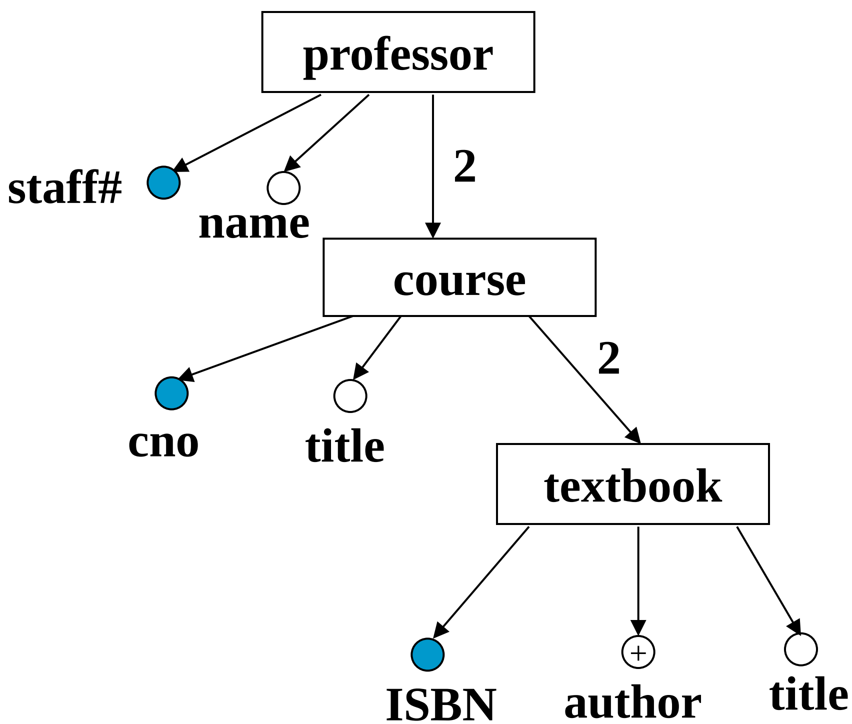
Applications of ORA-SS Model

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

Normal Form (NF) ORA-SS Schema Diagram

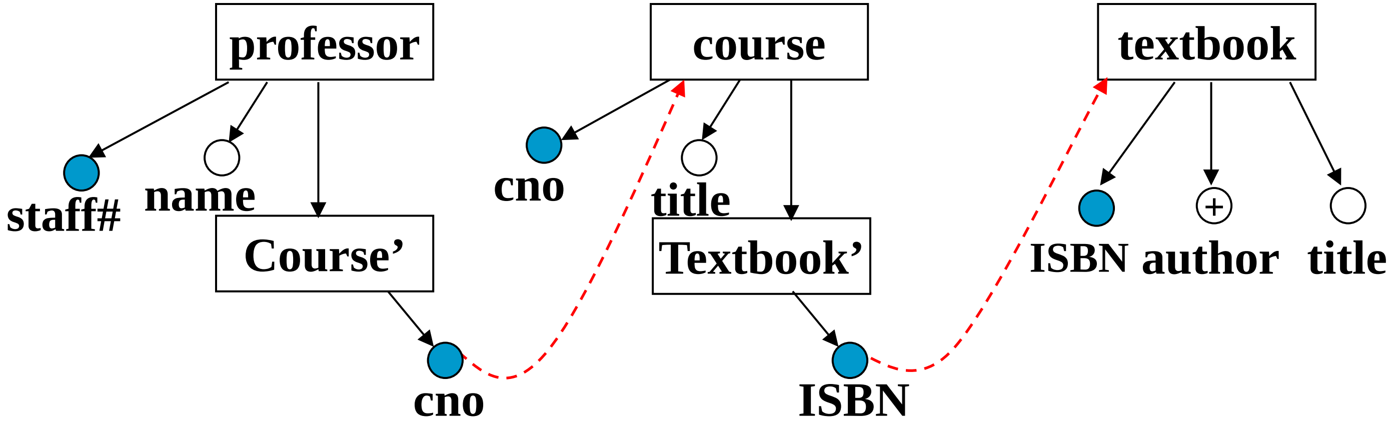
- The two binary relationship types are many-to-many
- Schema may have a lot of redundant data. Why? Where?
- Update anomalies. Why?
- Normal Form schema is needed. What?



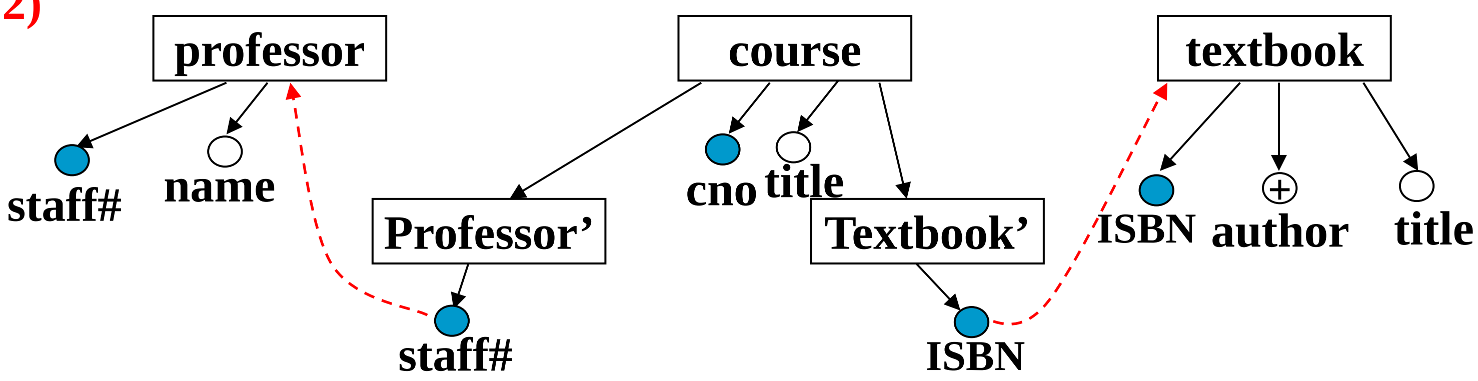
NF ORA-SS Schema Diagram (cont.)

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

(Solution 1)



(Solution 2)

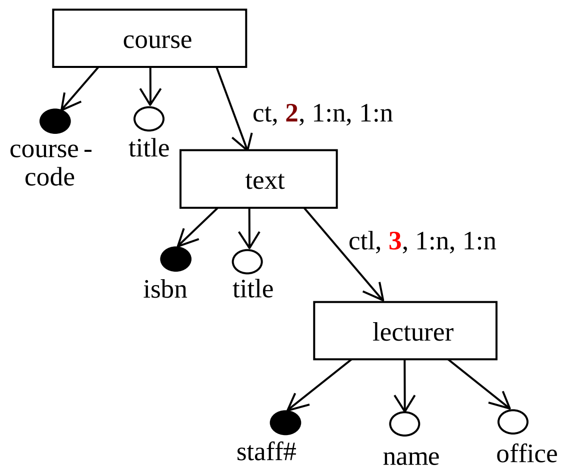


❖ **Q:** What are the problems of these two schemas/designs?
Symmetric queries cannot be processed equally efficiently.

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:

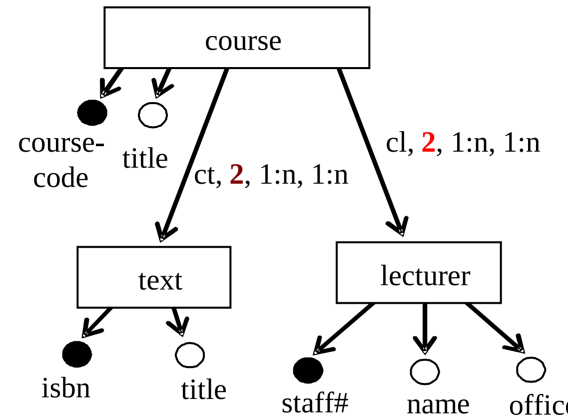
course-code $\rightarrow\rightarrow$ isbn | staff#



The relation for the relationship type ctl is:

ctl (<u>course-code, isbn, staff#</u>)

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:

ct (<u>course-code, isbn</u>)

cl (<u>course_code, staff#</u>)

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

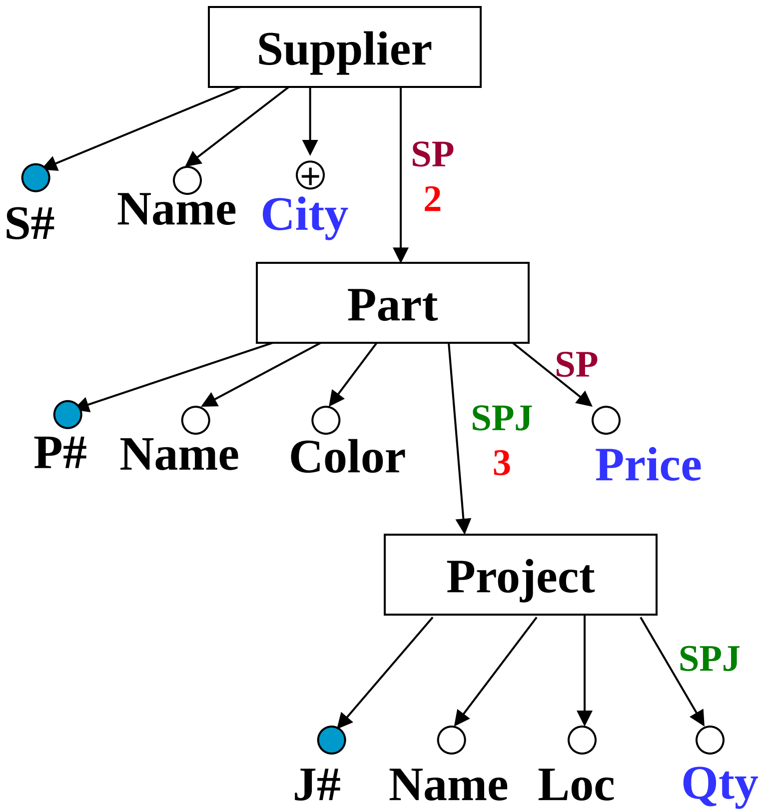
- ORA-SS reflects the nested structure of semi-structured data and store data in nested relations.

  - – less join in nested relations

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 8*i* *and newer versions*.

Storage Schema for ORA-SS/XML Databases



Object Relations

- Supplier (S#, Name, (City)*)

  Notation: (…)* indicates a **repeating group** in a nested relation; suppliers may have many cities.

  Part (P#, Name, color)

  Project(J#, Name, Loc)

Relationship relations

SP (S#, P#, price)

SPJ (S#, P#, J#, Qty)

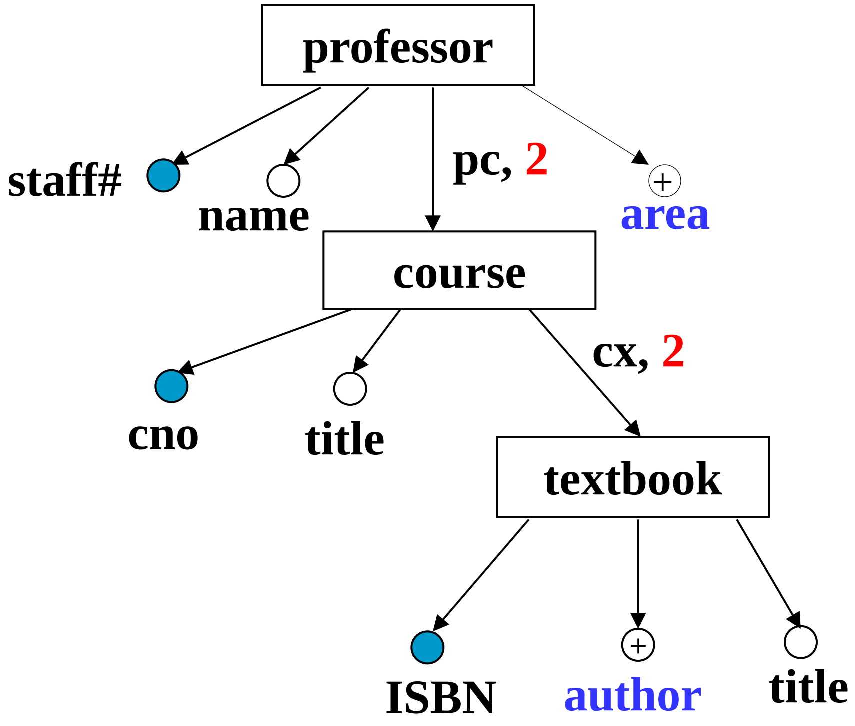
Constraints:

$$SPJ[S\#, P\#] \subseteq SP[S\#, P\#]$$

and other referential constraints.

Storage Schema for ORA-SS/XML Databases (cont.)

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] $\subseteq$ pc [cno]

and other referential constraints

Views

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

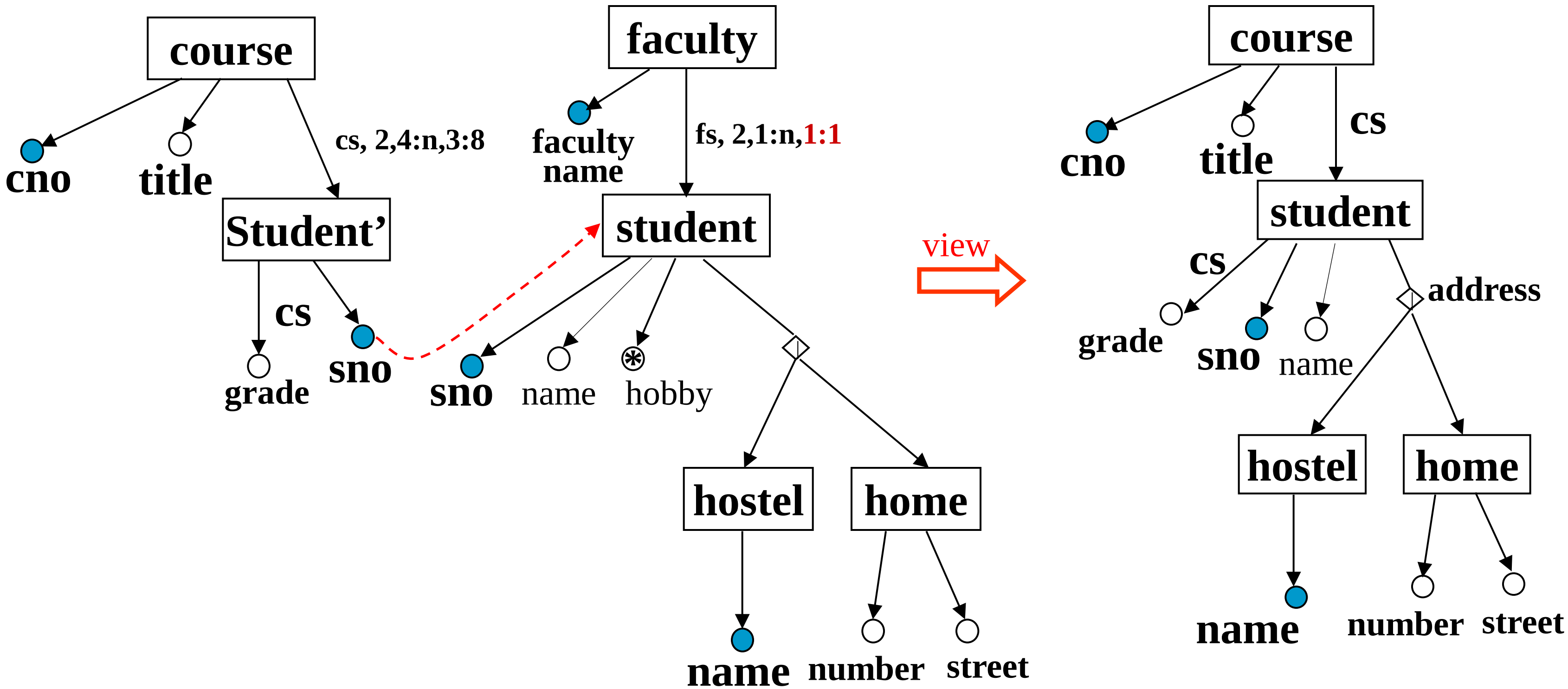


Fig. Referencing an object class in an ORA-SS schema diagram

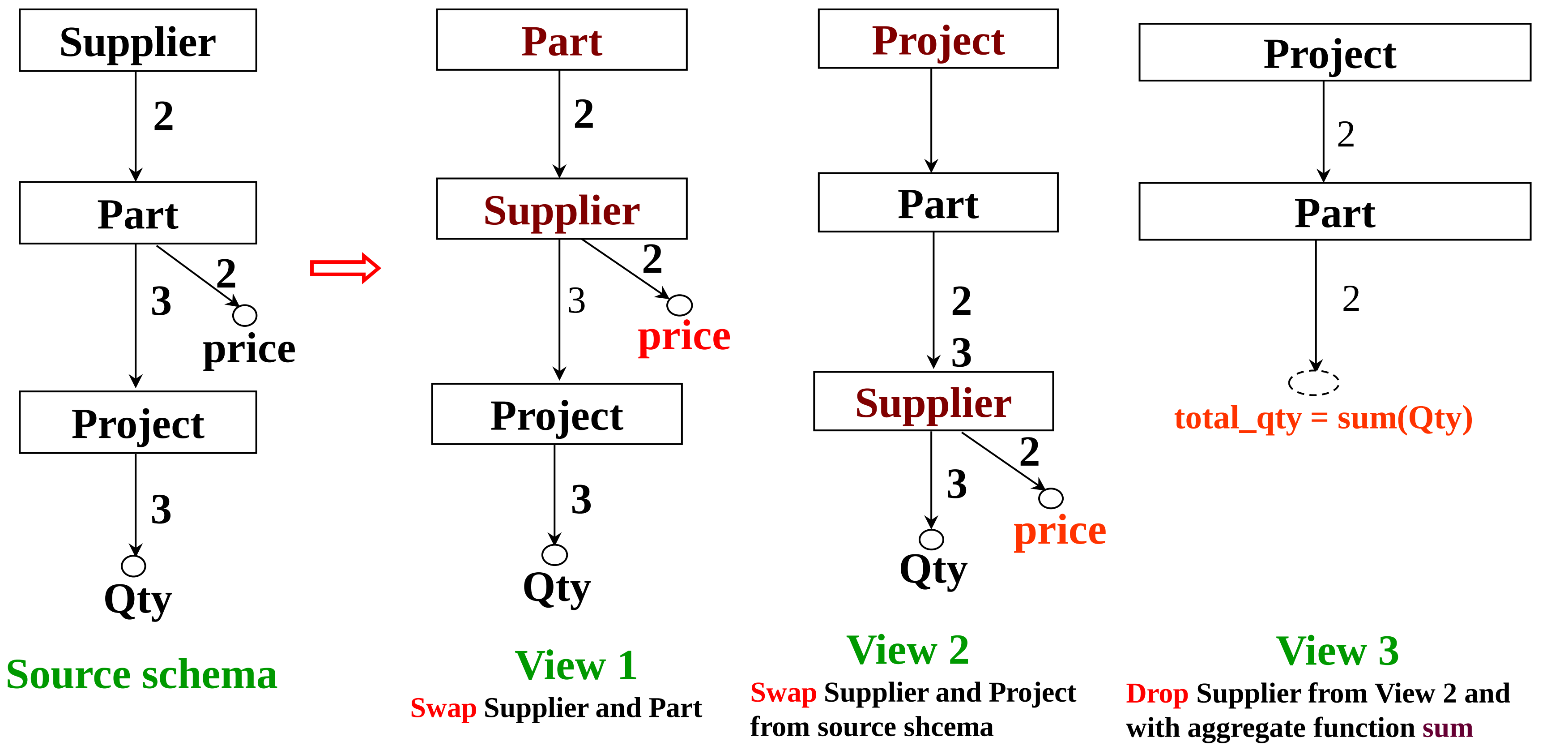
Fig. A **view** of the source schema

[5] Ya Bing Chen, Tok Wang Ling, Mong Li Lee, Automatic Generation of XQuery View Definitions from ORA-SS Views. ER 2003.
[6] Ya Bing Chen, Tok Wang Ling, Mong Li Lee: Automatic Generation of SQLX View Definitions from ORA-SS Views. DASFAA 2004

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.



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.

Views (cont.)

- 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

Views (cont.)

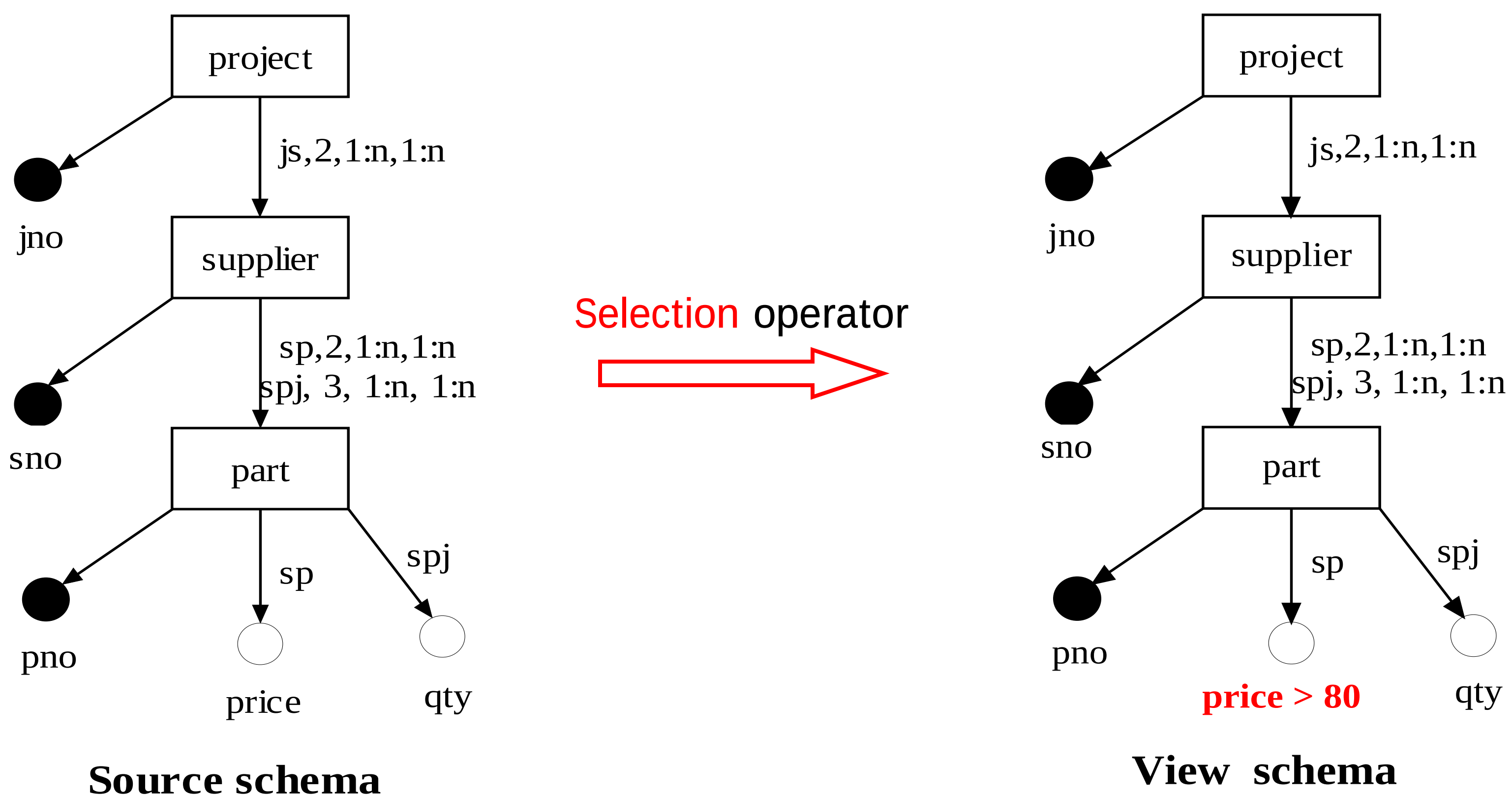
- Disadvantages of the main related work
  - Ignore semantic information in source data
    - For example, ignore the difference between object class, attribute and relationship in schema
  - Cannot check the validity of designed views
  - Difficult to use query languages to define views
    - Proprietary 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.

Views (cont.)

- 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

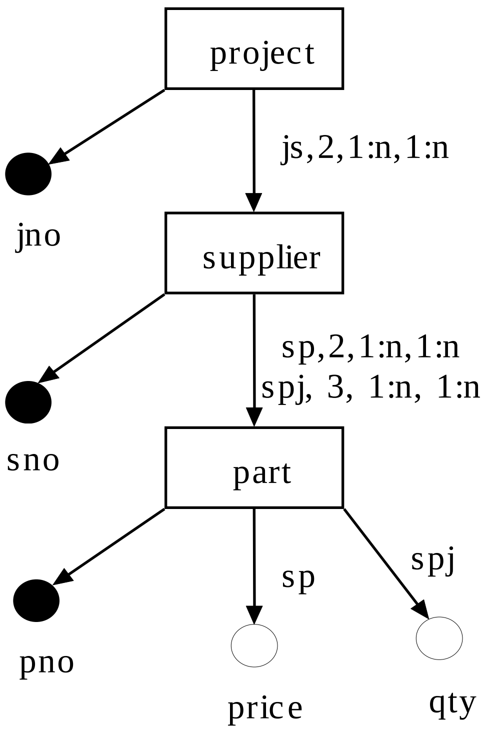
Views: Selection operator

- 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.

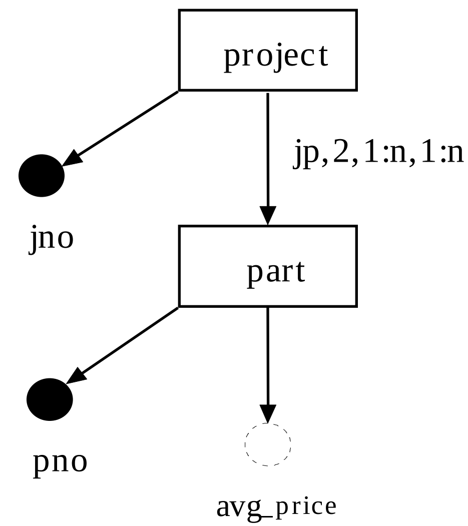


Views: Drop (projection) operator

- 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.



Source schema

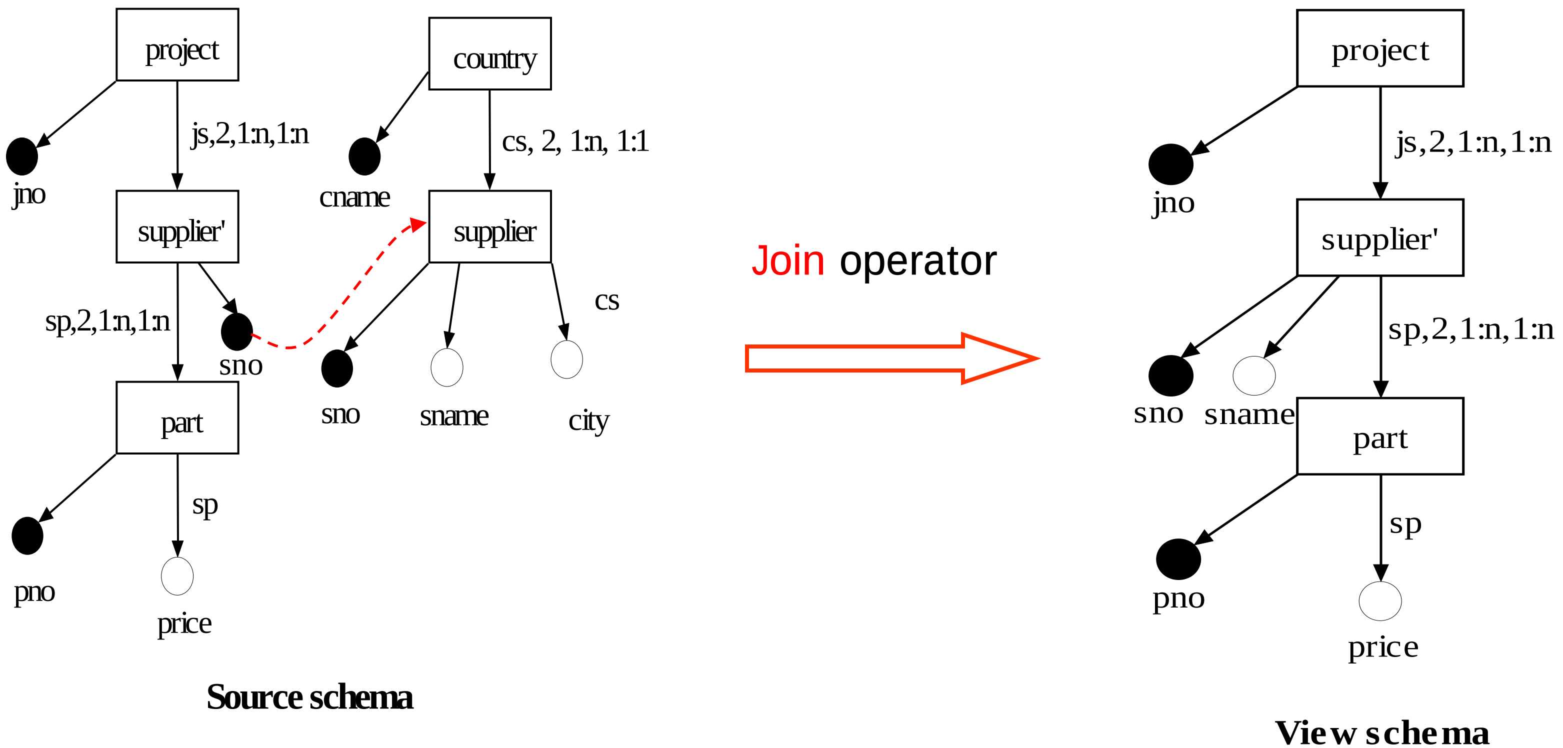


View schema

Note: We can have total_qty derived attribute below part also.

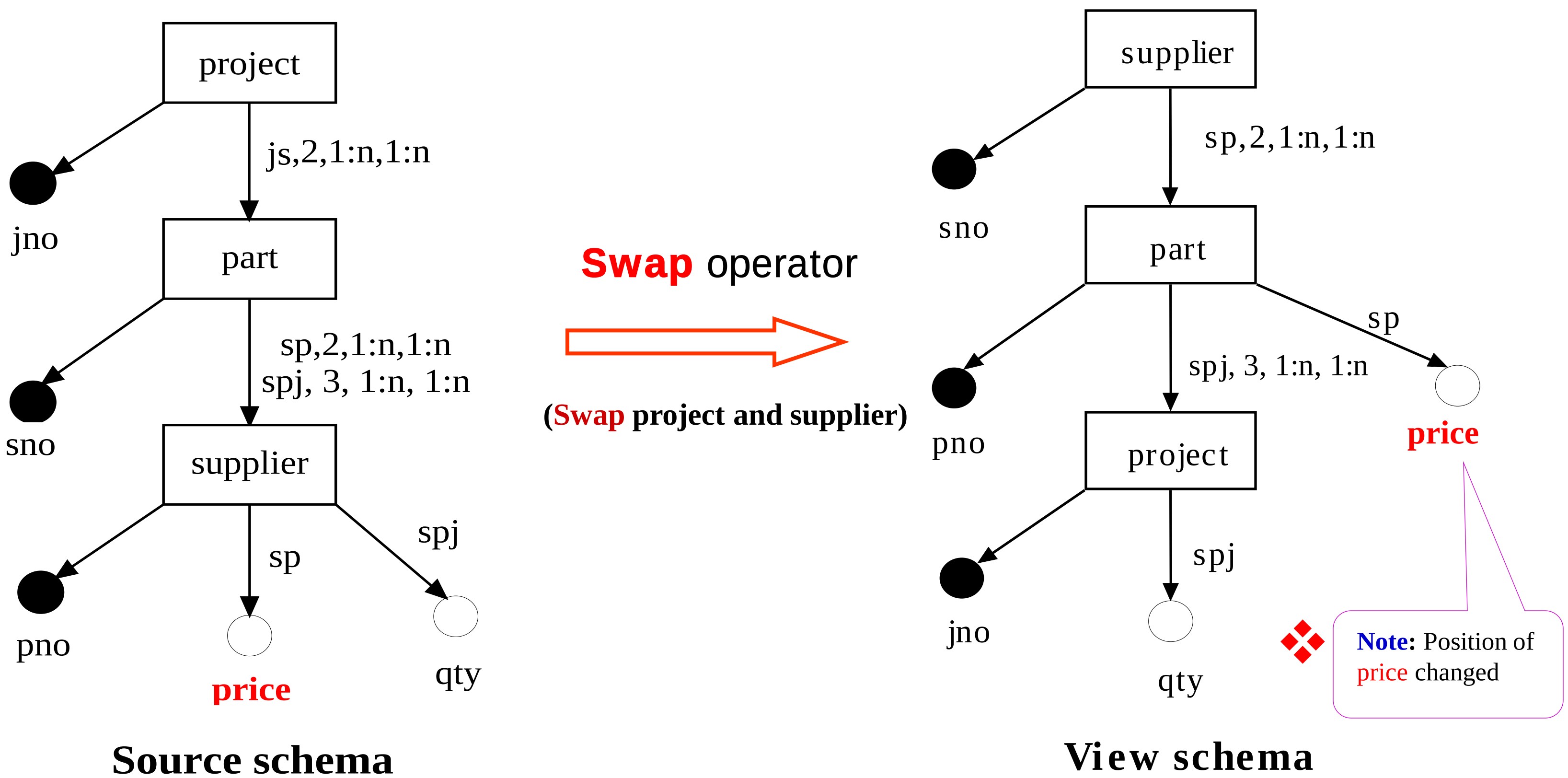
Views: Join operator

- 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.



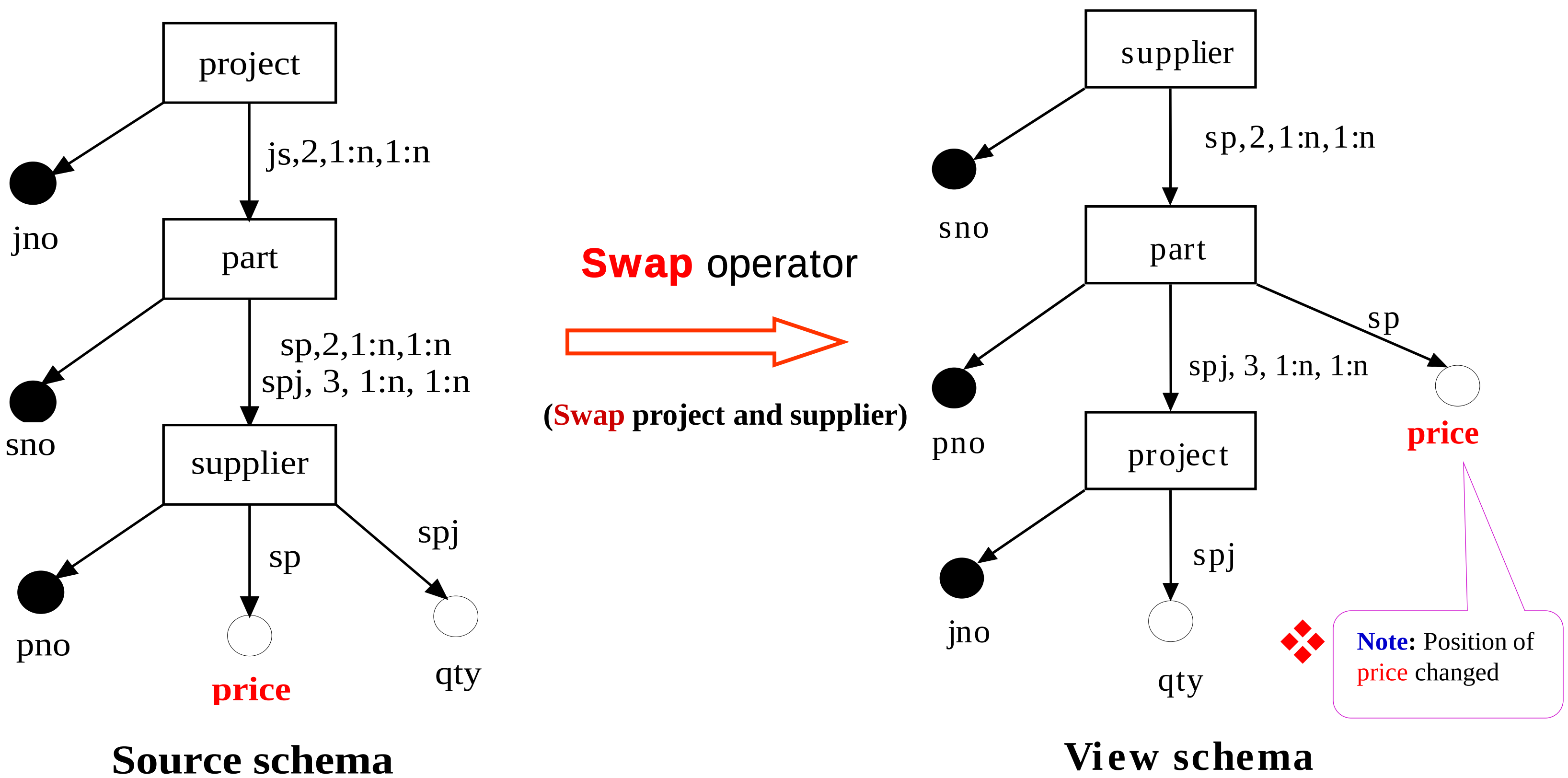
Views: Swap operator

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



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*.



Other Topics on Views

- – materialized views and maintenance
- – view update problem

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 **X**ML extension which can be directly evaluated in the object-relational database to produce XML documents.

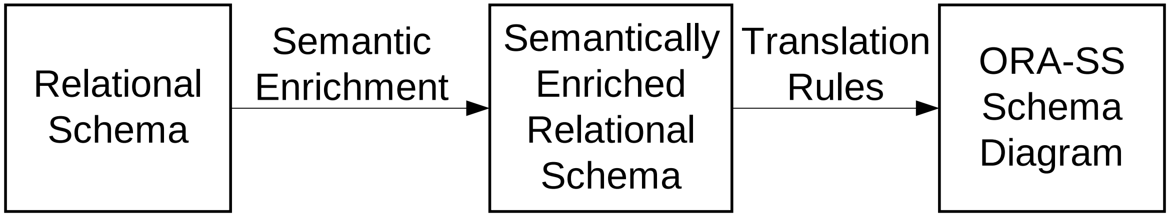
Note: The **semantics** in ORA-SS schema can be used to **optimize** Twig Pattern Query and XML Keyword Query processing and **remove redundant answers**.

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**.



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#}

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

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

Example: An original relational schema

COURSE (CODE, TITLE)

DEPT (D#, DNAME)

STUDENT (S#, DEGREE)

TUTORIAL (T#, DAY, TIME, ROOM)

HOBBIES (S#, HOBBY)

STUDENTDEPT (S#, D#)

C_S (CODE, S#, GRADE)

ATTEND (CODE, T#, S#)

CONSULTATION (CODE, S#, RECORD)

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)

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

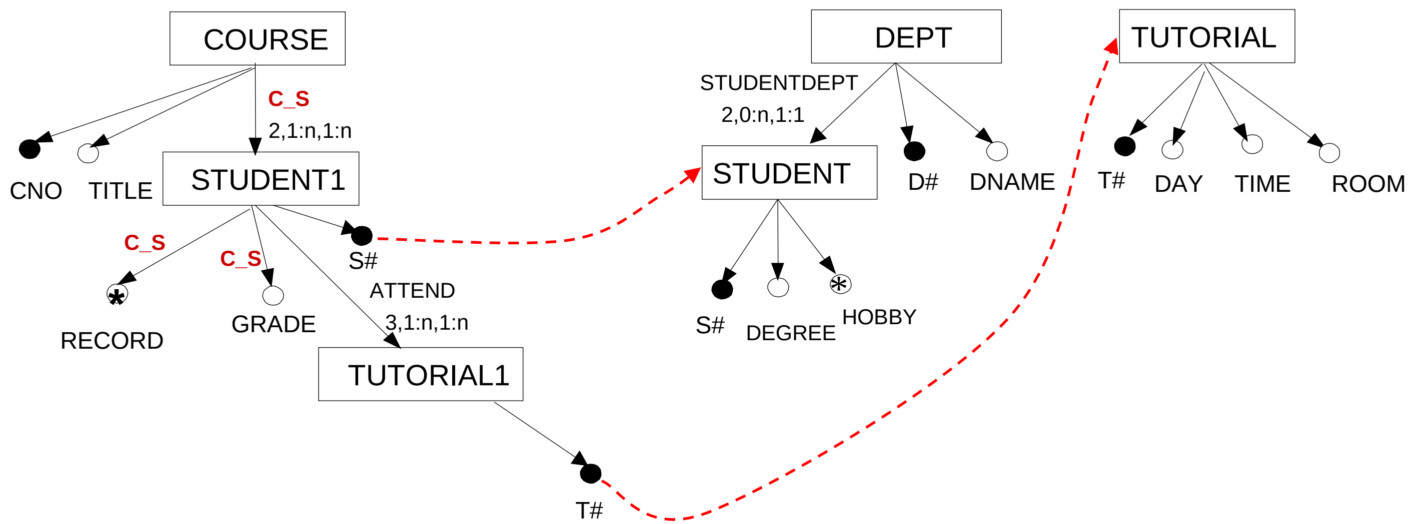
Translating Relational Schema into ORA-SS Schema (cont.)

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.

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

A possible derived ORA-SS schema diagram of the given relational schema.



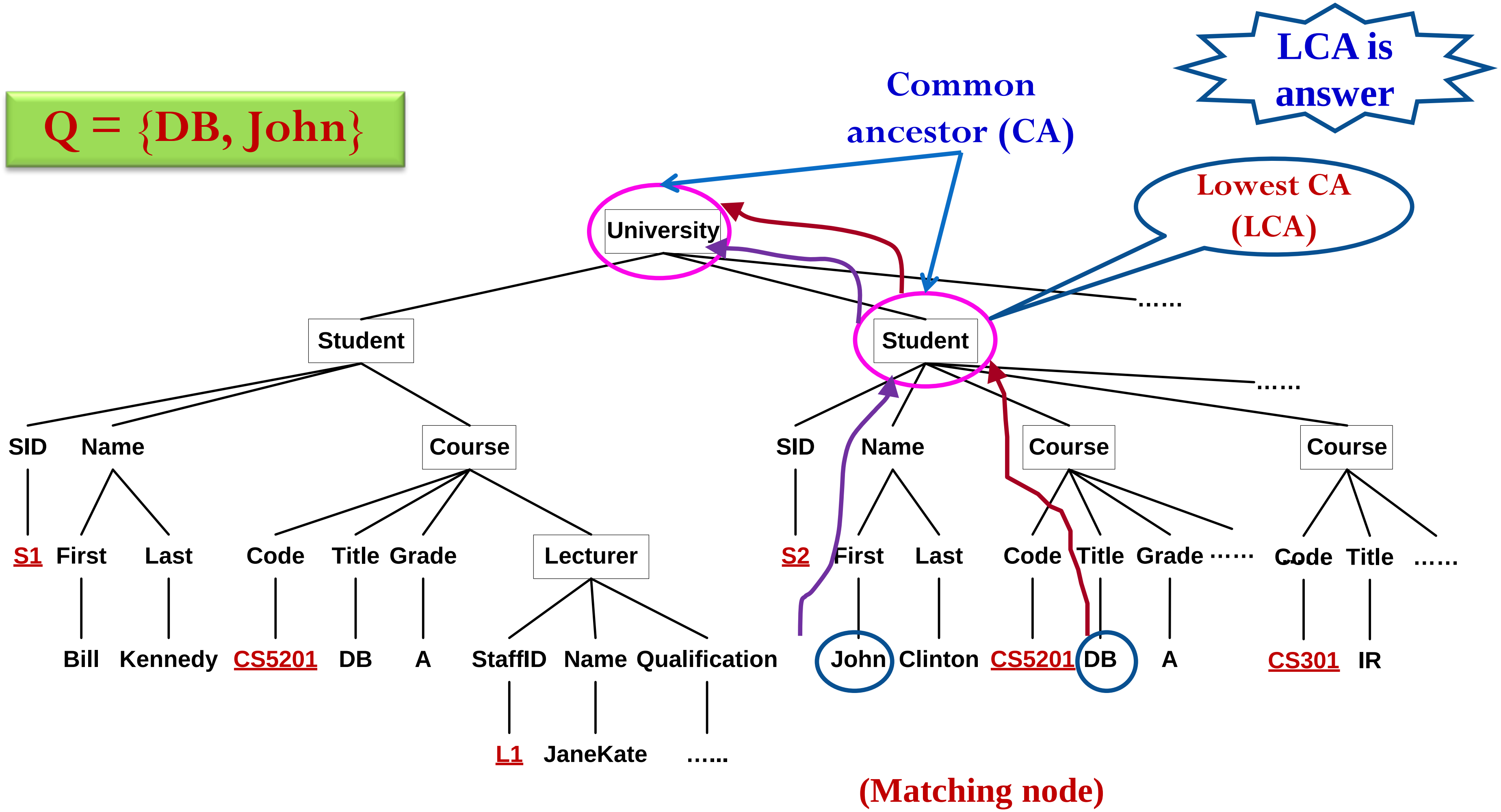
Note: There are many other possible ORA-SS schemas with different hierarchical structures.

ORA-Semantics based XML Keyword Search

Note: ORA-Semantics means Object-Relationship-Attribute Semantics

Current XML keyword search approaches :

Lowest Common Anscestor (LCA) - based



ORA-Semantics based XML Keyword Search

Comparing structured query and keyword query

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

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

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

Keyword Search (KWS) (keyword query)

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

Unsatisfactory answers

- **Meaningless answers** e.g. **Q={Jane, Kate}**
- **Missing answers** e.g. **Q={Bill, John}**
- **Duplicated answers** e.g. **Q={CS5201, DB}**
- **Incomplete answers** e.g. **Q={DB, A}**
- **Schema-dependent answers** e.g. **Q={Bill, John}**

ORA-Semantics based XML Keyword Search

Comparing structured query and keyword query

Structured Search (e.g., XPath, XQuery)	Keyword Search (KWS) (keyword query)

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

• precise (+) • expressive (+) • learn complex query languages (-) • need to know schema (-)	• unsatisfactory answers (-) • not expressive (-) • user friendly (+) • users do not know schema (+)

How to have advantages of both structured search and KWS?

ORA-Semantics based XML Keyword Search

Comparing structured query and keyword query

Structured Search (e.g., XPath, XQuery)	Keyword Search (KWS) (keyword query)

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

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

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

SEARCH → **Keyword SEARCH**

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

ORA-Semantics based XML Keyword Search

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

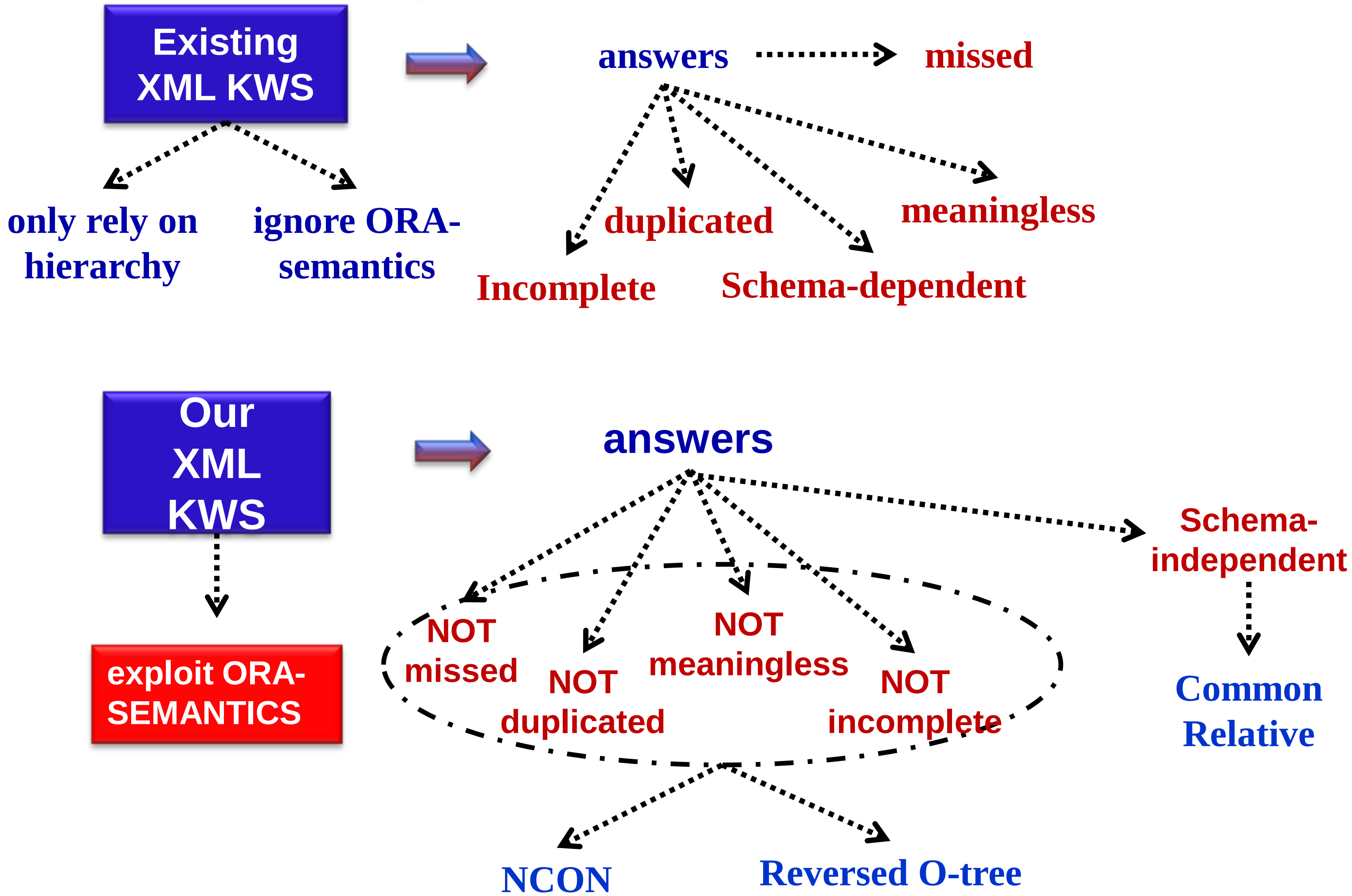
- Do not have concepts of relationship
  - → Cannot distinguish object attributes and relationship attributes
    - → Incomplete answers

- 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.

ORA-Semantics based XML Keyword Search

In Summary



ORA-Semantics based XML Keyword Search

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

- Thuy Ngoc Le, Tok Wang Ling, H. V. Jagadish, Jiaheng Lu: Object Semantics for XML keyword Search. DASFAA, 2014.
- Thuy Ngoc Le, Zhifeng Bao, TokWang Ling: Schema-independence in XML Keyword Search. ER, 2014.
- Thuy Ngoc Le, Zhifeng Bao, Tok Wang Ling, Gillian Dobbie: Group-by and Aggregate Functions in XML Keyword Search. DEXA (1) 2014: 105-121.

Summary

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

References

[1] Gillian Dobbie, Xiaoying Wu, Tok Wang Ling and Mong Li Lee. ORA-SS: An Object-Relationship-Attribute Model for Semistructured Data. Technical Report TR21/00, School of Computing, National University of Singapore, 2000.

[2] Wenyue Du, Mong Li Lee, Tok Wang Ling. XML Structures for Relational Data, in *Proceedings of the 2nd International Conference on Web Information Systems Engineering (WISE)* , IEEE Computer Society, Kyoto, Japan, December 2001.

[3] Mong Li Lee, Tok Wang Ling, Wai Lup Low. Designing Functional Dependencies for XML, in *VIII Conference on Extending Database Technology (EDBT)* , Prague, March 2002.

[4] Xiaoying Wu, Tok Wang Ling, Mong Li Lee, Gillian Dobbie. Designing Semistructured Databases Using the ORA-SS Model, in *Proceedings of the 2nd International Conference on Web Information Systems Engineering (WISE)* , IEEE Computer Society, Kyoto, Japan, December 2001.

[5] Ya Bing Chen, Tok Wang Ling, Mong Li Lee, Automatic Generation of XQuery View Definitions from ORA-SS Views. ER’2003, Chicago, October 2003, pp 158-171

[6] Ya Bing Chen, Tok Wang Ling, Mong Li Lee: Automatic Generation of SQLX View Definitions from ORA-SS Views. DASFAA 2004: 476-481

[7] Huayu Wu, Tok Wang Ling, Bo Chen: VERT: A Semantic Approach for Content Search and Content Extraction in XML Query Processing. ER 2007: 534-549

[8] Huayu Wu, Tok Wang Ling, Liang Xu, Zhifeng Bao: Performing grouping and aggregate functions in XML queries. WWW 2009: 1001-1010.

[9] Thuy Ngoc Le, Tok Wang Ling, H. V. Jagadish, Jiaheng Lu: Object Semantics for XML keyword Search. DASFAA, 2014.

[10] Thuy Ngoc Le, Zhifeng Bao, TokWang Ling. Schema-independence in XML Keyword Search. ER 2014.

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