ArangoDB

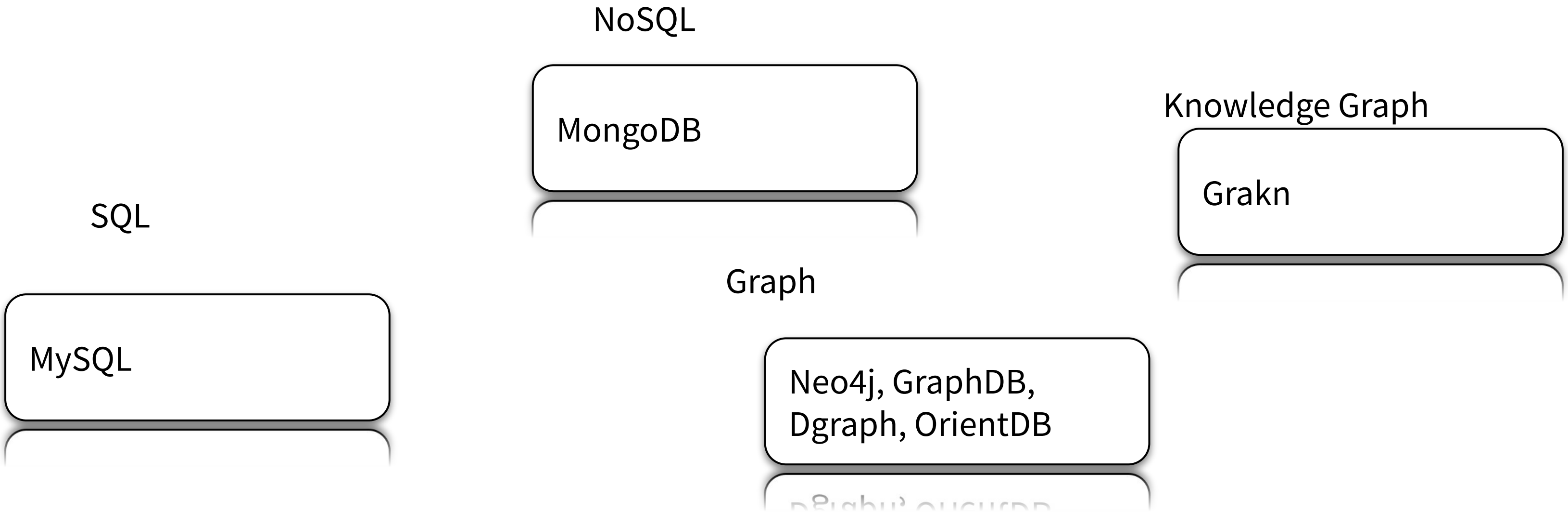
case for graphs

Alexander Belikov

Outline

1. Introduction
2. Arango setup
3. Practicalities
4. Toy model design: Publications
5. Interesting queries

Zooniverse of DBs



NoSQL

MongoDB

Knowledge Graph

Grakn

SQL

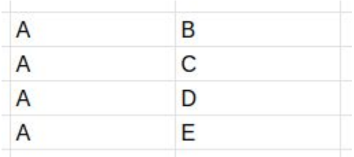
Graph

MySQL

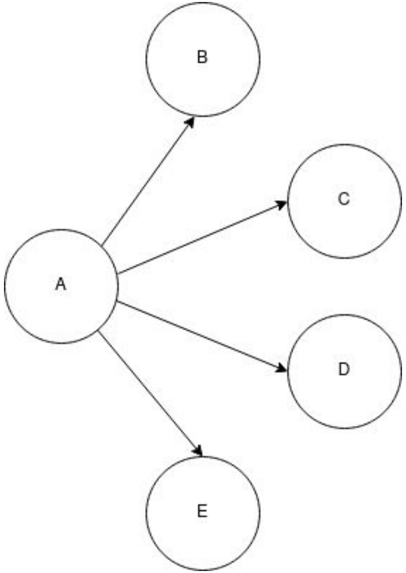
Neo4j, GraphDB, Dgraph, OrientDB

Advantages of graph representations

1. Flexible representation of relations: 1:n



A	B
A	C
A	D
A	E



2. Deep-link traversal.
3. Live updates

Arango: Installation

https://www.arangodb.com/download-major/ubuntu/

Available for major platforms, docker (through apt store on Deb systems).

Similar to mongodb. Starts as a service, uses js as shell language.

Practicalities

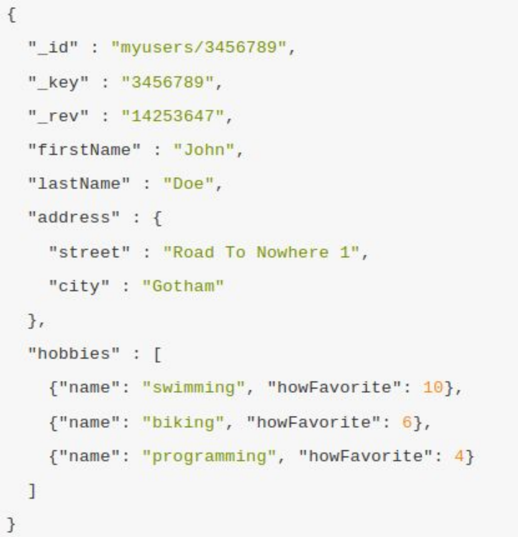
1. pyarango (https://pyarango.readthedocs.io/en/latest/)
2. aql (arango query language) (https://www.arangodb.com/docs/stable/aql/tutorial.html)

Recipe:

manage db from python; ingest using string aql queries sent from python

Document organization

Similar to Mongo, document is a json object:



```json
{
  "_id" : "myusers/3456789",
  "_key" : "3456789",
  "_rev" : "14253647",
  "firstName" : "John",
  "lastName" : "Doe",
  "address" : {
    "street" : "Road To Nowhere 1",
    "city" : "Gotham"
  },
  "hobbies" : [
    {"name": "swimming", "howFavorite": 10},
    {"name": "biking", "howFavorite": 6},
    {"name": "programming", "howFavorite": 4}
  ]
}
```

_key: in-collection id (can be specified by user (!))

_id: global id

Collection organization

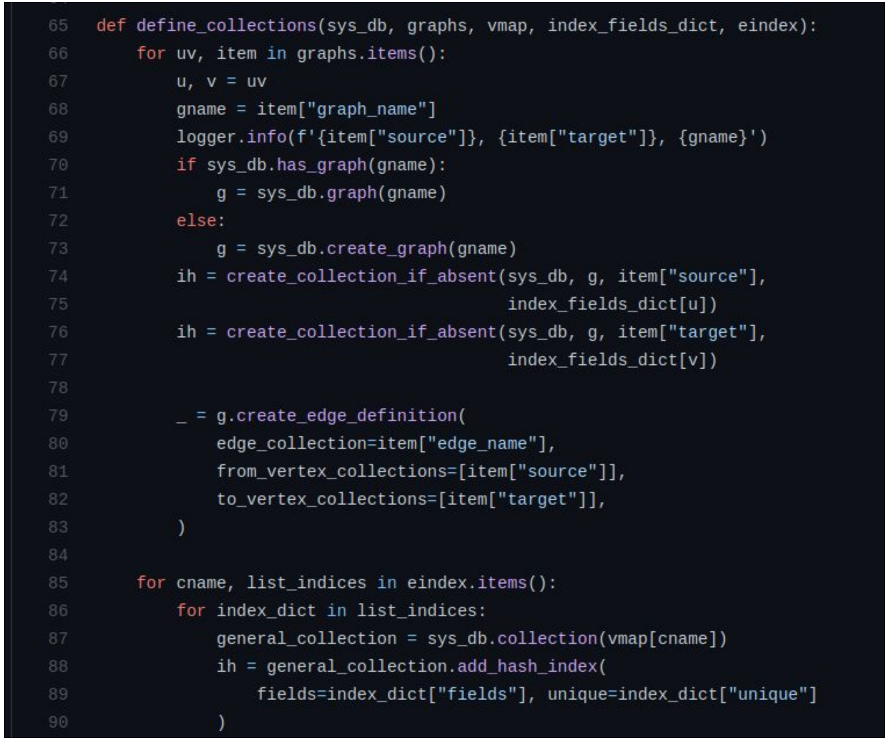
General collection.

Vertex collection.

Graph.

Edge collection.

NB: graph namespace limits graph traversal queries.



```python
def define_collections(sys_db, graphs, vmap, index_fields_dict, eindex):
    for uv, item in graphs.items():
        u, v = uv
        gname = item["graph_name"]
        logger.info(f'{item["source"]}, {item["target"]}, {gname}')
        if sys_db.has_graph(gname):
            g = sys_db.graph(gname)
        else:
            g = sys_db.create_graph(gname)
        ih = create_collection_if_absent(sys_db, g, item["source"],
                                         index_fields_dict[u])
        ih = create_collection_if_absent(sys_db, g, item["target"],
                                         index_fields_dict[v])

        _ = g.create_edge_definition(
            edge_collection=item["edge_name"],
            from_vertex_collections=[item["source"]],
            to_vertex_collections=[item["target"]],
        )

    for cname, list_indices in eindex.items():
        for index_dict in list_indices:
            general_collection = sys_db.collection(vmap[cname])
            ih = general_collection.add_hash_index(
                fields=index_dict["fields"], unique=index_dict["unique"]
            )
```

Features. Upsert.

```
UPSERT searchExpression INSERT insertExpression UPDATE updateExpression IN collection options
```

```
UPSERT searchExpression INSERT insertExpression REPLACE updateExpression IN collection options
```

```
UPSERT { name: 'superuser' }

INSERT { name: 'superuser', logins: 1, dateCreated: DATE_NOW() }

UPDATE { logins: OLD.logins + 1 } IN users
```

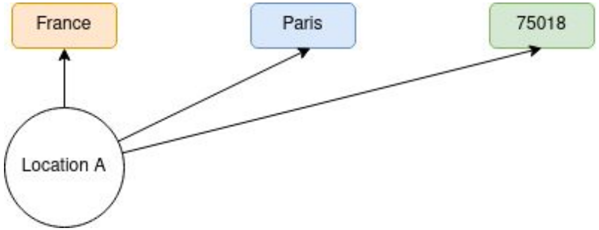
Architecture concerns

What should be a field and what should be a collection?

Sometimes it is useful to use blank nodes

Location: {country: France, city: Paris, postal_code: 75018}

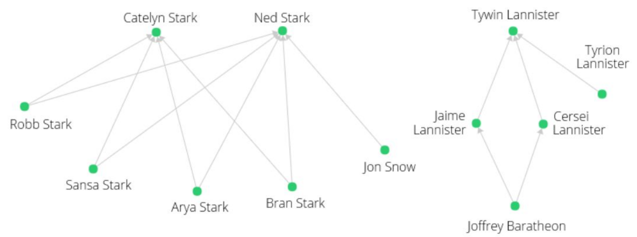
vs



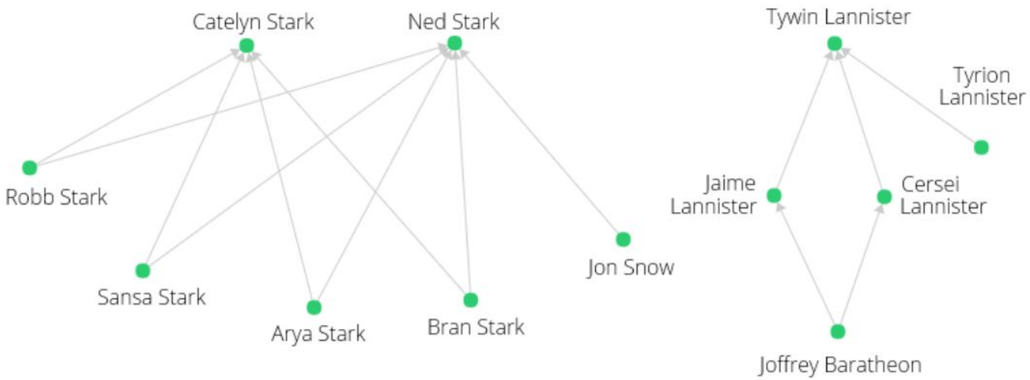
Queries. Graph traversal

```
LET data = [

  {

    "parent": { "name": "Ned", "surname": "Stark" },

    "child": { "name": "Robb", "surname": "Stark" }

  }, {

    "parent": { "name": "Ned", "surname": "Stark" },

    "child": { "name": "Sansa", "surname": "Stark" }

  }, {

    "parent": { "name": "Ned", "surname": "Stark" },

    "child": { "name": "Arya", "surname": "Stark" }

  }, {

    "parent": { "name": "Ned", "surname": "Stark" },

    "child": { "name": "Bran", "surname": "Stark" }

  },

...
```

```
FOR rel in data
    LET parentId = FIRST(
        FOR c IN Characters
            FILTER c.name == rel.parent.name
            FILTER c.surname == rel.parent.surname
            LIMIT 1
            RETURN c._id
    )
    LET childId = FIRST(
        FOR c IN Characters
            FILTER c.name == rel.child.name
            FILTER c.surname == rel.child.surname
            LIMIT 1
            RETURN c._id
    )
    FILTER parentId != null AND childId != null
    INSERT { _from: childId, _to: parentId } INTO ChildOf
    RETURN NEW
```



Queries. Graph traversal



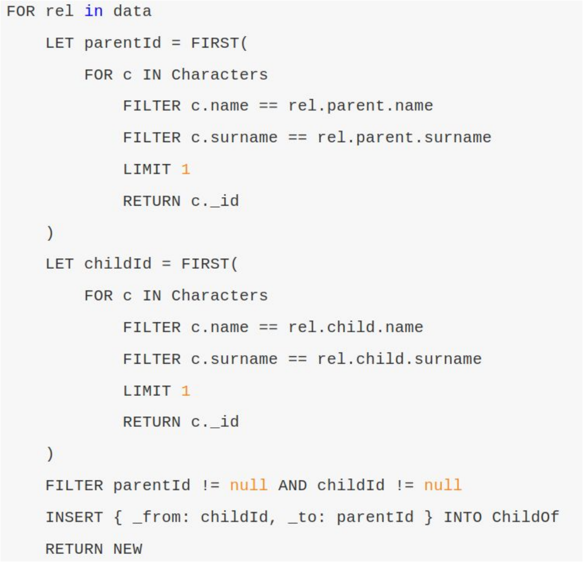
```
FOR v IN 1..1 OUTBOUND "Characters/2901776" ChildOf
    RETURN v.name
```

```
FOR c IN Characters
    FILTER c.name == "Bran"
    FOR v IN 1..1 OUTBOUND c ChildOf
        RETURN v.name
```

```
[
  "Ned",
  "Catelyn"
]
```

```
FOR c IN Characters
    FILTER c.name == "Tywin"
    FOR v IN 2..2 INBOUND c ChildOf
        RETURN v.name
```

Edge definition

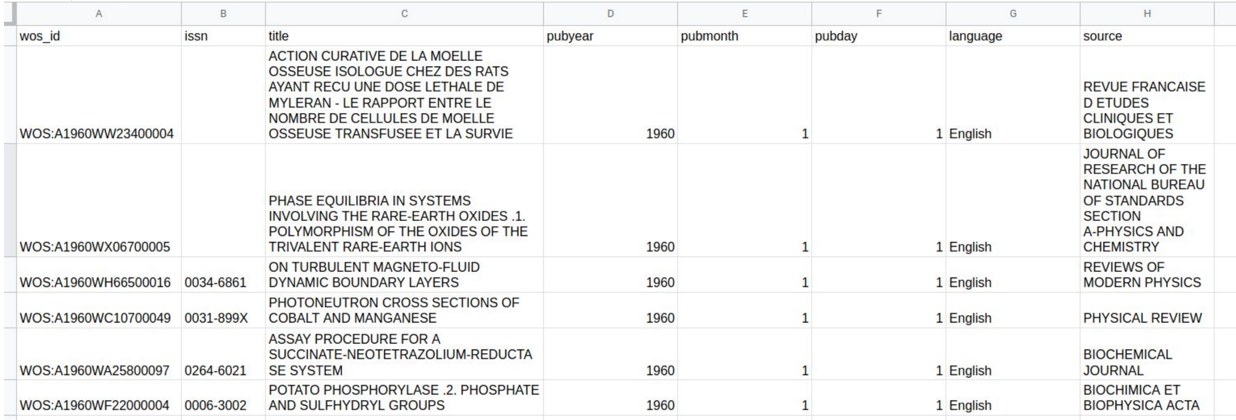


```
FOR rel in data
    LET parentId = FIRST(
        FOR c IN Characters
            FILTER c.name == rel.parent.name
            FILTER c.surname == rel.parent.surname
            LIMIT 1
            RETURN c._id
    )
    LET childId = FIRST(
        FOR c IN Characters
            FILTER c.name == rel.child.name
            FILTER c.surname == rel.child.surname
            LIMIT 1
            RETURN c._id
    )
    FILTER parentId != null AND childId != null
    INSERT { _from: childId, _to: parentId } INTO ChildOf
    RETURN NEW
```

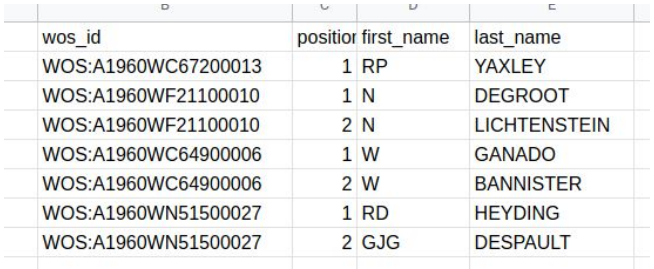
If vertices are completely specified by theirs id - the cost adding edges diminishes greatly! Otherwise ids have to be looked up.

Add attributes to edges: create extra structure, e.g. order.

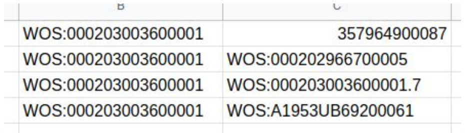
Toy model : publications



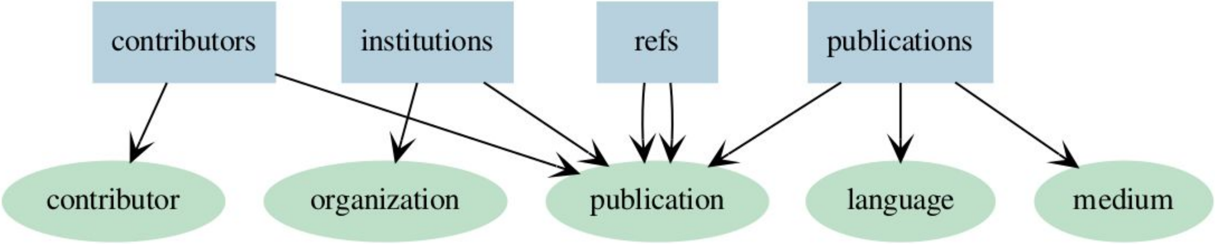
A	B	C	D	E	F	G	H
wos_id	issn	title	pubyear	pubmonth	pubday	language	source
WOS:A1960WW23400004		ACTION CURATIVE DE LA MOELLE OSSEUSE ISOLOGUE CHEZ DES RATS AYANT RECU UNE DOSE LETHALE DE MYLERAN - LE RAPPORT ENTRE LE NOMBRE DE CELLULES DE MOELLE OSSEUSE TRANSFUSEE ET LA SURVIE	1960	1	1	English	REVUE FRANCAISE D ETUDES CLINIQUES ET BIOLOGIQUES
WOS:A1960WX06700005		PHASE EQUILIBRIA IN SYSTEMS INVOLVING THE RARE-EARTH OXIDES .1. POLYMORPHISM OF THE OXIDES OF THE TRIVALENT RARE-EARTH IONS	1960	1	1	English	JOURNAL OF RESEARCH OF THE NATIONAL BUREAU OF STANDARDS SECTION A-PHYSICS AND CHEMISTRY
WOS:A1960WH66500016	0034-6861	ON TURBULENT MAGNETO-FLUID DYNAMIC BOUNDARY LAYERS	1960	1	1	English	REVIEWS OF MODERN PHYSICS
WOS:A1960WC10700049	0031-899X	PHOTONEUTRON CROSS SECTIONS OF COBALT AND MANGANESE	1960	1	1	English	PHYSICAL REVIEW
WOS:A1960WA25800097	0264-6021	ASSAY PROCEDURE FOR A SUCCINATE-NEOTETRAZOLIUM-REDUCTASE SYSTEM	1960	1	1	English	BIOCHEMICAL JOURNAL
WOS:A1960WF22000004	0006-3002	POTATO PHOSPHORYLASE .2. PHOSPHATE AND SULFHYDRYL GROUPS	1960	1	1	English	BIOCHIMICA ET BIOPHYSICA ACTA



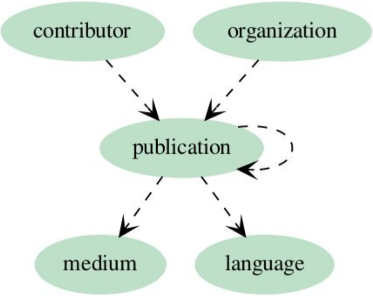
B	C	D	E
wos_id	positior	first_name	last_name
WOS:A1960WC67200013	1	RP	YAXLEY
WOS:A1960WF21100010	1	N	DEGROOT
WOS:A1960WF21100010	2	N	LICHTENSTEIN
WOS:A1960WC64900006	1	W	GANADO
WOS:A1960WC64900006	2	W	BANNISTER
WOS:A1960WN51500027	1	RD	HEYDING
WOS:A1960WN51500027	2	GJG	DESPAULT

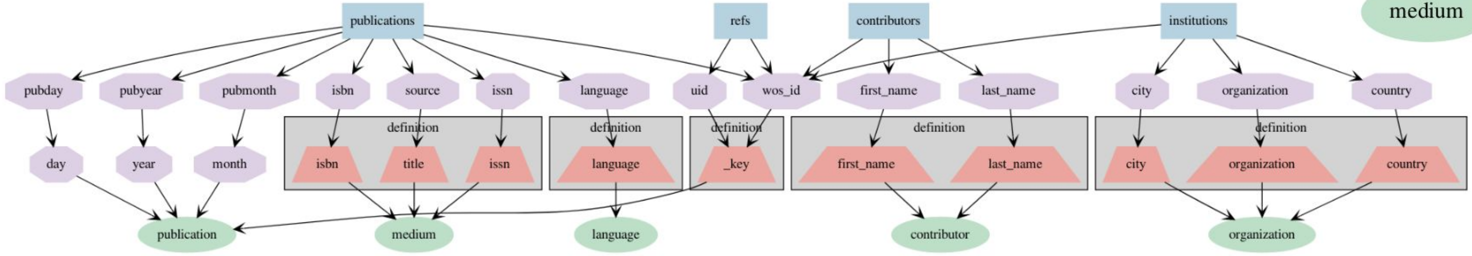


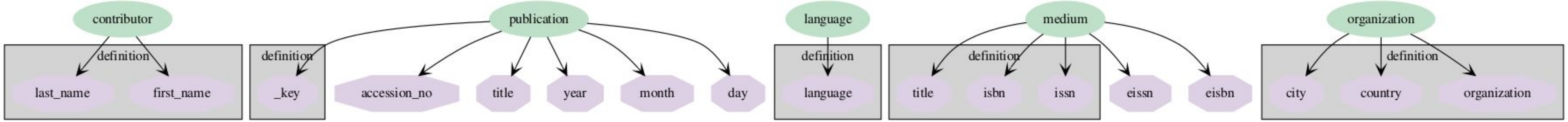
B	C
WOS:000203003600001	357964900087
WOS:000203003600001	WOS:000202966700005
WOS:000203003600001	WOS:000203003600001.7
WOS:000203003600001	WOS:A1953UB69200061



Collection mapping







Queries

Q1: the most popular journals by number of publications for 1978.

Q2: 1000 most popular words (minus stop words) from all available titles.

Q3: authors who changed their country more than twice.

Q4: for publication p compute the ratio of number of second order neighbors to first order neighbors in the directed network of citations.

Q5: count the number of times publications from journal *J* published in 1978 cite publications in journal *J*' published in period [1973, 1978).

Q6: given a subset of publications, compute the cardinality of the power set defined as papers cited by p, papers that are cited by papers cited by p etc of order 5.

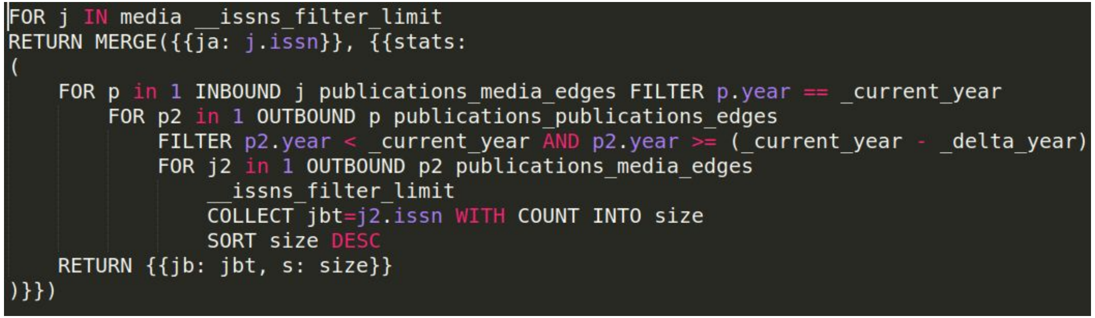
Query 5

Known at EigenFactor ™ (http://www.eigenfactor.org)

A cousin of Google PageRank.

$$M_{ij} = \frac{Z_{ij}}{\sum_k Z_{kj}} \qquad \mathbf{P} = \alpha\mathbf{M} + (1-\alpha)\mathbf{A},$$

make it irreducible and aperiodic. Then there is a unique stationary distributions, given by the eigenvector of unity.



```
FOR j IN media __issns_filter_limit
RETURN MERGE({{ja: j.issn}}, {{stats:
(
    FOR p in 1 INBOUND j publications_media_edges FILTER p.year == _current_year
        FOR p2 in 1 OUTBOUND p publications_publications_edges
            FILTER p2.year < _current_year AND p2.year >= (_current_year - _delta_year)
            FOR j2 in 1 OUTBOUND p2 publications_media_edges
                __issns_filter_limit
                COLLECT jbt=j2.issn WITH COUNT INTO size
                SORT size DESC
    RETURN {{jb: jbt, s: size}}
)}})
```

Query results

Arango vs SQL

16 Gb vs 128 Gb

Q1: return the most popular journals by number of publications for 1978.

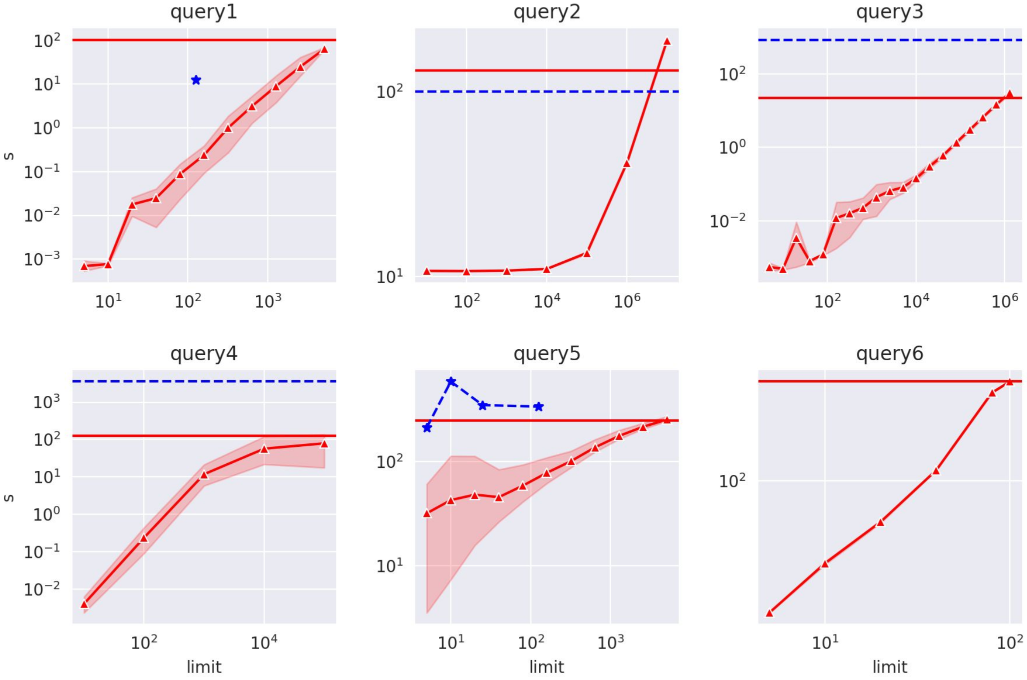
Q2: return 1000 most popular words (minus stop words) from all available titles.

Q3: return the authors who changed their country more than twice.

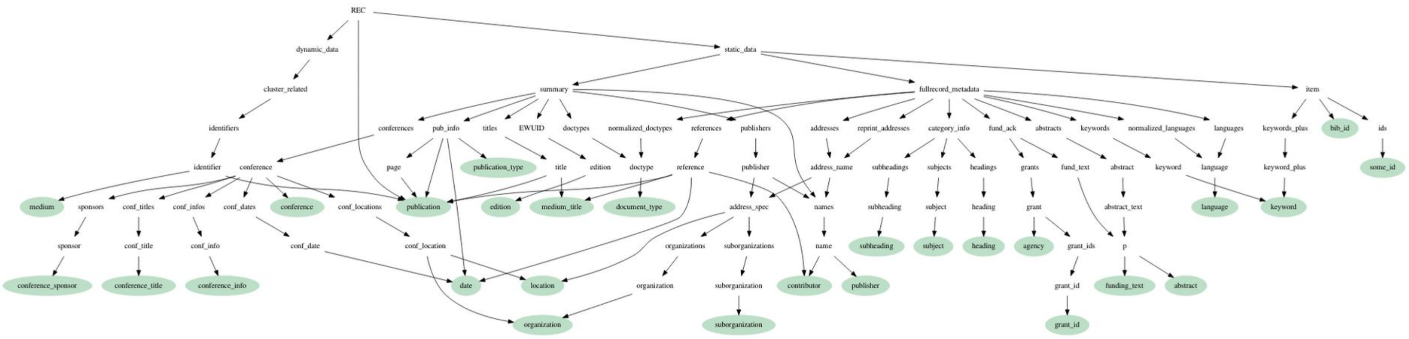
Q4: for publication p compute the ratio of number of second order neighbors to first order neighbors in the directed network of citations.

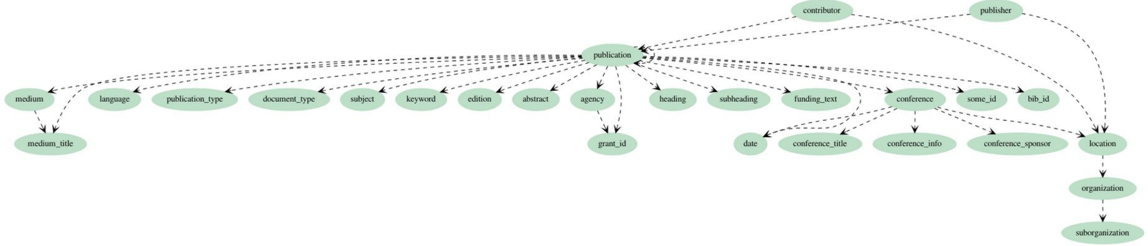
Q5: count the number of times publications from journal j published in 1978 cite publications in journal j' published in period [1973, 1978).

Q6: given a subset of publications, compute the cardinality of the power set defined as papers cited by p, papers that are cited by papers cited by p etc of order 5. As the subset of publications we take 100 publication from query 4 with the highest ratio.



Bonus: full WoS schema


REC
dynamic_data
static_data
cluster_related
summary
fullrecord_metadata
item
identifiers
conferences
pub_info
titles
EWUID
doctypes
normalized_doctypes
references
publishers
addresses
reprint_addresses
category_info
fund_ack
abstracts
keywords
normalized_languages
languages
keywords_plus
bib_id
ids
identifier
conference
page
publication_type
title
edition
doctype
reference
publisher
address_name
subheadings
subjects
headings
grants
fund_text
abstract
keyword
language
keyword_plus
some_id
medium
sponsors
conf_titles
conf_infos
conf_dates
conference
conf_locations
publication
edition
medium_title
document_type
address_spec
names
subheading
subject
heading
grant
abstract_text
language
keyword
sponsor
conf_title
conf_info
conf_date
conf_location
organizations
suborganizations
name
subheading
subject
heading
agency
grant_ids
p
conference_sponsor
conference_title
conference_info
date
location
organization
suborganization
contributor
publisher
grant_id
funding_text
abstract
organization
suborganization
grant_id


contributor
publisher
publication
medium
language
publication_type
document_type
subject
keyword
edition
abstract
agency
heading
subheading
funding_text
conference
some_id
bib_id
medium_title
grant_id
date
conference_title
conference_info
conference_sponsor
location
organization
suborganization