DBMS – An Introduction

Overview

- An Introduction
- What is DBMS?
- File systems
- Purpose of Database System
- Characteristics of Database
- View of Data
  - Data Abstraction
  - Instances and Schemas
- Data Models
- Database Languages

An Introduction

- A file is a collection or set (ordered or unordered) of data elements stored on a storage media.
- File organization:
  - Method of arranging a file of records on external storage.
  - Main objective is to reduce the file retrieval time (to reduce I/O time)
- Heap – a record can be placed anywhere in the file where there is space.
- Sequential – store records in sequential order, one after the other record.
- Hashing – a hash function computed on some attribute of each record; the result specifies in which block of the file the record should be placed.

An Introduction

- Indexed Files :
  - Record id (rid) is sufficient to physically locate record.
  - Indexes are data structures (Trees and Hashing) that allow us to find the record ids of records with given values in index search key fields.
- Direct (Access) Files : Access files without any indexes. Records of each relation may be stored in a separate file.
- In a multi-table clustering file organization, records of several different relations (applications) can be stored in the same file.
- Motivation: store related records on the same block to minimize I/O.

An Introduction

- Data – means known facts that can be recorded and that have implicit meaning
- Database – collection of related data with an implicit meaning
- Characteristics of database:
  - A database represents some aspects of the real world.
  - A database is a logically coherent collection of data with some inherent meaning.
  - A database is designed, built, and populated with data for a specific purpose.

What is DBMS?

- DBMS-
  - is a collection of programs that enables users to create and maintain the database.
  - A *general-purpose software system* that facilitates the processes of defining, constructing, and manipulating databases for various applications.

What is DBMS? (Cont.)

- *Defining a database* – involves specifying the data types, structures and constraints for the data to be stored in the database.
- *Constructing a database* – process of storing the data itself on storage medium that is controlled by DBMS.
- *Manipulating a database* – includes functions such as querying the DB for retrieval, updating the database to reflect changes in the miniworld and generating the reports, etc.,

File Systems

- Prior to DBMS, file system provided by OS was used to store information.
- In a file-based system, collection of application programs perform services for the end users.
- Each program defines and manages its own data.

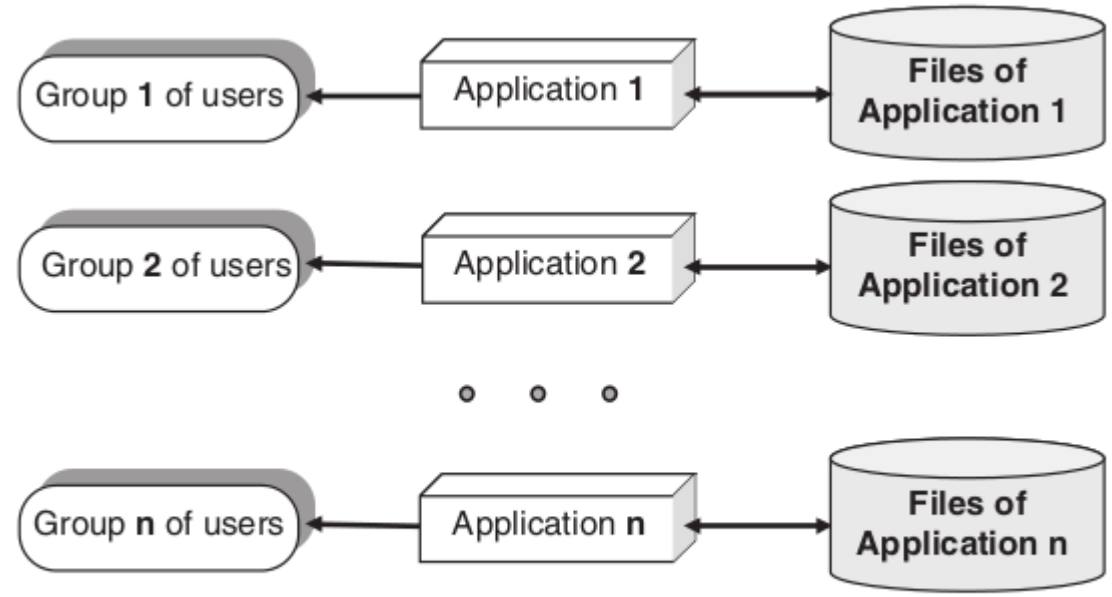


Fig. 1.4. File-based System

Purpose of Database Systems

- Earlier, database applications were built 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
    - Higher storage cost, data inconsistency
  - 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 part of program code
    - Hard to add new constraints or change existing ones

Purpose of Database Systems

- Drawbacks of using file systems (cont.)
  - Atomicity of updates
    - Failures may leave database in an *inconsistent state* with partial updates carried out
    - E.g. 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
      - » E.g. two people reading a balance and updating it at the same time
  - Security problems
  - Database systems offer solutions to all the above problems

Characteristics of Database

- 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.

Characteristics of Database

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.

Characteristics of Database

- **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.

Characteristics of Database

- **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.

Levels of Abstraction

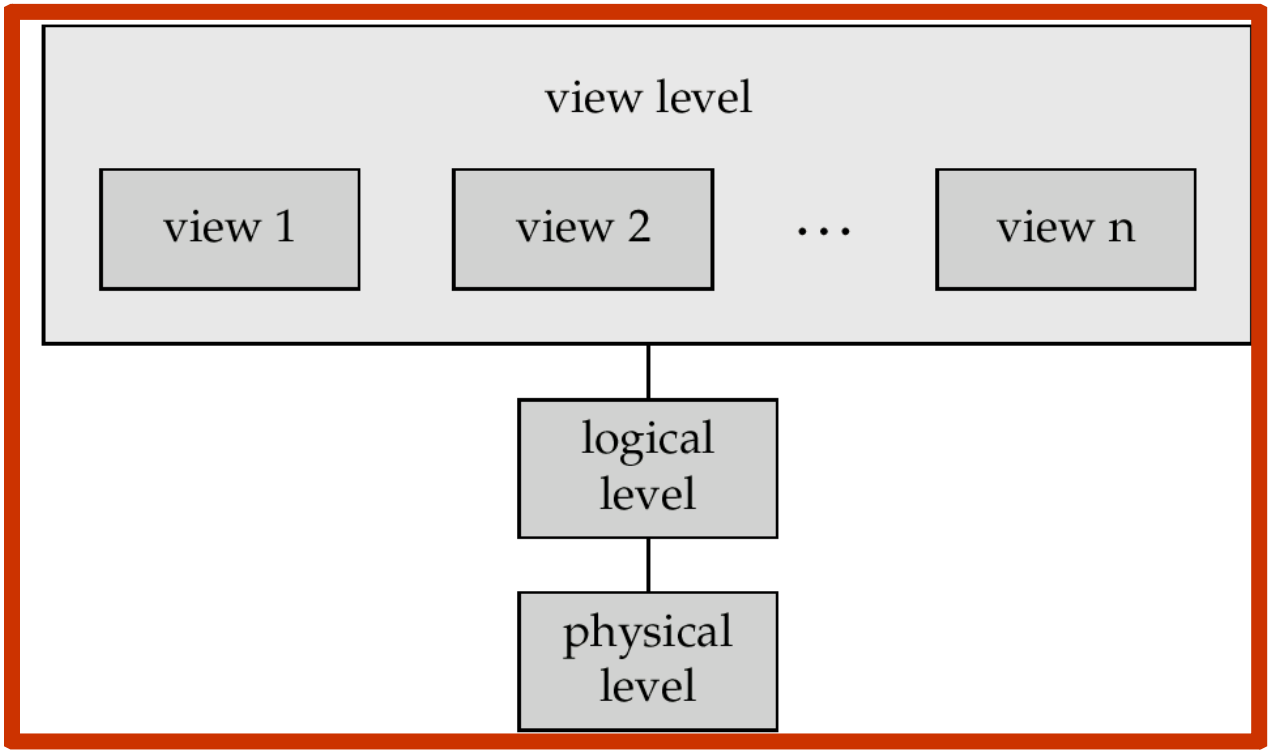
- Database purpose is to provide users with an *abstract* view of the data – the system hides certain details
- Physical level: describes *how* a record (e.g., customer) is stored.
- Logical level: describes data stored in database, and the relationships among the data.

```
type customer = record
    name : string;
    street : string;
    city : integer;
end;
```

- View level: application programs hide details of data types. Views can also hide information (e.g., salary) for security purposes.

View of Data

Three levels of data abstraction



Instances and Schema

- Similar to types and variables in programming languages
- **Schema** – the logical structure of the database
    - e.g., the database consists of information about a set of customers and accounts and the relationship between them
    - Analogous to type information of a variable in a program
    - **Physical schema**: database design at the physical level
    - **Logical schema**: database design at the logical level
  - **Instance** – the actual content of the database at a particular point in time
    - Analogous to the value of a variable

Data Independence

- **Physical Data Independence** – the ability to modify the physical schema without changing the logical (or view) schema.
  - files reorganized to improve the performance of retrieval/update
- **Logical Data Independence** – ability to modify the logical schema without changing the view schema or application programs.
  - expand the database or reduce the database
  - changes to the constraints

Data Models

- A collection of tools for describing:
  - data
  - data relationships
  - data semantics
  - data constraints
- Data Models can be classified into four categories:
- Relational Model
  - Uses a collection of tables to represent both data and the relationships among those data.
  - Each table has multiple columns, and each column has unique name
  - Widely-used in commercial DBMS

Data Models

- Entity-Relationship Model
  - Based on collection of objects, called entities and relationships among these objects.
  - An entity is a thing or object in the real world that is distinguishable.
- Object-Based Data Model
  - Extending the E-R model with notions of encapsulation, methods (functions) and object identity.
  - Object-relational data model combines features of OO Data model and relational model.
- Semistructured Data Model
  - eXtensible Markup Language (XML) is widely used to represent semistructured data.
- Legacy data models like network and hierarchical data model – widely used in the past.

Database Languages

- Data-definition language
  - To specify the database schema
- Data-manipulation language
  - To express database queries and updates

Data-definition Language

- Specify a database schema by a set of definitions.
- Storage definition language specifies the storage structure and access methods used by the database system.
- The data values stored in the database must satisfy certain consistency constraints.
- Database systems check these constraints every time the database is updated.
- Domain Constraints:
  - Every attribute must be associated with a domain of possible values
  - Ex: integer, character, date/time
  - Elementary form of integrity constraint
  - Tested when a new data is entered into the database

Data-definition Language (Cont.)

- Referential Integrity:
  - Ensure that a value appears in one relation for a given set of attributes also appears in another relation.
  - Modification is allowed only if it does not violate referential integrity
- Assertions:
  - An assertion is any condition that the database must always satisfy
  - Many constraints can not expressed by using only special forms
- Authorization:
  - To differentiate among the users with the type of access they are permitted on various data values in the database
  - Read/insert/update/delete authorization

Data-manipulation Language

- A language that enables users to access or manipulate data:
  - Retrieval of data
  - Insertion of new data
  - Deletion of data
  - Modification of existing data
- Two classes of languages:
  - Procedural – user specifies what data is required and how to get those data
  - Nonprocedural – user specifies what data is required without specifying how to get those data
- SQL is the most widely used query language

References

- Fundamentals of Database Systems
  Ramez Elmasri, Shamkant Navathe, 5th Edition
- Database System Concepts
  Abraham Silberschatz, Korth, Sudarshan, 5th Edition


THANK
YOU!