Reverse Engineering of Relational Database Schema based on Universal Metadata Queries

Stefan Reljic, Drazen Brdjanin, Goran Banjac
M-lab Research Group @ Faculty of Electrical Engineering
University of Banja Luka, Bosnia & Herzegovina
Presentation Outline

• Research context and motivation
• Metamodeling of data dictionaries
• Universal queries for RDBS extraction
• Implementation within the REDBUL tool
• An illustrative example of reverse engineering of database schema
• Conclusion and future work
Research Context & Motivation

Model-driven Software Engineering Laboratory

Faculty of Electrical Engineering • University of Banja Luka

http://m-lab.etf.unibl.org

M-lab long-term research project:
UML-based Reverse Database Engineering

Main project achievement:

http://m-lab.etf.unibl.org:8080/redbul

REDBUL
the first online web-based system for reverse database engineering which represents the extracted DB schema by standard UML class diagram
Research Context & Motivation

REDBUL (Reverse Engineering of Database to UML)

DMBS:
MSSQL Server
Server:

Authentication
Windows Authentication
User name
Password

Database
Output file name

Check connection

http://m-lab.etf.unibl.org:8080/redbul
Research Context & Motivation

Pre-existing RDBS extractor (limitations)

– Only two DBMSs supported: MySQL & MS SQL
– Different set of SQL queries for RDBS extraction for each DBMS
– Non-standardized and non-unified data dictionary

Research objectives

– Define an approach and implement a more flexible RDBS extractor ...
Data dictionary organization

Different Data dictionary organization in different DBMSs

MySQL

Oracle

S. Reljic, D. Brdjanin, G. Banjac: Reverse Engineering of Relational Database Schema based on Universal Metadata Queries
Data dictionary organization

Different Data dictionaries $\Rightarrow$ Different SQL queries

MySQL

```
SELECT TABLE_NAME
FROM information_schema.tables
WHERE TABLE_SCHEMA = <userSchema>;
```

Oracle

```
SELECT TABLE_NAME
FROM all_tables
WHERE OWNER = <userSchema>;
```
Proposed approach

Different Data Dictionary organization in different DBMSs

Universal set of SQL query templates

MySQL

Oracle

MSSQL

Postgre SQL

MariaDB

IBM DB2

Universal Data Dictionary Metamodel

MySQL

Oracle

JSON-based metadata representation

Concrete SQL queries for each DBMS

S. Reljic, D. Brdjanin, G. Banjac: Reverse Engineering of Relational Database Schema based on Universal Metadata Queries
Data dictionary metamodel / Top level view

JSON-based metadata representation

```
"databaseMetadata": {
  "dbName": "name of source DBMS",
  "connectionString": "connection string for source DBMS",
  "schemaMetadata": {...},
  "tableMetadata": {...},
  "columnMetadata": {...},
  "primaryKeyMetadata": {...},
  "foreignKeyMetadata": {...},
  "indexMetadata": {...},
  "viewMetadata": {...},
  "dataTypeMetadata": {...}
}
```

```
"mySQLMetadata": {
  "dbName": "MySQL",
  "connectionString": "jdbc:mysql://{0}:{1}/{2}",
  ...
}
```

```
"oracleMetadata": {
  "dbName": "Oracle",
  "connectionString": "jdbc:oracle:thin:@{0}:{1}/{2}",
  ...
}
```

S. Reljic, D. Brdjanin, G. Banjac: Reverse Engineering of Relational Database Schema based on Universal Metadata Queries
### System tables containing data about user tables

<table>
<thead>
<tr>
<th>DBMS</th>
<th>System tables</th>
<th>Columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySQL</td>
<td>INFORMATION_SCHEMA.TABLES</td>
<td>TABLE_SCHEMA,TABLE_NAME</td>
</tr>
<tr>
<td>Oracle</td>
<td>ALL_TABLES</td>
<td>OWNER, TABLE_NAME</td>
</tr>
<tr>
<td>MS SQL</td>
<td>INFORMATION_SCHEMA.TABLES</td>
<td>TABLE_SCHEMA,TABLE_NAME</td>
</tr>
<tr>
<td>MariaDB</td>
<td>INFORMATION_SCHEMA.TABLES</td>
<td>TABLE_SCHEMA,TABLE_NAME</td>
</tr>
<tr>
<td>PostgreSQL</td>
<td>INFORMATION_SCHEMA.TABLES</td>
<td>TABLE_SCHEMA,TABLE_NAME</td>
</tr>
<tr>
<td>IBM DB2</td>
<td>SYSCAT.TABLES</td>
<td>TABSCHEMA, TABNAME</td>
</tr>
</tbody>
</table>

### JSON-based metadata representation

MySQL metadata:

```
"tableMetadata": {
  "tablesTable": {
    "name": "information_schema.tables",
    "columns": {
      "schemaName": "TABLE_SCHEMA",
      "tableName": "TABLE_NAME"
    }
  }
}
```

Oracle metadata:

```
"tableMetadata": {
  "tablesTable": {
    "name": "all_tables",
    "columns": {
      "schemaName": "OWNER ",
      "tableName": "TABLE_NAME"
    }
  }
}
```

### Metamodel excerpt

```
"tableMetadata": {
  "tablesTable": {
    "name": "information_schema.tables", 
    "columns": { 
      "schemaName": "TABLE_SCHEMA", 
      "tableName": "TABLE_NAME"
    }
  }
}
```
Universal queries / Tables excerpt

Universal SQL query template

```
SELECT <tablesTable.column.tableName>
FROM <tablesTable.name>
WHERE <tablesTable.column.schemaName> = <userSchema>;
```

Query template for extraction of table names

MySQL metadata
```
"tableMetadata":{
  "tablesTable":{
    "name": "information_schema.tables",
    "columns":{
      "schemaName": "TABLE_SCHEMA",
      "tableName": "TABLE_NAME"
    }
  }
}
```

Oracle metadata
```
"tableMetadata":{
  "tablesTable":{
    "name": "all_tables",
    "columns":{
      "schemaName": "OWNER",
      "tableName": "TABLE_NAME"
    }
  }
}
```

SQL Query for MySQL
```
SELECT TABLE_NAME
FROM information_schema.tables
WHERE TABLE_SCHEMA = <userSchema>;
```

SQL Query for Oracle
```
SELECT TABLE_NAME
FROM all_tables
WHERE OWNER = <userSchema>;
```
Implementation

Improved RDBS Extractor service in REDBUL
Implementation

REGBUL (Reverse Engineering of Databases to UML)

DMBS: Oracle
Server:

Schema name:
Port:

Username:
Password:

Service name:
Output file name:

Extract schema

http://m-lab.etf.unibl.org:8081/
Illustrative example of reverse RDBS engineering

Sample source RDBS (MySQL, http://db4free.net)

Extracted RDBS (REDBUL, http://m-lab.etf.unibl.org:8081/)
Conclussion and future work

• In this paper we presented an approach of reverse RDBS engineering based on data dictionary metadata and universal meta-queries
• In comparison with existing REDBUL implementation, presented approach has three main advantages:
  – Adding support for new DBMSs by extending metadata bundle;
  – Only one codebase has to be maintained;
  – There is possibility to implement new way of extraction for any schema element, and to use it as hybrid solution for specified DBMS.
  – This solution currently enables extraction of six different DBMSs

• In the future we plan to:
  – Further improve the entire approach;
  – Further improve the implemented tool.
Thank you!

**Stefan Reljic**, Drazen Brdjanin, Goran Banjac
M-lab Research Group @ Faculty of Electrical Engineering
University of Banja Luka, Bosnia & Herzegovina