Alternatives to SQL: Peer Responses

Course: MSc Computer Science

Module: Object-Oriented Information Systems

Assignment: ePortfolio

Date: Friday 23rd July 2021

Student ID: 126853

Peer Response 1:

In response to:

Reply to Michael Justus from Sergio Rafael Zavarce Caldera 12 days ago

Peer response

Before researching this topic I used to believe the same. Why would anyone keep paying high fees for Oracle´s services when there are so many options out there? And that is correct for start-ups or small companies with low database impact. But for those who have been using Oracle for a long time, the story is different.

Many companies are trapped in Oracle's net because of the contractual lock-in strategies, which can cost a fortune to get rid of. Moreover, for those companies is not only about the Databases. They usually have CRM, ERP, HR amongst other solutions that would need migration. On top of that, there is always the cost associated with software migration or database development to implement the new database system (Guarente, 2020).

References:

Guarente, C. (November 30, 2020). Is Oracle Still Relevant?. *Palisade*. https://palisadecompliance.com/is-oracle-still-relevant/

[Accessed 10 July 2021]

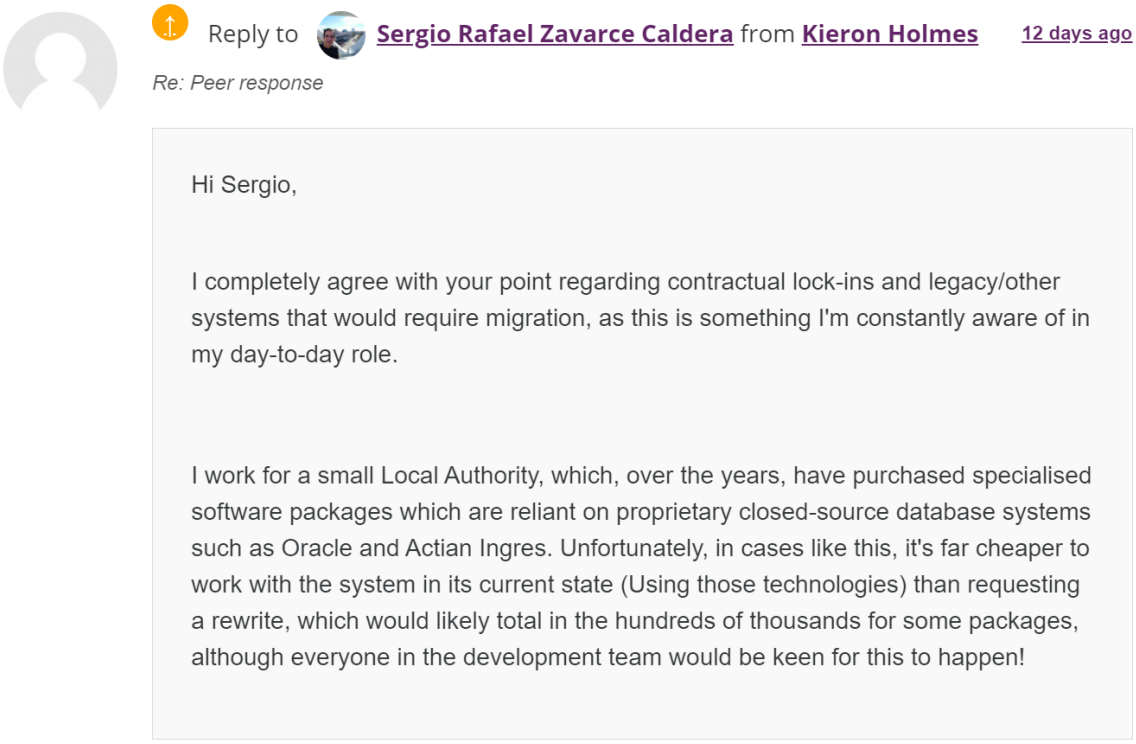
Post:

Hi Sergio,

I completely agree with your point regarding contractual lock-ins and legacy/other systems that would require migration, as this is something I'm constantly aware of in my day-to-day role.

I work for a small Local Authority, which, over the years, have purchased specialised software packages which are reliant on proprietary closed-source database systems such as Oracle and Actian Ingres. Unfortunately, in cases like this, it's far cheaper to work with the system in its current state (Using those technologies) than requesting a rewrite, which would likely total in the hundreds of thousands for some packages, although everyone in the development team would be keen for this to happen!

Screenshot:



Peer Response 2:

In response to:

Michael Justus

Initial Post

24 days ago

6 replies

Last 8 days ago

NoSQL databases are gaining popularity in recent years because of their claim to efficiently manage large volumes of data, support horizontal scalability, and be schema-agnostic (Fowler, 2015). Two main points stand out from a survey by Couchbase regarding relational databases: inflexible schemas and a lack of scalability (Lazzar, 2012). Problems NoSQL databases resolve because they adhere to Basically Available-Soft state-Eventual consistency (BASE) (Chandra, 2015). BASE differs from the traditional ACID principle (Atomic, Consistent, Isolated, Durable) in the degree of transaction support required of data. For instance, time-sensitive use cases requiring all-or-nothing commits are supported best by relational databases. Those that accept eventual consistency benefit from NoSQL databases (given their distributed nature).

Since NoSQL databases are non-relational and schema-agnostic, graph databases are interesting NoSQL databases because they maintain a graph of relations among *corresponding* data. In graph databases, relationships are first-class citizens and may contain data (much like an association class in UML). Such relationships allow users to think about the semantics of *how* data relates to other data—relational databases think about data sets, so users focus more on *schema* than *relations*.

NoSQL databases are not yet as mature as relational databases (6+ years versus 40+ years) (Chandra, 2015). Moreover, there exists no single query language for NoSQL databases. Neo4j introduced their Cypher query language--being adopted by more organisations (Francis et al., 2018); Couchbase has its query language called N1QL to traverse JSON documents (Ostrovsky et al., 2015); and Microsoft has extended standard T-SQL to support NoSQL-style graph queries using traditional relational structures (Microsoft, 2019). Overall, NoSQL databases are an essential tool in an organisation's data strategy, best suited for decentralised applications, multi-structure data, dynamic schemas. (Raj et al., 2018). Selection of *which* style of NoSQL to utilise (document, graph, key-value) ultimately will depend on the business use cases of data.

References

Chandra, D.G. (2015). BASE analysis of NoSQL database. Future Generation Computer Systems 52:13-21 DOI: https://doi.org/10.1016/j.future.2015.05.003

Fowler, A. (2015) NoSQL for dummies. New Jersey: John Wiley & Sons

Francis, N., Green, A., Guagliardo, P., Libkin, L., Lindaaker, T., Marsault, V., Plantikow, S., Rydberg, M., Selmer, P. & Taylor, A. (2018) Cypher: An evolving query language for property graphs. Proceedings of the 2018 International Conference on Management of Data 1433-1445. DOI: https://doi.org/10.1145/3183713.3190657

Lazzar, M. (2012) Couchbase Survey Shows Accelerated Adoption of NoSQL in 2012. Available from http://www.couchbase.com/press-releases/couchbase-survey-shows-accelerated-adoption-nosql-2012 [Accessed on 24 Jun 2021]

Microsoft (2019) Graph processing with SQL Server and Azure SQL Database. Available from https://docs.microsoft.com/en-us/sql/relational-databases/graphs/sql-graph-overview?view=sql-server-ver15 [Accessed on 24 Jun 2021]

Ostrovsky, D., Haji, M. & Rodenski, Y. (2015) The N1QL Query Language. Pro Couchbase Server. Berkeley, CA: Apress.

Raj, P. and Deka, G.C., 2018. A Deep Dive into NoSQL Databases: The Use Cases and Applications. Academic Press.

Bibliography

Moniruzzaman, A.B.M. & Hossain, S.A. (2013). NoSQL database: New era of databases for big data analytics-classification, characteristics and comparison. International Journal of Database Theory and Application 6(4). arXiv:1307.0191.

Simon, S., 2000. Brewer's cap theorem. CS341 Distributed Information Systems, University of Basel (HS2012).

Post:

Hi Michael,

I find your post very interesting, particularly the comparisons across NoSQL technologies. I have picked up on one quote regarding the lack of standardisation across different NoSQL variants.

"Moreover, there exists no single query language for NoSQL databases."

I find it quite interesting that there is little standardisation across the various NoSQL technologies, requiring an individual to specialise in a general product instead of the easily transferrable knowledge with standard SQL databases. A 2019 article produced by ScaleGrid shows that traditional SQL accounted for 60.48% of database usage, whilst NoSQL accounted for 39.52%.

The lack of language standardisation could lead to a skills shortage for particular NoSQL technologies. Do you believe that this is impacting the overall take-up of NoSQL within businesses? If standardisation was achieved, could you see NoSQL becoming the go-to database technology?

References:

Scalegrid. (2019) 2019 Database Trends – SQL vs. NoSQL, Top Databases, Single vs. Multiple Database Use. Available From: https://scalegrid.io/blog/2019-database-trends-sql-vs-nosql-top-databases-single-vs-multiple-database-use/ [Accessed 7th July 2021].

Screenshot: