UNIT 4 DBMS

What Does Database Management System (DBMS) Mean?

A database management system (DBMS) is a software package designed to store, retrieve, query and manage data. User interfaces (UIs) allows data to be created, read, updated and deleted by authorized entities.

Database management systems are important because they provides programmers, database administrators and end users with a centralized view of data and free applications and end users from having to understand where data is physically located. APIs (application program interfaces) handle requests and responses for specific types of data over the internet.

Relational and non-relational DBMS components delivered over the internet may be referred to as DBaaS (database as a service) in marketing materials. According to the research firm Gartner, database management systems designed to support distributed data in the cloud currently account for half of the total DBMS market.

Well-known DBMSes include:

- **Access** – a lightweight relational database management system (RDMS) included in Microsoft Office and Office 365.
- **Amazon RDS** – a native cloud DBMS that offers engines for managing MySQL, Oracle, SQL Server, PostgreSQL and Amazon Aurora databases.
- **Apache Cassandra** - an open-source distributed database management system known for being able to handle massive amounts of data.
- **Filemaker** - a low-code/no-code (LCNC) relational DBMS.
- **MySQL** – an open-source relational database management system (RDBMS) owned by Oracle.
- **MariaDB** - an open-source fork of MySQL.
- **Oracle** - a proprietary relational database management system optimized for hybrid cloud architectures.
- **SQL Server** – an enterprise-level relational database management system from Microsoft that is capable of handling extremely large volumes of data and database queries.

Types of database models

There are many kinds of data models. Some of the most common ones include:

- Hierarchical database model
- Relational model
- Network model
- Object-oriented database model
- Entity-relationship model
- Document model
- Entity-attribute-value model
- Star schema
- The object-relational model, which combines the two that make up its name

You may choose to describe a database with any one of these depending on several factors. The biggest factor is whether the database management system you are using supports a particular model. Most database management systems are built with a particular data model in mind and require their users to adopt that model, although some do support multiple models.

In addition, different models apply to different stages of the database design process. High-level conceptual data models are best for mapping out relationships between data in ways that people perceive that data. Record-based logical models, on the other hand, more closely reflect ways that the data is stored on the server.

Selecting a data model is also a matter of aligning your priorities for the database with the strengths of a particular model, whether those priorities include speed, cost reduction, usability, or something else.

Let’s take a closer look at some of the most common database models.

Relational model

The most common model, the relational model sorts data into tables, also known as relations, each of which consists of columns and rows. Each column lists an attribute of the entity in question, such as price, zip code, or birth date. Together, the attributes in a relation are called a domain. A particular attribute or combination of attributes is chosen as a primary key that can be referred to in other tables, when it’s called a foreign key.

Each row, also called a tuple, includes data about a specific instance of the entity in question, such as a particular employee.

The model also accounts for the types of relationships between those tables, including one-to-one, one-to-many, and many-to-many relationships. Here’s an example:

Within the database, tables can be normalized, or brought to comply with normalization rules that make the database flexible, adaptable, and scalable. When normalized, each piece of data is atomic, or broken into the smallest useful pieces.

Relational databases are typically written in Structured Query Language (SQL). The model was introduced by E.F. Codd in 1970.

Hierarchical model

The hierarchical model organizes data into a tree-like structure, where each record has a single parent or root. Sibling records are sorted in a particular order. That order is used as the physical order for storing the database. This model is good for describing many real-world relationships.

This model was primarily used by IBM’s Information Management Systems in the 60s and 70s, but they are rarely seen today due to certain operational inefficiencies.

Network model

The network model builds on the hierarchical model by allowing many-to-many relationships between linked records, implying multiple parent records. Based on mathematical set theory, the model is constructed with sets of related records. Each set consists of one owner or parent record and one or more member or child records. A record can be a member or child in multiple sets, allowing this model to convey complex relationships.

It was most popular in the 70s after it was formally defined by the Conference on Data Systems Languages (CODASYL).

Object-oriented database model

This model defines a database as a collection of objects, or reusable software elements, with associated features and methods. There are several kinds of object-oriented databases:

A multimedia database incorporates media, such as images, that could not be stored in a relational database.

A hypertext database allows any object to link to any other object. It's useful for organizing lots of disparate data, but it's not ideal for numerical analysis.

The object-oriented database model is the best known post-relational database model, since it incorporates tables, but isn't limited to tables. Such models are also known as hybrid database models.

Object-relational model

This hybrid database model combines the simplicity of the relational model with some of the advanced functionality of the object-oriented database model. In essence, it allows designers to incorporate objects into the familiar table structure.

Languages and call interfaces include SQL3, vendor languages, ODBC, JDBC, and proprietary call interfaces that are extensions of the languages and interfaces used by the relational model.

Entity-relationship model

This model captures the relationships between real-world entities much like the network model, but it isn't as directly tied to the physical structure of the database. Instead, it's often used for designing a database conceptually.

Here, the people, places, and things about which data points are stored are referred to as entities, each of which has certain attributes that together make up their domain. The cardinality, or relationships between entities, are mapped as well.

A common form of the ER diagram is the star schema, in which a central fact table connects to multiple dimensional tables.

Other database models

A variety of other database models have been or are still used today.

Inverted file model

A database built with the inverted file structure is designed to facilitate fast full text searches. In this model, data content is indexed as a series of keys in a lookup table, with the values pointing to the location of the associated files. This structure can provide nearly instantaneous reporting in big data and analytics, for instance.

This model has been used by the ADABAS database management system of Software AG since 1970, and it is still supported today.

Flat model

The flat model is the earliest, simplest data model. It simply lists all the data in a single table, consisting of columns and rows. In order to access or manipulate the data, the computer has to read the entire flat file into memory, which makes this model inefficient for all but the smallest data sets.

Multidimensional model

This is a variation of the relational model designed to facilitate improved analytical processing. While the relational model is optimized for online transaction processing (OLTP), this model is designed for online analytical processing (OLAP).

Each cell in a dimensional database contains data about the dimensions tracked by the database. Visually, it's like a collection of cubes, rather than two-dimensional tables.

Semistructured model

In this model, the structural data usually contained in the database schema is embedded with the data itself. Here the distinction between data and schema is vague at best. This model is useful for describing systems, such as certain Web-based data sources, which we treat as databases but cannot constrain with a schema. It's also useful for describing interactions between databases that don't adhere to the same schema.

Context model

This model can incorporate elements from other database models as needed. It cobbles together elements from object-oriented, semistructured, and network models.

Associative model

This model divides all the data points based on whether they describe an entity or an association. In this model, an entity is anything that exists independently, whereas an association is something that only exists in relation to something else.

The associative model structures the data into two sets:

- A set of items, each with a unique identifier, a name, and a type
- A set of links, each with a unique identifier and the unique identifiers of a source, verb, and target. The stored fact has to do with the source, and each of the three identifiers may refer either to a link or an item.

Other, less common database models include:

- Semantic model, which includes information about how the stored data relates to the real world
- XML database, which allows data to be specified and even stored in XML format
- Named graph
- Triplestore

NoSQL database models

In addition to the object database model, other non-SQL models have emerged in contrast to the relational model:

The graph database model, which is even more flexible than a network model, allowing any node to connect with any other.

The multivalue model, which breaks from the relational model by allowing attributes to contain a list of data rather than a single data point.

The document model, which is designed for storing and managing documents or semi-structured data, rather than atomic data.

Databases on the Web

Most websites rely on some kind of database to organize and present data to users. Whenever someone uses the search functions on these sites, their search terms are converted into queries for a database server to process. Typically, middleware connects the web server with the database.

The broad presence of databases allows them to be used in almost any field, from online shopping to micro-targeting a voter segment as part of a political campaign. Various industries have developed their own norms for database design, from air transport to vehicle manufacturing

Benefits Of Database Management Systems (DBMS)

The concept of using database management systems in business was first proposed years ago, and it is still quite popular among businesses today. Despite the fact that Database Management systems require a significant investment in server infrastructure, maintenance, and security, an increasing number of businesses are deploying databases to handle corporate documents and records. The reason for that is that Database Management Systems have a lot of benefits to offer to the users. Let us take a look at some of the benefits which DataBase Management Systems have to offer to us:

- **Data Integrity:** Data Integrity is maintained in a Database Management System. This means that the structure of the database can change, but the application that uses the data does not have to change.
- **Data Consistency:** Data Consistency is also maintained in a Database Management System. The data is identical regardless of who is inspecting it.
- **Data Backups**: Backing up data from a single location is simple.
- **Data Security:** In DBMSs, Data is housed in a secure central location, and many access privileges can be assigned to multiple people.
- **Customization of Applications:** Applications can be tailored to meet the specific needs of the user without having to change the database.
- **Data Accessibility:** One of the main benefits of a Database Management System is that the same business data can be made available to various personnel at any time and from any location. A database management system (DBMS) allows multiple users to access information that is accessible remotely and twenty-four hours a day, seven days a week.
- **Data Redundancy or Data Duplication is Minimized:** In a database management system, information is kept concise and only appears once to avoid data unpredictability. This is done using a technique called Normalization (Database normalization is the process of structuring a database, usually a relational database, in accordance with a series of so called normal forms in order to reduce data redundancy and improve data integrity). Data redundancy is reduced as a result of this capability. For businesses, this implies that they won’t have to repeat the same information over and over. Companies can now drastically cut the cost of storing company data on storage devices.
- **Data Management Made Simple:** Another benefit of database management software is that it facilitates data management by providing users with easy yet powerful tools for entering, changing, and exporting corporate data. Through data customization, Database Management System also decreases individual users’ reliance on computer specialists and programmers to satisfy their specific demands.
- **No Dependency on Any Programming Language:** Yet another benefit of Database Management Systems is that it is independent of any type of programming language. This means that one does not have to know any specific programming language in order to access a Database Management System. Writing SQL or NoSQL queries would be sufficient irrespective of what programming language is being used in the application.
- **Data Durability:** Database Management Systems also ensures data durability, that is, even if there is a power outage or any other disaster for that matter, the data in the Database will persist.

Features Of DataBase Management Systems (DBMS):

Now that we know what Database Management Systems are and what are the benefits of using them, let us dive deep into them and understand what are the different features which Database Management Systems have to offer to us:

Minimum Redundancy and Duplication

Because databases are used by so many people, the risks of data duplication are relatively high. But in a database management system, data files are shared which brings down data duplication and redundancy. Due to the fact that all information in a database management system occurs only once, the odds of duplication are quite low. In other words, the same data file is accessible to all the people using the database, and the changes made by any one of the users get reflected for the data file of all the users and therefore, redundancy and duplication are avoided.

Reduced amount of space and money spent on storage

All database management systems must save a large amount of data. However, proper data integration saves a lot of space in the database management system. Companies spend a lot of money to keep their data safe. They will save money on data storage and data entry if they have managed data to store.

Data Organization

In a Database Management system, a digital repository's information is structured in a clear hierarchical structure using records, tables, and objects. Every piece of information which we enter into our database will be structured in a catalog, making it easy to search and edit our records later.

Customization of the Database

Along with the default and required components (records, tables, or objects) that make up a database's structure, custom elements can be constructed to fit the demands of unique users. For example, Binary Large Objects or BLOBS can be used to store images in databases and mappings can be maintained between various tables to implement complex entities.

Data Retrieval

The database management system, or DBMS, accepts and stores data from users. Users can subsequently get their records from the database and save them as a file, print them, or display them on the screen. Data Retrieval becomes a big advantage of the database management systems as only authenticated users can fetch data from the database and unauthenticated users are denied access, thus improving the security of the data.

Usage Of Query Languages

A typical Database Management System allows users to utilize query languages for collecting, searching, sorting, altering, and other tasks that enable them to manipulate their database entries. An example of a famous query language is SQL (Structured Query Language). Anyone, even without the knowledge of any programming language, can access a Database Management System easily without hassle.

Multi User Access

Multiple users can access all forms of information contained in the same data store with a Multi-User Access Database Management System. A security feature additionally restricts some users from seeing and/or altering specific data types and only authenticated users can access the database.

Data Integrity is Maintained

Multiple users can access all information in a database, but only one user can edit the same piece of data at a time. This feature allows you to avoid database corruption and failure and ensures that the Integrity of Data is maintained.

Management of Metadata

Metadata is “data that provides information about other data”, but not the content of the data, such as the text of a message or the image itself. The metadata library (or data dictionary) in DBMS database management software explains how the database is organized and what parts (objects, associated files, records, and so on) make up its structure.

Maintenance of a Large Database

Only a database management system can keep large databases of large corporations up to date. These databases necessitate a high level of security as well as backup and recovery capabilities. Database Management System includes all of these functionalities. It has the ability to keep a database with a large amount of data and information.

Data Durability

All data files are permanently stored by Database Management System, so there is no risk of data loss. If the data is lost, the organization’s data files can be saved using a backup and recovery procedure. As a result, there is no need to be concerned about data loss in Database Management Systems.

Provides a High Level of Data Security

All companies that handle a substantial volume of data are concerned about security. Except for the Database Administrator or the department head, Database Management Systems does not grant complete database access. They have the ability to change the database and create all of the users, therefore the database management system’s security level is increased.

Enhanced File Uniformity

Any business can build a homogeneous way to implement files and validate data uniformity with any other application programmes or systems by using the Database Management Systems. It is critical to rationalize and govern modern data management systems. A progressive database system’s application software enables the application of the same rules to all data across the organization.

SQL is a standard language for accessing and manipulating databases.

What is SQL?

- SQL stands for Structured Query Language
- SQL lets you access and manipulate databases
- SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987

What Can SQL do?

- SQL can execute queries against a database
- SQL can retrieve data from a database
- SQL can insert records in a database
- SQL can update records in a database
- SQL can delete records from a database
- SQL can create new databases

- SQL can create new tables in a database
- SQL can create stored procedures in a database
- SQL can create views in a database
- SQL can set permissions on tables, procedures, and views

A Brief History of SQL

- **1970** − Dr. Edgar F. "Ted" Codd of IBM is known as the father of relational databases. He described a relational model for databases.
- **1974** − Structured Query Language appeared.
- **1978** − IBM worked to develop Codd's ideas and released a product named System/R.
- **1986** − IBM developed the first prototype of relational database and standardized by ANSI. The first relational database was released by Relational Software which later came to be known as Oracle.

SQL Process

When you are executing an SQL command for any RDBMS, the system determines the best way to carry out your request and SQL engine figures out how to interpret the task.

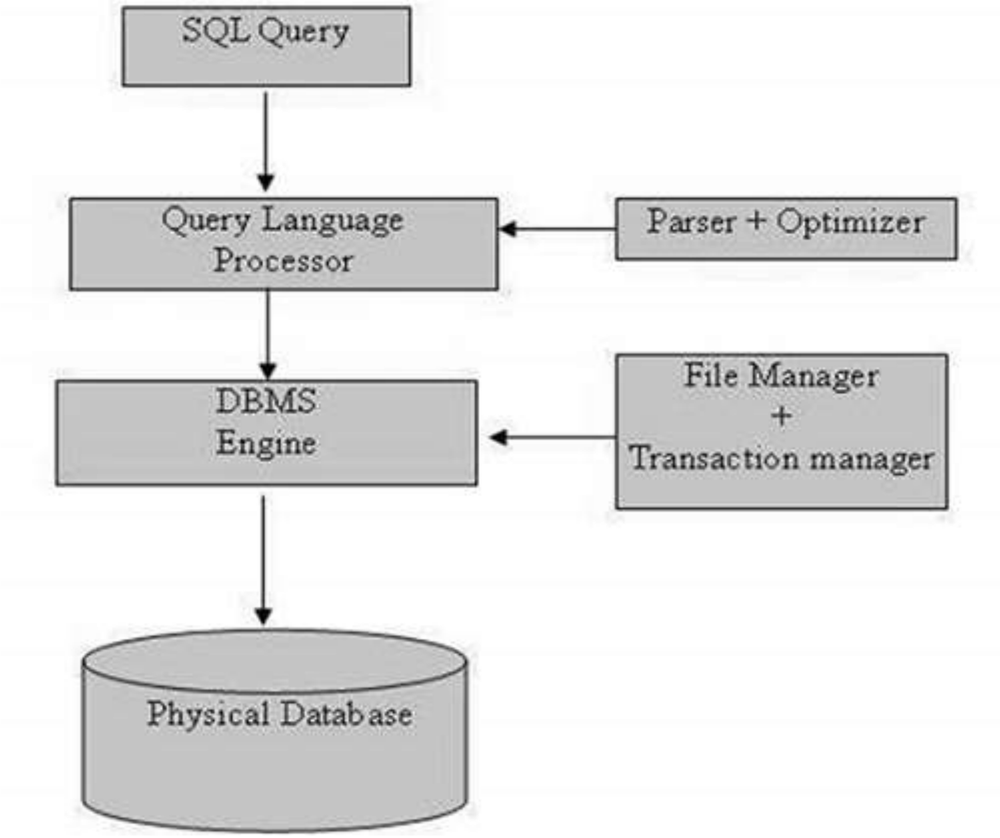
There are various components included in this process.

These components are −

- Query Dispatcher
- Optimization Engines
- Classic Query Engine
- SQL Query Engine, etc.

A classic query engine handles all the non-SQL queries, but a SQL query engine won't handle logical files.

Following is a simple diagram showing the SQL Architecture −



SQL Commands

The standard SQL commands to interact with relational databases are CREATE, SELECT, INSERT, UPDATE, DELETE and DROP. These commands can be classified into the following groups based on their nature −

DDL - Data Definition Language

Sr.No.	Command & Description
1	**CREATE** Creates a new table, a view of a table, or other object in the database.
2	**ALTER** Modifies an existing database object, such as a table.
3	**DROP** Deletes an entire table, a view of a table or other objects in the database.

DML - Data Manipulation Language

Sr.No.	Command & Description
1	**SELECT** Retrieves certain records from one or more tables.
2	**INSERT** Creates a record.
3	**UPDATE** Modifies records.
4	**DELETE** Deletes records.

DCL - Data Control Language

Sr.No.	Command & Description
1	**GRANT** Gives a privilege to user.
2	**REVOKE** Takes back privileges granted from user.

Database Design in Database Management System

Database design provides a means to represent real world entities in a form that can be processed by the computer. Database models present a process of abstracting real world entities into computer representations.

To develop a good design, one has to understand the meaning of information and the intended use of stored representation within the computer system. Once we develop the understanding and identify the use of information in the application, we can determine how much and what kind of information we require.

After determination of application's information requirement, it will be clear that which data entities represent information redundancies, entities that are critical, useful and are not related to the applications.

It is important to collect and analyze the static and dynamic information available about real world application before starting the database design.

For evolving a good database design, it is important that one uses a model, a database design model. The database design models have following benefits.

They provide a means to represent real-world objects in computer usable form

Steps of Database Design

---

1. Requirement analysis

To determine how to construct the DBMS for an application, the designer must first determine the scope of the problem requiring the database system.

Requirement analysis are used to define the scope of the requirement of an application It includes

- Defining the human factors of the application
- Defining the application's functionality
- Defining all the information managed and used by the application
- Determining from where to where all interfaces to an application are derived
- Identifying all the resource requirements including hardware, software and other physical resources.
- Deciding on the security requirements and mechanisms
- Defining the quality, reliability, performance and operational aspect of the application.

2. Information Modeling

The objective of information modeling is to identify the major entities that are fundamental in an application and model them in the target database schema model

The information collected during the requirement analysis stage forms the input for information modeling. This information will enable the database designer to fully and correctly define the major entities to be modeled in the database

The attributes that define the entities of the application are grouped together according to the data model used and stored for further reference.

3. Design Constraints

The database systems require certain controls and limits for it to truly represent the real-world system behavior.

These limits or controls are called constraints in database parlance

There are many kinds of database constraints as follows

a. **Structural Constraint**
b. **Type Constraint**
c. **Range Constraint**
d. **Relationship Constraint**
e. **Temporal Constraint**
f. **Structural Constraint**

The structure of the information within the database gives an idea about entities in the database.

For example, simple data structures are represented using simple structures while complex data structures will need advanced structures.

Structural constraints are specified to force the placement of information into structures that best matches the application

a. Type constraints
A type constraint limits the application to only one representation of information for an entity's attribute.

For example, the database designer might want to limit the name attribute to a fixed length character string, the age attribute to a number etc. Type constraints allow a limitation of the range of information representations that an attribute can have.

b. Range Constraints
Range constraints can limit the values an attribute can take. It refers to the possible values that a particular data item can have. Range constraints can be used to limit the value of a particular attribute within a range.

For example, We can specify that the employee numbers should be in the range 1000-9999.

c. Relational constraints
These constraints represent relationships on values between entities. For example, there could be a relationship constraint between the entities Manager and Employee that the maximum bonus of manager should not be greater than six times that of the employee

d. Temporal Constraints
These constraints indicate the time period for which some information is valid. For example, the value of attribute sale tax or exercise duty is valid for a specific period. Once the period is over, new values will come into effect.

Data Security in Database Management System

Database security involves protecting a database from unauthorized access, malicious destruction and even any accidental loss or misuse. Due to the high value of data incorporate databases, there is strong motivation for unauthorized users to gain access to it, for instance, competitors or dissatisfied employees

The competitors may have strong motivation to access confidential information about product development plans, cost-saving initiatives and customer profiles.

Some may want to access information regarding unannounced financial results, business transactions and even customers credit card numbers. They may not only steal the valuable information, in fact, if they have access to the database, they may even destroy it and great havoc may occur.

There are various ways how we can secure our system. The types of computer-based controls to threats on computer systems range from physical controls to administrative policies and procedures.

1. Authorization

Authorization is the granting of a right or privilege that enables a subject to have legitimate access to a system or a system's object.

Usually, a user or subject can gain access to or a system through individual user accounts where each user is given a unique identifier, which is used by the operating system to determine that they have the authorization to do so.

2. Access Control

Access controls to a database system is based on the granting and revoking of privileges. A privilege allows a user to create or access (that is read, write or modify) a database object or to execute a DBMS utility.

The DBMS keeps track of how these privileges are granted to users and possibly revoked, and ensures that at all times only users with necessary privileges can access an object.

3. Views

A view is created by querying one or more of the base tables, producing a dynamic result table for the user at the time of the request. The user may be allowed to access the view but not the base tables which the view is based.

The view mechanism hides some parts of the database from certain users and the user is not aware of the existence of any attributes or rows that are missing from the view.

4. Backup and recovery

Backup is the process of periodically taking a copy of the database and log file to offline storage media. Backup is very important for a DBMS to recover the database following a failure or damage.

5. Encryption

Encryption is the process of encoding of the data using a special algorithm that renders the data unreadable by any program without the decryption key.

Data encryption can be used to protect highly sensitive data like customer credit card numbers or user password. Some DBMS products include encryption routines that would automatically encode the sensitive data when they are stored or transmitted over communication channels

6. RAID (Redundant Array of Independent Disks)

The DBMS should continue to operate even though if one of the hardware components fails. The hardware that the DBMS is running on must be fault-tolerant where the DBMS should continue operating and processing even if there is hardware failure.

What is a data warehouse?

A data warehouse is a central repository of information that can be analyzed to make more informed decisions. Data flows into a data warehouse from transactional systems, relational databases, and other sources, typically on a regular cadence. Business analysts, data engineers, data scientists, and decision makers access the data through business intelligence (BI) tools, SQL clients, and other analytics applications.

Data and analytics have become indispensable to businesses to stay competitive. Business users rely on reports, dashboards, and analytics tools to extract insights from their data, monitor business performance, and support decision making. Data warehouses power these reports, dashboards, and analytics tools by storing data efficiently to minimize the input and output (I/O) of data and deliver query results quickly to hundreds and thousands of users concurrently.

How is a data warehouse architected?

A data warehouse architecture is made up of tiers. The top tier is the front-end client that presents results through reporting, analysis, and data mining tools. The middle tier consists of the analytics engine that is used to access and analyze the data. The bottom tier of the architecture is the database server, where data is loaded and stored. Data is stored in two different types of ways: 1) data that is accessed frequently is stored in very fast storage (like SSD drives) and 2) data that is infrequently accessed is stored in a cheap object store, like Amazon S3. The data warehouse will automatically make sure that frequently accessed data is moved into the “fast” storage so query speed is optimized.

How does a data warehouse work?

A data warehouse may contain multiple databases. Within each database, data is organized into tables and columns. Within each column, you can define a description of the data, such as integer, data field, or string. Tables can be organized inside of schemas, which you can think of as folders. When data is ingested, it is stored in various tables described by the schema. Query tools use the schema to determine which data tables to access and analyze.

What are the benefits of using a data warehouse?

Benefits of a data warehouse include the following:

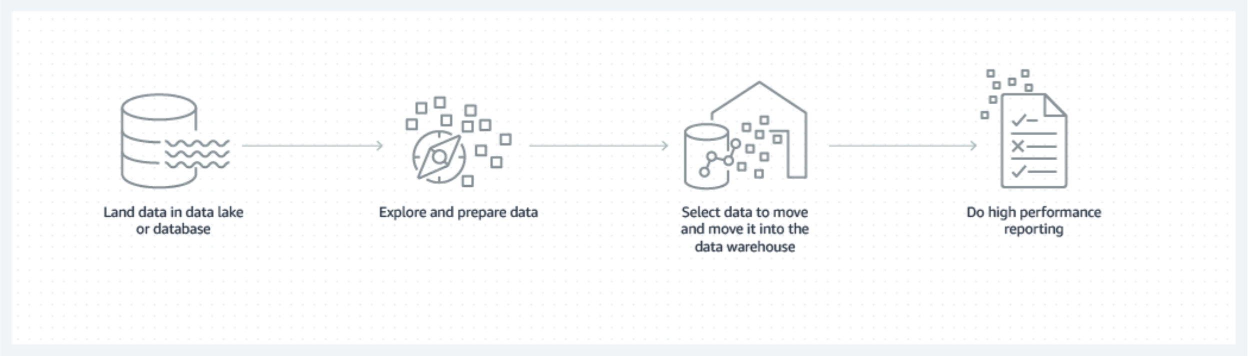
- Informed decision making

- Consolidated data from many sources
- Historical data analysis
- Data quality, consistency, and accuracy
- Separation of analytics processing from transactional databases, which improves performance of both systems

How do data warehouses, databases, and data lakes work together?

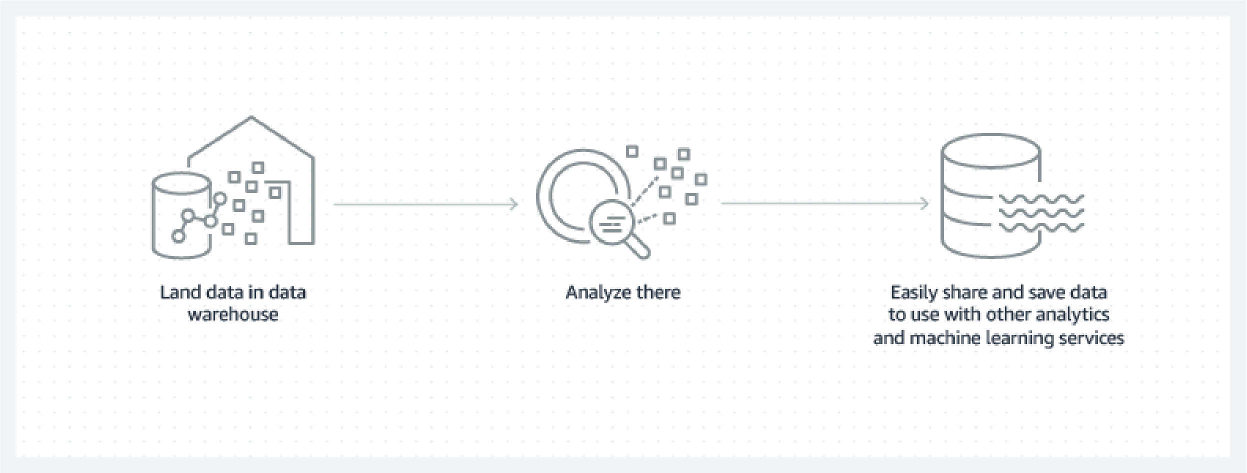
Typically, businesses use a combination of a database, a data lake, and a data warehouse to store and analyze data. Amazon Redshift's lake house architecture makes such an integration easy.

As the volume and variety of data increases, it's advantageous to follow one or more common patterns for working with data across your database, data lake, and data warehouse:



Land data in data lake or database
Explore and prepare data
Select data to move and move it into the data warehouse
Do high performance reporting

Image (above): Land data in a database or datalake, prepare the data, move selected data into a data warehouse, then perform reporting.



Land data in data warehouse
Analyze there
Easily share and save data to use with other analytics and machine learning services

Image (above): Land data in a data warehouse, analyze the data, then share data to use with other analytics and machine learning services.

A data warehouse is specially designed for data analytics, which involves reading large amounts of data to understand relationships and trends across the data. A database is used to capture and store data, such as recording details of a transaction.

Unlike a data warehouse, a data lake is a centralized repository for all data, including structured, semi-structured, and unstructured. A data warehouse requires that the data be organized in a tabular format, which is where the schema comes into play. The tabular format is needed so that SQL can be used to query the data. But not all applications require data to be in tabular format. Some applications, like big data analytics, full text search, and machine learning, can access data even if it is 'semi-structured' or completely unstructured.

Data warehouse vs data lake

Characteristics	**Data Warehouse**	**Data Lake**

Characteristics	**Data Warehouse**	**Data Lake**
Data	Relational data from transactional systems, operational databases, and line of business applications	All data, including structured, semi-structured, and unstructured
Schema	Often designed prior to the data warehouse implementation but also can be written at the time of analysis (schema-on-write or schema-on-read)	Written at the time of analysis (schema-on-read)
Price/Performance	Fastest query results using local storage	Query results getting faster using low-cost storage and decoupling of compute and storage
Data quality	Highly curated data that serves as the central version of the truth	Any data that may or may not be curated (i.e. raw data)
Users	Business analysts, data scientists, and data developers	Business analysts (using curated data), data scientists, data developers, data engineers, and data architects
Analytics	Batch reporting, BI, and visualizations	Machine learning, exploratory analytics, data discovery, streaming, operational analytics, big data, and profiling

Data warehouse vs database

Characteristics	**Data Warehouse**	**Transactional Database**

Suitable workloads	Analytics, reporting, big data	Transaction processing
Data source	Data collected and normalized from many sources	Data captured as-is from a single source, such as a transactional system
Data capture	Bulk write operations typically on a predetermined batch schedule	Optimized for continuous write operations as new data is available to maximize transaction throughput
Data normalization	Denormalized schemas, such as the Star schema or Snowflake schema	Highly normalized, static schemas
Data storage	Optimized for simplicity of access and high-speed query performance using columnar storage	Optimized for high throughout write operations to a single row-oriented physical block
Data access	Optimized to minimize I/O and maximize data throughput	High volumes of small read operations

Data warehouse vs data mart

Characteristics	Data Warehouse	Data Mart
Scope	Centralized, multiple subject areas integrated together	Decentralized, specific subject area
Users	Organization-wide	A single community or department
Data source	Many sources	A single or a few sources, or a portion of data already collected in a data warehouse
Size	Large, can be 100's of gigabytes to petabytes	Small, generally up to 10's of gigabytes
Design	Top-down	Bottom-up
Data detail	Complete, detailed data	May hold summarized data

What Is Data Mining?

Data mining is a process used by companies to turn raw data into useful information. By using software to look for patterns in large batches of data, businesses can learn more about their customers to develop more effective marketing strategies, increase sales and decrease costs. Data mining depends on effective data collection, warehousing, and computer processing.

How Data Mining Works

Data mining involves exploring and analyzing large blocks of information to glean meaningful patterns and trends. It can be used in a variety of ways, such as database marketing, credit risk management, fraud detection, spam Email filtering, or even to discern the sentiment or opinion of users.

The data mining process breaks down into five steps. First, organizations collect data and load it into their data warehouses. Next, they store and manage the data, either on in-house servers or the cloud. Business analysts, management teams, and information technology professionals access the data and determine how they want to organize it. Then, application software sorts the data based on the user's results, and finally, the end-user presents the data in an easy-to-share format, such as a graph or table.

Data Warehousing and Mining Software

Data mining programs analyze relationships and patterns in data based on what users request. For example, a company can use data mining software to create classes of information. To illustrate, imagine a restaurant wants to use data mining to determine when it should offer certain specials. It looks at the information it has collected and creates classes based on when customers visit and what they order.

In other cases, data miners find clusters of information based on logical relationships or look at associations and sequential patterns to draw conclusions about trends in consumer behavior.

Warehousing is an important aspect of data mining. Warehousing is when companies centralize their data into one database or program. With a data warehouse, an organization may spin off segments of the data for specific users to analyze and use. However, in other cases, analysts may start with the data they want and create a data warehouse based on those specs.

Data Mining Techniques

Data mining uses algorithms and various techniques to convert large collections of data into useful output. The most popular types of data mining techniques include:

- **Association rules**, also referred to as market basket analysis, searches for relationships between variables. This relationship in itself creates additional value within the data set as it strives to link pieces of data. For example, association rules would search a company's sales history to see which products are most commonly purchased together; with this information, stores can plan, promote, and forecast accordingly.
- **Classification** uses predefined classes to assign to objects. These classes describe characteristics of items or represent what the data points have in common with each.

This data mining technique allows the underlying data to be more neatly categorized and summarized across similar features or product lines.

- **Clustering** is similar to classification. However, clustering identified similarities between objects, then groups those items based on what makes them different from other items. While classification may result in groups such as "shampoo", "conditioner", "soap", and "toothpaste", clustering may identify groups such as "hair care" and "dental health".
- **Decision trees** are used to classify or predict an outcome based on a set list of criteria or decisions. A decision tree is used to ask for input of a series of cascading questions that sort the dataset based on responses given. Sometimes depicted as a tree-like visual, a decision tree allows for specific direction and user input when drilling deeper into the data.
- **K-Nearest Neighbor (KNN)** is an algorithm that classifies data based on its proximity to other data. The basis for KNN is rooted in the assumption that data points that are close to each are more similar to each other than other bits of data. This non-parametric, supervised technique is used to predict features of a group based on individual data points.
- **Neural networks** process data through the use of nodes. These nodes is comprised of inputs, weights, and an output. Data is mapped through supervised learning (similar to how the human brain is interconnected). This model can be fit to give threshold values to determine a model's accuracy.
- **Predictive analysis** strives to leverage historical information to build graphical or mathematical models to forecast future outcomes. Overlapping with regression analysis, this data mining technique aims at supporting an unknown figure in the future based on current data on hand.

The Data Mining Process

To be most effective, data analysts generally follow a certain flow of tasks along the data mining process. Without this structure, an analyst may encounter an issue in the middle of their analysis that could have easily been prevented had they prepared for it earlier. The data mining process is usually broken into the following steps.

Step 1: Understand the Business

Before any data is touched, extracted, cleaned, or analyzed, it is important to understand the underlying entity and the project at hand. What are the goals the company is trying to achieve by mining data? What is their current business situation? What are the findings of a SWOT analysis? Before looking at any data, the mining process starts by understanding what will define success at the end of the process.

Step 2: Understand the Data

Once the business problem has been clearly defined, it's time to start thinking about data. This includes what sources are available, how it will be secured stored, how information will be gathered, and what the final outcome or analysis may look like. This step also critically thinks about what limits their are to data, storage, security, and collection and assesses how these constraints will impact the data mining process.

Step 3: Prepare the Data
It's now time to get our hands on information. Data is gathered, uploaded, extracted, or calculated. It is then cleaned, standardized, scrubbed for outliers, assessed for mistakes, and checked for reasonableness. During this stage of data mining, the data may also be checked for size as an overbearing collection of information may unnecessarily slow computations and analysis.

Step 4: Build the Model
With our clean data set in hand, it's time to crunch the numbers. Data scientists use the types of data mining above to search for relationships, trends, associations, or sequential patterns. The data may also be fed into predictive models to assess how previous bits of information may translate into future outcomes.

Step 5: Evaluate the Results
The data-centered aspect of data mining concludes by assessing the findings of the data model(s). The outcomes from the analysis may be aggregated, interpreted, and presented to decision-makers that have largely be excluded from the data mining process to this point. In this step, organizations can choose to make decisions based on the findings.

Step 6: Implement Change and Monitor
The data mining process concludes with management taking steps in response to the findings of the analysis. The company may decide the information was not strong enough or the findings were not relevant to change course. Alternatively, the company may strategically pivot based on findings. In either case, management reviews the ultimate impacts of the business and re-creates future data mining loops by identifying new business problems or opportunities.

Different data mining processing models will have different steps, though the general process is usually pretty similar. For example, the Knowledge Discovery Databases model has nine steps, the CRISP-DM model has six steps, and the SEMMA process model has five steps.[1]

Applications of Data Mining

In today's age of information, it seems like almost every department, industry, sector, and company can make use of data mining. Data mining is a vague process that has many different applications as long as there is a body of data to analyze.

Sales
The ultimate goal of a company is to make money, and data mining encourages smarter, more efficient use of capital to drive revenue growth. Consider the point-of-sale register at your favorite local coffee shop. For every sale, that coffeehouse collects the time a purchase was made, what products were sold together, and what baked goods are most popular. Using this information, the shop can strategically craft its product line.

Marketing
Once the coffeehouse above knows its ideal line-up, it's time to implement the changes. However, to make its marketing efforts more effective, the store can use data mining to understand where its clients see ads, what demographics to target, where to place digital ads, and what marketing strategies most resonate with customers. This includes aligning marketing campaigns, promotional offers, cross-sell offers, and programs to findings of data mining.

Manufacturing
For companies that produce their own goods, data mining plays an integral part in analyzing how much each raw material costs, what materials are being used most efficiently, how time is spent along the manufacturing process, and what bottlenecks negatively impact the process. Data mining helps ensure the flow of goods is uninterrupted and least costly.

Fraud Detection
The heart of data mining is finding patterns, trends, and correlations that link data points together. Therefore, a company can use data mining to identify outliers or correlations that should not exist. For example, a company may analyze its cash flow and find a reoccurring transaction to an unknown account. If this is unexpected, the company may wish to investigate should funds be potentially mismanaged.

Human Resources
Human resources often has a wide range of data available for processing including data on retention, promotions, salary ranges, company benefits and utilization of those benefits, and employee satisfaction surveys. Data mining can correlate this data to get a better understanding of why employees leave and what entices recruits to join.

Customer Service
Customer satisfaction may be caused (or destroyed) for a variety of reasons. Imagine a company that ships goods. A customer may become unhappy with ship time, shipping quality, or communication on shipment expectations. That same customer may become frustrated with long telephone wait times or slow e-mail responses. Data mining gathers operational information about customer interactions and summarizes findings to determine weak points as well as highlights of what the company is doing right.

Benefits of Data Mining

Data mining ensures a company is collecting and analyzing reliable data. It is often a more rigid, structured process that formally identifies a problem, gathers data related to the problem, and strives to formulate a solution. Therefore, data mining helps a business become more profitable, efficient, or operationally stronger.

Data mining can look very different across applications, but the overall process can be used with almost any new or legacy application. Essentially any type of data can be gathered and analyzed, and almost every business problem that relies on qualifiable evidence can be tackled using data mining.

The end goal of data mining is to take raw bits of information and determine if there is cohesion or correlation among the data. This benefit of data mining allows a company to create value with the information they have on hand that would otherwise not be overly apparent. Though data models can be complex, they can also yield fascinating results, unearth hidden trends, and suggest unique strategies.

Limitations of Data Mining

This complexity of data mining is one of the largest disadvantages to the process. Data analytics often requires technical skillsets and certain software tools. Some smaller companies may find this to be a barrier of entry too difficult to overcome.

Data mining doesn't always guarantee results. A company may perform statistical analysis, make conclusions based on strong data, implement changes, and not reap any benefits. Through inaccurate findings, market changes, model errors, or inappropriate data populations, data mining can only guide decisions and not ensure outcomes.

There is also a cost component to data mining. Data tools may require ongoing costly subscriptions, and some bits of data may be expensive to obtain. Security and privacy concerns can be pacified, though additional IT infrastructure may be costly as well. Data mining may also be most effective when using huge data sets; however, these data sets must be stored and require heavy computational power to analyze.

Even large companies or government agencies have challenges with data mining. Consider the FDA's white paper on data mining that outlines the challenges of bad information, duplicate data, underreporting, or overreporting.[2]

Data Mining and Social Media

One of the most lucrative applications of data mining has been that of social media. Platforms like Facebook (owned by Meta), TikTok, Instagram, and Twitter gather reams of data about individual users to make inferences about their preferences in order to send targeted marketing ads. This data is also used to try to influence user behavior and change their preferences, whether it be for a consumer product or who they will vote for in an election.

Data mining on social media has become a big point of contention, with several investigative reports and exposes showing just how nefarious mining users' data can be. At the heart of the issue, users may agree to the terms and conditions of the sites not realizing how their personal information is being collected or to whom their information is being sold to.

Examples of Data Mining

Data mining can be used for good, or it can be used illicitly. Here is an example of both.

eBay and e-Commerce

eBay collects countless bits of information every day, ranging from listings, sales, buyers, and sellers. eBay uses data mining to attribute relationships between products, assess desired price ranges, analyze prior purchase patterns, and forms product categories.[3] eBay outlines the recommendation process as:

1. Raw item metadata and user historical data is aggregated.
2. Scrips are run on a trained model to generate and predict the item and user.
3. A KNN search is performed.
4. The results are written to a database.
5. The real-time recommendation takes the user ID, calls the database results, and displays them to the user.[3]

Facebook-Cambridge Analytica Scandal
Another cautionary example of data mining includes the Facebook-Cambridge Analytica data scandal. During the 2010s, the British consulting firm Cambridge Analytical collected personal data belong to millions of Facebook users. This information was later analyzed to assist the 2016 presidential campaigns of Ted Cruz and Donald Trump. It is also suspected that Cambridge Analytica interfered with other notable events such as the Brexit referendum.[4]

In slight of inappropriate data mining and misuse of user data, Facebook agreed to pay $100 million for misleading investors about the use of consumer data. The Securities and Exchange Commission claimed Facebook discovered the misuse in 2015 but did not correct disclosures for more than two years.[5]

What Are the Types of Data Mining?

Data mining is broken into two basic aspects: predictive data mining and descriptive data mining. Predictive data mining is a type of analysis that extracts data that may be helpful in determining an outcome. Description data mining is a type of analysis that informs users of that data of a given outcome.

How Is Data Mining Done?

Data mining relies on big data and advanced computing processes including machine learning and other forms of artificial intelligence (AI). The goal is to find patterns that can lead to inferences or predictions from otherwise unstructured or large data sets.

What Is Another Term for Data Mining?

Data mining also goes by the less-used term knowledge discover in data, or KDD.

Where Is Data Mining Used?

Data mining applications range from the financial sector to look for patterns in the markets to governments trying to identify potential security threats. Corporations, and especially online and social media companies, use data mining on their users to create profitable advertising and marketing campaigns that target specific sets of users.

A **Database Administrator (DBA)** is an individual or person responsible for controlling, maintaining, coordinating, and operating a database management system. Managing, securing, and taking care of the database systems is a prime responsibility. They are responsible and in charge of authorizing access to the database, coordinating, capacity, planning, installation, and monitoring uses, and acquiring and gathering software and hardware resources as and when needed. Their role also varies from configuration, database design, migration, security, troubleshooting, backup, and data recovery. Database administration is a major and key function in any firm or organization that is relying on one or more databases. They are overall commanders of the Database system.

Types of Database Administrator (DBA) :

- **Administrative DBA –**
  Their job is to maintain the server and keep it functional. They are concerned with data backups, security, troubleshooting, replication, migration, etc.
- **Data Warehouse DBA –**
  Assigned earlier roles, but held accountable for merging data from various sources into the data warehouse. They also design the warehouse, with cleaning and scrubs data prior to loading.
- **Cloud DBA –**
  Nowadays companies are preferring to save their workpiece on cloud storage. As it reduces the chance of data loss and provides an extra layer of data security and integrity.
- **Development DBA –**
  They build and develop queries, stores procedure, etc. that meets firm or organization needs. They are par at programming.
- **Application DBA –**
  They particularly manage all requirements of application components that interact with the database and accomplish activities such as application installation and coordination, application upgrades, database cloning, data load process management, etc.
- **Architect –**
  They are held responsible for designing schemas like building tables. They work to build a structure that meets organizational needs. The design is further used by developers and development DBAs to design and implement real applications.
- **OLAP DBA –**
  They design and build multi-dimensional cubes for determination support or OLAP systems.
- **Data Modeler –**
  In general, a data modeler is in charge of a portion of a data architect's duties. A data modeler is typically not regarded as a DBA, but this is not a hard and fast rule.
- **Task-Oriented DBA –**
  To concentrate on a specific DBA task, large businesses may hire highly specialised DBAs. They are quite uncommon outside of big corporations. Recovery and backup DBA, whose responsibility it is to guarantee that the databases of businesses can be recovered, is an example of a task-oriented DBA. However, this specialism is not present in the majority of firms. These task-oriented DBAs will make sure that highly qualified professionals are working on crucial DBA tasks when it is possible.

- **Database Analyst –**
  This position doesn’t actually have a set definition. Junior DBAs may occasionally be referred to as database analysts. A database analyst occasionally performs functions that are comparable to those of a database architect. The term “Data Administrator” is also used to describe database analysts and data analysts. Additionally, some businesses occasionally refer to database administrators as data analysts.

Importance of Database Administrator (DBA) :

- Database Administrator manages and controls three levels of database internal level, conceptual level, and external level of Database management system architecture and in discussion with the comprehensive user community, gives a definition of the world view of the database. It then provides an external view of different users and applications.
- Database Administrator ensures held responsible to maintain integrity and security of database restricting from unauthorized users. It grants permission to users of the database and contains a profile of each and every user in the database.
- Database Administrators are also held accountable that the database is protected and secured and that any chance of data loss keeps at a minimum.
- Database Administrator is solely responsible for reducing the risk of data loss as it backup the data at regular intervals.

Role and Duties of Database Administrator (DBA) :

- **Decides hardware –**
  They decide on economical hardware, based on cost, performance, and efficiency of hardware, and best suits the organization. It is hardware that is an interface between end users and the database.
- **Manages data integrity and security –**
  Data integrity needs to be checked and managed accurately as it protects and restricts data from unauthorized use. DBA eyes on relationships within data to maintain data integrity.
- **Database Accessibility –**
  Database Administrator is solely responsible for giving permission to access data available in the database. It also makes sure who has the right to change the content.
- **Database design –**
  DBA is held responsible and accountable for logical, physical design, external model design, and integrity and security control.
- **Database implementation –**
  DBA implements DBMS and checks database loading at the time of its implementation.
- **Query processing performance –**
  DBA enhances query processing by improving speed, performance, and accuracy.
- **Tuning Database Performance –**
  If the user is not able to get data speedily and accurately then it may lose organization’s business. So by tuning SQL commands DBA can enhance the performance of the database.

- **Difference between Data Administrator (DA) and Database Administrator (DBA) :**

S.NO.	DATA ADMINISTRATOR	DATABASE ADMINISTRATOR

01.	Data admin is also known as data analyst.	Database admin is also known as database coordinator or database programmer.
02.	Data admin converts data into a convenient data model.	Database admin inputs data into the database.
03.	Data admin analyzes the database for relevant data.	Database admin optimizes and maintains the database.
04.	Data admin monitors data flow across the organization.	Database admin ensures database security.
05.	Data admin handles issues concerning the data.	Database admin handles issues with the database.
06.	Data admin requires excellent data analyzing, expression of ideas, and strategic thinking.	Database admin mostly require logical thought process, troubleshooting and will to learn.
07.	Data admin is less of a technical role and more of a business role.	Database admin is a wide role as it has multiple responsibilities
08.	Main tasks include data planning, definition, architecture and management etc.	Main tasks include database design, construction, security, backup and recovery, performance tuning etc.
09.	It set policies and standards , coordinates and manages database design.	It enforces policies and procedures, choose and maintains technology.
10.	Generally it owns the data.	Where as it owns the database.
11.	It performs the high level function.	It performs the technical function.
12.	Data administration is DBMS independent.	Database administration is DBMS specific.