**DATA PIPELINE: MODELLING OF XML DATABASE OF A HEALTH INSURANCE COMPANY**

**ADEOGUN ASHIAT ASHAKE**

PROGRAMME: MSc APPLIED DATA ANALYTICS

1. ABSTRACT

This report presents a summary of the modelling and implementation of an XML database of a health insurance company. The model follows instructions given in the assignment description. The tasks performed include – selection of a modeling framework for the database, creation of the XML database, generation and validation of the schema using freeformater.com, creation and writing of queries in XSL and transformation to XSLT with output formats in HTML. Other queries are written using JSON and JSON-schema. An online tool – jsonschemavalidator.net was used to validate the JSON against schema. All queries were successfully evaluated.

1. MODELLING OF XML Database

*Question: Describe your modelling choices in natural language. Think about the advantages and disadvantages of your choices and discuss them.*

The selected model, i.e., structure for the XML database of an insurance company is shown in Figure 1.0. As shown in the figure, the model has a tree structure with Electronic Medical Records (EMR) as the root node with each employee/client as child nodes. Thus, the root node has its number of children equal to the total number of clients. For each client node, we have a total of 3 child nodes containing information about employee personal information, claims and payments. The node with employee personal information contains entries such as ID, name, address, age, DoB, gender, etc.

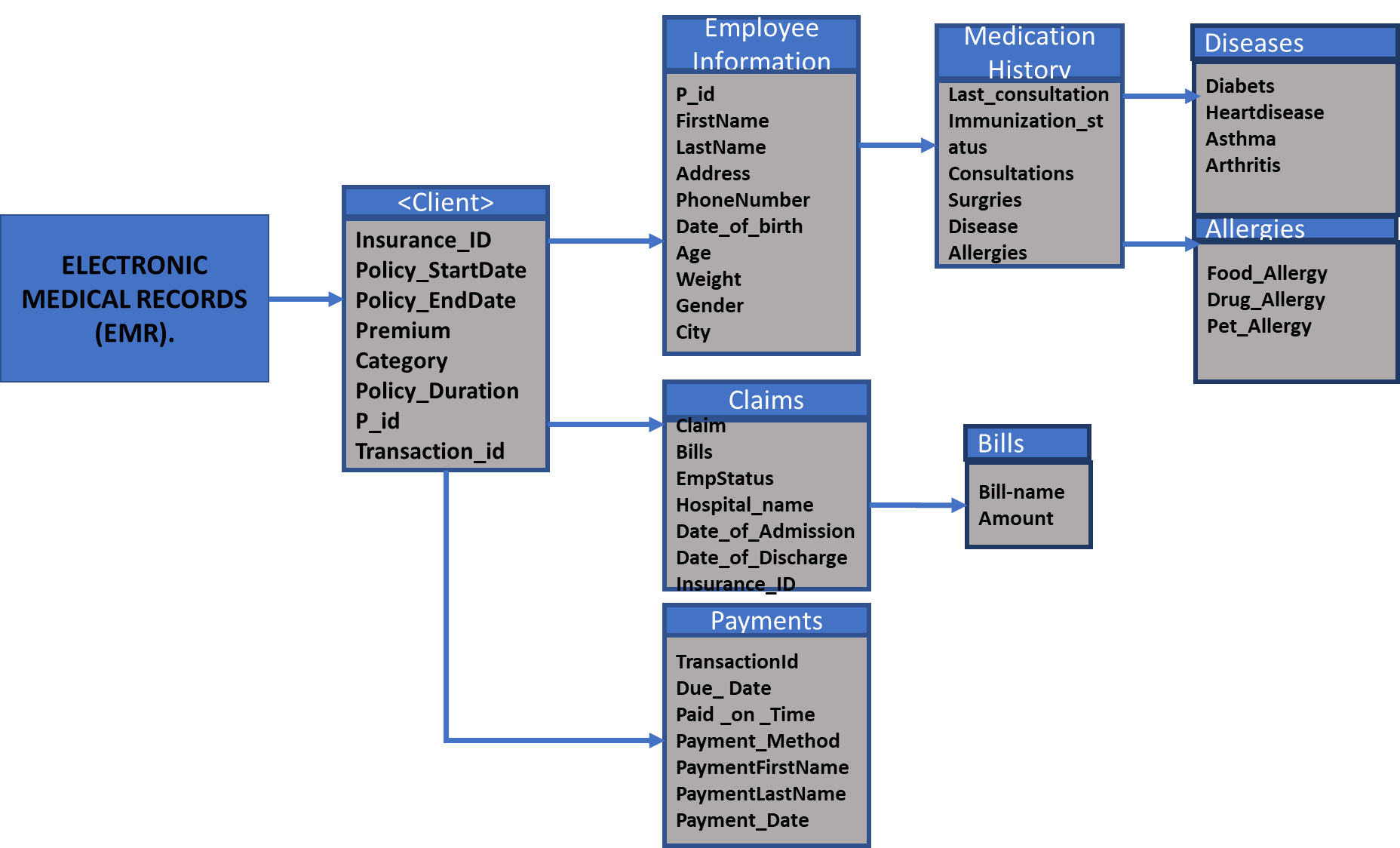


Figure 1.0: Illustration of the EMR model showing a single insurance client/employee.

Under claims, we have details about the corresponding employee claim history as child nodes with each claim denoted as Claim. Each claim node contains information about type of claim, the hospital, amount, etc. The payment node contains record about premium payments, date information and an indication whether payments were made on time.

As shown in Figure 1.0, the Client node contains entries detailing insurance information – insurance ID, policy start date, policy end date, premium, etc. The advantages and disadvantages of this modelling approach are highlighted below.

Advantages

* Clarity of model and implementation: Since all information are grouped according to employee/client, it becomes easier to implement the model, retrieve information from the database and/or make modifications to the XML database.
* Simple XSL Query: The modelling structure shown in Figure 1.0 allows retrieval of information from the XML database with simple XSL queries that are easy to understand.
* Ease of maintenance: The tree structure offers flexibility in maintenance of the database as it is easy to traverse the tree from the root to the leaves and vice-versa.

Disadvantages:

* The tree structure may in some cases lead to long chain of instructions to fetch multiple information at different nodes in the tree jointly.

1. USE CASES AND XSLT Transformations

Case 1: XSL to retrieve information about clients/employee with a particular insurance category. An XML query is written to extract a list of clients/employee having a particular insurance category. Screen shots of the XML, XSLT and HTML output for a query to fetch insurance category – GroupA.

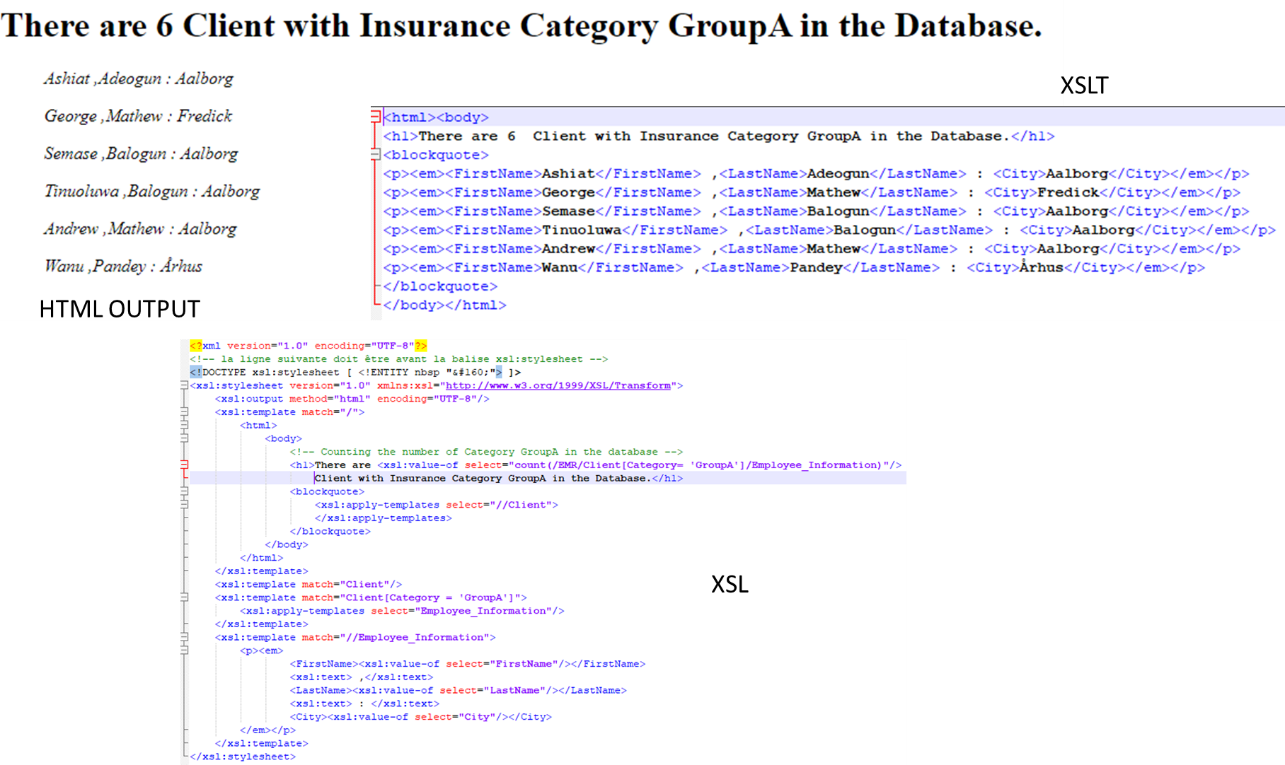


Figure C1: An example query to fetch details of all clients with GroupA insurance category.

Case 2: Medical history of clients with a particular disease type – e.g., diabetes as shown in the screenshots below.



Figure C2: Screenshot of XSLT, XSL and HTML for Case 2 - Medical history of clients with diabetics

Case 3: Retrieve information about clients with expired insurance policy. The query fetches names of clients as well as information about the expired policy option.



Figure C3: Screen shots of XML, XSLT and HTML for information about clients with expired insurance policy

Case 4: Retrieve all clients and calculate the total claim using their claims history. The allows the company to view clients with potentially excessive claims, etc. A screenshot example is shown in Figure 4.



**Figure C4**: Screenshot of XML, XSLT and HTML output for use case 4.

1. JSON – XML MODELS COMPARISON

From the attempts made, using JSON for modelling the database appears relatively easy when compared to the XML model created.

1. TOOLS, ONLINE SERVICES USED AND FILES INCLUDED

The following are the tools and online resources utilized for solving the problem.

1. Notepad ++: for writing XML, XSL and JSON scripts
2. www. freeformater.com
3. jsonschemavalidator.net

Details of the files included are as follows:

1. Source XML file and Schema**: Health.xml and Health.xsd**
2. XSL files: **Question1.xsl, Question2.xsl, Question3.xsl and Question4.xsl**
3. XSLT files: **Transfo1.xslt, Transfo2.xslt, Transfo3.xslt, and Transfo4.xslt**
4. Output html files: **Question1.html, Question2.html, Question3.html and Question4.html**
5. JSON and JSON-Schema files: **Health.json and Healthschema.json**
6. CONCLUSION

A conscise summary of the modelling and implementation of an XML database of a health insurance company is presented in this report. The process utilizes various software and online tools including notepad++ for creating XML, XSL and JSON (and JSON-schema) scripts and performing XSL to XSLT transformation, [www.freeformater](http://www.freeformater) for generating XML- schema and [www.jsonschemavalidator.net](http://www.jsonschemavalidator.net) for validating JSON-schema.