Maharishi International University
Master's in Software Development

CS418: Databases & Software Development

Lesson 1

Course Overview: Introduction to Databases and MySQL

Wholeness

- This lecture focuses on giving an overview of what Databases are by reviewing their history and the important role they play in a typical data-driven Software solution.
- A Database is an organized, structured collection of data that is useful to an enterprise. *Throughout this lesson, we will find that order is present everywhere; it is only our lack of understanding of the natural order of life that causes problems to arise.*

Lesson 1 Objectives

- Learn and understand the meanings of the terms – Data, Information, Database, Database Management system (DBMS).
- Appreciate the importance of database systems and some of their common uses in real-world.
- Appreciate the evolution and development of DBMSs and consider the various types (RDBMS e.g. MySQL versus Non-Relational DBs e.g. MongoDB).
- Learn the typical functions of a DBMS.
- Identify and understand the major components of the DBMS environment and the roles involved.
- Examine the advantages of having DBMSs as well as some of their disadvantages/limitations.

1.1 Overview

Meaning of essentials terms:

- Data: is facts or figures collected together through observation or measurement, for the purpose of referencing or analysis.
- Information: is what is derived, when data has been made to become meaningful or useful through processing, interpreting or presenting.
- Database: is a structured collection of related data.
- Database Management system (DBMS): is software that manages and controls access to and operation of, a database.

1.1 Overview

- Database Application: is any computer program that interacts with a database at some point during its execution.
- Database system: is the collection of application programs that interact with the database, together with the DBMS and the database itself.

Note: All access to the Database is through the DBMS.

1.1 Overview

Why study Databases?

- The Database system is perhaps the most important development in the field of software engineering.
- A Database typically forms the underlying framework of the information system on which most organizations operate.
- The importance of database systems keeps increasing, driven by significant developments in hardware capability/capacity and data communication and the emergence of the Internet, eCommerce, Business Intelligence and network/grid computing.

1.1 Overview

- Some areas of application of Database systems:
  - Purchases from the supermarket.
  - Online purchases using your credit card.
  - Booking a vacation with a travel agency
  - Using the local/school library
  - Taking out insurance
  - Finding and watch a movie
  - Conducting online or traditional banking
  - Studying at a college or university
  - Using the Internet/World-wide websites

Main Point 1

- It is important that we understand what Databases are, their usefulness within the context of an overall enterprise information system and the related terminologies surrounding the study of Database systems. *Knowledge is different in different states of consciousness.*

Maharishi International University
Master's in Software Development

CS418:
Databases & Software Development

1.2 Evolution of Databases

- To appreciate the history of Database systems, it is helpful to consider a short review of its predecessor – the File-based system – and the problems it had.
- Traditional File-based system: was a system consisting of one or more application programs where-by each program defined and managed its own data.

1.2 Evolution of Databases

- File-based system:
  - Were an early attempt at computerizing the manual filing system.
  - For example, an organization might have physical files set up to hold data relating to a project, product, task, client or employee etc.
  - The files are labeled and stored in one or more filing cabinets and with locks for security.
  - When some information is needed the user goes to the filing system & searches through.

1.2 Evolution of Databases

- File-based system – a case study: *DreamHome Rental Agency*

  DreamHome is a property rental agency that has two departments – Sales and Contracts.

  The Sales department is responsible for the selling and renting of properties.

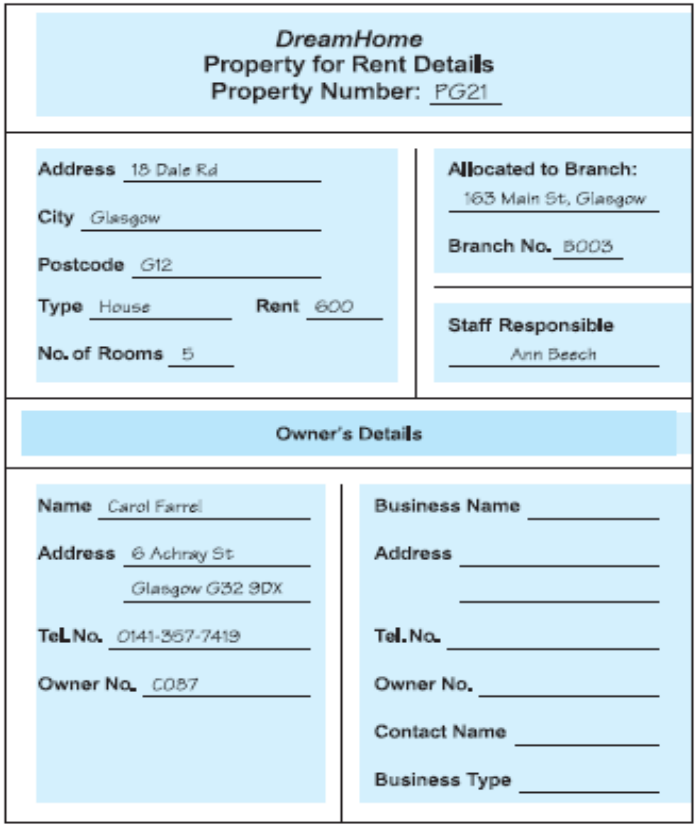
  For example, whenever a client who wishes to offer his or her property as a rental

  approaches the Sales Department, a form similar to the one shown in Figure 1.2(a)

  is completed.

1.2 Evolution of Databases

- File-based system – a case study: *DreamHome Rental Agency*



DreamHome
Property for Rent Details
Property Number: PG21

Address 18 Dale Rd
City Glasgow
Postcode G12
Type House Rent 600
No. of Rooms 5

Allocated to Branch:
163 Main St, Glasgow
Branch No. B003

Staff Responsible
Ann Beech

Owner's Details

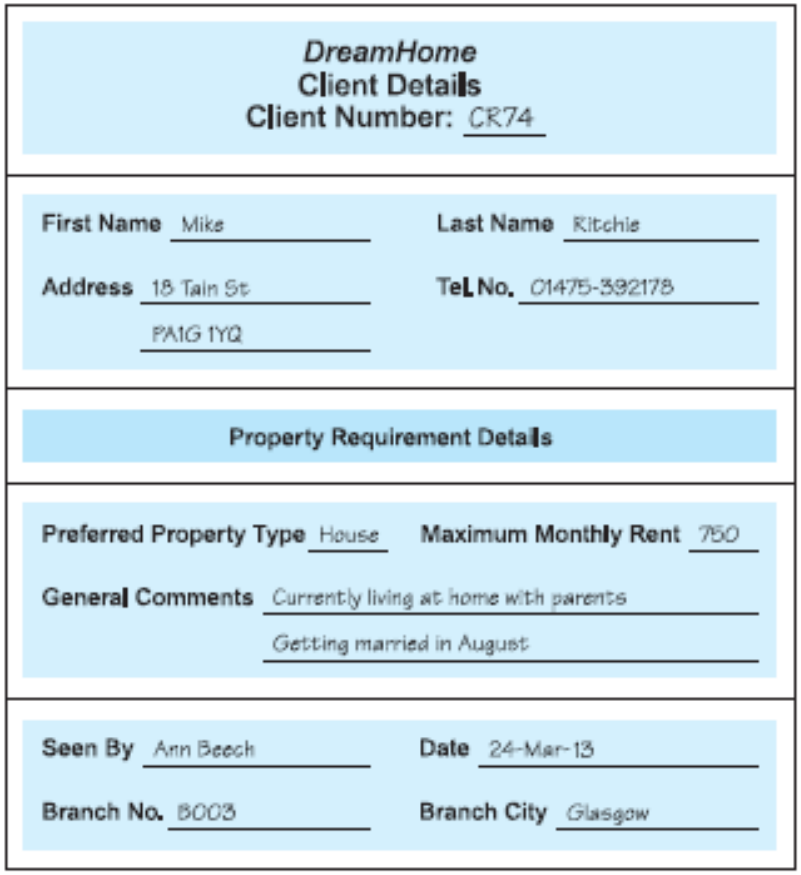
Name Carol Farrel
Address 6 Achray St
Glasgow G32 9DX
Tel. No. 0141-357-7419
Owner No. C087

Business Name
Address
Tel. No.
Owner No.
Contact Name
Business Type

Fig 1.2a – Sales department: Property for rent details form

1.2 Evolution of Databases

- File-based system – a case study: *DreamHome Rental Agency*



DreamHome
Client Details
Client Number: CR74

First Name Mike Last Name Ritchie

Address 18 Tain St Tel No. 01475-392178

PA1G 1YQ

Property Requirement Details

Preferred Property Type House Maximum Monthly Rent 750

General Comments Currently living at home with parents

Getting married in August

Seen By Ann Beech Date 24-Mar-13

Branch No. B003 Branch City Glasgow

The Sales Department also handles inquiries from clients, and a form similar to the one shown in Figure l.2 (b) is completed for each one.

Fig 1.2b – Sales department: Client details form

1.2 Evolution of Databases

- File-based system – a case study: *DreamHome Rental Agency*

  The Sales department's File-based system will consists of three files containing **property**, **owner**, and **client** details, as illustrated in Figure 1.2c

1.2 Evolution of Databases

- File-based system – a case study: *DreamHome Rental Agency*

PropertyForRent

propertyNo	street	city	postcode	type	rooms	rent	ownerNo
PA14	16 Holhead Rd	Aberdeen	AB7 5SU	House	6	650	CO46
PL94	6 Argyll St	London	NW2	Flat	4	400	CO87
PG4	6 Lawrence St	Glasgow	G11 9QX	Flat	3	350	CO40
PG36	2 Manor Rd	Glasgow	G32 4QX	Flat	3	375	CO93
PG21	18 Dale Rd	Glasgow	G12	House	5	600	CO87
PG16	5 Novar Dr	Glasgow	G12 9AX	Flat	4	450	CO93

PrivateOwner

ownerNo	fName	lName	address	telNo
CO46	Joe	Keogh	2 Fergus Dr, Aberdeen AB2 7SX	01224-861212
CO87	Carol	Farrel	6 Achray St, Glasgow G32 9DX	0141-357-7419
CO40	Tina	Murphy	63 Well St, Glasgow G42	0141-943-1728
CO93	Tony	Shaw	12 Park Pl, Glasgow G4 0QR	0141-225-7025

Client

clientNo	fName	lName	address	telNo	prefType	maxRent
CR76	John	Kay	56 High St, London SW1 4EH	0207-774-5632	Flat	425
CR56	Aline	Stewart	64 Fern Dr, Glasgow G42 0BL	0141-848-1825	Flat	350
CR74	Mike	Ritchie	18 Tain St, PA1G 1YQ	01475-392178	House	750
CR62	Mary	Tregear	5 Tarbot Rd, Aberdeen AB9 3ST	01224-196720	Flat	600

Fig. 1.2c – Sales department data files

1.2 Evolution of Databases

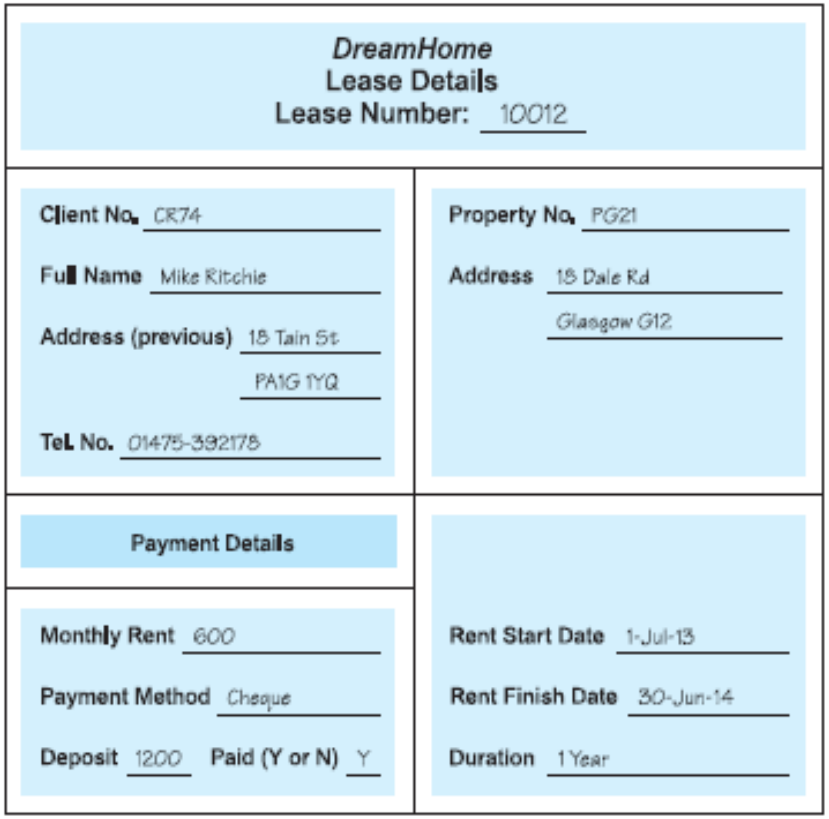
- File-based system – a case study: *DreamHome Rental Agency*

  DreamHome is a property rental agency that has two departments – Sales and Contracts.

  The Contracts Department is responsible for handling the lease agreements associated with properties for rent. Whenever a client agrees to rent a property, a form with the client and property details is filled in by one of the sales staff, as shown in Figure 1.2d. This form is passed to the Contracts Department, which allocates a lease number and completes the payment and rental period details.

1.2 Evolution of Databases

- File-based system – a case study: *DreamHome Rental Agency*



DreamHome
Lease Details
Lease Number: 10012

Client No. CR74
Full Name Mike Ritchie
Address (previous) 18 Tain St
PA1G 1YQ
Tel No. 01475-392178

Property No. PG21
Address 18 Dale Rd
Glasgow G12

Payment Details

Monthly Rent 600
Payment Method Cheque
Deposit 1200 Paid (Y or N) Y

Rent Start Date 1-Jul-13
Rent Finish Date 30-Jun-14
Duration 1 Year

Fig 1.2d – Contracts department: Lease details form

1.2 Evolution of Databases

- File-based system – a case study: *DreamHome Rental Agency*

  The Contracts department's file-based system will consists of three files that store **lease**, **property**, and **client** details, and that contain similar data to that held by the Sales Department, as illustrated in Figure 1.2e

1.2 Evolution of Databases

- File-based system – a case study: *DreamHome Rental Agency*

Lease

leaseNo	propertyNo	clientNo	rent	payment Method	deposit	paid	rentStart	rentFinish	duration
10024	PA14	CR62	650	Visa	1300	Y	1-Jun-13	31-May-14	12
10075	PL94	CR76	400	Cash	800	N	1-Aug-13	31-Jan-14	6
10012	PG21	CR74	600	Cheque	1200	Y	1-Jul-13	30-Jun-14	12

PropertyForRent

propertyNo	street	city	postcode	rent
PA14	16 Holhead	Aberdeen	AB7 5SU	650
PL94	6 Argyll St	London	NW2	400
PG21	18 Dale Rd	Glasgow	G12	600

Client

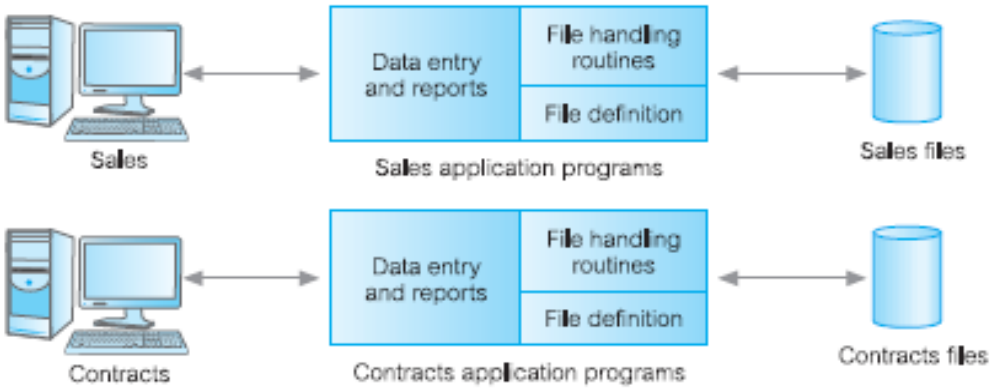
clientNo	fName	lName	address	telNo
CR76	John	Kay	56 High St, London SW1 4EH	0171-774-5632
CR74	Mike	Ritchie	18 Tain St, PA1G 1YQ	01475-392178
CR62	Mary	Tregear	5 Tarbot Rd, Aberdeen AB9 3ST	01224-196720

Fig. 1.2e – Contracts department data files

1.2 Evolution of Databases

- File-based system – a case study: *DreamHome Rental Agency*

Fig. 1.2f – File-based processing system - shows each department accessing their own files through application programs written especially for them. Each set of departmental application programs handles data entry, file maintenance, and the generation of a fixed set of specific reports. More important, the physical structure and storage of the data files and records are defined in the application code.



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)

1.2 Evolution of Databases

- File-based system – analysis
  - It can be seen quite clearly that there is a significant amount of duplication of data in these departments, and this is generally true of file-based systems.
  - To proceed in considering the limitations of this File-based approach, it may be useful to understand the terminology used in file-based systems.

1.2 Evolution of Databases

- File-based system – analysis
  - **A file is simply a collection of records, which contains logically related data.**

    For example, the **PropertyForRent** file in Figure 1.2c contains six records, one for each property. Each record contains a logically connected set of one or more fields, where each field represents some characteristic of the real-world object that is being modeled. In Figure 1.2c, the fields of the **PropertyForRent** file represent characteristics of properties, such as address, property type, and number of rooms.

1.2 Evolution of Databases

- Limitations of File-based approach
  - **Separation and Isolation of data.**

    When data is isolated in separate files, it is more difficult to access data that should be available.

    For example, if we want to produce a list of all houses that match the requirements of clients, we first need to create a temporary file of those clients who have “house” as the preferred type. We then search the **PropertyForRent** file for those properties where the property type is “house” and the rent is less than the client’s maximum rent. With file systems, such processing is difficult. The application developer must synchronize the processing of two files to ensure that the correct data is extracted. This difficulty is compounded if we require data from more than two files.

1.2 Evolution of Databases

- Limitations of File-based approach
  - **Duplication of data.**

    Due to the decentralized approach taken by each department, the file-based

    approach encouraged, if not necessitated, the uncontrolled duplication of data. For example, in Figure 1.2x we can clearly see that there is duplication of both property and client details in the Sales and Contracts Departments.

    This results in waste, with higher cost in time and money for entering data more than once. And the unnecessary additional data storage space being required.

1.2 Evolution of Databases

- Database approach
  - A Database and Database Management system offers a more effective solution to the above limitations posed by the traditional file-based systems.
  - A Database, more formally, can be defined as a shared collection of logically related data and its description, designed to meet the information needs of an organization.
  - It is a single, usually large repository of data that can be used by many departments and users

1.2 Evolution of Databases

- Database approach
  - Instead of disconnected files with redundant data, all data items are integrated with a minimum amount of duplication.
  - The database is no longer owned by one department but is a shared corporate resource.
  - The database holds not only the organization's operational data, but also a description of this data. For this reason, a database is also defined as **a self-describing collection of integrated records**. The description of the data is known as the **system catalog** (or **data dictionary** or **metadata**—the "data about data"). It is the self-describing nature of a database that provides program–data independence.

1.2 Evolution of Databases

- Database approach
  - Question: What do we mean by, “logically related”?
  - To design and implement a computerized system for an organization, we first need to perform some **Data analysis**, where-by we analyze the data needs of the organization.
  - During this Data analysis task, we attempt to identify **Entities**, **Attributes**, and **Relationships**.

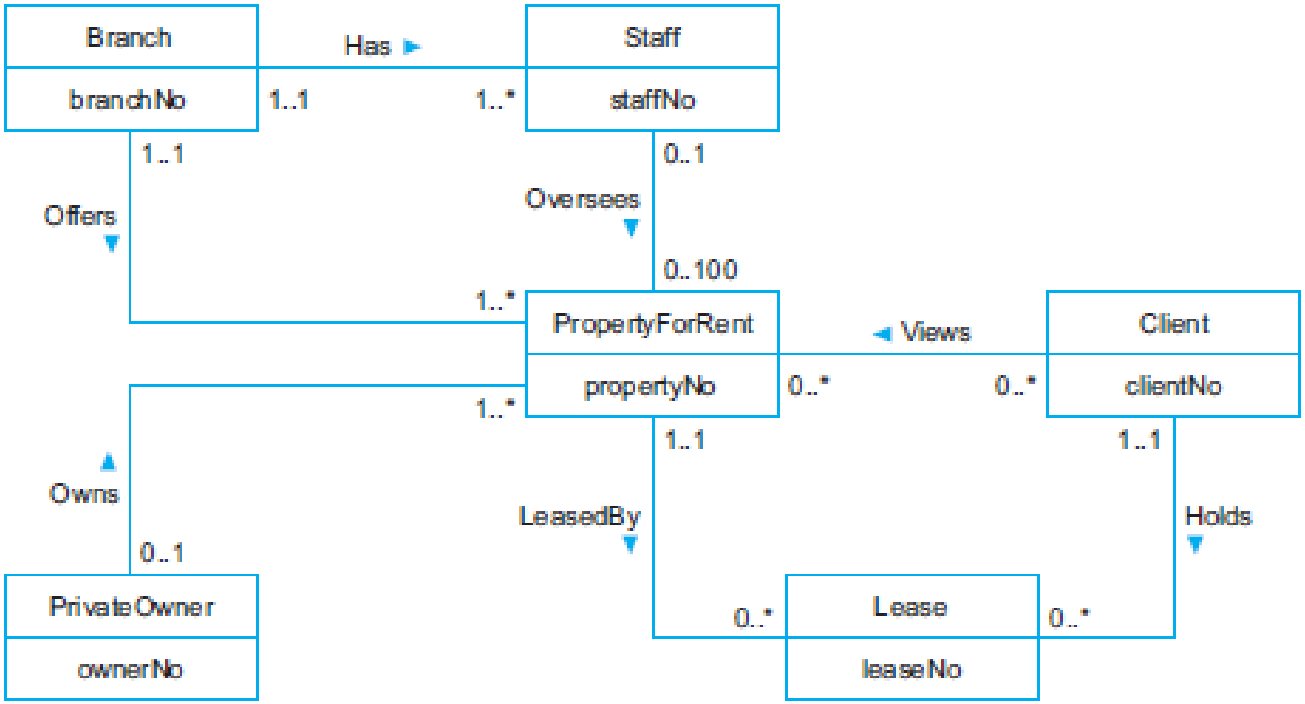
1.2 Evolution of Databases

- Database approach
  - An Entity is a distinct object (a person, place, thing, concept, or event) in the organization that is to be represented in the database.
  - An Attribute is a property that describes some aspect of the object (entity) that we wish to record.
  - A relationship is an association between entities.

1.2 Evolution of Databases

- Database approach

  Figure 1.2g shows an Entity–Relationship (ER) diagram for part of the *DreamHome* case study.



1.2 Evolution of Databases

- Database approach

  The E-R diagram consists of:

  – six entities (the boxes/rectangles): Branch, Staff, PropertyForRent, Client, PrivateOwner, and Lease.

  – seven relationships (the names adjacent to the lines): Has, Offers, Oversees, Views, Owns, LeasedBy, and Holds.

  – six attributes, one for each entity: branchNo, staffNo, propertyNo, clientNo, ownerNo, and leaseNo.

1.2 Evolution of Databases

- The Database
  - The database represents the entities, the attributes, and the logical relationships between the entities. In other words, the database holds data that is logically related.
  - We will examine/study the E-R diagrams and E-R modeling in further detail in Lesson 3.

1.2 Evolution of Databases

- The Database Management system (DBMS)
  - The DBMS is the software that interacts with the users' application programs and the database.
  - It is a software system that enables users to define, create, maintain, and control access to the database.

1.2 Evolution of Databases

- Features of a Database Management system (DBMS)
  - It allows users to define the database and its constituent objects, usually through a **Data Definition Language (DDL)**. The DDL allows users to specify the data types and structures and the constraints on the data to be stored in the database.

1.2 Evolution of Databases

- Features of a Database Management system (DBMS)
  - It allows users to insert, update, delete, and retrieve data from the database, usually through a **Data Manipulation Language (DML)**. Having a central repository for all data and data descriptions allows the DML to provide a general inquiry facility to this data, called a query language.
  - The most common query language is the Structured Query Language (SQL, pronounced “S-Q-L”, or sometimes “See-Quel”), which is now both the formal and de-facto standard language for relational DBMSs. We study SQL in Lesson 2.

1.2 Evolution of Databases

- Features of a Database Management system (DBMS)
  - It provides controlled access to the database; which includes:
    - a security system, which prevents unauthorized users accessing the database.
    - an integrity system, which maintains the consistency of stored data.
    - a concurrency control system, which allows shared access of the database.
    - a recovery control system, which restores the database to a previous consistent state following a hardware or software failure.

1.2 Evolution of Databases

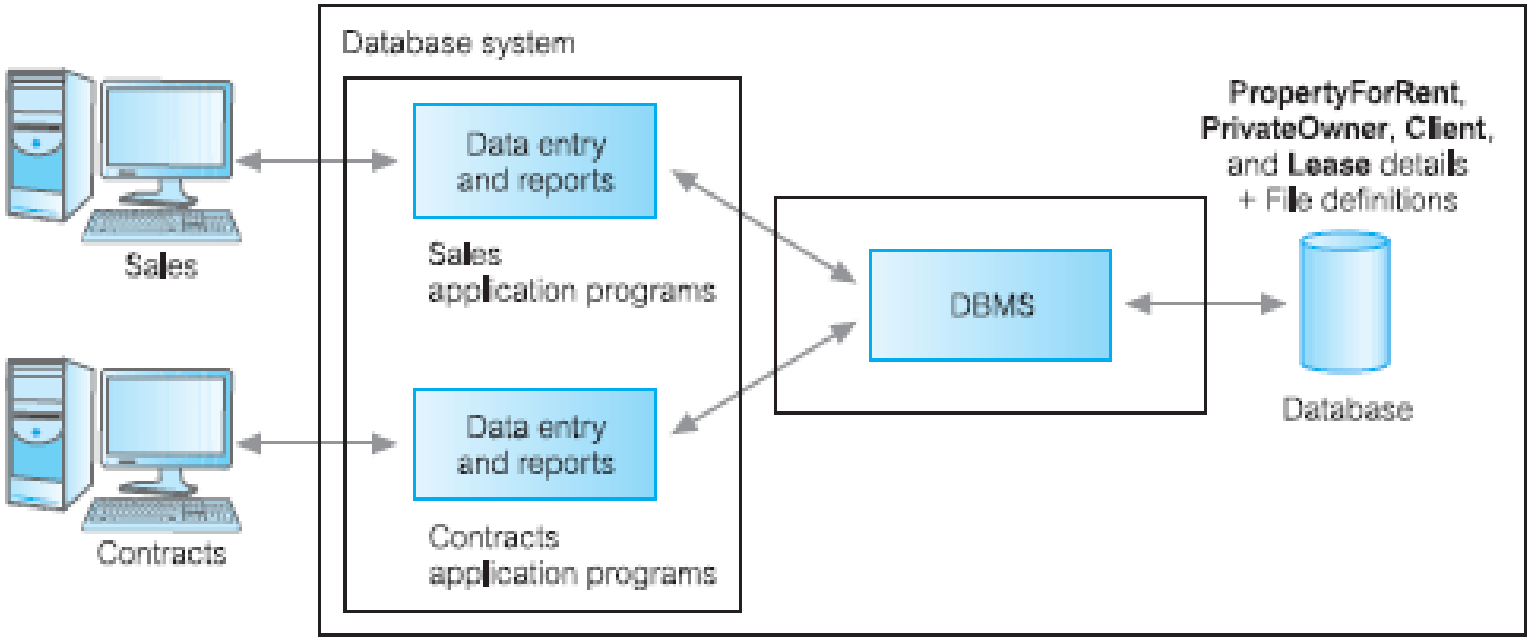
- Features of a Database Management system (DBMS)
  - It provides controlled access to the database; which includes:
    - a user-accessible catalog (aka the database schema or data dictionary), which contains descriptions of the data in the database.

1.2 Evolution of Databases

- Database Application Program
  - Is a computer program that interacts with the database by issuing an appropriate request (typically an SQL statement) to the DBMS.
  - Users interact with the database through a number of application programs that are used to create and maintain the database and to generate information.
  - Application programs can be command line interface app, Desktop GUI app, a batch application or an online/web app.

1.2 Evolution of Databases

- Fig 1.2h – the Database approach applied to the *DreamHome* case study



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)

Main Point 2

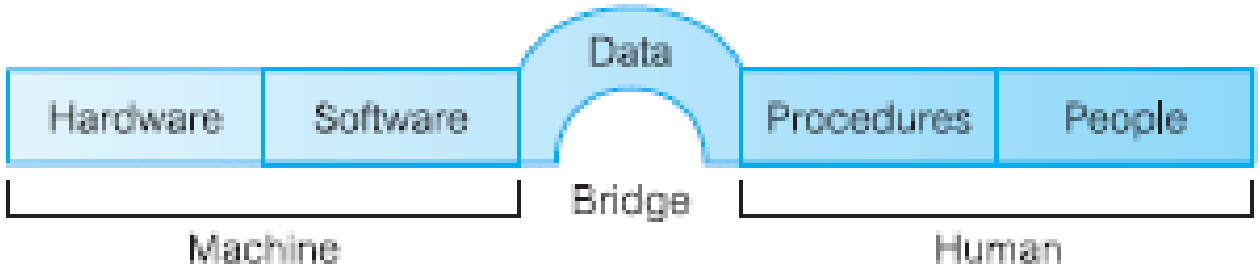
- Database systems have evolved over the years, changing from traditional File-based systems to what is the predominant technology today, the Database Management systems (DBMSs). *The Nature of life is to grow!*

Maharishi International University
Master's in Software Development

CS418:
Databases & Software Development

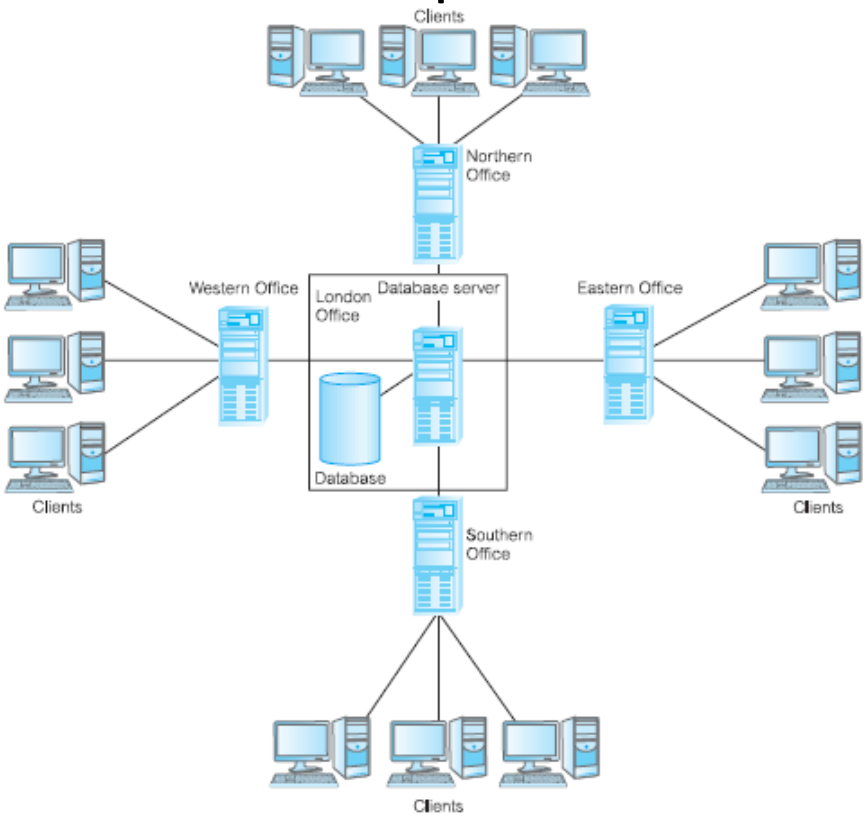
1.3 Components of the DBMS Environment

- The following 5 major components can be identified in the DBMS environment:
  - Hardware
  - Software
  - Data
  - Procedures
  - People



1.3 Components of the DBMS Environment

- Hardware: the DBMS require hardware to run. This can be a single personal computer or a large Mainframe computer or a network of servers.



1.3 Components of the DBMS Environment

- Software: The software component comprises the DBMS software itself and the application programs, together with the operating system, including network software if the DBMS is being used over a network.
- Data: This is quite easily the most important component of the DBMS, from the end-users' point of view
  - the data acts as a bridge between the machine components and the human components.
  - The database contains both the operational data and the metadata, the "data about the data".

1.3 Components of the DBMS Environment

- Data:
  - The structure of the database is called the schema.
  - For example, in Figure 1.2h, the schema consists of four files, or tables, namely: PropertyForRent, PrivateOwner, Client, and Lease.
  - The PropertyForRent table has eight fields, or attributes, namely: propertyNo, street, city, zipCode, type (the property type), rooms (the number of rooms), rent (the monthly rent), and ownerNo.
  - The ownerNo attribute models the relationship between PropertyForRent and PrivateOwner: that is, an owner Owns a property for rent, as depicted in the ER diagram of Figure 1.2g. For example, in Figure 1.2c we observe that owner CO46, Joe Keogh, owns property PA14.

1.3 Components of the DBMS Environment

- Procedures: This refers to the instructions and rules that govern the design and use of the database. The users of the system and the staff who manage the database require documented procedures on how to use or run the system. These may consist of instructions on how to:
  - Log on to the DBMS
  - Use a particular DBMS facility or application program
  - Start and stop the DBMS
  - Make backup copies of the database
  - Handle hardware or software failures
- People: This means the people or roles involved with the Database system; which will be the next topic in Section 1.4

1.4 Roles in the Database Environment

- Four (4) distinct types of people (roles) who participate in a DBMS environment, can be identified:
  - Data and Database Administrators (Das & DBAs): Responsible for management and control of the DBMS and its data
  - Database designers: These has the Logical Database designers (identify data entities, attributes and constrains – business rules) and the Physical Database designers (plans and decides how the Logical DB is to be physically implemented)
  - Application Developers: Responsible for developing the application programs which will make use of the Database.
  - End-users: these are the users of the database, whose information needs are to be served.

1.5 Advantages and disadvantages of DBMSs

- Advantages:
  - Control of data redundancy.
  - Greater guarantee of Data consistency.
  - Access to more information for the same amount of data
  - Sharing of data
  - Improved data integrity
  - Improved security
  - Lower cost of building and maintaining enterprise information systems

1.5 Advantages and disadvantages of DBMSs

- Disadvantages:
  - Complexity
  - Need for larger size of storage resources
  - Cost of DBMSs
  - Additional hardware cost
  - Cost of converting from non-DBMS to DBMS
  - Some Performance degradation
  - Greater impact of failure*

** It depends on the architecture. Can be avoided.*

1.6 Data Models and Conceptual Modeling

- A model is a representation of real-world objects and events, and their associations.
- A Data model is an integrated collection of concepts for describing and manipulating data, relationships between data, and constraints on the data in an organization.
- A Conceptual data model is one that is used to represent the logical view of an organization's data, which is DBMS-independent.
- Two of the broad categories of data models that exist are: **Object-based** and **Physical** model.

1.6 Data Models and Conceptual Modeling

- **Object-based data model** is used to describe data at the conceptual level
- Object-based data models use concepts such as **entities**, **attributes**, and **relationships**.
  - An entity is a distinct object (a person, place, thing, concept, event) in the organization that is to be represented in the database.
  - An attribute is a property that describes some aspect of the object that we wish to record.
  - A relationship is an association between entities.
- An example of an Object-based data model is the Entity-Relationship (ER) model

1.6 Data Models and Conceptual Modeling

- The **Entity-Relationship (ER) model** has emerged as one of the main techniques for database design and forms the basis for the database design methodology we study in this course.
- **Conceptual modeling or conceptual database design** is the process of constructing a model of the information use in an enterprise that is independent of implementation details, such as the target DBMS, application programs, programming languages, or any other physical considerations. This model is called a conceptual data model.

1.7 Functions of a DBMS

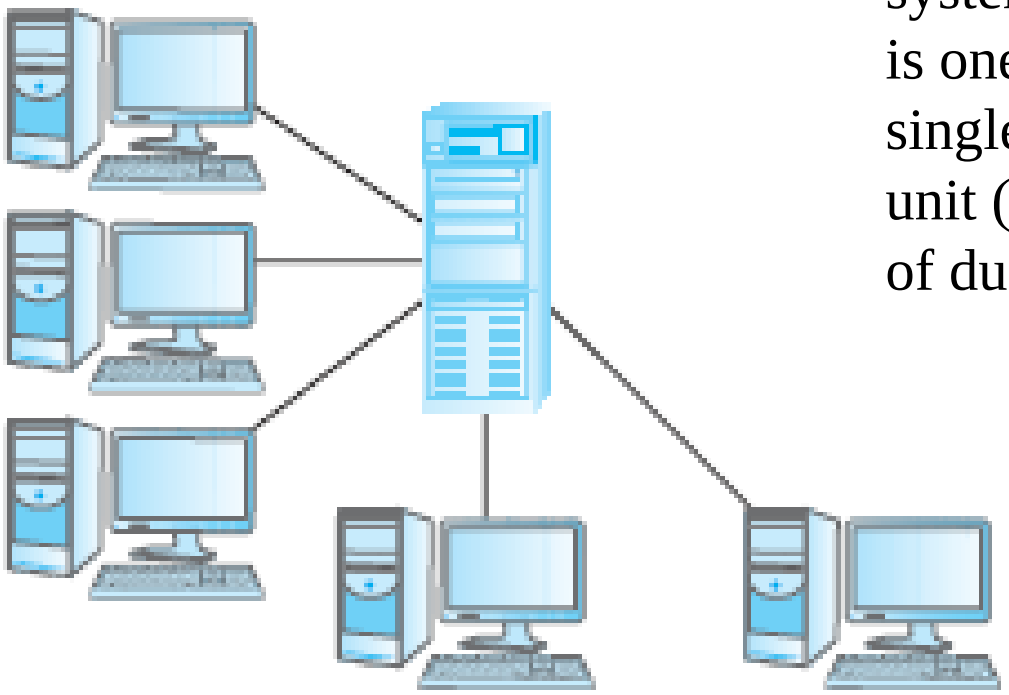
- A typical DBMS can be expected to provide the following functions and services:
  - Data storage, retrieval and update capabilities
  - A User-accessible catalog
  - Transaction support: Most DBMSs will ensure ACID properties in their transaction management capabilities (A: Atomicity, C: Consistency, I: Isolation, D: Durability)
  - Concurrency control services: multiuser support with various locking mechanisms to ensure data integrity and consistency
  - Recovery services
  - Authorization services
  - Support for data communication
  - Integrity services

1.7 Functions of a DBMS

- A typical DBMS can be expected to provide the following functions and services:
  - Services to promote data independence
  - Utility services – most DBMSs provide a set of utility programs and tools for the DBAs and AppDevs

1.8 Database Application Architectures

- Multi-user DBMS Architectures:
  - Teleprocessing



This was the traditional architecture for multi-user systems, where there is one computer with a single central processing unit (CPU) and a number of dumb terminals

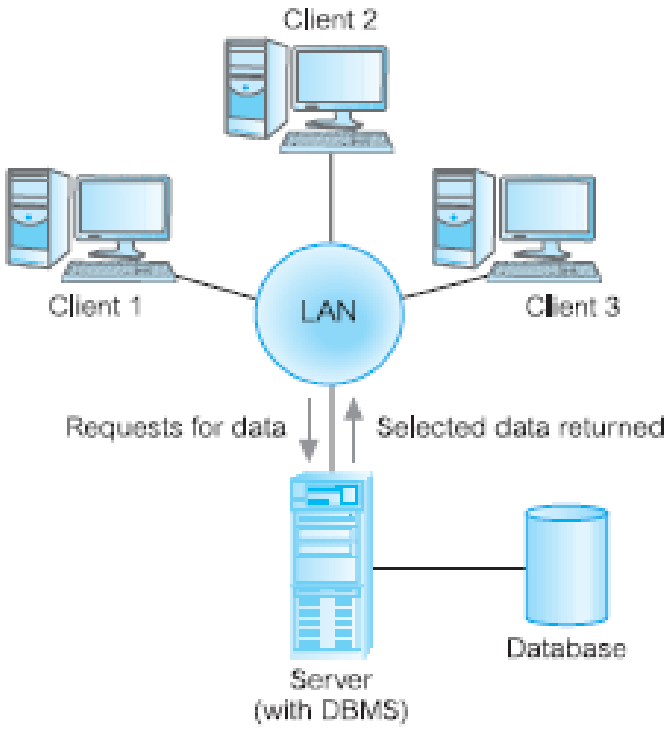
1.8 Database Application Architectures

- Multi-user DBMS Architectures:
  - File-Server Architecture

In this architecture, the processing is distributed about the network (LAN). The file-server holds the files required by the applications and the DBMS. However, the applications and the DBMS run on each workstation, requesting files from the file-server when necessary

1.8 Database Application Architectures

- Multi-user DBMS Architectures:
  - Traditional 2-tier Client-Server Architecture

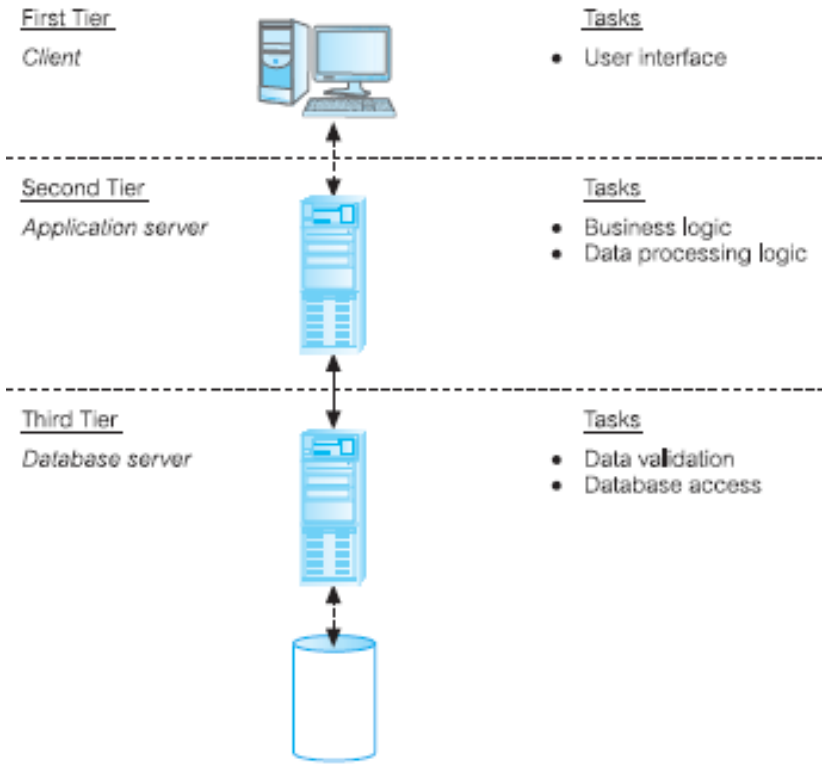


Client–server refers to the way in which software components interact to form a system.

There is a client process, which requires some resource, and a server, which provides the resource. There is no requirement that the client and server must reside on the same machine.

1.8 Database Application Architectures

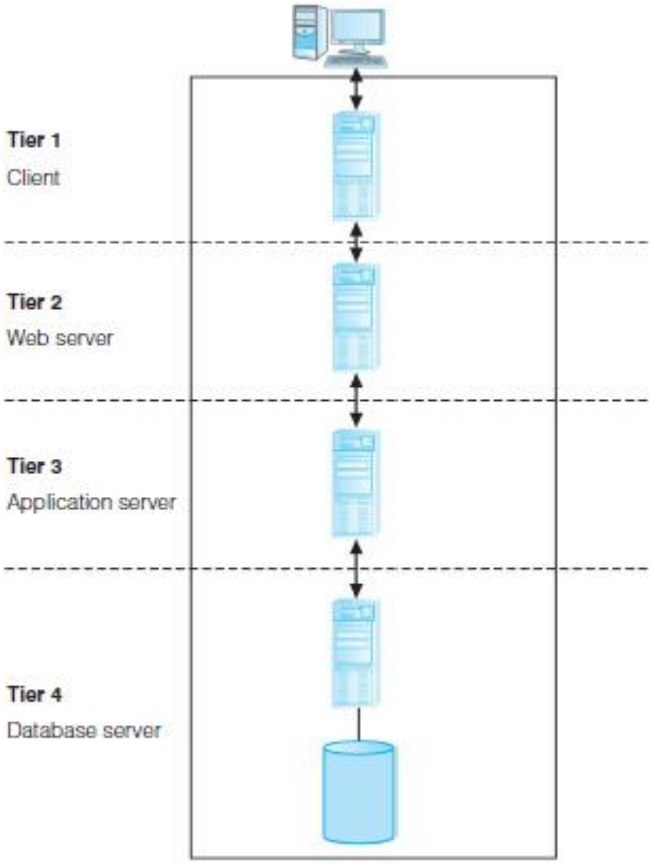
- Multi-user DBMS Architectures:
  - 3-tier Client-Server Architecture



This architecture proposed three layers, each potentially running on a different platform:
The user interface layer, which runs on the end-user's computer (the client).
The business logic and data processing layer. This middle tier runs on a server and is often called the application server.
A DBMS, which stores the data required by the middle tier. This tier may run on a separate server called the database server.

1.8 Database Application Architectures

- Multi-user DBMS Architectures:
  - N-tier Client-Server



The three-tier architecture can be expanded to n tiers, with additional tiers providing more flexibility and scalability.
The middle tier of the architecture could be split into two, with one tier for the Web server and another tier for the application server. In environments with a high volume of throughput, the single Web server could be replaced by a set of Web servers (or a Web farm) to achieve efficient load balancing.

1.9 Cloud Computing and Cloud-hosted Databases

- Cloud computing is the term given to the use of multiple servers over a digital network (such as the Internet) as if they were one computer.
- It is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.
- The following 3 service models are exist:
  - Infrastructure as a Service (IaaS)
  - Platform as a Service (PaaS)
  - Software as a Service (SaaS)

1.9 Cloud Computing and Cloud-hosted Databases

- Cloud-based Database Solutions: As a type of SaaS, cloud-based database solutions fall into two basic categories: Data as a Service (DaaS) and Database as a Service (DBaaS):
  - Daas offers the ability to define data in the cloud and subsequently query that data on demand. Unlike traditional database solutions, DaaS does not implement typical DBMS interfaces such as SQL. Instead, the data is accessed via a common set of APIs. e.g. Urban Mapping, a geography data service.
  - DBaaS offers full database functionality to application developers.

1.9 Cloud Computing and Cloud-hosted Databases

- Cloud-based Database Solutions: As a type of SaaS, cloud-based database solutions fall into two basic categories: Data as a Service (DaaS) and Database as a Service (DBaaS):
  - In DBaaS, a management layer is responsible for the continuous monitoring and configuring of the database to achieve optimized scaling, high availability, multi-tenancy (that is, serving multiple client organizations), and effective resource allocation in the cloud, thereby sparing the application developer from ongoing database administration tasks. Examples include: Amazon's RDS, Google's BigQuery, Microsoft Azure DB etc.

Main Point 3

- DBMS environment has 5 major components: Hardware, Software, Data, Procedures and People.
- Database Application architecture are typically structured in layers/tiers which demonstrates that *'Life is found in layers',* and that *'lower layers have more power'.*

Maharishi International University
Master's in Software Development

CS418:
Databases & Software Development

1.10 Introduction to MySQL

- MySQL is an open-source relational database management system (RDBMS)
- In this course we will be studying and implementing the various Database Management system concepts and techniques using a MySQL instance
- Therefore, you are required to download and install a MySQL Database server on your local machine (if you do not already have it).*
- MySQL can be obtained from https://dev.mysql.com/

***Note: A demonstration of how to obtain and install MySQL will be given.**

Maharishi International University
Master's in Software Development

CS418:
Databases & Software Development

Conclusion

- This lesson 1 has given you an overview of the Databases and Database Management systems.
- You should now be sufficiently familiar with general idea of what a Database is. Your understanding of these will guide the rest of our studies throughout the remainder of this course.
- The subsequent lessons will explore these topics in greater detail.

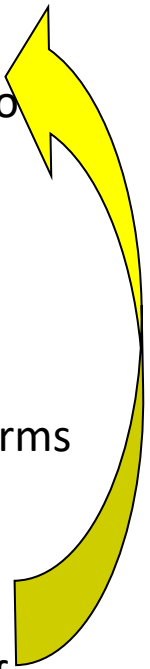
Connecting the Parts of Knowledge With the Wholeness of Knowledge

Overview and Introduction of Databases and DBMSs

1. A Database is a shared collection of logically related data and its description, designed to meet the information needs of an organization
2. Database Management system (DBMS) is a software system that enables users to define, create, maintain, and control access to the database.

---

3. **Transcendental consciousness** is the underlying basis of all levels of creation.
4. **Impulses within the Transcendental Field:** The well-designed Database which forms the basis of an elegant software solution, arises as an impulse of the Transcendental Field.
5. **Wholeness moving within itself**: In Unity Consciousness, one directly perceives that all expressions and levels of creation are nothing more than one's own Self – pure consciousness.



Maharishi International University
Master's in Software Development

CS418:
Databases & Software Development