DATABASE MANAGEMENT SYSTEMS: OUTLINE

What Is A Database Management System (DBMS)?

Why Do We Need Database Management Systems?

How Does A DBMS Manage Data?

What Are The Major Components Of A DBMS?
The problem we want to address

How best to manage stored data?

organize, access, share, protect, ...
A database is a collection of related data within a specific business process or problem setting.

Data are facts such as age, salary, name, address, etc.

- A database has the following properties.
  - It is designed, built and populated with data for a specific purpose.
    Applications: sales, human resources, manufacturing, banking, real estate, stock trading, inventory management, social media, ride sharing, …
  - It usually represents some aspect of the real world.
  - The data have some inherent meaning.

☞ Databases touch all aspects of our lives!
A **database management system (DBMS)** is a general-purpose software system used to manage databases.

- A DBMS provides support/facilities for:
  - defining what data to store (types, structures, constraints)
  - storing and managing data on a storage device
  - manipulating data (querying, updating)
  - sharing data among many users
  - protecting data from loss, corruption, unauthorized access

<table>
<thead>
<tr>
<th>Company</th>
<th>Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle</td>
<td>Oracle Database</td>
</tr>
<tr>
<td>IBM</td>
<td>DB2</td>
</tr>
<tr>
<td>Microsoft</td>
<td>Access, SQL Server</td>
</tr>
<tr>
<td>Sybase</td>
<td>Adaptive Server</td>
</tr>
<tr>
<td>Informix</td>
<td>Dynamic Server</td>
</tr>
</tbody>
</table>

A DBMS provides an environment for managing data that is both convenient and efficient to use.
What Is A Database Management System (DBMS)?

Why Do We Need Database Management Systems?

How Does A DBMS Manage Data

What Are The Major Components Of A DBMS?
Applications access stored data using the facilities provided by an *operating system file system*.

**Drawbacks**
- Data duplication and inconsistency
- Difficulty meeting unanticipated needs
- Data isolation
- Data integrity problems
- Atomicity of updates
- Concurrent access
- Security problems
A DBMS provides automated solutions for the data management problems encountered when using file systems.

### DATABASE APPROACH TO MANAGING DATA

**Major Principles**

- **integrates** an organization’s data.
- **separates** meta-data (description of data) and data.
- **supports** multiple views of data.
- **controls** definition and access of data **centrally**.

Applications access stored data using the facilities provided by a **DBMS**.

---

A DBMS provides automated solutions for the data management problems encountered when using file systems.
What Is A Database Management System (DBMS)?

Why Do We Need Database Management Systems?

How Does A DBMS Manage Data?

What Are The Major Components Of A DBMS?
A **data model** is a set of concepts for describing data that defines
– properties
– relationships
– semantics
– constraints

A data model is the **fundamental mechanism** used by a DBMS to **logically** describe and organize data and consists of:

1. **data structure types** $\implies$ specify logical organization (properties, relationships and semantics)

2. **integrity constraints** $\implies$ specify constraints (restrictions on properties and relationships)

3. **operations** $\implies$ specify how data is accessed (e.g., R,I,U,D—Read, Insert, Update, Delete)

A data model is used to **describe and organize** data as well as to state any restrictions on the data.
**DATA MODELS: EXAMPLE**

**Entity-Relationship (E-R) model**

<table>
<thead>
<tr>
<th>Student</th>
<th>EnrollsIn</th>
<th>Course</th>
</tr>
</thead>
<tbody>
<tr>
<td>studentId</td>
<td></td>
<td>code</td>
</tr>
<tr>
<td>name</td>
<td>grade</td>
<td>description</td>
</tr>
<tr>
<td>address</td>
<td></td>
<td>credits</td>
</tr>
<tr>
<td>admitYear</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

☞ Users view data as entities and explicit relationships among entities.

**Relational model**

<table>
<thead>
<tr>
<th>Student</th>
<th>EnrollsIn</th>
<th>Course</th>
</tr>
</thead>
<tbody>
<tr>
<td>studentId</td>
<td>code</td>
<td>code</td>
</tr>
<tr>
<td>name</td>
<td>code</td>
<td>description</td>
</tr>
<tr>
<td>address</td>
<td>code</td>
<td>credits</td>
</tr>
<tr>
<td>admitYear</td>
<td>code</td>
<td></td>
</tr>
</tbody>
</table>

☞ Users view data as tables and implicit relationships among tables.
LEVELS OF ABSTRACTION

- One big problem in application development is the separation of application programs (i.e., code) from the data that they access.

- Do I have to change my application program when I …
  - replace my hard drive?
  - partition the data into two physical files (or merge two physical files into one)?
  - store salary as a floating-point number instead of an integer?
  - develop other applications that use the same data?
  - add more data fields to support other applications?
  - index the data using a B⁺-tree instead of a hash index?

A DBMS provides separation of application programs and data via several levels of abstraction.
Divide the student records into 3 partitions and store them on disks 1, 2 and 3.

Type student = record
  id: string;
  name: string;
  address: string;
  admitYear: integer;
  birthdate: date;
  cga: decimal;
end;

Files on disks

HKUST student data

Logical level

Logical view

CSE Dept view

ARR view

Finance Office view

View level

Physical level

Physical view

LEVELS OF ABSTRACTION (cont’d)
SCHEMAS AND INSTANCES

- A database **schema** describes the overall design of a database according to a data model.
  - Stored in the system catalog; changes infrequently, if at all.

- A DBMS uses several schemas, one for each level of abstraction, which **describes the data** at the corresponding level.
  - A **view** (subschema) describes the data that a user can access.
  - A **logical schema** describes the logical structure of the database (e.g., the set of students, courses and the relationship between them).
  - A **physical schema** describes the file formats and locations where the data are stored on disk.

- A database **instance** refers to the actual content of the database at a particular point in time.
  - Conforms to its corresponding schema; changes frequently as data are changed.
Data independence is the ability to modify a schema definition in one level of abstraction without affecting a schema definition in a higher level.

**Logical data independence** (shields users from changes in the *logical structure*).

**Physical data independence** (shields users from changes in the *physical structure*).
DATA INDEPENDENCE: EXAMPLE

A program accessing the data via the OS file system must know:
• first 4 bytes is an id (a number)
• next 10 bytes is a name (a string)

Data on disk

| 1129 | John Law | … |

Schema

Student (id integer, name char(10),…)

A program accessing the data via a schema does not need to know its physical organization; it simply requests the student’s id, name, etc.

Data on disk

| 1129 | John Law | … |
What Is A Database Management System (DBMS)?

Why Do We Need Database Management Systems?

How Does A DBMS Manage Data?

What Are The Major Components Of A DBMS?
DBMS ARCHITECTURE

- Naïve users (tellers, web users)
  - Use
- Application programmers
  - Write
- Sophisticated users (analysts)
  - Use
- Database administrators
  - Use

- Application interfaces
- Application programs
- Query tools
- Administration tools

- Compiler and linker
- DML queries
  - DML compiler and organizer
  - DDL interpreter

- Query evaluation engine

- Storage manager
- Buffer manager
- File manager
- Authorization and integrity manager
- Transaction manager

- Query processor

- Disk storage
- Data
- Indices
- System catalog (data dictionary)
- Statistical data
The storage manager provides the interface between the low-level data stored in the database and the application programs and queries submitted to the system.
The **query processor** translates nonprocedural queries and updates into efficient physical disk-level operations.
**DATABASE MANAGEMENT SYSTEMS: SUMMARY**

- **Database management systems (DBMSs)** address the limitations of OS file systems for managing an enterprise’s data.

- **Data models** are the foundation for developing a database—the entity-relationship (E-R) model and relational model are commonly used in practice.

- **Data independence** is fundamental to understanding how a DBMS manages data at different abstraction levels.

- **A DBMS provides many facilities** for query processing and storage management to efficiently handle the data management and data access needs of various users.
DSAA 5012: SYLLABUS

- Database Management Systems

Entity-Relationship (E-R) Model and Database Design
  - Relational Algebra
  - Structured Query Language (SQL)
  - Relational Database Design
  - Storage and File Structure

- Indexing
- Query Processing
- Query Optimization
- Transactions
- Concurrency Control
- Recovery System
- Graph Databases
- Uncertain Databases