C

Robert Hoover

Converting databases into xml documents

CPSC4660

Contents

[Introduction 2](#_Toc373138847)

[Phase One – Creation of the MySQL database 2](#_Toc373138848)

[Database Schema 3](#_Toc373138849)

[Phase Two – Creation of the XML and XSD files 4](#_Toc373138850)

[XML File 4](#_Toc373138851)

[XSD File 4](#_Toc373138852)

[Phase 3 Query and Conversion Time Calculations 5](#_Toc373138853)

[Conclusion 6](#_Toc373138854)

# Introduction

The 4660 Project I chose was based on an article by Francis Pang, Joseph Fong, and Chris Bloor. In the article the researchers described a method of reverse engineering a RDB into an EER model then forward engineering the EER model to XML. My goal was to be able to convert directly from a MySQL database into a useable XML document. Then my goal was to run queries on both the XML document and the MySQL database and compare query times. Issues arose when trying to create relationships using the given information from the article. On completion I was only able to map entities from the MySQL database to elements in the XML schema. I was also able to map relationships to references.

# Phase One – Creation of the MySQL database

In order to complete the project I had to recreate the database from the schema provided in the article. However, there was some inconsistencies in the table design provided. According to the SQL schema, a department contains a department\_id which makes sense, but according to the schema, it is a foreign key that references both region\_a and region\_b which is impossible in a SQL database. This format would work in XML as shown below.

<region\_a>

<department>

<trip>

<car\_rental>

<car>

<people>

</car\_rental>

</region\_a>

<region b>

<department>

<trip>

<car\_rental>

<car>

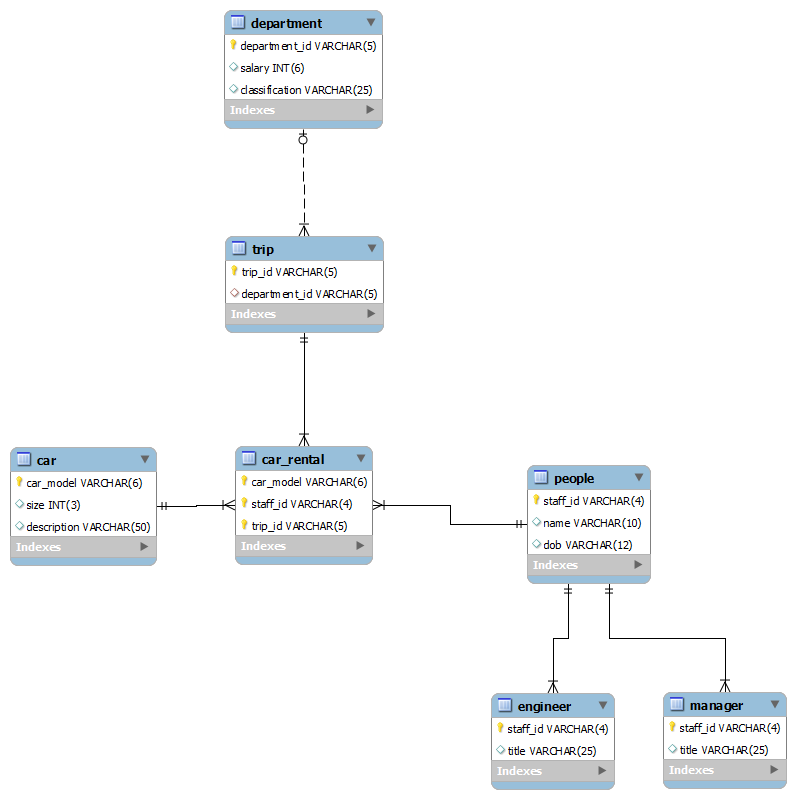
<people>

</car\_rental>

</region b>

In saying this, in my opinion it would be impossible to directly translate this information to XML without creating the SQL schema beforehand. Briefly in the article methods of creating the schema are explained. Knowledge Discovery in Database is one method of extracting useful data semantics. The decision I made was to modify the schema slightly by simply putting the region\_a and region\_b tables into the department table. The figure on the following page shows this modification as well as the database schema that was used for this project.

## Database Schema



# Phase Two – Creation of the XML and XSD files

## XML File

Once I had created the MySQL database using the previously mentioned schema, I started writing the code to create an XML document from it. This was a fairly simple task of finding the table names then selecting all attributes from each of the tables. This could have been more complex if I had discovered methods of extracting required information from the database to recreate a more complex schema.

## XSD File

When beginning with the schema document I researched into the semantics used on the research paper. After spending a few hours trying to locate the XML schema language they used I decided to just use the W3C standard that is popular today. I spent almost half of my time on this project researching the XSD data format as I had little to no experience using it before. Once I thought I had enough information, I began writing the code to convert the MySQL database into a XML schema document. I began with mapping entities to elements as suggested in the paper. This was a fairly simple step as it was similar to how you handle converting the database into an XML document.

Once I had created elements I moved onto creating relationships. This has two parts in a XSD file. First you must create keys for all of the tables using the <key> tags. Second is the use of the <keyref> tag, which actually creates the relationships between two tables. By first writing to the file directly I was able to determine the correct formatting that needed to be used in order for the XSD to validate. Once I knew the format, I wrote the PHP program to automatically generate the tags. In order to generate foreign keys, I was able to query the database information tables located in the MySQL information\_schema. In order to partially deal with cardinality the maxOccurs attribute is used; however, this only sets the maxOccurs to the table elements to infinite and does not deal with the possibility that an XML attribute can be mapped to as shown below.

Program Output:

<car\_rental>

<car\_model>mz-18</car\_model>

<staff\_id>a002</staff\_id>

<trip\_id>t0001</trip\_id>

</car\_rental>

<car\_rental>

<car\_model>r-023</car\_model>

<staff\_id>b004</staff\_id>

<trip\_id>t0001</trip\_id>

</car\_rental>

Article Output

<trip>

<trip\_id>t0001</trip\_id>

<car\_rental>

<car\_model>mz-18</car\_model>

<staff\_id>a002</staff\_id>

</car\_rental>

<car\_rental>

<car\_model>r-023</car\_model>

<staff\_id>b004</staff\_id>

</car\_rental>

</trip>

# Phase 3 Query and Conversion Time Calculations

Once I had a program that was writing both a valid XML document and XSD I began to write simple queries to test the query times of both. I decided first on using a simple query on people since it would likely be a large table in practice. Initially people only has 8 entries. Because of this an XPath query on the XML was actually faster than querying the database. However, once more data was introduced, querying the database became more efficient.

|  |  |  |
| --- | --- | --- |
| Number of Entries | XPath time | SQL time |
| 8 | 0.001 seconds | 0.002 seconds |
| 1000 | 0.005 seconds | 0.003 seconds |
| 10000 | 0.049 seconds | 0.017 seconds |

I also recorded the conversion times depending on the number of entries in the people table. The creation of the XSD file does not change based on the number of entries, only based on the number of tables, and keys. However, the time to convert to the XML document does change based on the number of entries as shown in the below table.

|  |  |
| --- | --- |
| Number of Entries | Time to Convert |
| 8 | 0.016 seconds |
| 10000 | 0.312 seconds |

# Conclusion

In conclusion I felt that my program achieved my goal of creating a useable XML file and XSD file. The XSD file correctly validates that key constraints and relationships are held and that is the primary concern of data integrity. The XML file can be queried easily using XPath and even though it is a simple representation of the data, it still is logical and representative.