The DBMS Approach

A Database Management System (DBMS) is a software system designed to manage and organize databases. It provides a structured way to store, retrieve, manipulate, and administer data in a database environment. The primary goal of a DBMS is to ensure efficient data management using tools such as a data dictionary, data modelling, building the database using logical schema ensuring the integrity and security of the database.

Data Dictionary

A DBMS uses a more structured approach to the management, organisation and maintenance of data in a database. An already-defined data structure can be used to set up and create the database. The entry of new data, the storage of data, the alteration and deletion of data are all managed by the DBMS.

Data Dictionary

A data dictionary, also known as a **metadata repository,** is a centralized collection of information that stores information about the structure, organization, and characteristics of the data stored in a database. It serves as a comprehensive reference guide for understanding and managing the database. . It provides detailed descriptions of each data element, including its name, data type, format, valid values, relationships to other data elements, and any other relevant information. The use of a data dictionary improves the integrity of the data stored, helping to ensure that it is accurate, complete and consistent.

Benefits of a data dictionary

* Improved data quality: The data dictionary helps to ensure data consistency and accuracy by providing a single source of truth for data definitions.
* Enhanced communication: The data dictionary facilitates communication between developers, database administrators, and other stakeholders by providing a common understanding of the data elements.
* Reduced development time: The data dictionary can save time during development by providing readily available information about data elements.
* Increased data integrity: The data dictionary can help to prevent data errors by providing information about valid values, data types, and relationships.
* Improved documentation: The data dictionary serves as a valuable documentation tool, providing a comprehensive overview of the data used in the system.

**Data modelling** is an important tool used to show the data structure of a database. An E-R diagram is an example of a data model.

A **logical schema** is a detailed representation of the data structures and relationships in a database. It defines the entities, attributes, and relationships that will be used to store and manage data. The logical schema is independent of the specific DBMS being used and serves as a blueprint for the database design.

Key characteristics of a logical schema:

* Entity-relationship model: The schema uses an entity-relationship (ER) model to represent the entities, attributes, and relationships in the database. Entities represent real-world objects or concepts, attributes represent the characteristics of entities, and relationships define the associations between entities.
* Data types: The schema specifies the data types for each attribute, such as integer, string, date, or boolean.
* Constraints: The schema defines constraints on the data, such as primary keys, foreign keys, and data integrity rules. These constraints ensure data consistency and prevent invalid data from being entered into the database.
* Normalization: The schema is normalized to eliminate data redundancy and improve data integrity. Normalization involves breaking down complex tables into smaller, more manageable tables with well-defined relationships.

**Benefits of using a logical schema:**

* **Improved communication:** The logical schema provides a clear and concise representation of the database design, which facilitates communication between developers, database administrators, and other stakeholders.
* **Reduced development time:** By defining the data structures and relationships upfront, the logical schema reduces the time and effort required to develop the physical database.
* **Increased data integrity:** The constraints defined in the logical schema help to ensure data consistency and prevent errors.
* **Flexibility:** The logical schema is independent of the specific DBMS, making it easier to migrate the database to a different platform if needed.

**Tools for creating logical schemas:**

* **Entity-relationship diagrams (ERDs):** Graphical representations of the entities, attributes, and relationships in the database.
* **Normalization**
* **Data modeling tools:** Software applications that allow users to create and manage logical schemas.

A DBMS helps to provide data security to prevent the unwanted alteration, corruption, deletion or sharing of data with others that have no right to access it.

Security measures taken by a DBMS can include

* using usernames and passwords to prevent unauthorised access to the database.
* using access rights to manage the actions authorised users can take, for example, users could read/write/delete, or read only, or append only.
* using **access rights** to manage the parts of the database they have access to, for example, the provisions of different views of the data for different users to allow only certain users access to some tables.
* automatic creation and scheduling of regular back-ups
* encryption of the data stored.
* automatic creation of an audit trail or activity log to record the actions taken by users of the database.

The use and purpose of DBMS software tools

*Developer interface*

A developer interface is a software tool that allows developers to interact with and manage a database management system (DBMS). It provides a graphical user interface (GUI) or command-line interface (CLI) for performing various tasks, including:

* Schema creation and modification: Defining the structure of the database, including tables, columns, data types, and relationships.
* Data manipulation: Inserting, updating, and deleting data in the database.
* Querying: Writing and executing queries to retrieve specific data from the database.
* Security management: Defining user roles and permissions to control access to the database.
* Performance monitoring: Analyzing the performance of the database and identifying bottlenecks.

Examples of developer interfaces:

MySQL Workbench

PostgreSQL pgAdmin

Microsoft SQL Server Management Studio

Oracle SQL Developer

The **developer interface** allows a developer to write queries in **structured query language (SQL)** rather than using query-by-example. These queries are then processed and executed by the **query processor**. This allows the construction of more complex queries to interrogate the database.

*Query processor*

A query processor is a software component within a DBMS that is responsible for parsing, optimizing, and executing queries. It receives a query from the developer interface or another application, analyzes its syntax and semantics, and then generates an execution plan to retrieve the requested data efficiently.

* ***Parsing:*** *Checking the syntax of the query and ensuring it conforms to the database language grammar.*
* ***Optimization:*** *Finding the most efficient way to execute the query, considering factors such as available indexes, table sizes, and query complexity.*
* ***Execution:*** *Retrieving data from the database according to the generated plan and returning the results to the user.*

The query processor takes a query written in SQL and processes it. The query processor includes a DDL interpreter, a DML compiler and a query evaluation engine. Any DDL statements are interpreted and recorded in the database’s data dictionary. DML statements are compiled into low level instructions that are executed by the query evaluation engine. The DML compiler will also optimize the query.

**Benefits of using query processors:**

* **Improved query performance:** By optimizing the execution plan, query processors can significantly reduce the time it takes to retrieve data.
* **Reduced developer effort:** Developers do not need to write complex and efficient queries themselves, as the query processor takes care of it.
* **Increased data consistency:** Query processors ensure that queries are executed correctly and return accurate results.