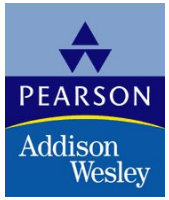
Database Systems

Spring 2024



Course Introduction

- **Course:** Database Systems (3 credit Hours theory + 1 Credit hour Lab)
- **Course Coordinator** :
- **Text Book:**
  - Ramez Elmasri, Fundamentals of Database Systems (5th /6th Edition)
  - Link :
- **Marks Division:**
  - Quizzes/Class participation 10%
  - Assignments 10%
  - Midterms 30%
  - Final 50%

Plagiarism policy

- Any attempt to plagiarize will result in ZERO. No further arguments in this case.

- **Marks Division(Tentative)**
  - Quizzes /Class Participation 10%
  - Assignments 10%
  - Midterms 30%
  - Final 50%

Attendance Policy

- Do not waste your leaves.
- Try to come to class on time.
- We can wait 5 minutes after that attendance will be marked.
- Students are **requested** not to leave during the class. Otherwise your attendance will be updated to Absent.
- Students coming late will be marked as late.

Quiz policy

- Quiz can be announced or unannounced.
- There will be **no retakes** of quizzes.
- Average of all quizzes will be marked
- Class participation will be marked based on your valuable participation. Quiz can also be conducted from the current lecture that will be marked in class participation.

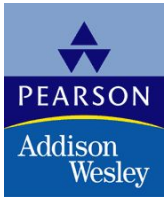
Announcements

- Make sure you check your classroom regarding all the contents and announcements all the deadlines will be posted there.

- ANY QUESTIONS ??
- THEN LETS GET STARTED …

Chapter 1

Databases and Database Users



Types of Databases and Database Applications

- Traditional Applications:
  - Numeric and Textual Databases
- More Recent Applications:
  - Multimedia Databases
  - Geographic Information Systems (GIS)
  - Biological and Genome Databases
  - Data Warehouses
  - Mobile databases
  - Real-time and Active Databases

Recent Developments (1)

- Social Networks started capturing a lot of information about people and about communications among people-posts, tweets, photos, videos in systems such as:

- Facebook

- Twitter

- Linked-In

- All of the above constitutes data
- Search Engines- Google, Bing, Yahoo : collect their own repository of web pages for searching purposes

Recent Developments (2)

- New Technologies are emerging from the so-called non-database software vendors to manage vast amounts of data generated on the web:

- Big Data storage systems involving large clusters of distributed computers
- NOSQL (Not Only SQL) systems
- A large amount of data now resides on the “cloud” which means it is in huge data centers using thousands of machines.

Introduction to DBMS

- **Data**
  - Meaningful facts, text, graphics, images, sound, video segments
- **Database**
  - An organized collection of logically related data
- **Information**
  - Data processed to be useful in decision making
- **Metadata**
  - Data that describes data
- **Database Management System**
  - Collection of interrelated data and a set of programs to access those data.
- **Goal of DBMS**
  - To provide a way to store and retrieve database information that is both convenient and e

Data vs Information

Baker, Kenneth D.	324917628
Doyle, Joan E.	476193248
Finkle, Clive R.	548429344
Lewis, John C.	551742186
McFerran, Debra R.	409723145
Sisneros, Michael	392416582

Figure 1-1a Data in Context

Class Roster

Course: MGT 500 Business Policy

Semester: Spring 200X

Section: 2

Large volume of facts, difficult to interpret or make decisions based on

Name	ID	Major	GPA
Baker, Kenneth D.	324917628	MGT	2.9
Doyle, Joan E.	476193248	MKT	3.4
Finkle, Clive R.	548429344	PRM	2.8
Lewis, John C.	551742186	MGT	3.7
McFerran, Debra R.	409723145	IS	2.9
Sisneros, Michael	392416582	ACCT	3.3

Figure 1-1b Summarized data

Useful information that managers can use for decision making and interpretation

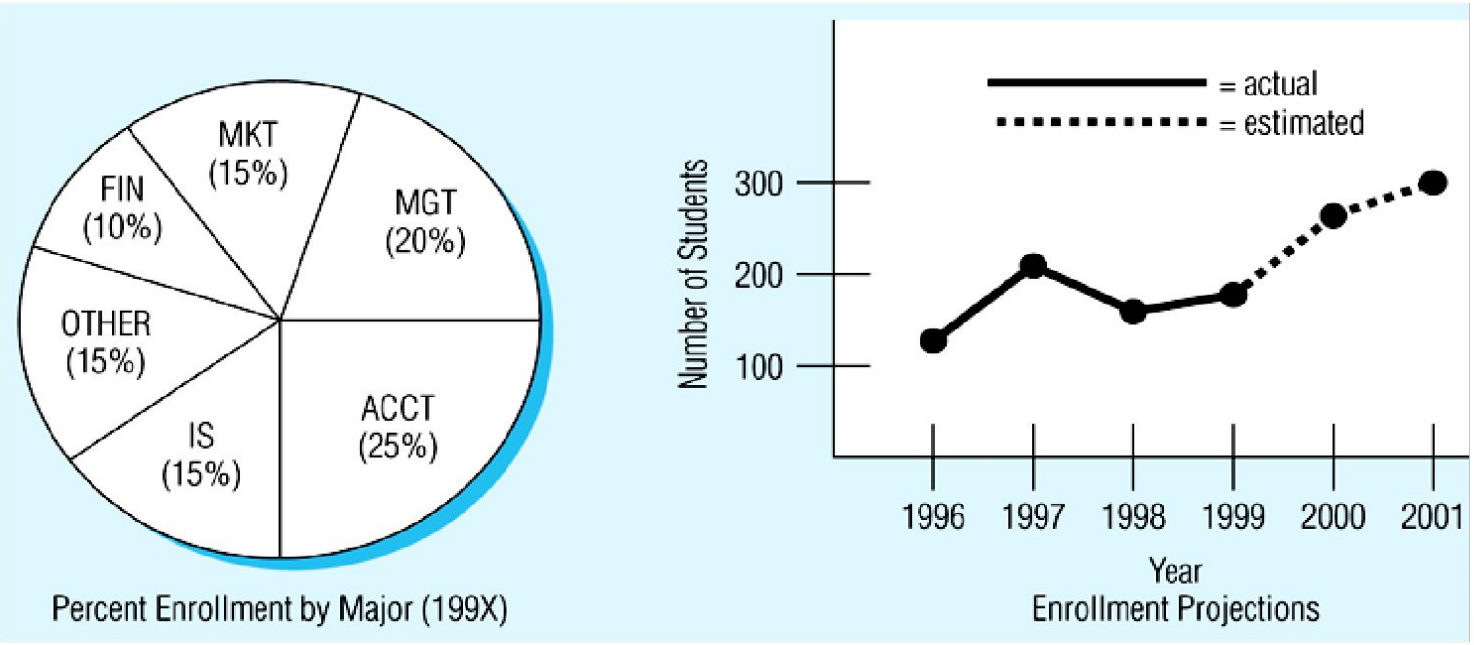


Table 1-1 Metadata

Descriptions of the properties or characteristics of the data, including data types, field sizes, allowable values, and documentation

Table 1-1 Example Metadata for Class Roster

Data Item		*Value*			
Name	**Type**	**Length**	**Min**	**Max**	**Description**
Course	Alphanumeric	30			Course ID and name
Section	Integer	1	1	9	Section number
Semester	Alphanumeric	10			Semester and year
Name	Alphanumeric	30			Student name
ID	Integer	9			Student ID (SSN)
Major	Alphanumeric	4			Student major
GPA	Decimal	3	0.0	4.0	Student grade point average

Introduction to DBMS

- **Why Required?**
  - To manage large bodies of information.
  - Management involves both defining structure for storage of information and providing mechanisms for the manipulation of information.
- **Examples:**
- **Banking:**
  - For customer information, accounts, loans and banking transactions etc.
- **Airlines:**
  - For reservations and schedule information.
- **Universities:**
  - For student information, course registration and grades.

Introduction to DBMS

- The internet revolution of the late 1990's sharply increased direct user access to databases.
- **Example:**
  - Through printed reports. E.g. credit card statements
  - Airline reservation agents.
  - For flight arrival/departure information use of dial phones.
  - Library card system (example of manual database)
- Emergence of web interfaces provided direct database interaction. Data on website is directly fetched from the database.
- An example of a large commercial database is Amazon.com

Types of databases

- Traditional databases
- Multimedia Databases

Properties of Databases

A database has the following implicit properties:

- A database represents some aspect of the real world, sometimes called the **miniworld** or the **universe of discourse (UoD)**. Changes to the miniworld are reflected in the database.

- A database is a logically coherent(logical) collection of data with some inherent meaning. A random assortment of data cannot correctly be referred to as a database.

- A database is designed, built, and populated with data for a specific purpose. It has an intended group of users and some preconceived applications in which these users are interested.

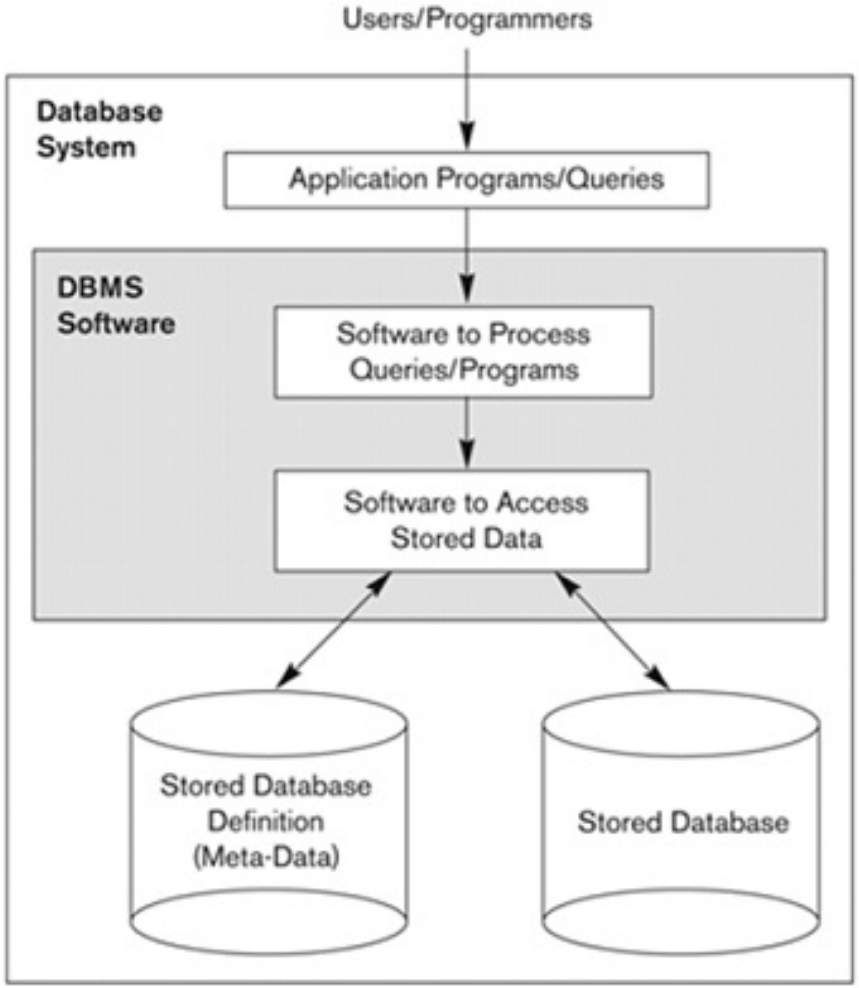
Impact of Databases and Database Technology

- Businesses: Banking, Insurance, Retail, Transportation, Healthcare, Manufacturing
- Service Industries: Financial, Real-estate, Legal, Electronic Commerce, Small businesses
- Education : Resources for content and Delivery
- More recently: Social Networks, Environmental and Scientific Applications, Medicine and Genetics
- Personalized Applications: based on smart mobile devices

DBMS Functionality

- **Define** a database in terms of data types, structures and constraints
- **Construct** or Load the Database on a secondary storage medium
- **Manipulating** the database by querying, generating reports, insertions, deletions and modifications to its content
- **Sharing** by a set of users and programs but keeping the data consistent at the same time.
- **Security** measures to prevent unauthorized access.
- **Presentation** and Visualization of data

A simplified database system environment



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

Application Activities Against a Database

- Applications interact with a database by generating

- Queries: that access different parts of data and formulate the result of a request

- Transactions: that may read some data and “update” certain values or generate new data and store that in the database

- Applications must not allow unauthorized users to access data
- Applications must keep up with changing user requirements against the database

Additional DBMS Functionality

- DBMS may additionally provide:
  - Protection or Security measures to prevent unauthorized access
  - “Active” processing to take internal actions on data
  - Presentation and Visualization of data
  - Maintenance of the database and associated programs over the lifetime of the database application
    - Called database, software, and system maintenance

Simple Example of a Database

- **Mini-world for the example**:
  - Part of a UNIVERSITY environment.

- **Some mini-world *entities***:
  - The STUDENT file stores data of each student
  - The COURSE file stores data of each course
  - The SECTION file stores data of each section of a course
  - The GRADE_REPORT file stores the grades that students receive in the various sections they have completed
  - The PREREQUISITE file stores the prerequisites of each course.

Example of a Database

- Notice that records in the various files may be related.

Some mini-world *relationships*:

- SECTIONs *are of* specific COURSEs
- STUDENTs *take* SECTIONs
- COURSEs *have* prerequisite COURSEs
- COURSEs *are offered by* DEPARTMENTs
- STUDENTs *major in* DEPARTMENTs

Example of a Database

- For example, the record for Smith in the STUDENT file is related to two records in the GRADE_REPORT file that specify Smith's grades in two sections.

- Similarly, each record in the PREREQUISITE file relates two course records: one representing the course and the other representing the prerequisite.

Examples of Queries/Updates

- A list of all courses and grades of a ‘Smith’
- List of pre requisite courses of ‘Database’

- Change the class of ‘smith’ to ‘3’
- Create a new section for Database course for this semester

STUDENT

Name	Student_number	Class	Major
Smith	17	1	CS
Brown	8	2	CS

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

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	07	King
92	CS1310	Fall	07	Anderson
102	CS3320	Spring	08	Knuth
112	MATH2410	Fall	08	Chang
119	CS1310	Fall	08	Anderson
135	CS3380	Fall	08	Stone

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

PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

Design Phases of Database

1. Requirement Definition and analysis
   - Design of a new application for an existing database

     OR
   - Design of a new database

2. Conceptual Design

- These requirements are documented in detail to form a conceptual design

-Conceptual design can be represented and manipulated using some computerized tool so that it can easily be maintained, modified and transformed to database implementation.

Example: Entity Relationship Model

Design Phases of Database

3. Logical Design
   - Conceptual design is transformed into logical design
   - Logical design can be expressed in a data model
   - Example: Relational Data Model

4. Physical Design
   - In this stage further specifications are provided for storing and accessing the database.
   - Database design is implemented , populated with actual data and continuously maintained to reflect the actual state of miniworld.

File System vs Database approach

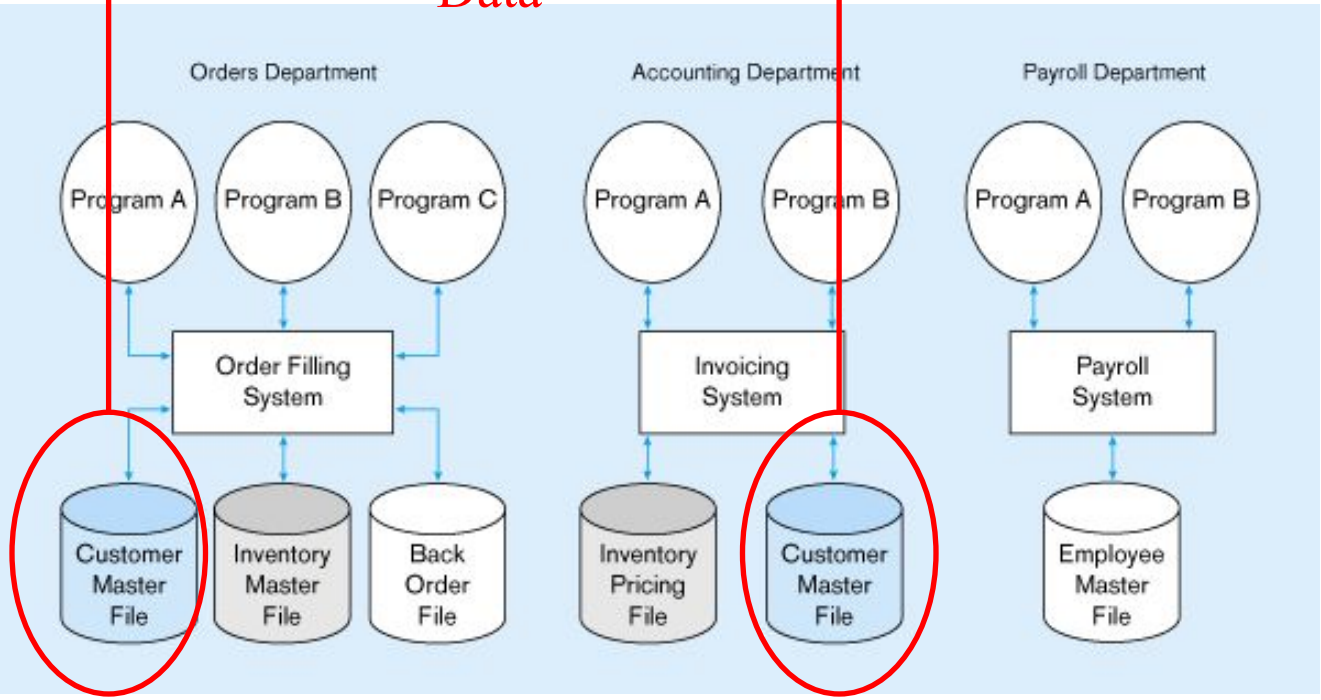
- Student Fee and Transcript Information
- In the database approach, a single repository maintains data that is defined once and then accessed by various users.
- In file systems, each application is free to name data elements independently. In contrast, in a database, the names or labels of data are defined once, and used repeatedly by queries, transactions, and applications.

Disadvantages of File Processing Systems

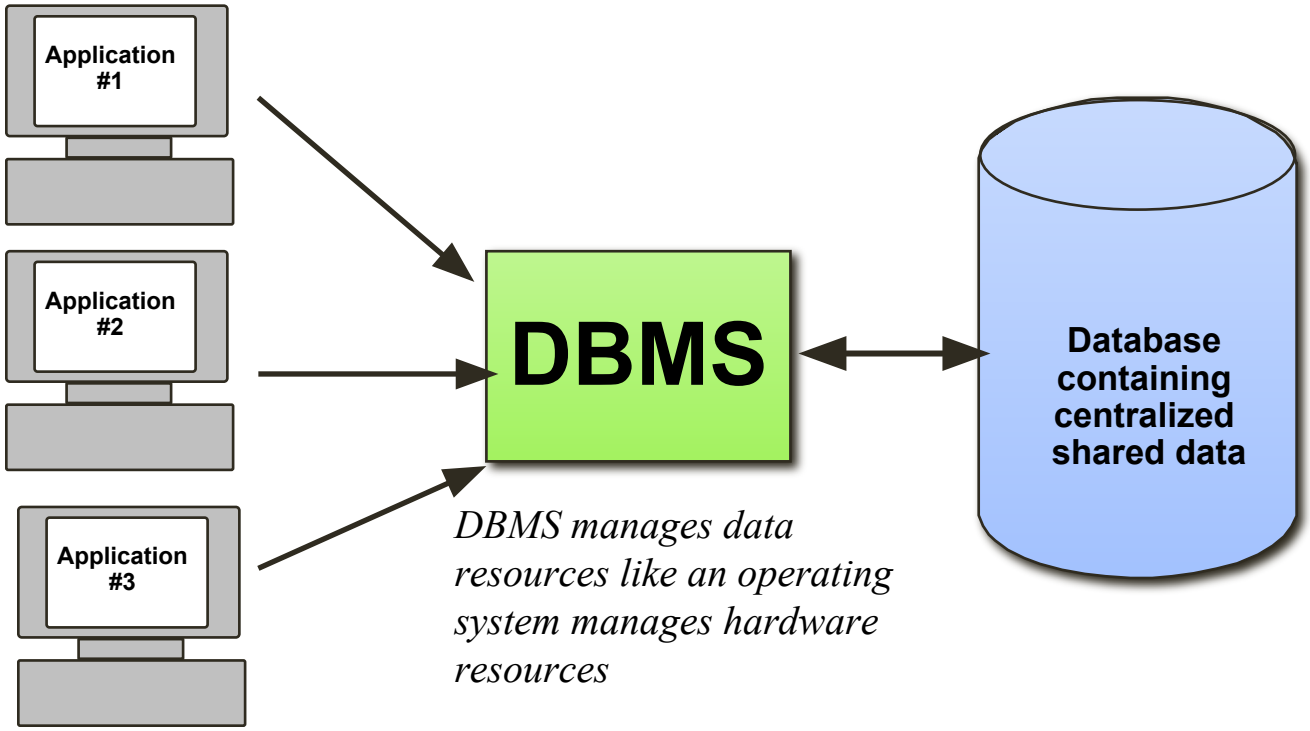
- **Program-Data Dependence**
  - All programs maintain metadata for each file they use
- **Data Redundancy (Duplication of data)**
  - Different systems/programs have separate copies of the same data
- **Limited Data Sharing**
  - No centralized control of data
- **Lengthy Development Times**
  - Programmers must design their own file formats
- **Excessive Program Maintenance**
  - 80% of information systems budget

Figure 1-2 Three file processing systems at Pine Valley Furniture

Duplicate Data



Database Management System



DBMS manages data resources like an operating system manages hardware resources

Advantages of Database Approach

- Program-Data Independence
  - Metadata stored in DBMS, so applications don't need to worry about data formats
  - Data queries/updates managed by DBMS so programs don't need to process data access routines
  - Results in: increased application development and maintenance productivity
- Minimal Data Redundancy
  - Leads to increased data integrity/consistency
- Improved Data Sharing
  - Different users get different views of the data
- Enforcement of Standards
  - All data access is done in the same way
- Improved Data Quality
  - Constraints, data validation rules
- Better Data Accessibility/ Responsiveness
  - Use of standard data query language (SQL)
- Security, Backup/Recovery, Concurrency
  - Disaster recovery is easier

Main Characteristics of the Database Approach

- Self-describing nature of a database system: A DBMS **catalog** stores the *description* of the database. The description is called **meta-data**. This allows the DBMS software to work with different databases.
- Insulation between programs and data: Called **program-data independence**. Allows changing data storage structures and operations without having to change the DBMS access programs.

Metadata/Data Dictionary/Catalo

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

Main Characteristics of the Database Approach

- Data Abstraction: A **data model** is used to hide storage details and present the users with a *conceptual view* of the database.
  - **program-operation independence**

- 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.

Views

TRANSCRIPT

Student_name	Student_transcript				
	Course_number	Grade	Semester	Year	Section_id
Smith	CS1310	C	Fall	08	119
	MATH2410	B	Fall	08	112
Brown	MATH2410	A	Fall	07	85
	CS1310	A	Fall	07	92
	CS3320	B	Spring	08	102
	CS3380	A	Fall	08	135

COURSE_PREREQUISITES

Course_name	Course_number	Prerequisites
Database	CS3380	CS3320
		MATH2410
Data Structures	CS3320	CS1310

Main Characteristics of the Database Approach

- Sharing of data and multiuser transaction processing : allowing a set of concurrent users to retrieve and to update the database.
- **Concurrency control** within the DBMS guarantees that each **transaction** is correctly executed or completely aborted.

Example Flight reservation

Transaction

- A transaction is an *executing program* or *process* that includes one or more database accesses, such as reading or updating of database records.

- Each transaction is supposed to execute a logically correct database access if executed in its entirety without interference from other transactions.

Database Users

Users may be divided

1. **Actors on the Scene :**

   Those who actually use and control the content

2. **Workers Behind the Scene:**

   Associated with the design, development, and operation of the DBMS *software and system environment.* These persons are typically not interested in the database content itself.

Database Users

Actors on the scene

- **Database administrators:** responsible for authorizing access to the database, for co-ordinating and monitoring its use, acquiring software, and hardware resources, controlling its use and monitoring efficiency of operations.
- **Database Designers:** responsible to define the content, the structure, the constraints, and functions or transactions against the database. They must communicate with the end-users and understand their needs.
- **End-users:** they use the data for queries, reports and some of them actually update the database content.

Categories of End-users

- **Casual** : access database occasionally when needed
- **Naïve or Parametric** : they make up a large section of the end-user population. They use previously well-defined functions in the form of "canned transactions" against the database. Examples are bank-tellers or reservation clerks who do this activity for an entire shift of operations.

Categories of End-users

- **Sophisticated** : these include business analysts, scientists, engineers, others thoroughly familiar with the system capabilities. Many use tools in the form of software packages that work closely with the stored database.
- **Stand-alone** : mostly maintain personal databases using ready-to-use packaged applications. An example is a tax program user that creates his or her own internal database.

Categories of End-users

- **Application Programmers/ Software Engineers**
- Determine the requirements of end users, especially naive and parametric end users, and develop specifications for standard canned transactions that meet these requirements.

- **Application programmers** implement these specifications as programs; then they test, debug, document, and maintain these canned transactions.

Workers Behind the Scene

- **DBMS system designers and implementers**
  - Design and implement the DBMS modules and interfaces as a software package.
- **Tool developers**
  - **D**esign and implement **tools**—the software packages that facilitate database modeling and design, database system design, and improved performance.
- **Operators and maintenance personnel**
  - **They** are responsible for the actual running and maintenance of the hardware and software environment for the database system.

Advantages of Using the Database Approach

- Controlling **redundancy** in data storage and in development and maintenance efforts.
- Sharing of data among multiple users.
- Restricting unauthorized access to data.
- Providing backup and recovery services.
- Providing multiple interfaces to different classes of users.
- Enforcing integrity constraints on the database.
  - Referential, uniqueness, datatype

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 access to data by multiple users is not required.