Database System

TIF 1205

Introduction to Databases

TIK

- Menjelaskan konsep dasar database

Outline

- File Based Systems
- Database Approach
- Database Management System (DBMS)
- Types of Database Users

What is a Database?

What is a Database?

- “A set of information held in a computer”
  - Oxford English Dictionary
- “One or more large structured sets of persistent data, usually associated with software to update and query the data”
  - Free On-Line Dictionary of Computing
- “A collection of data arranged for ease and speed of search and retrieval by a computer. ”
  - American Heritage Science Dictionary

Uses of Databases

- Library catalogues
- Medical records
- Bank accounts
- Stock market data
- Personnel systems
- Product catalogues
- Telephone directories
- Train timetables
- Airline bookings
- Credit card details
- Student records
- Customer histories
- Stock market prices
- Discussion boards
- and many more...

Why Study Databases?

- Many computing applications deal with large amounts of information.
- Database systems give a set of tools for storing, searching and managing this information.
- Databases are a ‘core topic’ in computer science and IT.
- Basic concepts and skills with database systems are part of the skill set you will be ***assumed*** to have as a CS and IT graduate.

Why Study Databases?

- Almost everything we see on a computer uses a DBMS.
- Philosophy - the nature of reality that we are trying to model.
  - no entities, just statements of communicated fact. This could well be the next trend in programming
- relation rather than object oriented programming

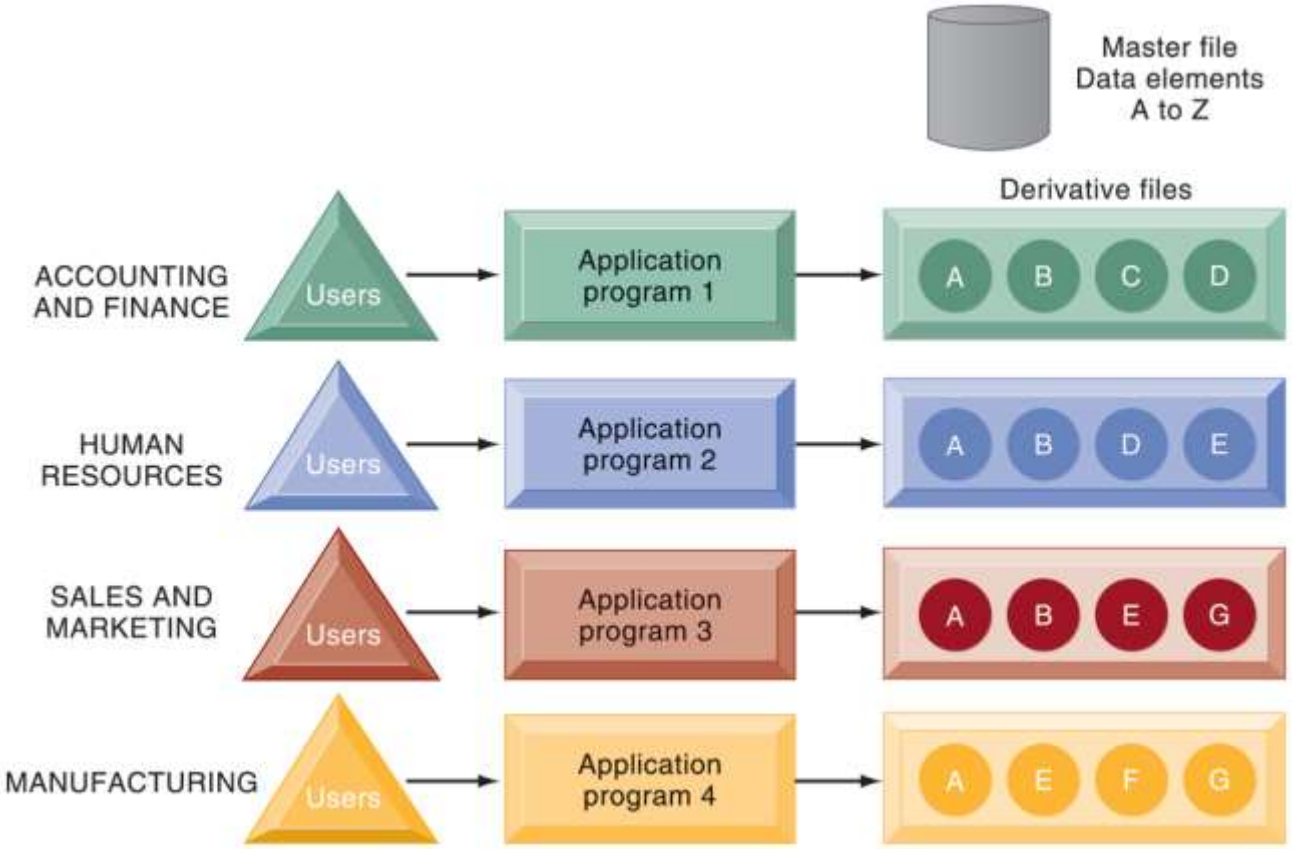
File Based Systems

- File-Based Approach
  - Sistem pemrosesan berbasis berkas (file based systems):
    - A collection of application programs that perform services for the system end-users such as the production of reports.
    - Each program defines and manages its own data.

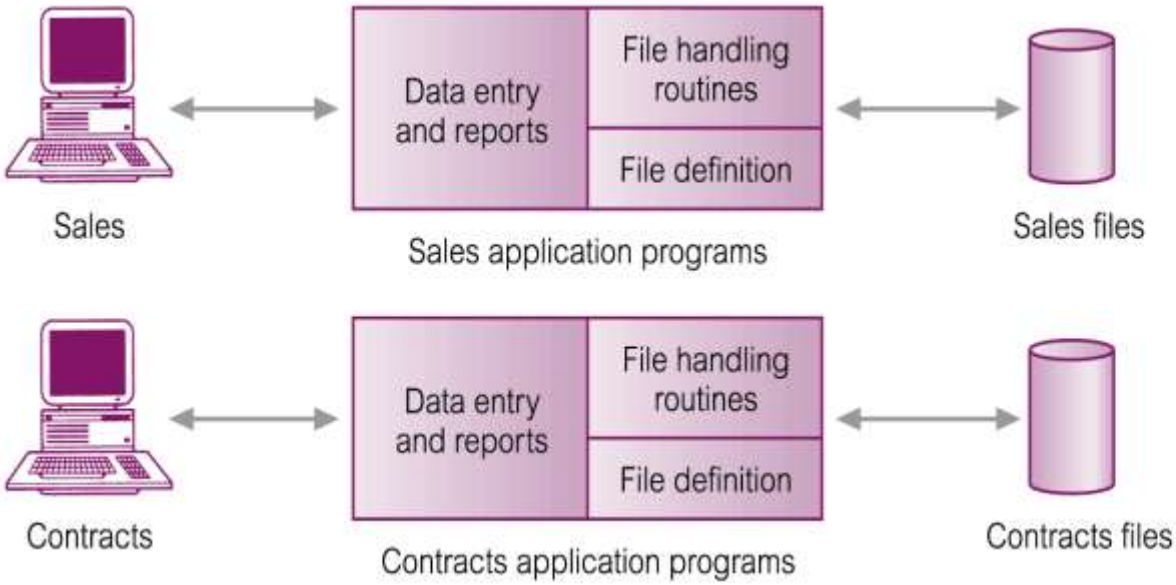
File Based Systems

- File based systems
  - Data is stored in files
  - Each file has a specific format
  - Programs that use these files depend on knowledge about that format

File Based Systems



File Based Systems



File-based processing.

Sales Files

PropertyForRent (propertyNo, street, city, postcode, type, rooms, rent, ownerNo)
PrivateOwner (ownerNo, fName, lName, address, telNo)
Client (clientNo, fName, lName, address, telNo, prefType, maxRent)

Contracts Files

Lease (leaseNo, propertyNo, clientNo, rent, paymentMethod, deposit, paid, rentStart, rentFinish, duration)
PropertyForRent (propertyNo, street, city, postcode, rent)
Client (clientNo, fName, lName, address, telNo)

Limitations of the File Based Systems

- Data redundancy:
  - Presence of duplicate data in multiple files
- Data inconsistency:
  - Same attribute has different values
- Program-data dependence:
  - When changes in program requires changes to data accessed by program

Limitations of the File-Based Approach

- Separation And Isolation Of Data
- Lack of flexibility
- Poor security
- Lack of data sharing and availability
- Incompatible File Formats

How do we resolve these problems?

- To become more effective, a new approach was required:
  - Database System
  - Database Management System (DBMS)

Database Approach

Definition

A shared collection of logically related data designed to meet the information requirements of an organization

Database Systems

- A DATABASE is a system to allow multiple users to share the SAME information.
  - Any item of data should only be stored once
  - Any changes to this data is thus available to all users

Database Management System (DBMS)

- A Database Management System (DBMS) is the software that implements a database
  - Oracle
  - DB2
  - MySQL
  - Ingres
  - PostgreSQL
  - Microsoft SQL Server
  - [MS Access]

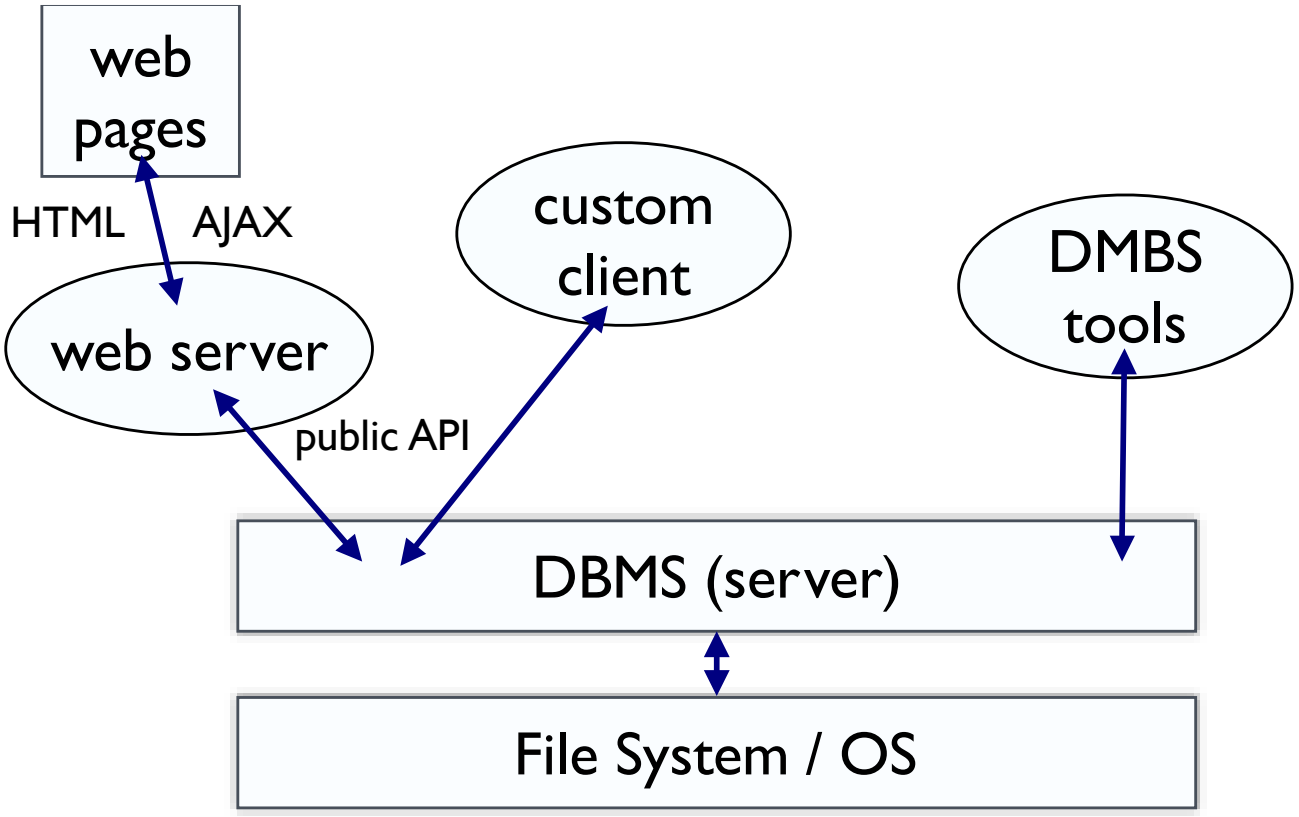
What a DBMS Does

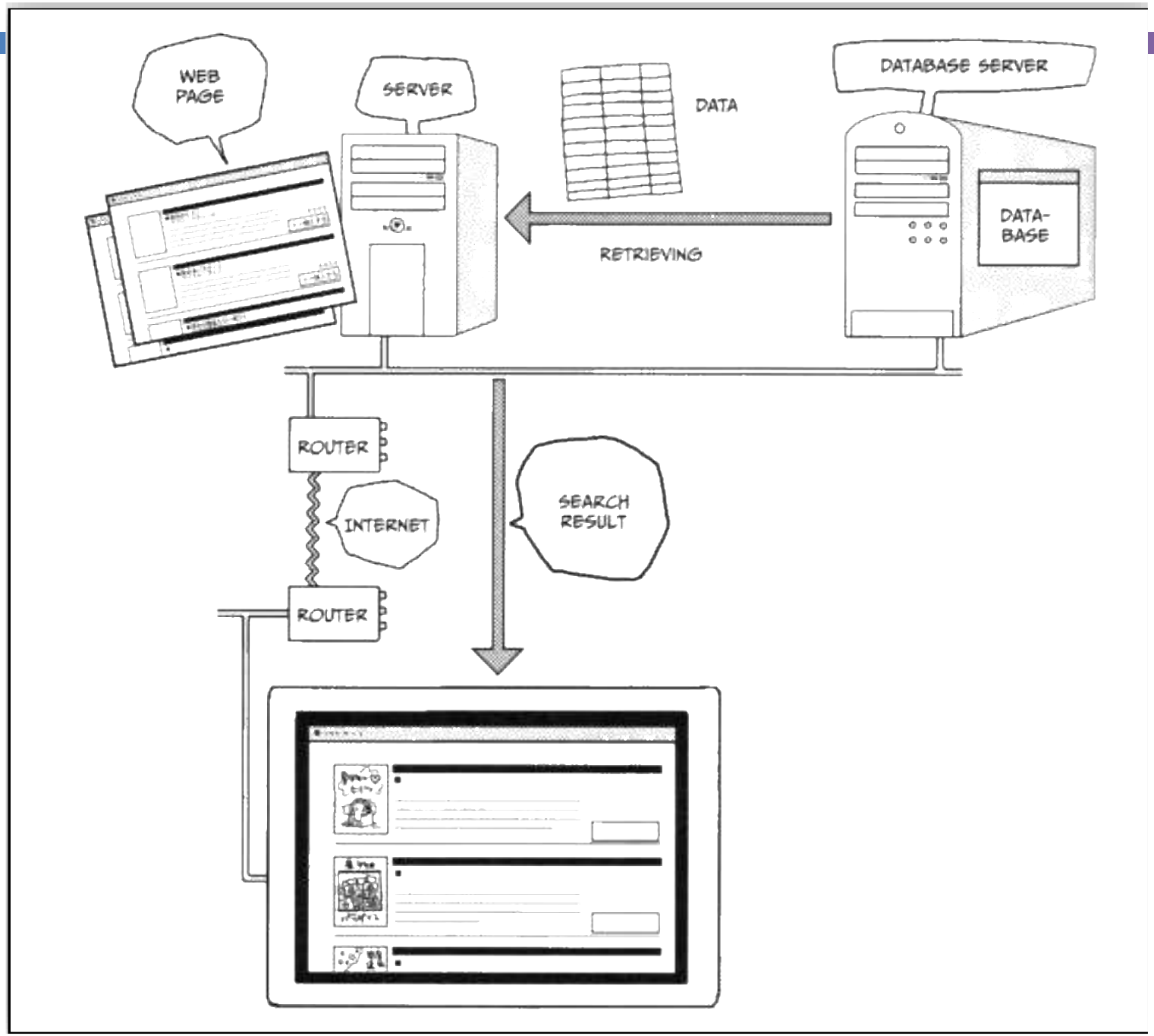
- Allows users to:
  - store data
  - manage change (updates)
  - organize data
  - retrieve data
  - retain privacy (security)

What a DBMS Does

- Provides:
  - Data Definition Language (DDL)
  - Data Manipulation Language (DML)
  - Data Control Language (DCL)
  - These may be the same language!

Where is the DBMS?




WEB PAGE
SERVER
DATA
DATABASE SERVER
DATA-BASE
RETRIEVING
ROUTER
INTERNET
ROUTER
SEARCH RESULT

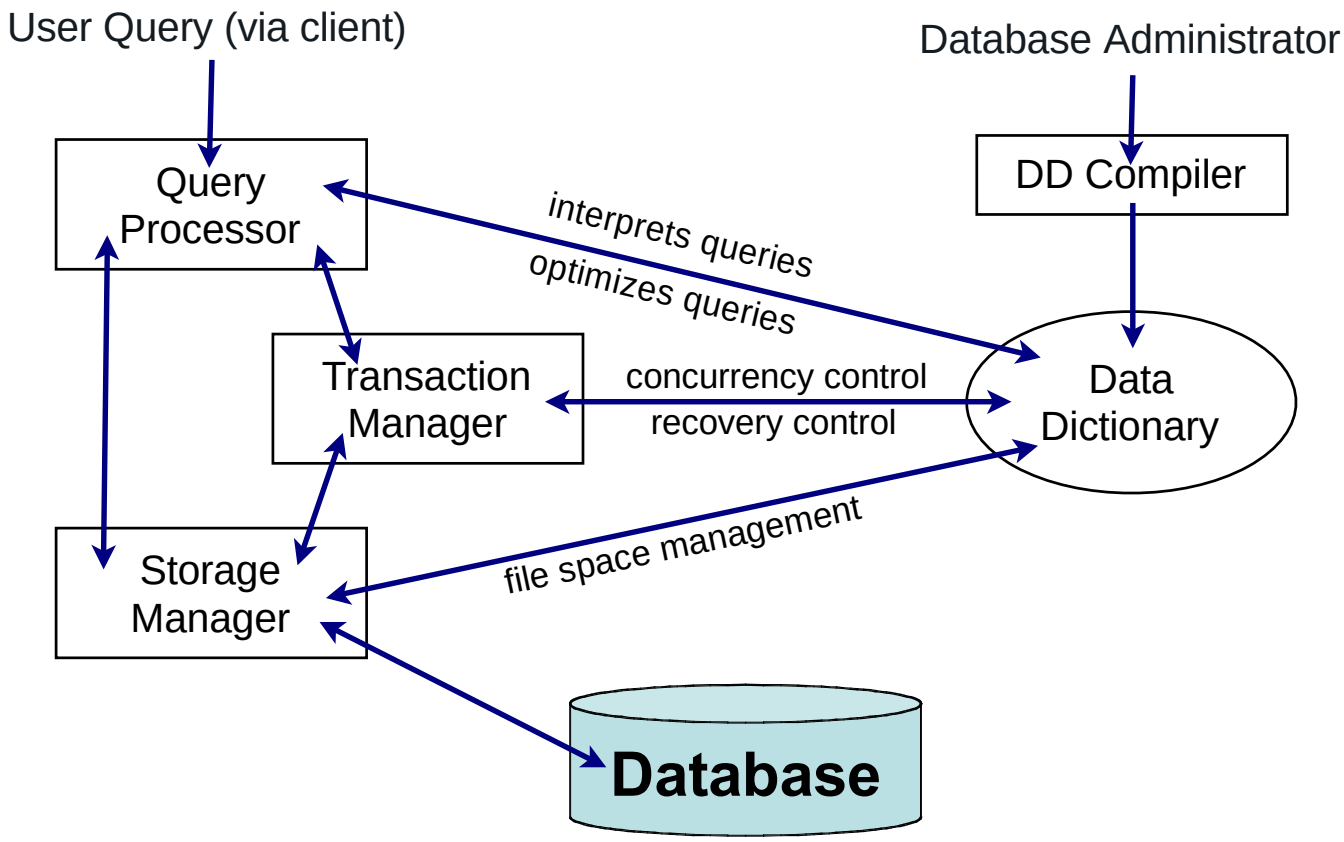
DBMS Components

- Data dictionary/system catalog
  - Stores information about database objects
- Query processor
  - Interprets and optimises user queries
- DML preprocessor
  - Converts DML statements in application programs into standard function calls. Interacts with query processor to generate code

DBMS Components

- DDL Compiler/Data Dictionary Compiler
  - Produces the data dictionary / system catalog
  - E.g. Converts DDL statements into tables
- Catalog Manager
  - Manages access to and maintains system catalog

Interaction of DBMS Components



DBMS Functions

- Data storage, retrieval and update
- User accessible catalog
- Transaction support (all or nothing)
- Concurrency control (correct updates)
- Recovery services (if something goes wrong)

DBMS Functions

- Authorisation services (security)
- Support communication software (remote applications)
- Integrity services (allow rules to be enforced)
- Promote data independence (from structure)
- Utility services (import/export, monitoring and logs, statistical analysis, consolidate files/indexes, reporting tools,...)

Provided Languages

- Data Definition Language (DDL) – Specify database format
- Data Manipulation Language (DML) – Specify and retrieve database contents
- Data Control Language (DCL) – Specify access controls
- Which are often all one piece of software – E.g. SQL

Types of Database Users

- Database systems programmers
  - writes the DBMS software
- Database administrators (DBA)
  - designs and manages the database system
- Applications developers
  - write the client software that allows end users to interface with the DBMS
- End users
  - use the information provided by the database to achieve a personal or organisational goal

Basic Definitions

- **Database**: A collection of related data.
- **Data**: Known facts that can be recorded and have an implicit meaning.
- **Mini-world**: Some part of the real world about which data is stored in a database. For example, student grades and transcripts at a university.
- **Database Management System (DBMS)**: A software package/ 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.

Database Environment

TIK

- Menjelaskan Three-level Database Architecture, Schemas, Data Independence, Data model, Database Language

Outline

- Three-level Database Architecture
- Schemas
- Data Independence
- Data model
- Database Language
- Database Design

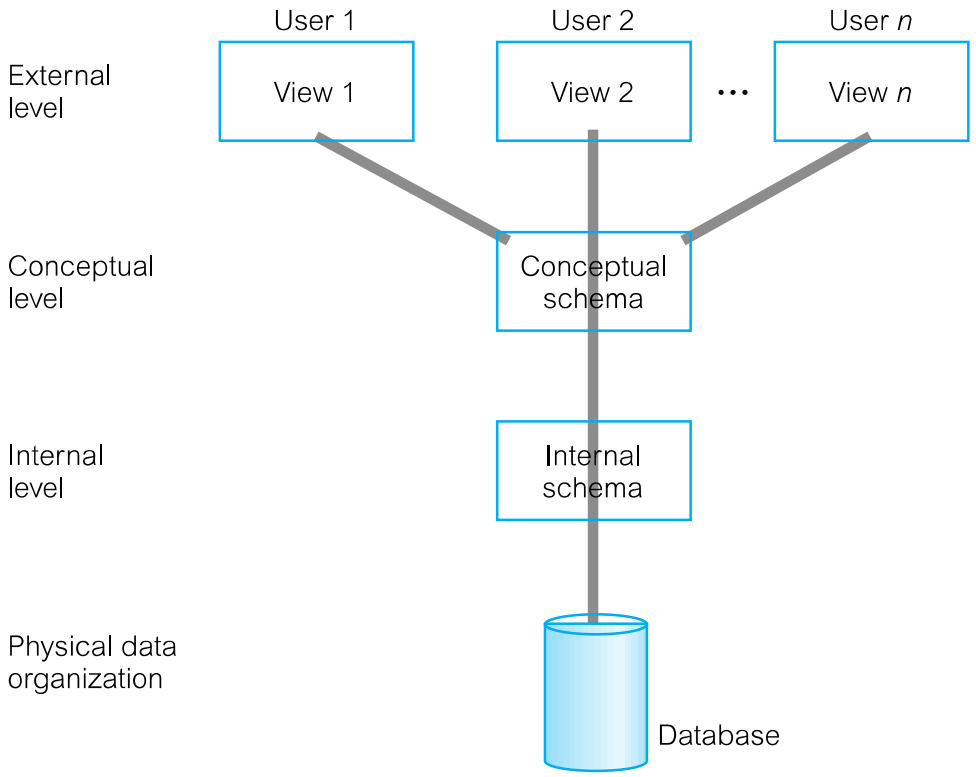
ANSI / SPARC Architecture

- Proposed a framework for DBMS in 1975
  - American National Standards Institute
  - Standards Planning Requirements Committee
- To separate each user's view of the database from the way the database is physically represented.

ANSI / SPARC Architecture

- Three tier architecture
  - Internal level
    - for systems designers
  - Conceptual level
    - for database designers
  - External level
    - for database users

ANSI / SPARC Architecture



Schemas

- **Database Schema**: The *description* of a database. Includes descriptions of the database structure and the constraints that should hold on the database.
- **Schema Diagram**: A diagrammatic display of (some aspects of) a database schema.
- **Schema Construct**: A component of the schema or an object within the schema, e.g., STUDENT, COURSE.
- **Database State/Snapshot**: The actual data stored in a database at a *particular moment in time*. Also called the **current set of occurrences/instances**).

An Example of Database

STUDENT	Name	StudentNumber	Class	Major
	Smith	17	1	CS
	Brown	8	2	CS

COURSE	CourseName	CourseNumber	CreditHours	Department
	Intro to Computer Science	CS1310	4	CS
	Data Structures	CS3320	4	CS
	Discrete Mathematics	MATH2410	3	MATH
	Database	CS3380	3	CS

SECTION	SectionIdentifier	CourseNumber	Semester	Year	Instructor
	85	MATH2410	Fall	98	King
	92	CS1310	Fall	98	Anderson
	102	CS3320	Spring	99	Knuth
	112	MATH2410	Fall	99	Chang
	119	CS1310	Fall	99	Anderson
	135	CS3380	Fall	99	Stone

GRADE_REPORT	StudentNumber	SectionIdentifier	Grade
	17	112	B
	17	119	C
	8	85	A
	8	92	A
	8	102	B
	8	135	A

PREREQUISITE	CourseNumber	PrerequisiteNumber
	CS3380	CS3320
	CS3380	MATH2410
	CS3320	CS1310

Schema diagram

STUDENT

Name	StudentNumber	Class	Major

COURSE

CourseName	CourseNumber	CreditHours	Department

PREREQUISITE

CourseNumber	PrerequisiteNumber

SECTION

SectionIdentifier	CourseNumber	Semester	Year	Instructor

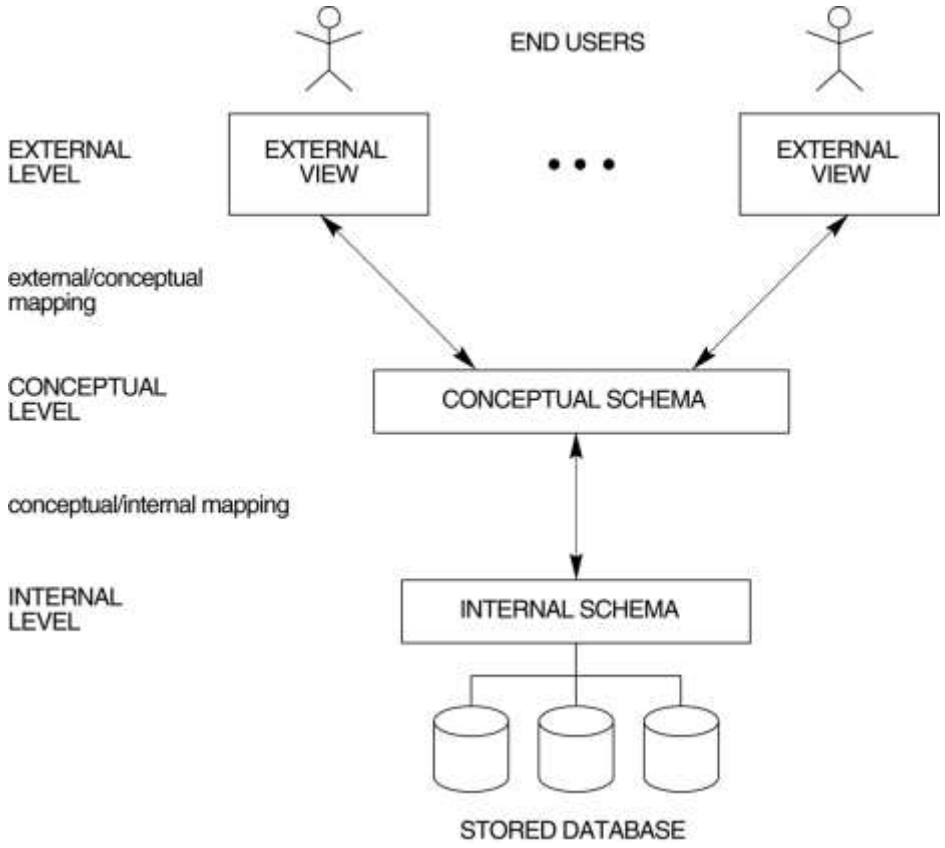
GRADE_REPORT

StudentNumber	SectionIdentifier	Grade

Database Schema Vs. Database State

- **Database State**: Refers to the content of a database at a moment in time.
- **Initial Database State**: Refers to the database when it is loaded
- **Valid State**: A state that satisfies the structure and constraints of the database.
- **Distinction**
  - The database schema changes very infrequently. The database state changes every time the database is updated.
  - Schema is also called **intension**, whereas state is called **extension.**

The Three-schema Architecture



Three-Schema Architecture

- Defines DBMS schemas at *three levels*:
  - **Internal schema** at the internal level to describe physical storage structures and access paths. Typically uses a *physical* data model.
  - **Conceptual schema** at the conceptual level to describe the structure and constraints for the *whole* database for a community of users. Uses a *conceptual* or an *implementation* data model.
  - **External schemas** at the external level to describe the various user views. Usually uses the same data model as the conceptual level.

External Level

- Provides a view of the database tailored to a user
  - Parts of the data may be hidden
  - Used by end users and application programmers
- External Schema:
  - For example:
    - **Create View myView as {**

      SELECT Name FROM Employee

      }

Conceptual Level

- Deals with the organisation of the entire database content
  - Abstractions are used to remove unnecessary details of the internal level
  - Used by DBAs and application programmers
- Conceptual Schema:
  - For example:

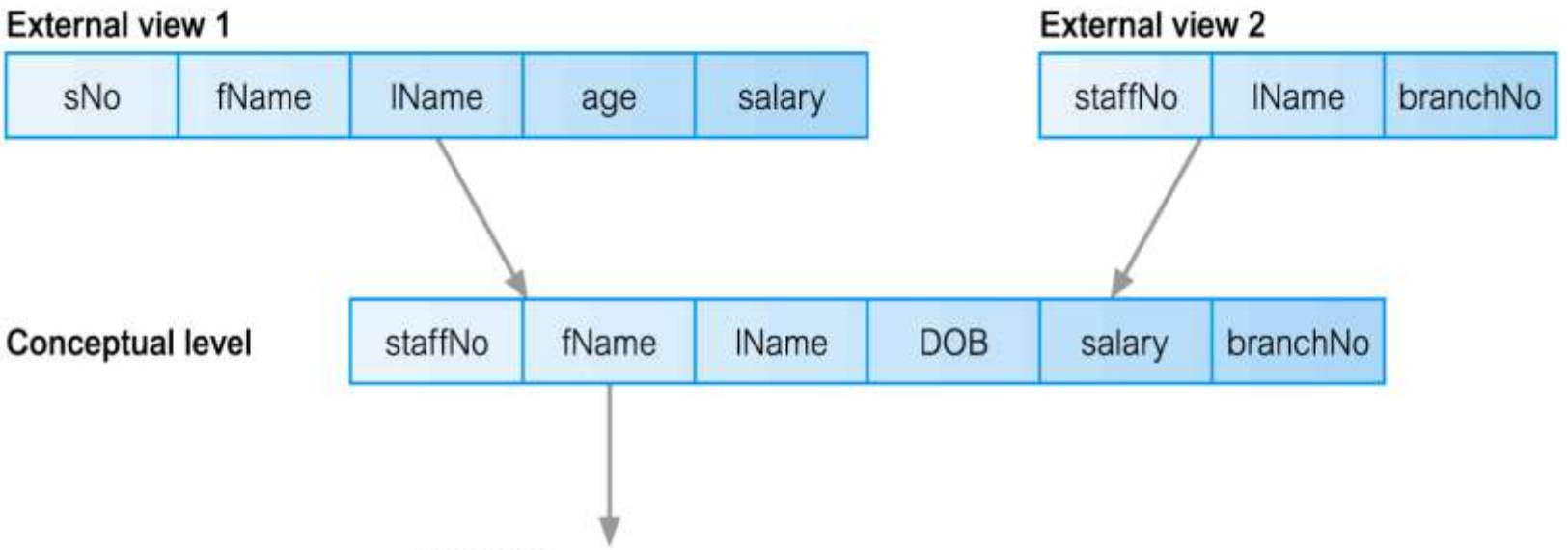
CREATE TABLE Employee (Name VARCHAR(25), Salary REAL, Department VARCHAR(10))

Internal Level

- Deals with physical storage of data
  - Structure of records on disk - files, pages, blocks
  - Indexes and ordering of records
  - Used by database system programmers
- Internal Schema:
  - For example:

```
RECORD EMP
LENGTH=44
HEADER: BYTE(5) OFFSET=0
NAME: BYTE(25) OFFSET=5
SALARY: FULLWORD OFFSET=30
DEPT: BYTE(10) OFFSET=34
```

Differences Between The Three Levels.



Internal level

```
struct STAFF {
        int staffNo;
        int branchNo;
        char fName [15];
        char lName [15];
        struct date dateOfBirth;
        float salary;
        struct STAFF *next;              /* pointer to next Staff record */
};
index staffNo; index branchNo;           /* define indexes for staff */
```

Mapping

- **Mappings** among schema levels are needed to transform requests and data. Programs refer to an external schema, and are mapped by the DBMS to the internal schema for execution.
- Mappings translate information from one level to the next
  - External/Conceptual
  - Conceptual/Internal

Mapping

- E.g., A request for a database retrieval, the DBMS will transform the request to the conceptual and physical levels, and extract the data from the stored database and reformatted the data to match the user's external view.

- These mappings provide data independence

Data Independence

- Ability that allows application programs not being affected by changes in irrelevant parts of the conceptual data representation, data storage structure and data access methods
- Upper levels are unaffected by changes to lower levels.
- Invisibility (transparency) of the details of entire database organization, storage structure and access strategy to the users
  - Both logical and physical

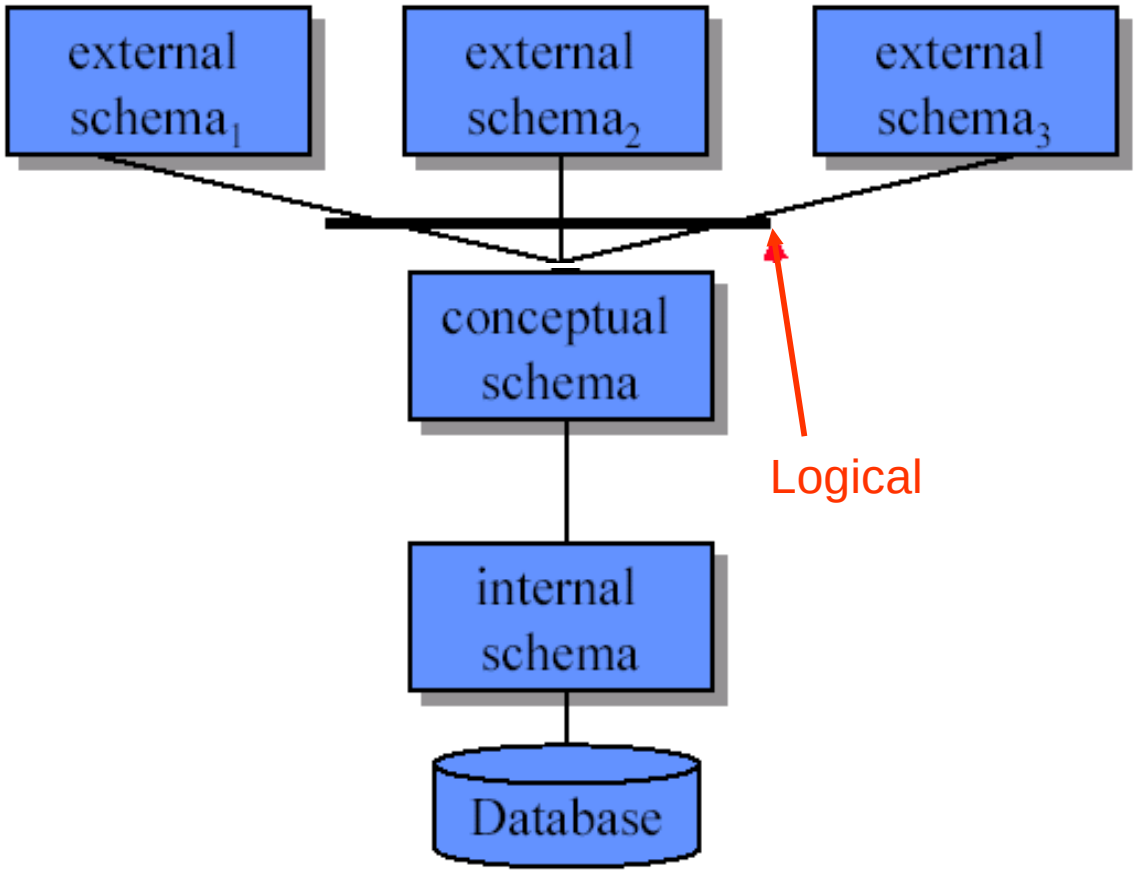
Data Independence

- Two kinds of data independence
  - **Logical**
  - **Physical**

Data Independence

- **Logical Data Independence**
  - Refers to immunity of external schemas to changes in conceptual schema.
  - Conceptual schema changes (e.g. addition/removal of entities).
  - Should not require changes to external schema or rewrites of application programs.

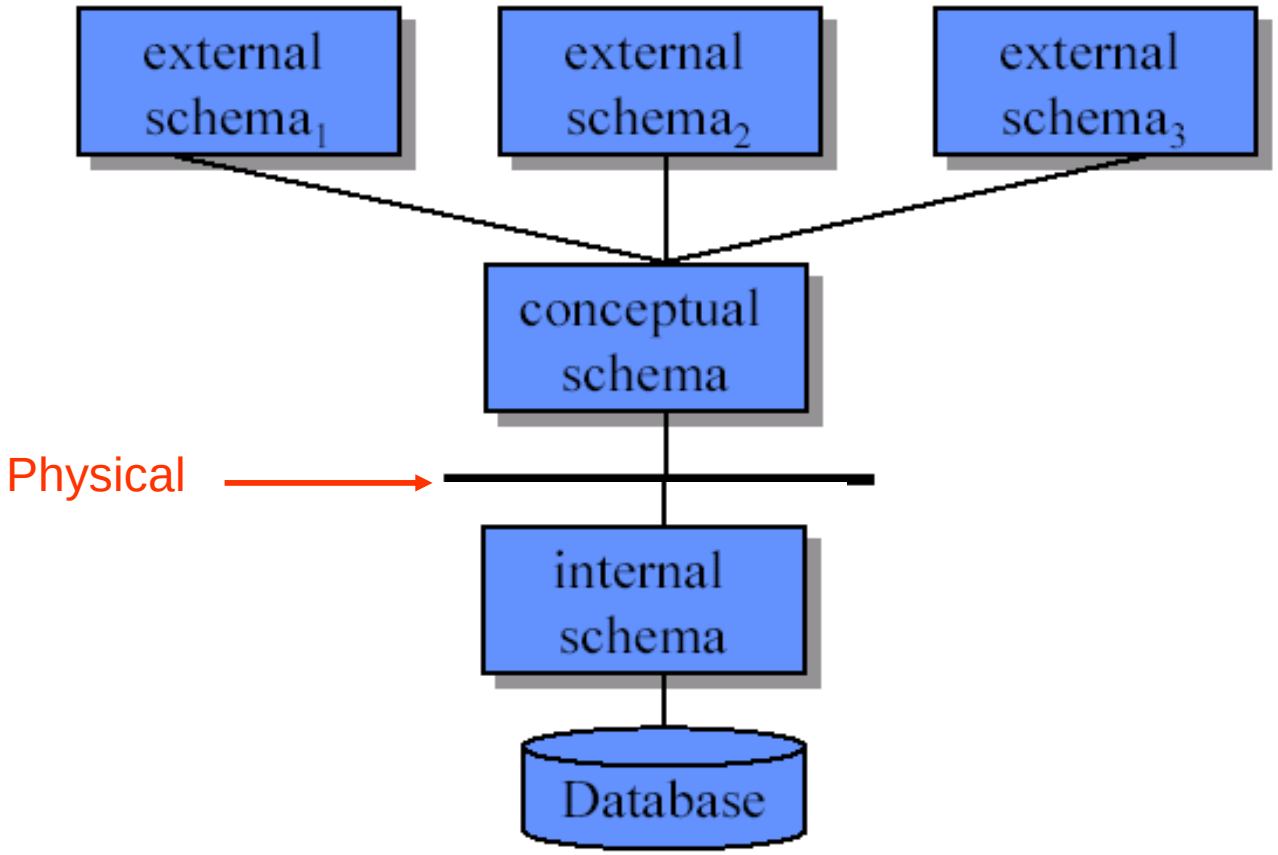
Logical Data Independence



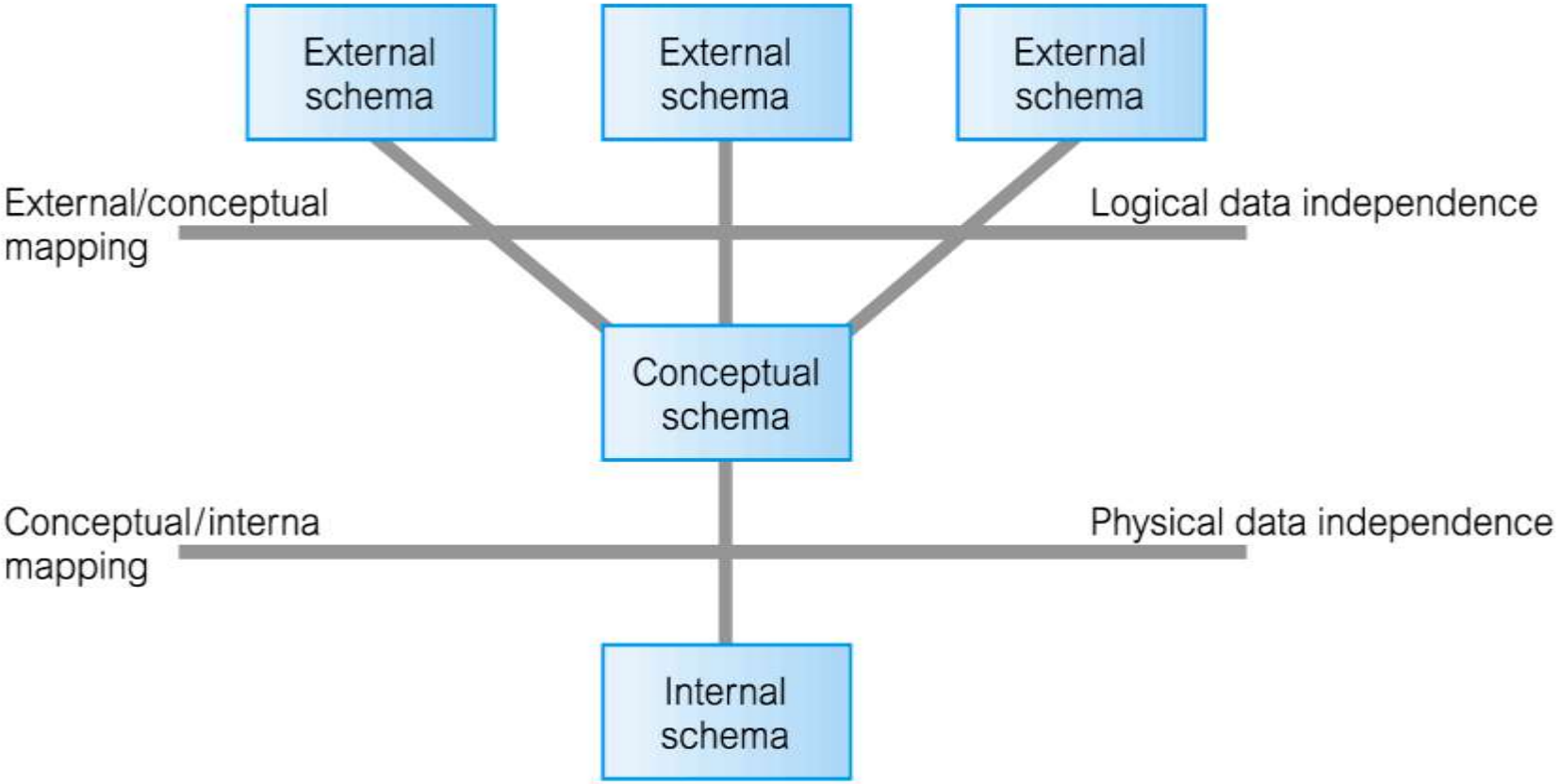
Data Independence

- **Physical Data Independence**
  - Refers to immunity of conceptual schema to changes in the internal schema.
  - Internal schema changes (e.g. using different file organizations, storage structures/devices).
  - Should not require change to conceptual or external schemas.
  - Physical files selalu perlu di reorganized, bisa karena disk space sudah penuh atau perlu penambahan / perubahan access structure untuk tujuan meningkatkan kinerja pencarian/perbaikan data

Physical Data Independence



Data Independence and the ANSI-SPARC Three-Level Architecture



Database Design

Data Model

- A **model** is a representation of ‘real world’ objects and events, and their associations.

- **Data Model**:
  - Integrated collection of concepts for describing data, relationships between data, and constraints on the data in an organization.
  - A set of concepts to describe the structure of a database, and certain constraints that the database should obey.

The Database System Development Lifecycle

Stage	Main activities
Database planning	Planning how the stages of the lifecycle can be realized most efficiently and effectively.
System definition	Specifying the scope and boundaries of the database system, including the major user views, its users, and application areas.
Requirements collection and analysis	Collection and analysis of the requirements for the new database system.
Database design	Conceptual, logical, and physical design of the database.
DBMS selection (optional)	Selecting a suitable DBMS for the database system.
Application design	Designing the user interface and the application programs that use and process the database.
Prototyping (optional)	Building a working model of the database system, which allows the designers or users to visualize and evaluate how the final system will look and function.
Implementation	Creating the physical database definitions and the application programs.
Data conversion and loading	Loading data from the old system to the new system and, where possible, converting any existing applications to run on the new database.
Testing	Database system is tested for errors and validated against the requirements specified by the users.
Operational maintenance	Database system is fully implemented. The system is continuously monitored and maintained. When necessary, new requirements are incorporated into the database system through the preceding stages of the lifecycle.

Database Design

- Three phases of database design:
  - Conceptual database design
  - Logical database design
  - Physical database design.

Conceptual Database Design

- Process of constructing a model of the data used in an enterprise, independent of all physical considerations.
- Data model is built using the information in users' requirements specification.
- Conceptual data model is source of information for logical design phase.

Logical Database Design

- Process of constructing a model of the data used in an enterprise based on a specific data model (e.g. relational), but independent of a particular DBMS and other physical considerations.
- Conceptual data model is refined and mapped on to a logical data model.

Physical Database Design

- Process of producing a description of the database implementation on secondary storage.
- Describes base relations, file organizations, and indexes used to achieve efficient access to data. Also describes any associated integrity constraints and security measures.
- Tailored to a specific DBMS system.

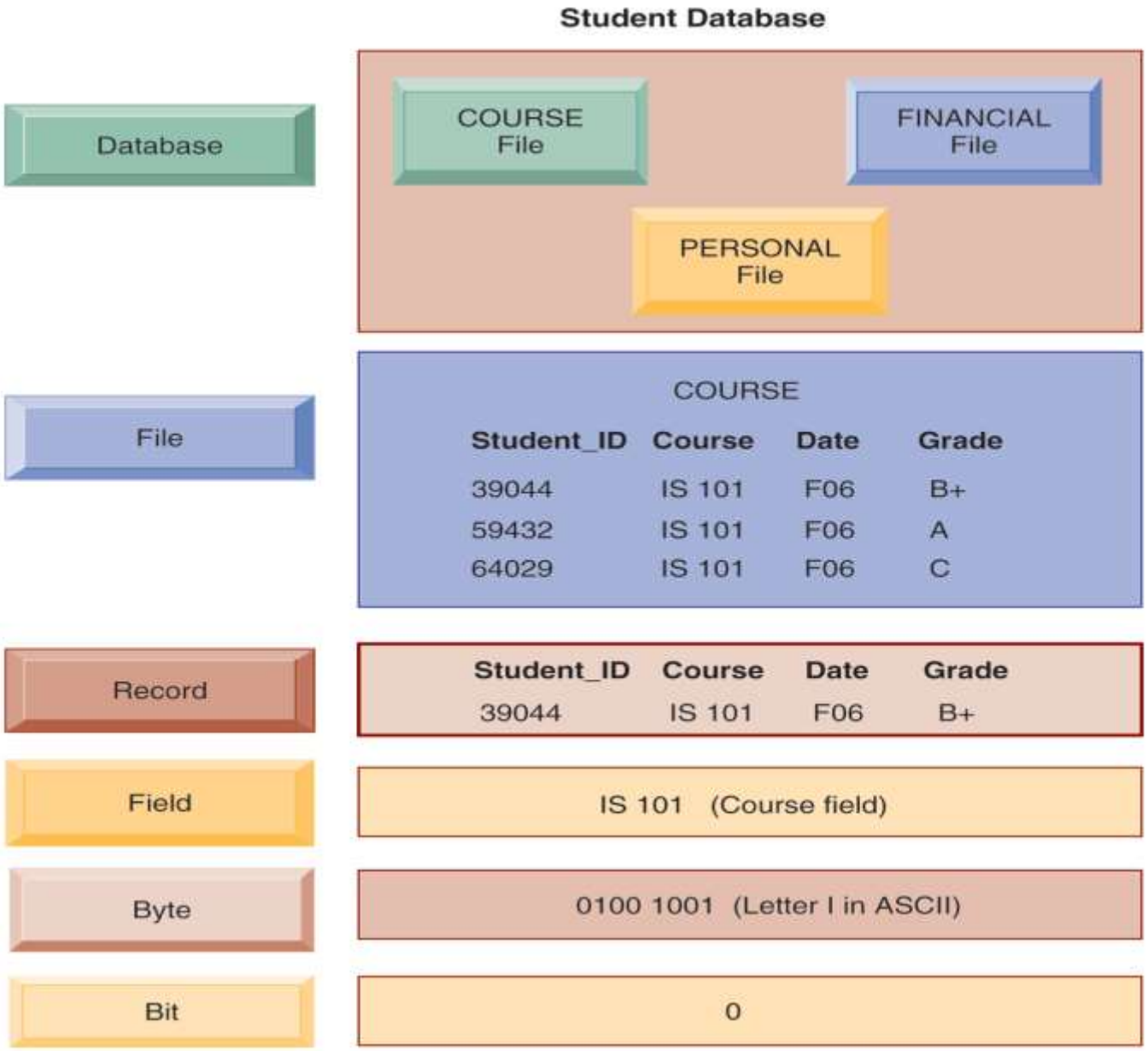
Database Terminology

- **Database**: Group of related files
- **File**: Group of records of same type

- **Record:** The main unit of data in a database. Group of related fields

- **Fields** : Items of data in a record. Group of characters as word(s) or number
  - Describes an **entity** (person, place, thing on which we store information)
  - **Attribute:** Each characteristic, or quality, describing entity
    - E.g., Attributes Date or Grade belong to entity COURSE

- Each record contains fields of the same type


Student Database
Database
COURSE File
FINANCIAL File
PERSONAL File
File
COURSE
Student_ID Course Date Grade
39044 IS 101 F06 B+
59432 IS 101 F06 A
64029 IS 101 F06 C
Record
Student_ID Course Date Grade
39044 IS 101 F06 B+
Field
IS 101 (Course field)
Byte
0100 1001 (Letter I in ASCII)
Bit
0

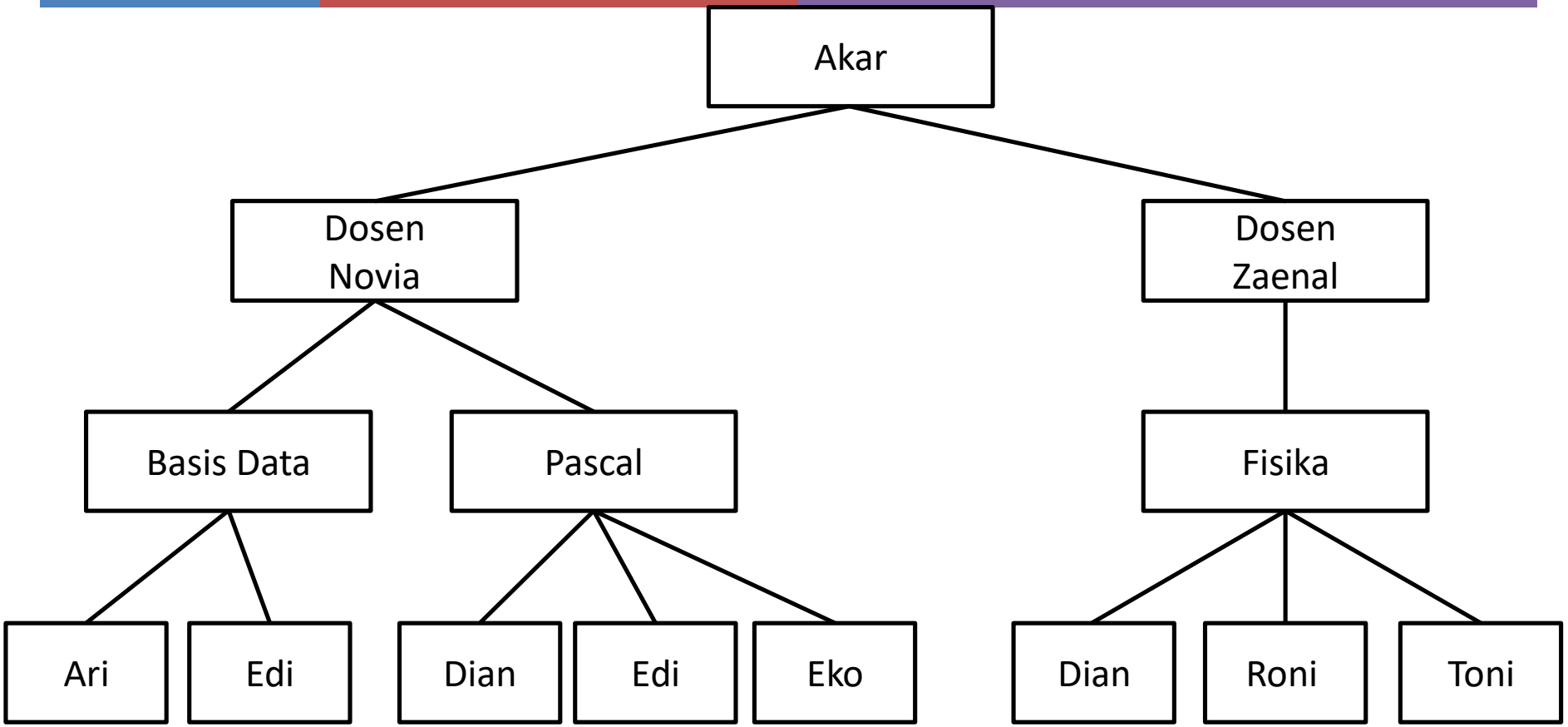
Types of Database

- There are several types of database (and DBMS
- The main difference is their **Data Model**
  - the rules for the organization of data
  - the mechanisms of association between data items
  - the means by which data may be manipulated
- Examples of data models:
  - hierarchical data model
  - network data model
  - relational data model

Hierarchical data Model

- Data is organized into a tree structure
  - records are generally linked to only one parent record
  - not all information naturally fits into hierarchies
  - management requires knowledge of physical location and order of data

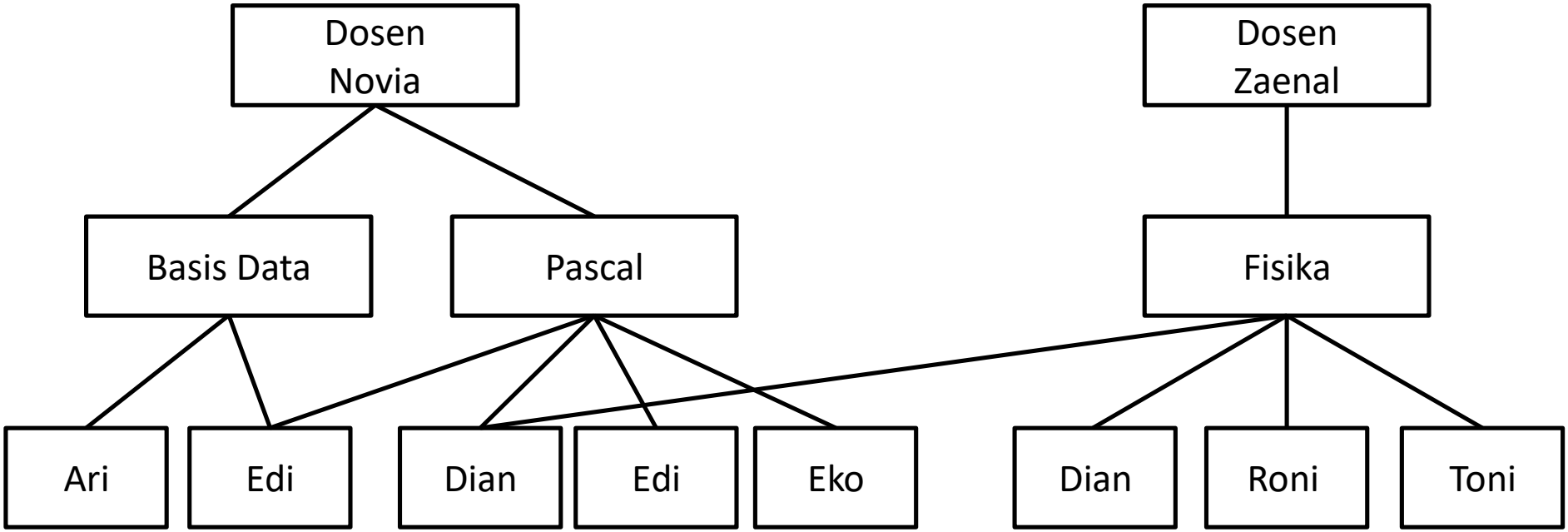
Hierarchical data Model



Network data Model

- Data is organized into a loose network structure
  - records may be linked arbitrarily
  - management still requires knowledge of physical location and order of data

Network data Model



Relational data Model

- Data is organised into sets
  - Sets are usually visualised as a table - for example:
    - the set of product codes
    - the set of data describing product 101
- The foundation of most (but not all) current database systems

Product Code	Product Name	Price
101	Melon	800G
102	Strawberry	150G
103	Apple	120G