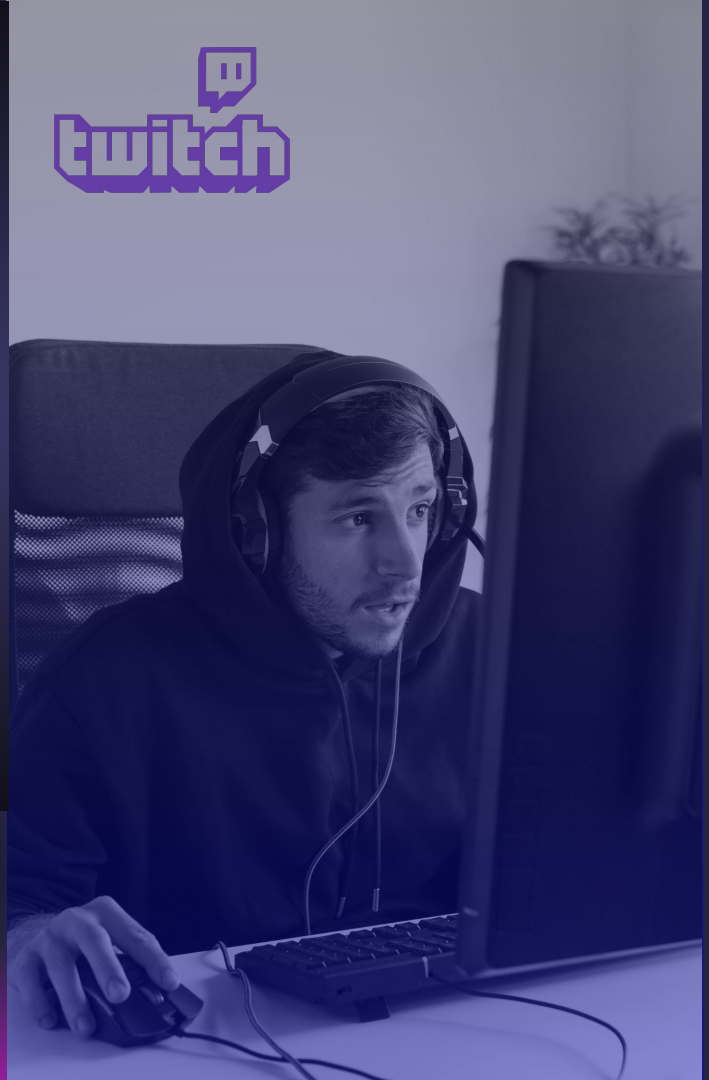
SparkVision agency



SparkVision
Marketing agency

Twitch Marketing Analysis

Analysis on which streamers to target in an advertising campaign

Gianluca Zappia
Roberto Antonio Mapa
Riccardo Pandolfi
Ary Jose Rubi Espinal

Sparkvision Agency

Introduction

SparkVision
Marketing agency

Scenario: The rise of eSports and online gaming has led to the emergence of a new industry of companies focused specifically on this space. Consequently, such companies would need the expertise of a Marketing Agency to guide advertising in online gaming platforms–Twitch being the largest and most opportunistic of all.

As a Marketing Agency, we are working with brands to curate sponsored content and advertising campaigns in order to reach the abundance of Twitch users and increase awareness for each type of brand.

Organization: Marketing Agency

Target Market: Twitch audience

Business Problem: How to effectively target and reach the Twitch audience through advertising

Objective: Offer a targeted advertising service to companies that are interested in reaching the Twitch streamer audience

DB Technologies - MongoDB

MongoDB is a non-relational database that is based on documents and holds data in the form of collections. Documents in the database are comprised of key-value pairs–a basic unit of data in MongoDB. Collections, on the other hand, contain multiple documents. A database is made up of one or more collections.

The advantages MongoDB brings is that the structure of the documents adhere to how developers usually construct classes. There is no need for a schema definition and fields can be quickly added and removed. It allows for arrays storage and has a really high scalability. Furthermore, it also allows for easy integration with other libraries.

MongoDB supports a wide range of programming languages and platforms, and allows the usage of a great number of features such as automatic sharding, which distributes data across multiple servers and replica sets, which provide high availability and automatic failover.

We have decided to use MongoDB since it is schema less, joins are not complex and the ability to query is very straightforward.

DB Technologies - Neo4j

SparkVision
Marketing agency

Sparkvision Agency

Neo4j is a graph database management system administered in Java, available in open source and commercial editions. Neo4j uses its own query language, Cypher, and handles complex queries efficiently at scale. The advantage of using Neo4j is that it brings: highly performant read and write scalability; reliability for mission-critical production applications; ease of learning, using and managing; and easy data modeling.

Neo4j is Atomicity, Consistency, Isolation, Durability (ACID) compliant, ensuring transactions are processed reliably and with high data integrity. ACID compliance is important for a Twitch dataset as it helps guarantee the data accurately displays user behaviors and preferences—allowing for analysis integral to forming advertising campaigns.

Moreover, Neo4j is ideal for Twitch data since it is ideal for storing and querying highly interconnected data using nodes and relationships–focal points to social media data. Neo4j's graph-based architecture allows for systematic traversal of relationships between data points, making it a powerful tool for analyzing and understanding user behavior on the platform.

Sparkvision Agency

Combination of MongoDB and Neo4j

SparkVision
Marketing agency

There are different reasons why we have chosen to adopt both MongoDB and Neo4j technologies; one of the main reasons regards its flexibility. The way data is stored and queried is straightforward and intuitive.

Another major reason why we have chosen to adopt those two together is the level of scalability: they are both designed to scale horizontally across multiple servers, which means that they can handle large amounts of data and traffic.

The two technologies provide complementary benefits in handling a Twitch dataset. While MongoDB takes care of large volumes of unstructured data, Neo4j handles complex queries on interconnected data and generates graphs. Using both provides a more a comprehensive and nuanced analysis of the data, allowing for exhaustive identification of influential users, popular games, and other insights that might be missed by using the databases separately.

The combination of both allowed us to store streamers profiles and their activity such as the number of minutes played, number of minutes streamed or the number of followers–on MongoDB–as well as the relations between the different users and the video games they play, what language they use, and what content they stream.

Sparkvision Agency

Data - Overview

SparkVision
Marketing agency

The data used contained information about Twitch users and details of their respective streams. At first we pulled two datasets: one in a JSON format (stored in MongoDB, containing data about the channels, their descriptions, and other metadata), and one in a Cypher format (stored in a Neo4j graph database containing details about the users, their channels, their interactions, and the like). We then merged the two datasets based on a common field: the channel name. To join the data from MongoDB and Neo4j, we used a technique called "polyglot persistence," wherein we stored data in multiple databases and then queried and joined the data as needed. Join the data: To join the data from MongoDB and Neo4j, we sought to use a Python script, whose programming language supports both databases and allowed us to retrieve data from MongoDB and Neo4j and perform joins based on the common identifier.

Sparkvision Agency

Data - Dataset 1 (MongoDB)

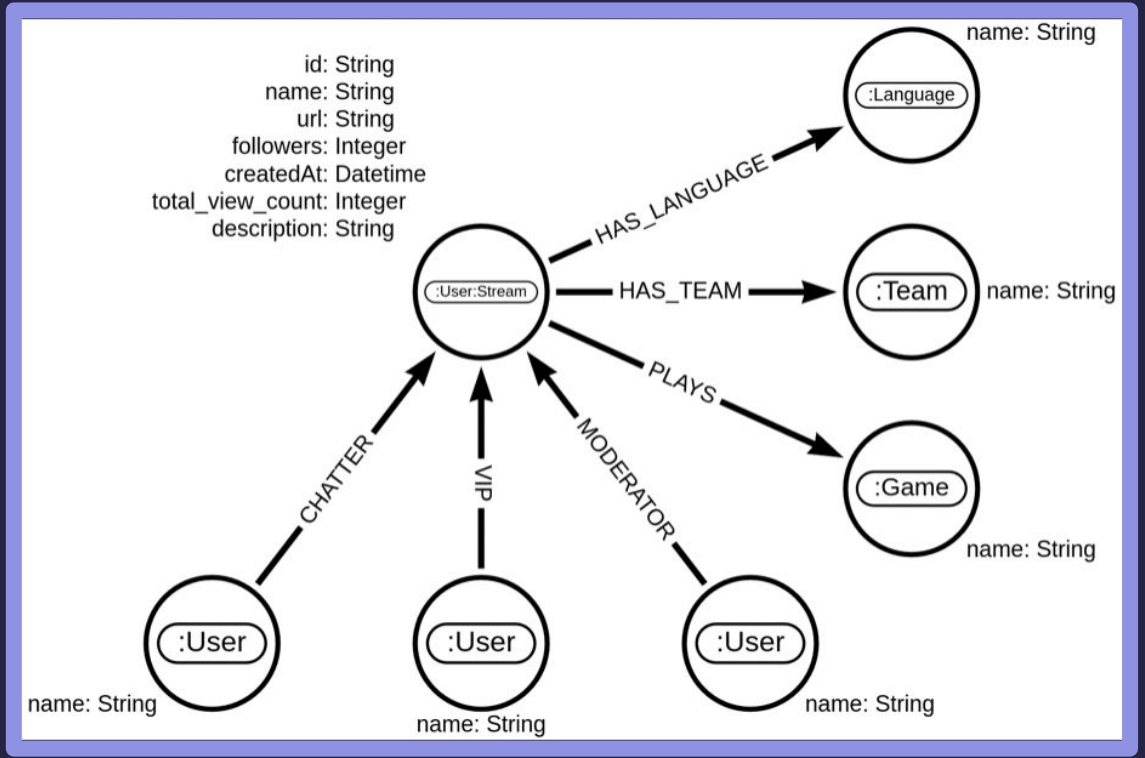
In order to gather as much information as possible from the streamers, we have been able to join two different datasets from MongoDB and Neo4j through the channel name.

	Description	Type
Channels	Name of the channel	String
Watch time	how much time the streamer has been watched for	Int64
Stream time	How much time the streamer has spent streaming	Int32
Peak viewers	highest peak of viewer per streamer	Int32
Average viewers	how many viewers watch a stream for a single channel on average	Int32
Followers	Number of follower per streamer	Int32
Followers gained	Rate of followers gained per stream	Int32
Views gained	Rate of views gained per stream	Int32
Partnered	whether the streamer partnered or not	Boolean
Mature	Whether the streamer displays mature content or not	Boolean

SparkVision
Marketing agency

Data - Dataset 2 (Neo4j)

The Twitch social network is composed of users, a percent of which stream their activities. The dataset, sourced from Neo4j's Sandbox, contains nodes of the streams on Twitch. As depicted in the graph model, the data contains details on users, users who stream (and information about which teams they belong to, which games they play on stream, which language they present their content, their number of followers, view count, and their user account creation date).



Sparkvision Agency

SparkVision
Marketing agency

Sparkvision Agency

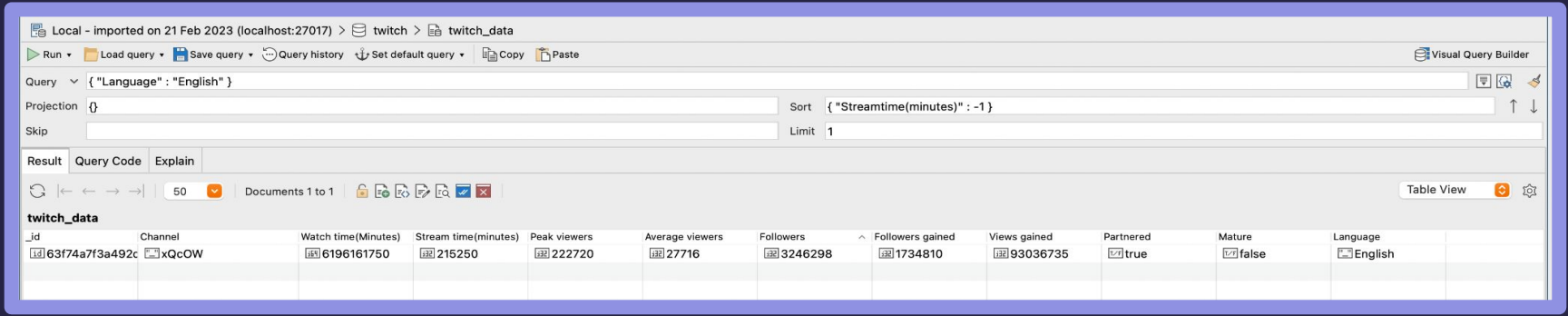
MongoDB - Query 1

SparkVision
Marketing agency

Query output: Streamer with the highest streaming time (in English)

Objective: Identify the channel with the most streaming minutes

Query components: This query filters data from a collection of Twitch users and streamers based on the language spoken during streams (English). The query then sorts the filtered data by the total amount of stream time in minutes, from highest to lowest. The query consists of a search and sort operation.



MongoDB - Query 2

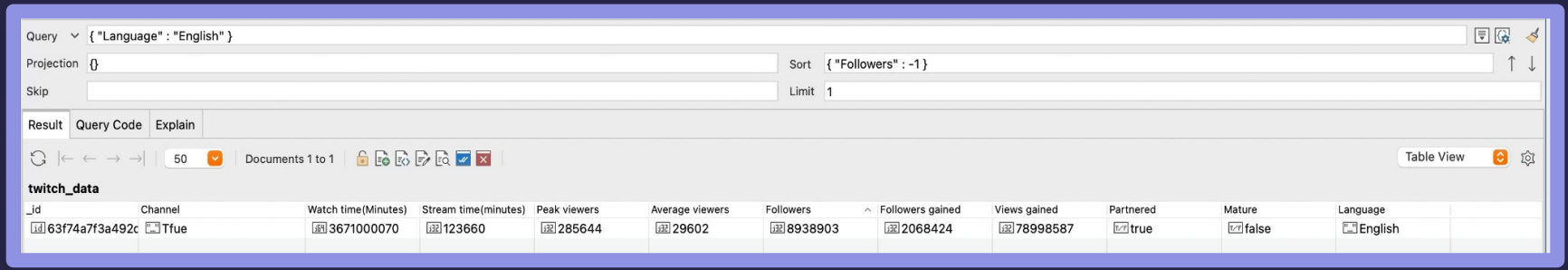
SparkVision
Marketing agency

Sparkvision Agency

Query output: Streamer with the highest number of followers (in English)

Objective: Identify the channel with the most followers

Query components: This query filters data from a collection of Twitch users and streamers based on the language spoken during streams (English) and then sorts the results based on the number of followers in descending order. The query consists of a search and sort operation to display Twitch users and streamers with the highest number of followers.



_id	Channel	Watch time(Minutes)	Stream time(minutes)	Peak viewers	Average viewers	Followers	Followers gained	Views gained	Partnered	Mature	Language
63f74a7f3a492c	Tfue	3671000070	123660	285644	29602	8938903	2068424	78998587	true	false	English

MongoDB - Query 3

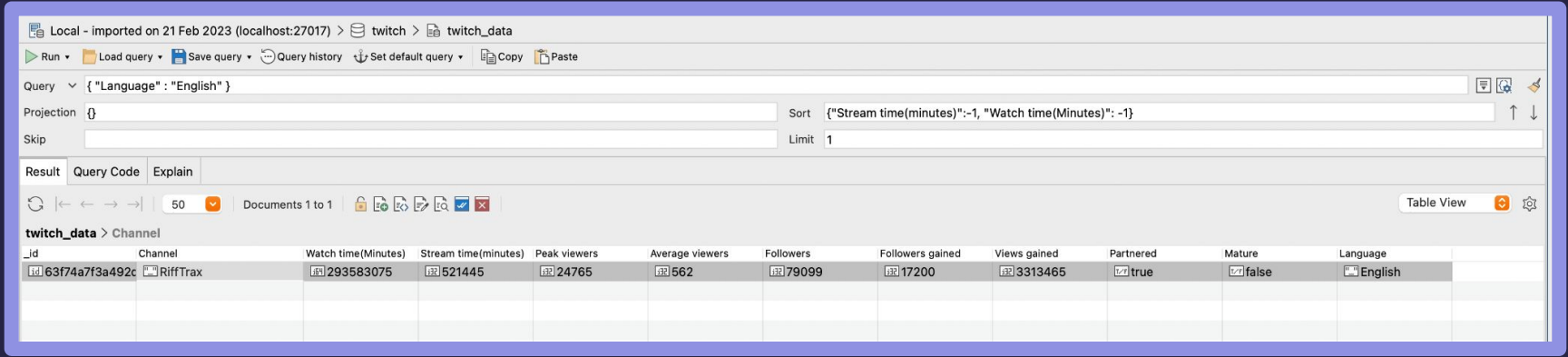
SparkVision
Marketing agency

Sparkvision Agency

Query output: Streamer with most watch time and stream time (in English)

Objective: Identify the channel with the most watch time and stream time

Query components: This query filters data from a collection of Twitch users and streamers based on the language spoken during streams (English) and sorts the results based on the total amount of stream time and watch time in minutes, from highest to lowest. The query consists of a search and sort operation to display Twitch users and streamers with the most stream and watch time.



Neo4j - Query 1

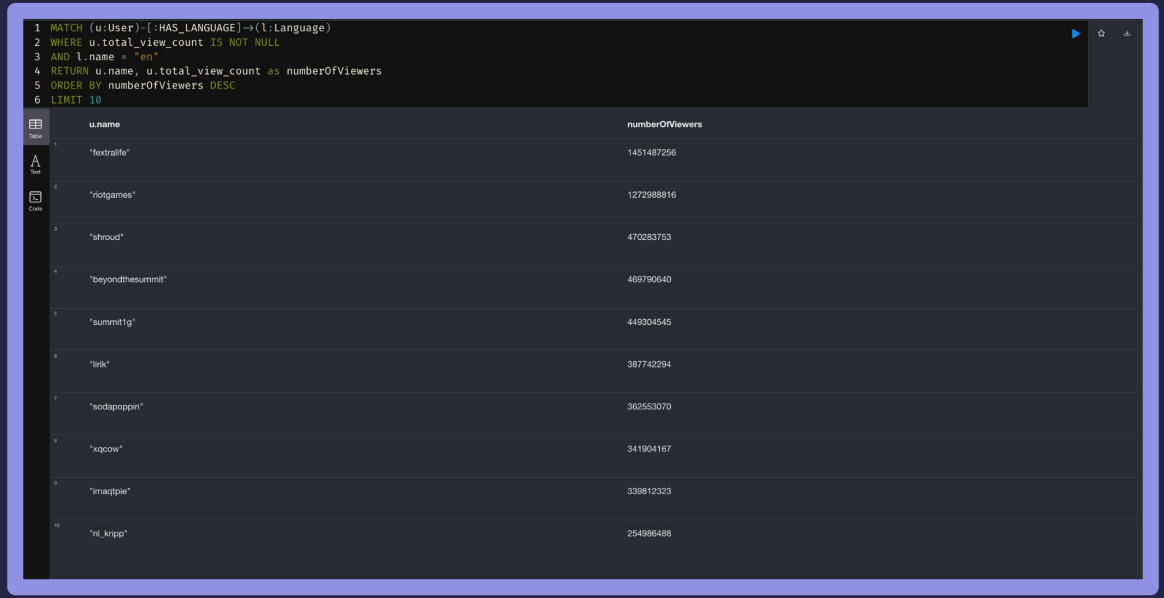
SparkVision
Marketing agency

Sparkvision Agency

Query output: Streamers with most number of viewers

Objective: Identify streamers with large followings and target ads towards them, potentially in the form of team sponsorships.

Query components: This query filters Twitch users and streamers based on the English language and total view count, generating the top 10 users with the highest view count and their respective names, sorted in descending order by number of viewers. The query is composed of four parts: "MATCH" to define the starting point, "WHERE" to filter the data, "RETURN" to specify the output, and "ORDER BY" and "LIMIT" to sort and limit the results.



```
MATCH (u:User)-[:HAS_LANGUAGE]→(l:Language)
WHERE u.total_view_count IS NOT NULL
AND l.name = "en"
RETURN u.name, u.total_view_count as numberOfViewers
ORDER BY numberOfViewers DESC
LIMIT 10
```

Table | Text | Code

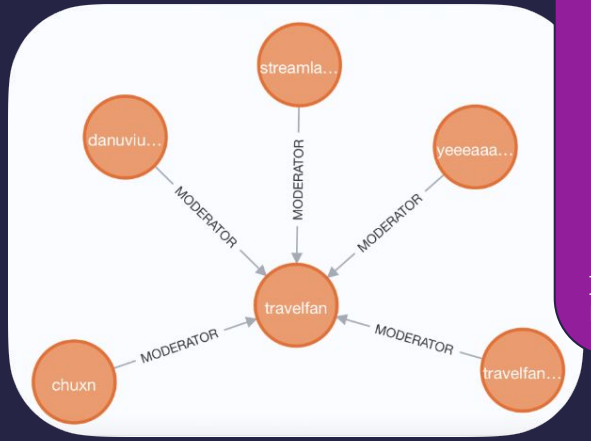
	u.name	numberOfViewers
1	"fextralife"	1451487256
2	"riotgames"	1272988816
3	"shroud"	470283753
4	"beyondthesummit"	469790640
5	"summit1g"	449304545
6	"lirik"	387742294
7	"sodapoppin"	362553070
8	"xqcow"	341904167
9	"imaqtpie"	339812323
10	"nl_kripp"	254986488

Neo4j - Query 2

Query output: Graph of RiffTrax's streams, along with up to 5 streams and up to 5 users that are connected to that stream

Objectives:

1. Identify the networks of the streamer with the most watch time and stream time (including involved fellow streamers, moderators, and VIPs) and target ads towards them, potentially in the form of sponsorships or collaborations.
2. Identify the most active chatters and use targeted advertising to reach them and their networks



Query:

```
MATCH (u:User {name:
"xqcow"})--(s:Stream)
WITH s LIMIT 1
CALL {
  WITH s
  MATCH p=(s)<--(:Stream)
  RETURN p
  LIMIT 5
  UNION
  WITH s
  MATCH p=(s)<--(u)
  RETURN p
  LIMIT 5
}
RETURN p
```

Query components: This query retrieves Twitch users and streamers data for a specific user ("rifftrax") and stream. It unites the user and stream nodes using the "--" relationship, restricts the stream nodes to one instance, and passes them to a "CALL" procedure that looks for incoming streams and users that have streamed the current stream, returning the paths produced by these searches. The results are limited to 10 paths.

SparkVision
Marketing agency

Sparkvision Agency

Neo4j - Query 3

SparkVision
Marketing agency

Sparkvision Agency

Query output: Graph of xQcOW's streams, along with up to 5 streams and up to 5 users that are connected to that stream

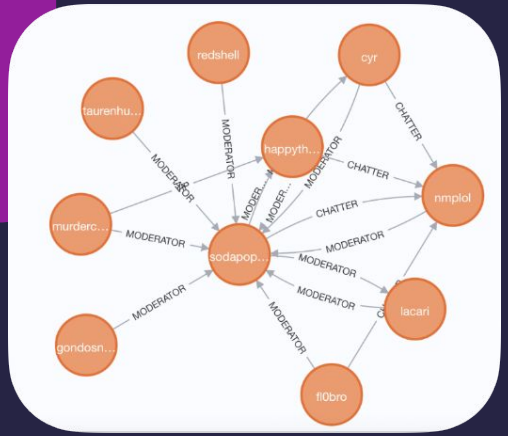
Objectives:

1. Identify the networks of the streamer with the most stream time (including involved fellow streamers, moderators, and VIPs) and target ads towards them, potentially in the form of sponsorships or collaborations.
2. Identify the most active chatters and use targeted advertising to reach them and their networks

Query:

```
MATCH (u:User {name:
"xqcow"})--(s:Stream)
WITH s LIMIT 1
CALL {
  WITH s
  MATCH p=(s)<--(:Stream)
  RETURN p
  LIMIT 5
  UNION
  WITH s
  MATCH p=(s)<--(u)
  RETURN p
  LIMIT 5
}
RETURN p
```

Query components: This query retrieves Twitch users and streamers data for a specific user ("xqcow") and stream. It unites the user and stream nodes using the "--" relationship, restricts the stream nodes to one instance, and passes them to a "CALL" procedure that looks for incoming streams and users that have streamed the current stream, returning the paths produced by these searches. The results are limited to 10 paths.



Conclusion

SparkVision
Marketing agency

Sparkvision Agency

In conclusion, MongoDB and Neo4j have proven extremely valuable for the marketing agency's decision-making by enabling the organizing, visualization, and analysis of complex data structures, resulting in deeper insights about streamers, their habits and the way they share their content; this enabled us to generate information for a more useful and powerful marketing campaign for eventual businesses interested in advertising on the Twitch platform.

Each technology has its own benefits. When it comes to MongoDB and Neo4j, the two platforms complement each other significantly in handling social network data. MongoDB was used to store large volumes of raw data and used queries that drew out the streamers most ideal for an advertising campaign. Afterwards, Neo4j used MongoDB's outputs to generate graphs that allowed for the analysis of the relationships between entities–projecting a more expansive view of the network of the chosen streamers. This provided valuable insights into the behavior and interactions between Twitch users and how each player in the network can be tapped on in an advertising campaign.