Cassandra report - DBLP

MANIP M'EBOBISSE Séthi - MOHAMED Soilhat - Ngatcha Nancy -ROMANO Léa

27 janvier 2019

1. How we import the dataset in the NoSQL database

1. Schema

First, we need to determine the schema on witch we'll construct our database. After reading the file and analyzing the different keys and values type, we came to a first schema:

```
CREATE TYPE pagesType (
   start INT,
   end INT
);

CREATE TYPE journalType (
   series VARCHAR,
   editor VARCHAR,
   volume VARCHAR,
   isbn LIST<VARCHAR>
);

CREATE TABLE IF NOT EXISTS DBLP (
   id VARCHAR,
   type VARCHAR,
   year INT,
   title VARCHAR,
   authors LIST<VARCHAR>,
   pages frozen<pagesType>,
   booktitle VARCHAR,
   journal frozen<journalType>,
   url VARCHAR,
   cites LIST <VARCHAR>,
   PRIMARY KEY(id)
);
```

2. Program for the dataset

We decided to write the loading program in python :

```
import json
```

```
from cassandra.cluster import Cluster

cluster = Cluster(['127.0.0.1'])

session = cluster.connect()

session.execute("CREATE KEYSPACE IF NOT EXISTS DBLP WITH REPLICATION =
{'class':'SimpleStrategy','replication_factor':3};")

session.set_keyspace('dblp')

session.execute("CREATE TYPE IF NOT EXISTS pagesType ( \

            start INT, \

            end INT\

          );")

session.execute("CREATE TYPE IF NOT EXISTS journalType ( \

          series VARCHAR,\

          editor VARCHAR,\

          volume VARCHAR,\

          isbn LIST<VARCHAR>\

         );")

session.execute("CREATE TABLE IF NOT EXISTS DBLP ( \

          id VARCHAR, \

          type VARCHAR,\

          year INT, \

          title VARCHAR,\

          authors LIST<VARCHAR>,\

          pages frozen<pagesType>,\

          booktitle VARCHAR, \

          journal frozen<journalType>,\

          url VARCHAR, \

          cites LIST<VARCHAR>,\

          PRIMARY KEY(id) \

         );")
```

```
session.execute('TRUNCATE dblp;')

with open('DBLP_clean.json', 'r') as file:

    for data in file.readlines():

        dataJSON = json.loads(data.replace("'", "''"))

        dataJSON['year'] = int(dataJSON['year'])

        if (dataJSON['pages']['start'] != None):

            dataJSON['pages']['start']= int(dataJSON['pages']['start'])

        if (dataJSON['pages']['end'] != None):

            dataJSON['pages']['end']= int(dataJSON['pages']['end'])

        data = str(dataJSON)

        data = data.replace("'", "")

        data = data.replace("''", "''")

        data = data.replace('\n', "")

        data = data.replace("_id", "id")

        data = data.replace('None', 'null')

        data = data.replace("'\"", "\"'")

        data = data.replace("\"',","'\",")

        data = data.replace("u\"","\"")



        statement = "INSERT INTO dblp JSON '"+ data + "';"

        session.execute(statement)
```

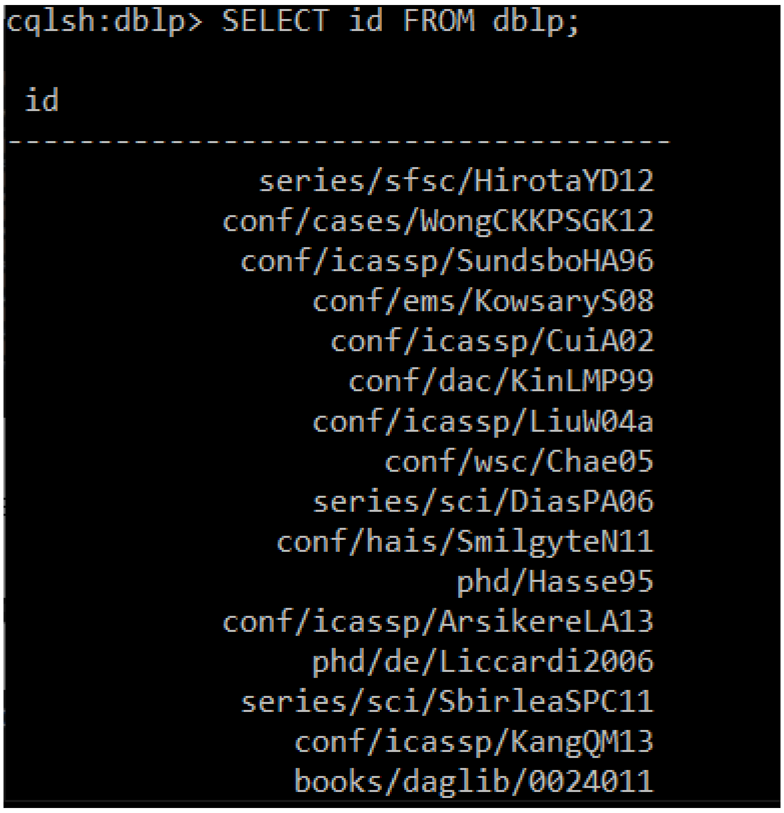
At each step of the loading we had to add changes to the program so that it cleans all the possible “error” we found.

2. Queries

#Simple queries (type exercise 1)

1. Find all the ids.

SELECT id FROM dblp ;

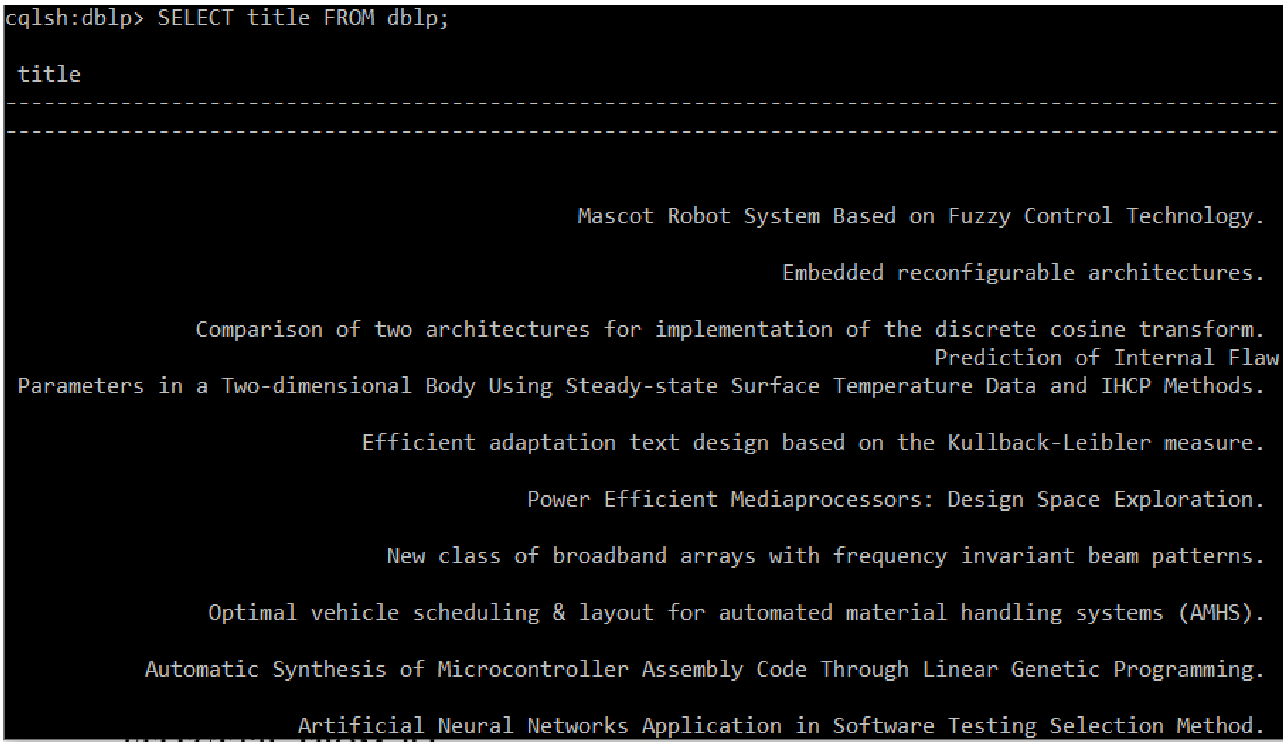


```
cqlsh:dblp> SELECT id FROM dblp;

 id
-------------------------------------
                series/sfsc/HirotaYD12
              conf/cases/WongCKKPSGK12
               conf/icassp/SundsboHA96
                   conf/ems/KowsaryS08
                  conf/icassp/CuiA02
                     conf/dac/KinLMP99
                  conf/icassp/LiuW04a
                      conf/wsc/Chae05
                  series/sci/DiasPA06
                conf/hais/SmilgyteN11
                         phd/Hasse95
              conf/icassp/ArsikereLA13
                   phd/de/Liccardi2006
               series/sci/SbirleaSPC11
                   conf/icassp/KangQM13
                   books/daglib/0024011
```

Here is an extract of the actual output

2. Find all the titles of publications

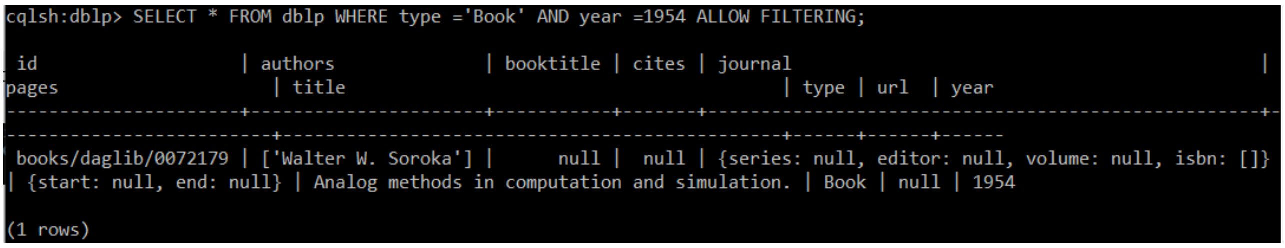
SELECT title FROM dblp;



```
cqlsh:dblp> SELECT title FROM dblp;

 title
--------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------


                                                  Mascot Robot System Based on Fuzzy Control Technology.

                                                                Embedded reconfigurable architectures.

                          Comparison of two architectures for implementation of the discrete cosine transform.
                                                                              Prediction of Internal Flaw
 Parameters in a Two-dimensional Body Using Steady-state Surface Temperature Data and IHCP Methods.

                                        Efficient adaptation text design based on the Kullback-Leibler measure.

                                                     Power Efficient Mediaprocessors: Design Space Exploration.

                                          New class of broadband arrays with frequency invariant beam patterns.

                           Optimal vehicle scheduling & layout for automated material handling systems (AMHS).

                       Automatic Synthesis of Microcontroller Assembly Code Through Linear Genetic Programming.

                                   Artificial Neural Networks Application in Software Testing Selection Method.
```

3. Find all books written in 1954.

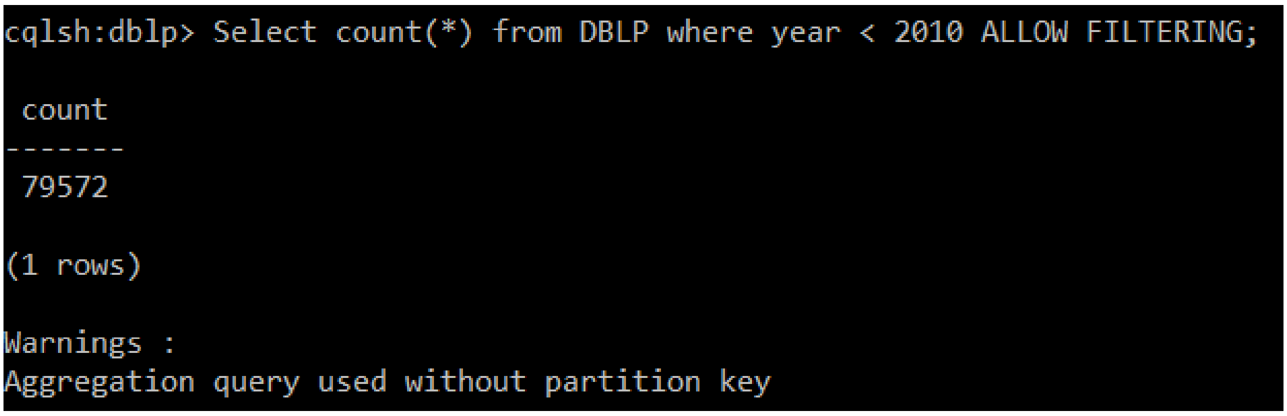
SELECT * FROM dblp WHERE type ='Book' AND year = 1954 ALLOW FILTERING;



```
cqlsh:dblp> SELECT * FROM dblp WHERE type ='Book' AND year =1954 ALLOW FILTERING;

 id                   | authors              | booktitle | cites | journal                                             |
pages                    | title                                          | type | url  | year
----------------------+----------------------+-----------+-------+-----------------------------------------------------+-
------------------------+------------------------------------------------+------+------+------
 books/daglib/0072179 | ['Walter W. Soroka'] |      null |  null | {series: null, editor: null, volume: null, isbn: []}
| {start: null, end: null} | Analog methods in computation and simulation. | Book | null | 1954

(1 rows)
```

4. Find the number of books or articles written before 2010

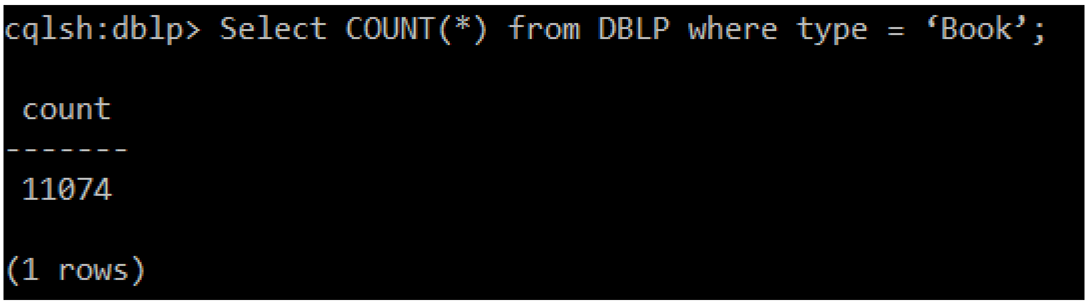
SELECT count(*) FROM dblp WHERE year < 2010 ALLOW FILTERING;



```
cqlsh:dblp> Select count(*) from DBLP where year < 2010 ALLOW FILTERING;

 count
-------
 79572

(1 rows)

Warnings :
Aggregation query used without partition key
```

5. Find the number of books in the database

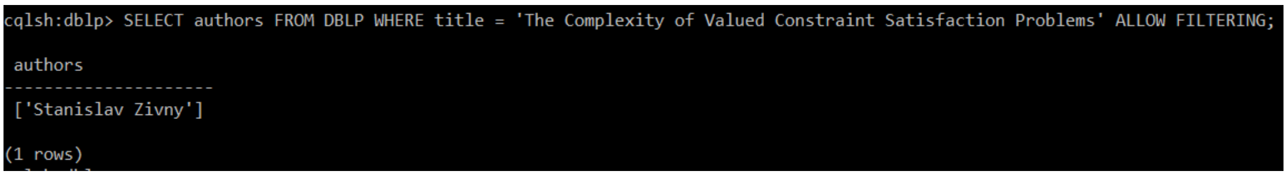
Such as we already create an index on type we don’t need to allow filtereing.
SELECT COUNT(*) FROM dblp WHERE type = ‘Book’;



```
cqlsh:dblp> Select COUNT(*) from DBLP where type = ‘Book’;

 count
-------
 11074

(1 rows)
```

6. Find the authors of the book "The Complexity of Valued Constraint Satisfaction Problems »

SELECT authors FROM DBLP WHERE title = 'The Complexity of Valued Constraint Satisfaction Problems' ALLOW FILTERING;

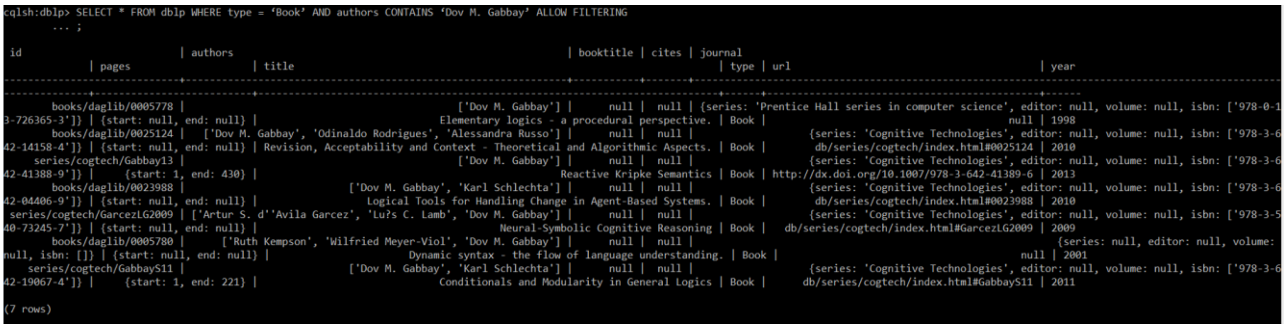


```
cqlsh:dblp> SELECT authors FROM DBLP WHERE title = 'The Complexity of Valued Constraint Satisfaction Problems' ALLOW FILTERING;

 authors
---------------------
 ['Stanislav Zivny']

(1 rows)
```

#Complex queries (type exercice 2)

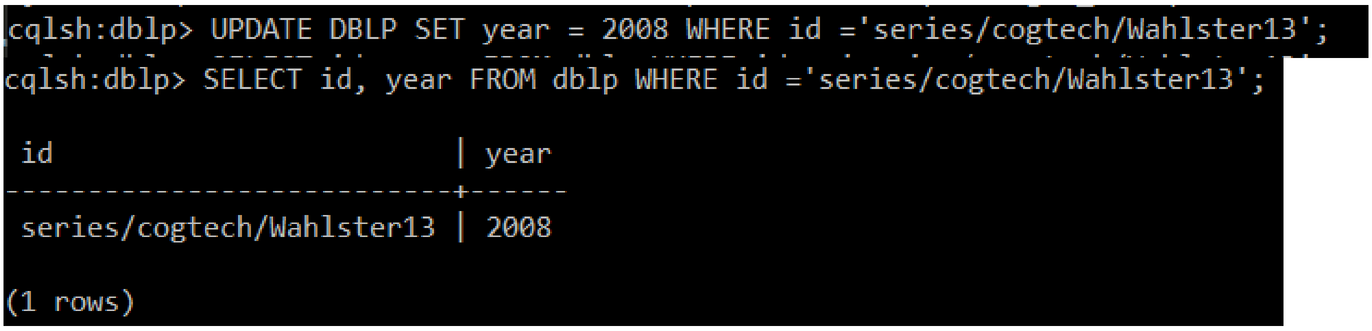
1. Find all books written by Dov M. Gabbay.

CREATE INDEX ON dblp (authors);
CREATE INDEX ON dblp(type);
SELECT * FROM dblp WHERE type = ‘Book’ AND authors CONTAINS ‘Dov M. Gabbay’ ALLOW FILTERING;



```
cqlsh:dblp> SELECT * FROM dblp WHERE type = ‘Book’ AND authors CONTAINS ‘Dov M. Gabbay’ ALLOW FILTERING
        ... ;

 id                          | authors                                                        | booktitle | cites | journal
              | pages                    | title                                                                          | type | url                                          | year
-----------------------------+----------------------------------------------------------------+-----------+-------+------------------------------------------------------------------------------------------------------------
--------------+--------------------------+--------------------------------------------------------------------------------+------+----------------------------------------------+------
        books/daglib/0005778 |                                                 ['Dov M. Gabbay'] |      null |  null | {series: 'Prentice Hall series in computer science', editor: null, volume: null, isbn: ['978-0-1
3-726365-3']} | {start: null, end: null} |                                            Elementary logics - a procedural perspective. | Book |                                         null | 1998
        books/daglib/0025124 |   ['Dov M. Gabbay', 'Odinaldo Rodrigues', 'Alessandra Russo'] |      null |  null |           {series: 'Cognitive Technologies', editor: null, volume: null, isbn: ['978-3-6
42-14158-4']} | {start: null, end: null} | Revision, Acceptability and Context - Theoretical and Algorithmic Aspects. | Book |       db/series/cogtech/index.html#0025124 | 2010
     series/cogtech/Gabbay13 |                                                 ['Dov M. Gabbay'] |      null |  null |           {series: 'Cognitive Technologies', editor: null, volume: null, isbn: ['978-3-6
42-41388-9']} |    {start: 1, end: 430} |                                                   Reactive Kripke Semantics | Book | http://dx.doi.org/10.1007/978-3-642-41389-6 | 2013
        books/daglib/0023988 |                               ['Dov M. Gabbay', 'Karl Schlechta'] |      null |  null |           {series: 'Cognitive Technologies', editor: null, volume: null, isbn: ['978-3-6
42-04406-9']} | {start: null, end: null} |                    Logical Tools for Handling Change in Agent-Based Systems. | Book |       db/series/cogtech/index.html#0023988 | 2010
 series/cogtech/GarcezLG2009 | ['Artur S. d''Avila Garcez', 'Lu?s C. Lamb', 'Dov M. Gabbay'] |      null |  null |           {series: 'Cognitive Technologies', editor: null, volume: null, isbn: ['978-3-5
40-73245-7']} | {start: null, end: null} |                                      Neural-Symbolic Cognitive Reasoning | Book |   db/series/cogtech/index.html#GarcezLG2009 | 2009
        books/daglib/0005780 |      ['Ruth Kempson', 'Wilfried Meyer-Viol', 'Dov M. Gabbay'] |      null |  null |                                             {series: null, editor: null, volume:
null, isbn: []} | {start: null, end: null} |                  Dynamic syntax - the flow of language understanding. | Book |                                         null | 2001
   series/cogtech/GabbayS11 |                               ['Dov M. Gabbay', 'Karl Schlechta'] |      null |  null |           {series: 'Cognitive Technologies', editor: null, volume: null, isbn: ['978-3-6
42-19067-4']} |    {start: 1, end: 221} |                            Conditionals and Modularity in General Logics | Book |      db/series/cogtech/index.html#GabbayS11 | 2011

(7 rows)
```

2. Update the year of a book depending on it's id.
UPDATE DBLP SET year = 2008 WHERE id ='series/cogtech/Wahlster13';



```
cqlsh:dblp> UPDATE DBLP SET year = 2008 WHERE id ='series/cogtech/Wahlster13';
cqlsh:dblp> SELECT id, year FROM dblp WHERE id ='series/cogtech/Wahlster13';

 id                         | year
----------------------------+------
 series/cogtech/Wahlster13  | 2008

(1 rows)
```

#Hard query (type exercice 3)

1. Create a new UDA to produce an equivalence to *"GROUP BY + COUNT"* on textual attributes

CREATE OR REPLACE FUNCTION state_group(state map<text, int>, type text) CALLED ON NULL INPUT RETURNS map<text, int> LANGUAGE java AS ' Integer val = (Integer) state.get(type); if (val == null) val = 0; else val++; state.put(type, val); return state; ' ;

CREATE OR REPLACE AGGREGATE state_group_and_max(text) SFUNC state_group STYPE map<text, int> INITCOND {};

We had trouble testing this function on Cassandra but normally it works.