William Blair

Jason Ader +1 617 235 7519
jader@williamblair.com

Billy Fitzsimmons +1 312 364 5112
bfitzsimmons@williamblair.com

Sebastien Naji +1 212 245 6508
snaji@williamblair.com

Database Software Market: The Long-Awaited Shake-up

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Contents

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Key Findings

Exploding volume of data + changing nature of applications + cloud adoption + open source = database market disruption. Ninety percent of the world's data was created in the past two years alone, according to *Forbes*. At the same time, lines of business are increasingly demanding a broader set of applications that can harness data to drive digital transformation. Many of these applications are cloud-native, and a growing percentage are being built on a microservices architecture. All of these applications must be supported by an underlying database, which increasingly is based on open-source software. The confluence of these trends is driving disruption, fragmentation, and innovation in the roughly $30 billion operational database management system (ODBMS) market.

Proliferation of database types has driven market confusion. To keep pace with the surging volume of data and changing nature of applications, the database itself has undergone a transformation. This has fueled the emergence of a plethora of database types and vendors over the past several years geared toward specific use-cases. However, the different places data can be stored (on-premises, cloud, hybrid) and different ways data can be stored (multiple database models) have led to hype and market confusion, though the dust appears to be settling as customers learn which products/vendors work best for different use-cases.

It's all about the use-case. Our research indicates that the historical distinction between relational databases (where data is organized by rows and columns) and nonrelational databases (where data is organized in means other than rows and columns) is blurring. Specifically, nonrelational databases (e.g., MongoDB) can increasingly address transactional use-cases like order processing, which rely on a system of record (the historical purview of relational DBs). Conversely, some relational DBs (e.g., Google Cloud Spanner) can address use-cases like gaming, which require distributed scaling (the historical purview of nonrelational DBs). Over time, this suggests that the religious debate on relational versus nonrelational will give way to a more pragmatic database selection process centered on use-case suitability.

Secular shift toward nonrelational DBs, but death of relational DBs is greatly exaggerated. We expect continued rapid adoption of nonrelational DBs alongside the boom in unstructured data creation and next-generation, cloud-native applications that demand the speed, flexibility, and scalability inherent in a nonrelational DB. Put simply, nonrelational DBs are better aligned with modern applications, agile software development, and the burgeoning DevOps ecosystem. Yet we do not believe this will eliminate the need for relational databases, which will continue to be heavily utilized for mission-critical, transactional applications (as well as for complex query data warehousing) and for which there remains substantial organizational inertia. To put this in context, IDC forecasts that relational DBs will still account for more than 80% of the total operational database market by 2022, though we suspect that this prediction is too conservative with respect to nonrelational DB adoption.

DBaaS (database-as-a-service) is becoming table stakes. The DBaaS model has taken the market by storm in recent years as it frees up developers from self-hosting and managing what is arguably the most complex and difficult-to-manage layer of the application stack (the database). Instead of worrying about managing and tuning the database (and its underlying infrastructure)—which is outsourced to the DBaaS provider—developers are able to focus on building applications with the speed and agility necessary in today's information-driven economy. Sold as a fully managed, subscription service by a CSP (cloud service provider) or independent database vendor (running on a public or private cloud), a DBaaS virtualizes the database from the application, allowing the database to be run and managed independent of the application (this is especially useful for microservices-based applications). For DBaaS vendors, a primary appeal is the ability to monetize free usage of open-source database software.

Multimodel databases are taking hold. To address market fragmentation and the resulting customer confusion, vendors have introduced multimodel databases, which incorporate multiple database structures in the same package, enabling data to be represented for numerous use-cases simultaneously (with the choice of mode based on what is most appropriate for the specific application being addressed). As noted above, these general-purpose databases can even offer both relational and nonrelational capabilities in the same platform. While this one-size-fits-all, one-throat-to-choke approach will appeal to certain users, we still see room for specialized databases such as graph and time-series to be successful given their uniqueness and unmatched performance for specific use-cases.

Open-source databases becoming more closed. The developer-centric nature of the operational database market explains in large part the popularity and proliferation of open-source offerings (about 170 open-source databases out there today, according to website DB-Engines), and commercial open-source vendors rely on the "freemium" model to spur adoption of their paid solutions. However, the ability of cloud providers to develop commercial DBaaS offerings from open-source projects like MongoDB, Redis, and PostgreSQL has spurred a backlash among open-source vendors. These vendors believe that CSPs are unfairly monetizing open-source software while contributing little to the community. As a result, several open-source vendors, including MongoDB and Redis Labs, have attempted to institute more restrictive licensing models to make it more difficult for CSPs to sell services based on their open-source technologies—although there is still no consensus in the open-source community for how to deal with so-called CSP strip mining.

Streaming data will increasingly be used for operational purposes. Historically, streaming data (e.g., machine-generated data) was used only for analytical purposes, but with advances in memory and chip processing, is now possible to process streaming data in real-time. With the emergence of internet of things (IoT), more data will be created in real-time, and enterprises will want and need to harness this real-time data to deliver personalized experiences to customers and extract instantaneous insights for strategic decision-making and business efficiency. For example, a package delivery company could meld real-time traffic and weather with customer pickup requests to optimize the routes taken by its delivery vehicles.

Growing convergence of transactional and analytical databases. Historically, the real-time operational/transactional database field was distinct from the post-process analytical database/data warehousing realm. The customer workflow required data from the transactional database to be moved to the data warehouse for analysis. This is beginning to change with the ability to perform analytics on-board the transactional database (in part due to the performance gains from in-memory database technology), eliminating the need to physically move the data. This approach offers significant benefits to end-users in terms of real-time analysis and decision-making as well as cost reduction.

CSPs and database insurgents are taking market share. The competitive landscape for database software is evolving rapidly with cloud-native databases, upstart independents, and incumbent providers battling for market share. The appeal of cloud-native databases is their simplicity, as developers can work in the same environment in which the application is located. The upstarts' value proposition is best-of-breed technology (and often lower cost, given open-source heritage) and multicloud portability. The incumbents meanwhile argue that neither the CSPs nor the upstarts can match the maturity and feature set of their tried-and-true software, especially for mission-critical, transactional applications.

The database remains sticky; but if you lose the app, you lose the database. Despite rising competitive intensity in the database market, we expect the incumbent DBMS vendors to continue to control the lion's share of the market in the near term. This is because existing applications in an organization are tied to traditional databases. While there are some database migration initiatives inside companies, and migration tools continue to improve, it remains expensive and cumbersome

to migrate existing applications to a new, underlying database. That said, we believe the long-term losers from the database market disruption will be the incumbent players that lack a strong IaaS platform (e.g., Oracle and IBM) and thus are poorly positioned to capture new applications (and their corresponding databases).

Vendor consolidation is inevitable … and necessary. With venture capital and private equity investors needing an exit from the $4.1 billion in capital they have invested in database companies over the past 10 years, and with CSPs co-opting the technology of many of the upstarts, the survival of more than a handful of stand-alone, upstart database companies is dubious. We expect perhaps four to five of these companies will reach escape velocity, with the rest getting swallowed up by bigger players, executing mergers of equals, or fading away.

MongoDB has emerged as the most viable next-generation database player. MongoDB's highly successful IPO in October 2017—the first IPO of an operational database player in more than 20 years—was a milestone event that heralded the tectonic shifts occurring in the database market (please see our companion MongoDB coverage initiation report: *Developer-Friendly Database Still Scratching the Surface; Initiating Coverage With Outperform*). With calendar 2018 revenue of $267 million, the company has established critical mass and separated itself from the crowded field of upstart vendors. Other fast-growing contenders with near- to medium-term IPO potential include DataStax, MarkLogic, Redis Labs, Couchbase, Neo4j, EnterpriseDB, MemSQL, and InfluxDB.

Introduction

If data is the fuel of the digital economy, the database is a critical part of the engine, allowing users to create, modify, retrieve, query, and ultimately organize the data tied to an application. Underneath every application lives a database, which explains the strategic value and historical stickiness of this market.

After having seen little innovation since the 1970s (the dawn of the relational database), the $30 billion ODBMS market has kicked into high gear over the past several years with groundbreaking new technologies and a slew of new vendors disrupting the status quo.

The main drivers of disruption, in our view, have been cloud computing and open-source software, which together have revolutionized how applications are built, where they are located, and how they interact with their underlying data. Because of the inherent application-database linkage, the upheaval in the application ecosystem has naturally precipitated an upheaval in the database market. Taken together with the rise of open-source software—which has catalyzed and democratized software development, while dramatically reducing the cost of software—the result has been a proliferation of database types and vendors.

From a customer perspective, however, the downside of all this innovation has been a multiplicity of choices, and a chaotic vendor landscape with incumbents, upstarts, and cloud service providers (CSPs) all vying for attention. To make matters more complicated for customers, there is often a disconnect between developers and IT operations inside an organization, with developers increasingly selecting the database that best fit their needs without regard for IT policies and standards.

The good news, in our view, is that the database market appears to have moved past much of the hype and confusion, and has entered a period of greater clarity and maturity that will spur strong growth in the years ahead. While we expect the inevitable vendor consolidation characteristic of a market in transition, we believe CSPs together with next-generation database players (led by MongoDB) will accelerate market disruption, creating fresh opportunities for public investors in a historically stagnant space.

It is fair to say that it has taken longer for the database market to change than most people expected, likely due to the aforementioned application-database stickiness. Yet it feels like we are finally at the proverbial tipping point, with the database market as wide open—and still as strategic—as it has ever been.
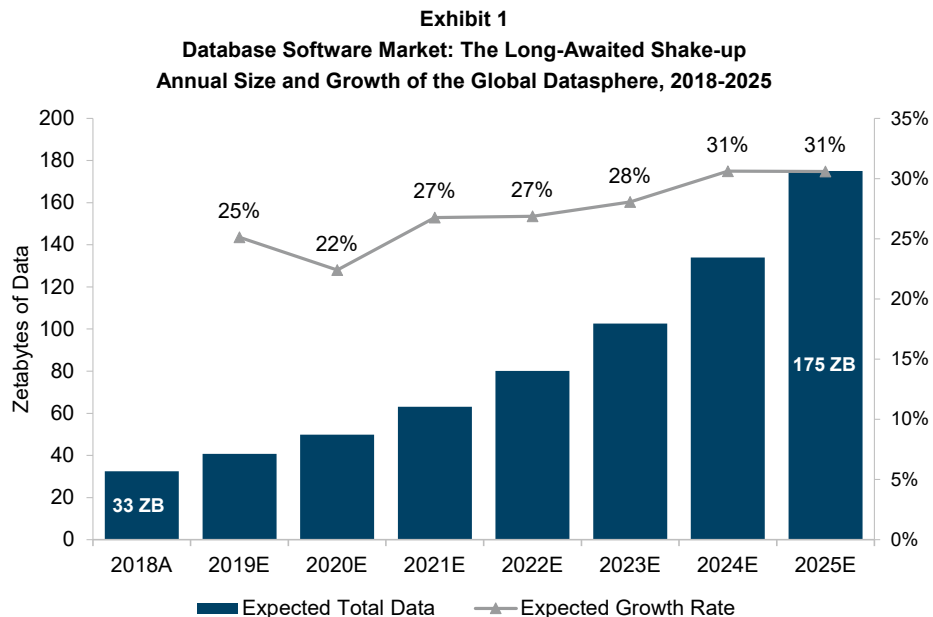
Scope of Study

In this report, we aim to shed light on the evolving market for database management system (DBMS) software, including key trends, technologies, and vendors that are shaping the future. Our report is focused on the operational part of the DBMS market (where the database is tied to a live application) versus the analytical or data warehousing segment of the market (which deals with processing and analyzing data imported from various sources). While we concede that the distinction between operational and analytical databases may increasingly be blurring, given the need for real-time responses based on analytics, a deep dive into the analytical segment of the market is beyond the scope of this report.

The Importance of the Database

Lines of business are demanding a broader, more modern set of software applications that allow them to deeply engage with their customers, optimize their businesses operations, inform strategic decision-making, and ultimately, better compete in their respective industries. Each of those applications must be supported by a database, in whatever form that database exists, whether it is cloud based, on premises, and/or delivered as a service. The proliferation of applications is driving a proliferation in databases.

Every software application requires a database to store, organize, and process data, and the database directly influences a particular application's performance, scalability, flexibility, and reliability. For this reason, the selection of a database is a highly strategic decision impacting application effectiveness and organizational competitiveness. In addition, as developers re-platform existing applications, they have the opportunity to reevaluate the underlying database platform that the application is built on to ensure that it will support the functionality required today and is flexible enough to adapt to future requirements.

Exhibit 1
Database Software Market: The Long-Awaited Shake-up
Annual Size and Growth of the Global Datasphere, 2018-2025

Source: Worldwide Global DataSphere Forecast (2019-2023), IDC; and Data Age 2025, IDC-Seagate White Paper for 2024 and 2025 estimates
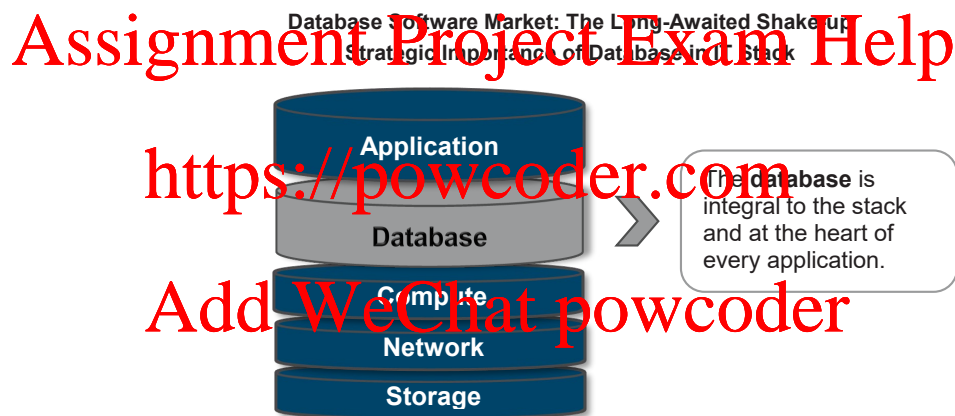
At the same time, we are witnessing an explosion of data. IDC forecasts that the global "datasphere" will grow from 33 zettabytes in 2018 to 175 zettabytes in 2025. A key driver here will be IoT devices, which IDC projects will create over 90 zettabytes of data in 2025 (or about half of the global datasphere). Accompanying this explosion in data volume is an expansion in the variety of data, including data with different structures, often called semi-structured data, and new patterns of data, such as time-series data.

Much of today's economy relies on data, and this reliance will only increase in the future as companies capture, catalog, and try to glean insights from data in every step of their supply chain; enterprises collect vast sums of customer data to provide greater levels of personalization; governments collect and use huge volumes of data for law enforcement and intelligence purposes; and consumers integrate social media, entertainment, cloud storage, and real-time personalized services into their lives.

While this data boom is enabling extraordinary user experiences and business opportunities, it is also delivering unprecedented challenges in data storage and access that must be overcome with new database technology. It is against this backdrop that we have seen a fundamental shake-up in the historically staid operational database market.

Exhibit 2
Database Software Market: The Long-Awaited Shake-up
Strategic Importance of Database in Stack

Application

Database

Compute

Network

Storage

The **database** is integral to the stack and at the heart of every application.

Sources: William Blair

Database Market History

The database market is rapidly evolving with its history generally defined by two distinct waves of technology. The first wave began in the 1970s and can be characterized by the growth and dominance of relational database offerings from the likes of Oracle, Microsoft, and IBM. The second wave, which began to pick up momentum about 10 years ago, was sparked by the introduction of nonrelational/NoSQL offerings and ushered in a flood of new vendors and innovation. The database popularity ranking compiled monthly by DB-Engines now covers more than 300 DBMSs, though we expect the market to undergo significant Darwinian rationalization and consolidation over time with a handful of natural winners emerging.

In the late 1960s, a mathematician working at IBM named Edgar Codd developed the theoretical underpinnings that would launch the relational database model. At its most fundamental level, the relational database that he developed described how data should be represented logically such that inconsistency and redundancy was eliminated. The general concept was that the order of records on disk, the presence of indexes, and the way in which related data was linked should not affect

the way in which a user or application might query this data (i.e., the database schema or structure should be disconnected from physical information storage). Building on some of this earlier work, Andreas Reuter and Theo Härder coined the acronym ACID (Atomicity, Consistency, Isolation, and Durability) in 1983, developing a model to characterize relational databases to ensure that all users had the same view of the data at any instance.

However, while the relational DBMS (RDBMS) worked well to store and manipulate structured data (i.e., data that is highly organized such as patient medical records), its shortcomings became apparent when dealing with unstructured data, which has grown exponentially over the last decade. Unstructured data is any data that resides in emails, files, or documents. In 1998, Berkeley computer scientist Eric Brewer developed what became known as the CAP (Consistency, Availability, and Partition Tolerance) theorem to describe the deficiencies of the ACID model. It posited that in the presence of a network partition, one has to choose between consistency and availability. In other words, if a network connection between two geographical locations were to be lost, an ACID-compliant database must fail in one of those two regions to maintain strict consistency.

In the late 2000s, to address the shortcomings of relational databases, NoSQL databases (nonrelational databases) were born to allow the database to continue operating in the presence of such a networking partition by sacrificing the strict consistency criterion. This type of nonrelational database followed a new set of rules known as the BASE system, where the strict consistency criteria gave way to eventual consistency such that a system becomes consistent over time rather than at the time of input. Since its inception, the nonrelational database model has led to a surge in database types for storing unstructured and semi-structured data, including key-value, column-oriented, document, time-series, and graph databases.

Exhibit 3
Database Software Market: The Long-Awaited Shake-up
A Brief History of the Database (1960-Present)

1970 - 1979	1980	1990 - 1999	2010 - 2017	2017
E.F. Codd publishes paper proposing use of relational database model. RDBMS becomes premier model with Ingres (developed at UC-Berkeley) and System R (created at IBM) becoming learning prototypes.	Oracle IPO	Advent of the Internet leads to exponential growth for databases. Increased use of cgi, MySQL, and Apache bring open source solutions to Internet.	Proliferation of NoSQL database companies continues as new use-cases and applications are developed using unstructured data.	MongoDB IPO

2000 — mySQL acquired by Sun Microsystems

1960 | 1970 | 1980 | 1990 | 2000 | 2010 | Present

1960 - 1969	1980 - 1989	2000 - 2009	2017 to Present
Computerized databases begin, with IBM's SABRE system becoming a commerical success by helping American Airlines manage reservation data.	SQL becomes standard query language. Relational database systems become commercial success. DB2 becomes flagship product for IBM.	Need for different database models to handle growth in unstructured data leads to development of NoSQL/nonrelational databases.	Upstart NoSQL database companies begin to mature. CSPs begin to disrupt the database market. DBaaS offerings become standard fare.

Sources: William Blair

Market Definitions

Database: A collection of information organized and indexed in digital form that can be easily accessed, managed, updated, and queried by an external user or application.

Database Management System (DBMS): Software that manages a database and defines how the data is stored and accessed, and how data within the database is related. The DBMS effectively serves as an interface between the database and the end-user or application program, ensuring that data is consistently organized and easily accessible.

Types of DBMSs: For the purposes of this report, we define DBMSs across two dimensions: 1) business use-case—i.e., operational or analytical, and 2) data structure—i.e., relational or nonrelational.

Operational DBMS (ODBMS): Encompasses relational and nonrelational DBMS products that sit underneath business applications and support transactional data. ODBMSs are used for two main types of transactional data: data of record and transient data. Data of record are typically tied to a packaged business application (e.g., ERP, CRM, security event management) or a custom-built transactional application and needs to be stored for future reference or analysis. Data of record is the traditional purview of relational databases. Transient data, on the other hand, is also tied to an application, but is more ephemeral in nature. It only exists to aid the operation of the application but does not necessarily need to be recorded for future analysis (e.g., data interactions tied to an airline ticket search). As a result, data consistency ("single source of truth") is less of a requirement with transient data compared with data of record. Transient data is most often associated with nonrelational databases.

Analytical DBMS (ADBMS): Involves relational and nonrelational DBMS products tied to data warehousing, business intelligence, and data science analytics on data from various sources. An ADBMS has no direct linkage to an application and typically developers are not involved in its operation. An ODBMS is often the source for a data warehouse and needs to be moved from the ODMBS to the ADBMS before analysis can commence. As noted earlier, the ADBMS market is beyond the scope of this report.

Relational DBMS (RDBMS): Developed by IBM in the 1970s, an RDBMS is fundamentally designed to support tabular data (i.e., data organized in rows and columns). The simplest analogy would be an Excel spreadsheet, where each column is a specific field of data in the database, and each row is an entry in that table. Technically speaking, all relational DBMSs require a predefined schema (structure) before adding records to the database and use the standard structured query language (SQL) for accessing and querying records.

Key benefits of a RDBMS are maturity, reliability, transactional guarantees, and ability to support complex queries. Major drawbacks are scalability (most RDBMSs were not designed to scale out due to a shared disk architecture, though we are starting to see the emergence of horizontally scalable relational databases) and flexibility (on account of its predefined schema requirement). This makes RDBMSs ill-suited for next-generation, cloud-native applications (such as customer experience apps) that are defined by large quantities of unstructured and semi-structured data where developers want the flexibility to modify the database with changing business requirements.

More specifically, the rigid structure of relational databases, where data is stored in tables of rows and columns, makes it costly and time consuming for developers to build, maintain, and update applications as required—even simple schema changes can be complicated. For example, developers are often required to spend significant time fixing and maintaining the linkages between modern applications and relational databases. In addition, the volume and variety of data today does not fit

easily into this predetermined row-and-column format, making it difficult and inefficient for developers to work with these applications, reducing application functionality, causing poor application performance, and risking costly application downtime.

Common RDBMS use-cases include traditional applications, ERP, CRM, and e-commerce. Examples of popular RDBMSs include Oracle Database, Microsoft SQL Server, IBM DB2, SAP Hana, Amazon Aurora, Amazon RDS, Azure SQL Database, EnterpriseDB (PostgreSQL), MySQL (owned by Oracle), and MemSQL.

Nonrelational DBMS (NDBMS): Foundationally designed to support large sets of unstructured, semi-structured, and distributed data, an NDBMS does not require a schema to be predefined because of its metadata structure (i.e., the format of the data is encapsulated with the data rather than predefined externally). Technically speaking, this is referred to as schema on-read, and allows a user to just drop the data into the database and add attributes on the fly. With unstructured data volumes having already surpassed total structured data generated globally, the appeal of NDBMS will only rise.

An NDBMS is also known as a NoSQL (not-only-SQL or non-SQL) DBMS because of its flexibility to use other programming languages besides SQL. There are multiple NDBMS types, defined by how their data is organized or represented, including document store, graph DBMS, key-value store, time-series DBMS, and wide column store (see definitions below). There has also been a strong movement toward multimodel databases (see below), which incorporate multiple modes of data representation in one package.

Key benefits of a NDBMS are ease of use, flexibility, and horizontal scaling to accommodate huge quantities of data with limited performance tradeoffs (NDBMSs employ a shared-nothing, distributed architecture). NDBMSs have taken off especially with application developers who want to deliver features and functions rapidly (often in the cloud) and do not want to spend time on rigid schema definitions up-front. In addition, because NDBMSs are typically built with APIs, developers can easily execute queries without having to learn SQL or understand the underlying architecture of their database. Lastly, NDBMSs tend to be significantly less expensive that traditional RDBMSs, especially as many of these NDBMSs are based on open-source software.

Major drawbacks of an NDBMS are lack of transactional guarantees, limited features (for instance, they lack the ability to join multiple tables in a single query), and relative immaturity compared to RDBMSs. Most NDBMS do not offer the guaranteed consistency and reliability for transactions (known as ACID) inherent in a relational DB, though vendors like MongoDB and Redis Labs now offer ACID compliance. ACID defines a set of properties that guarantees the integrity and validity of the data even in the event of errors or power failures.

Common NDBMS use-cases include webscale, IoT and mobile applications, DevOps, social networking, shopping carts, and recommendation engines. Examples of popular NDBMSs include MongoDB, Amazon DynamoDB, Amazon DocumentDB, Amazon Neptune, Azure CosmosDB, Google Spanner, DataStax, Neo4j, Couchbase, MarkLogic, InfluxDB, and Redis Labs.

We note that in our report we are leveraging IDC's market data for the DBMS market, but have excluded analytical databases from our analysis and combined IDC's separate classification of the NDBMS (primarily addressing mainframes) and DDMS (primarily NoSQL DBMSs) segments into one category that we call NDBMS. Please see our market size analysis for more details.

Exhibit 4
Database Software Market: The Long-Awaited Shake-up
Primary Database Types and Use Cases

	Operational	Analytical
Relational Database	**Key Applications**: ERP, CRM, credit card processing, e-commerce, and other data of record applications	**Key Applications**: Data warehousing, business intelligence, and data science
	How is Data Stored: Tables (rows and columns)	**How is Data Stored**: Tables (rows and columns)
	Popular Products: Oracle Database, Microsoft SQL Server, IBM DB2, SAP Hana, Amazon Aurora, Azure SQL Database, EnterpriseDB (PostgreSQL), MySQL, MemSQL	**Popular Products**: Oracle Exadata, Oracle Hyperion, Teradata, IBM Netezza, IBM dashDB, Amazon Redshift, Microsoft SQL Data Warehouse, Google BigQuery
	Pros: Transactional guarantees/data consistency, limitless indexing, large and mature ecosystem	**Pros**: Consistency of information and calculations
	Cons: Rigid schema definitions, cost, mainly vertical scaling, difficult to use with unstructured/semi-structured data	**Cons**: IT professionals need to maintain; data response in minutes instead of milliseconds like operational databases
Nonrelational Database	**Key Applications**: Web, mobile, and IoT applications, social networking, user recommendations, shopping carts	**Key Applications**: Indexing millions of data points, predictive analytics, fraud detection
	How is Data Stored: Multiple data structures (document, graph, column, key-value, time series)	**How is Data Stored**: Hadoop needs no inherent data structure; data can be stored across numerous servers
	Popular Products: MongoDB, Amazon DynamoDB, Amazon DocumentDB, Azure CosmosDB, DataStax, Neo4j, Couchbase, MarkLogic, Redis	**Popular Products**: Cloudera, Hortonworks, MapR, MarkLogic, Snowflake, DataBricks, ElasticSearch
	Pros: Ease of use, flexibility (no need for pre-defined schema), horizontal scaling (to accommodate massive data volumes), generally low cost (open source)	**Pros**: Good for batch processing, large files, and parallel scans; mainly open-source, so cost efficient
	Cons: Lack of transactional guarantees, limited querying features, relative immaturity	**Cons**: Slow response times; not good for fast lookups or quick updates

Source: William Blair

Exhibit 5
Database Software Market: The Long-Awaited Shake-up
Database Market Competitive Landscape

Source: William Blair

Types of Nonrelational Databases

Document Store: Maps data to documents not to rows and columns, with the data held in a hierarchical, tree-like format. This offers great flexibility to add semi-structured and unstructured data to the database. Documents are essentially a representation of an object in software programming. Each document in this type of database has its own data, and its own unique key, which is used to retrieve the document. Because document DBs typically describe data using web-centric interchange formats like JSON or XML, and these formats allow for easy mapping to web applications, they are a popular and intuitive choice for application developers. Common use-cases for document DBs include content management, personalization, and mobile applications. Popular products here include MongoDB, Amazon DocumentDB, Azure CosmosDB, MarkLogic, and Couchbase.
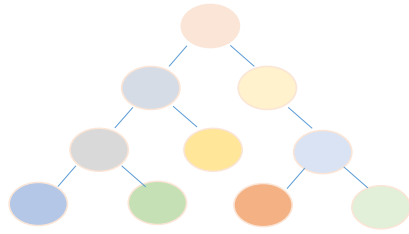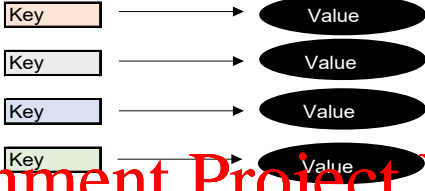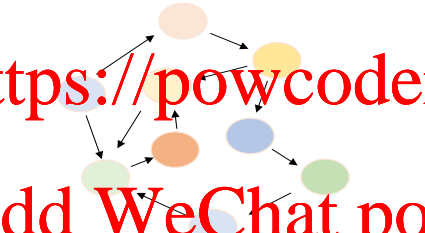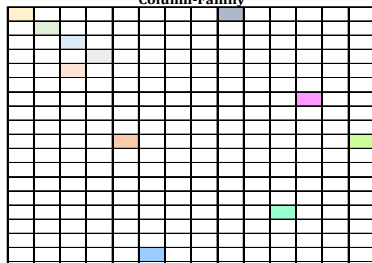
Key-Value Store: Stores data as a collection of key-value pairs in which a key serves as a unique identifier for the data. Both keys and values can be anything, ranging from simple objects to complex compound objects. Key-value databases are highly partitionable and allow horizontal scaling beyond what other types of databases can achieve. Common use-cases include session-oriented web applications, real-time bidding, shopping carts, and customer preferences. Popular products here include Amazon DynamoDB, Azure Table Storage, Redis Labs, Oracle NoSQL Database, Aerospike, Riak, and Oracle Berkeley DB.

Graph DBMS: Designed for use-cases where interconnected relationships among the data are as important as or more important than the individual data points. Architecturally, graph DBs use nodes to store data entities (person, thing, category, or other piece of data) and edges to store relationships between nodes. Nodes are similar to the objects familiar to software developers. Graph databases have some of the flexibility of key-value stores but also offer full support for relationships. Traversing the relationships is very fast because the relationships between nodes are not calculated at query times but are persisted in the database. This allows graph databases to deliver unmatched performance for applications with large amounts of connected data. Common use-cases include real-time recommendations, fraud detection, privacy, optimization engines, social networking, network and IT operations, and identity and access management. Popular products here include Neo4j, Amazon Neptune, Titan, and TigerGraph.

Time-series DBMS: A database purpose-built for collecting, synthesizing, and deriving insights from metrics and events or measurements that are time-stamped. Time series data are simply measurements or events that are tracked, monitored, downsampled, and aggregated over time. This could be server metrics, application performance monitoring, network data, sensor data, events, clicks, trades in a market, and many other types of analytics data. Time-series DBs optimize time series storage by removing duplicate information and grouping data by the producing device. DBMSs with time series support also include time-specific functions, such as range aggregations, time-based intersections and unions, time series filters, and smoothers. Common use-cases include IoT applications, DevOps, and industrial telemetry. Popular products here include InfluxDB, Amazon Timestream, and TimescaleDB.

Column Store: Organizes data into columns. Groups of these columns, called column families, have content and function similar to tables in relational databases. The main differences from RDBMSs are that column stores do not have relationships between rows and they support flexible schema definitions. These traits make them popular for storing semi-structured data, such as log or clickstream data that demand high performance and a highly scalable architecture. Column stores are also called wide-column stores, columnar databases, column-family database, or column-oriented DBMS. Popular products here include Apache Cassandra, HBase, Google BigTable, and HyperTable.

Exhibit 6
Database Software Market: The Long-Awaited Shake-up
Nonrelational Database Types

Document-based Store	**Document** 	*Data stored in JSON or XML object using hierarchical tree-like structure* Vendor Example: MongoDB Use-Cases: Content management, personalization mobile apps Advantage: Scale, ease-of-use
Key-Value Store	**Key-Value** 	*Every item stored as attribute name (or key) together with its value* Vendor Example: Redis Labs Use-Cases: session-oriented web apps, real-time bidding, shopping carts, Advantage: Performance at scale
Graph-based	**Graph** 	*Use nodes to store data entities (person, thing, category) and edges to store relationships between nodes* Vendor Example: Neo4j Use-Cases: Real-time recommendations, fraud detection Advantage: Connected data perforr
Time Series	**Time Series** 	Time	Time Series ID	Value
13:51:00	101	0.01
13:51:03	102	1.16
13:51:07	101	0.04
13:51:11	101	0.08
13:51:22	103	4.18
14:51:34	104	4.41
15:51:38	103	5.02
16:51:44	105	0.88		*Collect and synthesize time-stamped metrics and events* Vendor Example: InfluxDB Use-Case: Server metrics, sensor data Advantage: Ideal for IoT devices
Wide Column-based Store	**Column-Family** 	*Data organized into colums (no rows), supports flexible schema definition* Vendor Example: DataStax Use-Cases: Log or clickstream data Advantage: Performance at scale

Source: William Blair

Exhibit 7
Database Software Market: The Long-Awaited Shake-up
Nonrelational Database Vendors and Products, By Type

Document-based Store

Key-Value Store

Graph-based

Time-Series

Wide column-based Store

Multi-Model Database

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Source: Corporate Filings, Company Press Kits, and William Blair

Multimodel Database

Over the last several years, the sheer number of new vendors and database types has led to customer confusion and indecision, spurring the development of the multimodel database. Put simply, a multimodel database is a general-purpose database that enables data to be represented for multiple use-cases simultaneously, with the choice of data model based on what is most appropriate for the specific application being addressed. As noted above, data models (e.g., SQL, document, time-series, key-value, graph) represent how the data is stored within a database, which is important when considering the application/use-case.

In the past, a customer often had to use multiple databases in the same project. For example, in building an e-commerce application, the customer might use a relational database to persist structured tabular data such as customer information; a document store for unstructured object-like data, such as order history or product catalog; a key-value store for the shopping cart; and a graph database for highly linked referential data, such as product recommendations. This often led to operational friction (more complicated deployment, more frequent upgrades) as well as data consistency and duplication issues. The multimodel database aims to reduce such friction and reduce the number of vendors and products with which a customer needs to work.

As the database market has matured, single model vendors have gravitated toward multimodel DBs to address a broader set of use-cases, simplify the customer experience, and increase their strategic relevance to customers. Some of this feels like marketing, but several vendors including MongoDB, Microsoft (with Azure CosmosDB), Amazon DynamoDB, MarkLogic, Redis Labs, OrientDB, and ArrangoDB are seeing traction with their multimodel products.

Exhibit 8
Database Software Market: The Long-Awaited Shake-up
Multimodel Database
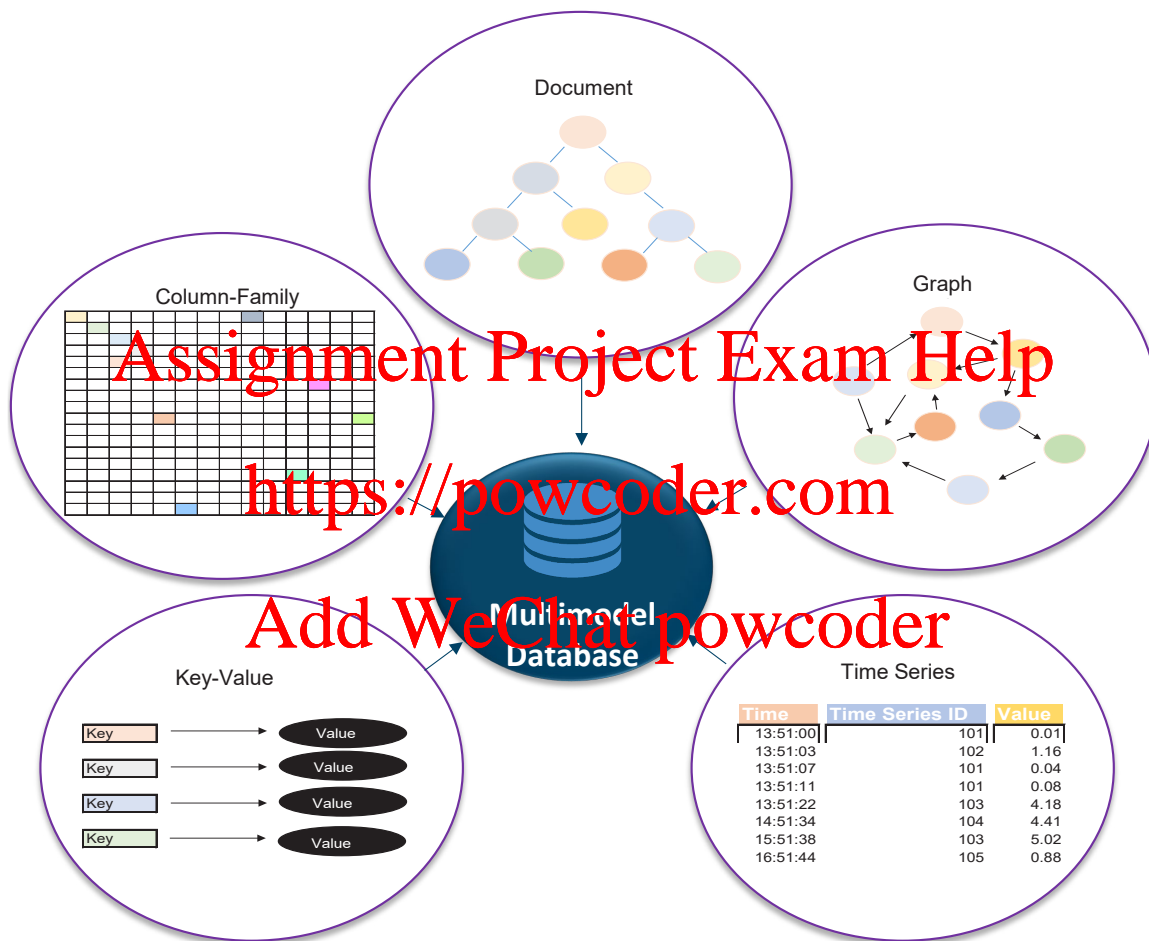
Source: William Blair

Within the multimodel trend, we are also starting to see a convergence of relational and nonrelational technologies. Specifically, some RDBMS products include features traditionally associated with nonrelational products (such as horizontal scaling and high availability), which allows the RDBMS to address a broader set of use-cases. Conversely, some NDBMS products include features associated with relational products (such as ACID transaction support), which removes traditional barriers of adoption for nonrelational DBs around serving mission-critical, transactional applications. This suggests that the historical distinction (and religious debate) between relational and nonrelational DBs will blur over time in favor of use-case suitability.

It is important to note that virtually every multimodel DB still has a native mode, with the other models essentially bolted on. For example, MongoDB is a document store natively while Redis is a key-value store natively. The question is are multimodel DBs "good enough" for most tasks or are there still use-cases where the way the data is modeled really matters. In other words, is there still a role for special purpose databases or will one-size-fits-all multimodel databases squeeze the stand-alone products out of the market over time?

Our research suggests that graph databases have the best chance to survive and thrive as a distinct category (versus the other NoSQL models) because connected data applications present serious performance problems that only a specialized graph DB can solve. Given the plethora of use-cases for connected data and the increasing business demand to extract insights in real-time from connections in data, this positions graph databases well for the future, though it is unclear how big the addressable market is for stand-alone graph databases.

We note that recent data from website DB-Engines, which ranks database management systems according to their popularity on a monthly basis, show an explosion in popularity in the graph DBMS model, especially in the past two years. In addition, several vendors, including Neo4j and Oracle, are working on creating a standard graph query language (today there are multiple different GQLs), which if approved by the standards body, would be the first database query language to be standardized since SQL. This would further make the case for the graph DB to be recognized and sustained as a separate category.

The other data model that in our view has a fighting chance of stand-alone success is the time-series DBMS, especially with the explosion in IoT data and the need to have strong tools for processing and analyzing time-stamped data. According to DB-Engines the time-series DBMS has seen the highest percent improvement in popularity over the past 12 months of any data model. Again, however, the concern here would be how big the addressable market is for stand-alone time-series databases.

Exhibit 19
Database Software Market: The Long-Awaited Shake-up
DBMS Popularity Broken Down by Database Models: January 2013 to Present

Sources: DB-Engines.com
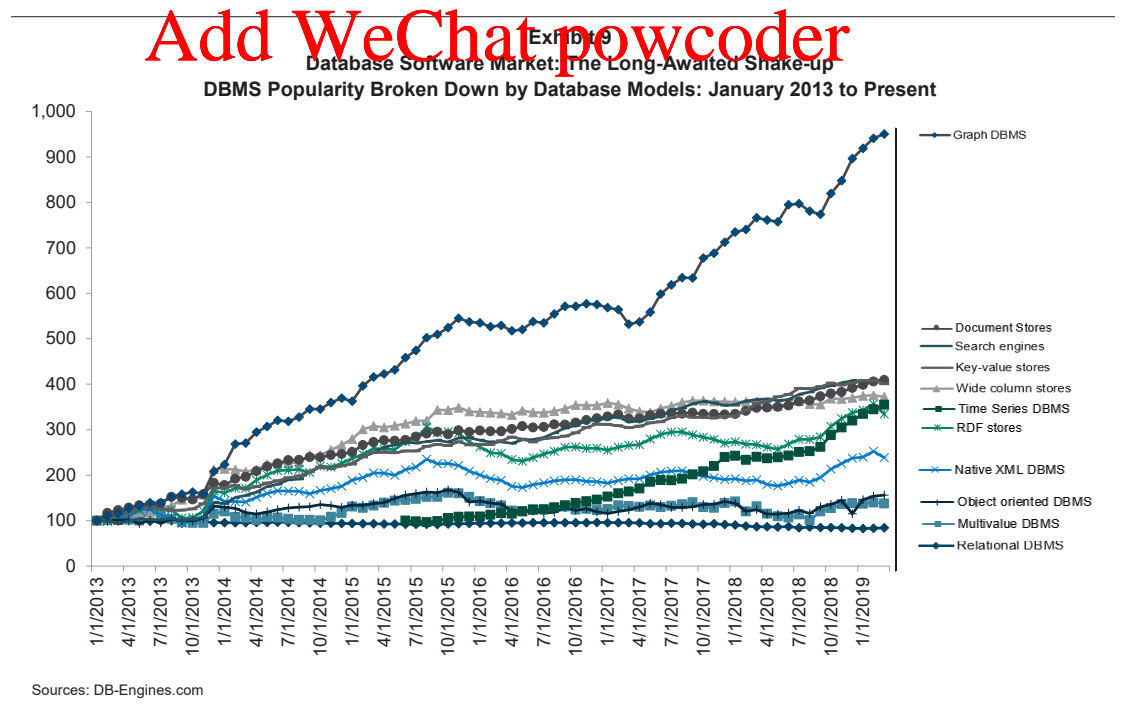
Exhibit 10
Database Software Market: The Long-Awaited Shake-up
DBMS Popularity Broken Down by Database Model: March 2017 to Present

Sources: DB-Engines.com
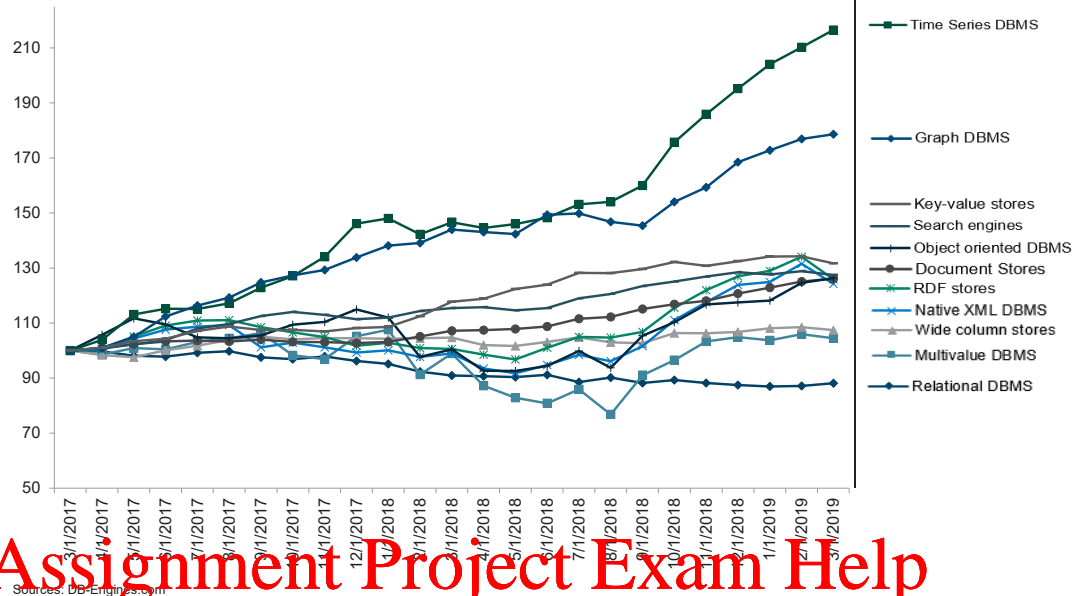
In-Memory Database

An in-memory database is a type of relational or non-relational database that relies primarily on memory for data storage, in contrast to databases that store data on hard disks or SSDs. In-memory databases are designed to attain minimal response time by eliminating the need to access disks. Because all data is stored and managed exclusively in main memory, it is at risk of being lost upon a process or server failure. To address this challenge, in-memory databases can persist data on disks by storing each operation in a log or by taking snapshots. With the continued price drop in memory and new persistent memory technologies (e.g., Intel Optane 3D XPoint, Samsung Z-SSD) accelerating the cost reduction curve, in-memory databases are likely to become even more attractive to customers in the future.

User demands on databases to deliver high throughput (sometime exceeding 1 million IOPS) and low latency (sub-milliseconds in many cases) are becoming increasingly commonplace. In-memory databases are ideal for applications that require lightning-fast response times and can have large spikes in traffic coming at any time such as gaming leaderboards, session stores, and real-time analytics.

In addition, in-memory technology is a key enabler for a number of hybrid use-cases incorporating transactions and analytics. The only way to tackle this level of complexity while still maintaining a high level of performance is to manage data in RAM. The advantage of this approach, which Gartner terms HTAP (hybrid transaction/analytical processing) and IDC calls ATP (analytic-transaction processing), is to provide customers with the ability to do analytics on-board the operational database without the need to move the data to a data warehouse. This is possible because both transactional and analytical RDBMSs use SQL.

For example, the SAP HANA in-memory database is architected to enable applications to support both transactional and analytical processing on a single system with one copy of the data. While more vendors are moving in the direction of delivering HTAP/ATP capabilities, we note that cloud vendors are maintaining their strategies of separate operational and analytical offerings in the form of dedicated data warehousing and operational DBMSs.

The DBaaS Wave

A concept that effectively combines two mega trends—cloud computing and open source—DBaaS is a DBMS sold as a fully managed, subscription service by a CSP (cloud service provider) or independent database vendor (running on a public or private cloud). While a DBaaS does not need to be based on open-source software, the practical reality is that many DBaaS offerings rely on open-source technology.

The concept is simple: a DBaaS frees up developers from self-hosting and managing what is arguably the most complex and difficult-to-manage layer of the application stack (the database). Instead of worrying about back-end provisioning, scaling, backup, monitoring, and management of their databases, which is taken care of by the DBaaS provider, developers are able focus on building applications with the speed and agility necessary in today's information-driven economy.

With the acceleration of workloads shifting to the cloud and the fact that the majority of new applications are cloud-native, DBaaS is seeing rapid adoption around the world. Early fears around security, data governance, and other perceived cloud vulnerabilities have been superseded by the tremendous value proposition of the cloud, which brings the benefits of pay-as-you-go pricing, elimination of talent-related constraints, self-service provisioning and operation, reduction in resource requirements, scalability, agility, and flexibility. As a result, DBaaS is becoming table stakes for DBMS vendors, with almost all of them already offering or soon to offer their platforms in this managed service deployment model.

Technically speaking, a DBaaS virtualizes the database from the application, allowing the database to be run and managed independent of the application (this is especially useful for microservices-based applications). The DBaaS URL is configured in the application server, directing the application where to send and receive the data. Key characteristics of a DBaaS are multi-tenancy, self-service operation, and consumption-based pricing (though in some cases, a DBaaS charges a flat price per user or per terabyte supported).

As noted above, beyond the well-understood shift from capital expenditure to an operating expenditure model, the appeal of DBaaS for end-users is to offload the operational hassle of managing and tuning the database (and its underlying infrastructure) to the DBaaS provider. Maintaining a traditional database requires a lot of expertise and manual work, and introduces many risks to the enterprise—a DBaaS thus lifts a great burden from database administrators (DBAs).

At the same time, the traditional infrastructure stack is expensive to scale at an enterprise level of SLAs. Unabated growth in data requires constant new purchases and refreshes of hardware and software, with infrastructure management increasingly complicated and database sprawl a constant challenge. A DBaaS effectively ensures high availability and scalability of the database, and is agnostic to whether the production application is in the cloud or on-premises (though for performance reasons most customers will co-locate their app and DBaaS). While we expect most of the traction for DBaaS will be with new, born-in-the-cloud workloads, we believe new on-premises workloads will increasingly leverage DBaaS as well due to the aforementioned end-user benefits.

For DBaaS vendors, the appeal is the ability to monetize free usage of open-source database software and generate recurring revenue with some degree of lock-in. To date, most DBaaS offerings have been used for dev/test use-cases (where DB sizes are smaller and issues of consistency and security are less of a concern), but increasingly they are being used for production applications.

Exhibit 11
Database Software Market: The Long-Awaited Shake-up
Understanding Database-as-a-Service
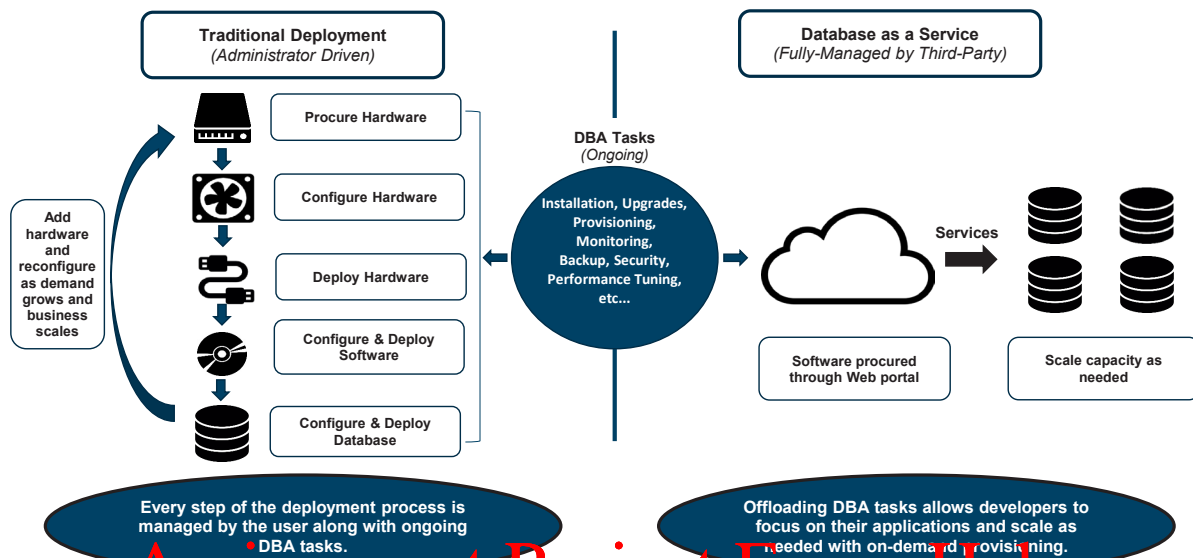
Source: William Blair

Sizing the Operational Database Market

Based on IDC data, our analysis suggests that the operational DBMS market will grow from $27 billion in 2017 to $40.4 billion in 2022 (five-year CAGR of 8.4%). To arrive at an ODBMS market estimate, we removed the data warehouse (analytical database) contribution from the RDMBS segment, as well as the nonrelational analytic data stores contribution (mainly Hadoop-oriented technology) from the dynamic database management systems (DDMS) segment. DDMS encompasses NoSQL databases like document stores, graph DBMSs, and other NoSQL databases. Elsewhere in this report, our definition of nonrelational databases includes DDMS.

Drilling deeper, the operational RDBMS segment (84% of total in 2017), which as noted excludes data warehousing, is expected to grow at a five-year compound annual rate of 8.12%. In contrast, the operational DDMS segment (3% of total in 2017) is expected to grow at a five-year compound annual rate of 30.9%, though IDC expects it will still fall short of reaching 10% of total operational DBMS revenue by 2022—a prediction that we view as overly conservative. Lastly, what IDC refers to as nonrelational database management systems (11.9% of total in 2017)—basically covering legacy database technology for mainframe-related applications—is expected to decline at a five-year compound annual rate of 2%.

In terms of public cloud versus on-premises deployments of these databases, IDC estimates that public cloud-based RDBMSs were 10.6% of the market in 2017, but will expand to 32% of the market in 2022. Meanwhile, IDC estimates that public cloud-based NoSQL (including Hadoop) databases were 70.4% of the market in 2017, but will expand to 86.7% of the market in 2022. This includes native cloud databases from the major IaaS providers, third-party database licenses purchased in the cloud (e.g., through AWS Marketplace), and third-party DBaaS offerings.

Exhibit 12
Database Software Market: The Long-Awaited Shake-up
Operational Database Market Size and Forecast

NDBMS CAGR	-2.03%
DDMS CAGR	30.94%
RDBMS CAGR	8.12%
Aggregate CAGR	8.37%

Nonrelational Database Market (NDBMS)
Dynamic Database Market (DDMS) (Excluding Hadoop)
Relational Database Management Systems (RDBMS) Market (Excluding Data Warehousing)
Source: William Blair Analysis Based on IDC Data
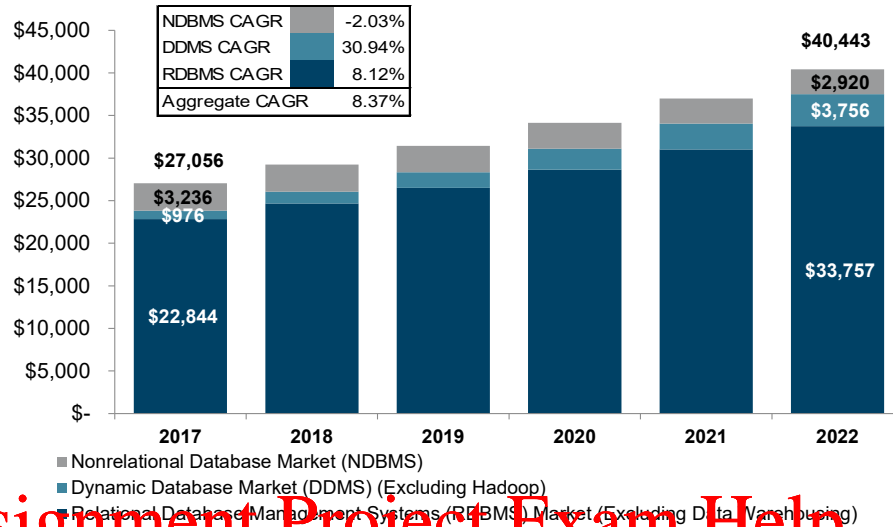
Demand Drivers
We believe that the aggregate demand for the operational database market is being driven by an explosion in data from various outlets, such as streaming, networked devices, edge computing, and customer experience applications. IDC forecasts that the global "datasphere" will grow from 33 zettabytes in 2018 to 175 zettabytes in 2025.

In general, the data volumes growing the fastest are in areas like machine-generated IoT, transaction data, log data, free-form text, social media data, geospatial data, sensor data, video and images, and audio data. These data types are characterized by high volume, in that there is a lot of it, but only specific subsegments of the data are actually valuable for decision-making. The vast majority of this data growth lends itself to the expansion of nonrelational databases (e.g., Amazon DynamoDB, Azure CosmosDB, MongoDB, DataStax, Couchbase), as these highly scalable databases can handle the tactical workloads associated with this massive volume of data. Furthermore, this data is often not orderly, making it better suited for nonrelational databases.

Streaming data for operational use-cases
A concrete example of this is streaming data, which was once solely the purview of analytical databases, but is moving toward operational use-cases since advances in technology are making it possible to process streaming data as it comes in. Analysts estimate that, in less than 10 years, more than one-quarter of data created will be real-time in nature. In addition, in a world where more things and people are digitally connected, our interactions with real-time data will grow, with the average rate per capita of data-driven interactions increasing 20-fold over the next 10 years. Enterprises will not be able to survive without the ability to harness this real-time data; not only to deliver highly personalized experiences to end-users, but also to extract up-to-the-moment business insights for internal decision-making.

A good example of how streaming data can be harnessed in real-time comes from the logistics business. A logistics firm might have at its fingertips real-time streams coming from various sources, including positional data for a fleet of trucks, real-time traffic information, and alerts from customers asking for packages to be picked up. This data can be synthesized and used to optimize the routes of the logistics company's trucks, avoiding congested roads and speeding package pickup and delivery.

Relational databases not going anywhere

While we expect robust growth in nonrelational databases, which are expanding off a smaller base, we do not think it will eliminate reliance on relational databases, which are still essential for most structured data (e.g., credit card transactions, medical records, ERP, and CRM systems). Rather, we expect relational and nonrelational databases to coexist in a hybrid model for enterprise customers for many years into the future.

While the relational database market has no doubt slowed due to the precipitous rise of nonrelational competitors, and evidence suggests that some customers are delaying relational database purchases as they consider moving to the cloud, an 8% CAGR over the next five years demonstrates that relational database growth should remain resilient. While there are some tasks that may always be best suited for the relational structure, relational databases have also tried to co-opt some key capabilities of nonrelational databases.

NewSQL databases

For instance, a new class of databases often called NewSQL aims to address some of the historical shortcomings of relational databases, including horizontal scalability, dynamic schema support, and operational distribution (across regions, for example). The technology is particularly well suited for any application that requires very high ingest rates and fast response times (average 1-2 milliseconds), but also demands transactional accuracy provided by ACID guarantees—for example, customer billing, real-time authorization, or real-time fraud detection. Key vendors pushing this NewSQL technology include MemSQL, NuoDB, Google Cloud Spanner, Clustrix (MariaDB), Cockroach Labs, and VoltDB.

Exhibit 13
Database Software Market: The Long-Awaited Shake-up
Postgres: The Linux of RDBMS

One of the more unsung database technologies in our view is PostgreSQL or simply Postgres, an open source relational database whose original claim to fame was compatibility with Oracle and Db2 (enabling fairly painless migration of these databases to Postgres). For more than 25 years, Postgres has been a highly respected relational database at the heart of a broad range of sophisticated transactional applications in verticals like financial services, insurance, health care, energy, media, retail, government, telco, and pharmaceuticals.

Active since 1996, Postgres is one of the oldest and most stable open source projects. Similar to the open source Linux operating system, Postgres offers enterprise-class technology without enterprise-class costs. The technology has also had a far-reaching influence on the entire database market with Amazon Redshift, HP Vertica, IBM Informix, IBM Netezza, Pivotal Greenplum Database, Teradata Aster, and VoltDB all tracing their roots back to Postgres code.

Like many other databases, the technology's roots can be traced back to UC Berkeley where legendary database inventor Michael Stonebraker built a proprietary database that he called Post-Ingres (Postgres) – the name indicating that it was his next-generation project after the original Ingres relational database project that he led. Postgres was first released in June 1989, but in 1995 Andrew Yu and Jolly Chen, students from Stonebraker's lab, developed an extended SQL version of Postgres, swapping out the native Postgres query language. They called the revamped database Postgres95 and released their code on the Web. An open source project based on that code release, renamed PostgreSQL, was founded the following year by Marc Fournier, Bruce Momjian, and Vadim Mikhe, who crafted the first full release, "Version 6.0," (made available in early 1997). Version 6.0 transformed what was originally an academic database into a commercial quality database with an extensive feature set to rival the most sophisticated commercial databases on the market.

Today, Postgres is developed by the PostgreSQL Global Development Group, an open source community consisting of a wide variety of volunteers, many of them employed by commercial vendors such as EnterpriseDB and Red Hat. Postgres has a reputation for high quality, standards-based code, extensive security features, excellent documentation and can address a range of workloads from ACID-compliant transactional applications to unstructured data like JSON documents and key-value stores. The PostgreSQL software license is quite liberal (very similar to the BSD license, see page 34) and allows use without cost or signed agreements. It also allows modification and redistribution without the need to contribute changes back to the community.

As organizations explore open source alternatives, Postgres is becoming an increasingly popular choice. Out of the roughly 200 database technologies measured by DB-Engines, Postgres was ranked the fourth most popular database for several years running, and was named "DBMS of the Year" for 2017. All of the major cloud vendors now offer Postgres databases, and the leading commercial player is EnterpriseDB. The value proposition is lower lifecycle license and maintenance costs relative to traditional proprietary databases, while providing full ACID features and enterprise-grade tools and management.

Source: William Blair

The influence of open source and DBaaS

One challenge for large, legacy RDBMS vendors is that open-source competitors can offer similar functionality for less. We expect this to serve as a drag on industry revenue growth. In an analogous manner to the shift from on-premises application software to the SaaS business model, enterprises and SMBs seem less inclined to want to pay the upfront licensing fees for database software. Pricing estimates suggest that open-source software like MySQL Enterprise Edition or EnterpriseDB EDB Postgres Enterprise can cost a fraction of what an enterprise might pay for a new Oracle Database Enterprise Edition license. To a company like Oracle's credit, the lock-in for large enterprises, with immense volumes of data, is powerful, making it difficult for some large customers to shift to other databases.

The same open-source influence exists on the nonrelational side, although at a much greater magnitude. Many nonrelational databases are based on open-source software. Thus, there are wide swaths of customers using these products, but not paying for it. The business model for these vendors relies on a customer converting to a paid model once the customer gets more serious and needs premium support and tools. What has made matters worse for the open-source vendors is co-opting of their technology by the big CSPs, who offer managed services (DBaaS) based on their technology. This has prompted virtually all of the open-source vendors to introduce fully managed DBaaS offerings of their own, which can be deployed on the various public clouds, or even, in some cases, on-premises.

The DBaaS trend allows both CSPs and independent vendors to better monetize previously free usage, as customers will need to pay to use these services as they are consumed. As a result, we expect DBaaS adoption will serve as a significant tailwind for revenue growth in the operational database market, both for the RDBMS and NDMBS segments.

Database Market Share

In sizing the market, IDC segments database vendors and market share based on the broad architectural distinctions of the database (i.e., the data structure, specifically relational versus nonrelational), which differs somewhat from our approach, which focuses on the business use-case first (operational versus analytical). IDC uses the following three categories for its market share data: RDBMS, DDMS, and NDBMS.

RDBMS include both operational relational databases and data warehousing products. DDMS includes both NoSQL operational DBMS vendors and NoSQL analytical data platforms like Hadoop. NDBMS, in IDC parlance, refers to databases that are schematic and include hierarchical, network, inverted-list, and multi-value. These databases are not inherently relational, usually information can be obtained without a SQL statement, and this market is usually associated with legacy mainframe vendors like IBM or Intersystems. This market is in modest decline due to reduced reliance on mainframes.

For the purposes of this report, as noted earlier, we combine DDMS and NDBMS into one category that we call nonrelational databases, which include products like MongoDB, Amazon DynamoDB, Azure CosmosDB, DataStax, Couchbase, Neo4j, and Redis Labs.

Exhibit 14
Database Software Market: The Long-Awaited Shake-up
Relational Database Vendor Share 2015 (in millions) ($29.4B in total)
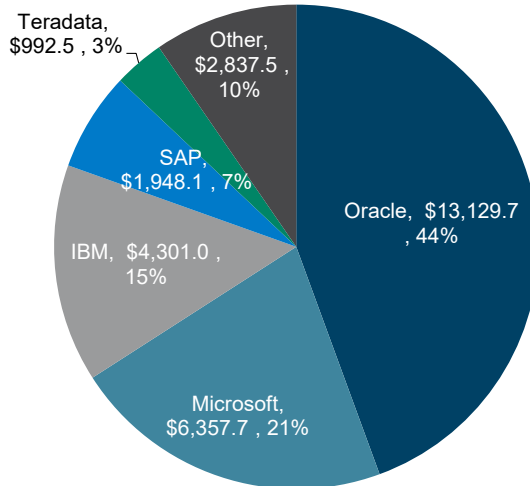
Teradata, $992.5 , 3%
Other, $2,837.5 , 10%
SAP, $1,948.1 , 7%
IBM, $4,301.0 , 15%
Oracle, $13,129.7 , 44%
Microsoft, $6,357.7 , 21%

*Data includes analytical databases

Sources: Worldwide Relational Database Management Systems Software Market Shares, 2017: The Race to the Cloud, IDC, Published June 2018

Exhibit 15
Database Software Market: The Long-Awaited Shake-up
Relational Database Vendor Share 2016 (in millions) ($31B in total)
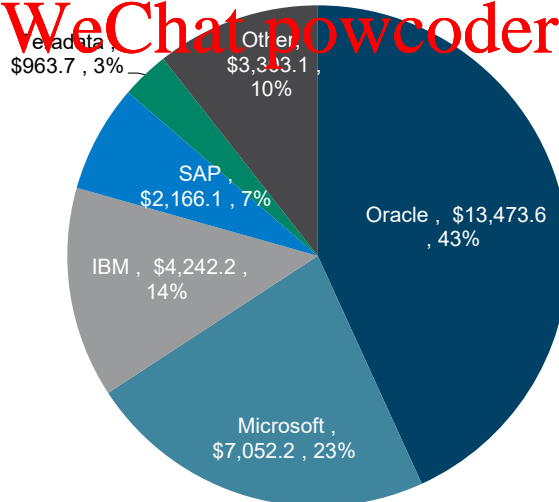
Teradata, $963.7 , 3%
Other, $3,383.1 , 10%
SAP , $2,166.1 , 7%
IBM , $4,242.2 , 14%
Oracle , $13,473.6 , 43%
Microsoft , $7,052.2 , 23%

*Data includes analytical databases

Sources: Worldwide Relational Database Management Systems Software Market Shares, 2017: The Race to the Cloud, IDC, Published June 2018

Exhibit 16
Database Software Market: The Long-Awaited Shake-up
Relational Database Vendor Share 2017 (in millions) ($33.1B in total)
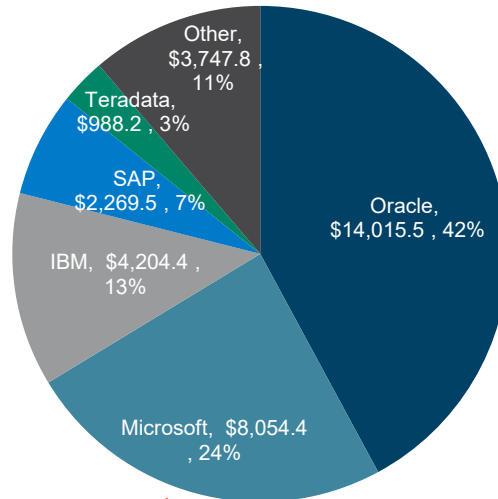
Other, $3,747.8 , 11%
Teradata, $988.2 , 3%
SAP, $2,269.5 , 7%
IBM, $4,204.4 , 13%
Oracle, $14,015.5 , 42%
Microsoft, $8,054.4 , 24%

*Data includes analytical databases
Sources: Worldwide Relational Database Management Systems Software Market Shares, 2017: The Race to the Cloud, IDC, Published June 2018

Exhibit 17
Database Software Market: The Long-Awaited Shake-up
Dynamic Data Management Systems (DDMS) Vendor Share 2015 (in millions) ($1.31B in total)

Other , $403.4 , 31%
Microsoft , $211.3, 16%
Amazon Web Services , $385.9, 30%
Hortonworks , $77.7 , 6%
Cloudera , $106.9, 8%
Google , $115.0, 9%

MongoDB was 4% of the market in 2015 with $56.4 million in revenue

*Data includes Hadoop

Sources: Worldwide Dynamic Data Management Systems Software Market Shares, 2017: Bringing Power and Flexibility to the Cloud, IDC, Published July 2018

Exhibit 18
Database Software Market: The Long-Awaited Shake-up
Dynamic Data Management Systems (DDMS) Vendor Share 2016 (in millions) ($2.15B in total)
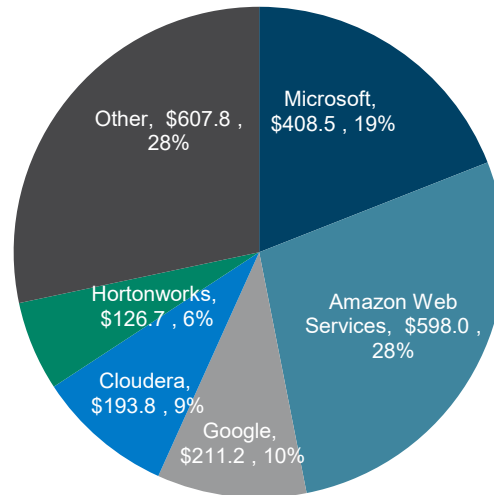
MongoDB was 4% of the market in 2016 with $82.9 million in revenue

*Data includes Hadoop

Sources: Worldwide Dynamic Data Management Systems Software Market Shares, 2017: Bringing Power and Flexibility to the Cloud, IDC, Published July 2018

Exhibit 19
Database Software Market: The Long-Awaited Shake-up
Dynamic Data Management Systems (DDMS) Vendor Share 2017 (in millions) ($3.4B in total)
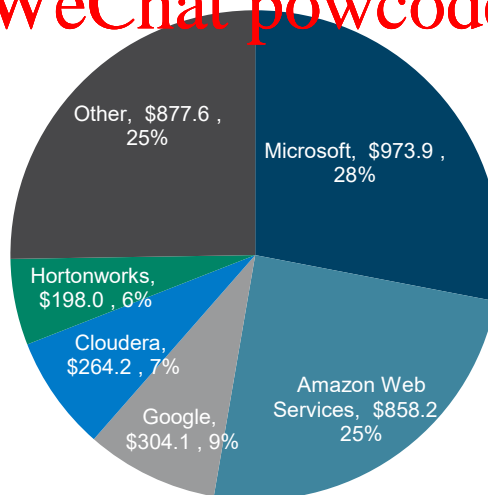
MongoDB was 4% of the market in 2017 with $134.4 million in revenue

*Data includes Hadoop

Sources: Worldwide Dynamic Data Management Systems Software Market Shares, 2017: Bringing Power and Flexibility to the Cloud, IDC, Published July 2018

Exhibit 20
Database Software Market: The Long-Awaited Shake-up
Nonrelational Database Management Systems Vendor Share 2015 (in millions) ($3.22B in total)
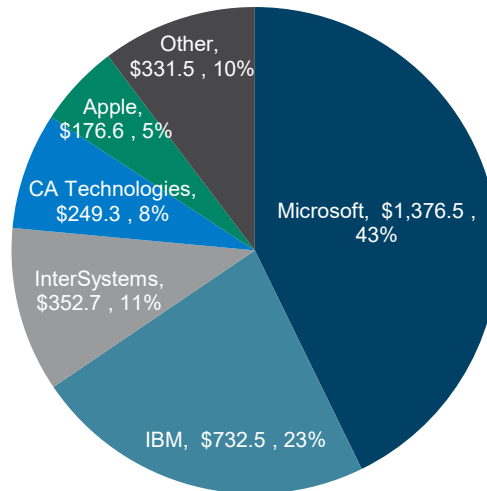
Microsoft, $1,376.5, 43%
IBM, $732.5, 23%
InterSystems, $352.7, 11%
CA Technologies, $249.3, 8%
Apple, $176.6, 5%
Other, $331.5, 10%

Sources: Nonrelational Database Management Systems Software Market Shares, 2017: Changing the Guard, IDC, Published August 2018

Exhibit 21
Database Software Market: The Long-Awaited Shake-up
Nonrelational Database Management Systems Vendor Share 2016 (in million) ($3.17B in total)

Microsoft, $1,369.1, 43%
IBM, $655.4, 21%
InterSystems, $381.4, 12%
CA Technologies, $238.1, 7%
Apple, $185.6, 6%
Other, $335.4, 11%

Sources: Nonrelational Database Management Systems Software Market Shares, 2017: Changing the Guard, IDC, Published August 2018

Exhibit 22
Database Software Market: The Long-Awaited Shake-up
Nonrelational Database Management Systems Vendor Share 2017 (in millions) ($3.24B in total)

Other, $347.7 , 11%
Apple, $197.0 , 6%
CA Technologies, $221.1 , 7%
InterSystems, $450.7 , 14%
Microsoft, $1,362.5 , 42%
IBM, $657.1 , 20%

Sources: Nonrelational Database Management Systems Software Market Shares, 2017: Changing the Guard, IDC, Published August 2018

Takeaways From Market Share Data

IDC's DDMS category (which as a reminder includes both NoSQL operational and analytical data-bases) grew 65% between 2015 and 2016, and 62% between 2016 and 2017. NoSQL insurgents such as MongoDB, MarkLogic, DataStax, and Couchbase each expanded briskly. One of the challenges for these smaller vendors is that Microsoft actually grew revenue and share the fastest over this three-year period following its introduction of Azure Cosmos DB.

Exhibit 23
Database Software Market: The Long-Awaited Shake-up
Total Database Market Share 2015-2017: IDC Data

— Oracle — Microsoft — IBM — SAP — Amazon Web Services — All Others
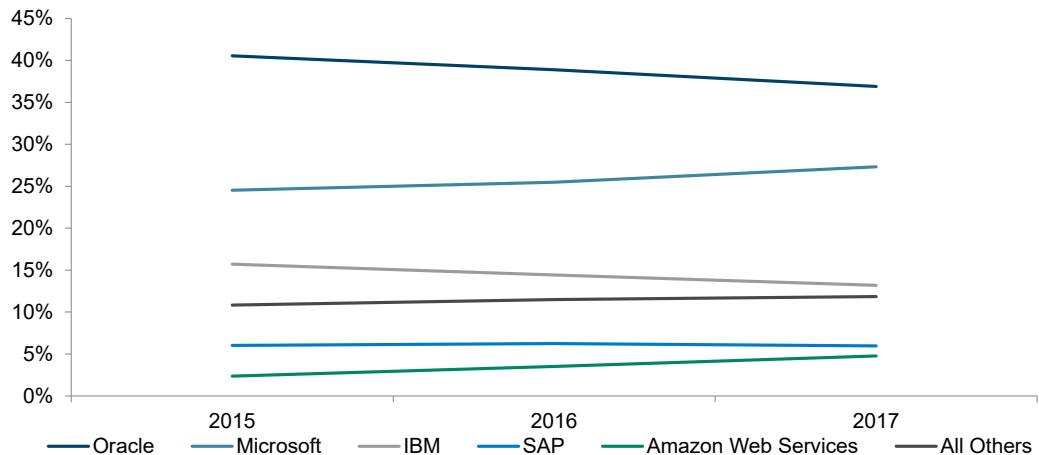
Sources: Consolidated Market Share Data from the Following Three IDC Reports
Worldwide Relational Database Management Systems Software Market Shares, 2017: The Race to the Cloud, IDC, Published June 2018
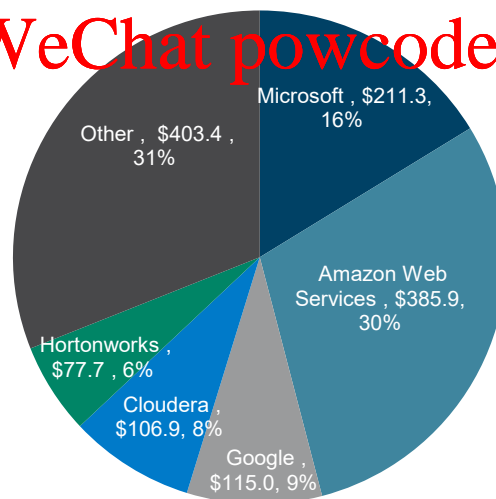Worldwide Dynamic Data Management Systems Software Market Shares, 2017: Bringing Power and Flexibility to the Cloud, IDC, Published July 2018
Nonrelational Database Management Systems Software Market Shares, 2017: Changing the Guard, IDC, Published August 2018

On the relational database side, we saw fewer changes over the three-year period. Incumbents such as Oracle, Microsoft, IBM, and SAP remained the top players, but we did see the entrance of Chinese vendors such as Alibaba and Tencent in 2016. Google and Amazon are two of the disruptive players in this space, with Amazon having strong success with Aurora and RDS, and Google introducing Cloud Spanner, which is a SQL database designed to bring the scalability of NoSQL technology.

The nonrelational database segment (which as a reminder under IDC's classification encompasses vendors associated mainly with the mainframe market) is modestly declining, and we are less concerned with the dynamics in this area for the purposes of this report.

The Open Source Insurgency

Open-Source Background

Netscape founder and venture capitalist Marc Andreessen famously said that software is "eating the world" to refer to the disruptive impact software is having across various industries (think Uber, Netflix, Amazon) and the fact that most businesses today run on software delivered as online/cloud services. By extension, we would argue that open-source software (OSS) is cannibalizing software itself by democratizing and commoditizing what has historically been a proprietary and closed model in the software industry. Examples of popular open-source projects include Linux, OpenStack, Hadoop, Docker, and MySQL, but according to open-source management vendor Black Duck Software, there are now more than a million different open-source software projects in existence.

The year 2018 heralded the mainstream arrival of OSS with a huge amount of M&A, including salesforce.com's $6.5 billion acquisition of MuleSoft, Adobe's $1.6 billion deal for Magento, Microsoft's $7.5 billion takeout of GitHub, and last but not least, IBM's planned $34 billion acquisition of Red Hat.
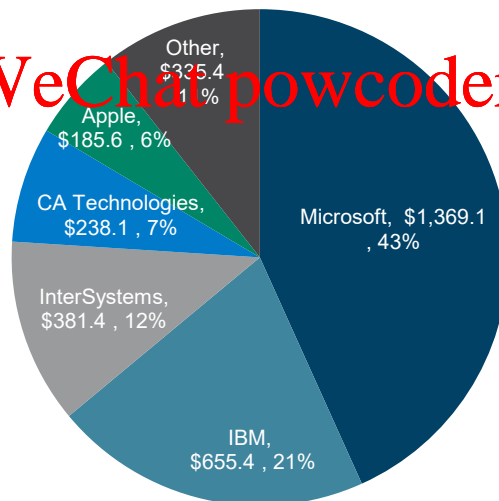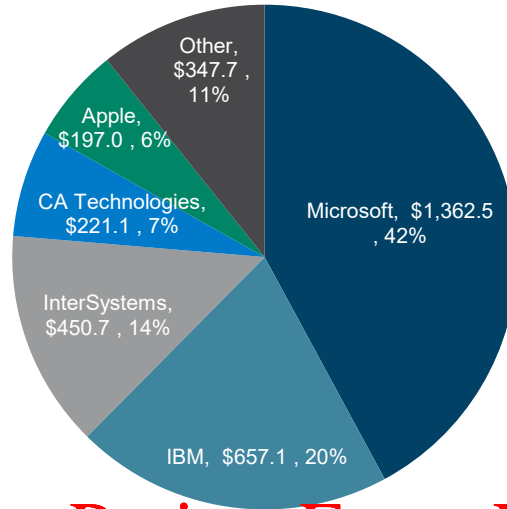
In its simplest form, OSS refers to software in which the source code—a collection of computer instructions written in human-readable computer language—is freely available to the general public for use or modification from its original design. Open-source code is typically created as a collaborative effort among programmers who improve upon the code and share the changes within the community. The OSS movement was born from enthusiasts and hobbyists who believe in freeing the innovation process from the confines of proprietary commercial interests. In this way, open-source software was a kind of grassroots response to the "tyranny" of proprietary software, where the developers or distributors maintain all rights to analyze, change, and share the software, and end-users are denied access.

As open-source code is technically free and designed to be a collaborative, iterative organism, there are inherent advantages in innovation, cost, and time to market over proprietary software alternatives. Proprietary code would be hard-pressed to match this pace of innovation and improvement, and the cost to the end-user of deploying OSS is significantly less than that of traditional software. Because of the community-driven nature of the projects, which can feature thousands of contributors, open-source code is a constantly improving entity with users of the software often dictating its direction.

While open source is a revolutionary concept, it has several drawbacks, including code maturity and stability (given its fluid nature), formal technical support models ("one throat to choke"), and guaranteed compatibility (certification) with various hardware systems, software infrastructure and applications, and public cloud platforms. As a result, most enterprise IT organizations do not want to take on the risk and cost of managing open-source software on their own. While the idea of unsupported software is appealing at first blush, any serious business manager will shy away from it as it places too much burden on employees and exposes the company to application quality problems.

OSS has been around for roughly two decades in various forms, starting in the consumer and web-scale space (with projects like Apache Web Server and Mozilla Firefox), but it has only been in the last 10 years or so that open-source solutions have become broadly accepted by mainstream enterprise IT departments, a mega trend that we believe is still in the early stages. While the plethora of OSS contributors can provide unparalleled insight and sophistication, the mass collaboration and fundamentally open and free nature can also often result in a messy piece of software without clear direction for a specific use-case.

In response, businesses were created to release hardened, enterprise-grade versions of popular OSS projects, monetized either via support and maintenance contracts or as privatized, "closed" versions of the originally "open" project. An example of the former would be Red Hat Enterprise Linux or Hortonworks (Hadoop distribution), while an example of the latter would be Redis Labs (for Redis), Couchbase (for CouchDB), or Cloudera (for Hadoop). In addition, CSP and independent vendors have created SaaS offerings based around various open-source offerings, which enables monetization of free usage of community editions.

Open-Source Business Models

Support-based model – In this model, the vendor does not sell software licenses or software subscriptions but provides OSS distribution and technical services to customers in the form of an annual subscription. The vendor provides the actual OSS code to customers (including periodic updates) and ongoing technical support to address any issues. On top of this, the vendor will offer consulting/ professional services to help with implementation and training. Examples in the database market are MySQL and MariaDB.

Open core – This model is similar to the support-based model in that the core software remains open-sourced, and continues to be developed by the community (a community edition is still freely available). However, special features and modules that extend or enhance the core product are only available as commercial software, for a fee. Examples in the database market are Redis Labs, Couchbase, and Neo4j.

Exhibit 24
Database Software Market: The Long-Awaited Shake-up
Overview of Open-Source Business Models
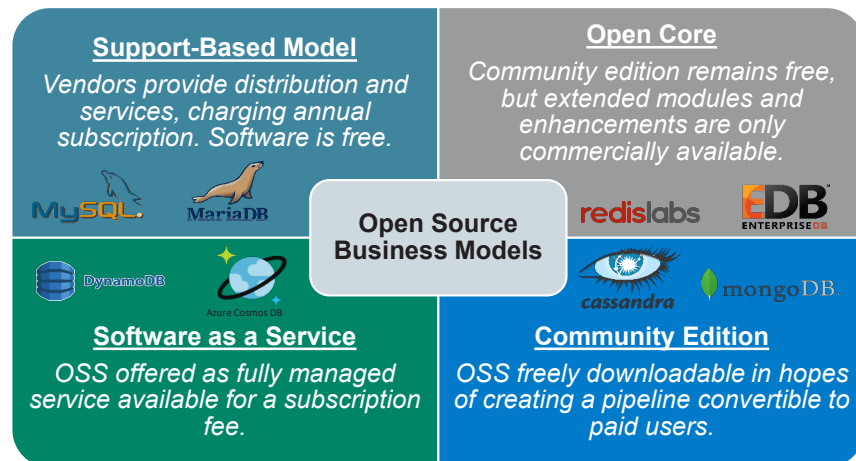
Source: William Blair

SaaS – In this model, the vendor offers the OSS as a fully managed service that is accessible only via a paid subscription. The main appeal here is to offload the operational hassle of managing the software (and its underlying infrastructure) to the SaaS provider. Examples in the database market are Amazon DynamoDB, Microsoft Cosmos DB, Google Cloud Spanner, and MongoDB Atlas.

Community Edition – This is the free, downloadable version of OSS, but is critical to the freemium business model of many commercial OSS vendors. This is because the community edition encourages usage and creates a pipeline of users who could potentially be paid users once they get to a point where they are more serious and need support, tools, or premium features.

Open-Source Databases Becoming More Closed
Because OSS is particularly popular with developers, and developers work closely with databases when building or migrating applications, it is no surprise that the database market has become one of the most fertile areas for open-source development. According to DB-Engines, there are 176 open-source database offerings available today across the relational and nonrelational segments of the market.

Exhibit 25
Database Software Market: The Long-Awaited Shake-up
Number and Type of Database Systems, March 2019

Open Source License, 176

Commercial License, 169

Sources: DB-Engines.com

Exhibit 26
Database Software Market: The Long-Awaited Shake-up
Database Popularity Scores, March 2019

Open Source License, 51.3%

Commercial License, 48.7%
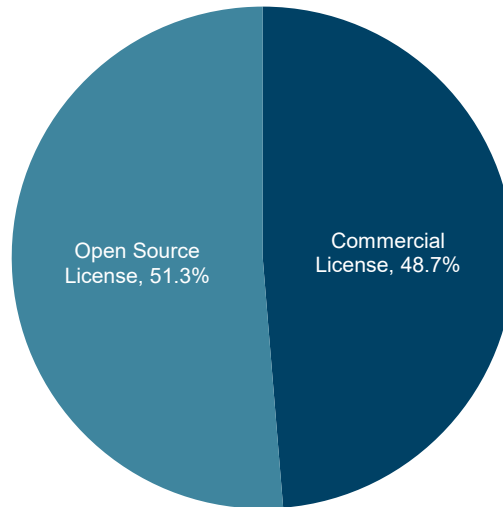
Sources: DB-Engines.com

Exhibit 27
Database Software Market: The Long-Awaited Shake-up
Popularity of Open-Source DBMS Versus Commercial DBMS: Popularity Trend
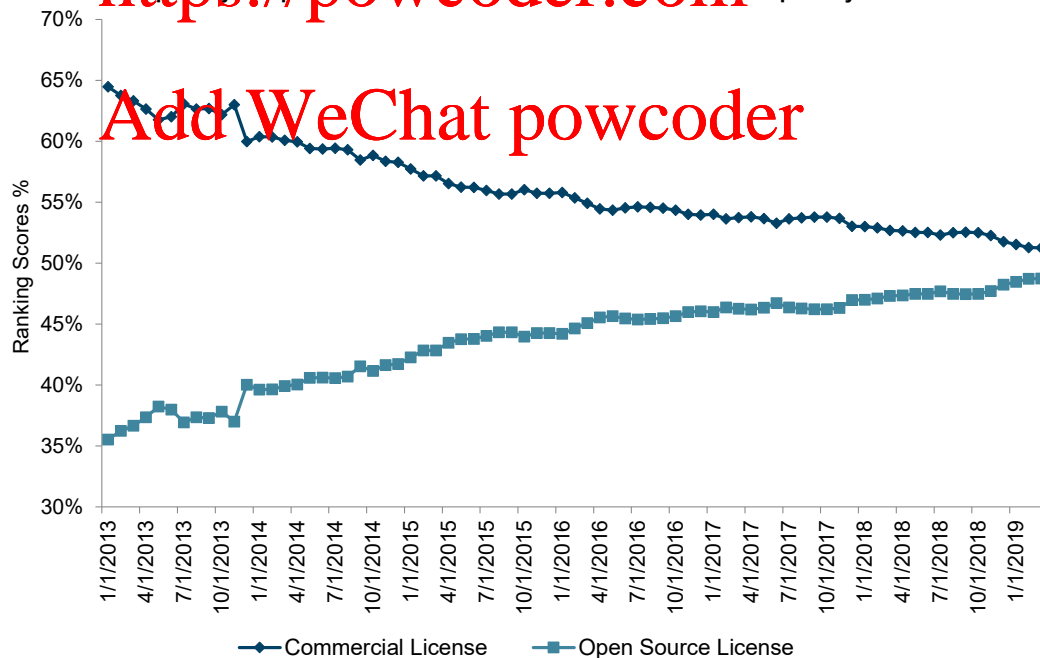
Sources: DB-Engines.com

Exhibit 28
Database Software Market: The Long-Awaited Shake-up
Popularity Broken Down by Database Model: March 2019

Sources: DB-Engines.com
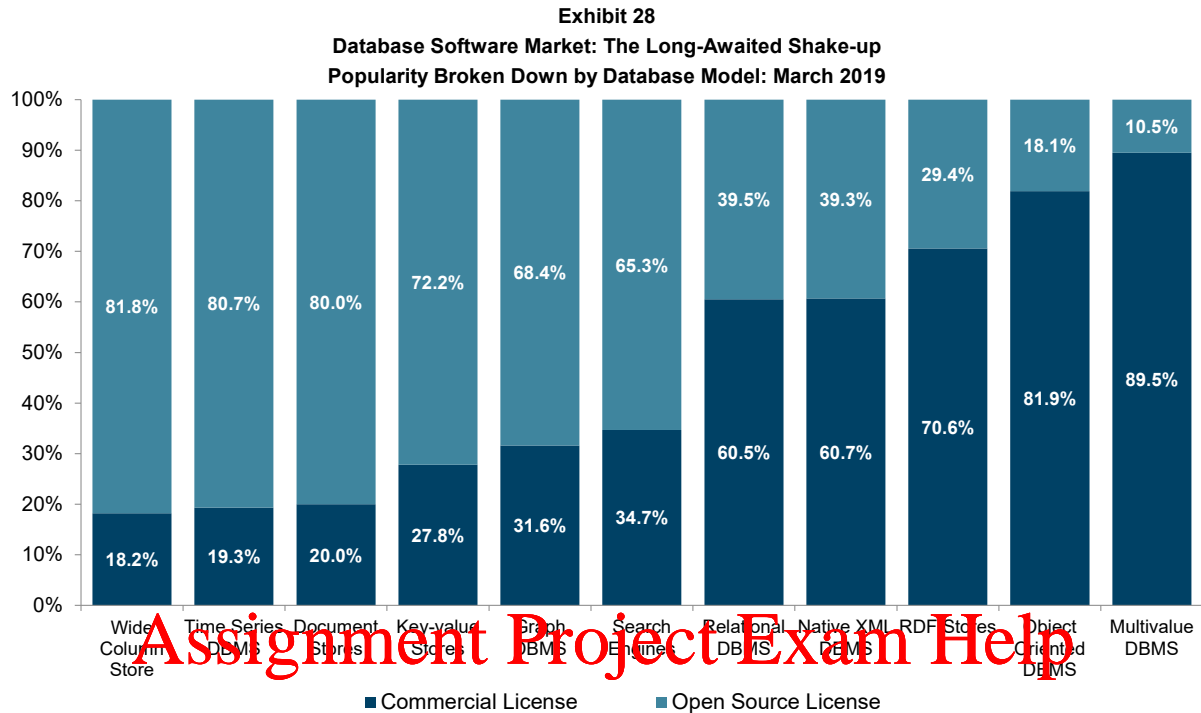
A key trend to watch is the ability of cloud providers like AWS to develop commercial DBaaS offerings from open-source projects like MongoDB (e.g., Amazon Document DB), Redis (e.g., Amazon Elasticache), PostgreSQL (e.g., Amazon RDS for PostgreSQL), and Elasticsearch (e.g., Amazon Elasticsearch). This is fueling a backlash among open-source vendors who believe that CSPs are unfairly monetizing open-source software while contributing little to the community. These vendors dispel the longstanding myth in the open-source world that projects are driven by a community of contributors. The reality is that the vast majority of the code in most modern open-source projects derives from paid developers, typically those employed at the primary commercial vendor that sponsors the project. For example, according to Redis Labs CEO Ofer Bengal, 99% of the contributions to the Redis open-source project were made by Redis Labs.

As a result, several open-source vendors, including MongoDB and Redis Labs, have attempted to institute more restrictive licensing models to make it harder for CSPs to offer a DBaaS built on open-source technology and gain access to key extensions and features built on top of the core open-source project. We note that both MongoDB and Redis Labs offer a community edition in addition to their open core commercial enterprise editions—the more restrictive licensing pertains to their community editions.

In October 2018, MongoDB issued a new software license, called Server Side Public License (SSPL), for MongoDB Community Server (the free community edition). The license outlines the conditions of deploying MongoDB—or any other open-source project licensed under the SSPL—as a service. The license retains all of the same freedoms that the open-source community had with MongoDB under the AGPL license, including freedom to use, review, modify, and redistribute the software. The only substantive change is an explicit condition that any organization attempting to exploit MongoDB Community Server as a service must open source the full software stack that it uses to offer such service.

However, as of its fourth quarter 2019 earnings release on March 13, 2019, MongoDB announced that it had withdrawn its submission of SSPL to the Open Source Initiative, citing a lack of community consensus regarding its legitimacy as an open source license. MongoDB management commented that the company has decided to focus its efforts on working with other stakeholders in the open source and broader tech community to either refine the SSPL or develop an alternative license that addresses these issues. For the time being, current and future versions of MongoDB Community Server, including patch releases to prior versions, will continue to be offered under SSPL until there is a broadly accepted alternative license that is designed for the cloud era.

In the same spirit as MongoDB, Redis Labs introduced in February 2019 a new type of license called RASL, which applies to certain Redis Modules. Combatting what it and others see as the trend of the large cloud providers taking advantage of their open-source projects without contributing back to the community, the RASL license restricts users from modifying and integrating code for a database product application, caching engine, steam processing engine, search engine, indexing engine, or ML/DL/AI serving engine. Although this added restriction in the new license means that it is no longer technically considered an open-source license, Redis Labs argues that in practice, it should not materially affect most developers who use its modules.

Exhibit 29
Database Software Market: The Long-Awaited Shake-up
Open-Source Licensing Models

- GPL: Widely used - freedom to run, study, share, and modify; restrictive in terms of viral, copyleft, and hereditary characteristics
- AGPL: Slightly less strict than GPL - software used on network server must be open-source
- APL: Apache License V2 - adds restriction that terminates the grant of patent rights if licensee sues over patent infringement
- BSD: Most liberal - cannot claim authorship or sue the developer

Sources: William Blair

The Pull of the Cloud

It is an understatement to say that cloud adoption is transforming the database landscape. As more applications are born in the cloud or move to the cloud, they pull with them more databases. Indeed, we estimate that cloud-resident databases (whether sold directly by CSPs or sold by third parties as cloud-based licenses or hosted DBaaS) will capture the lion's share of growth in the operational DBMS market over the next several years. In addition, we expect CSPs to accelerate their database migration promotional efforts, offering tools and utilities to help users lift-and-shift their workloads to the cloud.

Competitively speaking, the advent of myriad, cloud-native DBaaS offerings from the big three CSPs has put pressure not only on incumbent database providers but also upstart nonrelational and relational DBMS vendors. While cloud-native databases result in greater customer lock-in to a particular cloud, their main appeal is simplicity, as developers can work in the same environment in which the application is located. CSPs can also offer their own versions of popular open-source DBMSs; this creates potential headwinds for open-source vendors and, as noted above, has created a backlash on historically permissive open-source licensing models. The proliferation of DBaaS offerings from the big CSPs has also resulted in third-party vendors following suit with their own cloud-agnostic DBaaS offerings.

Ultimately, we do not see a zero-sum game here—i.e., there is room for both CSPs and independent vendors to be successful. However, with CSPs co-opting the technology of many of the insurgent players, the survival of more than a handful of stand-alone, upstart database companies is dubious. We expect perhaps four to five of these companies will reach escape velocity, with the rest either getting swallowed up by bigger players, executing mergers of equals, or fading away. The big losers in our view will be the incumbent database players without a strong IaaS platform (e.g., Oracle and IBM—see below).

Rise of Containers and Microservices

As noted earlier, a key contributor to the disruption taking place in the database market is the rise of next-generation application architectures, which are often better aligned with next-generation databases and at a minimum trigger a decision on which database to use. These next-generation application architectures are generally characterized by two interrelated technologies: containers and microservices.

By providing a uniform and efficient way to package the software code for an application with all of its dependencies, Linux containers have emerged as essential infrastructure building blocks for next-generation microservices-based application architectures. Microservices are in the spotlight because they offer benefits such as the decoupling of services, data store autonomy, functional isolation, and miniaturization in development and testing, as well as other advantages that expedite time-to-market for new applications or updates.

Microservices are based on small, independently deployable "services" or components (housed in containers) that when put together form an application. This allows specific functions of the application to be isolated and worked on by small teams, giving developers greater flexibility and agility in the application development process, and making it easy to make modifications as each service has a defined impact on the overall application. In addition, each service can be reused in other applications and is easily scaled by distributing these services across servers. In other words, a single

application can be composed of many discrete components, each of which can be independently managed and updated. This modular approach to system-building represents a major architectural change from prior isolated and monolithic application structures.

One of the core principles of a microservices architecture is the ability and freedom of each service to choose a database that best suits its own data model, rather than the reliance on a single large database, as is the case with a monolithic architecture. More specifically, microservices developers can employ a data model (database) that best suits the specific requirements of the microservice, such as performance, availability, consistency, and reliability. This helps ensure smooth operation of the application and prevents bottlenecks in the flow of data (which would severely hamper application performance). For example, the slow performance of a relational database makes it ill-suited for a microservices-based user-experience application that relies on accessing data with sub-millisecond latency.

Exhibit 30
Database Software Market: The Long-Awaited Shake-up
Database Architecture Before and After Microservices
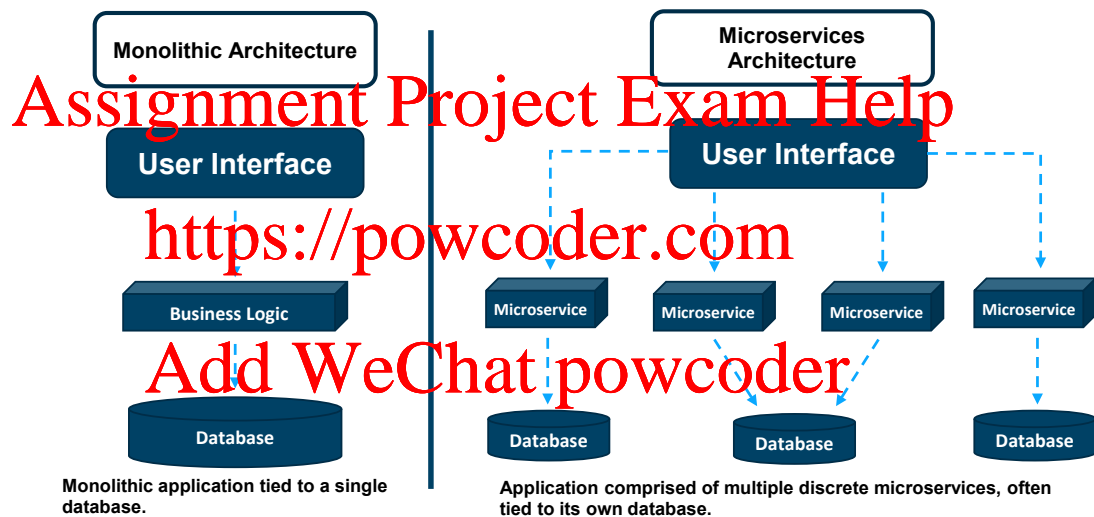
Source: William Blair

For some microservices, the database could hold data of record, while for others, it may just be a temporary store. More specifically, microservices that ingest transient data-like events, logs, messages, and signals before passing them to the appropriate destinations require a database that can hold the data temporarily while supporting high-speed writes. Since transient data is not stored anywhere else, high availability of the database used by this type of microservice is critical.

Other types of microservices need to process data in real-time to enable instant user experiences. An example here is a cache server, a temporary data store whose sole purpose is to improve the user experience by serving information in real time. While a database capturing ephemeral data such as this does not store the master copy of the data, it must be architected to be highly available, as failures could affect user experience resulting in lost revenue.

Still other types of microservices may need to process operational or transactional data. Information gathered from user sessions—such as user activity, shopping cart contents, clicks, likes, etc.—is considered operational data, even though it may be ephemeral in nature. These types of data power instant, real-time analytics and are typically used by microservices that interface directly with users. For this type of data, durability, consistency, and availability requirements are high. On the

other hand, transactional data, such as payment processing and order processing, must be stored as a permanent record in a database. This type of data demands high reliability and must employ a cost-effective means of storage as the volume of transactions grows.

Competitive Landscape

We see three main buckets of competition in the operational DBMS market: incumbent players—e.g., Oracle, IBM, Microsoft, SAP; cloud platforms—e.g., AWS, Microsoft, GCP; and pure-plays—e.g., MongoDB, DataStax, MarkLogic, Couchbase, Neo4j, Redis Labs, and EnterpriseDB.

Exhibit 31
Database Software Market: The Long-Awaited Shake-up
Operational Database Competitive Landscape

Sources: William Blair

Incumbents

Oracle
Oracle is the dominant player in the operational DBMS market based on its leading market share in the RDBMS space and strong reputation for product functionality, security, and performance. Oracle's historical success in this market can be largely attributed to its deep integration into tier-1 applications and the resulting customer partnership that the company forged, especially as tier-1 applications had a long history of running into issues. This made the choice of database vendor hugely strategic and sticky, which Oracle is still benefiting from today.

However, the cloud was a nonlinear transition point for Oracle's database business, and the company did not respond quickly enough, in our view. The key challenge is that Oracle is losing the IaaS battle, which poorly positions the company to retain database market share for the next generation of applications, which are overwhelmingly being built on the top three IaaS platforms. AWS, Azure, and GCP have their own databases or integrate with third-party (mainly open-source) databases.

Oracle has pursued an all-or-none strategy of rejecting easy integration of the Oracle Database into the big CSP platforms, instead focusing on trying to get customers to run their applications (and databases) on the Oracle Cloud. Oracle does not certify or change licensing metrics for customers wanting to use the Oracle Database on competitor clouds (Oracle requires double the number of licenses for using the Oracle Database on competitor clouds, while reducing the available functionality). In addition, Oracle has prevented third-party DBaaS offerings from integrating with the Oracle Cloud. Oracle appears to be betting that its large customers have very large and complex data management environments, which means it will take them a long time to move these to the cloud (if ever).

While this is likely the case for legacy applications, the problem for Oracle is with new applications, where most of the growth in the market lies. As we have noted in this report, if you lose the application, you lose the database—and because the Oracle Cloud is not seeing much adoption, Oracle is losing new applications (and their corresponding databases) to the big CSPs.

On top of this, Oracle has a notorious reputation for locking in its customers, aggressive auditing practices, and draconian pricing. This has created an unfavorable view of Oracle in the minds of many customers and a desire to move off Oracle's databases at the first possible chance. The market disruption from cloud has created a catalyst for customers to do so.

On the plus side, Oracle continues to innovate with new technology. Case in point is Oracle's Autonomous Database, a cloud-based offering that leverages machine learning to allow customers to tune, update, and patch their databases automatically with zero downtime. This approach will eliminate many of the mundane tasks of the DBA and allow for improved performance. Unfortunately, the Autonomous Database is only available on the Oracle Cloud, limiting its market reach.

On the open-source front, while Oracle offers a range of open-source databases, including MySQL, Oracle Berkeley DB, and Oracle NoSQL Database, Oracle is not viewed as a leader in the fastest-growing part of the RDBMS market—nonrelational databases. In the overall database market, as defined by IDC, Oracle lost roughly four points of market share from 2015 to 2017 (from 40.6% to 36.9%).

Exhibit 32
Database Software Market: The Long-Awaited Shake-up
Oracle Database Offerings

Database Enterprise Edition
NoSQL Database Enterprise Edition
Essbase
Exadata
MySQL
Autonomous Database
Berkeley DB
Big Data SQL

Sources: William Blair

IBM

IBM is a pioneer and leading player in the RDBMS market with a host of solutions, led by its long-standing Db2 platform, which is available across different operating systems (Linux, Unix, Windows, mainframe) and cloud environments (Db2 hosted or Db2 on IBM Cloud). While IBM also offers its own graph database (IBM Graph) and a proprietary version of the open-source CouchDB (IBM Cloudant), the company is not a major player in the nonrelational database space.

Like Oracle, the IBM Cloud has seen limited adoption, which we believe leaves IBM poorly positioned to capture the next generation of applications and their corresponding databases. While IBM does offer integration of Db2 hosted into third-party clouds, this service does not seem to be gaining much traction. In the overall database market, as defined by IDC, IBM lost roughly two and a half points of market share from 2015 to 2017 (from 15.7% to 13.2%).

Exhibit 33
Database Software Market: The Long-Awaited Shake-up
IBM Database Offerings

Db2 Database
Db2 on Cloud
Db2 Hosted
Db2 for z/OS
Db2 for i
Informix
Cloudant
Databases on PostgreSQL
Db2 Warehouse on Cloud

Sources: William Blair

Microsoft

Microsoft offers a range of operational DBMS products, including its flagship SQL Server, Azure SQL Database (a relational DBaaS based on SQL Server), and Azure Cosmos DB (a nonrelational, multimodel, document-oriented DBaaS). Microsoft's database market share has grown alongside the growth in its Azure cloud service. While Microsoft has a strong hybrid story overall, some customers complain about shortcomings in Microsoft's hybrid database offerings (including holistic management and security across on-premises environments and the cloud). In the overall database market, as defined by IDC, Microsoft gained roughly three points of market share from 2015 to 2017 (from 24.5% to 27.8%).

Exhibit 34
Database Software Market: The Long-Awaited Shake-up
Microsoft Database Offerings

SQL Server
SQL Database
Azure Database for MySQL
Azure Database for PostgreSQL
Azure Database for MariaDB
Azure CosmosDB
SQL Data Warehouse

Sources: William Blair

SAP

SAP offers several operational DBMS products include SAP Adaptive Server Enterprise (ASE), SAP SQL Anywhere (Sybase), and SAP HANA. Both SAP ASE and HANA are available as cloud offerings. SAP HANA claims more than 25,000 customers as of July 2018, and SAP has integrated HANA into its ERP and CRM offerings (S/4 and C/4). We also note that SAP now offers the OrientDB graph database from recently acquired Callidus Software. On whole, we see SAP as a less impactful stand-alone player going forward due to the absence of a strong IaaS platform and the company's strategic focus on applications versus infrastructure. In the overall database market, as defined by IDC, SAP's market share has held steady from 2015 to 2017 (at 6%).

Exhibit 35
Database Software Market: The Long-Awaited Shake-up
SAP Database Offerings

SAP Hana
SAP Adaptive Server Enterprise
SAP SQL Anywhere
SAP OrientDB
SAP IQ

Sources: William Blair

Cloud Platforms

As noted earlier in the report, each of the big three CSPs offers a menu of in-house operational database services, as well as integration with third-party databases. The main competitive advantage for the cloud vendors is simplicity/one-throat-to-choke, as developers can work in the same environment in which the application is located.

Amazon Web Services leads the way among CSPs with at least nine discrete database offerings, both for relational and nonrelational use-cases. Bucking the multimodel trend, Amazon continues to pursue a product strategy geared toward specific use-cases. Meanwhile, Microsoft Azure offers multiple database services beyond Azure SQL Database and Cosmos DB (see below), several of which are managed services for open-source databases (e.g., PostgreSQL, MySQL, Redis, MariaDB). Lastly, Google Cloud Platform lags a bit behind AWS and Azure, though GCP appears to be catching up with multiple in-house relational (e.g., Cloud Spanner, Cloud SQL) and nonrelational databases (e.g., Cloud Bigtable, Cloud Datastore) now available.

Exhibit 36
Database Software Market: The Long-Awaited Shake-up
Cloud Service Provider Database Offerings

	AWS	Microsoft Azure	Google Cloud
Relational	Amazon Aurora Amazon Compatible with MySQL Amazon compatible with PostgreSQL	SQL Server SQL Database Azure Database for MySQL Azure Database for PostgreSQL Azure Database for MariaDB SQL Data Warehouse	Cloud SQL Cloud Spanner (has nonrelational features) Google BigQuery
Nonrelational	Amazon DynamoDB Amazon Redshift Amazon Elasticache Amazon DocumentDB Amazon Neptune Amazon Timestream	Azure CosmosDB	Cloud Datastore Cloud Bigtable

Sources: William Blair

Pure-Plays

The landscape for pure-play database vendors is fast-evolving and highly fragmented, with over 300 databases listed in DB-Engines monthly popularity ranking, many of which are pure-play offerings (note: not all of these are tied to commercial vendors). With an estimated $4.065 billion of private capital invested in the database space (see exhibit below) over the past 10 years, and with the public cloud providers disrupting the space, database vendor consolidation is inevitable, in our view.

Exhibit 37
Database Software Market: The Long-Awaited Shake-up
Capital Invested in Private Database Companies Since 2009 (in millions)

Cumulative Capital Invested	Companies	Deals	Investors	Largest Individual Deal
$4.065 billion	37	123	271	$450 million

Sources: Pitchbook
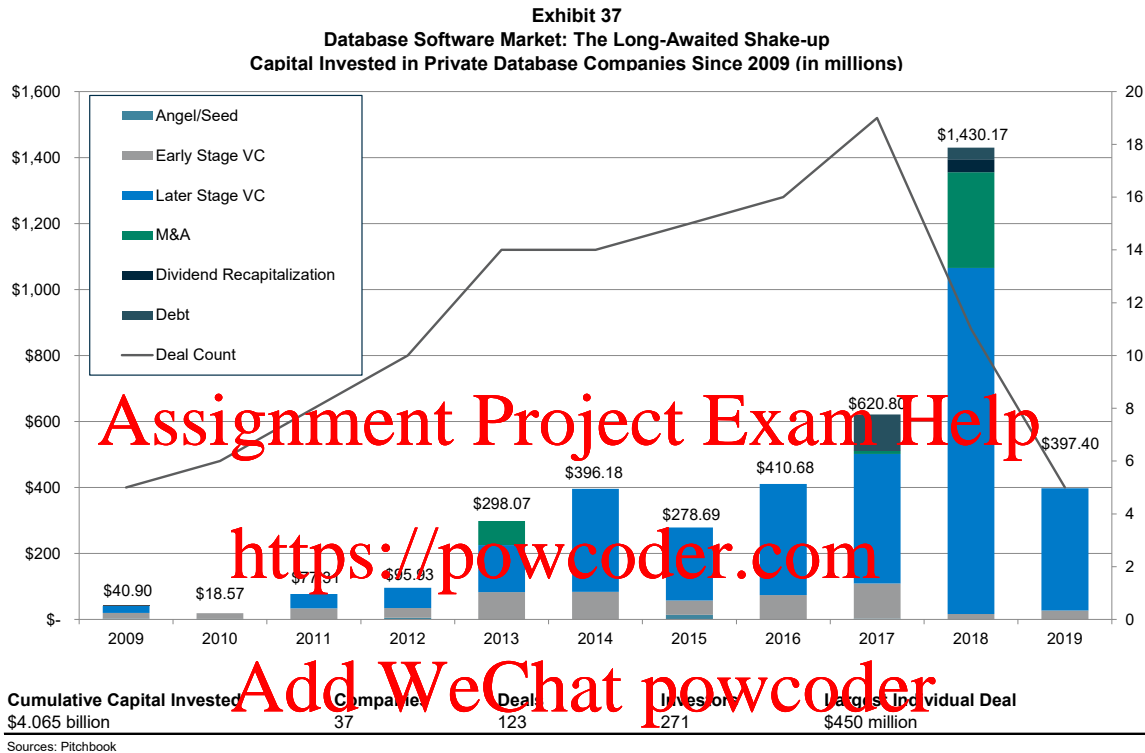
Within the crowded landscape of private companies, we see a handful of upstart vendors separating themselves from the pack that we believe have the potential to support stand-alone existences. These vendors include: (publicly traded) MongoDB, DataStax, Couchbase, MarkLogic, Redis Labs, EnterpriseDB, Neo4j, MemSQL, and InfluxDB. We profile MongoDB in a companion initiation report. The rest are profiled below, along with a host of other privately held database players.

Exhibit 38
Database Software Market: The Long-Awaited Shake-up
Capital Invested in Private Database Companies Since 2009, By Company

Number	Company Name	Last Financing Size (in Millions)	Total Capital Raised Since 2009 (in millions)
1	2ndQuadrant		
2	Actian	$ 40.00	$ 507.60
3	Aerospike	$ 15.83	$ 45.92
4	Almabase	$ 0.15	$ 0.26
5	ArangoDB		$ 7.07
6	BlueTalon	$ 15.98	$ 25.87
7	Cambridge Semantics	$ 9.65	$ 9.65
8	Cockroach Labs	$ 27.00	$ 53.50
9	Couchbase	$ 35.00	$ 164.00
10	Crunchy Data		
11	Databricks	$ 250.00	$ 497.36
12	DataStax		$ 190.79
13	DBmaestro	$ 4.50	$ 7.50
14	Delphix	$ 75.00	$ 120.51
15	Dgraph	$ 3.00	$ 4.06
16	EnterpriseDB		$ 21.93
17	GridGain Systems	$ 5.00	$ 27.50
18	Hazelcast	$ 3.12	$ 16.91
19	Iguazio		$ 48.00
20	InfluxData	$ 60.00	$ 120.73
21	MapR	$ 52.91	$ 380.34
22	MariaDB	$ 7.19	$ 105.09
23	MarkLogic		$ 142.08
24	Memgraph	$ 2.16	$ 2.26
25	MemSQL	$ 30.00	$ 110.93
26	Neo4j	$ 80.00	$ 163.19
27	NuoDB	$ 30.55	$ 95.03
28	OrientDB	$ 8.66	$ 8.66
29	Percona		
30	Redis Labs	$ 60.00	$ 146.00
32	ScaleOut Software	$ 0.05	
33	Snowflake	$ 450.00	$ 922.95
34	Solix Technologies		
35	Starcounter	$ 2.62	$ 10.21
36	TigerGraph	$ 31.00	$ 53.70
37	Timescale	$ 27.40	$ 31.10
38	YugaByte	$ 16.00	$ 24.00
		Total:	**$ 4,065**

Sources: Pitchbook

Exhibit 39
Database Software Market: The Long-Awaited Shake-up
Top 25 Database Vendors from DB-Engines in March 2019

345 systems in ranking, March 2019

Rank			DBMS	Database Model	Score		
Mar 2019	Feb 2019	Mar 2018			Mar 2019	Feb 2019	Mar 2018
1.	1.	1.	Oracle	Relational, Multi-model	1279.14	+15.12	-10.47
2.	2.	2.	MySQL	Relational, Multi-model	1198.25	+30.96	-30.62
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	1047.85	+7.79	-56.94
4.	4.	4.	PostgreSQL	Relational, Multi-model	469.81	-3.75	+70.46
5.	5.	5.	MongoDB	Document	401.34	+6.24	+60.82
6.	6.	6.	IBM Db2	Relational, Multi-model	177.20	-2.23	-9.47
7.	↑9.	7.	Microsoft Access	Relational	146.20	+2.18	+14.26
8.	↓7.	8.	Redis	Key-value, Multi-model	146.12	-3.32	+14.90
9.	↓8.	9.	Elasticsearch	Search engine, Multi-model	142.79	-2.46	+14.25
10.	10.	↑11.	SQLite	Relational	124.87	-1.29	+10.06
11.	11.	↓10.	Cassandra	Wide column	122.80	-0.58	-0.69
12.	12.	↑15.	MariaDB	Relational, Multi-model	84.31	+0.89	+21.21
13.	13.	13.	Splunk	Search engine	81.10	+0.29	+17.44
14.	14.	↓12.	Teradata	Relational		-0.75	+2.76
15.	15.	↑18.	Hive	Relational	73.00	+0.71	+16.00
16.	16.	↓14.	Solr	Search engine	60.01	-0.95	-4.80
17.	17.	17.	HBase	Wide column	58.80	-1.48	-2.14
18.	18.	↑19.	FileMaker	Relational	58.13	+0.34	+3.00
19.	↑20.	↓16.	SAP Adaptive Server	Relational	56.03	+0.29	-6.58
20.	↓19.	20.	SAP HANA	Relational, Multi-model	55.51	-1.03	+6.99
21.	21.	21.	Amazon DynamoDB	Multi-model	54.49	-0.45	+12.04
22.	22.	22.	Neo4j	Graph	48.58	+0.72	+7.68
23.	23.	23.	Couchbase	Document	33.80	-1.78	+0.90
24.	24.	24.	Memcached	Key-value	28.73	-0.72	-2.62
25.	25.	↑26.	Microsoft Azure SQL Database	Relational, Multi-model	27.93	+0.81	+3.31

Sources: DB-Engines.com

Exhibit 40
Database Software Market: The Long-Awaited Shake-up
Developer Database Usage by Type
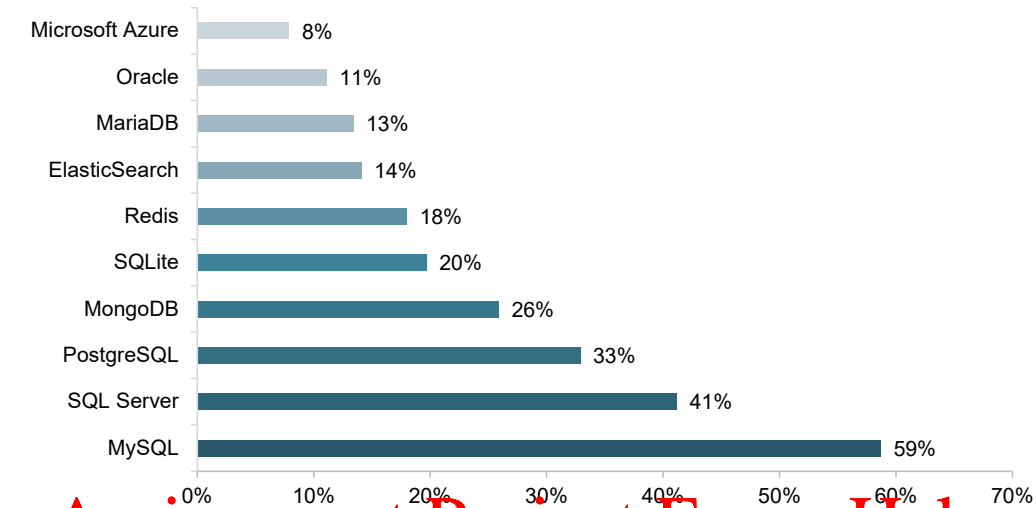
Database	Usage
Microsoft Azure	8%
Oracle	11%
MariaDB	13%
ElasticSearch	14%
Redis	18%
SQLite	20%
MongoDB	26%
PostgreSQL	33%
SQL Server	41%
MySQL	59%

Assignment Project Exam Help

1 Based on 66,264 responses

2 Respondents were able to select multiple databases, so sum is greater than 100%

Source: StackOverflow Developer Survey Results, 2018

https://powcoder.com

Exhibit 41
Database Software Market: The Long-Awaited Shake-up
Top 10 Most Desired Databases by Developers

Add WeChat powcoder

Database	Percent
MongoDB	18.6%
Elasticsearch	12.2%
PostgreSQL	11.4%
Redis	9.7%
MySQL	7.5%
Microsoft Azure	7.3%
Google Cloud Storage	7.3%
Cassandra	6.1%
Amazon DynamoDB	5.7%
Google BigQuery	5.6%

1 Based on 66,264 responses

2 Respondents were able to select multiple databases, so sum is greater than 100%

Source: StackOverflow Developer Survey Results, 2018

Exhibit 42
Database Software Market: The Long-Awaited Shake-up
Top 10 Most Loved Databases by Developers

Database	Percentage
SQL Server	51.6%
Google BigQuery	52.4%
MariaDB	53.3%
MongoDB	55.1%
Google Cloud Storage	56.5%
Microsoft Azure	56.7%
Amazon RDS / Aurora	58.8%
Elasticsearch	59.9%
PostgreSQL	62.0%
Redis	64.5%

1 Based on 66,264 responses
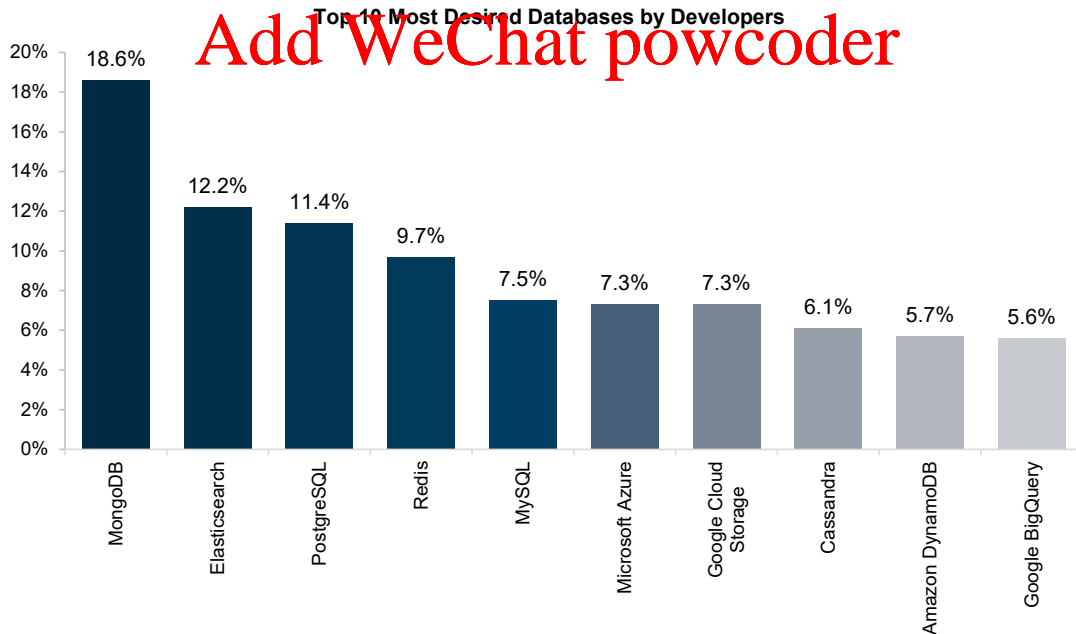2 Respondents were able to select multiple databases, so sum is greater than 100%
Source: StackOverflow Developer Survey Results, 2018

Private Company Profiles

2ndQuadrant
www.2ndquadrant.com

Management:
Simon Riggs, Founder and CTO
Faiz Husain, COO

Description:
2ndQuadrant, founded in 2001 by Simon Riggs, is a major developer and committer of the PostgreSQL RDBMS. The company provides a range of services that underpin the whole database lifecycle of PostgreSQL implementations. Its support engineering team has provided a plethora of code contributions to the PostgreSQL project, making it the only company to contribute enterprise features to all of the 13 latest releases. 2ndQuadrant's customer bases runs the gamut from financial services companies to national broadcasters, including companies such as tastyworks, Think Research, Telefonica del Sur, Animal Logic, Navionics, Agilis Systems, and Healthcare Software Solutions. The company is based in Oxford, United Kingdom.

Actian (acquired by HCL Technologies)
www.actian.com

Management:
Rohit de Souza, CEO
Lewis Black, CFO

Description:
Actian is a hybrid data management, analytics and integration company that was previously known as the Ingres database. Ingres Database is an open-source SQL-based relational database management system intended to support large commercial and government applications. Actian controls the development of Ingres and makes certified binaries available for download, while providing support globally. Its major products include Actian Vector, a columnar database; Actian DataConnect, a hybrid cloud data integration platform; and Actian X, a hybrid database for operational analytics. The company was founded in Palo Alto, California, in 1980 as Relational Technology Inc., before changing its name to Actian in 2011. In April 2018, Actian was acquired in a joint venture by HCL Technologies (retains 80% stake) and Sumeru Equity Partners (retains 20% stake) for a total of $330 million. Actian continues to operate as a separate entity while enhancing HCL's Mode 3 offerings in data management products and platforms. Actian customers include Bloomberg, ACME, Citigroup, Deutsche Bank, Intuit, Northrop Grumman, and Siemens.

Aerospike
www.aerospike.com

Management:
John Dillon, CEO
Srini Srinivasan, Founder and CPO

Description:
Mountain View, California–based Aerospike is a provider of NoSQL database solutions. It was the first SSD and flash-optimized, in-memory, operational NoSQL database with ACID properties and is used by revenue-critical applications to personalize the user experience. It offers enterprise and open-source versions of a NoSQL database that includes tools and packages with features such as

key-value store, flexible data models, user defined functions, geospatial aggregations, and geographic replication. Aerospike serves developers building Java, Node.JS, C#, .NET, C, Python, and Go applications for start-ups and *Fortune* 100 companies, which include Adobe, FlipKart, Kayak, Nielsen, and PayPal. The company was founded in 2009 and has received $30 million in funding over three rounds, led by New Enterprise Associates.

Altibase
www.altibase.com

Management:
James Jang, CEO

Description:
Altibase is a provider of in-memory, open-source relational database management systems addressing mission-critical enterprise use-cases. The Altibase database is highly scalable and ACID-compliant, with sharding ability, an architecture where data is distributed across commodity computers. Customers include HP, Samsung, E*Trade, Hyundai, and China Mobile. The company was founded in 1999 and is based in Seoul, South Korea.

ArangoDB
www.arangodb.com

Management:
Claudius Weinberger, CEO
Frank Celler, CTO

Description:
ArangoDB offers a cloud-based, open-source, NoSQL database for developers and architects. The ArangoDB database is multi-model, meaning that it can address documents, graphs, and key-values, enabling users to build high-performance applications using a SQL-like query language or JavaScript extensions. The company was founded in 2014 in Cologne, Germany, and has received €6.1 million in seed funding from Target Partners. ArangoDB's customers include Thomson Reuters, Liaison, InfoCamere, Oxford University, and Triton.

Blue Talon
www.bluetalon.com

Management:
Eric Tilenius, CEO
Pratik Verma, Founder and CPO

Description:
Blue Talon provides data-centric security along with visibility and control at the data layer across Hadoop, Spark, SQL, and big data. The company positions itself as creating a safe environment that allows users to access data in a governed and secure way. With the open-source platform structure in mind, Blue Talon enforces security policies across multiple data entry points and is integrated into the shared storage space to ensure there is no "dirt road access." Founded in Redwood City, California, the company has raised $27.4 million in funding over four rounds, led by Signia Venture Partners and Maverick Partners.

Cambridge Semantics
www.cambridgesemantics.com

Management:
Charles Pieper, Chairman and CEO
Alok Prasad, President

Description:
Cambridge Semantics operates as an enterprise analytics and data management software company. The company offers data integration, unstructured text integration, spreadsheet integration, meta-data management, data collection and curation, test analytics, and business intelligence solutions to connect and bring meaning to enterprise data. The company's Anzo Smart Data Lake product allows IT departments and business users to semantically link, analyze, and manage diverse data whether internal or external, structured or unstructured at big data scale and at the fraction of the implementation costs of using traditional approaches. The company was founded in 2007 and is based in Boston, Massachusetts, and its customers include the U.S. Air Force, Bristol-Myers Squibb, Credit Suisse, PwC, Lockheed Martin, Sanofi, and Staples.

Cockroach Labs
www.cockroachlabs.com

Management:
Spencer Kimball, Co-founder and CEO
Peter Mattis, Co-founder and VP of Engineering
Benjamin Darnell, Co-founder and CTO

Description:
New York City–based Cockroach Labs offers database software for application development through its open-source, distributed SQL database, CockroachDB. This database enables developers to build applications that can survive outages at the datacenter level through its ability to store copies of data in multiple locations and deliver requested data whenever it is needed. The company's disaster-recovery solutions support strongly consistent ACID transactions and can survive disk, machine, rack, or data center failures with minimal latency disruption and no manual intervention. Cockroach Labs was founded by ex-Google employees in 2015 and has received $53.5 million in funding over three rounds. Its major investors include Redpoint, Index Ventures, Benchmark, and Google Ventures. The company's customers include Baidu, Kindred, Bose, Rubrik, MetroNOM, and Tierion.

Couchbase
www.couchbase.com

Management:
Matt Cain, President and CEO
Greg Henry, CFO

Description:
Couchbase was formed through the merger of Membase and CouchOne in February 2011. The merged company aimed to build an easily scalable, high-performance NoSQL database system taking the best of both Membase and CouchOne technologies. Today, Couchbase is a leading commercial database vendor that helps enterprises build customer-engagement applications for web, mobile, and IoT use-cases that support massive data volumes in real time. The company's suite of platforms includes Couchbase Server, a high-performance NoSQL distributed database; Couchbase Lite, an embedded NoSQL database that lives locally on mobile devices; and Couchbase Sync Gateway, an internet-facing cloud component that securely synchronizes data between the mobile device and

the cloud. Customers include Amadeus, AT&T, Becton, Dickinson, Carrefour, Cisco, Comcast, Disney, DreamWorks Animation, eBay, Marriott, Neiman Marcus, Tesco, Tommy Hilfiger, United, Verizon, and Wells Fargo, and is based in Santa Clara, California. The company has raised funding totaling $146 million from investors that include Accel, Mayfield, Ignition Partners, Adams Street Partners, WestSummit Capital, North Bridge Venture Partners, and Sorenson Capital.

Crunchy Data
www.crunchydata.com

Management:
Bob Laurence, Co-founder and CEO
Paul Laurence, Co-founder

Description:
Crunchy Data is a provider of enterprise PostgreSQL support and open-source solutions. Its flagship product, Crunchy Certified PostgreSQL is an unmodified PostgreSQL packaged with popular extensions, including PostGIS (added support for spatial and geographic objects), and other important enterprise-grade features like audit logging. The company also provides subscription-based enterprise support services for assistance with troubleshooting, performance tuning, version upgrades, and other tasks. Crunchy Data was founded in 2012 with the mission of bringing PostgreSQL to security-conscious organizations while eliminating the need to use the expensive proprietary offerings of other providers. It started by working with the U.S. Department of Defense, where it developed its expertise in security and compliance and originated its inaugural PostgreSQL packages. The company is based in Charleston, South Carolina.

Databricks
www.databricks.com

Management:
Ali Ghodsi, Co-founder and CEO
Matei Zaharia, Co-founder and Chief Technologist
Reynold Xin, Co-founder and Chief Architect

Description:
Databricks provides customers with cloud-based big data processing solutions through the use of Apache Spark. The company was founded in 2013 by the team of UC Berkeley professors behind the creation of Apache Spark, a fast, in-memory data processing engine that allows users to execute streaming, machine learning, or SQL workloads requiring fast iterative access to data sets. Databricks provides a Unified Analytics Platform aimed at building digital pipelines between siloed data storage systems, helping engineers and data scientists communicate better. In 2017, the company partnered with Microsoft to debut Azure Databricks, a tool for processing and analyzing large streams of corporate data, which was integrated with other Azure data-related services including Azure Cosmos DB database, Azure SQL Data Warehouse, and Azure Active Directory. The company's customers include Viacom, HP Inc., Shell, Nielsen, Finra, Regeneron, and Sanford Health, among others. Databricks has raised a total of $497 million over five funding rounds, which most recently included a $250 million series E round in February 2019 lead by Andreessen Horowitz (other investors include Battery Ventures, Green Bay Ventures, Microsoft, and New Enterprise Associates). The company is based in San Francisco, California.

DataStax
www.datastax.com

Management:
Billy Bosworth, CEO
Jonathan Ellis, SVP and CTO
Robert O'Donovan, CFO

Description:
Santa Clara, California–based DataStax was founded in 2010 as Riptano and later changed its name to DataStax in 2011. The company's claim to fame is the primary commercial vendor supporting the open-source Apache Cassandra NoSQL database. Today, the company's suite of solutions includes DataStax Enterprise, a database for cloud applications that builds on Apache Cassandra used for online applications that require fast performance with no downtime; DataStax OpsCenter, a web-based visual management and monitoring solution for Apache Cassandra and DataStax Enterprise that provides architects, DBAs, and operations staff with capabilities to ensure their databases are running well; and DataStax DevCenter, which allows developers to create and run CQL (Cassandra query language) queries and commands against Apache Cassandra and DataStax Enterprise. Customers include Safeway, eBay, Samsung, ING, Macquarie, McDonald's, Delta Airlines, Macy's, Comcast, and Walmart. DataStax has raised a total of $190 million of funding through seven rounds, with investors that include Crosslink Capital, Meritech Capital Partners, Scale Venture Partners, and Kleiner Perkins. In 2016, DataStax acquired Datascale, which brought to the portfolio the Titan scalable graph database, which optimized for storing and querying graphs containing billions of vertices and edges distributed across a multi-machine cluster.

DBmaestro
www.dbmaestro.com

Management:
Yariv Tabac, Co-founder and CEO
Yaniv Yehuda, Co-founder and CTO

Description:
DBmaestro's flagship product—TeamWork—enables agile development and continuous integration and delivery for the database. The company's platform enables its customers to accelerate their overall application release cycle with database release automation solutions; increase the productivity of development teams; gain complete security, policy compliance, and transparent audits of its databases; and scale to support multi-database enterprise environments. DBmaestro was founded in 2008 and is based in Concord, Massachusetts, with its main R&D center located in Israel. Its customers include T-Mobile, Barclays, Visa, ING, BNY Mellon, GM, The Hartford, Allianz, Grupo Sura, and ING. The company has received $7.5 million in funding over three rounds from Vertex Ventures, StageOne Ventures, and lool ventures.

Delphix
www.delphix.com

Management:
Jedidiah Yueh, Founder and Executive Chairman
Chris Cook, President and CEO
Stewart Grierson, CFO

Description:
Delphix's Dynamic Data Platform delivers data-as-a-service (DaaS) solutions to help enterprises secure, automate, and accelerate application projects including ERP rollouts, custom developments, and migrations to private and public clouds. The platform secures data with automated and custom masking to ensure compliance before distribution; virtualizes data with compression to create lightweight virtual copies continuously synced with source data; enables easy replication of data for cloud migration or backup/DR; and enables easy viewing and management of test and copy data environments across users. Delphix's customers include AT&T, Clorox, Fannie Mae, HPE, Toyota, along with more than 30 of the *Fortune* 300. Founded in 2008 and based in Redwood City, California, the company has raised a total of $119.5 million in four funding rounds from investors such as Fidelity Investments, Icon Ventures, Lightspeed, and Greylock Partners.

Devart
www.devart.com

Management:
Patrik Gondža, CTO

Description:
Devart develops database tools for SQL Server, MySQL, Oracle, PostgreSQL, and Interbase and produces native connectivity solutions. Founded in 1997 and based in Prague, Czech Republic, the company has more than 40,000 customers including *Fortune* 100 and *Fortune* 500 companies, governmental agencies, educational and scientific institutions, freelancers, and private individuals. Devart is a Microsoft Silver Application Development Partner and a Silver Partner in the Oracle Partner Network Specialized program. Most recently, Devart announced an upgrade to its dbForge Studio for PostgreSQL upgrade, a graphical user interface tool for database development and management. The company's customers include 20th Century Fox, American Cancer Society, Boston University, Juventus F.C., T-Mobile, Unisys Corp, and SpaceX.

DGraph Labs
www.dgraph.io

Management:
Manish Jain, Founder and CEO

Description:
DGraph Labs provides graph database solutions. The company's DGraph is an open-source, horizontally scalable and distributed graph database, providing ACID transactions, consistent replication, and linearizable reads. DGraph positions itself as enabling webscale throughput, with low enough latency to serve real-time user queries over terabytes of structured data. DGraph supports GraphQL-like query syntax and responds in JSON and Protocol Buffers over GRPC and HTTP. Founded in 2016, the company is based in San Francisco and has received $3 million in seed funding from Bain Capital Ventures.

EnterpriseDB
www.enterprisedb.com

Management:
Ed Boyajian, President and CEO
Paul Blondin, SVP and CFO

Description:
EnterpriseDB provides enterprise-class products and services based on the open-source PostgreSQL database. PostgreSQL is one of the oldest and most popular relational databases supporting many of the world's largest enterprises and government agencies. The EnterpriseDB Postgres Platform consists of four major components, starting with two database options (open-source PostgreSQL or the proprietary EDB Postgres Advanced Server), then adds enterprise level tools (e.g., backup and recovery, replication, migration, management interface) and deployment options (virtualized, bare metal, containers, or public cloud), and finally service and support. The company was founded in 2004 and is based Bedford, Massachusetts. It currently has over 300 employees and has delivered a six-year subscription CAGR of 37%. Customers include ABN Amro, Rabobank, KT Corporation (formerly Korea Telecom), Ericsson, and U.S. Army. EnterpriseDB received $69.4 million in venture funding over six rounds from investors including Fidelity Ventures, CRV, and Valhalla Partners. In 2014, the company was acquired by private-equity firms Peak Equity Partners, NewSpring Capital, and NSI Stone Partners for an undisclosed amount.

GridGain Systems
www.gridgain.com

Management:
Abe Kleinfeld, CEO
Eoin O'Connor, CFO
Nikita Ivanov, CTO

Description:
GridGain is a provider of enterprise-grade in-memory computing solutions based on Apache Ignite, an open-source project for the Java, .NET, and C++ programming languages. The GridGain in-memory database can be used to power OLTP, OLAP, or HTAP use-cases and can be deployed on-premises, in the public cloud, or a hybrid environment. GridGain distributes a data set across a cluster of servers while the distributed SQL capabilities allow customers to read and write to a database using standard database commands through the ODBC/JDBC interface. The memory-centric Grid Gain architecture allows customers to execute distributed SQL, key-value, and other operations across different memory layers. For example, organizations that deploy a variety of memory technologies like DRAM, non-volatile memory, and 3D XPoint, can tune the configuration of their system to use a combination of memory options, which provides the best trade-off between price and performance. Based in Foster City, California, GridGain was founded in 2010 and has raised $38.6 million in funding over six rounds from investors that include Almaz Capital and FortRoss Ventures. Customers include ING, American Express, Société Générale, Microsoft, Workday, Newegg, Pitney Bowes, and e-therapeutics.

Hazelcast
www.hazelcast.com

Management:
Kelly Herrell, CEO
Marion Smith III, CFO
Greg Luck, CTO

Description:
Hazelcast positions itself as a leading in-memory data management company with millions of productive clusters and over 60 million server starts per month. The Hazelcast IMDG (In-Memory Data Grid) platform helps companies manage their data and distribute processing, using in-memory storage and parallel execution for faster application speed and scale. In a Hazelcast grid, data is evenly distributed among the nodes of a computer cluster, allowing for horizontal scaling of processing and available storage. It can run on-premises, in the cloud, in virtualized environments, and in Docker containers. The company also provides Hazelcast Jet, an application-embeddable, distributed computing platform for fast processing of big data sets and based on a parallel-streaming core engine, which enables data-intensive applications to operate at near real-time speeds. Founded in 2008 and based in San Francisco, the company has received $13.6 million in funding over three rounds from investors that include Earlybird Venture Capital and Bain Capital Ventures. Customers include TD Bank, UBS, T Systems, Ebay, HSBC, and Lloyds Banking Group.

Iguazio Systems
www.iguazio.com

Management:
Asaf Somekh, Founder and CEO
Yaron Haviv, Founder and CTO
Orit Nissan-Messing, VP R&D and Co-founder

Description:
Iguazio provides data management and storage solutions for continuous data science applications. It allows data scientists to focus on delivering solutions by reducing the time spent on data infrastructure, management, and deployment. The company also focuses on applications related to IoT, big data, and cloud infrastructure by simplifying the development and deployment of high-volume, real-time, data-driven applications. The company was founded in 2014 and is based in Tel Aviv, Israel. It raised a $15 million series A round in November of 2015 from Magma Venture Partners and $33 million in series B funding from lead investor Pitango Venture Capital. Its customers and partners include Bosch, Verizon Ventures, CME Group, Intel, Dell-EMC, and Equinix.

InfluxData
www.influxdata.com

Management:
Evan Kaplan, CEO
Winnie Cheng, CFO
Paul Dix, Founder and CTO

Description:
InfluxData delivers an open-source, time-series database platform, built from the ground up to manage time-series data at scale for DevOps and IoT applications. InfluxData empowers developers to build next-generation monitoring, analytics, and IoT applications, faster, easier, and to scale. The InfluxData Platform is composed of InfluxDB, a time-series data storage solution; Telegraf, a

metrics and events reporting tool; Chronograf, a management interface; and Kapacitor, a real-time data streaming data processing engine. The platform has two major deployment models: InfluxEnterprise, a highly scalable cluster that runs on an enterprise's own infrastructure, and InfluxCloud, a fully managed SaaS offering. Both models include a hardened version of the open-source core (TICK stack), clustering for high availability and scalability, and automated backup and restore functionality. Customers include Cisco, Citrix, eBay, Houghton Mifflin Harcourt, Siemens, IBM, and Twitter. Founded in 2012 and based in San Francisco, the company announced in February 2019 that it raised $60 million in a series D round led by Norwest Venture Partners and joined by Sorenson Capital and existing investors Sapphire Ventures, Battery Ventures, Mayfield Fund, Trinity Ventures, and Harmony Partners. The round brought the company's total capital raised to $119.9 million.

InterSystems
www.intersystems.com

Management:
Philip Ragon, Founder and CEO
Paul Grabscheid, VP of Strategic Planning

Description:
InterSystems offers high-performance database management, integration, and health information systems catering to healthcare providers, businesses, and governments. The company's IRIS Data Platform is designed for rapid development and deployment of low-latency, sensitive applications, supporting transactional, analytic, and transactional-analytic applications. IRIS scales vertically and horizontally to efficiently accommodate increasing workloads, data sizes, and concurrency. InterSystems Caché functions as a data management system, within which data can be modeled and stored as tables, objects, or multidimensional arrays. Different models can then access data without the need for performance-killing mapping between models and a network of many servers, behaving like a single data store to enhance scalability and performance of distributed applications. The company also offers InterSystems HealthShare, which leverages a healthcare informatics platform for hospitals, integrated delivery networks, and national health information exchanges. InterSystems was founded in 1978 and is based in Cambridge, Massachusetts. The company claims that healthcare services for two-thirds of the U.S. population run on its software. Key customers include Kaiser Permanente, TD Ameritrade, the European Space Agency, and Johns Hopkins Hospital.

MapR Technologies
www.mapr.com

Management:
John Schroeder, CEO and Chairman
Dan Atler, Executive Vice President and CFO
Ted Dunning, CTO

Description:
MapR is a business software company that provides access to a variety of data sources from a single computer cluster. It provides access to big data workloads like Apache Hadoop and Apache Spark as well as analytics solutions for mirroring, snapshots, and data placement control. The company enables enterprises to inject analytics into their business processes to increase revenue and mitigate costs. MapR addresses the issues associated with data complexity of high-scale and mission critical distributed processing from the cloud to the edge, IoT analytics, and container persistence. Customers include Audi, Cisco, Criteo, Ericsson, Eastern Bank, Credit Agricole, Idexx Labs, HP, Novartis, SAP, TransUnion, and United Healthcare. The company was founded in 2009 and is based in Santa Clara, California. It has raised $280 million through seven rounds of funding from investors including Lightspeed Venture Partners, Future Fund, CapitalG, and Redpoint.

MariaDB
www.mariadb.com

Management:
Michael Howard, CEO
Kenneth Paqvalen, CFO
Michael Widenius, CTO

Description:
MariaDB develops and delivers the MariaDB open-source relational database, a community-developed, commercially supported fork of the popular MySQL database. The company positions MariaDB Platform as the enterprise open-source database for hybrid transactional/analytical processing at scale. With the ability to de deployed on commodity hardware or any of the major public clouds, the platform's pluggable, purpose-built storage engines support workloads that previously required a variety of specialized databases. Deployed in minutes for transactional, analytical, or hybrid use-cases, MariaDB delivers operational agility without sacrificing key enterprise features, including real ACID compliance and full SQL capabilities. Customers include Deutsche Bank, Nasdaq, Red Hat, Home Depot, Nokia, ServiceNow, and Verizon. MariaDB was founded in 2014 in Espoo, Finland, and has received funding totaling over $98 million from investors including Alibaba Group, ServiceNow, Intel Capital, SmartFin Capital, and the European Investment Bank.

MarkLogic
www.MarkLogic.com

Management:
Gary Bloom, CEO
Peter Norman, CFO
Christopher Lindblad, Founder

Description:
MarkLogic offers an operational and transactional enterprise-grade, NoSQL database platform. The company's Data Hub technology and cloud services allow enterprises to gain a unified, 360-degree view of data with the quality, governance, and curation required to power AI and analytical systems, and without the inefficiencies of traditional ETL technology. MarkLogic's flexible, multimodel database enables customers to bring in data from anywhere—relational databases, mainframes, file servers, Hadoop—while delivering core enterprise features such as ACID compliance, advanced security, and built-in search. Among the company's over 2,000 global enterprises and government customers are Northern Trust, ABN Amro, Sony, NBC, Healthcare.gov, Abvie, Autoliv, BBC, KPMG, and Cisco. The company was founded in 2001 and is based in San Carlos, California. It has received a total of $173.2 million in funding over eight rounds from investors that include Sequoia Capital, Wellington Management, Tenaya Capital, and NTT Data.

MemGraph
www.memgraph.io

Management:
Dominik Tomicevic, Founder and CEO

Description:
MemGraph offers a real-time, scalable graph database built for the cloud with integrated streaming capabilities. The company targets connected data use-cases such as real-time fraud detection, real-time risk and trade analytics, IT network management, and customer intelligence. It provides a platform that is able to dynamically balance data and queries across a cluster of

industry standard hardware for massively parallel performance, concurrency, and availability. The in-memory, ACID-compliant architecture features highly concurrent data structures, multi-version concurrency control, and asynchronous I/O. The company's commercial offerings include MemGraph Core for independent users, and MemGraph Enterprise for customers with greater data needs. The company was founded in London in 2016 and has raised £1.7 million in seed funding from Connect Ventures.

MemSQL
www.memsql.com

Management:
Nikita Shamgunov, Co-founder and CEO
Adam Prout, CTO

Description:
MemSQL offers a real-time, massively scalable database targeting large enterprise customers. The product combines ultra-high-performance ingestion and query response, SQL scalability, unified transactions and analytics, enterprise security, and flexible deployment (public cloud or on-premises). Use-cases include real-time analytics, IoT, real-time data pipelines, risk management, infrastructure consolidation, Hadoop acceleration, and converged processing. Founded in 2011 and based in San Francisco, MemSQL has received six rounds of funding totaling $110.1 million from investors including Accel, Caffeinated Capital, Data Collective, First Round Capital, IA Ventures, REV, In-Q-Tel, and Khosla Ventures. Customers include Comcast, Dell, Kellogg's, Uber, Samsung, Akamai, Cisco, and Sony.

Neo Technology / Neo4j
www.neo4j.com

Management:
Emil Eifrem, CEO
Mike Asher, CFO
Lars Nordwall, President and COO

Description:
Neo Technology, also known as Neo4j, positions itself as a leading graph database platform. Its flagship product, the Neo4j Graph Platform, helps organizations make sense of their data by taking a connections-first approach to reveal how people, processes, locations, and systems are interrelated. This connections first approach powers applications tackling artificial intelligence, fraud detection, real-time recommendations, and master data. The company boasts the world's largest dedicated investment in graph technology, has amassed more than 20 million downloads, and has a huge developer community deploying graph applications around the globe. From its initial seed funding round in October 2009 to the latest series E funding raised in November of 2018, the company has raised a total of $160.1 million from venture groups including One Peak Partners and Sunstone Capital. Customers include Walmart, eBay, Adobe, UBS, IBM, Volvo, Microsoft, AstraZeneca, Monsanto, Telenor, Telia, Zurich Insurance Group, Airbnb, NASA, Orange, Swiss Air, and Financial Times.

NuoDB
www.nuodb.com

Management:
Bob Walmsley, President and CEO
Roger Blanchette, CFO

Description:
NuoDB offers a distributed SQL database that helps enterprise organizations solve the highly complex database challenges they face when trying to move enterprise-grade workloads to the cloud. NuoDB ensures that mission critical, SQL-dependent applications are capable of maximizing performance by providing a distributed, highly available, container-native solution. NuoDB's architecture is designed to deliver five core requirements: 1) horizontal scale-out, 2) zero downtime, 3) hardware and software fault tolerance, 4) multi-site operation for business continuity, and 5) automatic load balancing. The company was founded in 2010 and is based in Cambridge, Massachusetts. It has raised $82.5 million in funding over eight rounds from investors including Morgenthaler Ventures, Longworth Venture Partners, Temenos, Dassault Systems, and HWVP.

OrientDB (acquired by SAP)
www.orientdb.com

Management:
Luca Garulli, Founder and CEO

Description:
OrientDB's core offering OrientDB is an open-source, distributed multimodel database built on top of a graph database engine. Each month, the OrientDB project is downloaded more than 80,000 times. The company offers a Community Edition of OrientDB for small customers and developers, and OrientDB Enterprise, a commercial software version, for larger customers. The company also offers solutions for sync and migration of relational databases as well as the ability to embed OrientDB into applications. The company was founded in 2011 and based in London. At the end of 2017, OrientDB was acquired by Callidus Software for its high-performance transaction graph database technologies. In April 2018, SAP completed the acquisition of Callidus Software, bringing OrientDB under the SAP umbrella of companies. Customers include Ericsson, the UN, Pitney Bowes, Sky, CenturyLink, Accenture, Dell, Comcast, Cisco, and Sonatype.

Percona
www.percona.com

Management:
Peter Zaitsev, Founder and CEO
Vadim Tkachenko, Founder and CTO

Description:
Percona provide enterprise-class support, consulting, managed services, training, and software for MySQL, MariaDB, MongoDB, PostgreSQL, and other open-source databases in on-premises and cloud environments. Customers include Cisco Systems, Time Warner Cable, Alcatel-Lucent, Groupon, and the BBC, as well as to many smaller companies looking to maximize application performance while streamlining database efficiencies. Well established as thought leaders, Percona experts author content for the Percona Performance Blog. In addition, the popular Percona Live conferences draw attendees and acclaimed speakers from around the world. The company was founded in 2006 and is based in Raleigh, North Carolina. In April 2015, Percona acquired Tokutek, a company developing algorithms to accelerate key database operations.

data for analytics, compliance, infrastructure optimization, and data security. Solix EDMS, from a single web-based console database, offers archiving, test data management, data masking, and application retirement solutions. Solix was founded in 2000 in Parsippany, New Jersey, as NECA Services, changing its name to Solix in November 2005. The company's customers include Citi, Banco Stantander, Frost & Sullivan, Maximus, and BAE Systems.

Splice Machine
www.splicemachine.com

Management:
Monte Zweben, Co-founder and CEO
Gene Davis, Co-founder and VP of Product Management
Eran Pilovsky, CFO

Description:
Splice Machine is a developer of a hybrid data platform using operational AI, designed to unify streaming, analytics, and transactions in a single relational database system (powered by Apache Hadoop and Apache Spark). Its RDBMS is used for various applications that include digital marketing, ETL acceleration, operational analytics, and IoT use-cases. The company's platform leverages ANSI SQL support to accelerate offloading and analytical workloads from expensive Oracle, Teradata, and Netezza systems. It seamlessly integrates analytics and AI into mission-critical applications, enabling clients to remove latency, cost, and complexity from supporting modern big data applications. The company's customers include Anthem, Kaiser Permanente, Wells Fargo, Cetera Financial Group, Infinera, and Kroger. Splice Machine has raised a total of $47 million over four rounds, the most recent of which was a Series B round for $16 million in February 2019. Investors in the company include Mohr Davidow Ventures, InterWest Partners, Salesforce Ventures, GreatPoint Ventures, and Accenture. The company was founded 2012 and is based in San Francisco, California.

Starcounter
www.starcounter.com

Management:
Carl-Johan Runer, CEO
Johan Kalderen, CFO
Dan Skatov, CTO

Description:
Starcounter is an in-memory database engine and application server for ultra-fast development of high-performance business applications. All applications/modules built on Starcounter are fully compatible with each other through their patented, AI-driven cognitive mapping and blending Polyjuice solution. Polyjuice allows for data and screens to be shared without any integrations between applications, such that data fields, images, graphics, and actions from many apps can blend together on the screen to support a specific use-case. Founded in 2006 in Stockholm, Sweden, Starcounter has more than 40 employees and five patents. It has raised a total of $27.7 million in funding over seven rounds led by Industrifonden.

TigerGraph
www.tigergraph.com

Management:
Yu Xu, Founder and CEO
Todd Blaschka, COO

Description:
TigerGraph is a provider of a graph database platform for enterprise applications. Through its Native Parallel Graph product, it provides a complete, distributed, parallel graph computing platform, supporting web-scale data analytics in real-time. TigerGraph's technology is used by customers including Alipay, Visa, SoftBank, State Grid Corporation of China, Wish, and Elementum. Founded in 2012 in Redwood City, California, the company has raised a total of $31 million over two rounds of funding. Investors include Ant Financial, Baidu, and Qiming Venture Partners.

Timescale
www.timescale.com

Management:
Ajay Kulkarni, Co-founder and CEO
Michael Freedman, Co-founder CTO

Description:
Timescale is an open-source, time-series database optimized for fast ingest and complex queries, available on GitHub under the Apache license. TimescaleDB is engineered up from PostgreSQL and is packaged as an extension. It scales to 100 billion-plus rows even on a single server and natively supports standard SQL and relational database features. In September 2018, the company announced the release of TimescaleDB 1.0, paving the way for a rollout of its platform for enterprises. Founded in 2015, the company is based in New York City. It has received $31.1 million in funding over three rounds from Icon Ventures, Benchmark, and New Enterprise Associates. The company's customers include Eastlink Capital, Elementum, Hortonworks, Splunk, Stanford University, and Wish.

VoltDB
www.voltdb.com

Management:
David Flower, President and CEO

Description:
VoltDB is an in-memory SQL database that combines streaming analytics and transaction processing into a single platform. VoltDB powers application that require real-time intelligent decision on streaming data (using machine learning–based decision-making) without compromising on ACID requirements. Its relational database adds horizontal partitioning, active-active redundant clustering, full disk persistence, and low long-tail latency among other features to its operational app platform to provide sophisticated decision making in milliseconds. The company's customers include HPE, Nokia, Huawei, Openet, Flipkart, and *Financial Times.* Founded in 2009, VoltDB has raised $31.6 million in funding over five rounds with contributing investors that include Sigma Prime Ventures, Kepha Partners, and Sigma Partners. The company is based in Bedford, Massachusetts.

Redis Labs
www.redislabs.com

Management:
Ofer Bengal, Co-founder and CEO
Yiftach Shoolman, Co-founder and CTO

Description:
Redis Labs is the main commercial vendor selling Redis (short for "Remote Dictionary Server"), an open-source (BSD licensed), in-memory data structure store, used as a database, cache, and message broker. It combines the best of in-memory, schema-less design with optimized data structures and versatile modules to deliver high performance for instant experience, while delivering linear scaling for high performance. The company's flagship product is Redis Enterprise, which extends the open-source Redis platform with additional tools and services for enterprises, including active-active geo distribution, high availability, and built-in search. The company also offers managed cloud services, which give businesses the choice between hosting on public clouds like AWS, GCP, and Azure, as well as in their private clouds. For the second year in a row, Redis was voted the most loved database in the Stack Overflow Developer Survey, meaning that proportionally more developers wanted to continue working with it than any other database. Founded in 2008 and based in Mountain View, California (with primary R&D in Israel), the company has raised a total of $146 million in funding from investors that include Goldman Sachs, Bain Capital Ventures, Dell Technologies Capital, Carmel Ventures, Francisco Partners, and Viola Ventures. Italian software engineer Salvatore Sanfilippo, the original developer of Redis, joined Redis Labs in 2015 to lead open-source development. Redis Labs currently employs over 225 people across offices in California, London, and Tel Aviv. More than 8,300 customers, including American Express, Atlassian, Microsoft, Vodafone, Walmart, Dreamworks, and United Health, along with 6 of the *Fortune* 10 and 40% of the *Fortune* 100, subscribe to its Redis Cloud service and over 250 use its on-premises Redis Labs Enterprise Cluster solution.

ScaleOut Software
www.scaleoutsoftware.com

Management:
William Bain, Founder and CEO
David Brinker, COO

Description:
ScaleOut Software is a provider of software for in-memory data storage and integrated, real-time computing. Its key products include ComputeServer, StateServer, and GeoServe and combine a scal-able, highly available in-memory data grid that eliminates bottlenecks in storage of fast-changing data within server farms and other operational systems, and across multiple data centers. The company also offers hServer, the first in-memory, API-compatible, Hadoop MapReduce execution engine. Founded in 2003 in Bellevue, Washington, ScaleOut Software received angel seed funding in 2004. Its customers include, Angie's List, GoDaddy, Delta, Green Mountain Coffee, Adidas, and Farmers Insurance.

ScyllaDB
www.scylladb.com

Management:
Dor Laor, CEO
Avi Kivity, CTO

Description:
ScyllaDB offers a NoSQL column-store database with increased speed and lower latency compared to Apache Cassandra. ScyllaDB functions as an open source drop-in replacement for Cassandra with a close-to-the-hardware design that can deliver scale-up performance of 1 million IOPS per node, and scale out to hundreds of nodes. The company makes available a free community edition, but also sells an enterprise and cloud version running the Scylla database. The company's customers include Comcast, AppNexus, AdGear, CERN, IBM, and Samsung, among others. Founded in December of 2012 under the name Cloudius Systems, the company is based in Tel Aviv, Israel. It has received $35 million in funding over four rounds from investors that include TLV Partners, Magma Ventures, Qualcomm Ventures, Bessemer Venture Partners, and Samsung Ventures.

Snowflake
www.snowflake.com

Management:
Bob Muglia, CEO
Benoit Dageville, Co-founder and CTO

Description:
Snowflake positions itself as a developer of the only cloud-based data warehousing software solution worldwide. The company offers data warehouse architecture that provides complete relational database support for both structured data (CSV files and tables) as well as semi-structured data (JSON, Avro, Parquet, etc.), all within a single, logically integrated solutions. It essentially offers data warehouse-as-a-service, which features separate compute, storage, and cloud services that can scale independently and require no management. The company also offers Virtual Private Snowflake (VPS), a solution for industries demanding the highest level of security compliance such as financial services. Its customers include Adobe Systems, Sony Pictures, Square, Logitech, DoorDash, Lionsgate, and Capital One, among others. Snowflake has raised a total of $929 million over seven rounds from investors like Sequoia Capital, Altimeter Capital, ICONIQ Capital, Reporting, and Sutter Hill Ventures. The company was founded in 2012 and is based in San Mateo, California.

Solix
www.solix.com

Management:
Sai Gundavelli, Founder and CEO
John Ottman, Executive Chairman

Description:
Solix provides program and process management, regulatory compliance, and customer care services for businesses and government agencies in the United States. It offers call center solutions in the areas of consumer engagement, program management, and specialized teleservices, among others. It provides its services through two platforms: Solix Common Data Platform (Solix CDP), a unified big data management platform, and Solix Enterprise Data Management Suite (Solic EDMS), a comprehensive information lifecycle management software. Solix CDP leverages open-source big data technology to help organize, manage, and process all of a company's structured and unstructured

YugaByte, Inc.
www.yugabyte.com

Management:
Kannan Muthukkaruppan, Co-founder and CEO
Karthik Ranganathan, CTO

Description:
YugaByte develops open-source, cloud-native database software that runs on its YugaByteDB transactional, high-performance database. YugaByteDB claims that it is the only distributed database that is both non-=relational and relational at the same time. The database is supportive of transactional NoSQL APIs and distributed SQL (i.e., PostrgreSQL-compatible) on a document store, leveraging its scalability to power internet-scale online services. YugaByte liberates developers from the risk of lock-in by a particular provider of cloud-compute and reduces overall programming complexity. The company was founded in February 2016 and is based in Sunnyvale, California. YugaByte's founders are the engineers that led the massive scaling of Facebook's NoSQL platform, which powers Facebook Messenger and its internal time series monitoring system. YugaByte has received $24 million in funding from Dell Technologies Capital and Lightspeed Venture Partners.

Appendix: Glossary of Terms

ACID: ACID is a model for database design composed of four characteristics that a DBMS should strive for: Atomicity, Consistency, Isolation, and Durability. The atomicity standard states that database modification must follow an "all or nothing" rule, such that if one part of the transaction fails, the entire transaction fails. Consistency states that only valid data is written into the database, such that any executed transaction that is a violation of a database's consistency rules is rolled back and the database is restored to a state consistent with those rules. Isolation requires that multiple transactions occurring at the same time not impact each other's execution. Durability ensures that any transaction committed to the database will not be lost. Databases can enforce the ACID model through write-ahead logging (WAL), where any transaction detail is first written to a log that includes both redo and undo information, ensuring a back-safe in case of failure. Another method is shadow-paging, in which a shadow page is created when the data is to be modified and the query's updates are written to the shadow page such that the real data is only modified when the edit is complete.

Affero General Public License (AGPL): The AGPL name encompasses two version of a license, designed to close a perceived application service provider loophole that is found in the ordinary GPL. In a regular GPL, if the modified software is used on network servers and publicly available on the server, the source code does not need to be released. Within an AGPL, the full source code must be made available to any network user of the AGPL-licensed work. The two versions are based on the two corresponding version of the GPL (GPLv2 and GPLv3).

Apache Hadoop: Hadoop is an open-source distributed processing framework that manages data processing and storage for big data applications running in clustered systems. It is at the center of a growing ecosystem of big data technologies that are primarily used to support advanced analytics initiatives, including predictive analytics, data mining, and machine learning applications. It can handle various forms of structured and unstructured data, giving users more flexibility for collecting, processing, and analyzing data, than a typical relational database and data warehouse.

BSD License: BSD (Berkeley Source Distribution) is a one of the simplest and most liberal class of open source software licenses. The only restrictions placed on users of the software released under a BSD license are that it be distributed with the original copyright notice, a disclaimer of liability in

conjunction with the following two simple restrictions: 1) one cannot claim they wrote the software if it is not so, and 2) one should not sue the developer if the software does not function as expected or desired. Some BSD licenses also include an additional restriction on using the name of the project for endorsing derivative works.

Clustering: In the context of database, clustering refers to the ability of several servers or instances to connect to a single database. An instance is the collection of memory and processes that interacts with a database. Clustering offers two major advantages, particularly for high-volume database environments: fault tolerance, because of the availability of multiple servers for users to connect to, and load balancing, allowing users to be automatically allocated to the server with the least workload.

Commons Clause: A commons clause is a condition added to existing open source software licenses to a create a new, slightly more restrictive combined software license. The combined license maintains all conditions of the underlying open source license, but also limits commercial sale of the software by the user. A software that has applied a Commons Clause to an open source code is no longer considered to be open source software. The Commons Clause only applies to the specific software that it is applied to (i.e., a user may develop on top of the licensed software and distribute and sell the larger product).

Data Model: A data model represents how the data is stored within a database. It controls and standardizes how the data is added to and retrieved from the database and how the data relate to one another and to the properties of the real world entities. It can also be a reflection of the application itself.

Data Persistence: In the context of data storage, persistence means that the data survives after the process with which it was created has ended. In other words, the data must write to nonvolatile storage. There are four major design approaches for data storage: 1) pure in-memory, which has no persistence at all (such as memcached, Scalaris); 2) in-memory with periodic snapshots (such as Oracle Coherence, Redis); 3) disk-based memory with update in place writes (such as MySQL ISAM, MongoDB); 4) Commitlog-based memory found in all traditional OTLP databases (such as Oracle, SQL Server). Greater persistence tends to trade-off with in-memory's fast speeds.

Database (DB): A database is collection of data, organized in digital form that can be easily accessed, manipulated, and updated. Database architecture may be external, internal, or conceptual. The external level specifies the way in which every end-user type comprehends the organization of its corresponding relevant data in the database. The internal level deals with the performance, scalability, cost, and other operational matters. The conceptual level perfectly unifies the different external views into a defined and wholly global view. Typically, a database is an OLTP (see below) database, constrained to a single application.

Database Management System (DBMS): A database management system is essentially nothing more than a computerized data-keeping system. In most instances, it functions as a software package designed to define, manipulate, retrieve, and manage data within a database. A DBMS generally manipulates the data itself, the data format, field names, record structure, and file structure as well as defining rules to validate and manipulate this data. It relieves users from framing programs for data maintenance. A DBMS often employs fourth-generation query languages such as SQL along with the DBMS software package to interact with a database. Prominent DMBS software products includes: MySQL, SQL Server, Oracle, PostgreSQL, and IBM Db2.

Database as a Service (DBaaS): DBaaS refers to a new and growing cloud computing service model that enables users to access a database without the need for setting up physical hardware or installing software. The service provider takes charge of maintenance and administration of the physical database products, and leases usage to the user.

Data Warehouse: A data warehouse, contrasted to a database, exists as a layer on top of another database or databases. It takes data from all these databases and creates a layer optimized for and dedicated to performing analytics. A data warehouse can be considered an OLAP database, which accommodates data storage for any number of applications. See OLAP below for greater detail on structure.

DevOps: The term DevOps is commonly used to refer to the roles or processes involving the development and operations teams. It is considered by some a byproduct of Agile development, a design theory aimed at cross-functional development and in computer science specifically, the examination of code over multiple iterations to improve efficiency. It also includes the phenomenon by which skilled professionals automate formerly manual processes—where developers become users of their own software and manual labor related to an infrastructure becomes unnecessary. Cloud computing has led to new possibilities for DevOps in developing IT infrastructure.

Document DB: A document database, also called a document store or document-oriented database, is a subset of NoSQL databases. It is used for storing, retrieving, and managing semi-structured data. Unlike traditional relational databases, the data model in a document database is not structured in a table format of rows and columns. The schema can vary, providing more flexibility for data modelling than in a relational database. The document database uses documents as the structure for storage and queries. In this case, the term "document" often refers to a block of XML or JSON. Instead of columns with names and data types, a document contains a description of the data type and the value for that description. Each document can have the same or a different structure and be added without modification of the schema.

Edge Device: An edge device serves as an entry point into an enterprise or service provider's network. Routers, switches, and integrated access devices (IADs) are all considered edge devices.

Flat Database: A flat database is a simple database system in which each database is represented as a single table. Here, all of the records are stored as single rows of data, which are separated by delimiters such as tabs or commas. The table is usually stored and physically represented as a simple text file.

General Public License (GPL): Also known as GNU GPL, GPLs are widely-used, free software licenses, which guarantee end users the freedom to run, study, share, and modify the software. The GPL is provided through the Free Software Foundation, a nonprofit organization working to provide free software for the GNU Project—an open source OS development program. It is important to note that in this context, 'free' does not refer to price, but rather the freedom to distribute copies of the software. Users can receive the source code and make changes to the software, developing them into new free programs. A company can still charge customers for use of this software, as long as the underlying code is publicly accessible.

Graph DB: A graph database (GDB) is a database that uses graph structures for semantic queries. A graph can be described as composed of two elements, a node and a relationship. The node represents an entity, and each relationship represents how two nodes are associated. Unlike in other DBMS, relationships take first priority in GDBs, and connected data is equally important to individual data points. This connections-first approach to data means connections are persisted through every part of the data lifecycle such that a user's application does not have to infer data connections using alternatives like foreign keys or out-bound processing.

Hierarchal Database: A hierarchal database is a design that uses a one-to-many relationship for data elements. Hierarchal database models use a tree structure that links a number of disparate elements to one "owner" primary record. It is only useful for a certain type of data storage, and therefore, is confined to very specific use-cases.

In-Memory DB: An in-memory database (IMDB), also known as a main memory database system (MMDB), is a database whose data is stored in main memory to facilitate faster response time. Source data is loaded into system memory in a compressed, nonrelational format, which helps streamline the work involved in processing queries by eliminating seek time. IMDBs are faster than disk-optimized databases because disk access is slower than memory access, therefore, the internal optimization algorithms are simpler and execute fewer CPU instructions.

Internet of Things (IoT): The IoT is a computing concept that describes the collection, management, and exploitation of data from millions of edge devices. It describes the idea of physical objects being connected to the internet, thereby connecting various network devices such as vehicles and electronics to each other and allowing them to interact and exchange data.

JSON: Short for JavaScript Object Notation, JSON is a text-based data-interchange format that is used to exchange structured data between a browser and a server. It functions as a text format that is language independent, making it easy for humans to read and machines to parse and generate. JSON is often viewed as an alternative to XML, another plain text data interexchange format. In most cases, the JSON representation of an object is more compact than the XML representation because it does not require tags for each element.

Key-Value DB: A key-value database, also known as a key–value store and key–value store database, is a type of NoSQL database that uses a simple key/value method to store data. Values are identified and accessed via a specific digital key. Stored values can be numbers strings, counters, JSON, HTML, images, etc. It is the most flexible NoSQL model because the application has complete control over what is stored in the value.

MapReduce: MapReduce is a core component of the Apache Hadoop software framework. It filters and parcels out work to various nodes within the cluster (or map), and it organizes and reduces the results from each node into a cohesive answer to a query (functions sometimes termed the mapper and reducer, respectively).

Multimodel DB: The multimodel DB provides a solution to the challenge of heterogeneous data. It supports multiple data models in their organic form within a single, integrated back-end, and uses data and query standards appropriate for each model. Being able to incorporate multiple models into a single database lets users meet various application requirements without the need to deploy different database systems. The data models that such databases can accommodate include relational, hierarchical, and object databases. In contrast to relational data models, multimodel DBs do not uniformly store data in a row-based table structure, and can handle different forms of data such as unstructured and semi-structured data types.

Nonrelational DB: A nonrelational database is any database that does not follow the relational model provided by a traditional RDBMS. These databases, sometimes referred to as NoSQL databases, have grown in popularity due to their inherent scalability and flexibility. For example, they offer flexible schema design for data models, are designed to handle unstructured data, and can be scaled horizontally to take advantage of inexpensive, commodity servers.

NoSQL: A NoSQL database (stands for "non-SQL" or "not-only-SQL") provides a mechanism for storage and retrieval of data that is modeled in means other the tabular relations used in relational databases. NoSQL databases are generally more scalable and provide superior performance by addressing issues that relational databases cannot address, including large volumes of rapidly changing structured or unstructured data, agile sprints, quick schema iterations and frequent code pushes, object-oriented programming, and geographically distributed scale-out architecture. NoSQL database types include document databases, graph stores, key-value stores, and wide-column stores.

OLAP (Online Analytical Processing): OLAP is a computing method that enables users to perform multidimensional analysis of business data. OLAP business intelligence queries often provide capabilities for complex calculations, trend analysis, sales forecasting, financial reporting, budgeting, and data modeling. Data is collected from multiple data sources and stored in data warehouses, which are then cleansed and organized into data cubes. These OLAP cubes contain data categorized by dimensions (like region, time period) derived from dimensional tables in the data warehouse. Data in denormalized to enhance analytical query response times and provide ease of use. OLAP represents one-half of the general IT system structure (contrast to OLTP).

OLTP (Online Transition Processing): OLTP is a class of software programs involved in the operation of a particular system and characterized by many short online transactions. The OLTP system is typically used for order entry, financial transactions, CRM, and retail sales. Database queries for these short tractions are usually simple involving extremely short response times and returning relatively few records. An important attribute of an OLTP system is its ability to maintain concurrency, stemming from often decentralized operations to avoid single points of failure.

Open Source: The term "open source" generally refers to a philosophy within computing that promotes the free access and distribution of an end-product (technological information, design, code) so that it may be improved through multiple insights and iterations. Open-source software is often freely available as a long as the user abides by the software license agreement, most often a General Public License (GNU GPL). GPLs guarantee end-users the freedom to run, study, share, and modify the software, and are most often provided through the Free Software Foundation, a nonprofit company working to provide free software for the GNU Project, an open-source OS development program. Other common open-source licenses include Affero GPLs, which mandate that source code be released if the modified software is used to power web servers, and BSD licenses, which mandate that one cannot claim one wrote the software if false and that one should not sue the developer if the software does not function as desired. A Commons Clause can often be added to an open-source license to limit commercial sale of software using open-source code.

Partitioning: Partitioning is the process by which very large tables are divided into multiple smaller parts within a database. By splitting a larger table, queries that can access only a fraction of the data can run faster because they have to scan less data. Partitioning enhances the performance, manageability, and availability of a wide variety of applications, helping to reduce the total cost of ownership for storing large amounts of data. Partitioning is divided into two categories: vertical and horizontal partitioning. Vertical partitioning splits a table into two or more tables containing different columns and is usually used to increase a SQL server's performance. Horizontal partitioning divides a table into multiple tables that contain the same number of columns but fewer rows and is often used to separate out data based on date-time.

PostgreSQL: PostgreSQL is an open-source object-relational database management system (OR-DBMS) that is managed mostly though a coordinated online effort by an active global community of developers. An ORDBMS is similar to a relational database, except that it has an object-oriented database model, allowing it to support objects, classes, and inheritance in database schemas and query language.

Relational DB: A relational database (RDB) is a collective set of multiple data sets organized by tables, records, and columns. Tables communicate and share information, which facilitates data searchability, organization, and reporting. RDBs use SQL as the standard user application, providing an easy programming interface for database interaction.

Schema: A schema is the visual and logical architecture of a database created on a database management system. It provides a graphical view of the entire database architecture and structure. Schemas provide a means for logically grouping and displaying database objects such as tables, fields, functions, and relations.

Semi-Structured Data: Semi-structured data is data that is neither raw data, nor typed data in a conventional database system. It is structured data, but it is not organized in a rational model, like a table or object-based graph. Files that are semi-structured may contain rational data made up of records but is not organized in a recognizable structure (i.e., some fields may be missing or contain information that cannot be easily described in a database system).

Sharding: Sharding is a method for distributing data across multiple machines. It is a database architecture that partitions data by key ranges and distributes the data among two or more database instances. It enables horizontal scaling of databases.

SQL: SQL (Structured Query Language) is the standard language for dealing with relational databases. SQL programming can be used to insert, search, update, and delete database records. SQL commands are divided into several different types, among them data manipulation language (DML) and data definition language (DDL) statements, transaction controls, and security measures. The DML vocabulary is used to retrieve and manipulate data, while DDL statements are for defining and modifying database structures. The transaction controls help manage transaction processing, ensuring that transactions are either completed or rolled back if errors or problems occur. The security statements are used to control database access as well as to create user roles and permissions.

Streaming Data: Streaming data is data generated continuously by thousands of data sources, which typically send in the data record simultaneously in small bits at a time. This class of data can include a variety data like log files, e-commerce purchases, gaming activity, network from social networks, etc. The data needs to be processed sequentially and incrementally on a record-by-record basis and is mostly used for analytical purposes.

Structured Data: Structured data refers to information with a high degree of organization in a formatted repository, typically via rows and columns. Its elements are made addressable (i.e., easily and uniquely identifiable) for effective processing.

Time Series DB: A time-series database (TSDB) is a database optimized for time-stamped or time series data. Time series data are simply measurements or events that are tracked, monitored, downsampled, and aggregated over time, such as server metrics, application performance monitoring, network data, etc. A TSDB is built specifically for handling metrics and events over time, and is optimized for measuring change over time. TSDBs have unique architectural properties including time-stamp data storage and compression, data lifecycle management, data summarization, ability to handle large time series dependent scans of many records, and time series aware queries.

Unstructured Data: Unstructured data refers to information that does not reside in a traditional row-column database. In contrast to structured data, unstructured data files include text and media content (e.g., e-mail messages, videos, photos, audio files) and represent the majority of the data in any organization.

Wide-Column DB: A wide-column database, also known as a column family database, refers to a category of NoSQL databases that works well for storing enormous amounts of data. Its architecture uses a persistent/sparse matrix and dimensional mapping in a tabular format meant for massive scalability. It benefits from the ability to handle a high volume of data, extreme write speeds with relatively lower velocity reads, and data extraction by columns using row keys.

XML: XML (Extensible Markup Language) is a simple and flexible text format originally designed to meet the challenges of large-scale electronic publishing. XML documents are made up of storage units called entities, which contain parsed or unparsed data. XML was developed under the auspices of the World Wide Web Consortium (W3C) in 1996 to be easily usable over the internet, support a wide variety of applications, and easily create documents. Along with JSON, XML remains one of the most popular formats for storing data in a nonrelational document store database.

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

The prices of the common stock of other public companies mentioned in this report follow:

Adobe Inc.	$259.74
Alibaba Group	$181.28
Alphabet Inc. (Outperform)	$1,226.43
Amazon.com, Inc. (Outperform)	$1,797.27
Cloudera, Inc.	$11.46
IBM Corporation	$139.60
Intel Corp.	$53.82
Microsoft Corp. (Outperform)	$117.52
MongoDB Inc. (Outperform)	$143.73
Oracle Corp (Market Perform)	$52.64
Tencent Holdings, Ltd.	$47.00
salesforce.com, inc. (Outperform)	$163.51
SAP SE (Market Perform)	€100.10

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

IMPORTANT DISCLOSURES

This report is available in electronic form to registered users via R*Docs™ at https://williamblairlibrary.bluematrix.com or www.williamblair.com.

Please contact us at +1 800 621 0687 or consult williamblair.com/Research-and-Insights/Equity-Research/Coverage.aspx for all disclosures.

Jason Ader attests that 1) all of the views expressed in this research report accurately reflect his/her personal views about any and all of the securities and companies covered by this report, and 2) no part of his/her compensation was, is, or will be related, directly or indirectly, to the specific recommendations or views expressed by him/her in this report. We seek to update our research as appropriate. Other than certain periodical industry reports, the majority of reports are published at irregular intervals as deemed appropriate by the research analyst.

DOW JONES: 25962.50
S&P 500: 2854.88
NASDAQ: 7838.96

Additional information is available upon request.

Current Rating Distribution (as of March 21, 2019):

Coverage Universe	Percent	Inv. Banking Relationships *	Percent
Outperform (Buy)	67	Outperform (Buy)	18
Market Perform (Hold)	32	Market Perform (Hold)	8
Underperform (Sell)	1	Underperform (Sell)	17

*Percentage of companies in each rating category that are investment banking clients, defined as companies for which William Blair has received compensation for investment banking services within the past 12 months.

The compensation of the research analyst is based on a variety of factors, including performance of his or her stock recommendations; contributions to all of the firm's departments, including asset management, corporate finance, institutional sales, and retail brokerage; firm profitability; and competitive factors.

OTHER IMPORTANT DISCLOSURES

Stock ratings and valuation methodologies: William Blair & Company, L.L.C. uses a three-point system to rate stocks. Individual ratings reflect the expected performance of the stock relative to the broader market (generally the S&P 500, unless otherwise indicated) over the next 12 months. The assessment of expected performance is a function of near-, intermediate-, and long-term company fundamentals, industry outlook, confidence in earnings estimates, valuation (and our valuation methodology), and other factors. Outperform (O) - stock expected to outperform the broader market over the next 12 months; Market Perform (M) - stock expected to perform approximately in line with the broader market over the next 12 months; Underperform (U) - stock expected to underperform the broader market over the next 12 months; not rated (NR) - the stock is not currently rated. The valuation methodologies include (but are not limited to) price-to-earnings multiple (P/E), relative P/E (compared with the relevant market), P/E-to-growth-rate (PEG) ratio, market capitalization/revenue multiple, enterprise value/EBITDA ratio, discounted cash flow, and others. Stock ratings and valuation methodologies should not be used or relied upon as investment advice. Past performance is not necessarily a guide to future performance.

The ratings and valuation methodologies reflect the opinion of the individual analyst and are subject to change at any time.

Our salespeople, traders, and other professionals may provide oral or written market commentary, short-term trade ideas, or trading strategies-to our clients, prospective clients, and our trading desks-that are contrary to opinions expressed in this research report. Certain outstanding research reports may contain discussions or investment opinions relating to securities, financial instruments and/or issuers that are no longer current. Always refer to the most recent report on a company or issuer. Our asset management and trading desks may make investment decisions that are inconsistent with recommendations or views expressed in this report. We will from time to time have long or short positions in, act as principal in, and buy or sell the securities referred to in this report. Our research is disseminated primarily electronically, and in some instances in printed form. Research is simultaneously available to all clients. This research report is for our clients only. No part of this material may be copied or duplicated in any form by any means or redistributed without the prior written consent of William Blair & Company, L.L.C.

This is not in any sense an offer or solicitation for the purchase or sale of a security or financial instrument. The factual statements herein have been take from sources we believe to be reliable, but such statements are made without any representation as to accuracy or completeness or otherwise, except with respect to any disclosures relative to William Blair or its research analysts. Opinions expressed are our own unless otherwise stated and are subject to change without notice. Prices shown are approximate.

This material is distributed in the United Kingdom and the European Economic Area (EEA) by William Blair International, Ltd., authorised and regulated by the Financial Conduct Authority (FCA). William Blair International, Limited is a limited liability company registered in England and Wales with company number 03619027. This material is only directed and issued to persons regarded as Professional investors or equivalent in their home jurisdiction or persons falling within articles 19 (5), 38, 47, and 49 of the Financial Services and Markets Act of 2000 (Financial Promotion) Order 2005 (all such persons being referred to as "relevant persons"). This document must not be acted on or relied on by persons who are not "relevant persons."

William Blair

Equity Research Directory

John F. O'Toole, Partner   Manager and Director of Research   +1 312 364 8612
Kyle Harris, CFA, Partner   Operations Manager   +1 312 364 8230

FINANCIAL SERVICES AND TECHNOLOGY

Adam Klauber, CFA, Partner   +1 312 364 8232
Co-Group Head–Financial Services and Technology
Financial Analytic Service Providers, Insurance Brokers, Property & Casualty Insurance

Robert Napoli, Partner   +1 312 364 8496
Co-Group Head–Financial Services and Technology
Financial Technology, Specialty Finance

Chris Shutler, CFA   +1 312 364 8197
Asset Management, Financial Technology

GLOBAL SERVICES

Timothy McHugh, CFA, Partner   +1 312 364 8229
Group Head–Global Services
Consulting, HR Technology, Information Services, Staffing

Tim Mulrooney   +1 312 364 8123
Commercial Services

Stephen Sheldon, CFA, CPA   +1 312 364 5167
Real Estate Services and Technology

HEALTHCARE

Biotechnology

Tim Lugo, Partner   +1 415 248 2870
Co-Group Head–Biotechnology
Therapeutics

Y. Katherine Xu, Ph.D., Partner   +1 212 237 2758
Co-Group Head–Biotechnology
Biotechnology

Andy T. Hsieh, Ph.D.   +1 312 364 8169
Biotechnology

Matt Phipps, Ph.D.   +1 312 364 8602
Biotechnology

Raju Prasad, Ph.D.   +1 312 364 8469
Therapeutics

Healthcare Technology and Services

Ryan Daniels, CFA, Partner   +1 312 364 8418
Co-Group Head–Healthcare Technology and Services
Healthcare Technology, Healthcare Services

John Kreger, Partner   +1 312 364 8597
Co-Group Head–Healthcare Technology and Services
Distribution, Outsourcing, Pharmacy Benefit Management

Jeffrey Garro, CFA   +1 312 364 8022
Healthcare Technology

Margaret Kaczor, CFA, Partner   +1 312 364 8608
Medical Technology

Matt Larew   +1 312 364 8242
Healthcare Delivery

Brian Weinstein, CFA, Partner   +1 312 364 8170
Diagnostic Products, Medical Technology

CONSUMER

Sharon Zackfia, CFA, Partner   +1 312 364 5386
Group Head–Consumer
Lifestyle and Leisure Brands, Restaurants

Jon Andersen, CFA, Partner   +1 312 364 8697
Consumer Products

Dylan Carden   +1 312 801 7857
Apparel and Accessories

Daniel Hofkin   +1 312 364 8965
Hardlines, Specialty Retail

Ryan Sundby, CFA   +1 312 364 5443
Outdoor and Recreation

GLOBAL INDUSTRIAL INFRASTRUCTURE

Nick Heymann   +1 212 237 2740
Co-Group Head–Global Industrial Infrastructure
Multi-industry

Larry De Maria, CFA   +1 212 237 2753
Co-Group Head–Global Industrial Infrastructure
Capital Goods

Louie DiPalma, CFA   +1 312 364 5437
Aerospace and Defense

Brian Drab, CFA, Partner   +1 312 364 8280
Industrial Technology

Ryan Merkel, CFA   +1 312 364 8603
Commercial Services, Industrial Distribution

TECHNOLOGY, MEDIA, AND COMMUNICATIONS

Jason Ader, CFA, Partner   +1 212 235 7519
Co-Group Head–Technology, Media, and Communications
Enterprise and Cloud Infrastructure

Bhavan Suri, Partner   +1 312 364 5341
Co-Group Head–Technology, Media, and Communications
IT Services, Software, Software as a Service

Jim Breen, CFA   +1 617 235 7513
Internet Infrastructure and Communication Services

Justin Furby, CFA   +1 312 364 8201
Software as a Service

David Griffin   +1 312 364 8505
Software

Jonathan Ho, Partner   +1 312 364 8276
Cybersecurity, Security Technology

Maggie Nolan, CPA   +1 312 364 5090
IT Services

Matthew Pfau, CFA   +1 312 364 8694
Software as a Service

Ralph Schackart III, CFA, Partner   +1 312 364 8753
Digital Media, Internet

Alessandra Vecchi   +1 212 237 2764
Semiconductors/Wireless

EDITORIAL AND SUPERVISORY ANALYSTS

Steve Goldsmith, Head Editor and SA   +1 312 364 8540
Beth Pekol Porto, Editor and SA   +1 312 364 8924
Kelsey Swanekamp, Editor and SA   +1 312 364 8174
Lisa Zurcher, Editor and SA   +44 20 7868 4549