Неструктурирани бази на податоци

Компаративна анализа на **Релациони** и **Граф-базиранани** бази на податоци

Марија Вецовска

185008

Вовед

Целта на овој проект е да се направи компаративна анализа на две бази на податоци од различен тип – неструктурирана и релациона. Анализата се состои од извршување на прашалници врз истите податоци во двете бази и споредба на брзината на извршување.
Податочното множество е Identifying Influential Bloggers: Techcrunch и е преземено од Kaggle.
Множеството е достапно на линкот: https://www.kaggle.com/datasets/lakritidis/identifying-influential-bloggers-techcrunch.
Неструктурирана база на податоци која се користи во рамките на овој проект е Граф-базираната база на податоци Neo4j, а релациона ваза е базата PostgreSQL.

Податочно множество:

Множеството Identifying Influential Bloggers: Techcrunch се состои од 3 фајлови:

- authors.csv – податоци за 107 блогери на Techcrunch,
- posts.csv – податоци за 19464 блог постови на Techcrunch напишани од авторите,
- comments.csv – коментари од 746561 читатели на постовите
- inlinks.csv – податоци за 193808 страници кои ги линкуваат Techcrunch постовите.

Чистење на податоците

Во рамки на податоците се среќаваат многу нецелосни податоци, исти редови со различно ИД и други грешки. Од тие причини csv фајловите се трансформираат со помош на python скрипта:

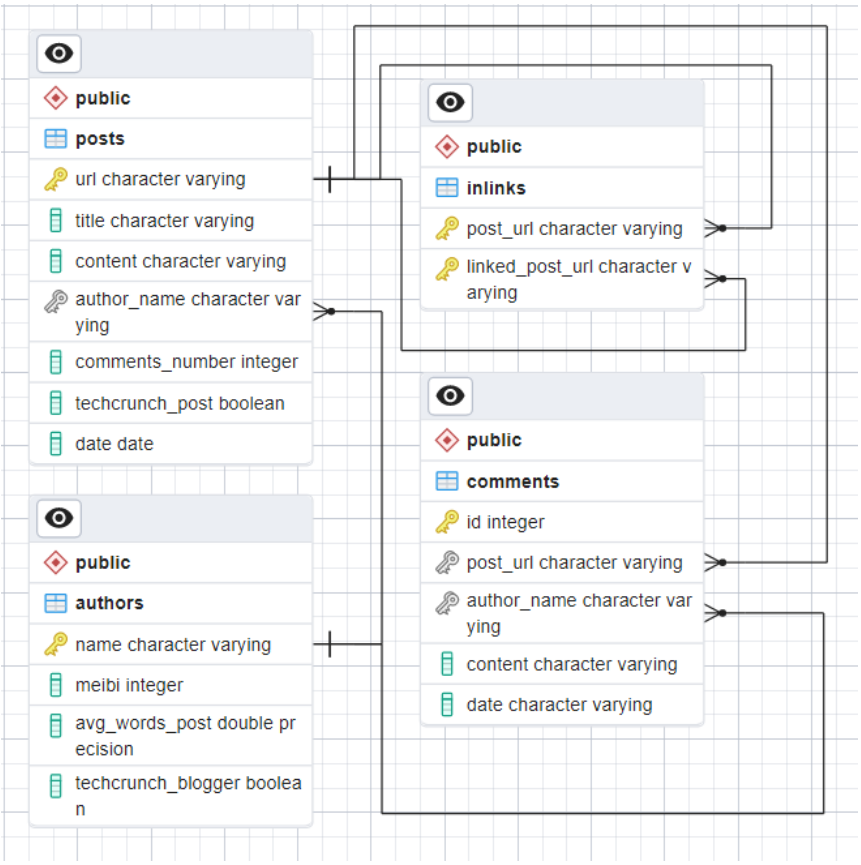
- authors_transformed.csv: ги содржи сите автори на Блогови, Постови и Коментари (автори од authors.csv, posts.csv и comments.csv).
- posts_transformed.csv: ги содржи податоците за Блогови и страници кои ги линкуваат (постови од authors.csv и inlinks.csv).
- comments_transformed.csv: ги содржи податоците за коментари од корисници (коментари од comments.csv).
- inlinks_transformed.csv: ги содржи линковите меѓу сите постови(линкови од inlinks.csv)

На крај се добиваат следните csv:

Внесување на податоците во PostgreSQL

- Креирање на табели и Шемата на базата на податоци:

```
create table Authors (
    name varchar primary key,
    meibi integer,
    avg_words_post float,
    techcrunch_blogger boolean not null
);
create table Posts (
    url varchar primary key,
    title varchar,
    content varchar,
    author_name varchar,
    comments_number integer,
    techcrunch_post boolean not null,
    date date,
    foreign key(author_name) references Authors(name)
);
create table Comments (
    id integer,
    post_url varchar,
    author_name varchar,
    content varchar,
    date varchar,
    foreign key(post_url) references Posts(url),
    foreign key(author_name) references Authors(name),
    primary key(id)
);
create table Inlinks (
    post_url varchar,
    linked_post_url varchar,
    primary key(post_url, linked_post_url),
    foreign key(post_url) references Posts(url),
    foreign key(linked_post_url) references Posts(url)
)
```



- **Внесување на податоците**
  - **Автори**

```
COPY authors(name, meibi, avg_words_post, techcrunch_blogger)
FROM 'C:/Users/marija/Documents/Neo4j/relate-data/dbmss/dbms-535955cc-1561-4ae6-ac9d-cb27f2726a0e/import/authors_transformed.csv'
DELIMITER ','
CSV HEADER;
```

```
COPY 196273

Query returned successfully in 2 secs 11 msec.
```

  - Постови

```
COPY posts(url, title, content, author_name, comments_number, date, techcrunch_post )
FROM 'C:/Users/marija/Documents/Neo4j/relate-data/dbmss/dbms-535955cc-1561-4ae6-ac9d-cb27f2726a0e/import/posts_transformed.csv'
DELIMITER ','
CSV HEADER;
```

```
COPY 170305

Query returned successfully in 8 secs 483 msec.
```

  - **Коментари**

```
COPY comments(id, post_url, content, author_name, date)
FROM 'C:/Users/marija/Documents/Neo4j/relate-data/dbmss/dbms-535955cc-1561-4ae6-ac9d-cb27f2726a0e/import/comments_transformed.csv'
DELIMITER ','
CSV HEADER;
```

```
COPY 741276

Query returned successfully in 48 secs 96 msec.
```

  - **Линкови**

```
COPY inlinks(post_url, linked_post_url)
FROM 'C:/Users/marija/Documents/Neo4j/relate-data/dbmss/dbms-535955cc-1561-4ae6-ac9d-cb27f2726a0e/import/inlinks_transformed.csv'
DELIMITER ','
CSV HEADER;
```

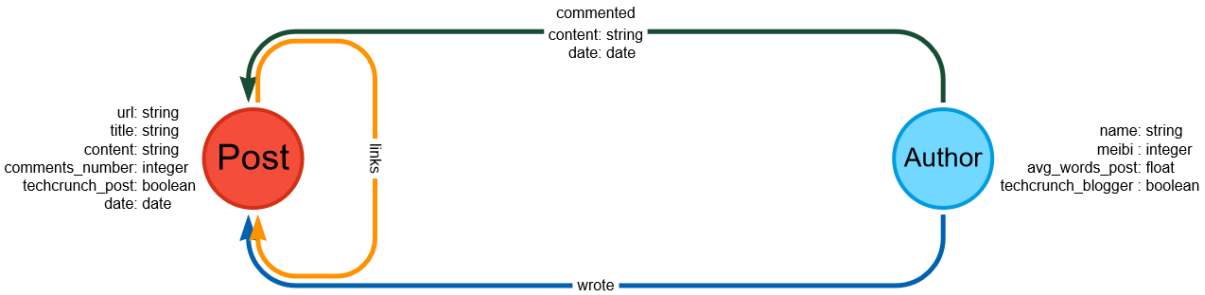
```
COPY 193504

Query returned successfully in 16 secs 460 msec.
```

Вкупно време на внесување во база: 75,05 секунди

Внесување на податоците во Neo4j

- Шема на базата:



- Подесување на ресурси достапни за базата

Edit settings

```
# Disable logging JMX endpoint.
server.jvm.additional=-Dlog4j2.disable.jmx=true

# Limit JVM metaspace and code cache to allow garbage collection. Used by cypher for code generation and
may grow indefinitely unless constrained.
# Useful for memory constrained environments
#server.jvm.additional=-XX:MaxMetaspaceSize=1024m
#server.jvm.additional=-XX:ReservedCodeCacheSize=512m

#*****************************************************************
# Wrapper Windows NT/2000/XP Service Properties
#*****************************************************************
# WARNING - Do not modify any of these properties when an application
#  using this configuration file has been installed as a service.
#  Please uninstall the service before modifying this section.  The
#  service can then be reinstalled.

# Name of the service
server.windows_service_name=neo4j

#*****************************************************************
# Other Neo4j system properties
#*****************************************************************

dbms.memory.heap.initial_size=3G
dbms.memory.heap.max_size=3G
dbms.memory.pagecache.size=3G
dbms.windows_service_name=neo4j-relate-dbms-535955cc-1561-4ae6-ac9d-cb27f2726a0e
dbms.jvm.additional=-Dlog4j2.formatMsgNoLookups=true
```

Reset to defaults Undo Apply Close

- **Ограничувања**

```
neo4j$ SHOW CONSTRAINTS
```

	id	name	type	entityType	labelsOrTypes	properties	ownedIndex
1	4	"author_name"	"UNIQUENESS"	"NODE"	["Author"]	["name"]	"author_name"
2	6	"post_url"	"UNIQUENESS"	"NODE"	["Post"]	["url"]	"post_url"

Started streaming 2 records after 40 ms and completed after 68 ms.

- **Автори**

```
:auto LOAD CSV WITH HEADERS FROM "file:///authors_transformed.csv" AS row
CALL {
    WITH row
    MERGE(a:Author{name:row.name})
    SET a.meibi=toInteger(row.meibi)
    SET a.avg_words_post=toFloat(row.avg_words_post)
    SET a.techcrunch_blogger=toBoolean(row.techcrunch_blogger)
} IN TRANSACTIONS
```

Added 196273 labels, created 196273 nodes, set 392760 properties, completed after 16115 ms.

Added 196273 labels, created 196273 nodes, set 392760 properties, completed after 16115 ms.

- **Постови**

```
:auto LOAD CSV WITH HEADERS FROM "file:///posts_transformed.csv" AS row
CALL {
    WITH row
    MERGE(p:Post{url:row.url})
    SET p.title=row.title
    SET p.content=row.content
    SET p.comments_number=toInteger(row.comments_number)
    SET p.techcrunch_post=toBoolean(row.techcrunch_post)
    SET p.date=date(row.date)
} IN TRANSACTIONS
```

Added 170305 labels, created 170305 nodes, set 700072 properties, completed after 15970 ms.

Added 170305 labels, created 170305 nodes, set 700072 properties, completed after 15970 ms.

- Релацијата :**wrote** која означува кој е авторот на кој пост

```
:auto LOAD CSV WITH HEADERS FROM "file:///posts_transformed.csv" AS row
CALL {
    WITH row
    MATCH(p:Post{url:row.url})
    MATCH(a:Author{name:row.author_name})
    MERGE(a)-[:wrote]→(p)
} IN TRANSACTIONS
```

Table

Code

Created 153901 relationships, completed after 10502 ms.

Created 153901 relationships, completed after 10502 ms.

- Релацијата **:commented** која претставува коментар од еден автор на некој пост

```
:auto LOAD CSV WITH HEADERS FROM "file:///comments_transformed.csv" AS row
CALL {
    WITH row
    MATCH(p:Post{url:row.post_url})
    MATCH(a:Author{name:row.author_name})
    MERGE(a)-[c:commented{id:row.id, date:row.date}]→(p) SET c.content = row.comment
} IN TRANSACTIONS
```

Table

Code

Set 1482552 properties, created 741276 relationships, completed after 63114 ms.

Set 1482552 properties, created 741276 relationships, completed after 63114 ms.

- Релацијата **:links** која означува линк од еден пост према друг

```
:auto LOAD CSV WITH HEADERS FROM "file:///inlinks_transformed.csv" AS row
CALL {
    WITH row
    MATCH(p:Post{url:row.post})
    MATCH(l:Post{url:row.linked_post})
    MERGE(p)-[:links]→(l)
} IN TRANSACTIONS
```

Table

Code

Created 193504 relationships, completed after 19657 ms.

Created 193504 relationships, completed after 19657 ms.

Вкупно време на внесување во база: 109.234 минути

- **Додавање на дополнителни лабели:**

```
MATCH (a:Author{techcrunch_blogger:true})
SET a:Blogger
RETURN a
```

Graph | Table | Text | Code

a


```
{
  "identity": 46964,
  "labels": [
    "Author",
    "Blogger"
  ],
  "properties": {
    "avg_words_post": 4.25791,
    "meibi": 43,
    "name": "Jason Kincaid",
    "techcrunch_blogger": true
  },
  "elementId": "46964"
}
```


```
{
```

Added 107 labels, started streaming 107 records after 3 ms and completed after 391 ms.

```
MATCH (p:Post{techcrunch_post:true})
SET p:Blog
RETURN p
```

Graph | Table | Text | Code

p


```
{
  "identity": 243237,
  "labels": [
    "Post",
    "Blog"
  ],
  "properties": {
    "date": "2010-04-01",
    "title": "We Just Tested Twitter&#8217;s @anywhere Platform (Screenshots)",
    "techcrunch_post": true,
    "url": "http://techcrunch.com/2010/04/01/we-just-tested-twitters-anywhere-platform-screen
    "content": "During his keynote at SXSW last month, Twitter CEO Evan Wiliams announced an
looks like we&#8217;ve just come across the first site to feature @anywhere.  Meet Eggboiling
&#8216;Connect With Twitter&#8217;; buttons to follow twitter users @jack, @biz, and @ev; a t
each of the &#8216;follow&#8217; buttons appropriately changed the status from &#8220;Follow
```


Added 19458 labels, started streaming 19458 records after 3 ms and completed after 70 ms, displaying first 1000 rows.

Прашалници:

1. Сите techrunch блогови кои се напишани помеѓу 2009 и 2010 година и во насловот содржат ‘twitter’

Neo4j:

```
MATCH(b:Blog)
WHERE b.date ⩾ date('2009-01-01') AND b.date ⩽ date('2010-12-31')
AND (b.title CONTAINS 'twitter' OR  b.title CONTAINS 'Twitter')
RETURN b.title
```

Table | Text | Code

	b.title
1	"We Just Tested Twitter’s @anywhere Platform (Screenshots)"
2	"Twitter Tweaks Search, A Few Popular Tweets Now At The Top Of Results"
3	"SharesPost Report Values Twitter At $750 Million"
4	"Twitter Sets Chirp Free… Well, Cheaper"
5	"Twitter Launches A New, Dynamic Homepage"
6	"Twitter And The Nine-Month Bounce"
7	

Started streaming 767 records in less than 1 ms and completed after 53 ms.

PostgreSQL:

```
SELECT p.title
FROM posts as p
WHERE p.techcrunch_post = 'true'
AND p.date>=date('2009-01-01') AND p.date<=date('2010-12-31')
AND ( p.title like '%twitter%' OR p.title like '%Twitter%')
```

Data Output | Messages | Notifications

	title character varying
1	We Just Tested Twitter’s @anywhere Platform (Screenshots)
2	Twitter Tweaks Search, A Few Popular Tweets Now At The Top Of Results
3	SharesPost Report Values Twitter At $750 Million
4	Twitter Sets Chirp Free… Well, Cheaper
5	Twitter Launches A New, Dynamic Homepage
6	Twitter And The Nine-Month Bounce

Total rows: 767 of 767 | Query complete 00:00:00.107

Разлика: 54 ms.

2. Постовите напишани од автор кои содржат линк кон пост од истиот автор

Neo4j:

```
MATCH(p1:Post)-[:links]→(p)
MATCH (a:Author)-[:wrote]→(p)
MATCH (a)-[:wrote]→(p1)
RETURN distinct(p.url)
```

Table | Text | Code

	(p.url)
1	"http://www.mobilecrunch.com/2010/02/17/opera-says-opera-mini-for-iphone-100-compliant-with-app-store-policies-video/"
2	"http://techcrunch.com/2009/12/16/google-android-market/"
3	"http://techcrunch.com/2010/03/05/global-smartphone-app-download-market-could-reach-15-billion-by-2013-report/"
4	"http://techcrunch.com/2010/02/28/19-startups-showing-their-wares-at-techcrunch-japans-tokyocamp-demo-event/"
5	"http://techcrunch.com/2010/02/15/layar-scores-3-4m-in-funding-global-distribution-agreement/"
6	"http://techcrunch.com/2009/10/14/layar-brings-augmented-reality-browser-to-the-iphone-screenshots/"
7	

Started streaming 42 records after 1 ms and completed after 446 ms.

PostgreSQL:

```
SELECT p.url
FROM posts as p
JOIN inlinks as il ON il.post_url=p.url
JOIN posts as p1 on il.linked_post_url = p1.url
JOIN authors as a ON p.author_name = a.name
WHERE p1.author_name=a.name
```

Data Output | Messages | Notifications

	url [PK] character varying
1	http://www.mobilecrunch.com/2008/12/09/android-rising-sony-ericsson-vodafone-and-14-others-join-open-handset-alliance/
2	http://www.mobilecrunch.com/2009/01/28/review-t-mobile-blackberry-8900/
3	http://www.mobilecrunch.com/2009/03/20/sekai-camera-mobile-social-tagging-is-coming-to-android-phones-too/
4	http://www.mobilecrunch.com/2010/03/03/skype-for-symbian-lands-on-ovi-store-more-than-200-million-possible-users/
5	http://www.mobilecrunch.com/2009/05/05/att-finally-releases-what-should-have-been-the-first-iphone-app/
6	http://www.crunchgear.com/2009/04/22/proof-that-att-needs-to-extend-that-ipl

Total rows: 48 of 48 | Query complete 00:00:00.266

✓ Successfully run. Total query runtime: 266

Разлика: - 180 ms

3. Блоговите напишани од некој автор кои содржат линк кон пост од истиот автор, кој има барем 30 коментари.

Neo4j:

```
MATCH(p1:Blog)-[:links]→(p)
WHERE p:Blog
MATCH (a:Blogger)-[:wrote]→(p)
MATCH (a)-[:wrote]→(p1)
MATCH (p)←[c:commented]-()
WITH p, COUNT(c) AS num
WHERE num ⩾30
RETURN p.url, num
```

	p.url	num
1	"http://techcrunch.com/2007/11/05/breaking-google-announces-android-and-open-handset-alliance/"	112
2	"http://techcrunch.com/2008/03/17/microsoft-adopts-flash-lite-for-windows-mobile-as-a-stopgap-measure/"	52
3	"http://techcrunch.com/2008/04/30/adobes-open-screen-project-write-once-flash-everywhere/"	66
4	"http://www.mobilecrunch.com/2008/11/16/adobe-to-demo-flash-on-mobile-but-only-windows-still-working-on-the-iphone/"	30
5	"http://www.mobilecrunch.com/2009/07/20/review-htc-hero/"	73
6	"http://www.crunchgear.com/2009/09/17/review-htc-hero-from-sprint/"	92

Started streaming 27 records after 1 ms and completed after 42 ms.

PostgreSQL:

```
SELECT distinct(p1.url)
FROM posts as p
JOIN inlinks as il ON il.post_url=p.url AND p.techcrunch_post IS TRUE
JOIN posts as p1 on il.linked_post_url = p1.url AND p1.techcrunch_post = 'true'
JOIN authors as a ON p.author_name = a.name AND p1.author_name=a.name
JOIN comments as c on c.post_url = p1.url
GROUP BY p1.url
HAVING count(c.id) >= 30
```

Data Output | Messages | Notifications

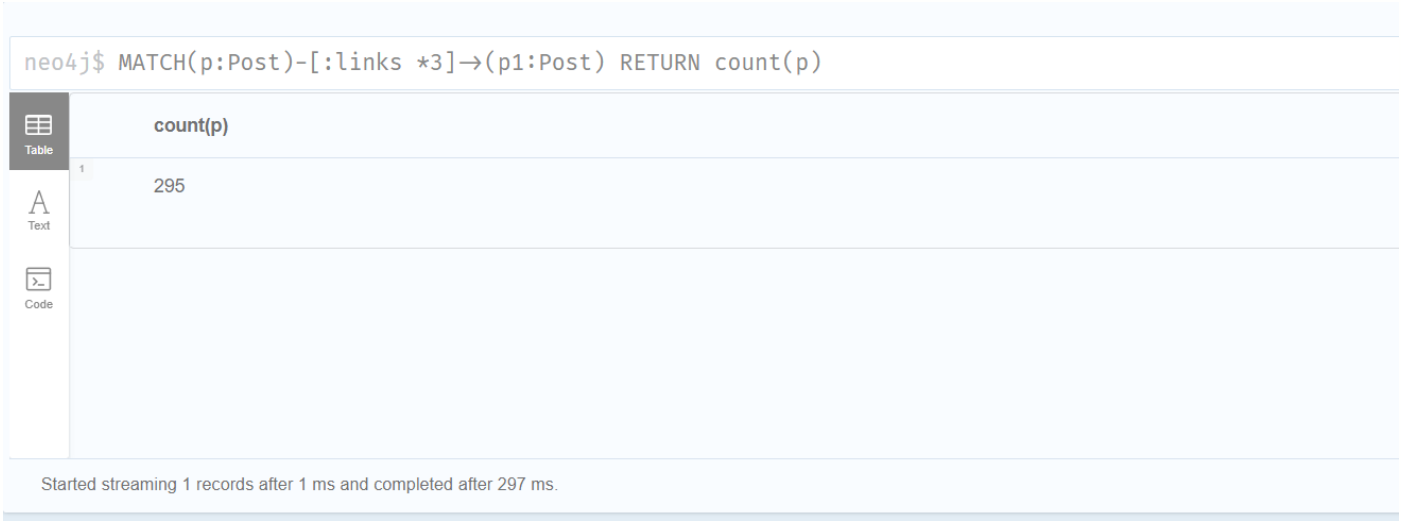
	url [PK] character varying
1	http://techcrunch.com/2007/11/05/breaking-google-announces-android-and-open-handset-alliance/
2	http://techcrunch.com/2008/03/17/microsoft-adopts-flash-lite-for-windows-mobile-as-a-stopgap-measure/
3	http://techcrunch.com/2008/04/30/adobes-open-screen-project-write-once-flash-everywhere/
4	http://techcrunch.com/2009/02/17/tagging-the-real-world-sekai-camera-for-the-iphone-is-alive-and-very-cool/
5	http://techcrunch.com/2009/04/02/skype-iphone-app-downloaded-one-million-times-in-first-two-days/
6	http://techcrunch.com/2009/04/14/shocker-att-wants-to-keep-keep-sitting-on-its-golden-egg-the-iphone/

Total rows: 27 of 27 | Query complete 00:00:00.274

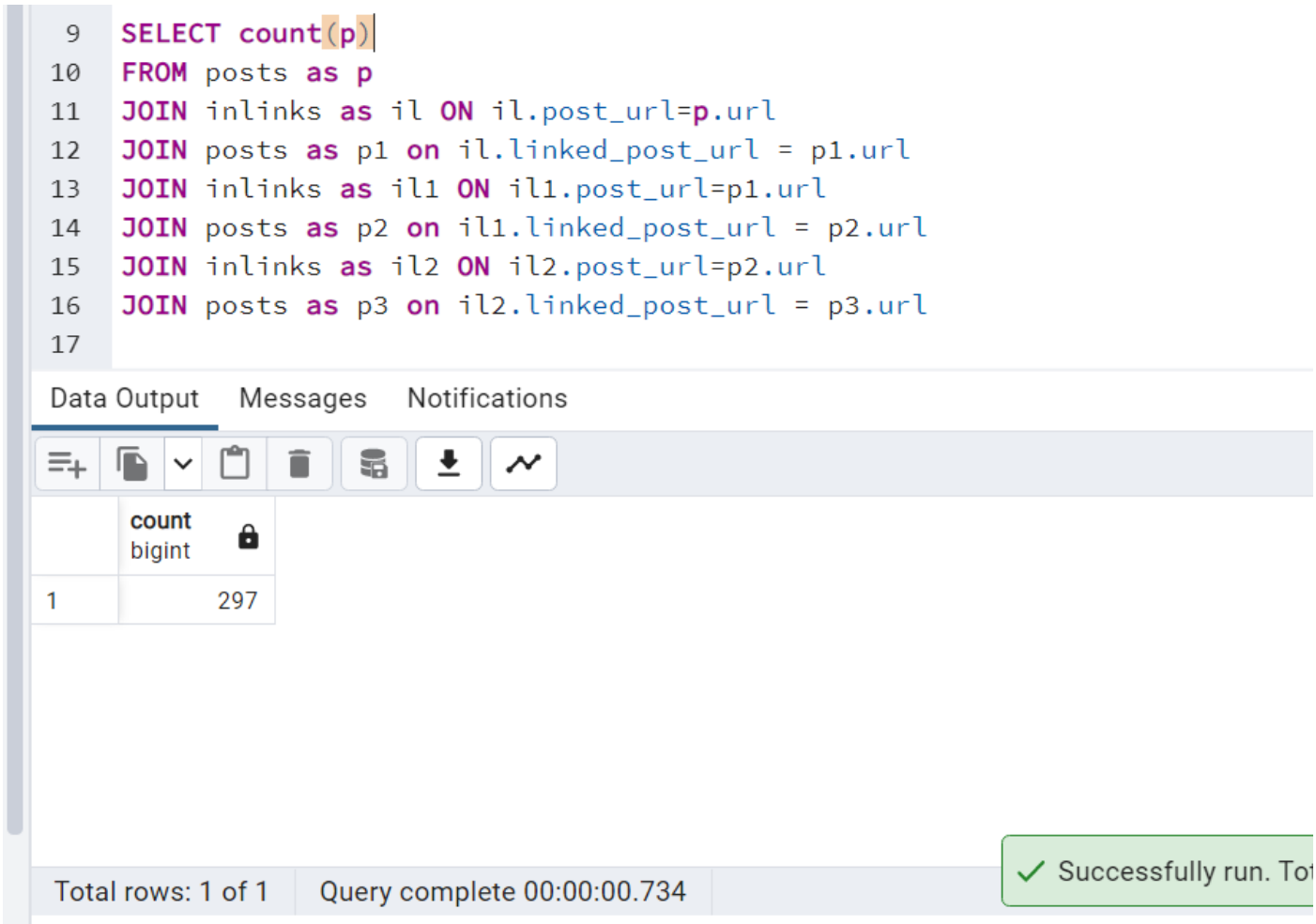
Разлика: 232 ms

4. Број на низи од 4 поста поврзани со линкови

Neo4j:



PostgreSQL:

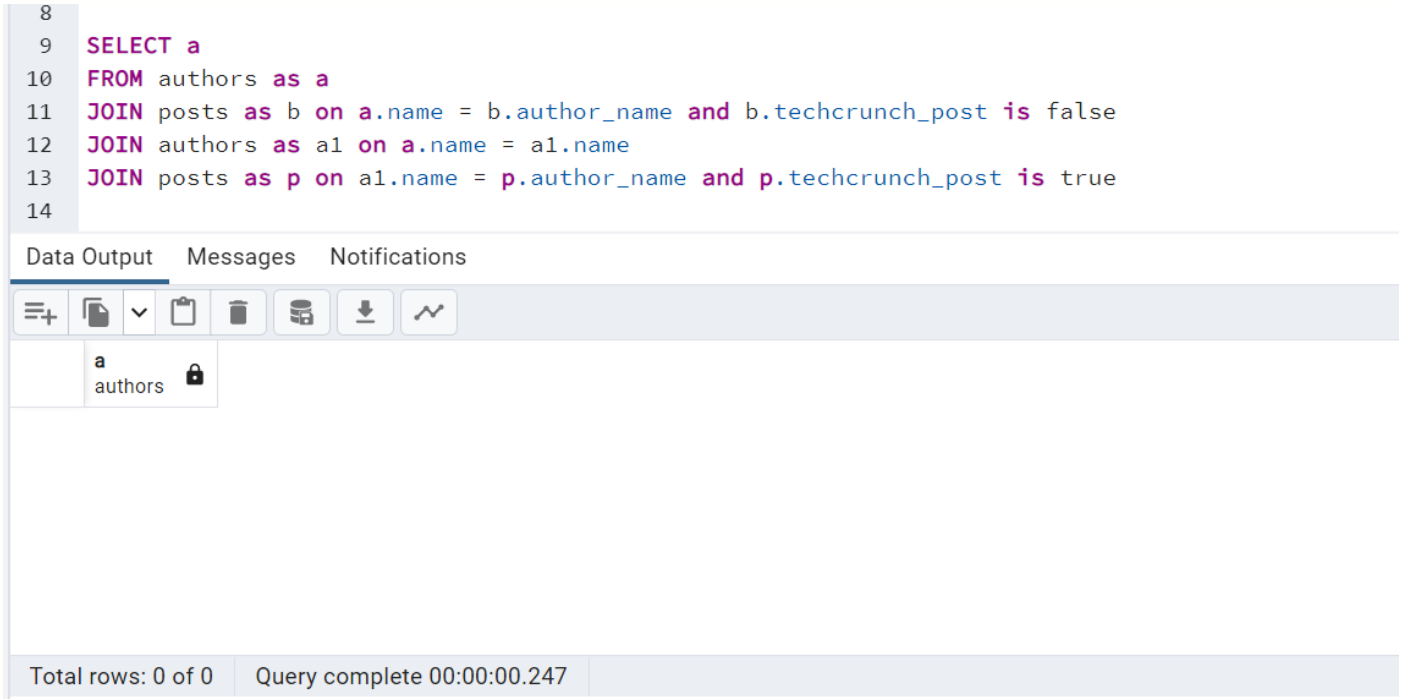


Разлика : 437 мс

5. Број на автори кои имаат напишано и techcrunch и обичен пост

Neo4j:

PostgreSQL:



Разлика: -6 699 мс, 240 мс.

6. Автори кои си коментирале на свој пост на истиот ден кога го напишале

Neo4j:

```
MATCH(a:Author)-[:wrote]→(p:Post)←[c:commented]-(a)
WHERE date(p.date)=date(c.date)
RETURN distinct a.name
```

Table | Text | Code

	a.name
1	"Michael Arrington"
2	"Devin Coldewey"
3	"MG Siegler"
4	"Andy Brett"
5	"Erick Schonfeld"
6	"Robin Wauters"
7	

Started streaming 57 records after 1 ms and completed after 438 ms.

PostgreSQL:

```
SELECT distinct(p.author_name)
FROM posts as p
JOIN comments as c on c.author_name = p.author_name
WHERE c.post_url = p.url and date(p.date) = date(c.date)
```

Data Output | Messages | Notifications

	author_name character varying
1	Andy Brett
2	Arden Pennell
3	Asad Akbar
4	Blake Robinson
5	Brian Solis
6	Chris Velazco
7	Dan Kimerling
8	Dan Romero
9	David Diaz
10	Devin Coldewey

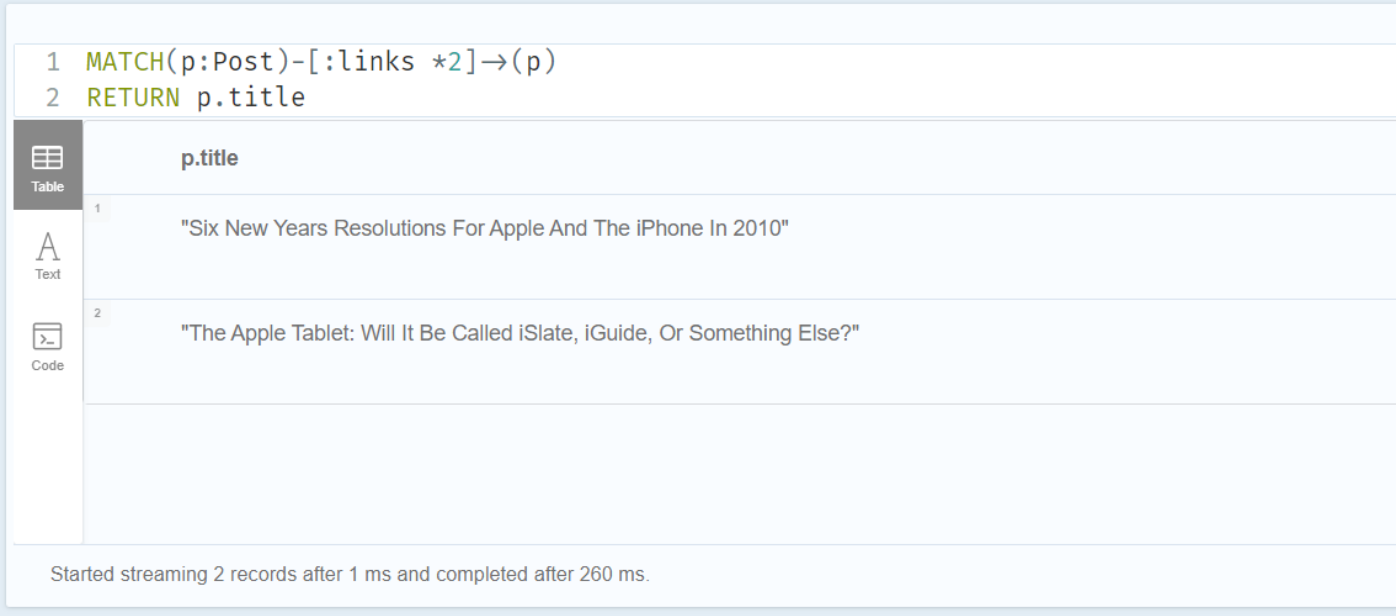
Total rows: 57 of 57 | Query complete 00:00:01.043

PostgreSQL 15/Bloggers - Database connected.

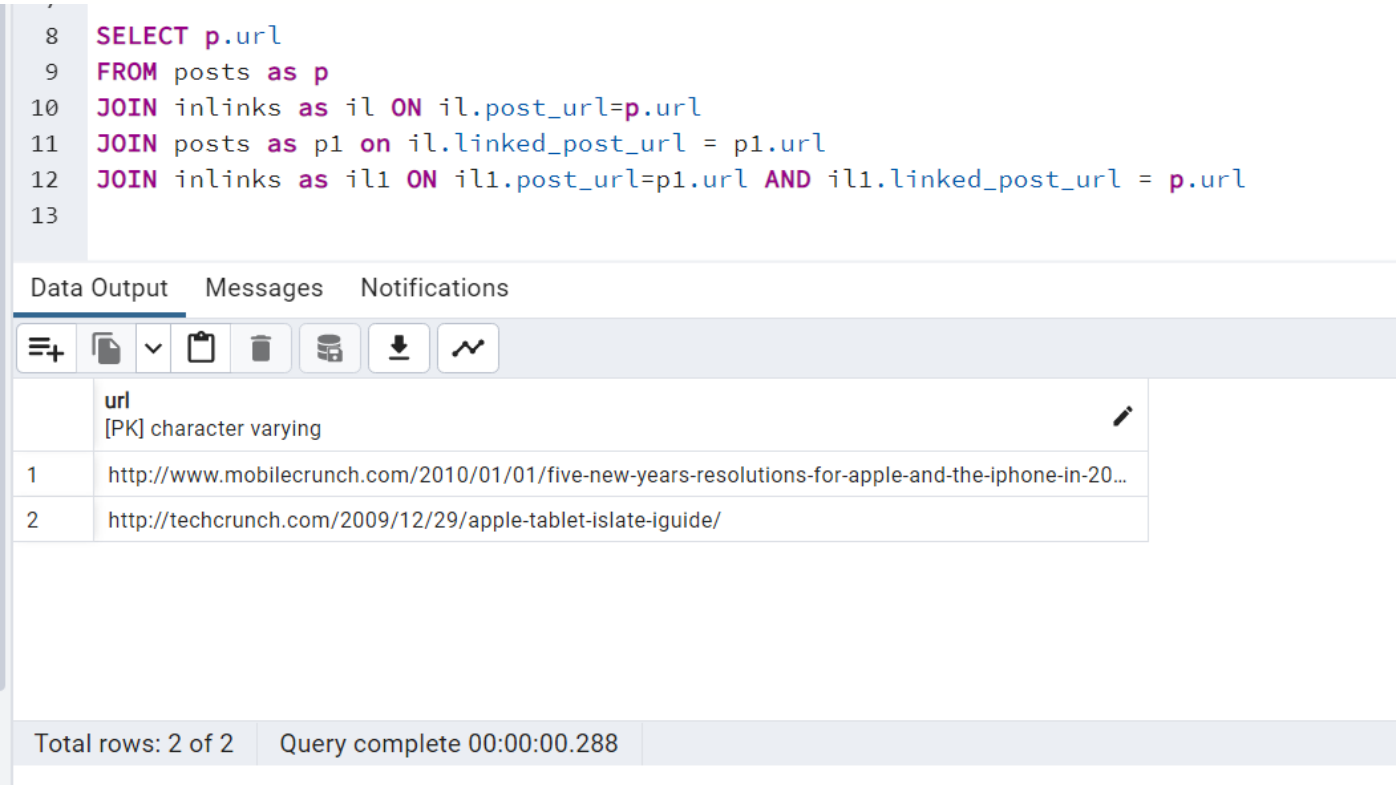
Разлика: 605 мс

7. Постови кои се линкуваат меѓусебно

Neo4j:



PostgreSQL:



Разлика 28 мс

8. Постови кои во содржината го содржат името на авторот

Neo4j:

PostgreSQL:

Разлика: -106 мс

9. Автори кои имат повеќе од 1000 коментари на сите постови вкупно, во опаѓачки редослед

Neo4j:

```
MATCH(a:Author)-[c:commented]→()
WITH a, count(c) as com
WHERE com > 1000
RETURN a.name, com
ORDER BY com DESC
```

	a.name	com
1	"Michael Arrington"	6652
2	"Chris"	5267
3	"Mike"	3274
4	"John"	3042
5	"Alex"	2730
6	"MG Siegler"	2673

Started streaming 42 records in less than 1 ms and completed after 233 ms.

PostgreSQL:

Query | Query History

```
SELECT auth
FROM (
    SELECT a.name as name , count(c) as comment_count
    FROM authors as a
    JOIN comments as c on a.name = c.author_name
    GROUP BY a.name
) as auth
WHERE auth.comment_count > 1000
ORDER BY auth.comment_count DESC
```

Data Output | Messages | Notifications

	auth record
1	("Michael Arrington",6652)
2	(Chris,5267)
3	(Mike,3274)
4	(John,3042)
5	(Alex,2730)
6	("MG Siegler",2673)

Total rows: 42 of 42 | Query complete 00:00:02.826 | Successfully run. Total query runtime: 2

Разлика: 2 593 мс

10. Блогери и вкупни коментари на сите нивни постови во опаѓачки редослед

Neo4j:

```
MATCH (a:Blogger)-[:wrote]→(p:Post)←[:commented]-(c)
WITH a, count(c) as comment_count
ORDER BY comment_count DESC
RETURN a.name, comment_count
```

	a.name	comment_count
1	"Michael Arrington"	245674
2	"Erick Schonfeld"	95017
3	"MG Siegler"	71739
4	"Jason Kincaid"	61966
5	"Robin Wauters"	45271
6	"Duncan Riley"	32708

Started streaming 102 records in less than 1 ms and completed after 90 ms.

PostgreSQL:

```
SELECT name, comment_count
FROM (
    SELECT a.name as name, count(c) as comment_count
    FROM authors as a
    JOIN posts as p on p.author_name = a.name and a.techcrunch_blogger is true
    JOIN comments as c on c.post_url= p.url
    GROUP BY a.name
) as bloggers
ORDER BY comment_count DESC
```

Data Output | Messages | Notifications

	name [PK] character varying	comment_count bigint
1	Michael Arrington	245674
2	Erick Schonfeld	95017
3	MG Siegler	71739
4	Jason Kincaid	61966
5	Robin Wauters	45271
6	Duncan Riley	32708

Total rows: 102 of 102 | Query complete 00:00:01.602 | Successfully run. Total query runtime: 1 secs 602 ms

Разлика: 1 512 мс

Заклучок:

За поголемиот дел од прашалниците важи дека Neo4j побрзо дава резултат, иако има и исклучоци.

Neo4j е впечатливо побрза во случајот 9 и 10, или случаите каде што во Neo4j се избегнува употреба на вгнездено квери.

Од 2. и 3. Може да се забележи дека додавањето на дополнителни лабели на ентитетите во Neo4j може значително да ги забрза кверињата, додека во релациона база нема таква можност.

Во 4 Neo4j повторно има подобри резултати и многу поедноставен начин на пишување на квери, отколку во PostgreSQL каде треба да се направат повеќе спојувања на табели.

Во 1 и 8 се работи за квериња кои се концентрираат на една табела, и нема голема разлика во перформансите во двете бази.

Во повеќето квериња каде има поврзување помеѓу различни табели Neo4j има подобри резутати, иако нема премногу голема разлика.

Од 2 и 5 може да се забележи дека кверињата во кои се работи за различни ентитети со иста лабела кои се во ист тип на релација со некој ентитет со различна лабела имаат многу полоши перформанси. Но, промена на формата на кварито може да ја промени брзината на извршување многу пати.

Кога пак се работи за за прашалници каде се кверираат ентитети од ист тип поврзани со истата релација, Neo4j има подобри перформанси(7 и 4).

Како и да е, прашалниот јазик на Neo4, Cypher има многу полесна синтакса за користење.

Од овој експеримент може да се заклучи дека Neo4j има подобри перформанси кога пребарувањето во базата се врши на база на релации, но потребно е да се внимава каков тип на прашалници се најчести.