MongoDB

using Compass and Azure Cosmos DB

Harshit Rai, Ishank Vasania, Danish Nadeem

BE BOUNDLESS

Contents

Introduction

- The system of our choice, MongoDB, is a NoSQL document-based database management system.
- MongoDB is an open-source, highly scalable, and flexible database that is designed to handle unstructured data.
- It is widely used in web applications and big data environments due to its ease of use and high performance.
- The aim of this project is to test the effectiveness of MongoDB in dealing with graph data.
- The dataset of choice is Amazon Metadata which represents the purchase of various products by customers.

About the Dataset

Amazon-metadata file with about 550000 rows containing information on the products sold on Amazon. The size of the file is ~1 GB.

Fields and their descriptions:

- ➔ Id: A unique identifier for the product, assigned by Amazon.
- ➔ ASIN: The Amazon Standard Identification Number for the product.
- ➔ Title: The title of the product.
- ➔ Group: The group that the product belongs to (e.g., Book, Music, DVD).
- ➔ Salesrank: The sales rank of the product on Amazon.
- ➔ Similar: A list of ASINs for products that are frequently co-purchased with the current product.
- ➔ Categories: A list of category paths that the product belongs to on Amazon.
- ➔ Reviews: A list of review objects, each containing information about a customer review for the product.

4 **node** files which represent different combinations of products bought together.

Node: Each node in the network represents a unique product. Each node has associated attributes such as product ID, product name, category, brand, description, etc.

Edge: The edges in the network represent co-purchasing relationships between products.

An edge exists between two nodes if those products have been purchased together by Amazon customers.

Fields and their descriptions:

- ➔ Date: The date that the review was posted.
- ➔ Customer: A unique identifier for the customer who posted the review.
- ➔ Rating: The rating that the customer gave the product (1-5 stars).
- ➔ Votes: The number of helpful votes that the review received.
- ➔ Helpful: The number of customers who found the review helpful.

```
{
  "From": 0,
  "To": 1
},
{
  "From": 0,
  "To": 2
},
{
  "From": 0,
  "To": 3
},
{
  "From": 0,
  "To": 4
},
{
  "From": 0,
  "To": 5
},
{
  "From": 0,
  "To": 6
},
{
  "From": 0,
  "To": 7
}
```


Raw JSON file


```
{
  "product_id": 0,
  "co_purchases": [
    1,
    2,
    3,
    4,
    5
  ],
  "Date": "March 02 2003",
  "month": "March"
},
{
  "product_id": 1,
  "co_purchases": [
    0,
    2,
    4,
    5,
    15
  ],
  "Date": "March 02 2003",
  "month": "March"
},
{
  "product_id": 2,
  "co_purchases": [
    0,
    11,
    12,
```


Processed JSON file

Data Model

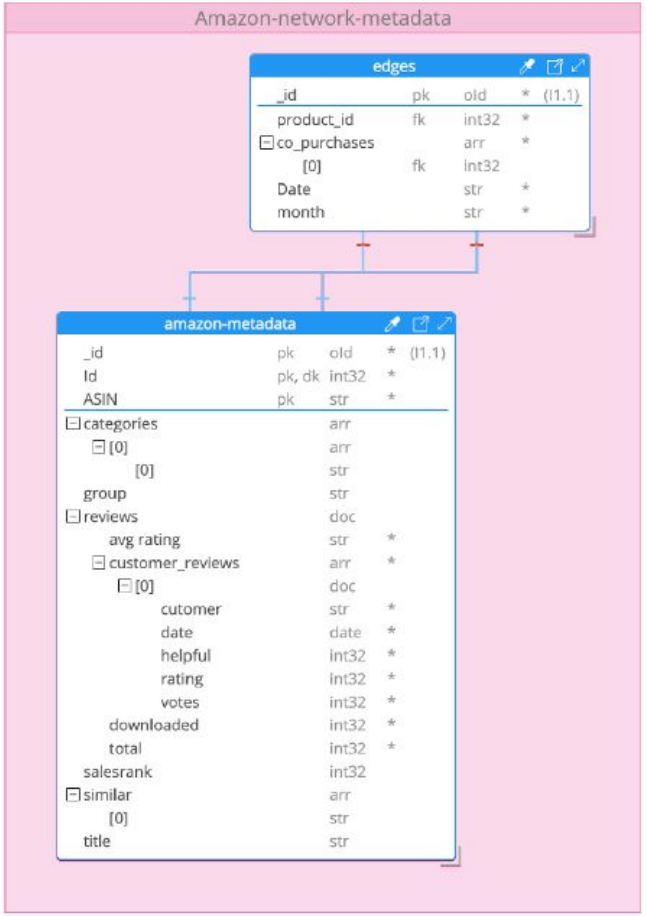
- While designing the data model, the key considerations include the performance requirements of the application, the complexity of the data relationships, and the need for data consistency and integrity.
- The original dataset had 4 different files to represent edges.
- If we included these 4 files directly into our mongodb database the size limit was exceeded as a total of 5 collections were found to be too intensive for MongoDB to handle given the resource constraints.
- To tackle this issue, all of the 4 files containing edges collection were merged into one file.
- We basically converted the data from long format to wide format hence saving memory consumption.
- In conclusion, for the Amazon product review dataset, we chose a document-based data model in MongoDB due to its flexibility, scalability, and ability to handle unstructured data efficiently.
- The images in the following slides represent the data model in the visual format.

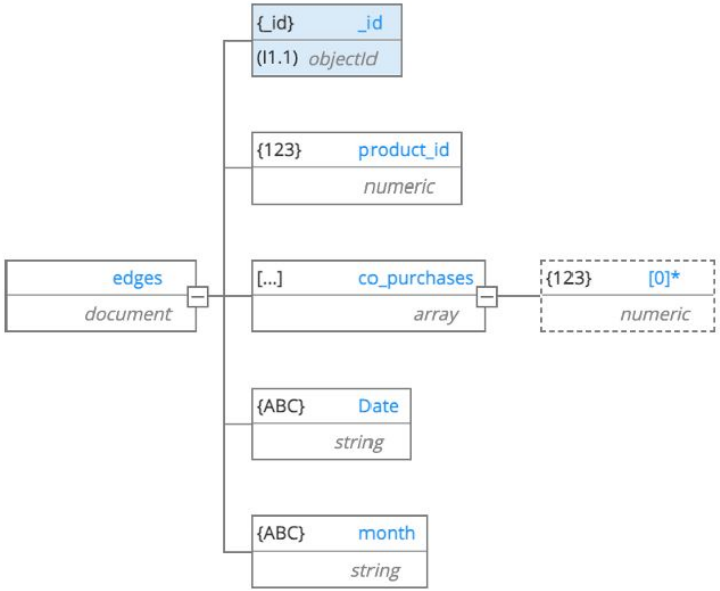
```
Id:   1
ASIN: 0827229534
  title: Patterns of Preaching: A Sermon Sampler
  group: Book
  salesrank: 396585
  similar: 5  0804215715  156101074X  0687023955  0687074231  082721619X
  categories: 2
   |Books[283155]|Subjects[1000]|Religion & Spirituality[22]|Christianity[12290]|Clergy[12360]|Preaching[12368]
   |Books[283155]|Subjects[1000]|Religion & Spirituality[22]|Christianity[12290]|Clergy[12360]|Sermons[12370]
  reviews: total: 2  downloaded: 2  avg rating: 5
    2000-7-28  cutomer: A2JW67OY8U6HHK  rating: 5  votes:  10  helpful:   9
    2003-12-14  cutomer: A2VE83MZF98ITY  rating: 5  votes:   6  helpful:   5

Id:   2
ASIN: 0738700797
  title: Candlemas: Feast of Flames
  group: Book
  salesrank: 168596
  similar: 5  0738700827  1567184960  1567182836  0738700525  0738700940
  categories: 2
   |Books[283155]|Subjects[1000]|Religion & Spirituality[22]|Earth-Based Religions[12472]|Wicca[12484]
   |Books[283155]|Subjects[1000]|Religion & Spirituality[22]|Earth-Based Religions[12472]|Witchcraft[12486]
  reviews: total: 12  downloaded: 12  avg rating: 4.5
    2001-12-16  cutomer: A11NCO6YTE4BTJ  rating: 5  votes:   5  helpful:   4
    2002-1-7  cutomer:  A9CQ3PLRNIR83  rating: 4  votes:   5  helpful:   5
    2002-1-24  cutomer: A13SG9ACZ9O5IM  rating: 5  votes:   8  helpful:   8
    2002-1-28  cutomer: A1BDAI6VEYMAZA  rating: 5  votes:   4  helpful:   4
    2002-2-6  cutomer: A2P6KAWXJ16234  rating: 4  votes:  16  helpful:  16
    2002-2-14  cutomer:  AMACWC3M7PQFR  rating: 4  votes:   5  helpful:   5
    2002-3-23  cutomer: A3GO7UV9XX14D8  rating: 4  votes:   6  helpful:   6
    2002-5-23  cutomer: A1GIL64QK68WKL  rating: 5  votes:   8  helpful:   8
    2003-2-25  cutomer:  AEOBOF2ONQJWV  rating: 5  votes:   8  helpful:   5
    2003-11-25  cutomer: A3IGHTES8ME05L  rating: 5  votes:   5  helpful:   5
    2004-2-11  cutomer: A1CP26N8RHYVVO  rating: 1  votes:  13  helpful:   9
    2005-2-7  cutomer:  ANEIANH0WAT9D  rating: 5  votes:   1  helpful:   1

Id:   3
ASIN: 0486287785
  title: World War II Allied Fighter Planes Trading Cards
  group: Book
  salesrank: 1270652
  similar: 0
  categories: 1
   |Books[283155]|Subjects[1000]|Home & Garden[48]|Crafts & Hobbies[5126]|General[5144]
```

Raw text file

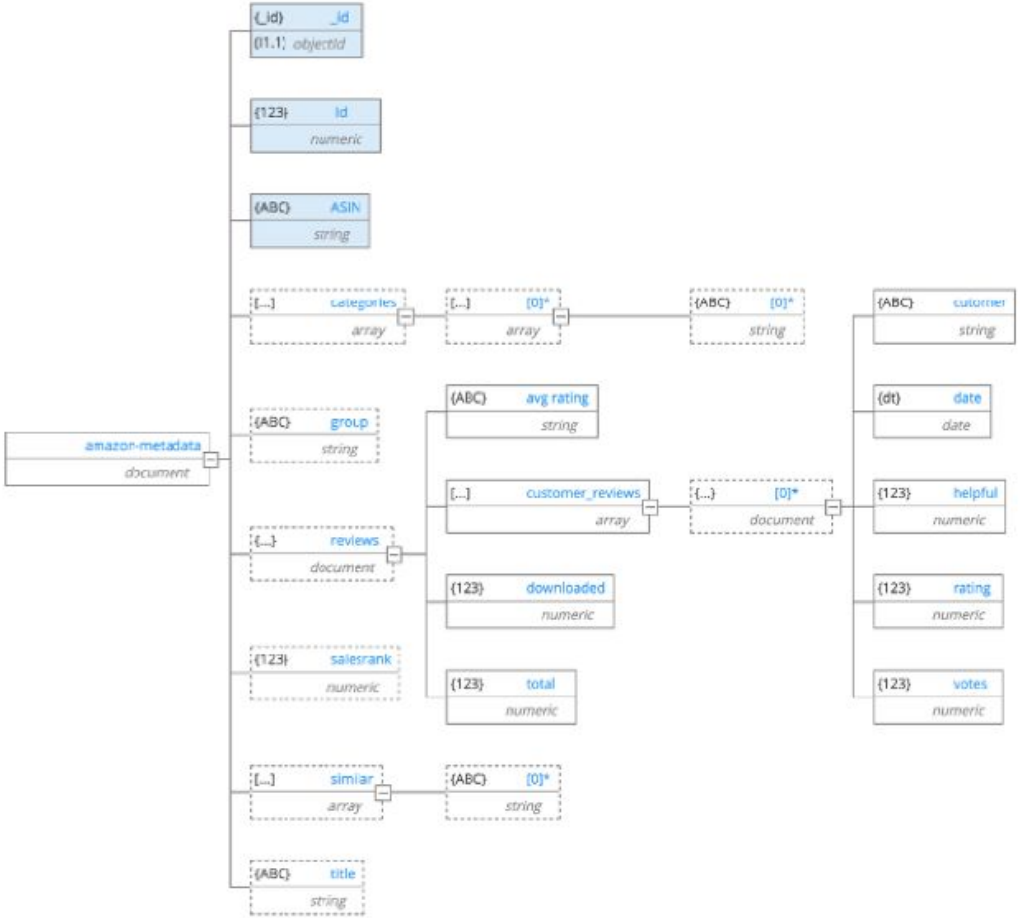
Processed JSON file

UNIVERSITY of WASHINGTON





UNIVERSITY of WASHINGTON


amazon-metadata
document
{_id} _id
(01.1) objectId
{123} id
numeric
{ABC} ASIN
string
[...] categories
array
[...] [0]*
array
{ABC} [0]*
string
{ABC} group
string
{ABC} avg rating
string
{...} reviews
document
[...] customer_reviews
array
{...} [0]*
document
{ABC} customer
string
{dt} date
date
{123} helpful
numeric
{123} rating
numeric
{123} votes
numeric
{123} downloaded
numeric
{123} total
numeric
{123} salesrank
numeric
[...] similar
array
{ABC} [0]*
string
{ABC} title
string

Data Preprocessing

Objective: Convert the raw data into a semi-structured and usable JSON format.

- ➔ Format of the raw data: tab-spaced .txt file
- ➔ This format is not supported by MongoDB
- ➔ Expected structure is JSON
- ➔ JSON format allows for easy data manipulation, querying, and integration with various tools and technologies
- ➔ Solution- A custom Python script that processes each product from the raw text file and converts it into a corresponding JSON object.

Preprocessing Steps

Amazon-Metadata

1. Data Extraction
    a. Read the raw text file containing the Amazon-Meta dataset.
    b. Load the entire dataset into memory.
2. Regular Expression Matching
    a. Apply regular expressions to extract relevant information from the dataset.
    b. Use different regex patterns to match global block, review, and customer review patterns.
3. JSON Conversion
    a. Parse the extracted information and convert it into a structured JSON format.
    b. Create nested JSON objects and arrays to represent different data components.
4. Write to file

Final JSON Structure

- ➔ Top-level object: Contains an array of individual product objects
- ➔ Product object: Represents a single product with its attributes and reviews.
- ➔ Attributes: ASIN, title, group, sales rank, similar products, and categories.
- ➔ Similar: ASIN of similar products.
- ➔ Categories: Categories to which the product was mapped
- ➔ Reviews: Includes total reviews, downloaded reviews, average rating, and customer reviews.

Node Files

1. Load JSON data
    a. Extract the month from the file name
    b. Create a product dictionary
    c. Iterate over each line in the data
    d. Extract "From" and "To" IDs
    e. Add the "To" ID to the co-purchases list of the "From" ID in the dictionary
2. Transform and store JSON:
    a. Create a list of products
    b. For each product in the dictionary:
    c. Assign product ID, co-purchases, date, and month
    d. Append the product to the list

Output: co-purchases.json that represents co-purchase relationships between products that is useful for further analysis and processing

Query Execution

Please visit the following link to watch various features of MongoDB and execution of queries in action:

Link to the Video

Link to the GitHub Repository

Azure Cosmos DB Primary Connection String (to access data storage):

mongodb://mongodb-data514:c74DCYlirQo4OnJil0bjA6gqwe3H3YeEejNw44b7DuPPgNO4raHMc8vtGtUbWnOGWmtJDscXzqjIACDbsZXWMA==@mongodb-data514.mongo.cosmos.azure.com:10255/?ssl=true&replicaSet=globaldb&retrywrites=false&maxIdleTimeMS=120000&appName=@mongodb-data514@

Conclusion

- ➔ Several native MongoDB (available in Compass) features are not supported on Azure Cosmos DB- lack of support for performance monitoring within Compass, unsupported index types, etc.
- ➔ Queries frequently timing out
- ➔ Several features are limited to Atlas
- ➔ No 'one place to do it all' like MS SQL Server (Hackolade Studio- third party application for data modeling and schema design being advertised in Compass application)
- ➔ In our experiments, MongoDB using the Cosmos DB API was often sluggish and wasn't efficient enough in dealing with graph data stored on Azure.
- ➔ However, these issues could be primarily due to the compute and cost restraints and we believe that a more powerful compute may be required to perform various CRUD operations successfully.