

Chapter 1. Introduction

Def. Database-management system (DBMS).

A collection of interrelated data (database)

and a set of programs to access those.

- * Primary goal : Store & retrieve
in a convenient & efficient manner.

(1.1) Database system applications.

Remark. Although user interfaces hide details of access to databases,
it is now an essential part of almost everybody.

(1.2) Purpose of database systems

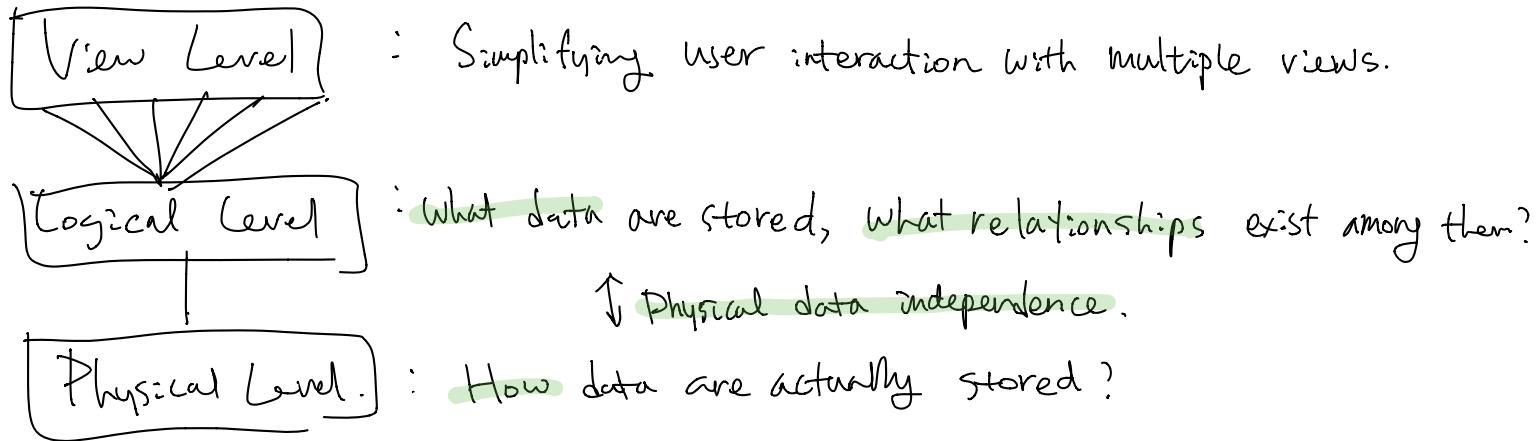
Remark. Before DBMS, file processing system, which is supported by
conventional OS, is used.

- * Disadvantages (= advantages of DBMS).
 - Data redundancy : Duplicates & different formats.
 - Data isolation : Scattered data.
 - Difficulties in accessing data : one code per function.
 - Data inconsistency : Various inconsistent copies.
 - Integrity problems : Consistency constraints.
ex) Bank balance always positive.
 - Concurrent access anomalies
 - Security problems : Different access boundaries per user.

1.3]. View of Data.

Remark. Providing **abstract** view of the data is a major purpose of DBMS.

* Three levels of Data Abstraction.



Remark. Each level has its own **schema** which describes the overall design.

Def. **Instance** of a database is a collection of information stored in the database at a particular moment.

Def. **Data model** is a collection of conceptual tools for describing data, data relationships, data semantics and consistency constraints.

- i.) **Relational Model** uses a collection of tables (**relations**) to represent both data and relationships between data.
- ii.) **Entity-Relationship (E-R) Model** uses a collection of basic objects (**entities**) and **relationship** among entities.
- iii.) **Object-based data model** extends E-R model. with notions of encapsulation, methods (functions) and object identity.
- iv.) Semistructured data model permits specification of data - individual items of same type may have different attributes. ex) XML.

I.4 Database Languages.

i) Data-Manipulation Language (DML).

: Enables access or manipulation as organized by the data model.

Namely, ① Retrieval. ② Insertion. ③ Deletion. ④ Modification.

- Procedural DML require a user to specify what data are needed and how to get those data.

- Declarative DML only require what data are needed.
So, the system has to figure out how to retrieve data efficiently.

Def. A query is a statement requesting a retrieval of info.

Query language is a portion of DML that involves retrieval of info.

ii) Data-Definition Language (DDL)

: Conveys specifications of a database schema with a set of definitions, as database has to satisfy certain consistency constraints.

ii) Assertions (consistency of data).

Special cases: Domain constraints & Referential integrity.
↳ type checking. ↳ foreign key, cascade deletion.

iii) Authorizations (consistency of user)

on read / insert / update / delete. (operations expressed by DML)

Def Data storage & definition language specifies implementation details.

Remark. Output of DDL is a metadata (data about data).

Results are stored in a special table (Data dictionary)

I.5) Relational Databases.

Remark. Record-based models

are structured in fixed-format records of several types.

Each table contains records of a particular type.

Each record type defines a fixed number of fields (attributes).

Remark. SQL is not a universal turing machine,

so it needs a host language for additional actions.

I.6) Database Design.

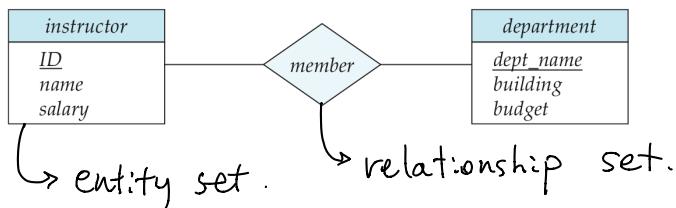
* Database design process.

Data model provides conceptual framework on requirements.

Designer translates requirements to a conceptual schema.

- What attributes we want to capture? \Rightarrow Business decision
- How to group these attributes? \Rightarrow CS problem.

* Unified Modelling Language (UML).



: A way to express E-R model graphically.

Def. Mapping cardinality is a type of constraint that expresses

number of entities to which another can be associated via a relationship set.

Def. Normalization is a process to remove redundant info.

yet allow easy retrieval of info.

(1.1) Data Storage and Querying.

Def. The **Storage manager** is a component of DBMS that provides the interface between low-level data & queries submitted.

The storage manager component includes:

- Authorizations and Integrity (assertions)
- Transaction
- File & Buffer.

which implements several data structures for physical implementation:

- Data files (data itself)
- Data dictionary (metadata)
- Indices (provides fast access).

Def. The **query processor** is a component of DBMS that simplify and facilitate access to data.

The query processor component includes:

- DDL interpreter
- DML compiler & query optimizer
- Query evaluation engine

1.8 Transaction Management.

Def. A **transaction** is a collection of operations that performs a single logical function in a database app.

Def. A **transaction manager** manages:

- **Consistency** (correctness of data)
- **Atomicity** (all-or-none operations applied)
- **Durability** (Persistence after a transaction)
- **Isolation** (among multiple transactions)

Recovery
manager

Concurrency
Control manager

1.9 Database Architecture

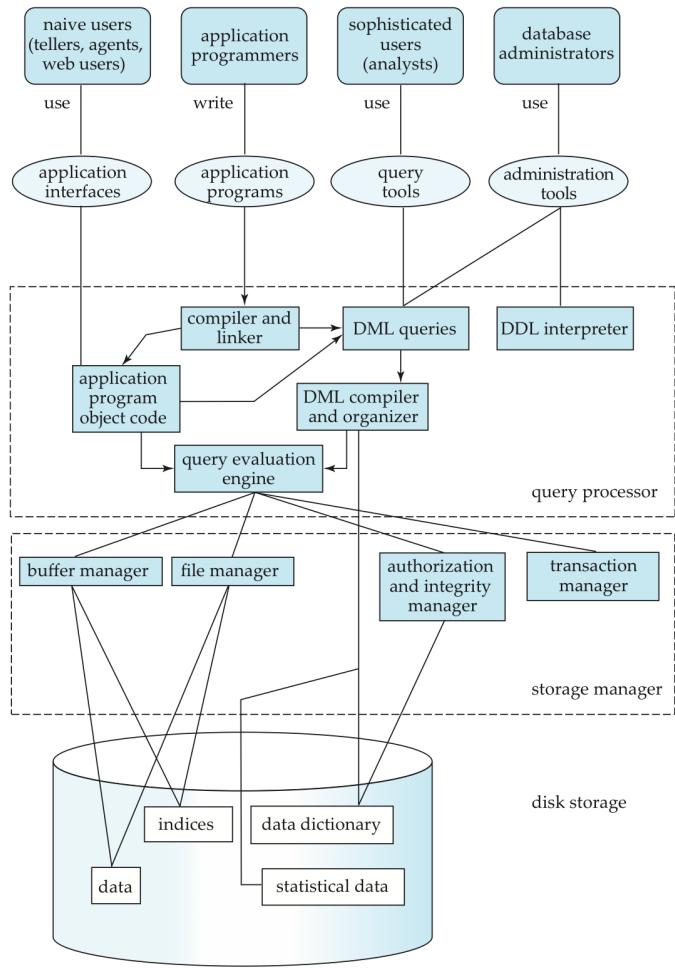


Figure 1.5 System structure.

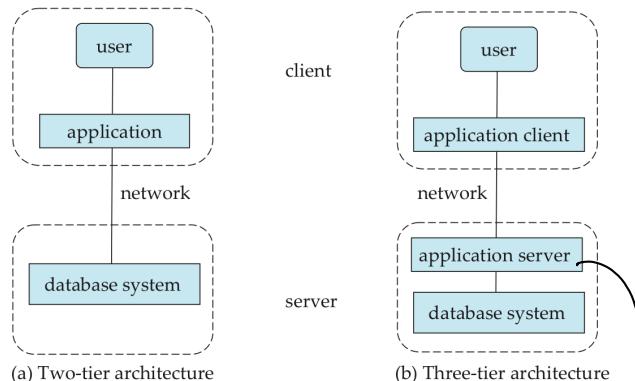


Figure 1.6 Two-tier and three-tier architectures.

*API server.
en RESTful.*

Remark. The **business logic** (What actions to do under what conditions) is embedded in the app server, making 3-tier app more appropriate for larger apps, or apps on www.