Chapter 1
Overview of Database

Content

- Introduction
- The evolution of database systems
- Characteristics of the database approach
- Database users
- Architecture of a DBMS
- Properties of DBMS
- Data models
- Database languages

Introduction

- Examples
  - Banking and finance
    - Customer information, accounts, loans, banking transactions
    - Information of holdings, sales and purchases
  - Education
    - Student information, course registrations and grades
  - Airline
    - Reservations and information of flights and ticket prices
  - Human resources
    - Information about employees, salaries, payroll taxes
  - …

Introduction

- Data
  - Facts that can be recorded and have meaning
  - Pieces of data are individual pieces of information

- Example
  - Name, address, phone number of customers
  - Name, address, salary, tax status of employees
  - Printing of reports such as sale, purchase, bill…
  - Tracking inventories of items in warehouses/stores
  - …

Introduction

- Database (DB)
  - A collection of related data
  - Contains information relevant to an enterprise

- Example: Corporate records
  - Sale, purchase
  - Payable and receivable accounts
  - Employees
  - Printing of employee’s weekly paychecks

Introduction

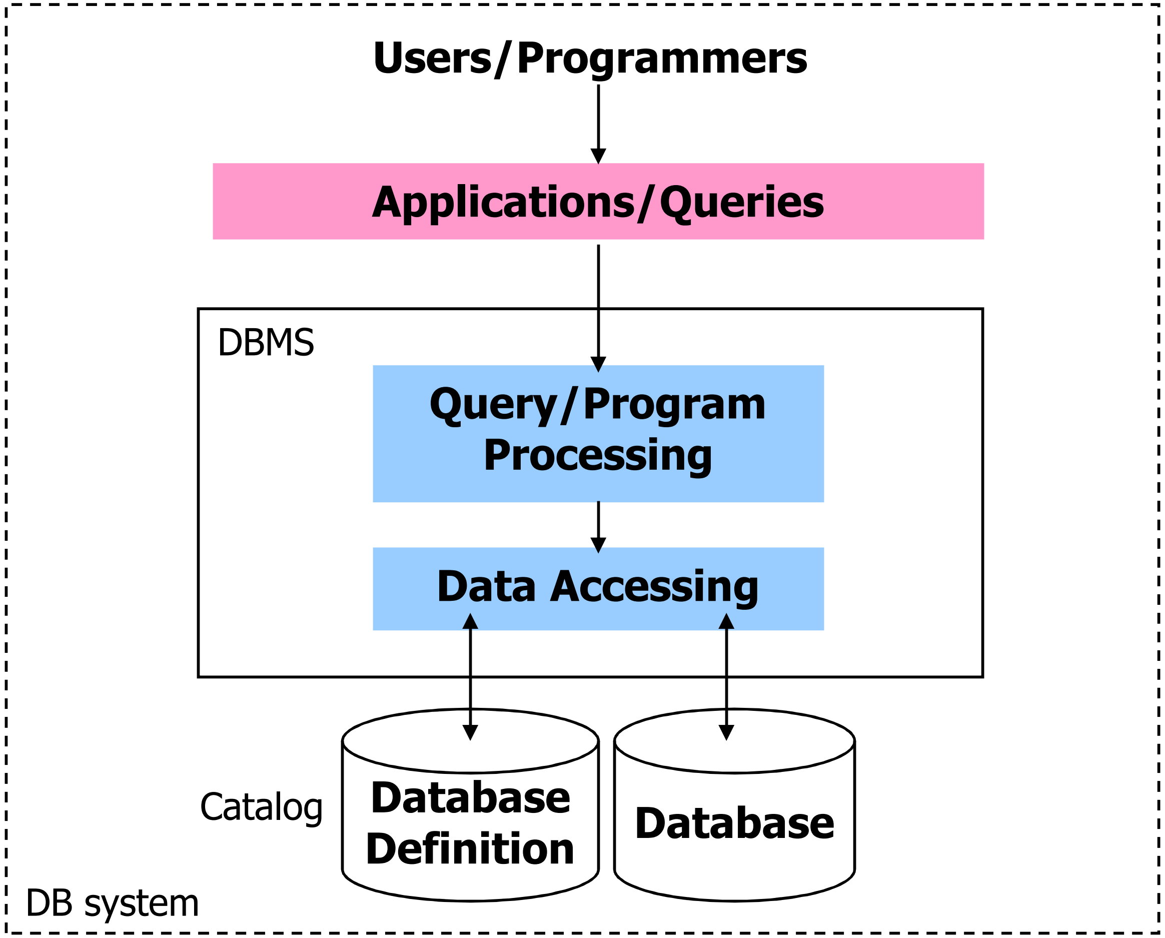
- Database (DB)
  - Represents some aspect of the real world
  - A logically coherent collection of data with some inherent meaning
    - Random assortment of data cannot correctly be a database
  - Is designed, built, and populated with data for a specific purpose, for intended group of users or applications

Introduction

- Database Management System (DBMS)
  - A collection of programs that enables users to create and maintain a database
  - A general-purpose software system that facilitates
    - ***Definition*** – specifying the data types, structures, and constraints for the data
    - ***Construction*** – storing the data itself on some storage medium
    - ***Manipulation*** – querying the database to retrieve data, updating the database to reflect changes, generating reports from the data
    - ***Sharing*** – allowing multiple users/programs to access the database concurrently

Introduction

- Database System



Example

EMPLOYEE	LNAME	MNAME	FNAME	SSN	BIRTHDATE	SUPERSSN	DNO
	Tran	Hong	Quang	987987987	03/09/1969	987654321	4
	Nguyen	Thanh	Tung	333445555	12/08/1955	888665555	5
	Nguyen	Manh	Hung	666884444	09/15/1962	333445555	5
	Tran	Thanh	Tam	453453453	07/31/1972	333445555	5

PROJECT	PNAME	PNUMBER	PLOCATION	DNUM
	San pham X	1	VUNG TAU	5
	San pham Y	2	NHA TRANG	5
	San pham Z	3	TP HCM	5
	Tin hoc hoa	10	HA NOI	4

WORKS_ON	SSN	PNO	HOURS
	123456789	1	32.5
	123456789	2	7.5
	666884444	3	40.0
	453453453	1	20.0

Example

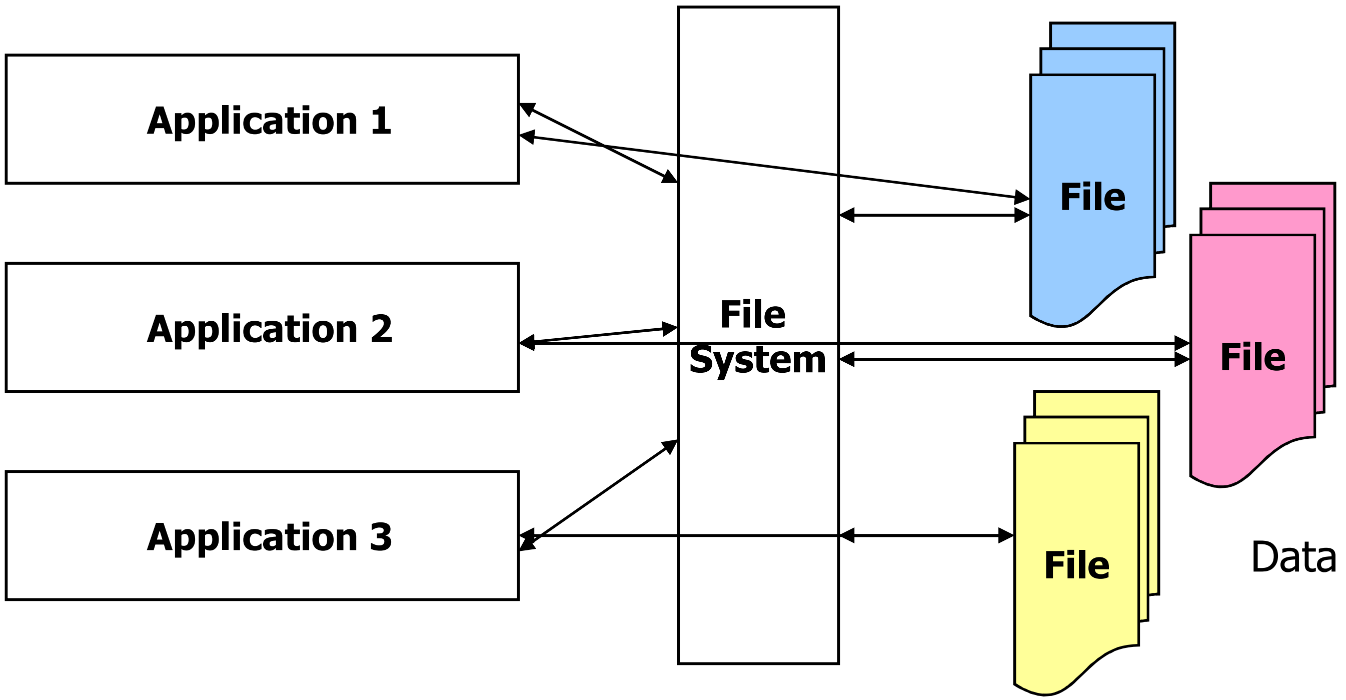
- Company database - project management
  - Definition
    - Specify the structure of records, including data elements, data types
  - Construction
    - Store data to represent an employee, project, department… as a record
  - Manipulation
    - Querying: “Select the employees whose department is 5”
    - Updating: “Move the employee Nguyen Thanh Tung to department 1”

Content

- Introduction
- **The evolution of database systems**
- Characteristics of the database approach
- Database users
- Architecture of a DBMS
- Properties of DBMS
- Data models
- Database languages

Evolution

- File



An application program has its own data

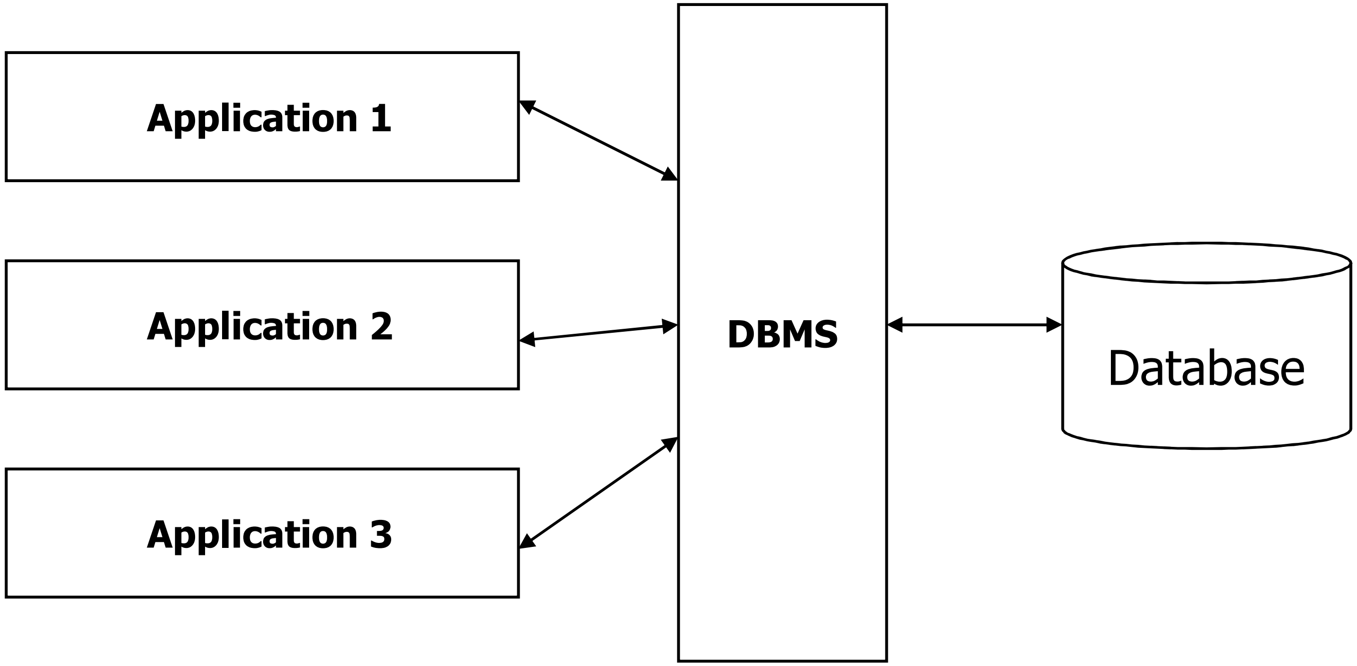
Evolution

- Limitations
  - Data redundancy
    - Wasted storage space
    - Opportunities of the inconsistency
  - Data sharing is limited
  - Difficult recovery
  - Low security

- But, still be used in some applications
  - Small size DB
    - Storing and accessing data only, not including other processing operations
  - Fee costs less
    - Operation or maintenance

Evolution

- Database



Content

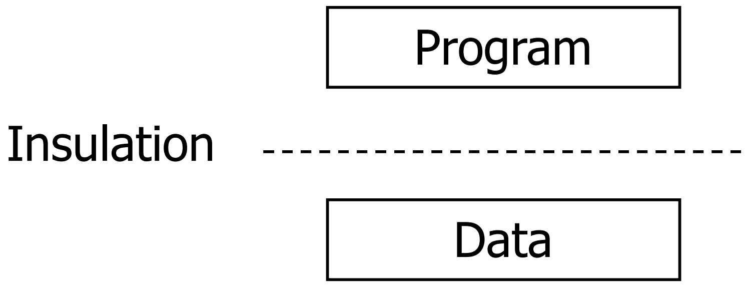
- Introduction
- The evolution of database systems
- **Characteristics of the database approach**
  - Self-describing
  - Insulation between programs and data
  - Data abstraction
  - Views of data
  - Sharing of data
- Database users
- Architecture of a DBMS
- Properties of DBMS
- Data models
- Database languages

Self-Describing

- The DB system contains not only the DB itself, but also a complete definition/description of the DB structure
- The definitions are stored in ***catalog***
  - Contains information such as the structure of data, type and storage format of data items, and constraints on the data
- Information stored in catalog is called ***meta-data*** (data of data)
- Many applications can access to the DB
  - Refer to catalog, knowing the structure of files in specific DB (type and format of data)

Insulation

- The structure of data is stored in *catalog* separately from the access programs
  - Program-Data independence



- A little change in the structure happens
  - Application programs are rarely revised

Data abstraction

- The DB system provides a ***conceptual representation*** of the data to hide certain details of how the data are stored and maintained

- Example
  - *Data model* is a type of data abstraction
    - Objects
    - Properties
    - Relationships
  - These logical concepts are easier for user to understand than computer storage concepts

Views of data

- A DB has many users
- Each user may require a different ***perspective or view*** of the database

- A view may be
  - A subset of the database
  - Aggregate data that are derived from the database

Sharing of data

- A multiuser DBMS
  - Allows users to access the DB at the same time
  - Data for many applications are to be integrated and maintained in *a single DB*

- Using concurrency control mechanisms to access the data reasonably
  - Avoid data contention
  - Ensure the data will always be valid when they are accessed

Content

- Introduction
- The evolution of database systems
- Characteristics of the database approach
- **Database users**
  - Database administrator (DBA)
  - Database designer
  - End user
- Architecture of a DBMS
- Properties of DBMS
- Data models
- Database languages

Database administrator

- Many people use the same resources
  - Need a chief administrator to oversee and manage

- Responsibility
  - Administering the DB
  - Authorizing access to the DB
  - Coordinating and monitoring the use of DB
  - Acquiring software and hardware resources as needed

Database designer

- Responsibility
  - Identifying the data to be stored in the DB
  - Choosing appropriate structures to represent and store the DB
  - Communicating with all DB users to understand their requirements, to come up with a design that meet the requirements
- Can be
  - Staff of the DBA
  - Other staffs taking responsibilities after the DB designed is completed

End user

- People whose jobs require to access to the DB
  - Querying, updating, generating reports

- Categories
  - Casual end user
  - Naïve or parametric end user
  - Sophisticated end user

End user

- People whose jobs require to access to the DB
  - Querying, updating, generating reports

- Categories
  - Casual end user
    - Occasionally access the DB
    - Need different information each time
    - Use sophisticated DB query language to specify requests
    - Middle or high level manager
  - Naïve or parametric end user
  - Sophisticated end user

End user

- People whose jobs require to access to the DB
  - Querying, updating, generating reports

- Categories
  - Casual end user
  - Naïve or parametric end user
    - Constantly query and update the DB
    - Use standard types of queries and updates that have been programmed and tested
    - Employee
  - Sophisticated end user

End user

- People whose jobs require to access to the DB
  - Querying, updating, generating reports

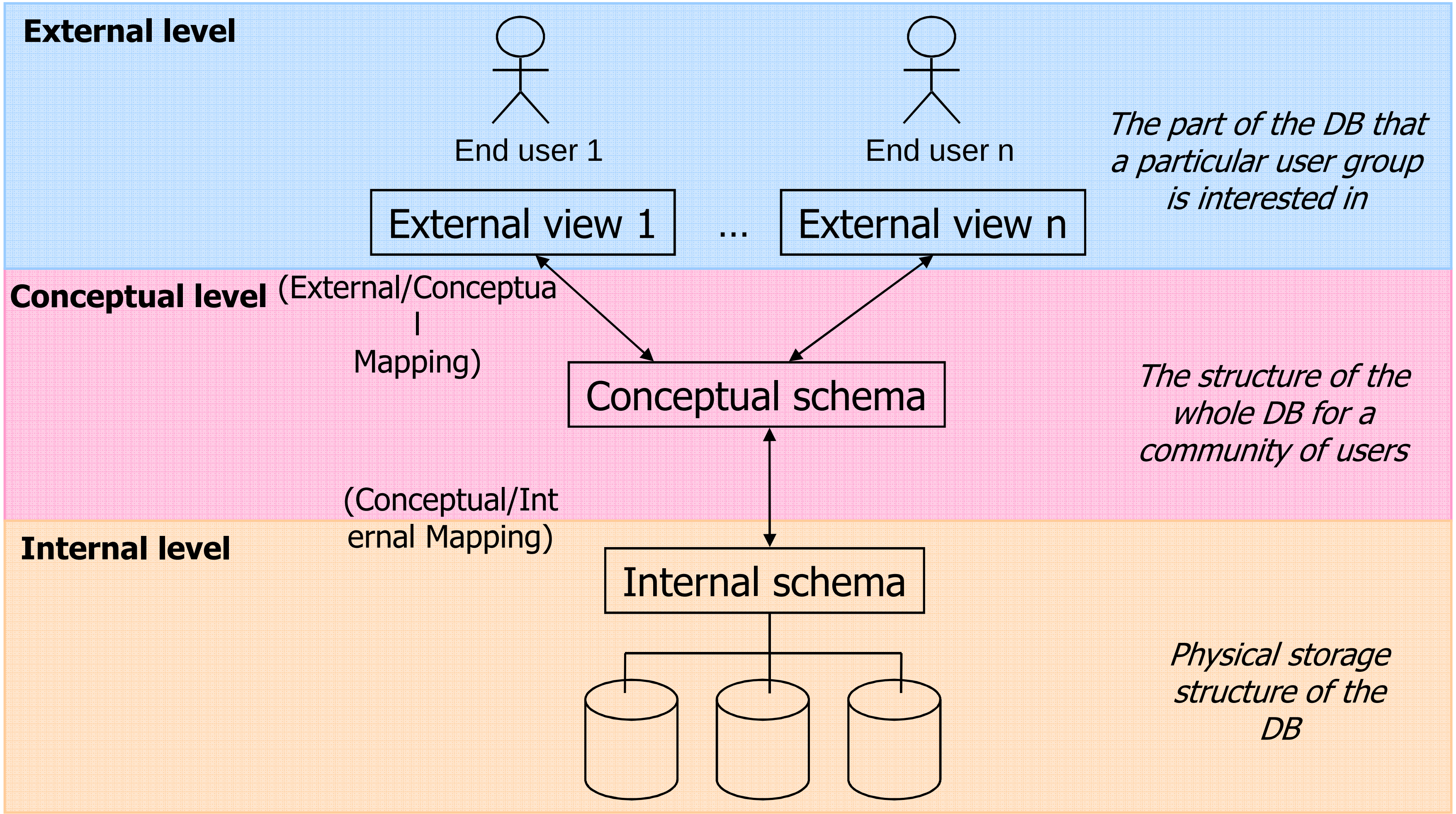
- Categories
  - Casual end user
  - Naïve or parametric end user
  - Sophisticated end user
    - Be familiar with the facilities of the DBMS
    - Implement the applications to meet the complex requirements
    - Engineers, scientists, business analysts

Content

- Introduction
- The evolution of database systems
- Characteristics of the database approach
- Database users
- **Architecture of a DBMS**
- Properties of DBMS
- Data models
- Database languages

Architecture

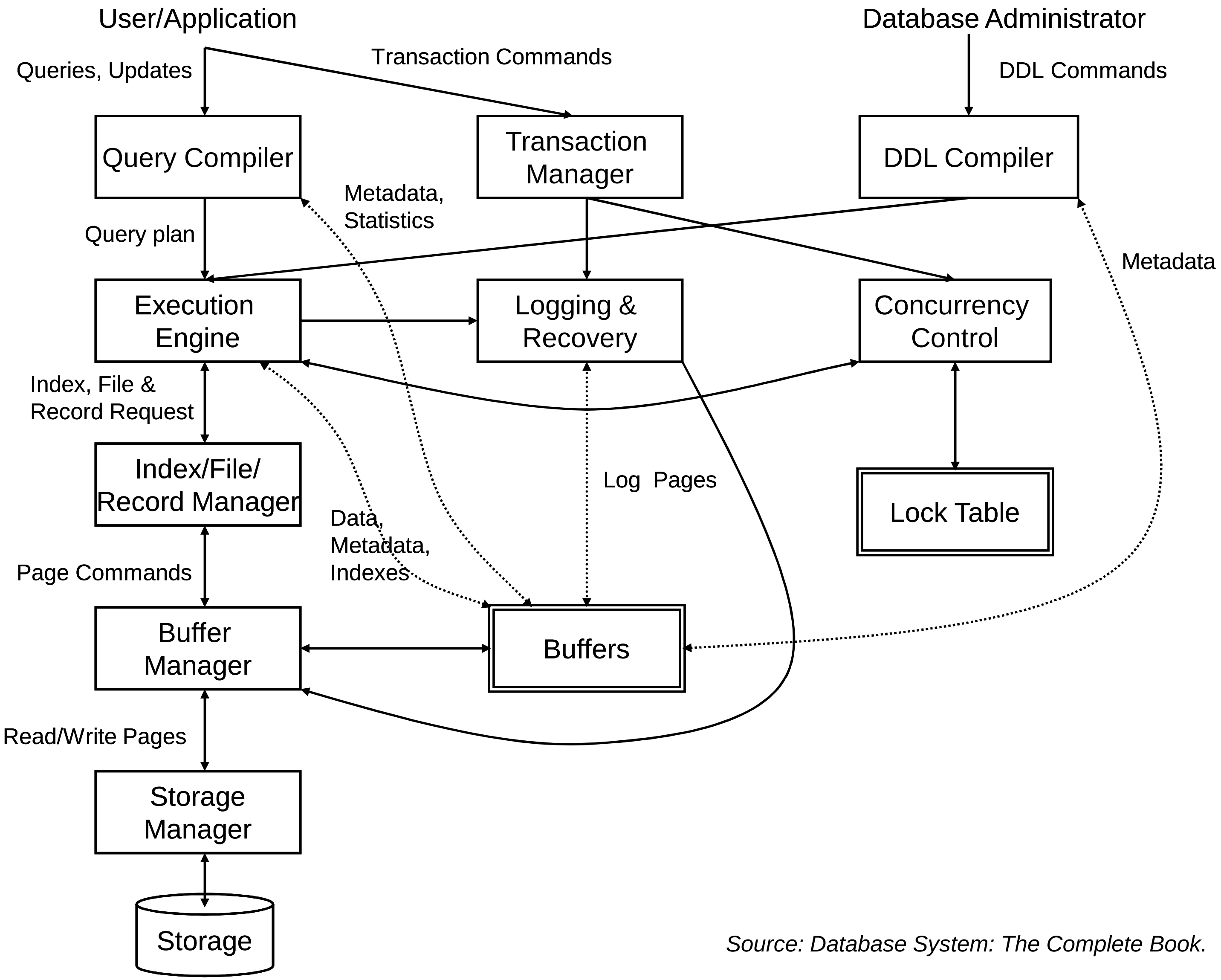
- Three-schema architecture



Architecture

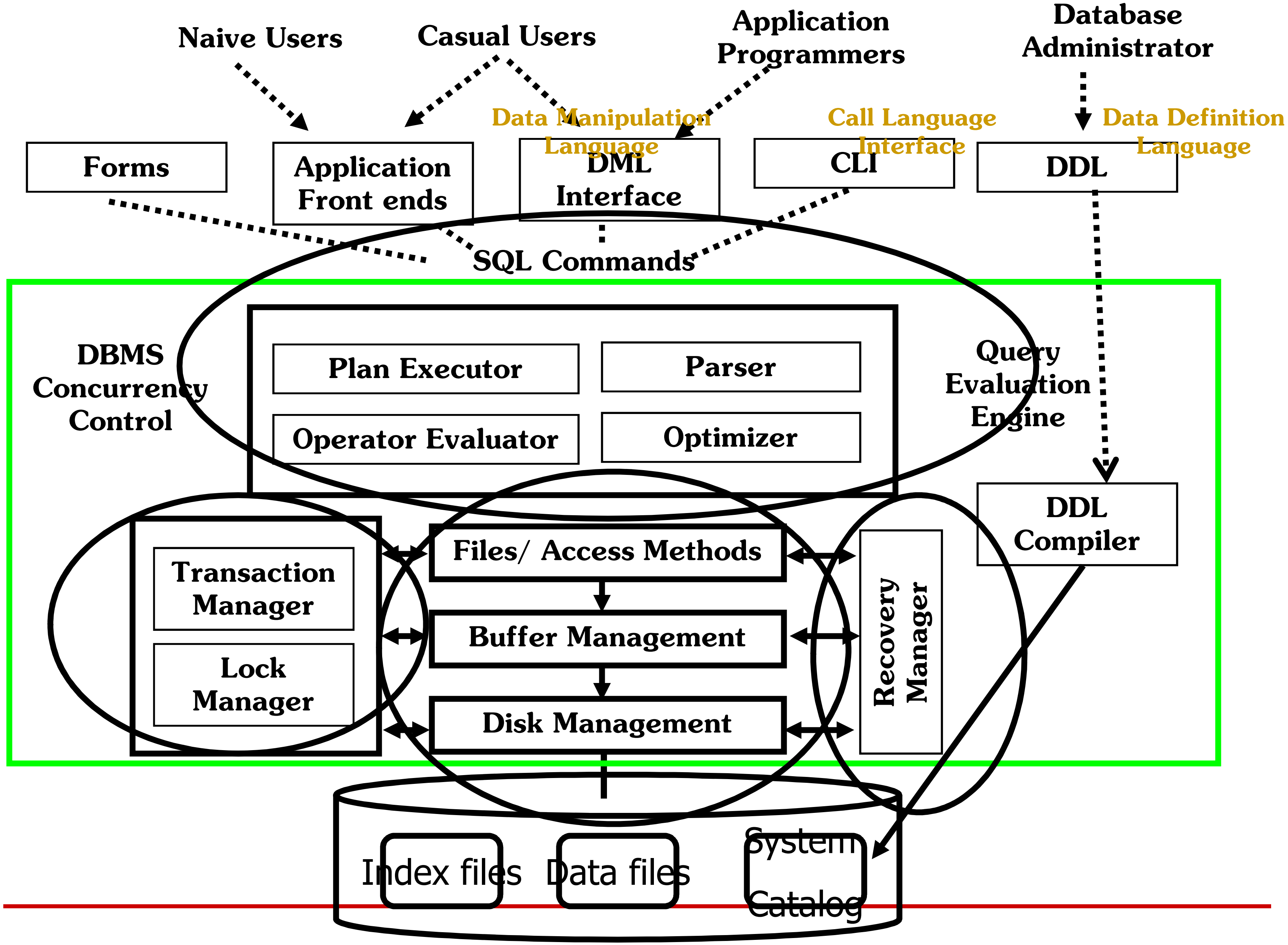
- Data independence
  - Logic data independence
    - The capability to change the conceptual schema without any change external schemas or application programs
    - Example
      - Adding/removing a record type or data item (expand/reduce DB)
      - Changing constrains
  - Physical data independence
    - The capability to change the internal schema without having any change the conceptual schema
    - Example
      - Physical files had to be reorganized to improve the performance of retrieval or update

Architecture



Source: Database System: The Complete Book.

Architecture of a DBMS



Content

- Introduction
- The evolution of database systems
- Characteristics of the database approach
- Database users
- Architecture of a DBMS
- **Properties of DBMS**
- Data models
- Database languages

Properties of DBMS

- Controlling redundancy
  - By placing all the data together, we do not have to search multiple files to collect this data
- Data sharing
  - In multiple user environment, concurrency data access is allowed
- Restricting unauthorized access
  - Users or user groups are given account numbers protected by passwords to gain access to the DB
- Providing multiple user interfaces
  - Provide query languages for casual users, programming language interfaces for programmers, forms and command codes for parametric users

Properties of DBMS

- Enforcing integrity constraints
  - Integrity constraints
    - Rules/conditions are derived from the meaning/semantics of the data or the miniworld it represents
  - Some constraints
    - Can be specified to the DBMS and automatically enforced
    - May have to be checked by update programs

- Providing backup and recovery
  - Provide facilities for recovering from hardware and software failures
  - Make sure the DB is restored to the state it was before

Properties of DBMS

- Others
  - Potential for enforcing standards
    - Permit DBA to define and enforce standards among database users in a large organization
  - Flexibility
    - It may be necessary to change the structure of a DB as requirements change without affecting the stored data and the existing application programs
  - Reduced application development time
  - Availability of up-to-date information
    - As soon as one user's update is applied to the DB, all other users can immediately see this update

Content

- Introduction
- The evolution of database systems
- Characteristics of the database approach
- Database users
- Architecture of a DBMS
- Properties of DBMS
- **Data models**
- Database languages

Data models

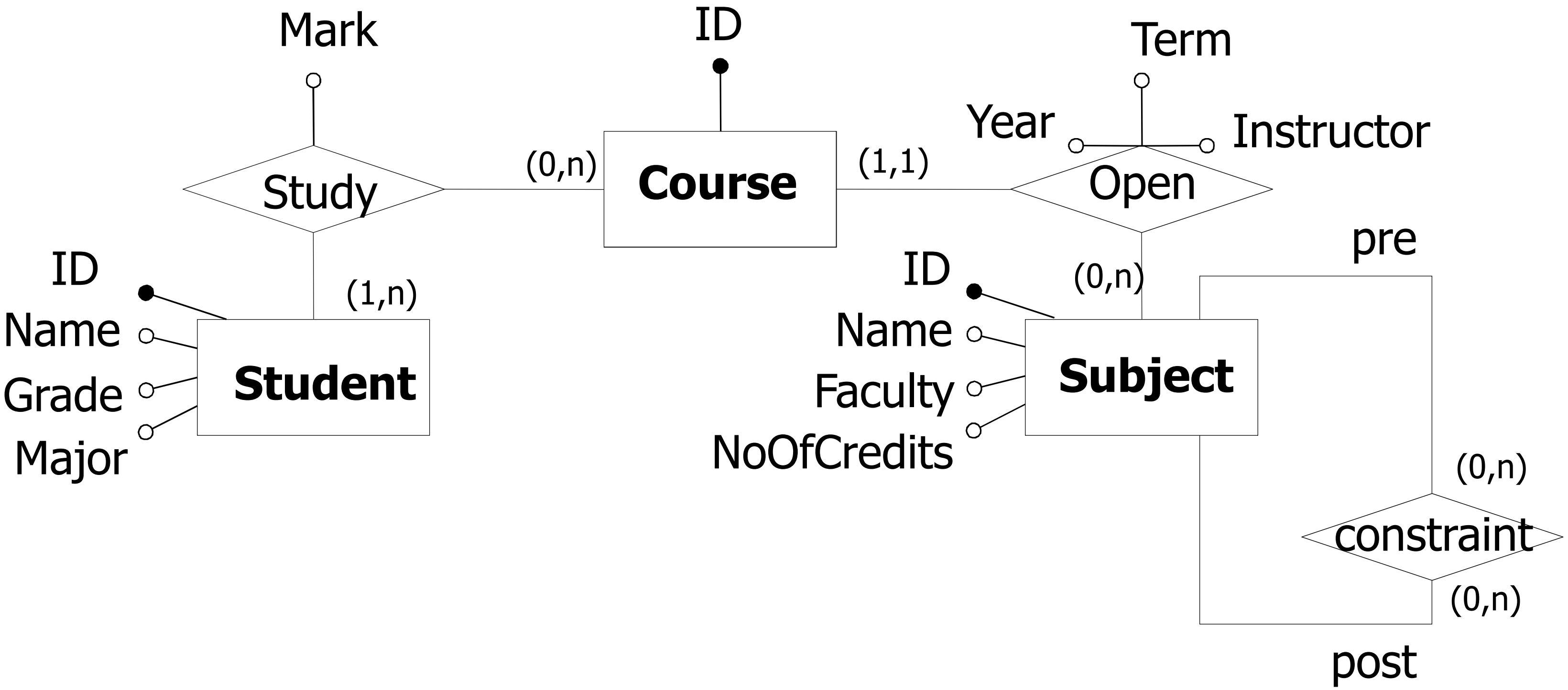
- Definition
  - A collection of ***concepts*** that can be used to describe the ***structure of a DB***
    - Data types, relationships, and constraints
  - Including a set of basic ***operations*** for specifying retrievals and updates on the DB

- Categories
  - High level or conceptual data models
  - Representational or implementation data models
  - Low level or physical data models

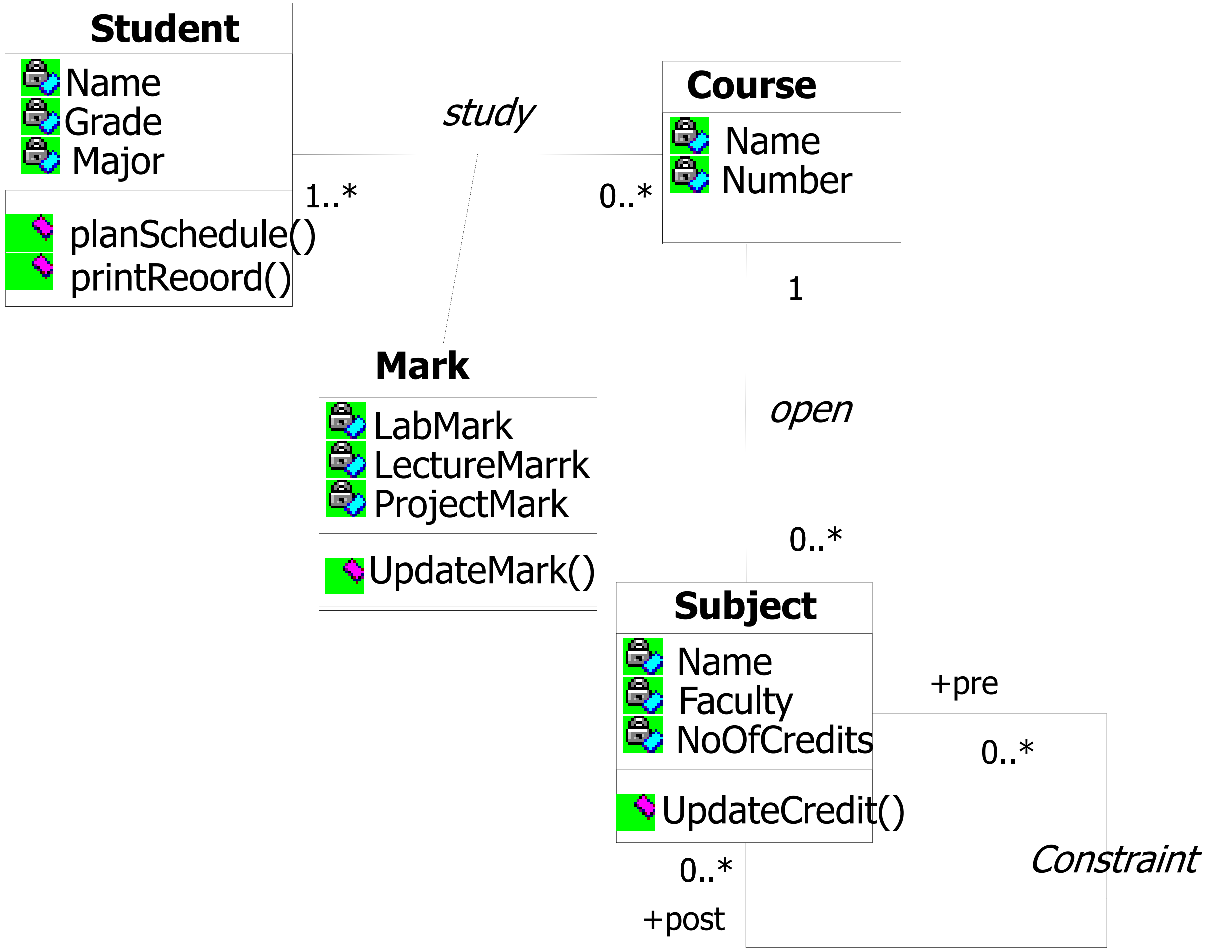
Data models

- High level data model
  - Provide concepts that are close to the way users perceive data
  - Eg: entity relationship model, object-oriented model…
- Implementation data model
  - Provide concepts that may be understood by end users, but that are not too far from the way data is organized within the computer
  - Eg: relational model, network and hierarchical models…
- Low level data model
  - Provide concepts the describe the details of how data is stored in the computer

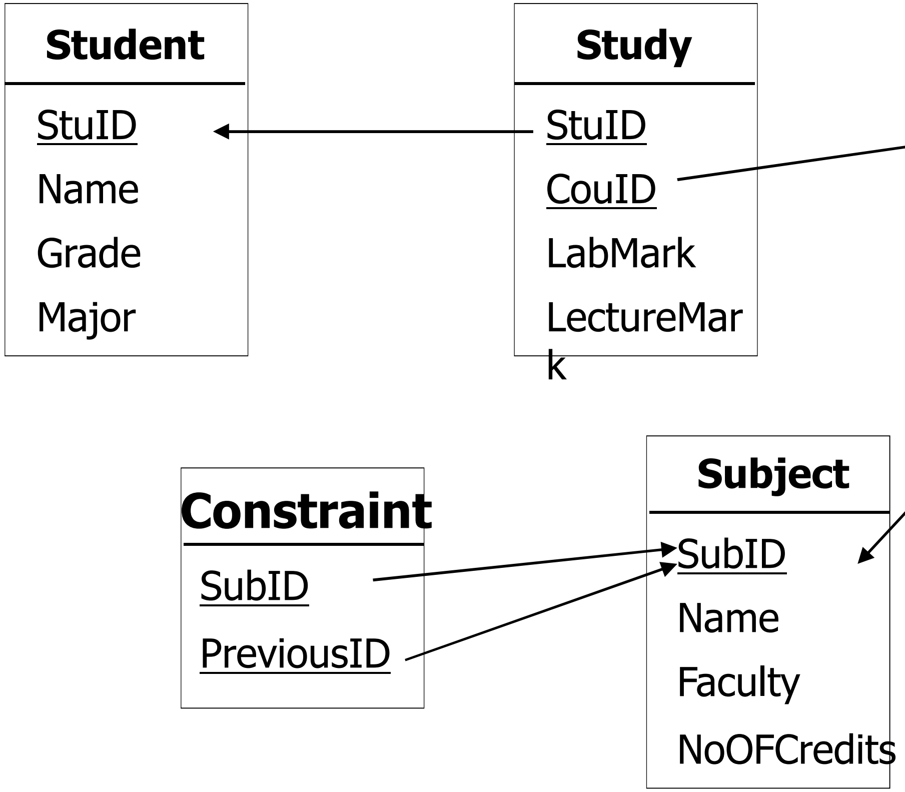
Example of ER Model



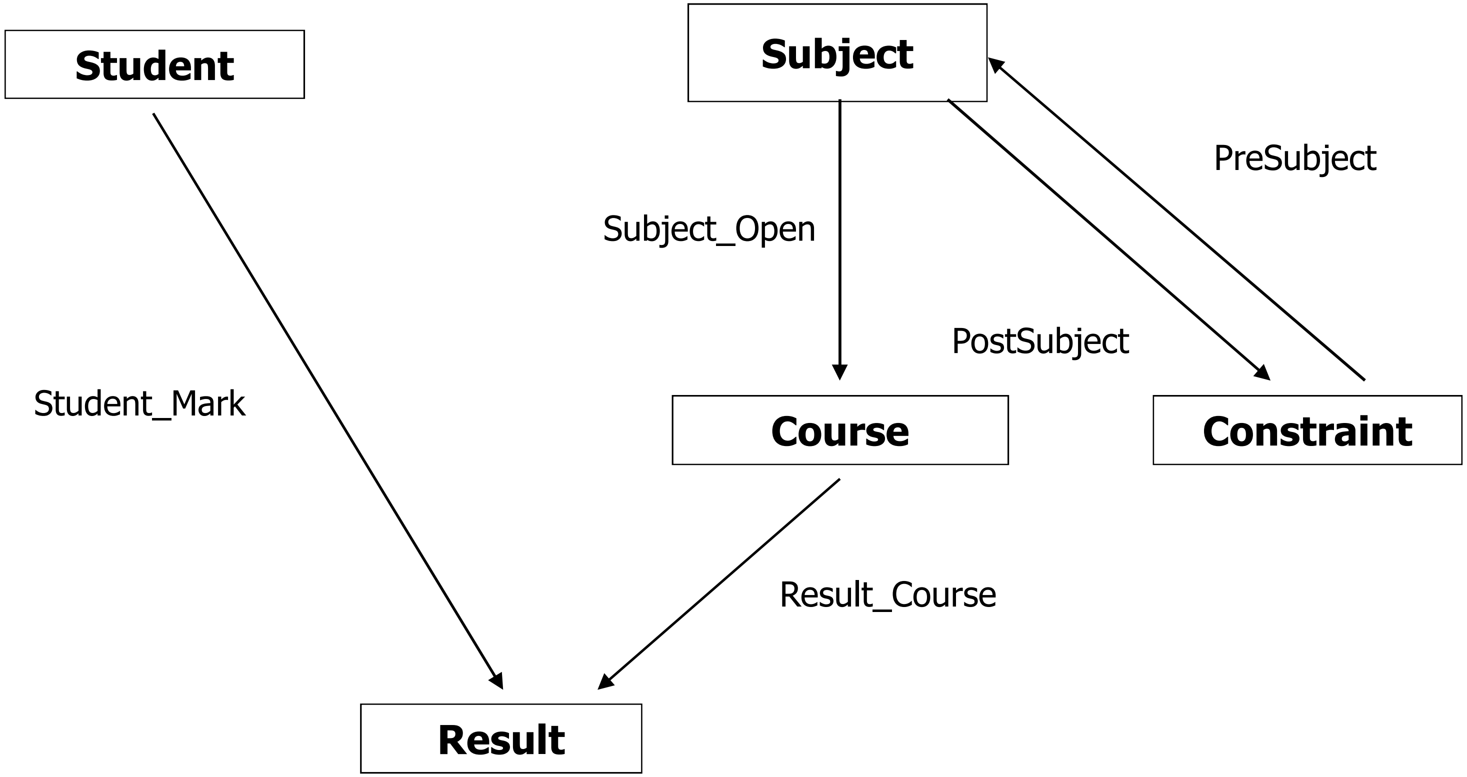
Example of Object-Oriented Model



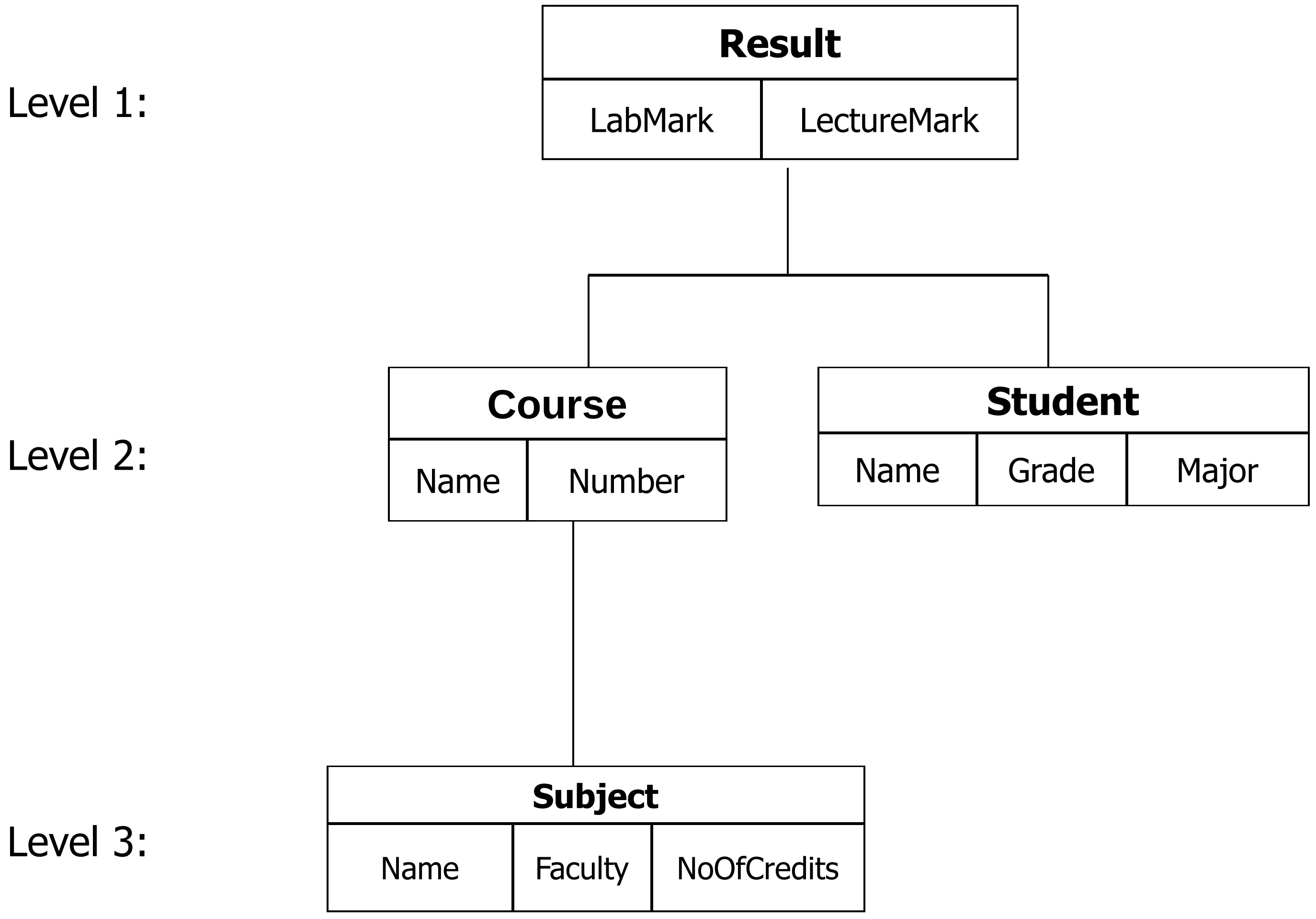
Example of relational model



Example of network data model



Example of hierarchical data model



Content

- Introduction
- The evolution of database systems
- Characteristics of the database approach
- Database users
- Architecture of a DBMS
- Properties of DBMS
- Data models
- **Database languages**

Database language

- DDL – Data Definition Language
  - Identify descriptions of the schema constructs
  - Store the schema description in the DBMS catalog
- SDL – Storage Definition Language
  - Specify the internal schema and the mappings between two schemas
- VDL – View Definition Language
  - Specify user views and their mapping to the conceptual schema

Database language

- DML – Data Manipulation Language
  - Provide a set of operations including retrieval, insertion, deletion and modification of the data
  - Two types
    - High level (nonprocedural)
      - Entered interactively from a display monitor/terminal
      - Embedded in a general-purpose programming language
    - Low level (procedural)
      - Must be embedded in a general-purpose programming language

Discussion

- When will we use or not use the DB approach?