
# 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 (**R**everse **E**ngineering of **D**ata**B**ases to **U**m**L**)

**DMBS:**

MSSQL Server

**Server:**

Authentication

Windows Authentication

User name

Password

Check connection

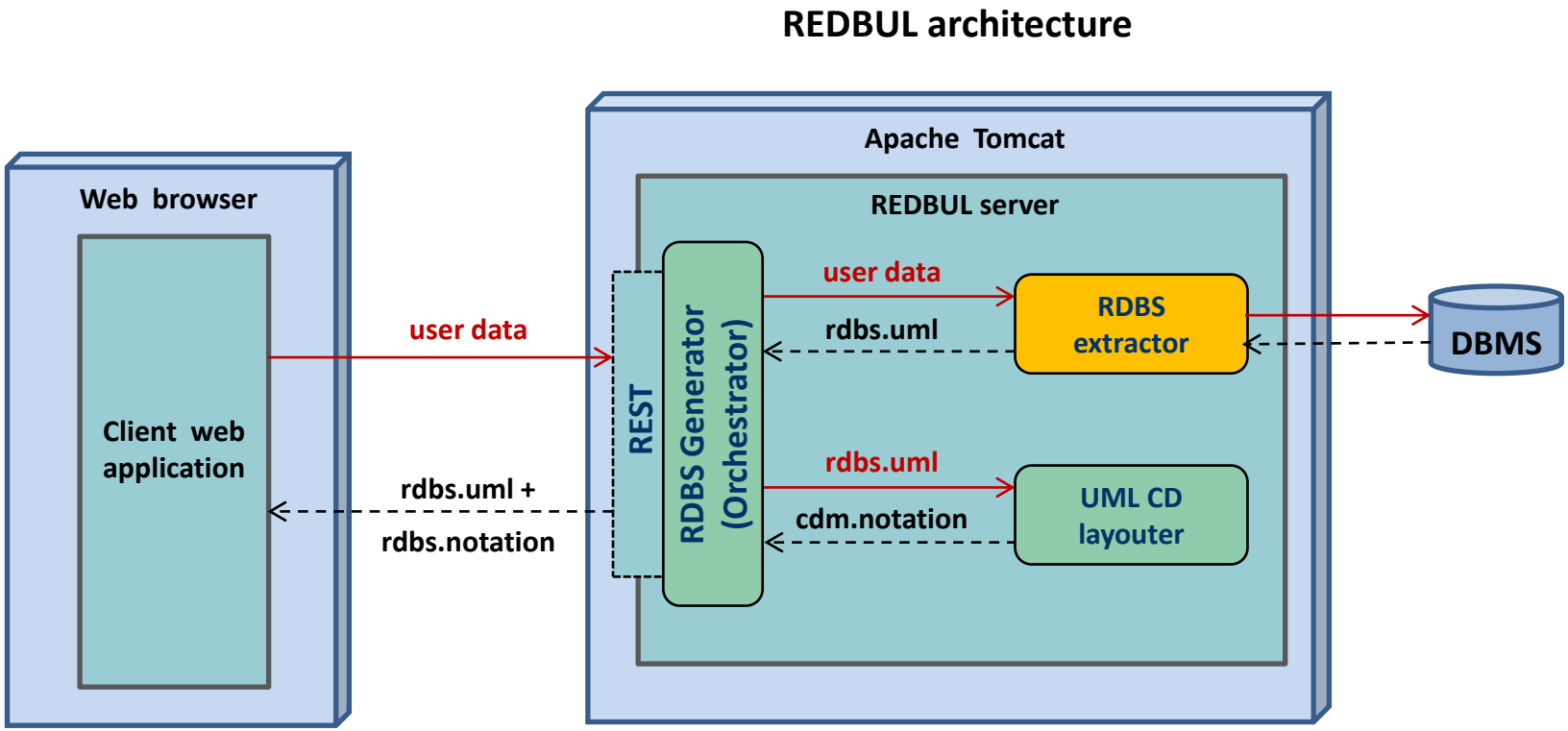
Database

Output file name

Extract Schema

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

# Research Context & Motivation



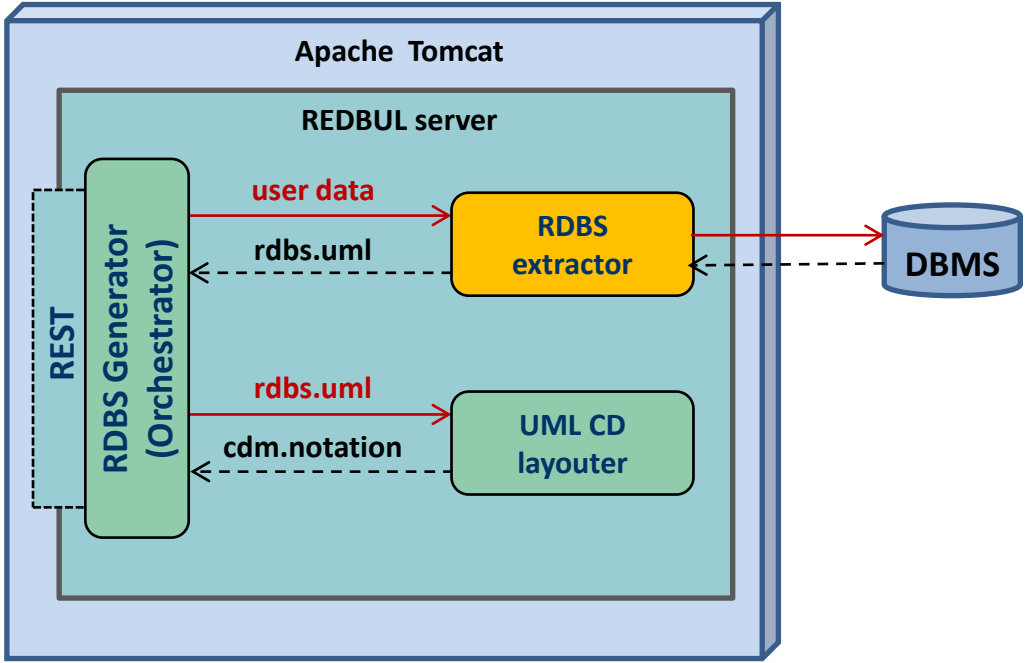
# Research Context & Motivation

## 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

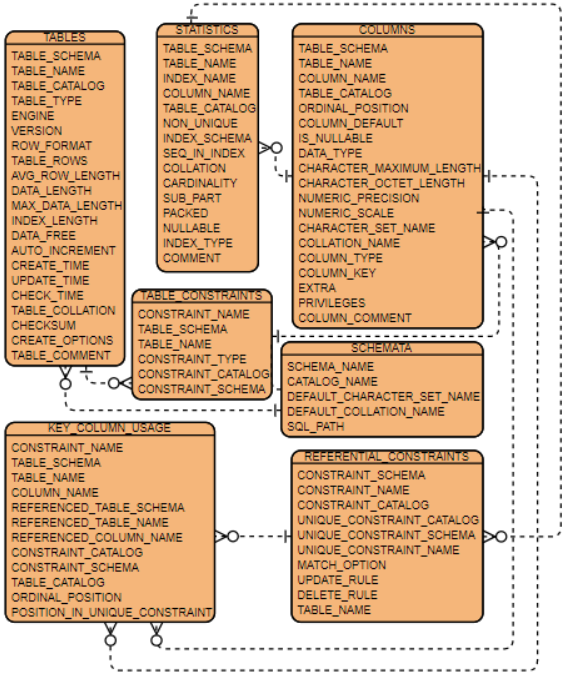
- Define an approach and **implement a more flexible RDBS extractor** …



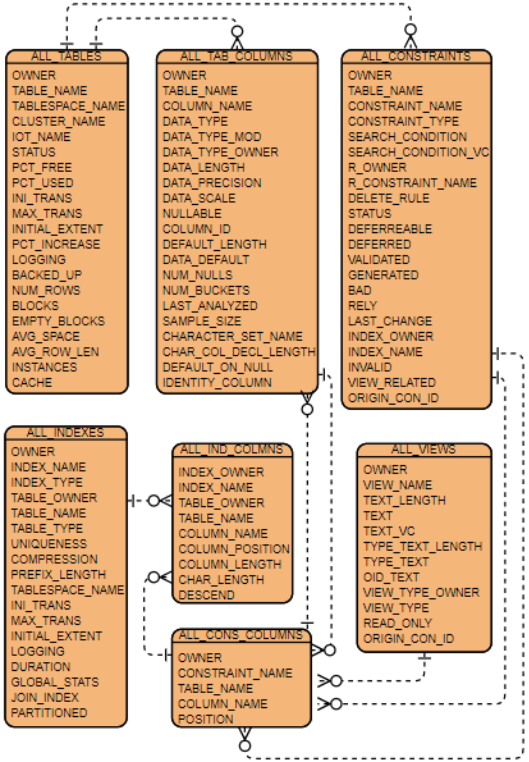
# Data dictionary organization

## Different Data dictionary organization in different DBMSs

**MySQL**



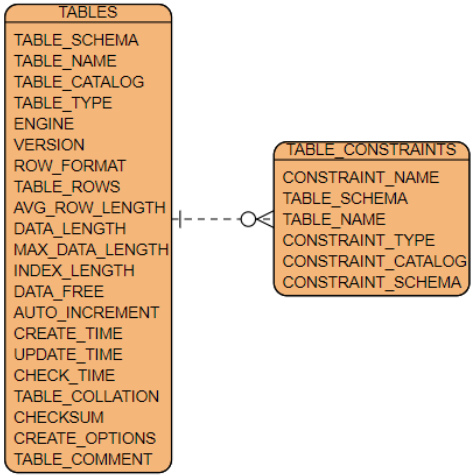

**Oracle**



# Data dictionary organization

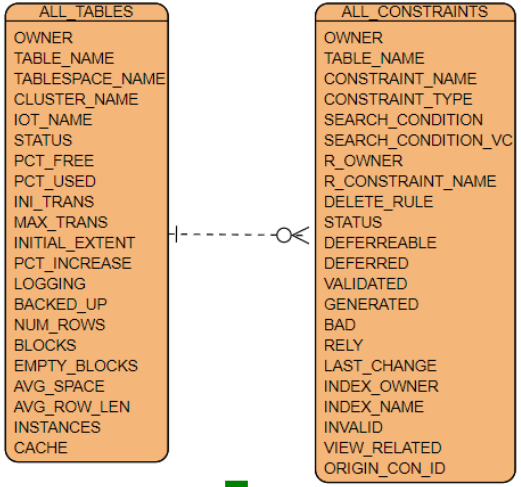
## Different Data dictionaries ⇒ Different SQL queries

**MySQL**




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

**Oracle**

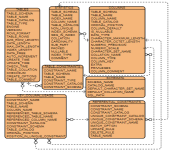



```
SELECT TABLE_NAME
FROM   all_tables
WHERE  OWNER = <userSchema>;
```

# Proposed approach

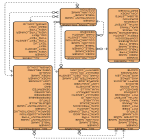
## Different Data Dictionary organization in different DBMSs

**MySQL**

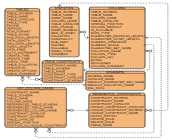


**Oracle**

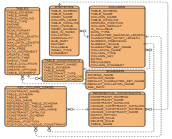
**MSSQL**



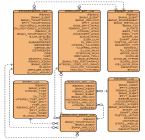
**Postgre SQL**



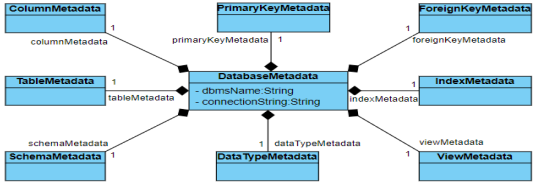
**MariaDB**



**IBM DB2**



## Universal Data Dictionary Metamodel




**MySQL**

```
"tableMetadata": {
  "tablesTable"
    "name": "in
    "columns":
      "schemal
      "tableNar
    }
  }
}
```

**Oracle**

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

• • •

## JSON-based metadata representation

## Universal set of SQL query templates

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

**MySQL**

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

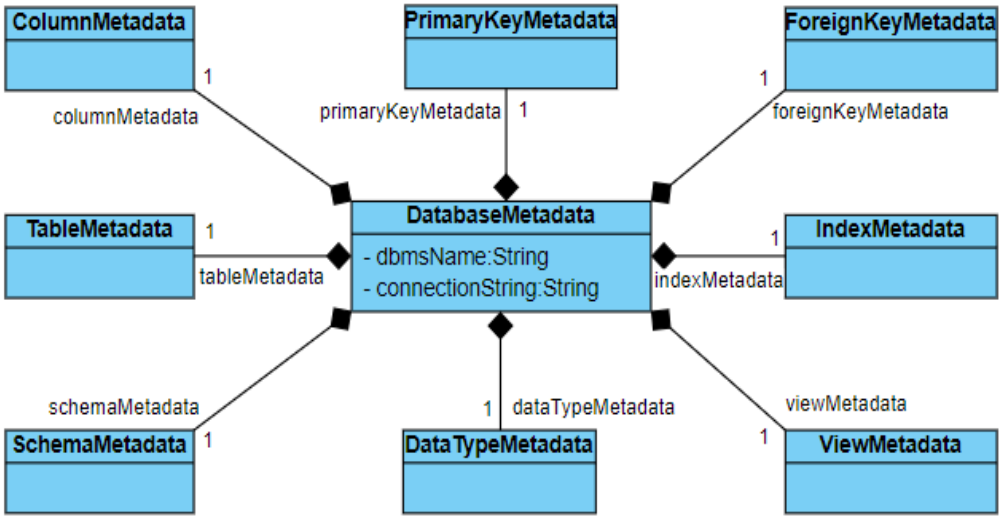
```
SELECT TABLE_NAME
FROM   all_tables
WHERE  OWNER = <userSchema>;
```

• • •

**Oracle**

## Concrete SQL queries for each DBMS

# Data dictionary metamodel / Top level view




## JSON-based metadata representation

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

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

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

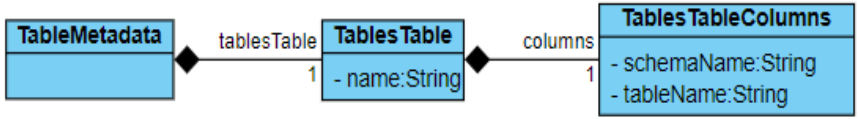
• • •

# Data dictionary metamodel / Tables excerpt

## System tables containing data about user tables

| DBMS | System tables | Columns |
|---|---|---|
| MySql | INFORMATION_SCHEMA.TABLES | TABLE_SCHEMA,TABLE_NAME |
| Oracle | ALL_TABLES | OWNER, TABLE_NAME |
| MS SQL | INFORMATION_SCHEMA.TABLES | TABLE_SCHEMA,TABLE_NAME |
| MariaDB | INFORMATION_SCHEMA.TABLES | TABLE_SCHEMA,TABLE_NAME |
| Postgre SQL | INFORMATION_SCHEMA.TABLES | TABLE_SCHEMA,TABLE_NAME |
| IBM DB2 | SYSCAT.TABLES | TABSCHEMA, TABNAME |

## Metamodel excerpt




## 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"
    }
  }
}
```

## JSON-based metadata representation

```
"tableMetadata": {
  "tablesTable": {
    "name": "name of table with table metadata",
    "columns": {
      "schemaName": "name of column with schema name",
      "tableName": "name of column with 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"
    }
  }
}
```

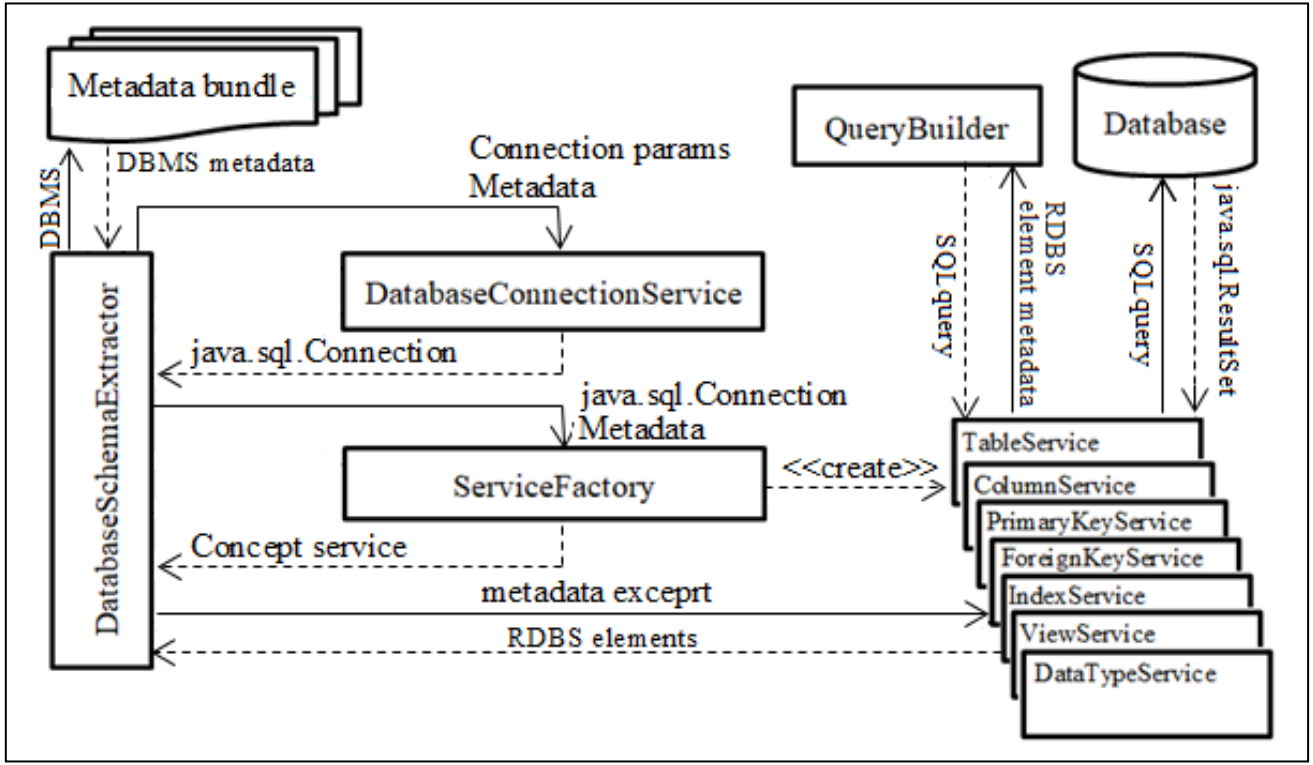
*SQL Query for MySQL*

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

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

*SQL Query for Oracle*

# Implementation




**Improved RDBS Extractor service in REDBUL**

# Implementation

REDBUL (**R**everse **E**ngineering of **D**ata**B**ases to **U**m**L**)

**DMBS:** Oracle

**Schema name:**

**Username:**

**Service name:**

Extract schema

**Server:**

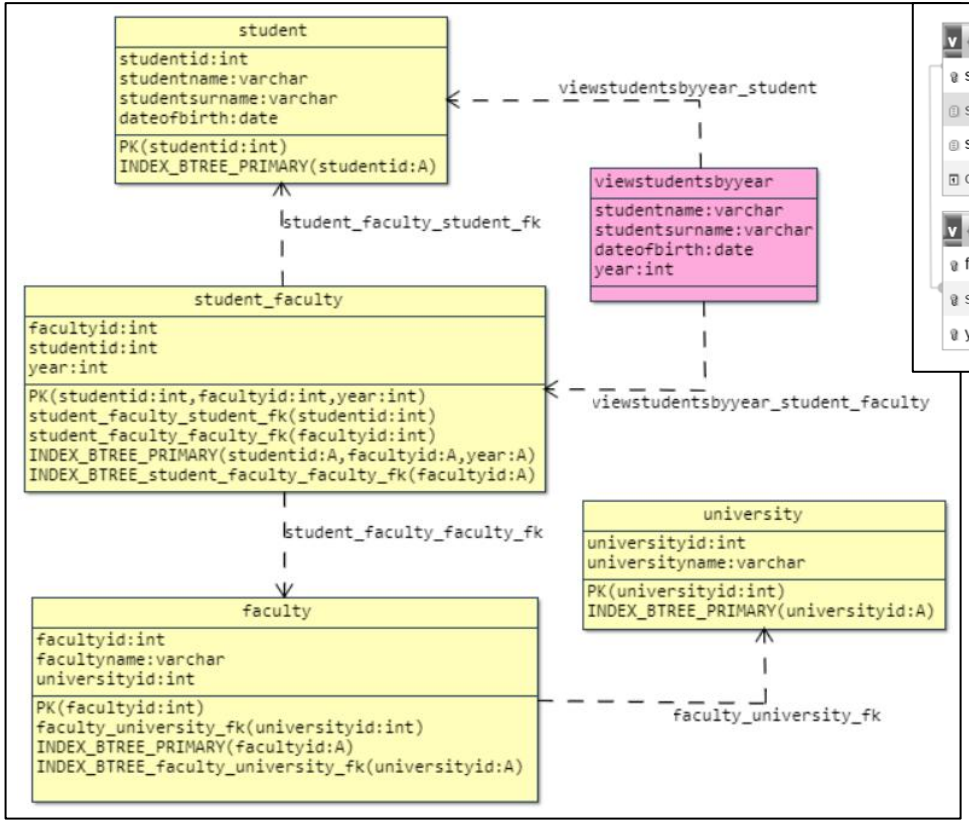
**Port:**

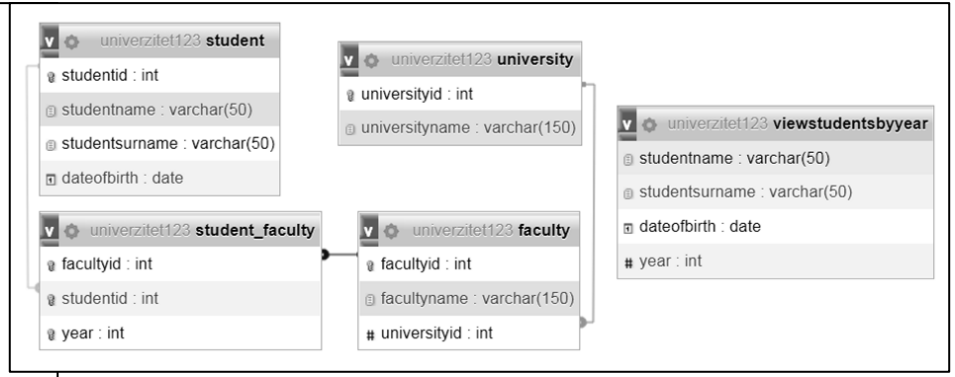
**Password:**

**Output file name:**

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/)**

# Conslusion 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