Object-oriented Information Systems August 2021

Home / / My courses/ / OOIS_PCOM7E August 2021 / / Unit 8 /

/ Collaborative Discussion 2: Alternatives to SQL / / Initial Post /

« Collaborative Discussion 2: Alternatives to SQL

Michael Botha

Initial Post

15 days ago

2 replies

Last 2 days ago

For more than 30 years the relational model has been the dominant representation applied in databases used in industry information systems (Connolly & Begg, 2014). The Relational Database Management System (RDBMS) is the core piece of software used to provision databases of this nature (Connolly & Begg, 2014). RDMSs provide the means to interface with the underlying storage hardware in an abstract way, enabling logical data structures known as relations to be defined in a tabular form, where columns have specified attributes that the associated values represent (Connolly & Begg, 2014). Rows represent each instance of a specific record, that an associated application saves to the database. Certain attributes of each relation collectively defined as unique and known as the primary key, are used in unison with those called foreign keys, which represent the same attributes in separately defined relations, to provide a means to set relationships between the data structures (Connolly & Begg, 2014).

In contrast to the relational model the Object-Oriented model for databases, implemented through an Object-Oriented Database Management System (OODBMS), allows data abstractions and their associated relationships to been defined using the same Object-Oriented paradigm commonly used in high-level programming languages (Connolly & Begg, 2014; Rao et al., 2018). OODBMS have historically been used to support interactions with larger volumes of data, as well as data that is more complex in nature than that traditionally stored in relational databases (Rao et al., 2018)

The two models have their respective advantages and disadvantages, and as with any technology there are proponents of each (Connolly & Begg, 2014; Rao et al., 2018). However, all can agree that the requirements of modern-day applications require more from their supportive databases than traditionally provided (Connolly & Begg, 2014; Rao et al., 2018).

References:

Connolly, t., Begg, c. (2014) *Database Systems: A Practical Approach to Design, Implementation, and Management*. Pearson Education available from: https://ebookcentral.proquest.com/lib/universityofessex-ebooks/detail.action?docID=5174902 [accessed 27 September 2021].

Rao, T., Haq, Ehsan., Khan Dost. (2018) Performance based Comparison between RDBMS and OODBMS. *International Journal of Computer Applications*. 180. 42-46.

Reply

Maximum rating: -

2 replies

1 Post by **Oliver Buckley**

4 days ago

Re: Initial Post

This is an interesting take on things, Michael. I am not so sure that I would class an Object Oriented Database as a NoSQL technology as at their heart they still rely on data being normalised and stored in interrelated objects (which mirror entities in an RDBMS).

I would have a look at something like MongoDB or Neo4J as examples of how different data management and storage can be with these technologies.

Reply

2 Reply to **Oliver Buckley** from **Michael Botha** ↑

2 days ago

Re: Initial Post

Thanks for your response Oli. I erred by reading the discussion instructions which state that Object databases are a type of NoSQL, and having an interest in the object-oriented topics we have covered, went straight on to perform comparative research on RDBMSs and OODBMs. I have, subsequent to your comment, browsed a couple of articles that mention OODMSs as having characteristics of the more modern NoSQL technologies (Krisciunas, 2014). Furthermore, according to Connolly and Begg (2014) definitions of what constitutes an OODBMS vary, which may cause some interpretive disparities, and therefore may be why it was mentioned as a NoSQL technology in the discussion instructions. Nevertheless, in my study of Object Databases I found some interesting points, like:

- The latest version of the SQL standard (SQL:2011) now includes control structures similar to standard high-level programming languages (Connolly & Begg, 2014).

- An issue with designing applications communicating with an RDBMS is the impedance mismatch of having a high-level language which the application is designed in, and a separate database query language, which causes significant extra coding for programmers (possibly up to 30%) (Connolly & Begg, 2014). Furthermore, because there are objects represented in the application, these typically need to be broken up when sent to the database for persistence, as the RDBMS tables will likely have separated various attributes into different relations (Connolly & Begg, 2014)
- OODBMSs try to move away from the conventional two-level storage model which uses transformation and type checking when moving data from local memory to a database, to a single-level model which doesn't have the in between process (Connolly & Begg, 2014).
- The relational representation is significantly based on a mathematical model (relational algebra) unlike Object-oriented Databases (Rao et al., 2018)

Following your suggestion to get a better perspective on NoSQL technologies, I read and viewed some intriguing content, and learnt the following key points:

- Object-oriented databases were originally developed as a solution to the impedance mismatch problem mentioned above, however they never took off as a technology, largely because many organisations were using their RDBMSs to integrate various applications (Fowler, 2012). Therefore, RDBMSs dominated the database space for an extended period (Fowler, 2012).
- The rise of the Internet meant large data requirements, and the need for database clusters where separate databases were integrated under a single DBMS (Fowler, 2012). RDBMSs were not designed for such scenarios, and in general did not scale very well (Fowler, 2012).
- With the requirement to increase developer productivity, it was necessary to remove the impedance matching issue which OODBMSs were originally designed to do (Fowler, 2012).
- The term NoSQL was actually a twitter handle originally created to represent a conference that took place in the 2000s (Fowler, 2012).
- The definition of what a NoSQL database comprises is somewhat unclear, however there are generally common characteristics of such (Fowler, 2012). For instance: they are non-relational; cluster-friendly (for the most part); often open-source; and have been a product of 21st century web requirements (Fowler, 2012).
- There are four main data models ascribed to NoSQL technologies, which are: the Key-value Store model, the Document model, the Wide-Column model, and the Graph model (mongoDB, N.D).
- The first three mentioned above are aggregate-oriented, where they enable the representation of complex data structures or objects as aggregations, unlike the normalised tables of relational databases (Fowler, 2012).
- In contrast to Relation and aggregate-oriented databases, Graph databases allow further granularity of data attributes, where various data structures are not combined but maintain various relationships allowing for fast and efficient interfacing with the many different data components (Fowler, 2012).
- NoSQL databases do not have a set schema, therefore can facilitate not only structured data, which is what RDBMS were designed for, but semi-structured and unstructured data too (Fowler, 2012).

- Structured data is data that needs to be able to fit into designated fields. For instance, zip codes, phone numbers, and credit card details (mongoDB, N.D). Unstructured data is data not able to fit into a pre-set model (mongoDB, N.D). Examples of this type is text, multimedia, emails, and so on (mongoDB, N.D). Semi-structured data is data which is generally unstructured but has some form of tags allowing basic classification and keyword searches (mongoDB, N.D).

- It is not likely that Relational databases will fall away in the future, but that polyglot persistence is performed in companies, where specific databases are allocated to the type of data predominantly processed by a particular application (Fowler, 2012).

As it pertains to the mongoDB database offering, I noted:

- It is a Document model NoSQL technology (mongoDB, N.D).
- Stores data in a semi-structured format as JavaScript Object Notation (JSON) objects, where the associated values can be any of a variety of data types (mongoDB, N.D).
- Provides a flexible schema (Genika, 2020).
- Enables large scaling of database systems (Genika, 2020).
- Supports data being distributed across multiple servers and regions, with intelligent geo-placing (mongoDB, N.D).
- Performs fast queries due to the underlying data model (Genika, 2020).

References:

Connolly, t., Begg, c. (2014) *Database Systems: A Practical Approach to Design, Implementation, and Management*. Pearson Education available from: https://ebookcentral.proquest.com/lib/universityofessex-ebooks/detail.action?docID=5174902 [accessed 27 September 2021].

Fowler, M. (2013) goto Conference: Introduction to NoSQL. Available from: **https://www.youtube.com/watch?v=ql_g07C_Q5I** [Accessed 14 October 2021].

Genika, Y. (2020) Data modeling with MongoDB. Available from: **https://www.youtube.com/watch?v=3GHZd0zv170** [Accessed 14 October 2021].

Krisciunas, A. (2014) Benefits of NoSQL. Available from: **https://www.devbridge.com/articles/benefits-of-nosql/** [Accessed 14 October 2021].

mondoDB. (N.D) What is NoSQL. Available from: **https://www.mongodb.com/nosql-explained** [Accessed 14 October 2021].

Rao, T., Haq, Ehsan., Khan Dost. (2018) Performance based Comparison between RDBMS and OODBMS. *International Journal of Computer Applications*. 180. 42-46.

Reply

Maximum rating: -

Add your reply

Your subject