I'm not a robot

A database is an organized collection of information that can be searched, sorted, and updated. This data is often stored electronically in a computer system called a database management system (DBMS). Databases typically organize data in rows and columns for easy processing and retrieval. Oftentimes, you'll need to use a programming language, such as structured query language (SQL), to interact with your database.Databases are similar to spreadsheets, but there are several key differences. In general, databases are much larger than spreadsheets and so can store more data, and they allow for multiple users to access data at the same time. For these reasons, people who work with data, such as data analysts and data scientists, often work with databases rather than spreadsheets.SQL is a programming language used on almost all relational databases to query and manipulate data. It is still widely used today for querying databases to access the right data needed, but new languages have begun to appear in this space.Common database typesThere are several types of databases, including:Relational database: A relational database stores and allows access to data. These types of databases are considered "relational" because the items within them have pre-determined relationships with one another. Data is stored in tables, which are connected by unique IDs or "keys." To access specific information, users enter the key to access the data that has been programmed to be related to that key.NoSQL database (or nonrelational database): A non-relational database (also known as a NoSQL database) stores data in whatever format is best for the type of data being stored. They tend to contain unstructured data, or data that is less defined, like emails, videos, images, and documents. They are called NoSQL because they don't use SQL.Distributed database: A distributed database stores data in several different physical locations. Processing data in this type of database is spread out. Distributed databases can be homogenous and have the same hardware in each physical location and run the same systems and applications, or they can be heterogeneous and have different operating systems in each location.Object-oriented database: An object-oriented database focuses on organizing objects rather than actions or logic. Instead of being assigned an alphanumeric value, it would remain its original object type.Graph database: A graph database is a type of NoSQL database. It stores, queries, and maps relationships according to the graph theory. Graph databases are used to analyze interactions and connections. They consist of nodes and edges and use a declarative programming language called SPARQL.Cloud database: A cloud database is built in a cloud to optimize for a virtual work setting and distribution. Organizations tend to be charged based on the amount of storage or bandwidth they use.Open-source database: These databases are open-source, meaning anyone can contribute or edit the source code. They can be SQL or NoSQL.Data warehouse: Data warehouses are central repositories for data. A data warehouse is designed to be swift, so users can query and analyze data quickly.Each database type is characterized by specific storage and retrieval practices, data types, job functions, and use cases.Learn more about databases from experts at Google:Components of a databaseThe different types of databases vary in terms of data structure, data types, and schema. Database schema is a blueprint that outlines a database's architecture, describing how the data inside is organized and how different elements (such as foreign and primary keys, fields, etc.) relate to each other.All types of databases have the following five components:Hardware: Database hardware is the physical device that the software runs on, so that users can query and pull data from it. Hardware examples include computers, hard drives, and servers.Software: From the hardware, database software allows users to manipulate the database for their needs. A database management system (DBMS) manages several databases.Data: The data itself is a core component of the database. It is the raw information that is stored on the database and managed by database administrators. Users can then access the data to derive meaning for specific teams and projects.Procedures: Procedures are the rules that determine how the database runs and handles data.Data access language: Data access languages are programming languages (such as SQL) that are used to control and manage the database. They must align with the DBMS and work in sync.Related termsLearn more about data base types and data types and data analysis on CourseraGain hands-on experience with gathering, cleaning, and analyzing data using databases and other analytics tools by pursuing the Google Data Analytics Professional Certificate on Coursera. Develop skills for an in-demand career in data analytics while you learn at the pace that works for you. Jobs Companies Articles Tracker We have collected some of the most popular database management systems (DBMS) nowadays. Let's start by defining what a database management system is. A database management system (DBMS) is a software used to define, manipulate, retrieve, store and manage data in databases. To sum up, database management systems are in charge of: Defining rules to validate and manipulate data. Interacting with databases, applications and end users. Retrieving, storing and analyzing data. Updating data. MySQL is a free, open source relational database management system (RDBMS). It was initially owned by MySQL AB, before being acquired by Sun Microsystems (part of Oracle Corporation since 2010). MySQL was originally developed by Ulf Michael Widenius, Swedes David Axmark and Allan Larsson, founders of MySQL AB. Many database-driven web applications, such as WordPress, Joomla and phpBB, as well as many popular websites like MediaWiki, Twitter and Facebook, use MySQL. Developer: Oracle Corporation. Original author: MySQL AB. Latest MySQL release: MySQL 8.0.32. MySQL license: GNU General Public License version 2 and proprietary. MariaDB is a community-developed, free and open source relational database management system. It is a fork of MySQL. MariaDB was originally developed by Ulf Michael Widenius, Swedes David Axmark and Allan Larsson, founders of MySQL AB and the MariaDB Foundation. Ulf Michael Widenius is the current lead developer and CTO of MariaDB. MariaDB is also included in numerous Linux distributions, such as CentOS, Debian and RHEL. Besides, it is used by many organizations such as Wikipedia, Google or Tumblr. Developer: MariaDB Corporation Ab and MariaDB Foundation. Latest MariaDB release: MariaDB 11.1.0. MariaDB license: GPL version 2. Microsoft SQL Server is a commercial relational database management system. It is available in multiple editions, divided into three main categories: mainstream, specialized and discontinued editions. Developer: Microsoft. Latest Microsoft SQL Server release: Microsoft SQL Server 2022. Microsoft SQL Server license: proprietary license. Oracle DBMS is a commercial, multi-model database management system. It is also known as Oracle Database or just Oracle. It is commonly used for running: online transaction processing (OLTP) and data warehousing (DW). Developer: Oracle Corporation. Latest Oracle DBMS long-term release: Oracle DBMS 19c. Latest Oracle DBMS release: Oracle DBMS 23c beta. Oracle DBMS license: proprietary license. PostgreSQL is a free, open source relational database management system (RDBMS). It was initially developed as a successor of the Ingres database, developed at the University of California, Berkeley. Developer: PostgreSQL Global Development Group. Latest PostgreSQL release: PostgreSQL 15.2. PostgreSQL license: PostgreSQL license. MongoDB is an open source, NoSQL, document-oriented database management system. MongoDB Inc. offers an integrated suite of cloud database services, as well as commercial support. This document-oriented database software is commonly used for high-volume data storage. Developer: MongoDB Inc. Latest MongoDB release: MongoDB 6.0.4. MongoDB license: Server Side Public License (SSPL). Redis, short for "Remote Dictionary Server", is an open source, NoSQL, key-value database management system. Developer: Redis. Original author: Salvatore Sanfilippo. Latest Redis release: Redis 7.0. Redis license: BSD 3-clause. IBM DB2 is a database management product developed by IBM, formerly known as DB2 for Linux, UNIX and Windows. Developer: IBM. Latest IBM DB2 release: IBM DB2 11.5.8. IBM DB2 license: proprietary license. Elasticsearch is a distributed, RESTful search and analytics engine. It is based on the Lucene library. Elasticsearch is the successor to a previous search engine called Compass, also designed by Shay Banon. Developer: Elastic NV. Original author: Shay Banon. Latest Elasticsearch release: Elasticsearch 8.7. Elasticsearch license: dual-licensed Elastic license and Server Side Public License. SQLite is a public domain database engine that belongs to the embedded, relational database management systems family. It has bindings to many programming languages. Developer: Dwayne Richard Hipp. Latest SQLite release: SQLite 3.41.2. SQLite license: Public domain. DBMSTypeOperating systemsLicenseWritten inMySQLRDBMSCanonical, FreeBSD, Linux, MacOS, Solaris and WindowsGNU GPL v2 and proprietaryC and C++MariaDBRDBMSLinux, MacOS and WindowsGNU GPL v2Bash, C, C++, and PerlMicrosoft SQL ServerRDBMSLinux and WindowsProprietaryC and C++Oracle DBMSMulti-model database management systemAIX, BS2000, HP-UX, Linux, MacOS and WindowsProprietaryAssembly language, C and C++PostgreSQLRDBMSFreeBSD, Linux, MacOS, OpenBSD and WindowsPostgreSQL licenseCMongoDBDocument-oriented databaseFreeBSD, Linux, MacOS and WindowsServer Side Public LicenseC++, JavaScript and PythonRedisKey-value databaseUnix-likeBSD 3-clauseCIBM DB2RDBMSLinux, Unix-like and WindowsProprietaryAssembly, C, C++ and JavaElasticsearchSearch and indexLinux, MacOS and WindowsDual-licensed Elastic license and Server Side Public LicenseJavaSQLiteRDBMSAndroid, BSD, iOS, Linux, MacOS, Solaris, VxWorks and WindowsPublic domainC Finally, according to DB-Engines ranking, as of April 2023*, these are the top 10 database management systems: Oracle MySQL Microsoft SQL Server PostgreSQL MongoDB Redis IBM DB2 Elasticsearch SQLite Microsoft Access *DB-Engines ranking is updated on a monthly basis. DBMS > Microsoft SQL ServerPlease select another system to compare it with Microsoft SQL Server.Our visitors often compare Microsoft SQL Server with Oracle, PostgreSQL and MySQL.Share this page Organized collection of data in computing This article is about the computing concept. For instances of the general concept, see Lists of databases. It has been suggested that Data bank be merged into this article. (Discuss) Proposed since May 2025. An SQL select statement and its result In computing, a database is an organized collection of data or a type of data store based on the use of a database management system (DBMS), the software that interacts with end users, applications, and the database itself to capture and analyze the data. The DBMS additionally encompasses the core facilities provided to administer the database. The sum total of the database, the DBMS and the associated applications can be referred to as a database system. Often the term "database" is also used loosely to refer to any of the DBMS, the database system or an application associated with the database. Before digital storage and retrieval of data have become widespread, index cards were used for data storage in a wide range of applications and environments: in the home to record and store recipes, shopping lists, contact information and other organizational data; in business to record presentation notes, project research and notes, and contact information; in schools as flash cards or other visual aids; and in academic research to hold data such as bibliographical citations or notes in a card file. Professional book indexers used index cards in the creation of book indexes until they were replaced by indexing software in the 1980s and 1990s. Small databases can be stored on a file system, while large databases are hosted on computer clusters or cloud storage. The design of databases spans formal techniques and practical considerations, including data modeling, efficient data representation and storage, query languages, security and privacy of sensitive data, and distributed computing issues, including supporting concurrent access and fault tolerance. Computer scientists may classify database management systems according to the database models that they support. Relational databases became dominant in the 1980s. These model data as rows and columns in a series of tables, and the vast majority use SQL for writing and querying data. In the 2000s, non-relational databases became popular, collectively referred to as NoSQL, because they use different query languages. Formally, a "database" refers to a set of related data accessed through the use of a "database management system" (DBMS), which is an integrated set of computer software that allows users to interact with one or more databases and provides access to all of the data contained in the database (although restrictions may exist that limit access to particular data). The DBMS provides various functions that allow entry, storage and retrieval of large quantities of information and provides ways to manage how that information is organized. Because of the close relationship between them, the term "database" is often used casually to refer to both a database and the DBMS used to manipulate it. Outside the world of professional information technology, the term database is often used to refer to any collection of related data (such as a spreadsheet or a card index) as size and usage requirements typically necessitate use of a database management system.[1] Existing DBMSs provide various functions that allow management of a database and its data which can be classified into four main functional groups: Data definition – Creation, modification and removal of definitions that detail how the data is to be organized. Update – Insertion, modification, and deletion of the data itself.[2] Retrieval – Selecting data according to specified criteria (e.g., a query, a position in a hierarchy, or a position in relation to other data) and providing that data either directly to the user, or making it available for further processing by the database itself or by other applications. The retrieved data may be made available in a more or less direct form without modification, as it is stored in the database, or in a new form obtained by altering it or combining it with existing data from the database.[3] Administration – Registering and monitoring users, enforcing data security, monitoring performance, maintaining data integrity, dealing with concurrency control, and recovering information that has been corrupted by some event such as an unexpected system failure.[4] Both a database and its DBMS conform to the principles of a particular database model.[5] "Database system" refers collectively to the database model, database management system, and database.[6] Physically, database servers are dedicated computers that hold the actual databases and run only the DBMS and related software. Database servers are usually multiprocessor computers, with generous memory and RAID disk arrays used for stable storage. Hardware database accelerators, connected to one or more servers via a high-speed channel, are also used in large-volume transaction processing environments. DBMSs are found at the heart of most database applications. DBMSs may be built around a custom multitasking kernel with built-in networking support, but modern DBMSs typically rely on a standard operating system to provide these functions.[citation needed] Since DBMSs comprise a significant market, computer and storage vendors often take into account DBMS requirements in their own development plans.[7] Databases and DBMSs can be categorized according to the database model(s) that they support (such as relational or XML), the type(s) of computer they run on (from a server cluster to a mobile phone), the query language(s) used to access the database (such as SQL or XQuery), and their internal engineering, which affects performance, scalability, resilience, and security. The sizes, capabilities, and performance of databases and their respective DBMSs have grown in orders of magnitude. These performance increases were enabled by the technology progress in the areas of processors, computer memory, computer storage, and computer networks. The concept of a database was made possible by the emergence of direct access storage media such as magnetic disks, which became widely available in the mid-1960s; earlier systems relied on sequential storage of data on magnetic tape. The subsequent development of database technology can be divided into three eras based on data model or structure: navigational,[8] SQL/relational, and post-relational. The two main early navigational data models were the hierarchical model and the CODASYL model (network model). These were characterized by the use of pointers (often physical disk addresses) to follow relationships from one record to another. The relational model, first proposed in 1970 by Edgar F. Codd, departed from this tradition by insisting that applications should search for data by content, rather than by following links. The relational model employs sets of ledger-style tables, each used for a different type of entity. Only in the mid-1980s did computing hardware become powerful enough to allow the wide deployment of relational systems (DBMSs plus applications). By the early 1990s, however, relational systems dominated in all large-scale data processing applications, and as of 2018[update] they remain dominant: IBM Db2, Oracle, MySQL, and Microsoft SQL Server are the most searched DBMS.[9] The dominant database language, standardized SQL for the relational model, has influenced database languages for other data models.[citation needed] Object databases were developed in the 1980s to overcome the inconvenience of object–relational impedance mismatch, which led to the coining of the term "post-relational" and also the development of hybrid object–relational databases. The next generation of post-relational databases in the late 2000s became known as NoSQL databases, introducing fast key–value stores and document-oriented databases. A competing "next generation" known as NewSQL databases attempted new implementations that retained the relational/SQL model while aiming to match the high performance of NoSQL compared to commercially available relational DBMSs. Further information: Navigational database Basic structure of navigational CODASYL database model The introduction of the term database coincided with the availability of direct-access storage (disks and drums) from the mid-1960s onwards. The term represented a contrast with the tape-based systems of the past, allowing shared interactive use rather than daily batch processing. The Oxford English Dictionary cites a 1962 report by the System Development Corporation of California as the first to use the term "data-base" in a specific technical sense.[10] As computers grew in speed and capability, a number of general-purpose database systems emerged; by the mid-1960s a number of such systems had come into commercial use. Interest in a standard began to grow, and Charles Bachman, author of one such product, the Integrated Data Store (IDS), founded the Database Task Group within CODASYL, the group responsible for the creation and standardization of COBOL. In 1971, the Database Task Group delivered their standard, which generally became known as the CODASYL approach, and soon a number of commercial products based on this approach entered the market. The CODASYL approach offered applications the ability to navigate around a linked data set which was formed into a large network. Applications could find records by one of three methods: Use of a primary key (known as a CALC key, typically implemented by hashing) Navigating relationships (called sets) from one record to another Scanning all the records in a sequential order Later systems added B-trees to provide alternate access paths. Many CODASYL databases also added a declarative query language for end users (as distinct from the navigational API). However, CODASYL databases were complex and required significant training and effort to produce useful applications. IBM also had its own DBMS in 1966, known as Information Management System (IMS). IMS was a development of software written for the Apollo program on the System/360. IMS was generally similar in concept to CODASYL, but used a strict hierarchy for its model of data navigation instead of CODASYL's network model. Both concepts later became known as navigational databases due to the way data was accessed: the term was popularized by Bachman's 1973 Turing Award presentation The Programmer as Navigator. IMS is classified by IBM as a hierarchical database. IDMS and Cincom Systems' TOTAL databases are classified as network databases. IMS remains in use as of 2014[update].[11] Edgar F. Codd worked at IBM in San Jose, California, in an office primarily involved in the development of hard disk systems.[12] He was unhappy with the navigational model of the CODASYL approach, notably the lack of a "search" facility. In 1970, he wrote a number of papers that outlined a new approach to database construction that eventually culminated in the groundbreaking A Relational Model of Data for Large Shared Data Banks.[13] The paper described a new system for storing and working with large databases. Instead of records being stored in some sort of linked list of free-form records as in CODASYL, Codd's idea was to organize the data as a number of "tables", each table being used for a different type of entity. Each table would contain a fixed number of columns containing the attributes of the entity. One or more columns of each table were designated as a primary key by which the rows of the table could be uniquely identified; cross-references between tables always used these primary keys, rather than disk addresses, and queries would join tables based on these key relationships, using a set of operations based on the mathematical system of relational calculus (from which the model takes its name). Splitting the data into a set of normalized tables (or relations) aimed to ensure that each "fact" was only stored once, thus simplifying update operations. Virtual tables called views could present the data in different ways for different users, but views could not be directly updated. Codd used mathematical terms to define the model: relations, tuples, and domains rather than tables, rows, and columns. The terminology that is now familiar came from early implementations. Codd would later criticize the tendency for practical implementations to depart from the mathematical foundations on which the model was based. In the relational model, records are "linked" using virtual keys not stored in the database but defined as needed between the data contained in the records. The use of primary keys (user-oriented identifiers) to represent cross-table relationships, rather than disk addresses, had two primary motivations. From an engineering perspective, it enabled tables to be relocated and resized without expensive database reorganization. But Codd was more interested in the difference in semantics: the use of explicit identifiers made it easier to define update operations with clean mathematical definitions, and it also enabled query operations to be defined in terms of the established discipline of first-order predicate calculus; because these operations have clean mathematical properties, it becomes possible to rewrite queries in provably correct ways, which is the basis of query optimization. There is no loss of expressiveness compared with the hierarchic or network models, though the connections between tables are no longer so explicit. In the hierarchic and network models, records were allowed to have a complex internal structure. For example, the salary history of an employee might be represented as a "repeating group" within the employee record. In the relational model, the process of normalization led to such internal structures being replaced by data held in multiple tables, connected only by logical keys. For instance, a common use of a database system is to track information about users, their name, login information, various addresses and phone numbers. In the navigational approach, all of this data would be placed in a single variable-length record. In the relational approach, the data would be normalized into a user table, an address table and a phone number table (for instance). Records would be created in these optional tables only if the address or phone numbers were actually provided. As well as identifying rows/records using logical identifiers rather than disk addresses, Codd changed the way in which applications assembled data from multiple records. Rather than requiring applications to gather data one record at a time by navigating the links, they would use a declarative query language that expressed what data was required, rather than the access path by which it should be found. Finding an efficient access path to the data became the responsibility of the database management system, rather than the application programmer. This process, called query optimization, depended on the fact that queries were expressed in terms of mathematical logic. Codd's paper inspired teams at various universities to research the subject, including one at University of California, Berkeley[12] led by Eugene Wong and Michael Stonebraker, who started INGRES using funding that had already been allocated for a geographical database project and student programmers to produce code. Beginning in 1973, INGRES delivered its first test products which were generally ready for widespread use in 1979. INGRES was similar to System R in a number of ways, including the use of a "language" for data access, known as QUEL. Over time, INGRES moved to the emerging SQL standard. IBM itself did one test implementation of the relational model, PRTV, and a production one, Business System 12, both now discontinued. Honeywell wrote MRDS for Multics, and now there are two new implementations: Alphora Dataphor and Rel. Most other DBMS implementations usually called relational are actually SQL DBMSs. In 1970, the University of Michigan began development of the MICRO Information Management System[14] based on D.L. Childs' Set-Theoretic Data model.[15][16][17] The university in 1974 hosted a debate between Codd and Bachman which Bruce Lindsay of IBM later described as "throwing lightning bolts at each other!".[12] MICRO was used to manage very large data sets by the US Department of Labor, the U.S. Environmental Protection Agency, and researchers from the University of Alberta, the University of Michigan, and Wayne State University. It ran on IBM mainframe computers using the Michigan Terminal System.[18] The system remained in production until 1998. Main article: Database machine In the 1970s and 1980s, attempts were made to build database systems with integrated hardware and software. The underlying philosophy was that such integration would provide higher performance at a lower cost. Examples were IBM System/38, the early offering of Teradata, and the Britton Lee, Inc. database machine. Another approach to hardware support for database management was ICL's CAFS accelerator, a hardware disk controller with programmable search capabilities. In the long term, these efforts were generally unsuccessful because specialized database machines could not keep pace with the rapid development and progress of general-purpose computers. Thus most database systems nowadays are software systems running on general-purpose hardware, using general-purpose computer data storage. However, this idea is still pursued in certain applications by some companies like Netezza and Oracle (Exadata). IBM formed a team led by Codd that started working on a prototype system, System R despite opposition from others at the company.[12] The first version was ready in 1974/5, and work then started on multi-table systems in which the data could be split so that all of the data for a record (some of which is optional) did not have to be stored in a single large "chunk". Subsequent multi-user versions were tested by customers in 1978 and 1979, by which time a standardized query language – SQL[citation needed] – had been added. Codd's ideas were establishing themselves as both workable and superior to CODASYL, pushing IBM to develop a true production version of System R, known as SQL/DS, and, later, Database 2 (IBM Db2). Larry Ellison's Oracle Database (or more simply, Oracle) started from a different chain, based on IBM's papers on System R. Though Oracle V1 implementations were completed in 1978, it was not until Oracle Version 2 when Ellison beat IBM to market in 1979.[19] Stonebraker went on to apply the lessons from INGRES to develop a new database, Postgres, which is now known as PostgreSQL. PostgreSQL is often used for global mission-critical applications (the .org and .info domain name registries use it as their primary data store, as do many large companies and financial institutions). In Sweden, Codd's paper was also read and Mimer SQL was developed in the mid-1970s at Uppsala University. In 1984, this project was consolidated into an independent enterprise. Another data model, the entity–relationship model, emerged in 1976 and gained popularity for database design as it emphasized a more familiar description than the earlier relational model. Later on, entity–relationship constructs were retrofitted as a data modeling construct for the relational model, and the difference between the two has become irrelevant.[citation needed] Besides IBM and various software companies such as Sybase and Informix Corporation, most large computer hardware vendors by the 1980s had their own database systems such as DEC's VAX Rdb/VMS.[20] The decade ushered in the age of desktop computing. The new computers empowered their users with spreadsheets like Lotus 1-2-3 and database software like dBASE. The dBASE product was lightweight and easy for any computer user to understand out of the box. C. Wayne Ratliff, the creator of dBASE, stated: "dBASE was different from programs like BASIC, C, FORTRAN, and COBOL in that a lot of the dirty work had already been done. The data manipulation is done by dBASE instead of by the user, so the user can concentrate on what he is doing, rather than having to mess with the dirty details of opening, reading, and closing files, and managing space allocation."[21] dBASE was one of the top selling software titles in the 1980s and early 1990s. By the start of the decade databases had become a billion-dollar industry in about ten years.[20] The 1990s, along with a rise in object-oriented programming, saw a growth in how data in various databases were handled. Programmers and designers began to treat the data in their databases as objects. That is to say that if a person's data were in a database, that person's attributes, such as their address, phone number, and age, were now considered to belong to that person instead of being extraneous data. This allows for relations between data to be related to objects and their attributes and not to individual fields.[22] The term "object–relational impedance mismatch" described the inconvenience of translating between programmed objects and database tables. Object databases and object–relational databases attempt to solve this problem by providing an object-oriented language (sometimes as extensions to SQL) that programmers can use as alternative to purely relational SQL. On the programming side, libraries known as object–relational mappings (ORMs) attempt to solve the same problem. Main articles: NoSQL and NewSQL Database sales grew rapidly during the dotcom bubble and, after its end, the rise of ecommerce. The popularity of open source databases such as MySQL has grown since 2000, to the extent that Ken Jacobs of Oracle said in 2005 that perhaps "these guys are doing to us what we did to IBM".[20] XML databases are a type of structured document-oriented database that allows querying based on XML document attributes. XML databases are mostly used in applications where the data is conveniently viewed as a collection of documents, with a structure that can vary from the very flexible to the highly rigid: examples include scientific articles, patents, tax filings, and personnel records. NoSQL databases are often very fast,[23][24] do not require fixed table schemas, avoid join operations by storing denormalized data, and are designed to scale horizontally. In recent years, there has been a strong demand for massively distributed databases with high partition tolerance, but according to the CAP theorem, it is impossible for a distributed system to simultaneously provide consistency, availability, and partition tolerance guarantees. A distributed system can satisfy any two of these guarantees at the same time, but not all three. For that reason, many NoSQL databases are using what is called eventual consistency to provide both availability and partition tolerance guarantees with a reduced level of data consistency. NewSQL is a class of modern relational databases that aims to provide the same scalable performance of NoSQL systems for online transaction processing (read-write) workloads while still using SQL and maintaining the ACID guarantees of a traditional database system. This section does not cite any sources. Please help improve this section by adding citations to reliable sources. Unsourced material may be challenged and removed. (March 2013) (Learn how and when to remove this message) Databases are used to support internal operations of organizations and to underpin online interactions with customers and suppliers (see Enterprise software). Databases are used to hold administrative information and more specialized data, such as engineering data or economic models. Examples include computerized library systems, flight reservation systems, computerized parts inventory systems, and many content management systems that store websites as collections of webpages in a database. One way to classify databases involves the type of their contents, for example: bibliographic, document-text, statistical, or multimedia objects. Another way is by their application area, for example: accounting, music compositions, movies, banking, manufacturing, or insurance. A third way is by some technical aspect, such as the database structure or interface type. This section lists a few of the adjectives used to characterize different kinds of databases. An in-memory database is a database that primarily resides in main memory, but is typically backed-up by non-volatile computer data storage. Main memory databases are faster than disk databases, and so are often used where response time is critical, such as in telecommunications network equipment. An active database includes an event-driven architecture which can respond to conditions both inside and outside the database. Possible uses include security monitoring, alerting, statistics gathering and authorization. Many databases provide active database features in the form of database triggers. A cloud database relies on cloud technology. Both the database and most of its DBMS reside remotely, "in the cloud", while its applications are both developed by programmers and later maintained and used by end-users through a web browser and Open APIs. Data warehouses[citation needed] archive data from operational databases and often from external sources such as market research firms. The warehouse becomes the central source of data for use by managers and other end-users who may not have access to operational data. For example, sales data might be aggregated to weekly totals and converted from internal product codes to use UPCs so that they can be compared with ACNielsen data. Some basic and essential components of data warehousing include extracting, analyzing, and mining data, transforming, loading, and managing data so as to make them available for further use. A deductive database combines logic programming with a relational database. A distributed database is one in which both the data and the DBMS span multiple computers. A document-oriented database is designed for storing, retrieving, and managing document-oriented, or semi structured, information. Document-oriented databases are one of the main categories of NoSQL databases. An embedded database system is a DBMS which is tightly integrated with an application software that requires access to stored data in such a way that the DBMS is hidden from the application's end-users and requires little or no ongoing maintenance.[25] End-user databases consist of data developed by individual end-users. Examples of these are collections of documents, spreadsheets, presentations, multimedia, and other files. Several products[which?] exist to support such databases. A federated database system comprises several distinct databases, each with its own DBMS. It is handled as a single database by a federated database management system (FDBMS), which transparently integrates multiple autonomous DBMSs, possibly of different types (in which case it would also be a heterogeneous database system), and provides them with an integrated conceptual view. Sometimes the term multi-database is used as a synonym for federated database, though it may refer to a less integrated (e.g., without an FDBMS and a managed integrated schema) group of databases that cooperate in a single application. In this case, typically middleware is used for distribution, which typically includes an atomic commit protocol (ACP), e.g., the two-phase commit protocol, to allow distributed (global) transactions across the participating databases. A graph database is a kind of NoSQL database that uses graph structures with nodes, edges, and properties to represent and store information. General graph databases that can store any graph are distinct from specialized graph databases such as triplestores and network databases. An array DBMS is a kind of NoSQL DBMS that allows modeling, storage, and retrieval of (usually large) multi-dimensional arrays such as satellite images and climate simulation output. In a hypertext or hypermedia database, any word or a piece of text representing an object, e.g., another piece of text, an article, a picture, or a film, can be hyperlinked to that object. Hypertext databases are particularly useful for organizing large amounts of disparate information. For example, they are useful for organizing online encyclopedias, where users can conveniently jump around the text. The World Wide Web is thus a large distributed hypertext database. A knowledge base (abbreviated KB, kb or Δ[26][27]) is a special kind of database for knowledge management, providing the means for the computerized collection, organization, and retrieval of knowledge. Also a collection of data representing problems with their solutions and related experiences. A mobile database can be carried on or synchronized from a mobile computing device. Operational databases store detailed data about the operations of an organization. They typically process relatively high volumes of updates using transactions. Examples include customer databases that record contact, credit, and demographic information about a business's customers, personnel databases that hold information such as salary, benefits, skills data about employees, enterprise resource planning systems that record details about product components, parts inventory, and financial databases that keep track of the organization's money, accounting and financial dealings. A parallel database seeks to improve performance through parallelization for tasks such as loading data, building indexes and evaluating queries. The major parallel DBMS architectures which are induced by the underlying hardware architecture are: Shared memory architecture, where multiple processors share the main memory space, as well as other data storage. Shared disk architecture, where each processing unit (typically consisting of multiple processors) has its own main memory, but all units share the other storage. Shared-nothing architecture, where each processing unit has its own main memory and other storage. Probabilistic databases employ fuzzy logic to draw inferences from imprecise data. Real-time databases process transactions fast enough for the result to come back and be acted on right away. A spatial database can store the data with multidimensional features. The queries on such data include location-based queries, like "Where is the closest hotel in my area?". A temporal database has built-in time aspects, for example a temporal data model and a temporal version of SQL. More specifically the temporal aspects usually include valid-time and transaction-time. A terminology-oriented database builds upon an object-oriented database, often customized for a specific field. An unstructured data database is intended to store in a manageable and protected way diverse objects that do not fit naturally and conveniently in common databases. It may include email messages, documents, journals, multimedia objects, etc. The name may be misleading since some objects can be highly structured. However, the entire possible object collection does not fit into a predefined structured framework. Most established DBMSs now support unstructured data in various ways, and new dedicated DBMSs are emerging. Connolly and Begg define database management system (DBMS) as a "software system that enables users to define, create, maintain and control access to the database."[28] Examples of DBMS's include MySQL, MariaDB, PostgreSQL, Microsoft SQL Server, Oracle Database, and Microsoft Access. The DBMS acronym is sometimes extended to indicate the underlying database model, with RDBMS for the relational, OODBMS for the object (oriented) and ORDBMS for the object–relational model. Other extensions can indicate some other characteristics, such as DDBMS for a distributed database management systems. The functionality provided by a DBMS can vary enormously. The core functionality is the storage, retrieval and update of data. Codd proposed the following functions and services a fully-fledged general purpose DBMS should provide:[29] Data storage, retrieval and update User accessible catalog or data dictionary describing the metadata Support for transactions and concurrency Facilities for recovering the database should it become damaged Support for authorization of access and update of data Access support from remote locations Enforcing constraints to ensure data in the database abides by certain rules It is also generally to be expected the DBMS will provide a set of utilities for such purposes as may be necessary to administer the database effectively, including import, export, monitoring, defragmentation and analysis utilities.[30] The core part of the DBMS interacting between the database and the application interface sometimes referred to as the database engine. Often DBMSs will have configuration parameters that can be statically and dynamically tuned, for example the maximum amount of main memory on a server the database can use. The trend is to minimize the amount of manual configuration, and for cases such as embedded databases the need to target zero-administration is paramount. The large major enterprise DBMSs have tended to increase in size and functionality and have involved up to thousands of human years of development effort throughout their lifetime.[a] Early multi-user DBMS typically only allowed for the application to reside on the same computer with access via terminals or terminal emulation software. The client–server architecture was a development where the application resided on a client desktop and the database on a server allowing the processing to be distributed. This evolved into a multitier architecture incorporating application servers and web servers with the end user interface via a web browser with the database only directly connected to the adjacent tier.[32] A general-purpose DBMS will provide public application programming interfaces (API) and optionally a processor for database languages such as SQL to allow applications to be written to interact with and manipulate the database. A special purpose DBMS may use a private API and be specifically customized and linked to a single application. For example, an email system performs many of the functions of a general-purpose DBMS such as message insertion, message deletion, attachment handling, blocklist lookup, associating messages an email address and so forth however these functions are limited to what is required to handle email. Main article: Database application External interaction with the database will be via an application program that interfaces with the DBMS.[33] This can range from a database tool that allows users to execute SQL queries textually or graphically, to a website that happens to use a database to store and search information. A programmer will code interactions to the database (sometimes referred to as a datasource) via an application program interface (API) or via a database language. The particular API or language chosen will need to be supported by DBMS, possibly indirectly via a preprocessor or a bridging API. Some API's aim to be database independent, ODBC being a commonly known example. Other common API's include JDBC and ADO.NET. Database languages are special-purpose languages, which allow one or more of the following tasks, sometimes distinguished as sublanguages: Data control language (DCL) – controls access to data; Data definition language (DDL) – defines data types such as creating, altering, or dropping tables and the relationships among them; Data manipulation language (DML) – performs tasks such as inserting, updating, or deleting data occurrences; Data query language (DQL) – allows searching for information and computing derived information. Database languages are specific to a particular data model. Notable examples include: SQL combines the roles of data definition, data manipulation, and query in a single language. It was one of the first commercial languages for the relational model, although it departs in some respects from the relational model as described by Codd (for example, the rows and columns of a table can be ordered). SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987. The standards have been regularly enhanced since and are supported (with varying degrees of conformance) by all mainstream commercial relational DBMSs.[34][35] OQL is an object model language standard (from the Object Data Management Group). It has influenced the design of some of the newer query languages like JDOQL and EJB QL. XQuery is a standard XML query language implemented by XML database systems such as MarkLogic and eXist, by relational databases with XML capability such as Oracle and Db2, and also by in-memory XML processors such as Saxon. SQL/XML combines XQuery with SQL.[36] A database language may also incorporate features like: DBMS-specific configuration and storage engine management Computations to modify query results, like counting, summing, averaging, sorting, grouping, and cross-referencing Constraint enforcement (e.g. in an automotive database, only allowing one engine type per car) Application programming interface version of the query language, for programmer convenience Main articles: Computer data storage and Database engine Database storage is the container of the physical materialization of a database. It comprises the internal (physical) level in the database architecture. It also contains all the information needed (e.g., metadata, "data about the data", and internal data structures) to reconstruct the conceptual level and external level from the internal level when needed. Databases as digital objects contain three layers of information which must be stored: the data, the structure, and the semantics. Proper storage of all three layers is needed for future preservation and longevity of the database.[37] Putting data into permanent storage is generally the responsibility of the database engine a.k.a. "storage engine". Though typically accessed by a DBMS through the underlying operating system (and often using the operating systems' file systems as intermediates for storage layout), storage properties and configuration settings are extremely important for the efficient operation of the DBMS, and thus are closely maintained by database administrators. A DBMS, while in operation, always has its database residing in several types of storage (e.g., memory and external storage). The database data and the additional needed information, possibly in very large amounts, are coded into bits. Data typically reside in the storage in structures that look completely different from the way the data look at the conceptual and external levels, but in ways that attempt to optimize (the best possible) these levels' reconstruction when needed by users and programs, as well as for computing additional types of needed information from the data (e.g., when querying the database). Some DBMSs support specifying which character encoding was used to store data, so multiple encodings can be used in the same database. Various low-level database storage structures are used by the storage engine to serialize the data model so it can be written to the medium of choice. Techniques such as indexing may be used to improve performance. Conventional storage is row-oriented, but there are also column-oriented and correlation databases. Main article: Materialized view Often storage redundancy is employed to increase performance. A common example is storing materialized views, which consist of frequently needed external views or query results. Storing such views saves the expensive computing them each time they are needed. The downsides of materialized views are the overhead incurred when updating them to keep them synchronized with their original updated database data, and the cost of storage redundancy. See also: Replication (computing) § Database replication Occasionally a database employs storage redundancy by database objects replication (with one or more copies) to increase data availability (both to improve performance of simultaneous multiple end-user accesses to the same database object, and to provide resiliency in a case of partial failure of a distributed database). Updates of a replicated object need to be synchronized across the object copies. In many cases, the entire database is replicated. With data virtualization, the data used remains in its original locations and real-time access is established to allow analytics across multiple sources. This can aid in resolving some technical difficulties such as compatibility problems when combining data from various platforms, lowering the risk of error caused by faulty data, and guaranteeing that the newest data is used. Furthermore, avoiding the creation of a new database containing personal information can make it easier to comply with privacy regulations. However, with data virtualization, the connection to all necessary data sources must be operational as there is no local copy of the data, which is one of the main drawbacks of the approach.[38] This article appears to contradict the article Database security. Please discuss at the talk page and do not remove this message until the contradictions are resolved. (March 2013) Main article: Database security Database security deals with all various aspects of protecting the database content, its owners, and its users. It ranges from protection from intentional unauthorized database uses to unintentional database accesses by unauthorized entities (e.g., a person or a computer program). Database access control deals with controlling who (a person or a certain computer program) are allowed to access what information in the database. The information may comprise specific database objects (e.g., record types, specific records, data structures), certain computations over certain objects (e.g., query types, or specific queries), or using specific access paths to the former (e.g., using specific indexes or other data structures to access information). Database access controls are set by special authorized (by the database owner) personnel that uses dedicated protected security DBMS interfaces. This may be managed directly on an individual basis, or by the assignment of individuals and privileges to groups, or (in the most elaborate models) through the assignment of individuals and groups to roles which are then granted entitlements. Data security prevents unauthorized users from viewing or updating the database. Using passwords, users are allowed access to the entire database or subsets of it called "subschemas". For example, an employee database can contain all the data about an individual employee, but one group of users may be authorized to view only payroll data, while others are allowed access to only work history and medical data. If the DBMS provides a way to interactively enter and update the database, as well as interrogate it, this capability allows for managing personal databases. Data security in general deals with protecting specific chunks of data, both physically (i.e., from corruption, or destruction, or removal; e.g., see physical security), or the interpretation of them, or parts of them to meaningful information (e.g., by looking at the strings of bits that they comprise, concluding specific valid credit-card numbers; e.g., see data encryption). Change and access logging records who accessed which attributes, what was changed, and when it was changed. Logging services allow for a forensic database audit later by keeping a record of access occurrences and changes. Sometimes application-level code is used to record changes rather than leaving this in the database. Monitoring can be set up to attempt to detect security breaches. Therefore, organizations must take database security seriously because of the many benefits it provides. Organizations will be safeguarded from security breaches and hacking activities like firewall intrusion, virus spread, and ransom ware. This helps in protecting the company's essential information, which cannot be shared with outsiders at any cause.[39] Further information: Concurrency control Database transactions can be used to introduce some level of fault tolerance and data integrity after recovery from a crash. A database transaction is a unit of work, typically encapsulating a number of operations over a database (e.g., reading a database object, writing, acquiring or releasing a lock, etc.), an abstraction supported in database and also other systems. Each transaction has well defined boundaries in terms of which program/code executions are included in that transaction (determined by the transaction's programmer via special transaction commands). The acronym ACID describes some ideal properties of a database transaction: atomicity, consistency, isolation, and durability. See also: Data migration § Database migration A database built with one DBMS is not portable to another DBMS (i.e., the other DBMS cannot run it). However, in some situations, it is desirable to migrate a database from one DBMS to another. The reasons are primarily economical (different DBMSs may have different total costs of ownership or TCOs), functional, and operational (different DBMSs may have different capabilities). The migration involves the database's transformation from one DBMS type to another. The transformation should maintain (if possible) the database related application (i.e., all related application programs) intact. Thus, the database's conceptual and external architectural levels should be maintained in the transformation. It may be desired that also some aspects of the architecture internal level are maintained. A complex or large database migration may be a complicated and costly (one-time) project by itself, which should be factored into the decision to migrate. This is in spite of the fact that tools may exist to help migration between specific DBMSs. Typically, a DBMS vendor provides tools to help import databases from other popular DBMSs. Main article: Database tuning After designing a database for an application, the next stage is building the database. Typically, an appropriate general-purpose DBMS can be selected to be used for this purpose. A DBMS provides the needed user interfaces to be used by database administrators to define the needed application's data structures within the DBMS's respective data model. Other user interfaces are used to select needed DBMS parameters (like security related, storage allocation parameters, etc.). When the database is ready (all its data structures and other needed components are defined), it is typically populated with initial application's data (database initialization, which is typically a distinct project; in many cases using specialized DBMS interfaces that support bulk insertion) before making it operational. In some cases, the database becomes operational while empty of application data, and data are accumulated during its operation. After the database is created, initialized and populated it needs to be maintained. Various database parameters may need changing and the database may need to be tuned (tuning) for better performance; application's data structures may be changed or added, new related application programs may be written to add to the application's functionality, etc. Main article: Backup Sometimes it is desired to bring a database back to a previous state (for many reasons, e.g., cases when the database is found corrupted due to a software error, or if it has been updated with erroneous data). To achieve this, a backup operation is done occasionally or continuously, where each desired database state (i.e., the values of its data and their embedding in database's data structures) is kept within dedicated backup files (many techniques exist to do this effectively). When it is decided by a database administrator to bring the database back to this state (e.g., by specifying this state by a desired point in time when the database was in this state), these files are used to restore that state. Static analysis techniques for software verification can be applied also in the scenario of query languages. In particular, the *Abstract interpretation framework has been extended to the field of query languages for relational databases as a way to support sound approximation techniques.[40] The semantics of query languages can be tuned according to suitable abstractions of the concrete domain of data. The abstraction of relational database systems has many interesting applications, in particular, for security purposes, such as fine-grained access control, watermarking, etc. Other DBMS features might include: Database logs – This helps in keeping a history of the executed functions. Graphics component for producing graphs and charts, especially in a data warehouse system. Query optimizer – Performs query optimization on every query to choose an efficient query plan (a partial order (tree) of operations) to be executed to compute the query result. May be specific to a particular storage engine. Tools or hooks for database design, application programming, application program maintenance, database performance analysis and monitoring, database configuration monitoring, DBMS hardware configuration (a DBMS and related database may span computers, networks, and storage units) and related database mapping (especially for a distributed DBMS), storage allocation and database layout monitoring, storage migration, etc. Increasingly, there are calls for a single system that incorporates all of these core functionalities into the same build, test, and deployment framework for database management and source control. Borrowing from other developments in the software industry, some market such offerings as "DevOps for database".[41] Main article: Database design The first task of a database designer is to produce a conceptual data model that reflects the structure of the information to be held in the database. A common approach to this is to develop an entity–relationship model, often with the aid of drawing tools. Another popular approach is the Unified Modeling Language. A successful data model will accurately reflect the possible state of the external world being modeled: for example, if people can have more than one phone number, it will allow this information to be captured. Designing a good conceptual data model requires a good understanding of the application domain; it typically involves asking deep questions about the things of interest to an organization, like "can a customer also be a supplier?", or "if a product is sold with two different forms of packaging, are those the same product or different products?", or "if a plane flies from New York to Dubai via Frankfurt, is that one flight or two (or maybe even three)?". The answers to these questions establish definitions of the terminology used for entities (customers, products, flights, flight segments) and their relationships and attributes. Producing the conceptual data model sometimes involves input from business processes, or the analysis of workflow in the organization. This can help to establish what information is needed in the database, and what can be left out. For example, it can help when deciding whether the database needs to hold historic data as well as current data. Having produced a conceptual data model that users are happy with, the next stage is to translate this into a schema that implements the relevant data structures within the database. This process is often called logical database design, and the output is a logical data model expressed in the form of a schema. Whereas the conceptual data model is (in theory at least) independent of the choice of database technology, the logical data model will be expressed in terms of a particular database model supported by the chosen DBMS. (The terms data model and database model are often used interchangeably, but in this article we use data model for the design of a specific database, and database model for the modeling notation used to express that design). The most popular database model for general-purpose databases is the relational model, or more precisely, the relational model as represented by the SQL language. The process of creating a logical database design using this model uses a methodical approach known as normalization. The goal of normalization is to ensure that each elementary "fact" is only recorded in one place, so that insertions, updates, and deletions automatically maintain consistency. The final stage of database design is to make the decisions that affect performance, scalability, recovery, security, and the like, which depend on the particular DBMS. This is often called physical database design, and the output is the physical data model. A key goal during this stage is data independence, meaning that the decisions made for performance optimization purposes should be invisible to end-users and applications. There are two types of data independence: Physical data independence and logical data independence. Physical design is driven mainly by performance requirements, and requires a good knowledge of the expected workload and access patterns, and a deep understanding of the features offered by the chosen DBMS. Another aspect of physical database design is security. It involves both defining access control to database objects as well as defining security levels and methods for the data itself. Main article: Database model Collage of five types of database models A database model is a type of data model that determines the logical structure of a database and fundamentally determines in which manner data can be stored, organized, and manipulated. The most popular example of a database model is the relational model (or the SQL approximation of relational), which uses a table-based format. Common logical data models for databases include: Navigational databases Hierarchical database model Network model Graph database Relational model Entity–relationship model Enhanced entity–relationship model Object model Document model Entity–attribute–value model Star schema An object–relational database combines the two related structures. Physical data models include: Inverted index Flat file Other models include: Multidimensional model Array model Multivalue model Specialized models are optimized for particular types of data: XML database Semantic model Content store Event store Time series model Traditional view of data[42] A database management system provides three views of the database data: The external level defines how each group of end-users sees the organization of data in the database. A single database can have any number of views at the external level. The conceptual level (or logical level) unifies the various external views into a compatible global view.[43] It provides the synthesis of all the external views. It is out of the scope of the various database end-users, and is rather of interest to database application developers and database administrators. The internal level (or physical level) is the internal organization of data inside a DBMS. It is concerned with cost, performance, scalability and other operational matters. It deals with storage layout of the data, using storage structures such as indexes to enhance performance. Occasionally it stores data of individual views (materialized views), computed from generic data, if performance justification exists for such redundancy. It balances all the external views' performance requirements, possibly conflicting, in an attempt to optimize overall performance across all activities. While there is typically only one conceptual and internal view of the data, there can be any number of different external views. This allows users to see database information in a more business-related way rather than from a technical, processing viewpoint. For example, a financial department of a company needs the payment details of all employees as part of the company's expenses, but does not need details about employees that are in the interest of the human resources department. Thus different departments need different views of the company's database. The three-level database architecture relates to the concept of data independence which was one of the major initial driving forces of the relational model.[43] The idea is that changes made at a certain level do not affect the view at a higher level. For example, changes in the internal level do not affect application programs written using conceptual level interfaces, which reduces the impact of making physical changes to improve performance. The conceptual view provides a level of indirection between internal and external. On the one hand it provides a common view of the database, independent of different external view structures, and on the other hand it abstracts away details of how the data are stored or managed (internal level). In principle every level, and even every external view, can be presented by a different data model. In practice usually a given DBMS uses the same data model for both the external and the conceptual levels (e.g., relational model). The internal level, which is hidden inside the DBMS and depends on its implementation, requires a different level of detail and uses its own types of data structure types. Database technology has been an active research topic since the 1960s, both in academia and in the research and development groups of companies (for example IBM Research). Research activity includes theory and development of prototypes. Notable research topics have included models, the atomic transaction concept, related concurrency control techniques, query languages and query optimization methods, RAID, and more. The database research area has several dedicated academic journals (for example, ACM Transactions on Database Systems-TODS, Data and Knowledge Engineering-DKE) and annual conferences (e.g., ACM SIGMOD, ACM PODS, VLDB, IEEE ICDE). For a topical guide, see Outline of databases. Comparison of database tools Comparison of object database management systems Comparison of object–relational database management systems Comparison of relational database management systems Data hierarchy Data bank Data store Database theory Database testing Database-centric architecture Datalog Database-as-IPC DBOS Flat-file database INP (database) Journal of Database Management Question-focused dataset ^ This article quotes a development time of five years involving 750 people for DB2 release 9 alone.[31] ^ Ullman & Widom 1997, p. 1. ^ "Update Definition & Meaning". Merriam-Webster. Archived from the original on Feb 25, 2024. ^ "Retrieval Definition & Meaning". Merriam-Webster. Archived from the original on Jun 27, 2023. ^ "Administration Definition & Meaning". Merriam-Webster. Archived from the original on Dec 6, 2023. ^ Tsitchizris & Lochovsky 1982. ^ Beynon-Davies 2003. ^ Nelson & Nelson 2001. ^ Bachman 1973. ^ "TOPDB Top Database index". pypl.github.io. ^ "database, n". OED Online. Oxford University Press. June 2013. Retrieved July 12, 2013. (Subscription required.) ^ IBM Corporation (October 2013). "IBM Information Management System (IMS) 13 Transaction and Database Servers delivers high performance and low total cost of ownership". Retrieved Feb 20, 2014. ^ a b c d "RDBMS Plenary 1: Early Years" (PDF) (Interview). Interviewed by Burton Grad. Computer History Museum. 2007-06-12. Retrieved 2025-05-30. ^ Codd 1970. ^ Hershey & Easthope 1972. ^ North 2010. ^ Childs 1968a. ^ Childs 1968b. ^ M.A. Kahn; D.L. Rumelhart; B.L. Bronson (October 1977). MICRO Information Management System (Version 5.0) Reference Manual. Institute of Labor and Industrial Relations (ILIR), University of Michigan and Wayne

State University. ^ "Oracle 30th Anniversary Timeline" (PDF). Archived (PDF) from the original on 2011-03-20. Retrieved 23 August 2017. ^ a b c "RDBMS Plenary Session: The Later Years" (PDF) (Interview). Interviewed by Burton Grad. Computer History Museum. 2007-06-12. Retrieved 2025-05-30. ^ Interview with Wayne Ratliff. The FoxPro History. Retrieved on 2013-07-12. ^ Development of an object-oriented DBMS; Portland, Oregon, United States; Pages: 472–482; 1986; ISBN 0-89791-204-7 ^ Jordan, Meghan. "NoSQL Latency". ScyllaDB. Retrieved 2025-06-09. ^ "SQL vs. NoSQL: Full comparison of features, differences, and more". www.testgorilla.com. Retrieved 2025-06-09. ^ Graves, Steve. "COTS Databases For Embedded Systems" Archived 2007-11-14 at the Wayback Machine, Embedded Computing Design magazine, January 2007. Retrieved on August 13, 2008. ^ Argumentation in Artificial Intelligence by Iyad Rahwan, Guillermo R. Simari ^ "OWL DL Semantics". Retrieved 10 December 2010. ^ Connolly & Begg 2014, p. 64. ^ Connolly & Begg 2014, pp. 97–102. ^ Connolly & Begg 2014, p. 102. ^ Chong et al. 2007. ^ Connolly & Begg 2014, pp. 106–113. ^ Connolly & Begg 2014, p. 65. ^ Chapple 2005. ^ "Structured Query Language (SQL)". International Business Machines. October 27, 2006. Retrieved 2007-06-10. ^ Wagner 2010. ^ Ramalho, J.C.; Faria, L.; Helder, S.; Coutada, M. (31 December 2013). "Database Preservation Toolkit: A flexible tool to normalize and give access to databases". Biblioteca Nacional de Portugal. University of Minho. ^ Paiho, Satu; Tuominen, Pekka; Rökman, Jyri; Ylikerälä, Markus; Pajula, Juha; Siikavirta, Hanne (2022). "Opportunities of collected city data for smart cities". IET Smart Cities. 4 (4): 275–291. doi:10.1049/smc2.12044. ISSN 2631-7680. S2CID 253467923. ^ David Y. Chan; Victoria Chiu; Miklos A. Vasarhelyi (2018). Continuous auditing : theory and application (1st ed.). Bingley, UK: Emerald Publishing. ISBN 978-1-78743-413-4. OCLC 1029759767. ^ Halder & Cortesi 2011. ^ Ben Linders (January 28, 2016). "How Database Administration Fits into DevOps". Retrieved April 15, 2017. ^ itl.nist.gov (1993) Integration Definition for Information Modeling (IDEFIX) Archived 2013-12-03 at the Wayback Machine. 21 December 1993. ^ a b Date 2003, pp. 31–32. Bachman, Charles W. (1973). "The Programmer as Navigator". Communications of the ACM. 16 (11): 653–658. doi:10.1145/355611.362534. Beynon-Davies, Paul (2003). Database Systems (3rd ed.). Palgrave Macmillan. ISBN 978-1403916013. Chapple, Mike (2005). "SQL Fundamentals". Databases. About.com. Archived from the original on 22 February 2009. Retrieved 28 January 2009. Childs, David L. (1968a). Description of a set-theoretic data structure (PDF) (Technical report). CONCOMP (Research in Conversational Use of Computers) Project. University of Michigan. Technical Report 3. Childs, David L. (1968b). Feasibility of a set-theoretic data structure: a general structure based on a reconstituted definition (PDF) (Technical report). CONCOMP (Research in Conversational Use of Computers) Project. University of Michigan. Technical Report 6. Chong, Raul F.; Wang, Xiaomei; Dang, Michael; Snow, Dwaine R. (2007). "Introduction to DB2". Understanding DB2: Learning Visually with Examples (2nd ed.). IBM Press Pearson plc. ISBN 978-0131580183. Retrieved 17 March 2013. Codd, Edgar F. (1970). "A Relational Model of Data for Large Shared Data Banks" (PDF). Communications of the ACM. 13 (6): 377–387. doi:10.1145/362384.362685. S2CID 207549016. Connolly, Thomas M.; Begg, Carolyn E. (2014). Database Systems – A Practical Approach to Design Implementation and Management (6th ed.). Pearson. ISBN 978-1292061184. Date, C. J. (2003). An Introduction to Database Systems (8th ed.). Pearson. ISBN 978-0321197849. Halder, Raju; Cortesi, Agostino (2011). "Abstract Interpretation of Database Query Languages" (PDF). Computer Languages, Systems & Structures. 38 (2): 123–157. doi:10.1016/j.cl.2011.10.004. ISSN 1477-8424. Hershey, William; Easthope, Carol (1972). A set theoretic data structure and retrieval language. Spring Joint Computer Conference, May 1972. ACM SIGIR Forum. Vol. 7, no. 4. pp. 45–55. doi:10.1145/1095495.1095500. Nelson, Anne Fulcher; Nelson, William Harris Morehead (2001). Building Electronic Commerce: With Web Database Constructions. Prentice Hall. ISBN 978-0201741308. North, Ken (10 March 2010). "Sets, Data Models and Data Independence". Dr. Dobb's. Archived from the original on 24 October 2012. Tsitchizris, Dionysios C.; Lochovsky, Fred H. (1982). Data Models. Prentice–Hall. ISBN 978-0131964280. Ullman, Jeffrey; Widom, Jennifer (1997). A First Course in Database Systems. Prentice–Hall. ISBN 978-0138613372. Wagner, Michael (2010), SQL/XML:2006 – Evaluierung der Standardkonformität ausgewählter Datenbanksysteme, Diplomica Verlag, ISBN 978-3836696098 Ling Liu and Tamer M. Özsu (Eds.) (2009). "Encyclopedia of Database Systems, 4100 p. 60 illus. ISBN 978-0-387-49616-0. Gray, J. and Reuter, A. Transaction Processing: Concepts and Techniques, 1st edition, Morgan Kaufmann Publishers, 1992. Kroenke, David M. and David J. Auer. Database Concepts. 3rd ed. New York: Prentice, 2007. Raghu Ramakrishnan and Johannes Gehrke, Database Management Systems. Abraham Silberschatz, Henry F. Korth, S. Sudarshan, Database System Concepts. Lightstone, S.; Teorey, T.; Nadeau, T. (2007). Physical Database Design: the database professional's guide to exploiting indexes, views, storage, and more. Morgan Kaufmann Press. ISBN 978-0-12-369389-1. Teorey, T.; Lightstone, S. and Nadeau, T. Database Modeling & Design: Logical Design, 4th edition, Morgan Kaufmann Press, 2005. ISBN 0-12-685352-5. CMU Database courses playlist MIT OCW 6.830 | Fall 2010 | Database Systems Berkeley CS W186 DB File extension – information about files with the DB extension Database at Wikipedia's sister projects:Definitions from WiktionaryMedia from CommonsNews from WikinewsQuotations from WikiquoteTexts from WikisourceTextbooks from WikibooksResources from Wikiversity Retrieved from " Share — copy and redistribute the material in any medium or format for any purpose, even commercially. Adapt — remix, transform, and build upon the material for any purpose, even commercially. The licensor cannot revoke these freedoms as long as you follow the license terms. Attribution — You must give appropriate credit , provide a link to the license, and indicate if changes were made . You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. ShareAlike — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits. You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation . No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material. DBMS > PostgreSQLPlease select another system to compare it with PostgreSQL.Our visitors often compare PostgreSQL with MySQL, Oracle and SQLite.Share this page Skip to content Optimize your data integration with Hevo! HomeAboutSolutionsProductsBlogsContactCareersThis article provides an overview of popular Database Management Systems (DBMS) used across industries. It covers PLSQL, MSSQL, MySQL, MariaDB, MongoDB, and SQLite, highlighting their key features, strengths, weaknesses, best use cases, and demand in the industry. PLSQL and MSSQL are widely used in enterprise environments, while MySQL and MariaDB are favored for web applications. MongoDB stands out in NoSQL solutions for scalability and flexibility, and SQLite is a lightweight solution for mobile and small-scale apps. The demand for these technologies continues to grow, with salaries varying by experience level.Databases are the backbone of most software applications today, enabling efficient data storage, retrieval, and management. In this article, we will explore some of the most popular Database Management Systems (DBMS) used in the industry. From traditional relational databases to modern NoSQL solutions, we will analyze them. 1. PLSQL Definition PLSQL (Procedural Language for SQL) is an extension of SQL used specifically in Oracle databases. It allows for procedural programming within SQL, enabling complex logic and error handling. Key Features: Procedural Constructs: Supports loops, conditions, and exception handling. Integration with SQL: Can seamlessly integrate with SQL queries for data manipulation. Cursors: Used for handling query result sets. Triggers and Stored Procedures: Allows creating stored procedures and triggers to automate tasks. Unique Features: Error Handling: PLSQL provides robust exception handling capabilities. Performance: Stored procedures can improve performance by reducing the number of database calls. Strengths and Weaknesses: Strengths: Efficient in managing large and complex datasets. Enhances performance through modularity and reusability. Weaknesses: Limited to Oracle databases, making it less flexible for cross-platform applications. Learning curve can be steep for beginners. Best For: Large enterprises using Oracle databases. Systems requiring high-level data manipulation and automation. Some Examples Where Used: Enterprise Resource Planning (ERP) systems. Financial applications requiring complex transaction processing. Demand in the Industry: PLSQL remains in high demand for roles focused on Oracle-based systems, especially in industries like banking, telecommunications, and healthcare. Entry-Level Job Average Income: $65,000 - $85,000 annually. Senior-Level Job Average Income: $110,000 - $140,000 annually. 2. MSSQL (Microsoft SQL Server) Definition MSSQL is a relational database management system (RDBMS) developed by Microsoft. It is designed for a wide range of enterprise applications and supports various tools for data management and analysis. Key Features: T-SQL Support: A proprietary extension of SQL for writing scripts, queries, and stored procedures. Data Security: Strong security features like data encryption and role-based access control. High Availability: Includes features like Always On Availability Groups for minimizing downtime. Unique Features: SQL Server Integration Services (SSIS): A tool for data integration and workflow management. SQL Server Reporting Services (SSRS): A tool for designing and delivering reports. Strengths and Weaknesses: Strengths: Excellent integration with Microsoft tools and services. Strong community support and documentation. Weaknesses: Can be resource-heavy, especially for smaller applications. Licensing can be expensive for larger deployments. Best For: Businesses that are already using Microsoft-based technologies. Data warehousing and business intelligence applications. Some Examples Where Used: E-commerce platforms. Data-driven decision-making applications in businesses. Demand in the Industry: MSSQL is widely used across many sectors, including finance, healthcare, and retail. Entry-Level Job Average Income: $70,000 - $90,000 annually. Senior-Level Job Average Income: $120,000 - $150,000 annually. 3. MySQL Definition MySQL is an open-source relational database management system that uses SQL for querying and managing data. It is widely used for web-based applications. Key Features: Open-Source: Free to use and customizable. Replication: Allows for replication of databases across multiple servers for high availability. ACID-Compliant: Ensures reliable and safe transactions. Unique Features: InnoDB Engine: Supports foreign key constraints and transactions. Wide Adoption in Web Development: Commonly used in LAMP stack (Linux, Apache, MySQL, PHP/Perl/Python). Strengths and Weaknesses: Strengths: Open-source, which makes it accessible for small to medium businesses. Large developer community for support and resources. Weaknesses: Limited in terms of advanced analytics and data processing capabilities. Not as feature-rich as enterprise-grade solutions like Oracle. Best For: Startups and small-to-medium businesses. Web applications and content management systems (CMS). Some Examples Where Used: WordPress-powered websites. E-commerce platforms like Magento and WooCommerce. Demand in the Industry: MySQL remains popular, particularly for web development and open-source projects. Entry-Level Job Average Income: $60,000 - $80,000 annually. Senior-Level Job Average Income: $100,000 - $130,000 annually. 4. MariaDB Definition MariaDB is a community-developed fork of MySQL that is designed to be fully compatible with MySQL while offering additional features and improvements. Key Features: Improved Performance: Faster than MySQL in certain scenarios, with better query optimization. Compatibility: Fully compatible with MySQL, making it easy to migrate. Security Features: Improved encryption and authentication mechanisms. Unique Features: Storage Engines: Supports multiple storage engines like Aria, TokuDB, and ColumnStore. Open-Source: Completely open-source with no commercial licensing. Strengths and Weaknesses: Strengths: Better performance and scalability compared to MySQL. No licensing costs. Weaknesses: Fewer enterprise tools and features compared to larger DBMS options. Less widely adopted in the enterprise space. Best For: Open-source web applications. Companies looking for a free and scalable database solution. Some Examples Where Used: Online platforms using LAMP stack. Media content delivery applications. Demand in the Industry: MariaDB is growing in popularity among companies seeking open-source database solutions. Entry-Level Job Average Income: $55,000 - $75,000 annually. Senior-Level Job Average Income: $90,000 - $120,000 annually. 5. MongoDB Definition MongoDB is a NoSQL database designed for handling large volumes of unstructured or semi-structured data. It stores data in a flexible, JSON-like format called BSON. Key Features: Document-Oriented: Stores data in documents (JSON/BSON), allowing for flexible schema. Horizontal Scaling: Designed for scalability across multiple servers. Aggregation Framework: Powerful querying and data transformation capabilities. Unique Features: Schema-Less: Allows for the easy storage of data without a predefined schema. Sharding: Distributes data across multiple servers for load balancing. Strengths and Weaknesses: Strengths: Highly scalable and flexible for handling unstructured data. Ideal for big data and real-time analytics applications. Weaknesses: Lack of joins, which can make complex queries difficult. Data consistency can be an issue in distributed environments. Best For: Applications that require flexible schema and scalability. Real-time analytics and big data applications. Some Examples Where Used: Social media platforms. Internet of Things (IoT) applications. Demand in the Industry: MongoDB is a leader in the NoSQL space and is in high demand, especially in startups and big data companies. Entry-Level Job Average Income: $70,000 - $90,000 annually. Senior-Level Job Average Income: $120,000 - $150,000 annually. 6. SQLite Definition SQLite is a lightweight, serverless relational database engine that is used for small-scale applications. It is self-contained and requires minimal setup. Key Features: Serverless: No need for a separate server process; it runs directly within the application. Lightweight: Suitable for embedded systems and mobile devices. Cross-Platform: Works on various operating systems, including Windows, macOS, and Linux. Unique Features: Zero Configuration: Simple setup, making it ideal for embedded and mobile applications. File-Based: Database is stored as a single file on the filesystem. Strengths and Weaknesses: Strengths: Lightweight and easy to integrate into applications. Ideal for mobile apps and small-scale websites. Weaknesses: Not suitable for large-scale, high-performance applications. Lacks advanced features like clustering and replication. Best For: Mobile applications. Small-scale projects and prototypes. Some Examples Where Used: Mobile apps like Android and iOS apps. Lightweight desktop applications. Demand in the Industry: SQLite is commonly used in smaller applications and embedded systems but is less common for enterprise solutions. Entry-Level Job Average Income: $50,000 - $70,000 annually. Senior-Level Job Average Income: $80,000 - $100,000 annually. Conclusion The DBMS landscape offers a wide variety of options, each with its strengths and weaknesses. The right choice depends on the specific needs of the project, including scalability, flexibility, and ease of use. Relational databases like PLSQL, MSSQL, and MySQL are still widely used in many industries, while MongoDB and MariaDB are gaining traction in modern, scalable applications. As companies continue to shift towards cloud-based and data-driven solutions, the demand for skilled professionals in these technologies will only increase.what is a database management systemdatabase management systemsbusiness intelligence databasedata integrity in nosqlsql vs mongodb redditmariadb vs mysql differencebest databases for small-scale projectsdatabase performance and query optimizationRelated articlesTurkey's IT industry is rapidly growing, driven by its strategic geographic location, a young and educated workforce, and strong government support. The country's thriving startup ecosystem, emphasis on digital transformation, and increasing IT exports are positioning Turkey as a significant player in the global technology landscape. This growth trajectory highlights Turkey's potential and ambition in shaping the future of technology both domestically and internationally.Turkey is quickly becoming a key player in the IT outsourcing industry, thanks to its strategic location, skilled workforce, and cost-effective solutions. Positioned between Europe and Asia, Turkey offers businesses access to top-tier IT talent at competitive prices, with minimal language and cultural barriers. The country's commitment to data protection and innovation further enhances its appeal as a reliable and cutting-edge IT outsourcing partner. A database is a structured and systematic way of storing information to be accessed, analyzed, transformed, updated and moved (to other databases). A database is a means of organizing information in a way that users can quickly navigate data, spot trends and perform other actions. Although databases may come in different formats, most are stored on computers for greater convenience. To begin understanding databases, consider an Excel notebook or Google sheet. Spreadsheets like these are a basic form of a table. Databases are almost exclusively organized in tables and those tables have rows and columns. So, think of a simple database as a collection of spreadsheets (or tables) joined together in a systematic way. Related Reading From Built In ExpertsPython Databases 101: How to Choose a Database Library Database Tutorial for Beginners. | Video: Lucid Software What Is a Database? A database is a structured collection of information, or data, stored electronically in a computer system, often managed using a database management system (DBMS). Databases come in many formats — like relational, non-relational and more — in order to handle different types of data and perform various tasks. Databases can be hosted on servers located on-premises at an organization's office, off-premises at an organization's data center or within an organization's cloud infrastructure. Why Do We Use Databases? Computerized databases are the foundation for handling information related to products, data analysis, business processes and more. Many of the services you use online every day (banking, social media, shopping, email) are all built on top of databases. Today, databases are used for many reasons. Databases Hold Data Efficiently We use databases because they are an extremely efficient way of holding vast amounts of data and information. Databases around the world store everything from your credit card transactions to every click you make within one of your social media accounts. Given there are billions of people on the planet, that's a lot of data. Databases Allow Smooth Transactions Databases allow access to various services which, in turn, allow you to access your accounts and perform transactions all across the internet. For example, your bank's login page will ping a database to figure out if you've entered the right password and username. Your favorite online shop pings your credit card's database to pull down the funds needed for you to buy that item you've been eyeing. Databases Update Information Quickly Databases allow for easy information updates on a regular basis. Creating a post on a social media account, directly depositing your salary into your bank account or buying a plane ticket for your next vacation are all updates made to a database and displayed back to you almost instantaneously. Databases Simplify Data Analysis Databases make research and data analysis much easier because they are highly structured storage areas of data and information. This means businesses and organizations can easily analyze databases once they know how a database is structured. Common structures (e.g. table formats, cell structures like date or currency fields) and common database querying languages (e.g., SQL) make database analysis easy and efficient. Types of Databases There are many types of databases used today. Below are some of the more prominent ones. 1. Hierarchical Databases Hierarchical databases were the earliest form of databases. You can think of these databases like a simplified family tree. There's a singular parent object (like a table) that has child objects (or tables) under it. A parent can have one or many child objects but a child object only has one parent. The benefit of these databases is that they're incredibly fast and efficient, plus there's a clear, threaded relationship from one object to another. The downside to hierarchical databases is that they're very rigid and highly structured. 2. Relational Databases Relational databases are perhaps the most popular type of database. Relational databases are set up to connect their objects (like tables) to each other with keys. For example, there might be one table with user information (name, username, date of birth, customer number) and another table with purchase information (customer number, item purchased, price paid). In this example, the key that creates a relationship between the tables is the customer number. 3. Non-Relational or NoSQL Databases Non-relational databases were invented more recently than relational databases and hierarchical databases in response to the growing complexity of web applications. Non-relational databases are any database that doesn't use a relational model. You might also see them referred to as NoSQL databases. Non-relational databases store data in different ways, such as unstructured data, structured document format or as a graph. Relational databases are based on a rigid structure, whereas non-relational databases are more flexible. 4. Cloud Databases Cloud databases refer to information that's accessible in a hybrid or cloud environment. All users need is an internet connection to reach their files and manipulate them like any other database. A convenience of cloud databases is that they don't require extra hardware to create more storage space. Users can either build a cloud database themselves or pay for a service to get started. 5. Centralized Databases Centralized databases are contained within a single computer or another physical system. Although users may access data through devices connected within a network, the database itself operates from one location. This approach may work best for larger companies or organizations that want to prioritize data security and efficiency. 6. Distributed Databases Distributed databases run on more than one device. That can be as simple as operating several computers on the same site, or a network that connects to many devices. An advantage of this method is that if one computer goes down, the other computers and devices keep functioning.   7. Object-Oriented Databases Object-oriented databases perceive data as objects and classes. Objects are specific data — like names and videos — while classes are groups of objects. Storing data as objects means users don't have to distribute data across tables. This makes it easier to determine the relationships between variables and analyze the data. 8. Graph Databases Graph databases highlight the relationships between various data points. While users may have to do extra work to determine trends in other types of databases, graph databases store relationships right next to the data itself. Users can then immediately see how various data points are connected to each other. What Are the Components of a Database? The components of a database vary slightly depending on whether the database is hierarchical, relational or non-relational. However, here's a list of database components you might expect to be associated with any database. Schema The database schema is essentially the design of the database. A schema is developed at the early conceptual stages of building a database. It's also a valuable source of ongoing information for those wanting to understand the database's design. Constraints and Rules Databases use constraints to determine what types of tables can (and cannot) be stored and what types of data can live in the columns or rows of the database tables, for example. These constraints are important because they ensure data is structured, less corruptible by unsanctioned data structures and that the database is regulated so users know what to expect. These constraints are also the reason why databases are considered rigid. Metadata Metadata is essentially the data about the data. For example, the metadata of a photo may include the date it was taken, its location and its file format. Each database or object has metadata, which the database software reads in order to understand what's in the database. You can think of metadata as the database schema design and constraints combined together so a machine knows what kind of database it is and what actions can (or can't) be performed within the database. Query Language Each database can be queried. In this case, "queried" means people or services can access the database. That querying is done by way of a particular language or code snippet. The most common querying language is SQL (Structured Query Language) but there are also many other languages and even SQL variations like MySQL, Presto and Hive. Objects Each database is a collection of objects. There are a few different types of objects stored within databases such as tables, views, indexes, sequences and synonyms. The most well known of these are tables, which are like spreadsheets that organize data in rows and columns. You may also hear the term "object instance," which is simply an instance or element of an object. For example, a table called "Transactions" in a database is an instance of the object-type table. Database Advantages The structured nature of databases offers a range of benefits for professional and casual users alike. Below are some of the more prominent advantages: Improved data sharing and handling Improved data storage capacity Improved data integrity and data security Reduced data inconsistency Quick data access Increased productivity Improved data-driven decision making Database Disadvantages Although databases can be helpful for many, there are some limitations to consider before investing in a database: High cost High complexity Required dedicated database management staff Risk of database failure Applications of Databases When used correctly, databases can be a helpful tool for organizations in various industries looking to better arrange their information. Common use cases include: Healthcare: storing massive amounts of patient data. Logistics: monitoring and analyzing route information and delivery statuses. Insurance: storing customer data like addresses, policy details and driver history. Finance: handling account details, invoices, stock information and other assets. E-commerce: compiling and arranging data on products and customer behavior. Transportation: storing passengers' names, scheduled flights and check-in status. Manufacturing: keeping track of machinery status and production goals. Marketing: collecting data on demographics, purchasing habits and website visits. Education: tracking student grades, course schedules and more. Human resources: organizing personnel info, benefits and tax information. What Is a Database Management System? A database management system (DBMS) is a software package used to create and manage databases. In other words, a DBMS makes it possible for users to actually interact with the database. The DBMS is the user interface (UI) that allows us to access, add, modify and delete content from the database. There are several types of database management systems, including relational, non-relational and hierarchical. Evolution of Databases Storing information is nothing new, but the rise of computers in the 1960s marked a shift toward more digital forms of databases. While working for GE, Charles Bachman created the Integrated Data Store, ushering in a new age of computerized databases. IBM soon followed suit with its Information Management System, a hierarchical database. In the 1970s, IBM's Edgar F. Codd released a paper touting the benefits of relational databases, leading to IBM and the University of California, Berkeley releasing their own models. Relational databases became popular in the following years, with more businesses developing models and using SQL. Even though object-oriented databases became an alternative in the 1980s, relational databases remained the gold standard. The invention of the World Wide Web led to greater demand for databases in the 1990s. MySQL and NoSQL databases entered the scene, competing with the commercial databases developed by businesses. Object-oriented databases also began to replace relational databases in popularity.   During the 2000s and 2010s, organizations began to collect larger volumes of data, and many turned to the scalability offered by NoSQL databases. Distributed databases provided another way to organize this proliferating data, storing it away in multiple locations.   Future of Databases As organizations handle increasing amounts of data, future databases must be able to keep up. Users will expect databases to be accessible across the globe and able to deal with limitless volumes of data. As a result, it's likely that more companies will migrate their data to cloud environments. With the increase in data has also come a spike in cybersecurity threats, so organizations can be expected to complement their cloud environments with reinforced security measures. Databases will become more easily accessible for authorized personnel only, while companies adopt tools and best practices for keeping their data out of the wrong hands. Spreadsheets organize data into rows and columns, with each individual cell housing the actual data. Databases also employ rows and columns, but each cell contains a record of data gathered from an external table. As a result, databases provide more ways to arrange and structure information as opposed to spreadsheets. Relational databases are the most commonly used database type. A database is a highly organized information that is designed to be easily accessible and navigable for users. Most databases are stored on computers, making it possible to quickly analyze, transform and manipulate data in other ways. Common types of databases include: Relational databases Non-relational (NoSQL) databases Hierarchal databases Cloud databases Centralized databases Distributed databases Object-oriented databases Graph databases No, Microsoft Excel is a spreadsheet software program used to organize data in a spreadsheet format. However, Excel can connect to data sources like Microsoft Access and Microsoft SQL Server databases. Data is the cornerstone of any modern software application, and databases are the most common way to store and manage data used by applications. With the explosion of web and cloud technologies, databases have evolved from traditional relational databases to more advanced types of databases such as NoSQL, columnar, key-value, hierarchical, and distributed databases. Each type has the ability to handle structured, semi-structured, and even unstructured data. On top of that, databases are continuously handling mission-critical and sensitive data. When this is coupled with compliance requirements and the distributed nature of most data sets, managing databases has become highly complex. As a result, organizations require robust, secure, and user-friendly tools to maintain these databases. This is where database management systems come into play—by offering a platform to manage databases. Let's take a look. Introduction of DBMS What is a database management system (DBMS)? A database management system (DBMS) is a software tool for creating, managing, and reading a database. With DBMS, users can access and interact with the underlying data in the database. These actions can range from simply querying data to defining database schemas that fundamentally affect the structure of DBMS. Furthermore, DBMS allows users to interact with a database securely and concurrently without interfering with each user and while maintaining data integrity. Unlock the potential of IT Service Management with BMC Helix ITSM. › What are the functions of DBMS? The typical DBMS tasks or functions include: User access and control. Administrators can easily configure user accounts, define access policies, modify restrictions and access scopes to limit access to underlying data, control user actions, and manage database users. Data backups and snapshots. DBMS can simplify the database backup process through a simpler and straightforward interface for managing backups and snapshots. For safekeeping, users can move these backups to third-party locations, such as cloud storage. Performance tuning. DBMS can monitor database performance using integrated tools. Users can tune databases by creating optimized indexes to reduce I/O usage and optimize SQL queries for the best database performance. Data recovery. DBMS provides a recovery platform and the necessary tools to fully or partially restore databases to their previous state—effortlessly. Database query language and APIs. Access and use data via a variety of query languages and API connections. Data dictionary management. Dictionaries include metadata about the structure of the data and relationships between data points so that functionality can rely on structural abstractions rather than complex coding. Data transformation and display. DBMS transforms data on command, such as assembling attributes for the month, day and year as December 14, 2024, or 12/14/24 or another specified display format. Management of data integrity. DBMS establishes and maintains data consistency and minimizes duplications. User access. This policy permits more than one user to access the database at a time and follows ACID to accommodate multiple users. User interface. Whether accessing data through a web form, a direct dashboard, or a third-party distributed network, a browser-based interface makes it easy. All these administrative tasks are facilitated using a single management interface. Most modern DBMS support handling multiple database workloads from a centralized DBMS software, even in a distributed database scenario. Furthermore, they allow organizations to have a governable top-down view of all the data, users, groups, locations, etc., in an organized manner. (Explore the role of DBAs, or database administrators.) How does DBMS work? The various DBMS components work together to create an integrated system for structuring and storing data, supporting user queries and access, ensuring consistency and integrity, control, security, backups, and logging. The following DBMS schematic illustrates how a DBMS system works: What are the components of a DBMS? All DBMS comes with various integrated components and tools necessary to carry out almost all database management tasks. Some DBMS software even provides the ability to extend beyond the core functionality by integrating with third-party tools and services, directly or via plugins. In this section, we will look at the common components of a DBMS that are universal across all database software: 1. Storage engine in a database The database storage engine is the core component of the DBMS that interacts with the file system at an OS level to store data. All SQL queries which interact with the underlying data go through the storage engine. Which storage engine is the best for a database? The right storage engine depends on your data model. SQL engines supporting transactions work well with relational databases. Non-relational models, especially those that require scalability, work best with MongoDB or Cassandra. 2. Database query language What is a database access language? A database access language is required for interacting with a database, from creating databases to simply inserting or retrieving data. A proper DBMS must support one or multiple query languages and language dialects. Structured query language (SQL) and MongoDB Query Language (MQL) are two query languages that are used to interact with the databases. What are the 4 types of DBMS languages? In many query languages, the query language functionality can be further categorized according to specific tasks: Data Definition Language (DDL). This consists of commands that can be used to define database schemas or modify the structure of database objects. Data Manipulation Language (DML). Commands that directly deal with the data in the database. All CRUD operations come under DML. Data Control Language (DCL). This deals with the permissions and other access controls of the database. Transaction Control Language (TCL). Command which deals with internal database transactions. 3. Query processor The query processor is the intermediary between user queries and the database. In DBMS, query processing is the process of interpreting user queries, such as SQL, and making them actionable commands that the database can understand to perform the appropriate functionality. What are the components of the query processor? The query processor components each work together to extract data. Parser. This component translates a user query into a database language such as SQL, parses it for correct syntax, and verifies its logical meaning. Optimizer. This component converts the query into logical relational operations, identifies how much time and energy it will take to execute the query, and then specifies the exact operations and sequence for the most efficient execution. Execution engine. This is the component that carries out the query, implements algorithms and operators according to the optimized plan, and finally retrieves and formats the results. Query cache. Some systems include a component that stores frequently executed queries and results to save time and improve performance. 4. Optimization engine in DBMS The optimization engine allows the DBMS to provide insights into the performance of the database in terms of optimizing the database itself and queries. When coupled with database monitoring tools, it can provide a powerful toolset to gain the best performance out of the database. 5. Metadata catalog A metadata catalog, also referred to as a data catalog, is the centralized catalog of all the objects within the database. When an object is created, the DBMS keeps a record of that object with some metadata about it using the metadata catalog. Then, this record can be used to: Verify user requests to the appropriate database objects Provide an overview of the complete database structure 6. Log manager The log manager is a component that will keep all the logs of the DBMS. These logs will consist of user logins and activity, database functions, backups and restore functions, etc. The log manager ensures all these logs are properly recorded and easily accessible. (Compare logs to monitoring.) 7. Reporting & monitoring tools Reporting and monitoring tools are another standard component that comes with a DBMS. DBMS reporting tools will enable users to generate reports while monitoring tools enable monitoring the databases for resource consumption, user activity, etc. 8. Data utilities In addition to all the above, most DBMS software comes with additional inbuilt utilities to provide functionality such as: Data integrity checks Backup and restore Simple database repair Data validations Etc. Scale operational effectiveness with an artificial intelligence for IT operations. Learn more about AIOps with BMC! › What are the different types of DBMS? The evolution of data models, how data is structured, and the use cases of each has led to various types of DBMS. The most commonly used are: 1. Relational database management systems (RDBMS) Relational Database Management Systems are the most common type of DBMS. Relational databases interact with databases that contain structured data in a table format with predefined relationships. Moreover, they use structured query language (SQL) to interact with databases. Some popular examples of RDBMS include: Microsoft SQL MySQL Oracle Database MariaDB PostgreSQL 2. NoSQL databases NoSQL (nonrelational) databases are designed for semi-structured and unstructured data. They offer greater data modeling flexibility and often don't use a schema. They also support scaling across distributed systems. Examples of nonrelational or NoSQL databases include: MongoDB Azure Cosmos DB Apache Cassandra CouchDB Amazon DynamoDB 3. Object-oriented DBMS (OODBMS) This type of database stores data and data relationships as objects that can be used by object-oriented programming languages like C++ and Java in applications such as CAD systems, databases containing scientific research, and multimedia. Examples of object-oriented databases include: ObjectDB Versant GemStone/S Objectivity/DB 4. Hierarchical DBMS This type of database uses tree-like structures to organize data in parent-child relationships. A parent node can have many children and grandchildren, but each child node has only one parent. These DBMSs work well when data has well-defined relationships that can be organized into files and directories. Examples of hierarchical databases include: IBM Information Management System (IMS) RDM Mobile Windows Registry XML data storage 5. Network DBMS This type of database supports complex interconnections in many-to-many data relationships, with records that have multiple and complex links. Examples of databases that use the network model include: IDMS (Integrated Database Management System) Oracle CODASYL 6. Columnar database management systems (CDBMS) As the name suggests, CDMBS is used to manage columnar databases that store data in columns instead of rows, emphasizing high performance. Some databases that use columnar format are Apache Cassandra, Apache HBase, etc. What are the advantages of DBMS? DBMS was introduced to solve the fundamental issues associated with storing, managing, accessing, securing, and auditing data in traditional file systems. Software users and organizations can gain the following advantages of DBMS: 1. Increased data security DBMS provides the ability to control users and enforce policies for security and compliance management. This controlled user access increases the database security and makes the data less vulnerable to security breaches. 2. Simple data sharing DBMS enables users to access the database securely regardless of their location. Thus, they can handle any database-related task promptly without the need for complex access methods or worrying about database security. On top of that, DBMS allows multiple users to collaborate effectively when interacting with the database. 3. Data integration DBMS allows users to gain a centralized view of databases spread across multiple locations and manage them using a single interface rather than operating them as separate entities. 4. Abstraction & independence DBMS enables users to change the physical schema of a database without changing the logical schema that governs database relationships. As a result, organizations can scale the underlying database infrastructure without affecting the database operations. Furthermore, any change to the logical schema can also be carried out without affecting applications that access the databases. 5. Streamlined backup & recovery mechanism Most databases have built-in backup and recovery tools. Yet, DBMS offers centralized tools to facilitate backup and recovery functionality more conveniently and thereby provide a better user experience. Securing data has become easier than ever with functionality like: Automated snapshots Backup scheduling Backup verifications Multiple recovery methods 6. Uniform management & monitoring DBMS provides a single interface to carry out all the management and monitoring tasks, thus simplifying the workload of database administrators. These tasks can range from database creation and schema modifications to reporting and auditing. Why is DBMS important? Considering the many advantages, DBMS is essential for any organization when managing databases. With different DBMS providing different feature sets, it is paramount that organizations rigorously evaluate the DBMS software before committing to a single system. However, a properly configured DBMS will greatly simplify the management and maintenance of databases at any scale. The scale, complexity, and feature set of a DBMS will depend on the specific DBMS and the organization's requirements. Related reading These postings are my own and do not necessarily represent BMC's position, strategies, or opinion. See an error or have a suggestion? Please let us know by emailing . Every digital product — from banking apps to ride-sharing platforms — relies on a database under the hood.But here's the thing: there's no one-size-fits-all database.Some applications need strict consistency and structure, while others require flexible schemas, lightning-fast writes, or massive scale across regions. That's why there are multiple types of databases, each optimized for a different kind of workload.This guide breaks it all down for you:What are the main types of databases used in modern systems?When should you choose a relational database vs. a NoSQL store?What's the difference between a graph database and a document database?Which databases are used in real-world companies — and why?If you're building software, managing infrastructure, or just exploring how data systems work, understanding the core types of databases is a critical foundation.In a few minutes, you'll explore 9 major types of databases, how each works, and when to use them — so you can build faster, more innovative, and more scalable applications. Looking to sync data across different types of databases — in real time?Estuary Flow lets you connect relational, NoSQL, time-series, and more using streaming pipelines with zero code. Explore the platform → | or create a free account →What is a Database?A database is a system that stores, organizes, and retrieves data efficiently, usually through software called a Database Management System (DBMS).In technical terms:A database is a structured collection of data that can be easily accessed, managed, and updated.But here's the more straightforward way to think about it:Imagine a giant, organized digital filing cabinet. Each drawer holds data like customer records, product inventories, or user activity logs. A database is that cabinet — optimized to answer questions like:"What's this user's last login?""How many orders came in today?""What's the relationship between these two products?"Key Characteristics of a Database:Structured or unstructured data storageFast and reliable data accessScalability for small apps or massive systemsAbility to read, write, update, and delete dataSecurity and access controlWhy Databases MatterWhether you're tracking transactions in a financial system or collecting sensor data from a smart device, you need a reliable way to store and query that information. Databases do exactly that — and choosing the right one affects everything from speed to scalability to cost.In short, a database isn't just a place to store data — it's the brain behind your application's ability to think, respond, and scale.Different Types of DatabasesThere are several types of databases, each optimized for a specific data structure, performance goal, or use case. Below, we break down the 9 most widely used database types, with definitions, examples, and when to use each.1. Relational Databases (RDBMS)A relational database organizes data into predefined tables made up of rows and columns — much like a spreadsheet. Each table holds data about a specific entity (like users or orders), and relationships between tables are defined using keys.Definition: A relational database stores and allows access to related data points through structured schemas and SQL (Structured Query Language).Key Features:Structured schema: Data follows a strict formatACID compliance: Ensures data accuracy through transactionsSQL support: Industry-standard querying languageTable relationships: Foreign keys connect different datasetsRelational Database Examples:MySql - Popular open-source RDBMS, widely used in web appsPostgreSQL - Advanced features, open-source, strong for analyticsOracle Database - Enterprise-grade performance and securitySQL Server - Microsoft's commercial RDBMS, great for .NET applicationsCommon Use Cases:E-commerce (orders, products, customers)Banking and finance (transactions, accounts)ERP and CRM systemsAny application requiring strict consistency and structured dataPros:Highly reliable and mature technologyStrong data integrity and consistencyGreat tooling and community supportLimitations:It is not ideal for highly flexible or unstructured dataScaling horizontally can be challengingWhen to Use a Relational Database:Choose a relational database when your data has clear structure, relationships, and you need strong consistency — especially for financial systems, inventory, or user records.2. NoSQL DatabasesA NoSQL database offers a flexible alternative to traditional relational models by allowing developers to store and retrieve data in formats such as key-value pairs, documents, wide columns, or graphs. These databases are built to handle unstructured and semi-structured data, scale horizontally, and deliver high performance for modern applications.Definition: A NoSQL (Not Only SQL) database is a non-relational database designed for distributed data storage. It offers dynamic schemas and high scalability, ideal for large volumes of diverse or rapidly changing data.Key Features:Schema flexibility: No predefined structure is requiredHigh performance: Optimized for fast reads/writes at scaleHorizontal scalability: Easily distributes data across multiple serversSupport for diverse data models: Document, key-value, column, graphNoSQL Database Examples:MongoDB – Document-oriented NoSQL database using JSON-like documentsRedis – In-memory key-value store known for ultra-fast performanceApache Cassandra – Wide-column database built for high availability and scaleCouchbase – Distributed document database with built-in cachingCommon Use Cases:Real-time analytics and dashboardsIoT and sensor data ingestionMobile apps with dynamic data modelsPersonalization engines and recommendation systemsCaching and session managementPros:Highly flexible for evolving data structuresBuilt for web-scale applicationsExcellent read/write performanceCan store large volumes of diverse dataLimitations:Data consistency may be weaker (eventual consistency models)Complex queries and relationships can be more complicated to manageLacks standardized querying language like SQLWhen to Use a NoSQL Database:Use a NoSQL database when your application handles large-scale, rapidly changing, or non-relational data — such as user activity logs, product catalogs, or content feeds. NoSQL is ideal when flexibility and horizontal scalability are more important than strict data consistency.3. Time-Series DatabasesA time-series database (TSDB) is optimized for storing and querying data that changes over time. Instead of general-purpose storage, these databases are purpose-built for time-stamped data, making them ideal for IoT applications, system monitoring, and real-time analytics.Definition: A time-series database is a type of database designed to store, index, and analyze data points associated with specific timestamps — typically collected at regular intervals over time.Key Features:Timestamp indexing: Every entry is linked to a time valueEfficient storage and compression for large volumes of sequential dataHigh write throughput for continuous data ingestionBuilt-in functions for aggregation, downsampling, and windowingTime-Series Database Examples:InfluxDB – Purpose-built TSDB with high performance and query supportTimescaleDB – PostgreSQL extension for time-series workloadsPrometheus – Monitoring system and TSDB used for cloud-native observabilityOpenTSDB – Scalable time-series platform built on HBaseCommon Use Cases:IoT data collection (sensors, smart devices)Server and infrastructure monitoringReal-time financial market dataEnergy usage tracking and forecastingIndustrial telemetry systemsPros:Highly optimized for time-based queriesScales well for high-ingest workloadsNative support for time-based analysis (e.g., averages, anomalies)Often integrates well with visualization tools like GrafanaLimitations:Not suited for general-purpose relational dataQuerying non-time data can be inefficientLimited support for complex joins or transactionsWhen to Use a Time-Series Database:Choose a time-series database when your primary data consists of events over time, such as IoT sensor metrics, application performance stats, or stock prices. It best fits systems that rely on real-time monitoring and historical trend analysis.4. Graph DatabasesA graph database is designed to represent and store data with complex relationships using structures called nodes (entities) and edges (relationships). Instead of relying on tables and joins, graph databases make relationship-based queries fast, intuitive, and efficient.Definition: A graph database, a type of NoSQL database, employs graph structures with nodes, edges, and properties to store and query data. This structure makes graph databases ideal for analyzing and navigating relationships within the data.Key Features:Native graph storage: Built around nodes and edges, not tablesHigh-performance traversals: Find deep relationships quicklyFlexible schema: Adapts easily to evolving structuresOptimized for connected data: Ideal for many-to-many relationshipsGraph Database Examples:Neo4j – Leading open-source graph database with Cypher query languageAmazon Neptune – Fully managed graph DB service on AWSArangoDB – Multi-model database supporting graph, document, and key-valueTigerGraph – Enterprise-grade graph analytics platformCommon Use Cases:Social networks (e.g., user connections, followers)Recommendation engines (e.g., "users who bought this also bought…")Fraud detection in financial systemsKnowledge graphs and semantic searchSupply chain and logistics path optimizationPros:Excellent for querying complex relationshipsSimplifies many-to-many data modelingIdeal for real-time relationship analysisHighly visual and intuitive data structureLimitations:Not suitable for flat, tabular dataIt may require a learning curve with specialized query languagesIt can become resource-intensive for extremely large graphsWhen to Use a Graph Database:Use a graph database when relationships are first-class citizens in your data model — like in social graphs, identity resolution, or recommendation systems. If your app frequently asks questions like "how is A connected to B?" — graph is the right fit.5. Key-Value DatabasesA key-value database stores data as a simple collection of key-value pairs. Each key acts as a unique identifier, and the value holds the associated data — which can be anything from a string to a complex object. This design delivers ultra-fast performance for basic read/write operations.Definition: A NoSQL database that stores data in a dictionary-like format, using unique keys to access each piece of data, is known as a key-value database.Key Features:Efficient and straightforward: Ideal for quick lookupsIn-memory or persistent storage optionsLow latency access even under heavy loadNo fixed schema, supporting flexible valuesKey-Value Database Examples:Redis – In-memory key-value store, great for caching and real-time applicationsAmazon DynamoDB – Fully managed key-value and document DB at a massive scaleRiak KV – Distributed key-value database for high-availabilityAerospike – High-performance NoSQL DB with strong consistencyCommon Use Cases:Caching and session managementUser profile storageShopping cart data in e-commerce appsLeaderboard and gaming statsAPI rate limiting and token storagePros:High-speed performanceSimple data model and APIHorizontally scalableEasy to deploy and maintainLimitations:It is not ideal for complex queries or relationshipsLack of native data integrity featuresIt can be inefficient for analytics or structured reportingWhen to Use a Key-Value Database:Use a key-value database when you need fast, simple access to data — like in caching systems, real-time gaming, or session storage. It's best for high-volume workloads that don't require relational logic.6. Document DatabasesA document database stores data as documents, typically in JSON or BSON format. Each document contains key-value pairs, but unlike traditional databases, the structure of each document can vary — making it highly flexible for modern applications.Definition: A document database is a type of NoSQL database that stores and retrieves data as semi-structured documents, allowing for nested fields and flexible schemas.Key Features:Schema-less structure: Documents can have different shapesRich data types: Supports nesting, arrays, and complex objectsPowerful query capabilities on document fieldsHorizontal scalability for large-scale applicationsDocument Database Examples:MongoDB – The most popular document store; it supports dynamic schemas and advanced queriesCouchDB – JSON-based document database with a RESTful HTTP APICouchbase – Combines key-value and document features with built-in cachingFirestore (Firebase) – Cloud-hosted NoSQL document store by Google for mobile/web appsCommon Use Cases:Content management systems (blogs, product catalogs)User profiles and preferencesE-commerce product dataMobile or IoT applicationsReal-time collaboration toolsPros:Excellent flexibility for evolving app requirementsMore straightforward mapping to modern programming objectsSuitable for both small and large-scale systemsSchema changes don't require downtimeLimitations:Complex relationships can be more challenging to modelQuery performance can degrade if not indexed properlyMay lack strong consistency in distributed setupsWhen to Use a Document Database:Use a document database when your application works with semi-structured or evolving data, like product catalogs, user profiles, or CMS content. It's a great fit for schema flexibility with powerful query capabilities.7. Columnar DatabasesA columnar database stores data by columns rather than rows, enabling fast analytical queries over massive datasets. Unlike row-based databases, which retrieve an entire row, column-oriented systems fetch only the specific columns needed, making them ideal for business intelligence and data warehousing.Definition: A columnar database is a type of database optimized for reading and aggregating data stored in columns instead of rows, making it well-suited for high-performance analytical workloads.Key Features:Column-based storage: Efficient for scanning large datasetsHigh compression: Repeated values compress well by columnOptimized for OLAP (Online Analytical Processing)Ideal for aggregation-heavy queriesColumnar Database Examples:Apache HBase – Wide-column NoSQL database built on top of HadoopClickHouse – Open-source columnar DB for real-time analyticsAmazon Redshift – Managed data warehouse with columnar storageGoogle BigQuery – Serverless, highly scalable cloud data warehouseCommon Use Cases:Business intelligence dashboardsData lake queryingAd hoc data explorationReporting and KPI trackingETL pipelines and analytics platformsPros:Blazing-fast performance for SELECT and aggregation queriesLower I/O usage and better compressionScales well with petabyte-scale datasetsIntegrated with popular BI and visualization toolsLimitations:Not ideal for frequent writes or transactional workloadsMore complex to manage for real-time updatesNot suited for high-volume, low-latency operational queriesWhen to Use a Columnar Database:Choose a columnar database when your primary goal is to analyze large volumes of structured data, such as in data warehouses, marketing analytics, or performance monitoring systems. It's the go-to choice for read-heavy, analytical environments.9. NewSQL DatabasesNewSQL databases aim to bridge the gap between traditional relational databases and modern NoSQL systems. They offer the scalability and performance of NoSQL while preserving the ACID guarantees, SQL interface, and relational structure of RDBMSs.Definition: A NewSQL database is a modern, distributed SQL database that delivers horizontal scalability and high throughput without sacrificing data consistency or relational capabilities.Key Features:Full SQL support with joins, constraints, and transactionsACID compliance even at a global scaleDistributed architecture for high availability and scalabilityAutomatic sharding and fault toleranceNewSQL Database Examples:Google Cloud Spanner – Globally distributed RDBMS with external consistencyCockroachDB – Open-source, cloud-native SQL database built for scaleTiDB – MySQL-compatible distributed SQL database with hybrid OLTP/OLAP capabilitiesVoltDB – In-memory NewSQL platform for fast transaction processingCommon Use Cases:SaaS platforms and multi-tenant appsGlobal financial systemsReal-time fraud detectionLarge-scale transactional systemsEnterprise systems requiring both scale and integrityPros:Combines relational consistency with a horizontal scaleFamiliar SQL interface for easy adoptionGreat for mission-critical systemsBuilt-in resilience and cloud-native featuresLimitations:Newer ecosystem compared to traditional RDBMSsOperational complexity can be higherHigher resource requirements in some configurationsWhen to Use a NewSQL Database:Use a NewSQL database when your application demands relational data integrity with NoSQL-level scalability — for example, in high-traffic financial platforms, e-commerce systems, or cloud-native enterprise apps that can't compromise consistency.Working Across Databases in Real-Time?If you're building systems that connect multiple database types — say, syncing data from PostgreSQL to Snowflake or MongoDB to BigQuery — consider using a platform like Estuary Flow.Estuary Flow is a real-time data integration tool that supports both streaming and batch pipelines, with native connectors for relational, NoSQL, time-series, and warehouse destinations. Whether you're dealing with CDC, analytics, or hybrid architectures, Flow helps unify your pipeline — without writing custom code.Conclusion: Choosing the Right Type of DatabaseUnderstanding the types of databases is crucial for designing scalable, efficient, and reliable applications. From relational databases like PostgreSQL to NoSQL options like MongoDB and from time-series systems to graph databases, each type is optimized for a specific use case.The key is choosing the right database type for your workload — whether you need consistency, speed, flexibility, or scalability.If you're building:A transactional financial system → go with a relational or NewSQL databaseA flexible content app → choose a document databaseA high-speed caching layer → use a key-value storeAn analytics dashboard → opt for a columnar databaseA social network or fraud detection system → leverage a graph databaseA time-based monitoring tool → implement a time-series databaseStill unsure? Many modern systems use polyglot persistence, combining multiple database types to serve different application components.Frequently Asked Questions (FAQs)What are the main types of databases?The most common types of databases include:Relational databases (RDBMS)NoSQL databases (document, key-value, graph, wide-column)Time-series databasesGraph databasesColumnar databasesObject-oriented databasesNewSQL databasesWhich type of database is best for structured data?Relational databases like MySQL or PostgreSQL are best for structured data with well-defined schemas and relationships.What's the difference between SQL and NoSQL databases?SQL (Relational) databases use tables and require fixed schemas.NoSQL databases allow for flexible schemas and scale horizontally. They include document, key-value, graph, and column-oriented models.Can one application use multiple types of databases?Yes — this approach is called polyglot persistence. It's common in complex systems that need to balance flexibility, speed, and consistency.When should I use a time-series database?Use a time-series database when you're storing time-stamped metrics — such as server logs, IoT sensor data, or financial trading history.Are NewSQL databases better than traditional RDBMS?NewSQL databases offer the familiar SQL interface and ACID guarantees of traditional RDBMS, with the added benefit of horizontal scalability. They're great for modern, cloud-native applications.If your architecture includes multiple database types, Estuary Flow makes it easy to unify them into a single, real-time pipeline — without writing custom code. Database applications are crucial tools for businesses of all sizes, helping them organize and manage vast amounts of data efficiently. With the rise of big data and the need for real-time analytics, the demand for advanced database applications has never been higher. In this article, we will explore the top 5 of the current top 10 database applications, along with an analysis of why they are leading the pack. 1. Oracle Database: Oracle Database has long been a dominant player in the database market, known for its reliability, scalability, and comprehensive feature set. It offers a range of advanced features, including support for multi-tenancy, in-memory processing, and advanced analytics. Oracle Database is particularly popular in large enterprises that require a high-performance database solution for mission-critical applications. 2. Microsoft SQL Server: Microsoft SQL Server is another popular choice for businesses looking for a robust and user-friendly database solution. It offers excellent integration with other Microsoft products, such as Excel and Power BI, making it a preferred choice for organizations that rely heavily on Microsoft technologies. SQL Server also excels in performance, security, and scalability, making it a top choice for businesses of all sizes. 3. MySQL: As an open-source database application, MySQL has gained popularity for its cost-effectiveness and ease of use. It is particularly well-suited for small to medium-sized businesses that require a reliable database solution without the hefty price tag. MySQL is known for its speed, scalability, and flexibility, making it a versatile option for a wide range of applications. 4. PostgreSQL: PostgreSQL is an advanced open-source database application that is highly regarded for its robust feature set and extensibility. It offers strong support for complex data types, advanced indexing, and sophisticated query optimization, making it a top choice for businesses with demanding database requirements. PostgreSQL is particularly popular among developers who appreciate its flexibility and rich ecosystem of extensions. 5. MongoDB: MongoDB is a leading NoSQL database application that has gained popularity for its flexibility, scalability, and ease of use. It is particularly well-suited for applications that require fast and agile development, such as web and mobile apps. MongoDB's document-based data model and distributed architecture make it a top choice for businesses looking to handle large volumes of unstructured data efficiently. Emerging Trends in Database Applications: 1. Cloud-Based Database Solutions: With the increasing adoption of cloud computing, businesses are turning to cloud-based database solutions to streamline their data management processes. Cloud databases offer scalability, flexibility, and cost-effectiveness, making them an attractive option for businesses of all sizes. Leading cloud database providers, such as Amazon Web Services (AWS) and Microsoft Azure, offer a range of database services tailored to meet the specific needs of businesses. 2. AI and Machine Learning Integration: AI and machine learning are revolutionizing the way businesses analyze and leverage their data. Database applications are increasingly incorporating AI and machine learning capabilities to automate tasks, improve data quality, and provide real-time insights. By harnessing the power of AI, businesses can uncover valuable patterns and trends in their data, enabling them to make more informed decisions. 3. Real-Time Data Processing: In today's fast-paced business environment, real-time data processing is essential for staying competitive. Database applications are evolving to support real-time analytics, enabling businesses to make instant decisions based on the latest data. Real-time data processing allows businesses to respond quickly to changing market conditions, optimize operations, and deliver personalized customer experiences. 4. Data Security and Compliance: Data security and compliance are top priorities for businesses in light of increasing data breaches and regulatory requirements. Database applications are enhancing their security features to protect sensitive data from unauthorized access and ensure compliance with data privacy regulations, such as GDPR and CCPA. Advanced encryption, access controls, and audit trails are becoming standard features in modern database applications to safeguard data integrity. 5. Multi-Model Databases: Multi-model databases are gaining traction as businesses seek to manage diverse data types in a unified platform. These databases support multiple data models, such as relational, document, graph, and key-value, allowing businesses to store and analyze different types of data efficiently. Multi-model databases offer flexibility and scalability, enabling businesses to adapt to changing data requirements and drive innovation. 6. Edge Computing: Edge computing is reshaping the way businesses process and analyze data by moving compute resources closer to the data source. Database applications are evolving to support edge computing architectures, enabling businesses to process data locally and reduce latency. Edge computing is particularly valuable for applications that require real-time processing, such as IoT devices, autonomous vehicles, and industrial automation. 7. Blockchain Integration: Blockchain technology is disrupting the database landscape by providing a secure and transparent way to record transactions. Database applications are exploring blockchain integration to enhance data integrity, traceability, and security. By leveraging blockchain technology, businesses can create immutable records of transactions, establish trust among stakeholders, and streamline data sharing across networks. In conclusion, the top 5 database applications, including Oracle Database, Microsoft SQL Server, MySQL, PostgreSQL, and MongoDB, are leading the pack with their advanced features, scalability, and reliability. Businesses are increasingly turning to cloud-based solutions, integrating AI and machine learning, adopting real-time data processing, enhancing data security and compliance, embracing multi-model databases, leveraging edge computing, and exploring blockchain integration to stay ahead in the competitive landscape. By staying abreast of emerging trends in database applications, businesses can unlock the full potential of their data and drive innovation across their organizations. There are various database management systems known in the market that many companies use according to their need and use. As we know, almost every service from social media, payment applications, various booking services through applications, etc, has enormous amounts of data on the server. Suppose you are playing a multiplayer game, and there is a score that gets changed every second. Suppose a situation where a video gets viral, and people all over the globe add comments to the video. All these activities lead to the addition of data to the database center.

This data continuously gets added to the database, and people update and manipulate it using database management systems. The prime concern for these services is data that needs to be stored, fetched on time, and also deleted if needed. And database management system does this very efficiently for the organization. Some of the most popular database names are Oracle, MySQL, MongoDB, etc., which we will discuss in this article. Before this, we need to get some brief knowledge about the databases that will be a prerequisite for this article.What is a Database?The database is defined as a collection of enormous information stored in a logical way so that the user can retrieve this data easily anytime they want. Generally, a database is not only stored in a computer. Any stored information such as a paper file having all the information about a person, a contact file of your cell phone, etc all considered a database.In the language of computer science, a database is used for organizing data of very large size. When we surf various websites on the internet or use software for our utility, a database is used to store all the information and logical activities done by the user. And to control (add, store, fetch, and delete data) the database, we use software known as a database management system. Using the DBMS, users can interact with the data stored in the database.Top 10 Most Popular Databases in 20251. OracleThe Oracle database was developed by the company namely Oracle Corporation. The name of the oracle developer was Lawrence Ellison and it was developed in the year 1977. This database was the first database that supported the features like enterprise grid computing and data warehousing. Enterprise grid computing refers to the distribution of computer nodes over various geographical locations to obtain results for a common goal. This feature helps to work the database flexibly and also it costs lesser. We use SQL queries to interact with the oracle database. Oracle database is supported in various operating systems such as IBM, macOS, Windows, etc. Also, it supports the Linux OS after the versions in the late 1990s. The average salary for an Oracle database developer is 4.8 lakh per annum in India. Various features of the oracle database are:SecurityAvailabilityScalabilityMulti-platform supportedBackup and recoverySupports cloud-based serviceApart from features, there are also some disadvantages of the oracle database such as High price, Complexity, etc.2. MySQLMySQL is an open-source database that is most widely used across the globe. This database was developed in the year 1995. Later in 2009, Oracle acquired MySQL. The term SQL stands for “structured query language”. MySQL database stores data in tables using rows and columns. The working of MySQL takes place in three steps.The client stores the data in tables by defining the relationship between the entities. Then the clients make requests to fetch any data from the database using MySQL queries. In last, the server application responds to these requested queries and provides the appropriate data to the client. The server of MySQL is fast, safe, scalable, and user-friendly. MySQL is supported on both the client and server sides. The average salary for a Mysql database developer is 4.0 lakh per annum in India.3. Microsoft SQL ServerMS SQL server is a very popular database around the globe that was developed by the company “Microsoft” in 1989. This database was written in language C and C++. SQL. The server can be run on either computer or over a network. It is supported in both 32-bit and 64-bit systems. There are three editions of SQL server that are Enterprise edition, Standard edition, and Server WEB. My SQL server follows a Client-Server Architecture so the users can directly fetch the data from the server. The three components of the SQL server areNetwork ProtocolsDatabase EnginesSQL OSIn the MS SQL server, we can install different versions of this database on a single machine that works independently. It uses the concept of “Instances”. This makes it possible to work on different databases on a single computer and also there is less risk of data security due to the instances. The average salary for a SQL Server Developer is 2.0 lakh per annum in India.4. PostgreSQLPostgreSQL is an open-source database management system that is very popular among users. It stands for (post-gress-Q-L). This database follows the concept of ACID properties (Atomicity, Consistency, Isolation, and Durability). It supports many advanced programming languages such as Java, Ruby, Perl, etc. Various operating systems supported by PostgreSQL are macOS, windows, UX, Solaris, etc. Various data types that can be stored in this database are text, images, sounds, etc. PostgreSQL also supports Open Database Connectivity (ODBC). In PostgreSQL, various modern and advanced features are supported such as:It supports complex queriesIt supports Trigger and View featuresStreaming replicationForeign keysSome of the disadvantages of PostgreSQL are it is not organized and maintained by a single company. The transaction is slow as compared to MySQL. Also, the installation may be difficult for beginners. The average salary for a PostgreSQL Developer is 3.6 lakh per annum in India.Do read about Database Management Systems to gain in-depth knowledge.5. MongoDBMongoDB is an open-source database developed in the year 2009 by a company namely 10gen Software. It stores data in the form of a document and collections which means the data are not stored using rows and columns. The schema of this database is dynamic. There are key-value pairs in the MongoDB database, which is the basic unit of this database. Although, it is not a relational database the performance of MongoDB is very much faster as compared to a relational database. Various features of MongoDB are scalability. indexing, master-slave replication, high performance, etc. It does not support joint operations. Data is stored in a JSON document and it is able to store heterogeneous data. The average salary for a MongoDB Developer is 6.0 lakh per annum in India.Some of the disadvantages of MongoDB are:Utilises high memory storageLimited data in the BSON document6. IBM Db2The IBM DB2 database was developed by a company namely “IBM” in 1990. This database was developed keeping in mind that the data should be effortlessly stored and retrieved. Using IBM DB2, we can do both transactional and analytical operations in the database. The IBM DB2 also supports some no SQL features such as XML, JSON, etc. In this database, the logical and physical structures are managed separately and are used to create the IBM DB2 database is DDL (Data Definition Language). IBM DB2 is capable to supports the machine learning algorithm and augmented data explorer. Various platforms supported by the IBM DB2 database are Windows, Linux, and Unix. The average salary for an IBM Db2 Developer is 10.0 lakh per annum in India.7. RedisRedis is an open-source database management system that is licensed by the BSD. Redis is a no SQL database and it works on the principle of key-value pair. This database is also termed a “Data Structure Database” because the key of this database contains strings, hashes, lists, sets, and sorted sets. The development of the Redis database was done in the C language. There are two types of architecture supported in the Redis database that is:Redis Client Redis ServerThere are various features of the Redis database such as speed, persistence, sharding, atomic operation, portability, etc. The average salary for a Redis Database Developer is 5.0 lakh per annum in India.8. ElasticSearchElasticSearch is a non-relational NoSQL database. It is a real-time database management system and it was developed by Shay Banon in 2010 in the Java language. Elasticsearch is a highly scalable database that can be used to store larger logs containing large data. This database is developed in RESTfil APIs that help this database to execute any request and give an appropriate response for the requested query. In Elasticsearch, we can perform both structured and unstructured searches. It can be used for textual search, data aggregation, metrics analysis, etc. This database used the concept of denormalization. As it is developed using the Java programming language so it is compatible with every platform. Some of the advantages of ElasticSearch database are scalability, integrity, real-time, etc.9. Microsoft AccessMicrosoft Access is a database that is developed by Microsoft company in 1992. It also comes as a part of the Microsoft 365 office suite. This database is a combination of Jet Database Engine and software development tools with a pinch of GUI. It is similar to MS Excel but it is more efficient than Excel in terms of data storage and manipulation. It stored the data in tables in the form of rows and columns. Microsoft Access is able to link to the data stored in other databases directly. The various components of the MS Access database are Tables, Macros, Modules, Queries, Reports, etc. The four steps to access the data from MS Access are as follows:Creation of dataData inputQuery to fetch the required dataReport10. SQLiteSQLite is a subset of MySQL and t is a serverless and embedded database. It is a library that is self-contained and needs a zero-configuration. Many top multinational companies such as Adobe uses SQLite database in their organization. We can simply use this database over the network. It is a transactional SQL database engine. We do not need to configure this SQLite in our system. This database stands on the principle of ACID property. SQLite was written in the ANSI-C language and it is easy to use ASPI. We can use the SQLite database where there is embedded software such as cell phones, televisions, etc. It is also able to control low HTTP requests. Using this database, we can work on multiple datasets without any intervention with makes it more flexible. Various OS supporting the SQLite databases are LINUX, macOS, Android, etc. The average salary for an SQLite Developer is 3.0 lakh per annum in India.It is important to have a look at relational databases and non-relational databases. ConclusionThe database is defined as a collection of enormous amounts of information stored logically so that the user can retrieve this data easily anytime they want. The two types of databases are Relational databases and Non-relational databases. Relational database stores data in tables using rows and columns and non-relational database stores data in a documented format. There are many databases that are used by professionals around the globe according to their uses and priorities. Some of the most popular databases are Oracle SQL, MySQL, SQLite, MongoDB, etc. DBMS > MySQLPlease select another system to compare it with MySQL.Our visitors often compare MySQL with PostgreSQL, Microsoft SQL Server and Oracle.Share this page RSS FeedRanking > Complete RankingThe DB-Engines Ranking ranks database management systems according to their popularity. The ranking is updated monthly.Read more about the method of calculating the scores.424 systems in ranking, July 20251.1.1.OracleRelational, Multi-model 1217.05-13.33-23.312.2.2.MySQLRelational, Multi-model 940.73-12.85-98.733.3.3.Microsoft SQL ServerRelational, Multi-model 771.14-5.61-36.514.4.4.PostgreSQLRelational, Multi-model 680.89+0.23+41.985.5.5.MongoDB Detailed vendor-provided information availableDocument, Multi-model 403.83+0.99-25.996.6. 7.SnowflakeRelational176.17+1.68+39.647.7. 6.RedisKey-value, Multi-model 149.72-2.01-7.058.8. 9.IBM Db2Relational, Multi-model 127.51+2.38+3.119.9. 8.ElasticsearchMulti-model 118.83-2.45-12.0010.10.10.SQLiteRelational115.44-1.60+5.4911.11. 12.Apache CassandraWide column, Multi-model Wide column storeVector DBMS108.76+0.49+9.6312.12. 15.DatabricksMulti-model Document storeRelational DBMS108.04+3.36+24.7413.13. 14.MariaDB Detailed vendor-provided information availableRelational, Multi-model 95.44+0.91+4.8614.14. 11.Microsoft AccessRelational90.47+2.19-10.1715.15. 17.Amazon DynamoDBMulti-model Document storeKey-value store84.15+0.81+13.2016. 17.16.Microsoft Azure SQL DatabaseRelational, Multi-model 76.51+1.20-2.2417. 16. 19.Apache HiveRelational75.74-0.94+18.4518.18. 13.SplunkSearch engine69.52-0.10-23.4019.19. 18.Google BigQueryRelational64.19-0.35+6.3820.20. 21.Neo4jGraph54.64+3.30+8.8821. 22. 24.Apache SolrSearch engine, Multi-model Search engineSpatial DBMS34.17+0.71-4.7122. 21. 20.FileMakerRelational33.69-0.33-14.9023.23. 22.TeradataRelational, Multi-model 33.30+0.71-11.2224.24. 23.SAP HANARelational, Multi-model 32.52+0.79-11.6025.25.25.SAP Adaptive ServerRelational, Multi-model Relational DBMSSpatial DBMS27.64-0.26-7.1026.26. 33.Apache Spark (SQL)Relational23.80+1.07+6.0927.27. 26.Apache HBaseWide column22.72+0.04-5.0128. 29.28.InfluxDBTime Series, Multi-model Time Series DBMSSpatial DBMS22.65+0.91-0.9529. 28. 27.Microsoft Azure Cosmos DBMulti-model Document storeGraph DBMSKey-value storeWide column storeSpatial DBMS21.68-0.75-5.4430.30. 37.ClickHouseRelational, Multi-model 18.54-0.06+3.3731.31. 29.PostGISSpatial, Multi-model Spatial DBMSRelational DBMS18.38-0.10-2.5832.32. 35.OpenSearchMulti-model Search engineVector DBMSVector DBMS17.30+0.40+0.6633.33. 30.FirebirdRelational17.02+0.45-3.3934.34. 32.MemcachedKey-value16.37+0.67-1.3735. 36. 34.InformixRelational, Multi-model 14.68+0.18-2.8836. 35. 38.Amazon RedshiftRelational14.64-0.33-0.4237.37. 31.Microsoft Azure Synapse AnalyticsRelational14.44+0.11-5.4038.38. 39.Firebase Realtime DatabaseDocument14.35+0.12+0.4739.39. 36.CouchbaseMulti-model 12.96-0.83-3.4540.40.40.Apache ImpalaRelational, Multi-model Relational DBMSDocument store12.75+0.62+0.1841.41.41.Apache FlinkRelational12.42+0.61+2.1942.42.42.VerticaRelational, Multi-model 9.41-0.04-0.5543.43. 54.Google Cloud FirestoreDocument8.11+0.32+1.3644. 46. 43.dBASERelational7.62+0.38-1.8945. 44. 52.Amazon AuroraRelational, Multi-model 7.52+0.17+0.2046. 48. 45.GreenplumRelational, Multi-model 7.15+0.09-1.0847. 45. 65.DuckDBRelational7.15-0.15+2.3648. 50. 51.PrometheusTime Series7.12+0.17-0.2149. 51. 48.CouchDBDocument, Multi-model Document storeSpatial DBMS7.12+0.35-0.7050. 52. 49.KdbMulti-model Time Series DBMSVector DBMSRelational DBMS7.10+0.36-0.4851. 49. 53.etcdKey-value6.89-0.16-0.1052. 47. 47.H2Relational, Multi-model Relational DBMSSpatial DBMS6.89-0.18-1.1353. 54. 46.NetezzaRelational6.75+0.36-1.4154. 53. 50.RealmDocument6.45-0.14-0.8855. 56. 58.AlgoliaSearch engine6.06+0.29+0.4956. 55. 44.PrestoRelational6.05-0.11-2.4557.57. 56.SphinxSearch engine5.62-0.05-0.3358. 61. 57.Microsoft Azure AI SearchSearch engine, Multi-model Search engineVector DBMS5.27+0.04-0.4159. 58.59.TrinoRelational, Multi-model 5.26-0.11-0.1760. 62. 55.Oracle EssbaseRelational5.11+0.01-0.8761. 60. 60.HazelcastKey-value, Multi-model Key-value storeDocument store5.04-0.19-0.2762. 59. 61.Aerospike Detailed vendor-provided information availableMulti-model Document storeGraph DBMSKey-value storeSpatial DBMSVector DBMS5.00-0.26-0.1663.63. 64.GraphiteTime Series4.41-0.07-0.6364.64. 66.CockroachDBRelational4.35+0.26-0.1065. 68. 88.PineconeVector4.29+0.41+1.1666. 65. 75.ScyllaDB Detailed vendor-provided information availableWide column, Multi-model Wide column storeKey-value store3.98+0.02+0.1567.67.67.MarkLogicMulti-model Document storeNative XML DBMSRDF storeSearch engine3.94+0.04-0.4968. 66. 62.Apache JackrabbitContent3.83-0.12-1.3269.69. 74.TimescaleDBTime Series, Multi-model Time Series DBMSRelational DBMS3.77+0.14-0.1370. 71. 63.Datastax EnterpriseWide column, Multi-model 3.66+0.04-1.4471. 78. 82.TiDBRelational, Multi-model Relational DBMSDocument store3.63+0.35+0.1772.72. 71.Google Cloud DatastoreDocument3.540.00-0.5273. 76. 93.Apache DruidMulti-model Relational DBMSTime Series DBMS3.53+0.14+0.5974. 75. 80.OpenEdgeRelational3.50+0.09-0.0875. 73. 70.SingleStoreRelational, Multi-model 3.45-0.02-0.6776. 74. 94.Oracle NoSQLMulti-model Document storeKey-value storeRelational DBMS3.41-0.05+0.4877. 80.77.IngresRelational3.38+0.15-0.4278. 85. 89.MilvusVector3.36+0.34+0.2479. 77. 76.InterbaseRelational3.33-0.03-0.5080. 70. 69.EhcacheKey-value3.31-0.32-0.9581.81. 73.Riak KVKey-value3.24+0.04-0.7182. 79. 104.QuestDBTime Series, Multi-model Time Series DBMSRelational DBMS3.24+0.00+0.5883. 82. 92.Apache IgniteMulti-model Key-value storeRelational DBMS3.12-0.06+0.1184. 88. 97.RocksDBKey-value3.12+0.26+0.2385. 83. 68.Apache DerbyRelational3.10-0.07-1.3286. 84. 81.Microsoft Azure Data ExplorerRelational, Multi-model 3.01-0.03-0.5587. 86. 79.SAP SQL AnywhereRelational2.97-0.03-0.6188. 87. 85.ArangoDBMulti-model Document storeGraph DBMSKey-value storeSearch engine2.83-0.06-0.5889. 91. 86.HyperSQLRelational2.71+0.01-0.5790. 97. 100.Google Cloud SpannerRelational2.71+0.12-0.0691. 89. 72.VirtuosoMulti-model Document storeGraph DBMSNative XML DBMSRelational DBMSRDF storeSearch engineSpatial DBMS2.64-0.07-1.3392. 90. 90.OrientDBMulti-model Document storeGraph DBMSKey-value store2.64-0.06-0.4293. 95. 84.Microsoft Azure Table StorageWide column2.61+0.00-0.8094. 92. 99.RavenDBDocument, Multi-model 2.59-0.06-0.2295. 94. 102.AdabasMultivalue2.58-0.04-0.1496. 93. 103.IBM CloudantDocument2.57-0.06-0.1097. 96. 96.Google Cloud BigtableMulti-model Key-value storeWide column store2.56-0.03-0.3798.98.98.GraphDBMulti-model Graph DBMSRDF store2.54-0.02-0.3399. 102. 163.QdrantVector2.54+0.12+1.27100.100. 95.MemgraphGraph2.53+0.05-0.40101.101. 138.OceanBaseMulti-model 2.51+0.08+0.83102. 99. 101.GemFireKey-value, Multi-model Key-value storeDocument store2.51+0.02-0.24103. 110. 87.InterSystems IRISMulti-model Document storeKey-value storeObject oriented DBMSRelational DBMS2.35+0.22-0.90104. 109. 115.Amazon NeptuneMulti-model Graph DBMSRDF store2.34+0.21+0.21105. 103. 91.Apache Jena - TDBRDF2.34-0.05-0.71106. 104. 105.SAP IQRelational2.320.00-0.29107.107. 122.CoveoSearch engine2.31+0.08+0.33108. 106. 109.UniData,UniVerseMultivalue2.29+0.05-0.09109. 105. 106.RethinkDBDocument, Multi-model Document storeSpatial DBMS2.29-0.03-0.25110. 108. 83.Apache AccumuloWide column2.21+0.04-1.25111. 126. 148.WeaviateVector2.15+0.34+0.63112. 121. 120.NebulaGraphGraph2.09+0.20+0.08113. 112. 78.DolphinDBMulti-model Time Series DBMSVector DBMSRelational DBMS2.06+0.02-1.56114. 111. 110.YugabyteDBRelational, Multi-model 2.03-0.06-0.29115. 116. 129.ChromaVector2.00+0.07+0.12116. 122. 112.4DRelational1.92+0.05-0.39117.117. 118.Percona Server for MySQLRelational1.91-0.02-0.13118. 120. 114.LevelDBKey-value1.91+0.02-0.28119. 118. 108.InfinispanKey-value1.91-0.01-0.49120. 119. 116.CitusRelational, Multi-model Relational DBMSDocument store1.90+0.01-0.19121. 114. 113.MaxDBRelational1.90-0.05-0.30122. 113. 121.GridDB Detailed vendor-provided information availableTime Series, Multi-model 1.90-0.08-0.09123. 115. 111.PouchDBDocument1.89-0.06-0.42124. 127. 131.Amazon DocumentDBDocument1.84+0.04-0.04125.125. 126.Apache PhoenixRelational1.83+0.00-0.08126. 130. 133.Amazon CloudSearchSearch engine1.83+0.09+0.04127. 124. 124.Apache DrillMulti-model Document storeRelational DBMS1.78-0.05-0.17128. 123. 117.StardogMulti-model Graph DBMSRDF store1.78-0.06-0.29129. 128. 125.CloudKitDocument1.78-0.02-0.14130. 129. 128.JanusGraphGraph1.73-0.02-0.16131.131. 123.LMDBKey-value1.72+0.00-0.25132.132. 127.Oracle Berkeley DBMulti-model Key-value storeNative XML DBMS1.68-0.03-0.22133. 134. 132.Amazon SimpleDBKey-value1.67+0.03-0.18134. 133. 119.Oracle CoherenceKey-value1.670.00-0.36135. 136. 137.MonetDBRelational, Multi-model 1.61+0.04-0.09136. 135. 130.EDB PostgresRelational, Multi-model 1.61+0.01-0.27137. 148. 166.HFSQLRelational1.57+0.15+0.33138. 137. 107.TDengineTime Series, Multi-model Time Series DBMSRelational DBMS1.550.00-0.90139. 138. 134.RRDtoolTime Series1.55+0.01-0.20140. 152. 172.VictoriaMetricsTime Series1.55+0.19+0.42141. 140. 136.EXASOLRelational1.50+0.00-0.21142. 141.ValkeyKey-value1.47-0.03143. 144. 146.DatomicRelational1.45+0.01-0.12144. 159. 154.TiberoRelational1.45+0.20+0.03145. 146. 150.FaunaMulti-model Document storeGraph DBMSRelational DBMSTime Series DBMS1.45+0.01-0.04146. 143. 141.OpenTSDBTime Series1.44-0.01-0.18147. 145. 144.SpatiaLiteSpatial, Multi-model Spatial DBMSRelational DBMS1.44-0.01-0.15148. 142. 167.Apache IoTDBTime Series1.43-0.04+0.20149.149. 145.TigerGraphGraph1.43+0.02-0.16150. 151. 184.GBaseRelational1.39+0.03+0.36151. 147. 139.BaseXNative XML1.39-0.04-0.30152. 154. 155.jBASEMultivalue1.38+0.06-0.01153. 139. 140.IMSNavigational1.36-0.17-0.29154. 150. 143.TarantoolMulti-model Document storeKey-value storeRelational DBMSSpatial DBMS1.34-0.04-0.26155. 153. 165.D3Multivalue1.330.00+0.08156. 155. 142.GeodeKey-value1.33+0.02-0.29157. 156. 152.GridGainMulti-model ColumnarKey-value storeObject oriented DBMSRelational DBMS1.32+0.02-0.12158. 174. 218.StarRocksRelational1.31+0.20+0.56159. 161. 174.Amazon TimestreamTime Series1.30+0.08+0.18160. 157. 135.EmpressRelational1.28+0.01-0.46161. 168. 204.TDSQL for MySQLRelational, Multi-model 1.25+0.11+0.41162. 158. 215.Alibaba Cloud PolarDBRelational1.25-0.01+0.49163. 172. 187.MeilisearchMulti-model Search engineVector DBMS1.24+0.10+0.27164.164. 151.VoltDBRelational1.22+0.01-0.23165. 160. 149.Actian NoSQL DatabaseObject oriented1.22-0.01-0.28166. 165. 153.DgraphGraph1.200.00-0.22167. 166. 157.ObjectStoreObject oriented1.19+0.00-0.19168. 175. 175.SQLBaseRelational1.19+0.08+0.08169. 162. 147.HEAVY.AIRelational, Multi-model Relational DBMSSpatial DBMS1.18-0.04-0.34170. 163. 159.IBM Db2 warehouseRelational1.17-0.04-0.14171. 167. 170.DatameerDocument1.17+0.02+0.02172. 170. 161.MnesiaDocument1.15+0.01-0.13173. 171. 164.TimesTenRelational1.12-0.01-0.13174. 169. 156.Db4oObject oriented1.10-0.04-0.28175. 177. 181.EventStoreDBEvent1.10+0.02+0.06176. 178. 173.ObjectBoxMulti-model Object oriented DBMSVector DBMSTime Series DBMS1.09+0.01-0.04177. 181. 162.PlanetScaleRelational, Multi-model 1.08+0.02-0.19178. 186. 211.Alibaba Cloud MaxComputeRelational1.08+0.07+0.30179. 173. 160.mSQLRelational1.07-0.04-0.22180. 176. 191.GigaSpacesMulti-model 1.06-0.04+0.13181. 182. 169.GiraphGraph1.06+0.01-0.10182. 180. 180.SurrealDBMulti-model Document storeGraph DBMS1.06-0.01+0.02183. 184. 168.LiteDBDocument1.04+0.00-0.16184. 179. 190.CubridRelational1.04-0.03+0.09185. 183. 205.VitessRelational, Multi-model 1.03-0.01+0.19186. 185. 171.Apache KylinRelational1.03+0.03-0.12187.187. 192.Amazon KeyspacesWide column1.01+0.03+0.09188. 190. 179.openGaussRelational, Multi-model 0.97+0.03-0.07189. 191. 182.Oracle RdbRelational0.96+0.04-0.07190. 189. 183.NonStop SQLRelational0.950.00-0.09191. 195. 185.FoundationDBMulti-model Document storeKey-value storeRelational DBMS0.92+0.03-0.10192. 194. 178.Apache HAWQRelational0.91+0.03-0.13193. 188. 194.IDMSNavigational0.90-0.07-0.01194. 192. 186.GT.MKey-value0.90+0.00-0.11195. 201. 224.TypesenseSearch engine0.88+0.03+0.18196.196. 176.DataEaseRelational0.88+0.00-0.19197. 203. 196.AltibaseRelational0.87+0.06-0.03198. 193. 231.VespaMulti-model Search engineVector DBMS0.86-0.02+0.23199. 206.199.Alibaba Cloud AnalyticDB for MySQLRelational, Multi-model Relational DBMSDocument store0.85+0.05-0.03200. 197. 189.M3DBTime Series0.85-0.02-0.10201. 198. 233.eXtremeDBMulti-model Relational DBMSTime Series DBMS0.83-0.04+0.20202.202. 193.NCacheKey-value, Multi-model 0.83+0.00-0.08203. 199. 202.HPE Ezmeral Data FabricMulti-model Document storeWide column store0.83-0.03-0.03204. 200. 197.RocksetDocument, Multi-model 0.82-0.03-0.07205. 204. 201.NuoDBRelational0.82+0.01-0.05206. 208. 195.InfobrightRelational0.80+0.03-0.11207. 223. 158.FireboltRelational0.79+0.13-0.56208. 213. 213.SciDBMultivalue0.75+0.03-0.02209. 207. 207.YellowbrickRelational0.75-0.03-0.07210. 211. 198.AllegroGraphMulti-model Document storeGraph DBMSRDF storeVector DBMSSpatial DBMS0.75-0.01-0.14211. 205. 200.MatrixOneRelational0.74-0.06-0.12212. 215. 209.BigchainDBDocument0.74+0.02-0.06213. 209. 217.XapianSearch engine0.74-0.02-0.01214. 217. 220.Alibaba Cloud AnalyticDB for PostgreSQLRelational0.73+0.02+0.00215. 210. 221.BlazegraphMulti-model Graph DBMSRDF store0.73-0.03-0.01216.216. 206.Model 204Multivalue0.72+0.00-0.11217. 219. 208.GeoMesaSpatial0.71+0.02-0.09218. 214. 240.RisingWaveRelational0.71-0.01+0.13219. 212. 237.KeyDBKey-value0.71-0.03+0.08220. 221. 212.Datacom/DBRelational0.69+0.01-0.09221. 222. 219.eXist-dbNative XML0.68+0.00-0.06222. 218. 203.ZODBKey-value0.68-0.03-0.16223. 220. 214.Actian VectorRelational0.68-0.01-0.09224.224. 228.RDF4JRDF0.64-0.01-0.02225. 226.225.Objectivity/DBObject oriented0.630.00-0.05226. 227. 177.SednaNative XML0.63+0.00-0.44227. 225. 222.CrateDBMulti-model Document storeSpatial DBMSSearch engineTime Series DBMSVector DBMSRelational DBMS0.62-0.02-0.10228.228. 230.SQream DBRelational0.62+0.00-0.02229. 230. 216.BoltDBKey-value0.62+0.02-0.14230. 238. 236.TypeDBMulti-model Graph DBMSObject oriented DBMSRelational DBMS0.62+0.05-0.01231. 229. 188.DoltRelational, Multi-model Relational DBMSDocument store0.61+0.00-0.34232. 237. 229.DBISAMRelational0.61+0.04-0.05233. 235. 232.Graph EngineMulti-model Graph DBMSKey-value store0.59+0.02-0.05234. 232. 226.1010dataRelational0.58+0.00-0.09235. 240. 238.KairosDBTime Series0.58+0.03-0.02236. 231. 227.solidDBRelational0.58-0.02-0.09237. 234. 245.R:BASERelational0.570.00+0.04238. 249. 255.Apache DorisRelational0.57+0.08+0.07239.239. 234.FrontBaseRelational0.55-0.01-0.08240. 245. 272.Alibaba Cloud Log ServiceSearch engine0.55+0.06+0.18241. 236. 235.ObjectDBObject oriented0.54-0.03-0.09242. 250. 279.ITTIATime Series, Multi-model Time Series DBMSRelational DBMS0.54+0.06+0.20243. 241. 246.Splice MachineRelational0.540.00+0.00244. 233. 284.YDBMulti-model Document storeRelational DBMS0.51-0.06+0.19245. 252. 251.Percona Server for MongoDBDocument0.51+0.050.00246. 242. 242.HarperDocument0.51-0.02-0.05247. 244. 266.DragonflyKey-value0.49-0.01+0.08248. 247. 254.atotiObject oriented0.49+0.01-0.01249. 246. 273.Apache PinotRelational0.49+0.00+0.11250. 253. 241.NexusDBRelational0.48+0.02-0.08251. 248. 223.WebSphere eXtreme ScaleKey-value0.480.00-0.22252. 251. 249.MapDBKey-value0.460.00-0.07253. 243. 244.SQL.JSRelational0.46-0.05-0.08254. 255. 257.VistaDBRelational0.45+0.01-0.03255. 263. 263.KingbaseRelational, Multi-model 0.44+0.04+0.02256.256. 278.Apache SedonaSpatial0.43+0.00+0.08257. 259. 243.GemStone/SObject oriented0.420.00-0.14258. 260. 250.ScalarisKey-value0.410.00-0.12259. 262. 252.ScaleArcRelational0.41+0.01-0.10260. 257. 274.SearchBloxSearch engine0.41-0.01+0.04261. 258. 285.FlureeGraph0.40-0.02+0.09262. 254. 253.KineticaRelational, Multi-model 0.40-0.04-0.11263. 261. 288.Deep LakeVector0.40-0.01+0.10264. 266. 259.AlaSQLMulti-model Document storeRelational DBMS0.40+0.04-0.06265. 269.265.LokiJSDocument0.39+0.05-0.02266. 264. 261.OpenInsightMultivalue0.39+0.00-0.03267. 271.267.CnosDBTime Series0.38+0.06-0.02268. 272. 378.RaptorDBDocument0.38+0.06+0.38269. 265. 260.RasdamanMultivalue0.38+0.01-0.05270. 273. 269.SequoiadbMulti-model Document storeRelational DBMS0.37+0.06-0.02271. 274. 286.Cloudflare Workers KVKey-value0.37+0.06+0.07272. 268. 247.HibariKey-value0.35+0.01-0.18273. 267. 287.MarqoSearch engine0.34-0.01+0.04274. 280. 248.PerstObject oriented0.34+0.04-0.19275.275. 262.Kyligence EnterpriseRelational0.34+0.03-0.08276. 270. 283.Comdb2Relational0.340.00+0.01277.277. 271.ModeShapeContent0.31+0.01-0.08278. 294. 310.FeatureBaseRelational0.31+0.07+0.10279. 276. 258.Postgres-XLRelational, Multi-model 0.300.00-0.16280. 289. 294.DatabendRelational0.30+0.04+0.03281. 279. 210.Northgate RealityMultivalue0.290.00-0.49282. 281. 289.ElassandraWide column, Multi-model Wide column storeSearch engine0.290.00-0.01283. 282. 281.StarcounterObject oriented0.280.00-0.05284. 278. 268.StrabonRDF0.28-0.01-0.12285. 283. 275.Project VoldemortKey-value0.28+0.00-0.08286. 301. 334.YTsaurusMulti-model Document storeKey-value store0.28+0.06+0.15287. 292. 296.OpenQMMultivalue0.28+0.03+0.02288. 293. 297.BrytlytRelational0.28+0.03+0.02289. 295. 290.LeanXcaleMulti-model Key-value storeRelational DBMS0.28+0.040.00290. 284. 276.InfiniteGraphGraph0.28+0.01-0.08291. 299. 292.Fujitsu Enterprise PostgresRelational, Multi-model 0.27+0.04-0.01292. 297. 314.RDFoxMulti-model Graph DBMSRDF storeRelational DBMS0.27+0.04+0.07293. 296. 309.Alibaba Cloud TSDBTime Series0.26+0.03+0.06294. 288. 303.Manticore SearchSearch engine, Multi-model Search engineTime Series DBMS0.25-0.01+0.02295. 290. 264.Raima Database ManagerMulti-model Relational DBMSTime Series DBMS0.250.00-0.17296. 298. 305.AnzoGraph DBMulti-model Graph DBMSRDF store0.25+0.02+0.02297. 285. 322.Valentina ServerRelational0.24-0.02+0.07298. 302. 299.ImmudbKey-value, Multi-model Key-value storeRelational DBMS0.24+0.02+0.00299. 286. 270.ValdVector0.24-0.02-0.15300. 310. 293.AxibaseTime Series0.24+0.05-0.03301. 291. 239.4storeRDF0.23-0.02-0.35302. 287. 315.MyScaleMulti-model Relational DBMSVector DBMS0.23-0.03+0.04303.303. 316.Faircom DBMulti-model Key-value storeRelational DBMS0.23+0.01+0.04304. 300. 313.RedStoreRDF0.220.00+0.03305. 304. 300.Actian FastObjectsObject oriented0.210.00-0.02306. 313. 308.YottaDBKey-value, Multi-model Key-value storeRelational DBMS0.21+0.030.00307.307. 256.TajoRelational0.21+0.01-0.27308. 306. 337.XTDBDocument0.200.00+0.08309.309. 291.Mimer SQLRelational0.20+0.00-0.08310. 317. 341.QuasardbTime Series0.20+0.04+0.10311. 308. 277.RedlandRDF0.200.00-0.16312.312. 282.PipelineDBRelational0.19+0.01-0.13313. 331. 306.PieCloudDBRelational0.19+0.06-0.03314. 311. 326.IBM Db2 Event StoreMulti-model Event StoreTime Series DBMS0.19+0.00+0.03315. 305. 378.chDBRelational, Multi-model Relational DBMSTime Series DBMS0.19-0.02+0.19316. 340. 298.LovefieldRelational0.18+0.08-0.07317. 314. 321.TransLatticeRelational0.180.00+0.00318. 335. 320.TerminusDBGraph, Multi-model 0.17+0.050.00319. 320. 330.Apache HugeGraphGraph0.17+0.02+0.04320. 318. 344.Warp 10Time Series0.17+0.01+0.10321. 315.FalkorDBGraph0.17+0.00322. 341. 343.BangdbMulti-model Document storeGraph DBMSTime Series DBMSSpatial DBMS0.17+0.07+0.08323. 337. 336.UltipaGraph0.16+0.04+0.04324. 316. 304.Tibco ComputeDBRelational0.160.00-0.07325. 332. 317.AgensGraphMulti-model Graph DBMSRelational DBMS0.16+0.03-0.03326. 319. 324.Riak TSTime Series0.160.000.00327. 322. 323.NEventStoreEvent0.150.00-0.01328. 323. 329.XtremeDataRelational0.150.00+0.01329. 325. 332.BigObjectRelational0.140.00+0.01330. 327. 328.EsgynDBRelational0.140.000.00331. 321. 338.BadgerKey-value0.14-0.01+0.02332. 329. 307.ElevateDBRelational0.140.00-0.07333.333. 352.GreptimeDBMulti-model Time Series, ObservabilityTime Series DBMS0.14+0.01+0.09334. 345. 378.ArcadeDBMulti-model Document storeGraph DBMSKey-value storeTime Series DBMS0.14+0.05+0.14335. 330. 319.HyperGraphDBGraph0.130.00-0.05336. 328. 335.Actian PSQLRelational0.13-0.01+0.00337. 336. 312.SpeedbKey-value0.13+0.01-0.07338. 326. 339.MulgaraRDF0.13-0.01+0.01339. 344. 318.EJDBDocument0.13+0.04-0.06340. 324. 301.FlockDBGraph0.12-0.02-0.11341. 338. 327.GraphBaseGraph0.12+0.01-0.03342. 339. 295.JadeObject oriented0.09-0.01-0.17343. 342. 302.Kyoto TycoonKey-value0.090.00-0.13344. 343. 341.SparkseeGraph0.090.00-0.01345. 334. 311.Alibaba Cloud Table StoreWide column0.09-0.04-0.11346. 380. 355.TransbaseRelational0.09+0.09+0.05347. 355. 356.Machbase NeoTime Series0.08+0.03+0.03348. 380. 378.ReductStoreTime Series0.07+0.07+0.07349. 346. 346.SiteWhereTime Series0.070.00+0.01350. 347. 348.ScaleOut StateServerKey-value0.070.00+0.01351. 348. 345.ExorbyteSearch engine0.070.00+0.01352. 363. 358.OpenMLDBTime Series, Multi-model Time Series DBMSRelational DBMS0.07+0.05+0.03353. 351. 347.BluefloodTime Series0.07+0.01+0.01354. 357. 349.TinkerGraphGraph0.07+0.03+0.01355. 350. 360.TigrisMulti-model Document storeKey-value storeSearch engineTime Series DBMS0.060.00+0.02356. 353. 350.EllipticsKey-value0.06+0.01+0.01357. 380. 378.JaguarDBMulti-model Key-value storeVector DBMS0.06+0.06+0.06358. 349. 333.CubicWebRDF0.060.00-0.07359. 375. 340.AntDBRelational0.06+0.05-0.05360. 353. 351.SenseiDBDocument0.050.000.00361. 352. 325.Tokyo TyrantKey-value0.05-0.01-0.11362. 380. 378.GalaxybaseGraph0.04+0.04+0.04363. 380. 354.openGeminiTime Series0.04+0.04-0.01364. 359. 363.JethroDataRelational0.04+0.01+0.01364. 356. 361.NosDBDocument0.040.000.00366. 380. 378.TerarkDBKey-value0.04+0.04+0.04367. 358. 362.WakandaDBObject oriented0.030.000.00368. 359. 363.STSdbKey-value0.030.000.00369. 361. 280.HeroicTime Series0.020.00-0.31370. 367. 378.Sadas EngineRelational0.02+0.00+0.02371. 362.371.Hawkular MetricsTime Series0.020.00+0.01372. 364. 365.DydraRDF0.020.000.00373. 365. 366.SmallSQLRelational0.02+0.000.00374. 366. 378.KuzuGraph0.02+0.00+0.02375. 380. 378.SiriDBTime Series0.02+0.02+0.02376.376. 368.OushuDBRelational0.02+0.01+0.00377. 368. 367.SparkleDBRDF0.020.000.00378. 369. 369.AcebaseDocument0.01+0.000.00379. 370. 370.Resin CacheKey-value0.010.000.00380.380. 378.VelocityDBMulti-model Graph DBMSObject oriented DBMS0.01+0.01+0.01381. 371. 357.gStoreMulti-model Graph DBMSRDF store0.01+0.00-0.03382. 380.SvectorDBVector0.01+0.01+0.01383. 372. 372.SWC-DBWide column, Multi-model Wide column storeTime Series DBMS0.01+0.000.00384. 374. 374.EloqueraObject oriented0.010.000.00385. 373. 373.Faircom EDGEMulti-model Key-value storeRelational DBMS0.010.000.00386. 377. 375.SiaqodbObject oriented0.00+0.00+0.00387. 378. 376.LedisDBKey-value0.00+0.00+0.00388. 379. 377.SwayDBKey-value0.00+0.00+0.00389. 380. 378.ActorDBRelational0.00±0.00±0.00389. 380. 378.BergDBKey-value0.00±0.00±0.00389. 380. 378.BrightstarDBRDF0.00±0.00±0.00389. 380. 378.Cachelot.ioKey-value0.00±0.00±0.00389. 380. 378.CortexDBMulti-model Document storeKey-value store0.00±0.00±0.00389. 380. 378.CovenantSQLRelational0.00±0.00±0.00389. 380. 378.DaggerDBRelational0.00±0.00±0.00389. 380. 359.DataFSObject oriented, Multi-model Object oriented DBMSGraph DBMS0.00±0.00-0.04389. 380. 378.Edge IntelligenceRelational0.00±0.00±0.00389. 380. 378.EdgelessDBRelational0.00±0.00±0.00389. 380. 378.H2GISSpatial, Multi-model Spatial DBMSRelational DBMS0.00±0.00±0.00389. 380. 378.HeliumKey-value0.00±0.00±0.00389. 380. 378.HGraphDBGraph0.00±0.00±0.00389. 380. 378.HyperLevelDBKey-value0.00±0.00±0.00389. 380. 378.iBoxDBDocument0.00±0.00±0.00389. 380. 378.IndicaSearch engine0.00±0.00±0.00389. 380. 378.InfinityDBKey-value0.00±0.00±0.00389. 380. 378.JasDBDocument0.00±0.00±0.00389. 380. 378.K-DBRelational0.00±0.00±0.00389. 380. 353.LinterRelational, Multi-model Relational DBMSSpatial DBMS0.00±0.00-0.05389. 380. 378.NewtsTime Series0.00±0.00±0.00389. 380. 378.NSDbTime Series0.00±0.00±0.00389. 380. 378.OpenTenBaseRelational0.00±0.00±0.00389. 380. 378.OrigoDBMulti-model Document storeObject oriented DBMS0.00±0.00±0.00389. 380. 378.RizhiyiSearch engine, Multi-model Search engineTime Series DBMS0.00±0.00±0.00389. 380. 378.searchxmlMulti-model Native XML DBMSSearch engine0.00±0.00±0.00389. 380. 331.SkytableKey-value0.00±0.00-0.14389. 380. 378.SpaceTimeSpatial, Multi-model Spatial DBMSRelational DBMS0.00±0.00±0.00389. 380. 378.TkrzwKey-value0.00±0.00±0.00389. 380. 378.TomP2PKey-value0.00±0.00±0.00389. 380. 378.Transwarp ArgoDBRelational, Multi-model Relational DBMSSearch engine0.00±0.00±0.00389. 380. 378.Transwarp HippoVector0.00±0.00±0.00389. 380. 378.Transwarp KunDBRelational0.00±0.00±0.00389. 380. 378.Transwarp StellarDBGraph0.00±0.00±0.00389. 380. 378.UpscaledbKey-value0.00±0.00±0.00389. 380. 378.WhiteDBDocument0.00±0.00±0.00Share this page In the dynamic landscape of web development, the importance of a robust database cannot be overstated. As businesses, whether small or large, continue their digital transformation journey, efficient data management becomes paramount. The database management system software emerges as a pivotal solution, serving as the backbone for organizing and safeguarding sensitive information generated at every touchpoint. Diverse Database Ecosystem With the evolution of technology, traditional SQL databases face competition from various fronts like NoSQL, NewSQL, and Cloud databases. This diversification is fueled by the rise of Microservices, Cloud computing, and the handling of Semi-Structured Data and Big Data with low latency requirements. This shift presents both opportunities and challenges for data integration specialists, including those working with tools like SSIS, as they need to adapt to a more diverse and dynamic data landscape. Strategic Insights Through Data Visualization As companies delve into the expansive realm of web applications, extracting meaningful insights from stored data becomes a strategic imperative. Through the adept use of data visualization, businesses can transform raw data into actionable intelligence, driving informed decision-making. Web App Development Trends In web development, particularly Java web application development, choosing the right database is pivotal. Whether it's for custom web app development services or aligning with the best web development languages, selecting a suitable database is critical. Market Trends and Projections Reports from reputable sources underscore the significance of databases, especially in Customer Relationship Management (CRM) applications. According to business.com, a staggering 91% of companies with more than 11 employees utilize CRM software. The mobile CRM market, as projected by Future Market Insights, is anticipated to witness a substantial Compound Annual Growth Rate (CAGR) of around 13% from 2019 to 2029. In this blog, we will navigate through the ever-expanding databases list, offering insights for businesses seeking to optimize their web development endeavors. Stay tuned for an in-depth exploration of the databases shaping the future of web applications. Discover seamless integration with ValueCoders' services for unparalleled web app success. Hiring proficient developers from top web application development company india becomes paramount for those embarking on this transformative journey. Our blog will not only shed light on powerful web development tools but also guide you in selecting the right database for your needs. Whether you're exploring front-end development services or delving into the intricacies of backend development services, our comprehensive guide will be your compass in the complex world of web application databases. Choosing the Best Database for Web Applications in 2025: A Comparative Overview In web application development, selecting a suitable database is critical. In 2022, two standout options are Oracle and MySQL, each bringing its unique strengths. 1. Oracle Oracle is the most widely used commercial relational database management system (RDBMS), incorporating assembly languages like C, C++, and Java. Advantages: Performance: Processes data faster and is known for efficient space utilization. Innovative Features: Version 21c introduces new features, including enhanced JSON capabilities in SQL. 2. MySQL MySQL remains a stalwart in the database landscape, particularly favored in web application development for its emphasis on stability, robustness, and maturity. Key Features: Stability: Prioritizes stability, ensuring reliable performance. Maturity: Known for its mature and well-established nature. Latest Version: MySQL 8.0 introduces improved recovery options, enhancing data security. 3. MS SQL Server MS SQL Server, a flagship product, is backed by Microsoft's robust toolset support, ensuring a seamless experience for users both on-premise and in the cloud. Notably, MS SQL Server exhibits compatibility with diverse operating systems, including Linux and Windows, offering flexibility in deployment. Setting it apart, MS SQL is a versatile multi-model database supporting various data models, including Structured Data (SQL), Semi-Structured Data (JSON), and Spatial Data. Key Features: Structured and Semi-Structured Support: MS SQL's strength lies in its ability to handle both traditional Structured Data, managed through SQL, and Semi-Structured Data like JSON. Spatial Data Management: A distinctive feature is its support for Spatial Data, catering to applications where geographic or spatial information is integral. Evolution Over the Years: While not claiming the same level of innovation as some newer databases, MS SQL Server has consistently evolved. Substantial improvements and overhauls showcase a commitment to staying relevant in the dynamic database landscape. 4. PostgreSQL PostgreSQL, initially known as POSTGRES, holds a rich history. Notably, its creator, Michael, received the prestigious Turing Award for his significant contributions to the development of PostgreSQL. Key Attributes: Written in C: PostgreSQL is crafted in the C programming language, ensuring a robust and efficient foundation for managing extensive datasets. Versatile Industry Applications: Tailored for businesses dealing with large volumes of data, PostgreSQL finds applications in diverse sectors, including gaming apps, database automation tools, and domain registrations. Contributions Recognized: Michael's recognition with the Turing Award underscores the substantial impact of his contributions to PostgreSQL. This accolade highlights the database's significance in the realm of data management. Wide-Spanning Usage: Gaming Apps: PostgreSQL's usage extends to the dynamic world of gaming applications, emphasizing its suitability for real-time data processing. Database Automation Tools: Businesses leverage PostgreSQL for streamlined database management through automation tools, enhancing efficiency and reliability. Domain Registrations: In critical domains like domain registrations, where data accuracy and integrity are paramount, PostgreSQL emerges as a trusted database solution. 5. MongoDB In the landscape of popular databases for 2022, MongoDB stands out as a trailblazer in the NoSQL category. Released in 2009, MongoDB represents the inaugural Document Database management software. MongoDB emerged in response to challenges encountered in loading and accessing data into Relational Database Management Systems (RDBMS) using object-oriented programming languages. Key Considerations: Document Data Handling: MongoDB's primary strength lies in its adept handling of Document Data, offering a more streamlined approach compared to traditional RDBMS. Eliminating Mapping Challenges: The database mitigates complexities associated with loading data into RDBMS, particularly the need for additional application-level mapping. Use Case Significance: Industry Adoption: MongoDB's significance extends beyond its inception, gaining popularity across diverse industries and applications. NoSQL Dominance: As the first Document Database, MongoDB plays a pivotal role in shaping the dominance of NoSQL databases in modern data management. 6. IBM DB2 IBM's DB2, available as DB2 LUW for Windows, Linux, and Unix, showcases its adaptability across diverse operating systems. The most recent iteration, DB2 11.5, not only reflects IBM's commitment to innovation but notably enhances query execution speed. Key Features: Relational Model Support: Originally known for its robust support for the relational model, DB2 has evolved to encompass a more extensive range of functionalities. Object-Relational Capabilities: Keeping pace with industry trends, DB2 now seamlessly integrates object-relational features, offering a more comprehensive data management approach. Non-Relational Forms Inclusion: In response to the evolving needs of modern databases, DB2 extends its support to non-relational forms such as JSON and XML. Adapting to Modern Requirements: Mobile App Integration: The significance of databases for mobile apps cannot be understated, and DB2 aligns with this demand by not only supporting the relational model but also catering to newer forms and structures. Continuous Growth: The expansion of DB2's capabilities over recent years positions it as a dynamic and evolving player in the database landscape. Partner with ValueCoders for Java web application development expertise. 7. Redis Redis stands out as a widely embraced open-source database project, garnering significant attention and usage within the developer community. According to Stack Overflow's Annual Developer Survey, Redis holds the prestigious title of being the Most Loved Database platform, a testament to its popularity and positive reception. Versatile Capabilities: Distributed, In-Memory Database: Redis positions itself as a distributed, in-memory key-value database, emphasizing speed and efficiency in data retrieval and storage. Beyond a Database: Redis extends its utility beyond traditional database functions. It can seamlessly serve as a distributed cache, optimizing data access times, and also function as a message broker. Durability as an Option: Redis offers flexibility as an optional feature, ensuring data persistence according to specific use case requirements. Community Recognition: Annual Developer Survey: Redis's acknowledgment as the Most Loved Database in Stack Overflow's survey reflects not only its technical prowess but also its positive impact on the developer experience. Widespread Adoption: The versatility of Redis, spanning database, cache, and message broker functionalities, contributes to its widespread adoption across various development scenarios. 8. Elasticsearch Elasticsearch, an open-core full-text search engine, builds on the robust foundation of Lucene. Shay Banon introduced this powerful tool in 2010, and it has since become a cornerstone in the realm of full-text search engines. Key Features: Distributed Capability: Elasticsearch boasts a distributed, multi-tenant architecture, enhancing its scalability and enabling efficient handling of diverse search requirements. REST API Integration: With a REST API at its core, Elasticsearch ensures seamless integration into various applications and systems, providing a standardized and accessible interface. Scalability and Sharing: Horizontal Scaling: Elasticsearch offers horizontal scaling through automatic sharing, a critical feature for managing large volumes of data efficiently. Structured and Schema-Less Data Support: The platform's adaptability shines through its support for structured and schema-less data in JSON format. This flexibility is particularly advantageous for analyzing, Logging, or Monitoring data. Versatile Use Cases: Elasticsearch's support for JSON and its distributed nature makes it especially well-suited for the analysis of Logging or Monitoring data. 9. Cassandra Cassandra, developed in 2008, stands as an open core, distributed, wide column store, and is a prevalent choice for applications requiring robust database solutions. Key Characteristics: Distributed Architecture: Cassandra's distributed nature is a key strength, allowing it to efficiently handle large datasets across various nodes. Wide Column Store: Operating as a wide column store, Cassandra is optimized for handling vast amounts of data with a flexible and scalable structure. Scalability at its Core: Cassandra shines as a highly scalable database management software, making it a go-to solution for industries dealing with massive datasets and demanding workloads. Fault Tolerance and Reliability: Decentralized Database: A standout feature is its decentralized (Leaderless) database architecture, promoting fault tolerance and ensuring robustness without single points of failure. Automatic Replication: Cassandra's automatic replication further contributes to its fault-tolerant nature, enhancing data redundancy and availability. Multi-Data Center Replication: With multi-data center replication, Cassandra extends its reliability across geographically dispersed locations, catering to diverse business needs. Operational Diversity: Cassandra offers a range of operations and infrastructure, providing flexibility for varied use cases within different industries. Comparative Insight: Cassandra vs. HBase: Cassandra and HBase, while sharing some similarities, cater to distinct use cases based on their inherent characteristics and strengths. 10. MariaDB MariaDB is a robust Relational Database Management System (RDBMS), distinguished by its compatibility with the MySQL Protocol and Clients. One of MariaDB's notable strengths is its seamless compatibility with the MySQL server. It allows for a straightforward replacement, requiring no code changes and ensuring a smooth transition for users. Innovative Features: Columnar Storage: MariaDB introduces columnar storage, enhancing data retrieval efficiency and optimizing performance for specific types of queries. Massively Parallel Distributed Data Architecture: Embracing a massively parallel distributed data architecture, MariaDB ensures scalability and high-performance capabilities. Community-Driven Dynamics: MariaDB distinguishes itself by being more community-driven compared to MySQL. This emphasis on community involvement fosters a collaborative environment for continuous improvement and innovation. Seamless Transition from MySQL: MariaDB's compatibility with MySQL extends to code-level harmony, facilitating a transition without altering existing codebases. 11. OrientDB OrientDB stands out as an open-source NoSQL multi-model database program, offering businesses a versatile solution that obviates the need for multiple systems to handle diverse data types. Key Features: Graph Database Capabilities: One of OrientDB's notable strengths is its ability to serve as a graph database management software, providing businesses with the power of graph-based data modeling. Eliminating System Proliferation: By supporting various database models, including graph, document, key-value, and object-oriented, OrientDB eliminates the need for businesses to build separate systems for handling different data types. Versatility in Management Solutions: Multi-Model Support: OrientDB goes beyond traditional databases by supporting multiple models, optimizing performance and security while offering scalability. Graph, Document, Key-Value, and Object-Oriented Models: The inclusion of diverse models ensures that businesses can leverage the most suitable approach for their specific data management needs. Performance and Security Focus: Enhanced Performance: OrientDB is designed to enhance performance across different data models, catering to the varied requirements of businesses in today's dynamic landscape. Security Measures: Alongside performance optimization, OrientDB prioritizes security, providing a robust foundation for safeguarding sensitive data. Scalability Advantage: The multi-model support not only improves performance and security but also allows for scalability, ensuring that businesses can grow their databases seamlessly. 12. SQLite SQLite stands as a powerful open-source SQL database, offering users an integrated relational database management system since its creation in the year 2000. Notable for its simplicity, SQLite requires no complex configuration, server, or installation. This ease of deployment makes it an attractive choice for various applications. Key Attributes: No Configuration Required: One of SQLite's distinctive features is its minimalistic setup – users can start leveraging its capabilities without the need for intricate configurations. Server-Free Operation: Unlike traditional databases, SQLite doesn't rely on a server, contributing to its lightweight nature and making it an efficient option for diverse use cases. Functional Richness Despite its simplicity, SQLite doesn't compromise on functionality. It encompasses a range of commonly used database management system software features, making it a versatile choice for various applications. Mobile Web Development Compatibility: SQLite's lightweight and server-free nature make it particularly well-suited for mobile web development, including platforms like React Native. Its efficient handling of data aligns with the demands of mobile applications. ValueCoders crafts solutions in the best web development languages for lasting success. 13. DynamoDB DynamoDB takes center stage as Amazon's nonrelational best database, specifically designed to cater to the dynamic needs of modern applications. Positioned as a serverless database, DynamoDB seamlessly scales up and down in response to demand, ensuring optimal performance for mobile apps while relieving users of infrastructure management burdens. Key Features: Automated Scaling: DynamoDB's automatic scaling is a standout feature, dynamically adjusting resources to accommodate varying workloads. This ensures efficient utilization of resources while maintaining responsiveness for mobile applications. Data Backups: The database offers automatic data backups, providing an added layer of data protection and resilience against unexpected events. Security and Performance: Built-In Security: DynamoDB prioritizes security with built-in measures, safeguarding sensitive data and ensuring compliance with industry standards. In-Memory Caching: The inclusion of in-memory caching enhances performance, reducing latency and optimizing response times for seamless user experiences. Consistent Latency: DynamoDB's commitment to consistent latency ensures that applications deliver predictable and reliable performance, critical for mobile app success. 14. Neo4j Neo4j stands as a pioneering open-source NoSQL graph database, leveraging Java for its dynamic architecture. Launched in 2007, it revolutionizes data representation and querying through a graph-based paradigm. Key Characteristics: Graph-Based Representation: Neo4j distinguishes itself by adopting a graph-based approach, where data is stored and processed as interconnected graphs rather than traditional tables. This unique structure enables a more intuitive representation of relationships within the data. Cypher Query Language: Neo4j employs Cypher, a specialized query language designed for expressing relationship queries efficiently and expressively. This language simplifies the exploration of complex relationships within the graph. Graph Storage Advantages: Relationship-Centric Model: Neo4j's innovative design focuses on relationships as a core aspect of data, allowing for the creation of efficient and interconnected structures. This approach is particularly advantageous for scenarios where relationships play a crucial role. Dynamic Relationship System: Neo4j's dynamic relationship system offers agility, allowing the creation and utilization of new relationships as needed. This flexibility, often referred to as “shortcutting,” accelerates domain data retrieval and manipulation. Efficiency through Relationships: Neo4j's relationship system is optimized for speed, facilitating the swift creation and utilization of relationships. This efficiency is paramount for scenarios where real-time relationship exploration is critical. 15. Firebird SQL Firebird stands out as a free SQL relational database management system, offering robust functionality without the constraints of licensing costs. It seamlessly operates on Mac OS X, Linux, Microsoft Windows, and various Unix platforms, making it a versatile choice for diverse environments. Key Features: Cross-Platform Compatibility: Firebird's versatility is underscored by its ability to operate seamlessly across Mac OS X, Linux, Microsoft Windows, and various Unix platforms. This cross-platform capability ensures accessibility for a broad user base. SQL Relational Database Management: Positioned as a SQL relational database management system, Firebird excels in managing structured data through relational models, catering to the requirements of web applications. Financial Flexibility: Free and Open-Source Model: Firebird embraces an open-source model, eliminating licensing costs and making it an attractive option for businesses and developers seeking a cost-effective database solution. Varied Financing Choices: Firebird goes beyond traditional funding models by offering a range of financing choices, from memberships to sponsorship commitments. This diversity allows users to contribute and support the development of Firebird based on their preferences. Multi-Platform Enhancement: Firebird continually enhances its multi-platform Relational Database Management System (RDBMS), ensuring that users benefit from the latest improvements and optimizations. Unlocking the Power of Databases: Choosing the Right Path In database software tools, the ease of use often hinges on the user's familiarity and the purpose at hand. Understanding the specific needs is crucial for determining the easiest database to navigate. Here's a breakdown tailored to different scenarios: Microsoft SQL Server: For novice students embarking on database essentials, Microsoft SQL Server is often the starting point at universities. Its structured learning curve prepares students to transition seamlessly to other systems. SQL Databases: Learning the fundamentals of popular SQL databases is key. Starting with SQL statements rather than relying on a graphical user interface (GUI) approach provides a solid foundation applicable across various systems. MySQL and Oracle: Once the basics are mastered, progression to more advanced systems like MySQL and Oracle becomes natural. This sequential learning approach ensures a smooth transition and comprehensive understanding. Key Concepts Understanding: Regardless of the chosen database, grasping key concepts is paramount. A solid understanding of SQL basics allows for seamless application to different database management software, irrespective of the specific GUI or IDE used. SQL Statements Mastery: Composing SQL statements serves as a practical and universal learning method. This hands-on approach facilitates adaptability to various databases, reinforcing the core concepts learned. Choosing the easiest database depends on the learning context and the user's objectives. Starting with foundational systems like Microsoft SQL Server and progressively exploring MySQL and Oracle ensures structured learning journey. However, the real ease comes from a thorough understanding of SQL basics, allowing users to navigate and apply their knowledge across a diverse range of database systems with confidence. Optimal Database Choices for Python Development Python, renowned for its robust database programming features, seamlessly integrates with various databases, offering flexibility for diverse applications. Among the plethora of options, one standout choice for Python development is: SQLite: Simplifying Python Database Interaction Ease of Integration: SQLite emerges as a straightforward and popular SQL database for Python applications. Notably, it requires no external installations or additional Python SQL database modules. Inherent Python Support: Within the default Python installation, the SQLite3 library is readily available. This built-in library streamlines the connection and interaction with an SQLite database, eliminating the need for additional configurations. Python's Database Programming Support: Versatility: Python accommodates a range of databases, including SQLite, Oracle, MySQL, and PostgreSQL. DDL, DML, and Data Query: Python supports essential database operations, encompassing Data Definition Language (DDL), Data Manipulation Language (DML), and Data Query Statements. Python DB-API Standard: Adhering to the Python DB-API standard, most Python database interfaces ensure consistency and compatibility. SQLite's Distinct Advantages: No External Dependencies: SQLite's inherent advantages lie in its simplicity and lack of external dependencies, making it an ideal choice for seamless Python integration. Built-in Library: With SQLite3 as part of the default Python installation, skilled developers can initiate interactions with an SQLite database without the hassle of external setups. Leverage ValueCoders' powerful web development tools for unmatched web app excellence. Concluding Thoughts A robust database for mobile apps serves as the bedrock for thriving software applications, playing a pivotal role in organizational success. For unparalleled assistance in SQL database selection and app development, ValueCoders stands as your reliable partner. As one of the top web development companies in India, we specialize in crafting scalable applications that deliver impressive results across desktop and mobile devices. Our expertise extends to custom web app development services, ensuring scalability and optimal application performance. Whether it's Java web application development, hiring web application developers, or utilizing the best web development languages, our team excels in seamless integration to meet your specific needs. ValueCoders, among the top web app development company india, empowers your vision with powerful web development tools and top-notch website development services. Our comprehensive solutions cater to every aspect of your web application journey from front-end development services to robust backend development. Your Journey to Excellence Begins with ValueCoders. Connect with Us for Unparalleled Web Development Services. Software Development Expert Turning Ideas into Innovative Digital Experiences – 15+ Years in Software Development. I specialize in creating innovative and user-focused digital solutions. A rich background of over one and a half decades enables me to deliver high-quality projects that drive business success. Here's how I can help you: Extensive Experience: Over 15 years in software development, skilled in React, Angular, Swift, Kotlin, and responsive design principles. Leadership in Development: Proven track record in leading teams to successfully deliver projects on time and within scope, using Agile and Scrum methodologies. Innovation and Problem-Solving: Dedicated to creating engaging digital experiences that meet business needs and enhance user satisfaction. Collaboration and Communication: Strong team player, collaborating closely with designers, product managers, and stakeholders to integrate solutions seamlessly. Continuous Learning: Committed to staying ahead of industry trends and technological advancements. Let's connect and explore how we can turn your digital ideas into reality! Explore More of My Work: Expertify | Hackernoon | Habr Stay Connected: linkedin #MobileDevelopment #WebDevelopment #Agile #Innovation #UserExperience #TechLeadership #ContinuousLearning DBMS > OraclePlease select another system to compare it with Oracle.Our visitors often compare Oracle with PostgreSQL, Microsoft SQL Server and Oracle Coherence.Share this page , the free encyclopedia that anyone can edit. 107,583 active editors 7,025,866 articles in English Taylor Swift on the Red Tour in 2013 Red is the fourth studio album by the American singer-songwriter Taylor Swift (pictured). It was released in October 2012 by Big Machine Records. Swift designated Red as a breakup album and her last to be promoted as country music. The album incorporates styles of pop, rock, folk and country, and is composed of acoustic instruments, electronic synths and drum machines. Initial reviews mostly praised Swift's songwriting for its emotional exploration and engagement, but critics deemed the production inconsistent and questioned her identity as a country artist. Swift supported Red with the Red Tour (2013–2014). Red topped the charts and received multi-platinum certifications in Australia, Canada, New Zealand and the United Kingdom. In the United States it spent seven weeks atop the Billboard 200. Red was nominated for Album of the Year at the 2013 Country Music Association Awards, and Album of the Year and Best Country Album at the 2014 Grammy Awards. (Full article...) Recently featured: Sir William Gordon-Cumming, 4th Baronet Great Wilbraham (causewayed enclosure) Henry de Hinuber Archive By email More featured articles About Trichogenes claviger ... that Trichogenes claviger (examples pictured) can climb nets using outgrowths from its head? ... that Philippe Housiaux took a law examination instead of competing in the 1972 Summer Olympics? ... that helical strakes can be used to reduce structural fatigue on structures such as chimneys and pipelines? ... that "Whistle" was composed because of a miscommunication between a film director and the songwriters? ... that Fletcher Ransom wrote and illustrated a book of satirical cartoons about Theodore Roosevelt titled My Policies in Jungleland? ... that art students designed the stained-glass windows of the Chapel at George Fox University? ... that Don Summers signed with an NFL team despite having gone more than 400 days without catching a football? ... that AHOF, a K-pop group formed on a reality show, includes two members who were not on the winning team? ... that Worrall Reed Carter wrote about beans, bullets and black oil? Archive Start a new article Nominate an article Muhammadu Buhari Former president of Nigeria Muhammadu Buhari (pictured) dies at the age of 82. Clashes between Druze militias and the Syrian Armed Forces result in hundreds of deaths. The International Criminal Court issues arrest warrants for Taliban leaders Hibatullah Akhundzada and Abdul Hakim Haqqani over their alleged persecution of women in Afghanistan. Flooding in Central Texas, United States, leaves at least 140 people dead. Ongoing: Gaza war Russian invasion of Ukraine timeline Sudanese civil war timeline Recent deaths: Andrea Gibson Raymond Guiot Felix Baumgartner Fauja Singh Bradley John Murdoch Frank Barrie Nominate an article July 21: Belgian National Day (1831), Marine Day in Japan (2025) Sirimavo Bandaranaike 905 – Louis III, Holy Roman Emperor, was captured and blinded during his attempt to restore Carolingian power over Italy by King Berengar I. 1877 – Much of central Pittsburgh, Pennsylvania, was burned and looted during the Great Railroad Strike of 1877. 1960 – Sirimavo Bandaranaike (pictured) was sworn in as prime minister of Ceylon, becoming the world's first democratically elected female head of government. 1990 – The Taiwanese military ordered the deportation of dozens of illegal immigrants from mainland China in sealed boat holds, causing 25 deaths due to suffocation. 2007 – Harry Potter and the Deathly Hallows, the final book in the popular Harry Potter series by J. K. Rowling, was released and sold 15 million copies in its first 24 hours, making it the fastest-selling book in history. Nelson Dewey (d. 1889)Fiammetta Wilson (d. 1920)Johnny Peirson (b. 1925)Erling Haaland (b. 2000) More anniversaries: July 20 July 21 July 22 Archive By email List of days of the year About One Direction performing in 2015 The English–Irish boy band One Direction released five studio albums, seventeen singles, and numerous music videos between 2011 and 2015, becoming one of the most commercially successful groups of the 2010s. Formed during the seventh series of The X Factor in 2010 and signed to Simon Cowell's Syco Records, One Direction quickly rose to international fame. Their debut single, "What Makes You Beautiful", was a global hit, followed by chart-topping albums Up All Night, Take Me Home, and Midnight Memories. The latter made them the first band ever to have their first three albums debut at number one on the Billboard 200. They extended this streak with their fourth album, Four, and capped their discography with Made in the A.M. in 2015 before going on an indefinite hiatus. (Full list...) Recently featured: Kerivoulines Accolades received by Inception 1956 Summer Olympics medal table Archive More featured lists The southern scrub robin (Drymodes brunneopygia) is a species of bird in the family Petroicidae, the Australasian robins. It is endemic to Australia, where it occurs in mallee and heathland in the semi-arid southern parts of the continent. Its range includes several disjoint regions from central New South Wales through western Victoria and southern South Australia, through to the southwestern area of Western Australia. It is a relatively dull and large robin, with adults around 19 to 20 centimetres (7.5 to 7.9 in) in length, of which around a third are the tail feathers. Most of the plumage is grey. It breeds between July and December and has a nest on the ground, built of twigs and lined with twigs, grass and bark. Unusually for a passerine, it lays only a single egg. This southern scrub robin was photographed in the Nombinnie Nature Reserve in New South Wales. Photograph credit: JJ Harrison Recently featured: C/2022 E3 (ZTF) Passion fruit Basilica of St Paul, Rabat Archive More featured pictures Community portal – The central hub for editors, with resources, links, tasks, and announcements. Village pump – Forum for discussions about Wikipedia itself, including policies and technical issues. Site news – Sources of news about Wikipedia and the broader Wikimedia movement. Teahouse – Ask basic questions about using or editing Wikipedia. Help desk – Ask questions about using or editing Wikipedia. Reference desk – Ask research questions about encyclopedic topics. Content portals – A unique way to navigate the encyclopedia. Wikipedia is written by volunteer editors and hosted by the Wikimedia Foundation, a non-profit organization that also hosts a range of other volunteer projects: CommonsFree media repository MediaWikiWiki software development Meta-WikiWikimedia project coordination WikibooksFree textbooks and manuals WikidataFree knowledge base WikinewsFree-content news WikiquoteCollection of quotations WikisourceFree-content library WikispeciesDirectory of species WikiversityFree learning tools WikivoyageFree travel guide WiktionaryDictionary and thesaurus This Wikipedia is written in English. Many other Wikipedias are available; some of the largest are listed below. 1,000,000+ articles العربية Deutsch Español فارسی Français Italiano Nederlands 日本語 Polski Português Русский Svenska Українська Tiếng Việt 中文 250,000+ articles Bahasa Indonesia Bahasa Melayu Bân-lâm-gú Български Català Čeština Dansk Eesti Ελληνικά Esperanto Euskara עברית Հայերեն Magyar Norsk bokmål Română Simple English Slovenčina Srpski Srpskohrvatski Suomi Türkçe Oʻzbekcha 50,000+ articles Asturianu Azərbaycanca □□□□□ Bosanski کوردی Frysk Gaeilge Galego Hrvatski ქართული Kurdî Latviešu Lietuvių □□□□□□ Македонски □□□□□□□□□□ Norsk nynorsk □□□□□□ Shqip Slovenščina □□□ □□□□□□ اردو Retrieved from " 2Event 56th Annual Grammy AwardsOfficial posterDateJanuary 26, 2014 (2014-01-26)5:00–8:30 p.m. PSTLocationStaples Center, Los Angeles, California, U.S.Hosted byLL Cool JMost awards Daft Punk (5) Most nominationsJay-Z (9)Website56th Annual Grammy AwardsTelevision/radio coverageNetworkCBSViewership28.5 million viewers.[1] ← 55th · Grammy Awards · 57th → The 56th Annual Grammy Awards presentation was held on January 26, 2014, at Staples Center in Los Angeles.[2] The show was broadcast on CBS at 8 p.m. ET/PT and was hosted for the third time by LL Cool J. The show was moved to January to avoid competing with the 2014 Winter Olympics in Sochi, as was the case in 2010. The eligibility period for the 56th Annual Grammy Awards was October 1, 2012, to September 30, 2013.[3] The nominations were announced on December 6, 2013 during a live televised concert on CBS, The Grammy Nominations Concert Live – Countdown to Music's Biggest Night.[4] Jay-Z received the most nominations with nine. Justin Timberlake, Kendrick Lamar, Macklemore & Ryan Lewis and Pharrell Williams each received seven nominations. Daft Punk and Pharrell Williams were nominated twice for both Album of the Year and Record of the Year. Sound engineer Bob Ludwig received the most nominations by a non-performing artist, with five.[5] Daft Punk won five awards,[6] including Album of the Year for Random Access Memories and Record of the Year, with Pharrell Williams, for "Get Lucky", Best Pop Duo/Group Performance, and an additional win for Best Engineered Album, Non-Classical

completing a clean sweep for the project that night.[7][8] Macklemore and Ryan Lewis won four trophies, including Best New Artist, and led an industry show of support for gay marriage with a performance of their song "Same Love" to accompany a mass wedding of gay and heterosexual couples, which was presided over by Queen Latifah.[9] Lorde's "Royals" received awards for Best Pop Solo Performance and Song of the Year.[10][11] Carole King was honored as MusiCares Person of the Year on January 24, two days prior to the awards ceremony.[12] On June 4, 2013, the Recording Academy approved a number of changes recommended by its Awards & Nominations Committee, including adding a new category for Best American Roots Song to the American Music field. This songwriters' award will encompass all the subgenres in this field such as Americana, bluegrass, blues, folk, and regional roots music. The Best Hard Rock/Metal Performance category was renamed Best Metal Performance and became a stand-alone category. Hard rock performances will now be screened in the Best Rock Performance category. The Music Video field will become the Best Music Video/Film field. Its two categories will be renamed: Best Short Form Music Video will now be known as Best Music Video and Best Long Form Music Video will change into Best Music Film. These changes bring the total number of categories at the 2014 Grammy Awards to 82, up from 81 at the 2013 Grammy Awards.[13] Artist(s) Song(s) BeyoncéJay-Z "Drunk in Love" Lorde "Royals" Hunter Hayes "Invisible" Katy PerryJuicy J "Dark Horse" Robin ThickeChicago "Does Anybody Really Know What Time It Is?""Beginnings""Saturday in the Park""Blurred Lines" Keith UrbanGary Clark Jr.[14] "Cop Car" John Legend "All of Me" Taylor Swift[15] "All Too Well" P!nkNate Ruess "Try""Just Give Me a Reason" Ringo Starr "Photograph" Kendrick LamarImagine Dragons[16] "m.A.A.d city""Radioactive" Kacey Musgraves "Follow Your Arrow" Paul McCartneyRingo Starr "Queenie Eye" Merle HaggardKris KristoffersonWillie NelsonBlake Shelton[17] "Highwayman""Okie from Muskogee""Mammas Don't Let Your Babies Grow Up to Be Cowboys" Daft PunkNile RodgersStevie WonderPharrell Williams "Get Lucky""Harder, Better, Faster, Stronger""Lose Yourself to Dance""Le Freak""Another Star""Around the World" Sara BareillesCarole King[15] "Beautiful""Brave" MetallicaLang Lang[18] "One" Macklemore & Ryan Lewis[15]Mary LambertMadonna[19]Queen LatifahTrombone Shorty & Orleans Avenue "Same Love""Open Your Heart" Billie Joe ArmstrongMiranda Lambert[14] Tribute to Phil Everly"When Will I Be Loved" Nine Inch NailsQueens of the Stone AgeDave GrohlLindsey Buckingham[20] "Copy of A""My God Is the Sun" Pharrell Williams and Anna Kendrick – presented Best New Artist Anna Faris and Juanes – presented Best Pop Duo/Group Performance Steve Coogan – introduced Katy Perry Bruno Mars – introduced P!nk Ariana Grande and Miguel – presented Best Pop Solo Performance Black Sabbath – introduced Paul McCartney and Ringo Starr Jamie Foxx – presented Best Rap/Sung Collaboration Julia Roberts – introduced Paul McCartney Gloria Estefan and Marc Anthony – presented Best Pop Vocal Album Jeremy Renner – introduced Willie Nelson, Blake Shelton, Merle Haggard and Kris Kristofferson Zac Brown and Martina McBride – presented Best Country Album Neil Patrick Harris – introduced Daft Punk Cyndi Lauper – introduced Sara Bareilles and Carole King Sara Bareilles and Carole King – presented Song of the Year Jared Leto – introduced Metallica Smokey Robinson and Steven Tyler – presented Record of the Year Queen Latifah – introduced Macklemore & Ryan Lewis John Legend and Ryan Seacrest – presented Music Educator Award Lang Lang – introduced "In Memoriam" Olivia Harrison, Alicia Keys and Yoko Ono – presented Album of the Year The winners and nominees per category were:[21] Record of the Year "Get Lucky" – Daft Punk, Pharrell Williams & Nile Rodgers Thomas Bangalter & Guy-Manuel de Homem-Christo, producers; Peter Franco, Mick Guzauski, Florian Lagatta & Daniel Lerner, engineers/mixers; Bob Ludwig & Antoine Chabert, mastering engineers "Radioactive" – Imagine Dragons Alex Da Kid, producer; Manny Marroquin & Josh Mosser, engineers/mixers; Joe LaPorta, mastering engineer "Royals" – Lorde Joel Little, producer; Joel Little, engineer/mixer; Stuart Hawkes, mastering engineer "Locked Out of Heaven" – Bruno Mars Jeff Bhasker, Emile Haynie, Mark Ronson & The Smeezingtons, producers; Alalal, Josh Blair, Wayne Gordon, Ari Levine, Manny Marroquin & Mark Ronson, engineers/mixers; David Kutch, mastering engineer "Blurred Lines" – Robin Thicke featuring T.I. and Pharrell Williams Pharrell, producer; Andrew Coleman & Tony Maserati, engineers/mixers; Chris Gehringer, mastering engineer Album of the Year Random Access Memories – Daft Punk Julian Casablancas, DJ Falcon, Todd Edwards, Chilly Gonzales, Giorgio Moroder, Panda Bear, Nile Rodgers, Paul Williams & Pharrell Williams, featured artists; Thomas Bangalter, Julian Casablancas, Guy-Manuel de Homem-Christo, DJ Falcon & Todd Edwards, producers; Peter Franco, Mick Guzauski, Florian Lagatta, Guillaume Le Braz & Daniel Lerner, engineers/mixers; Antoine Chabert & Bob Ludwig, mastering engineers The Blessed Unrest – Sara Bareilles Sara Bareilles, Mark Endert & John O'Mahony, producers; Jeremy Darby, Mark Endert & John O'Mahony, engineers/mixers; Greg Calbi, mastering engineer Good Kid, M.A.A.D City – Kendrick Lamar Mary J. Blige, Dr. Dre, Drake, Jay Rock, Jay-Z, MC Eiht & Anna Wise, featured artists; DJ Dahi, Hit-Boy, Skhye Hutch, Just Blaze, Like, Terrace Martin, Dawaun Parker, Pharrell, Rahki, Scoop DeVille, Sounwave, Jack Splash, Tabu, Tha Bizness & T-Minus, producers; Derek Ali, Dee Brown, Dr. Dre, James Hunt, Mauricio "Veto" Iragorri, Mike Larson, Jared Scott, Jack Splash & Andrew Wright, engineers/mixers; Mike Bozzi & Brian "Big Bass" Gardner, mastering engineers Red – Taylor Swift Gary Lightbody & Ed Sheeran, featured artists; Jeff Bhasker, Nathan Chapman, Dann Huff, Jacknife Lee, Max Martin, Shellback, Taylor Swift, Butch Walker & Dan Wilson, producers; Joe Baldridge, Sam Bell, Matt Bishop, Chad Carlson, Nathan Chapman, Serban Ghenea, John Hanes, Sam Holland, Michael Ilbert, Tyler Johnson, Jacknife Lee, Steve Marcantonio, Manny Marroquin, Justin Niebank, John Rausch, Eric Robinson, Pawel Sek, Jake Sinclair, Mark "Spike" Stent & Andy Thompson, engineers/mixers; Tom Coyne & Hank Williams, mastering engineers The Heist – Macklemore & Ryan Lewis Ab-Soul, Ben Bridwell, Ray Dalton, Eighty4 Fly, Hollis, Mary Lambert, Baffalo Madonna, Evan Roman, Schoolboy Q, Allen Stone, The Teaching & Wanz, featured artists; Ryan Lewis, producer; Ben Haggerty, Ryan Lewis, Amos Miller, Reed Ruddy & Pete Stewart, engineers/mixers; Tom Coyne & Brian "Big Bass" Gardner, mastering engineer Song of the Year "Royals" Ella Yelich O'Connor, Joel Little, songwriters (Lorde) "Just Give Me a Reason" P!nk, Jeff Bhasker and Nate Ruess, songwriters (P!nk featuring Nate Ruess) "Locked Out of Heaven" Bruno Mars, Philip Lawrence, Ari Levine, songwriters (Bruno Mars) "Roar" Katy Perry, Lukasz Gottwald, Max Martin, Bonnie McKee and Henry Walter, songwriters (Katy Perry) "Same Love" Ben Haggerty, Ryan Lewis and Mary Lambert, songwriters (Macklemore & Ryan Lewis featuring Mary Lambert) Best New Artist Macklemore & Ryan Lewis James Blake Kendrick Lamar Kacey Musgraves Ed Sheeran Best Pop Solo Performance "Royals" – Lorde "Brave" – Sara Bareilles "When I Was Your Man" – Bruno Mars "Roar" – Katy Perry "Mirrors" – Justin Timberlake Best Pop Duo/Group Performance "Get Lucky" – Daft Punk, Pharrell Williams & Nile Rodgers "Just Give Me a Reason" – P!nk featuring Nate Ruess "Stay" – Rihanna & Mikky Ekko "Blurred Lines" – Robin Thicke featuring T.I. and Pharrell Williams "Suit & Tie" – Justin Timberlake & Jay-Z Best Pop Instrumental Album Steppin' Out – Herb Alpert The Beat – Boney James HandPicked – Earl Klugh Summer Horns – Dave Koz, Gerald Albright, Mindi Abair & Richard Elliot Hacienda – Jeff Lorber Fusion Best Pop Vocal Album Unorthodox Jukebox – Bruno Mars Paradise – Lana Del Rey Pure Heroine – Lorde Blurred Lines – Robin Thicke The 20/20 Experience – The Complete Experience – Justin Timberlake Best Dance Recording "Clarity" – Zedd & Foxes Anton Zaslavski (Zedd), producer and mixer "Need U (100%)" – Duke Dumont featuring A*M*E & MNEK Adam Dyment & Tommy Forrest, producers; Adam Dyment & Tommy Forrest, mixers "Sweet Nothing" – Calvin Harris & Florence Welch Calvin Harris, producer; Calvin Harris, mixer "Atmosphere" – Kaskade Finn Bjarnson & Ryan Raddon, producers; Ryan Raddon, mixer "This Is What It Feels Like" – Armin Van Buuren & Trevor Guthrie Armin Van Buuren & Benno De Goeij, producers; Armin Van Buuren & Benno De Goeij, mixers Best Dance/Electronica Album Random Access Memories – Daft Punk Settle – Disclosure 18 Months – Calvin Harris Atmosphere – Kaskade A Color Map of the Sun – Pretty Lights Best Traditional Pop Vocal Album To Be Loved – Michael Bublé (with Bob Rock)[22] Viva Duets – Tony Bennett and various artists The Standards – Gloria Estefan Cee Lo's Magic Moment – Cee Lo Green Now – Dionne Warwick Best Rock Performance "Radioactive" – Imagine Dragons "Kashmir (Live)" – Led Zeppelin "Always Alright" – Alabama Shakes "The Stars (Are Out Tonight)" – David Bowie "My God Is the Sun" – Queens of the Stone Age "I'm Shakin'" – Jack White Best Metal Performance "God Is Dead?" – Black Sabbath "T.N.T." – Anthrax "The Enemy Inside" – Dream Theater "In Due Time" – Killswitch Engage "Room 24" – Volbeat & King Diamond Best Rock Song "Cut Me Some Slack" Dave Grohl, Paul McCartney, Krist Novoselic & Pat Smear, songwriters (Paul McCartney, Dave Grohl, Krist Novoselic, Pat Smear) "Ain't Messin 'Round" Gary Clark Jr., songwriter (Gary Clark Jr.) "Doom and Gloom" Mick Jagger & Keith Richards, songwriters (The Rolling Stones) "God Is Dead?" Geezer Butler, Tony Iommi & Ozzy Osbourne, songwriters (Black Sabbath) "Panic Station" Matthew Bellamy, songwriter (Muse) Best Rock Album Celebration Day – Led Zeppelin 13 – Black Sabbath The Next Day – David Bowie Mechanical Bull – Kings of Leon ...Like Clockwork – Queens of the Stone Age Psychedelic Pill – Neil Young & Crazy Horse Best Alternative Music Album Modern Vampires of the City – Vampire Weekend The Worse Things Get, The Harder I Fight, The Harder I Fight, The More I Love You – Neko Case Trouble Will Find Me – The National Hesitation Marks – Nine Inch Nails Lonerism – Tame Impala Best R&B Performance "Something" – Snarky Puppy & Lalah Hathaway "Love and War" – Tamar Braxton "Best of Me" – Anthony Hamilton "Nakamarra" – Hiatus Kaiyote & Q-Tip "How Many Drinks?" – Miguel & Kendrick Lamar Best Traditional R&B Performance "Please Come Home"– Gary Clark Jr. "Get It Right" – Fantasia "Quiet Fire" – Maysa "Hey Laura" – Gregory Porter "Yesterday'" – Ryan Shaw Best R&B Song "Pusher Love Girl" James Fauntleroy, Jerome Harmon, Timothy Mosley & Justin Timberlake, songwriters (Justin Timberlake) "Best of Me" Anthony Hamilton & Jairus Mozee, songwriters (Anthony Hamilton) "Love and War" Tamar Braxton, Darhyl Camper Jr., LaShawn Daniels & Makeba Riddick, songwriters (Tamar Braxton) "Only One" PJ Morton, songwriter (PJ Morton & Stevie Wonder) "Without Me" Fantasia, Missy Elliott, Al Sherrod Lambert, Harmony Samuels & Kyle Stewart, songwriters (Fantasia featuring Kelly Rowland & Missy Elliott) Best Urban Contemporary Album Unapologetic – Rihanna Love and War – Tamar Braxton Side Effects of You – Fantasia One: In the Chamber – Salaam Remi New York: A Love Story – Mack Wilds Best R&B Album Girl on Fire – Alicia Keys R&B Divas – Faith Evans Love in the Future – John Legend Better – Chrisette Michele Three Kings – TGT Best Rap Performance "Thrift Shop" – Macklemore & Ryan Lewis featuring Wanz "Swimming Pools (Drank)" – Kendrick Lamar "Berzerk" – Eminem "Started from the Bottom" – Drake "Tom Ford" – Jay-Z Best Rap/Sung Collaboration "Holy Grail" – Jay-Z & Justin Timberlake "Now or Never" – Kendrick Lamar & Mary J. Blige "Power Trip" – J. Cole & Miguel "Remember You" – Wiz Khalifa & The Weeknd "Part II (On the Run)" – Jay-Z & Beyoncé Best Rap Song "Thrift Shop" Ben Haggerty & Ryan Lewis, songwriters (Macklemore & Ryan Lewis featuring Wanz) "F**in' Problems" Tauheed Epps, Aubrey Graham, Kendrick Lamar, Rakim Mayers & Noah Shebib, songwriters (ASAP Rocky featuring Drake, 2 Chainz & Kendrick Lamar) "Holy Grail" Shawn Carter, Terius Nash, J. Harmon, Timothy Mosley, Justin Timberlake & Ernest Wilson, songwriters (Kurt Cobain, Dave Grohl & Krist Novoselic, songwriters) (Jay-Z & Justin Timberlake) "New Slaves" Christopher Breaux, Ben Bronfman, Mike Dean, Louis Johnson, Malik Jones, Elon Rutberg, Sakiya Sandifer, Che Smith, Kanye West & Cydell Young, songwriters (Anna Adamis & Gabor Presser, songwriters) (Kanye West) "Started from the Bottom" W. Coleman, Aubrey Graham & Noah Shebib, songwriters (Bruno Sanfilippo, songwriter) (Drake) Best Rap Album The Heist – Macklemore & Ryan Lewis Nothing Was the Same – Drake good kid, m.A.A.d city – Kendrick Lamar Magna Carta... Holy Grail – Jay-Z Yeezus – Kanye West Best Country Solo Performance "Wagon Wheel" – Darius Rucker "I Drive Your Truck" – Lee Brice "I Want Crazy" – Hunter Hayes "Mama's Broken Heart" – Miranda Lambert "Mine Would Be You" – Blake Shelton Best Country Duo/Group Performance "From This Valley" – The Civil Wars "Don't Rush" – Kelly Clarkson & Vince Gill "Your Side of the Bed" – Little Big Town "Highway Don't Care" – Tim McGraw, Taylor Swift & Keith Urban "You Can't Make Old Friends" – Kenny Rogers & Dolly Parton Best Country Song "Merry Go 'Round" Shane McAnally, Kacey Musgraves & Josh Osborne, songwriters (Kacey Musgraves) "Begin Again" Taylor Swift, songwriter (Taylor Swift) "I Drive Your Truck" Jessi Alexander, Connie Harrington & Jimmy Yeary, songwriters (Lee Brice) "Mama's Broken Heart" Brandy Clark, Shane McAnally & Kacey Musgraves, songwriters (Miranda Lambert) "Mine Would Be You" Jessi Alexander, Connie Harrington & Deric Ruttan, songwriters (Blake Shelton) Best Country Album Same Trailer Different Park – Kacey Musgraves Night Train – Jason Aldean Two Lanes of Freedom – Tim McGraw Based on a True Story... – Blake Shelton Red – Taylor Swift Best New Age Album Love's River – Laura Sullivan Lux – Brian Eno Illumination – Peter Kater Final Call – Kitaro Awakening The Fire – R. Carlos Nakai & Will Clipman Best Improvised Jazz Solo "Orbits" – Wayne Shorter, soloist "Don't Run" – Terence Blanchard, soloist "Song for Maura" – Paquito D'Rivera, soloist "Song Without Words #4: Duet" – Fred Hersch, soloist "Stadium Jazz" – Donny McCaslin, soloist Best Jazz Vocal Album Liquid Spirit – Gregory Porter The World According to Andy Bey – Andy Bey Attachments – Lorraine Feather WomanChild – Cécile McLorin Salvant After Blue – Tierney Sutton Best Jazz Instrumental Album Money Jungle: Provocative in Blue – Terri Lyne Carrington Guided Tour – The New Gary Burton Quartet Life Forum – Gerald Clayton Pushing the World Away – Kenny Garrett Out Here – Christian McBride Trio Best Large Jazz Ensemble Album Night in Calisia – Randy Brecker, Włodek Pawlik Trio & Kalisz Philharmonic Brooklyn Babylon – Darcy James Argue's Secret Society Wild Beauty – Brussels Jazz Orchestra featuring Joe Lovano March Sublime – Alan Ferber Intrada – Dave Slonaker Big Band Best Latin Jazz Album Song for Maura – Paquito D'Rivera and Trio Corrente La Noche Más Larga – Buika Yo – Roberto Fonseca Egg_n – Omar Sosa Latin Jazz-Jazz Latin – Wayne Wallace Latin Jazz Quintet Best Gospel/Contemporary Christian Music Performance "Break Every Chain" (Live) – Tasha Cobbs "Hurricane" – Natalie Grant "Lord, I Need You" – Matt Maher "If He Did It Before... Same God" (Live) – Tye Tribbett "Overcomer" – Mandisa Best Gospel Song "If He Did It Before... Same God" (Live) Tye Tribbett, songwriter (Tye Tribbett) "Deitrick Haddon" Calvin Frazier & Deitrick Haddon, songwriters (Deitrick Haddon) "If I Believe" Wirlie Morris, Michael Paran, Charlie Wilson & Mahin Wilson, songwriters (Charlie Wilson) "A Little More Jesus" Erica Campbell and Tina Campbell & Warryn Campbell, songwriters (Erica Campbell) "Still" Percy Bady, songwriter (Percy Bady featuring Lowell Pye) Best Contemporary Christian Music Song "Overcomer" David Garcia, Ben Glover & Christopher Stevens, songwriters (Mandisa) "Hurricane" Matt Bronleewe, Natalie Grant & Cindy Morgan, songwriters (Natalie Grant) "Love Take Me Over" Steven Curtis Chapman, songwriter (Steven Curtis Chapman) "Speak Life" Toby McKeehan, Jamie Moore & Ryan Stevenson, songwriters (Tobymac) "Whom Shall I Fear (God of Angel Armies)" Ed Cash, Scott Cash & Chris Tomlin, songwriters (Chris Tomlin) "In God's Time" – Tehrah Best Gospel Album Greater Than (Live) – Tye Tribbett Grace (Live) – Tasha Cobbs Best For Last: 20 Year Celebration Vol. 1 (Live) – Donald Lawrence Best Days Yet – Bishop Paul S. Morton God Chaser (Live) – William Murphy Best Contemporary Christian Music Album Overcomer – Mandisa We Won't Be Shaken – Building 429 All the People Said Amen (Live) – Matt Maher Your Grace Finds Me (Live) – Matt Redman Burning Lights – Chris Tomlin Best Latin Pop Album Vida – Draco Rosa Faith, Hope y Amor – Frankie J Viajero Frecuente – Ricardo Montaner Syntek – Aleks Syntek 12 Historias – Tommy Torres Best Latin Rock Urban or Alternative Album Treinta Días – La Santa Cecilia El Objeto Antes Llamado Disco – Café Tacuba Ojo Por Ojo – El Tri Chances – Illya Kuryaki and the Valderramas Repeat After Me – Los Amigos Invisibles Best Regional Mexican Music Album (Including Tejano) A Mi Manera – Mariachi Divas de Cindy Shea El Free – Banda Los Recoditos En Peligro De Extinción – Intocable Romeo y Su Nieta – Paquita la del Barrio 13 Celebrando El 13 – Joan Sebastian Best Tropical Latin Album Pacific Mambo Orchestra – Pacific Mambo Orchestra 3.0 – Marc Anthony Como Te Voy A Olvidar – Los Angeles Azules Sergio George Presents Salsa Giants – Various Artists Corazón Profundo – Carlos Vives Best American Roots Song "Love Has Come For You" Edie Brickell & Steve Martin, songwriters (Steve Martin & Edie Brickell) "Build Me Up From Bones" Sarah Jarosz, songwriter (Sarah Jarosz) "Invisible" Steve Earle, songwriter (Steve Earle & The Dukes (& Duchesses)) "Keep Your Dirty Lights On" Tim O'Brien & Darrell Scott, songwriters (Tim O'Brien and Darrell Scott) "Shrimp Po-Boy, Dressed" Allen Toussaint, songwriter (Allen Toussaint) Best Americana Album Old Yellow Moon — Emmylou Harris & Rodney Crowell Love Has Come for You — Steve Martin & Edie Brickell Buddy and Jim — Buddy Miller and Jim Lauderdale One True Vine — Mavis Staples Songbook — Allen Toussaint Best Bluegrass Album The Streets of Baltimore — Del McCoury Band It's Just a Road — The Boxcars Brothers of the Highway — Dailey & Vincent This World Oft Can Be — Della Mae Three Chords and the Truth — James King Best Blues Album Get Up! — Ben Harper With Charlie Musselwhite Remembering Little Walter — Billy Boy Arnold, Charlie Musselwhite, Mark Hummel, Sugar Ray Norcia & James Harman Cotton Mouth Man — James Cotton Seesaw — Beth Hart with Joe Bonamassa Down In Louisiana — Bobby Rush Best Folk Album My Favorite Picture of You — Guy Clark Sweetheart of the Sun — The Greencards Build Me Up from Bones — Sarah Jarosz The Ash & Clay — The Milk Carton Kids They All Played for Us: Arhoolie Records 50th Anniversary Celebration — Various Artists; Chris Strachwitz, producer Best Regional Roots Music Album Dockside Sessions — Terrance Simien & The Zydeco Experience The Life & Times Of...The Hot 8 Brass Band — Hot 8 Brass Band Hula Ku'i — Kahulanui Le Fou — Zachary Richard Apache & Crown Dance Songs — Joe Tohonnie Jr. Best Reggae Album Ziggy Marley In Concert – Ziggy Marley Reincarnated – Snoop Lion One Love, One Life – Beres Hammond The Messiah – Sizzla Reggae Connection – Sly & Robbie and the Jam Masters Best World Music Album Savor Flamenco – Gipsy Kings Live: Singing for Peace Around the World – Ladysmith Black Mambazo (tie) No Place for My Dream – Femi Kuti The Living Room Sessions Part 2 – Ravi Shankar Best Children's Album Throw a Penny in the Wishing Well – Jennifer Gasoi Blue Clouds – Elizabeth Mitchell & You Are My Flower The Mighty Sky – Beth Nielsen Chapman Recess – Justin Roberts Singing Our Way Through: Songs for the World's Bravest Kids – Alastair Moock & Friends Best Spoken Word Album (Includes Poetry, Audio Books & Story Telling) America Again: Re-becoming The Greatness We Never Weren't – Stephen Colbert Carrie and Me – Carol Burnett Let's Explore Diabetes with Owls – David Sedaris Still Foolin' 'Em – Billy Crystal The Storm King – Pete Seeger Best Comedy Album Calm Down Gurrl – Kathy Griffin I'm Here to Help – Craig Ferguson A Little Unprofessional – Ron White Live – Tig Notaro That's What I'm Talkin' About – Bob Saget Best Musical Theater Album Kinky Boots Billy Porter & Stark Sands, principal soloists; Sammy James Jr., Cyndi Lauper, Stephen Oremus & William Wittman, producers; Cyndi Lauper, composer & lyricist (Original Broadway Cast) Matilda the Musical Bertie Carvel, Sophia Gennusa, Laurence, Bailey Ryon, Miley Shapiro & Lauren Ward, principal soloists; Michael Croiter, Van Dean & Chris Nightingale, producers; Tim Minchin, composer & lyricist (Original Broadway Cast) Motown: The Musical Brandon Victor Dixon & Valisia Lakae, principal soloists; Frank Filipetti & Ethan Popp, producers (Robert Bateman, Al Cleveland, Georgia Dobbins, Lamont Dozier, William Garrett, Marvin Gaye, Berry Gordy, Freddie Gorman, Cornelius Grant, Brian Holland, Ivy Jo Hunter, Michael Lovesmith, Alphonzo Mizell, Freddie Perren, Deke Richards, William Stevenson, Norman Whitfield & Stevie Wonder, composers; Nickolas Ashford, Marvin Gaye, Berry Gordy, Lula Mae Hardaway, Eddie Holland, Michael Lovesmith, Deke Richards, Smokey Robinson, Barrett Strong, Ronald White, Stevie Wonder & Syreeta Wright, lyricists) (Original Broadway Cast) Best Compilation Soundtrack for Visual Media Sound City: Real to Reel – Dave Grohl & Various Artists Django Unchained – Various Artists The Great Gatsby (Deluxe Edition) – Various Artists Les Misérables (Deluxe Edition) – Various Artists Muscle Shoals – Various Artists Best Score Soundtrack for Visual Media Skyfall Thomas Newman, composer Argo Alexandre Desplat, composer The Great Gatsby Craig Armstrong, composer Life of Pi Mychael Danna, composer Lincoln John Williams, composer Zero Dark Thirty Alexandre Desplat, composer Best Song Written for Visual Media "Skyfall" (from Skyfall) Adele Adkins & Paul Epworth, songwriters (Adele) "Atlas" (from The Hunger Games: Catching Fire) Guy Berryman, Jonny Buckland, Will Champion & Chris Martin, songwriters (Coldplay) "Silver Lining (Crazy 'Bout You)" (from Silver Linings Playbook) Diane Warren, songwriter (Jessie J) "We Both Know" (from Safe Haven) Colbie Caillat & Gavin DeGraw, songwriters (Colbie Caillat featuring Gavin DeGraw) "Young and Beautiful" (from The Great Gatsby) Lana Del Rey & Rick Nowels, songwriters (Lana Del Rey) "You've Got Time" (from Orange Is the New Black) Regina Spektor, songwriter (Regina Spektor) Best Instrumental Composition "Pensamientos For Solo Alto Saxophone And Chamber Orchestra" Clare Fischer, composer (The Clare Fischer Orchestra) " Away" Chuck Owen, composer (Chuck Owen & The Jazz Surge) "California Pictures For String Quartet" Gordon Goodwin, composer (Quartet San Francisco) "Koko On The Boulevard" Scott Healy, composer (Scott Healy Ensemble) "String Quartet No. 1: Funky Diversion In Three Parts" Vince Mendoza, composer (Quartet San Francisco) Best Instrumental Arrangement "On Green Dolphin Street" Gordon Goodwin, arranger (Gordon Goodwin's Big Phat Band) "Invitation" Kim Richmond, arranger (The Kim Richmond Concert Jazz Orchestra) "Side Hikes – A Ridge Away" Chuck Owen, arranger (Chuck Owen & The Jazz Surge) "Skylark" Nan Schwartz, arranger (Amy Dickson) "Wild Beauty" Gil Goldstein, arranger (Brussels Jazz Orchestra Featuring Joe Lovano) Best Instrumental Arrangement Accompanying Vocalist(s) "Swing Low" Gil Goldstein, arranger (Bobby McFerrin & Esperanza Spalding) "La Vida Nos Espera" Nan Schwartz, arranger (Gian Marco) "Let's Fall In Love" Chris Walden, arranger (Calabria Foti Featuring Seth MacFarlane) "The Moon's A Harsh Mistress" John Hollenbeck, arranger (John Hollenbeck) "What A Wonderful World" Shelly Berg, arranger (Gloria Estefan) Best Recording Package Long Night Moon Sarah Dodds & Shauna Dodds, art directors (Reckless Kelly) Automatic Music Can Be Fun Mike Brown, Zac Decamp, Brian Grunert & Annie Stoll, art directors (Geneseo) Magna Carta... Holy Grail Brian Roettinger, art director (Jay-Z) Metallica Through The Never (Music From The Motion Picture) Bruce Duckworth, Sarah Moffat & David Turner, art directors (Metallica) The Next Day Jonathan Barnbrook, art director (David Bowie) Best Boxed or Special Limited Edition Package Wings Over America (Deluxe Edition) Simon Earith & James Musgrave, art directors (Paul McCartney and Wings) The Brussels Affair Charles Dooher & Scott Sandler, art directors (The Rolling Stones) How Do You Do (Limited Edition Box Set) Mayer Hawthorne, art director (Mayer Hawthorne) The Road To Red Rocks (Special Edition) Ross Stirling, art director (Mumford & Sons) The Smith Tapes Masaki Koike, art director (Various Artists) Best Album Notes Afro Blue Impressions (Remastered & Expanded) Neil Tesser, album notes writer (John Coltrane) Call It Art 1964–1965 Ben Young, album notes writer (New York Art Quartet) Electric Music for the Mind and Body Alec Palao, album notes writer (Country Joe and the Fish) Stravinsky: Le Sacre Du Printemps Jonathan Cott, album notes writer (Leonard Bernstein & New York Philharmonic) 360 Sound: The Columbia Records Story Sean Wilentz, album notes writer (various artists) Work Hard, Play Hard, Pray Hard: Hard Time, Good Time & End Time Music, 1923–1936 Nathan Salsburg, album notes writer (various artists) Best Historical Album Charlie Is My Darling – Ireland 1965 (tie) Teri Landi, Andrew Loog Oldham & Steve Rosenthal, compilation producers; Bob Ludwig, mastering engineer (The Rolling Stones) The Complete Sussex And Columbia Albums (tie) Leo Sacks, compilation producer; Joseph M. Palmaccio, Tom Ruff & Mark Wilder, mastering engineers (Bill Withers) Call It Art 1964–1965 Joe Lizzi & Ben Young, compilation producers; Steve Fallone, Joe Lizzi & Ben Young, mastering engineers (New York Art Quartet) Pictures Of Sound: One Thousand Years Of Educed Audio: 980–1980 Patrick Feaster & Steven Ledbetter, compilation producers; Michael Graves, mastering engineer (Various Artists) Wagner: Der Ring Des Nibelungen (Deluxe Edition) Philip Siney, compilation producer; Ben Turner, mastering engineer (Sir Georg Solti) Best Engineered Album, Non-Classical Random Access Memories Peter Franco, Mick Guzauski, Florian Lagatta & Daniel Lerner, engineers; Bob Ludwig, mastering engineer (Daft Punk) Annie Up Chuck Ainlay, engineer; Bob Ludwig, mastering engineer (Pistol Annies) The Blue Room Helik Hadar & Leslie Ann Jones, engineers; Bernie Grundman, mastering engineer (Madeleine Peyroux) The Devil Put Dinosaurs Here Paul Figueroa & Randy Staub, engineers; Ted Jensen, mastering engineer (Alice in Chains) ...Like Clockwork Joe Barresi & Mark Rankin, engineers; Gavin Lurssen, mastering engineer (Queens of the Stone Age) The Moorings Trina Shoemaker, engineer; Eric Conn, mastering engineer (Andrew Duhon) Producer of the Year, Non-Classical Pharrell Williams "BBC" (Jay-Z) "Blurred Lines" (Robin Thicke featuring T.I. & Pharrell) "Happy" (Pharrell Williams) "I Can't Describe (The Way I Feel)" (Jennifer Hudson featuring T.I.) "Nuclear" (Destiny's Child) "Oceans" (Jay-Z featuring Frank Ocean) "Reach Out Richard" (Mayer Hawthorne) "The Stars Are Ours" (Mayer Hawthorne) Rob Cavallo All That Echoes (Josh Groban) "Bright Lights" (Gary Clark Jr.) ¡Dos! (Green Day) "If I Loved You" (Delta Rae featuring Lindsey Buckingham) "Love They Say" (Tegan and Sara) "Things Are Changin'" (Gary Clark Jr.) ¡Tré! (Green Day) "When My Train Pulls In" (Gary Clark Jr.) "You've Got Time" (Regina Spektor) Dr. Luke Bounce It (Juicy J Featuring Wale & Trey Songz) "Crazy Kids" (Kesha) "Fall Down (will.i.am featuring Miley Cyrus) "Give It 2 U" (Robin Thicke featuring Kendrick Lamar) "Roar" (Katy Perry) "Rock Me" (One Direction) "Wrecking Ball" (Miley Cyrus) "Play it Again" (Becky G) Ariel Rechtshaid Days Are Gone (Haim) "Everything Is Embarrassing" (Sky Ferreira) "Lost in My Bedroom" (Sky Ferreira) Modern Vampires of the City (Vampire Weekend) Reincarnated (Snoop Lion) True Romance (Charli XCX) "You're No Good" (Major Lazer featuring Santigold, Vybz Kartel, Danielle Haim and Yasmin) Jeff Tweedy The Invisible Way (Low) One True Vine (Mavis Staples) Wassaic Way (Sarah Lee Guthrie & Johnny Irion) Best Remixed Recording, Non-Classical "Summertime Sadness" (Cedric Gervais Remix) Cedric Gervais, remixer (Lana Del Rey) "Days Turn Into Nights" (Andy Caldwell Remix) Andy Caldwell, remixer (Delerium featuring Michael Logen) "If I Lose Myself" (Alesso Vs. OneRepublic) Alesso, remixer (OneRepublic) "Locked Out of Heaven" (Sultan + Ned Shepard Remix) Sultan & Ned Shepard, remixers (Bruno Mars) "One Love/People Get Ready" (Photek Remix) Rupert Parkes, remixer (Bob Marley and the Wailers) Best Surround Sound Album Live Kisses Al Schmitt, surround mix engineer; Tommy LiPuma, surround producer (Paul McCartney) Sailing the Seas of Cheese (Deluxe Edition) Les Claypool & Jason Mills, surround mix engineers; Stephen Marcussen, surround mastering engineer; Les Claypool & Jeff Fura, surround producers (Primus) Signature Sound Opus One Leslie Ann Jones, surround mix engineer; Michael Romanowski, surround mastering engineer; Herbert Waltl, surround producer (various artists) Sixteen Sunsets Jim Anderson, surround mix engineer; Darcy Proper, surround mastering engineer; Jim Anderson & Jane Ira Bloom, surround producers (Jane Ira Bloom) Sprung Rhythm Daniel Shores, surround mix engineer; Daniel Shores, surround mastering engineer; Dan Merceruio, surround producer (Richard Scerbo & Inscape) Best Engineered Album, Classical Winter Morning Walks David Frost, Brian Losch & Tim Martyn, engineers; Tim Martyn, mastering engineer (Dawn Upshaw, Maria Schneider, Australian Chamber Orchestra & Saint Paul Chamber Orchestra) Hymn to the Virgin Morten Lindberg, engineer (Tone Bianca, Sparre Dahl, & Schola Cantorum) La Voie Triomphale Morten Lindberg, engineer (Ole Kristian Ruud & Staff Band of the Norwegian Armed Forces) Roomful of Teeth Mark Donahue & Jesse Lewis, engineers (Brad Wells & Roomful of Teeth) Vinci: Artaserse Hans-Martin Renz, Wolfgang Rixius & Ulrich Ruscher, engineers (Diego Fasolis, Philippe Jaroussky, Max Emanuel Cenčić, Daniel Behle, Franco Fagioli, Valer Barna-Sabadus, Yuriy Mynenko & Concerto Köln) Producer of the Year, Classical David Frost Andres: Home Stretch (Timo Andres, Andrew Cyr & Metropolis Ensemble) Angel Heart, A Music Storybook (Matt Haimovitz & Uccello) Beethoven: Piano Sonatas, Vol. 2 (Jonathan Biss) Ben-Haim: Chamber Works (ARC Ensemble) Celebrating The American Spirit (Judith Clurman & Essential Voices USA) Elgar: Enigma Variations; Vaughan Williams: The Wasps; Greensleeves (Michael Stern & Kansas City Symphony) Guilty Pleasures (Renée Fleming, Sebastian Lang-Lessing & Philharmonia Orchestra) Verdi: Otello (Riccardo Muti, Aleksandrs Antonenko, Krassimira Stoyanova, Carlo Guelfi, Chicago Symphony Chorus & Chicago Symphony Orchestra) Winter Morning Walks (Dawn Upshaw, Maria Schneider, Australian Chamber Orchestra & St. Paul Chamber Orchestra) Manfred Eicher Beethoven: Diabelli-Variationen (András Schiff) Canto Oscuro (Anna Gourari) Pärt: Adam's Lament (Tõnu Kaljuste, Latvian Radio Choir, Vox Clamantis, Sinfonietta Riga, Estonian Philharmonic Chamber Choir & Tallinn Chamber Orchestra) Tabakova: String Paths (Maxim Rysanov) Marina A. Ledin, Victor Ledin Bizet: Symphony In C; Jeux D'Enfants; Variations Chromatiques (Martin West & San Francisco Ballet Orchestra) Traveling Sonata – European Music For Flute & Guitar (Viviana Guzmán & Jérémy Jouve) Voyages (Conrad Tao) Zia (Del Sol String Quartet) James Mallinson Berlioz: Grande Messe Des Morts (Colin Davis, London Symphony Chorus, London Philharmonic Choir & London Symphony Orchestra) Bloch: Symphony In C-Sharp Minor & Poems Of The Sea (Dalia Atlas & London Symphony Orchestra) Fauré: Requiem; Bach: Partita, Chorales & Ciaccona (Nigel Short, Tenebrae & London Symphony Orchestra Chamber Ensemble) Nielsen: Symphonies Nos. 2 & 3 (Colin Davis & London Symphony Orchestra) Wagner: Das Rheingold (Valery Gergiev, René Pape, Stephan Rügamer, Nikolai Putilin & Mariinsky Orchestra) Wagner: Die Walküre (Valery Gergiev, Anja Kampe, Jonas Kaufmann, René Pape, Nina Stemme & Mariinsky Orchestra) Weber: Der Freischütz (Colin Davis, Christine Brewer, Sally Matthews, Simon O'Neill, London Symphony Chorus & London Symphony Orchestra) Jay David Saks Adams: Nixon in China (John Adams, Russell Braun, Ginger Costa-Jackson, James Maddalena, Janis Kelly, Richard Paul Fink, Robert Brubaker, Kathleen Kim, Metropolitan Opera Chorus and Orchestra) Adès: The Tempest (Thomas Adès, Audrey Luna, Isabel Leonard, Alan Oke, Simon Keenlyside, Metropolitan Opera Chorus & Orchestra) The Enchanted Island (William Christie, Joyce DiDonato, David Daniels, Danielle De Niese, Luca Pisaroni, Lisette Oropesa, Plácido Domingo, Metropolitan Opera Orchestra & Chorus) Handel: Rodelinda (Harry Bicket, Renée Fleming, Andreas Scholl, Joseph Kaiser, Stephanie Blythe, Iestyn Davies, Shenyang & The Metropolitan Opera Orchestra) Live At Carnegie Hall (James Levine, Evgeny Kissin & The Metropolitan Opera Orchestra) Verdi: Rigoletto (Michele Mariotti, Željko Lučić, Diana Damrau, Piotr Beczała, Oksana Volkova, Štefan Kocán, The Metropolitan Opera Orchestra & Chorus) Best Orchestral Performance Sibelius: Symphonies Nos. 1 & 4 Osmo Vänskä (conductor), Minnesota Orchestra Atterberg: Orchestral Works Vol. 1 Neeme Järvi (conductor), Gothenburg Symphony Orchestra Lutoslawski: Symphony No. 1 Esa-Pekka Salonen (conductor), Los Angeles Philharmonic Schumann: Symphony No. 2; Overtures Manfred & Genoveva Claudio Abbado (conductor), Orchestra Mozart Stravinsky: Le Sacre du Printemps Simon Rattle (conductor), Berliner Philharmoniker Best Opera Recording Adès: The Tempest Thomas Adès (conductor); Simon Keenlyside, Isabel Leonard, Audrey Luna, Alan Oke (soloists); Luisa Bricetti and Victoria Warivonchick (producers) Britten: The Rape of Lucretia Oliver Knussen (conductor); Ian Bostridge, Peter Coleman-Wright, Susan Gritton, Angelika Kirchschlager (soloists); John Fraser (producer) Kleiberg: David and Bathsheba Tõnu Kaljuste (conductor); Anna Eimarsson and Johannes Weisser (soloists); Morten Lindberg (producer) Vinci: Artaserse Diego Fasolis (conductor); Valer Barna-Sabadus, Daniel Behle, Max Emanuel Cenčić, Franco Fagioli, Philippe Jaroussky (soloists); Ulrich Russcher (producer) Wagner: Der Ring des Nibelungen Christian Thielemann (conductor); Katarina Dalayman, Albert Dohmen, Stephen Gould, Eric Halfvarson, Linda Watson (soloists); Ohmar Eichinger (producer) Best Choral Performance Performers who are not eligible for an award (such as orchestras, soloists or choirs) are mentioned in parentheses Pärt: Adam's Lament Tõnu Kaljuste (conductor) (with Tui Hirv & Rainer Vilu; Estonian Philharmonic Chamber Choir; Sinfonietta Riga & Tallinn Chamber Orchestra; Latvian Radio Choir & Vox Clamantis) Berlioz: Grande Messe de Morts Colin Davis (conductor) (with Barry Banks, London Symphony Orchestra, London Philharmonic Choir and London Symphony Chorus) Parry: Works for Chorus & Orchestra Neeme Järvi (conductor), Adrian Partington (chorus master) (with Amanda Roocroft, BBC National Orchestra of Wales and BBC National Chorus of Wales) Whitbourn: Annelies James Jordan (conductor) (with Arianna Zukerman, The Lincoln Trio and the Westminster Williamson Voices) Palestrina: Volume 3 Harry Christophers (conductor) (with The Sixteen) Best Chamber Music/Small Ensemble Performance Roomful of Teeth Brad Wells & Roomful of Teeth Beethoven: Violin Sonatas Leonidas Kavakos & Enrico Pace Cage: The 10,000 Things Vicki Ray, William Winant, Aron Kallay & Tom Peters Duo Hélène Grimaud & Sol Gabetta Times Go By Turns New York Polyphony Best Classical Instrumental Solo Corigliano: Conjurer – Concerto for Percussionist & String Orchestra Evelyn Glennie (soloist), David Alan Miller (conductor) Bartók, Eötvös & Ligeti Patricia Kopatchinskaja (soloist), Peter Eötvös (conductor) The Edge of Light Gloria Cheng Lindberg: Piano Concerto No. 2 Yefim Bronfman (soloist), Alan Gilbert (conductor) Salonen: Violin Concerto; Nyx Leila Josefowicz (soloist), Esa-Pekka Salonen (conductor) Schubert: Piano Sonatas D. 845 & D. 960 Maria João Pires Best Classical Vocal Solo Winter Morning Walks Dawn Upshaw Drama Queens Joyce DiDonato Mission Cecilia Bartoli Schubert: Winterreise Christoph Prégardien Wagner Jonas Kaufmann Best Classical Compendium Hindemith: Violinkonzert; Symphonic; Konzertmusik Christoph Eschenbach (conductor) Holmboe: Concerto Dima Slobodeniouk (conductor), Preben Iwan (producer) Tabakova: String Paths Maxim Rysanov (conductor), Manfred Eicher (producer) Best Classical Contemporary Composition Winter Morning Walks Maria Schneider Piano Concerto No. 2 Magnus Lindberg Adam's Lament Arvo Pärt Violin Concerto Esa-Pekka Salonen Partita for 8 Voices Caroline Shaw Best Music Video "Suit & Tie" – Justin Timberlake & Jay-Z David Fincher, video director; Timory King, video producer "Safe and Sound" – Capital Cities Grady Hall, video director; Buddy Enright, video producer; Daniel Weisman, video producer; Danny Lockwood, video producer "Picasso Baby: A Performance Art Film" – Jay-Z Mark Romanek, video director; Shawn Carter & Aristides McGarry, video producers "Can't Hold Us" – Macklemore & Ryan Lewis featuring Ray Dalton Jon Jon Augustavo, Jason Koenig & Ryan Lewis, video directors; Tricia Davis, Honna Kimmerer & Jenny Koenig, video producers "I'm Shakin'" – Jack White Dori Oskowitz, video director; Raquel Costello, video producer Best Music Film Live Kisses – Paul McCartney Jonas Åkerlund, video director; Violaine Etienne, Aron Levine & Scott Rodger, video producers Live 2012 – Coldplay Paul Dugdale, video director; Jim Parsons, video producer ¡Cuatro! – Green Day Tim Wheeler, video director; Tim Lynch, video producer I'm in I'm Out and I'm Gone: The Making of Get Up! – Ben Harper with Charlie Musselwhite Danny Clinch, video director; Ben Harper, video producer The Road to Red Rocks – Mumford & Sons Nicolas Jack Davies & Frederick Scott, video directors; Dan Bowen, video producer Carole King[12] The Beatles[23] Clifton Chenier The Isley Brothers Kraftwerk Kris Kristofferson Armando Manzanero Maud Powell Rick Hall[23] Jim Marshall Ennio Morricone Emile Berliner Lexicon Kent Knappenberger (of Westfield Academy and Central School in Westfield, New York) Main article: Grammy Hall of Fame Title Artist Record Label Year of Release Genre Format After the Gold Rush Neil Young and Crazy Horse Reprise 1970 Folk Rock Album All Things Must Pass George Harrison Apple 1970 Rock Album The Chicago Transit Authority Chicago Columbia 1969 Progressive Rock Album Cosmo's Factory Creedence Clearwater Revival Fantasy 1970 Rock Album Doc Watson Doc Watson Vanguard 1964 Folk Album "Fortunate Son" Creedence Clearwater Revival Fantasy 1969 Rock Single "Georgia on My Mind" Hoagy Carmichael & his Orchestra (featuring Bix Beiderbecke on Cornet) Victor 1930 Jazz Single "Get Up (I Feel Like Being a) Sex Machine" James Brown King 1970 Funk Single "Honky Tonk Women" The Rolling Stones London 1969 Hard Rock Single "Jolene" Dolly Parton RCA Nashville 1973 Country Pop Single The Joshua Tree U2 Island 1987 Rock Album Kristofferson Kris Kristofferson Monument 1970 Country Album "Low Rider" War United Artists 1975 Funk Single Mary Poppins Cast including Julie Andrews, Dick Van Dyke, David Tomlinson, Karen Dotrice, Matthew Garber and others Walt Disney 1964 Soundtrack Album "Nobody Knows the Trouble I've Seen" Louis Armstrong Decca 1938 Negro Spiritual Single "Raindrops Keep Fallin' on My Head" B. J. Thomas Scepter 1969 Easy Listening Single "Rapper's Delight" The Sugarhill Gang Sugar Hill 1979 Old-school hip hop Single Relaxin' with the Miles Davis Quintet Miles Davis & his Quintet Prestige 1958 Jazz Album "The Revolution Will Not Be Televised" Gil Scott-Heron Flying Dutchman 1971 Funk Single "Strange Things Happening Every Day" Sister Rosetta Tharpe Decca 1944 Negro Spiritual Single "Sweet Home Chicago" Robert Johnson Vocalion 1936 Blues Single "3 O'Clock Blues" B.B. King RPM 1951 R&B Single "Under the Boardwalk" The Drifters Atlantic 1964 Pop Single "Walk This Way" Run-D.M.C. featuring Aerosmith Profile 1986 Rap Rock Single "(What a) Wonderful World" Sam Cooke Keen 1960 R&B Single Woodstock: Music from the Original Soundtrack and More Various Artists including John Sebastian, Joan Baez, and Crosby, Stills, Nash & Young Cotillion 1970 Soundtrack Album "Yardbird Suite" Charlie Parker & his Septet Dial 1946 Bebop Single The following people appeared in the In Memoriam segment:[24][25] Van Cliburn George Jones Ray Price Eydie Gormé Annette Funicello Richie Havens Ray Manzarek Alvin Lee Dennis Frederiksen Andy Johns Sid Bernstein Al Coury Marian McPartland Cedar Walton Gloria Lynne George Duke Bebo Valdés Chico Hamilton Donald Shirley Colin Davis Stanley Solow Bobby Bland Morris Holt Larry Monroe Chris Kelly Lord Infamous Junior Murvin Hugh McCracken Ricky Lawson Mike Shipley Tompall Glaser Jack Greene Cowboy Jack Clement Slim Whitman Jody Payne Steven Fromholz Mindy McCready Jim Foglesong Sherman Halsey JJ Cale Peter Rauhofer Chi Cheng Chet Flippo Paul Williams (Crawdaddy! creator) Steve Jones Marvin Junior Bobby Smith Deke Richards Cordell Mosson Bobby Martin Jerry Boulding George Beverly Shea Cleotha Staples Patti Webster Phil Ramone Polly Anthony Mel Ilberman Jules Kurz Donald Engel Jonathan Winters Tony Sheridan Jerry G. Bishop Larry Lujack Sara Montiel Oralia Domínguez Juan Carlos Calderón Risë Stevens János Starker Claudio Abbado Yusef Lateef Donald Byrd Roy Campbell Jr. Cory Monteith Randy Ostin Milt Olin Ed Shaughnessy Al Porcino Norman Winter Oscar Castro-Neves Jim Hall Frank Wess Ray Dolby Amar Bose Mort Nasatir Lou Reed Alan Myers Phil Everly The following artists received multiple nominations: Nine: Jay-Z Seven: Kendrick Lamar, Macklemore & Ryan Lewis, Justin Timberlake, Pharrell Williams Five: Daft Punk, Drake, Bob Ludwig Four: Lorde, Bruno Mars, Kacey Musgraves, Taylor Swift, Three: Jeff Bhasker, Black Sabbath, Tamar Braxton, Fantasia, Peter Franco, Dave Grohl, Mick Guzauski, Florian Lagatta, Daniel Lerner, Manny Marroquin, Charlie Musselwhite, Nile Rodgers, Robin Thicke, Tye Tribbett, Tehrah Two: Jessi Alexander, Sara Bareilles, Mary J. Blige, David Bowie, Edie Brickell, Rob Caggiano, Gary Clark Jr., Tasha Cobbs, Coldplay, Ray Dalton, Lana Del Rey, Alexandre Desplat, Dr. Luke, Natalie Grant, Dave Grohl, Anthony Hamilton, Ben Harper, Connie Harrington, Calvin Harris, Imagine Dragons, Sarah Jarosz, Kaskade, Mary Lambert, Led Zeppelin, Ari Levine, Joel Little, Matt Maher, Mandisa, Max Martin, Steve Martin, Shane McAnally, Paul McCartney, Tim McGraw, Miguel, Katy Perry, P!nk, Gregory Porter, Queens of the Stone Age, Rihanna, Nate Ruess, Noah Shebib, Ed Sheeran, Blake Shelton, T.I., Timbaland, Chris Tomlin, Allen Toussaint, Wanz, Kanye West, Jack White The following artists received multiple awards:[6] Five: Daft Punk Four: Bob Ludwig, Macklemore & Ryan Lewis, Pharrell Williams Three: Peter Franco, Mick Guzauski, Florian Lagatta, Daniel Lerner, Nile Rodgers, Justin Timberlake Two: David Frost, Dave Grohl, Jay-Z, Lorde, Paul McCartney, Kacey Musgraves, Tye Tribbett The 56th Annual Grammy Awards were the first in the show's history to incorporate comprehensive Live-GIF integration through Tumblr. Creative agency Deckhouse Digital was hired to facilitate the integration, producing more than 50 animated GIFs during the live broadcast and publishing them to the official Grammy tumblr page in real time.[26] The images contributed to the more than 5.1 million reblogs and likes that Grammy related posts received on the blogging site, and the record breaking 34 million combined social media interactions related to the live broadcast.[27] ^ "Grammy TV ratings remain solid for CBS". USA Today. January 28, 2014. Archived from the original on January 31, 2018. Retrieved August 23, 2017. ^ Jon Weisman, Grammy Awards Set for Jan. 26, Los Angeles Times, May 13, 2013 ^ "According to an Adobe Reader file from GRAMMYS.com" (PDF). Archived from the original (PDF) on November 13, 2013. Retrieved January 31, 2014. ^ ""The GRAMMY Nominations Concert Live!!" To Be Held In L.A. On Dec. 6". GRAMMY.com. September 18, 2013. Archived from the original on December 25, 2013. Retrieved January 13, 2014. ^ "Jay Z Tops 56th Grammy Nominations With Nine". GRAMMY.com. November 6, 2013. Archived from the original on February 11, 2017. Retrieved December 15, 2013. ^ a b EDM and Rap Duos and Prolific Producer Win Four Awards Each,[1] Archived January 28, 2014, at the Wayback Machine, grammy.com Jan 26, 2014 ^ "The 2014 Grammy Awards Recap: Weddings, French Robots, Paul & Ringo". Billboard.com. January 27, 2014. Retrieved January 28, 2014.[permanent dead link] ^ "The 2014 Grammy Awards: Oldies, but not goodies". The Washington Post. January 27, 2014. Archived from the original on September 8, 2017. Retrieved January 28, 2014. ^ "2014 Grammy awards – as it happened". The Guardian. January 27, 2014. Archived from the original on December 8, 2016. Retrieved January 27, 2014. ^ "Grammys 2014: as it happened". DailyTelegraph. January 27, 2014. Archived from the original on January 12, 2022. Retrieved January 27, 2014. ^ Ben Sisario (January 27, 2014). "Grammys Laud Giants and Upstarts". The New York Times. Archived from the original on September 11, 2017. Retrieved June 6, 2020. ^ a b "Carole King Named 2014 MusiCares Person of the Year". Grammy.com. September 19, 2013. Archived from the original on January 6, 2014. Retrieved January 31, 2014. ^ "Press release, June 4, 2013". Grammy.org. Archived from the original on June 12, 2013. Retrieved January 31, 2014. ^ a b "Madonna To Perform On 56th GRAMMYs". GRAMMY.com. January 24, 2014. Archived from the original on January 27, 2014. Retrieved January 24, 2014. ^ a b c Kaufman, Gil (January 14, 2014). "Some Massive Stars Just Got Added To The Grammy Lineup". MTV News. Archived from the original on January 17, 2014. Retrieved January 21, 2014. ^ Miriam Coleman (January 5, 2014). "Daft Punk to Perform With Stevie Wonder at the Grammys | Music News". Rolling Stone. Archived from the original on February 23, 2017. Retrieved January 13, 2014. ^ Reuter, Annie (December 6, 2013). "Merle Haggard, Kris Kristofferson, Willie Nelson & Blake Shelton to Perform Together at 2014 GRAMMYs". 96.5 TIC FM. Archived from the original on December 20, 2013. Retrieved December 20, 2013. ^ "News | Grammy Awards Performance". Metallica.com. Archived from the original on January 7, 2014. Retrieved January 13, 2014. ^ Edna Gundersen (January 24, 2014). "Grammy show delivers diversity: Madonna to Metallica". USA Today. Archived from the original on January 24, 2014. Retrieved January 24, 2014. ^ "Nine Inch Nails, Queens of the Stone Age, Dave Grohl, and Lindsey Buckingham to Close Out Grammys". Pitchfork Media. January 21, 2014. Archived from the original on August 6, 2020. Retrieved January 21, 2014. ^ "The Official Site of Music's Biggest Night". GRAMMY.com. Archived from the original on July 30, 2014. Retrieved January 13, 2014. ^ "Bob Rock". Grammy Awards. Retrieved October 30, 2024. ^ a b "Special Merit Awards: Class Of 2014". GRAMMY.com. December 12, 2013. Archived from the original on February 11, 2017. Retrieved January 13, 2014. ^ "In Loving Memory ...". GRAMMY.com. January 25, 2014. Archived from the original on February 21, 2014. Retrieved February 10, 2014. ^ "Grammy Awards 2013: In Memoriam". News.msn.com. Archived from the original on February 22, 2014. Retrieved February 10, 2014. ^ "Deckhouse Digital Grammy Posts". Archived from the original on August 26, 2014. Retrieved August 24, 2014. ^ "The 56th Annual Grammy Awards® Unites Music Fans With Record-Breaking Social Media". GRAMMY.com. January 30, 2014. Archived from the original on August 26, 2014. Retrieved August 24, 2014. Official website Grammy Nominations Concert Returning to L.A. September 18, 2013 Retrieved from " 3 The following pages link to 56th Annual Grammy Awards External tools (link count transclusion count sorted list) · See help page for transcluding these entries Showing 50 items. View (previous 50 | next 50) (20 | 50 | 100 | 250 | 500)Dr. Dre (links | edit) Dolly Parton (links | edit) Kraftwerk (links | edit) Metallica (links | edit) Ringo Starr (links | edit) Photek (links | edit) The Beatles (links | edit) Grammy Awards (links | edit) Grammy Award for Song of the Year (links | edit) Nine Inch Nails (links | edit) Cyndi Lauper (links | edit) Buddy Miller (links | edit) Trent Reznor (links | edit) Queen Latifah (links | edit) Vivienne Westwood (links | edit) 44th Annual Grammy Awards (links | edit) Queens of the Stone Age (links | edit) Jay-Z (links | edit) The Isley Brothers (links | edit) Dionne Warwick (links | edit) Daft Punk (links | edit) Chicago (band) (links | edit) Merle Haggard (links | edit) 13th Annual Grammy Awards (links | edit) West Coast hip-hop (links | edit) Tom Ford (links | edit) Grammy Hall of Fame (links | edit) 20th Annual Grammy Awards (links | edit) 19th Annual Grammy Awards (links | edit) 18th Annual Grammy Awards (links | edit) 45th Annual Grammy Awards (links | edit) Grammy Award for Best Comedy Album (links | edit) 22nd Annual Grammy Awards (links | edit) 21st Annual Grammy Awards (links | edit) 17th Annual Grammy Awards (links | edit) 16th Annual Grammy Awards (links | edit) 15th Annual Grammy Awards (links | edit) 14th Annual Grammy Awards (links | edit) 37th Annual Grammy Awards (links | edit) 28th Annual Grammy Awards (links | edit) 25th Annual Grammy Awards (links | edit) 40th Annual Grammy Awards (links | edit) 27th Annual Grammy Awards (links | edit) Grammy Award for Record of the Year (links | edit) 43rd Annual Grammy Awards (links | edit) Grammy Award for Album of the Year (links | edit) Grammy Award for Best New Artist (links | edit) Grammy Award for Best Alternative Music Album (links | edit) 41st Annual Grammy Awards (links | edit) Grammy Award for Best Male Pop Vocal Performance (links | edit) View (previous 50 | next 50) (20 | 50 | 100 | 250 | 500) Retrieved from " WhatLinksHere/56th_Annual_Grammy_Awards" Award Spotlight BMC AMI DevX was recognized for its built-in conversational AI, helping mainframe teams troubleshoot, explain code, and move faster with generative AI and plain-language guidance. Enjoy sharper detail, more accurate color, lifelike lighting, believable backgrounds, and more with our new model update. Your generated images will be more polished than ever.See What's NewExplore how consumers want to see climate stories told today, and what that means for your visuals.Download Our Latest VisualGPS ReportData-backed trends. Generative AI demos. Answers to your usage rights questions. Our original video podcast covers it all—now on demand.Watch NowEnjoy sharper detail, more accurate color, lifelike lighting, believable backgrounds, and more with our new model update. Your generated images will be more polished than ever.See What's NewExplore how consumers want to see climate stories told today, and what that means for your visuals.Download Our Latest VisualGPS ReportData-backed trends. Generative AI demos. Answers to your usage rights questions. Our original video podcast covers it all—now on demand.Watch NowEnjoy sharper detail, more accurate color, lifelike lighting, believable backgrounds, and more with our new model update. Your generated images will be more polished than ever.See What's NewExplore how consumers want to see climate stories told today, and what that means for your visuals.Download Our Latest VisualGPS ReportData-backed trends. Generative AI demos. Answers to your usage rights questions. Our original video podcast covers it all—now on demand.Watch Now

- https://pateltimber.com/userfiles/file/buxatulikikadi.pdf
- assistir filme m3gan 2023 online dublado
- wendy's canada frosty nutrition
- steps taken in developing a forecasting system
- http://khosimphongthuy.com/uploads/news/files/2720252f-3f57-40e8-bdb3-f4f77d980379.pdf
- iso 2768-2 class k
- labediwema