EML Data Manager
Data Query Specification

Ben Leinfelder

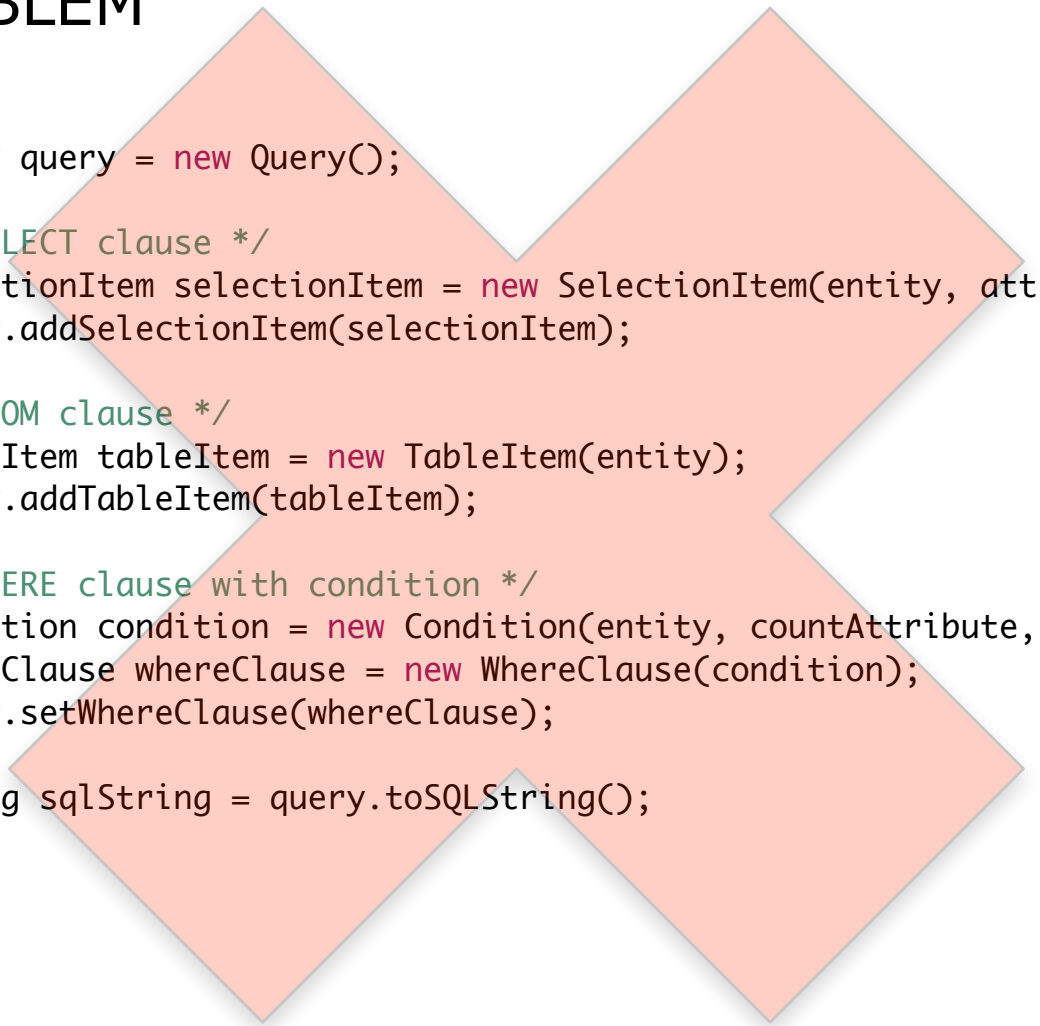
National Center for Ecological Analysis and Synthesis (NCEAS)
University of California Santa Barbara

DevLunch

August 6th, 2008

Query as Code

The PROBLEM

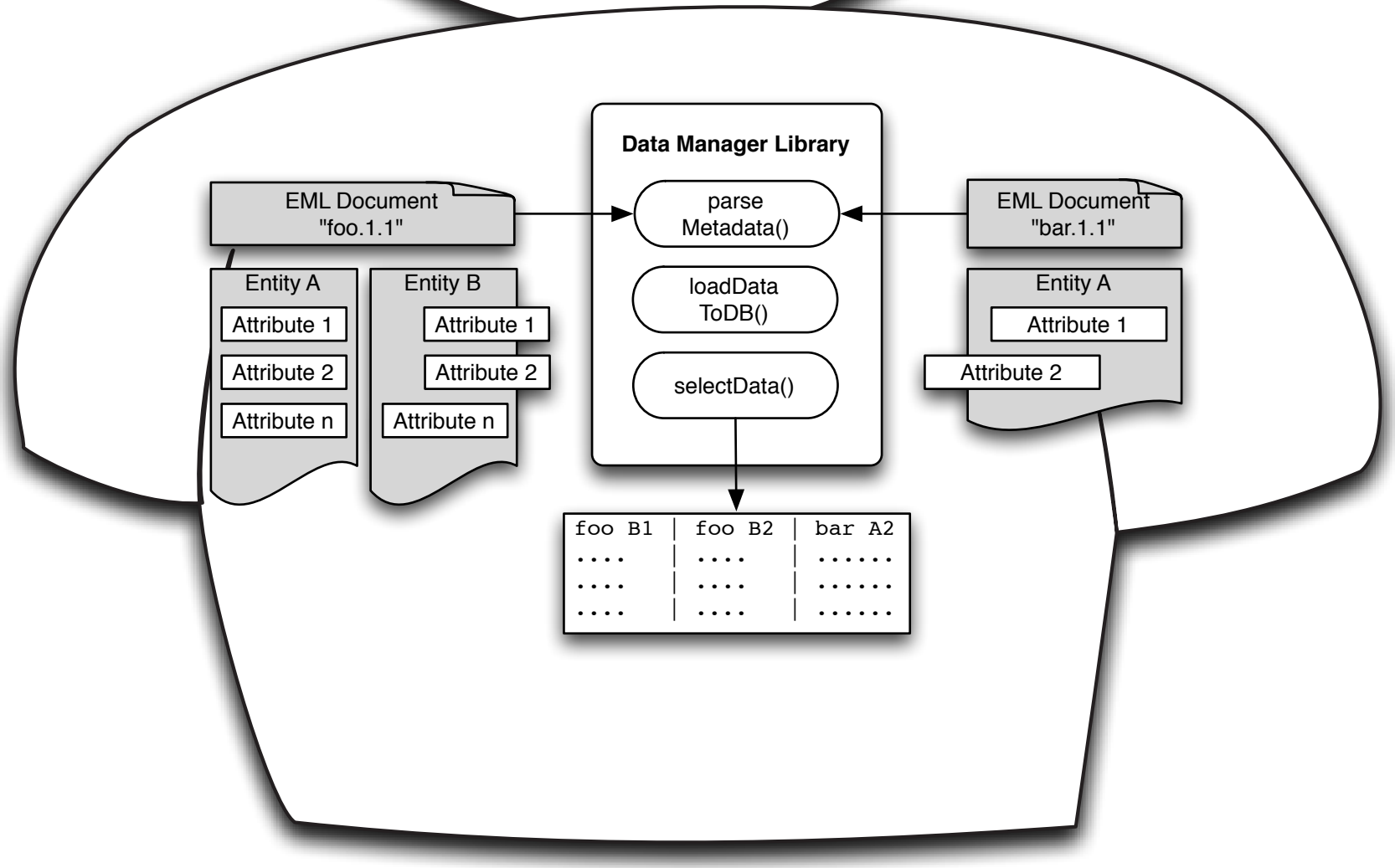


```
Query query = new Query();

/* SELECT clause */
SelectionItem selectionItem = new SelectionItem(entity, attribute);
query.addSelectionItem(selectionItem);

/* FROM clause */
TableItem tableItem = new TableItem(entity);
query.addTableItem(tableItem);

/* WHERE clause with condition */
Condition condition = new Condition(entity, countAttribute, operator, value);
WhereClause whereClause = new WhereClause(condition);
query.setWhereClause(whereClause);

String sqlString = query.toSQLString();
```

Data Manager API

- Current Use
  - Queries are created with Java objects
  - Assembled from Datapackages that have lists of:
    - Entities (tables) that have lists of:
      - Attributes (columns)
  - Joins, where clauses, subqueries, etc... supported

Anatomy of the Query



Extended Data Manager API

- Data Query Specification
  - Use XML document to express the desired query
    - dataquery.xsd (draft version)
  - Similar to Metacat pathquery that is used when searching for metadata documents
  - SAX parser for translation:
    - XML -> Java code
    - nested items are harder than you'd think!

NCEAS

Extend because...

Datamanager + Metacat

- Enhance Metacat API
  - Select data directly from Metacat
  - Request:
    - action=dataquery
    - dquery=<dataquery XML instance>
    - qformat=[csv|zip]
  - Response:
    - csv or zip containing the tabular data results

Data Query Features

- Selection
  - choose multiple Attributes from:
    - same parent Entity
    - parent's sibling Entity
    - Entity within a separate Datapackage

- Concatenation
  - stack similar data sets and query results
  - UNION and UNION ALL support

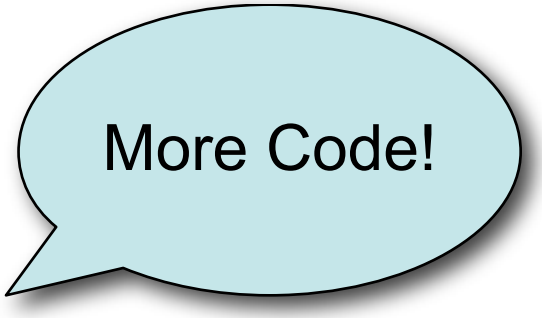
Data Query Features

- Conditions
  - simple:
    - <Entity>.<Attribute> <operator> <value>
  - join:
    - <Entity>.<Attribute> = <Entity>.<Attribute>
  - subquery:
    - <Entity>.<Attribute> <operator> <subquery>

=, >, <, >=, <=, !=

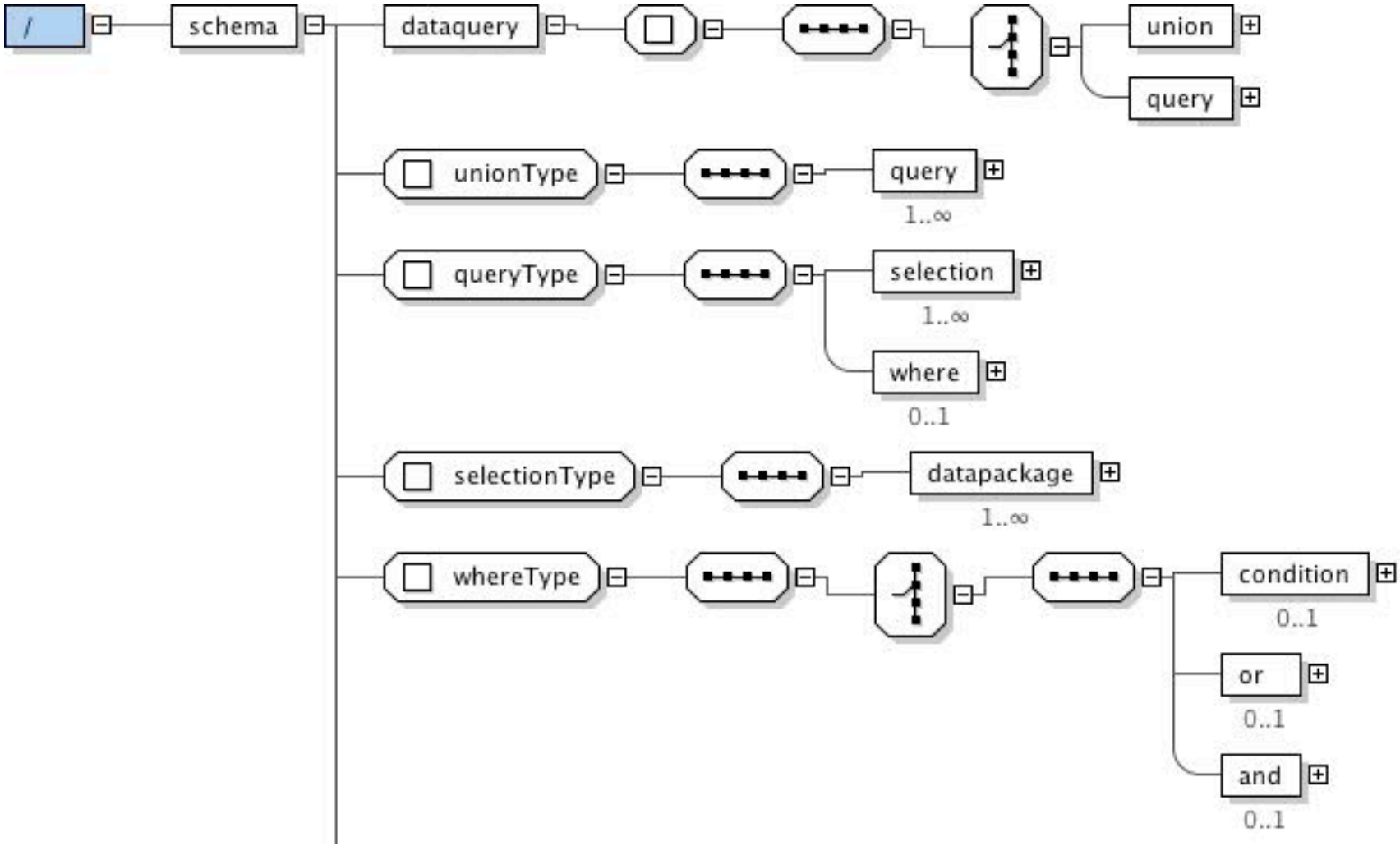
[IN\NOT IN]

Data Query Schema

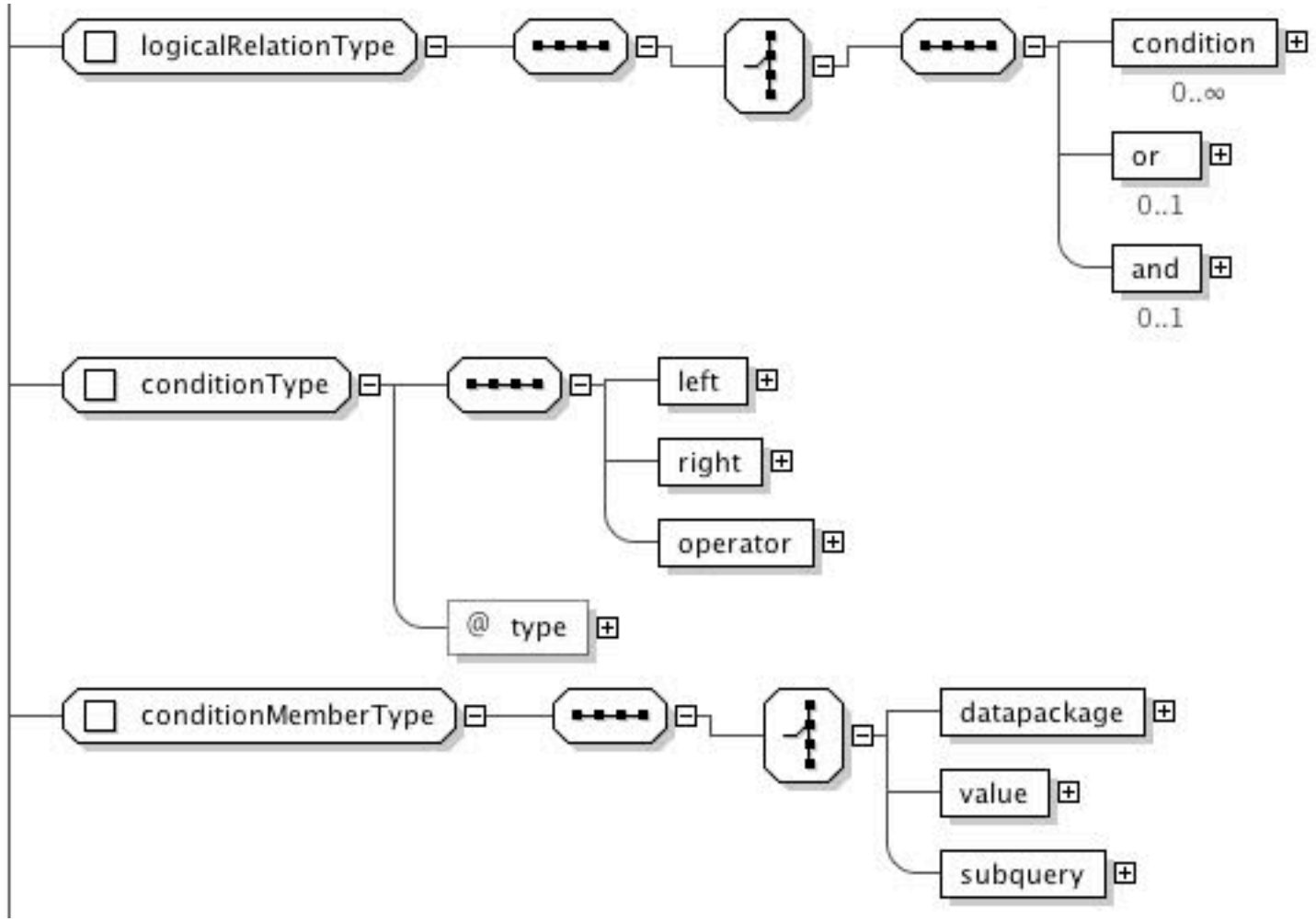



```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns="eml://ecoinformatics.org/dataquery"
    targetNamespace="eml://ecoinformatics.org/dataquery">

    <xs:element name="dataquery">
        <xs:complexType>
            <xs:sequence>
                <xs:choice>
                    <xs:element name="union" type="unionType"/>
                    <xs:element name="query" type="queryType"/>
                </xs:choice>
            </xs:sequence>
        </xs:complexType>
    </xs:element>.....................................
```

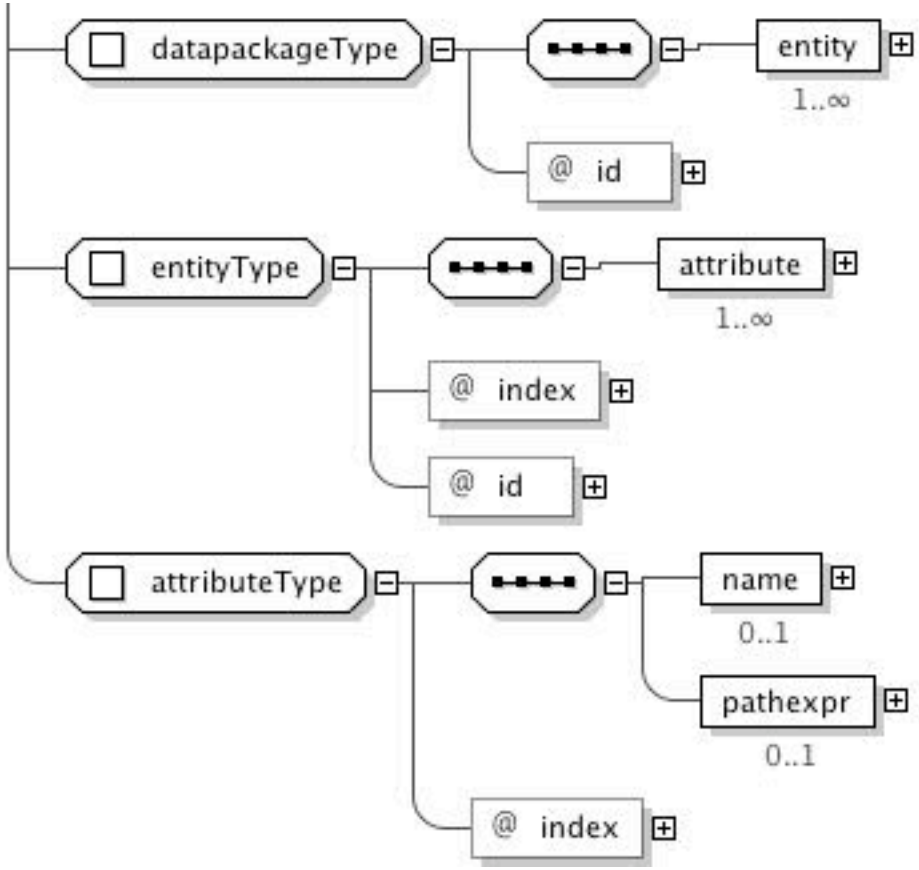
Query Details



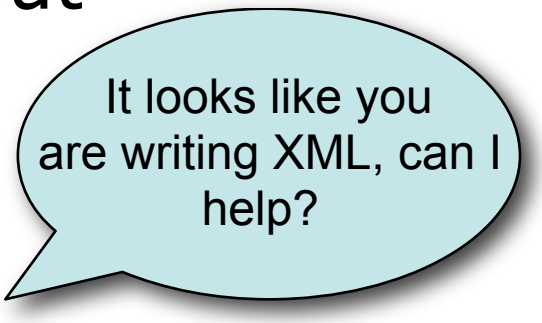
Condition Details



Data Structure Details



Example XML input




```
<?xml version="1.0"?>
<dq:dataquery>
        <union>
                <!-- first query in UNION -->
                <query>
                        <selection>
                                <datapackage id="tao.1.1">
                                        <entity index="0">
                                                <attribute index="0"/>
                                                <attribute index="1"/>
                                        </entity>
                                </datapackage>
                        </selection>
                        <where>
                                <condition type="condition">
                                        <left>
                                                <datapackage id="tao.1.1">
                                                        <entity index="0">
                                                                <attribute index="0"/>
                                                        </entity>
                                                </datapackage>
                                        </left>
                                        <right>
                                                <value>1</value>
                                        </right>
                                        <operator>=</operator>
                                </condition>
                        </where>
                </query>.....................
```

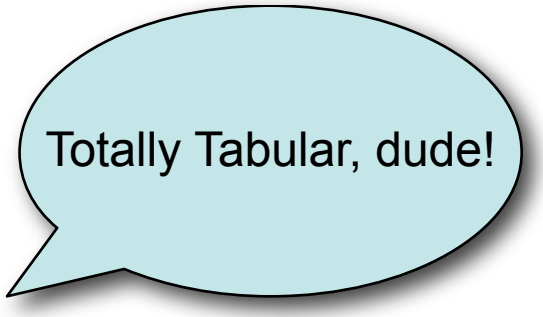
Example SQL output

```
<!--
        This sample XML is parsed by the Data Manager to produce the following SQL
        (formatted for improved legibility):

        SELECT Datos_Meteorologicos.DATE,Datos_Meteorologicos.TIME
        FROM Datos_Meteorologicos  where Datos_Meteorologicos.DATE = '1'

        UNION ALL

        SELECT Datos_Meteorologicos.DATE,Datos_Meteorologicos.TIME
        FROM Datos_Meteorologicos
        where  Datos_Meteorologicos.DATE = Datos_Meteorologicos.TIME
        AND Datos_Meteorologicos.TIME = 'happy'
        AND Datos_Meteorologicos.TIME IN (
                  SELECT Datos_Meteorologicos.TIME FROM Datos_Meteorologicos
                  where  Datos_Meteorologicos.TIME = '100'
                  AND Datos_Meteorologicos.TIME = '200'
        )
        AND (
                  Datos_Meteorologicos.DATE = '1234' OR Datos_Meteorologicos.DATE = '5678'
        ) ;

-->
```

Example Data output

```
SELECT
Datos_Meteorologicos.DATE,Datos_Meteorologicos.TIME,Datos_Meteorologicos.T_AIR
FROM Datos_Meteorologicos
where Datos_Meteorologicos.T_AIR LIKE '%15%';
```

date	time	t_air
2001-01-01	00:00:00	0001-01-01 00:00:00 BC	15
2001-01-01	00:00:00	0001-01-01 00:00:00 BC	15.9
2001-01-01	00:00:00	0001-01-01 00:00:00 BC	15.6
2001-01-01	00:00:00	0001-02-01 00:00:00 BC	15.2
2002-01-01	00:00:00	0001-01-01 00:00:00 BC	15.5
2002-01-01	00:00:00	0001-01-01 00:00:00 BC	15.2
2003-01-01	00:00:00	0001-01-01 00:00:00 BC	15.3
2003-01-01	00:00:00	0001-01-01 00:00:00 BC	15.1
2004-01-01	00:00:00	0001-09-01 00:00:00 BC	15.2



Thanks!

- Email
  - leinfelder@nceas.ucsb.edu
  -