Database Management System

Instructor

Dr. Mahammad Humayoo
PhD in Computer Science and Technology, UCAS

Senior Lecturer,
Khwopa Engineering College
Email: mahammad.humayoo@khec.edu.np

Mobile: 9844059625

Chapter 2

Database System Concepts and Architecture

Outlines

- Data Models, Schemas, and Instances
- Three-Schema Architecture and Data Independence
- Database Languages and Interfaces
- The Database System Environment
- Centralized and Client/Server Architectures for DBMSs
- Classification of Database Management Systems

Scanned with CamScanner

Data Models

- **Data Model**: A set of concepts to describe the structure of a database, and certain constraints that the database should obey.
  - Help to achieve data abstraction
  - By structure of a database we mean the data types, relationships, and constraints that apply to the data
- **Data Model Operations:** Operations for specifying database retrievals and updates by referring to the concepts of the data model. Operations on the data model may include basic operations and user-defined operations.
- **Categories of data models**
- **Conceptual (high-level, semantic) data models:** Provide concepts that are close to the way many users perceive (see) data. (Also called entity-based or object-based data models.)
  - Use concepts such as entities, attributes and relationships
  - Entities -> represent real world object or concept. For example: Employee, Student etc.
  - Attributes -> Further describe an entity. For Example: student name, rollnol etc.
  - Relationships -> association among 2 or more entities. For Example: Relationship among student and course.
- **Physical (low-level, internal) data models:** Provide concepts that describe details of how data is stored in the computer.
  - By representing information like format and access path.
  - Access path -> structure for efficient searching of database records. Ex: indexing or hashing

1A-4

Data Models

- **Implementation (representational) data models:** Provide concepts that fall between the above two, balancing user views with some computer storage details.
- Representation data models are three types
  - Relational data model
  - Hierarchical models
  - Network models
- Representational data models represent data by using record structures and hence are sometimes called record-based data models.

1A-5

Terminologies

- **Database Schema:** Description of a database.
  - DB schema is specified during database design and is not expected to change frequently
  - Includes descriptions of the database structure and the constraints that should hold on the database
- **Schema Diagram:** Displayed schema

Student				
Name	Roll No	Class	Major	Address

- **Schema Construct:** Each object within the schema.
  - Ex: Student, Course, Section
- **Database State (or instance or snapshot):** The data in the database at a particular moment. Also called database occurrence.
- The actual data in a database may change quite frequently. For example, University database changes every time we add a new student or enter a new grade
- **schema evolution:** the schema is not supposed to change frequently,
- it is not uncommon that changes occasionally need to be applied to the schema as the application requirements change. For example, we may decide that another data item needs to be stored for each record in a file, such as adding the Date_of_birth to the STUDENT schema in University DB. This is known as schema evolution.

Terminologies

- Example:
  - Database schema: *instructor (ID, name, dept_name, salary)*
  - Database Instance:

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	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

1A-7

Scanned with CamScanner

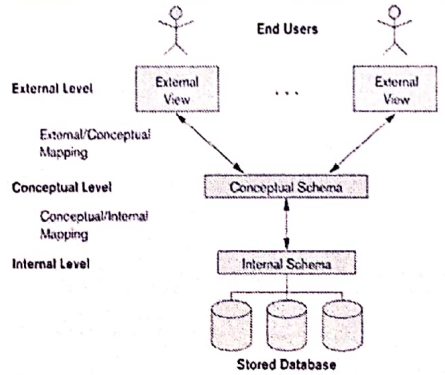
Database Schema Vs. Database State

- **Database State:** Refers to the content of a database at a moment in time.
- **Initial Database State:** Refers to the database when it is loaded
- **Valid State:** A state that satisfies the structure and constraints of the database.
- **Distinction**
- The database schema changes very infrequently. The database state changes every time the database is updated.
- Schema is also called intension, whereas state is called extension.

Three-Schema Architecture

- **Goal of three-schema architecture is to separate the user applications from the physical database**
- **Proposed to support DBMS characteristics of:**
  - Program-data independence.
  - Support of multiple views of the data.
- **Defines DBMS schemas at three levels:**
- **Internal schema:** at the internal level to describe physical storage structures and access paths of database. Typically uses a physical data model.
- **Conceptual schema:** at the conceptual level to describe the structure and constraints for the whole database for a community of users. Uses a conceptual or an implementation data model.
  - Hides the details of the physical storage structure and concentrates on describing entities, data types, relationships, constraints etc.
- **External schemas:** at the external level to describe the various user views. Usually uses the same data model as the conceptual level.
  - Describes the part of the database that a user is interested in and hides the rest of the database from the user group
- **Mappings:** The processes of transforming requests and results between levels are called mappings

Three-Schema Architecture



Data Independence

- **Data Independence:** The capacity to change the schema at one level of a database system without having to change the schema at the next higher level
- **Two types of data independence:**
- **Logical Data Independence:** The capacity to change the conceptual schema without having to change the external schemas and their application programs.
- **Physical Data Independence:** The capacity to change the internal schema without having to change the conceptual schema.
  - Changes may be need to improve performance. For example: changing access path (indexing), Changing file location

When a schema at a lower level is changed, only the mappings between this schema and higher-level schemas need to be changed in a DBMS that fully supports data independence. The higher-level schemas themselves are unchanged. Hence, the application programs need not be changed since they refer to the external schemas.

Scanned with CamScanner

DBMS Languages

Data Definition Language (DDL): Used by the DBA and database designers to specify the conceptual schema of a database. In many DBMSs, the DDL is also used to define internal and external schemas (views). In some DBMSs, separate storage definition language (SDL) and view definition language (VDL) are used to define internal and external schemas.

- In short, DDL allows you to create SQL statements to make operations with database data structures (schemas, tables etc.).
- **DDL Command List:** CREATE, DROP, ALTER, RENAME, TRUNCATE
- **Data Manipulation Language (DML):** Used to specify database retrievals and updates.
  - DML commands (data sublanguage) can be embedded in a general-purpose programming language (host language), such as COBOL, C or java. (low level language)
  - Alternatively, stand-alone DML commands can be applied directly (query language). (High-level language)
- **DML Command List:** SELECT, INSERT, UPDATE, DELETE
- In short, DML used to build SQL queries to manipulate (select, insert, update, delete etc.) data in the database.

DBMS Languages

- **Types of DML:**
- **High Level or Non-procedural Languages:** e.g., SQL, are set-oriented and specify what data to retrieve than how to retrieve. Also called declarative languages.
- **Low Level or Procedural Languages:** record-at-a-time; they specify how to retrieve data and include constructs such as looping.
- **Data Control Language (DCL):** its commands are responsible for access restrictions inside of the database.
- **DCL Command List:** Grant, Revoke, Deny etc.

DBMS Interfaces

- Stand-alone query language interfaces.
- Programmer interfaces for embedding DML in programming languages:
  - Pre-compiler Approach
  - Procedure (Subroutine) Call Approach
- User-friendly interfaces:
  - **Menu-based, popular for browsing on the web:** Pull-down menus are a very popular technique in Web-based user interfaces.
  - **Forms-based, designed for naïve users :** A forms-based interface displays a form to each user. Users can fill out all of the form entries to insert new data, or they can fill out only certain entries, in which case the DBMS will retrieve matching data for the remaining entries.
  - **Graphics-based (Point and Click, Drag and Drop etc.):** A Graphical User Interfaces (GUI) typically displays a schema to the user in diagrammatic form. The user then can specify a query by manipulating the diagram. In many cases, GUIs utilize both menus and forms.
    - Uses a pointing device to pick certain parts of the displayed diagram
  - **Natural language:** These interfaces accept requests written in English or some other language and attempt to understand them.
    - A natural language interface usually has its own schema, which is similar to the database conceptual schema, as well as a dictionary of important words
    - The natural language interface refers to the words in its schema, as well as to the set of standard words in its dictionary, that are used to interpret the request
    - If interpretation is successful, the interface generates a high-level query corresponding to the natural language request and submits it to the DBMS for processing
    - If interpretation is failed, a dialogue is started with the user to clarify the request

DBMS Interfaces

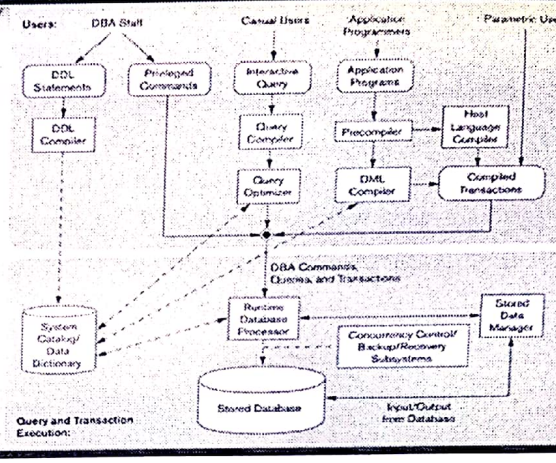
- **Interfaces for Parametric Users:** Parametric users or naive users such as bank tellers, airline booking agent etc. often have small set of operations that they must perform repeatedly
  - Systems analysts and programmers design and implement a special interface for each known class of naive users.
  - Usually a small set of abbreviated commands is included, with the goal of minimizing the number of keystrokes required for each request.
- **Interfaces for the DBA:** Most database systems contain privileged commands that can be used only by the DBA staff.
  - These include commands for creating accounts, setting system parameters, granting account authorization, changing a schema, and reorganizing the storage structures of a database.
- **Apps for Mobile Devices:** These interfaces present mobile users with access to their data.
  - For example, banking, reservations, and insurance companies, among many others
  - provide apps that allow users to access their data through a mobile phone or mobile device.
- **Keyword-based Database Search:** These are somewhat similar to Web search engines, which accept strings of natural language (like English or Spanish) words and match them with documents at specific sites (for local search engines) or Web pages on the Web at large (for engines like Google or Ask).
- **Speech Input and Output:** Limited use of speech as an input query and speech as an answer to a question or result of a request is becoming commonplace
  - Applications with limited vocabularies, such as inquiries for telephone directory, flight arrival/departure, and credit card account information, are allowing speech for input and output to enable customers to access this information.

Scanned with CamScanner

Database System Environment

- A DBMS is a complex software system. Below we discuss types of software components that constitute a DBMS and the types of computer system software with which the DBMS interacts
- **DBMS Component Modules**: Below figure illustrates, in a simplified form, the typical DBMS components.
- The figure is divided into two parts.
  - The top part of the figure refers to the various users of the database environment and their interfaces.
  - The lower part shows the internal modules of the DBMS responsible for storage of data and processing of transactions
- Below figure shows DBMS component modules and their interaction.

Database System Environment



Database System Environment

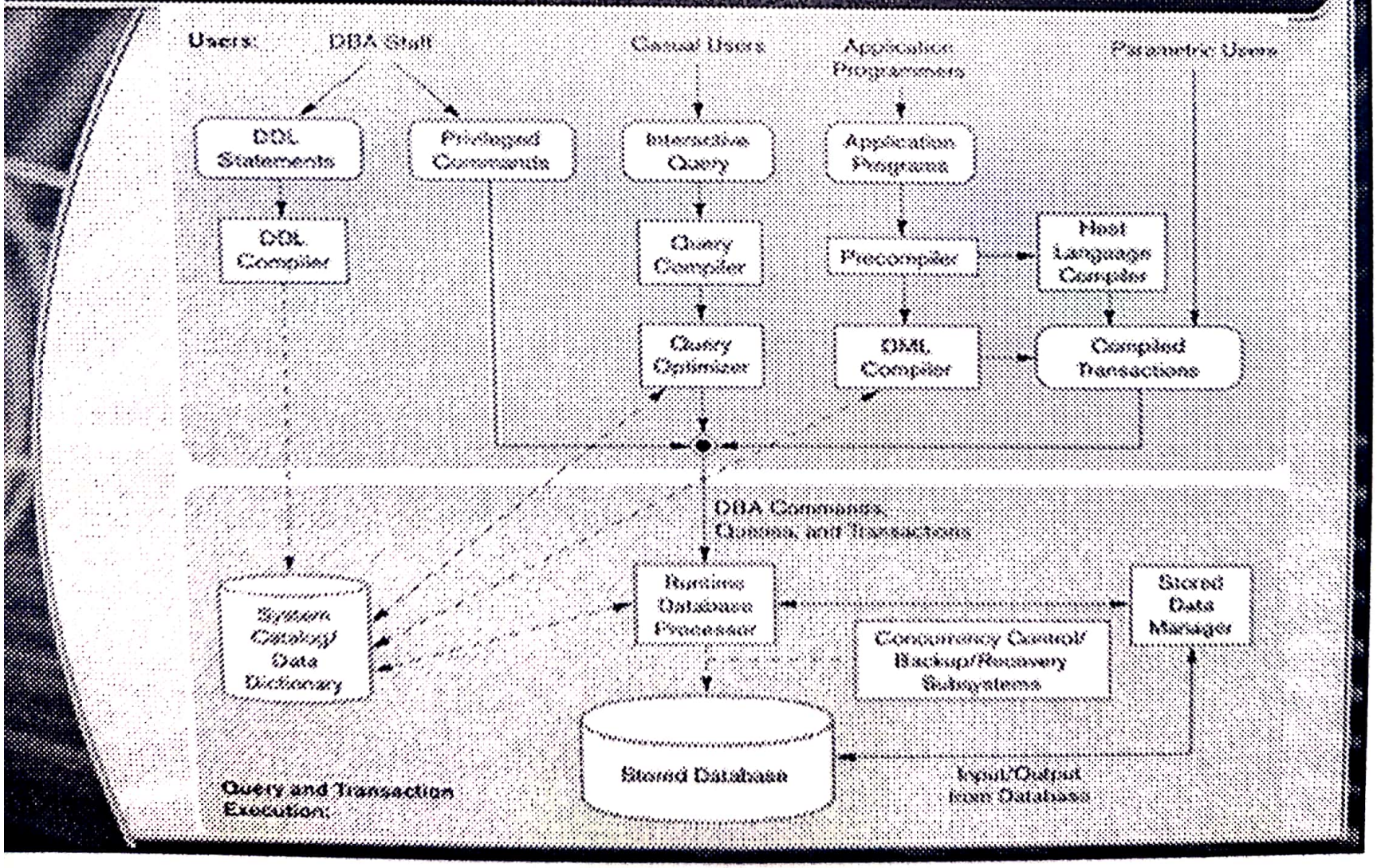
- Figure shows interfaces for the DBA staff, casual users who work with interactive interfaces to formulate queries,
- Application programmers who create programs using some host programming languages,
- Parametric users who do data entry work by supplying parameters to predefined transactions
- The DBA staff works on defining the database and tuning it by making changes to its definition using the DDL and other privileged commands.
- DDL compiler --> Processes schema definition --> stores it in the catalog
- Query Compiler --> handles high-level queries. it analyzes and interprets the queries by creating database access code. Then Query compiler call runtime database processor for executing that code.
- Query optimizer is concerned with the rearrangement and possible reordering of operations, elimination of redundancies, and use of efficient search algorithms during execution
- Pre-compiler --> extracts DML commands from an application program and send to DML Compiler. DML compiler --> compilation of DML commands into object code for access to database.

Database System Environment

- Runtime database processor -> handles database accesses at runtime. It receive retrieve or update request and it carries them on the stored database.
- Stored data manager-> carries data transfer between disk and main memory. This module control access to information that store on disk. Once data is in main memory, it can be processed by other DBMS module or application program

Scanned with CamScanner

Database System Environment



Scanned with CamScanner

Database System Utilities

- The most DBMSs have database utilities that help the DBA manage the database system
- Common utilities perform the following type of functions such as:
  - Loading data stored in files into a database. Includes data conversion tools.
    - The source file format and the target data file structure are mentioned to the utility
    - Utility automatically reformats the data stored it in the database
  - Backing up the database periodically on tape or other mass storage medium.
    - we can restore database from the backup copy in case of system failure
    - incremental backups are often used. Though it is complex, it save spaces
    - in incremental backups, only changes since the previous backup are recorded
  - Reorganizing database file structures
    - Used to reorganize a database file into a different file organization.
    - why is this required, to increase performance
  - Report generation utilities.
  - Performance monitoring utilites
    - Monitors database usage and provides complete statistics of the performance of database to the DBA
    - The DBA uses the statistics in making decisions such as whether or not to reorganize files or whether to add or drop indexes to improve performance
  - Other functions, such as sorting, user monitoring, data compression, etc.

1A-21

Other Tools

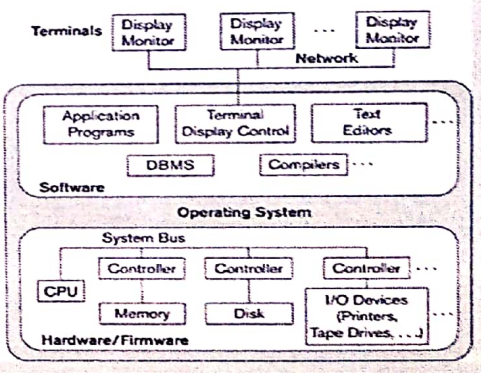
- CASE tools are used in the design phase of database systems. Many CASE tools are used primarily for database design
- Data dictionary / repository:
  - Used to store schema descriptions and other information such as design decisions, application program descriptions, user information, usage standards, etc.
  - A data dictionary utility is similar to the DBMS catalog, but it includes a wider variety of information and is accessed mainly by users rather than by the DBMS software.
- Application Development Environments: provides an environment for developing database applications.
  - Examples – Power builder (Sybase),JBuilder (Borland)
- Communication Software --> remote access to the database through computer terminals, workstations, or personal computers.
  - These are connected to the database site through data communications hardware such as Internet routers, phone lines, long-haul networks, local networks, or satellite communication devices
  - Many commercial database systems have communication packages that work with the DBMS. The integrated DBMS and data communications system is called a DB/DC system
  - Distributed DBMSs are physically distributed over multiple machines. In this case, communications networks are needed to connect the machines. These are often local area networks (LANs), WANs etc.

1A-22

DBMS Architecture: Centralized DBMSs Architecture

- Earlier, mainframe computers were used to process all system functions.
- Users accessed systems via computer terminals --> provided only display capabilities but no processing capabilities.
- All processing --> performed remotely on the computer system Only display information --> sent to the computer terminals
- These computer terminal --> connected to remote computer system via network
- Prices of hardware declined --> terminals replaced by PCs & workstations
- Initial Database systems used these computers in similar way as display terminals. But DBMS is centralized where all the functionality of DBMS, application program execution, and user interface processing were carried out on one machine.
- Slowly DBMS started making use of the processing ability of the systems at the user side also, not also just server side. This led to client/server DBMS architectures.
- **Centralized DBMS**: in short, combines everything into single system including- DBMS software, hardware, application programs and user interface processing software.

Centralized DBMSs Architecture



Scanned with CamScanner

Basic Client/Server Architectures

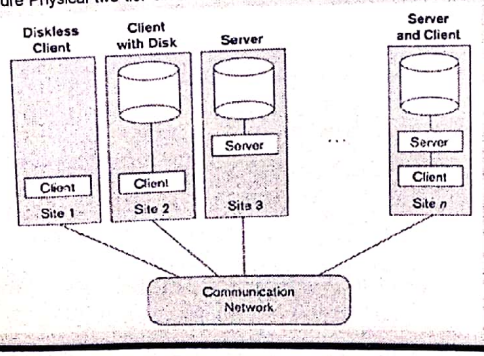
- The client/server architecture was developed to deal with computing environments in which a large number of PCs, workstations, file servers, printers, database servers, Web servers, e-mail servers, and other software and equipment are connected via a network.
- Goal --> define **specialized servers with specific functionalities**
  - For example, it is possible to connect a number of PCs or small workstations as clients to a file server that maintains the files of the client
  - Example2, Another machine can be designated as a printer server by being connected to various printers; all print requests by the clients are forwarded to this machine.
  - Web servers or e-mail servers also fall into the specialized server category. The resources provided by specialized servers can be accessed by many client machines
- Clients --> user machine that provides user interface capabilities and local processing
- Client provide appropriate interfaces and a client-version of the system to access and utilize the server resources.
- Clients maybe diskless machines or PCs or Workstations with disks with only the client software installed.
- Client connected to the servers via some form of a network. (LAN: local area network, wireless network, etc.)

Basic Client/Server Architectures

- When a client requires access to additional functionality—such as database access—that does not exist at the client, it connects to a server that provides the needed functionality.
- DBMS Server: is a system containing both hardware and software that can provide services to the client machines, such as file access, printing, archiving, or database access.
- DBMS server provides database query and transaction services to the clients
- DBMS server sometimes called query and transaction servers
- Two main types of basic DBMS Client/Server architectures are:
  - Two-tier architecture
  - Three-tier architecture
- which will be we discussing now.

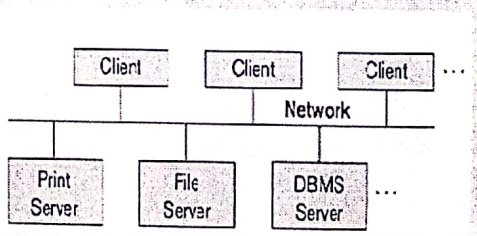
Basic Client/Server Architectures

Figure Physical two-tier client/server architecture.



Basic Client/Server Architectures

Figure Logical two-tier client/server architecture.



Scanned with CamScanner

Two-Tier Client/Server Architectures for DBMSs

User Interface Programs and Application Programs run on the client side

- In RDBMS, user interface and application programs --> moved to client side
- Query and transaction functionality --> on server side (Query server / transaction server)
- When DBMS access is required --> program establishes a connection with the DBMS (server side)
- Interface called ODBC (Open Database Connectivity – see Ch 9) provides an Application program interface (API) allow client side programs to call the DBMS. Most DBMS vendors provide ODBC drivers.
- A similar standard for the Java programming language, called JDBC, has also been defined.
  - This allows Java client programs to access one or more DBMSs through a standard Interface.
- A client program may connect to several DBMSs.
- Other variations of clients are possible: e.g., in some DBMSs, more functionality is transferred to clients including data dictionary functions, optimization and recovery across multiple servers, etc. In such situations the server may be called the Data Server.

Two-Tier Client/Server Architectures (cont.)

- The advantages of two-tier architecture are:
  - simplicity
  - and seamless compatibility with existing systems
- The architectures described above are called two-tier architectures because the software components are distributed over two systems: client and server.
- The emergence of the Web changed the roles of clients and servers, leading to the three-tier architecture.

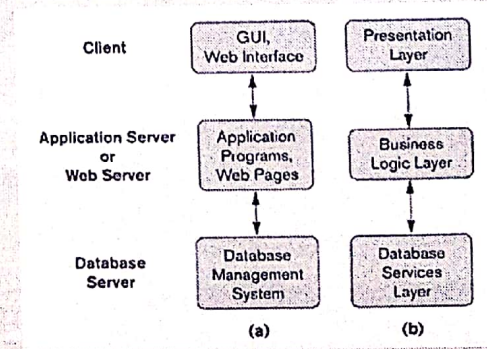
Three-Tier and n-Tier Architectures for Web Applications

- Common for Web applications
- Intermediate Layer called **middle tier or Application Server or Web Server:**
  - stores the web connectivity software and the rules and business logic (constraints) part of the application used to access the right amount of data from the database server
  - acts like a conduit for sending partially processed data between the database server & client.
- Additional Features- Security:
  - encrypt the data at the server before transmission
  - decrypt data at the client

Application server or Web server
↓
Stores rules used to access data
↓
Accepts requests from client
↓
Processes the requests
↓
sends command to database server

Three-Tier and n-Tier Architectures for Web Applications

- Figure Logical three-tier client/server architecture, with a couple of commonly used nomenclatures.

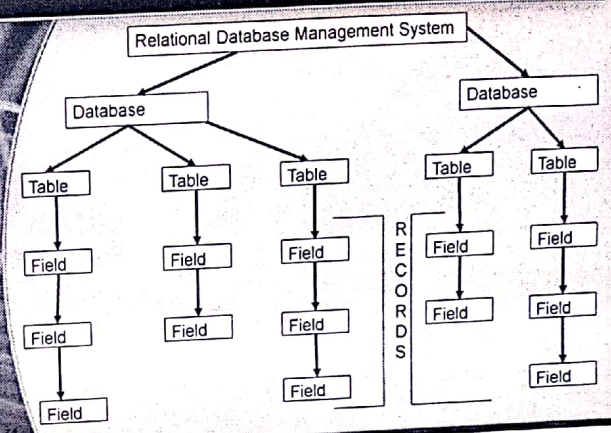


Scanned with CamScanner

Classification of Database Management Systems

- Several criteria used to classify DBMS:
- Data Model
- Number of users
- Cost of DBMS
- Number of sites
- Types of access path
- Generality
- Based on the data model used:
  - Traditional: Relational, Network, Hierarchical.
  - Emerging: Object-oriented, Object-relational.
- Other classifications:
  - Single-user (typically used with micro- computers) vs. multi-user (most DBMSs)
  - Centralized (uses a single computer with one database) vs. distributed (uses multiple computers, multiple databases)
- Data Model
  - Relational data model: represents database as a collection of tables. DBMS systems like oracle, MS Sqlserver use this kind of data model.

Classification of Database Management Systems

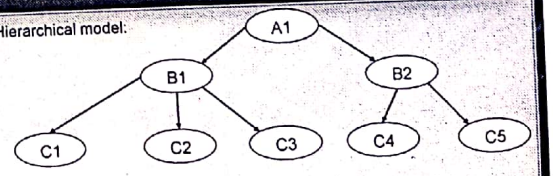


Classification of Database Management Systems

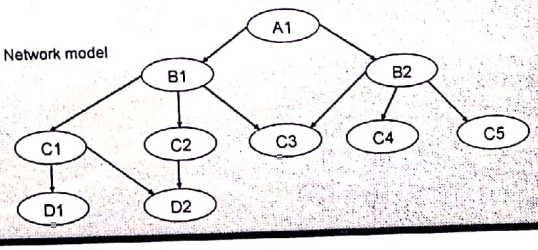
- Object-Oriented database model: defines database in terms of objects, their properties & their operations
- Hierarchical model: represents data as tree structures.
- Network model: represents object and their relationships in the form of graph

Classification of Database Management Systems

- Hierarchical model:



- Network model



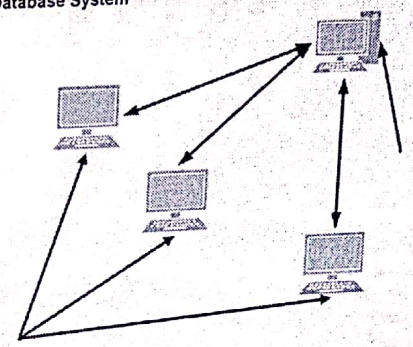
Scanned with CamScanner

Classification of Database Management Systems

- **Number of users:** Another criterion used to classify DBMSs is the number of users supported by the system.
  - Single-user systems support only one user at a time and are mostly used with PCs.
  - Multiuser systems, which include the majority of DBMSs, support concurrent multiple users.
- **Cost of DBMS:**
  - Low cost DBMS: Cost of these systems --> between $100 and $3000.
  - Medium cost DBMS: Cost varies between $10,000 and $100,000
  - High cost DBMS: Cost of these systems --> more than $100,000
  - Free cost DBMS: Open source (free) database MySQL and PostgreSQL
- **Number of sites:** Another criterion is the number of sites over which the database is distributed.
  - **Centralized database system:** Both DBMS and the database --> reside at a single computer site.
  - **Distributed database system:** DBMS & the database --> distributed over many sites, connected by a computer network.
- **Types of access path:** classify a DBMS on the basis of the types of access path options for storing files. One well-known family of DBMSs is based on inverted file structures.

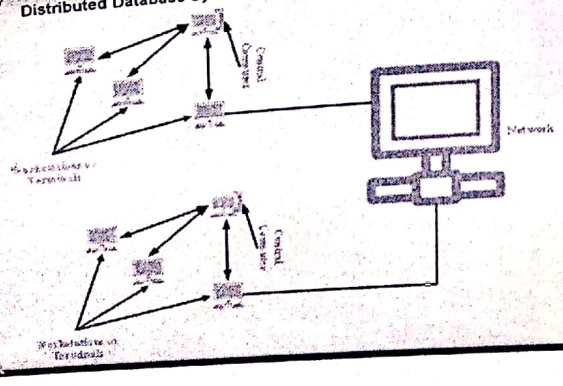
Classification of Database Management Systems

- **Central Database System**



Classification of Database Management Systems

- **Distributed Database System**



Classification of Database Management Systems

General purpose: A DBMS can be general purpose or special purpose. When performance is a primary consideration, a special-purpose DBMS can be designed and built for a specific application; such a system cannot be used for other applications without major changes.

- Many airline reservations and telephone directory systems developed in the past are special-purpose DBMSs.

Scanned with CamScanner

Variations of Distributed Environments

- **Homogeneous DDBMS:** use the same DBMS software at all the sites
- **Heterogeneous DDBMS:** can use different DBMS software at each site. It is also possible to develop middleware software to access several autonomous preexisting databases stored under heterogeneous DBMSs
- **Federated or Multi-database Systems:** in which the participating DBMSs are loosely coupled and have a degree of local autonomy. Many DDBMSs use client-server architecture

1A-41

Review Questions

- Define the following terms: data model, database schema, database state, internal schema, conceptual schema, external schema, data independence, DDL, DML, SDL, VDL, query language, host language, data sublanguage, database utility, catalog, client/server architecture, three-tier architecture, and n-tier architecture.
- Discuss the main categories of data models. What are the basic differences among the relational model, the object model, and the XML model?
- What is the difference between a database schema and a database state?
- Describe the three-schema architecture. Why do we need mappings among schema levels? How do different schema definition languages support this architecture?
- What is the difference between logical data independence and physical data independence? Which one is harder to achieve? Why?
- What is the difference between procedural and nonprocedural DMLs?
- Discuss the different types of user-friendly interfaces and the types of users who typically use each.

1A-42

Review Questions

- With what other computer system software does a DBMS interact?
- What is the difference between the two-tier and three-tier client/server architectures?
- Discuss some types of database utilities and tools and their functions.
- What is the additional functionality incorporated in n-tier architecture
- (n . 3)?

Any Questions

Scanned with CamScanner