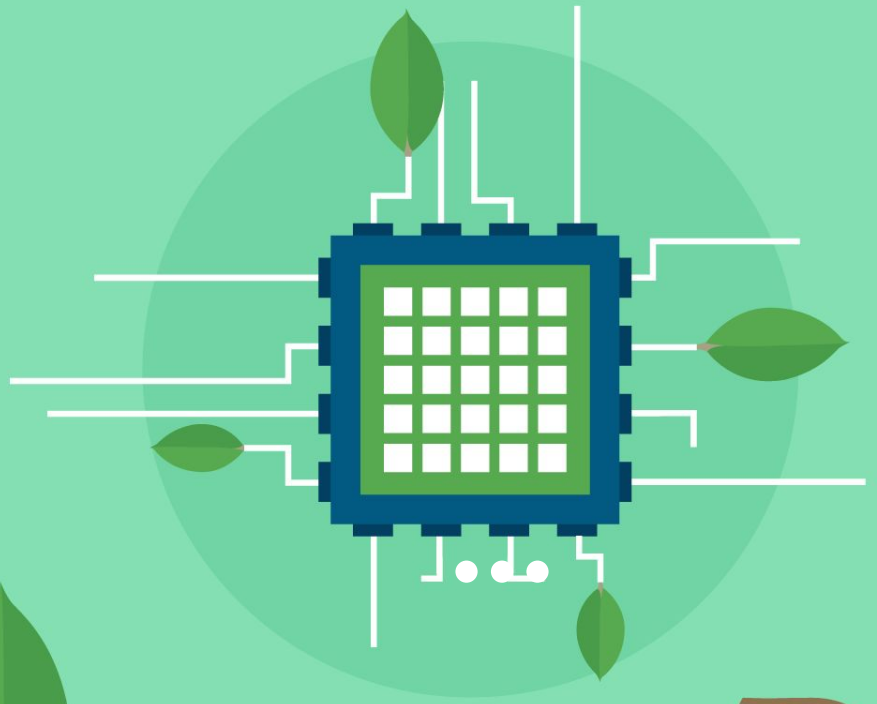


mongoDB

Harshit Rai, Ishank Vasania, Mohammad Danish Nadeem

Main Goal

- Demonstrate various features of MongoDB

- Test the effectiveness of MongoDB in handling graph data

Features of MongoDB

- Flexible Schema
- Scalability
- High Performance
- Rich Query Language
- Replication and High Availability
- Horizontal Scaling
- Easy Integration
- Full Text Search
- Geospatial Capabilities
- Aggregation Framework
- Document-Oriented

Applications of MongoDB

1. Real-time Analytics
2. Internet of Things (IoT)
3. Catalog and Product Inventories
4. Social Networks and Collaboration
5. Data Caching and Session Storage
6. Mobile Application

Famous Companies using MongoDB: Ebay, EA, Shutterfly and many others

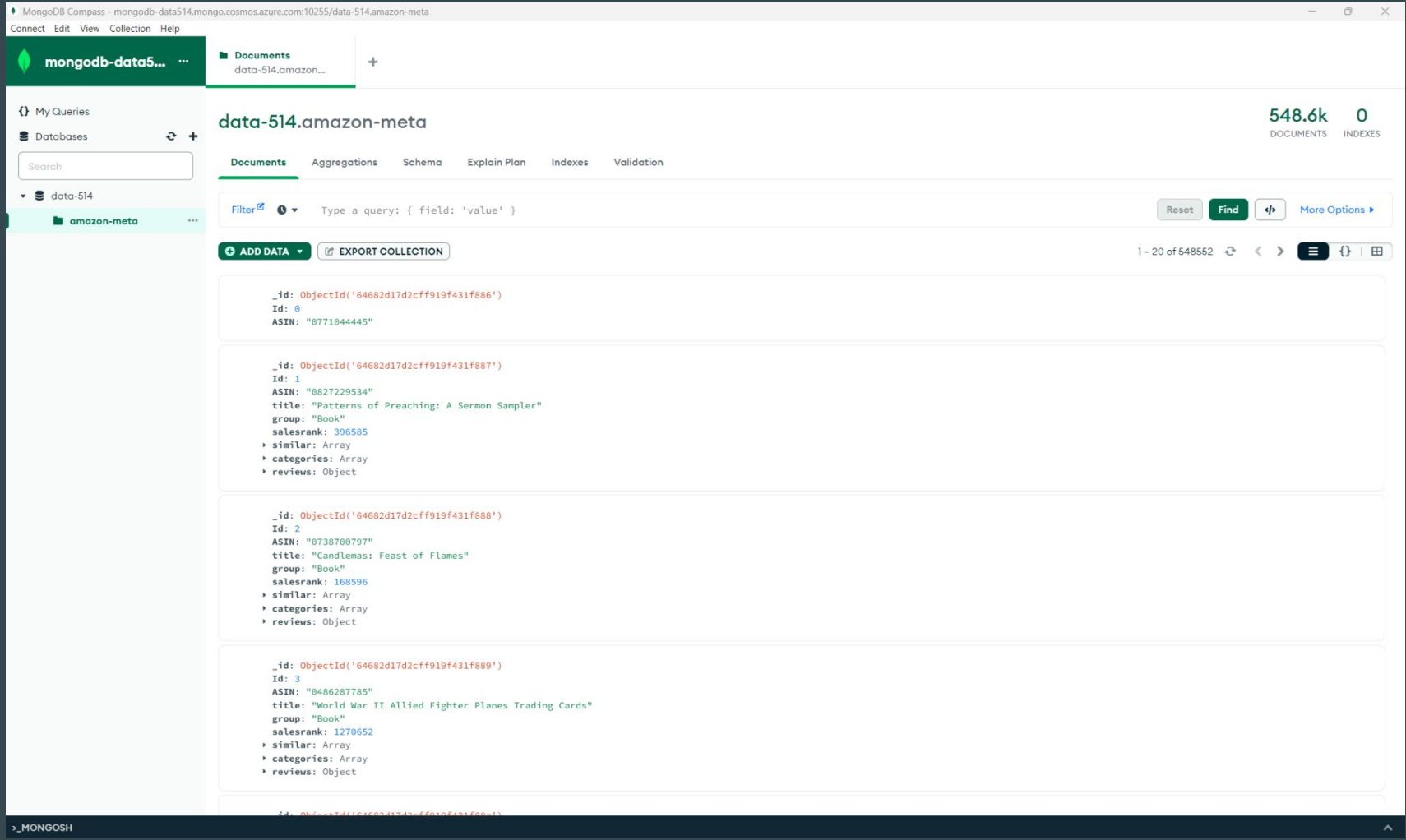
MongoDB Compass

Advantages:

- User-Friendly Interface
- Schema Visualization
- Query Building
- Index Optimization

Disadvantages:

- Limited Functionality
- Learning Curve

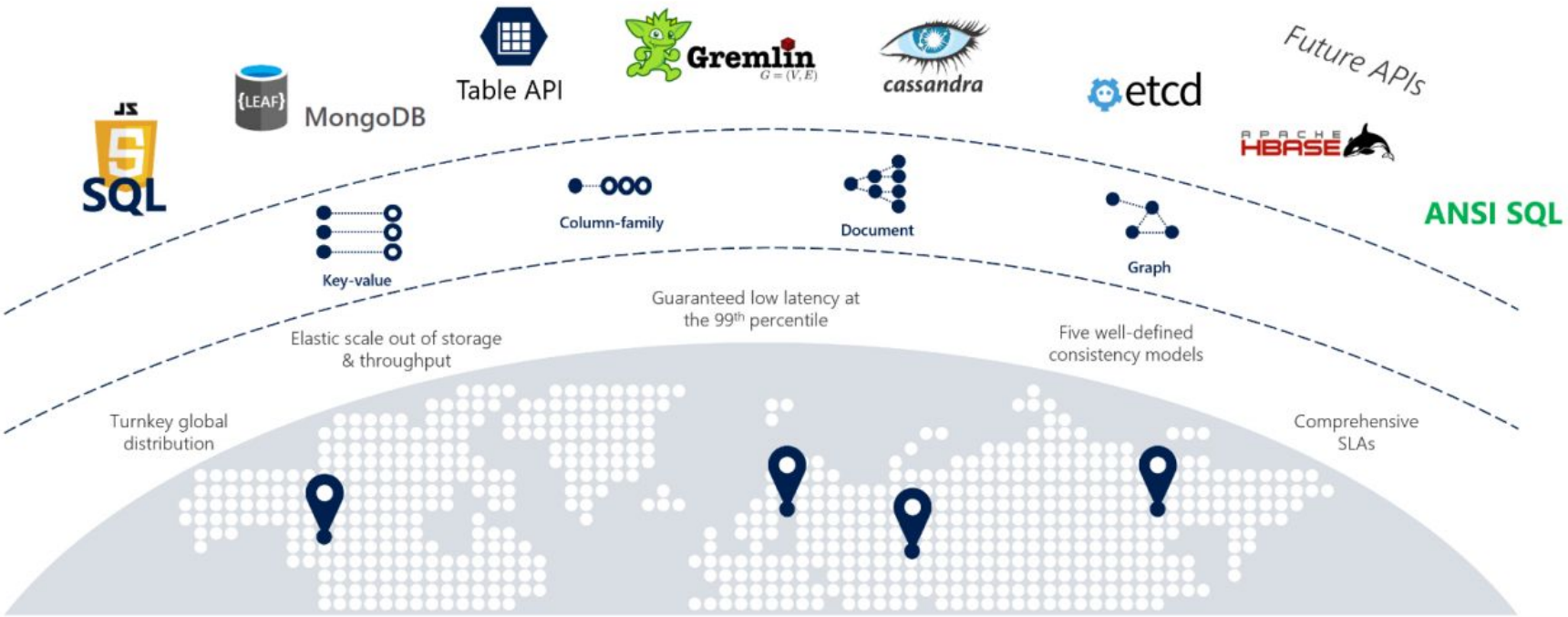

MongoDB Compass - mongodb-data514.mongo.cosmos.azure.com:10255/data-514.amazon-meta
Connect Edit View Collection Help
mongodb-data5...
Documents
data-514.amazon...
My Queries
Databases
Search
data-514
amazon-meta
data-514.amazon-meta
548.6k
DOCUMENTS
0
INDEXES
Documents
Aggregations
Schema
Explain Plan
Indexes
Validation
Filter
Type a query: { field: 'value' }
Reset
Find
More Options
ADD DATA
EXPORT COLLECTION
1 - 20 of 548552
_id: ObjectId('64682d17d2cff919f431f886')
Id: 0
ASIN: "0771044445"
_id: ObjectId('64682d17d2cff919f431f887')
Id: 1
ASIN: "0827229534"
title: "Patterns of Preaching: A Sermon Sampler"
group: "Book"
salesrank: 396585
similar: Array
categories: Array
reviews: Object
_id: ObjectId('64682d17d2cff919f431f888')
Id: 2
ASIN: "0738700797"
title: "Candlemas: Feast of Flames"
group: "Book"
salesrank: 168596
similar: Array
categories: Array
reviews: Object
_id: ObjectId('64682d17d2cff919f431f889')
Id: 3
ASIN: "0486287785"
title: "World War II Allied Fighter Planes Trading Cards"
group: "Book"
salesrank: 1270652
similar: Array
categories: Array
reviews: Object
>_MONGOSH

MongoDB Atlas v/s Cosmos DB

Comparison Criteria	Cosmos DB	Atlas
Vendor and Ecosystem	Microsoft Azure	MongoDB
Supported Databases	DocumentDB, MongoDB, SQL, Gremlin, Table	MongoDB
Compatibility	Compatible with multiple APIs	Compatible with MongoDB's wire protocol
Global Distribution	Multiple data centers, global replication	Multiple regions, data replication across regions
Pricing Model	Provisioned throughput	Usage-based pricing

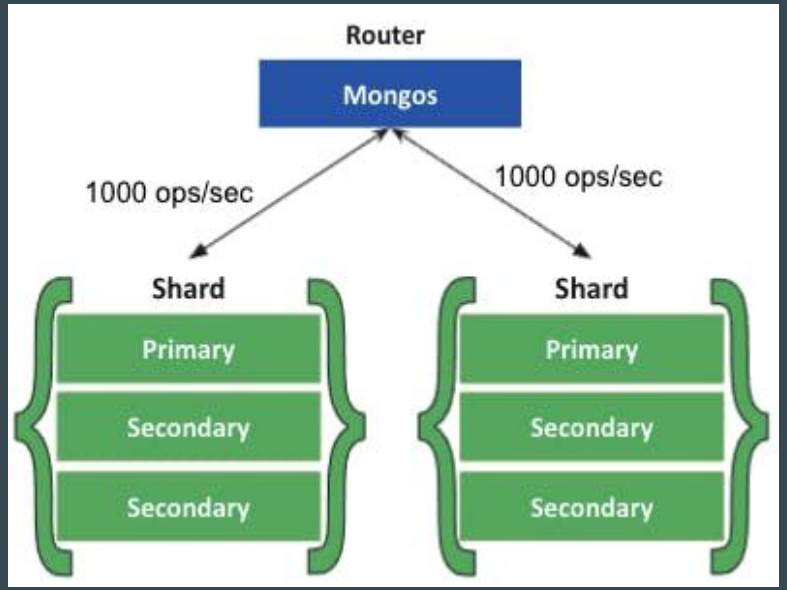
Azure Cosmos DB

Microsoft's globally distributed, massively scalable, multi-model database service



Sharding in MongoDB

- Sharding: MongoDB's method for distributing data across multiple machines.
- Challenges: Large data sets or high throughput can strain a single server's capacity.
- Vertical Scaling: Increasing a server's capacity (CPU, RAM, storage), but with limitations.
- **Horizontal Scaling**: Dividing workload among multiple servers, providing efficiency and flexibility.
- Trade-off: Horizontal scaling requires more complex infrastructure but can be cost-effective.
- Sharding != Partitioning



ACID in MongoDB

1. Atomicity, Consistency, Isolation, and Durability.
2. Initial support for ACID principles on single-document level.
3. Added support for multi-document ACID transaction in v4.0 (2018).
4. Added support for distributed multi-document ACID transaction in v4.2 (2019).
5. Application of multi-document ACID transactions-
    a. Updating the status of an account across all those users' documents.
    b. Logging custom application actions -- say when a user transfers ownership of an entity, the write should not be successful if the logging isn't.
    c. Aggregations of hundreds of thousands of trades, need to be updated every time trades are added or modified.

Data Profile

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.

```
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


```
_id: ObjectId('64682d17d2cff919f431f887')
Id: 1
ASIN: "0827229534"
title: "Patterns of Preaching: A Sermon Sampler"
group: "Book"
salesrank: 396585
similar: Array
  0: "0804215715"
  1: "156101074X"
  2: "0687023955"
  3: "0687074231"
  4: "082721619X"
categories: Array
  0: Array
    0: "Books[283155]"
    1: "Subjects[1000]"
    2: "Religion & Spirituality[22]"
    3: "Christianity[12290]"
    4: "Clergy[12360]"
    5: "Preaching[12368]"
  1: Array
    0: "Books[283155]"
    1: "Subjects[1000]"
    2: "Religion & Spirituality[22]"
    3: "Christianity[12290]"
    4: "Clergy[12360]"
    5: "Sermons[12370]"
reviews: Object
  total: 2
  downloaded: 2
  avg rating: "5"
  customer_reviews: Array
    0: Object
      date: "2000-7-28"
      cutomer: "A2JW67OY8U6HHK"
      rating: 5
      votes: 10
      helpful: 9
    1: Object
      date: "2003-12-14"
      cutomer: "A2VE83MZF98ITY"
      rating: 5
      votes: 6
      helpful: 5
```


Processed JSON file

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 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.

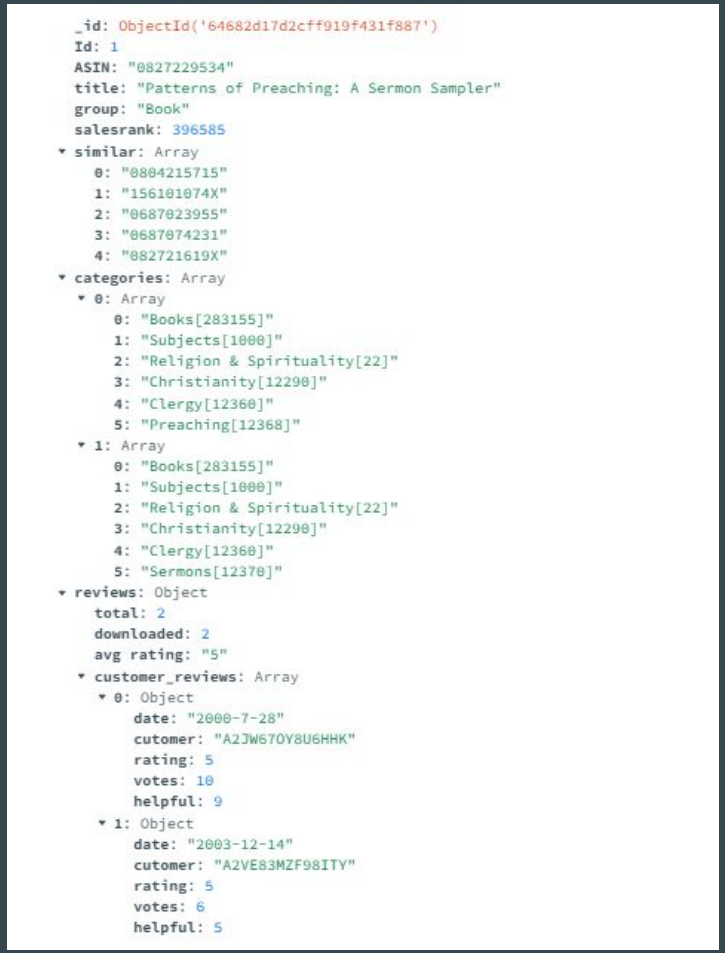
Preprocessing Steps

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

Mongodb System Summary

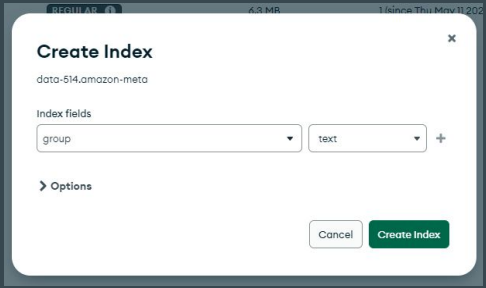
Data Model

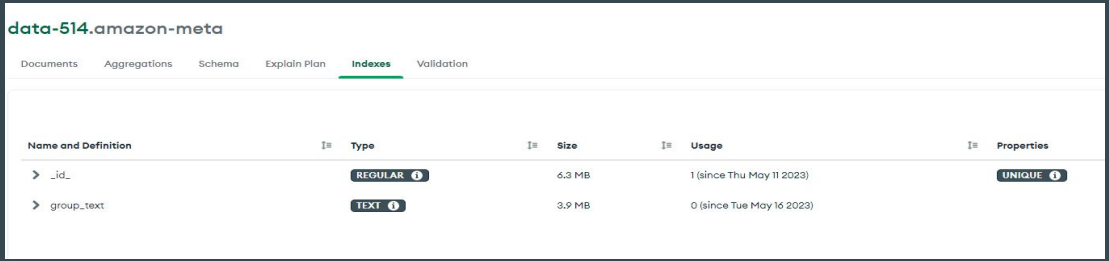
MongoDB uses a document-based data model. Documents are stored as JSON-like objects, and each document can have a different structure. Each document contains key-value pairs, where the keys represent field names and the values represent data. MongoDB supports dynamic schema, which means that we can add new fields to a document at any time. Below is an example of how data can be stored in a MongDB document-based data *collection*. The Data Model used for the Amazon dataset is explained in more detail in the later part of this document

MongoDB System Summary

Indexes

MongoDB supports various types of indexes, including single field indexes, compound indexes, multikey indexes, geospatial indexes, and text indexes. Indexes can significantly improve query performance by reducing the number of documents that need to be scanned. Below is an example of an index created on the *group* field of the dataset





System Design

Storing Data in MongoDB
The following steps will be followed to store the dataset in MongoDB.

1. **Initialize System:** Setup the MongoDB system on a local machine and create a corresponding MongoDB instance on Microsoft Azure.
2. **Data Overview:**
    a. Exploring the Raw Data
    b. Identifying Useful Fields
    c. Designing the Data Model

3. **Data Preprocessing and Conversion:** Perform preprocessing to convert the raw data in text format to JSON format using Python. This will help loading the data into a format supported by MongoDB and the data model designed in the previous step.

4. **Data Ingestion:** Create a MongoDB database and collection to store the dataset. We use the MongoDB (Compass) graphical user interface to create the database and collection.
5. **Querying the Data:** Once the data is stored in MongoDB, perform various queries and analysis on the data using the supported query language and aggregation framework.

MongoDB Data Model Variants

- Embedded Data Model
  - In this variant, related data is stored within a single document as nested fields or arrays.
  - It allows for efficient retrieval of complete data in a single query.
  - Suitable for one-to-one and one-to-many relationships where the embedded data is small and does not require frequent updates.
- Referenced Data Model
  - In this variant, data is distributed across multiple collections, and relationships are established using references or foreign keys.
  - It allows for flexibility in managing and updating related data independently.
  - Suitable for many-to-many relationships or scenarios where the related data is large, frequently updated, or subject to changes.
- Hybrid Data Model
  - This variant combines elements of both the embedded and referenced data models.
  - It leverages the benefits of both approaches, allowing for flexibility and performance optimizations.
  - Suitable for complex relationships or scenarios where a balance between data duplication and data consistency is required.

Data Model Variants

Variant 1

Edges collection:

```
{
  "_id": ObjectId("61a76f104e5f151a482f9844"),
  "product_id":
ObjectId("61a76f104e5f151a482f9845"),
  "co_purchases": [
    ObjectId("61a76f104e5f151a482f9846"),
    ObjectId("61a76f104e5f151a482f9847")
  ],
  "date": ISODate("2023-05-20T00:00:00Z"),
  "month": "May"
}
```

Selected Model

Edges collection:

```
{
   "_id":"ObjectId",
   "product_id":"NumberInt",
   "co_purchases":[
      "NumberInt"
   ],
   "date":"ISODate",
   "month":"String"
}
```

Data Variants

Variant 2:

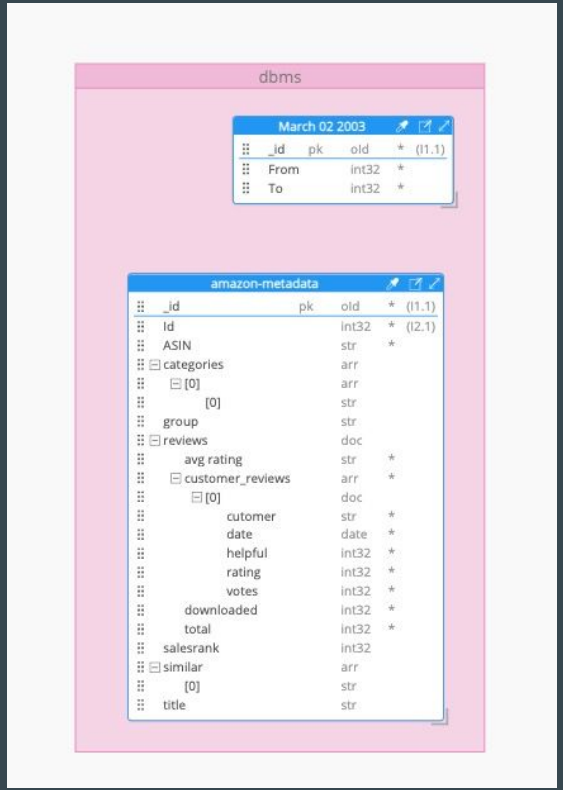
```
{
  "_id":
ObjectId("61a76f104e5f151a482f
9844"),
  "From": NumberInt(12345),
  "To": NumberInt(12345)
}
```

Selected Model

Edges collection:

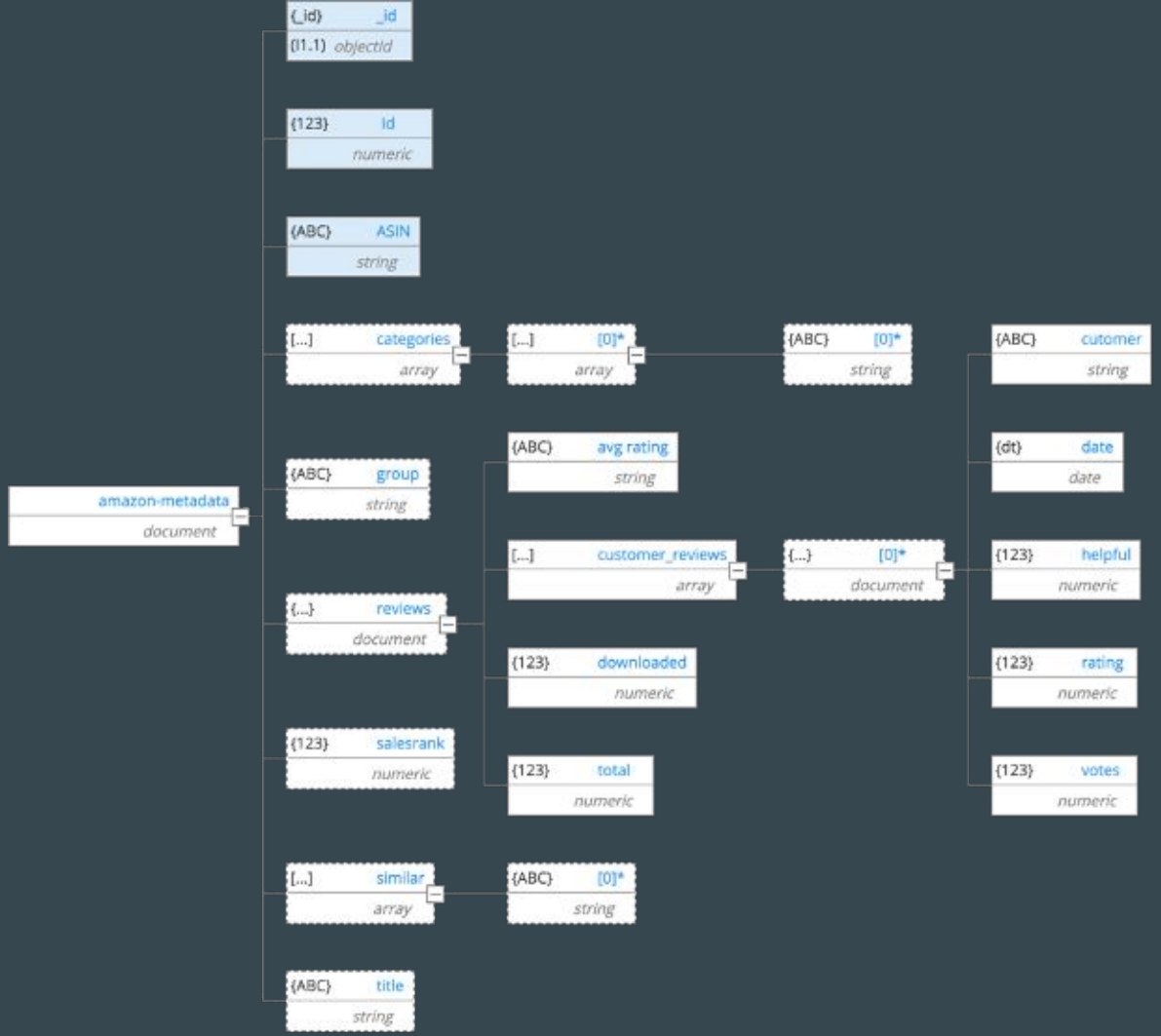
```
{
  "_id":"ObjectId",
  "product_id":"NumberInt",
  "co_purchases":[
    "NumberInt"
  ],
  "date":"ISODate",
  "month":"String"
}
```

Variant 2:


dbms
March 02 2003
_id pk oid * (I1.1)
From int32 *
To int32 *
amazon-metadata
_id pk oid * (I1.1)
Id int32 * (I2.1)
ASIN str *
categories arr
[0] arr
[0] str
group str
reviews doc
avg rating str *
customer_reviews arr *
[0] doc
cutomer str *
date date *
helpful int32 *
rating int32 *
votes int32 *
downloaded int32 *
total int32 *
salesrank int32
similar arr
[0] str
title str

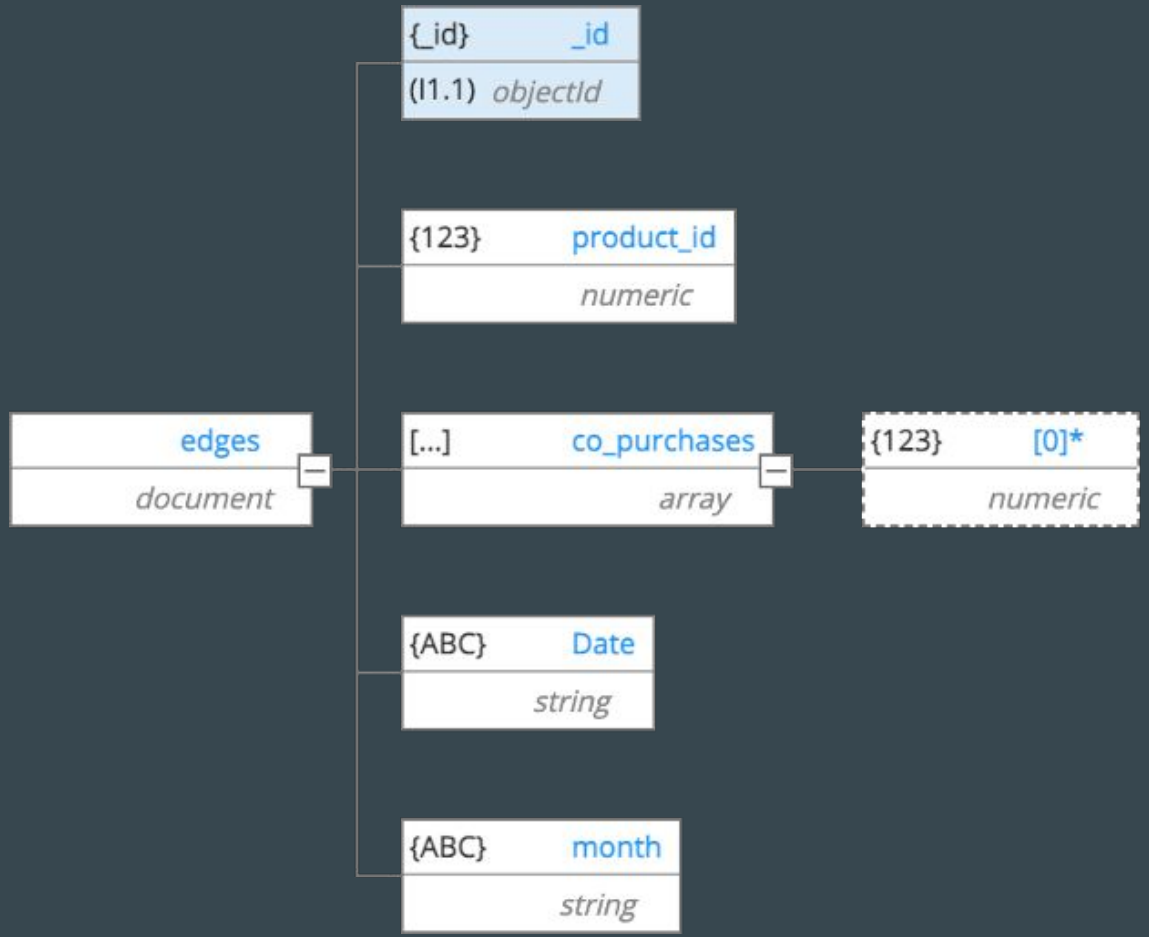
System Design

The diagram illustrate the schema of the *amazon-metadata* collection that are part of the Amazon dataset under consideration.



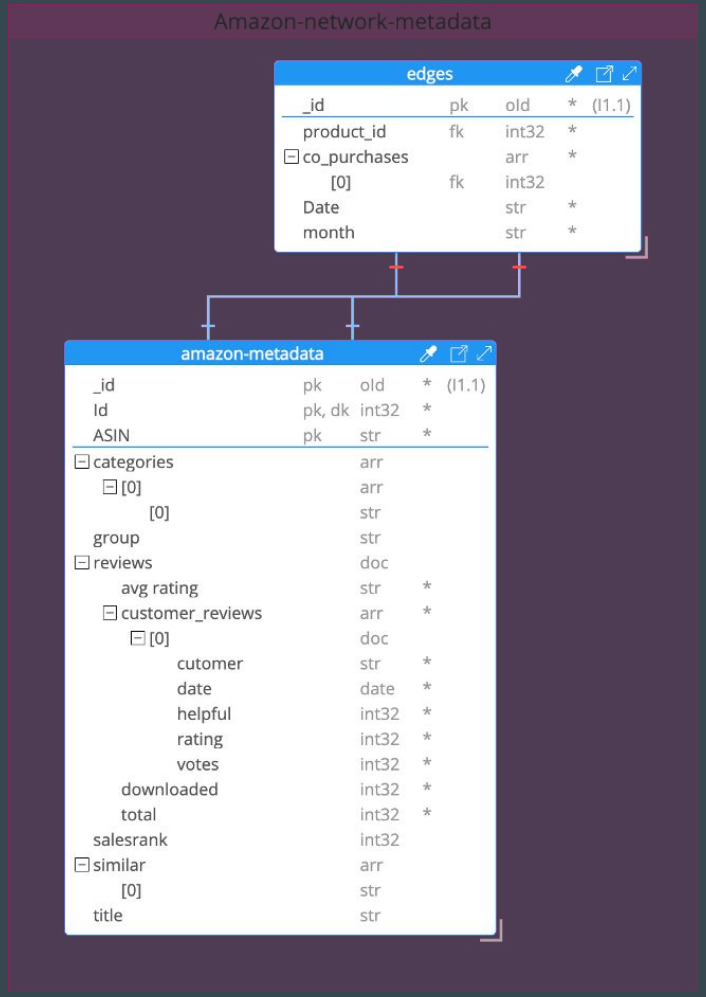
System Design

The diagram illustrate the schema of the *edges* collection that are part of the Amazon dataset under consideration.



System Design

ER/Class Diagram



System Design

Indexes used

Amazon-metadata collection

1. Unique Index on ASIN: The **ASIN** field is unique for each document in the **products** collection, hence we created a unique index on this field to prevent duplicate products from being inserted.
2. Unique Index on Product ID: The **product_id** field is unique for each document in the **products** collection, hence we created a unique index on this field to prevent duplicate products from being inserted.

System Design

Reason for selecting the data model and its potential variant

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. In the variant, as could be seen in the variant 2 image, we had the edges from 02-March-2003. To tackle this issue, all of the 4 files containing *edges* collection were merged into one file. The new resultant file created is shown in the ER diagrams in the previous parts and now also has date as well as co-purchases. 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.

System Design

Preprocessing to restructure the data

The "amazon-meta.txt" input file contains data that our custom Python script analyses and converts to JSON format. The input file includes details on different items sold on Amazon's website. To extract particular patterns, including customer reviews, overall reviews, and information about global blocks, our script makes use of regular expressions. The aforementioned Python script handles a range of keys and values, transforms some values into integers or floats, and arranges the information into dictionaries. The parsed data is then written in JSON format to a new file called "sample.json" and kept in a list. Overall, the script organizes and extracts the pertinent data from the input file into a more understandable and consistent format for additional processing or analysis.

Aggregations in MongoDB

In MongoDB, aggregation and querying are two fundamental operations used to retrieve and manipulate data.

Querying:

Querying in MongoDB involves searching for documents in a collection that match certain criteria. It allows you to specify conditions, projections, and sorting parameters to filter and retrieve data. MongoDB provides a flexible and powerful query language that supports a wide range of operators and expressions for complex querying.

```
db.users.find({ name: "John" })
```

Aggregation:

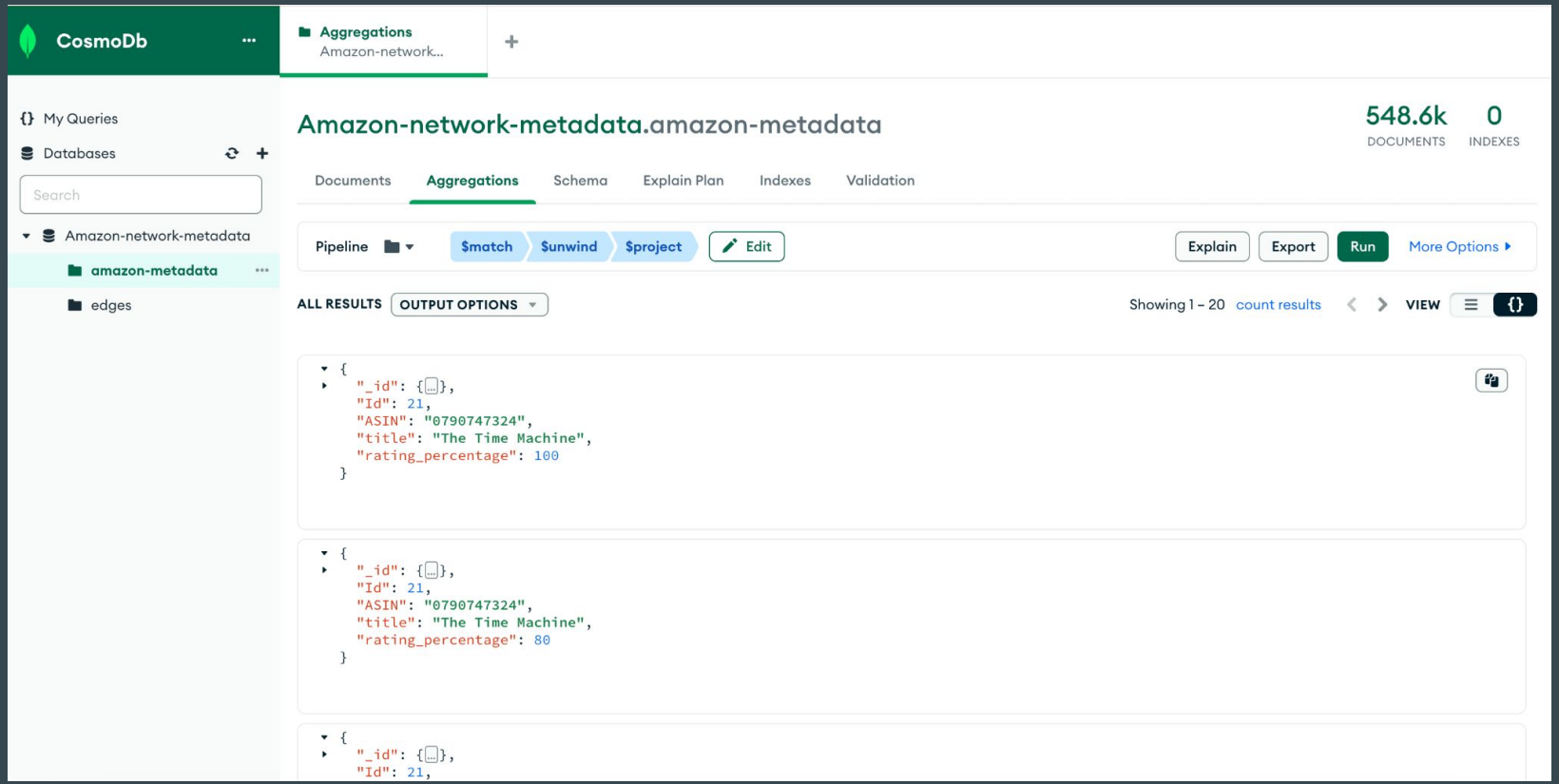
Aggregation in MongoDB involves processing and transforming data from multiple documents in a collection to produce aggregated results. It allows you to perform operations such as grouping, filtering, sorting, calculating averages, sums, counts, and other aggregations on the data.

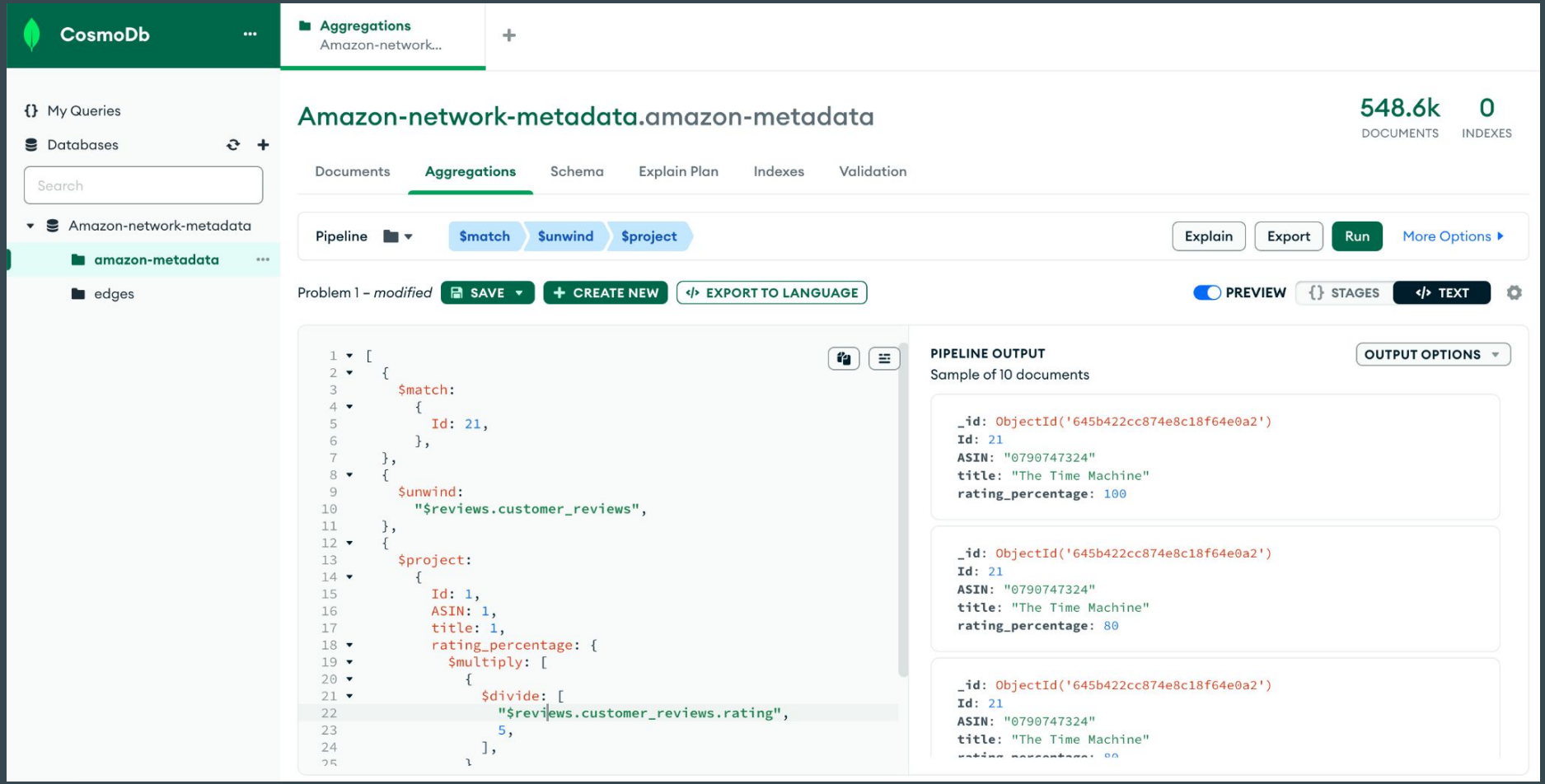
```
db.users.aggregate([

  { $group: { _id: "$city", averageAge: { $avg: "$age" } } }

])
```

Query to answer:

What are the percentages of each rating digit for the product with id: 21?

Aggregations and Querying




CosmoDb
Aggregations
Amazon-network...
My Queries
Databases
Search
Amazon-network-metadata
amazon-metadata
edges
Amazon-network-metadata.amazon-metadata
548.6k
DOCUMENTS
0
INDEXES
Documents
Aggregations
Schema
Explain Plan
Indexes
Validation
Pipeline
$match
$unwind
$project
Explain
Export
Run
More Options
Problem 1 – modified
SAVE
CREATE NEW
EXPORT TO LANGUAGE
PREVIEW
STAGES
TEXT
[
{
$match:
{
Id: 21,
},
},
{
$unwind:
"$reviews.customer_reviews",
},
{
$project:
{
Id: 1,
ASIN: 1,
title: 1,
rating_percentage: {
$multiply: [
{
$divide: [
"$reviews.customer_reviews.rating",
5,
],
PIPELINE OUTPUT
Sample of 10 documents
OUTPUT OPTIONS
_id: ObjectId('645b422cc874e8c18f64e0a2')
Id: 21
ASIN: "0790747324"
title: "The Time Machine"
rating_percentage: 100
_id: ObjectId('645b422cc874e8c18f64e0a2')
Id: 21
ASIN: "0790747324"
title: "The Time Machine"
rating_percentage: 80
_id: ObjectId('645b422cc874e8c18f64e0a2')
Id: 21
ASIN: "0790747324"
title: "The Time Machine"

Conclusion

Journey so far:

- Explore various features offered by MongoDB and Cosmos DB
- Preprocess raw data into meaningful and useful JSON format
- Extract co-purchase relationships between products

Next steps:

- Answer various questions by executing queries-
  - Identify products frequently purchased together
  - Detected patterns and trends in purchasing behavior
- Analyze the effectiveness of MongoDB in handling graph data

References

- https://www.mongodb.com/docs/manual/tutorial/
- https://learn.mongodb.com/
- https://techcrunch.com/2018/09/24/microsoft-updates-its-planet-scale-cosmos-db-database-service/
- https://www.mongodb.com/docs/manual/crud/
- https://www.mongodb.com/docs/manual/sharding/
- https://www.mongodb.com/basics/acid-transactions
- https://chat.openai.com/
- https://www.mongodb.com/docs/manual/core/aggregation-pipeline/
- https://www.mongodb.com/docs/compass/current/query/filter/

License

Contact: tufte@pdx.edu