With the data explosion across every industry, storing all of your enterprise's Big Data in a Data Lake is the right answer. Data Lakes easily provide both linear and modular scalability. Moreover, they unlock the value of unstructured data while significantly reducing overall infrastructure costs.

To ensure a smooth, swift, and successful deployment of your Data Lake – and to maximize delivery of business value from it – IT, Sales, Marketing, Finance, Supply Chain, and other business leaders should ensure that their teams are aware of the fundamental differences between traditional data approaches today and leveraging data through the Data Lake tomorrow.

Navigating the Shift from Traditional Relational Database Management (RDBMS) to a Data Lake

Source: Matt Turck / FirstMark

Traditional RDBMS ***Data Lake***

From our experience, we have assembled the table below with the goal of empowering leaders in Information Technology and their Sales, Marketing, Finance, Supply Chain, and other business partners with perspectives on the paradigm shift driven by a Data Lake.

Dimension	*Traditional RDBMS*	*Data Lake*	*Kaizen Viewpoint*
Query Execution	Query execution for properly indexed tables are tuned to a certain extent by the RDBMS for better performance with the following limitations: • Tables cannot grow beyond certain size • Re-indexing live tables is costly	Data is stored as files in a distributed file system called HDFS. Since queries are executed on each HDFS node, query execution is slower than RDBMS on the HDFS systems. • Queries are split into several map-reduce functions as a query plan • Execution happens within the edge node and not at a central location • All data transformations happen at each data node	***Data Lakes will need additional tools on top of traditional Hadoop / Yarn HDFS to ensure ease of querying.*** ***For column-based data structures, systems like HBase or Impala add the ability to query Data Lakes and respond in real time.***

Dimension	*Traditional RDBMS*	*Data Lake*	*Kaizen Viewpoint*
Caching	Traditional RDBMS systems come with several levels of in-memory / fast access stores for caching: • Queries • Execution plans • Data	Queries get queued onto the HDFS in a system like Hive. By design, indexing does not exist in HDFS. Caching can be configured; however, cache invalidation criterion can cause full table scans each time a query is executed. With this, when some active users are executing queries, queries from others get queued. As a result, execution timings may be high to a point they impact resource productivity.	***Tools like Apache Spark provide the caching with speed. Spark enables in-memory datasets (RDDs) as a first-class construct. Multiple stages of reads and writes can happen purely in-memory without accessing disk. By storing lineage information about data transformations, they provide fault tolerance even for in-memory datasets through re-computation.***
Transactionality (ACID) & CRUD Data Operations	RDBMS Systems are built on transactionality concepts where each transaction is • Atomic • Consistent • Isolated • Durable In Create, Read, Update, and Delete (CRUD) operations performed on data, create, update and delete are transactional operations. In traditional RDBMS systems, these operations are basic concepts and are done at row level.	Data updates are possible only on some row/columnar file formats like RC (Row Columnar), ORC (Optimized Row Columnar), Parquet etc. • Columnar file formats have the flexibility of reading and updating the row level data with good performance, but they compromise on the write performance • If the data has constant changes to the schema, ORC must incur high performance costs to update the entire schema with the new column added • Uncompressed CSV files are fast to write but they lack column-orientation and are slow for reads. Updating records would mean deleting current records and adding new files to the system • Not all Big Data deployments support all file formats	***Data lake implementers should be aware of the requirement of the underlying table.*** ***Choosing the optimal file format in Hadoop is one of the most essential drivers of functionality and performance for Big Data processing and query in a Data Lake.***

Dimension	*Traditional RDBMS*	*Data Lake*	*Kaizen Viewpoint*
Application Programming & Business Analytics Implementations	Capabilities inherent in RDBMS approaches allow for a degree of flexibility in Application programming and Analytics tools • Since RDBMS optimizes query plan, application programmers can typically write queries without thinking much about the performance implications of a query • Analytics resources can use tools to point to the database and run both queries and analysis without much application programming knowledge	Application programmers must match coding style to Big Data distributed environment. This requires understanding of responsibilities of Data Layer and Programming Layer. While RDBMS can be forgiving, Big Data environments have revived the importance of data driven design. A few key areas to focus on are: • Adopting iterative design • Defining the platform • Establishing sources and integration requirements • Focusing on data collection, assembly, and feedback	***Analytics and Operations Research teams should have or be supported by strong technical expertise in Data Lake implementations to ensure a smooth and successful delivery of your Data Lake.***
Performance Testing	During implementation, a Performance Improvements phase can run in parallel with Functional phases.	Projects typically require a Performance Tuning phase. Big Data applications often require additional focus on performance testing • Must keep track of system resources	***Performance rework can be minimized with the "right" upfront Data Lake design and with an understanding of implications from design choices.***
Integration into Tools through APIs	Nearly all analytics tools provide integration into all RDBMS systems.	Integration is evolving and not fully "there" yet • Example: Microsoft Excel integration through Power Query integration is very specific to version of Microsoft Excel, and this is not fully developed yet • Need to use tools like Paxata, AtScale for integration into existing BI tools and EDW platforms • Need to use HiveJDBC/ODBC in accessing DataLake from Application layer	***Given the multitude of Big Data tools, knowing the right tool to use and how to integrate it will be crucial for ensuring a smooth and successful delivery of your Data Lake.***
Expertise / Skill Availability	Large pools of resources are available to support initiatives.	Large pools of resources who claim to have required skill sets are available, but typically these resources come with RDBMS baggage.	***Having resources with the right skill sets is the #1 key for Big Data success.***

Kaizen Analytix: Masters of the Big Data Lake

Kaizen Analytix has a wealth of experience helping companies quickly generate insights from their own and industry data sources, with a technology-agnostic perspective.

This experience brings our company into direct contact with Big Data and Data Lakes, and we are now considered authoritative sources on this subject.

In working with clients on Data Lake projects, we have successfully…

- Served in both leadership and support/facilitator capacities
- Designed and implemented not only Data Lakes, but also the advanced analytical models that deliver business insights from it
- Served in these capacities in industries as diverse as Automotive, Media & Entertainment, Retail and Consumer Goods
- Worked alongside client IT teams, vendors, and system integrator firms – we are flexible in any model
- Helped clients identify and incorporate new external data sources – some of which our clients are not aware of – into the Data Lake, and then leverage those data sources for predictive and prescriptive analytics