



Database Systems

Lecture 1: Course Overview

Dr. Momtazi
momtazi@aut.ac.ir

based on the slides of the course book



Outline

- **Administrative Information**
- Introduction to the Course
- Overview of the Semester



Course

- Credit point: 3
- Lecture
 - Saturdays 10:45 - 12:15
 - Mondays 10:45 - 12:15



Course Home Page

- Administrative information
- Slides
- Exercises



Assessment

- Regular attendance in the class
 - More than 3/16 absences will be reported
 - Do not enter the class when the door is close

- Final exam (40%)
- Midterm exam (30%)
- Exercises (15%)
- Final project (15%)



Teaching

- Both theoretical and practical concepts in main sessions
- More practical sessions by teacher assistant
 - Goals:
 - Solving exercises
 - Teaching MySQL and MongoDB
 - Timeslot: ?



Contact

- Email: momtazi@aut.ac.ir
- Consultations: Sundays 13:30– 15:00
- Phone: 021-64542737



Text Book

Database System Concepts

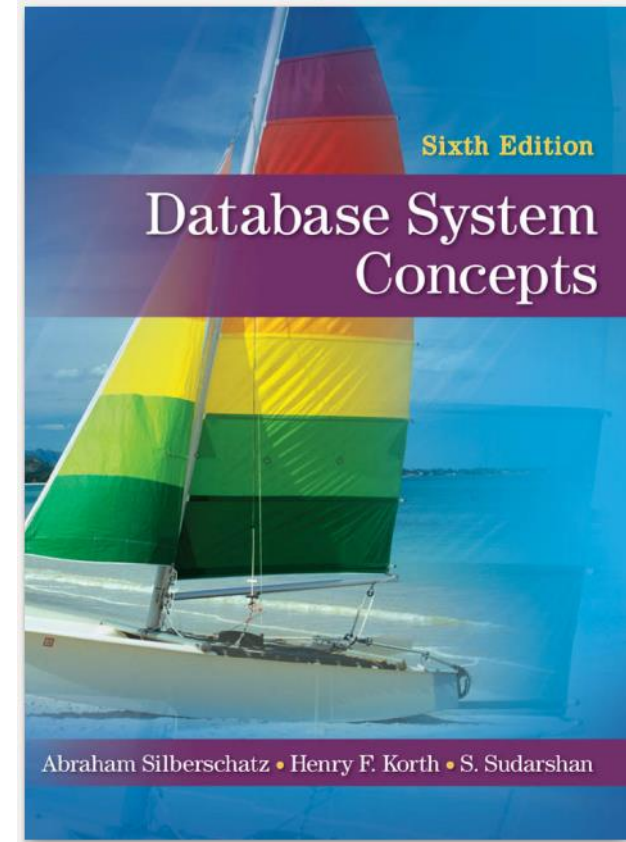
by Abraham Silberschatz

Henry F. Korth

S. Sudarshan

SIXTH EDITION

Publisher: McGraw-Hill





Rules of the Game

- In case you don't understand something:
 - Ask!!!
 - Ask!!!
 - Ask!!!



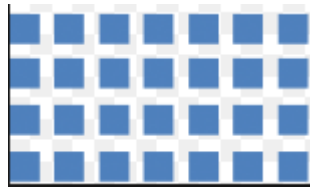
Outline

- Administrative Information
- **Introduction to the Course**
- Overview of the Semester

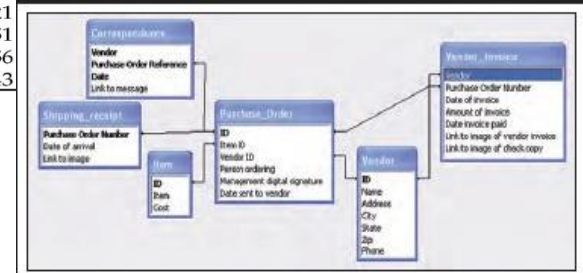


Data

Structured



ID	name	dept_name	salary
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821			
15151			
33456			
76543			



Unstructured



Techniques such as [data mining](#), [Natural Language Processing \(NLP\)](#), and [text analytics](#) provide different means to interpret this information. Common techniques for structuring text usually involve manual [tagging](#) with [metatags](#) or further [text mining](#)-based structuring. [Unstructured Information Management Architecture \(UIMA\)](#) provides a framework for information to extract meaning and create structured data about the information.^[5]

Software that creates machine-processable structure exploits the linguistic, auditory, and visual structure in communication.^[6] Algorithms can infer this inherent structure from text, for instance, by examining word [small](#)- and [large-scale](#) patterns. Unstructured information can then be enriched and tagged to address ambiguity then used to facilitate search and discovery. Examples of "unstructured data" may include books, journals, [audio](#), [video](#), [analog data](#), images, files, and unstructured text such as the body of an [e-mail](#) message. [Web](#) pages, the main content being conveyed does not have a defined structure, it generally comes packaged in objects themselves have structure and are thus a mix of structured and unstructured data, but collectively this is still unstructured. For example, an [HTML](#) web page is tagged, but HTML mark-up typically serves solely for rendering. It does not contain elements in ways that support automated processing of the information content of the page. [XHTML](#) tagging elements, although it typically does not capture or convey the semantic meaning of tagged terms.

Since unstructured data commonly occurs in [electronic documents](#), the use of a [content](#) or [document management system](#) for entire documents is often preferred over data transfer and manipulation from within the documents. Documents are then conveyed structure onto [document collections](#).

[Search engines](#) have become popular tools for indexing and searching through such data, especially text.

A small sailboat with a vibrant rainbow-colored sail is beached on a sandy shore. The sail features horizontal stripes of red, orange, yellow, green, and blue. The boat is white and positioned on the left side of the frame. The background shows a clear blue sky with some clouds and a calm sea extending to the horizon.



Data

■ Storage



■ Retrieval





The Need for Databases

- The Internet revolution of the late 1990s sharply increased direct user access to databases.
- Converting many of phone interfaces into Web interfaces
- Making a variety of services and information available online.
 - Accessing an online bookstore and browse a book or music collection
 - Entering an order online
 - Accessing a bank Web site and retrieving bank balance and transaction information
 - Accessing a Web site and browsing its advertisement



The Need for Databases

- Database system vendors are among the largest software companies in the world





Database Management System (DBMS)

- DBMS contains information about a particular enterprise
 - Collection of interrelated data
 - Set of programs to access the data
 - An environment that is both *convenient* and *efficient* to use
- Databases can be very large.
- Databases touch all aspects of our lives



Database Applications

- Banking: transactions
- Airlines: reservations, schedules
- Universities: registration, grades
- Sales: customers, products, purchases
- Online retailers: order tracking, customized recommendations
- Manufacturing: production, inventory, orders, supply chain
- Human resources: employee records, salaries, tax deductions



University Database Example

- Application program examples
 - Add new students, instructors, and courses
 - Register students for courses, and generate class rosters
 - Assign grades to students, compute grade point averages (GPA) and generate transcripts
- In the early days, database applications were built directly on top of file systems



Drawbacks of using file systems to store data

- Data redundancy and inconsistency
 - Multiple file formats, duplication of information in different files
- Difficulty in accessing data
 - Need to write a new program to carry out each new task
- Data isolation
 - Multiple files and formats
- Integrity problems
 - Integrity constraints (e.g., account balance > 0) become “buried” in program code rather than being stated explicitly
 - Hard to add new constraints or change existing ones



Drawbacks of using file systems to store data

- Atomicity of updates
 - Failures may leave database in an inconsistent state with partial updates carried out
 - Example: Transfer of funds from one account to another should either complete or not happen at all
- Concurrent access by multiple users
 - Concurrent access needed for performance
 - Uncontrolled concurrent accesses can lead to inconsistencies
 - Example: Two people reading a balance (say 100) and updating it by withdrawing money (say 50 each) at the same time
- Security problems
 - Hard to provide user access to some, but not all, data



Outline

- Administrative Information
- Introduction to the Course
- **Overview of the Semester**



View of Data

- A database system is a collection of interrelated data and a set of programs
- Allow users to access and modify these data
- Major purposes:
 - Providing users with an abstract view of the data
 - Hiding certain details of how the data are stored and maintained.



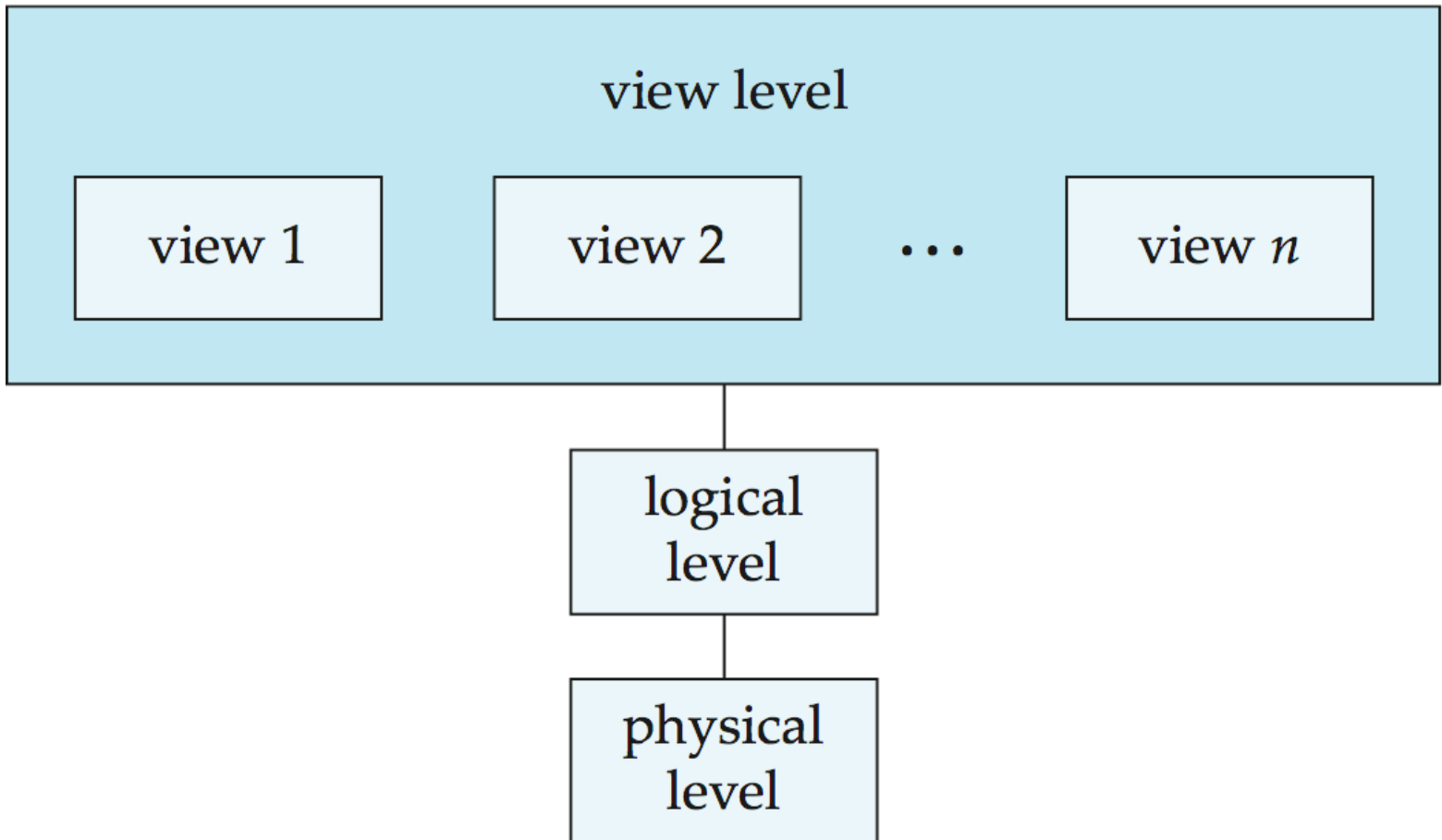
Levels of Abstraction

- **Physical level:** describes how a record (e.g., instructor) is stored.
- **Logical level:** describes data stored in database, and the relationships among the data.
- **View level:** application programs hide details of data types. Views can also hide information (such as an employee's salary) for security purposes.



View of Data

An architecture for a database system





Example

```
type instructor = record  
    ID : string;  
    name : string;  
    dept_name : string;  
    salary : integer;  
end;
```

- A university organization may have several such record types:
 - Department (dept name, building, and budget)
 - Course (course id, title, dept name, and credits)
 - Student (ID , name, dept name, and tot_cred)



Data Models

- A collection of tools for describing
 - Data
 - Data relationships
 - Data semantics
 - Data constraints

- Data Models:
 - Relational model
 - Entity-Relationship data model (mainly for database design)
 - Object-based data models (Object-oriented and Object-relational)
 - Semistructured data model (XML)



Relational Model

- All the data is stored in various tables.
- Example of tabular data in the relational model

Columns

<i>ID</i>	<i>name</i>	<i>dept_name</i>	<i>salary</i>
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

Rows

(a) The *instructor* table



Example of Relational Database

<i>ID</i>	<i>name</i>	<i>dept_name</i>	<i>salary</i>
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

(a) The *instructor* table

<i>dept_name</i>	<i>building</i>	<i>budget</i>
Comp. Sci.	Taylor	100000
Biology	Watson	90000
Elec. Eng.	Taylor	85000
Music	Packard	80000
Finance	Painter	120000
History	Painter	50000
Physics	Watson	70000

(b) The *department* table



Topics

- Data Definition Language (DDL)
- Data Manipulation Language (DML)
- Structured Query Language (SQL)
- Database Design Approaches
 - Entity Relationship (ER) Model
 - Normalization
- Advanced Topics:
 - Semistructured data model (XML)
 - Object-based data models (Object-oriented and Object-relational)
 - NoSQL



Question?