

NOSQL

Not Only SQL
-> niet relationeel

Enkele Voorbeelden

- Wide Column Store
 - Bv Cassandra, HBase, ScyllaDB
- Document Store
 - Bv MongoDB, Couchbase, Realm
- Graph based
 - Bv Neo4J, Memgraph, NebulaGraph
- Key Value
 - Bv Redis, Berkeley DB, MemcacheDB
- Multivalue
 - Bv SciDB, Rasdaman, OpenQM
- Search Engines
 - Bv Elasticsearch, Solr, OpenSearch

Enkele Voorbeelden

- Time Series DBMS
 - Bv InfluxDB, Prometheus, TimescaleDB
- RDF Stores
 - Bv Apache Jena – TDB, RDF4J, 4store
- Vector DBMS
 - Bv Chroma, Weaviate, Milvus
- Spatial DBMS
 - Bv PostGIS, spatialite, GeoMesa
- Native XML DBMS
 - Bv BaseX, Sedna, eXist-db
- Object oriented DBMS
 - Bv ObjectBox, Perst, WakandaDB

Enkele Voorbeelden

- Event stores
 - Bv EventStoreDB, NeventStore
- Content stores
 - Bv Jackrabbit, ModeShape
- Multi-model DBMS
 - Verschillende RDBMS ondersteunen vaak verschillende paradigma
 - Bv PostgreSQL, MariaDB, ClickHouse

Mongo db

- Binaire JSON
- JSON voorbeeld:
- CRUD
- Index
- Aggregation
- Replication
- Sharding

```
{
  "firstName": "John",
  "lastName" : "Smith",
  "age"       : 25,
  "address"   :
  {
    "streetAddress": "21 2nd
                      Street",
    "city"         : "New York",
    "state"        : "NY",
    "postalCode"   : "10021"
  },
  "phoneNumber":
  [
    {
      "type" : "home",
      "number": "212 555-1234"
    },
    {
      "type" : "fax",
      "number": "646 555-4567"
    }
  ]
}
```

KV

- Look up Value using Key
- Structuur?
- Cf Dictionary (ODMS)
- RDBMS oplossingen..
 - Postgresql : hstore of gewoon..

```
CREATE TABLE kvp
(id    SERIAL NOT NULL,
key   text    NOT NULL,
value text    ,
CONSTRAINT kvp_pk PRIMARY KEY id );
```

```
-- KeyValuePair
```

```
-- index op key maken
```

KV

- Look up Value using Key..
- Cf Dictionary (ODMS)
- RDBMS oplossingen..
 - Postgresql : hstore of gewoon..

BASE

- vs ACID
- Basically Available: zo beschikbaar mogelijk
- Soft-state: geen garantie op consistentie, inhoud kan veranderen zonder invoer van de gebruiker
- Eventually consistent: doel is consistentie, gegeven genoeg tijd

XML

(CC BY-NC-SA 4.0)

Wim.bertels@ucll.be

XML

- eXtensible Markup Language
- DTD (of xml schema)
- 1996
- hierarchische structuren in een nieuw kleedje
- platte tekstbestanden
- ISO standaard
- einde RDBMS..?

eenvoudige XML Syntax

- men gebruikt een tag om aan te geven waar men begint en eindigt
- elke document moet in 1 root tag omvat zijn
- beginnen met een cijfer mag niet
- case sensitive
- bv
 <example> tekst </example>
- tags mogen genest worden

Voorbeeld

<energie>

<gerecht>

<naam> frieten </naam>

<quoting> zeer wel </quoting>

</gerecht>

<gerecht>

<naam> sojapuddingzonderchoco </naam>

<quoting> twijfelachtig </quoting>

</gerecht>

</energie>

Doel

- xml (data) vs html (vorm)
(xsl vs css)
- om zowel hierarchische datastructuren te omschrijven als deze te bevatten
- om data uit wisselen tussen verschillende gegevensbronnen (cf s-patroon:mediator)
 - een gemeenschappelijke taal
 - geef hiervan een voorbeeld.

METADATA

- je kan attributen aan je tags meegeven, deze moeten tussen " " of ' ' staan
- gebruik dit enkel om METADATA weer te geven, bv eigenschappen
- de data zelf horen daar niet thuis
- bv

`<example type='music'> lalala.mp3 </example>`

Uitbreidingen

- xml word bv ook gebruikt om configuraties van software bij te houden bv menubalk, ..
- er zijn verschillende formaten die van deze standaard gebruik maken:
xhtml, xml dom, xsl, xslt, xpath, xsl-fo, xlink, xpointer, dtd, xsd, xforms, xquery, soap, wsd, rdf, rss, svg, wap, smil, ..

Pro Contra

- ISO
 - uitwisselbaarheid
 - eenvoud
 - geen hiërarchische engine nodig
 - ..
- redundantie tov relationeel model, dus niet misbruiken in die zin
 - sequentieel, traag
 - ..

RDBMS

- Verschillende RDBMS voorzien manieren om xml te gebruiken
 - <http://www.postgresql.org/docs/current/static/datatype-xml.html>
 - <http://www.postgresql.org/docs/current/static/functions-xml.html>

Voorbeelden

```
SELECT xmlcomment('hallo');
```

xmlcomment

<!--hallo-->

```
SELECT xmlelement(name jos,  
  xmlattributes(current_date as ke), 'hal', 'lo');
```

xmlelement

<jos ke="2014-01-26">hallo</jos>

```
select xmlforest(divisie, teamnr) from teams;
```

xmlforest

```
<divisie>ere </divisie><teamnr>1</teamnr>  
<divisie>tweede</divisie><teamnr>2</teamnr>
```

```
SELECT xmlpi(name php, 'echo "Patat";');
```

xmlpi

```
<?php echo "Patat";?>
```

..
table_to_xml(tbl regclass, nulls boolean,
tableforest boolean, targetns text)

query_to_xml(query text, nulls boolean,
tableforest boolean, targetns text)

cursor_to_xml(cursor refcursor, count int, nulls
boolean, tableforest boolean, targetns text)

..

Links

- <http://www.postgresql.org/docs/current/static/datatype-xml.html>
- <http://www.postgresql.org/docs/current/static/functions-xml.html>

Wim Bertels

Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International Public License