Efficient Database Model to Represent Numerical Research Data of Material Flow Analysis.

Master Thesis - Computer Science

UNI FREIBURG

Albert-Ludwigs-Universität Freiburg

06 June 2018

Mohammad Mahadi Hasan

Outline

UNI FREIBURG

- **Introduction**
  - Problem Statement
  - Approach
- **Background**
  - Relational Database
  - The Semantic Web
  - Triple stores
- **Construction of Relational Database**
  - MySQL architecture and design process
  - Implementation of web interface
- **Construction of RDF Triple Store**
  - Mapping relational database to RDF triples
  - Apache Jena TDB architecture
- **Database Performance Analysis**
- **Discussion**

Introduction

Problem Statement

Usually, the research data of Industrial Ecology (IE) is stored in .mat or .csv files on a local machine which imposes challenges like:

- Lack of an efficient data storing mechanism.
- Lack of an online platform to query and retrieve data publicly.
- Data sharing and re-using terms has not been discussed for such data.

Approach

- Study on Relational Database (RDB)
  - MySQL database.
  - Implementation of web interface.
- Study on Semantic Web and its components
  - RDF - Data Model.
  - Apache Jena TDB triple store.

Background

UNI FREIBURG

- **Relational Database (RDB)**

  - A collection of data sets organized and stored in relational tables.
  - Each of these tables has one primary key column and shares at least one column (referred as foreign key) with another table to establish relationship.
  - Data can be extracted by querying with query language like SQL.
  - RDBMS is a database management system that is used to create and maintain relational databases.
  - Popular RDBMS examples include Microsoft Access, SQL Server, Oracle Database, MySQL, PostgreSQL etc.

MySQL Database Architecture

- **The application layer:** client-side layer which handles the connection string, authentication and most importantly security.
- **The server layer:** the brain of the overall architecture. Any kind of query statement is executed in this layer.
- **The storage engine layer:** offers different storage engines.

- **Indexes:**

  - Indexes are used to find a data entry quickly.
  - Let us consider following example:

*SELECT * FROM table WHERE id=1;*

- Without the indexes, the query will go through every row and column.
- The optimization here is to add an index, for example, primary key. It runs the query only against the table indexes rather than all the column data.

The Semantic Web

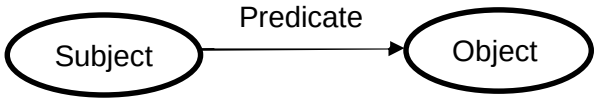
- "The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries“.
- Data is structured and published following semantic web standards: RDF.

RDF- The Data Model

Resource Description Framework is a "framework for representing information in the Web“.

- The basic elements of RDF are triples.
- A triple is a set of three entities that form a statement in the form of *subject-predicate-object* expressions.

Harry Potter → has author → J. K. Rowling



URIs, Vocabularies & Ontologies

- Uniform Resource Identifiers (URIs) are short strings that identify resources in the web.

```
<http://dbpedia.org/resource/Berlin>                    ..(1)
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>       ..(2)
<http://dbpedia.org/class/yago/CapitalsInEurope>        ..(3)
```

subject → predicate → object

(1) → (2) → (3)

This triple refers that resource (1) has a relationship (2) with the resource (3).

- The “vocabularies” on the Semantic Web is used for data integration between data sets.
- The term “ontology” is for more complex, and more formal collection of terms, where “vocabulary” is used for more basic use.

Background

SPARQL – Query Language

- SPARQL is the standardized query language for RDF, just like SQL is the standardized query language for relational databases.
- Like SQL, SPARQL also follow the same “SELECT...FROM...WHERE...” query structure.

```
:id1 foaf:name "André Schürrle"
:id1 foaf: based_near : Dortmund
:id2 foaf:name "Nils Petersen"
:id2 foaf:based_near : Freiburg
```

```
SELECT ?name
WHERE {
?x foaf:name ?name .
?x foaf:based_near : Freiburg .
}
```

The predicate has a constant value of foaf:based_near and the object has a constant value of : *Freiburg* match to one of RDF triples. The result is *Nils Petersen*.

Triple Stores

- Triple Store (RDF triple store) is a specialized DBMS for RDF triples.
- RDF data can be stored in two ways:
  - in *files,* triples are stored following one of the serialization formats, or
  - in special kind of databases for triples, called *triple stores*.
- *Triple stores* have three possible architectures:
  - In-memory: stores the triples in main memory. It is fast but expensive.
  - Native Store: storage systems with own database. For example- Jena TDB, Sesame Native, Virtuoso, AllegroGraph, Oracle 11g etc.
  - Non-native Store: storage using a third-party RDBMS. For example- Jena SDB backed by MySQL database.

Construction of Relational Database

MySQL Design Process:

- Six different CSV files containing the data of country, unit specification, dataset (list of datasets), process list, stock data and flow data between processes.

- Database Schema, Tables, and Constraints
  - Database Schema => database name.
  - 6 tables.
  - 6 primary keys and 11 foreign keys; are said to be database constraints.
  - 589165 data rows.
  - Roughly 55mb of SQL dump.

Construction of Relational Database

UNI FREIBURG

stocks
seq_id
system_id
stock_dataset_number
process_id
ISO_code
…
…
Unit_id
comment

countries
seq_id
ISO_code
name
…
…
alternative_name7

flows
seq_id
system_id
flow_dataset_number
process_id_source
process_id_target
region_source
region_target
…
…
Unit_id
comment

dataset
system_id
system_definition
dataset_name
…
…
comment

process_list
system_id
process_id
process_name
…
…
YPos

Unit_classification
unit_id
si_unit_id
unit_code
…
…
factor

Green columns = primary keys.

Purple columns = foreign keys.

Construction of Relational Database

Implementation of web interface

- Back-End Development
  - Mechanisms to communicate with server-side database.
- Front-End Development
  - Mechanisms to present data in client-side.

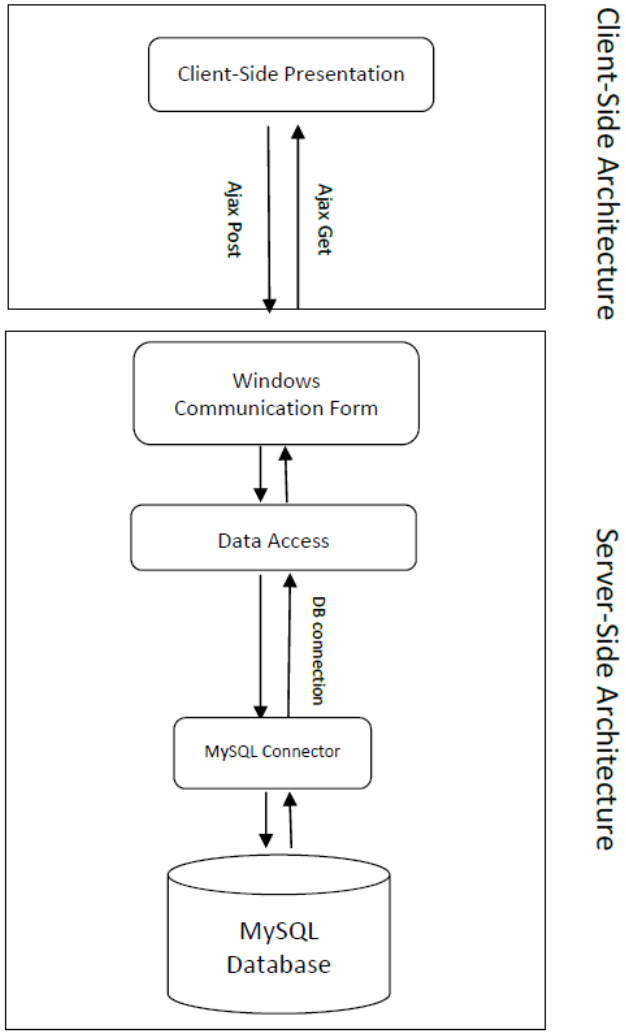


Figure: web interface architecture

Construction of Relational Database

Data Table with the catalogue of datasets:

- ***Query 1:*** *Select all the dataset available in the database from dataset table;*

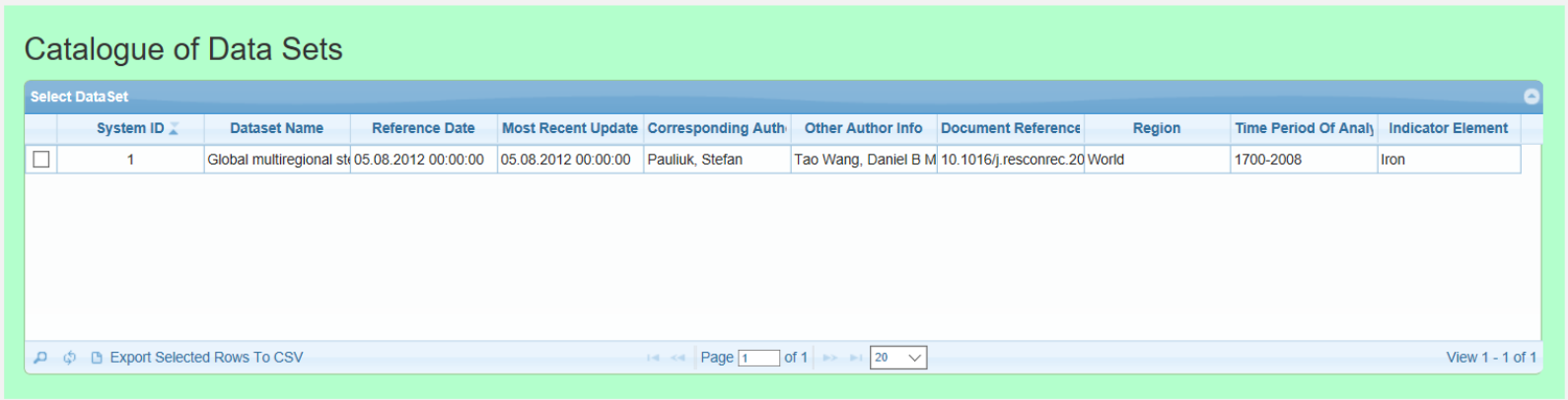


Figure: DataGrid with list of datasets available in the database

Construction of Relational Database

Data Table with material stocks:

- ***Query 2:*** *Select all the stocks within the dataset (identified by the id) and filter the data with year (2007, 2008) and country (Austria, Germany, Switzerland and Italy) from stock table;*

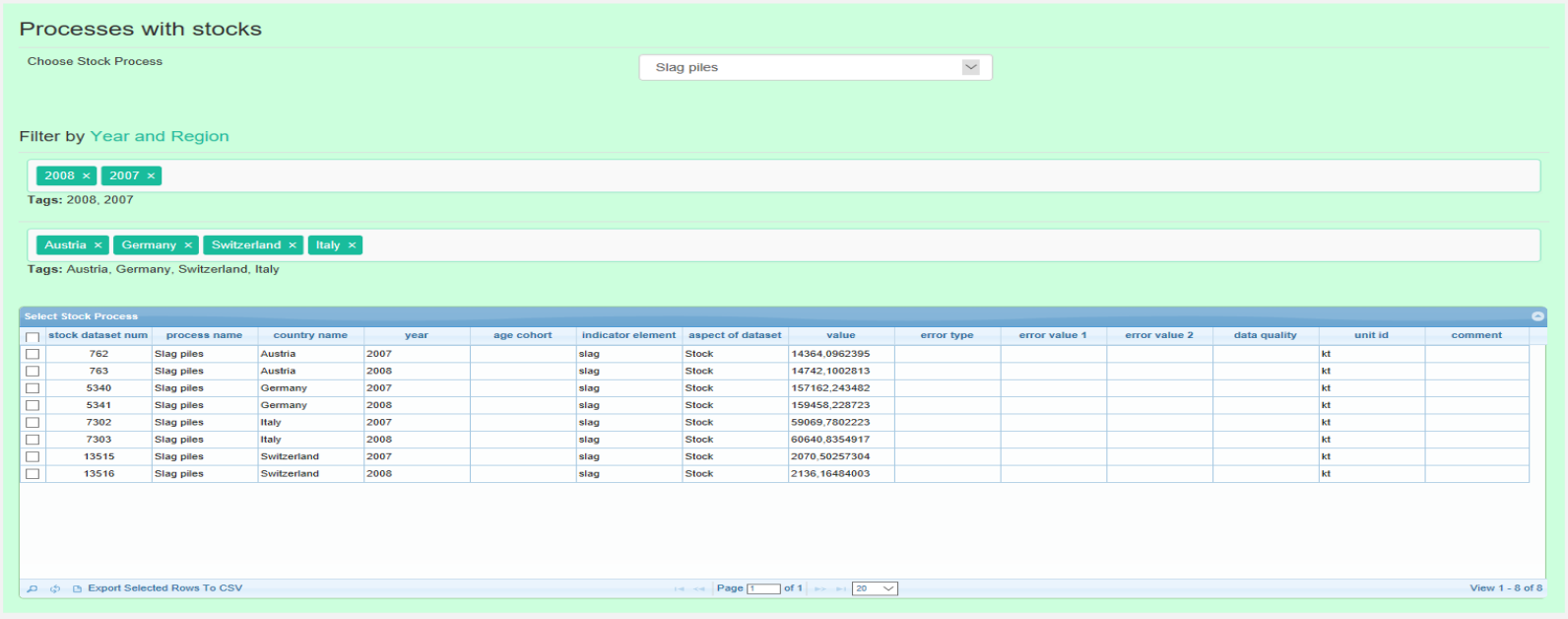
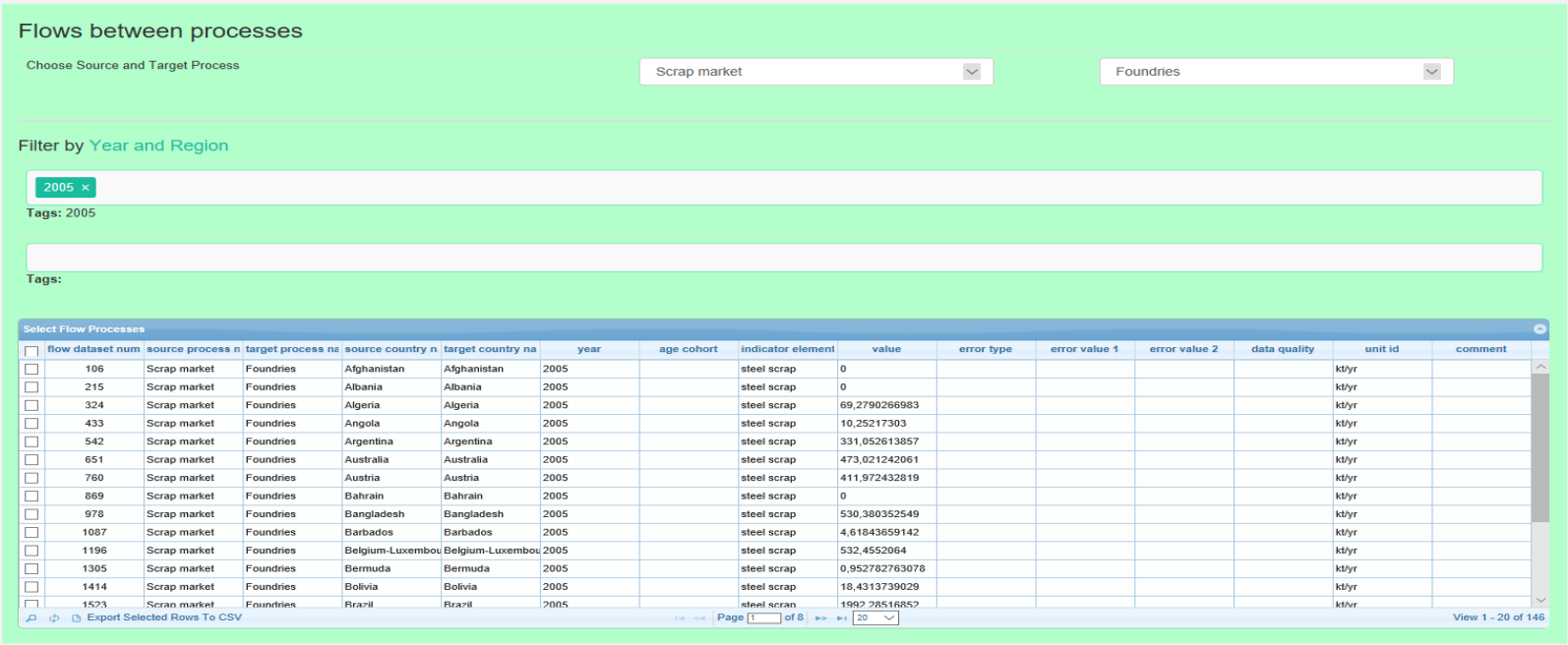


Figure: Data Grid with stock data between processes

Construction of Relational Database

Data Table with material flows:

- ***Query 3:*** *Select all the flows within the dataset (identified by the id) and filter the data with year (2005) and source process 'scrap market' and target process 'foundries' from flow table;*



flow dataset num	source process n	target process na	source country n	target country na	year	age cohort	indicator element	value	error type	error value 1	error value 2	data quality	unit id	comment
106	Scrap market	Foundries	Afghanistan	Afghanistan	2005		steel scrap	0					kt/yr	
215	Scrap market	Foundries	Albania	Albania	2005		steel scrap	0					kt/yr	
324	Scrap market	Foundries	Algeria	Algeria	2005		steel scrap	69,2790266983					kt/yr	
433	Scrap market	Foundries	Angola	Angola	2005		steel scrap	10,25217303					kt/yr	
542	Scrap market	Foundries	Argentina	Argentina	2005		steel scrap	331,052613857					kt/yr	
651	Scrap market	Foundries	Australia	Australia	2005		steel scrap	473,021242061					kt/yr	
760	Scrap market	Foundries	Austria	Austria	2005		steel scrap	411,972432819					kt/yr	
869	Scrap market	Foundries	Bahrain	Bahrain	2005		steel scrap	0					kt/yr	
978	Scrap market	Foundries	Bangladesh	Bangladesh	2005		steel scrap	530,380352549					kt/yr	
1087	Scrap market	Foundries	Barbados	Barbados	2005		steel scrap	4,61843659142					kt/yr	
1196	Scrap market	Foundries	Belgium-Luxembou	Belgium-Luxembou	2005		steel scrap	532,4552064					kt/yr	
1305	Scrap market	Foundries	Bermuda	Bermuda	2005		steel scrap	0,952782763078					kt/yr	
1414	Scrap market	Foundries	Bolivia	Bolivia	2005		steel scrap	18,4313739029					kt/yr	
1523	Scrap market	Foundries	Brazil	Brazil	2005		steel scrap	1992,28516852					kt/yr	

Figure: Data Grid with flow data between processes

Construction of Triple Store

UNI FREIBURG

- The majority of data on the current Web is stored in relational databases.
- Semantic web is useful, especially if data from different sources has to be shared or integrated.
- Therefore, it is important to introduce mapping technologies between relational database and RDF.

- **D2RQ:** A mapping mechanism to use RDB as RDF graphs (in addition, it exposes RDB to RDF triples)

Construction of Triple Store

D2RQ architecture

- A D2RQ Engine accesses a Non-RDF database.
- The mapping language describes the relationship between an ontology (or vocabulary) and a relational data model.
- RDF APIs can be embedded with Java applications via Jena or Sesame APIs.
- The RDF dump file can be stored in any triple store.
- D2RQ server provides an HTML view to explore the mapped database.

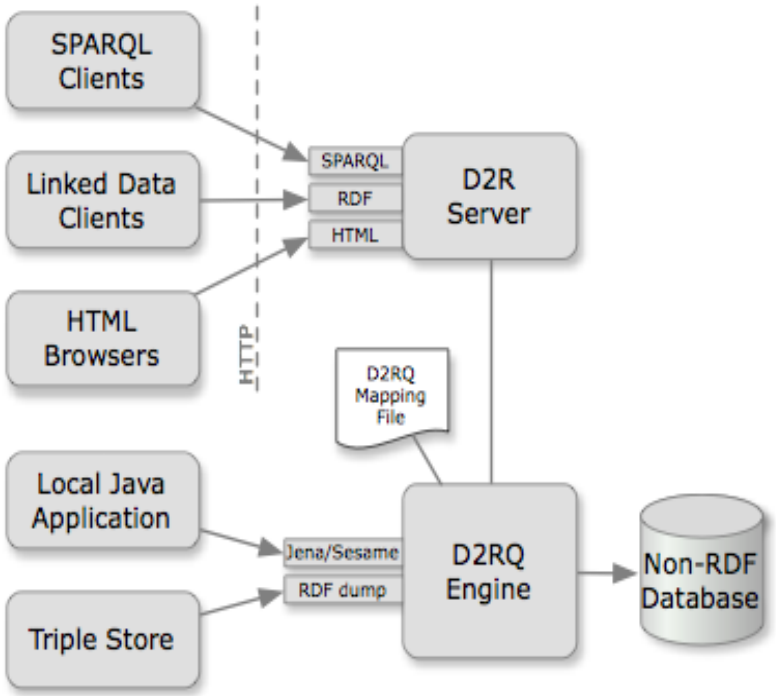


Figure: Architecture of D2RQ

Construction of Triple Store

UNI FREIBURG

D2RQ: Overview and Features

- The D2RQ platform serves different tools to offer RDF-based access to the content of relational databases.

  - The ***generate-mapping*** tool creates a D2RQ mapping file by analysing the relational database schema.

```
>generate-mapping -u username -p password -o mapping.ttl jdbc:mysql:///ief
```

  - The ***dump-rdf*** tool is used to dump the contents of the entire database into a single RDF file.

```
>dump-rdf -f TURTLE -b http://localhost:2020/mapping.ttl > iefdatabase.ttl
```

  - The ***D2R-server*** takes the mapping file as input and provides a web interface where RDF data can be browsed.

```
>d2r-server mapping.ttl
```

Construction of Triple Store

Database table to RDF triples with D2RQ:

<“primary key column value“ “tableName_columnName“ “columnValue“>

id	country	aspect_of_dataset	indicator_element	process	system_id	unit_id	value	year
17011	Germany	Stock	slag	Scrap market	1	kilo ton	444.0907	1906

```
<"stocks17011" "stocks_country" "Germany">
<"stocks17011" "stocks_aspect_of_dataset" "Stock">
<"stocks17011" "stocks_indicator_element" "slag">
<"stocks17011" "stocks_process" "Scrap market">
<"stocks17011" "stocks_system_id" "1">
<"stocks17011" "stocks_unit_id" "kilo ton">
<"stocks17011" "stocks_value" "444.0907">
<"stocks17011" "stocks_year" "1906">
```

Construction of Triple Store

UNI FREIBURG

- **Apache Jena TDB**
  - TDB is a component of Jena that used as native RDF storage.
  - TDB is stored in a single directory in the filing system backed by a dataset.
  - A complete TDB dataset consists of:
    - **The node table:** stores the representation of RDF terms. It consists of two mappings: Node to NodeId and NodeId to Node.
    - **Triple and Quad indexes:** used for the default graph. Triples are held as 3-tuples of NodeIds in triple indexes where quads are held as 4-tuples of NodeIds.
    - **The prefixes table:** stores index for Graph->Prefix->URI mapping. It provides the mechanism for Jena API to serialize the triples in RDF/XML or Turtle.

Database Performance Analysis

- **Experimental Settings:**
  - Two training datasets:
    - **IEF dataset:** consisting the material stock and flow data of Industrial Ecology Freiburg research group.
      - 6 tables, 589165 rows.
      - D2RQ generates roughly 10 million triples.
      - 7596007 final triples (ignoring the empty “ ” column values).
    - **ISWC dataset:** sample dataset with the information about conferences, papers, authors and topics from ISWC 2002 conference[1].
      - 9 tables, 96 rows.
      - 322 triples after mapping.

[1]http://iswc2002.semanticweb.org/

Database Performance Analysis

UNI FREIBURG

Results Evaluation:

SQL Queries on MySQL		**SPARQL Queries on Jena TDB**	
	Query Execution Time (ms) (average of 10 run)		Query Execution Time (ms) (average of 10 run)
Query 1	9	*Query 5*	145
Query 2	127	*Query 6*	1673
Query 3	80	*Query 7*	11448
Query 4	16	*Query 8*	10257

Table 1: Query execution time comparisons for ief dataset

Database Performance Analysis

UNI FREIBURG

Results Evaluation:

SQL Queries on MySQL		**SPARQL Queries on Jena TDB**	
	Query Execution Time (ms) (average of 10 run)		Query Execution Time (ms) (average of 10 run)
Query 9	3	*Query 13*	132
Query 10	3	*Query 14*	7
Query 11	2	*Query 15*	5
Query 12	2	*Query 16*	12

Table 2: Query execution time comparisons for iswc dataset

Database Performance Analysis

UNI FREIBURG

Results Evaluation:

- MySQL works faster than TDB
  - MySQL storage engines are faster architecturally.
  - Use of indexing to identify records.
- TDB works slower because of enumeration
  - TDB query execution method does not finish until results are fully enumerated.
  - Enumeration process takes longer time depending on number of predicates (number of columns in case of MySQL) and triples.

```
ResultSet results = qexec.execSelect();
long numResults = ResultSetFormatter.consume(results);
```

The variation of query execution time in TDB (when triples are available in main memory!).

Discussions

UNI FREIBURG

- Most of the web contents are backed by RDB.
- RDB's are easy to implement; become complex with the number of tables increases.
- RDB works as "closed loop".
- Data can not be integrated from different sources if the database schemas are not in same structure.

- Triple Stores allow data to be extended across different data sources.
- Data can be integrated with multiple data sources by adding properties.
- A small data source can be enlarged into a bigger and richer data source.

Discussions

- **Future Work**
  - Query optimization for SQL joins.
  - Allow filtering data with a range of years (1901-2000 instead of using 1901,1902, …., 2000).
  - Introduce filtering based on a collection of regions (not only country wise, also continents wise like “Europe“).
  - Improvement of default RDF vocabulary terms within D2RQ (use of more meaningful and publicly well-known vocabulary).
  - Build a domain-specific complete ontology from the dataset and make it public.

UNI FREIBURG

Thank You!