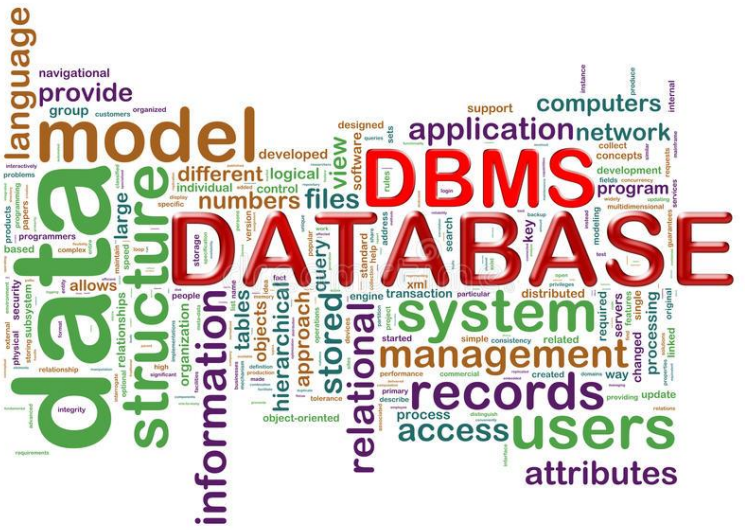
Notes On

Database Management Systems

Unit-I : Introduction & RDBMS Design



B.TECH II YEAR - I SEM

LENDI INSTITUTE OF ENGINEERING AND TECHNOLOGY

An Autonomous Institution

Approved by AICTE & Permanently Affiliated to JNTU-GV, Vizianagaram

Accredited by NAAC with "A" Grade and NBA (CSE,ECE, EEE & ME)

Jonnada (Village), Denkada (Mandal), Vizianagaram Dist – 535 005

Phone No. 08922-241111, 241112

E-Mail: lendi_2008@yahoo.com

Website: www.lendi.org

Dr. M. Satyanarayana., Professor, Dept. of CSE, LENDI

LENDI INSTITUTE OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SYLLABUS

Course Code	Subject Name	L	T	P	C
R20CSE-PC2101	**Database Management Systems Common to CSE & CSIT**	**3**	**0**	**0**	**3**

Course objectives:

- Learn the fundamental concepts of database systems.
- Enable students to design ER diagram for any customized applications
- Learn simple and Complex queries using SQL.
- Learn schema refinement techniques (Normalization).
- Knowledge about transaction and recovery techniques.

Course Outcomes:

1. Understand File System Vs Databases.
2. Design and implement ER-model and Relational models.
3. Construct simple and Complex queries using SQL.
4. Analyze schema refinement techniques.
5. Design and build database system for a given real world problem

Unit-I

Introduction- Database Vs File System, RDBMS, Database Users, Data Models; Instance and DataIndependence; Three Tier Schema Architecture, Database System Structure,
RDBMS Design: Introduction, Entities, Attributes Entity Set, Relationship Set,Specialization, andGeneralization. Types of Keys

Learning outcomes:

Student will be able to
1. Distinguish between Database System and File System (L2)
2. Design a database relational model using ER diagrams. (L5)

UNIT-II

Relational operations & Basic SQL: Relational Algebra, Relational Operations, Relational Calculus, Tuple And Domain Relational Calculus.
PL/ SQL : Database Languages, Data Types, Integrity Constraints, Simple And Nested Queries,Implementation Of Different Types Of Joins, Stored Procedures

Learning Outcomes:

Student will be able to
1. Understand relational operations and calculus (L1)
2. Implement simple and complex queries for relational data (L3)

UNIT-III

Schema Refinement (Normalization): Types Of Anomalies, Concept Of Functional Dependency,Normalization, Advantages ,Types Of Normal forms(1NF, 2NF And 3NF), Boyce-Codd NormalForm(BCNF), Fourth Normal Form(4NF) .Lossless Join And Dependency PreservingDecomposition,.

Learning Outcomes:

Student will be able to
1. Identify anomalies and remove redundancies using Normal Forms (L2)
2. Understand lossy and lossless decomposition.(L3)

Unit-IV

Transaction Management: Transaction, Transaction States, ACID Properties, Schedule, Serializability And Types, Concurrent Control, Concurrency Control Protocols, Crash Recovery: Introduction To ARIES, The Log, Write-Ahead Log Protocol, Recovering From A System Crash

Learning Outcomes:

Student will be able to

1. Understand transaction and serializability schedules. (L1)
2. Understand concurrency control protocols on transactions. (L1)

UNIT-V

File Organization and Indexing, Physical Storage Media, RAID, File Organization. Indexing, B & B+ Tree Index files, Hashing Vs Indexing

Learning Outcomes:

Student will be able to

1. Understand basic concepts of File Organization and storage (L1).
2. Understand Indexing and hashing for file processing.(L1)

Text Books:

1. Database Management Systems, 3/e, Raghurama Krishnan, Johannes Gehrke,TMH
2. Database System Concepts,5/e, Silberschatz, Korth, TMH

Reference Books:

1. Database Management System, 6/e RamezElmasri, Shamkant B. Navathe, PEA
2. Database Principles Fundamentals of Design Implementation and Management, CarlosCoronel, Steven Morris, Peter Robb, Cengage Learning.
3. Introduction to Database Systems, 8/e C J Date, PEA..

S NO	UNIT	Topic of syllabus – No. of Hrs Required - 11	No. of Classes	
1	I	Introduction- Database Vs File System	1	
2	I	RDBMS, Database Users, Data Models	1	
3	I	RDBMS, Database Users, Data Models	1	
4	I	Instance and DataIndependence	1	
5	I	Three Tier Schema Architecture	1	
6	I	Database System Structure	1	
7	I	RDBMS Design: Introduction	1	
8	I	Entities, Attributes Entity Set, Relationship Set	1	
9	I	Entities, Attributes Entity Set, Relationship Set	1	
10	I	Specialization, and Generalization, Types of Keys	1	

UNIT – I

DBMS stands for Database Management System. We can break it like this

DBMS = Database + Management System. Database is a collection of data and Management System is a set of programs to store and retrieve those data. Based on this we can define DBMS like this: DBMS is a collection of inter-related data and set of programs to store & access those data in an easy and effective manner. Here are the DBMS notes to help you learn database systems in a Systematic manner. Happy Learning!!

What is data: Data is the known facts or figures that have implicit meaning. It can also be defined as itis the representation of facts ,concepts or instruction in a formal manner, which is suitablefor understanding and processing. Data can be represented in alphabets(A-Z, a-z),indigits(0-9) and using special characters(+,-.#,$, etc)e.g: 25, “ajit” etc.

Information: Information is the processed data on which decisions and actions are based. Informationcan be defined as the organized and classified data to provide meaningful values.

Eg: “This is LENDI Institute of Engineering and Technology”

File: File is a collection of related data stored in secondary memory.

File Oriented approach:

The traditional file oriented approach to information processing has for each application a separate master file and its own set of personal file. In file oriented approach the programdependent on the files and files become dependent on the files and files becomedependents upon the programs

Disadvantages of file oriented approach:

1) Data redundancy and inconsistency:

The same information may be written in several files. This redundancy leads tohigher storage and access cost. It may lead data inconsistency that is the variouscopies of the same data may longer agree for example a changed customer addressmay be reflected in single file but not elsewhere in the system.

2) Difficulty in accessing data :

The conventional file processing system do not allow data to retrieved in aconvenient and efficient manner according to user choice.

3) Data isolation :

Because data are scattered in various file and files may be in different formatswith new application programs to retrieve the appropriate data is difficult.

4) Integrity Problems:

Developers enforce data validation in the system by adding appropriate code inthe various application program. However when new constraints are added, it isdifficult to change the programs to enforce them.

5) Atomicity:

It is difficult to ensure atomicity in a file processing system when transactionfailure occurs due to power failure, networking problems etc.(atomicity: either all operations of the transaction are reflected properly in thedatabase or non are)

6) Concurrent access:

In the file processing system it is not possible to access a same file fortransaction at same time

7) Security problems:

There is no security provided in file processing system to secure the data fromunauthorized user access.

Database:A database is organized collection of related data of an organization stored informatted way which is shared by multiple users.The main feature of data in a database are:

1. It must be well organized
2. it is related
3. It is accessible in a logical order without any difficulty
4. It is stored only once

for example:

- consider the roll no, name, address of a student stored in a student file. It is collection of related data with an implicit meaning.
- Data in the database may be persistent, integrated and shared.

Persistent: If data is removed from database due to some explicit request from user to remove.

Integrated:A database can be a collection of data from different files and when any redundancyamong those files are removed from database is said to be integrated data.

Sharing Data: The data stored in the database can be shared by multiple users simultaneously with outaffecting the correctness of data.

Why Database:

In order to overcome the limitation of a file system, a new approach was required.Hence a database approach emerged. A database is a persistent collection of logicallyrelated data. The initial attempts were to provide a centralized collection of data. Adatabase has a self-describing nature. It contains not only the data sharing and integrationof data of an organization in a single database.

A small database can be handled manually but for a large database and havingmultiple users it is difficult to maintain it, In that case a computerized database is useful.The advantages of database system over traditional, paper based methods of recordkeeping are:

- compactness: No need for large amount of paper files
- speed: The machine can retrieve and modify the data more faster way then human being
- Less drudgery: Much of the maintenance of files by hand is eliminated
- Accuracy: Accurate, up-to-date information is fetched as per requirement of the user at any time.

File Management System	Database Management System
File System is a general, easy-to-use system to store general files which require less security and constraints.	Database management system is used when security constraints are high.
Data Redundancy is more in file management system.	Data Redundancy is less in database management system.
Data Inconsistency is more in file system.	Data Inconsistency is less in database management system.
Centralization is hard to get when it comes to File Management System.	Centralisation is achieved in Database Management System.
User locates the physical address of the files to access data in File Management System.	In Database Management System, user is unaware of physical address where data is stored.
Security is low in File Management System.	Security is high in Database Management System.
File Management System stores unstructured data as isolated data files/entities.	Database Management System stores structured data which have well defined constraints and interrelation.

Database Management System (DBMS):

A database management system consists of collection of related data and refers to a set of programs for defining, creation, maintenance and manipulation of a database.

Function of DBMS:

1. Defining database schema: it must give facility for defining the database structure also specifies access rights to authorized users.
2. Manipulation of the database: The dbms must have functions like insertion of record into database updation of data, deletion of data, retrieval of data
3. Sharing of database:The DBMS must share data items for multiple users by maintaining consistency of data.
4. Protection of database: It must protect the database against unauthorized users.
5. Database recovery: If for any reason the system fails DBMS must facilitate data base recovery.

Advantages of DBMS:

Reduction of redundancies:

Centralized control of data by the DBA avoids unnecessary duplication of data andeffectively reduces the total amount of data storage required avoiding duplication in theelimination of the inconsistencies that tend to be present in redundant data files.

Sharing of data:

A database allows the sharing of data under its control by any number of applicationprograms or users.

Data Integrity:

Data integrity means that the data contained in the database is both accurate andconsistent. Therefore data values being entered for storage could be checked to ensurethat they fall within a specified range and are of the correct format.

Data Security:

The DBA who has the ultimate responsibility for the data in the dbms can ensure thatproper access procedures are followed including proper authentication schemas for accessto the DBS and additional check before permitting access to sensitive data.

Conflict resolution:

DBA resolve the conflict on requirements of various user and applications. The DBAchooses the best file structure and access method to get optional performance for theapplication.

Data Independence:

Data independence is usually considered from two points of views; physically dataindependence and logical data independence.

- Physical data Independence allows changes in the physical storage devices or organization of the files to be made without requiring changes in the conceptual

view or any of the external views and hence in the application programs using the data base.

- Logical data independence indicates that the conceptual schema can be changed without affecting the existing external schema or any application program.

Disadvantage of DBMS:

1. DBMS software and hardware (networking installation) cost is high
2. The processing overhead by the dbms for implementation of security, integrityand sharing of the data.
3. centralized database control
4. Setup of the database system requires more knowledge, money, skills, and time.
5. The complexity of the database may result in poor performance.

Applications of DBMS

Below are the popular database system applications:

Sector	Use of DBMS
Banking	For customer information, account activities, payments, deposits, loans, etc.
Airlines	For reservations and schedule information.
Universities	For student information, course registrations, colleges and grades.
Telecommunication	It helps to keep call records, monthly bills, maintaining balances, etc.
Finance	For storing information about stock, sales, and purchases of financial instruments like stocks and bonds.
Sales	Use for storing customer, product & sales information.
Manufacturing	It is used for the management of supply chain and for tracking production of items. Inventories status in warehouses.
HR Management	For information about employees, salaries, payroll, deduction, generation of paychecks, etc.

Types of DBMS

Types of DBMS

The main Four Types of Database Management System are:

- Hierarchical database
- Network database
- Relational database
- Object-Oriented database

RDBMS stands for *Relational Database Management System.*
All modern database management systems like SQL, MS SQL Server, IBM DB2, ORACLE, My-SQL, and Microsoft Access are based on RDBMS.
It is called Relational Database Management System (RDBMS) because it is based on the relational model introduced by E.F. Codd.

Difference between DBMS and RDBMS

Although DBMS and RDBMS both are used to store information in physical database but there are some remarkable differences between them.
The main differences between DBMS and RDBMS are given below:

DBMS	RDBMS
DBMS applications store data as file.	RDBMS applications store data in a tabular form
In DBMS, data is generally stored in either a hierarchical form or a navigational form.	In RDBMS, the tables have an identifier called primary key and the data values are stored in the form of tables.
Normalization is not present in DBMS.	Normalization is present in RDBMS.
DBMS does not apply any security with regards to data manipulation.	RDBMS defines the integrity constraint for the purpose of ACID (Atomocity, Consistency, Isolation and Durability) property.
DBMS uses file system to store data, so there will be no relation between the tables.	In RDBMS, data values are stored in the form of tables, so a relationship between these data values will be stored in the form of a table as well.
DBMS has to provide some uniform methods to access the stored information.	RDBMS system supports a tabular structure of the data and a relationship between them to access the stored information.
DBMS does not support distributed database.	RDBMS supports distributed database.
DBMS is meant to be for small organization and deal with small data. it supports single user.	RDBMS is designed to handle large amount of data. it supports multiple users.
Examples of DBMS are file systems, xml etc.	Example of RDBMS are mysql, postgre, sql server, oracle etc.

After observing the differences between DBMS and RDBMS, you can say that RDBMS is an extension of DBMS. There are many software products in the market today who are compatible for both DBMS and RDBMS. Means today a RDBMS application is DBMS application and vice-versa.

Database users

Naive users :

Users who need not be aware of the presence of the database system or any othersystem supporting their usage are considered naïve users . A user of an automatic tellermachine falls on this category.

Online users :

These are users who may communicate with the database directly via an onlineterminal or indirectly via a user interface and application program. These users areaware of the database system and also know the data manipulation language system.

Application programmers :

Professional programmers who are responsible for developing application programsor user interfaces utilized by the naïve and online user falls into this category.

Database Administration (Database Administrator) :

A person who has central control over the system is called database administrator .

The function of DBA are :

1. creation and modification of conceptual Schemadefinition
2. Implementation of storage structure and access method.
3. schema and physical organization modifications .
4. granting of authorization for data access.
5. Integrity constraints specification.
6. Execute immediate recovery procedure in case of failures
7. ensure physical security to database

Data Models

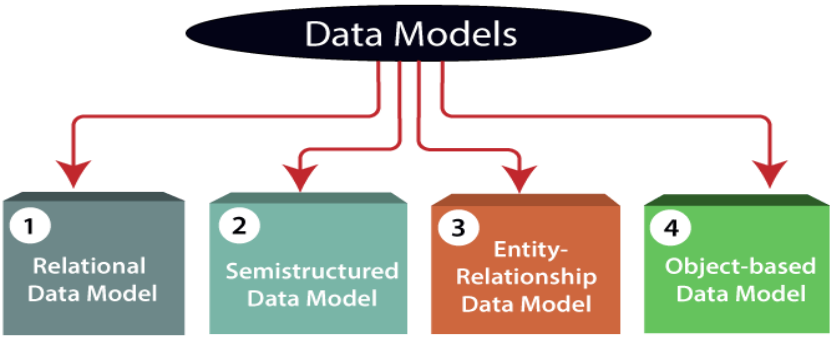
A data model is a conceptual tool that can be used to describe the structure ofdatabase. A database model instants of

I. Description of data type's relationship & constraints.

II. Set of basic operation for access & update database.

III. Specification of a set of valid users to define operation on database.

Types of data models :-



1. **Relational Data Model**: This type of model designs the data in the form of rows and columns within a table. Thus, a relational model uses tables for representing data and in-between relationships. Tables are also called relations. This model was initially

described by Edgar F. Codd, in 1969. The relational data model is the widely used model which is primarily used by commercial data processing applications.

2. **Entity-Relationship Data Model**: An ER model is the logical representation of data as objects and relationships among them. These objects are known as entities, and relationship is an association among these entities. This model was designed by Peter Chen and published in 1976 papers. It was widely used in database designing. A set of attributes describe the entities. For example, student_name, student_id describes the 'student' entity. A set of the same type of entities is known as an 'Entity set', and the set of the same type of relationships is known as 'relationship set'.
3. **Object-based Data Model**: An extension of the ER model with notions of functions, encapsulation, and object identity, as well. This model supports a rich type system that includes structured and collection types. Thus, in 1980s, various database systems following the object-oriented approach were developed. Here, the objects are nothing but the data carrying its properties.
4. **Semistructured Data Model**: This type of data model is different from the other three data models (explained above). The semistructured data model allows the data specifications at places where the individual data items of the same type may have different attributes sets. The Extensible Markup Language, also known as XML, is widely used for representing the semistructured data. Although XML was initially designed for including the markup information to the text document, it gains importance because of its application in the exchange of data.

Some of the types of Data Model: -

1. Hierarchical Model
2. Network Model
3. Entity-Relational Model
4. Relational Model
5. Object-Oriented Model
6. Semi-structured Model
7. Associative Model
8. Entity-Attribute-Value (EAV) Model
9. Context Model

Instance :

The data which is stored in the database at a particular moment of time is called an instance of the database.

Data Independence

- Data independence can be explained using the three-schema architecture.

- Data independence refers characteristic of being able to modify the schema at one level of the database system without altering the schema at the next higher level.

There are two types of data independence:

1. Logical Data Independence

- Logical data independence refers characteristic of being able to change the conceptual schema without having to change the external schema.
- Logical data independence is used to separate the external level from the conceptual view.
- If we do any changes in the conceptual view of the data, then the user view of the data would not be affected.
- Logical data independence occurs at the user interface level.

2. Physical Data Independence

- Physical data independence can be defined as the capacity to change the internal schema without having to change the conceptual schema.
- If we do any changes in the storage size of the database system server, then the Conceptual structure of the database will not be affected.
- Physical data independence is used to separate conceptual levels from the internal levels.
- Physical data independence occurs at the logical interface level.

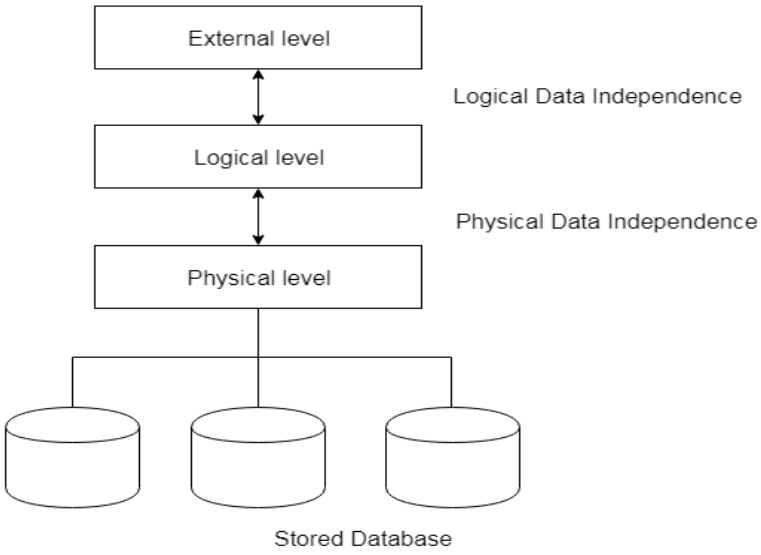


Fig: Data Independence

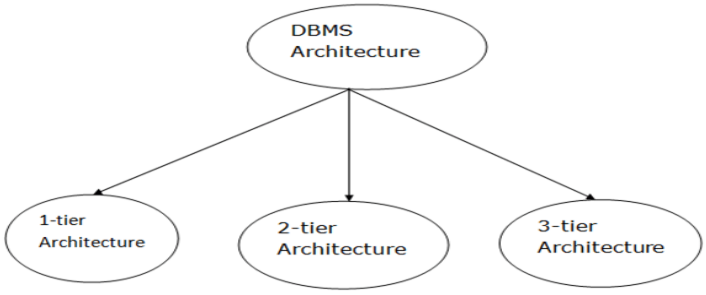
DBMS Architecture

The DBMS design depends upon its architecture. The basic client/server architecture is used to deal with a large number of PCs, web servers, database servers and other components that are connected with networks.

The client/server architecture consists of many PCs and a workstation which are connected via the network.

DBMS architecture depends upon how users are connected to the database to get their request done.

Types of DBMS Architecture

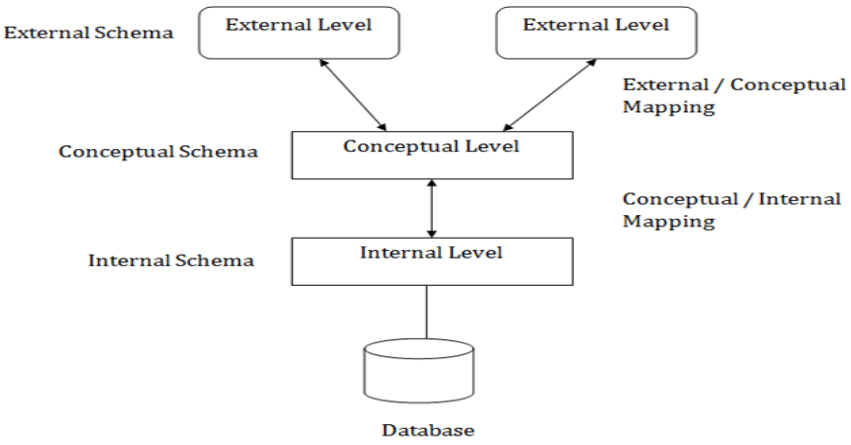


The goal of the three-schema architecture, illustrated in the below Figure, is to separate the user applications and the physical database. In this architecture, schemas can be defined at the following three levels:

1. The internal level has an internal schema, which describes the physical storage structure ofthe database. The internal schema uses a physical data model and describes the completedetails of data storage and access paths for the database.

2. The conceptual level has a conceptual schema, which describes the structure of the wholedatabase for a community of users. The conceptual schema hides the details of physicalstorage structures and concentrates on describing entities, data types, relationships, useroperations, and constraints. A high-level data model or an implementation data model can beused at this level.

3. The external or view level includes a number of external schemas or user views. Eachexternal schema describes the part of the database that a particular user group is interested inand hides the rest of the database from that user group. A high-level data model or animplementation data model can be used at this level.



The three-schema architecture is as follows:

Structure of a DBMS

A database system is partitioned into modules that deal with each of the responsibilities of the overall system. The functional components of a database system can be broadly divided into the storage manager and the query processor components. The storage manager is important because databases typically require a large amount of storage space. The query processor is important because it helps the database system simplify and facilitate access to data.

It is the job of the database system to translate updates and queries written in a nonprocedural language, at the logical level, into an efficient sequence of operations at the physical level.

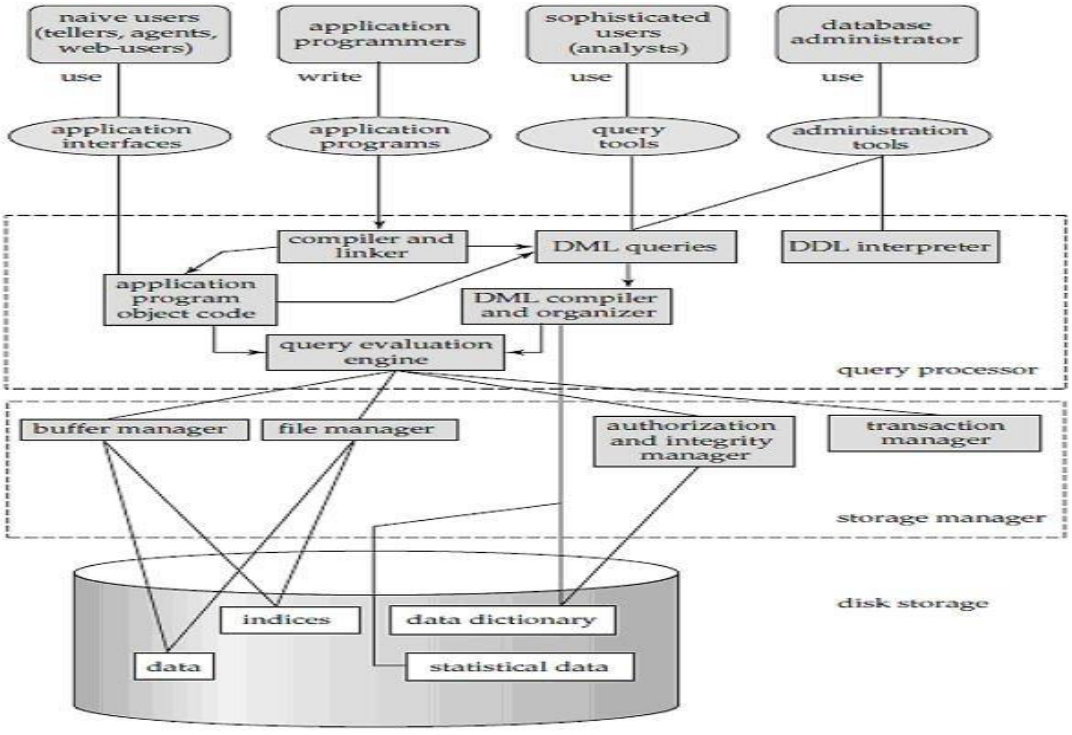


Figure: Database System Architecture

Query Processor

The query processor components include

- DDL interpreter, which interprets DDL statements and records the definitions in the data dictionary.
- DML compiler, which translates DML statements in a query language into an evaluation plan consisting of low-level instructions that the query evaluation engine understands.

A query can usually be translated into any of a number of alternative evaluation plans that all give the same result. The DML compiler also performs **query optimization**, that is, it picks the lowest cost evaluation plan from among the alternatives.

- **Query evaluation engine**, which executes low-level instructions generated by the DML compiler.

Storage Manager

A storage manager is a program module that provides the interface between the low-level data stored in the database and the application programs and queries submitted to the system. The storage manager is responsible for the interaction with the file manager. The raw data are stored on the disk using the file system, which is usually provided by a conventional operating system. The storage manager translates the various DML statements into low-level file-system commands. Thus, the storage manager is responsible for storing, retrieving, and updating data in the database.

The storage manager components include:

- **Authorization and integrity manager**, which tests for the satisfaction of integrity constraints and checks the authority of users to access data.
- **Transaction manager**, which ensures that the database remains in a consistent (correct) state despite system failures, and that concurrent transaction executions proceed without conflicting.
- **File manager**, which manages the allocation of space on disk storage and the data structures used to represent information stored on disk.
- **Buffer manager**, which is responsible for fetching data from disk storage into main memory, and deciding what data to cache in main memory. The buffer manager is a critical part of the database system, since it enables the database to handle data sizes that are much larger than the size of main memory.

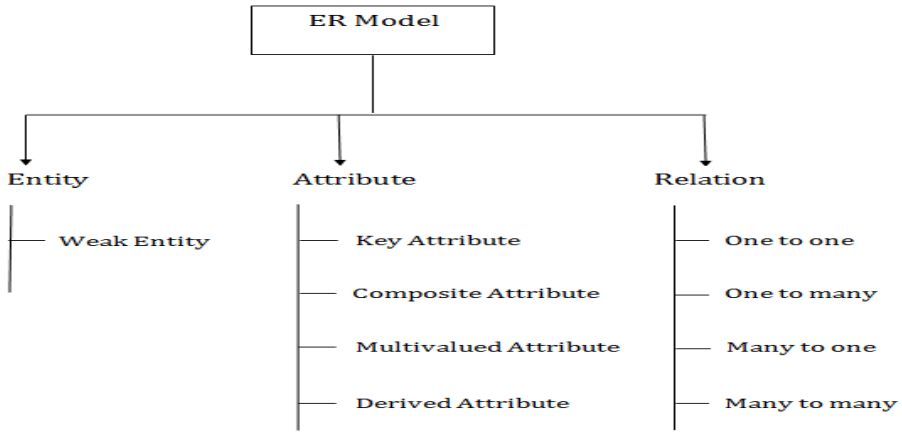
Transaction Manager

A transaction is a collection of operations that performs a single logical function in a database application. Each transaction is a unit of both atomicity and consistency. Thus, we require that transactions do not violate any database-consistency constraints. That is, if the database was consistent when a transaction started, the database must be consistent when the transaction successfully terminates. Transaction - manager ensures that the database remains in a consistent (correct) state despite system failures (e.g., power failures and operating system crashes) and transaction failures.

What is ER Modeling?

A graphical technique for understanding and organizing the data independent of the actual database implementation We need to be familiar with the following terms to go further.

Component of ER Diagram

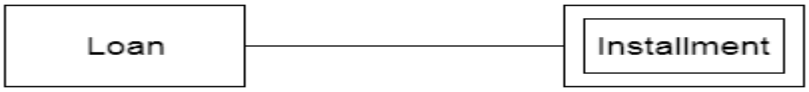


1. Entity:

An entity may be any object, class, person or place. In the ER diagram, an entity can be represented as rectangles. Consider an organization as an example- manager, product, employee, department etc. can be taken as an entity.

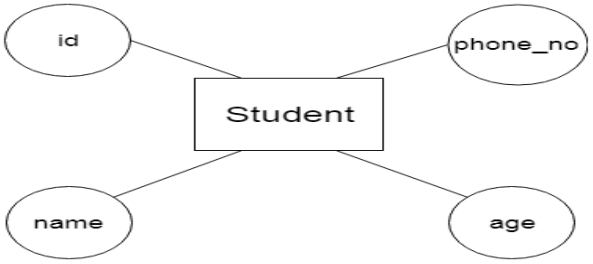
a. Weak Entity

An entity that depends on another entity called a weak entity. The weak entity doesn't contain any key attribute of its own. The weak entity is represented by a double rectangle.



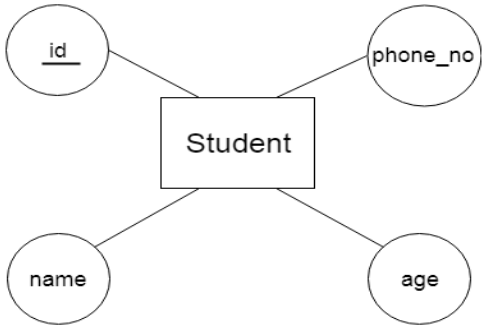
2. Attribute

The attribute is used to describe the property of an entity. Eclipse is used to represent an attribute. For example, id, age, contact number, name, etc. can be attributes of a student.



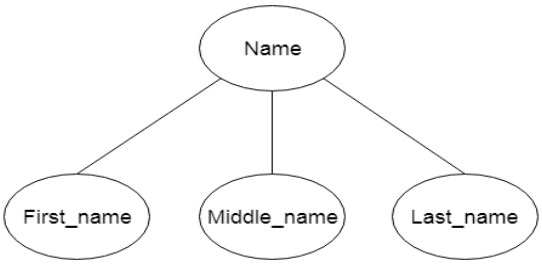
a. Key Attribute

The key attribute is used to represent the main characteristics of an entity. It represents a primary key. The key attribute is represented by an ellipse with the text underlined.



b. Composite Attribute

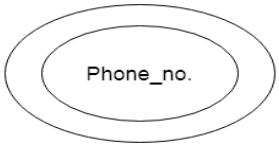
An attribute that composed of many other attributes is known as a composite attribute. The composite attribute is represented by an ellipse, and those ellipses are connected with an ellipse.



c. Multivalued Attribute

An attribute can have more than one value. These attributes are known as a multivalued attribute. The double oval is used to represent multivalued attribute.

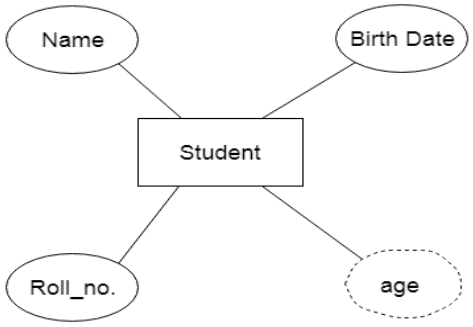
For example, a student can have more than one phone number.



d. Derived Attribute

An attribute that can be derived from other attribute is known as a derived attribute. It can be represented by a dashed ellipse.

For example, A person's age changes over time and can be derived from another attribute like Date of birth.



3. Relationship

A relationship is used to describe the relation between entities. Diamond or rhombus is used to represent the relationship.

Types of relationship are as follows:

a. One-to-One Relationship

When only one instance of an entity is associated with the relationship, then it is known as one to one relationship.

For example, A female can marry to one male, and a male can marry to one female.

b. One-to-many relationship

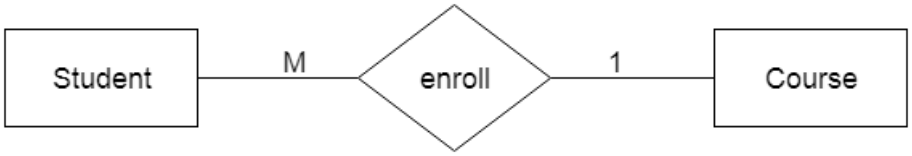
When only one instance of the entity on the left, and more than one instance of an entity on the right associates with the relationship then this is known as a one-to-many relationship.

For example, Scientist can invent many inventions, but the invention is done by the only specific scientist.

c. Many-to-one relationship

When more than one instance of the entity on the left, and only one instance of an entity on the right associates with the relationship then it is known as a many-to-one relationship.

For example, Student enrolls for only one course, but a course can have many students.



Generalization and Specialization

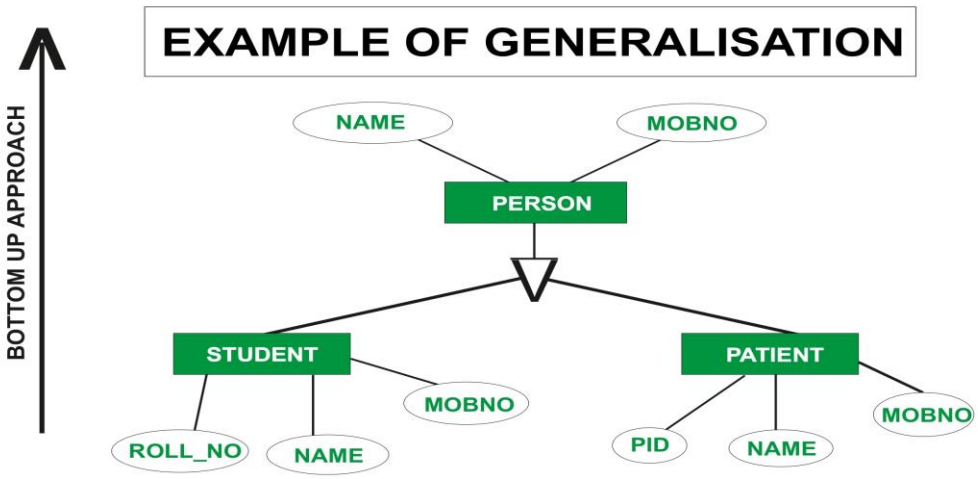
1. Generalization :

It works on the principle of bottom up approach. In Generalization lower level functions are combined to form higher level function which is called as entities. This process is repeated further to make advanced level entities.

In the Generalization process properties are drawn from particular entities and thus we can create generalized entity. We can summarize Generalization process as it combines subclasses to form superclass.

Example of Generalization –

Consider two entities Student and Patient. These two entities will have some characteristics of their own. For example Student entity will have Roll_No, Name and Mob_No while patient will have PId, Name and Mob_No characteristics. Now in this example Name and Mob_No of both Student and Patient can be combined as a Person to form one higher level entity and this process is called as Generalization Process.

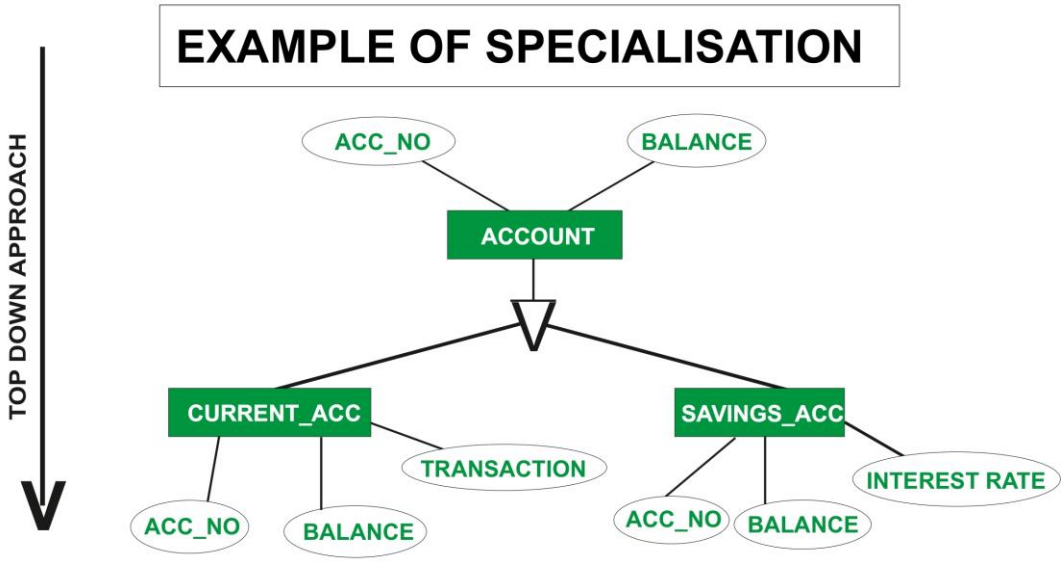


2. Specialization :

We can say that Specialization is opposite of Generalization. In Specialization things are broken down into smaller things to simplify it further. We can also say that in Specialization a particular entity gets divided into sub entities and it's done on the basis of it's characteristics. Also in Specialization Inheritance takes place.

Example of Specialization –

Consider an entity Account. This will have some attributes consider them Acc_No and Balance. Account entity may have some other attributes like Current_Acc and Savings_Acc. Now Current_Acc may have Acc_No, Balance and Transactions while Savings_Acc may have Acc_No, Balance and Interest_Rate henceforth we can say that specialized entities inherits characteristics of higher level entity.



Difference between Generalization and Specialization :

GENERALIZATION	SPECIALIZATION
Generalization works in Bottom-Up approach.	Specialization works in top-down approach.
In Generalization, size of schema gets reduced.	In Specialization, size of schema gets increased.
Generalization is normally applied to group of entities.	We can apply Specialization to a single entity.
Generalization can be defined as a process of creating groupings from various entity sets	Specialization can be defined as process of creating subgrouping within an entity set
In Generalization process, what actually happens is that it takes the union of two or more lower-level entity sets to produce a higher-level entity sets.	Specialization is reverse of Generalization. Specialization is a process of taking a subset of a higher level entity set to form a lower-level entity set.
Generalization process starts with the number of entity sets and it creates high-level entity	Specialization process starts from a single entity set and it creates a different entity set

with the help of some common features.	by using some different features.
In Generalization, the difference and similarities between lower entities are ignored to form a higher entity.	In Specialization, a higher entity is split to form lower entities.
There is no inheritance in Generalization.	There is inheritance in Specialization.

Types of Keys in DBMS

A key refers to an attribute/a set of attributes that help us identify a row (or tuple) uniquely in a table (or relation). A key is also used when we want to establish relationships between the different columns and tables of a relational database. The individual values present in a key are commonly referred to as key values.

Keys are of seven broad types in DBMS:

1. **Candidate Key**
2. **Primary Key**
3. **Foreign Key**
4. **Super Key**
5. **Alternate Key**
6. **Composite Key**
7. **Unique Key**

1. Primary Key: The primary key refers to a column or a set of columns of a table that helps us identify all the records uniquely present in that table. A table can consist of just one primary key. Also, this primary key cannot consist of the same values reappearing/repeating for any of its rows. All the values of a primary key have to be different, and there should be no repetitions.

The PK (PRIMARY KEY) constraint that we put on a column/set of columns won't allow these to have a null value or a duplicate. Any table can consist of only a single primary key constraint. A foreign key (explained below) that refers to it can never change the values present in the primary key.

2. Super Key: A super key refers to the set of all those keys that help us uniquely identify all the rows present in a table. It means that all of these columns present in a table that can identify the columns of that table uniquely act as the super keys. A super key is a candidate key's superset (candidate key has been explained below). We need to pick the primary key of any table from the super key's set so as to make it the table's identity attribute.

3. Candidate Key: The candidate keys refer to those attributes that identify rows uniquely in a table. In a table, we select the primary key from a candidate key. Thus, a candidate key has similar properties as that of the primary keys that we have explained above. In a table, there can be multiple candidate keys.

4. Alternate Key: As we have stated above, any table can consist of multiple choices for the primary key. But, it can only choose one. Thus, all those keys that did not become a primary key are known as alternate keys.

5. Foreign Key: We use a foreign key to establish relationships between two available tables. The foreign key would require every value present in a column/set of columns to match the referential table's primary key. A foreign key helps us to maintain data as well as referential integrity.

6. Composite Key : The composite key refers to a set of multiple attributes that help us uniquely identify every tuple present in a table. The attributes present in a set may not be unique whenever we consider them separately. Thus, when we take them all together, it will ensure total uniqueness.

7. Unique Key: A unique key refers to a column/a set of columns that identify every record uniquely in a table. All the values in this key would have to be unique. Remember that a unique key is different from a primary key. It is because it is only capable of having one null value. A primary key, on the other hand, cannot have a null value.

Why do we require Keys in DBMS?

We use a key for defining various types of integrity constraints in a database. A table, on the other hand, represents a collection of the records of various events for any relation. Now, there might be thousands of these records, and some of these might even be duplicated.

Thus, we need a way in which one can identify all of these records uniquely and separately, i.e., without any duplicates. This hassle is removed with the help of keys.

For example, let us consider a database of all the students who are studying in a college. What attribute of all the students, according to you, will identify each of these people uniquely? We can refer to these students by their names, departments, sections, and year. Similarly, we can also mention only the university roll number and fetch all the other details based on that roll number.

The keys in DBMS can be a combination of multiple attributes (or columns), or they can be just one single attribute. The primary motive of the keys is to provide every record with a unique identity of its own.

Important Questions from Unit – 1 (Question Bank)

1	Define Database Management System? Explain in detail about Database Management System advantages over file management system [14 Marks;CO1;Understand]
2	A)Explain in detail about the three tier schema architecture of DBMS. [7 Marks;CO1;Understand] B)What is the difference between a database schema and a database state? Explain [7 Marks;CO1;Understand]
3	Discuss the main categories of data models. What are the basic differences between the relational model, the object model [14 Marks;CO1;Understand]
4	A)Define DBA? What are the responsibilities of the DBA? [7 Marks;CO1;Understand] B)What is logical data independence and why is it important? [7 Marks;CO1;Understand]
5	A)Explain in detail about the following terms i) Composite attributeii) Multivalued attributeiii) Derived attribute B)Briefly describe the different types of Database users. [7 Marks;CO1;Understand]
6	Discuss in detail about the concepts of E-R model with suitable examples. [14 Marks;CO1;Understand]
7	A).Explain application of Database Management System? [7 Marks;CO1;Understand] B)What is the difference between a candidate key and the primary key for a given relation? What is a super key? [7 Marks;CO1;Understand]
8	A)What is the difference between specialization and generalization? Why do we not display this difference in schema diagrams?. [7 Marks;CO1;Analyze] B)Design an ER diagram for the Students information system taking in account at least four entities. [7 Marks;CO1;Analyze]
9	A)Compare theDatabase Management System and Relational Database Management System [7 Marks;CO1;Analyze]
10	A)Draw and explain the detailed system architecture of DBMS. [14 Marks;CO1;Understand]


LENDI
INSTITUTE OF ENGINEERING & TECHNOLOGY
A Step Ahead Into Tomorrow Careers