USE CASE OF NOSQL

Once only used by the likes of Google, Amazon and Facebook, many industries are now adopting NoSQL database technology for crucial business applications, replacing their relational database deployments to gain flexibility and scalability. Here are 10 enterprise use cases best addressed by NoSQL:

* **Personalization.** A personalized experience requires data, and lots of it – demographic, contextual, behavioral and more. The more data available, the more personalized the experience. However, relational databases are overwhelmed by the volume of data required for personalization. In contrast, a distributed NoSQL database can scale elastically to meet the most demanding workloads and build and update visitor profiles on the fly, delivering the low latency required for real-time engagement with your customers.

Profile Management. User profile management is core to Web and mobile applications to enable online transactions, user preferences, user authentication and more. Today, Web and mobile applications support millions – or even hundreds of millions – of users. While relational databases can struggle to serve this amount of user profile data as they are limited to a single server, distributed databases can scale out across multiple servers. With NoSQL, capacity is increased simply by adding commodity servers, making it far easier and less expensive to scale.

* **Real-Time Big Data.** The ability to extract information from operational data in real-time is critical for an agile enterprise. It increases operational efficiency, reduces costs, and increases revenue by enabling you to act immediately on current data. In the past, operational databases and analytical databases were maintained as different environments. The operational database powered applications while the analytical database was part of the business intelligence and reporting environment. Today, NoSQL is used as both the front-end – to store and manage operational data from any source, and to feed data to Hadoop – as well as the back-end to receive, store and serve analytic results from Hadoop.

* **Content Management.** The key to effective content is the ability to select a variety of content, aggregate it and present it to the customer at the moment of interaction.NoSQL document databases, with their flexible data model, are perfect for storing any type of content – structured, semi-structured or unstructured –

because NoSQL document databases don’t require the data model to be defined first. Not only does it enable enterprises to easily create and produce new types of content, it also enables them to incorporate user-generated content, such as comments, images, or videos posted on social media, with the same ease and agility.

* **Catalog.** Catalogs are not only referenced by Web and mobile applications, they also enable point-of-sale terminals, self-service kiosks and more. As enterprises offer more products and services, and collect more reference data, catalogs become fragmented by application and business unit or brand.Because relational databases rely on fixed data models, it’s not uncommon for multiple applications to access multiple databases, which introduces complexity and data management challenges. By contrast, a NoSQL document database, with its flexible data model, enables enterprises to more easily aggregate catalog data within a single database.

* **Customer 360° View.** Customers expect a consistent experience regardless of channel, while the enterprise wants to capitalize on upsell/cross-sell opportunities and to provide the highest level of customer service. However, as the number of products and services, channels, brands and business units increases, the fixed data model of relational databases forces enterprises to fragment customer data because different applications work with different customer data. NoSQL document databases use a flexible data model that enables multiple applications to access the same customer data as well as add new attributes without affecting other applications.

* **Mobile Applications**. With nearly two billion smartphone users, mobile applications face scalability challenges in terms of growth and volume. For instance, it is not uncommon for mobile games to reach tens of millions of users in a matter of months.With a distributed, scale-out database, mobile applications can start with a small deployment and expand as the user base grows, rather than deploying an expensive, large relational database server from the beginning.

* **Internet of Things.** Today, some 20 billion devices are connected to the Internet – everything from smartphones and tablets to home appliances and systems installed in cars, hospitals and warehouses. The volume, velocity, and variety of machine-generated data are increasing with the proliferation of digital telemetry, which is semi-structured and continuous. Relational databases struggle with the

three well-known challenges from big data IoT applications: scalability, throughput, and data variety. By contrast, NoSQL allows enterprises to scale concurrent data access to millions of connected devices and systems, store large volumes of data, and meet the performance requirements of mission-critical infrastructure and operations.

* **Digital Communications.** In an enterprise environment, digital communication may take the form of online interaction via direct messaging to help visitors find a product or complete the checkout process. And as with mobile text messaging, the application may need to support millions of website visitors. Relational databases are limited in responsiveness and scalability while NoSQL databases, thanks to their distributed architecture, deliver the sub-millisecond responsiveness and elastic scalability that digital communication applications require.

* **Fraud Detection.** For financial service organizations, fraud detection is essential to reducing profit loss, minimizing financial exposure and complying with regulations. When customers pay with a credit or debit card, they expect immediate confirmation. The process impacts both the enterprise and its customers. Fraud Detection relies on data – detection algorithm rules, customer information, transaction information, location, time of day and more – applied at scale and in less than a millisecond. While relational databases struggle to meet this low latency requirement, elastically scalable NoSQL databases can reliably deliver the required performance..