MASWS – Relational Databases and RDF

Kate Byrne

29th January 2009
Outline

Information Management and Relational Databases
  some history
  relational database basics

Converting Relational Databases to RDF
  why would you want to?
  how? – the basic conversion procedure

What’s Wrong With the Basic Procedure?
  bnodes
  literals
  redundancy

Useful Websites
Milestones in Information Management

- 1450s: Gutenberg invents movable type printing
- 1950s: First "modern" computers come into use
- 1970s: The Internet emerges; database management systems begin to be widely used
- 1990s: Tim Berners-Lee invents the World Wide Web (1990); first version of RDF becomes W3C Recommendation; Tim Berners-Lee proposes the Semantic Web (1999)
Milestones in Information Management

- **1450s:**
  - Gutenberg invents movable type printing

- **1950s:**
  - First "modern" computers come into use

- **1970s:**
  - The Internet emerges
  - Database management systems begin to be widely used
  - Ted Codd proposes the Relational Model

- **1990s:**
  - Tim Berners-Lee invents the World Wide Web (1990)
  - First version of RDF becomes W3C Recommendation
  - Tim Berners-Lee proposes the Semantic Web (1999)
Milestones in Information Management

- **1450s:**
  - Gutenberg invents movable type printing

- **1950s:**
  - First “modern” computers come into use

- **1970s:**
  - The Internet emerges
  - Database management systems begin to be widely used
  - Ted Codd proposes the Relational Model

- **1990s:**
  - Tim Berners-Lee invents the World Wide Web (1990)
  - First version of RDF becomes W3C Recommendation
  - Tim Berners-Lee proposes the Semantic Web (1999)
Milestones in Information Management

• 1450s:
  • Gutenberg invents movable type printing

• 1950s:
  • first “modern” computers come into use

• 1970s:
  • the Internet emerges
  • database management systems begin to be widely used
  • Ted Codd proposes the Relational Model

• 1990s:
  • Tim Berners-Lee invents the World Wide Web (1990)
  • first version of RDF becomes W3C Recommendation
  • Tim Berners-Lee proposes the Semantic Web (1999)
Milestones in Information Management

• 1450s:
  • Gutenberg invents movable type printing

• 1950s:
  • first “modern” computers come into use

• 1970s:
  • the Internet emerges
  • database management systems begin to be widely used
  • Ted Codd proposes the Relational Model

• 1990s:
  • Tim Berners-Lee invents the World Wide Web (1990)
  • first version of RDF becomes W3C Recommendation
  • Tim Berners-Lee proposes the Semantic Web (1999)
Relational Database Basics
Relations – Tables – Entities – ?Classes

SITE

<table>
<thead>
<tr>
<th>siteNo</th>
<th>name</th>
<th>parish</th>
<th>status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dirleton Castle</td>
<td>Dirleton</td>
<td>scheduled</td>
</tr>
<tr>
<td>2</td>
<td>Dirleton Cottage</td>
<td>Dirleton</td>
<td>listed</td>
</tr>
<tr>
<td>3</td>
<td>Drem Airfield</td>
<td>Dirleton</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Jamie’s Neuk</td>
<td>Dirleton</td>
<td></td>
</tr>
</tbody>
</table>

ARCHIVE

<table>
<thead>
<tr>
<th>archNo</th>
<th>category</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>photo</td>
<td>North face</td>
</tr>
<tr>
<td>102</td>
<td>drawing</td>
<td>Site plan</td>
</tr>
<tr>
<td>103</td>
<td>map</td>
<td>Parish map</td>
</tr>
</tbody>
</table>

- Each row (tuple) describes an instance
- Each column contains values for an attribute
- Null values (generally) permitted
- A relation is set of tuples:
  \text{site}(\text{siteNo}, \text{name}, \text{parish}, \text{status})
Relational Database Basics
Relations – Tables – Entities – Classes

SITE

<table>
<thead>
<tr>
<th>siteNo</th>
<th>name</th>
<th>parish</th>
<th>status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dirleton Castle</td>
<td>Dirleton</td>
<td>scheduled</td>
</tr>
<tr>
<td>2</td>
<td>Dirleton Cottage</td>
<td>Dirleton</td>
<td>listed</td>
</tr>
<tr>
<td>3</td>
<td>Drem Airfield</td>
<td>Dirleton</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Jamie’s Neuk</td>
<td>Dirleton</td>
<td></td>
</tr>
</tbody>
</table>

ARCHIVE

<table>
<thead>
<tr>
<th>archNo</th>
<th>category</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>photo</td>
<td>North face</td>
</tr>
<tr>
<td>102</td>
<td>drawing</td>
<td>Site plan</td>
</tr>
<tr>
<td>103</td>
<td>map</td>
<td>Parish map</td>
</tr>
</tbody>
</table>

- Each row (tuple) describes an **instance**
- Each column contains values for an **attribute**
- Null values (generally) permitted
- A relation is **set** of tuples:
  site\((siteNo, name, parish, status)\)
The photo, drawing and map all pertain to Dirleton Castle.

Dirleton Cottage and Jamie’s Neuk are also on the parish map.

There is no archive associated with Drem Airfield.
The photo, drawing and map all pertain to Dirleton Castle
Dirleton Cottage and Jamie’s Neuk are also on the parish map
There is no archive associated with Drem Airfield
Where is RDF stored anyway?

• If your RDF graph is too big to fit in memory...
  • ...you store it in a relational database
  • Many triple stores available: Jena, Kowari, Sesame,...
  • These all use database back-ends, typically MySQL
    • 3-column table, for Subject, Predicate, Object
    • usually GraphId too, so table becomes 4-column
    • physical implementations differ
  • Relational database query language is SQL
  • ⇒ SPARQL has to be translated (invisibly) into SQL

(Note: not all RDF stores use relational dbs.)
Where is RDF stored anyway?

- If your RDF graph is too big to fit in memory...
- ...you store it in a relational database
  - Many triple stores available: Jena, Kowari, Sesame,...
  - These all use database back-ends, typically MySQL
    - 3-column table, for Subject, Predicate, Object
    - Usually GraphId too, so table becomes 4-column
    - Physical implementations differ
  - Relational database query language is SQL
  - ⇒ SPARQL has to be translated (invisibly) into SQL

(Note: not all RDF stores use relational dbs.)
Where is RDF stored anyway?

- If your RDF graph is too big to fit in memory...
- ...you store it in a relational database
- Many triple stores available: Jena, Kowari, Sesame,...
- These all use database back-ends, typically MySQL
  - 3-column table, for Subject, Predicate, Object
  - usually GraphId too, so table becomes 4-column
  - physical implementations differ
- Relational database query language is SQL
- ⇒ SPARQL has to be translated (invisibly) into SQL

(Note: not all RDF stores use relational dbs.)
Where is RDF stored anyway?

- If your RDF graph is too big to fit in memory...
- ...you store it in a relational database
- Many triple stores available: Jena, Kowari, Sesame,...
- These all use database back-ends, typically MySQL
  - 3-column table, for Subject, Predicate, Object
  - usually GraphId too, so table becomes 4-column
  - physical implementations differ
- Relational database query language is SQL
- \(\Rightarrow\) SPARQL has to be translated (invisibly) into SQL

(Note: not all RDF stores use relational dbs.)
Where is RDF stored anyway?

- If your RDF graph is too big to fit in memory...
- ...you store it in a relational database
- Many triple stores available: Jena, Kowari, Sesame,...
- These all use database back-ends, typically MySQL
  - 3-column table, for Subject, Predicate, Object
  - usually GraphId too, so table becomes 4-column
  - physical implementations differ
- Relational database query language is SQL
  - $\Rightarrow$ SPARQL has to be translated (invisibly) into SQL

(Note: not all RDF stores use relational dbs.)
Where is RDF stored anyway?

• If your RDF graph is too big to fit in memory...
• ...you store it in a relational database
• Many triple stores available: Jena, Kowari, Sesame,...
• These all use database back-ends, typically MySQL
  • 3-column table, for Subject, Predicate, Object
  • usually GraphId too, so table becomes 4-column
  • physical implementations differ
• Relational database query language is SQL
• ⇒ SPARQL has to be translated (invisibly) into SQL

*(Note: not all RDF stores use relational dbs.)*
Storing RDF triples

- traverse links by “self-joins” on table
RDB2RDF conversion problems websites

Terminology

RDF triple:

- subject → predicate → object

(Or, if you prefer, Subject – Verb – Object.)

In database terms:

- rowId → attribute → value

- :site1 → :name → "Dirleton Castle"

- Note that :name just depends on schema designer’s choice...
- ...whereas “Dirleton Castle” is real data
Terminology

RDF triple:

subject ➔ predicate ➔ object

(Or, if you prefer, Subject − Verb − Object.)

In database terms:

rowId ➔ attribute ➔ value

:site1 ➔ :name ➔ "Dirleton Castle"

• Note that :name just depends on schema designer’s choice...
• ...whereas “Dirleton Castle” is real data
Information Management and Relational Databases
some history
relational database basics

Converting Relational Databases to RDF
why would you want to?
how? – the basic conversion procedure

What’s Wrong With the Basic Procedure?
bnodes
literals
redundancy

Useful Websites
The Hidden/Invisible/Deep Web Problem

- Most data is (still) in databases, especially “good” data:
  - carefully curated datasets, built over decades/centuries
  - like CANMORE – architecture and archaeology across Scotland – at http://www.rcahms.gov.uk/
- Web crawlers can’t see inside databases –
- – unless you “expose” individual search results:
Linking Data

- Related databases everywhere:
  - RCAHMS archaeological sites – NMS excavation finds
  - company merger; personnel records

- Interconnecting relational databases is hard:
  - you need to know the schema in detail
  - security issues
  - complex networking protocols – not http
  - whereas RDF was designed for data linking...
Linking Data

• Related databases everywhere:
  • RCAHMS archaeological sites – NMS excavation finds
  • company merger; personnel records

• Interconnecting relational databases is hard:
  • you need to know the schema in detail
    • with SQL, you cannot query without knowing attribute name
    • with SPARQL, you can
  • security issues
  • complex networking protocols – not http
  • whereas RDF was designed for data linking...
Linking Data

- Related databases everywhere:
  - RCAHMS archaeological sites – NMS excavation finds
  - company merger; personnel records

- Interconnecting relational databases is hard:
  - you need to know the schema in detail
    - with SQL, you **cannot** query without knowing attribute name
    - with SPARQL, you can
  - security issues
  - complex networking protocols – not http
  - whereas RDF was designed for data linking...
Linking Data

- Related databases everywhere:
  - RCAHMS archaeological sites – NMS excavation finds
  - company merger; personnel records

- Interconnecting relational databases is hard:
  - you need to know the schema in detail
    - with SQL, you cannot query without knowing attribute name
    - with SPARQL, you can
  - security issues
    - complex networking protocols – not http
    - whereas RDF was designed for data linking...
Linking Data

- Related databases everywhere:
  - RCAHMS archaeological sites – NMS excavation finds
  - company merger; personnel records

- Interconnecting relational databases is hard:
  - you need to know the schema in detail
    - with SQL, you cannot query without knowing attribute name
    - with SPARQL, you can
  - security issues
  - complex networking protocols – not http
  - whereas RDF was designed for data linking...
Linking Data

- Related databases everywhere:
  - RCAHMS archaeological sites – NMS excavation finds
  - company merger; personnel records
- Interconnecting relational databases is hard:
  - you need to know the schema in detail
    - with SQL, you cannot query without knowing attribute name
    - with SPARQL, you can
  - security issues
  - complex networking protocols – not http
  - whereas RDF was designed for data linking...
Dataset Linking in RDF

- Same node appears in two graphs?
- – graphs are automatically linked
Alternatives to Conversion

- You don’t have to instantiate the database as RDF:
  - query SQL database using SPARQL – eg SquirrelRDF, R2D2
  - virtual RDF graph interface for relational dbs – eg D2RQ
  - hybrid “middleware” database engine – eg Virtuoso
- Saves duplicating data – but more work at query time
- All need the database schema available
- Principles same as for full conversion
RDB2RDF conversion problems websites

How to Convert DBs – W3C Guidance

- RDF has been around since 1999...
- From the W3C Semantic Web FAQ site, http://www.w3.org/RDF/FAQ#relodb:

  “How do I export my data from a Relational Database?”

  “This is one of the active areas of R&D, and no final answer is yet available...”

- Lots of W3C activity in late 2008...
How to Convert DBs – W3C Guidance

- RDF has been around since 1999...
- From the **W3C Semantic Web FAQ** site, http://www.w3.org/RDF/FAQ#reldb:

  “How do I export my data from a Relational Database?”

  “This is one of the active areas of R&D, and no final answer is yet available…”

- Lots of W3C activity in late 2008…
How to Convert DBs – W3C Guidance

• RDF has been around since 1999...

• From the **W3C Semantic Web FAQ** site, http://www.w3.org/RDF/FAQ#reldb:

  “How do I export my data from a Relational Database?”

  “This is one of the active areas of R&D, and no final answer is yet available…”

• Lots of W3C activity in late 2008...
W3C RDB2RDF Incubator Group

• Group formed in mid-2008
• Now (end Jan 2009) finalising its report
  • Final report: http://www.w3.org/2005/Incubator/rdb2rdf/XGR/
  • Detailed “StateOfTheArt” discussion: http://esw.w3.org/topic/Rdb2RdfXG/StateOfTheArt
  • Public mailing list: http://lists.w3.org/Archives/Public/public-xg-rdb2rdf/
• Summary of recommendation: *W3C to form WG to standardise a language for RDB2RDF, based around RIF (Rule Interchange Format) principles*

• In the meantime...? Think spider plant.
W3C RDB2RDF Incubator Group

• Group formed in mid-2008
• Now (end Jan 2009) finalising its report
  • Final report: [http://www.w3.org/2005/Incubator/rdb2rdf/XGR/](http://www.w3.org/2005/Incubator/rdb2rdf/XGR/)
  • Detailed “StateOfTheArt” discussion: [http://esw.w3.org/topic/Rdb2RdfXG/StateOfTheArt](http://esw.w3.org/topic/Rdb2RdfXG/StateOfTheArt)
  • Public mailing list: [http://lists.w3.org/Archives/Public/public-xg-rdb2rdf/](http://lists.w3.org/Archives/Public/public-xg-rdb2rdf/)

• Summary of recommendation: *W3C to form WG to standardise a language for RDB2RDF, based around RIF (Rule Interchange Format) principles*

• In the meantime…? Think spider plant.
## DB Conversion – Growing Spider Plants

### Relational Database

<table>
<thead>
<tr>
<th>siteNo</th>
<th>name</th>
<th>parish</th>
<th>status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dirleton Castle</td>
<td>Dirleton</td>
<td>scheduled</td>
</tr>
<tr>
<td>2</td>
<td>Dirleton Cottage</td>
<td>Dirleton</td>
<td>listed</td>
</tr>
<tr>
<td>3</td>
<td>Drem Airfield</td>
<td>Dirleton</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Jamie’s Neuk</td>
<td>Dirleton</td>
<td></td>
</tr>
</tbody>
</table>

### ARCHIVE

<table>
<thead>
<tr>
<th>arcNo</th>
<th>arcType</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>photo</td>
<td>North face</td>
</tr>
<tr>
<td>102</td>
<td>drawing</td>
<td>Site plan</td>
</tr>
<tr>
<td>103</td>
<td>map</td>
<td>Parish map</td>
</tr>
</tbody>
</table>

### RDF

- siteNo
- arcNo
- siteNo
- arcNo
- scheduled
- listed
- Dirleton Castle
- Dirleton Cottage
- Drem Airfield
- Jamie’s Neuk
- North face
- Site plan
- Parish map
A Simple Example – One Database Record

<table>
<thead>
<tr>
<th>siteNo</th>
<th>name</th>
<th>parish</th>
<th>classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dirleton Castle</td>
<td>Dirleton</td>
<td>defence</td>
</tr>
<tr>
<td>2</td>
<td>Dirleton Cottage</td>
<td>Dirleton</td>
<td>residential</td>
</tr>
<tr>
<td>3</td>
<td>Drem Airfield</td>
<td>Dirleton</td>
<td>military</td>
</tr>
<tr>
<td>4</td>
<td>Jamie’s Neuk</td>
<td>Dirleton</td>
<td>military</td>
</tr>
</tbody>
</table>

```
@prefix       :    <http://www.ltg.ed.ac.uk/tether/> .
@prefix   rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
```

- Each row, or **instance**, of SITE forms a new offshoot –
- – central node, surrounded by cluster of attributes
- **Table as Class, Column as Predicate** conversion
Information Management and Relational Databases
  some history
  relational database basics

Converting Relational Databases to RDF
  why would you want to?
  how? – the basic conversion procedure

What’s Wrong With the Basic Procedure?
  bnodes
  literals
  redundancy

Useful Websites
To Bnode or Not to Bnode?

- duck typing – is it good data management?
- primary keys: important data items need direct reference
- needs schema knowledge
To Bnode or Not to Bnode?

- duck typing – is it good data management?
- primary keys: important data items need direct reference
- needs schema knowledge
Literals or Resources?

```
rdfs:Class
:site
rdf:type
rdf:type
"Dirleton Castle"
:name
"Dirleton"
:parish
"defence"
:classification
:SiteNo#1
:dirletonCastle
"East Lothian"
:location
```
Literals or Resources?

- rdfs:Class
- :site
- rdf:type
- :site
- rdf:type
- "Dirleton Castle"
- :classification
- "defence"
- :parish
- "Dirleton"
- :dirletonCastle
- :location
- "East Lothian"
Literals or Resources?
Literals or Resources?

- Avoid literals!
- Graph is sterilised at literals – no further links
- Encode database values as URIs
- Some unlikely URIs:
  - PhotoDesc – '#5: 6″x4″ neg, B&W'
    - http://www.ex.com/Pdesc#%235:%206%22x4%22%20neg%2C%20B%26W
- Take care with URI generation:
  - is http://www.example.com/place/edinburgh the same resource as http://www.example.com/city/edinburgh?
Literals or Resources?

- Avoid literals!
- Graph is sterilised at literals – no further links
- Encode database values as URIs
- Some unlikely URIs:
  - PhotoDesc – ’#5: 6”x4” neg, B&W’
    http://www.ex.com/Pdesc#%235:%206%22x4%22%20neg%2C%20B%26W
- Take care with URI generation:
  - is http://www.example.com/place/edinburgh the same resource as http://www.example.com/city/edinburgh?
Literals or Resources?

- Avoid literals!
- Graph is sterilised at literals – no further links
- Encode database values as URIs
- Some unlikely URIs:
  - PhotoDesc – ’#5: 6” x 4” neg, B&W’
    - http://www.ex.com/Pdesc#%235:%206%22x4%22%20neg%2C%20B%26W
- Take care with URI generation:
  - is http://www.example.com/place/edinburgh the same resource as http://www.example.com/city/edinburgh?
Redundant Nodes at Relational Joins
Repetition of RDB Metadata

- Standard approach duplicates data (bad!):
  - database attribute name becomes RDF predicate
  - database attribute name appears in target node (data source)
  - database attribute name becomes RDF Class name

- Instead:
  - small RDF predicate set, not tied to RDB attributes
  - use RDFS Classes to indicate RDB provenance –
    - Column as Class not Column as Predicate
  - requires a manual RDF design step
Repetition of RDB Metadata

- Standard approach duplicates data (bad!):
  - database attribute name becomes RDF predicate
  - database attribute name appears in target node (data source)
  - database attribute name becomes RDF Class name
- Instead:
  - small RDF predicate set, not tied to RDB attributes
  - use RDFS Classes to indicate RDB provenance –
    - *Column as Class not Column as Predicate*
  - requires a manual RDF design step
### Example of Designed RDF Schema – *Tether*

<table>
<thead>
<tr>
<th>siteNo</th>
<th>name</th>
<th>parish</th>
<th>classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dirleton Castle</td>
<td>Dirleton</td>
<td>defence</td>
</tr>
<tr>
<td>2</td>
<td>Dirleton Cottage</td>
<td>Dirleton</td>
<td>residential</td>
</tr>
<tr>
<td>3</td>
<td>Drem Airfield</td>
<td>Dirleton</td>
<td>military</td>
</tr>
<tr>
<td>4</td>
<td>Jamie’s Neuk</td>
<td>Dirleton</td>
<td>military</td>
</tr>
</tbody>
</table>

```prefix : <http://www.ltg.ed.ac.uk/tether/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>. 
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>. ```
Other Issues We Don’t Have Enough Time For!

- Hash fragments in URIs and “303 redirection”
  - what will your URIs serve, and how?
- Should the conversion process be reversible?
  - full data source references in all URIs
  - is the relational database the master copy?
- Data bloat:
  - up to 3 triples per database value
  - short fields become long URIs
- Null fields
- Coded values
- Data maintenance!
- (See Chapter 5 of my PhD thesis if interested)
Other Issues We Don’t Have Enough Time For!

- Hash fragments in URIs and “303 redirection”
  - what will your URIs serve, and how?
- Should the conversion process be reversible?
  - full data source references in all URIs
  - is the relational database the master copy?
- Data bloat:
  - up to 3 triples per database value
  - short fields become long URIs
- Null fields
- Coded values
- Data maintenance!
- (See Chapter 5 of my PhD thesis if interested)
Other Issues We Don’t Have Enough Time For!

- Hash fragments in URIs and “303 redirection”
  - what will your URIs serve, and how?
- Should the conversion process be reversible?
  - full data source references in all URIs
  - is the relational database the master copy?
- Data bloat:
  - up to 3 triples per database value
  - short fields become long URIs
- Null fields
- Coded values
- Data maintenance!
- (See Chapter 5 of my PhD thesis if interested)
Other Issues We Don’t Have Enough Time For!

- Hash fragments in URIs and “303 redirection”
  - what will your URIs serve, and how?
- Should the conversion process be reversible?
  - full data source references in all URIs
  - is the relational database the master copy?
- Data bloat:
  - up to 3 triples per database value
  - short fields become long URIs
- Null fields
  - Coded values
- Data maintenance!
- (See Chapter 5 of my PhD thesis if interested)
Other Issues We Don’t Have Enough Time For!

- Hash fragments in URIs and “303 redirection”
  - what will your URIs serve, and how?
- Should the conversion process be reversible?
  - full data source references in all URIs
  - is the relational database the master copy?
- Data bloat:
  - up to 3 triples per database value
  - short fields become long URIs
- Null fields
- Coded values
  - Data maintenance!
- (See Chapter 5 of my PhD thesis if interested)
Other Issues We Don’t Have Enough Time For!

- Hash fragments in URIs and “303 redirection”
  - what will your URIs serve, and how?
- Should the conversion process be reversible?
  - full data source references in all URIs
  - is the relational database the master copy?
- Data bloat:
  - up to 3 triples per database value
  - short fields become long URIs
- Null fields
- Coded values
- Data maintenance!

(See Chapter 5 of my PhD thesis if interested)
Other Issues We Don’t Have Enough Time For!

- Hash fragments in URIs and “303 redirection”
  - what will your URIs serve, and how?
- Should the conversion process be reversible?
  - full data source references in all URIs
  - is the relational database the master copy?
- Data bloat:
  - up to 3 triples per database value
  - short fields become long URIs
- Null fields
- Coded values
- Data maintenance!
- (See Chapter 5 of my PhD thesis if interested)
Useful websites

- Tim Berners-Lee design note: http://www.w3.org/DesignIssues/RDB-RDF.html
- RDB2RDF “StateOfTheArt”: http://esw.w3.org/topic/Rdb2RdfXG/StateOfTheArt
- RDFAndSQL Wiki: http://esw.w3.org/topic/RdfAndSql
- Linked Data: http://linkeddata.org/