

Chapter 1

Introduction to Database

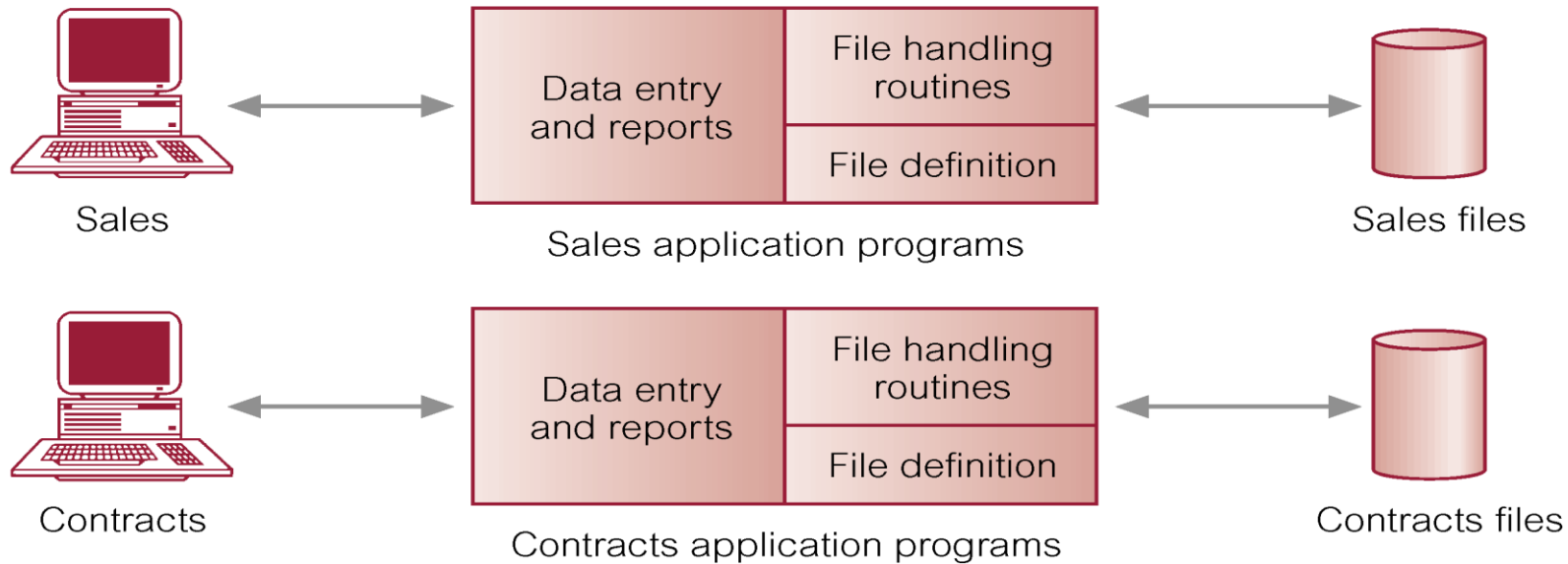
Introduction

- Because **data is so important** in most organizations, computer scientists have developed a large body of concepts and techniques for storing and managing data
- *Data*: Known facts that can be recorded and have an implicit meaning. Ex: name, mobile number, grades, temperature etc.....
- ***What is the difference between data and information***
- Data that need to be stored is everywhere. In universities, supermarkets, banks, interactive voice response (IVR), etc.....
- In the early days, these applications were built directly on top of **file systems**

File Based Systems

- It is a collection of application programs that perform services for the users wishing to access information. Each program defines and manages its own data.
- File-based systems were an early attempt to **computerize** the manual filing system that we are all familiar with
- The manual filing system works well while **the number of items to be stored is small**. It even works quite adequately when there are large numbers of items and we have only to **store and retrieve** them.
- It is not adequate when we have to **cross-reference or process the information in the files**

File Based Processing



Sales Files

PropertyForRent (propertyNo, street, city, postcode, type, rooms, rent, ownerNo)

PrivateOwner (ownerNo, fName, lName, address, telNo)

Client (clientNo, fName, lName, address, telNo, prefType, maxRent)

Contracts Files

Lease (leaseNo, propertyNo, clientNo, rent, paymentMethod, deposit, paid, rentStart, rentFinish, duration)

PropertyForRent (propertyNo, street, city, postcode, rent)

Client (clientNo, fName, lName, address, telNo)

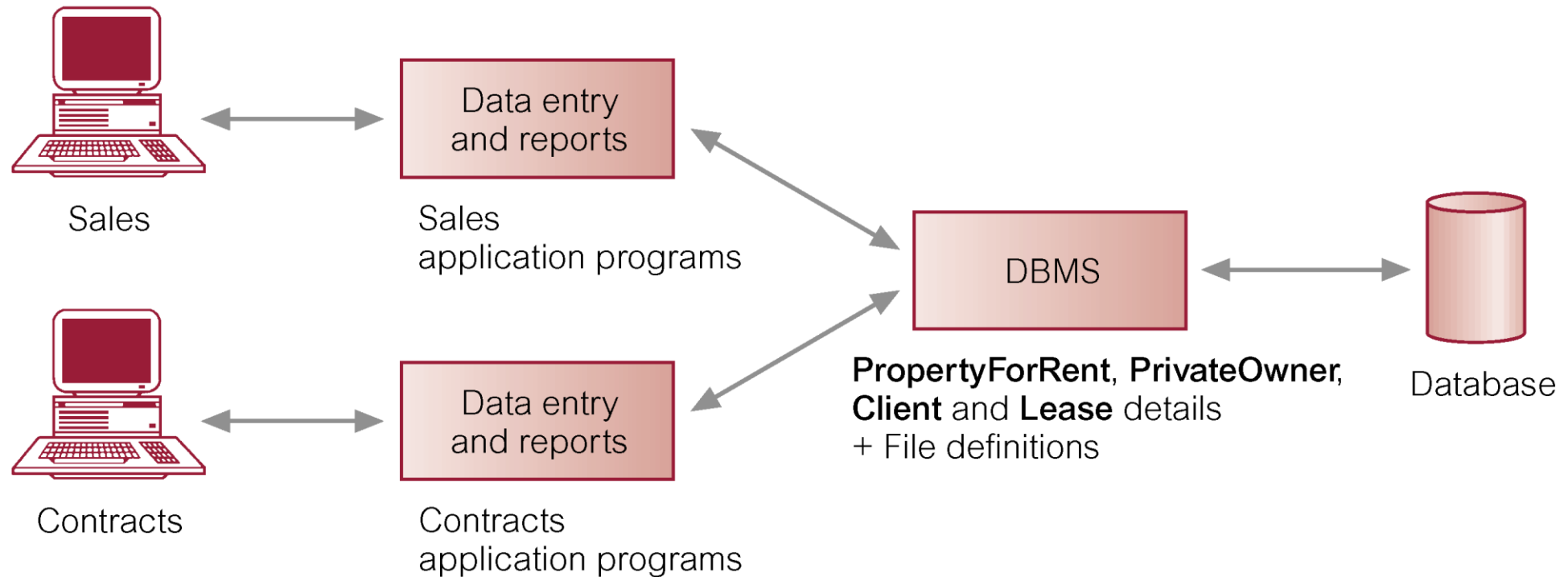
Disadvantages of File Based Systems

- **Separation and isolation of data:** When data is isolated and scattered in separate files and files may be in different format, it is more difficult to access needed data.
- **Duplication of data:** Different systems/programs have separate copies of the same data which result in:
 - Wasting time and storage space to enter data twice
 - Difficulty in maintenance
 - Result in inconsistent data
- **Data Dependence:** The physical structure and storage of the data files and records are defined in the application code. This means that changes to an existing structure are difficult to make

Disadvantages of File Based Systems (cont.)

- **Integrity Problem:** The data values stored in files must satisfy certain types of constraints to ensure its integrity. **Ex: Salary should not exceed 50.000, student ID shouldn't be duplicated**
- **Atomicity Problem:** Transactions done must happen in its entirety or not at all. It is difficult to ensure atomicity in a conventional file based system
- **Security Problems:** Not every user of the system should access all the stored data. Enforcing such security constraint is difficult. Also, there is no mechanism to track who did what when and where.
- **Limited Data Sharing:** Access to the files was restricted to one user at a time
- **Disaster recovery:** Recovery, in the event of a hardware or software failure, was limited or non-existent

Solution: Database



PropertyForRent (propertyNo, street, city, postcode, type, rooms, rent, ownerNo)

PrivateOwner (ownerNo, fName, lName, address, telNo)

Client (clientNo, fName, lName, address, telNo, prefType, maxRent)

Lease (leaseNo, propertyNo, clientNo, paymentMethod, deposit, paid, rentStart, rentFinish)

What is Database

- *Database* is a collection of **related** data
- *Database management system* is the software that enables users to create and maintain database. It facilitates the process of **defining, constructing, manipulating and sharing** databases among users
- *Database application* is a program that interacts with the database at some point in its execution
- *Database represents some aspect of the real world. This is called miniworld*

DBMS Functions

- **Defining a database** involves specifying the data types, structures and constraints of the data to be stored in the database. This definition is also stored in the database and is called **meta data**. This meta data is stored in what is called database catalog or dictionary.
- **Constructing a database** is the process of storing the data on some storage medium that is controlled by the DBMS
- **Manipulating a database** involves functions such as querying the database to retrieve data, updating the database

DBMS Functions

- **Sharing a database** allows multiple users and programs to access the database simultaneously
- **Protecting a database** includes system protection against hardware or software failure and security protection against unauthorized and malicious access.

Database Advantages

- **Control of data redundancy:** The database attempts to control the redundancy by integrating the files so that multiple copies of the same data are not stored
- **Data consistency:** If a data item is stored only once in the database, any update to its value must be performed only once and the new value is available immediately to all users
- **Sharing of data:** data may be accessed by different users at the same time. A fundamental role of multiuser DBMS software is to ensure that **concurrent transactions** operate correctly and efficiently. A **transaction** is a **process that includes one or more database accesses**, such as reading or updating of database records.

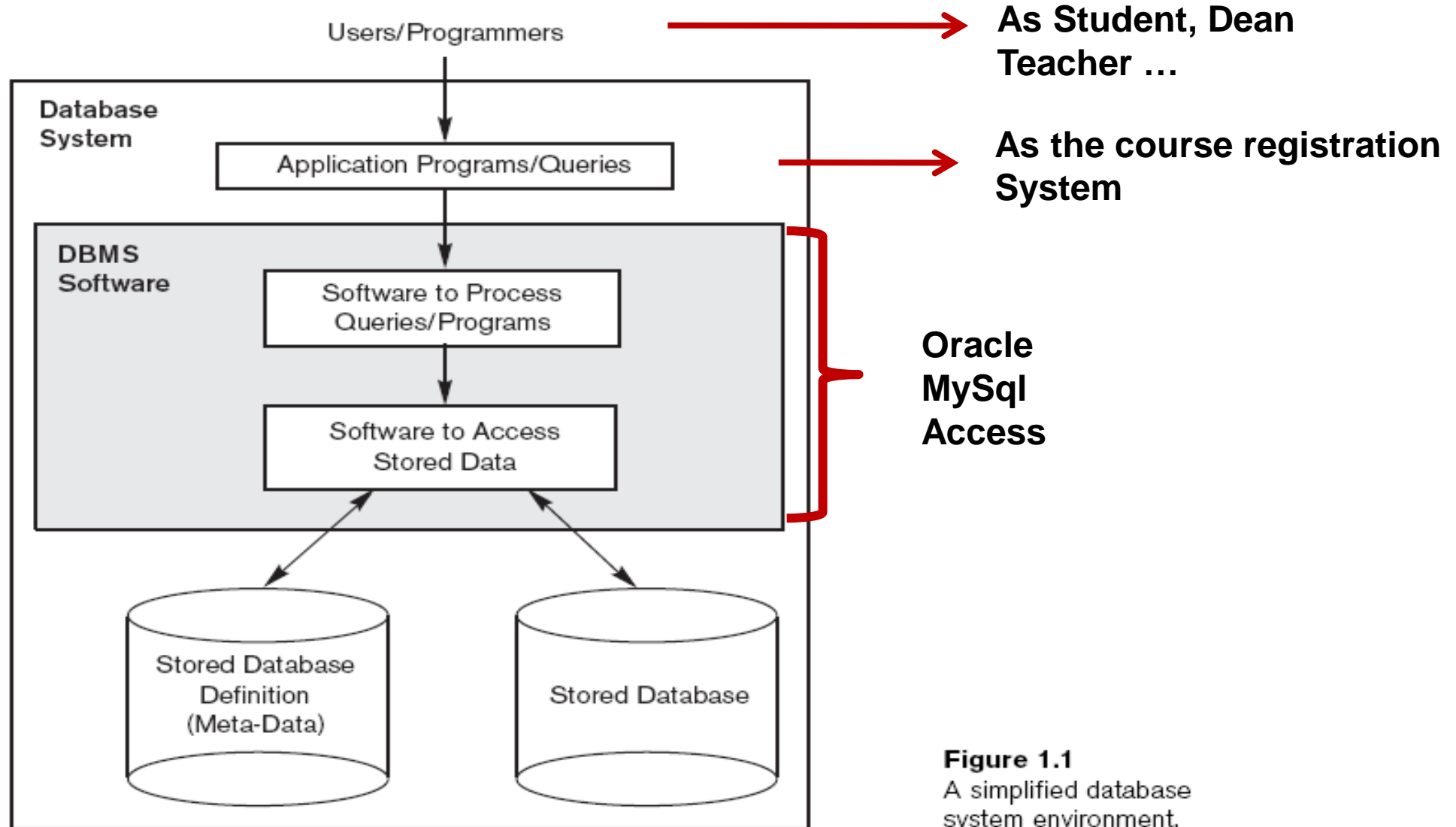
Database Advantages (Cont.)

- **Improved data integrity:** applying constraints to the data stored in the database to achieve data integrity become possible and easy
- **Support multiple view of the data:** A database typically has many users, each of whom may require a different perspective or **view** of the database. A view is a subset of the database
- **Improved security:** database allows the enforcement of different security measures that prevent unauthorized access to the data.
- **Program data independence**
- **More information from the same amount of data**
- **Improved backup and recovery services**

Database Languages

- There are two classes of languages:
 - **Procedural**: user specifies what data is required and how to get those data
 - **Declarative (nonprocedural)**: user specifies what data is required without specifying how to get those data
- **Data Definition Language (DDL)**: It is the language used to specify the structure of the database (tables, attributes, data types, constraints)
- **Data Manipulation Language (DML)**: It is the language used to manipulate data in the database (insert, update, delete, retrieve)
- **Query**: It is a statement requesting the retrieval of data

Database Environment



Database Disadvantages

- **Complexity:** Provisioning of such functionality makes the DBMS a complex piece of software. Failure to understand the system can lead to bad design decisions, which can have serious consequences
- **Size:** The complexity and breadth of functionality makes the DBMS an extremely large piece of software, occupying many megabytes of disk space and requiring substantial amounts of memory to run efficiently
- **Cost:** The cost of DBMSs varies significantly, depending on the environment and functionality provided
- **Additional hardware cost:** To achieve the required performance, it may be necessary to purchase a larger machine and additional storage space
- **Performance:** DBs are designed to store large amount of data and accessed by many different applications The effect is that some applications may not run as fast as they used to.

When Not to Use DB

- **More desirable to use regular files for:**
 - Simple, well-defined database applications not expected to change at all
 - Embedded systems with limited storage capacity
 - No multiple-user access to data

An Example

- **University Database**

- Information concerning students, courses, and grades in a university

- **Data Records**

- Student
- Course
- Section
- Grade report
- Prerequisite

An Example

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 |

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 |

An Example

- In order to create the previous database, specify data types, structure and constraints
- Store data in database
- Query (retrieve or update) data in database

Database Users

- **Database Administrator:** The DBA is responsible for authorizing access to the database, coordinating and monitoring its use, and acquiring software and hardware resources as needed. The DBA is accountable for problems such as security breaches and poor system response time
- **Database Designers:** are responsible for identifying the data to be stored in the database and for choosing appropriate structures to represent and store this data. These tasks are mostly undertaken before the database is actually implemented and populated with data.

Database Users

- **End Users:** are the people whose jobs require access to the database for querying, updating, and generating reports; the database primarily exists for their use. There are several categories of end users
- **Systems Analysts and Application Programmers:** system analysts determine requirements for end users and interact with database designers. Application programmers develop programs that interact with the database to satisfy end users' needs.

References:

- Fundamentals of Database Systems, 7th edition, Elmasri, Navathe
- Database_Systems_A_Practical_Approach_to_Design,_Implementation_and_Management_6th edition
- Good video link:
<https://www.youtube.com/watch?v=djEZeF4KTaM>