Introduction to Database Systems

Data Intelligence and Learning (DIAL) Lab

Prof. Jongwuk Lee

Acknowledgements

- **I truly appreciate Prof. Sang-won Lee for sharing slides.**

- **I also referred to the following DB classes:**
  - CSE 344, Introduction to Database Management, University of Washington
    - https://courses.cs.washington.edu/courses/cse344/18sp/
  - CSE 444, Database Systems Internals, University of Washington
    - https://courses.cs.washington.edu/courses/cse444/17wi/
  - CSE145, Introduction to Databases, Stanford University
    - http://web.stanford.edu/class/cs145/

Pop-up Quiz

- **What is a database?**

A. A file that stores and organizes large amounts of related data

B. An application to carry out book-keeping tasks

C. A file that stores your computer’s configuration details

D. A device used to store deleted files

E. None of the above

Pop-up Quiz

- **What is a database?**

  A. A file that stores and organizes large amounts of related data

  B. An application to carry out book-keeping tasks

  C. A file that stores your computer’s configuration details

  D. A device used to store deleted files

  E. None of the above

Basic Definitions

- **Database: A collection of data**

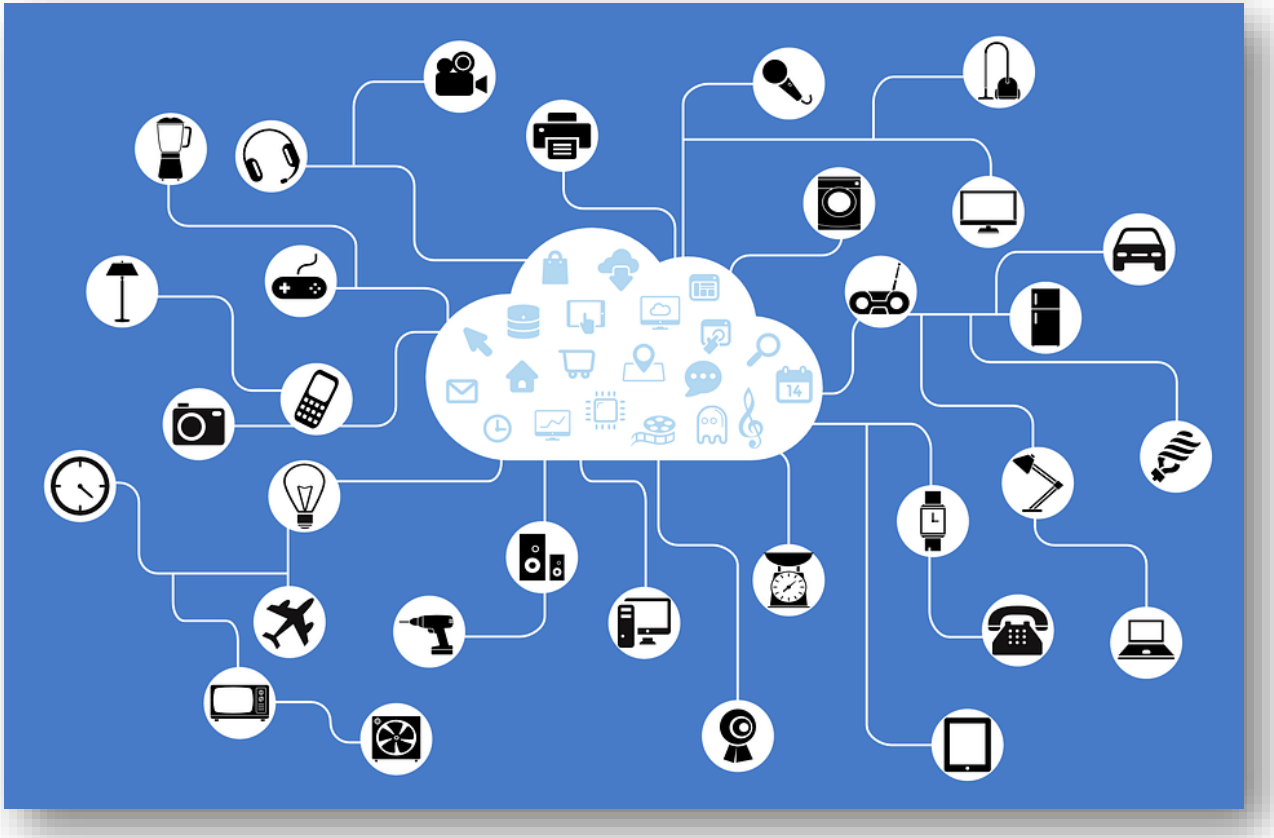
- **Data: Known facts that can be recorded with implicit meaning**

- **Database Management System (DBMS)**
  - A **software system** to facilitate the **creation** and **maintenance** of a computerized **database**

- **Database System**
  - The **DBMS software** together with the **data** itself
  - Sometimes, the **applications** are also included.

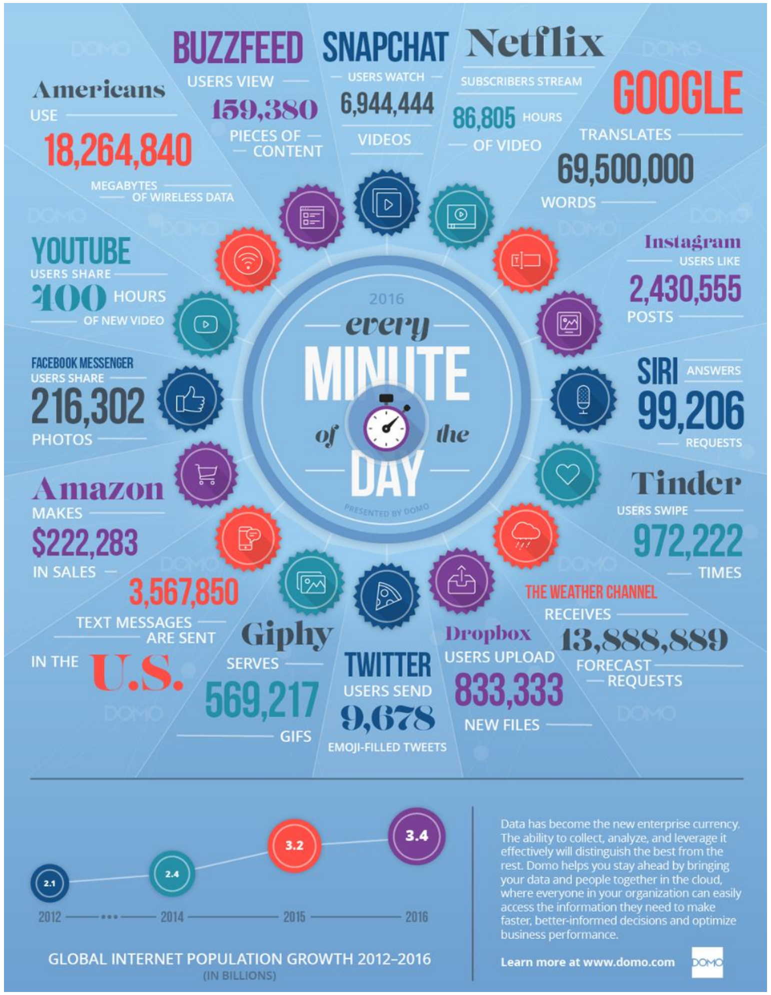
Data Exist Everywhere!

- **Major sources of abundant data**
  - **Business**: Web, e-commerce, transactions, stocks
  - **Society and everyone**: News, digital cameras, YouTube



Data Exist Everywhere!

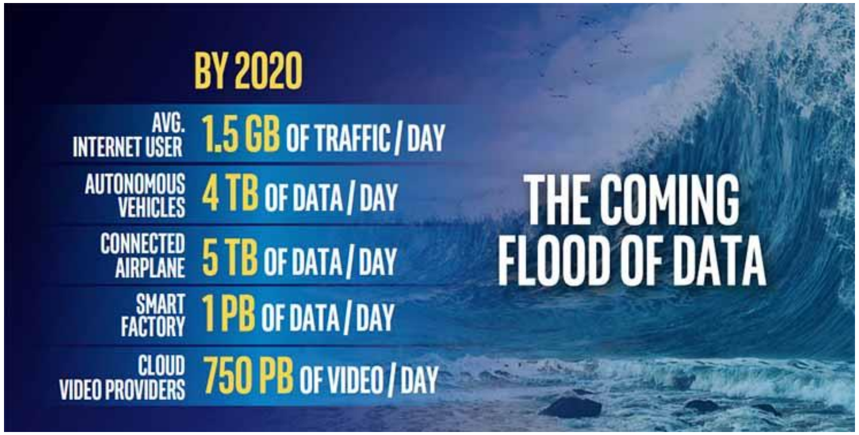
Data Never Sleeps (2016)



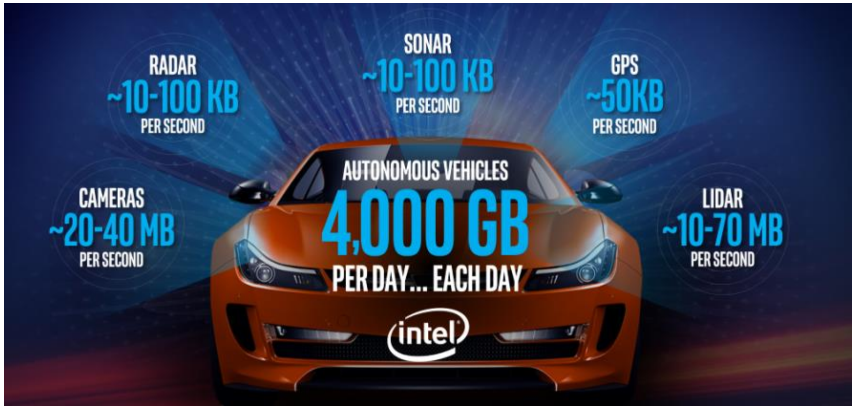
Data has become the new enterprise currency. The ability to collect, analyze, and leverage it effectively will distinguish the best from the rest. Domo helps you stay ahead by bringing your data and people together in the cloud, where everyone in your organization can easily access the information they need to make faster, better-informed decisions and optimize business performance.

Learn more at www.domo.com

Driven by IoT


BY 2020
AVG. INTERNET USER 1.5 GB OF TRAFFIC / DAY
AUTONOMOUS VEHICLES 4 TB OF DATA / DAY
CONNECTED AIRPLANE 5 TB OF DATA / DAY
SMART FACTORY 1 PB OF DATA / DAY
CLOUD VIDEO PROVIDERS 750 PB OF VIDEO / DAY
THE COMING FLOOD OF DATA

Driven by Autonomous Vehicles

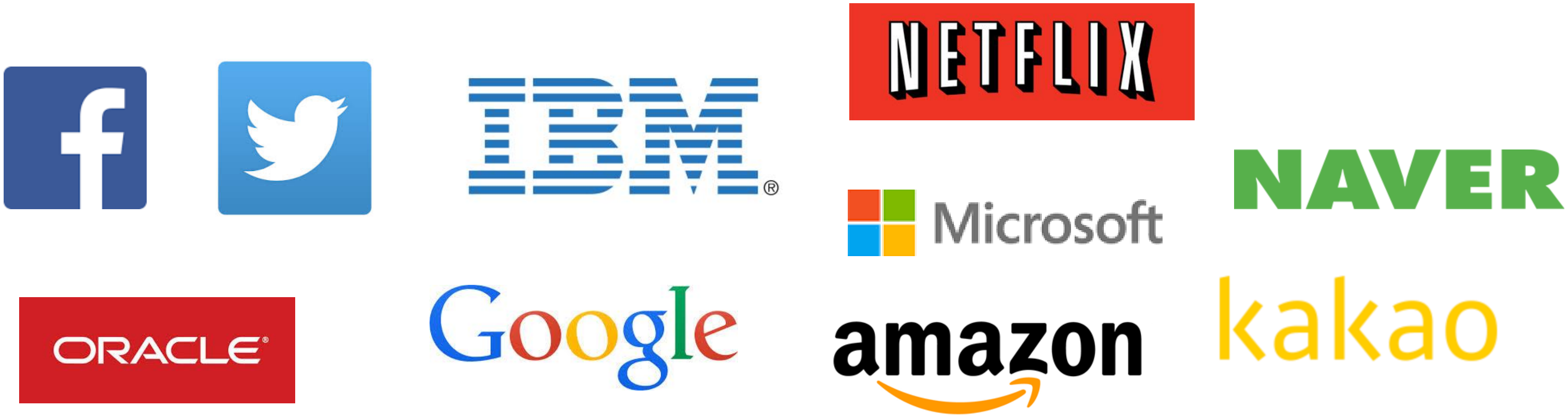


https://www.domo.com/blog/data-never-sleeps-4-0/

https://newsroom.intel.com/editorials/krzanich-the-future-of-automated-driving/#gs.y9qd6k

Data Exist Everywhere!

- **The world is drowning in data.**
  - **Big data**: **3V** = **Volume**, **Variety**, **Velocity**
  - New domains: Social networks, mobile devices, IoT, …



- Many IT companies consider themselves as **data-driven** ones.

What is a Database?

- A **database** is defined as an organized collection of data.

- Examples of databases in your daily life
  - A telephone book
  - Papers in your filing cabinet
  - Files on your computer
  - Amazon’s product database
  - SKKU’s student database

- To support **efficient data retrieval**, a collection of related data is usually compiled in a table of records.
  - **Data** vs. **Information**

What is a Database?

- **What data do we need?**
  - Data about books, customers, pending orders, order histories, trends, preferences, etc.
  - Data about sessions (clicks, pages, searches)
  - **Note: Data is large… it cannot fit all in memory!**

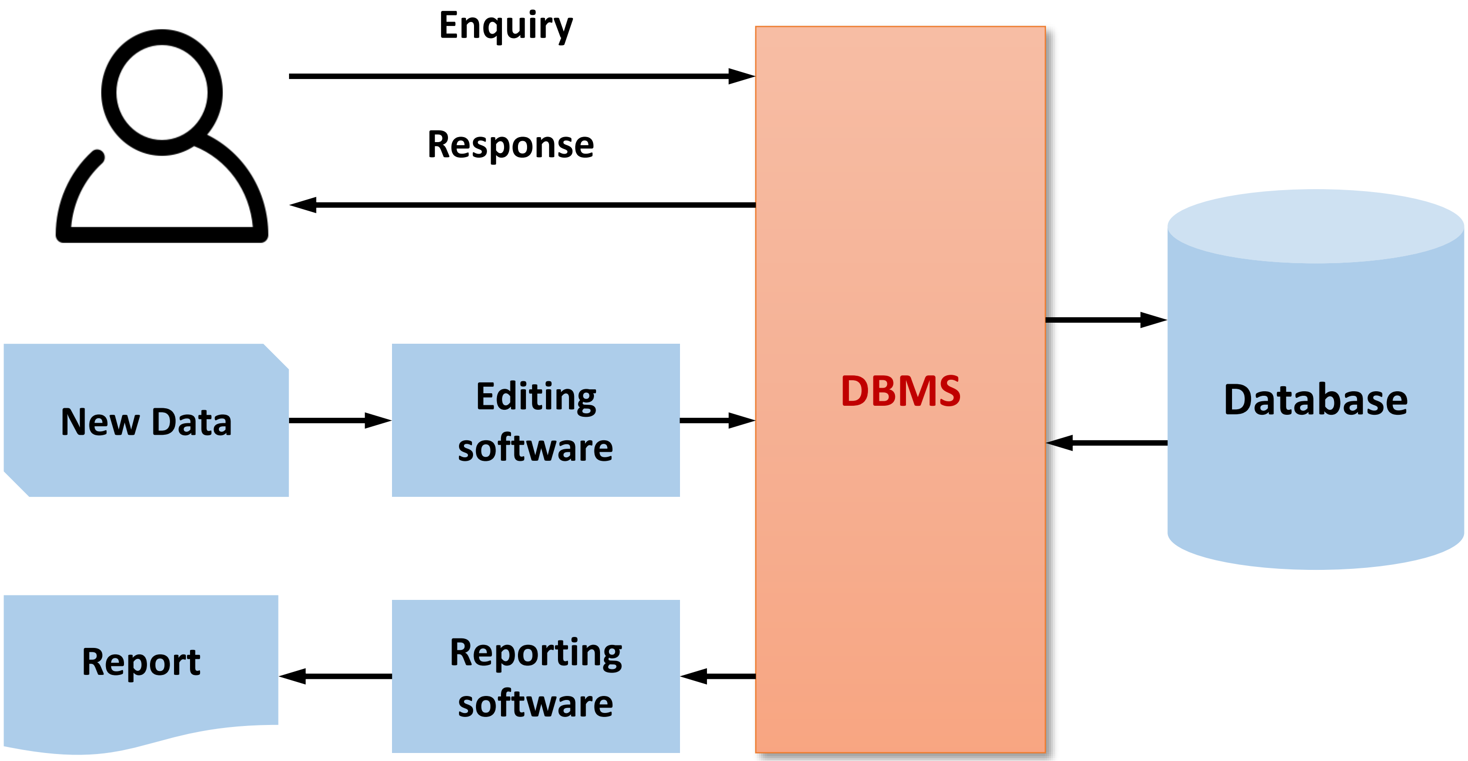
- **What capabilities on the data do we need?**
  - Insert/remove books, find books by author/title/etc., analyze past order history, recommend books, …
  - Data must be accessed efficiently, by **many users**.
    - **Concurrent access**
  - Data must be safe from failures and malicious users.
    - **Consistency**, **recovery**

What is a DBMS?

- A **database management system (DBMS)** is system software to store, maintain and retrieve data efficiently.
  - Oracle, IBM DB2, Microsoft SQL Sever
  - Postgre SQL, MySQL
  - MongoDB, MariaDB, SAP HANA, SQLite, …

What is a DBMS?

- A database management system (DBMS) is system software to store, maintain and retrieve data efficiently.



Typical DBMS Functionality

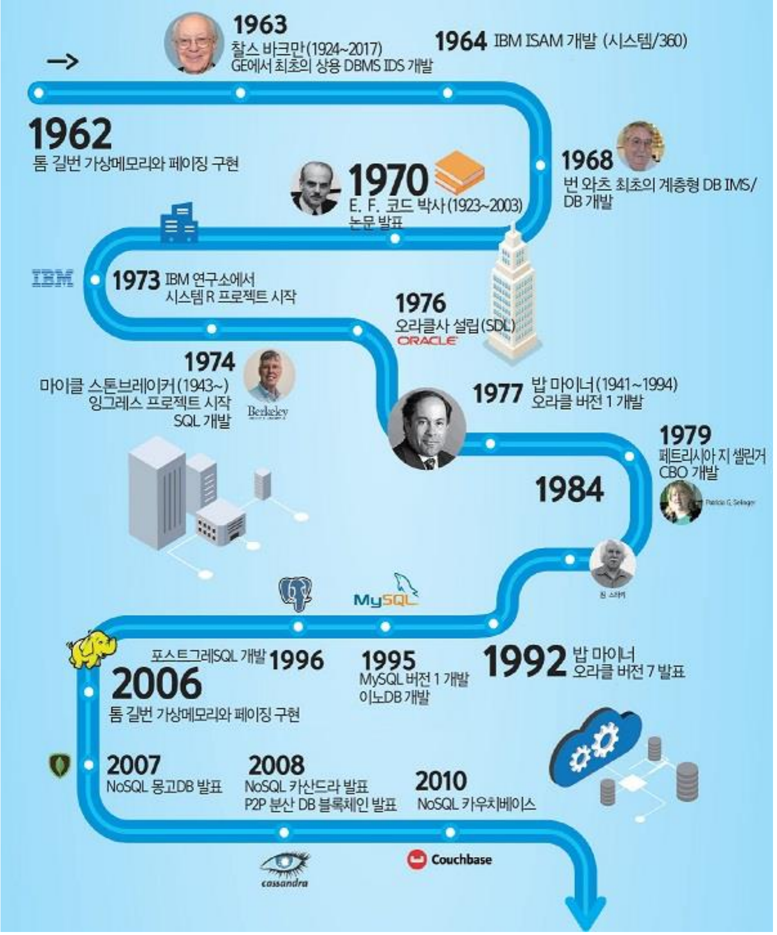
- **Defining a particular database in terms of its data types, structures, and constraints**
  - Loading initial database contents on a **secondary storage medium**

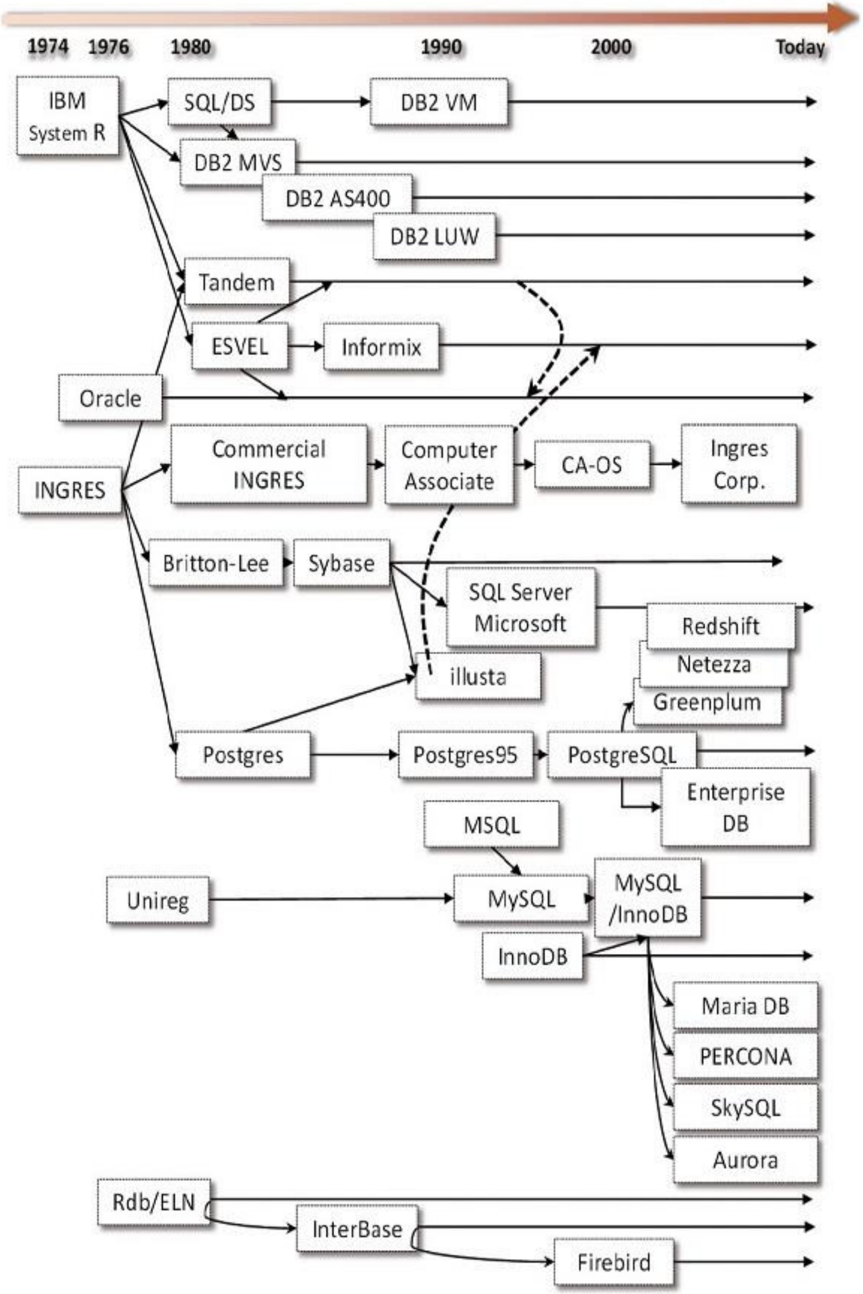
- **Manipulating the database:**
  - **Retrieval**: Querying and generating reports
  - **Modification**: Insertions, deletions and updates contents
  - Accessing the database through Web applications

- **Processing and sharing by a set of concurrent users and application programs**
  - Keeping all data valid and consistent

History of DBMS

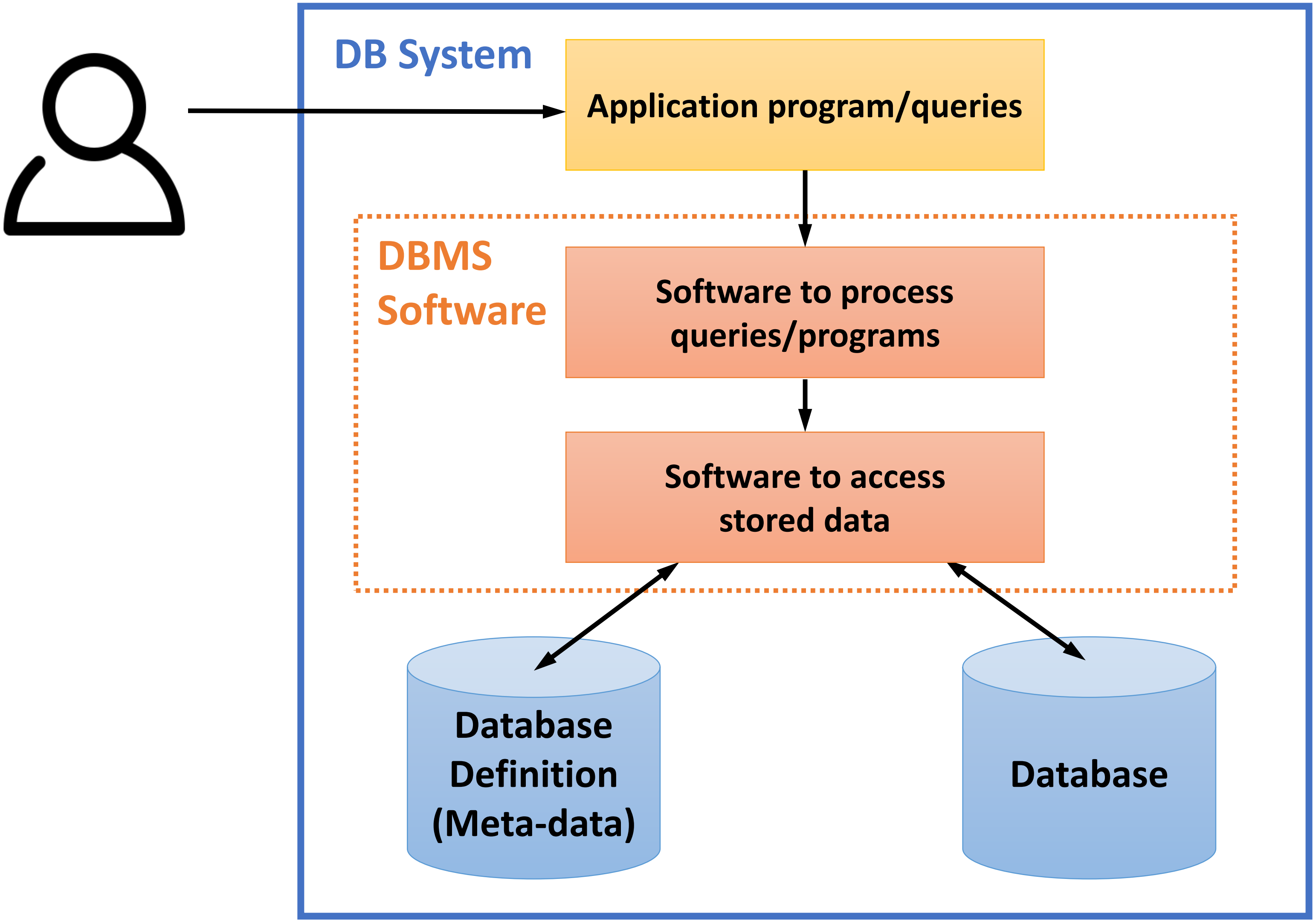
- ➢ **From IBM System R to today**





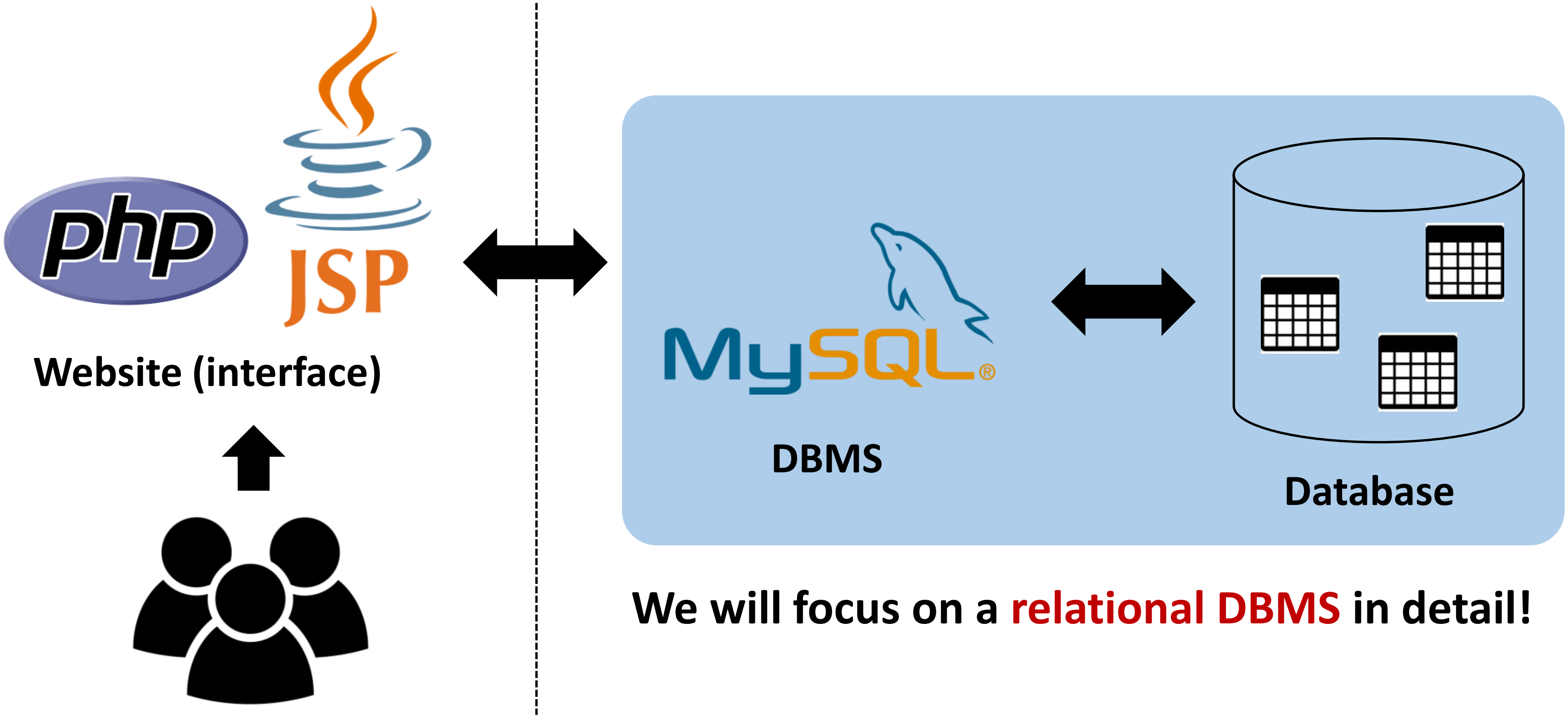
http://www.datanet.co.kr/news/articleView.html?idxno=114558

Simplified Database System


DB System
Application program/queries
DBMS Software
Software to process queries/programs
Software to access stored data
Database Definition (Meta-data)
Database

Example: Enrolling a Course

- **Two-tier architecture**
  - **Website**: Interface for end-users
  - **DBMS**: Managing users’ transactions
  - **Databases**: Storing users’ enrollments



We will focus on a relational DBMS in detail!

Course Objective

- **We will cover the following topics in DBMS.**
  - **Data model**: How to design a database?
    - Relational model, ER diagram
    - Functional dependency, Normalization
  - **SQL**: How to store/retrieve data from the database?

Database design

- **DBMS internals**: How to implement DBMS?
  - Data storage, Index structure, query processing
- **Transaction**: How to control DBMS with many users?
  - Concurrency control

Database implementation

- **Beyond the existing DBMS**
  - Distributed database, Hadoop
  - NoSQL, Data mining

Recent advances in DBMS

Objective: Database Design

1. Foundations: Relational models & SQL

- Basic concepts of relational DBMS
- How to manipulate data with SQL

2. DB design: ER model and design theory

- Conceptual database design: ER model
- Transforming relational schema from the ER diagram
- Functional dependency, Normalization

3. DB programming

- Implementing your own project with DBMS

Objective: Database Implementation

4. DB internals: File organization, Indexing, Transaction

- Storing/indexing data
- Relational algebra, basics of query optimization
- Locking, concurrency control

5. Recent advances in DB: Big data, Hadoop, NoSQL, data mining

- Hadoop programming
- Key-value stores and its variants
- Data analytics with ML&AI

Database Design

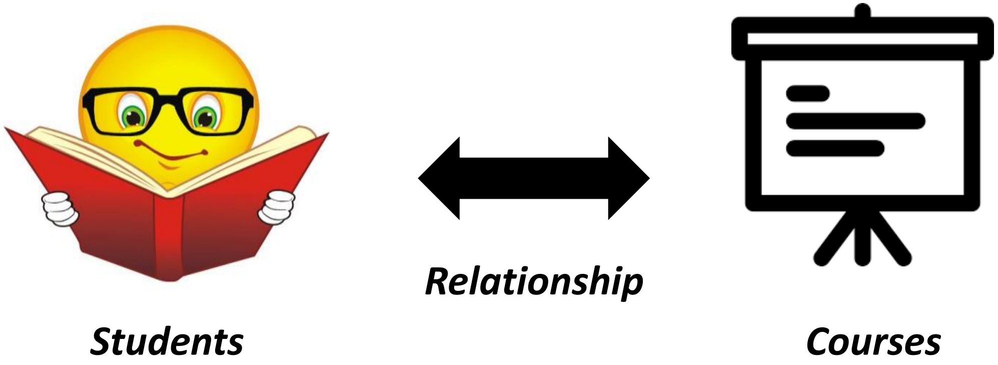
Motivation: Relational Model

- Consider the mini-world for SKKU management system.
  - **Students**
  - **Courses**
  - **Professors**

Entities

  - **Students** take **Courses**
  - **Professors** teach **Courses**

Relationships



- How to store data?

Flat File Database

- A database consists of a **single table.**

SID	Name	Course Name	CID
100	Alice	Data Structures	201
100	Alice	Database	301
100	Alice	Machine Learning	401
200	Bob	Data Structure	201
200	Bob	Database	301

- **Advantage**
  - Simple to create, easy to use, inexpensive
- **Disadvantage**
  - Data redundancy and inconsistency

Splitting into Multiple Relations

SID	Name	Course Name	CID
100	Alice	Data Structures	201
100	Alice	Database	301
100	Alice	Machine Learning	401
200	Bob	Data Structure	201
200	Bob	Database	301

Students take ***Courses***

Students

SID	Name
100	Alice
200	Bob

Enrolled

SID	CID
100	201
100	301
100	401
200	201
200	301

Courses

CID	Course Name
201	Data Structure
301	Database
401	Machine Learning

Detail: Splitting into Multiple Relations

SID	Name	Course Name	CID
100	Alice	Data Structures	201
100	Alice	Database	301
100	Alice	Machine Learning	401
200	Bob	Data Structure	201
200	Bob	Database	301

Spitting into two relations

Students

SID	Name
100	Alice
200	Bob

SID	Course Name	CID
100	Data Structures	201
100	Database	301
100	Machine Learning	401
200	Data Structure	201
200	Database	301

Detail: Splitting into Multiple Relations

SID	Course Name	CID
100	Data Structures	201
100	Database	301
100	Machine Learning	401
200	Data Structure	201
200	Database	301

Spitting into *Enrolled* and *Courses*

Enrolled

SID	CID
100	201
100	301
100	401
200	201
200	301

Courses

CID	Course Name
201	Data Structure
301	Database
401	Machine Learning

Relational Database

- A database is comprised of **multiple relations.**

Students

SID	Name
100	Alice
200	Bob

Enrolled

SID	CID
100	201
100	301
100	401
200	201
200	301

Courses

CID	Course Name
201	Data Structure
301	Database
401	Machine Learning

Advantage

- Reduce data redundancy, consistency, shared data

Disadvantage

- More complex, time latency

Combining Multiples Relations

Students

SID	Name
100	Alice
200	Bob

Enrolled

SID	CID
100	201
100	301
100	401
200	201
200	301

SID	Name	CID
100	Alice	201
100	Alice	301
100	Alice	401
200	Bob	201
200	Bob	301

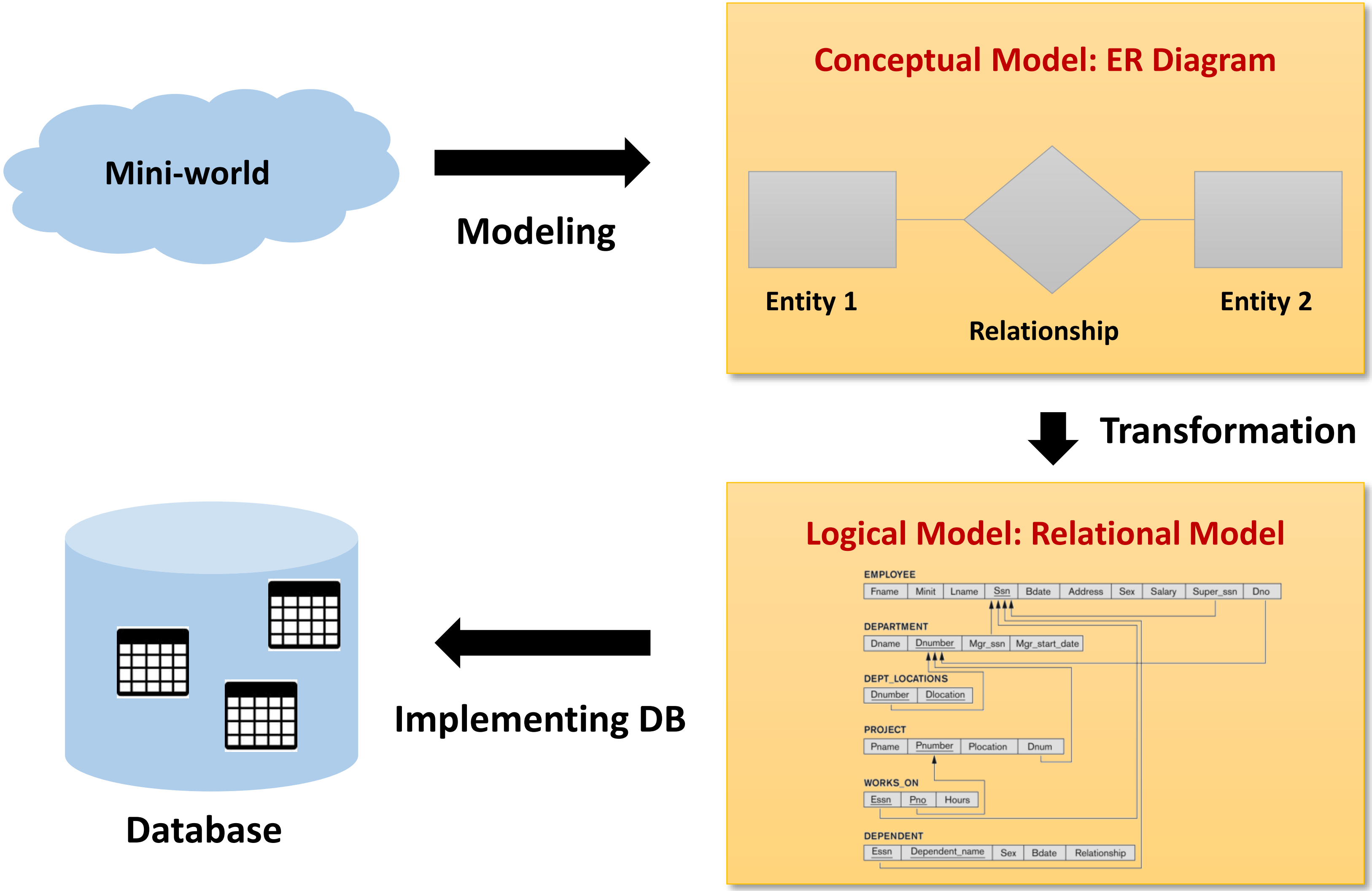
SID	Name	CID
100	Alice	201
100	Alice	301
100	Alice	401
200	Bob	201
200	Bob	301

Courses

CID	Course Name
201	Data Structure
301	Database
401	Machine Learning

SID	Name	CName	CID
100	Alice	Data Structures	201
100	Alice	Database	301
100	Alice	Machine Learning	401
200	Bob	Data Structure	201
200	Bob	Database	301

What is a Database Design?



Mini-world

Modeling

Conceptual Model: ER Diagram

Entity 1

Relationship

Entity 2

Transformation

Logical Model: Relational Model

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date

DEPT_LOCATIONS

Dnumber	Dlocation

PROJECT

Pname	Pnumber	Plocation	Dnum

WORKS_ON

Essn	Pno	Hours

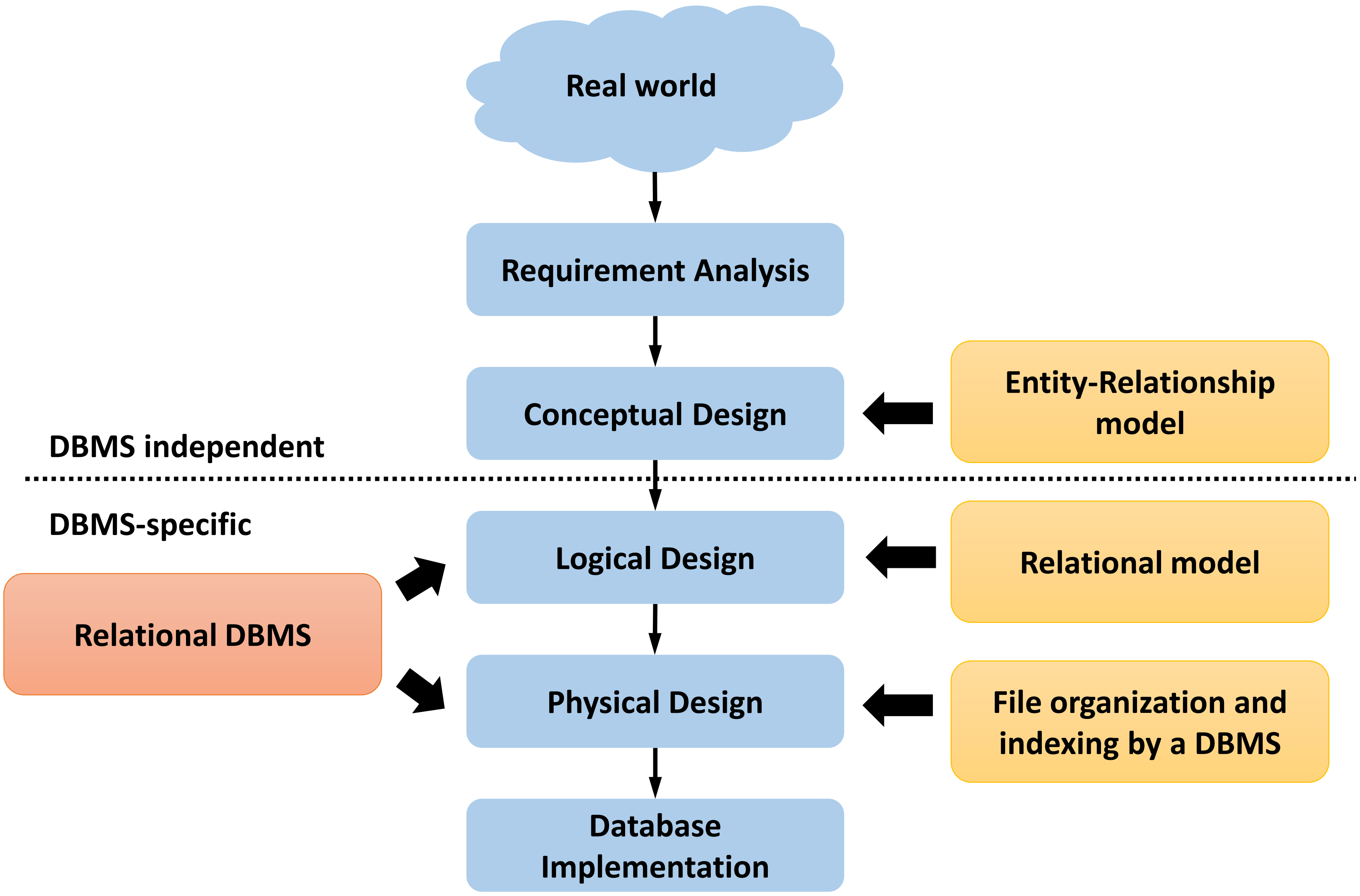
DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship

Implementing DB

Database

Overview of Database Design



Real world

Requirement Analysis

Conceptual Design

Entity-Relationship model

DBMS independent

DBMS-specific

Logical Design

Relational model

Relational DBMS

Physical Design

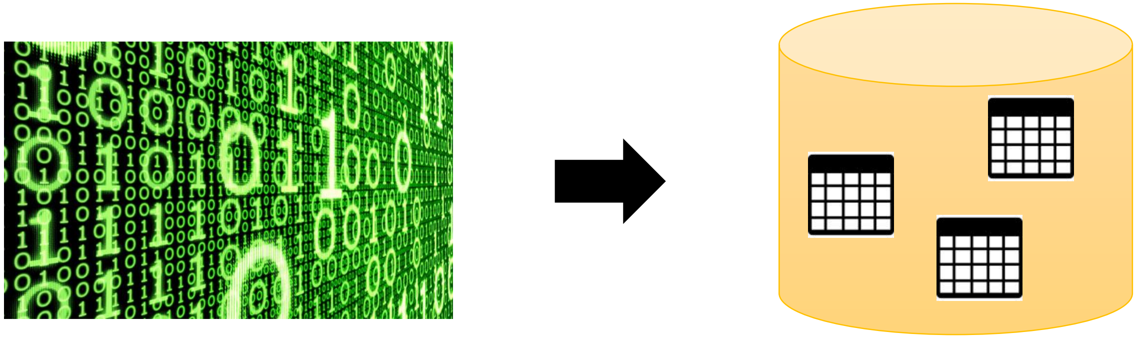
File organization and indexing by a DBMS

Database Implementation

Relational Data Model

- **A data model is a collection of high-level concepts for describing data.**
  - It hides many low-level details in the physical design.

- **The relational model has been widely used.**
  - **Relation ≈ Table**
  - A **schema** is a description of a particular collection of data, using the given data model.



Raw data

Database

Definition of Relational Databases

- **Relational database has a set of relations.**

- **A relation consists of two parts:**
  - **Schema**: relation name + name and type of each column
  - **Instance**: a table with rows and columns
    - # of rows = **cardinality**, # of fields = **degree** / **arity**
    - A relation = a set of rows (or **tuples**)

Courses

CID	Name	Credit	Department
101	C programming	3	CS
102	Discrete Math	2	Math
301	Databases	4	CS
302	Artificial Intelligence	3	CS
405	Data Mining	3	CS

Example: Relational Data Model

- **Logical schema**
  - *Students*(*sid*: string, *name*: string, *gpa*: float)
  - *Courses*(*cid*: string, *name*: string, *credits*: int)
  - *Enrolled*(*sid*: string, *cid:* string, *grade:* string)

Students

SID	Name	GPA
100	Alice	3.5
200	Bob	3.0

Courses

CID	Name	Credits
202	DS	3
301	DB	4

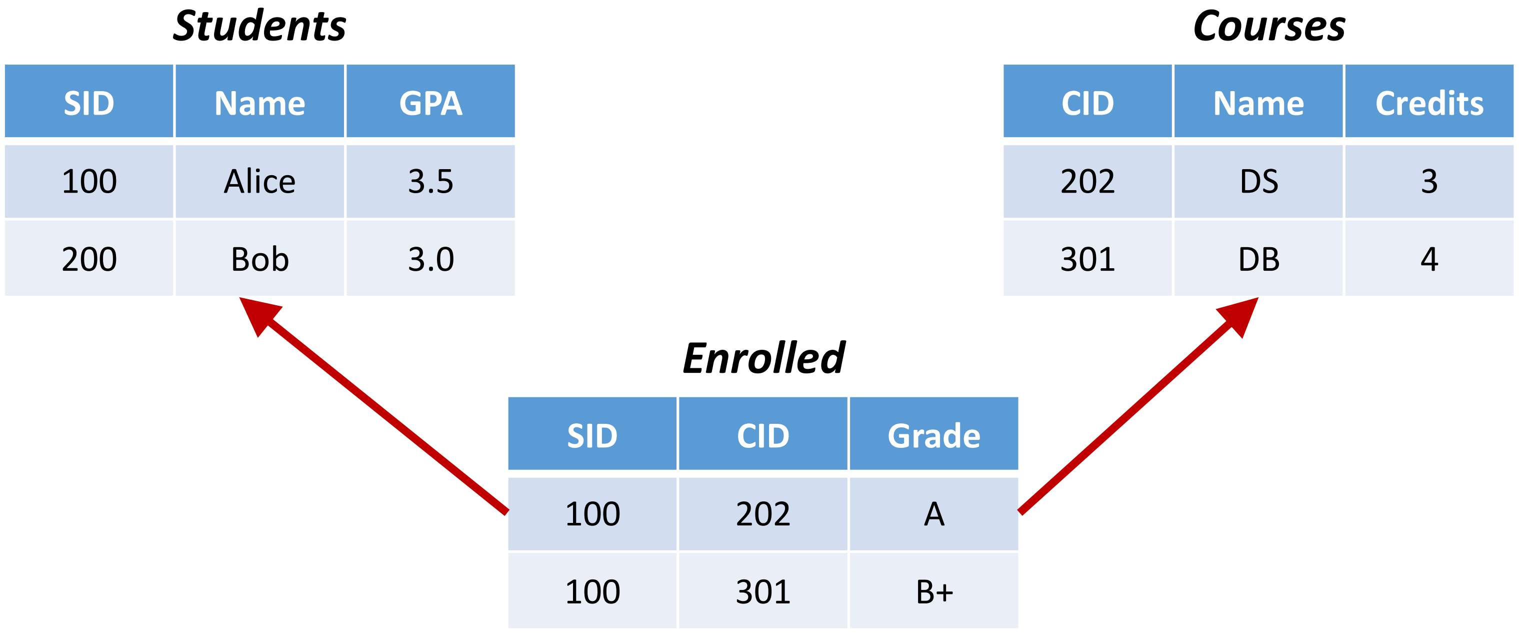
Enrolled

SID	CID	Grade
100	202	A
100	301	B+

Example: Relational Data Model

➢ Logical schema

- *Students*(*sid*: string, *name*: string, *gpa*: float)
- *Courses*(*cid*: string, *name*: string, *credits*: int)
- *Enrolled*(*sid*: string, *cid:* string, *grade:* string)



Students

SID	Name	GPA
100	Alice	3.5
200	Bob	3.0

Courses

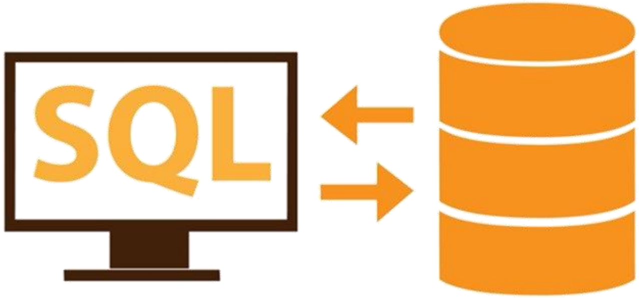
CID	Name	Credits
202	DS	3
301	DB	4

Enrolled

SID	CID	Grade
100	202	A
100	301	B+

What is SQL?

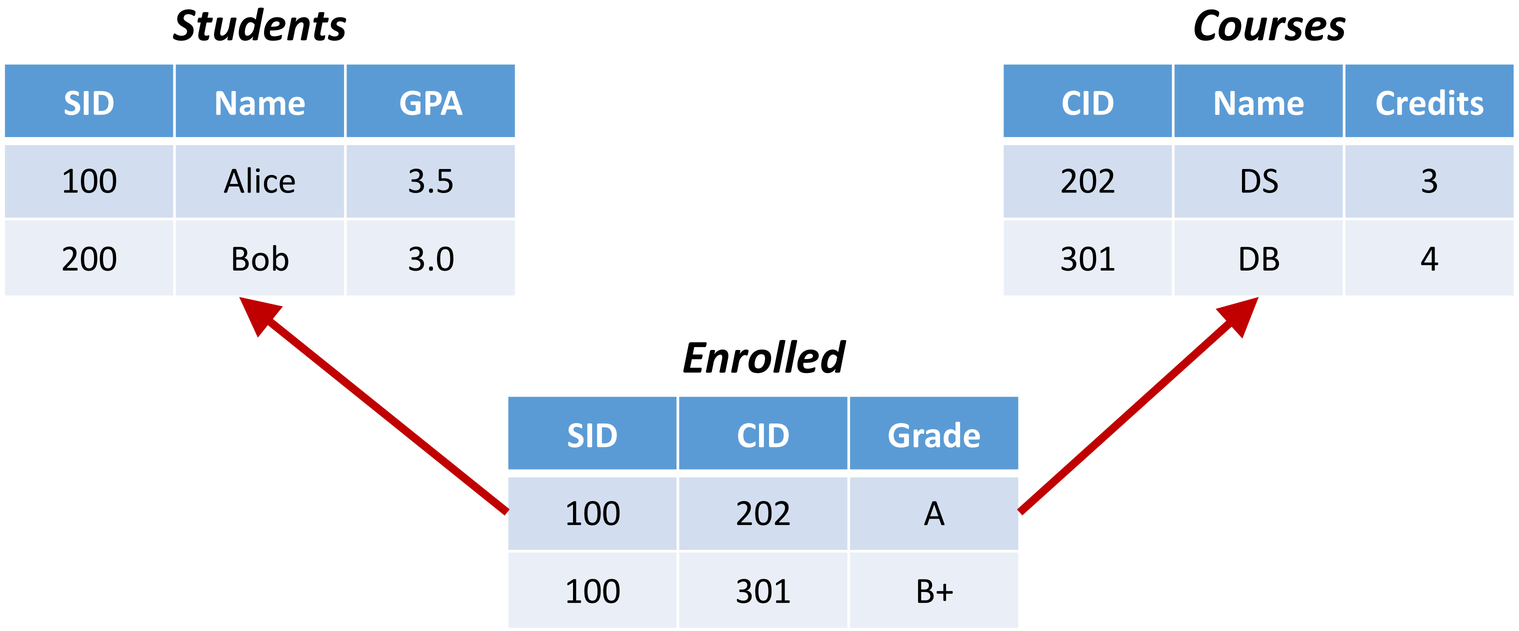
- **The origin of SQL is relational predicate calculus**
  - SQL comes from the word “SEQUEL”.
  - Popularly known as “**Structured query language**.”



- **SQL tutorial**
  - https://www.w3schools.com/sql/default.asp
  - Quiz: https://www.w3schools.com/sql/sql_quiz.asp

Queries in DBMS

- **Queries: Accessing different parts of data and formulate the result of a request**
- **What does a user want to retrieve?**
  - What is the name of the student with SID 100?
  - How many students are enrolled in CID 301?



Students

SID	Name	GPA
100	Alice	3.5
200	Bob	3.0

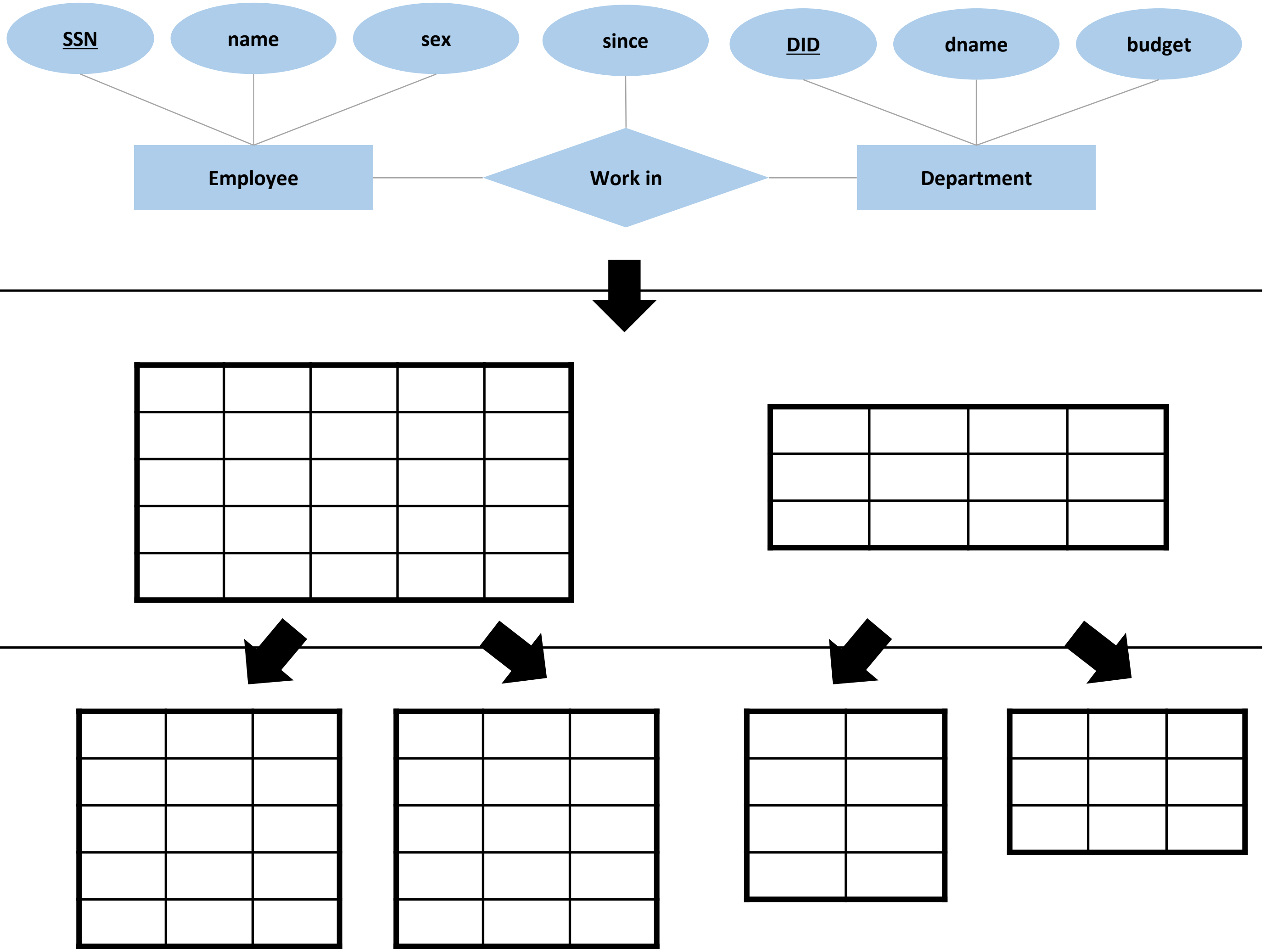
Courses

CID	Name	Credits
202	DS	3
301	DB	4

Enrolled

SID	CID	Grade
100	202	A
100	301	B+

Database Design Process



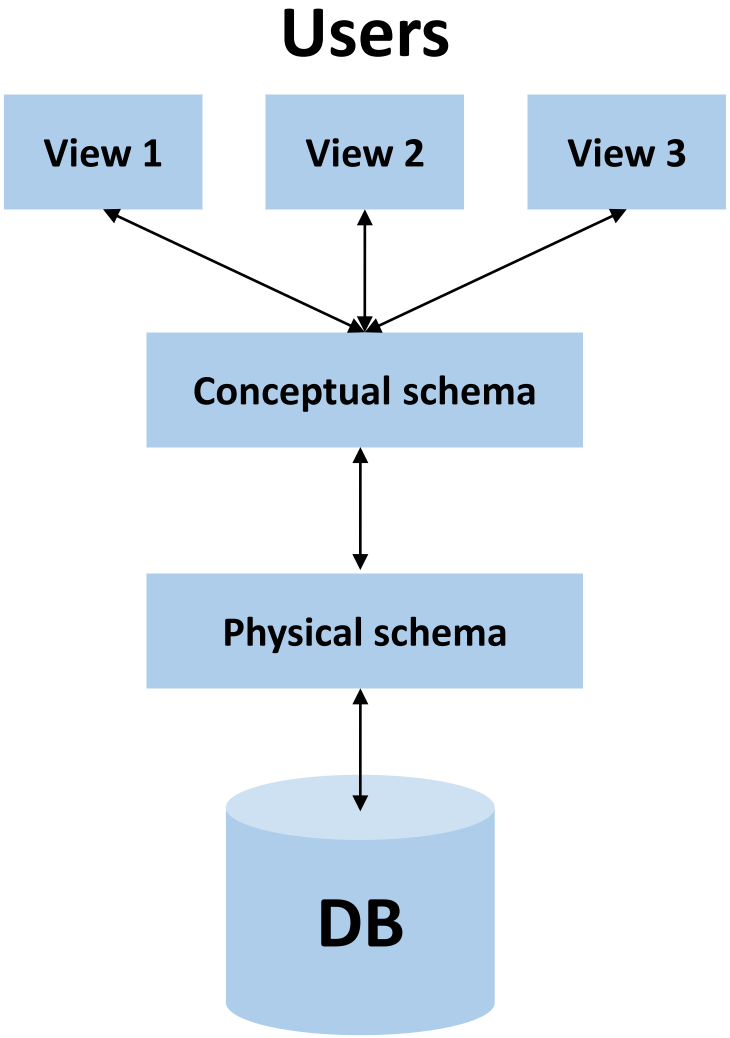
Conceptual model:

Relational Model: Tables + constraints

Normalization: Eliminates anomalies

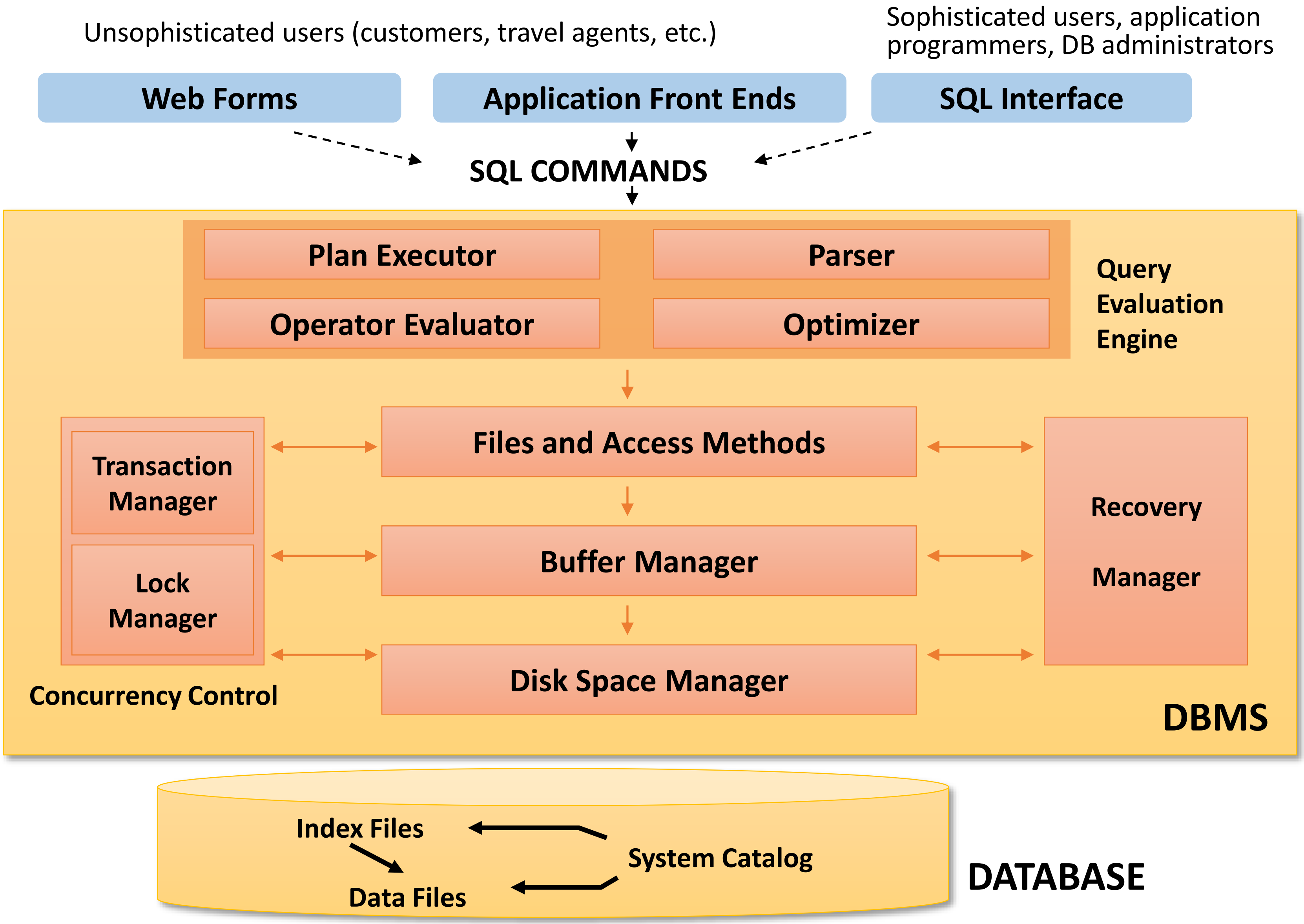
Levels of Schemata

- **External schemas (or views)**
  - Describe how users see the data.
- **Conceptual (or logical) schema**
  - Defines logical structures.
- **Physical schema**
  - Describes the files and indexes used
    - Relations as unordered files.
    - Some data in sorted order (index).



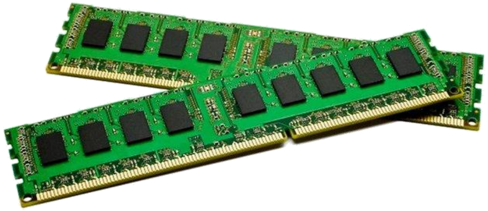
Database Implementation

Anatomy of an RDBMS



Storing Data on Disks

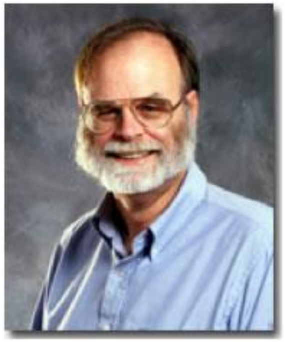
- **The DBMS stores data on disks.**
  - Electronic (CPU, DRAM) vs. Mechanical (HDD)



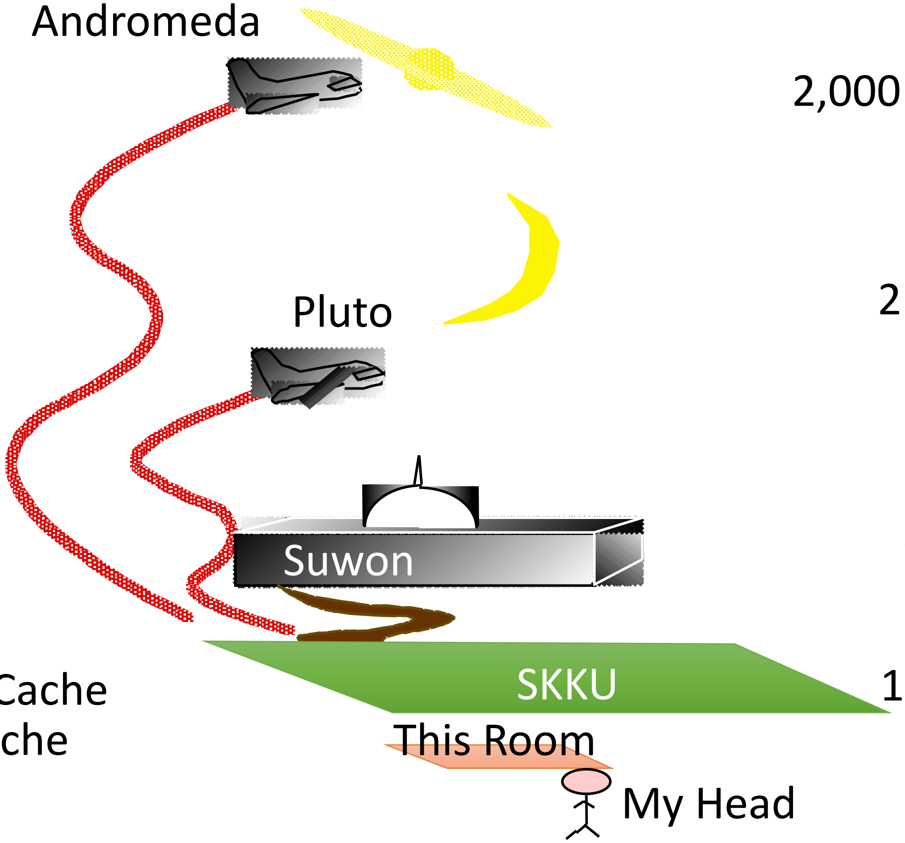
- **This has major implications for DBMS design!**
  - **READ**: transfer data from disk to RAM for data processing.
  - **WRITE**: transfer data (new/modified) from RAM to disk.
  - Both are high-cost operations, relative to in-memory operations, so must be **planned carefully**!

How Far Away is the Data?

- Jim Gray's storage latency analogy



Jim Gray
(Turing award in 1998)



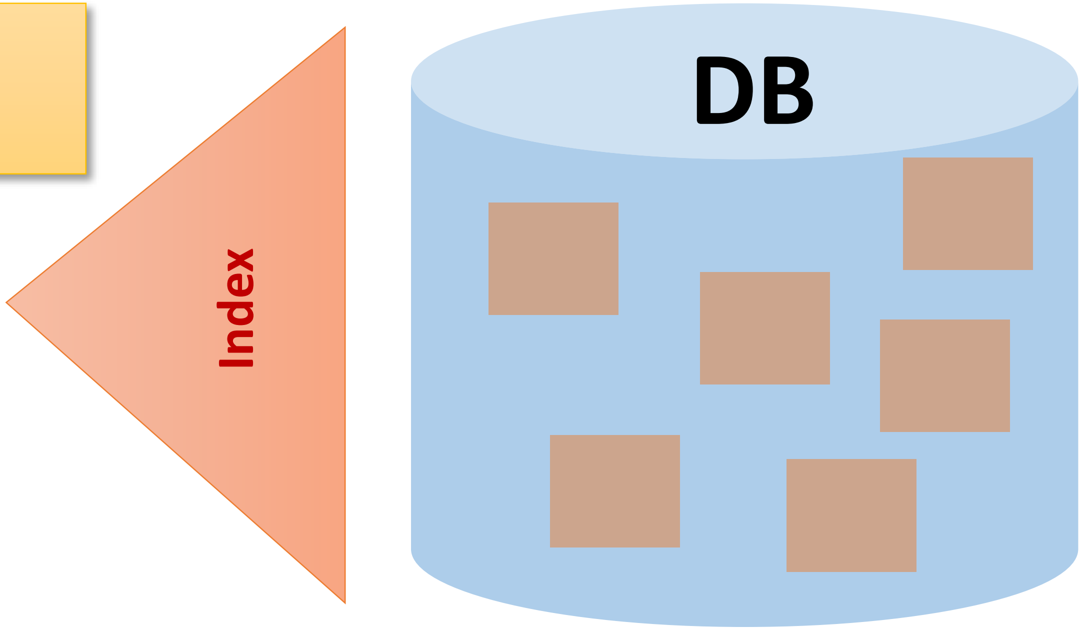
10^9	Tape	Andromeda	2,000 Years
10^6	Disk	Pluto	2 Years
100	Memory	Suwon	1.5 hr
10	On Board Cache	SKKU	10 min
2	On Chip Cache	This Room	
1	Registers	My Head	1 min

We need a 'gap filler'!

Indexing Data

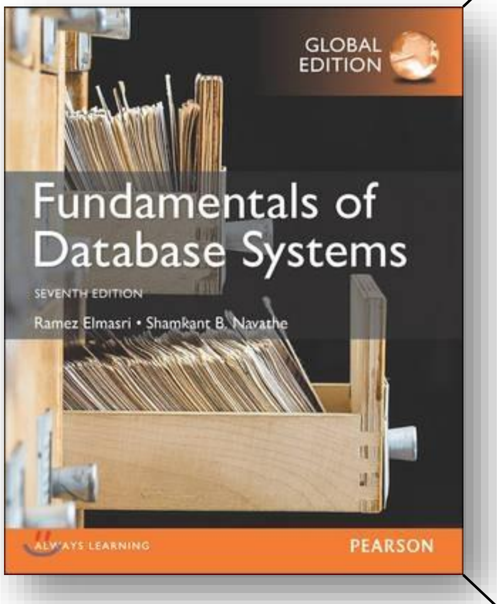
- A data structure for **efficient search** through large databases
- Two key ideas
  - The records are mapped to the disk blocks in specific ways
  - **Auxiliary data structures** are maintained for quick search

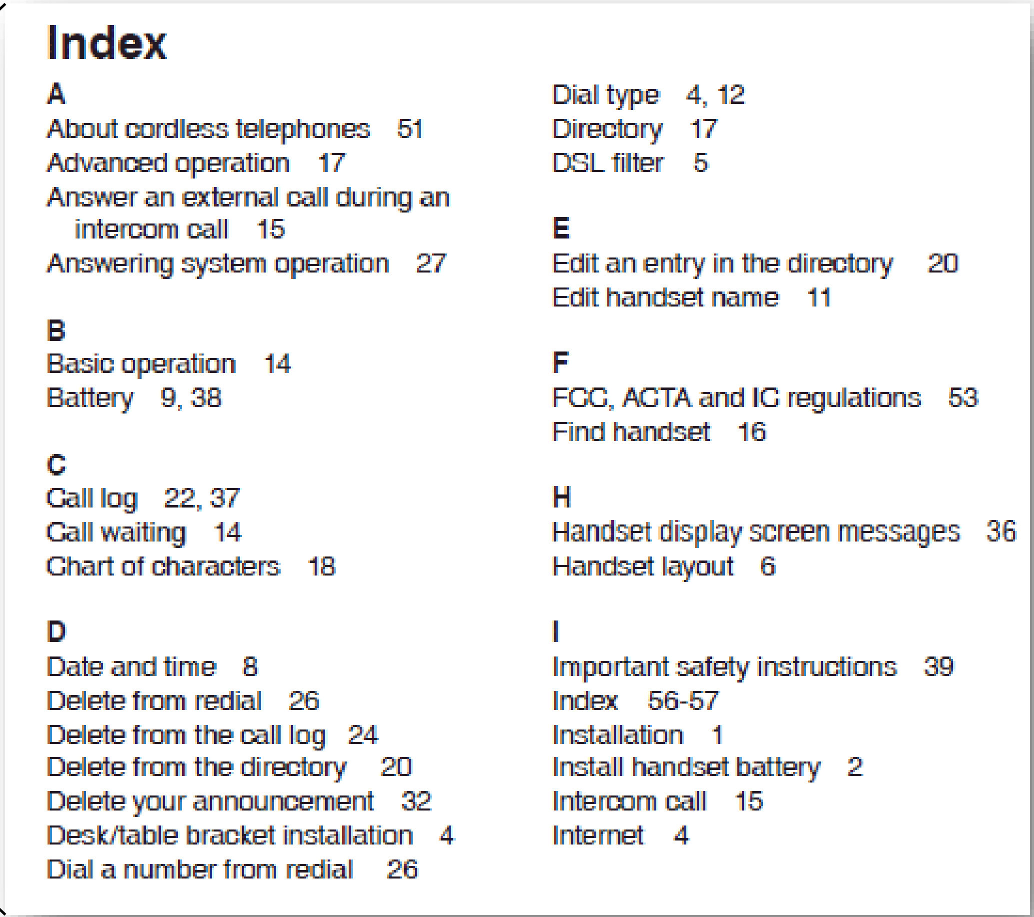
How to access records as efficiently as possible?



Example: Index in a Book

- How to find a specific term?


GLOBAL EDITION
Fundamentals of Database Systems
SEVENTH EDITION
Ramez Elmasri • Shamkant B. Navathe
ALWAYS LEARNING
PEARSON



Index

A
About cordless telephones 51
Advanced operation 17
Answer an external call during an intercom call 15
Answering system operation 27

B
Basic operation 14
Battery 9, 38

C
Call log 22, 37
Call waiting 14
Chart of characters 18

D
Date and time 8
Delete from redial 26
Delete from the call log 24
Delete from the directory 20
Delete your announcement 32
Desk/table bracket installation 4
Dial a number from redial 26
Dial type 4, 12
Directory 17
DSL filter 5

E
Edit an entry in the directory 20
Edit handset name 11

F
FCC, ACTA and IC regulations 53
Find handset 16

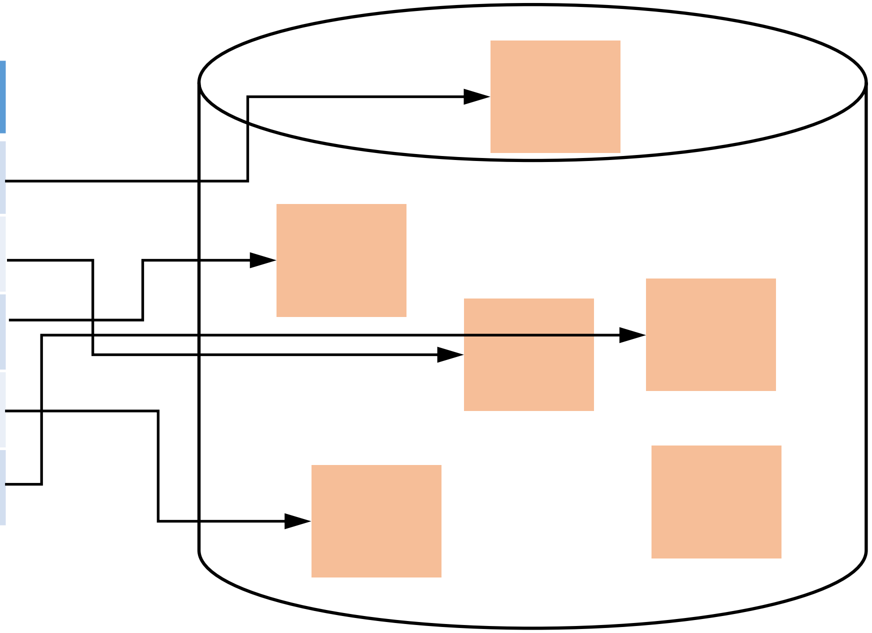
H
Handset display screen messages 36
Handset layout 6

I
Important safety instructions 39
Index 56-57
Installation 1
Install handset battery 2
Intercom call 15
Internet 4

Indexes as Access Paths

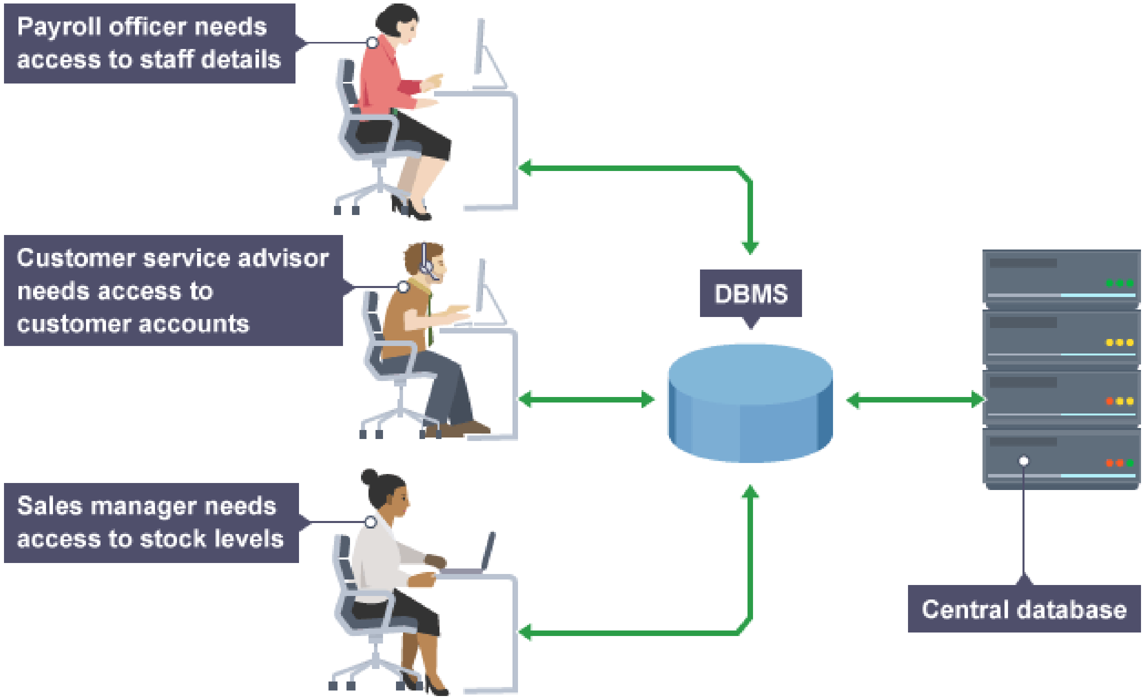
- **An index is an auxiliary file that makes it more efficient to search for a record in the data file.**
  - The index is usually specified on **one field** of the file (although it could be specified on several fields).
  - One form of an index is a file of entries <**field value**, **pointer to record**>, which is ordered by field value.

Data	Block address
1	FF0011
2	FE0021
3	…
4	…
5	…



Single-user vs. Multi-user

- **Single-user system**
  - At most **one user** at a time can **use** the system.
- **Multi-user system**
  - **Many users** can access the system **concurrently**.



Payroll officer needs access to staff details
Customer service advisor needs access to customer accounts
Sales manager needs access to stock levels
DBMS
Central database

Challenges with Many Users

- **Suppose that our application serves 1000 users or more.**
  - **Performance**: Need to provide concurrent access

    Disk/SSD access is slow, DBMS hide the **latency** by doing **more CPU work concurrently**.

  - **Consistency**: Concurrency can lead to update problems.

    DBMS allows users to write programs as if **they were the only user**

  - **Security**: Different users, different roles

Application Activities Against a DB

- **Applications interact with a database by generating:**
  - **Queries**: it accesses different parts of data and formulate the result of a request.
  - **Transactions**: it may read some data and “update” certain values or generate new data and store that in the database.

- **For the DBMS,**
  - Must not allow unauthorized users to access data.
  - Must keep up with changing user requirements against DB.

What is a Transaction (TXN)?

- An atomic sequence of DB actions (reads/writes)
- Four properties of transaction: ACID
  - Atomicity, Consistency, Isolation, Durability

- Example

Account	Balance
A1	10,000
A2	20,000

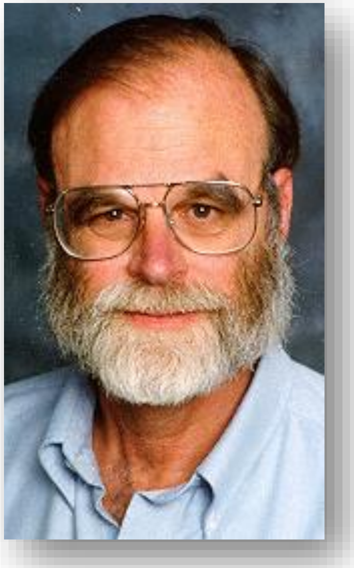
Transfer $3k from A1 to A2:
1. Debit $3k from A1.
2. Credit $3k to A2.

Account	Balance
A1	7,000
A2	20,000

Possible cases:

- Crash before 1,
- **After 1 but before 2,**
- After 2.

ACID Transactions



Jim Gray
(Turing Award
Winner 1998)

Atomicity:

Each transaction is "all or nothing"

Consistency:

Data should be valid according to all defined rules.

Isolation:

Transactions do not affect each other.

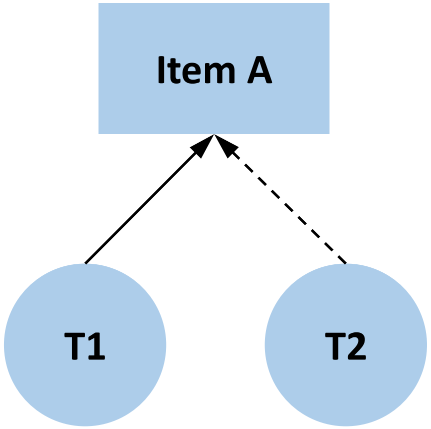
Durability:

Committed data would not be lost, even after power failure.

Concurrency Control

- To enforce **isolation** among conflicting transactions
- To preserve **database consistency** through consistency preserving execution of transactions

- Example
  - If **T1** conflicts with **T2** over a data item A,
  - The concurrency control manager decides if **T1** or **T2** should get item A.

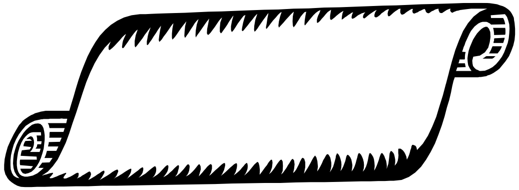


Ensuring Atomicity and Durability

- **Recovery: DBMS ensures atomicity even if a TXN crashes!**

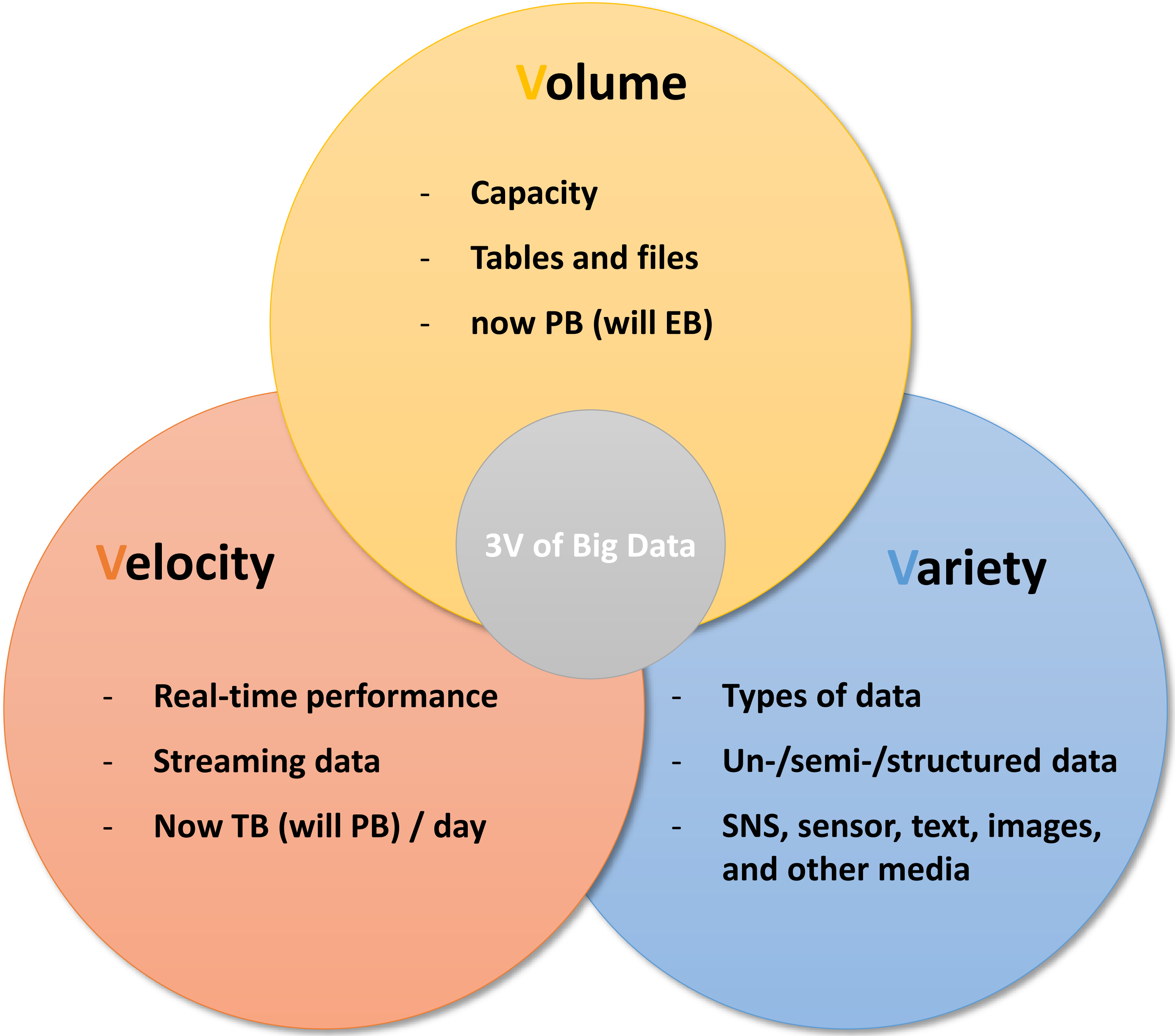
- **One way to accomplish this:**
  - **Write-ahead logging (WAL):** Before any action is finalized, a corresponding log entry is forced to disk.

- **Key idea**
  - Keep a log of all the writes done.
  - After a crash, the partially executed TXNs are undone using the **log**.



Recent Advances in DBMS

What is Big Data?



Volume

- Capacity
- Tables and files
- now PB (will EB)

3V of Big Data

Velocity

- Real-time performance
- Streaming data
- Now TB (will PB) / day

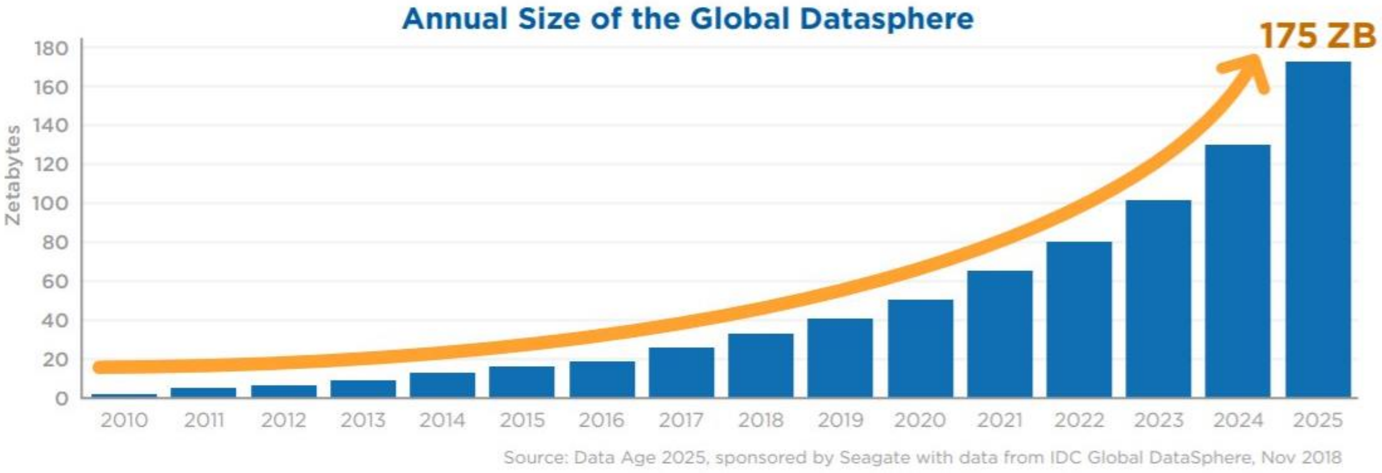
Variety

- Types of data
- Un-/semi-/structured data
- SNS, sensor, text, images, and other media

Volume (Scale)

- In 2018, International Data Corporation (IDC) estimated the global datasphere has reached **33 zettabytes** and is expected to reach **175 zettabytes** by 2025.

- 1 ZB = 10^{21} bytes = 1 trillion GB = 1,000,000,000,000 GB



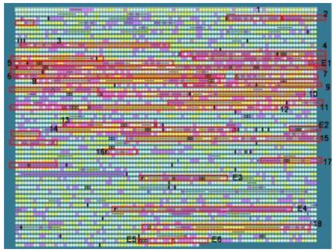
Source: Data Age 2025, sponsored by Seagate with data from IDC Global DataSphere, Nov 2018

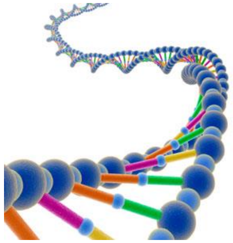
https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf

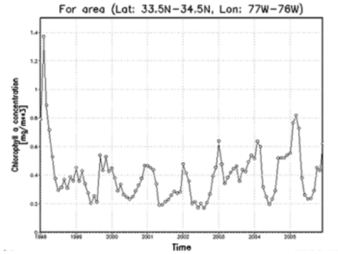
Variety (Structure)

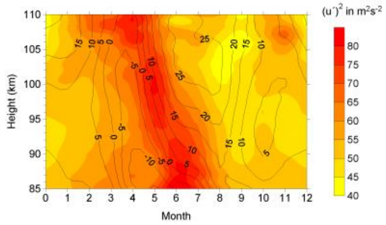
- Relational Data (Tables/Transaction/Legacy Data)
- Text Data (Web)
- Semi-structured Data (XML)
- Graph Data
  - Social Network, Semantic Web (RDF), …
- Streaming Data
  - You can only scan the data once.

To extract knowledge , all these types of data need to be **linked together**.



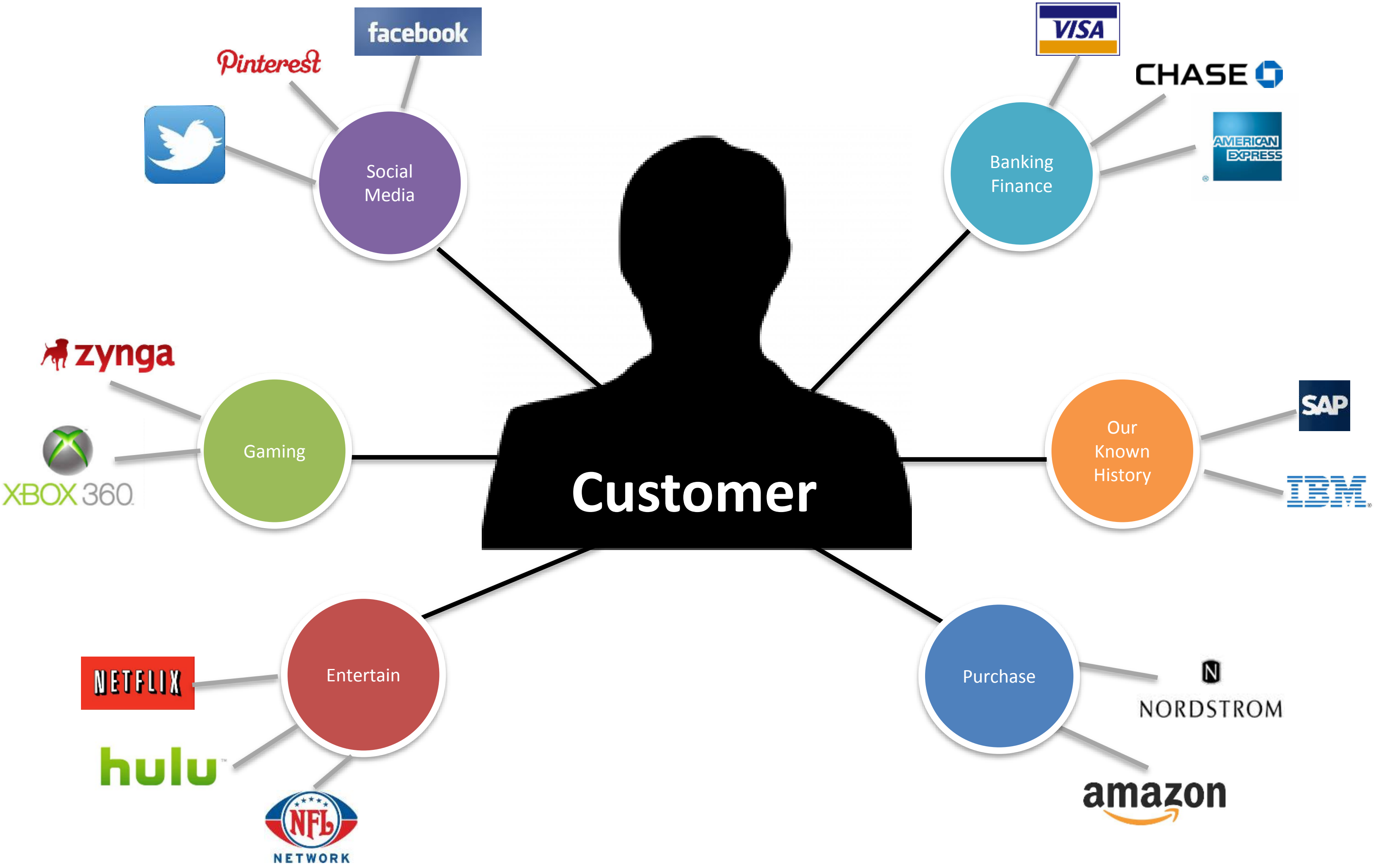



For area (Lat: 33.5N–34.5N, Lon: 77W–76W)
Chlorophyll a concentration [mg/m**3]
Time



Variety (Structure)

- A single view to the customer



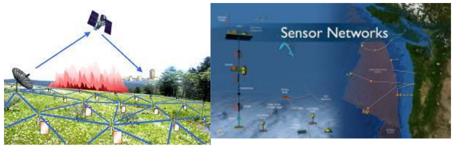
Velocity (Speed)

- The ability to manage, analyze, summarize, visualize, and discover knowledge in a **timely fashion**

Social media and networks
(all of us are generating data)

Scientific instruments
(collecting all sorts of data)

Mobile devices
(tracking all objects all the time)



Sensor technology and networks
(measuring all kinds of data)

Velocity (Speed)

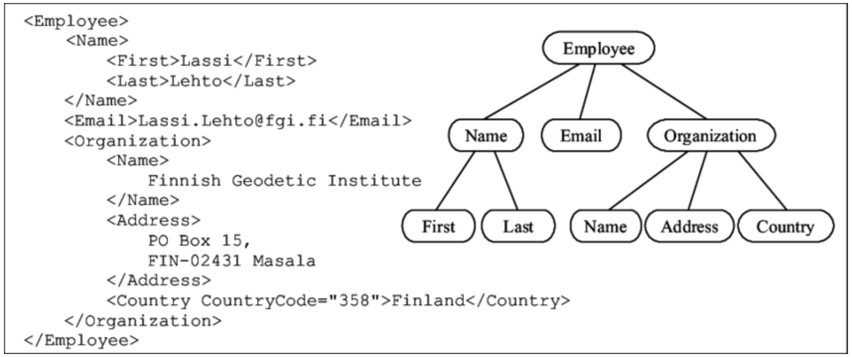
- **Online data analytics**
  - Late decisions → missing opportunities
- **Examples**
  - **E-Promotions**: Based on your current location, your purchase history, and what you like → send promotions right now to you.
  - **Healthcare monitoring**: sensors monitor your activities and body → any abnormal measurements require immediate reaction.

Why is the RDBMS *not* Suitable?

- **RDBMS assumes that data are**
  - Dense, largely uniform **structured** data

- **Data coming from Internet are**
  - Massive and sparse **semi-structured** or **unstructured** data

Structured data	Semi-structured data	Unstructured data
Databases	XML / JSON data Email Web pages	Audio Video Image data Natural language Documents



```
<Employee>
    <Name>
        <First>Lassi</First>
        <Last>Lehto</Last>
    </Name>
    <Email>Lassi.Lehto@fgi.fi</Email>
    <Organization>
        <Name>
            Finnish Geodetic Institute
        </Name>
        <Address>
            PO Box 15,
            FIN-02431 Masala
        </Address>
        <Country CountryCode="358">Finland</Country>
    </Organization>
</Employee>
```

What is NoSQL?

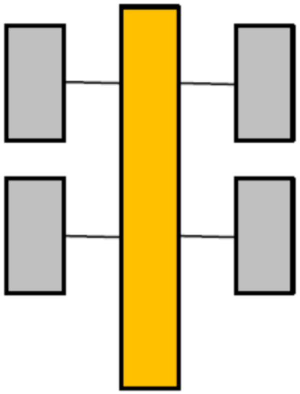
- **NoSQL stands for:**
  - **Non-**SQL (or **No RDBMS**)
  - **Not only** SQL

- **An umbrella term for all databases and data stores that do NOT follow the RDBMS principles**
  - Often related to **unstructured** large-scale datasets.

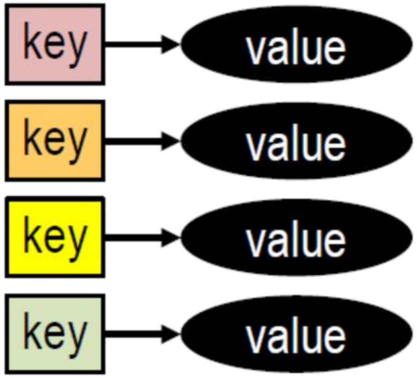
After NoSQL

Relational

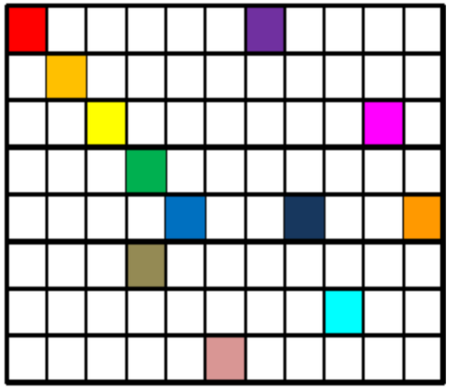
Analytical (OLAP)



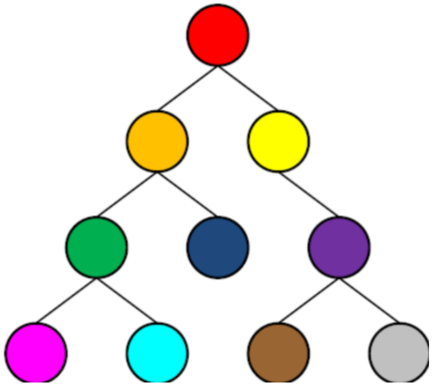
Key-value



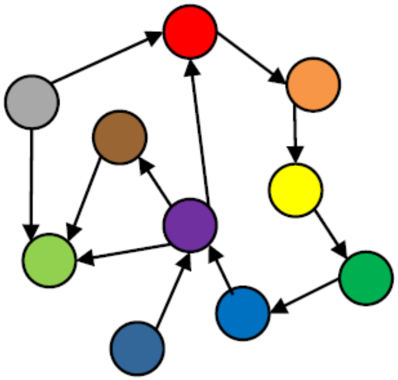
Column-oriented



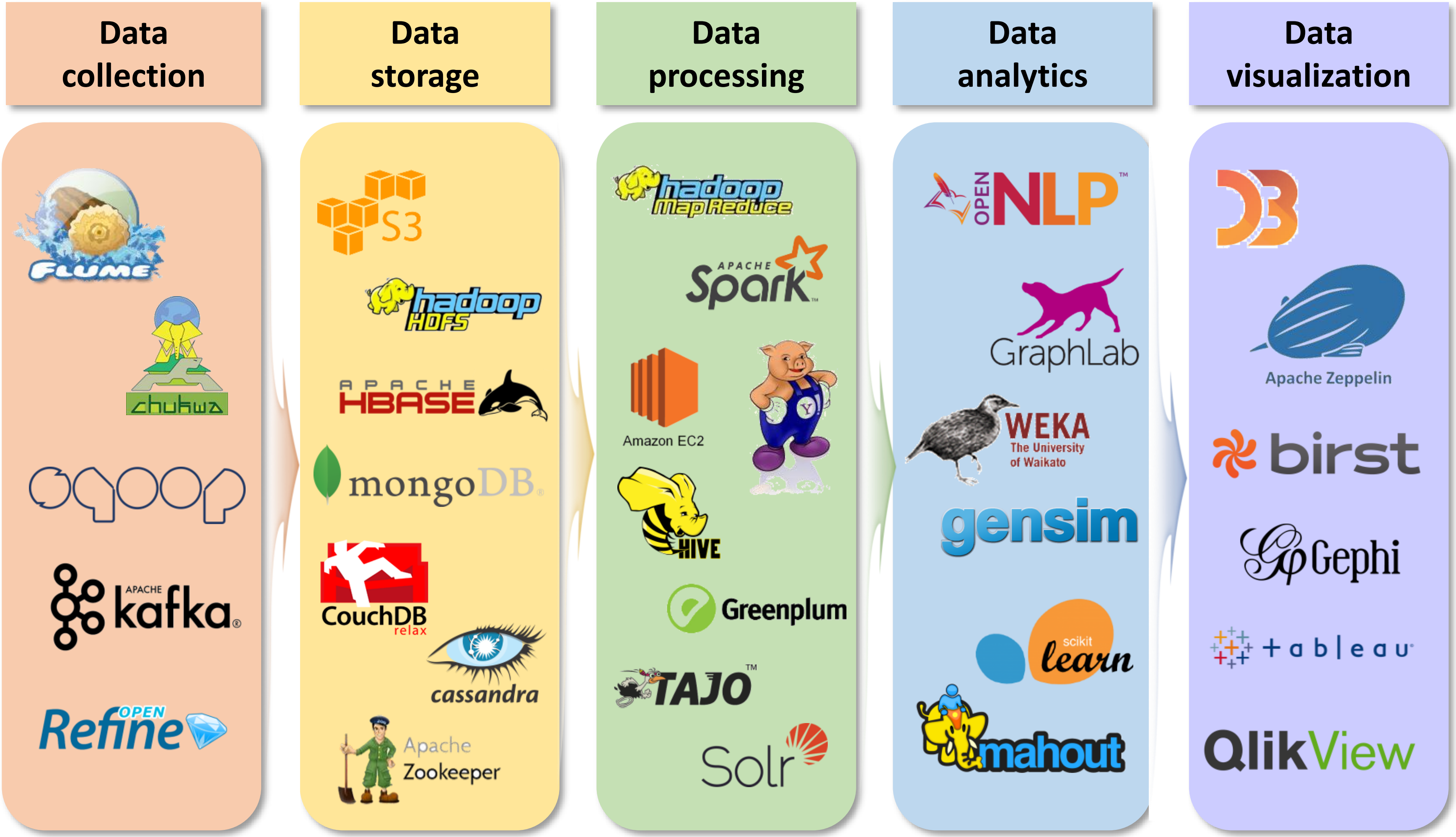
Document



Graph



Big Data Technology Stack



Data collection

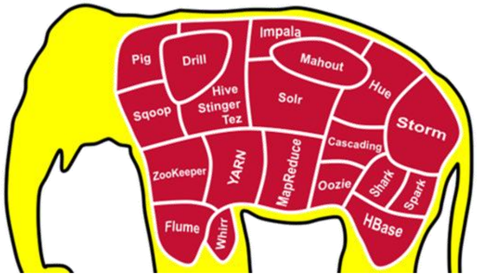
Data storage

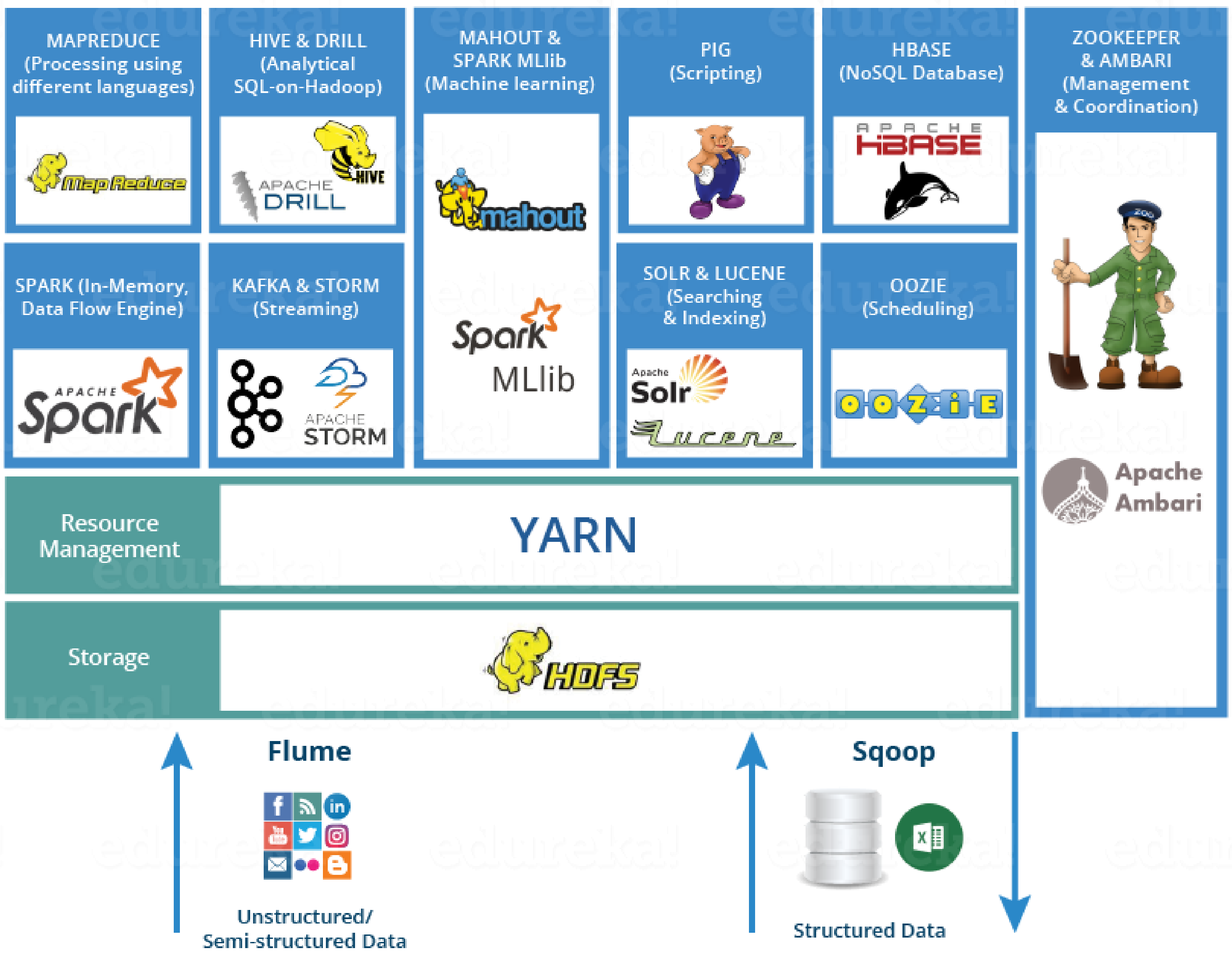
Data processing

Data analytics

Data visualization

The Hadoop Ecosystem




MAPREDUCE (Processing using different languages)

HIVE & DRILL (Analytical SQL-on-Hadoop)

MAHOUT & SPARK MLlib (Machine learning)

PIG (Scripting)

HBASE (NoSQL Database)

ZOOKEEPER & AMBARI (Management & Coordination)

SPARK (In-Memory, Data Flow Engine)

KAFKA & STORM (Streaming)

SOLR & LUCENE (Searching & Indexing)

OOZIE (Scheduling)

Resource Management

YARN

Storage

Flume

Unstructured/ Semi-structured Data

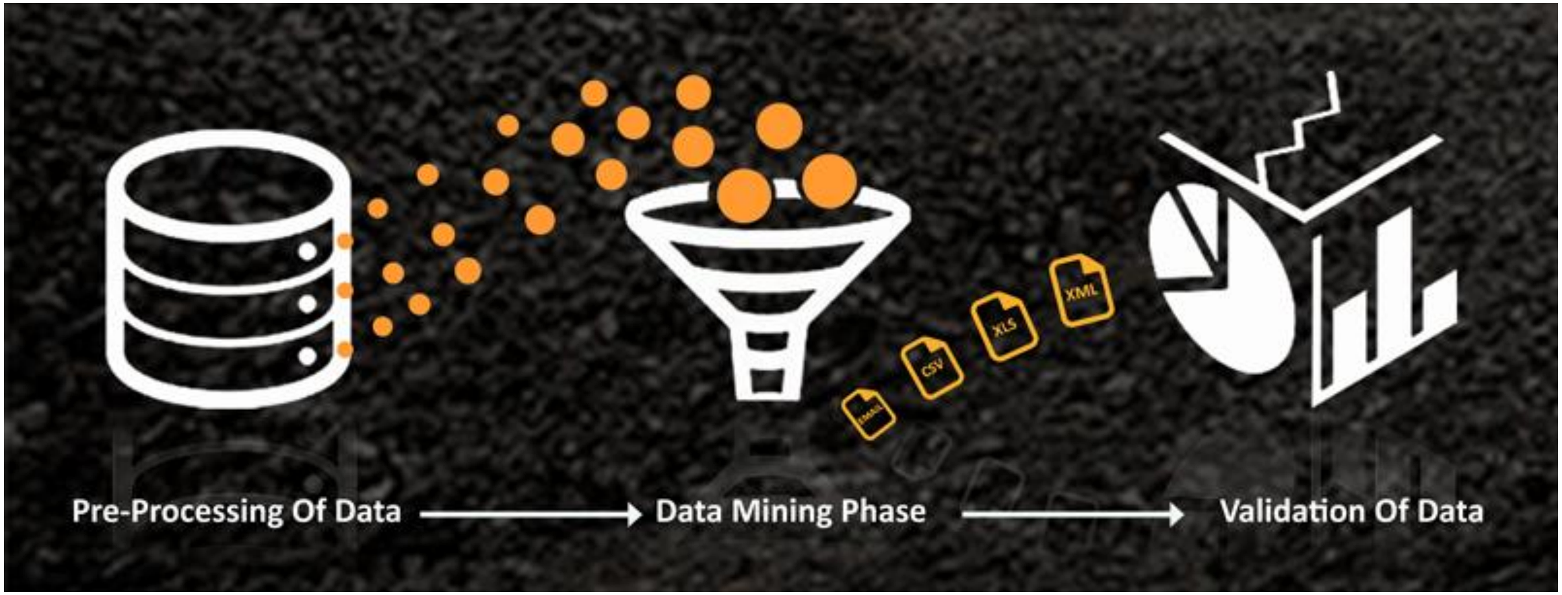
Sqoop

Structured Data

What is Data Mining (DM)?

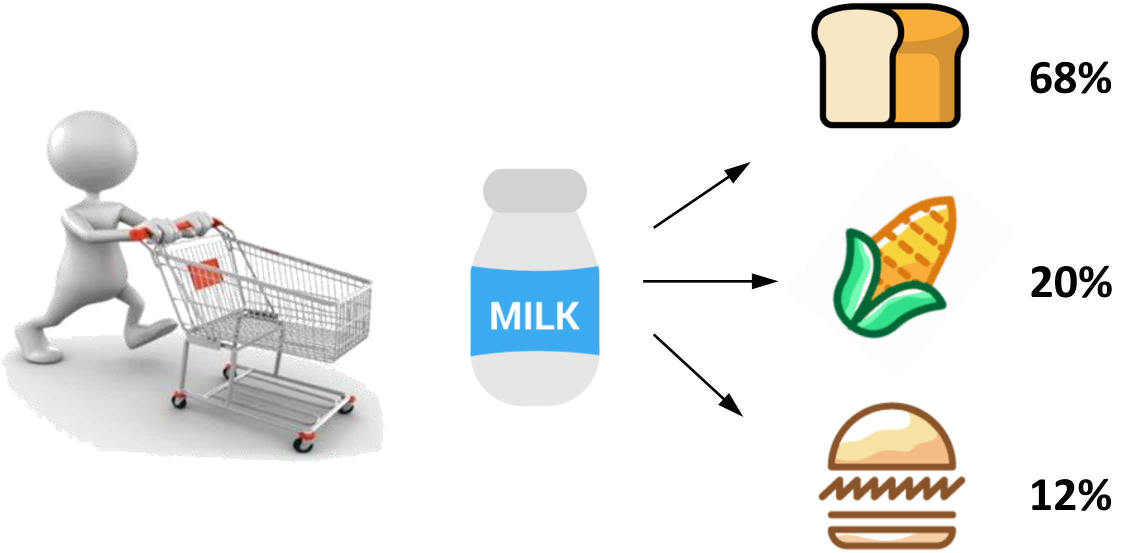
- **Drowning in data, but starving for knowledge!**

- **Data mining (knowledge discovery from data)**
  - Extraction of **interesting** (**non-trivial**, **implicit**, **unknown** and **potentially useful**) **patterns** or **knowledge** from data



Example: Association Rule Mining

- ➢ What items are frequently purchased together in Walmart?



68%

20%

12%

- ➢ An interesting association rule is Diaper → Beer.

Q&A