Graph Databases

Using Neo4j and Amazon SNAP Dataset

Database Management Systems Laboratory (CS39202)

Team CheeseCake

Shivam Choudhury

Ayush Mundada

Tanishka Rahate

Omkar Bhandare

Jadhav Udaykiran

Problem Statement

The objective of this project is to process large graphs using a graph database. The core tasks include:

i. Install any graph processing system, such as ApacheGraph, Pregel (GoldenOrb), Giraph, Neo4j, or Stanford GPS.

ii. Load a large graph from the Stanford SNAP large graph repository.

iii. Provide an interface to run simple graph queries. Bonus marks may be awarded for implementing PageRank.

iv. Profile and evaluate the performance of the system.

Background

Graph databases represent a powerful paradigm in database technology that focuses on relationships between data entities. Graph databases are designed to efficiently store, process, and query highly connected data using graph structures with nodes, edges, and properties.

They consist of three fundamental elements:

- **Nodes**: Represent entities or instances such as products, categories, or other items. They function similar to records, like in relational databases.
- **Edges**: Connect nodes, representing the relationships between entities. They can either be directed or undirected, and they often carry properties that describe the nature of the relationship.
- **Properties**: Information associated with nodes and edges that describe their characteristics and attributes.

At its core, a graph database portrays data as it exists conceptually in the real world. For example, in a social network like Twitter, each user would be represented as a node, with “follows” relationships connecting them. This intuitive representation makes graph databases particularly effective for modelling complex, interconnected data.

They are built on graph theory principles but do not require users to understand the mathematical complexities. Instead, they offer an intuitive way to model data that closely resembles how we think about relationships.

They typically use specialised query languages designed for traversing relationships.

Cypher, developed for Neo4j, is one of the most popular graph query languages, allowing users to express complex relationship patterns in a relatively simple syntax.

Graph databases are well-suited for e-commerce applications like the Amazon product recommendation system, which is the one attempted here. Because of their natural affinity for relationship-based data, graph databases are widely adopted across various industries. In modern applications, they play a pivotal role where understanding context and connection is essential - such as recommendation engines, fraud detection systems, and social media analytics. They enable systems to go beyond flat data queries and derive insights through patterns, proximity, and relevance. Their schema flexibility also allows easy evolution with dynamic datasets, a necessity in fast-paced domains like e-commerce and finance. As a result, graph databases are increasingly becoming a foundational choice for building intelligent, adaptive, and scalable data-driven systems.

Objective

The primary objective of this project is to design and implement a robust graph-based product recommendation system leveraging the Amazon SNAP dataset and Neo4j as the core database technology. The system aims to enhance user experience through intuitive navigation, advanced search capabilities, and intelligent, data-driven recommendations by capitalising on the inherent interconnectedness within product data. This project positions graph databases as a powerful alternative to traditional recommendation systems by exploiting relationships, patterns, and influence within product networks.

To achieve this goal, the project focuses on the following key objectives:

1. **Efficient Product Discovery via Hierarchical Structures**: Implement mechanisms for seamless product discovery by utilising categorical and group-based relationships inherent in the dataset.

2. **Customised PageRank Algorithm for Product Relevance**: Develop and integrate a tailored PageRank-inspired algorithm to rank products based on relevance, popularity, and contextual importance within the graph.

3. **Advanced Multi-Category Search Functionality**: Enable flexible and precise search across multiple product categories simultaneously, allowing for complex and user-specific query resolution.

4. **Performance Monitoring Framework**: Establish a comprehensive performance evaluation and monitoring framework to ensure system scalability, reliability, and responsiveness under varying loads.

5. **User Influence Scoring Mechanism**: Introduce a user influence scoring model that quantifies the impact of user behaviour on product visibility and recommendation outcomes.

6. **Temporal Analysis of Product Trends**: Analyse temporal dynamics in product popularity and user interaction to uncover evolving trends and seasonality within the dataset.

7. **Visualisation of Product Networks**: Design interactive visualisations to represent the structure and relationships among products, categories, and user interactions within the graph.

8. **Integration of a RAG-Based Natural Language Query System**: Incorporate a Retrieval-Augmented Generation (RAG) based natural language processing interface to facilitate intuitive and conversational product discovery.

Methodology

The development of the graph-based product recommendation system follows a structured and multi-faceted approach, encompassing key stages such as data modelling, algorithm design, system implementation, and user interface development. Each phase is carefully designed to align with the overall objective of leveraging graph structures for enhanced product discovery and recommendation. The methodology ensures a coherent integration of data engineering, graph analytics, and user-centric design to deliver a scalable and effective recommendation platform.

Data Schema Design

The underlying data schema is designed to effectively represent the complex relationships among products, consumers, categories, and groups within the Amazon SNAP dataset. This schema serves as the structural backbone of the recommendation system, enabling efficient querying, relationship traversal, and algorithmic computations.

Node Types (along with the attributes)

- **Product Node**: Id, ASIN, title, salesrank, avg_rating, intrinsic_score, total_score
- **Group Node**: name
- **Category Node**: name

- **Consumer Node**: Id, Customer, name, score

Edge Types

- **BELONGS_TO:** Product Node → Category Node
- **PART_OF:** Product Node → Group Node
- **REVIEWED:** Consumer Node → Product Node
  Attributes: Date, Rating, Votes, Helpful
- **SIMILAR:** Product Node → Product Node
- **COPURCHASED_WITH:** Product Node → Product Node
  Attributes: Frequency

Indexing Strategy

Given the dataset's variety and volume of queries, implementing appropriate indexes is critical for ensuring efficient data access and optimal performance. The following indexes were established, each tailored to anticipated query patterns:

Name	Type	Entity Type	Labels or Types	Properties
`category_name_index`	RANGE	NODE	{Category}	{name}
`consumer_customer_index`	RANGE	NODE	{Consumer}	{Customer}
`group_name_index`	RANGE	NODE	{Group}	{name}
`product_asin_index`	RANGE	NODE	{Product}	{ASIN}
`product_id_index`	RANGE	NODE	{Product}	{Id}

Table 1: Indexing Strategy

Custom PageRank Algorithm

To accurately reflect the dynamics of product relevance and interconnectedness within the Amazon SNAP dataset, we designed and implemented a novel two-phase PageRank algorithm tailored specifically for graph-based product recommendation. Unlike traditional PageRank—which primarily relies on nodes' structural connectivity—our approach integrates product-specific attributes and co-purchase behaviour to derive a more context-aware and impactful relevance score.

Phase 1: Computing the Intrinsic Score

$$\text{intrinsic\_score} = \left(2^{\text{avg\_rating}}\right) \cdot \left(\frac{\log_{10}(1 + \text{sum of all helpful})}{\log_{10}(2)}\right) \tag{1}$$

This formulation provides a meaningful balance: exponential scaling of the rating gives prominence to top-rated products. In contrast, logarithmic scaling of helpful votes accounts for user trust without allowing it to dominate the score.

Phase 2: Incorporating Network Influence via Co-Purchase Relationships

$$\text{total\_score} = \text{intrinsic\_score} + 0.5 \cdot \sum \left(\frac{\text{frequency}_j}{\text{total\_frequency}} \cdot \text{total\_score}_j \right) \tag{2}$$

Note: Only co-purchased products with frequency ≥ 2 are considered in the summation.

Cypher Implementation

Phase 1: Intrinsic Score Calculation

```
MATCH (p:Product)
OPTIONAL MATCH (u:Consumer)-[r:REVIEWED]->(p)
WITH p, AVG(r.Rating) AS avg_rating, SUM(r.Helpful) AS total_helpful
SET p.intrinsic_score = round(
    CASE
        WHEN avg_rating IS NULL THEN 0.0
        ELSE (2.0 ^ avg_rating) * (LOG(1.0 + COALESCE(total_helpful, 0)) /
   LOG(2))
    END
, 2)
RETURN count(p) AS updated_count
```

Phase 2: Incorporating Co-Purchase Influence

```
MATCH (p:Product)
WITH p, p.intrinsic_score AS intrinsic_score
OPTIONAL MATCH (p)-[co:COPURCHASED_WITH]->(other:Product)
 WHERE co.Frequency >= 2
WITH p, intrinsic_score,
    collect({
        product: other,
        frequency: co.Frequency,
        score: other.total_score
    }) AS co_purchases
WITH p, intrinsic_score, co_purchases,
    REDUCE(total = 0, item IN co_purchases | total + item.frequency) AS
   total_frequency
WITH p, intrinsic_score,
    CASE WHEN total_frequency > 0 THEN
        REDUCE(sum = 0, item IN co_purchases |
            sum + (item.frequency / toFloat(total_frequency)) * item.score
        )
    ELSE 0 END AS co_purchase_score
SET p.total_score = round(intrinsic_score + co_purchase_score * 0.5, 2)
RETURN count(p) AS updated_count
```

Only co-purchase edges with frequency ≥ 2 are considered, effectively filtering out noise and incidental relationships. The final `total_score` represents a balanced combination of intrinsic product quality and network influence, with a damping factor (0.5) applied to avoid disproportionate influence from the co-purchase network.

User Influence Scoring

To assess the influence of individual reviewers within the system, we developed a custom algorithm that calculates a user-specific influence score based on the helpfulness of their reviews. This approach prioritises users who consistently contribute meaningful and helpful reviews across various products while mitigating the potential impact of outliers.

Formula

The user influence score is calculated as follows:

$$\text{score} = \sum \left(\min \left(\frac{\log_{10}(1 + \text{helpful})}{\log_{10}(2)},\ 5.0 \right) \right) \tag{3}$$

Cypher Implementation

```
MATCH (c:Consumer)-[r:REVIEWED]->()
WHERE c.Customer IS NOT NULL AND r.Helpful IS NOT NULL
RETURN DISTINCT c
MATCH (c)-[r:REVIEWED]->()
WHERE r.Helpful IS NOT NULL
WITH c, collect(r.Helpful) AS helpful_values
WITH c, [val IN helpful_values |
    CASE
    WHEN val <= 0 THEN 0.0
    ELSE
        CASE
        WHEN log10(1 + val) / log10(2) > 5.0 THEN 5.0
        ELSE toFloat(log10(1 + val) / log10(2))
        END
    END
] AS logs
WITH c, reduce(score = 0.0, x IN logs | score + x) AS final_score
SET c.score = final_score
```

For each consumer, the algorithm computes a base score for each review using logarithmic scaling, caps this score at 5.0 to prevent exceptionally high helpfulness values from disproportionately affecting the overall influence score, and sums up these capped scores across all reviews to derive the final influence score. This method ensures that users who consistently contribute helpful reviews across different products are rewarded, while logarithmic scaling prevents a few highly helpful reviews from dominating the score.

Trending Product Analysis

To track and identify trending products, we developed a functionality that focuses on identifying products that have seen a notable surge in popularity within a specified time frame. The algorithm performs a time-based analysis of review activity for each product. This approach considers the number of reviews as the primary ranking attribute. It combines it with the product's total score, derived from the custom PageRank algorithm, as the secondary ranking attribute. This dual-ranking system prioritises products with a high volume of recent reviews and strong recommendation scores. By incorporating the quantity and quality of reviews, this analysis provides a robust measure of product popularity, making it highly effective for identifying products trending due to recent customer engagement.

Formula

$$\text{TrendingScore}(p) = \text{sort}\Big(\text{review\_count}_{[t_1,t_2]}(p),\ \text{total\_score}(p)\Big) \tag{4}$$

where $[t_1, t_2]$ represents the time window within which reviews are counted. Products are sorted first by review count (descending), and then by total score (descending) in case of ties.

Cypher Implementation

```
MATCH (c:Consumer)-[r:REVIEWED]->(p:Product)
WHERE date(r.Date) >= date($start_date) AND date(r.Date) <= date($end_date)
WITH p, count(r) AS review_count, p.total_score AS total_score
ORDER BY review_count DESC, total_score DESC
LIMIT 10
RETURN p.title AS product_title, p.ASIN AS product_asin, review_count,
   total_score
```

Multi-Category Search

To empower users with more refined product discovery, we implemented a Multi-Category Search feature that allows seamless filtering across multiple product categories. This functionality is handy when users are interested in items falling under overlapping categories. To make category selection intuitive, especially when users may not know a category's exact spelling or phrasing, we integrated fuzzy matching using the TheFuzz library. Users can type approximate or partial terms and retrieve relevant category names.

Cypher Implementation

```
MATCH (p:Product)
WHERE EXISTS {
  MATCH (p)-[:BELONGS_TO]->(:Category {name: $categories[0]})
}
AND ALL(category IN $categories[1..] WHERE
    EXISTS {
        MATCH (p)-[:BELONGS_TO]->(:Category {name: category})
    })
AND p.avg_rating >= $min_avg_rating
RETURN p.title AS title, p.ASIN AS asin, p.avg_rating AS rating,
     p.total_score AS total_score
ORDER BY p.total_score DESC
SKIP $skip
LIMIT $limit
```

RAG-Based Chatbot Implementation

We developed a Retrieve and Generate (RAG)-based chatbot to enhance natural language product discovery, leveraging advanced embedding techniques and efficient similarity search mechanisms. The core of this implementation utilises the 'BAAI/beg-small-en-v1.5' model to generate embeddings for product titles, groups, and categories. These embeddings are then indexed using FAISS (Facebook AI Similarity Search), accelerating the similarity search process and ensuring efficient retrieval of relevant products in response to user queries. This approach effectively translates user queries into embeddings, performs a similarity search across a range of products, categories, and groups, and returns the top matches ranked by similarity. A similarity threshold of 0.3 ensures that only highly relevant results are returned, providing users with accurate and meaningful recommendations.

The RAG engine integrates multiple components to process user queries efficiently:

- **Text Cleaning and Preprocessing:** User input is first cleaned using the `cleantext` library, removing noise such as URLs, emails, and unnecessary punctuation. A stop-word filtering process is also applied to ensure that only the most relevant terms remain in the query.
- **Embedding-Based Search:** After cleaning the query, it is passed through an embedding engine to obtain a vector representation. The system then searches for similar products, groups, and categories and ranks them based on their cosine similarity to the query vector.
- **Results Filtering and Response Generation:** The top matching products, categories, and groups are retrieved and enriched with additional product details (e.g., ASIN, title, sales rank). A well-structured response is generated to ensure an engaging user experience, highlighting the most relevant results.

Performance Monitoring Framework

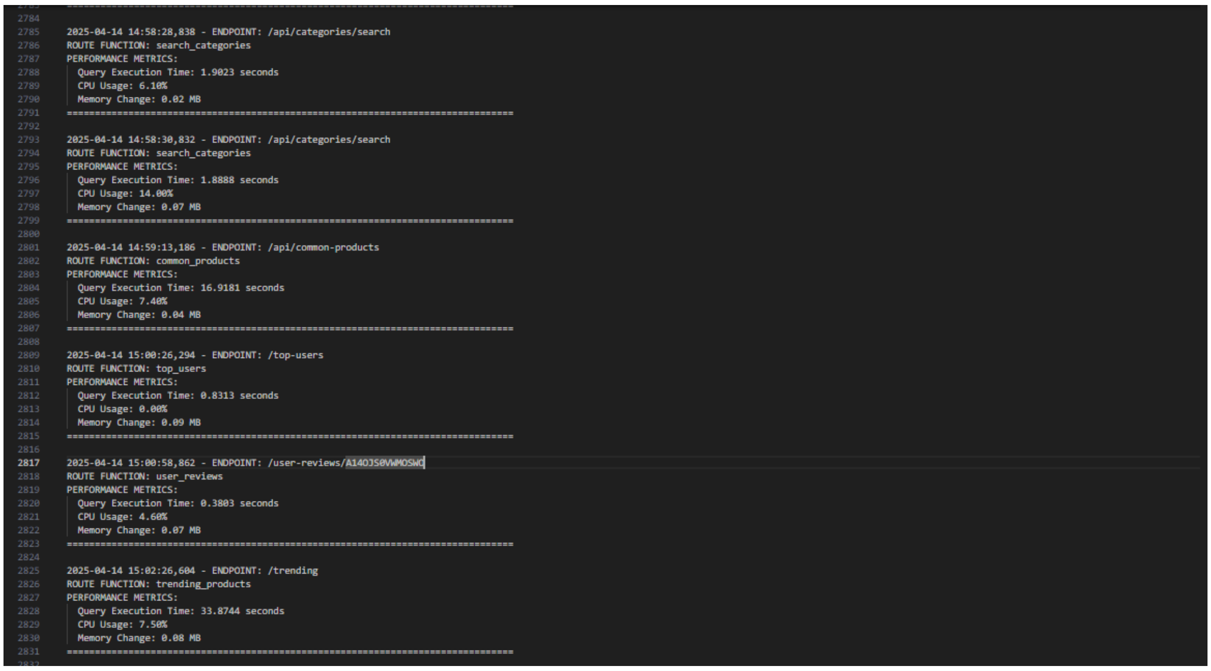
We have integrated a comprehensive performance monitoring framework to ensure optimal system efficiency and transparency across our application. This framework provides consistent and real-time tracking of key system metrics for every route and query execution.

Key Metrics Tracked

- **Query Execution Time (in seconds):** Measures the duration to execute individual route handlers and database interactions.
- **CPU Usage (percentage):** Captures the CPU consumption during processing.
- **Memory Consumption (in megabytes):** Tracks the memory footprint before and after route execution to detect potential memory leaks or spikes.

This monitoring system was implemented using a custom decorator, `@log_performance`, which wraps all relevant route handler functions. This approach ensures minimal intrusion into the application logic while enabling uniform performance tracking across the board.

Each time a route is executed, the decorator logs the stats:



```
================================================================================

2025-04-14 14:58:28,838 - ENDPOINT: /api/categories/search
ROUTE FUNCTION: search_categories
PERFORMANCE METRICS:
  Query Execution Time: 1.9023 seconds
  CPU Usage: 6.10%
  Memory Change: 0.02 MB
================================================================================

2025-04-14 14:58:30,832 - ENDPOINT: /api/categories/search
ROUTE FUNCTION: search_categories
PERFORMANCE METRICS:
  Query Execution Time: 1.8888 seconds
  CPU Usage: 14.00%
  Memory Change: 0.07 MB
================================================================================

2025-04-14 14:59:13,186 - ENDPOINT: /api/common-products
ROUTE FUNCTION: common_products
PERFORMANCE METRICS:
  Query Execution Time: 16.9181 seconds
  CPU Usage: 7.40%
  Memory Change: 0.04 MB
================================================================================

2025-04-14 15:00:26,294 - ENDPOINT: /top-users
ROUTE FUNCTION: top_users
PERFORMANCE METRICS:
  Query Execution Time: 0.8313 seconds
  CPU Usage: 0.00%
  Memory Change: 0.09 MB
================================================================================

2025-04-14 15:00:58,862 - ENDPOINT: /user-reviews/A14OJS0VWMOSWO
ROUTE FUNCTION: user_reviews
PERFORMANCE METRICS:
  Query Execution Time: 0.3803 seconds
  CPU Usage: 4.60%
  Memory Change: 0.07 MB
================================================================================

2025-04-14 15:02:26,604 - ENDPOINT: /trending
ROUTE FUNCTION: trending_products
PERFORMANCE METRICS:
  Query Execution Time: 33.8744 seconds
  CPU Usage: 7.50%
  Memory Change: 0.08 MB
================================================================================
```

Figure 1: Performance Log Entries

User Interface Components

The user interface has been thoughtfully designed to provide an intuitive, flexible, and engaging browsing experience across multiple dataset dimensions. Below is a detailed overview of the key components:

- **Home Page:** The landing page presents all available Groups in the dataset. As groups encapsulate broader themes or collections, this is a natural entry point for product exploration.

- **Products by Group:** Displays all products under a selected Group, sorted by default using the `total_score` in descending order to first surface the most relevant or high-performing products. Users can dynamically sort the product list by either `total_score`, `title`, `avg_rating`, or `salesrank`. Additional filters, such as minimum average rating, are also available to refine results.
- **Products by Categories:** This view displays all products belonging to a selected Category, again sorted by default using `total_score` in descending order. It shares the same interactive capabilities as above.
- **Products by Multiple Categories:** This page enables the discovery of products that belong simultaneously to multiple categories. It leverages a fuzzy-matching powered search bar and returns products that match all selected categories. Sorting and filtering options identical to other product views.
- **Product Detail Page:** Provides a comprehensive view of a product, including core product details, associated categories, co-purchased and similar products, and a complete list of user reviews with metadata.
- **Top Reviewers Page:** Ranks the top reviewers based on a custom reviewer scoring algorithm. This helps surface influential voices within the dataset and provides users with a trusted source for quality opinions.
- **Trending Products Page:** Displays products identified as trending using our internal scoring and trend-detection logic. These may be determined through review activity, popularity spikes, or recent high ratings.
- **User Reviews Page:** Showcases the most helpful reviews written by a selected top reviewer, ranked by the helpful count metric. This promotes valuable user-generated insights and supports more informed decision-making.

Query Complexity Analysis

Let the number of nodes and edges of defined types be as follows:

Node Types:	Category = N, Consumer = M, Group = G, Product = P
Edge Types:	BELONGS_TO = B, COPURCHASED_WITH = C, PART_OF = X, REVIEWED = R, SIMILAR = S

Finding all Groups (sorted): $O(G)$

Products in the same Group (sorted): $O(1 + X + P \cdot \log P)$

Products in the same Category (sorted): $O(1 + B + P \cdot \log P)$

Products in multiple selected Categories: $O(N \cdot P + P \cdot \log P)$

Trending Products: $O(R + P \cdot \log P)$

Top Reviewers (with score calculation): $O(M \cdot \log M + R + M)$

PageRank score calculation: $O(P + R + P + C) = O(P + R + C)$

Reviews of a selected top Reviewer: $O(R + R \cdot \log R)$

Selected Product data (all the linked info): $O(1+N+S+S\cdot \log S+C+C\cdot \log C+ R + R \cdot \log R)$

Screenshots of the Demonstration

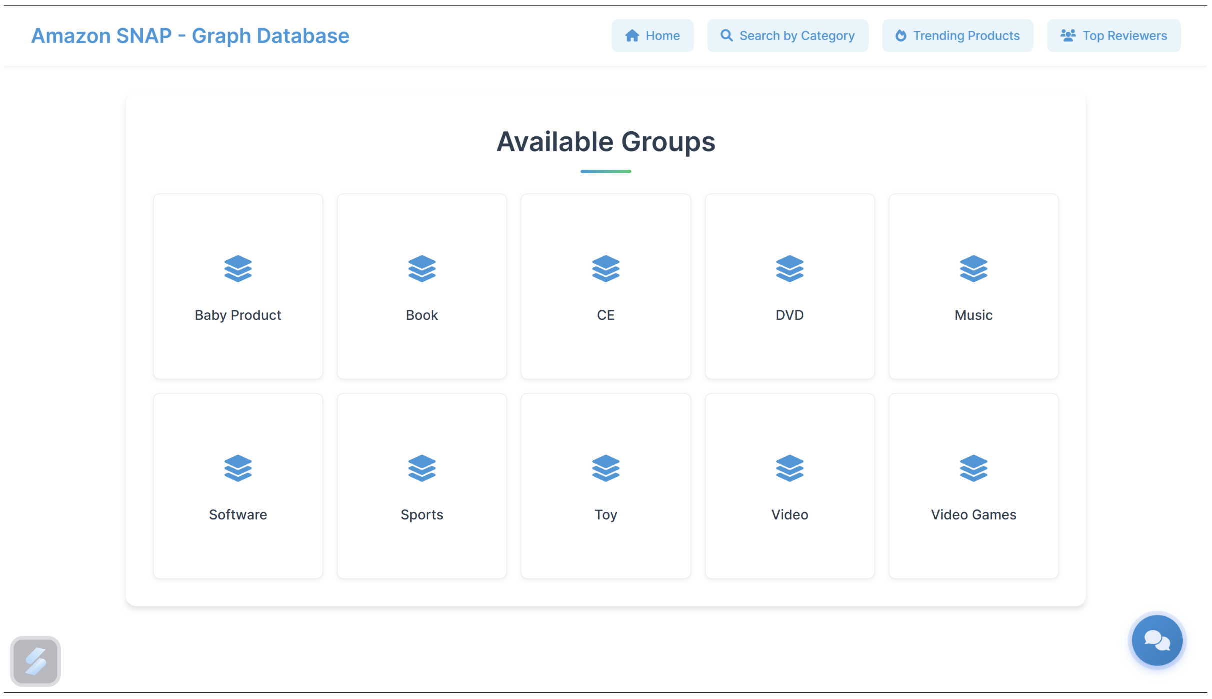
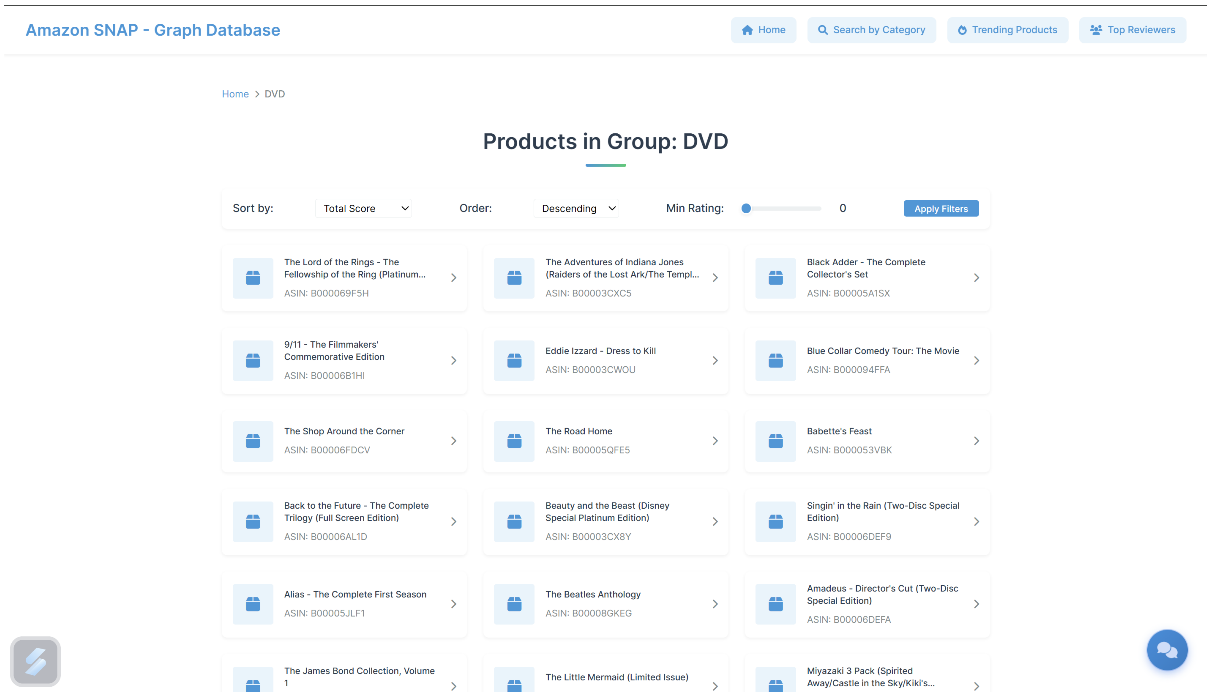


Figure 2: Home Page


Amazon SNAP - Graph Database

Home | Search by Category | Trending Products | Top Reviewers

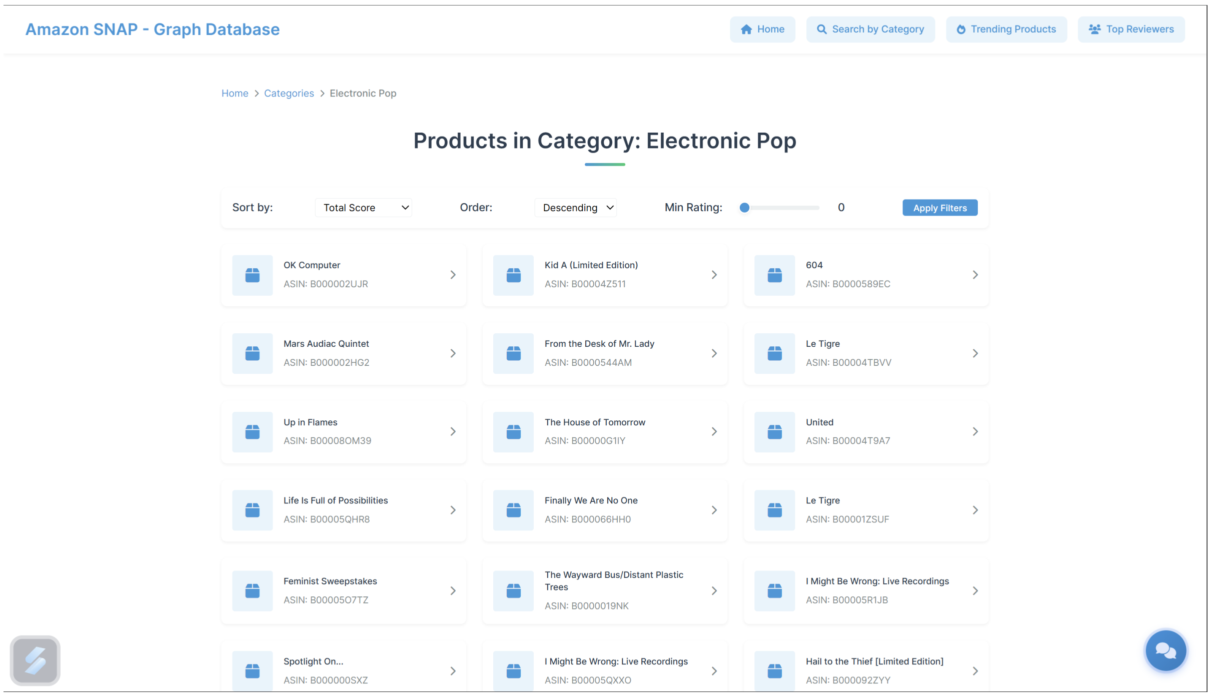
Home > DVD

Products in Group: DVD

Sort by: Total Score | Order: Descending | Min Rating: 0 | Apply Filters

The Lord of the Rings - The Fellowship of the Ring (Platinum... ASIN: B000069F5H	The Adventures of Indiana Jones (Raiders of the Lost Ark/The Templ... ASIN: B00003CXC5	Black Adder - The Complete Collector's Set ASIN: B00005A1SX
9/11 - The Filmmakers' Commemorative Edition ASIN: B00006B1HI	Eddie Izzard - Dress to Kill ASIN: B00003CWOU	Blue Collar Comedy Tour: The Movie ASIN: B000094FFA
The Shop Around the Corner ASIN: B00006FDCV	The Road Home ASIN: B00005QFE5	Babette's Feast ASIN: B000053VBK
Back to the Future - The Complete Trilogy (Full Screen Edition) ASIN: B00006AL1D	Beauty and the Beast (Disney Special Platinum Edition) ASIN: B00003CX8Y	Singin' in the Rain (Two-Disc Special Edition) ASIN: B00006DEF9
Alias - The Complete First Season ASIN: B00005JLF1	The Beatles Anthology ASIN: B00008GKEG	Amadeus - Director's Cut (Two-Disc Special Edition) ASIN: B00006DEFA
The James Bond Collection, Volume 1	The Little Mermaid (Limited Issue)	Miyazaki 3 Pack (Spirited Away/Castle in the Sky/Kiki's...

Figure 3: Products in selected Group


Amazon SNAP - Graph Database

Home | Search by Category | Trending Products | Top Reviewers

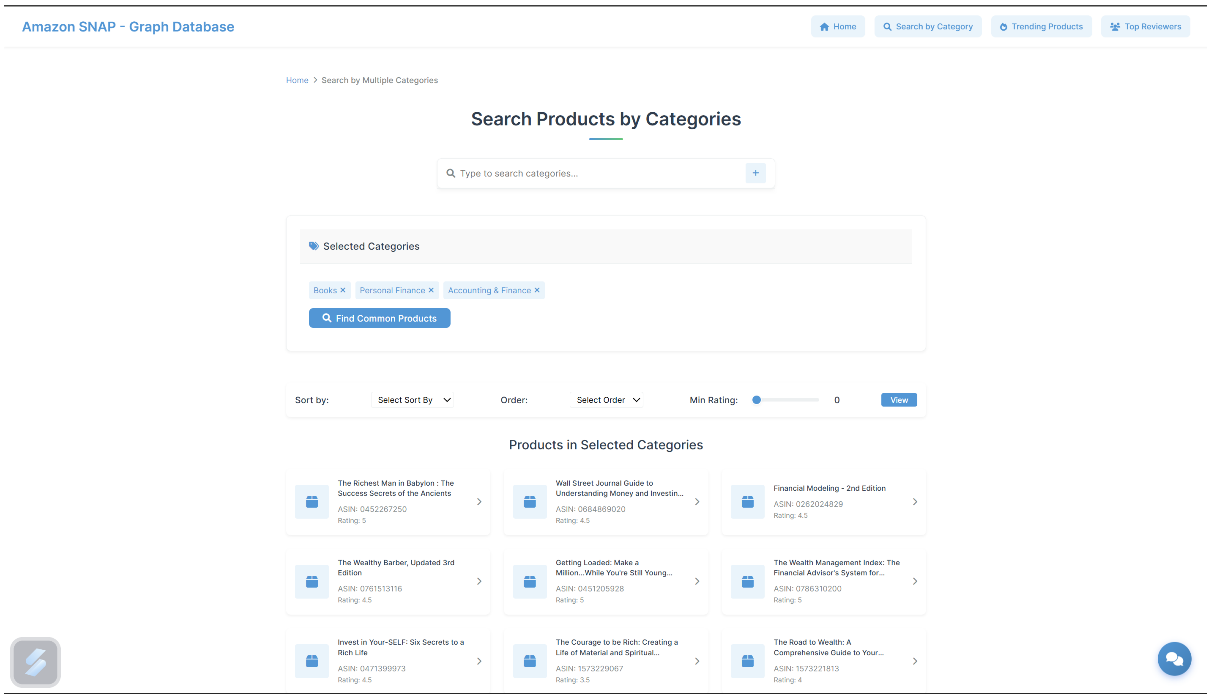
Home > Categories > Electronic Pop

Products in Category: Electronic Pop

Sort by: Total Score | Order: Descending | Min Rating: 0 | Apply Filters

OK Computer ASIN: B000002UJR	Kid A (Limited Edition) ASIN: B00004Z511	604 ASIN: B0000589EC
Mars Audiac Quintet ASIN: B000002HG2	From the Desk of Mr. Lady ASIN: B0000544AM	Le Tigre ASIN: B00004TBVV
Up in Flames ASIN: B00008OM39	The House of Tomorrow ASIN: B00000G1IY	United ASIN: B00004T9A7
Life Is Full of Possibilities ASIN: B00005QHR8	Finally We Are No One ASIN: B000066HH0	Le Tigre ASIN: B00001ZSUF
Feminist Sweepstakes ASIN: B00005O7TZ	The Wayward Bus/Distant Plastic Trees ASIN: B0000019NK	I Might Be Wrong: Live Recordings ASIN: B00005R1JB
Spotlight On... ASIN: B000000SXZ	I Might Be Wrong: Live Recordings ASIN: B00005QXXO	Hail to the Thief [Limited Edition] ASIN: B000092ZYY

Figure 4: Products in selected Category


Amazon SNAP - Graph Database

Home | Search by Category | Trending Products | Top Reviewers

Home > Search by Multiple Categories

Search Products by Categories

Type to search categories...

Selected Categories

Books × | Personal Finance × | Accounting & Finance ×

Find Common Products

Sort by: Select Sort By | Order: Select Order | Min Rating: 0 | View

Products in Selected Categories

The Richest Man in Babylon : The Success Secrets of the Ancients
ASIN: 0452267250
Rating: 5

Wall Street Journal Guide to Understanding Money and Investin...
ASIN: 0684869020
Rating: 4.5

Financial Modeling - 2nd Edition
ASIN: 0262024829
Rating: 4.5

The Wealthy Barber, Updated 3rd Edition
ASIN: 0761513116
Rating: 4.5

Getting Loaded: Make a Million....While You're Still Young...
ASIN: 0451205928
Rating: 5

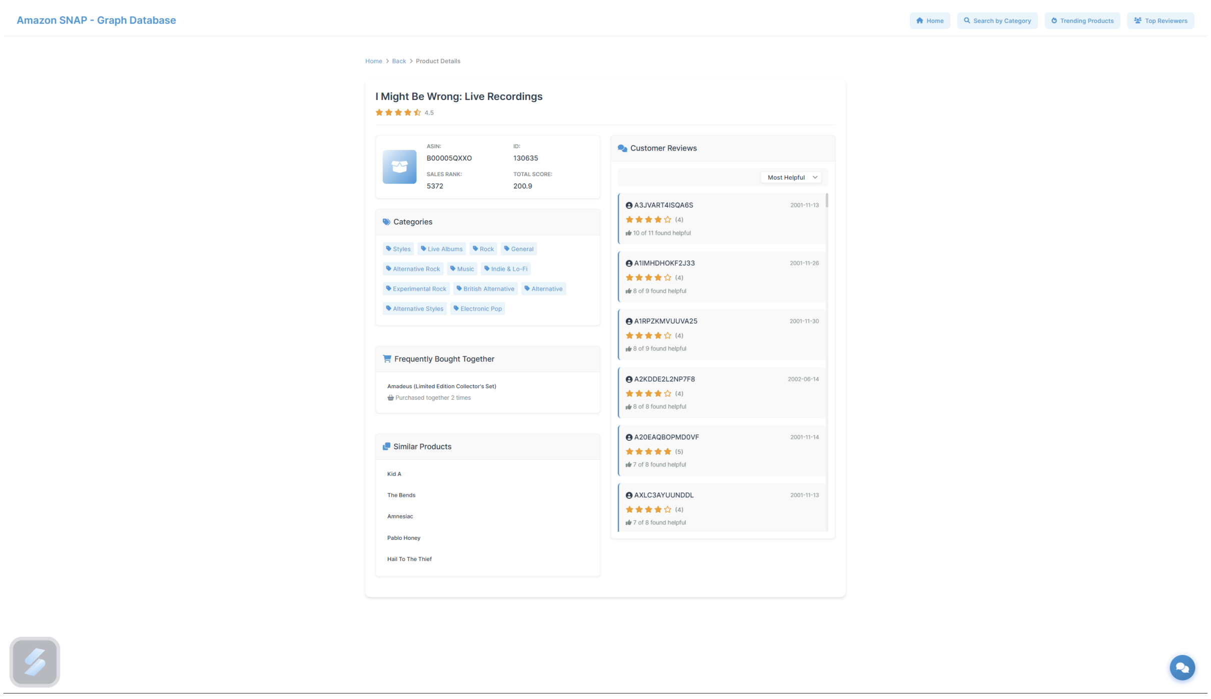
The Wealth Management Index: The Financial Advisor's System for...
ASIN: 0786310200
Rating: 5

Invest in Your-SELF: Six Secrets to a Rich Life
ASIN: 0471399973
Rating: 4.5

The Courage to be Rich: Creating a Life of Material and Spiritual...
ASIN: 1573229067
Rating: 3.5

The Road to Wealth: A Comprehensive Guide to Your...
ASIN: 1573221813
Rating: 4

Figure 5: Products in selected Categories


Amazon SNAP - Graph Database

Home | Search by Category | Trending Products | Top Reviewers

Home > Back > Product Details

I Might Be Wrong: Live Recordings
4.5

ASIN: B00005QXXO
ID: 130635
SALES RANK: 5372
TOTAL SCORE: 200.9

Categories

Styles | Live Albums | Rock | General | Alternative Rock | Music | Indie & Lo-Fi | Experimental Rock | British Alternative | Alternative | Alternative Styles | Electronic Pop

Frequently Bought Together

Amadeus (Limited Edition Collector's Set)
Purchased together 2 times

Similar Products

Kid A
The Bends
Amnesiac
Pablo Honey
Hail To The Thief

Customer Reviews

Most Helpful

A3JVART4ISQA6S 2001-11-13
(4)
10 of 11 found helpful

A1IMHDHOKF2J33 2001-11-26
(4)
8 of 9 found helpful

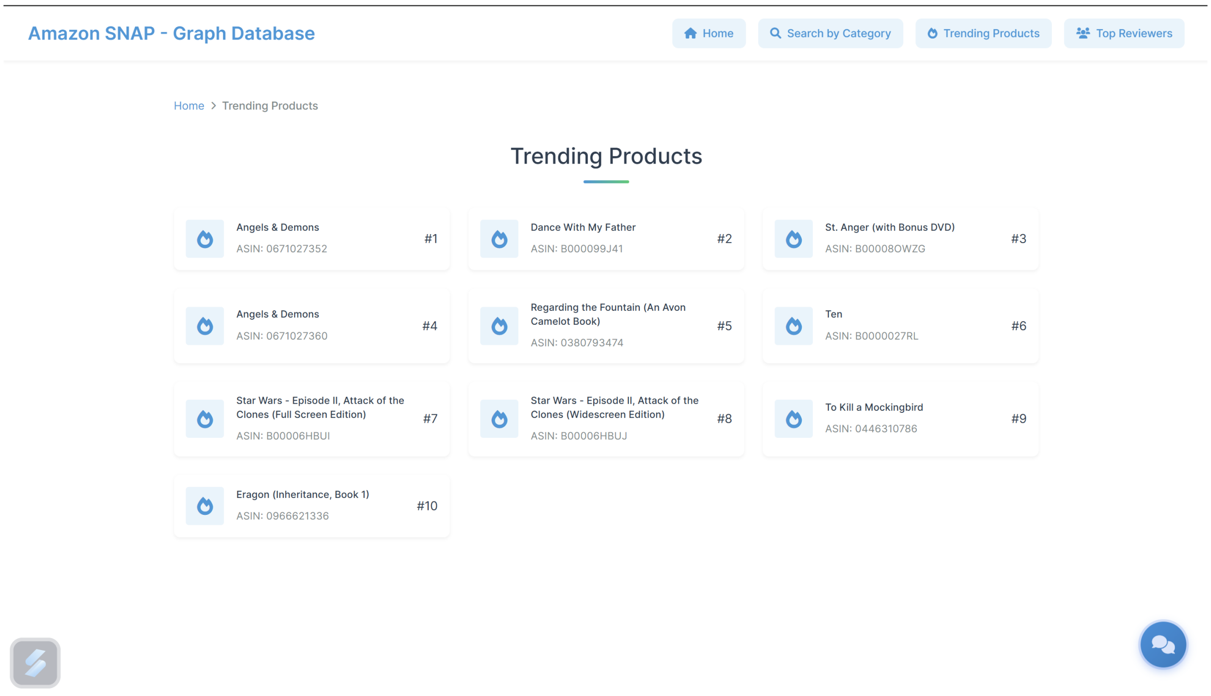
A1RPZKMVUUVA25 2001-11-30
(4)
8 of 9 found helpful

A2KDDE2L2NP7F8 2002-06-14
(4)
8 of 8 found helpful

A20EAQBOPMD0VF 2001-11-14
(5)
7 of 8 found helpful

AXLC3AYUUNDDL 2001-11-13
(4)
7 of 8 found helpful

Figure 6: Product Details



Trending Products

Angels & Demons
ASIN: 0671027352
#1

Dance With My Father
ASIN: B000099J41
#2

St. Anger (with Bonus DVD)
ASIN: B00008OWZG
#3

Angels & Demons
ASIN: 0671027360
#4

Regarding the Fountain (An Avon Camelot Book)
ASIN: 0380793474
#5

Ten
ASIN: B0000027RL
#6

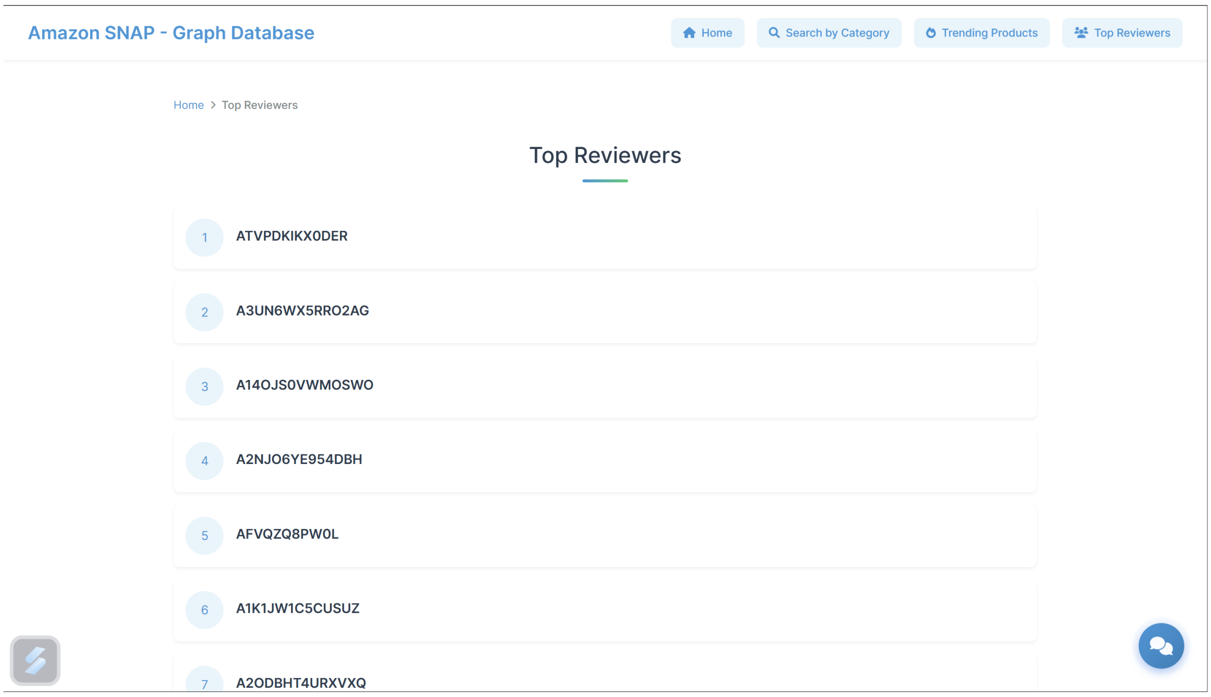
Star Wars - Episode II, Attack of the Clones (Full Screen Edition)
ASIN: B00006HBUI
#7

Star Wars - Episode II, Attack of the Clones (Widescreen Edition)
ASIN: B00006HBUJ
#8

To Kill a Mockingbird
ASIN: 0446310786
#9

Eragon (Inheritance, Book 1)
ASIN: 0966621336
#10

Figure 7: Trending Products



Top Reviewers

1 ATVPDKIKX0DER

2 A3UN6WX5RRO2AG

3 A14OJS0VWMOSWO

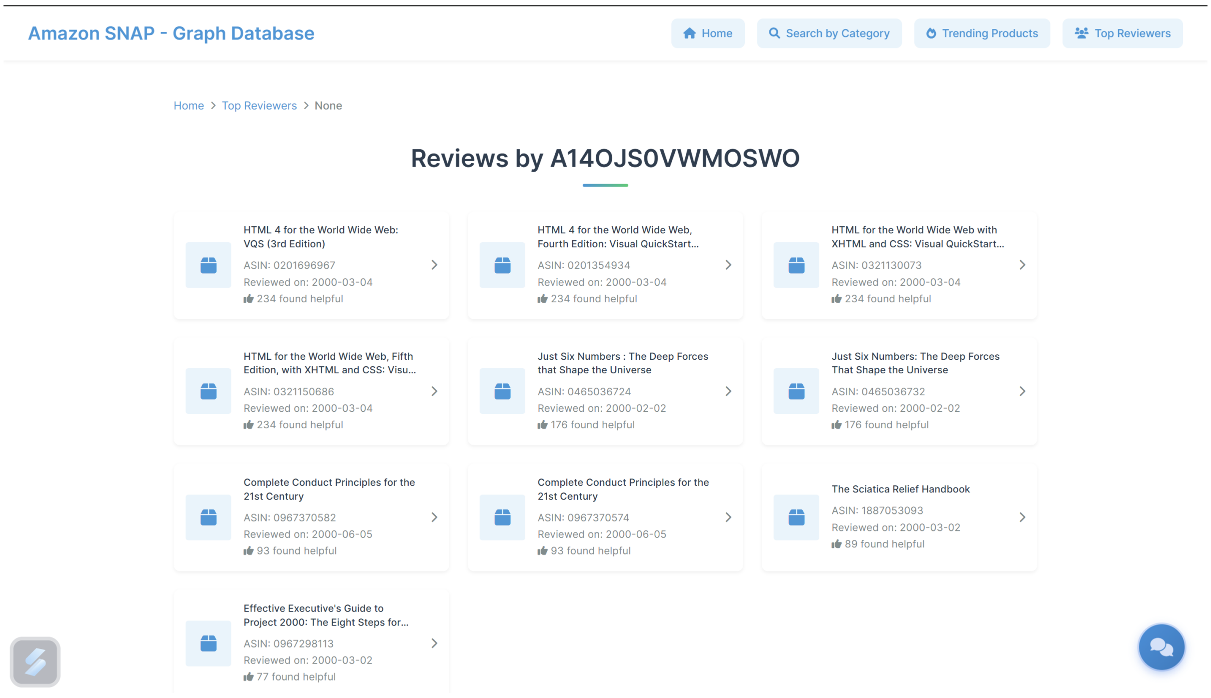
4 A2NJO6YE954DBH

5 AFVQZQ8PWOL

6 A1K1JW1C5CUSUZ

7 A2ODBHT4URXVXQ

Figure 8: Top Reviewers


Amazon SNAP - Graph Database

Home | Search by Category | Trending Products | Top Reviewers

Home > Top Reviewers > None

Reviews by A14OJS0VWMOSWO

HTML 4 for the World Wide Web: VQS (3rd Edition)
ASIN: 0201696967
Reviewed on: 2000-03-04
234 found helpful

HTML 4 for the World Wide Web, Fourth Edition: Visual QuickStart...
ASIN: 0201354934
Reviewed on: 2000-03-04
234 found helpful

HTML for the World Wide Web with XHTML and CSS: Visual QuickStart...
ASIN: 0321130073
Reviewed on: 2000-03-04
234 found helpful

HTML for the World Wide Web, Fifth Edition, with XHTML and CSS: Visu...
ASIN: 0321150686
Reviewed on: 2000-03-04
234 found helpful

Just Six Numbers : The Deep Forces that Shape the Universe
ASIN: 0465036724
Reviewed on: 2000-02-02
176 found helpful

Just Six Numbers: The Deep Forces That Shape the Universe
ASIN: 0465036732
Reviewed on: 2000-02-02
176 found helpful

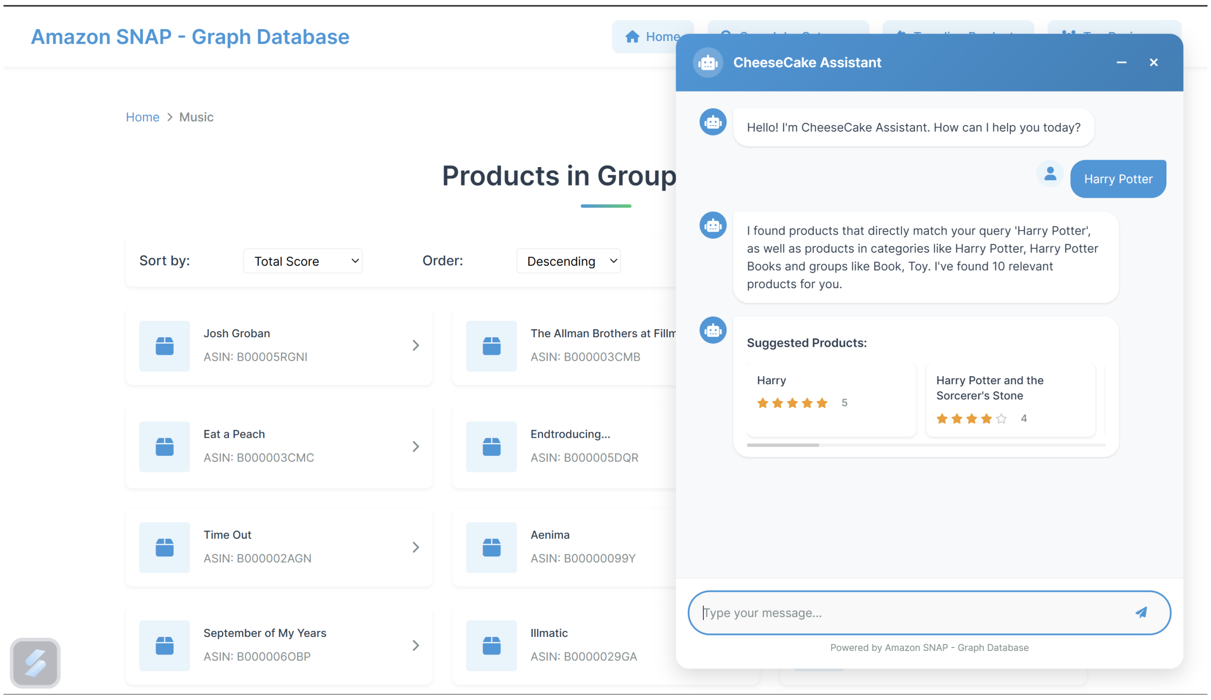
Complete Conduct Principles for the 21st Century
ASIN: 0967370582
Reviewed on: 2000-06-05
93 found helpful

Complete Conduct Principles for the 21st Century
ASIN: 0967370574
Reviewed on: 2000-06-05
93 found helpful

The Sciatica Relief Handbook
ASIN: 1887053093
Reviewed on: 2000-03-02
89 found helpful

Effective Executive's Guide to Project 2000: The Eight Steps for...
ASIN: 0967298113
Reviewed on: 2000-03-02
77 found helpful

Figure 9: Best Reviewed Products of top Reviewers


Amazon SNAP - Graph Database

Home

Home > Music

Products in Group

Sort by: Total Score   Order: Descending

Josh Groban
ASIN: B00005RGNI

The Allman Brothers at Fillm
ASIN: B000003CMB

Eat a Peach
ASIN: B000003CMC

Endtroducing...
ASIN: B000005DQR

Time Out
ASIN: B000002AGN

Aenima
ASIN: B00000099Y

September of My Years
ASIN: B000006OBP

Illmatic
ASIN: B0000029GA

CheeseCake Assistant

Hello! I'm CheeseCake Assistant. How can I help you today?

Harry Potter

I found products that directly match your query 'Harry Potter', as well as products in categories like Harry Potter, Harry Potter Books and groups like Book, Toy. I've found 10 relevant products for you.

Suggested Products:

Harry ★★★★★ 5

Harry Potter and the Sorcerer's Stone ★★★★☆ 4

Type your message...

Powered by Amazon SNAP - Graph Database

Figure 10: Integrated RAG-based ChatBox

Future Scope

1. Advanced Hybrid Recommendation Techniques

- Add knowledge graph reasoning for explainable recommendations.
- Integrate real-time session-based recommendations using streaming graph updates.

2. **Enhanced Graph Learning**
   - Implement Graph Neural Networks (GNNs) for dynamic relationship modelling.
   - Add temporal graph networks for dynamic tracking of evolving product popularity.
   - Explore heterogeneous graph transformers for multi-modal data.
3. **Scalability Improvements**
   - Implement Graph Partitioning for distributed Neo4j clusters.
   - Add incremental graph updates for real-time score recalculations.
   - Explore vector-graph hybrid search combining FAISS with Cypher.
4. **RAG System Enhancements**
   - Integrate multi-modal embeddings.
   - Add conversational memory for chat-based recommendations.
   - Implement query intent analysis using graph-aware LLMs.
5. **Social Network Integration**
   - Add social graph features for friend-based recommendations.
   - Track cross-product network effects through user communities.

References

- `https://en.wikipedia.org/wiki/Graph_database`
- `https://aws.amazon.com/nosql/graph/`
- `https://youtu.be/y7sXDpffzQQ?si=m-o_b-sak4HKdg29`
- `https://neo4j.com/docs/getting-started/`
- `https://neo4j.com/docs/getting-started/appendix/tutorials/guide-cypher-basics/`
- `https://youtu.be/8jNPelugC2s?si=9TsH3yZc_4r3oeld`
- `https://youtu.be/REVkXVxvMQE?si=SKpIsYs08T-kQVq0`
- `https://youtu.be/oRtVdXvtD3o?si=0CooM0pvgB2FxWyP`