Comparative analysis of data representing models using e-commerce data*

Ivan Osiichuk[1,*,†], Nataliya Zagorodna[1,†], Dmytro Dmytriv[1], Inna Skarga-Bandurova[1,2]
Oleksandr Zaiets[3]

[1] Ternopil Ivan Puluj National Technical University, 56 Ruska str., Ternopil 46001, Ukraine
[2] Oxford Brookes University, Headington Rd, Headington, Oxford OX3 0BP, United Kingdom
[3] National Scientific Center "Institute of Agrarian Economics", 10 Heroiv Oborony str., Kyiv 03127, Ukraine

Abstract
The rapid growth in the volume, variety, and value of data has led to changing requirements for its storage, processing, and transmission. The problem of selecting data storage strategies that are optimally suited to the functional requirements and operational conditions is discussed in this paper. The article includes the comparison of 4 models of data representation: relational, non-relational, taxonomies, and knowledge graphs based on predefined criteria. To create a more holistic understanding of these models, an e-commerce example has been chosen. The mentioned models were examined using the Amazon Review dataset to demonstrate their advantages and disadvantages on real data.

Keywords
data representation model, relational model, non-relational model, taxonomy, knowledge graph

1. Introduction

Web applications are the backbone of the modern Internet, serving a wide range of purposes, including informational purposes, entertainment, education, e-commerce, etc. They accumulate a certain state, data of a different nature, and therefore, the task of storing such data is crucial. For the e-commerce industry, data storage is even more important, as this field produces a huge amount of information, which may contain the products sold on the marketplace, data on transactions performed by users, such as purchases, and much more. Many vendors provide the service in this field, and a much larger number of people use such services. Moreover, E-commerce hugely depends on information systems, particularly web applications that have databases as a foundation of data storage. Therefore, e-commerce is perfectly suited for comparing different data representation models used by such databases.

Considering a large set of different approaches to data storage and representation, the task of choosing a specific model is quite complex. This process includes understanding the key factors:

- The nature of existing data.
- The structure of the data, including the possibility of the data being semi-structured.
- The upstream and downstream of the data, i.e., the way the data enters a system and is further used.
- The capabilities of the data storage, including validation, restrictions, or potential data size.

*ITTAP'2025: 5th International Workshop on Information Technologies: Theoretical and Applied Problems, October 22-24, 2025, Ternopil, Ukraine, Opole, Poland
[1*] Corresponding author.
[†] These authors contributed equally.
osiichuk100@gmail.com (I. Osiichuk); zagorodna.n@gmail.com (N. Zagorodna); dmytrivd75@gmail.com (D. Dmytriv); skarga_bandurova@ukr.net (I. Skarga-Bandurova); ool.zaiets@gmail.com (O. Zaiets)
0009-0003-6844-2374 (I. Osiichuk); 0000-0002-1808-835X (N. Zagorodna); 0000-0003-3164-5832 (D. Dmytriv); 0000-0003-3458-8730 (I. Skarga-Bandurova); 0009-0004-7167-208X (O. Zaiets)

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2026-01-02
CEUR-WS.org/Vol-4146/paper08.pdf

Thus, choosing the right model and data storage directly affects the work of the system, further challenges, and is ultimately important for business success. To understand the variety and the circumstances in which a specific approach arises, it is worth looking through the timeline of the models considered and their fundamental principles.

Relational model

One of the most common data representation models is relational. It is a first-order predicate logic-based data management method introduced in 1969 by English computer scientist Edgar F. Codd, that uses relational algebra as a foundation. All data are stored as tuples, grouped into relations. Relational databases follow the relational model and organize data in tables with rows and columns [1]. An important component of systems that use the relational model is the ACID (atomicity, consistency, isolation, durability) principles. The idea behind these principles is to provide a highly reliable system that maintains a coherent and consistent data model and data itself in the face of many simultaneous requests, network problems, and query errors [2].

Non-relational model

Non-relational models have existed for a long time, since about the middle of the 20th century. The concept was used as a separate term in the 90s of the 20th century to refer to databases that do not use the relational model [3]. The data is stored in common information-representation formats such as JSON, allowing flexible schemas that change over time. The non-relational model does not have a unified theory, but is associated with the BASE (basically available, soft state, eventually consistent) principles, which were originally defined by Eric A. Brewer [4] and prioritize scalability and availability over strict consistency.

Taxonomy model

Taxonomies are a way to store data that is convenient for classification and categorization. The term was invented in 1813 by the Swiss botanist A. P. de Candolle and was used, among other things, for classifications in the natural sciences. This model organizes data hierarchically with parent-child relationships, so it is easy to create and understand hierarchy and convenient for classification tasks, yet it is not a good model to create relationships that do not fall into such connectivity.

Knowledge graph model

The knowledge graph (KG) is a flexible data model that uses a graph approach to describe real-world concepts and their relationships in an organized structure. Nodes represent items like things, people, or more general concepts, and edges show the connections between these entities, representing the semantic (carrying meaning) relation between nodes, how different subjects relate to each other, providing a technical basis and a convenient structure to work with knowledge [5]. Knowledge graphs are closely related to Semantic Web approaches, which are designed to make web data machine-readable by defining formal standards for data representation and data sharing [6].

2. Related work

Relational, non-relational, taxonomies, and knowledge graphs are the models that have existed for a long time, especially taking relational and non-relational approaches. As expected, there are many articles comparing these approaches based on various criteria.

Researchers often compare not exactly the models but rather specific implementations of relational and non-relational approaches. The article [7] compares relational and non-relational databases. The paper is devoted to an overview of the different types of such databases, providing a significant list of differences between these models. While the research provides quite a comprehensive comparison, it delves into the technical rather than conceptual details of each of the models. Still, the conceptual side is crucial while drafting and implementing the architecture of the software system. Having a single example area to display the data model application is useful for demonstrating the conceptual difference between the models more clearly, which helps to select a specific model based on the tasks that the specified area is used for and the data it processes.

Several articles [8-9] review relational and non-relational models in the field of web applications. The sources describe the use of these models when working with data in general, including big data and data engineering [10-11]. Taking into consideration the key role of these models for this area, the criteria for choosing one or another model. Although these two models are quite well covered, they are not well compared to taxonomies and knowledge graphs within a single scope example.

Referring to the other two models: taxonomies and knowledge graphs, they are often used in pairs for natural language processing and machine learning tasks [12-13]. This reflects well on the practical use of these models, but does not compare them with the other participants in the comparison in this article.

E-commerce is also often used as an example to compare or demonstrate the application of these models [14-15]. At the same time, the literature review did not expose any materials comparing all 4 models using examples from a single e-commerce system.

This article compares all 4 models on a selected example of an e-commerce system, identifying both the conceptual and the technical differences between these models, including the type and structure of data, purpose of the system, and organizational principles of each model, leading to a better understanding performance of such models.

3. Comparative analysis of the fundamental principles of each model

Before moving on to an overview of the data models representing e-commerce data, it is useful to review these models from a more conceptual perspective, defining the technical and fundamental basis of the selected models and their features, without reference to specific data. The purpose of this comparison is to highlight the distinctive characteristics, advantages, and disadvantages of each model. The comparison considers 4 different data presentation models: relational, non-relational, taxonomy, and knowledge graph. The analysis is based on a set of predefined criteria depending on the common data management and knowledge representation requirements:

- Structure – the fundamental organizational principle of data models
- Relations nature – how relationships between separate subjects are defined and managed
- Flexibility – the adaptability of the schema to evolving data models and handling different structures
- Performance – efficiency in querying and data operations.

Comparison results according to these criteria were summarized in Table 1.

Table 1
Comparison of the fundamental features of each data model

Criterion	Relational	Non-relational	Taxonomy	Knowledge graph
Structure	Tabular (tables with rows and columns)	Document, key-value, wide-column, or graph formats	Hierarchical (tree-like structure)	Graph-based (nodes and edges)
Relations	Foreign keys and joins	Varies by type, often implicit or embedded	Parent-child (is-a, part-of)	Explicit, typed edges with semantics
Flexibility	Rigid schema, low adaptability	High flexibility, schema-less or dynamic	Low flexibility, fixed hierarchy	High flexibility, can evolve dynamically

Performance	High for structured data and complex queries (with tuning)	High for large-scale, unstructured, or semi-structured data	Efficient for encompassing hierarchies	Optimized for relationship-heavy and semantic queries

All the above characteristics define the foundations of these data models. As can be seen from the table, high performance and constraints of the relational model perfectly fit data warehouse tasks, including BI and data analytics. High-flexible non-relational models often form the basis of data lakes with large amounts of semi-structured data used for research or ML workloads. Hierarchical structure of taxonomy is widely used in medicine and biological sciences. In turn, knowledge graphs are well-suited for AI tasks, enabling semantic reasoning and the extraction of new knowledge from existing data. These characteristics lead to the specific advantages and disadvantages described in Table 2.

Table 2
Advantages and disadvantages of each model

Data model	Advantages	Disadvantages
Relational	• Strong support for structured queries • Easy to enforce data integrity with constraints • Good performance for working with structured data • Mature ecosystem with robust tooling and optimization	• Poor at handling complex, hierarchical, or interconnected relationships (the complexity of queries may increase swiftly) • Schema rigidity makes it less flexible for evolving data models • Limited in representing semantic relationships beyond foreign keys
Non-relational	• Flexible schema allows easy handling of semi-structured data • Good for storing nested or hierarchical data • Easy scalability for large datasets	• Querying relationships between documents can be complex and inefficient • Relations between subjects may be implicit or complex • Often needs external tools for advanced analytics
Taxonomy	• Simple and intuitive for hierarchical organization • Useful for filtering and navigating large datasets • Easy to visualize and implement	• Does not support complex relationships or cross-category links • Limited semantic depth, as it often lacks contextual information • Inflexible when data does not fit neatly into a hierarchy
Knowledge graph	• Rich representation of relationships with semantic meaning • High flexibility in data modeling;	• More complex to integrate and maintain • Requires specialized tools (e.g., Neo4j, RDF triple stores)

	schema evolves dynamically • Efficient traversal of complex relationships • Excellent for reasoning, recommendations, and insights through inference	• Performance may degrade for certain large-scale queries

As you can see, each model is good for certain tasks and bad for others, which further emphasizes the importance of choosing the right model based on the tasks the system is planned to do.

4. Evaluation of the data models based on the Amazon Review Data dataset

Having defined the fundamentals and key features of the selected data models, we can apply these models to represent e-commerce data. An important feature of data in this area is the presence of a large amount of heterogeneous data, which has a different structure and nature, sometimes semi-structured and with complex relationships, thereby bringing functional requirements for the data models that represent such information.

To better understand the practical differences and strengths of the listed data models, we take a real dataset containing e-commerce data and present its data using each model, which allows us to understand the specific advantages and disadvantages of the models for this data. We use the Amazon review data dataset (2018) [16-17], which primarily contains product reviews, but was enriched with the metadata describing the products themselves. The dataset page provides an example of metadata with a product description (Figure 1), which is slightly simplified here.


```
{
  "asin": "0000031852",
  "title": "Girls Ballet Tutu Zebra Hot Pink",
  "feature": ["Botiquecutie Trademark exclusive Brand",
              "Hot Pink Layered Zebra Print Tutu",
              "Fits girls up to a size 4T",
              "Hand wash / Line Dry",
              "Includes a Botiquecutie TM Exclusive hair flower bow"],
  "description": "This tutu is great for dress up play for your little ballerina. Botiquecute Trade
Mark exclusive brand. Hot Pink Zebra print tutu.",
  "price": 3.17,
  "imageURL": "http://ecx.images-amazon.com/images/I/51fAmVkTbyL._SY300_.jpg",
  "imageURLHighRes": "http://ecx.images-amazon.com/images/I/51fAmVkTbyL.jpg",
  "also_buy": ["B00JHONN1S", ...],
  "also_viewed": ["B002BZX8Z6", ...],
  "salesRank": {"Toys & Games": 211836},
  "brand": "Coxlures",
  "categories": [["Sports & Outdoors", "Other Sports", "Dance"]]
}
```


Figure 1: An example of the product metadata from the Amazon review data dataset
data

The fields have the following description [16]:

- asin – product ID;
- title – product name;
- feature – list of the product features;
- description – product description;
- price – price (US dollars);
- imageURL – product image URL;

- imageURLHighRes – product image URL (high resolution);
- related – related products (including also_buy, also_viewed);
- salesRank – information about sales rank;
- brand – name of the brand;
- categories – the categories the product is related to.

4.1. Relational model

Using a relational model for such data, consider the following structure in Figure 2. It is worth noting that the metadata has nested data as values of certain fields (e.g., sales_rank, which is another JSON object), which is not a typical representation for the relational model, so this structure needs to be normalized and broken down into separate relations, as in the entity relationship diagram below, which form tables with foreign key relations according to Table 1.

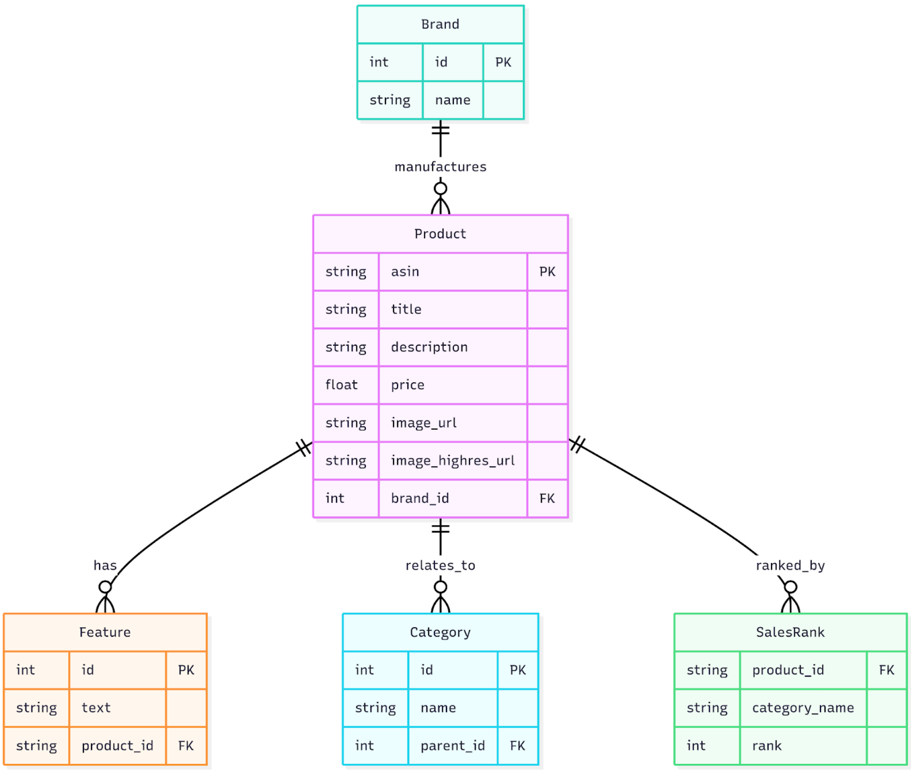

Figure 2: The ER diagram showing e-commerce data relations

Assessing the suitability of the relational model for such data, we can say that this approach ensures data consistency and allows us to detect the deviations of the data structure from the defined schema (schema anomalies) at the write step, but at the same time, this approach reduces the flexibility present in the metadata example. Possible changes in the data schema will be quite difficult to implement. implement. Such properties may be defined by the following advantages and disadvantages.

Advantages:

- Well-normalized structure enforces integrity.
- Optimized for structured queries (e.g., "Get all products under $5 by brand X").
- Scalable for large data sets using joins and indexes.

Disadvantages:

- Features must be stored as separate rows in a separate relation.
- Querying relations (e.g., also_viewed, also_bought) requires JOINs.
- Schema changes (adding new product fields) require migration.

4.2. Non-relational model

The JSON representation is natural for this model, so the structure in Figure 1 can be used without changes (Table 2).

Table 2
The data schema using the non-relational model

Field	Data type	Is always present
asin	string	true
title	string	true
feature	array<string>	true (length varies)
description	string	true
price	float	true
image_url	string	true
image_hd_url	string	true
also_buy	array<string>	false (length varies)
also_viewed	array<string>	false (length varies)
sales_rank	object({string, int})	false (length varies)
brand	string	true
categories	array<array<string>>	true (length and nesting vary)

The non-relational model is great for this kind of metadata structure, considering the possibility of nested fields and complex data structures, but this flexibility also has the downside represented by possible inconsistencies in the data schema. For example, incorrect or incomplete data can be stored, and thus, this approach requires more careful design of validation steps. Additionally, this model does not enforce clear and fixed relations (e.g., also_bought can have no direct relation to different products and can be just a list of strings), which negatively affects the performance of data operations. So, these features may be defined by the following advantages and disadvantages.

Advantages:

- Schema flexibility (e.g., features, images, ranks, and lists can vary per product).
- Easier to ingest and query with dynamic/optional fields.
- Suited for full document retrieval (getProduct("asin")).

Disadvantages:

- Implicit relationships (e.g., also_bought is just a string list).
- No referential integrity (risk of invalid asin references).
- Complex aggregations or filtering documents are less efficient.

4.3. Taxonomy

Considering taxonomies and hierarchical approaches to data storage, this model is not suitable for storing the full metadata (most of the data fields have a non-hierarchical nature), but it is perfect for storing the product categories (Figure 3) and can be used to represent similar hierarchical data.

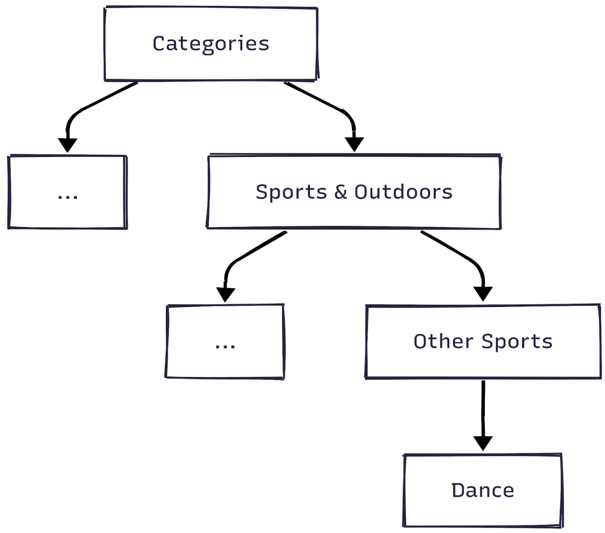

Figure 3: The product categories taxonomy

Taxonomy can be used as part of a search engine for filtering, or it can be used to store a category model to be referenced from other data models with the remaining metadata. For example, the product category taxonomy shown in Figure 3 can be stored as a separate model, providing quick and convenient access for other parts of the system that require this list of categories. Therefore, we can define the following advantages and disadvantages.

Advantages:

- Intuitive for classification-based navigation or filtering.
- Easy to group/filter products for category pages.
- Very lightweight structure.

Disadvantages:

- This model is not suitable for representing features, price, or other metadata.
- Not a complete data model – only works for hierarchical data.

4.4. Knowledge graph

Knowledge graphs are not well-suited for storing a large amount of identical data, but they are great for displaying a domain of knowledge based on the data. The presented metadata suits as a good starting point for knowledge graph construction, as it contains a list of features of each product, which can be used to build a knowledge graph showing various features users can specify in a search query, and links to products that have such features. Figure 4 provides a simplified visualized subgraph of the knowledge graph. Particularly, this part describes the product “Girls Ballet Tutu”.

Considering the diagram, the product itself refers to additional descriptive metadata, the brand in this case, but there may be additional nodes. It's also important to note that, taking the “Coxlures” brand node as an example, it may have many other products with a similar subgraph, so these linked nodes do not exist just for the needs of only that product but are interlinked components of the full graph.

Within this subgraph, we can see another green subgraph, which represents the hierarchy of categories with which this product is associated. Nodes that represent similar products are marked in blue. And the red subgraph defines the features of this product that are directly important when using such a graph to make a recommendation based on a list of requirements.

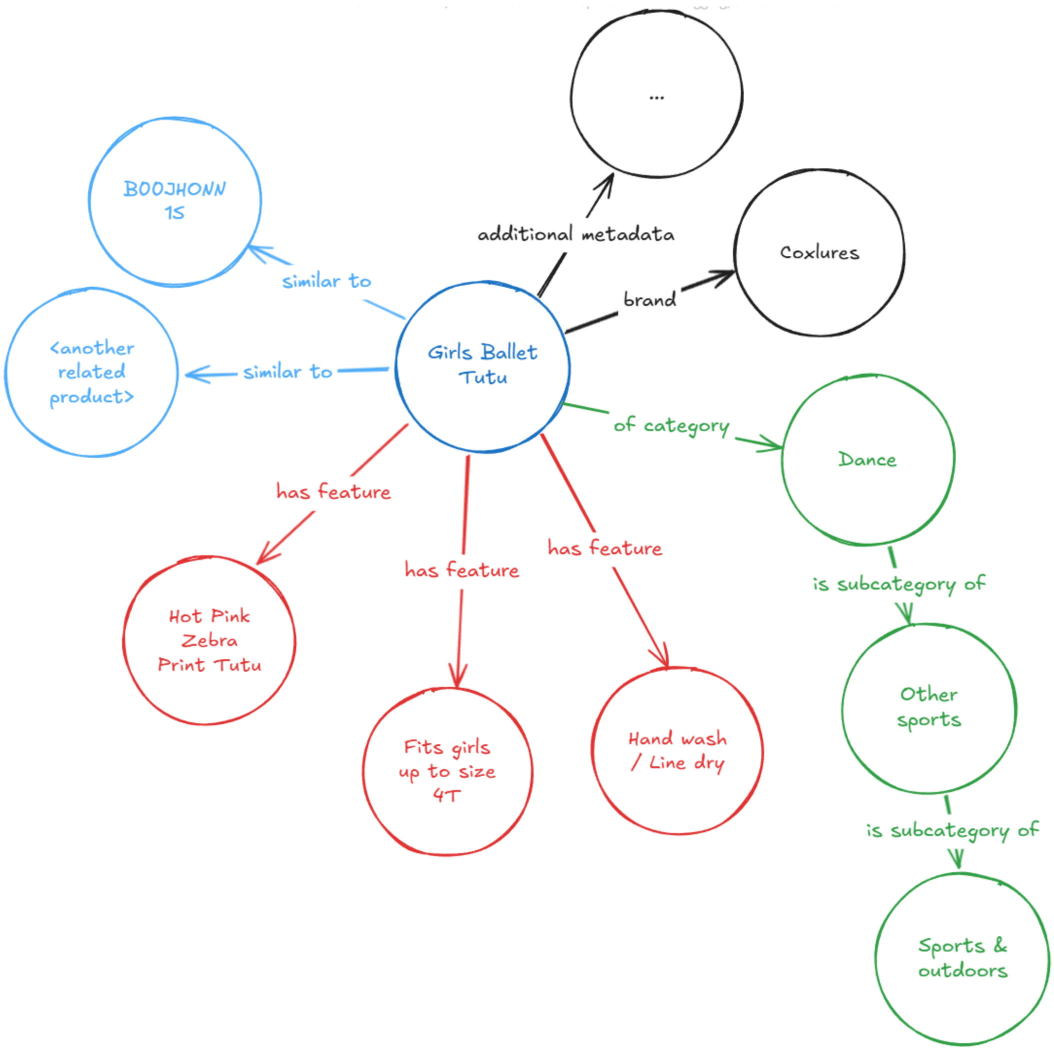

Figure 4: Subgraph of the KG describing the product “Girls Ballet Tutu”

Therefore, such a KG can be a good foundation for a recommender system to help a user choose a product based on a text description. This system can parse a search query into individual tokens containing a product description, features, and category, and then use this graph to identify products that have such characteristics. Examples of similar systems for different knowledge domains can be found across scientific publications [18-19]. When modeling data in a knowledge graph, it may be necessary to define a structure in accordance with the knowledge domain – an ontology [20] that provides a shared vocabulary and logical structure for how entities relate to one another for a particular scenario.

The following list of advantages and disadvantages can be defined for such a case.

Advantages:

- The ability to obtain insights based on the data
- Rich-typed semantic relations of the data subjects

Disadvantages:

- Creating a knowledge graph based on the described data can be resource-consuming and require a complex procedure of acquisition

5. Conclusion

The article compares four ways to organize data: relational and non-relational models, taxonomies, and knowledge graphs. The focus lies on what structure and rules they use, how they are formalized and organized, and for which part of the data is best suited. To better demonstrate the practical application of these models, the article represents real e-commerce data with each model, defining possible scope for a specific model (e.g., representing complete data or only a subset), advantages, and disadvantages of this model.

The relational model is great for handling organized tabular data that has fixed relation constraints with good performance at scale. They work well for transaction systems and reporting. However, they may be inefficient with complicated relations and non-persistent data structures. Considering e-commerce data, this approach is well-suitable for representing large amounts of data of different nature and origin as long as the data have a consistent, permanent schema. Therefore, it naturally reduces flexibility and is not compatible with semi-structured data.

Non-relational models are flexible enough to keep semi-structured data in formats like JSON. This data model is quite good for storing data with a dynamic schema. However, they are less efficient and more implicit at handling complex relationships between records. This model can be a good choice for a variety of e-commerce data. As shown in the example, the data often has a different structure and contains nested fields, and a non-relational approach widely supports such data features. At the same time, this model requires more careful validation, and subsequent schema inconsistencies can be present and detected only during further data manipulation if validation was not implemented well. Another important point to consider is the performance of data operations, which may not be as high as for relational models.

Taxonomies are strict in terms of relation types and provide less flexibility for different tasks. They are perfect for classification tasks, but less suitable for everything else. This model can be used to sort or arrange material. However, their rigid hierarchical structure and limited ability to capture complex relationships (only parent-child relations) make them less suitable for dynamic or overlapping data. Speaking of the application in e-commerce, it is only suitable for hierarchical data (e.g., product categories), for which taxonomies are a good choice and can simplify the work with such hierarchies and improve storage efficiency.

Knowledge graphs naturally best represent knowledge areas and semantic relationships between subjects. Their graph-based structure provides the possibility to define dynamic schemas, enables semantic reasoning, and efficient traversal of complex relationships. Compared to other data models, an important advantage is the ability to define generic relationships between entities. The model is abstract enough to represent different areas of knowledge and the relation between related entities. It makes them uniquely suited for search engines with semantic search, machine learning, and decision systems. In e-commerce, this model is not used to store raw data, but rather to extract insights about the data based on their relationships, such as making recommendations with a list of requirements. Within these tasks, the model is essential, though more complex to implement and operate.

After all, modern integrated data systems use combinations of these models. Therefore, the best choice is a hybrid system architecture, where different data models are used at different stages of data processing and for different tasks within a single data processing system framework.

Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT in order to: Drafting content, Generate literature review and Citation management. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] Codd, Edgar F. "A relational model of data for large shared data banks." *Communications of the ACM* 13.6 (1970): 377-387.

[2] Haerder, Theo, and Andreas Reuter. "Principles of transaction-oriented database recovery." *ACM computing surveys (CSUR)* 15.4 (1983): 287-317.

[3] Strauch, Christof, Ultra-Large Scale Sites, and Walter Kriha. "NoSQL databases." *Lecture Notes, Stuttgart Media University* 20.24 (2011): 79.

[4] Brewer, Eric A. "Towards robust distributed systems." *PODC*. Vol. 7. No. 10.1145. 2000.

[5] Bergman, Michael K., Michael K. Bergman, and Lagerstrom-Fife. *Knowledge Representation Practionary*. Springer International Publishing, 2018.

[6] Berners-Lee, Tim, James Hendler, and Ora Lassila. "The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities." *Linking the world's information: essays on Tim Berners-Lee's invention of the World Wide Web*. 2023. 91-103.

[7] Jatana, Nishtha, et al. "A survey and comparison of relational and non-relational database." *International Journal of Engineering Research & Technology* 1.6 (2012): 1-5.

[8] Gyorödi, Cornelia, Robert Gyorödi, and Roxana Sotoc. "A comparative study of relational and non-relational database models in a web-based application." *International Journal of Advanced Computer Science and Applications* 6.11 (2015): 78-83.

[9] Mihai, Gianina. "Comparison between relational and NoSQL databases." *Econ. Appl. Inform* 3 (2020): 38-42.

[10] Maowa, Jannatul. "A comparative study on big data handling using relational and non-relational data model." *International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol* 7 (2017).

[11] Ordonez, Carlos, Il-Yeol Song, and Carlos Garcia-Alvarado. "Relational versus non-relational database systems for data warehousing." *Proceedings of the ACM 13th international workshop on Data warehousing and OLAP*. 2010.

[12] Martel, Félix, and Amal Zouaq. "Taxonomy extraction using knowledge graph embeddings and hierarchical clustering." *Proceedings of the 36th Annual ACM Symposium on applied computing*. 2021.

[13] Shang, Chao, et al. "Taxonomy construction of unseen domains via graph-based cross-domain knowledge transfer." *Proceedings of the 58th annual meeting of the Association for Computational Linguistics.* 2020.

[14] Temaj, Granit. "A comparison between SQL and NoSQL databases in case of an e-commerce platform."

[15] Maalla, Allam, and Qiwei Jia. "Research on Key Technologies of E-commerce Big Data Analysis Platform." *IOP Conference Series: Earth and Environmental Science*. Vol. 242. No. 5. IOP Publishing, 2019.

[16] Ni, Jianmo. Amazon Review Data. University of California, San Diego, 2018, nijianmo.github.io/amazon/index.html.

[17] Ni, Jianmo, Jiacheng Li, and Julian McAuley. "Justifying recommendations using distantly-labeled reviews and fine-grained aspects." *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP).* 2019.

[18] Zhao, Xuejiao, et al. "Brain-inspired search engine assistant based on knowledge graph." *IEEE Transactions on Neural Networks and Learning Systems* 34.8 (2021): 4386-4400.
[19] Chicaiza, Janneth, and Priscila Valdiviezo-Diaz. "A comprehensive survey of knowledge graph-based recommender systems: Technologies, development, and contributions." *Information* 12.6 (2021): 232.
[20] Lypak, O. H., Lytvyn, V., Lozynska, O., Vovnyanka, R., Bolyubash, Y., Rzheuskyi, A., & Dosyn, D. (2018, September). Formation of Efficient Pipeline Operation Procedures Based on Ontological Approach. In Conference on Computer Science and Information Technologies (pp. 571-581). Cham: Springer International Publishing.