**Cloud Data Modeling - Mongo dB**

**Final Term Paper**

***CS777-prof. Farshid Alizadeh-Shabdiz***

**Yiduo Feng**

**11/02/2022**

**Abstract**

MongoDB is a database management system designed for web applications and Internet infrastructure. MongoDB is NoSQL type of database. The reason I didn't choose NoSQL but Mongo dB is because it's a product between relational and non-relational databases, which is very appealing to me and will be more useful for data analysis. In my subsequent paper, I want to discuss in depth mainly through MongoDB's unique model and how can we implement machine learning through the use of Mongo DB.

In business, there is growing interest in artificial intelligence and machine learning. The predictive power of ML/AI enables rapid insight to be gained from detected patterns at a faster rate than human analysis. Companies realize that this can lead to increased profits, lower costs and accelerated innovation. While businesses large and small can benefit from the power of AI, implementing predictive analytics projects can be complex and time-consuming.

Mongo BD's data relational modeling will allow huge data to be better planned and simplified in this regard. The first is a one-to-one embedded document model in Mongo DB. When we look for specific elements in huge data, the one-to-one embedded document model can help us to find the corresponding element by known attribute lookup. This model will be very useful when we do some vision or graphics machine learning. [2] The second is a one-to-many embedded document model [3]. We know that with the continuous development of computer languages, there are already many efficient supervised machine learning in python, java and R language that can predict classification problems with high accuracy, but I found that unsupervised machine learning predicts The accuracy of the results is often unsatisfactory. Because unsupervised machine learning has no known classifications and answers, and their attributes are not one-to-one but one-to-many, one-to-many embedded document models are needed. The third is one-to-many relationship modeling: document reference pattern [4]. It can summarize data to better help us draw histograms and so on.

**Introduction to Cloud Database MongoDB**

MongoDB is written in C++, and the popular open source database MySQL is also developed in C++. Released in 1983, C++ is a widely used computer programming language.[1] It is a general-purpose programming language that supports multiple programming patterns. Cloud Database MongoDB provides users with the capabilities of NoSQL databases as a service, which shows great advantages in terms of flexibility and ease of use, high availability, fully managed operation and maintenance, and data security and reliability. MongoDB is not a new database. As far as I know, MongoDB has been widely used in China's e-commerce field as early as a few years ago, and it has officially positioned itself as a general-purpose database on its official website. MongoDB is actually somewhat similar to MySQL in terms of data processing. Even though it has been used by some small businesses in China over the past few years, it still hasn't replaced MySQL, it has even been replaced by MySQL in many ways. However, many people said that MongoDB is not uncompetitive. It has many similarities with the imperfect MySQL in the past, and some of its advantages cannot be replaced. It is believed that it will become more powerful and popular over time.

MongoDB is a document-based NoSQL database. Data is stored in MongoDB in the form of documents. So what is a document-based database?

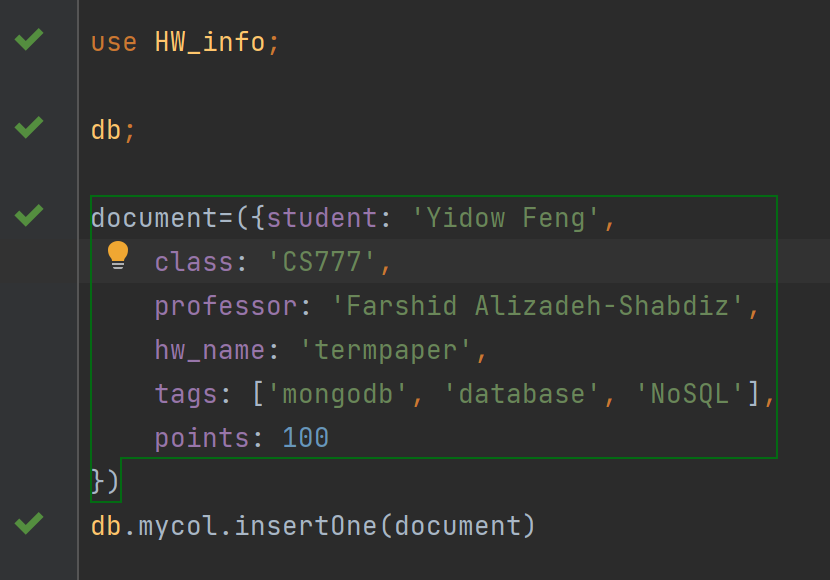
* A document database is a type of non-relational database. A document is a record in a document database. Documents typically store information about an object and any associated metadata. Documents store data in field-value pairs. Values ​​can be of various types and structures, including strings, numbers, dates, arrays, and more. The format of document storage can be JSON, BSON and XML. A collection is a set of documents. Documents in a collection usually have a similar structure. All documents in a collection do not need to have consistent fields. Some document-based databases provide format validation, so a collection's fields can also be fixed if desired.

A well-known disadvantage of document databases is that they do not support multi-document transactions. The so-called multi-document transaction refers to the need to operate multiple documents at the same time in a transaction. However, MongoDB can already support multi-document transactions.

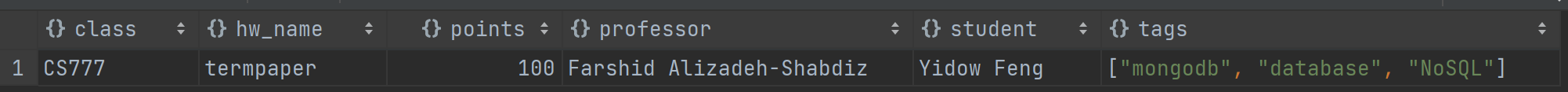
**Advantages of MongoDB**

Based on the current big data processing, Python is the most mainstream computer language (when we use Spark, we also choose PySpark in the Python environment), and the JSON format is very stable in python, and python also provides good support. Therefore, the data in JSON format read from MongoDB can be used directly by developers without secondary processing. Another point, the Key-Value key value in a MongoDB document supports a variety of data structures. Value can be an integer, a string, an array, or a nested subdocument. The advantage of using nesting is that in MongoDB Get the data you need with just one simple query.

These unique attributes can be directly nested in the product document in the form of JSON subdocuments, and one query can directly obtain all the content without using "join" in multiple tables. Another great feature of MongoDB documents is schema flexibility. The value type of the same key in different documents can be an integer or other types such as strings, and different documents can have different keys. MongoDB query also provides a very rich set of operators, which can be combined in a query to double the efficiency. The flexibility of MongoDB's schema is more reflected in the convenience and speed of adding fields. For example, we are inserting a document into the collection now.



As can be seen from the above operations, MongoDB does not need to build a table in advance, nor does it need to specify each field in the table, and it is flexible to use.

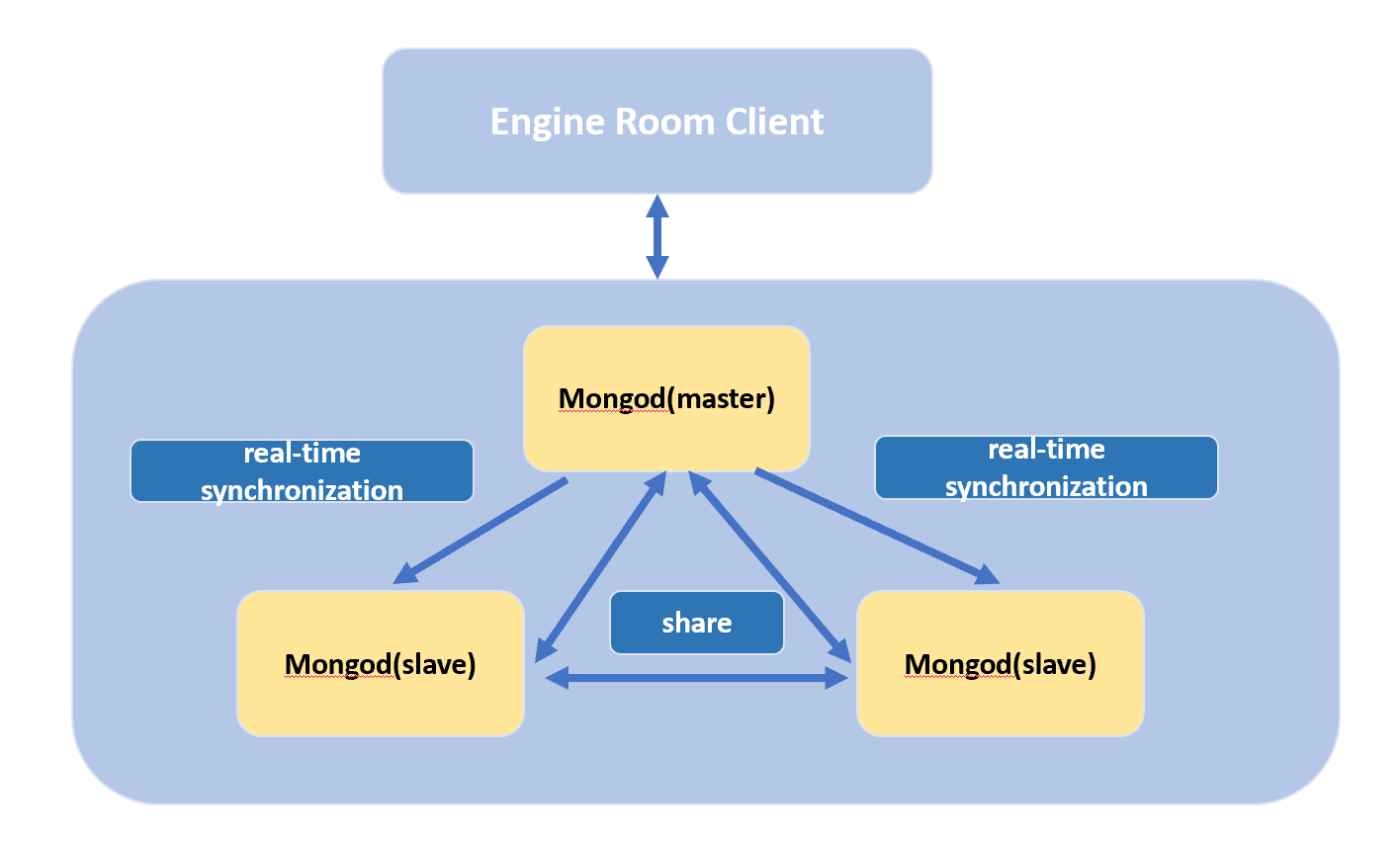


Then, we can get the results like this directly.

In addition, MongoDB also supports us to add a non-existing field directly to an existing document. Compared with dataframe in Spark, we can understand that in MongoDB, we are not constrained by "Schema", so we can operate and change the database more easily.

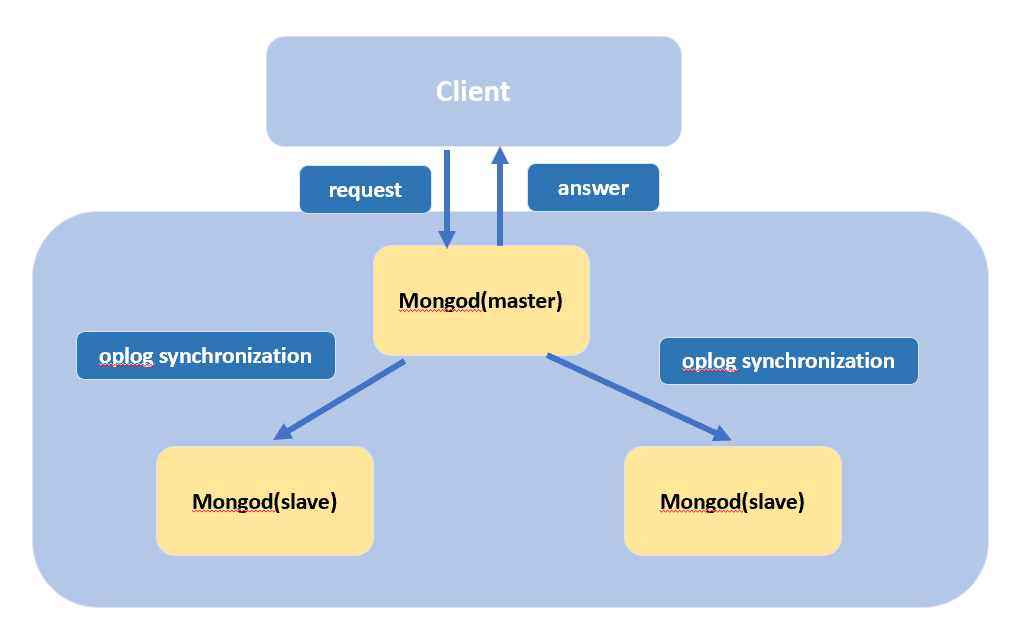
The document-based flexible data model is a major advantage of MongoDB. For business scenarios with diverse or changeable data models, compared with databases such as MySQL, there is no need to modify the table structure; compared with other Key-Value databases, Since the Value field of MongoDB is non-transparent to MongoDB, it can be indexed and full-text search can be performed, which has more advantages in query efficiency. This mode is very suitable in the fields of games, e-commerce, social networking, live video, logistics, etc. It can achieve fast query by nesting sub-documents with different purposes in users or products. It is also applicable to scenarios such as monitoring, log data storage, and third-party information capture, because the fields contained in different monitoring data, log records, and captured data are often different, and to some extent, they are uncontrollable. At the same time, the flexible model is also suitable for products or scenarios such as game market activities and mobile apps that require rapid development and launch but with relatively large changes in demand.

The current open source database Mysql, when the master server in the same shard replication set (assuming one master and two slaves) is abnormal or the instance exits abnormally, it needs to rely on a third-party MHA plug-in to realize the election of the new master. If there is no third-party MHA plug-in, or the third-party MHA plug-in is abnormal, the new master node cannot be elected, resulting in a write failure. Compared with MySQL's dependence on third-party MHA plug-ins, the MongoDB database comes with a natural high-availability election function.



As shown in the figure above, when the master node is abnormal, the keep-alive messages of the two Mongod slave nodes will detect the exception, and then the two slave nodes will conduct a new round of master node voting (raft protocol). The master node is abnormal, and the remaining two nodes can vote normally. Because the raft majority requirement is met, a new master node will be quickly selected among the two master nodes.

Because of this structure, complete data consistency and security guarantees are also the advantages of MongoDB.



Finally, in order to meet the needs of different business links, Mongodb supports two thread models: one link and one thread and the adaptive dynamic thread pool model, so high concurrency and high performance are also a major advantage of MongoDB.

**Model One-to-One Relationships with Embedded Documents**

In MongoDB official website, it described this model is “the model uses embedded documents to describe a one-to-one relationship between connected data. Embedding connected data in a single document can reduce the number of read operations required to obtain data. [2]”

Here I downed the MovieLens 10M dataset, which consists of 10million movie ratings from “grouplens.org” web site as an example [5]. This source provides tree related datasets, and they are ratings, movies and tags.

This dataset divides a large amount of data into these three different files because of the "schema" limitation of the data and in order to let the downloader understand the data more clearly. When we read the data details, we will find that these three data documents have several fields that are the same and can be merged. MongoDB's One-to-One Relationships model can better organize this dataset.

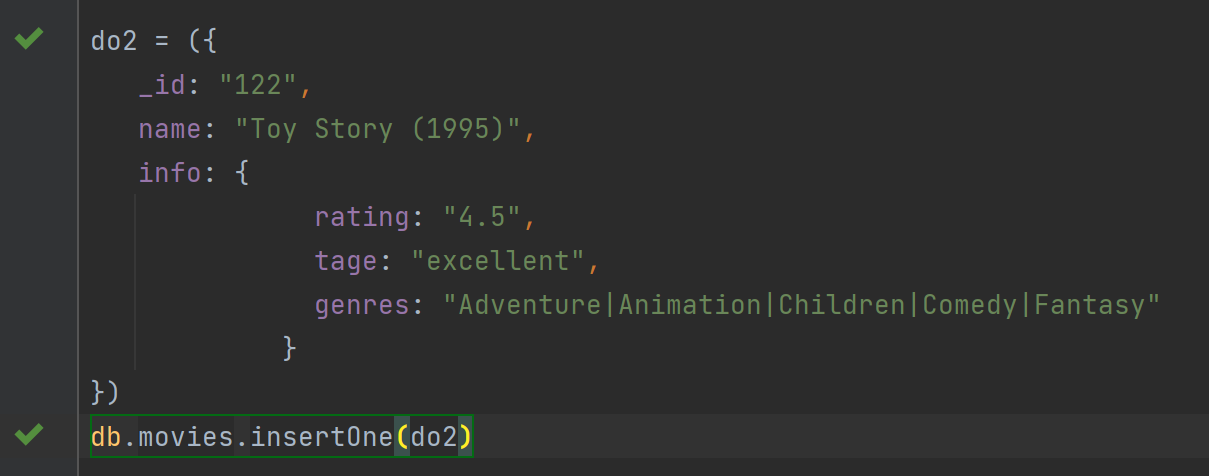
First let's see what columns are in the three data files:

Ratings: UserID::MovieID::Rating::Timestamp

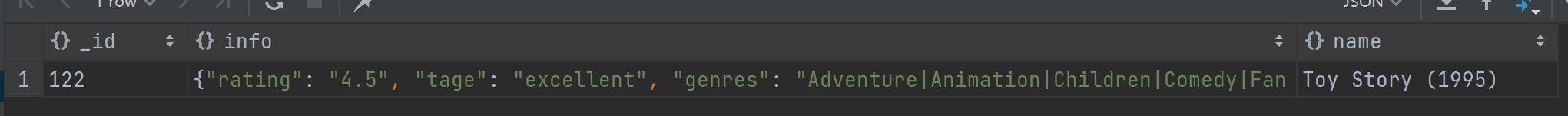
Tags: UserID::MovieID::Tag::Timestamp

Movies: MovieID::Title::Genres

We can embed ratings, tags, genres in a movie as a whole.



Then when we search the id of the movies, we can get the information of movie quickly by using the database in MongoDB, and the results will look like below:

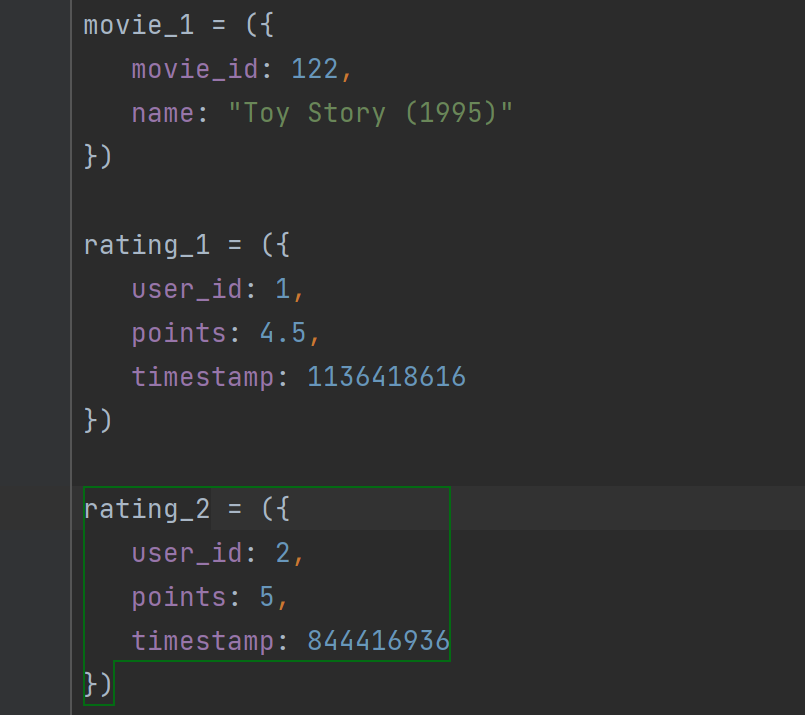


In this way, when merging data, developers can not only quickly clean up the data and combine multiple documents into one document, but also avoid the repetition of movie information and reduce the difficulty of data cleaning. What’s more, after we finish inserting data, the subsequent search work becomes faster and more convenient.

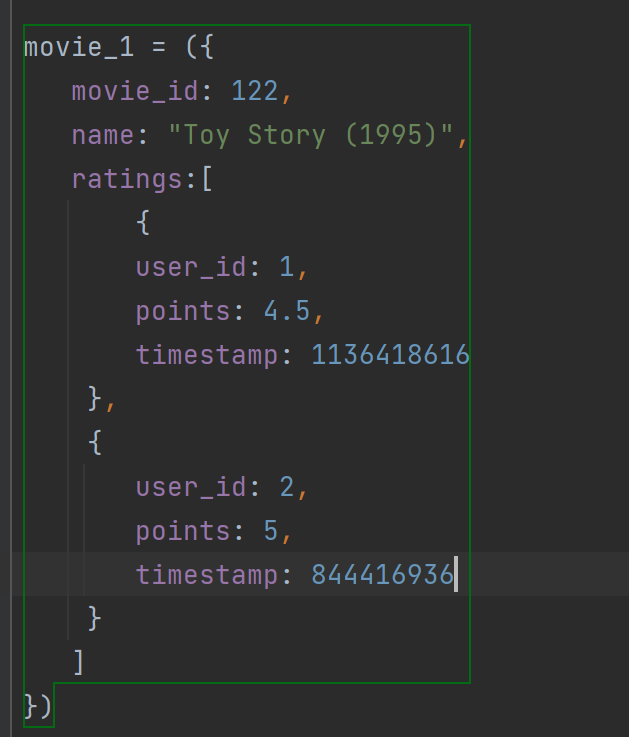
**Model One-to-Many Relationships with Embedded Documents**

Similar to the model above, it “uses embedded documents to describe a one-to-many relationship between connected data. Embedding connected data in a single document can reduce the number of read operations required to obtain data [3].”

Using the same example as the previous model, a movie will be rated differently by different people at different times, and the information is scattered.



Now we can use One-to-Many model to embed the data.

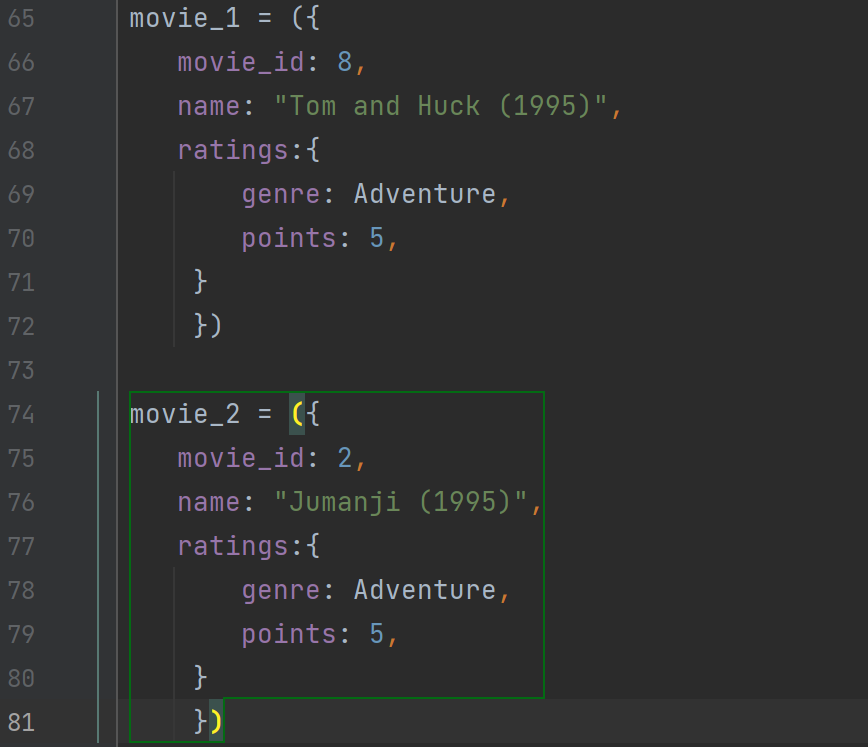


With this model, we can better aggregate the data. When we want to get the average rating of a movie, we don't need to search and recalculate in two datasets, we just need to search the movie between the MongoDB databases to get all the ratings for a single movie.

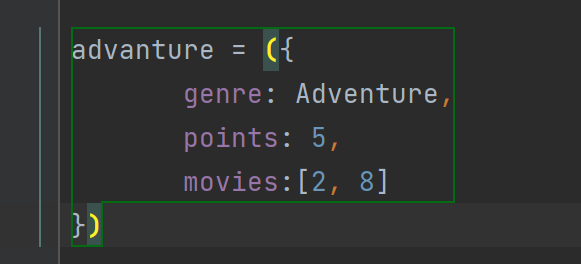
**Model One-to-Many Relationships with Document References**

This model “uses references between documents to describe one-to-many relationships between connected data. [4]”

When we sort out the reviews of movies, we find that there are many movies of the same genre with the same ratings. To aggregate movies of the same genre and with the same rating, we can use this model.



After using One-to-Many Relationships with Document References, we get:



**Conclusion**

MongoDB's non-relational document database has great advantages in inserting and organizing data. MongoDB's high concurrency, high performance, schema freedom, natural high availability support, and perfect data consistency and security guarantees are irreplaceable advantages for many databases at present. Although MongoDB is not currently a mainstream database, its embedded model can quickly clean up useless and duplicate data for complex documents.

**References**

[1] “MongoDB.” *Wikipedia*, Wikimedia Foundation, 28 Oct. 2022, https://en.wikipedia.org/wiki/MongoDB.

[2] *Model One-to-One Relationships with Embedded Documents - MongoDB Manual*, https://www.mongodb.com/docs/manual/tutorial/model-embedded-one-to-one-relationships-between-documents/.

[3] *Model One-to-Many Relationships with Embedded Documents - MongoDB Manual*, https://www.mongodb.com/docs/manual/tutorial/model-embedded-one-to-many-relationships-between-documents/.

[4] *Model One-to-Many Relationships with Documents References - MongoDB Manual*, <https://www.mongodb.com/docs/manual/tutorial/model-embedded-one-to-many-relationships-between-documents/>.

[5] “Movielens.” *GroupLens*, 8 Dec. 2021, https://grouplens.org/datasets/movielens/.

[6] “MongoDB vs Mysql.” *MongoDB*, https://www.mongodb.com/zh-cn/compare/mongodb-mysql.

[7] Shitalpande. “What Is Document Database.” *Numpy Ninja*, Numpy Ninja, 29 Jan. 2022, https://www.numpyninja.com/post/what-is-document-database.