Relational Databases Ingestion into a NoSQL Data Warehouse

Authors: Rym JEMMALI
Fatma ABDELHEDI
Gilles ZURFLUH

Presented By: Rym JEMMALI
Toulouse Institute of Computer Science Research (IRIT), CBI²- Trimane, Paris, France
Email: rym.jemmali@trimane.fr
Rym JEMMALI

Engineer diploma in computer science (2019)

2nd year PhD student : PhD in computer science, Big Data and Business Intelligence

CBI²- Trimane, Paris, France &
Toulouse Institute of Computer Science Research (IRIT)
Summary

1. Context & issue
2. Case study
3. Related work
4. Contribution & implementation
5. Conclusion & perspectives
Context & issue
Case study
Related work
Contribution
Conclusion & perspectives

Context

Data Lake
- Relational DB
- NoSQL DB
- Files

Decisional processing on a Data Lake

Decision makers
• The diversity of data types and formats
• The volumes stored which can reach several terabytes
• The raw nature of the data in a Data Lake

Decisional processing on a Data Lake
Case study

Medical application

Hospitalizations
Analysis

IRM
Radiology
Interpretation

ENS

Health insurance companies

Diagnosis
Prescriptions

Conclusion & perspectives
Case study

Medical application

Create a Data Warehouse from a Data Lake

Private health insurance companies

ENS

Relational DBs

DW
## Related work

<table>
<thead>
<tr>
<th></th>
<th>Data ingestion from a Data Lake</th>
<th>No data ingestion (without restructuring data)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Data management</td>
<td>Links processing</td>
</tr>
<tr>
<td></td>
<td>Proposing metamodels</td>
<td>Data transferring</td>
</tr>
<tr>
<td>Diamantini et al. (2018)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Candel et al. (2021)</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Hanine et al. (2015)</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Stanescu et al. (2016)</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Chikerur et al. (2015)</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Liyanaarachchi et al. (2016)</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Duggan et al. (2015)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alotaibi et al. (2020)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Data ingestion process

Architecture of the data ingestion process from a Data Lake: RDBToNoSDW
MDA (Model Driven Architecture)

OMG (Object Management Group)

Objectives: Functional specifications VS technical specifications

Principle
- ✔ Use of models
- ✔ Automation of transformations between models

Concepts
- ✔ Model
- ✔ Meta-model
Implementation tools

**EMF** (Eclipse Modeling Framework)
- ✓ Definition of a metamodel
- ✓ Creation of models
- ✓ Automation of transformations

**Technical tools**
- ✓ Ecore
- ✓ XMI
- ✓ QVT
Implementation tools

Eclipse IDE (Java Integrated Development Environment)

✓ Java coding
✓ Algorithmic
CreateDW Module

Source metamodel:
CreateDW Module

Target metamodel:
ConvertLinks Module

Experimentation : Medical application

The « Service Provision » database

The « Analysis » database

Extracts from the relational schemas of the two Data Lake databases
CreateDW Module

Experimentation: Medical application

Context & issue
Case study
Related work
Contribution
Conclusion & perspectives

CreateDW Module

Experimentation: Medical application

Context & issue

Case study

Related work

Contribution

Conclusion & perspectives

Experimentation: Medical application

Context & issue

Case study

Related work

Contribution

Conclusion & perspectives

ICSEA 2021
CreateDW Module

**Experimenteration**: Medical application

<table>
<thead>
<tr>
<th>Name</th>
<th>Color</th>
<th>SuperClasses</th>
<th>Alias</th>
<th>Abstract</th>
<th>Clusters</th>
<th>Default Cluster</th>
<th>Cluster Selection</th>
<th>Records</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysis_Insured</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[34, 35, 36, 37]</td>
<td>34</td>
<td>round-robin</td>
<td>5</td>
<td>RENAME, QUERY ALL, NEW RECORD, DROP</td>
</tr>
<tr>
<td>Analysis_Philosopher</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[42, 43, 44, 45]</td>
<td>42</td>
<td>round-robin</td>
<td>9</td>
<td>RENAME, QUERY ALL, NEW RECORD, DROP</td>
</tr>
<tr>
<td>Doctor_DW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[66, 67, 68, 69]</td>
<td>66</td>
<td>round-robin</td>
<td>78</td>
<td>RENAME, QUERY ALL, NEW RECORD, DROP</td>
</tr>
<tr>
<td>ServiceProvision_Doctor</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[22, 23, 24, 25]</td>
<td>22</td>
<td>round-robin</td>
<td>9</td>
<td>RENAME, QUERY ALL, NEW RECORD, DROP</td>
</tr>
<tr>
<td>Analysis_Insured</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[26, 27, 28, 29]</td>
<td>26</td>
<td>round-robin</td>
<td>3</td>
<td>RENAME, QUERY ALL, NEW RECORD, DROP</td>
</tr>
</tbody>
</table>

Extract from the list of the Data Warehouse classes stored in OrientDB
**Experimentation**: Medical application

Record from the «ServiceProvision_Patients» class

```json
{
  "@type": "d",
  "@id": "#26:0",
  "@version": 1,
  "@class": "ServiceProvision_Patients",
  "Email": "ramon.sadi@gmail.com",
  "FNameDoc": "Ramon",
  "LNameDoc": "Sadi",
  "NoPat": "45657709",
  "Doctor": "5685983"
}
```

Record from the «ServiceProvision_Doctor» class

```json
{
  "@type": "d",
  "@id": "#42:0",
  "@version": 1,
  "@class": "ServiceProvision_Doctor",
  "idDoc": "5685983",
  "FNameDoc": "Olivier",
  "LNameDoc": "Durand",
  "TelNDoc": "06577899"
}
```

The «Service Provision» database

Record from the «ServiceProvision_Patients» class after converting links

Record from the «ServiceProvision_Doctor» class after converting links
MergeClasses Module

Experimentation: Medical application

Record from the « ServiceProvision_Insured » class

```json
{
  "@type": "d",
  "@id": "#34:0",
  "@version": 1,
  "@class": "ServiceProvision_Insured",
  "Gender": "M",
  "FNameIns": "Ramon",
  "INameIns": "Saadi",
  "NoInsured": "45657703",
  "Spouse": "#36:0"
}
```

Record from the « Analysis_Patients » class

```json
{
  "@type": "d",
  "@id": "#26:1",
  "@version": 1,
  "@class": "Analysis_Patients",
  "Email": "ramon.saadi@gmail.com",
  "FNamePat": "Ramon",
  "INamePat": "Saadi",
  "NoPat": "45657703",
  "Doctor": "#22:0"
}
```

Record from the new created « Insured_DW » class

```json
{
  "@type": "d",
  "@id": "#26:0",
  "@version": 1,
  "@class": "Insured_DW",
  "Email": "ramon.saadi@gmail.com",
  "FNameIns": "Ramon",
  "INameIns": "Saadi",
  "NoInsured": "45657703",
  "Spouse": "#36:0"
}
```
MergeClasses Module

Experimentation: Medical application

Extract from the new created "Insured_DW" class
Conclusion

- Creation of a NoSQL Data Warehouse from a Data Lake
- Source: Relational databases
- Target: NoSQL Data Warehouse

Perspectives

- Extending the Data Lake to other types of sources
- Data processing and ingestion from these sources
Thank you for your attention