Lecture 1

Introduction to Data Base Management System

Dr. Vandana Kushwaha

Department of Computer Science

Institute of Science, BHU, Varanasi

CS107	Database Management Systems	L	T	P
		4	0	2

Introduction: Database Systems, View of Data Models, Database Languages, DBMS Architecture, Database Users and Data Independence.

ER Modeling: relation types, role and Structural Constraints, Extended ER Modeling Features, Design of an ER Database Schema, Reduction of ER Schema to Tables.

Relational Model: Relational Model Concepts, Relational Algebra.

Introduction to SQL: SQL data types and literals, Types of SQL commands, SQL operators, Tables, views and indexes, Queries and sub queries, Aggregate functions.

Relational Database Design: Functional and multi-valued Dependencies, Desirable Properties of Decomposition, Normalization up to 3 NF and BCNF.

Selected Database Issues: Security, Transaction Management, Introduction to Query Processing and Query Optimization, Concurrency Control, and Recovery Techniques.

Suggested Readings:

1. C. J. Date, An Introduction to Database Systems, Vol I & II, Addison Wesley.
2. A. Silberschatz, H. F. Korth, S. Sudarshan, Data Base System Concepts, McGraw Hill.
3. J. D. Ullman, Principles of Database Systems, Galgotia.
4. R. Elmasri, S. B. Navathe, Fundamentals of Database Systems, Pearson Education Asia.
5. R. Ramakrishnan, Database Management Systems, McGraw-Hill Education.

Introduction

- A **database** is a **collection** of **related data.**
- For **example**, consider the **names, telephone numbers**, and **addresses** of the **people** you know.
- This data can be recorded in an indexed address book or **stored** on a **hard drive,** using a **personal computer** and **software** such as **Microsoft Access** or **Excel.**
- A **database** can be of **any size** and **complexity.**
- A **database** of even **greater size** and **complexity** would be maintained by a **social media company** such as **Facebook**, which has more than a **billion users.**
- An example of a **large commercial database** is **Amazon.com.**

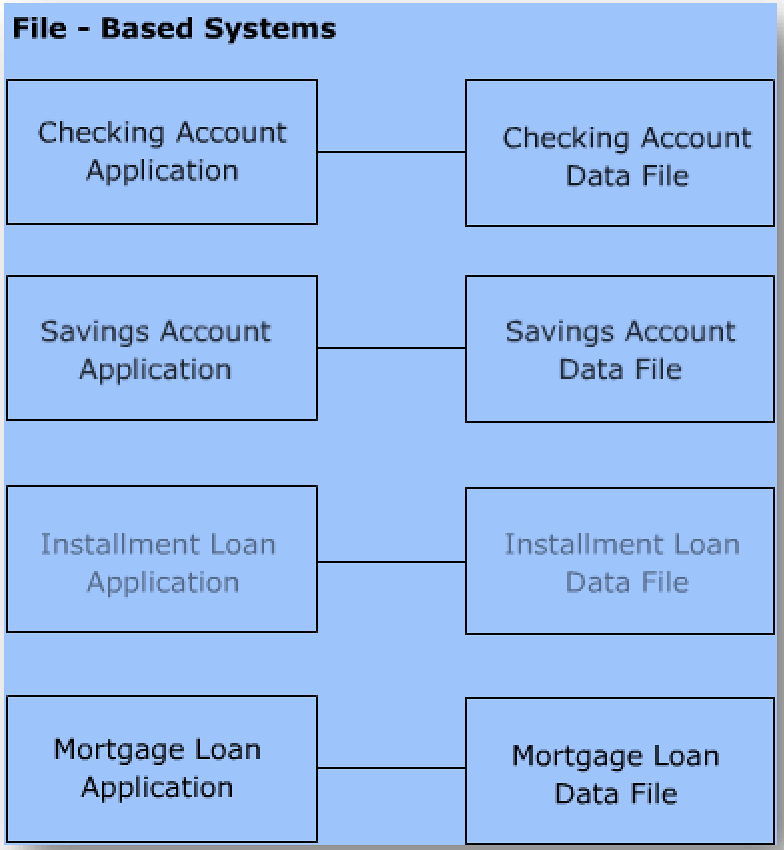
Introduction

- It contains **data** for over **60 million active users**, and **millions of books**, **CDs, videos, DVDs, games, electronics, apparel**, and other items.
- The **database** occupies over **42 terabytes** and is stored on hundreds of computers.
- **Millions of visitors** access **Amazon.com each day** and use the **database** to **make purchases.**
- The **database** is **continually updated** as **new books** and other **items** are **added** to the **inventory**, and **stock quantities** are **updated** as **purchases** are **transacted.**

File-processing system

- Typical **file-processing system** is supported by a **conventional operating system**.
- The system stores **permanent records** in various **files,** and it needs different **application programs** to **extract records** from, and **add records** to, the appropriate files.
- Before **database management systems (DBMSs)** were introduced, organizations usually stored information in such **File-processing system.**
- Consider a traditional **banking system** that uses the **file-based system** to **manage** the organization's **data.**
- There are **different departments** in the bank, each has its own applications that manage and manipulate different data files.
- For **banking systems**, the **programs** may be used to **debit or credit** an **account**, **find the balance** of an **account,** add a **new mortgage loan** and generate monthly statements.

File-processing system



Disadvantages of File-based approach

- **Data redundancy and inconsistency**
- The **same information** may be **duplicated in several places** (files).
- **Example:** Suppose one **College** is having different **departments** and each department is maintaining a **file based data base** of all the s**tudents** enrolled in that **department.**
- If a **student** has enrolled in **both** the **departments** say, **music** and **mathematics**.
- The **address** and **telephone number** of that **student** may appear in two different files one maintained by **Music department** and another by **Mathematics department**.
- This gives rise to **redundancy** in the **student's data.**
- This **redundancy** leads to **higher storage** and **access cost.**

Disadvantages of File-based approach

- In addition, it may **lead** to **data inconsistency.**
- For **example,** a changed student address may be reflected in the Music department records but not elsewhere in the system.

Disadvantages of file-based approach

- **Difficulty in accessing data**
- Suppose that one of the university clerks needs to find out the **names of all students** who **live** within a **particular postal-code area**.
- The clerk asks the **data-processing department** to generate such a list.
- Because the designers of the **original system** did not anticipate this request, there is **no application program** on hand to meet it.
- There is, however, an **application program** to **generate** the list of ***all*** **students.**
- The university clerk has now **two choices**: either obtain the list of all students and extract the needed information manually or **ask a programmer** to **write** the necessary **application program.**
- Both alternatives are obviously **unsatisfactory.**

Disadvantages of the file-based approach

- Suppose that such a program is written, and that, several days later, the same clerk needs to trim that list to include only those students who have taken at least 60 credit hours.
- As expected, a program to generate such a list does not exist.
- Again, the clerk has the preceding two options, neither of which is satisfactory.
- Conventional **file-processing environments** do **not allow needed data** to be retrieved in a **convenient and efficient manner.**

Disadvantages of File-based approach

- **Integrity problems**
- **Data values** must **satisfy** certain **consistency constraints** that are specified in the **application programs.**
- It is **difficult to make changes** to the **application programs** in order to enforce **new constraints.**
- Suppose the university maintains an account for each department, and records the balance amount in each account.
- Suppose also that the university requires that the account balance of a department may never fall below zero.
- Developers enforce these constraints in the system by adding appropriate code in the various application programs.
- However, when new constraints are added, it is difficult to change the programs to enforce them.

Disadvantages of File-based approach

- **Atomicity problems**
- A computer system, like any other device, is subject to **failure**.
- In many applications, it is **crucial** that, if a **failure occurs**, the **data** be **restored** to the **consistent state** that existed **prior to the failure**.
- Consider a program to **transfer $500** from the account balance of **department *A* to** the account balance of **department *B*.**
- If a **system failure** occurs during the execution of the program, it is possible that the $500 was removed from the balance of department *A* but was not credited to the balance of department *B*, resulting in an **inconsistent database state.**
- Clearly, it is essential to **database consistency** that ***either both the credit and debit occur, or that neither occur.***
- That is, the **funds transfer** must be **atomic**—it must happen in its entirety or not at all.
- It is **difficult to ensure atomicity** in a conventional **file-processing system.**

Disadvantages of File-based approach

- **Concurrent-access anomalies**.
- Many systems allow **multiple users** to **update** the **data simultaneously.**
- In such an environment, interaction of **concurrent updates** is possible and may result in **inconsistent data**.
- Consider **department *A*,** with an **account balance** of **$10,000.**
- If **two department clerks debit** the **account balance** (by say $500 and $100, respectively) of **department *A*** at almost **exactly the same time**, the result of the **concurrent executions** may leave the budget in an **incorrect (or inconsistent) state.**
- To **guard** against this possibility, the **system** must maintain some form of **supervision**.
- But supervision is difficult to provide because data may be accessed by many **different application programs** that have **not been coordinated** previously.

Disadvantages of the file-based approach

- **Security problems**
- Not every **user** of the **database system** should be **allowed to access all** the **data.**
- For **example**, in a **university, payroll personnel** need to **see only** that part of the database that has **financial information.**
- They **do not need access** to information about **academic records.**
- But, since **data** is stored in a **redundant** manner enforcing such **security constraints** is **difficult.**

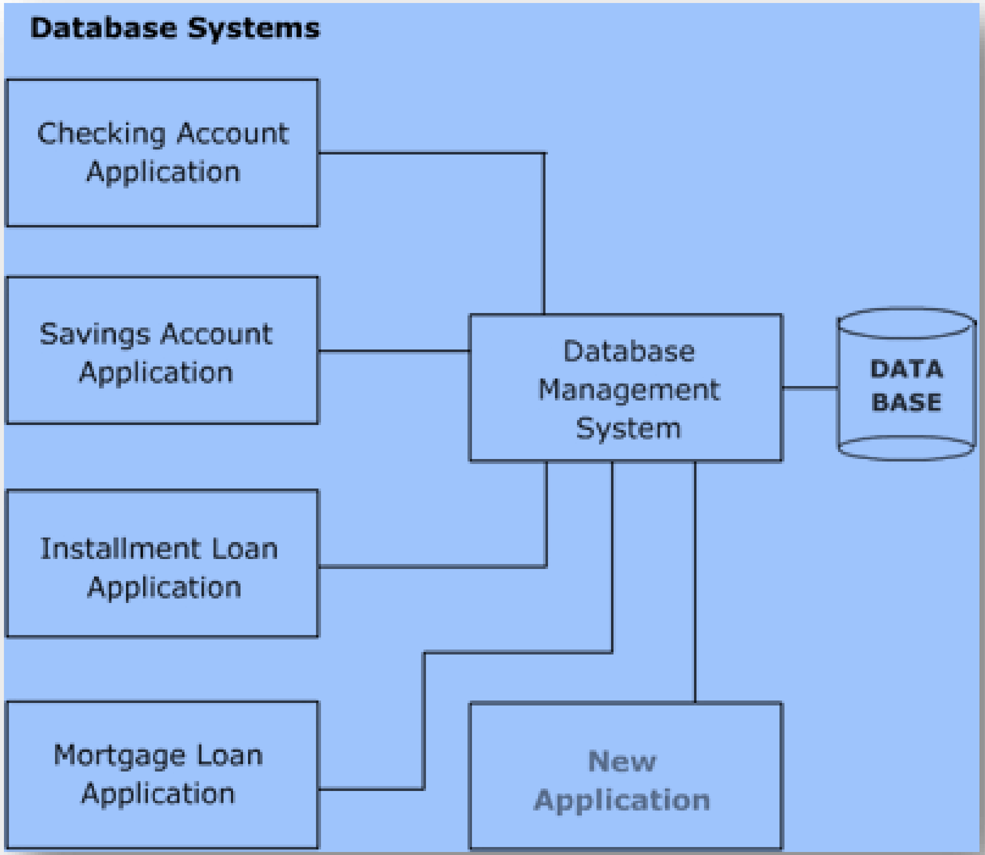
Disadvantages of the file-based approach

- Data Redundancy and Inconsistency
- Difficulty in accessing data
- Integrity problems
- Atomicity problems
- Concurrent-access anomalies
- Security problems

Data Base Management System

- A **Database-management system (DBMS)** is a **collection of interrelated data** and a **set of programs** to **access** those **data.**
- The **collection of data**, usually referred to as the **database**, contains information relevant to an enterprise.
- The **primary goal** of a **DBMS** is to provide a way to **store** and **retrieve database information** that is both ***convenient*** and ***efficient*****.**
- In addition, the **database system** must ensure the **safety** of the **information stored**, despite **system crashes** or attempts at **unauthorized access.**
- If **data** are to be **shared among several users**, the **system** must **avoid** possible **anomalous results.**

Data Base Management System



Advantages of Using the DBMS

- **Minimal Data Redundancy**
- Since the **whole data** resides in **one central database**, the **various programs** in the **application** can **access data .**
- This **reduces data redundancy.**
- However, this does not mean all redundancy can be eliminated.
- There could be business or technical reasons for having **some amount of redundancy.**
- Any such redundancy should be carefully controlled and the DBMS should be aware of it.
- **Data Consistency**
- Reduced **data redundancy** leads to better **data consistency.**

Advantages of Using the DBMS

- **Improved data sharing & Improved data security**
- The **DBMS** helps to create an environment in which **end users** have **better access** to **more** and **better-managed data**.
- The more users access the data, the greater the **risks** of **data security** breaches.
- Corporations invest considerable amounts of time, effort, and money to ensure that corporate data are used properly.
- A **DBMS provides a framework** for better enforcement of **data privacy** and **security policies.**

Advantages of Using the DBMS

- **Support for Multiple Views of data**
- A **database** supports **multiple views** of **data.**
- A ***view*** is a **subset** of the **database**, which is defined and dedicated for **particular users** of the system.
- **Multiple users** in the **system** might have **different views** of the **system.**
- Each **view** might contain only the **data of interest** to a user or group of users.
- **Student(enrolment_no, name, age, address, course, email_id, marks)**
- **View1(enrolment_no, name, age, course, email_id)**
- **View2(enrolment_no, name, course, marks)**

Advantages of Using the DBMS

- **Enforcement of integrity constraints**
- **DBMS** must provide the **ability to define** and **enforce certain constraints** to ensure that users **enter valid information** and maintain **data integrity.**
- A **database constraint** is a **restriction** or **rule** that dictates **what can be entered** or edited in a **table** such as a **postal code** using a **certain format** .
- There are **different types** of **database constraints**.
- ***Data type***, for example, determines the **sort of data permitted** in a field, for **example** numbers only.
- ***Data uniqueness*** such as the **primary key** ensures that **no duplicates** are entered.

Advantages of Using the DBMS

- **Backup and recovery facilities**
- **Backup** and **Recovery** are **methods** that allow you to **protect your data** from **loss.**
- In a **file-based computer system**, the user has to **create a backup** of the data **regularly** to **protect** it from being **damaged or lost** in the event of **system crash** or **failure.**
- This can be a **very time-consuming process**, and is **prone** to **human error.**
- Most of the **DBMS** have a **Backup-and-Recovery feature built within** them, that **automatically backs-up** all important data, and **restores** it when needed.

Advantages of Using the DBMS

- **Data Independence**
- The separation of **data** from the **application program** used to access it is known as **data independence.**
- Typically, in a DBMS, the **database** and the **application program** are maintained **separately** from each other, with the **DBMS** acting as a **mediator** between them.
- This proves to be a **big advantage**, as one can easily **change** the **database structure** **without affecting** the **application program.**
- **Data Abstraction**
- **Data abstraction** allows the **DBMS** to provide an **abstract view** of the **data**, without revealing the details of its **physical storage** or **method of implementation**.

Advantages of Using the DBMS

- **Ease of Application Development**
- Many **data-related issues**, like **concurrent access, security, data integrity**, etc., are taken care of by the **DBMS.**
- Therefore, when an **application programmer** develops a **program,** he can **focus explicitly** on the **needs** of the **users.**
- This makes the **task of application development** much **easier.**

- **Examples of DBMS Software:**
  - Oracle
  - Microsoft's SQL Server
  - MySQL

Applications of DBMS