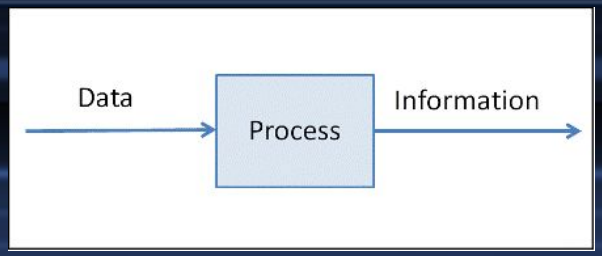
INTRODUCTION

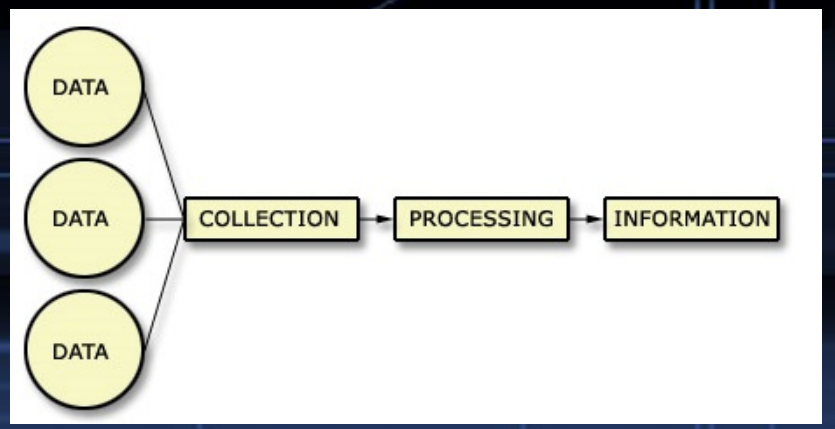
DATABASE MANAGEMENT SYSTEM

Sujan Tamrakar

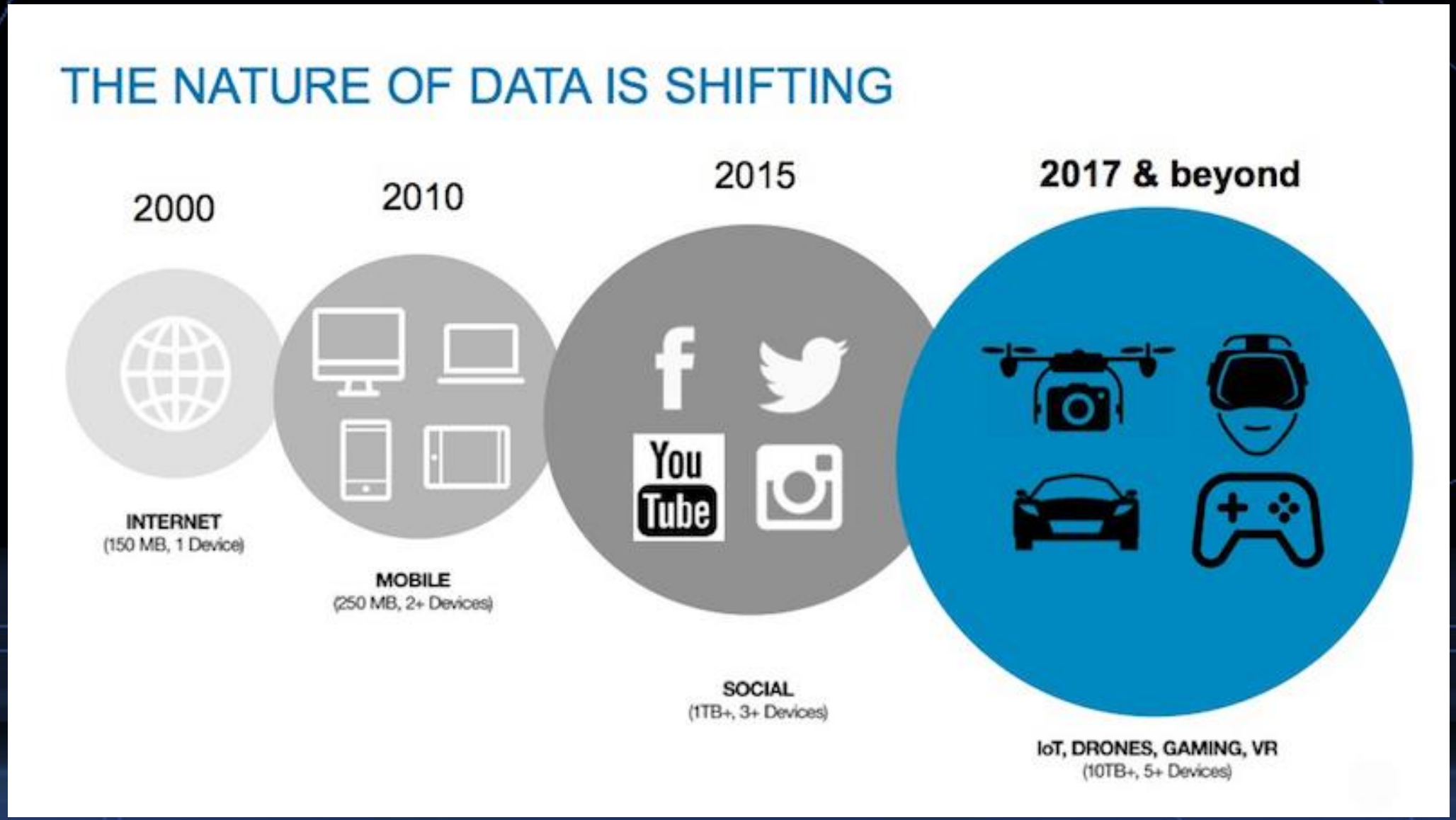
Concepts

- Data
  - Collection of facts & figures on any topic
  - May or may not give meaning
  - When processed gives information
  - Stored for future reference
  - Examples:
    - Text: numbers, scripts, any textual contents, etc.
    - Image: photographs, medical/ satellite charts, etc.
    - Audio: songs & music, speech, conversation, etc.
    - Video: documentary, live telecast, movies, etc.





Concepts


THE NATURE OF DATA IS SHIFTING
2000
2010
2015
2017 & beyond
INTERNET
(150 MB, 1 Device)
MOBILE
(250 MB, 2+ Devices)
SOCIAL
(1TB+, 3+ Devices)
IoT, DRONES, GAMING, VR
(10TB+, 5+ Devices)

United / logical

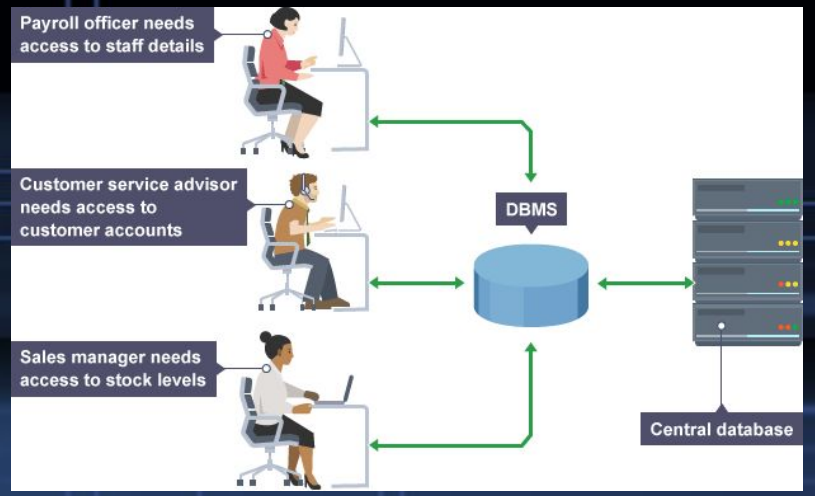
essential

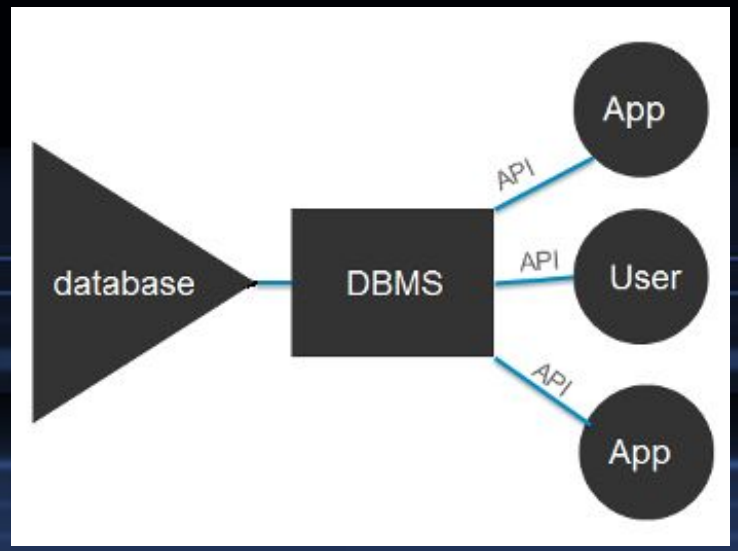
Concepts

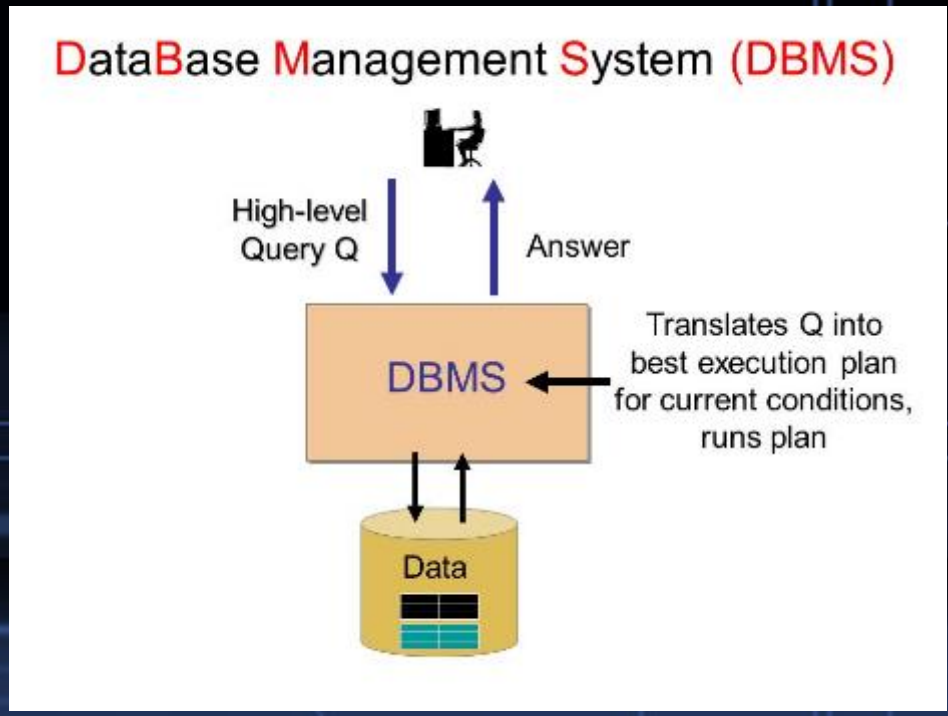
- Database
  - Collection of related data on a specific kind
  - Collection of coherent data with inherent meaning
  - Data are collected and integrated
  - Can be accessed by many users concurrently but security should be maintained
  - Used in: banks, universities, hospitals, research centres, industries, telecommunications, sales, human resources, etc.

Concepts

- Database Management System
  - Collection of programs that allows users to manipulate data in db
  - Responsible for defining, constructing, manipulating & sharing db
  - Takes request from users via application interface
  - Processes and returns results to user
  - DB + Software = DBMS



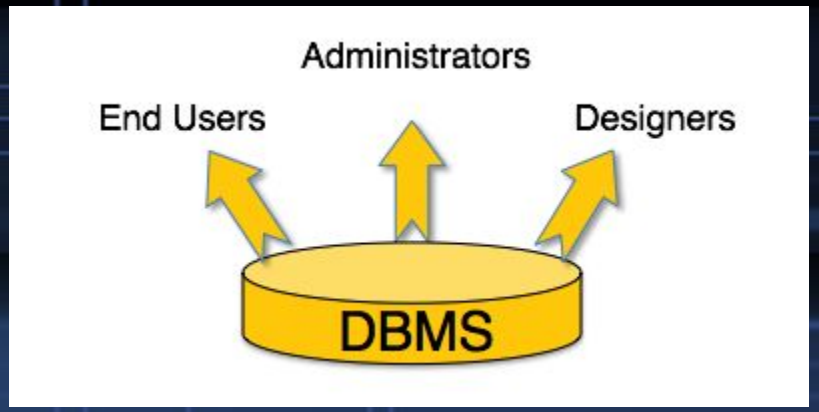




Pros | Need | Necessity | Objective

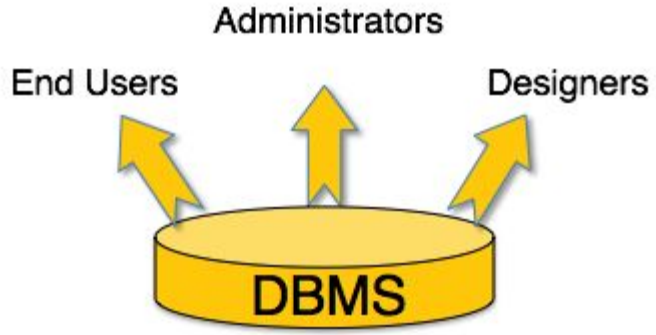
- Speed
- Compactness
- Less drudgery (mechanical work)
- Accuracy & up to date info
- Security
- Share ability
- Redundancy check
- Inconsistency check
- Backup and Recovery
- Persistence storage
- Multi user-interface

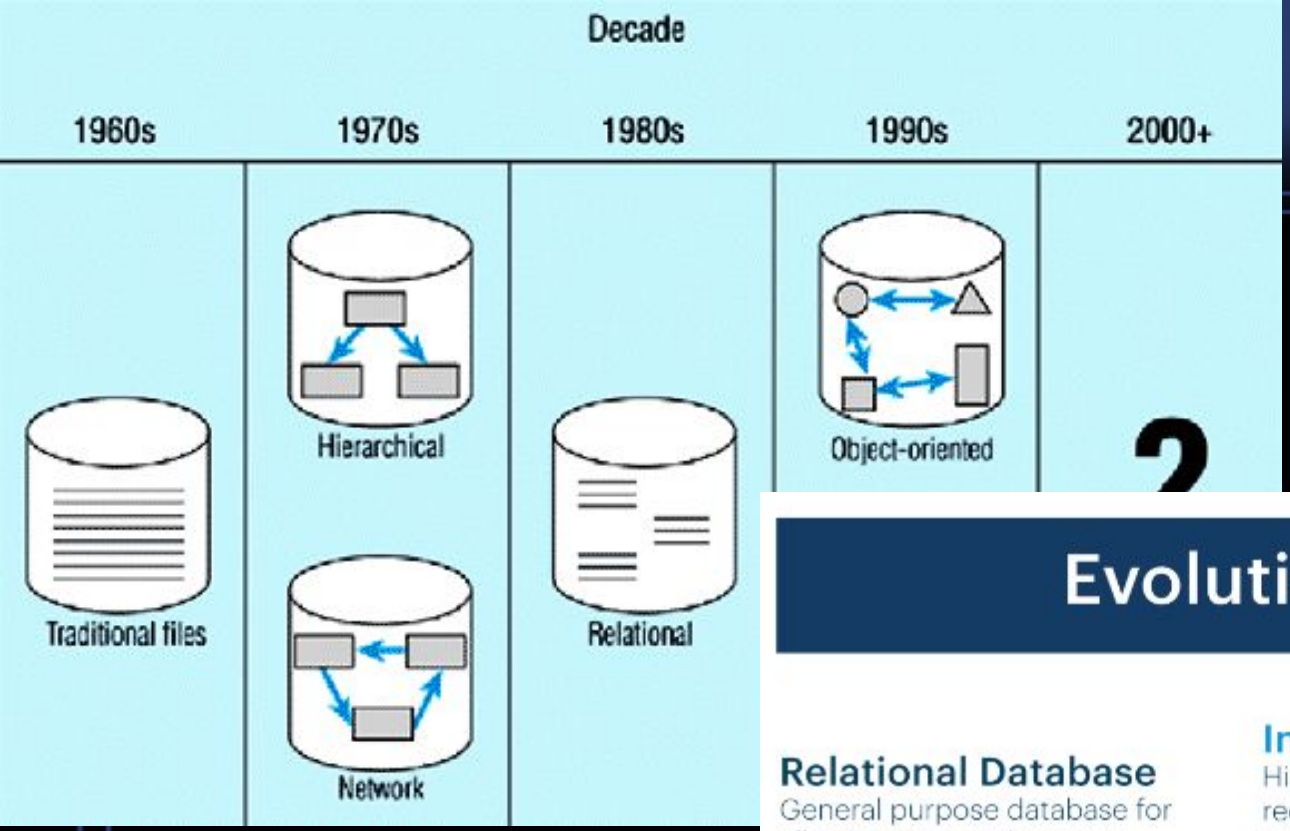
- Database Users
  - Application Programmer
    - Writes db application in various languages
  - Database Administrator
    - Administers db, dbms and related software, authorizes users to access db



- Database Users (cont…)
  - End User
    - Access the db by various queries, Uses to fulfil his goal
    - Types:
      - Casual end user: ocassionally uses db (middle/high level manager)
      - Naïve end user: frequently uses db (bank tellers/reservation counters)
      - Sophisticated end user: uses for complex requirement (scientists, analysts)
      - Standalone end user: uses personal db for simpler frequent tasks (normal layman)

Lack of experience





Evolution

Evolution of Database Technology

Relational Database
General purpose database for all enterprise applications

1970's

In-Memory Database
High volume applications required a new level of database performance

1996

Webscale (noSQL)
Internet services with billions of users created the first use case for vertical database technology

2008

2010

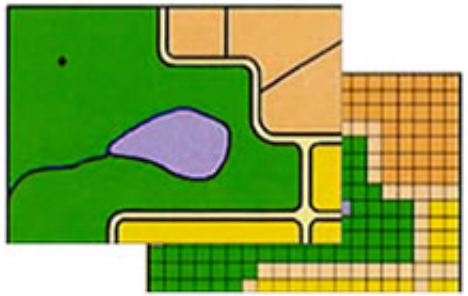
Vertical Databases
On-demand applications across multiple industries with specialized requirements (performance, scale and consistency)

Data Abstraction

- Db's records are needed to be retrieved efficiently
- Designers have used complex data structure to represent & operate on data
- All db users aren't trained & lack knowledge
- To simplify user interaction, complexity are hidden through several levels of abstraction.

Data Abstraction

Reality



Conceptual Model

- Discrete Objects
- Continuous Fields

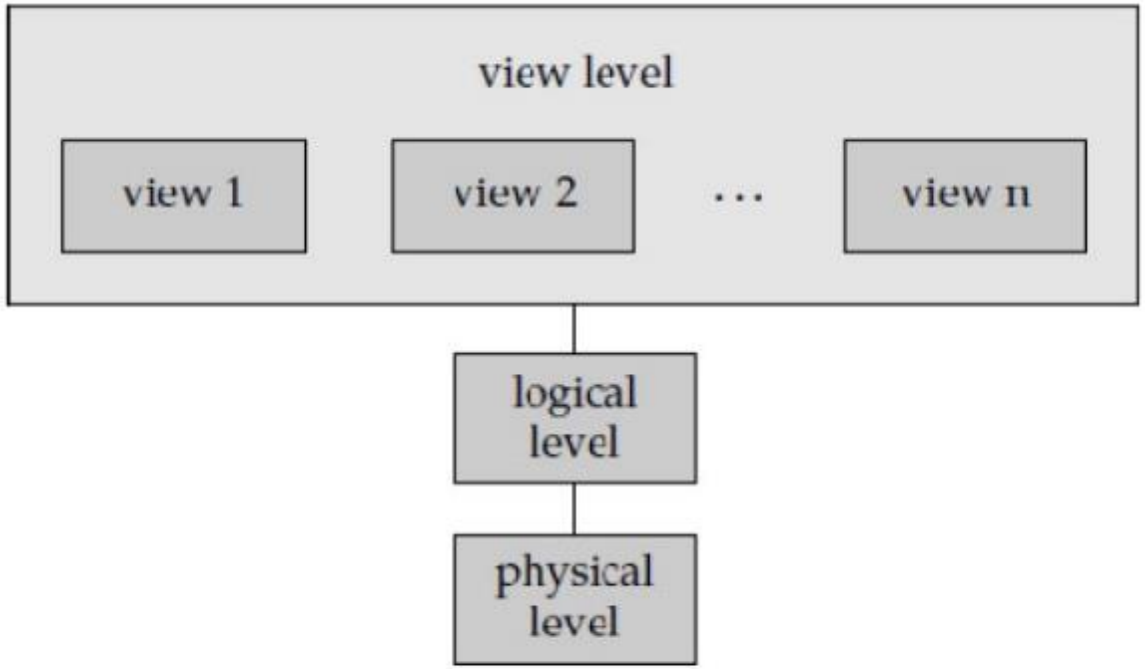
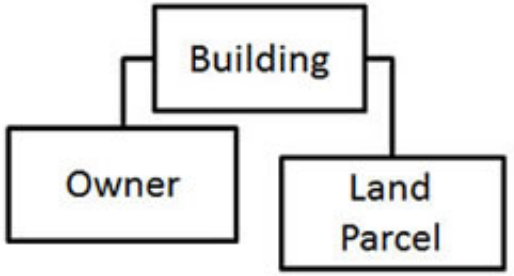


Figure 3 : Three Levels of Data Abstractions

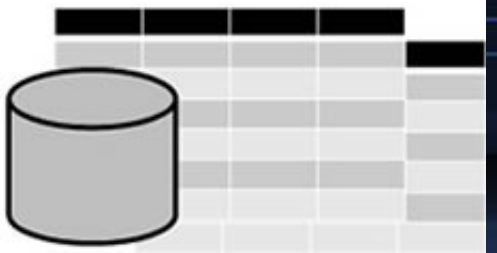
Logical Model

- Object
- Attribute
- Class



Physical Model

- Row
- Column
- Table



Data Abstraction (cont…)

- Physical Level
  - Lowest level of abstraction
  - Describes **HOW** data are actually stored
  - Describes complex low level data structure in detail

- Logical Level
  - Next higher level of abstraction
  - Describes **WHAT** data are stored in db & **WHAT** relationship exist among those data
  - Describes entire db in simpler structure (but complex in physical level)
  - Used by DBA (as he decides WHAT info to keep)

Data Abstraction (cont…)

- View Level
  - Highest level of abstraction
  - Provides interface to various users
  - Simplifies user interaction with the system
  - System may provide many views for same db (based on roles and access policy)

Data Abstraction (cont…)

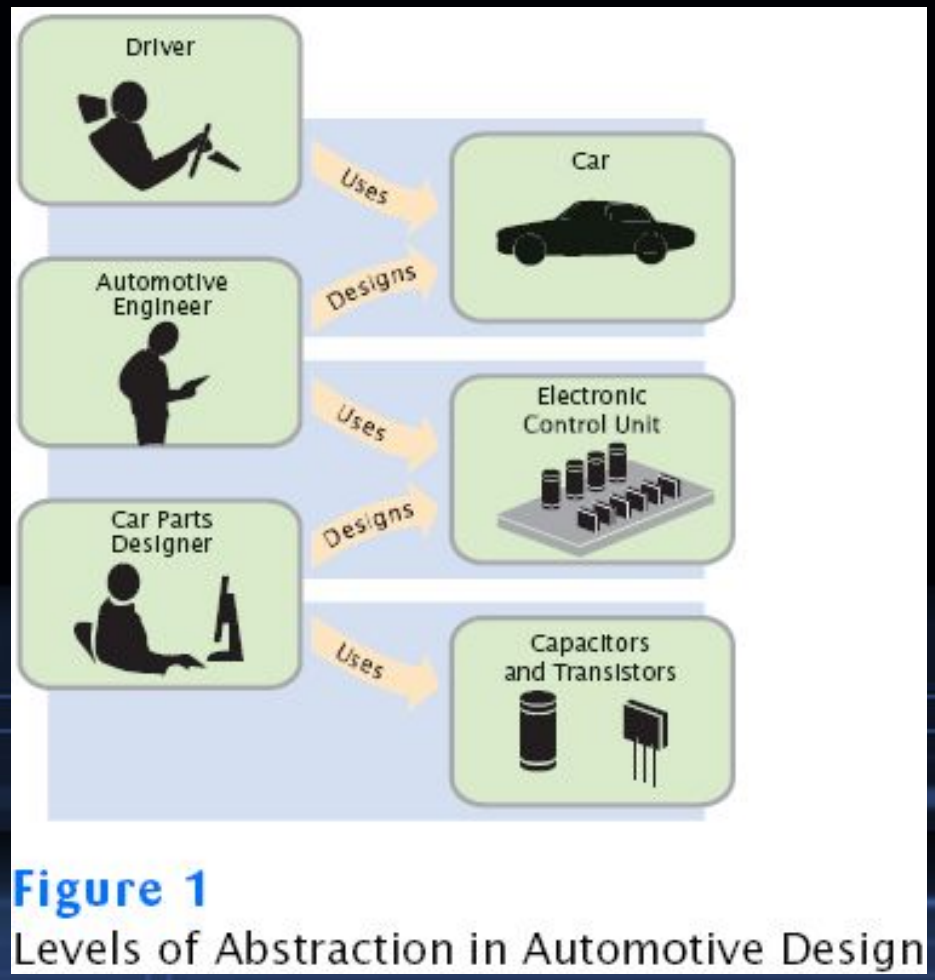
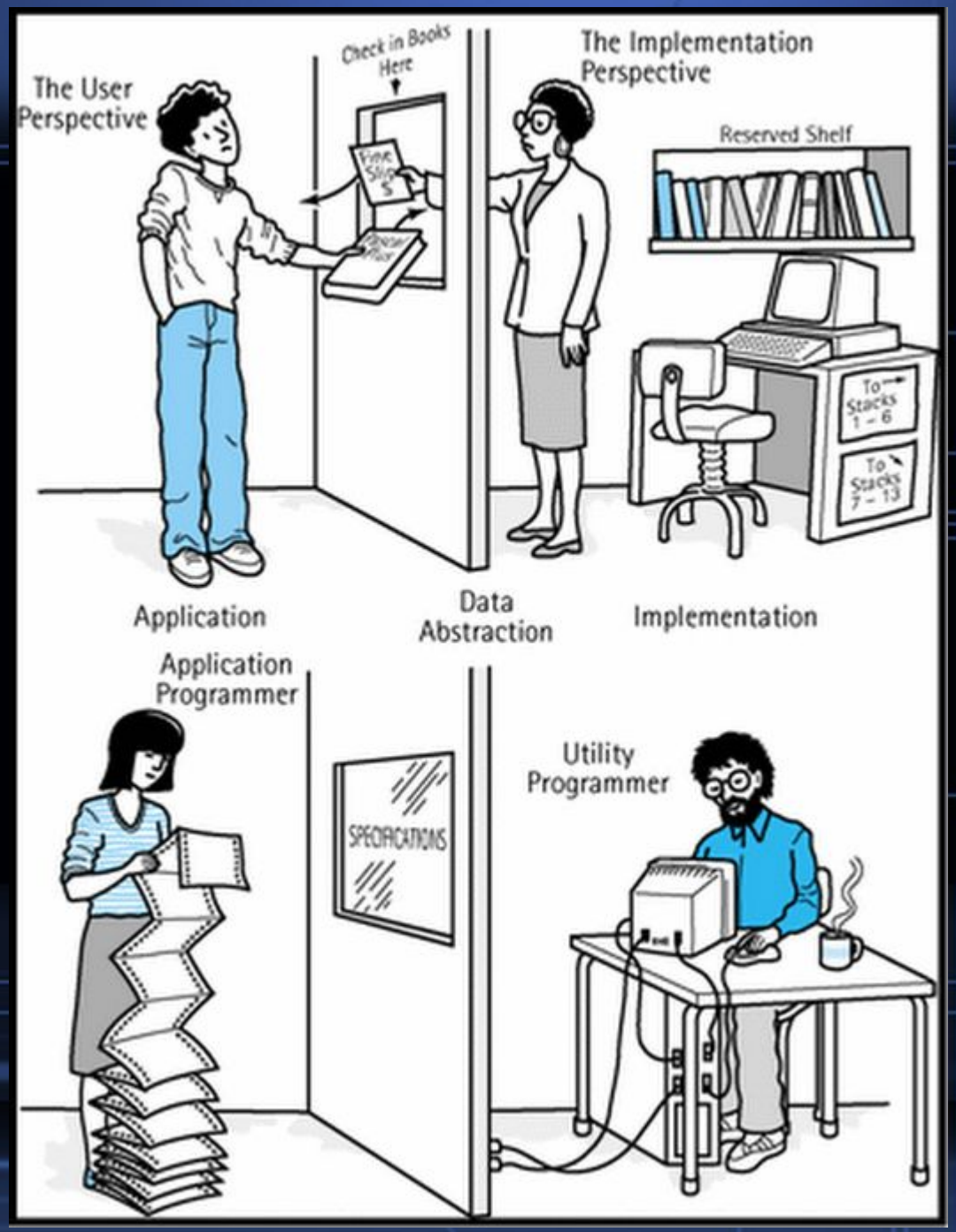


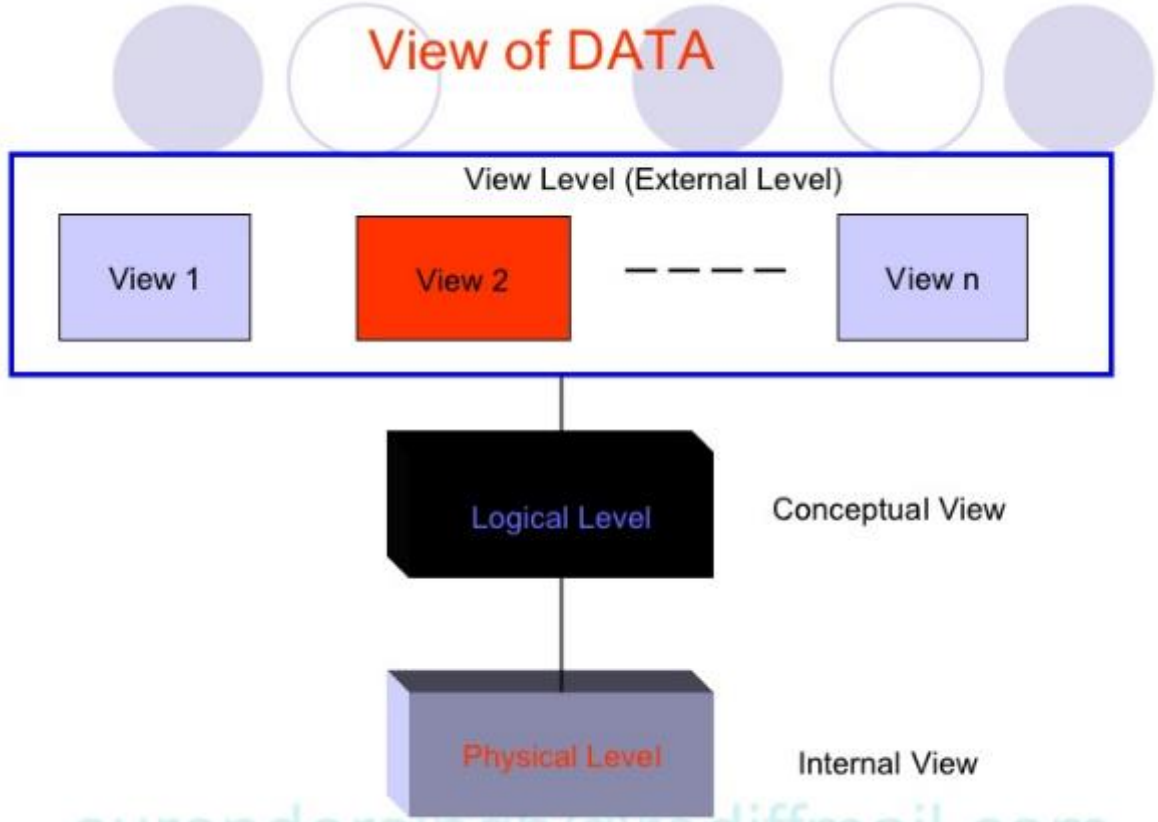
Figure 1
Levels of Abstraction in Automotive Design

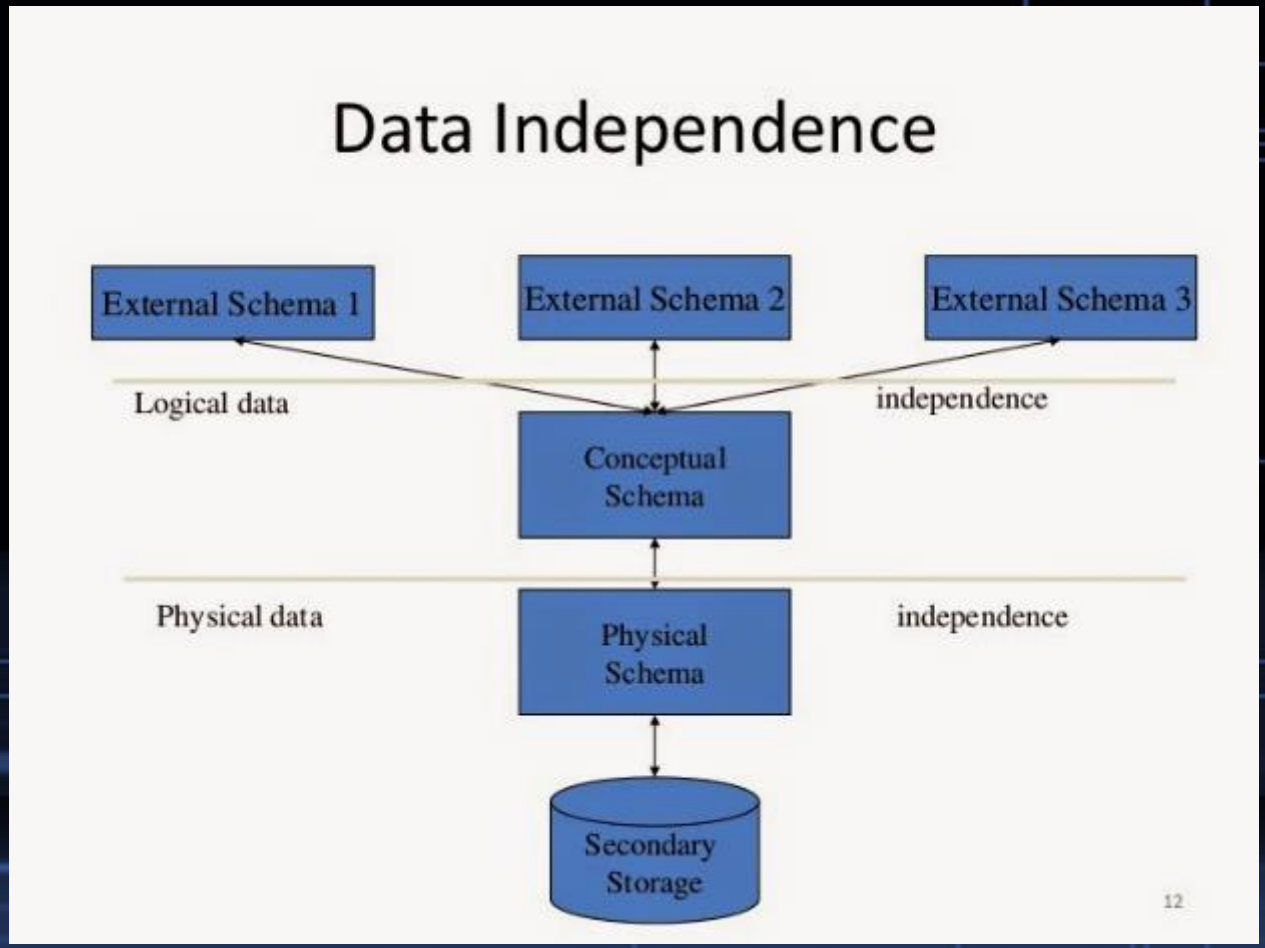


Data Independence

- Immunity of applications to change in one layer without affecting other layers.
- Capacity to change schema at one level of a DBMS without having to change schema at next higher level.
- Why?
  - Portability
  - Changing requirement (changes in any layer)
  - Dynamic nature of app

Data Independence (cont…)





Data Independence (cont…)

Types:

1. Logical Data Independence
   - Capacity to change logical level without having to change external view.
   - Occurs very often (for better adaptive product)
   - Changes in logical level like db manipulation, constraint changes, database schema changes, etc.

Data Independence (cont…)

Types:

2. Physical Data Independence
   - Capacity to change internal schema (physical level) without having to change logical level
   - Changes may be needed as some physical files may be reorganized.
   - App remains unaware of these details
   - Exists in most db as exact location of data on disk, hardware details of storage, compression, etc. are hidden from users.

Thank you !