless answers
  - Cannot discover object duplication
    - Duplicated answers

- Do not have concepts of relationship
  - Cannot distinguish object attributes and relationship attributes
    - Incomplete answers

- Only based on the hierarchical structure of data. However, data can be represented by different hierarchical structures.
  - Missing answers & Schema-dependent answers

To solve the above problems, must discover and use ORA-semantics.
In Summary

Existing XML KWS
- only rely on hierarchy
- ignore ORA-semantics
- duplicated
- incomplete
- schema-dependent

Our XML KWS
- exploit ORA-semantics
- answers
  - NOT missed
  - NOT meaningless
  - NOT duplicated
  - NOT incomplete

Applications of ORA-SS Model
ORA-Semantics based XML Keyword Search
Applications of ORA-SS Model

ORA-Semantics based XML Keyword Search

For more details, see the references on ORA-semantics based XML keyword search:

- Thuy Ngoc Le, Zhifeng Bao, Tok Wang Ling, Gillian Dobbie: Group-by and Aggregate Functions in XML Keyword Search. DEXA (1) 2014: 105-121.
Applications of ORA-SS

Summary

- Introduced ORA-SS model
- Briefly discussed topics using ORA-SS Model
  - Normal form ORA-SS schema diagram
  - Storage schema for ORA-SS/XML databases
  - ORA-SS/XML views
  - Evaluating XML queries on ORA-SS databases
  - Translating relational schema into ORA-SS schema
  - ORA-Semantics based XML Keyword Search
References


