INTRODUCTION TO DATABASE SYSTEMS

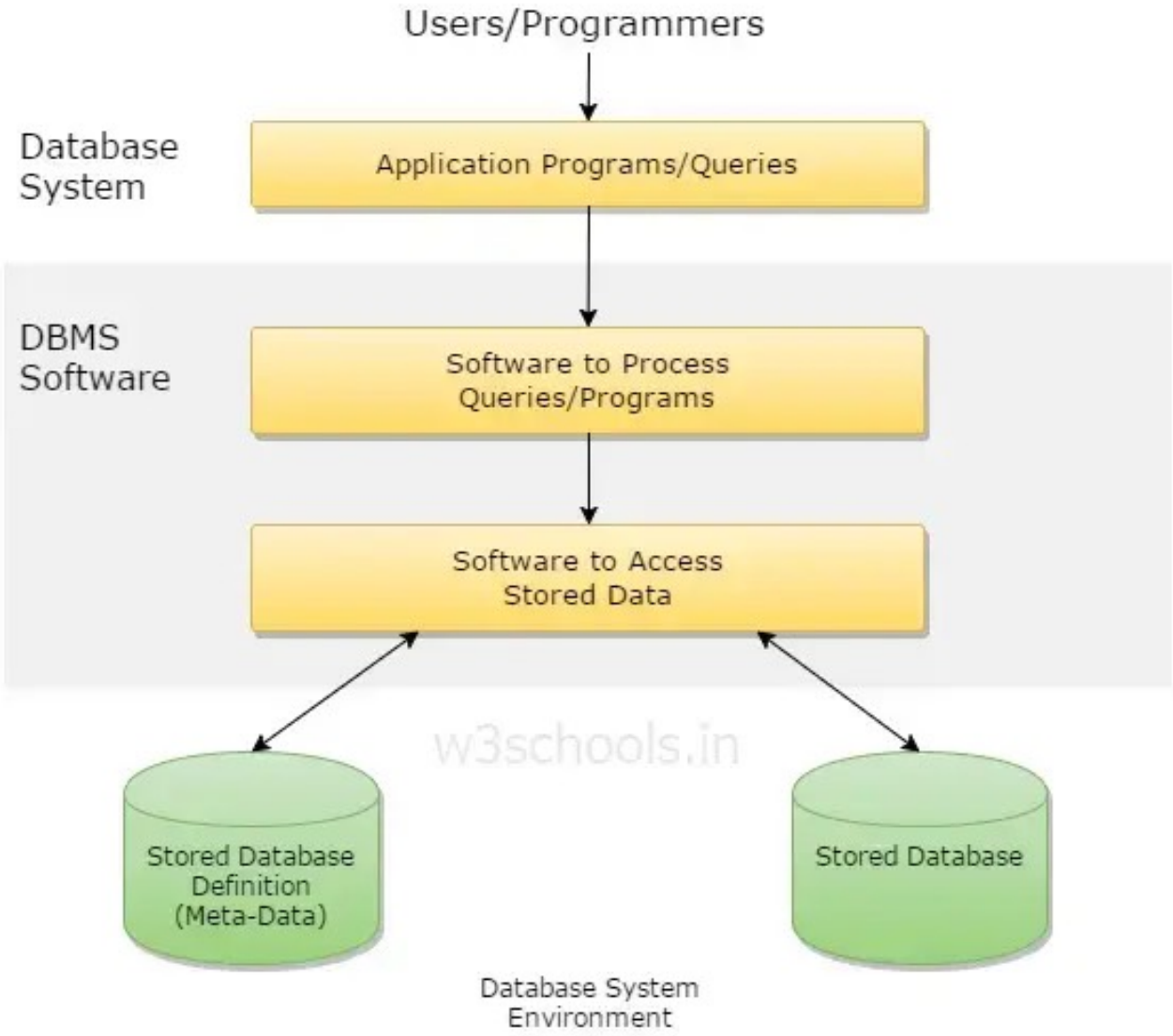
EGCO321 DATABASE SYSTEMS

KANAT POOLSAWASD
DEPARTMENT OF COMPUTER ENGINEERING
MAHIDOL UNIVERSITY

WHAT IS DATABASE ?

- A database is a collection of related data.
- By data, we mean known facts that can be recorded and that have implicit meaning.
- A database management system (DBMS) is a collection of programs that enables users to create and maintain a database.

DATABASE SYSTEM ENVIRONMENT



* https://www.w3schools.in/dbms/database-environment

PRE-DATABASE ERA

- Imagine you want build an online shopping website
  - Maintain products/categories (price, picture, etc.)
  - Customers accounts
- File is uninterpreted, unstructured collection of information
- File operations: delete, catalog, create, rename, open, close, read, write, find, etc.
- Access methods: Algorithms to implement operations along with internal file organisation

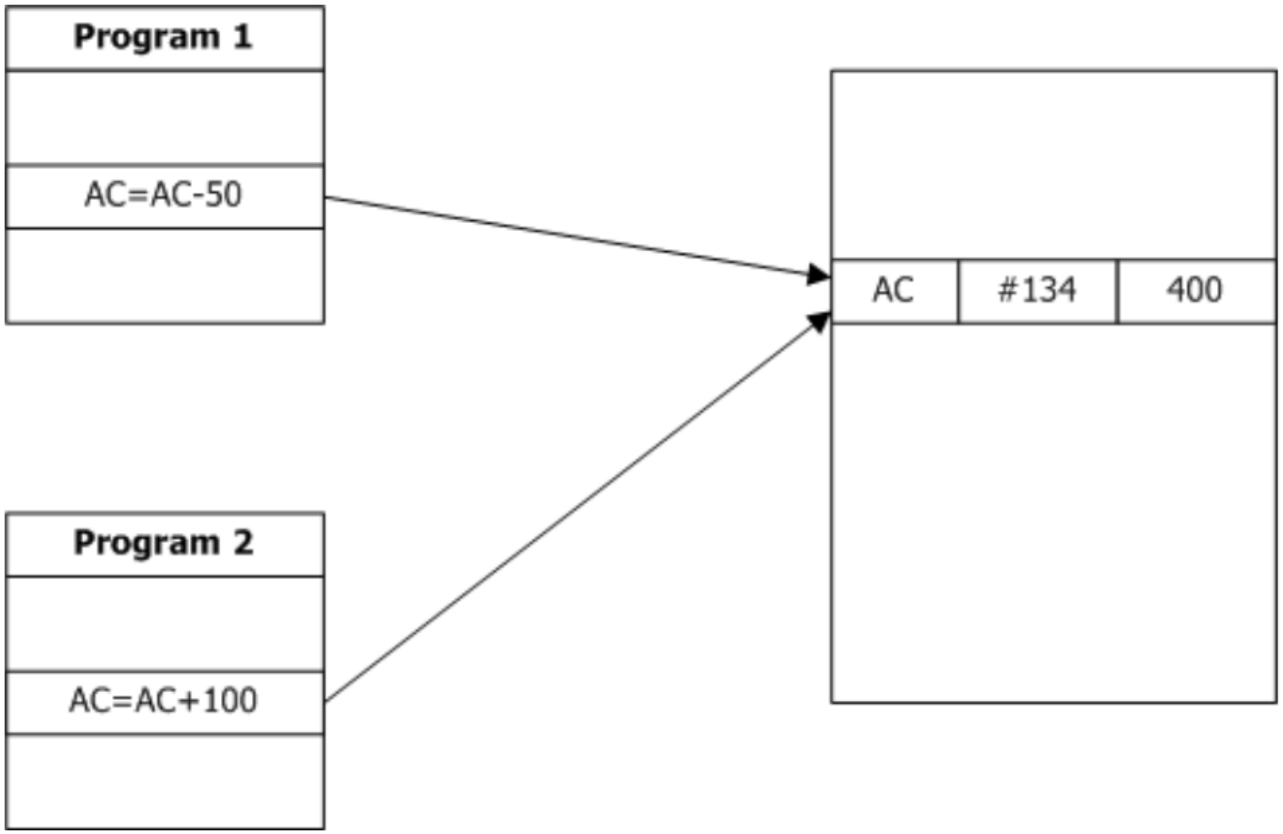
C++ FILE PROGRAMMING

- fopen() - open a file- specify how its opened (read/write) and type (binary/text)
- fclose() - close an opened file
- fread() - read from a file
- fwrite() - write to a file
- fseek() - move a file pointer to somewhere in a file

FILE MANAGEMENT SYSTEM PROBLEMS

- Multiple application (concurrent program) on the same file
- Data redundancy
- Data is not isolated from the access implementation (different format)

CONCURRENT PROGRAM EXECUTION



SECURITY PROBLEMS

- Allow access to the file only to the authorized personnel
- Ability to restrict access to parts of the record
- Ability to control operation usage by different users
- Protection from unauthorized use
- Protection from the derivation of unauthorised information

DATA INTEGRITY

- A database constraint is a logical constraint about the data expressed in a logical language.

```
STUDENT.AGE = 15
If (STUDENT.CLASS == 'CS43005') then
    (STUDENT.PRIOR_CLASS == 'CS31001')
```

- Database is consistent if data at each time satisfies all integrity constraints.
- Input to any application is a set of consistent data. An application output is a set of consistent data.

THREE ASPECTS TO STUDYING DBMS

- Data Modeling and Design
  - Data Models
  - Entity-Relationship Diagram
  - Normalization
- Database Languages and Querying
  - SQL
- DBMS Architecture and Internals
  - Three-Level Architecture (Physical/Conceptual/View)
  - Transaction Management
  - Storage Structures and Indexing
  - Recovery Techniques

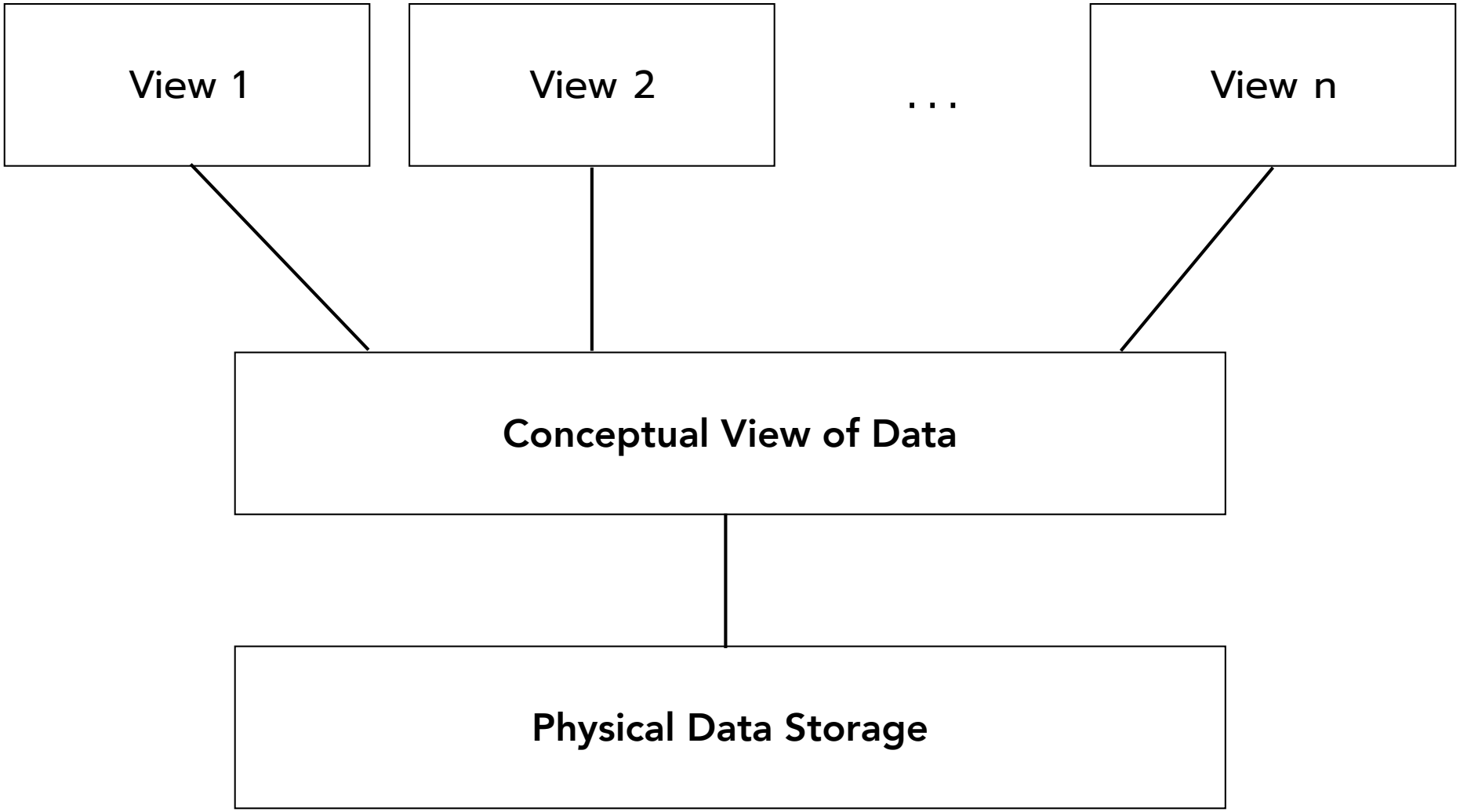
DATA VS. INFORMATION

- Data constitute building blocks of information
- Information produced by processing data
- Information reveals meaning of data
- Good, timely, relevant information key to decision making
- Good decision making key to organizational survival

DATA LEVELS (1)

- **Physical** – corresponds to the first view of data: How data is stored, how is it accessed, how data is modified, is data ordered, how data is allocated to computer memory and/or peripheral devices, how data items are actually represented (ASCI, EBCDIC,…)
- **Conceptual** – corresponds to the second view of data: What we want the data to express and what relationships between data we must express, what " story" data tells, are all data necessary for the "story' are discussed.
- **View** – corresponds to the third view of data: What part of the data is seen by a specific application

DATA LEVELS (2)

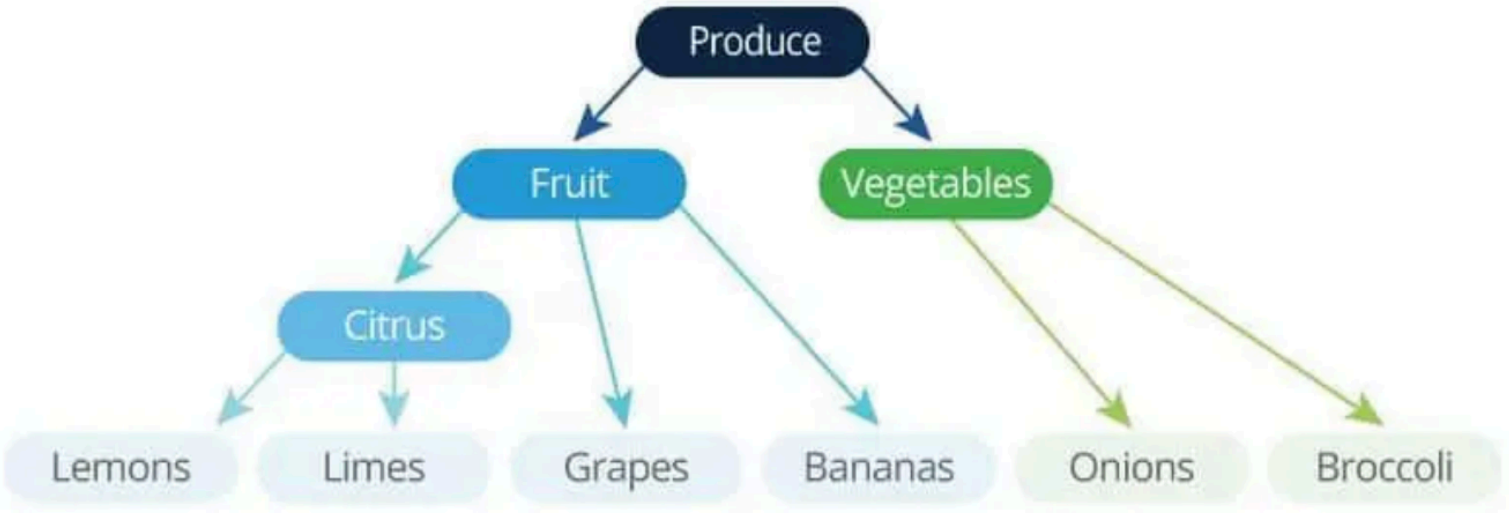


HISTORY OF DATABASE APPLICATIONS

- Pre-Database Era
- Hierarchical Database Model
- Network Database Model
- Relational Database Model
- Object-Oriented Database Model
- Data Warehouse /Data Mining
- Big Data

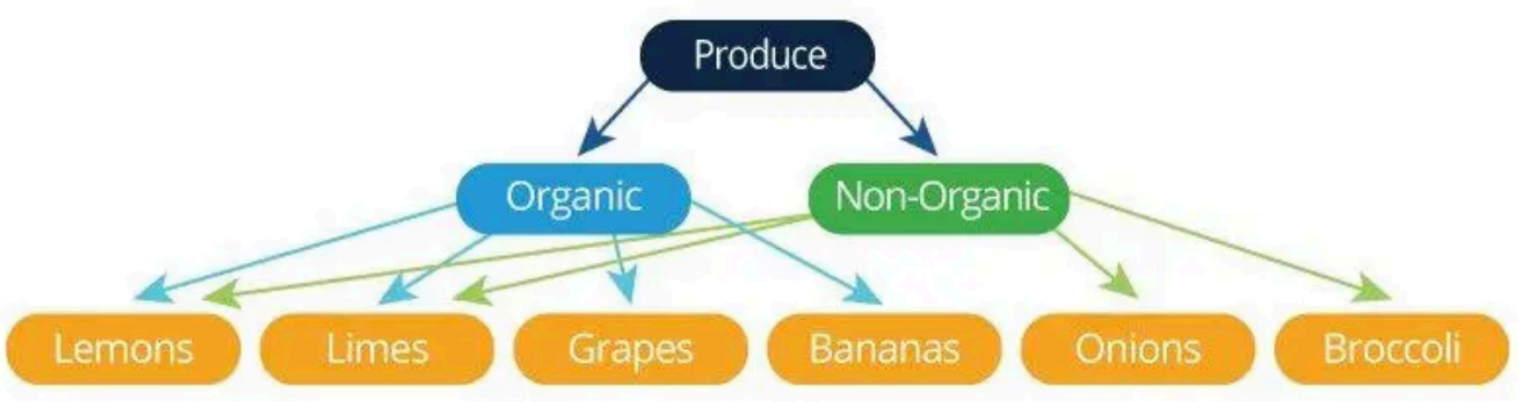
HIERARCHICAL DATABASE MODEL

- Early database applications using hierarchical and network systems



NETWORK DATABASE MODEL

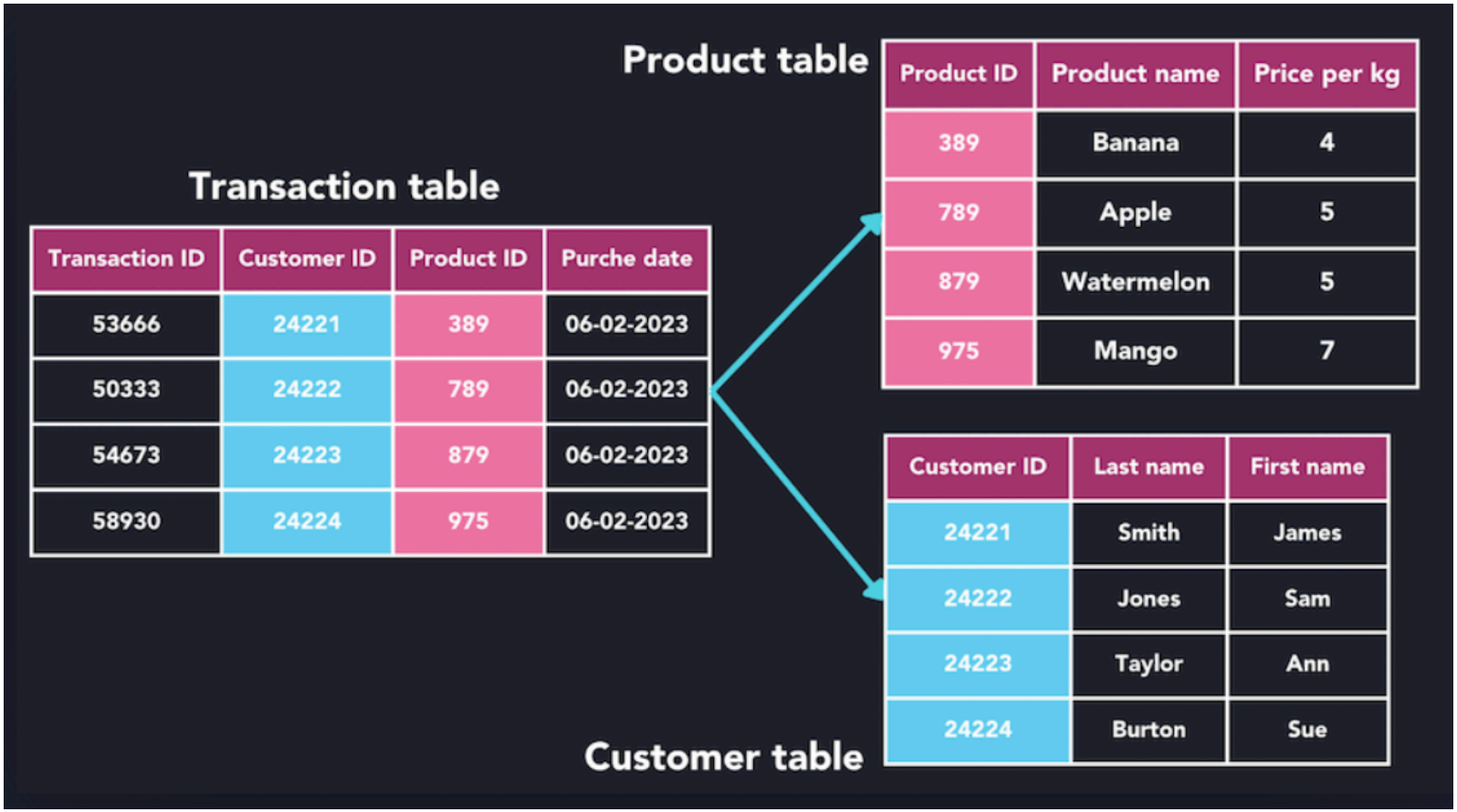
- Each record can have multiple parents
  - Composed of sets
  - Each set has owner record and member record
  - Member may have several owners



RELATIONAL DATABASE MODEL (1)

- Perceived by user as a collection of tables for data storage
- Tables are a series of row/column intersections
- Tables related by sharing common entity characteristic(s)

RELATIONAL DATABASE MODEL (2)



OBJECT-ORIENTED DATABASE MODEL

- Objects or abstractions of real-world entities are stored
  - Attributes describe properties
  - Collection of similar objects is a class
    - Methods represent real world actions of classes
    - Classes are organized in a class hierarchy
- Inheritance is ability of object to inherit attributes and methods of classes above it

NOSQL

- Stands for No-SQL or Not Only SQL.
- Class of non-relational data storage systems
  - E.g. MongoDB, Neo4j, …
- Usually do not require a fixed table schema nor do they use the concept of joins
  - Distributed data storage systems
- All NoSQL offerings relax one or more of the ACID properties.

DATA WAREHOUSE / DATA MINING

- **Data Warehousing** is the process of constructing and using a data warehouse. A data warehouse is constructed by integrating data from multiple heterogeneous sources that support analytical reporting, structured and/or ad hoc queries, and decision making.
- **Data Mining** is the practice of automatically searching large stores of data to discover patterns and trends that go beyond simple analysis. Data mining uses sophisticated mathematical algorithms to segment the data and evaluate the probability of future events.

BIG DATA

- Big data is a term that describes the large volume of data – both structured and unstructured – that inundates a business on a day-to-day basis. But it's not the amount of data that's important. It's what organizations do with the data that matters. Big data can be analyzed for insights that lead to better decisions and strategic business moves.

DATABASE SYSTEMS

- A database is a collection of stored operational data used by various applications and/or users by some particular enterprise or by a set of outside authorized applications and authorized users
- A **Database Management System (DBMS)** is a software system that manages execution of users applications to access and modify database data so that the data security, data integrity, and data reliability is guaranteed for each application and each application is written with an assumption that it is the only application active in the database.

DATABASE ARCHITECTURE

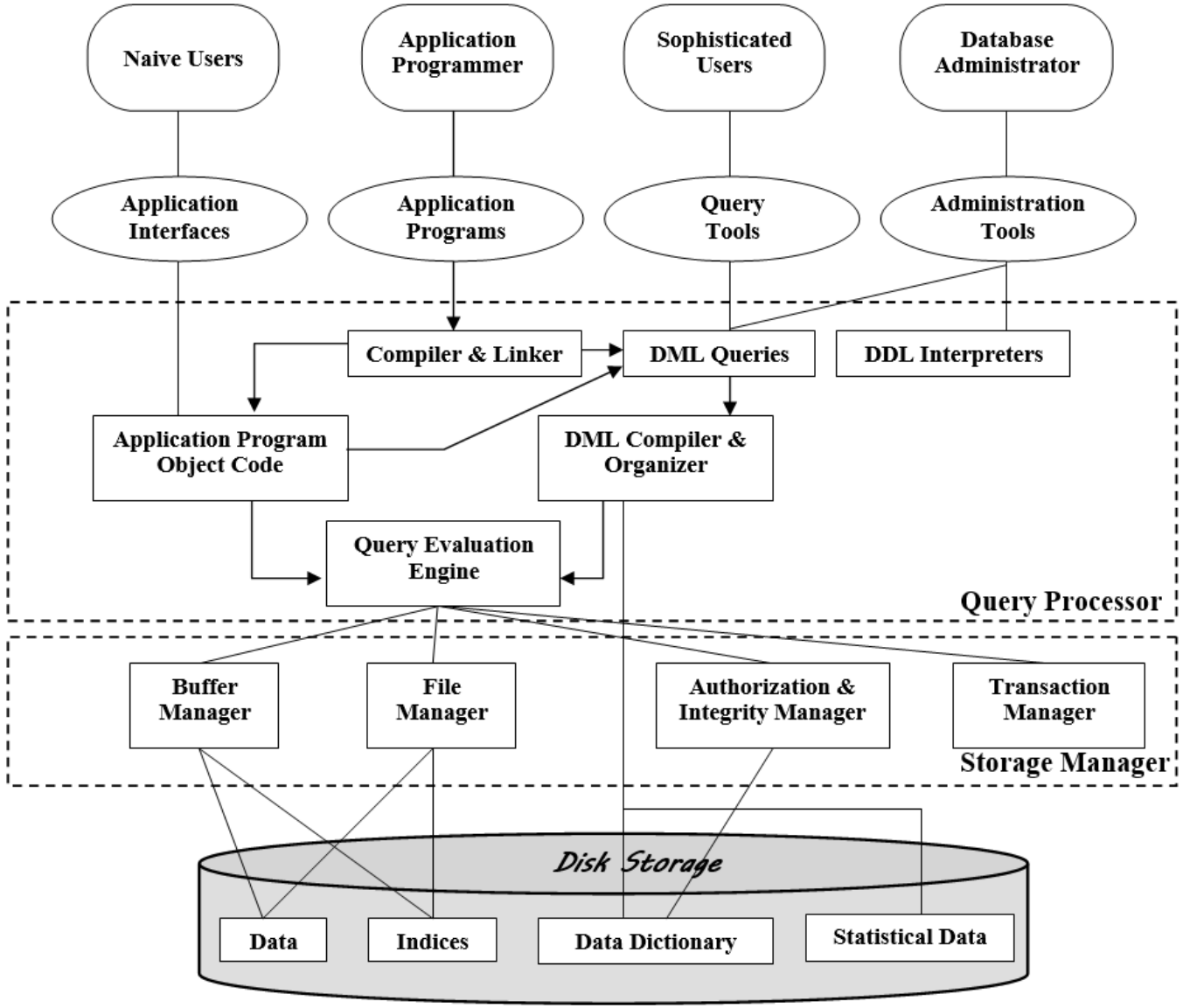


Figure: System Architecture

* https://onlineclassnotes.com/draw-database-system-architecture-dbms/

DATABASE ARCHITECTURE

- **Logical Components**
  - Data Definition Language (DDL)
  - Data Manipulation Language (DML)
  - Host Language Interface
  - Data Administrator
- **Physical Components**
  - Query Processor
    - Compiler
    - Optimiser
  - Management
    - Transaction Manager
    - File Manager
    - Buffer Manager
    - Authorisation and Integrity Manager

DATA DEFINITION LANGUAGE (DDL)

- Specification notation for defining the database schema

```
CREATE TABLE account(
    account-number    char(10),
    balance           integer)
```

- DDL compiler generates a set of tables stored in a data dictionary
- Data dictionary contains metadata (data about data)
  - Database schema
  - Data storage and definition language
    - Language in which the storage structure and access methods used by the database system are specified
    - Usually an extension of the data definition language

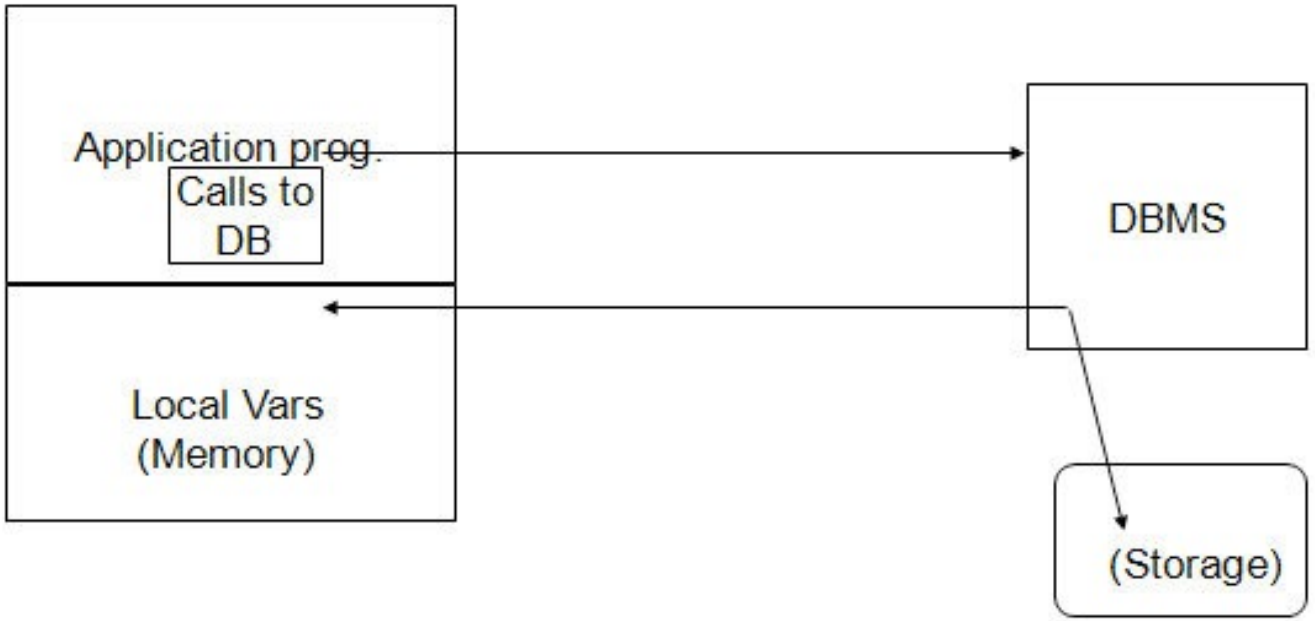
DATA MANIPULATION LANGUAGE (DML)

- SQL is a standard language for accessing and manipulating databases.

```
SELECT StudentID FROM MyClass
   WHERE DeptID = 'EGCO';
```

DATABASE HOST LANGUAGES

- Host language is completely general
- Query language — less general "non procedural" and optimizable



QUERY PROCESSOR

- Compiler – verifies whether a program or query is written in accordance with DDL and DML rules
- Optimizer – Finds the most effective way to access the required data and supply it in a user requested form. Monitors the query execution and modifies a query evaluation plan if necessary.

TRANSACTION MANAGER

- A transaction is a collection of operations that performs a single logical function in a database application
- Transaction-management component ensures that the database remains in a consistent (correct) state despite system failures (e.g., power failures and operating system crashes) and transaction failures.
- Concurrency-control manager controls the interaction among the concurrent transactions, to ensure the consistency of the database.

STORAGE MANAGEMENT

- Storage manager is a program module that provides the interface between the low-level data stored in the database and the application programs and queries submitted to the system.
- The storage manager is responsible to the following tasks:
  - Interaction with the file manager
  - Efficient storing, retrieving and updating of data

FILE MANAGER

- File Manager is responsible for mapping logical database units (objects, relations, etc.) into a set of low level files.
- It is responsible for maintenance of files and indexes on them. It should be able to create and destroy index and collect unused storage space to eliminate an unneeded gaps on disks.

BUFFER MANAGER

- Buffer Manager is responsible for the allocation and maintenance buffer space in a memory to facilitate processing database data by several concurrent applications.
- Buffer Manager decides when to load data from a buffer to a database or discard the data and under what conditions a new data should be put into a buffer

AUTHORIZATIONS AND INTEGRITY MANAGER

- This manager is responsible for granting an access to database or portions thereof only to authorized users and preventing the access to unauthorized users
- Integrity manager must assure data integrity during normal database operations as well as during the database failures

DATABASE ADMINISTRATOR

- Coordinates all the activities of the database system; the database administrator has a good understanding of the enterprise’s information resources and needs.
- Database administrator's duties include:
  - Schema definition
  - Storage structure and access method definition
  - Schema and physical organization modification
  - Granting user authority to access the database
  - Specifying integrity constraints
  - Acting as liaison with users
  - Monitoring performance and responding to changes in requirements

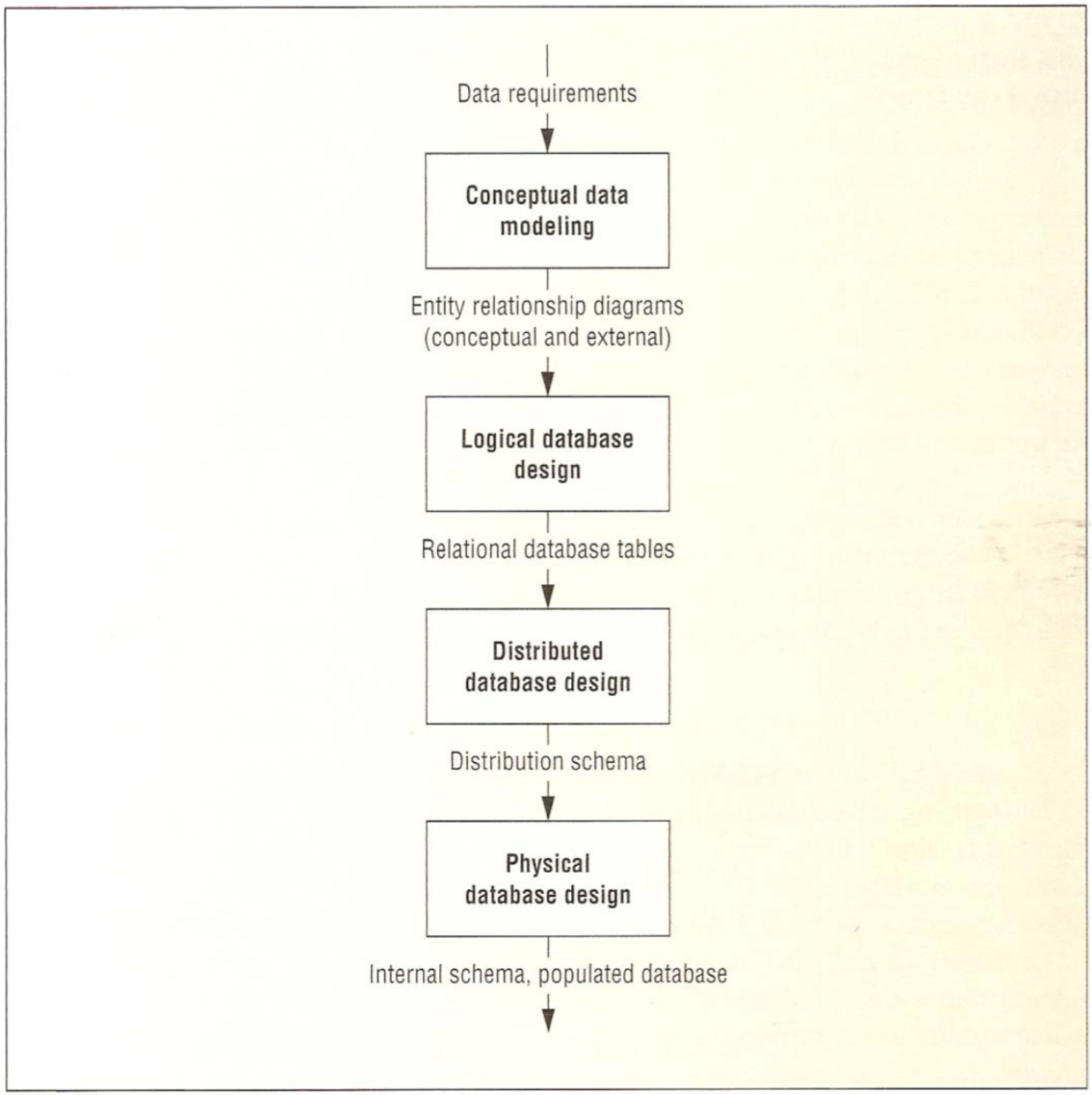
DATABASE USERS

- Actor on the scene
  - Database Administrators
  - Database Designers
  - End Users
  - System Analysts & Application Programmers (Software Engineer)
- Worker behind the scene
  - DBMS System Designers and Implementers
  - Tool Developers
  - Operators and Maintenance Personnel

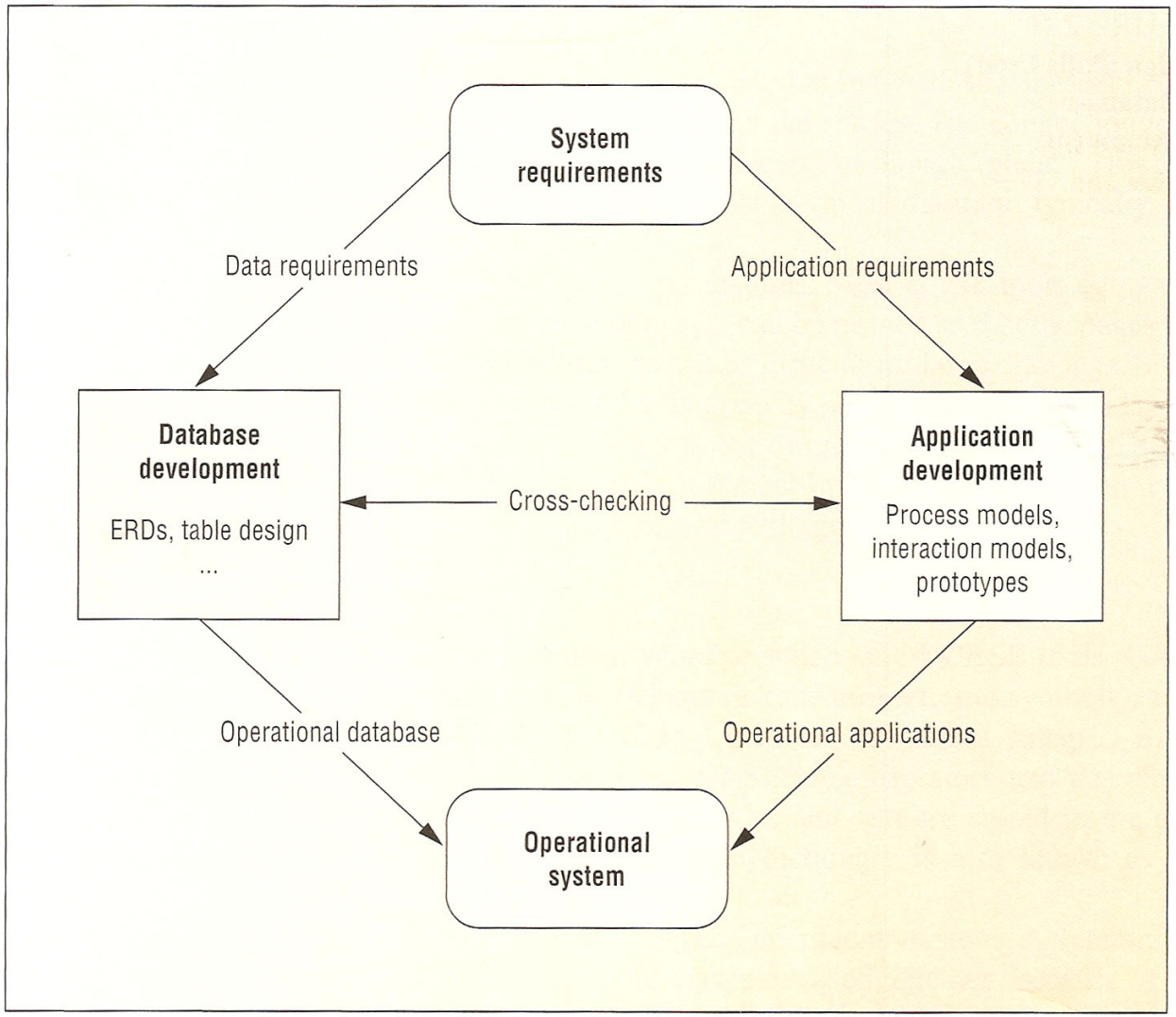
DATA INDEPENDENCE

- A database should have an identity separate from the applications (computer programs, forms, and reports) that use it.
- The separate identity allows the database definition to be changed without effecting related applications.

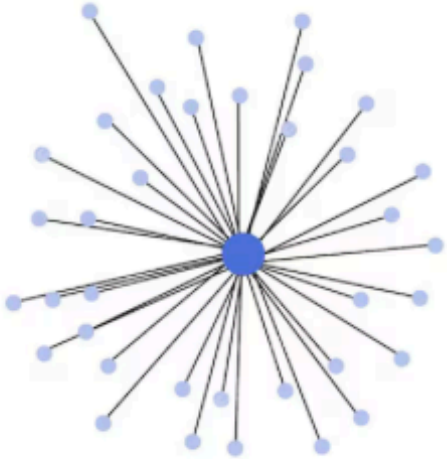
DATABASE DEVELOPMENT PROCESS



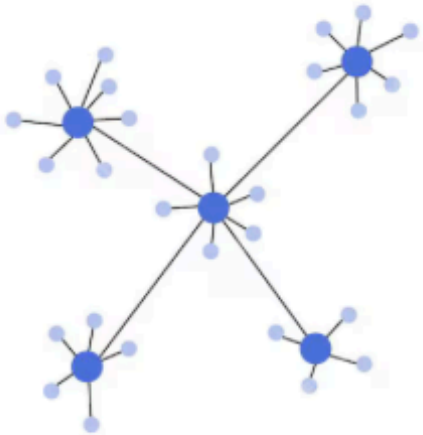
INTERACTION BETWEEN DATABASE AND APPLICATION DEVELOPMENT



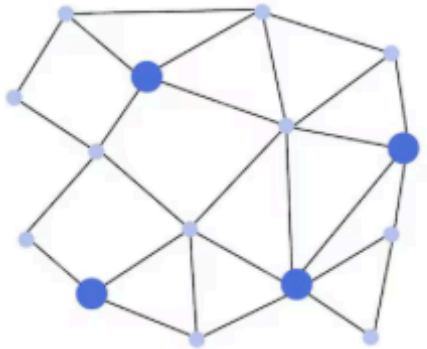
CENTRALIZED / DECENTRALIZED / DISTRIBUTED DATABASE SYSTEMS



Centralized



Decentralized



Distributed

CASE TOOLS FOR DATABASE DEVELOPMENT

- A CASE (computer-aided software engineering) tool is a software package that provides support for the design and implementation of information systems.
- Example CASE tools for database development:
  - Oracle
  - Microsoft SQL Server
  - MySQL / MariaDB
  - PostgreSQL
  - MongoDB
  - Etc.