Introduction

Database Systems CS203

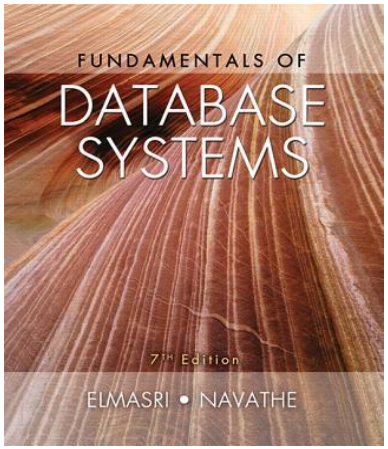
Week 01

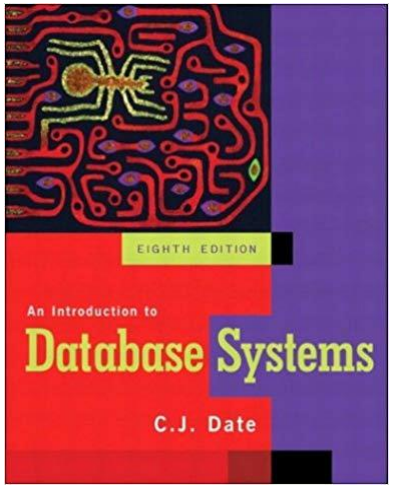
27th-Aug-2018

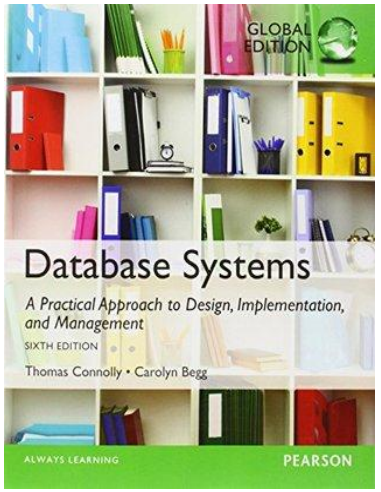
By: Anam Qureshi

Course Information

- Prerequisite: CS-201 Data Structures
- Text Book
  - Ramez Elmasri & Shamkant B. Navathe, *Database Systems, Models, Languages, Design and Application Programming,* 7th Edition, 2016.
- Reference Books
  - Thomas Connolly, Carolyn Begg, *Database Systems: A practical approach to design, implementation and Management*, 6th Edition, 2015
  - C.J. Date, *An Introduction to Database Systems*, 8th Edition, 2004







Tentative Course Outline

Week 1	Chapter No. 1 Introduction, Characteristics of Database Approach, Files Vs. Databases, Characteristics of Database approach, Advantages of using DBMS, When not to use DBMS.
Week 2	Chapter No. 2 Data Model, Schema and Instance, three schema architecture and data independence, classification of DBMS, database languages & Interfaces, Database systems environment. Relational Model Concepts, Relational Model
Week 3	Chapter No. 5 Constraints and Relational Database Schemas, Update Operations, Transactions, and Dealing with Constraint Violations SQL Data Definition and Data Types Specifying Constraints in SQL Basic Retrieval Queries in SQL
Week 4	Chapter No. 6 INSERT, DELETE, and UPDATE Statements in SQL Additional Features of SQL
Week 5	Chapter No. 7 More Complex SQL Retrieval Queries Specifying Constraints as Assertions and Actions as Triggers Views (Virtual Tables) in SQL Schema Change Statements in SQL

Tentative Course Outline

Week 6	Mid Term Exam 1
Week 7	Chapter No. 3 Using High-Level Conceptual Data Models for Database Design, A Sample Database Application. Entity Types, Entity Sets, Attributes, and Keys Relationship Types, Relationship Sets, Roles, and Structural Constraints Weak Entity Types, Refining the ER Design for the COMPANY Database ER Diagrams, Naming Conventions, and Design Issues, Relationship Types of Degree Higher than Two Relational Database Design Using ER-to-Relational Mapping (Chapter No. 9)
Week 8	Chapter No. 8 Unary Relational Operations: SELECT and PROJECT Relational Algebra Operations from Set Theory Binary Relational Operations: JOIN and DIVISION Examples of Queries in Relational Algebra The Tuple Relational Calculus /The Domain Relational Calculus
Week 9	Chapter No. 18 Translating SQL Queries into Relational Algebra and Other Operators Algorithms for External Sorting Algorithms for SELECT Operation
Week 10	Chapter No. 14 Informal Design Guidelines for Relation Schemas Functional Dependencies/Normal Forms Based on Primary Keys General Definitions of Second and Third Normal Forms
Week 11	Mid Term Exam 2

Tentative Course Outline

Week 12	Chapter No. 14 Boyce-Codd Normal Form Multivalued Dependency and Fourth Normal Form Join Dependencies and Fifth Normal Form
Week 13	Chapter No. 20 Introduction to Transaction Processing Transaction and System Concepts Desirable Properties of Transactions Characterizing Schedules Based on Recoverability Characterizing Schedules Based on Serializability Transaction Support in SQL
Week 14	Chapter No. 21 Two-Phase Locking Techniques for Concurrency Control Concurrency Control Based on Timestamp Ordering Multiversion Concurrency Control Techniques Validation (Optimistic) Concurrency Control Techniques Granularity of Data Items and Multiple Granularity Locking
Week 15	Chapter No. 22 Recovery Concepts NO-UNDO/REDO Recovery Based on Deferred Update Recovery Techniques Based on Immediate Update
Week 16	Chapter No. 24 Introduction to NOSQL Systems Document-Based NOSQL Systems and MongoDB NOSQL Key-Value Stores Column-Based or Wide Column NOSQL Systems

Grading Policy

Assignments	5
Quizzes	5
Course Project	10
Mid Exams	30
Final Exam	50

Course Project

- Desktop application
- Mobile application
- Web application
- Use any platform
- Apply the learned concepts
- Team Work (3 Members)
- Start Exploration

How to Get Good Score?

- Be Punctual
- Attend the Class Regularly
- Be Attentive During Lecture
- Ask Questions
- Actively Participate
- Give Suggestions
- Must read the textbook
- Do not CHEAT yourselves

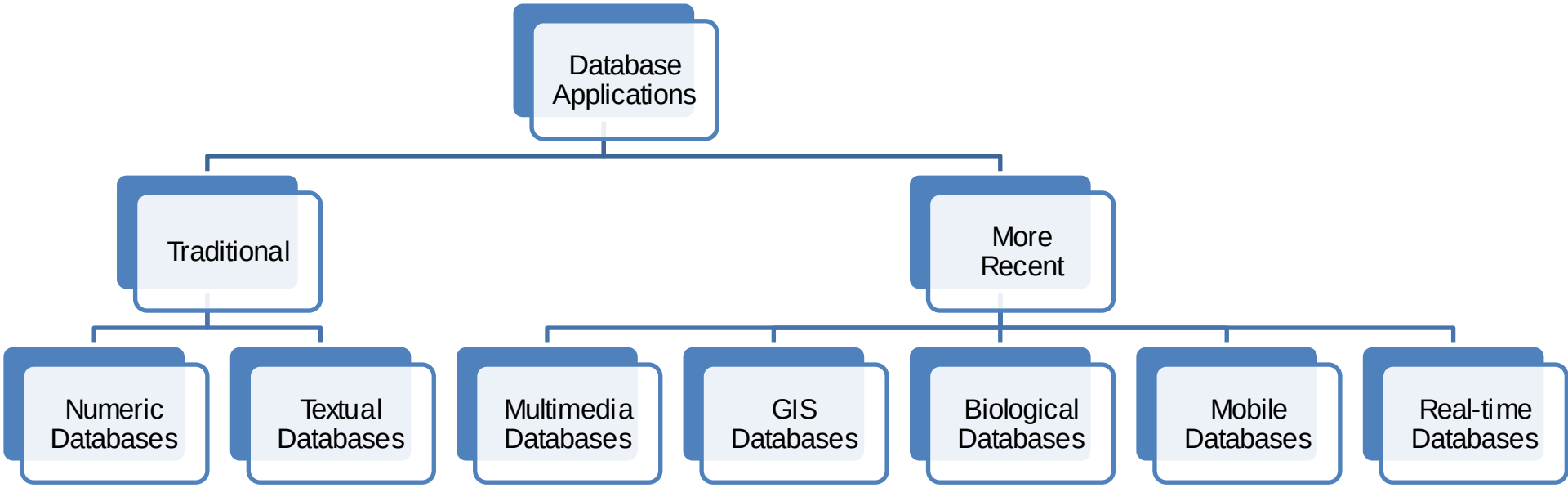
Chapter 01: Databases and Database Users

Outline

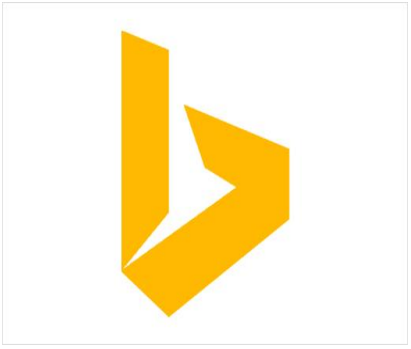
- Types of Databases and Database Applications
- Basic Definitions
- Example of a Database (University)
- Typical DBMS Functionality
- Main Characteristics of the Database Approach
- Types of Database Users
- Advantages of Using the Database Approach
- Historical Development of Database Technology
- Extending Database Capabilities
- When Not to Use Databases

Types of Databases and Database Applications

Types of Databases and Database Applications



Recent Developments



Basic Definitions

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.

An Example of a University Database

Example of a Database

- **Mini-world for the example:**

  Part of a UNIVERSITY environment.

- **Some mini-world *entities*:**

  STUDENTs
  COURSEs
  SECTIONs (of COURSEs)
  (academic) DEPARTMENTs
  INSTRUCTORs

Example of a Database

- **Some mini-world *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

Example of a 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

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

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

Figure 1.2
A database that stores student and course information.

Example of Metadata

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

Note: Major_type is defined as an enumerared type with all known majors. XXXXNNNN is used to define a type with four alpha characters followed by four digits

Figure 1.3
An example of a database catalog for the database in Figure 1.2.

DBMS Functionality

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

Characteristics of Database

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, and constraints)
- The description is called **meta-data***.
- This allows the DBMS software to work with different database applications.

Insulation between programs and data:

- Called **program-data independence**.
- Allows changing data structures and storage organization without having to change the DBMS access programs.

Main Characteristics of the Database Approach (Continued)

Data Abstraction:

- A **data model** is used to 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.

Images of Views

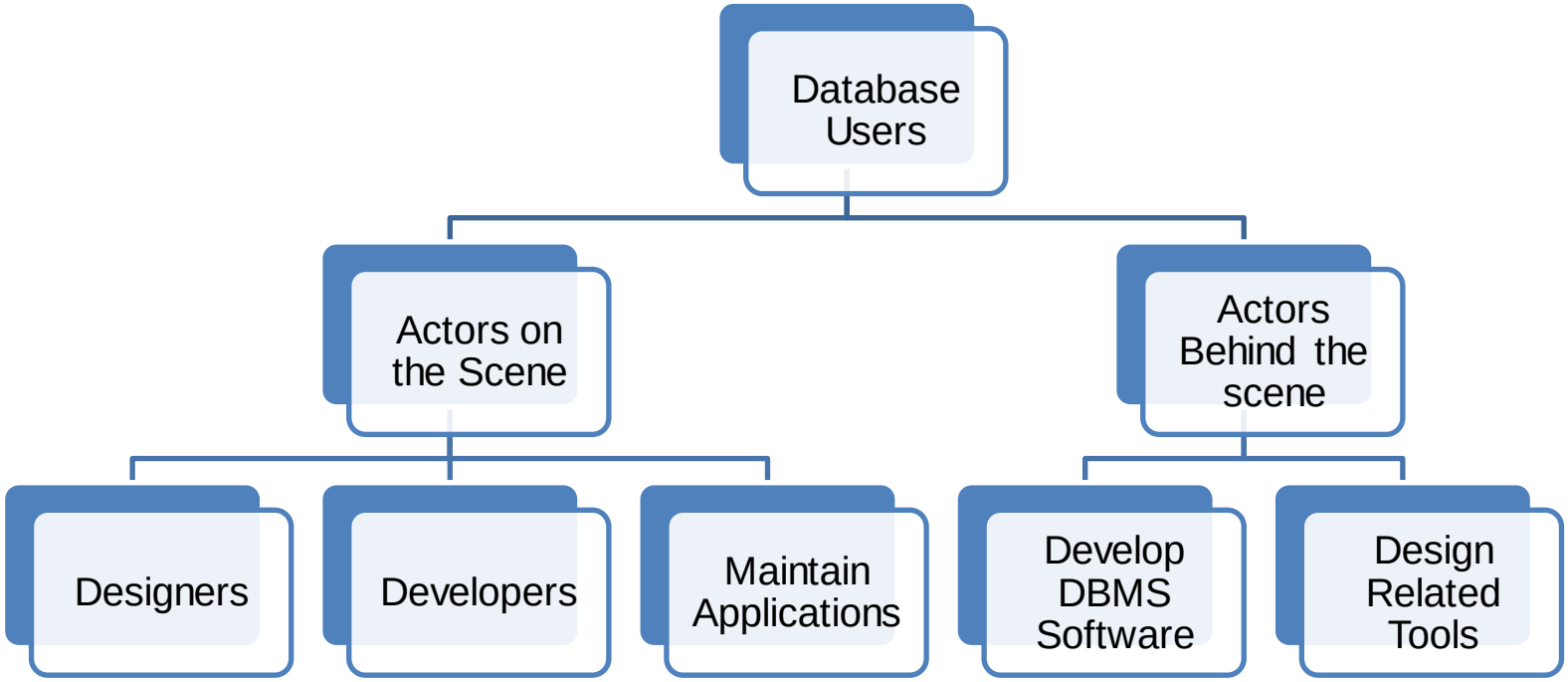
Main Characteristics of the Database Approach (Continued)

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
- *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.

Database Users

Database Users



Database Users – Actors on the Scene

Actors on the scene

Database administrators:

- Responsible for authorizing access to the database, for coordinating 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.

Database End Users

Actors on the scene (continued)

End-users: They use the data for queries, reports and some of them update the database content. End-users can be categorized into:

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.
- Users of Mobile Apps mostly fall in this category
- Bank-tellers or reservation clerks are parametric users who do this activity for an entire shift of operations.
- Social Media Users post and read information from websites

Database End Users (Continued)

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 the user of a tax program that creates its own internal database.
- Another example is a user that maintains a database of personal photos and videos.

Database Users- Actors on the Scene (Continued)

System Analysts and Application Developers

This category currently accounts for a very large proportion of the IT work force.

System Analysts: They understand the user requirements of naïve and sophisticated users and design applications including canned transactions to meet those requirements.

Application Programmers: Implement the specifications developed by analysts and test and debug them before deployment.

Business Analysts: There is an increasing need for such people who can analyze vast amounts of business data and real-time data ("Big Data") for better decision making related to planning, advertising, marketing etc.

Database Users- Actors Behind the Scene

***System Designers and Implementors*:** Design and implement DBMS packages in the form of modules and interfaces and test and debug them. The DBMS must interface with applications, language compilers, operating system components, etc.

Tool Developers: Design and implement software systems called tools for modeling and designing databases, performance monitoring, prototyping, test data generation, user interface creation, simulation etc. that facilitate building of applications and allow using database effectively.

Operators and Maintenance Personnel: They manage the actual running and maintenance of the database system hardware and software environment.