A collection of data, typically describing the activities of one or more related organizations.

Software designed to assist in maintaining and utilizing large collections of data.

network data model -> hierarchical data model -> relational data model + SQL

The ability to store new data types and to ask more complex queries.

A substantial layer of customizable application-oriented features on top of a DBMS.

- The system won't be able to store (or address) all the data in memory, so programs have to bring data into main memory as needed.
- Each query type will require a special program.
- Applications are forced to deal with details of concurrent access and maintaining data consistency.
- Password security mechanisms are not flexible enough for complex permissions.

- Data independence.
- Efficient data access.
- Data integrity and security.
- Data administration.
- Concurrent access and crash recovery.
- Reduced application development time.

DBMSs hide the details of data representation and storage from the application.

A DBMS can enforce integrity constraints on data inserted into the database. For example, making sure a person doesn't have a negative age. Also it can enforce access controls, giving each user a different levels of access to the data.

Centralizing data allows experienced professionals to control organization, access, and efficiency.

A DBMS schedules concurrent access to data in such a manner that users can think of data as being accessed by only one user at a time.

- The DBMS provides many common data-accessing functions.
- The DBMS provides a high-level interface.
- The DBMS is probably more robust and reliable than new code rolled out with the application for achieving the same functionality.

- If you need extremely high performance for a simple set of functions.
- If you need to manipulate data in ways not supported by the query language.

A collection of high-level data description constructs that hide many low-level storage details.

A DBMS allows a user to define the data to be stored in terms of a data model.

A more abstract, high-level data model that makes it easier for a user to come up with a good initial description of the data in an enterprise.

It's needed because a DBMS's data model, though it hides many details, is still much lower level than how a user thinks about the underlying enterprise.

... a set of records.

A description of data in terms of a data model.

... an *attribute* or a *column*.

Its name, the name of each field, and the type of each field.

Records.

Conditions that the records in a relation must satisfy.

- relational
- network
- hierarchical
- object-oriented
- object-relational

Schemas for the *conceptual* (AKA *logical*), *physical*, and *external* levels of abstraction.

It describes the stored data in terms of the data model of the DBMS.

In a relational DBMS, the conceptual schema describes all relations that are stored in the database.

It summarizes how the relations described in the conceptual schema are actually stored on secondary storage devices such as disks and tapes.

This includes determining file organization and indexes (auxiliary data structures for speeding up retrieval).

Either a collection of records or a collection of pages, **not** a string of characters as in an operating system.

It allows data access to be customized (and authorized) at the level of individual users or groups of users.

A collection of one or more views and relations from the conceptual schema.

Conceptually it is a relation, but the records in a view are not actually stored in the DBMS because they can be computed from relations already in the DBMS.

They have powerful query languages, allowing a rich class of questions to be posed easily.

- Relational calculus.
- Relational algebra.

Any *one execution* of a user program in a DBMS.

It is the basic unit of change as seen by the DBMS. Partial transactions are not allowed.

A set of rules to be followed by each transaction (and enforced by the DBMS) to ensure that, even though actions of several transactions might be interleaved, the net effect is identical to executing all transactions in the same serial order.

A lock is a mechanism used to control access to database objects.

- Shared lock - can be held by two different transactions simultaneously.
- Exclusive lock - ensures no other transaction holds *any* lock on the locked object.

A log of all writes to the database, written to *before* the action is taken on the database.

Since writes to the log occur first, the system can be brought back to consistency if the database crashes between the log write and the corresponding database change.
The log also guarantees that changes made by successfully completed transactions aren't lost either.

Periodically forcing some information to the disk to reduce the time required to recover from a crash.

- Objects read or written by a transaction must first be locked. This has a performance cost.
- For efficient log maintenance the DBMS must be able to selectively force a collection of pages in main memory to disk. OS support is not always satisfactory.
- Periodic checkpointing can reduce time needed to recover from a crash, but can decrease normal execution performance.

- Access DBMS through an external schema to produce packages that facilitate data access for end users. They use the host/data languages and software tools the DBMS vendor provides.

- Design and maintain a enterprise database. Responsible for conceptual/physical schemas, security, data availability and recovery, and database tuning.