Global Art Exploration and Analysis

DSC202 – Winter 2025

John Driscoll
Taylor Martinez
Thuy Nguyen

1 Problem Statement

1.1 Project Overview

The goal of this project is to explore and analyze the relationships between artists, their artworks, and various artistic movements over time and geography. By leveraging different technologies like relational databases (PostgreSQL), graph databases (Neo4j), and caching systems (Redis), we provide insights that are not possible with a single standalone data management system. The integration of these systems enables real-time access to frequently viewed artworks, uncovers meaningful relationships, and performs complex queries efficiently, making it a powerful tool for cultural preservation, data-driven insights, interdisciplinary research, and performance optimization.

1.2 Importance of the Project

Our project plays a critical role in cultural preservation by uncovering patterns, trends, and relationships in art history, offering a deeper understanding of how artistic movements evolve and influence one another, how artists have influenced each other, and how they may be relevant to trends in society today.

1.3 Data Requirements

The project integrates multiple datasets and real-time data sources to provide a comprehensive exploration of artists, artworks, and their relationships. The primary datasets are the MoMA (Museum of Modern Art) dataset and the PainterPalette dataset, which are enhanced with additional data gathered from WikiData queries, and popularity data obtained by scraping Google Trends and caching.

The MoMA dataset provides metadata on artworks, including titles, artists, creation dates, mediums, dimensions, and acquisition dates, as well as biographical details about artists such as names, nationalities, genders, birth and death years. However, the unique identifier in the table Getty ULAN ID isn’t present in other datasets, making linking artworks across datasets and external sources a challenge. We use a composite key of artwork and creation date to ensure stability throughout this project.

The PainterPalette dataset (enhanced with additional data from WikiData queries) complements MoMA’s data by offering additional biographical information, artistic styles, movements, and relationships between artists, such as influences, mentorships, and collaborations.

To ensure real-time relevance, the platform integrates recent popularity metrics by scraping Google Trends. This data includes changes in popularity surrounding an item of interest over time and is cached using Redis for fast retrieval and improved performance.

2 Detailed Coverage

2.1 Procedure Overview

This project integrates structured data, graph-based relationships, and real-time trend data into a cohesive system. All databases are connected through Python.

2.2 Postgres Informational Database

The database schema for this project was designed to organize and store comprehensive data about artists, their works, movements, and various other related attributes. The PostgreSQL setup was structured to accommodate a normalized relational model in order to efficiently handle the complex relationships between entities such as artists, artworks, movements, schools, and places. The key components of the schema isolate repetitive fields into their own tables to assist with data minimization.

The Artists table serves as the central table, containing key information about each artist, including their name, birth year, nationality, citizenship, gender, career start and end years, and death year. A critical feature of this table is the inclusion of a birth_year_key column, which is a generated column that uses the COALESCE function to replace any NULL birth years with a placeholder value (-999999). This approach allows for the unique identification of artists, even when their birth year is unknown. The primary key for the Artists table is a composite key, consisting of the artist’s name and the birth_year_key. Additionally, a partial unique constraint is applied to ensure that no two artists with NULL birth years share the same name, enforcing data consistency.

The Places table holds the names of various places related to the artists. This table is linked to the Artists table through the Artist_Birth_Places and Artist_Death_Places many-to-many tables. These two tables capture the birth and death places of the artists, respectively, using a composite key made up of the artist’s name, birth year, and place name. The use of composite primary keys and foreign keys ensures referential integrity and efficient deletion handling if an artist or place is removed from the database.

The Occupations and Artist_Occupations tables model the relationship between artists and their occupations over time. Each occupation is stored in the Occupations table with a primary key on the occupation name. The Artist_Occupations table creates a many-to-many relationship between artists and occupations, capturing which occupation each artist held during their career. Similar to the birth and death places, the Artist_Occupations table uses a composite primary key of artist name, birth year, and occupation name, with foreign key references to the Artists and Occupations tables.

The Schools and Artist_Schools tables capture information about the schools that artists attended, with the Artist_Schools table storing time periods as a JSONB data type. This structure enables flexibility in representing the varying durations for which artists attended each school. A composite primary key is again used for the Artist_Schools table, consisting of the artist's name, birth year, and school name, ensuring that relationships are properly established between artists and their respective schools.

The Artworks table stores details about individual artworks, including the title, artwork date, medium, department, acquisition date, and classification. The Artworks_Artists table captures the many-to-many relationship between artworks and artists, as a single artwork can be created by multiple artists, and an artist can create multiple artworks. The Artworks_Artists table also includes a composite primary key of artwork title, artwork date, artist name, and birth year key, ensuring the correct mapping of artists to their artworks.

The Movements table stores the names of movements, while the Artist_Movements table creates a many-to-many relationship between artists and movements. The Artist_Movements table includes an additional attribute, years_active, to track the years during which an artist was involved in each movement.

The Styles table contains information about artistic styles, and the Artist_Styles table tracks the styles associated with each artist. The Artist_Styles table also includes attributes such as style_count and style_years, which store the frequency of style usage and the years the artist worked within that style, respectively.

The Artist_Relationships table stores information regarding the relationships between artists – whether they had a teacher/pupil relationship, were friends, or influenced one another.

2.3 Neo4j Graph Database

To complement the relational data model in PostgreSQL and enable deeper insights into complex relationships, the project sets up multi-level relationships in Neo4j through its representation of entities as nodes and their connections as relationships.

This project’s Neo4j implementation connects artists, artworks, art movements, geographical locations, occupations, and art schools through direct and indirect relationships to facilitate rapid traversal and cluster identification.

We implemented the Nodes and their attributes as follows:

1. **Artist**: Stores artist-specific details, including name, birth and death dates, gender, and career timelines.
2. **Artwork**: Captures information about individual artworks such as title, creation date, medium, and classification.
3. **Art Movement**: Represents artistic movements (e.g., Impressionism, Cubism) with attributes such as movement name and period.
4. **Place**: Includes possible birth places and death places of artists.
5. **Art School**: Contains educational institution information, including school names, locations, and active periods.
6. **Occupation**: Represents various professional roles (e.g., painter, sculptor, engraver) held by artists.

We implemented the relationships between nodes as follows:

1. **Artist → INFLUENCED BY→ Artist:** Represents one artist being influenced by another, tracking historical artistic influences.
2. **Artist → INFLUENCED → Artist:** Represents one artist influencing another, tracking historical artistic influences.
3. **Artist → TAUGHT → Artist:** Captures mentorship or educational relationships between artists.
4. **Artist → IS PUPIL OF → Artist:** Captures studentship or educational relationships between artists.
5. **Artist → FRIENDS_WITH → Artist:** Records personal friendships between artists, providing social context.
6. **Artist → COLLABORATED_WITH → Artist:** Indicates artists who collaborated professionally on projects or artworks.
7. **Artist → CREATED → Artwork:** Links artists to their respective artworks.
8. **Artist → BELONGS_TO → Movement:** Connects artists to specific artistic movements they participated in.
9. **Artwork → BELONGS_TO → Movement:** Associates artworks directly with relevant artistic movements.
10. **Artist → CITIZEN_OF → Country:** Captures citizenship or national affiliations of artists.
11. **Artist → WORKED_IN → Country:** Tracks geographic locations where artists lived, worked, or produced art.

12. **Artist → WORKED_AS → Occupation:** Documents occupations or professional roles artists undertook throughout their careers.

2.4 REDIS Trend Caching Database

We implemented Redis as a caching layer and recent trends storage solution to complement our structured PostgreSQL database and interconnected Neo4j graph database. Redis's flexible, unstructured data format allows us to store and retrieve frequently accessed data and recent activity trends quickly, without the constraints of a rigid schema.

Our Redis implementation leverages hashes to efficiently store weeks of recent data. We append new data points directly to these hashes, enabling rapid access to current trends and popular entities. To clearly distinguish between different entity types and avoid naming collisions, we format Redis keys consistently as follows:

```
trends:{entity_type}:{entity_id}
```

This naming convention ensures each entity's trend data remains distinct and easily retrievable. Additionally, we store metadata alongside each Redis hash to provide context and simplify data management. We add metadata fields such as entity type, entity identifier, and the timestamp of the last update to make each cached entity self-descriptive, streamlining data retrieval, debugging, and maintenance tasks.

Finally, we configure Redis to force periodic saves to disk, ensuring local persistence of cached data. This step is crucial because our infrastructure does not run continuously, and forced persistence prevents data loss during downtime or system restarts.

2.3 Data Preprocessing and Integration

Combining and Processing Data Sources

To create a comprehensive dataset for our analyses of artists, their works, movements, and collaborations, we combined and processed several datasets, including MOMA's artist and artwork data, PainterPalette, and WikiData. The process involved cleaning, transforming, and merging the data to ensure consistency and accuracy across the combined dataset.

We began by cleaning the MOMA Artists dataset, ensuring that the ConstituentID column contained only unique values by removing duplicates. We then extracted the birth and death years of artists from the ArtistBio column using regular expressions. Any missing values were filled with zeros, and the data type was converted to integers. Additionally, we filled in missing values in the Nationality and Gender columns with "Unknown" to maintain data consistency.

Next, we cleaned the MOMA Artworks dataset by removing duplicate entries based on the ObjectID and Title columns. The ConstituentID column, which contained a list of artist IDs associated with each artwork, was split and exploded to represent each artwork-artist relationship as a separate row. We then filtered out any invalid ConstituentID values, such as 0. We also extracted the start and end years from the Date column of the artworks dataset. In cases where the end year was represented with two digits (e.g., "1976-77"), we processed the data to generate a complete four-digit year, ensuring consistent and accurate year formatting.

For the PainterPalette dataset, we filled in missing values in key columns such as Nationality, gender, styles, movement, and occupations with "Unknown." We also converted columns containing multiple entries, like styles and Influencedby, into lists, allowing for more efficient merging with the other datasets. We cleaned the Wikidata dataset in a similar manner.

After cleaning each dataset, we merged them into a single, unified dataset. First, we merged PainterPalette and WikiData based on the artist's name, aligning key attributes like Nationality, citizenship, gender, and birth year. In cases where both datasets provided values for the same column, we filled the missing values from PainterPalette with the corresponding values from WikiData. We then merged this combined dataset with the MOMA Artists dataset, ensuring that the Nationality field from MOMA was retained and eliminating any duplicate columns.

The final merge involved incorporating data from the MOMA Artworks dataset. We merged this dataset with the combined artist data based on the artist and Artist columns, adding relevant information about the artists' artworks, such as titles, creation dates, and start and end years. During this merge, we addressed duplicate columns, such as those with suffixes _x and _y, by combining them and retaining the most accurate data. For example, we merged the birth_year and death_year columns, keeping the _x values and filling any missing entries with data from the _y columns.

PostgreSQL

To ensure the data was compatible with our PostgreSQL database, we converted each column to the appropriate data type. We verified string columns, such as Nationality, gender, and styles, were in the correct format, while numeric columns like birth_year and death_year needed to be converted to integers. We reformatted list-based columns, such as styles and Influencedby, for insertion. We also addressed any invalid or inconsistent data, such as ensuring that all years were valid and that no years were set in the future. For any missing values that could not be filled with valid data, we inserted the placeholder "Unknown."

After cleaning and transforming the data, we exported the final dataset to a CSV file and prepared it for insertion into the PostgreSQL database. Using psycopg2, we connected to the PostgreSQL database and inserted the data into the appropriate tables. During this process, we ensured that the data types in the final dataset matched the structure of the PostgreSQL tables, and we applied necessary cleaning steps, such as splitting and trimming the occupations column.

Neo4J

The process of integrating data into Neo4j required a comprehensive data preprocessing pipeline to convert the structured relational data from PostgreSQL into an appropriate graph format. As the original dataset was stored in relational tables, we focused on transforming this data to fit the graph model, enabling effective querying and exploration of relationships between artists, artworks, movements, and other related entities.

The initial phase of preprocessing involved resolving missing and inconsistent data. For example, we decided to handle NULL values in birth years by utilizing a generated column in the PostgreSQL schema. The birth_year_key was introduced, using the COALESCE function to substitute missing birth years with a placeholder value (-999999). This ensured consistency in handling NULL birth years while also ensuring uniqueness in composite primary keys. This approach was especially important when mapping data to Neo4j, which requires unique identifiers for nodes. We enforced unique artist names for NULL birth years by creating a partial unique constraint, ensuring that no two artists with missing birth years had the same name. Neo4j was more sensitive about escape characters than Postgres, so we had to scrub and reformat strings before inserting the data.

We began data integration into Neo4j by creating nodes for each entity in the schema. Each entity, such as Artists, Artworks, Movements, Schools, Occupations, Styles, Places, and Exhibitions, was represented as a node, with attributes corresponding to the relevant fields in the relational tables. For example, the Artist node was populated with attributes such as name, nationality, gender, birth year, and career. During this process, we used MERGE operations to prevent the duplication of existing nodes in the graph. Relationships between nodes, such as works as, attended, created, influenced, and belongs to, were created to model the connections in the data. For instance, artists were linked to their occupations with the relationship *(:Artist)-[:WORKED_AS]->(:Occupation)*.

A significant challenge we faced was the creation of many-to-many relationships, which are common in the relational schema but naturally represented as edges in Neo4j. In PostgreSQL, we captured many-to-many relationships using intermediate tables such as Artist_Birth_Places, Artworks_Artists, and Artist_Movements. In Neo4j, we directly modeled these relationships. For example, artworks were linked to artists through the relationship *(:Artist)-[:CREATED]->(:Artwork)*, while artists were linked to movements through *(:Artist)-[:BELONGS_TO]->(:Movement)*. This direct mapping simplified the graph structure and allowed for more efficient traversal of connected data.

We also used TEXT columns in PostgreSQL to store complex attributes like time periods and years active. These attributes were appropriately transformed to maintain their integrity and structure when transferred to Neo4j. For example, the years_active for artists in movements was stored as an in-text array in PostgreSQL and was preserved in the graph as an attribute of the relationship *[:BELONGS_TO]*, allowing for detailed querying about the duration of an artist's involvement in specific movements.

We created indexes in Neo4j to optimize query performance, similar to PostgreSQL. Indexing key attributes like artist names, artwork titles, and movement names enabled efficient searches and quick retrieval of relevant data.

2.4 Real-Time Data Handling

Redis

To populate Redis with accurate and timely trend data, we utilized Google Trends as a primary data source. Given that Google Trends actively blocks automated web scraping, we configured and initialized a Python-controlled browser using Selenium. To bypass scraping restrictions, we carefully edited the browser headers, user-agent strings, and interaction behaviors to closely emulate human browsing patterns. This approach allowed us to reliably access and download popularity data from Google Trends without triggering automated detection mechanisms.

After downloading the popularity data files from Google Trends, we performed thorough data cleaning and standardization procedures. This included removing inconsistencies, standardizing date formats, and ensuring uniformity across all downloaded files. The cleaned and standardized data was then ready for ingestion into Redis, ensuring consistency and accuracy for subsequent analysis.

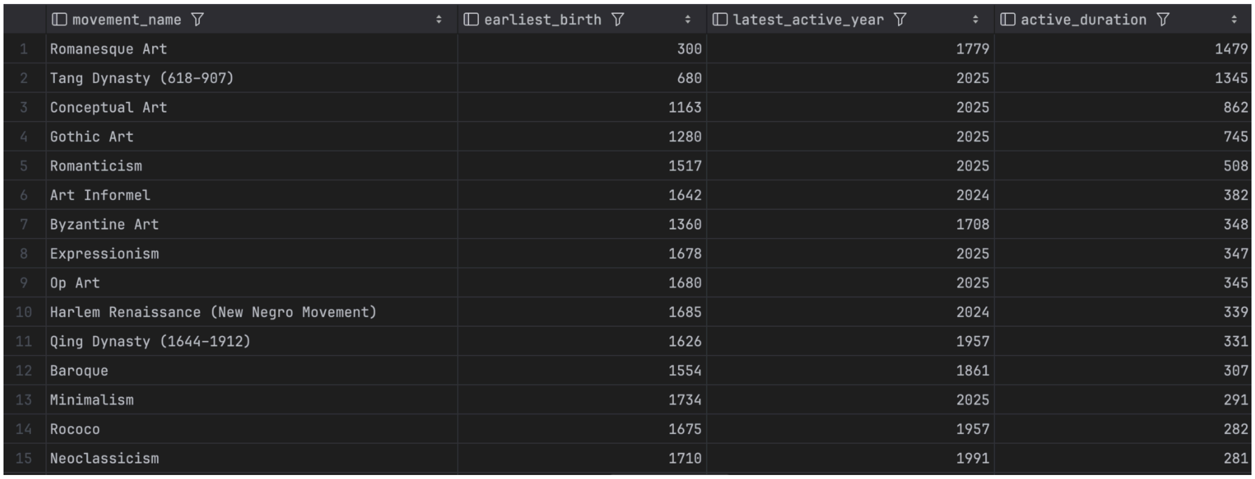
Within Redis, recent popularity data was stored using hashes, taking advantage of Redis's unstructured format. Each hash was designed to hold several weeks of recent data points, allowing rapid retrieval and analysis of current trends. Keys were systematically formatted as descriptive slugs (e.g., artist:vincent-van-gogh:popularity) to ensure clarity and ease of access. To maintain contextual clarity and facilitate efficient data management, metadata was appended to each Redis hash. Metadata included the entity type, entity identifier, and timestamps indicating the last update. This metadata was stored alongside the main data hashes.

2.5 Querying and Data Analysis

PostgreSQL

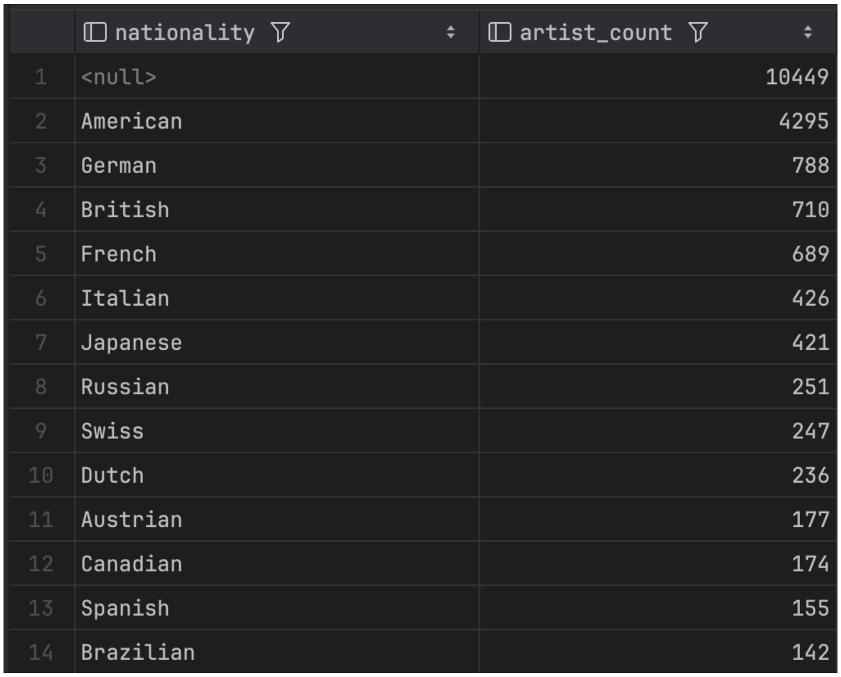
We utilized PostgreSQL to answer several key questions related to artists, movements, and trends that shaped the art world over the last two centuries. By querying the relational database, we were able to uncover insights into the longevity of art movements, the geographical origins of artists, the evolution of artistic occupations, the influence of art schools, and the longevity of artistic styles.

We first decided to look at which art movements had the longest periods of active artists. This question aimed to identify whether certain movements lasted longer and whether those movements had a lasting impact on art history. We analyzed the start and end years of various movements and filtered the data for those with the longest durations. Our SQL query for this involved grouping artists by their associated movements and calculating the span of each movement based on the earliest birth and the latest active year found. Based on the results below, we can see that the Romanesque Art and Tang Dynasty movements have both survived for over 1,300 years and still exist today, among other movements like Maximalism that have not been around for nearly as long.



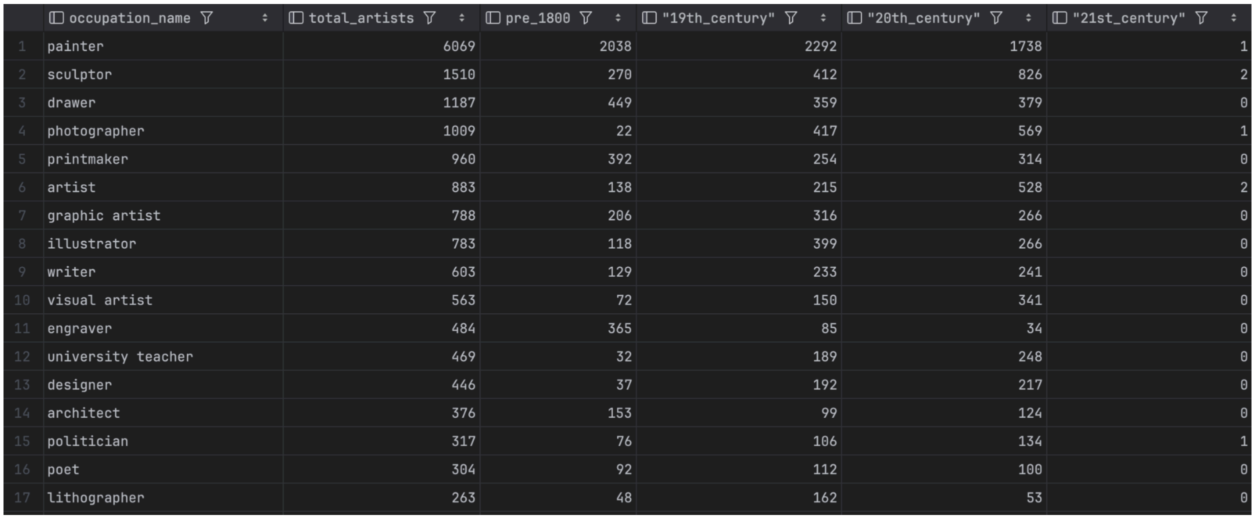
	movement_name	earliest_birth	latest_active_year	active_duration
1	Romanesque Art	300	1779	1479
2	Tang Dynasty (618-907)	680	2025	1345
3	Conceptual Art	1163	2025	862
4	Gothic Art	1280	2025	745
5	Romanticism	1517	2025	508
6	Art Informel	1642	2024	382
7	Byzantine Art	1360	1708	348
8	Expressionism	1678	2025	347
9	Op Art	1680	2025	345
10	Harlem Renaissance (New Negro Movement)	1685	2024	339
11	Qing Dynasty (1644-1912)	1626	1957	331
12	Baroque	1554	1861	307
13	Minimalism	1734	2025	291
14	Rococo	1675	1957	282
15	Neoclassicism	1710	1991	281

Next, we queried the nationality data and aggregated it over time to determine which nationalities have produced the most artists in the past 200 years. This helped us understand how different regions have contributed to global art over time and whether certain countries, like France in the 19th century or the U.S. in the 20th century, led in artistic production. The resulting data showed that the United States has dominated within the past two centuries, though we attribute this to bias within our input data files. For instance, a significant portion of our data comes from MOMA, which is based in the United States. We can also see that the majority of top nationalities are from the Western world and/or imperialist nations.



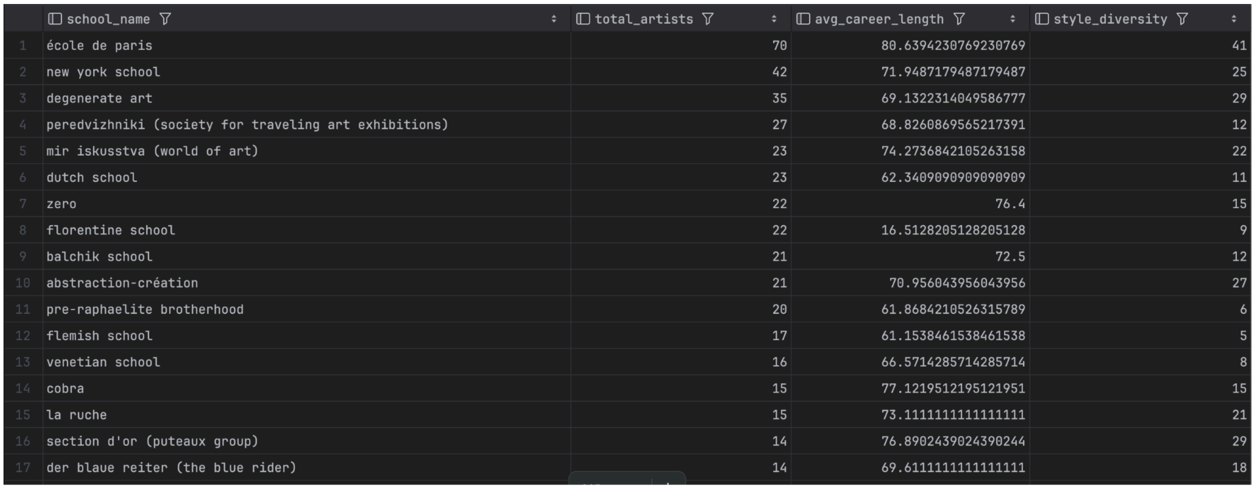
	nationality	artist_count
1	<null>	10449
2	American	4295
3	German	788
4	British	710
5	French	689
6	Italian	426
7	Japanese	421
8	Russian	251
9	Swiss	247
10	Dutch	236
11	Austrian	177
12	Canadian	174
13	Spanish	155
14	Brazilian	142

We also looked at the most common occupations among artists and how they have changed over time, tracing the shift in artistic professions from roles like painters and engravers to roles such as sculptors or photographers. This analysis highlighted how certain professions gained prominence while others declined, pointing to advancements in technology along with cultural shifts.



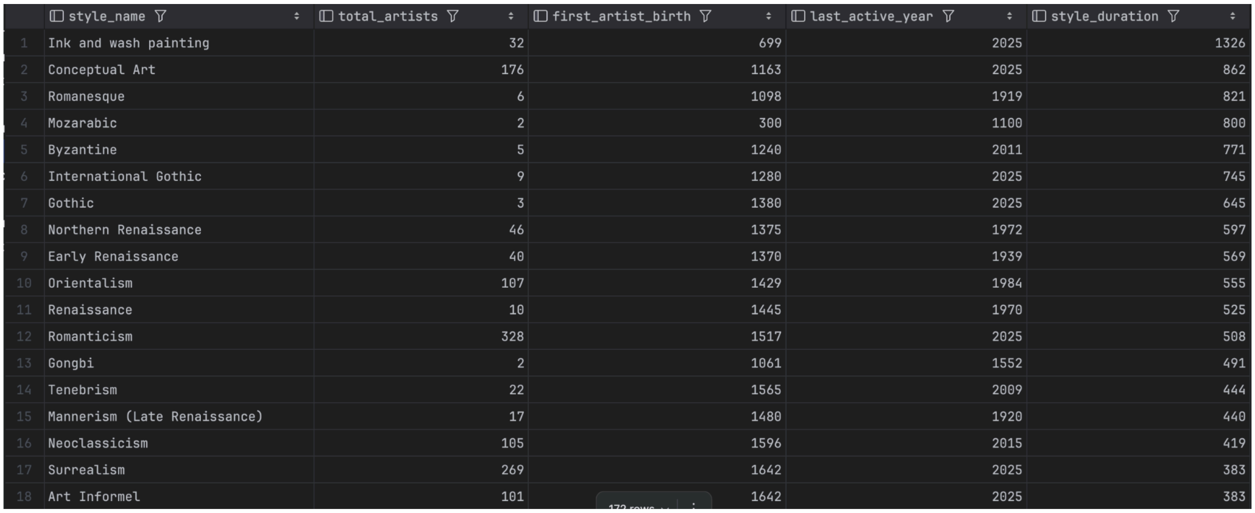
	occupation_name	total_artists	pre_1800	"19th_century"	"20th_century"	"21st_century"
1	painter	6069	2038	2292	1738	1
2	sculptor	1510	270	412	826	2
3	drawer	1187	449	359	379	0
4	photographer	1009	22	417	569	1
5	printmaker	960	392	254	314	0
6	artist	883	138	215	528	2
7	graphic artist	788	206	316	266	0
8	illustrator	783	118	399	266	0
9	writer	603	129	233	241	0
10	visual artist	563	72	150	341	0
11	engraver	484	365	85	34	0
12	university teacher	469	32	189	248	0
13	designer	446	37	192	217	0
14	architect	376	153	99	124	0
15	politician	317	76	106	134	1
16	poet	304	92	112	100	0
17	lithographer	263	48	162	53	0

To understand the impact of educational institutions on art history, we determined which schools produced the most successful artists based on influence, longevity, and style diversity, implying they had a significant impact on the art world. We found that certain art schools like Ecole de Paris and New York School have potentially had a large influence on shaping global art movements.



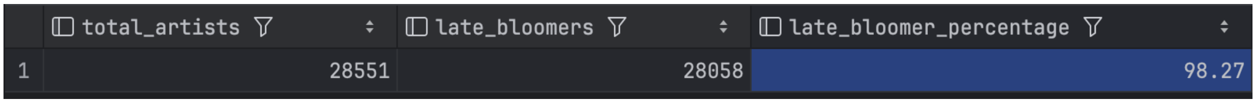
	school_name	total_artists	avg_career_length	style_diversity
1	école de paris	70	80.6394230769230769	41
2	new york school	42	71.9487179487179487	25
3	degenerate art	35	69.1322314049586777	29
4	peredvizhniki (society for traveling art exhibitions)	27	68.8260869565217391	12
5	mir iskusstva (world of art)	23	74.2736842105263158	22
6	dutch school	23	62.3409090909090909	11
7	zero	22	76.4	15
8	florentine school	22	16.5128205128205128	9
9	balchik school	21	72.5	12
10	abstraction-création	21	70.956043956043956	27
11	pre-raphaelite brotherhood	20	61.8684210526315789	6
12	flemish school	17	61.1538461538461538	5
13	venetian school	16	66.5714285714285714	8
14	cobra	15	77.1219512195121951	15
15	la ruche	15	73.1111111111111111	21
16	section d'or (puteaux group)	14	76.8902439024390244	29
17	der blaue reiter (the blue rider)	14	69.6111111111111111	18

In exploring the most common artistic styles and their relevance over time, we were able to determine whether some styles evolved quickly or stayed dominant for longer periods of time. We analyzed the styles associated with each artist and tracked their career duration, identifying which styles were short-lived and which had long-term appeal. Some styles were found to be transient, with artists moving quickly from one trend to the next, while others remained relevant for much longer, influencing art movements across multiple generations. This analysis revealed the cyclical nature of artistic trends and styles. For example, we can see from the results below that ink and wash painting style of art has existed for over 1,300 years and continues to live on. Other styles like Gongbi lost popularity several centuries ago.



	style_name	total_artists	first_artist_birth	last_active_year	style_duration
1	Ink and wash painting	32	699	2025	1326
2	Conceptual Art	176	1163	2025	862
3	Romanesque	6	1098	1919	821
4	Mozarabic	2	300	1100	800
5	Byzantine	5	1240	2011	771
6	International Gothic	9	1280	2025	745
7	Gothic	3	1380	2025	645
8	Northern Renaissance	46	1375	1972	597
9	Early Renaissance	40	1370	1939	569
10	Orientalism	107	1429	1984	555
11	Renaissance	10	1445	1970	525
12	Romanticism	328	1517	2025	508
13	Gongbi	2	1061	1552	491
14	Tenebrism	22	1565	2009	444
15	Mannerism (Late Renaissance)	17	1480	1920	440
16	Neoclassicism	105	1596	2015	419
17	Surrealism	269	1642	2025	383
18	Art Informel	101	1642	2025	383

Lastly, we completed an analysis to determine how many artists reached their peak productivity late in life. Our findings indicated that a significant number of artists achieved their most important works in the second half of their lives (“late bloomers”).



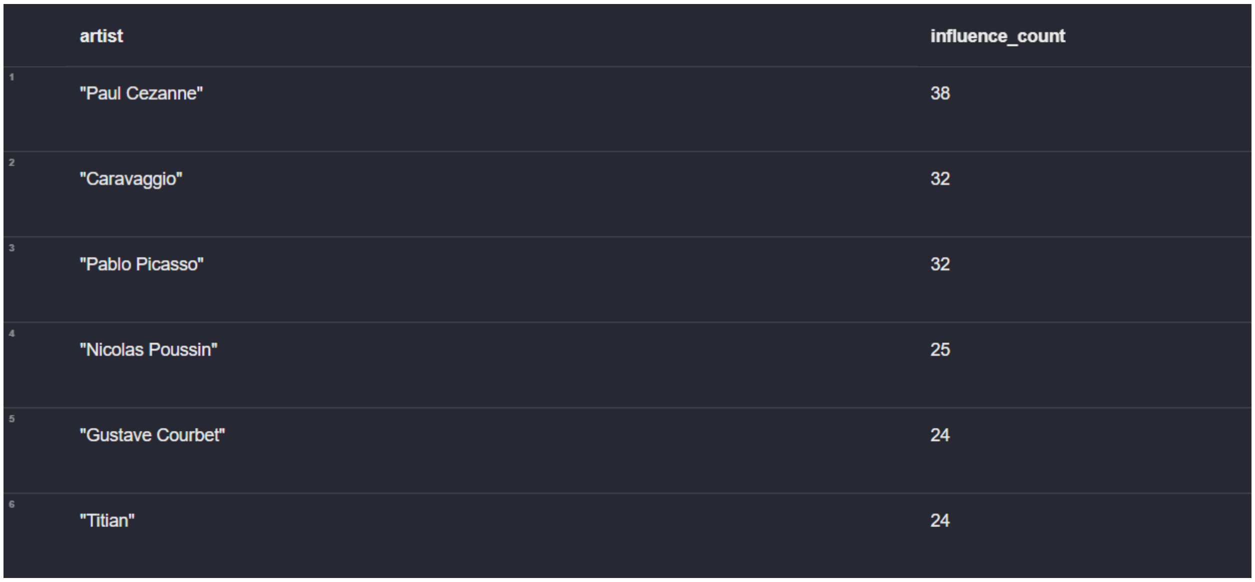
	total_artists	late_bloomers	late_bloomer_percentage
1	28551	28058	98.27

In conclusion, PostgreSQL enabled us to explore trends in art movements, nationalities, occupations, and more, offering meaningful insights into the evolution of art and artists over time. The final data outputs from these analyses, as well as the SQL queries used, are available in our GitHub repository for further reference.

Neo4j

We utilized Neo4j to analyze complex relationships between artists, movements, and artistic styles, leveraging the power of graph databases to uncover hidden patterns in the relationships that would be difficult to detect in a traditional relational database like PostgreSQL. By modeling the art world as an interconnected network, we were able to explore how artistic influence spreads across generations, how art schools shape collaboration patterns, and how certain artists act as bridges between different styles and movements.

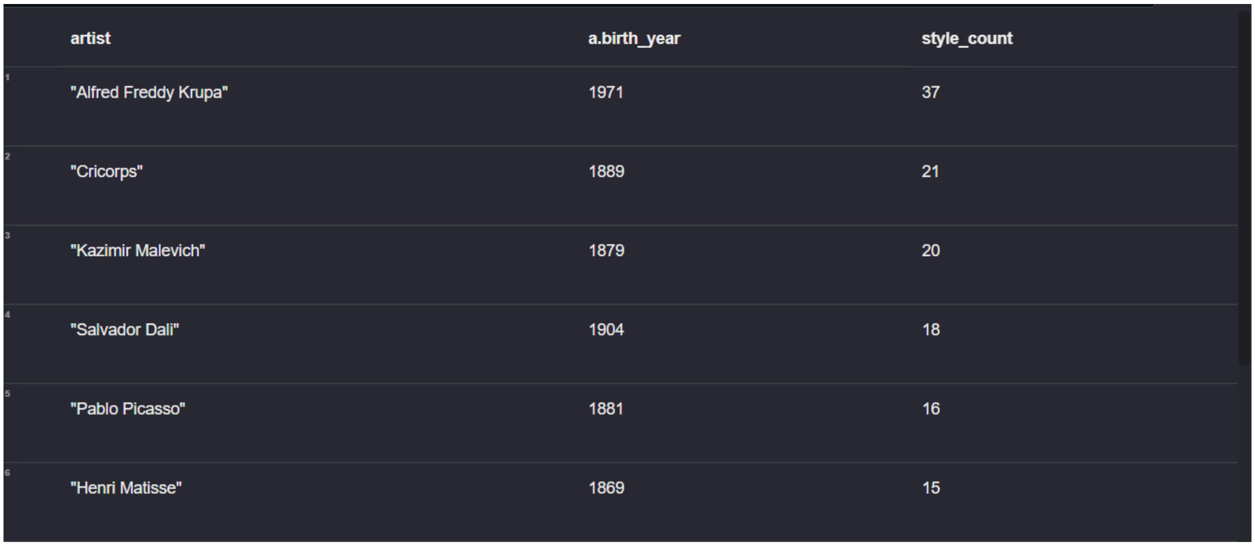
One of our primary investigations focused on identifying the most influential artists in terms of mentorship, collaboration, and cross-movement connections. Using graph traversal queries, we analyzed how influence propagates over multiple levels—whether an artist directly influenced another or if their impact traveled through multiple intermediaries. This was extremely insightful because this is something that is not able to be done solely using PostgreSQL, and it highlighted how deep a graph network can go. We identified key figures who served as inspiration in their respective artistic network, facilitating the transmission of ideas and styles across different eras.



	artist	influence_count
1	"Paul Cezanne"	38
2	"Caravaggio"	32
3	"Pablo Picasso"	32
4	"Nicolas Poussin"	25
5	"Gustave Courbet"	24
6	"Titian"	24

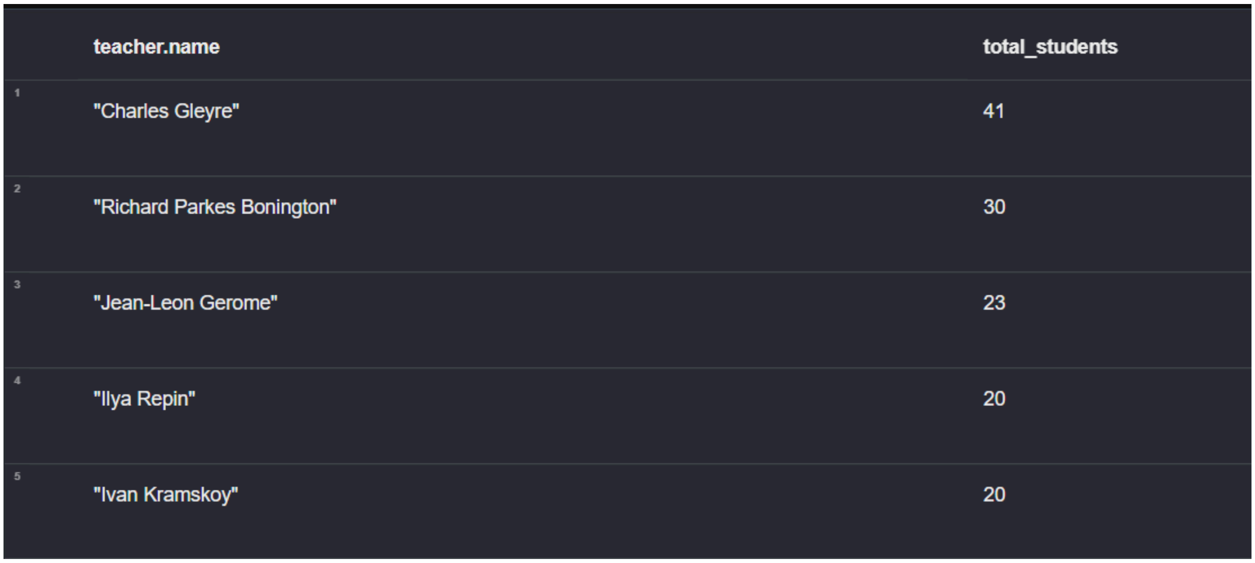
Next, we examined art schools as potential sources of collaboration, determining whether artists who studied at the same institutions were more likely to collaborate in their careers. Our query revealed that artists who attended the same school tended to form tight-knit clusters, with certain institutions, such as the École des Beaux-Arts and the Bauhaus, which produced particularly dense collaboration networks. We also explored whether these school-based artistic connections extended beyond direct contemporaries by analyzing multi-hop relationships between alumni across different generations.

In addition, we explored how different artistic styles interconnect by identifying artists who served as bridges between multiple styles. Unlike SQL, which relies on joins across multiple tables, Neo4j allowed us to traverse these relationships dynamically, revealing that some artists—such as Pablo Picasso and Kazimir Malevich—were instrumental in linking historically distinct styles. These artists not only worked in multiple artistic styles but also influenced artists beyond their direct stylistic affiliations, creating a web of artistic evolution.



	artist	a.birth_year	style_count
1	"Alfred Freddy Krupa"	1971	37
2	"Cricorps"	1889	21
3	"Kazimir Malevich"	1879	20
4	"Salvador Dali"	1904	18
5	"Pablo Picasso"	1881	16
6	"Henri Matisse"	1869	15

We developed a Neo4j query to analyze student counts and identify the most prolific teachers in art history. By leveraging graph traversal, this query follows mentorship chains up to three levels deep, revealing which artists mentored the highest number of students. The results highlight key figures who played a crucial role in shaping future generations of artists, either through direct mentorship or extended influence. This analysis provides insights into which artists had the greatest educational impact, helping us understand the transmission of artistic techniques, philosophies, and innovations across time. Notably, artists with a high student count often contributed to the establishment of major art movements or schools of thought, reinforcing the importance of mentorship in the evolution of art.



	teacher.name	total_students
1	"Charles Gleyre"	41
2	"Richard Parkes Bonington"	30
3	"Jean-Leon Gerome"	23
4	"Ilya Repin"	20
5	"Ivan Kramskoy"	20

In conclusion, Neo4j allowed us to explore the deep interconnectedness of the art world, leveraging graph analytics to uncover relationships and trends that would be challenging to detect with SQL-based queries. The final outputs from these analyses, as well as the Cypher queries used, are available in our GitHub repository for further exploration.

Redis and Web Scraping

We used REDIS and web-scraping of Google Trends to answer questions about current art topics surrounding the following question:

How do real life events impact public awareness and visibility of art topics?

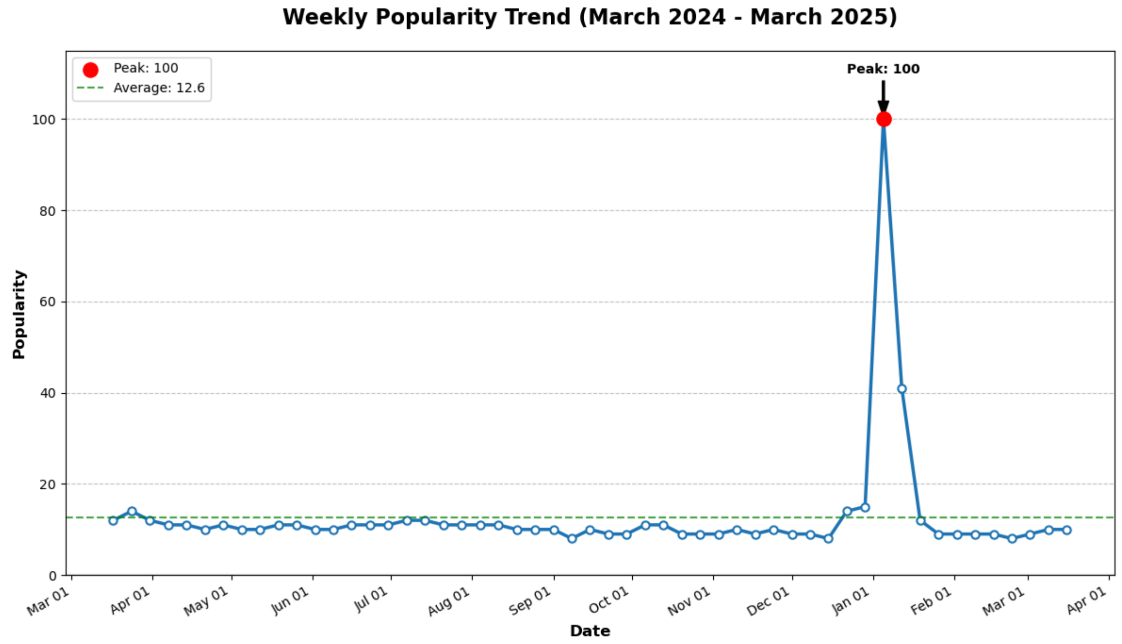
These sub-questions follow from investigating the idea that real-life events impact art-related popularity trends. We can analyze how catastrophic events can paradoxically increase the visibility and cultural relevance of art institutions, potentially changing visitation patterns, online searches, social media engagement, and public discourse about cultural preservation. An example of a current catastrophic event is the LA wildfires.

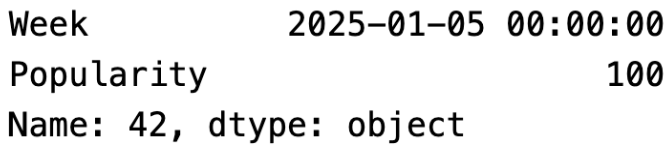
The code used to answer these questions is located in the github repository in “redis_questions.ipynb”.

1. How did the Los Angeles wildfires impact public awareness and visibility of The Getty Museum?

The fires nearly destroyed the Getty Museum and its priceless collections, and likely generated substantial media coverage and public interest. We can investigate if this is true using our redis integration.

After scraping popularity data for The Getty and populating redis. We find that the time with most interest was during the fires.





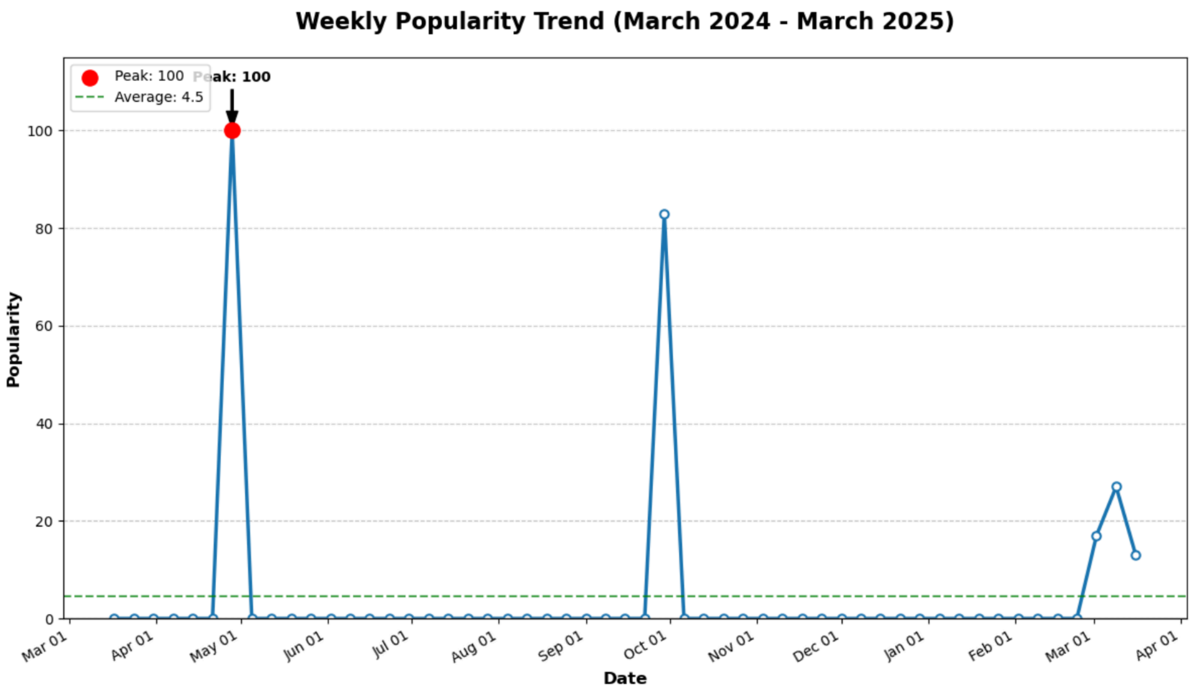
```
Week          2025-01-05 00:00:00
Popularity                    100
Name: 42, dtype: object
```

We see that the Los Angeles wildfires that threatened The Getty Museum altered public perception and awareness of this cultural institution.

2. How did the LA wildfires impact the popularity of marginalized artists?

LA wildfires damaged art and culture, and had impacts on marginalized artists like Christina Quarles. “Christina Quarles is a queer, mixed contemporary American artist and writer, living and working in Los Angeles, whose gestural, abstract paintings confront themes of racial and sexual identities, gender, and queerness.” - Wikipedia

We can investigate this question by comparing the popularity of Christina Quarles before and after the fires.

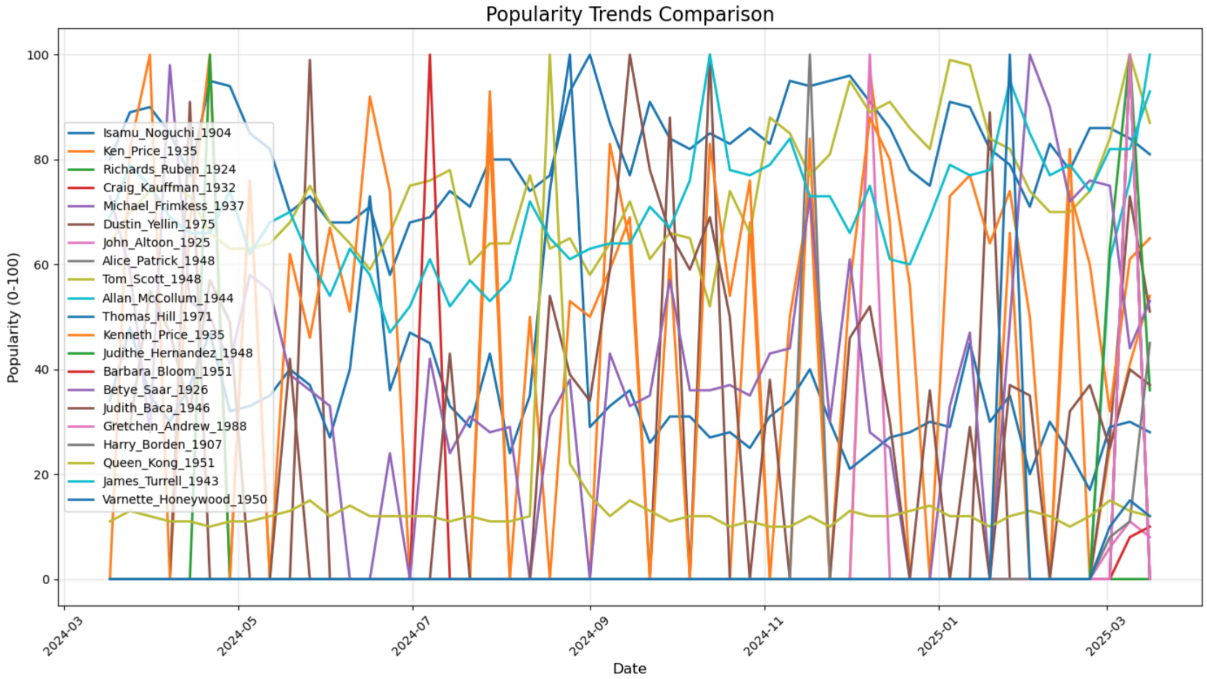


```
=== Statistical Comparison ===
Before January 2025 (n=42):
  Mean: 4.36
  Median: 0.00
  Standard Deviation: 19.81

January 2025 (n=4):
  Mean: 0.00
  Median: 0.00
  Standard Deviation: 0.00

After January 2025 (n=7):
  Mean: 8.14
  Median: 0.00
  Standard Deviation: 10.98

T-test (Before vs After, excluding January):
  t-statistic: -0.7346
  p-value: 0.4750
  Statistically significant difference: No
```

We find that there was no significant difference in popularity before and after the wildfires for this particular artist, but the trend seen starting in march is likely as a result of the wildfires' impact when she opened up about it publicly.

3. Did artists born in LA (postgres) follow a similar popularity trend surrounding the fire?

This question examines whether artists with Los Angeles origins experienced popularity trends similar to The Getty Museum during the wildfire crisis.



We find a significant divergence in public attention patterns. While The Getty Museum experienced heightened visibility during the wildfire crisis, artists born in Los Angeles did not demonstrate comparable surges in popularity or public interest. This distinction suggests fascinating nuances in how the public processes cultural significance during disasters. The geographic connection alone (being from Los Angeles) appears insufficient to generate increased attention for individual artists during a localized crisis. Instead, the findings indicate that public concern may center more specifically on tangible cultural repositories and their physical collections rather than on creative individuals associated with the threatened region.

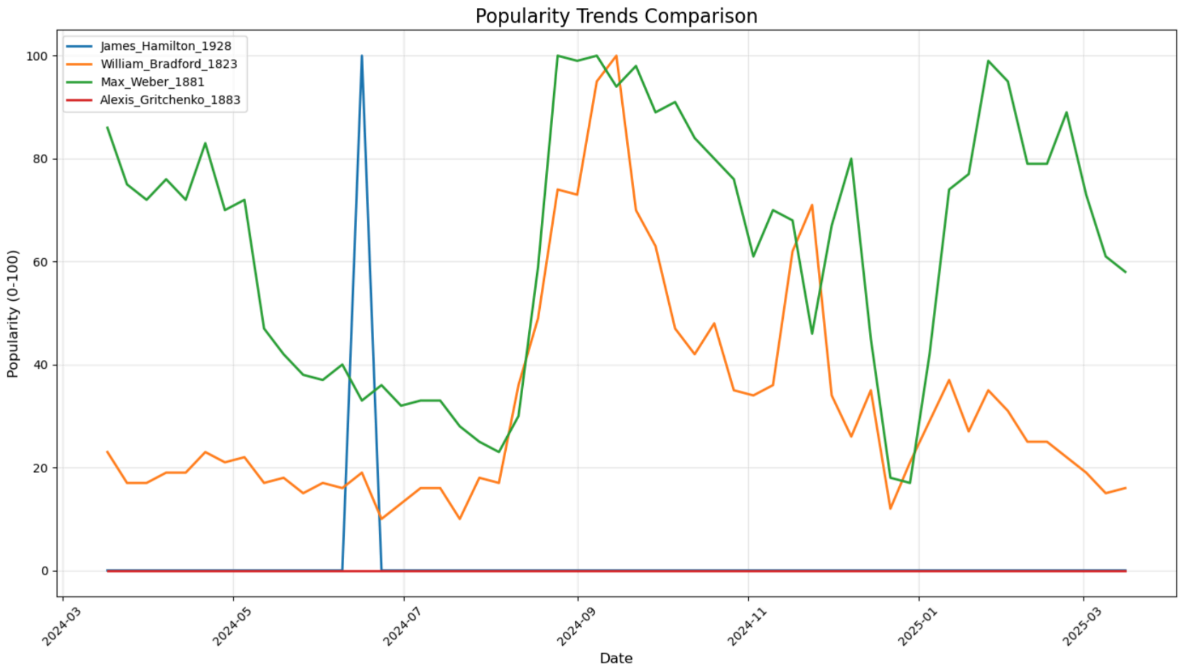
Combined Analyses

We deliberately crafted example questions below that require integration across all database systems to demonstrate the essential nature of our multi-database architecture. Each database—PostgreSQL for structured artist and artwork data, Neo4j for mapping complex relationships between movements and influences, and Redis for real-time trend analysis—serves a unique and irreplaceable function in our art analytics platform. By designing questions that necessitated sequential queries across these systems, we proved that eliminating any single database would render certain analytical pathways impossible. For instance, our movement influence analysis required Neo4j's graph capabilities to identify the most influential art movements, PostgreSQL's relational structure to identify prolific artists within those movements, and Redis's performance capabilities to analyze current popularity trends.

Do the most prolific artists from the two most influential movements trend together?

We investigated whether the most prolific artists from the art world's most influential movements demonstrate correlated popularity trends. Using Neo4j graph database analysis, we identified Romanticism and Expressionism as two of the most influential movements in art history. We excluded Baroque, despite its significant influence, due to insufficient artist representation in our dataset.

We then queried our PostgreSQL database to identify the most prolific artists within each movement, defining prolificity by the number of artworks present in our collection. This analysis revealed James Hamilton and William Bradford as the most productive Romanticists, while Max Weber and Alexis Gritchenko emerged as the leading Expressionists. Our Redis-backed Google Trends analysis revealed fascinating and unexpected relationships.



Contrary to our hypothesis, artists within the same movement did not demonstrate synchronized popularity trends. James Hamilton and William Bradford, despite sharing Romanticist roots, exhibited distinctly different public interest patterns. Similarly, the Expressionists Max Weber and Alexis Gritchenko showed independent trend trajectories.

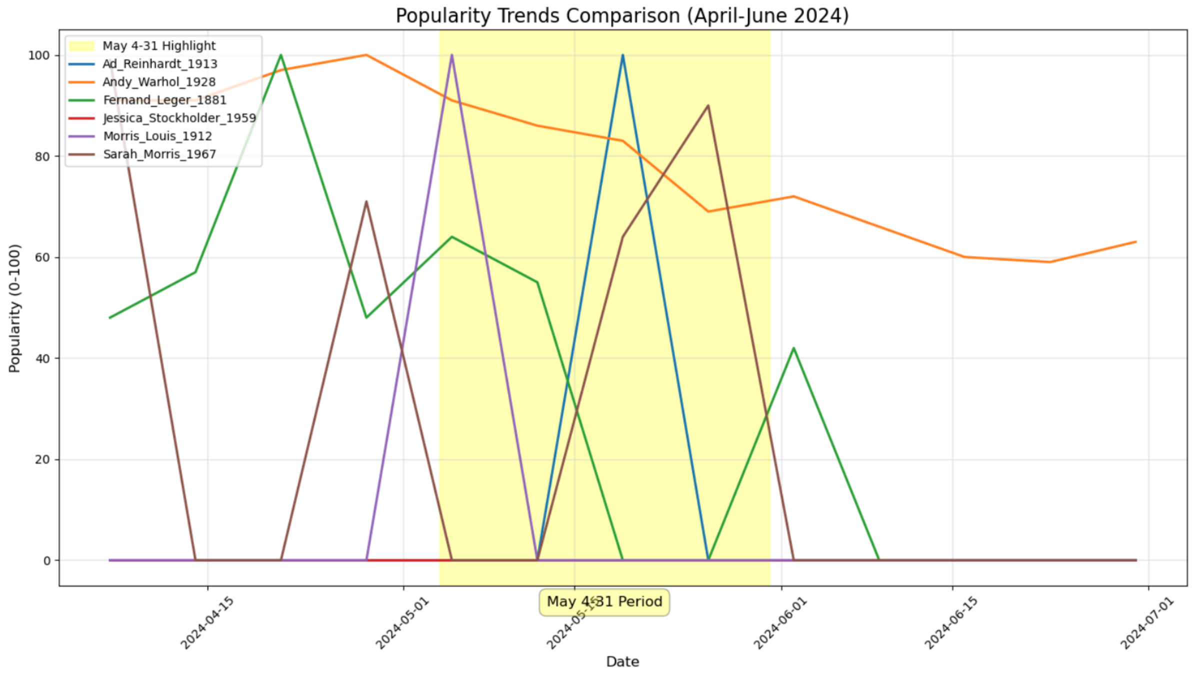
Most surprisingly, our data revealed that William Bradford (Romanticism) and Max Weber (Expressionism) demonstrated comparable trend patterns despite belonging to different artistic movements. This cross-movement correlation suggests that factors beyond artistic school affiliation may drive public interest in artists.

Our conclusion definitively shows that artists' belonging to the same influential movement does not predict similar popularity trends, challenging conventional assumptions about how artistic movements influence public engagement with their practitioners.

Does an artist's death influence their connections' popularity?

We investigated whether an artist's death creates measurable ripple effects across their artistic network, potentially driving increased public interest in connected artists. Our methodology deliberately integrated multiple database systems to capture the full complexity of artistic relationships and temporal popularity patterns. From our PostgreSQL database, we identified recently deceased notable artists, focusing on individuals with sufficient prominence to potentially influence broader artistic discourse. Our Neo4j graph database then mapped these artists' connections, revealing both direct collaborators and those linked through shared artistic movements or educational backgrounds.

This analysis identified Frank Stella, who died on May 4, 2024, at age 87 in New York's West Village, as an ideal case study due to his extensive and well-documented artistic network. We focused our investigation on his connections to Ad Reinhardt, Andy Warhol, Fernand Leger, Jessica Stockholder, Morris Louis, and Sarah Morris. We then utilized the Google Trends API to extract temporal popularity data for these artists, storing these time series in our Redis database for efficient analysis.



Our initial broad-scope annual analysis revealed no immediately apparent pattern across Stella's artistic network. However, when we narrowed our temporal focus to the specific period surrounding Stella's death, distinct popularity spikes emerged. Morris Louis and Fernand Leger demonstrated immediate popularity increases following Stella's passing, while Ad Reinhardt and Sarah Morris showed significant attention spikes in the subsequent month.

These findings provide compelling evidence that an artist's death creates measurable popularity effects within their artistic network, suggesting that mortality events prompt public reconsideration not only of the deceased artist but also of their creative connections and influences.

3 Lessons Learned

Throughout the course of this project, we encountered several challenges related to data cleaning, which provided valuable insights into the complexities of working with real-world datasets, particularly when pulling data from various sources. One of the most significant hurdles we faced was inconsistencies in formatting across different datasets. Each dataset, whether it came from MOMA, WikiData, or PainterPalette, had its own structure and naming conventions, which created discrepancies and threw errors when attempting to merge them. For instance, some columns had different naming conventions for similar attributes, such as artist versus artist_names, and similarly for birth years, or movement names. Resolving these inconsistencies required extensive preprocessing to ensure that the data from each source could be aligned correctly.

Another challenge arose from the presence of null values throughout the datasets. Many attributes, such as birth and death years, nationalities, and artistic movements, were missing or partially incomplete. In these cases, we had to make decisions about how to handle missing data. For instance, we chose to fill missing values with placeholders like "Unknown" where we deemed the columns may be valuable to our analysis and had enough complete rows. When possible, we attempted to fill gaps with values from additional sources likeWikiData. This approach, while effective in creating a usable dataset, still left us with fields, such as location, that we originally wanted to use but decided not to in the end because there were not enough rows with values for the data to make sense contextually.

Additionally, we had to address issues of data duplication, especially when merging large datasets. Artists sometimes appeared under different names across sources, and artworks were occasionally listed multiple times with slight variations in title or artist information. We tackled this by dropping duplicates, normalizing artist names, and ensuring that artworks were accurately attributed to the correct artist.

These challenges highlighted the complexities of real-world data integration. The discrepancies, inconsistencies, and missing values are common in datasets drawn from diverse sources, and cleaning the data became an iterative process of refining and standardizing the data to ensure its quality. This experience reinforced the need for robust data cleaning techniques when working with external sources and emphasized the importance of flexibility and attention to detail in managing and preparing data for analysis. Despite the challenges, the data cleaning process proved invaluable in ensuring that our analysis could proceed effectively and yield meaningful insights.

Another key lesson learned was the complementary strengths of PostgreSQL, Neo4j, and Redis when used together. Each of these tools offered unique capabilities – PostgreSQL served as a reliable relational database for storing structured data and performing complex queries, Neo4j excelled at uncovering relationships between artists, movements, styles, and artworks, allowing for deeper insights into the artistic ecosystem, and Redis played a crucial role in optimizing performance by drastically reducing query response times for frequently accessed data. The combined strengths of these technologies allowed us to perform a more comprehensive analysis.

Finally, the analysis reinforced the idea that art is constantly evolving. While new styles emerge, some styles persist for extended periods, suggesting that certain aesthetic movements have a lasting cultural impact. The relationships between art movements, their underlying philosophies, and historical events also revealed a dynamic interplay between art and societal developments. Our findings indicate that technological advancements and contemporary events continue to shape artistic expression, just as previous historical moments have influenced the art world in the past. This evolving nature of art presents both challenges and opportunities for artists and curators alike, suggesting that art's trajectory will remain ever-changing, yet deeply connected to the forces of its time.

4 Future Work

In future work, we would expand the scope of this project by incorporating additional data sources to address the gaps in the current dataset. Gathering more comprehensive location data, for example, could provide a deeper understanding of the geographical influences on artists and their works. This could involve integrating location-based information from regional art institutions, museums, and archives.

We would also explore the integration of auction price data from reputable sources like Sotheby's or Christie's. By acquiring data on the prices of artworks sold at auctions, we could investigate how the prices of different art pieces correlate with their styles, movements, and artists. This would allow us to analyze trends in the valuation of artworks over time.

Additionally, we would leverage machine learning models to predict future artwork prices or artist popularity based on historical data and current trends in the art world to forecast how specific artists or art styles might evolve in terms of market value and public interest.

Lastly, we would develop an interactive, web-based interface to allow users to explore the data in a more accessible and engaging manner. Users would be able to query the database, visualize trends, and gain deeper insights into the relationships between artists, movements, artworks, and more. Findings from this project would be more widely accessible to historians, researchers, and the general public, fostering greater interaction with the data and encouraging more extensive exploration of the art world.