Lecture1: Ch1. Introduction: Databases and Database Users

Dr. Alaa eldin Abdallah Yassin

outline

- Introduction and Basic Definitions.
- Types of Databases and Database Applications.
- **DBMS** and its functions.
- Example of a Database (UNIVERSITY).
- Database system environment.
- Main Characteristics of the Database Approach.
- Advantages of Using the **DBMS** Approach.
- When Not to Use DBMS.

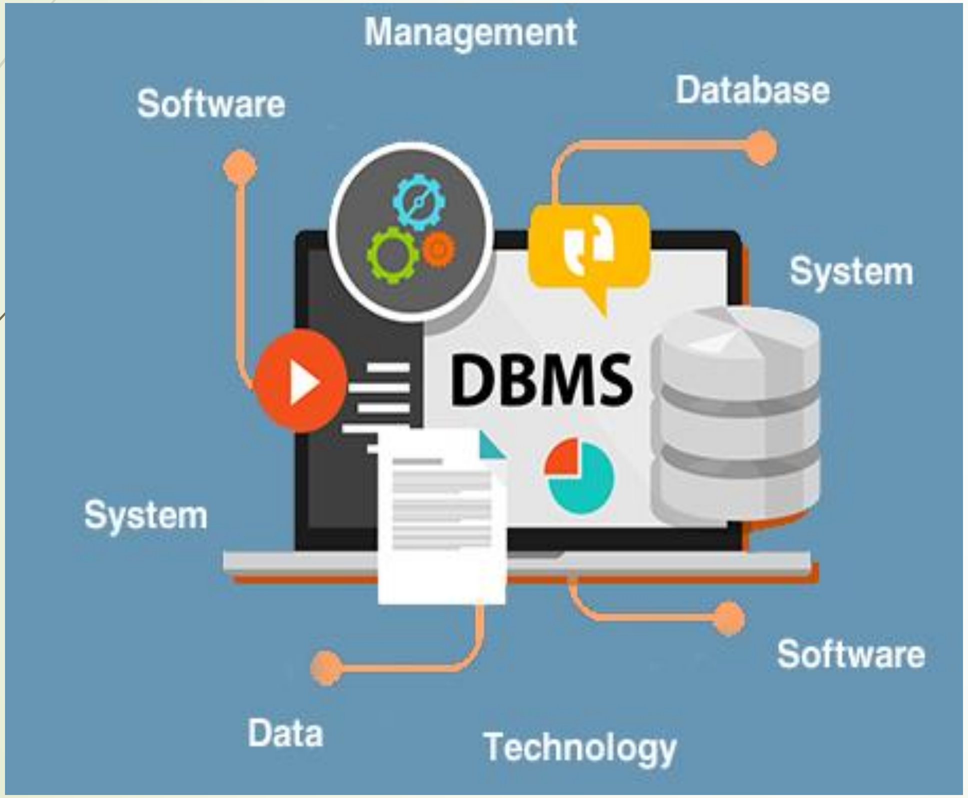
Introduction & Basic Definitions

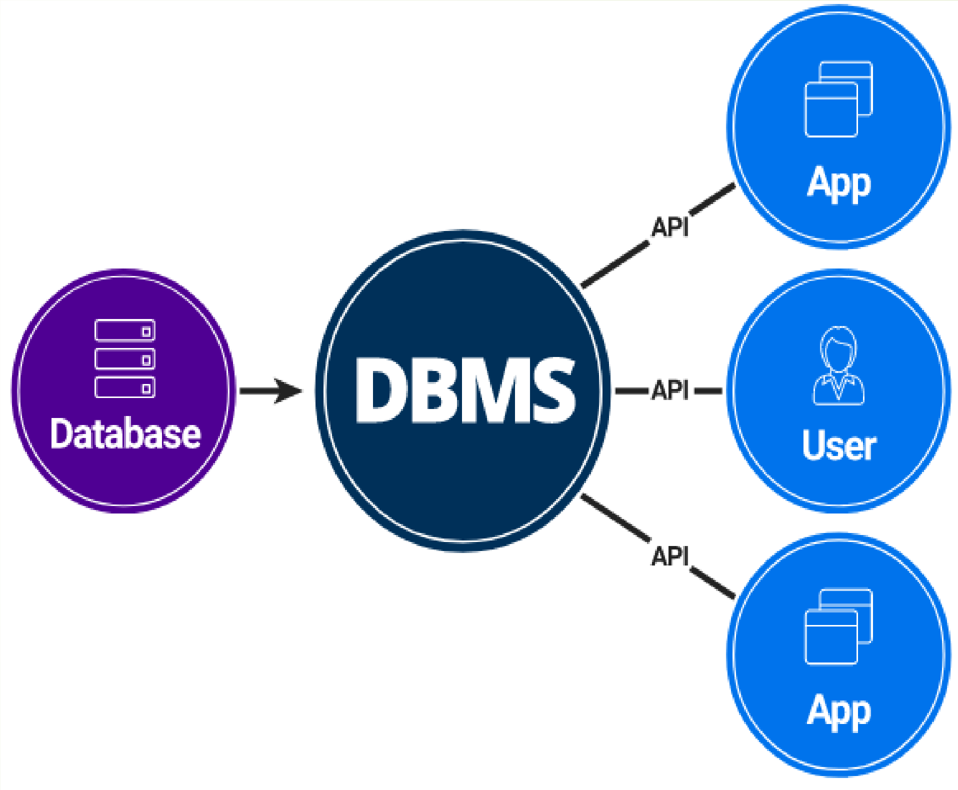
- **Data**:
  - ***“ Known facts that can be recorded and have an implicit meaning”.***
    - Like names, telephone numbers and addresses.
- **Database**:
  - Database is a ***“collection of related data ”***. ☺
  - **Database has the following implicit properties:**
    1. A database represents some aspect of the real world, sometimes called the “*mini world*”.
    2. A database is a logically coherent collection of data with some inherent meaning. (related data not a random data)
    3. A database is **designed**, **built**, and **populated** with data for specific purpose.
  - Examples: Airline reservation system, Students’ registration system.

Types of Databases and Database Applications

- **Traditional Applications**:
  - Numeric and Textual Databases
- **More Recent Applications**:
  - Multimedia Databases
    - store images, audio, video…
  - Geographic Information Systems (**GIS**)
    - store and analyze maps, weather data, and satellite images
  - Data Warehouses and online analytical processing (**OLAP**)
    - Extract and analyze information from different and very large databases; to support decision making.

DBMS


Management
Software
Database
System
DBMS
System
Software
Data
Technology


Database
DBMS
API
API
API
App
User
App

Basic Definitions…

Database Management System (DBMS):

- **DBMS**: "*Is a collection of programs that enables users to* ***create*** *and* ***maintain*** *a database*".
- **DBMS** : " *is a general-purpose software system that facilitates the processes of* ***defining***, ***constructing***, ***manipulating*** *and* ***sharing*** *databases among various users and applications*".

Defining Database:

- **Defining** a database involves :
  - Specifying the **data types**, **structures**, and **constrains** of the data to be stored in the database.
  - The database definition or descriptive information is stored by DBMS in database **catalog** or dictionary; it is called "***meta-data***".

DBMS functions…

- **Constructing Database:**
  - **Constructing** a database is the process of **storing the data** on some storage medium (hard drive, cloud-based,…) that is controlled by the DBMS.
- **Manipulating Database:**
  - **Manipulating** a database include **functions** such as **querying** the database to retrieve specific data, **updating** the database to reflect changes in the mini-world, and **generating reports** from the data.
- **Sharing Database:**
  - **Sharing** a database allows multiple users and programs to access the database **simultaneously** (train tickets, …).
- **Protecting Database:**
  - Protection includes system protection against hardware and software malfunction or crashes, and security **protection against unauthorized access(***privilege***).**

Typical **DBMS** Functionality

- **Define** a particular database in terms of its data types, structures, and constraints.
- Construct or **Load the initial database contents** on a secondary storage medium.
- **Manipulating** the database:
  - Retrieval: Querying, generating reports
  - Modification: Insertions, deletions and updates to its content
  - Accessing: the database through Web applications
- **Processing** and **Sharing** by a set of concurrent users and application programs – yet, keeping all data valid and consistent

Typical **DBMS** Functionality…

Other features:

- Protection or **Security** measures to prevent unauthorized access.
- Presentation and **Visualization** of data.
- **Maintaining** the database and associated programs over the lifetime of the database application.

Summarize the previous slides

Summary

- **D**ata **B**ase **M**anagement **S**ystem (**DBMS**) :
  - Collection of programs.
  - Enable users to create and maintain database.
- **Defining** a database :
  - Specify the data type, structure and constraints of the data to be stored.
- **Meta-data:**
  - Database definition or descriptive information.
  - Stored by the DBMS in the form of database catalog or dictionary.
- **Manipulating a database:**
  - Query and update the database.
  - Generate reports.

Summary…

- **Sharing** a database:
  - Allow multiple users and programs to access the database simultaneously.
- **Application program**:
  - Access database by sending queries to DBMS.
- **Query**:
  - Causes some data to be retrieved.
- **Transaction**:
  - May cause some data to be read and some data to be written into the DB.
- **Maintain** the database system:
  - Allow the system to improve as requirements change over time.

Real example “ **university db**. “

A real example “**University** data base”

- **University** database :
  - Information concerning students, courses, grades in the university environment.
- **Data records** (I want to store data about what ?):
  - STUDENTs
  - COURSEs
  - SECTIONs (of COURSEs)
  - (academic) DEPARTMENTs
  - INSTRUCTORs
  - PREREQUSITE
  - GRADES

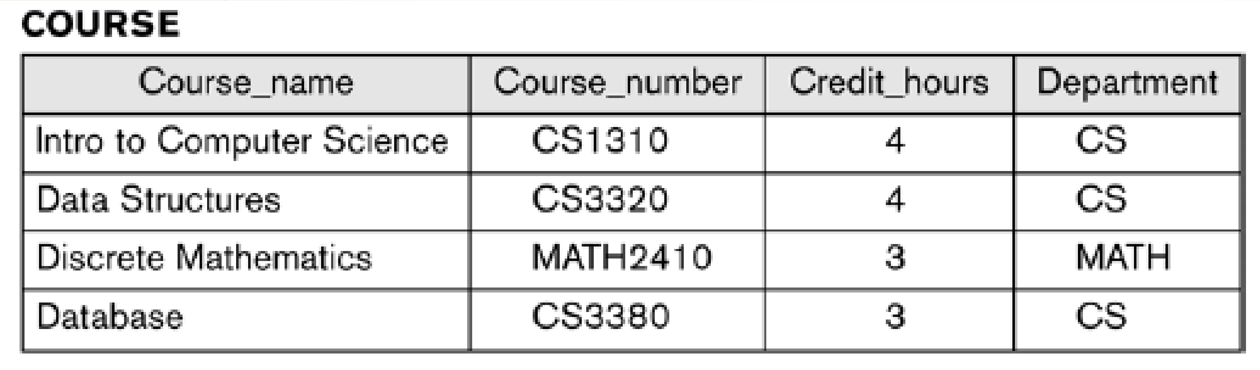
"**University** data base"…

- Specify **structure** of records of each file by specifying **data type** for each **data element**.
  - Integer
  - String
  - Etc.
- **Construct** "***university***" database:
  - **Store data** to represent each student, course, section, and prerequisite as a record in appropriate file.
  - **Relationship** among the records.
  - **Manipulating** involves querying and updating.
    - Query example : list the names of students who register "database" course.
    - Another query ex: list the prerequisite of the "database" course.

Example of a Database (with a Conceptual Data Model)…

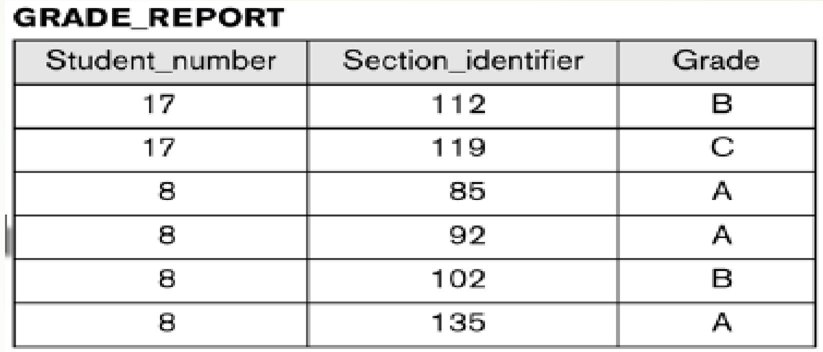
- Some mini-world (university database)**relationships**:
  - **SECTIONs** are of **specific COURSEs**
  - **STUDENTs take SECTIONs**
  - **COURSEs have prerequisite COURSEs**
  - **INSTRUCTORs teach SECTIONs**
  - **COURSEs** are **offered by DEPARTMENTs**
  - **STUDENTs** major **in DEPARTMENTs**
- **Note**: The above entities and relationships are typically expressed in a conceptual data model, such as the **ENTITY-RELATIONSHIP** data model (Chapters 3, 4)

University database…



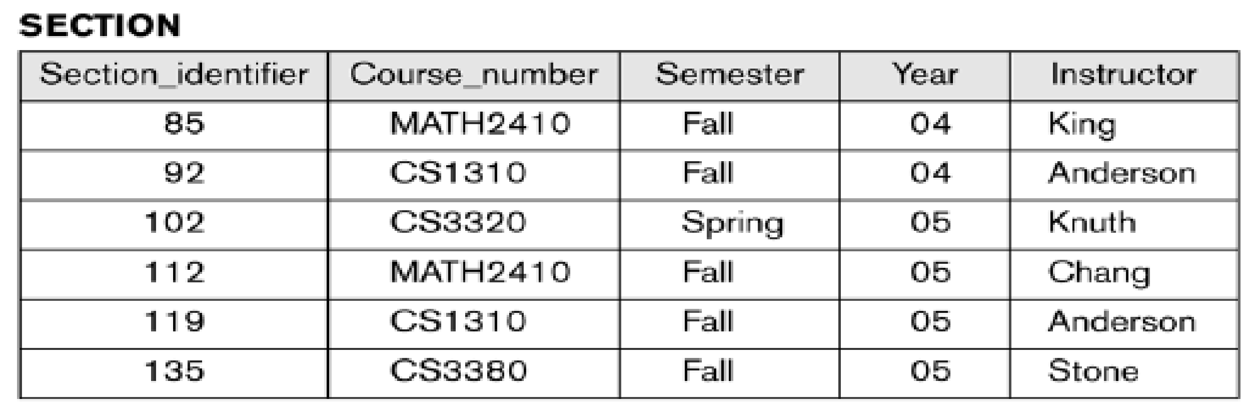
COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS



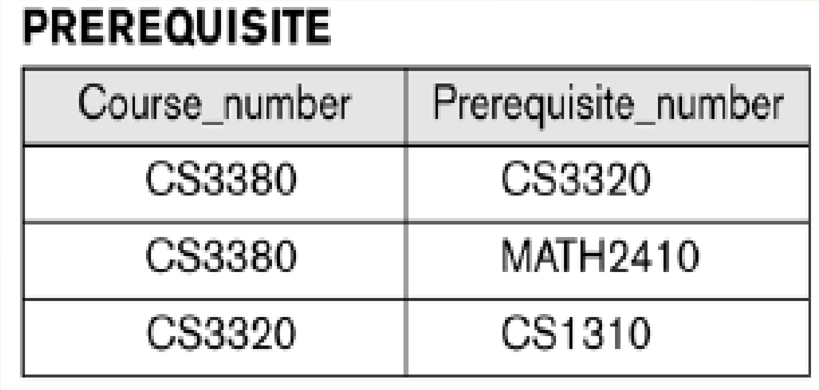
GRADE_REPORT

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A



SECTION

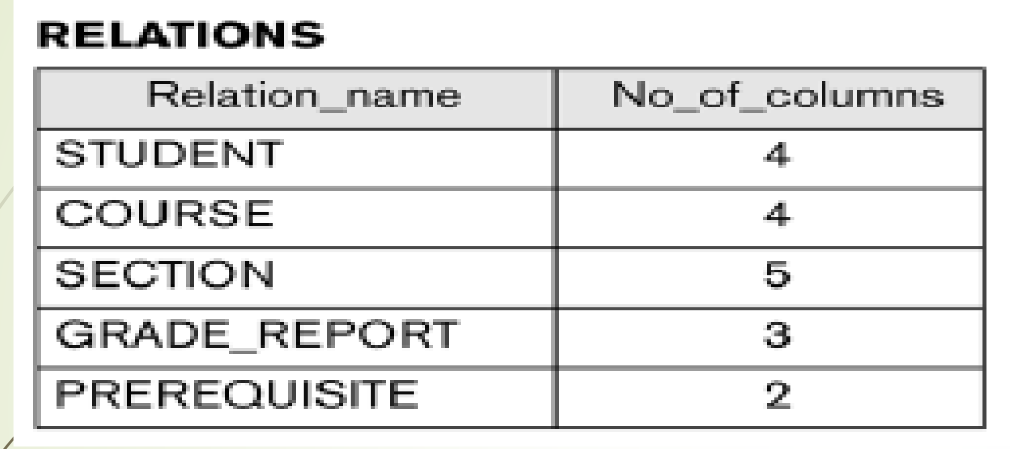
Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	04	King
92	CS1310	Fall	04	Anderson
102	CS3320	Spring	05	Knuth
112	MATH2410	Fall	05	Chang
119	CS1310	Fall	05	Anderson
135	CS3380	Fall	05	Stone



PREREQUISITE

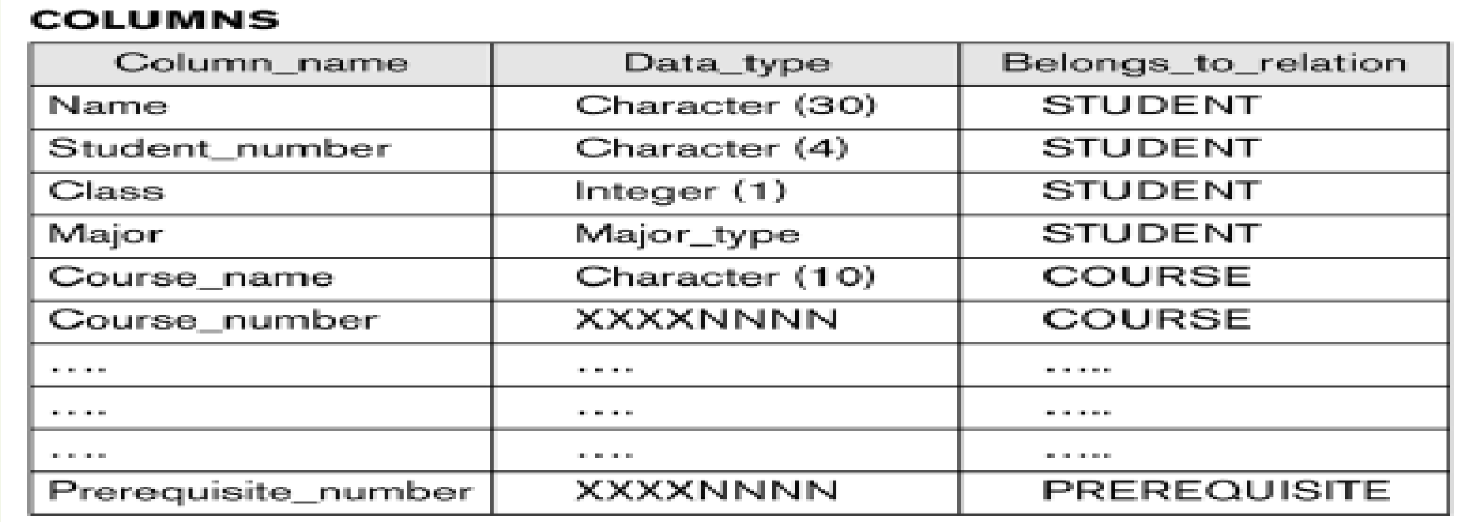
Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

Example of a simplified database **catalog**



RELATIONS

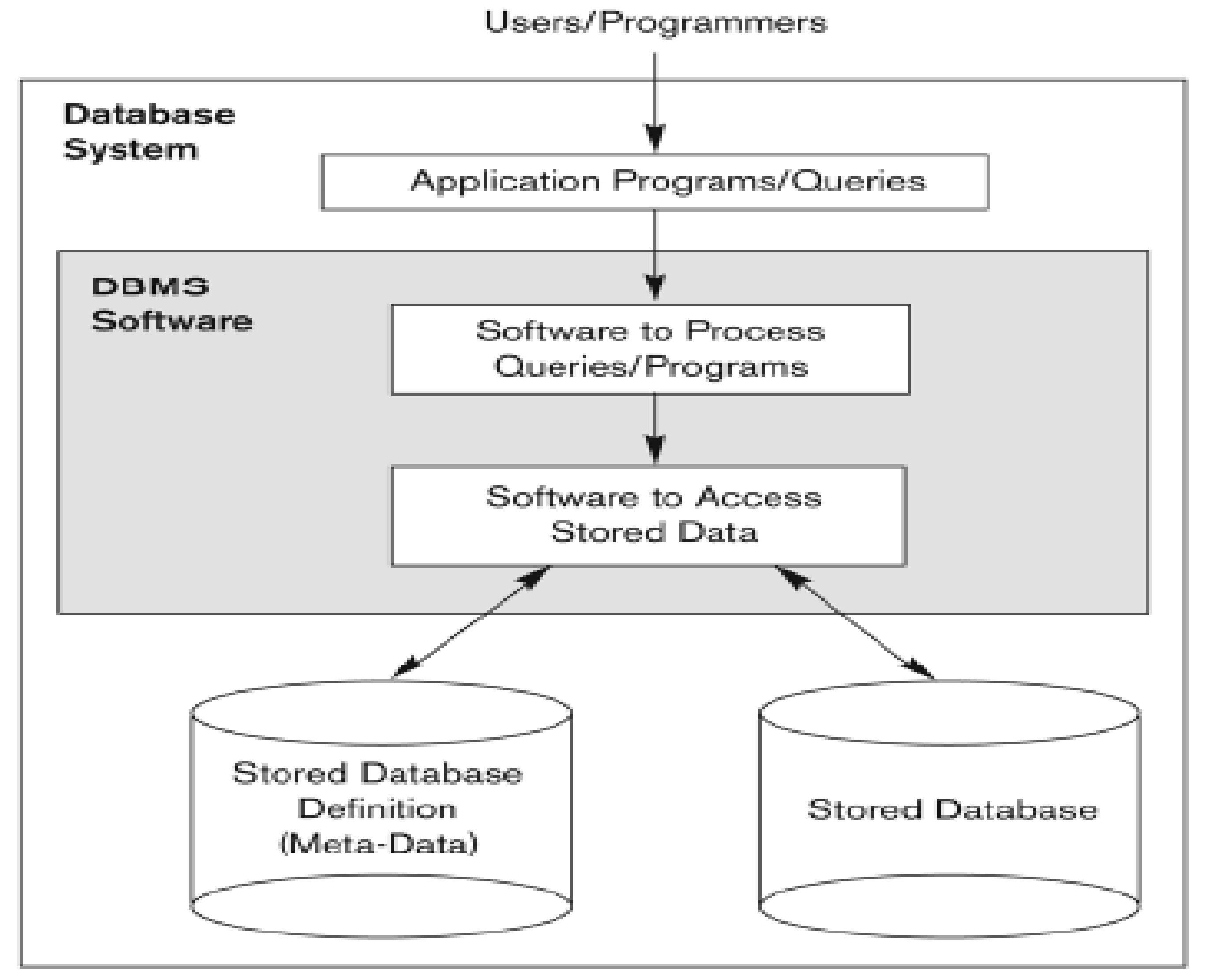
Relation_name	No_of_columns
STUDENT	4
COURSE	4
SECTION	5
GRADE_REPORT	3
PREREQUISITE	2



COLUMNS

Column_name	Data_type	Belongs_to_relation
Name	Character (30)	STUDENT
Student_number	Character (4)	STUDENT
Class	Integer (1)	STUDENT
Major	Major_type	STUDENT
Course_name	Character (10)	COURSE
Course_number	XXXXNNNN	COURSE
....	....	.....
....	....	.....
....	....	.....
Prerequisite_number	XXXXNNNN	PREREQUISITE

Simplified database system environment ☺


Users/Programmers
Database System
Application Programs/Queries
DBMS Software
Software to Process Queries/Programs
Software to Access Stored Data
Stored Database Definition (Meta-Data)
Stored Database

Managing Data

- There are two approaches to manage data

1. **File-based approach**:

   - ❖An approach that utilizes a collection of application programs which performs services to end-users (e.g., Reports).
   - ❖**Each program defines and manages its own data.**

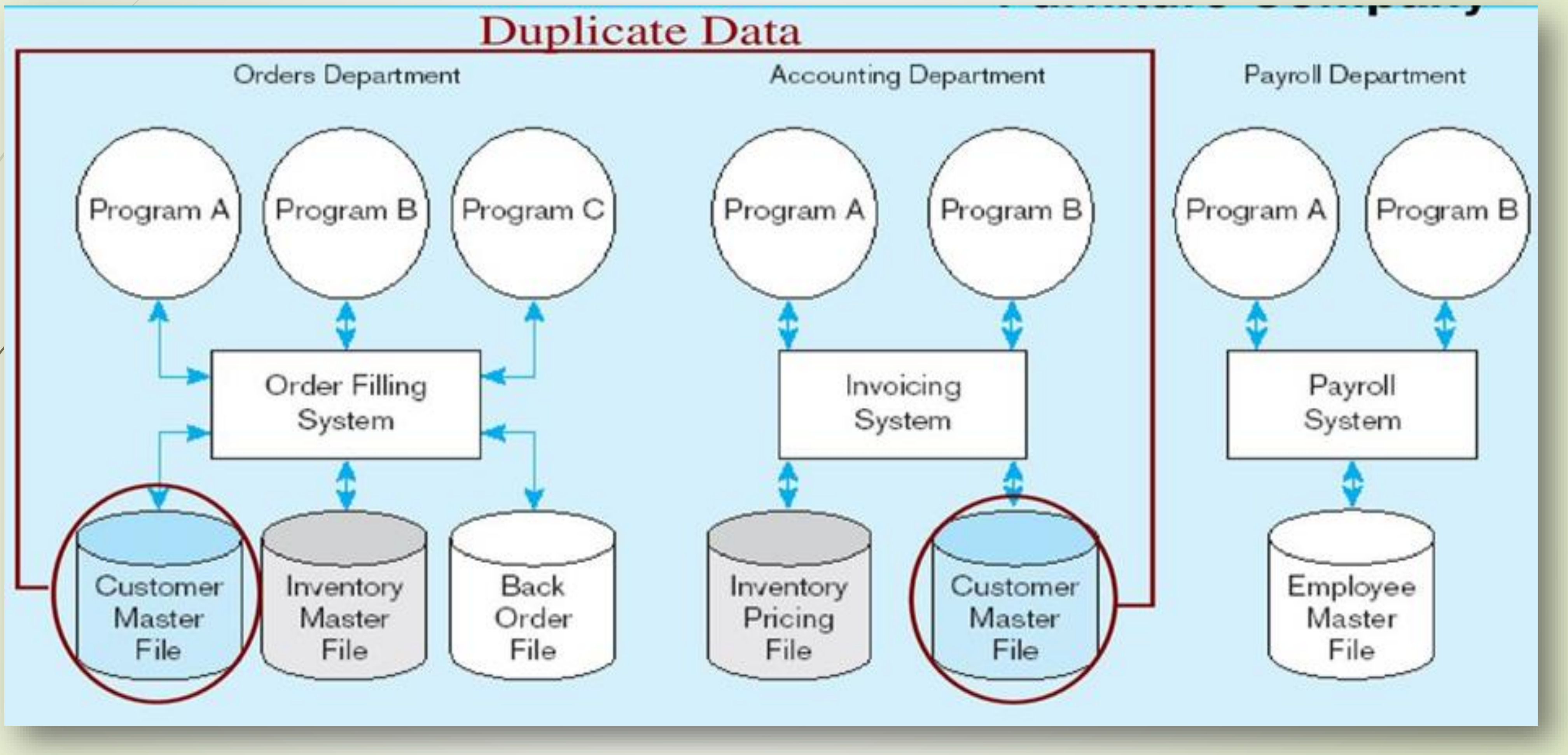
2. **Database approach**:

   - ❖ An approach that data is collected and manipulated using specific software called Database Management System, and **many programs share this data.**

Disadvantages of **file processing** approach

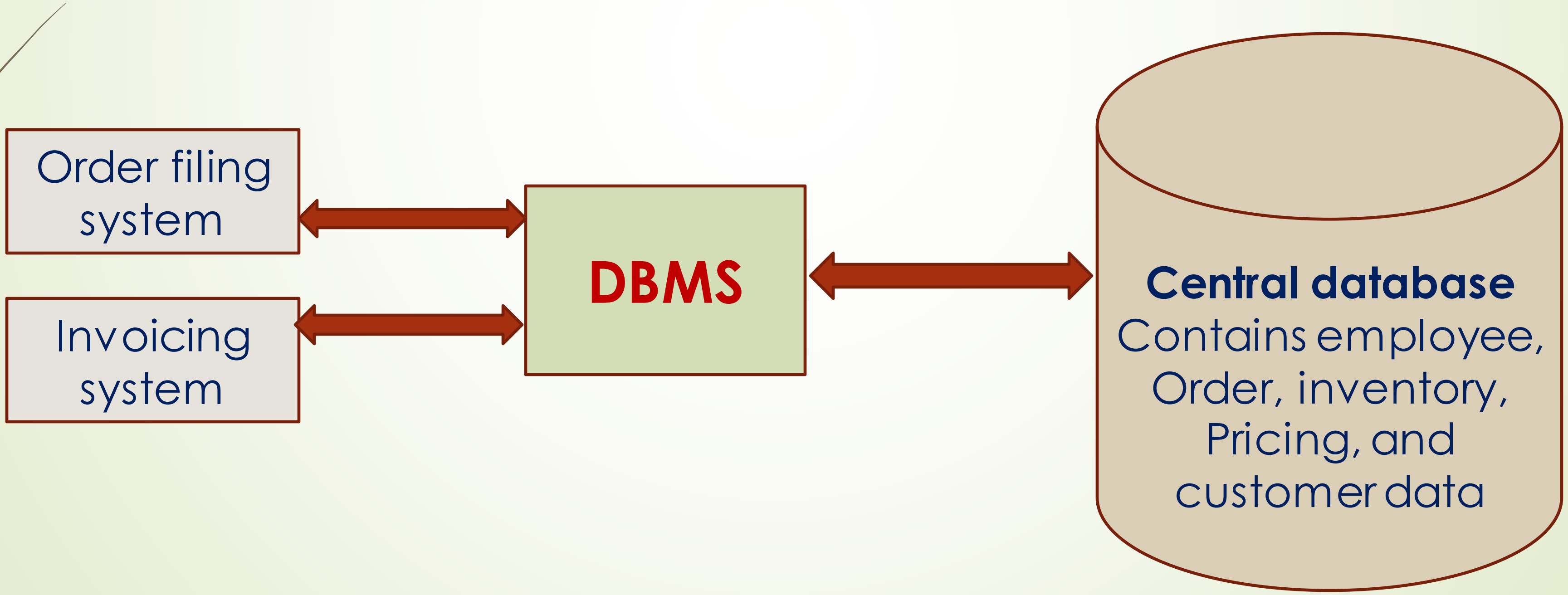
- **Program data dependance**:
  - All programs maintain meta data for each file they use.
- **Duplication of data**:
  - Different system/programs have separate copies of the same data(***no sharing***)
- **Limited data sharing**:
  - No centralized control of data.
- **Lengthy development times**:
  - Programmers must design their own file formats.
- **Excessive program maintenance**:
  - 80% of information system budget.

Disadvantages of **file processing** approach



Database approach is the solution

- Central storage of **shared data**.
- Data is managed by a controlling agent.
- Stored in standardized convenient form.


Order filing system
Invoicing system
DBMS
Central database
Contains employee, Order, inventory, Pricing, and customer data

Main Characteristics of the **Database Approach**

- **Self-describing nature** of a database system:
  - A **DBMS catalog** stores the description of a particular database (e.g., data structures, types, storage format, and **constraints**).
  - The description is called **meta-data**.
  - This allows the DBMS software to work with different database applications; because DBMS use catalog.
- **Insulation** between **programs** and **data**:
  - Called **program-data independence (PDI)**.
  - Allows changing data structures and storage organization without having to change the DBMS access programs.
  - **Program operation independence (POI)** (OO system).
  - **Data abstraction** = PDI + POI.

Main Characteristics of the Database Approach (continued)…

- **Data Abstraction**:
  - A **data model** is a type of data abstraction (ERD,…).
  - It hide storage details and present the users with a conceptual view of the database.
  - Programs refer to the data model constructs rather than data storage details.
- Support of **multiple views** of the data:
  - Each user may see a different view of the database, which describes **only** the data of interest to that user.
  - This characteristic used in **security**.

Main Characteristics of the Database Approach (continued)…

- **Sharing of data** and **multi-user transaction** processing:
  - ❖Allowing a set of **concurrent users** to retrieve from and to update the database.
  - ❖Concurrency control within the DBMS guarantees that each **transaction** is correctly executed or aborted (ticket reservation).
  - ❖Recovery subsystem ensures each completed transaction has its effect permanently recorded in the database
  - ❖**OLTP** (Online Transaction Processing) is a major part of database applications. This allows hundreds of concurrent transactions to execute per second (Bank transfer).

Advantages of Using the **DBMS** Approach

1. Controlling redundancy in data storage and in development and maintenance efforts.
   - ➢ Sharing of data among multiple users.
2. Restricting unauthorized access to data.
   - ➢ Security and privilege
3. Providing persistent storage for program Objects
   - ➢ Save object in database.
   - ➢ Impedance mismatch problem.
4. Providing Storage Structures and search techniques for efficient query processing.
   - ➢ indexes for efficient Query Processing
   - ➢ Buffering and caching.

Advantages of Using the **DBMS** Approach, continued ...

5. Providing backup and recovery services.
6. Providing multiple interfaces to different classes of users (GUIs)
7. Representing complex relationships among data.
8. Enforcing integrity constraints on the database.
9. Drawing inferences and actions from the stored data using deductive and active rules
    - ❖Deductive database
    - ❖Active database (triggers)

Additional Implication of Using the Database Approach

- Potential for enforcing standards:
  - This is very crucial for the success of database applications in large organizations. Standards refer to data item names, display formats, screens, report structures, meta-data (description of data), Web page layouts, etc.
- Reduced application development time:
  - Incremental time to add each new application is reduced.

Additional Implication of Using the Database Approach, continued …

- **Flexibility** to change data structures:
  - Database structure may evolve as new requirements are defined.
- **Availability** of current information:
  - Extremely important for on-line transaction systems such as airline, hotel, car reservations.
- **Economies** of scale:
  - Wasteful overlap of resources and personnel can be avoided by consolidating data and applications across departments.

When not to use a DBMS

- Main inhibitors (**costs**) of using a DBMS:
  - High initial investment and possible need for additional hardware.
  - Overhead for providing generality, security, concurrency control, recovery, and integrity functions.
- When a DBMS may be unnecessary:
  - If the database and applications are simple, well defined, and not expected to change.
  - If there are stringent real-time requirements that may not be met because of DBMS overhead.
  - If access to data by multiple users is not required.

When not to use a DBMS

- When **no DBMS** may **suffice**:
  - If the database system is not able to handle the complexity of data because of modeling limitations
  - If the database users need special operations not supported by the DBMS.

Team code in Microsoft team

g23d8bv