

The Couchbase Blog

Couchbase, the NoSQL Database

[Topics](#)
[Specializations](#)
[Download](#)

The Value of Schema-Less Databases

Sandhya Krishnamurthy, Senior Solutions Engineer, Couchbase on January 7, 2016

Schema-less (or “schemaless”) databases are the latest buzz word in the IT world. Geek programmers seem to love the flexibility and low cost and these attributes have fired up many a start-up. Having come from a heavily relational database background the value that schema-less databases bring to an enterprise were eye-opening for me. It is time to take schema-less databases from the developer’s backyard and bring it to the wider enterprise.

This blog examines why it is so relevant in today’s data-centric world. If there is one single thing that enterprises would kill to do today, it is to understand and quickly elicit actionable insights from their data. They continue to invest millions of dollars to do it effectively. Big Data is a huge part of this equation. Gartner has immortalized the 3Vs of big data – *Volume, Variety and Velocity*. And if there is one area where traditional database systems struggle it is in keeping up with the velocity at which data is arriving, almost every tool in IT toolkit imposes some form of a data-model or format making it difficult to read or instantiate data quickly.

This is where schema-less databases step in to add tremendous value.

What is a Schema-Less Database?

Let us first define what a schema-less database is.

1. A schema-less database does not require conforming to a rigid schema (database, schema, data types, tables etc.) that one is required to live up to through the life of a system.
2. Does not enforce data type limitations on individual values pertaining to one single column type
3. Models the business usage and not a database schema, application or product.
4. Can store structured and unstructured data.
5. Eliminates the need to introduce additional layers (ORM layer) to abstract the relational model and expose it in an object oriented format.

Put in plain English fundamentally schema-less databases:

1. Do not require any modelling (3NF form of normalization) that has made the careers of lot of database architects including yours truly
2. Do not require pre-setting data types in your repository which greatly reduces the time required to stand-up a data repository
3. Can store data with different characteristics and can tolerate change to that definition without having to plan ahead for complex outages and changes and eliminates complex schema migrations
4. Can be easily transformed e.g. an Account Number could start as being all characters or it could be numbers or a combination of both. It is limited only by the user’s definition and imagination!

Search our blog



Subscribe to Blog via Email

Enter your email address to subscribe to this blog and receive notifications of new posts by email.

Subscribe

Recent Posts

[Full Text Search Indexing Best Practices by Use Case](#)

[Index Advisor Service for Couchbase N1QL\(SQL for JSON\)](#)

[Utilizing Multiple Disks to Speed Up Couchbase Analytics Queries](#)

[Extend Couchbase Analytics with SQL Server using CData](#)

[ASP.NET Core Microservices: Getting Started](#)

Blog Topics

Select Topic



Schema-less databases store data as Key/Value pairs (also known as KV) or as JSON documents.

Based on the use cases users have the choice to either store data as KV pairs or as JSON documents. JSON documents are generally very rich in the way data is represented and allows users to very closely model the entity relationship model that we are all very familiar with and have found very useful. For e.g.: An Account entity can be modeled in a JSON document with all the required attributes and nested values that go with a typical Account object – the multiple address, emails, aliases etc. JSON documents also provide the added advantage of being able to index individual values making the access much more performant, allowing pieces of data from different documents to be joined together.

This blog does not focus on the intricacies of a JSON document but just to provide a quick introduction a JSON document in a schema-less database world is much like a row in a relational database with the ability for say one Account row to be completely different from another Account row which is exactly how real life business data looks like.

Benefits of Schema-less Databases

What does all this mean to an enterprise that has been investing several millions of dollars into complex proprietary hardware and software database solutions to rein in the 3Vs? Let us first examine the amount of time enterprises spend in the above listed activities. In my mind the activities that are the most time consuming and expensive are

1. Database Design or Normalization

Designing the schema generally referred to as Normalization of data to get it to a relational format takes several weeks if not months and very skilled resources. In the relational world every piece of data belongs to a table, database or schema and stays there for the most part for the entire life of the product. Any change will require applications and users to create copies of this data which could cause anomalies and goes against the tenets of normalization.

Schema-less database eliminates this activity to a very large extent and greatly reduces the complexity of this activity. The only decisions that a user needs to pre-determine is which attributes of an entity they want to keep collocated.

2. ETL/ELT of required to reformat and store structured/unstructured data

Extracting data from sources and landing (commonly referred to as ETL or ELT) it unto a staging area into databases require the source and target to understand the structure of the data in a relational world. The user needs to know the database/table/column/field layout. There are limitations and performance implications when extracting the data. Unless the amount of data is very small there is really no way to extract and load this data into the database system without some kind of batch stream. And this usually translates to many hours or overnight processes and few thousands of line of code or expensive ETL tools. There is very little tolerance for error in this back to back process and any error usually results in not having accurate data on time. For an enterprise that depends on data for survival this is a huge setback and any enterprise that is not depending on its data and can tolerate latency probably does not care about competition. Attempting to meet latency requires huge capital and operational expenditure to acquire state of the art equipment. And this expense is ongoing and grows every year.

Schema-less database does not need long drawn transforming and cleansing processes due to the fact that the model is flexible and user is not pigeon-holed into conforming to a strict schema.

3. Ongoing change management including schema changes

that could cause massive code changes in systems (Application layer, ORM layer) that read and write data to and from the database

Once a schema is arrived at, changing the schema requires planned outages to get the affected objects offline and complete and test changes to the application layer to leverage the changes. Manual deployment errors could cause impacts that could run into several days or months. In a world where change is the only constant this model seems a little outdated. Schema-less databases model the business rather than a data-model. The definition and attributes of a piece of data is constantly changing so why live with a database that forces you to conform to the same definition?

Schema-less databases eliminates complex migrations activities and change synchronization due to its flexible data storage model.

4. Real-time analytics on streaming data

The need to be able to extract value from the data is of paramount importance in today's data driven world where companies are generating petabytes of data from different sources- some that are of great value and some that are not. Having to wait till the data is completely cleansed and loaded into the system to understand the value means mounting costs to store, cleanse and transform the data. A schema-less database lets you land, analyze and size up the value of data very quickly. If the data is not of any value one would want to fail fast and eliminate data with no value to cut storage and processing costs.

Some use cases like fraud detection, just in time marketing require the need to analyze data and provide insights in real-time. The needs are very tactical. There is a need to combine new data with existing data to provide insights. All this translates to big money and a huge competitive advantage. Today it is not just the tech savvy engineers accessing and analyzing data many of the users of this data are not highly technical.

A schema-less database lets you define your view of data rather than create a schema that you have to fight to extract value. This lets your non-technical IT dependent staff extract information quickly and easily and do what they are hired to do rather than understand complex data models and spend hours creating their universe of the data. This again translates to huge costs savings.

5. On demand scalability to meet extreme Volumes, Velocity and Variety of data

The days when enterprises had the comfort of planning ahead for any anomalies in the volumes, velocity and variety of data are sadly gone. If you don't scale you simply fail – as stark as it sounds this is really the truth as it costs millions of dollars in profits and lost opportunities.

Schema-less databases do not require complex /proprietary infrastructure that require huge capital/operational expenditure to scale. A small instance can be scaled very easily at the push of a button. While some traditional databases do offer this feature just the rigidity of the schema does impose limits. Scalability is not merely the ability to shard the data it is also the ability to adapt to changing schema quickly. Schema-less databases do this seamlessly. And all this can be deployed on cheap commodity hardware.

Summary

As described in this article it is clear that schema-less databases have a very important role to play in the current data-centric world. They are clearly here to stay in some form or the other. Does this mean the world is going to stop using traditional databases, stop ETL/ELT of their data and stop using proprietary solutions? While only time will answer that question, Wisdom tells me traditional data sources cannot continue to ignore the benefits that schema-less databases bring to the table. It is only a matter of time before they adopt some of these features of schema-less databases and this could be

a win-win for today's discerning customers.

This article has been written by Sandhya Krishnamurthy, Senior Solutions Engineer at **Couchbase**, a leading provider of Schema-less databases.

Contact the author at sandhya.krishnamurthy@couchbase.com

- Talk to us at Forums
- Follow us @couchbasedev and @couchbase

Visit the sites below to learn more about Couchbase products, for free product downloads and free training

www.couchbase.com

<http://www.couchbase.com/nosql-databases/downloads>

<http://training.couchbase.com/online>

Posted in: Best Practices and Tutorials **Tagged in:** schema-less databases, value of NoSQL databases

Posted by Sandhya Krishnamurthy, Senior Solutions Engineer, Couchbase

Sandhya Krishnamurthy is a technologist with a strong database development background and pre-sales experience. She is a part-time artist, part-time singer and full-time mom.

[All Posts](#)

2 Comments

New Idea: Integration With Logs and Events | My Way With Java

DECEMBER 17, 2018 AT 11:05 AM

[...] entries can be stored in (scalable) schema-less (NoSQL) database like [...]

[Log in to Reply](#)

How to Maximize the Value Out of Hadoop? | .Net Updates

NOVEMBER 2, 2019 AT 12:26 AM

[...] design: Hadoop is schema-less in nature by design, but that doesn't completely exclude it from the discussion around [...]

[Log in to Reply](#)

Leave a reply

You must be logged in to post a comment.

Contact



COMPANY

ABOUT

LEADERSHIP

NEWS & PRESS

CAREERS

EVENTS

PRICING

CONTACT US

SUPPORT

DEVELOPER PORTAL

FORUMS

PROFESSIONAL SERVICES

SUPPORT LOGIN

TRAINING

QUICKLINKS

BLOG

DOWNLOADS

GET STARTED

ONLINE TRAINING

RESOURCES

WHY NOSQL

SECURITY

GDPR

We use cookies to offer you a better browsing experience, analyze site traffic, personalize content, and serve targeted advertisements. Read about how we use cookies and how you can control them by clicking "Cookie Settings." By continuing to browse or by clicking "Accept All Cookies", you consent to our use of cookies. [Cookie Policy](#)

[Cookie Settings](#)

Accept All Cookies

