GEETHANJALI INSTITUTE OF SCIENCE & TECHNOLOGY

Unit of USHODAYA EDUCATIONAL SOCIETY

An ISO 9001:2015 certified Institution: Recognized under Sec. 2(f)& 12(B) of UGC Act, 1956
3rd Mile, Bombay Highway, Gangavaram (V), Kovur(M), SPSR Nellore (Dt), Andhra Pradesh, India- 524137
Ph. No. 08622-212769, E-Mail: geethanjali@gist.edu.in, Website: www.gist.edu.in

DATABASE MANAGEMENT SYSTEMS

(Common to CSE, AI&ML, DS, CS)

Course Code	L:T:P:S	Credits	Exam Marks	Exam Duration	Course Type
22A0512T	3:0:0:0	3	CIE: 30 SEE:70	3 Hours	PCC

Course Objectives:

This course will enable students to:

- To teach the role of database management system in an organization.
- To design databases using data modeling and Logical database design techniques.
- To construct database queries using relational algebra and calculus and SQL.
- To explore implementation issues in database transaction.
- To familiarize database security mechanisms.

Course Outcomes (CO):

On completion of this course, student will be able to

- Understand the Basic Concepts of Database languages, Relational model, SQL.
- Choose the specific Data models for large enterprise database design.
- Analyze the data efficiently through SQL instructions.
- Apply Normal forms on database for eliminating the redundancy.
- Demonstrate the Basic Concepts of transaction management techniques.
- Apply concurrency control techniques for Database recovery.

Syllabus		Total Hours:48
Module-I	**Introduction to Database concepts &Modeling**	**10Hrs**

Conceptual Modeling Introduction: Introduction to Data bases, Purpose of Database Systems, View of Data, Data Models, Database Languages, Database Users, Database Systems architecture.

The Entity-Relationship Model: Overview of Database Design, Beyond ER Design, Entities, Attributes and Entity sets, Relationships and Relationship sets, Conceptual Design with the ER Model.

Module-II	Relational Model, Relational Algebra	9Hrs

Relational Model: Introduction to the Relational Model – Integrity Constraints over Relations, Enforcing Integrity constraints, querying relational data, Logical data base Design, Views.

Relational Algebra: Introduction to Relational algebra, selection and projection, set operations, renaming, joins, division.

Module-III	SQL	10Hrs

SQL: Basic form of SQL Query, DDL, DML queries, Views in SQL, Joins, Nested & Correlated queries, Operators, predefined functions, Aggregate Functions.

PL/SQL: Introduction, Functions & Procedures, Triggers, Cursors.

Module-IV	Normalization	9Hrs

Relational database design: Introduction, Functional Dependencies (FDs), Normalization for relational databases: 1NF, 2NF, 3NF and BCNF, Basic definitions of Multi Valued Dependencies, 4NF and 5NF.

Module-V	Transaction Management &Concurrency Control and Recovery	10Hrs

Transaction Management: Transaction processing, Transaction Concept, Transaction States, Implementation of Atomicity and Durability, Concurrent Executions.

Concurrency Control: Lock-Based Protocols, Timestamp- Based Protocols, Validation-Based Protocols, Multiple Granularity.

Recovery: Failure Classification, Recovery and Atomicity, Log-Based Recovery.

Text Books:

1. Abraham Silberschatz, Henry F. Korth, S. Sudarshan, Database System Concepts, 6th Edition, Tata McGraw-Hill Publishing Company,2017.
2. Raghu Ramakrishnan, Database Management System, 3rd Edition, Tata McGraw-Hill Publishing Company, 2014.

Reference Books:

1. Peter Rob, A.Ananda Rao, Corlos Coronel, Database Management Systems (for JNTU), Cengage Learning, 2011.
2. Hector Garcia Molina, Jeffrey D. Ullman, Jennifer Widom, Database System Implementation, 1st Edition, Pearson Education, United States, 2000.
3. E. Ramez and Navathe, Fundamental of Database Systems, 7th Edition, Pearson Education
4. R.P. Mahapatra & Govind Verma, Database Management Systems, Khanna Publishing House, 2016.
5. Carlos Coronel and Steven Morris, Database Systems: Design, Implementation, and Management, 12th edition, Cengage Learning,2016.
6. John V. ,Absolute beginner's guide to databases, Petersen, QUE

Web References:

1. https://www.coursera.org/learn/database-management
2. https://www.coursera.org/learn/sql-data-science
3. https://www.w3schools.com/sql/
4. https://www.youtube.com/watch?v=fHAfc7Hjq28&list=PLWPirh4EWFpGrpcMfZ6UcdI786QdtSxV8
5. https://www.youtube.com/watch?v=HwmEcudlv44&list=PL4OCRJojkV1jN-Ed6RkQpWfBvqe0utRd6
6. http://www.w3schools.in/dbms/
7. https://www.geeksforgeeks.org/dbms/
8. https://www.javatpoint.com/dbms-tutorial
9. https://www.edureka.co/blog/dbms-tutorial/

Module - 1 Introduction to Data bases and ER - Model

Introduction:-

Generally The Computers are used to Store the date, Manipulate the Data and Present the Date. From IIIrd Generation of the computer the data storage is played a virtual role for the programmers. They were some personal computers are used to store the data.

Good decisions require good information that is derived from raw facts known as data. Data are likely to be managed most efficiently when they are stored in a data base.

DBMS was the first tool to store the data and manipulated data. It stands for Data Base Management System

---

Data :- Collection of **Raw facts** that have some meaning is called data. It is the heart of DBMS. (OR) Raw facts that is facts that have not yet been processed to reveal their meaning to the end user is called data

Facts are recommended and stored on computer memory facts are texts, graphics, images, sounds and video segments that have meaning in users environment.

Information:- The processed data is called information. Information is also nothing but a collection facts which are arranged in a meaningful manner. The terms data and information are closely related but data is the base component to frame the information

Ex:- Student Information

S.no	Name	Course	Address
1	VENU	DS	Nellore
2	SRUTHI	AIML	Kavali
3	ASRITH	CSE	Tirupathi

Data Vs. Information:- Generally before processing the collection of Raw facts is called data and after processing the data is called information

Ex: Student Data

1	VENU	DS	Nellore
2	SRUTHI	AIML	Kavali
3	ASRITH	CSE	Tirupathi

Ex:- Student Information

S.No	Name	Course	Address
1	XYZ	DS	Nellore
2	PQR	AIML	Kavali
3	STU	CSE	Tirupathi

Meta Data:- Data about data is called Meta data (**OR**) The data concerning the characteristics and relationships of other data is called Meta data

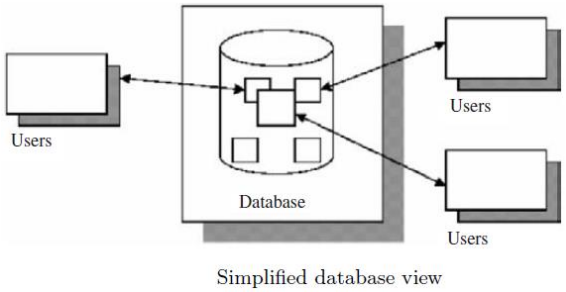
Ex:- Student Information

NAME	**TYPE**	**WIDTH**	**DESCRIPTION**
SNO	Number	5	Student No
SNAME	Text	25	Student Name
ADDRESS	Text	35	Student Address

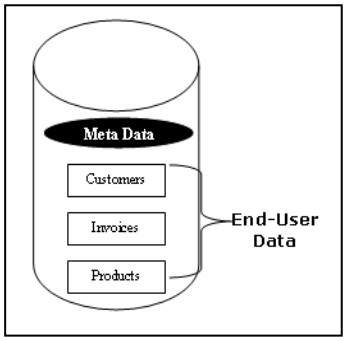
Data Base

A data base is nothing but a collection of files while containing logically related data **(OR)** A Data base is well organized collection of data that are related in a meaning full way.

The primary importance of database is to store huge amount of data into long term. A Data base contains two types of data i.e End-User data (raw facts) and Meta data.



Simplified database view



Here, Several users can access the data from the database in an organization, The database maintain the Data Integrity. So the same information is not recorded in two places.

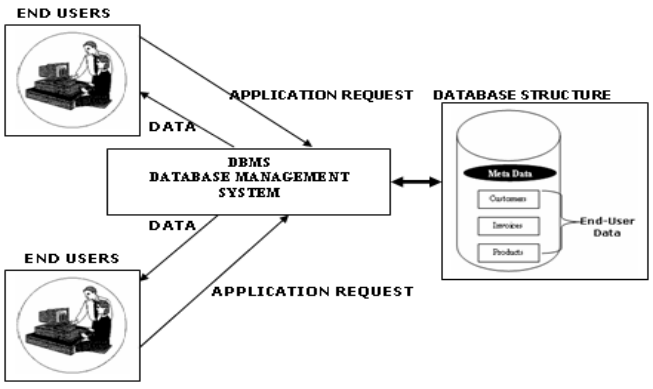
DBMS (Data Base Management System)

DBMS is a collection of inter related files and a set of programs these are used to access DB data. **(OR)** Data base is a collection of programs these programs are used for managing the database.

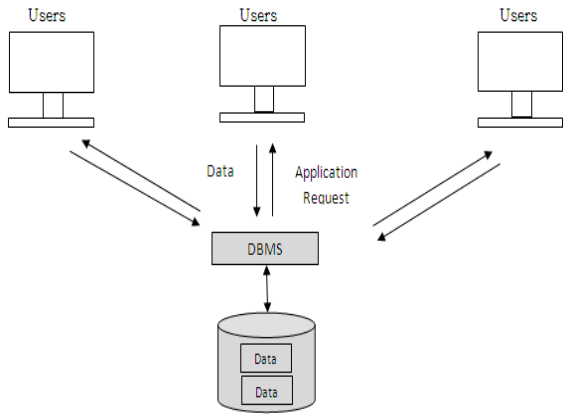
The DBMS is to provide convenient and efficient environment to retrieving and storing data in the data base.

The DBMS allows the users to input data, share the data, Edit the data, Manipulate the data, and Display the data in the database. The DBMS is software, which translates data from its logical representation to its physical representation.

Structure of DBMS



(OR)



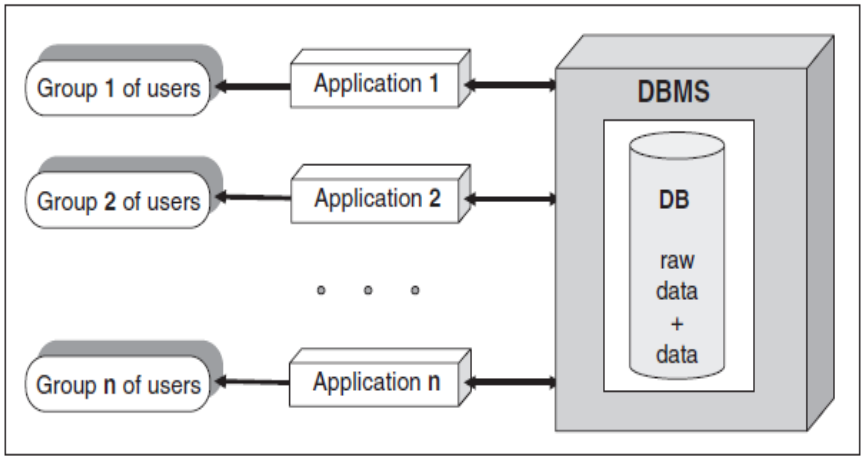
The above figure shows DBMS receives all application requests and translates them into the complex operations required to fulfill those requests. The application program might be written by a programmer using a programming language such as COBAL, VB or C++ or it might be created through a DBMS utility program.

The DBMS helps make data management more efficient and effective.

DBMS Approach

DBMS is a collection of inter related files and a set of programs these are used to access DB data. The DBMS is to provide convenient and efficient environment to retrieving and storing data in the database.

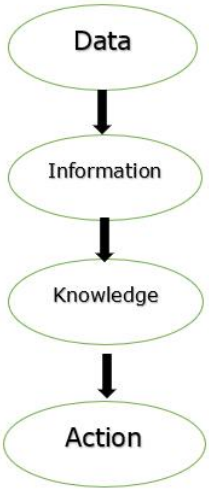
The DBMS is software provides



Data access through DBMS

Purpose of Database System

The **Database Management System** (DBMS) is defined as a software system that allows the user to define, create and maintain the database and provide control access to the data.



It is a collection of programs used for managing data and simultaneously it supports different types of users to create, manage, retrieve, update and store information.

Purpose of DBMS :-

The purpose of DBMS is to transform the following –

- Data into information.
- Information into knowledge.
- Knowledge to the action.

☞ **The Main purpose of DBMS is to overcome the draw backs of File system**

File-Based System

Traditionally, any small organization files were stored into files cabinet manually. But it is a time taking process for retrieving required information, to overcome this problem we use Database system.

The file system is provided by Operating system. In a file system the data was able to store into Computer. Each file is stored as separate application in the computer.

Drawbacks of File-Based System

The following are the drawbacks of file system.

1) Structural dependency
2) Data Dependency
3) Data Redundancy (Or) Duplication of data
4) Data Inconsistency
5) Limited Data sharing

1) Structural dependency Or Incompatible file formats

In file system accessing a file is depends on its structure. For example adding a customer date of birth field to the CUSTOMER file, none of the previous programs will not work properly. So all of the file system application programs must be modified.

2) Data Dependence :-

Data dependence means the application programs depends on data. If we change a field from integer to decimal, then the previous application program has to be rewritten.

3) Duplication of Data (Or) (Data Redundancy):-

Duplication of data means the same data is stored more than once in a computer. (Or) Having a separate copies of same information is called Data redundancy. It is the major drawback of file based approach, because file system follows decentralized approach.

For example, Agent table contains Agent_Id and Customer table contains Agent_Id. So Agent_Id column appears both tables i.e redundancy.

4) Data Inconsistency:-

Data inconsistency occurs becoz of data redundancy. When a data is repeated, modifying only at one location will cause inconsistency.

Ex:- The employee details are stored both department file and employee file. Now the employee changes his contact address in the employee file. The changed address is modified in the employee file but not in the department file. If some important information has to be sent to his contact address from the department then that information will be lost.

5) Limited Data Sharing:-

In a file based system there is no centralized data storage location, so data sharing is not possible. Each application program contains its own personal file. For example the users in accounting department cannot access the sales department files

To solve the above problems DBMS has been invented

Advantages of DBMS

DBMS is a collection of inter related data and a set of programs to access that data. The DBMS is to provide convenient and efficient environment to retrieving and storing data in the database. The DBMS The DBMS provides many advantages when compared to File-Based system.

1. Centralized data management.
2. Data Independence.
3. Improved Data sharing
4. Minimized Data Redundancy
5. Minimized data Inconsistency
6. Improved decision making
7. Improved data security
8. Support for multiple views

1) Centralized Data Management :-

In DBMS all files are stored into one system, thus reducing redundancies and making data management more efficient

2) Data Independence :-

The DBMS is the interface between the application programs and the data base data. Data independence means if any changes are made on the data, then the previous application programs may need not to be rewritten. Data independence can be Physical data independence (or) Logical data independence.

Ex:- If we add a new attribute in to existing data, then these changes are affected on application programs in the file processing system. But in a DBMS environment these changes are not affected on application programs.

3) Improved Data Sharing:- The database is designed as a shared resource to authorized users. The users can be granted permissions to other users to use the database.

4) Minimized data redundancy: The database approach does not completely eliminates the redundancy. Because database approach can stored the information in centralized location. So that data redundancy reduced upto minimum level.

Ex:- If we take two tables EMP & DEPT tables. We can observe that there is a common field 'Deptno'. It is necessary to maintain this much amount of redundancy to maintain the relations between the tables.

5) Minimized Data Inconsistency:- Data inconsistency exists when the same data appear in different places. By eliminating the redundancy the data inconsistency will also be reduced.

Ex:- If a Customer address is stored only once, then we can update the data very easily. Finally by removing the redundancy, we can also save the memory.

6) Improved Decision Making:- In Database environment better managed data and improved data access generates better quality information. It helps to end users take better decisions

7) Improved Data Security:- In database approach the database developers provides so many security features to database objects (tables). So in database approach security is very high compared to file system.

8) Support for multiple views :- Support for multiple views means DBMS allows different users to see different "views" of the database data, according to their requires. This concept is used to enhance the security of the database

Application and Uses of Database Management System (DBMS)

Now a days many companies are wants to store and retrieve data in the database for future reference. So some of the applications and uses of database management system (DBMS) as shown below**.**

Railway Reservation System

Database is required to keep record of ticket booking, train's departure and arrival status. Also if trains get late then people get to know it through database update.

Library Management System

There are thousands of books in the library so it is very difficult to keep record of all the books in a copy or register. So DBMS used to maintain all the information relate to book issue dates, name of the book, author and availability of the book.

Banking

We make thousands of transactions through banks daily and we can do this without going to the bank. So how banking has become so easy that by sitting at home we can send or get money through banks. That is all possible just because of DBMS that manages all the bank transactions.

Universities and colleges

Examinations are done online today and universities and colleges maintain all these records through DBMS. Student's registrations details, results, courses and grades all the information are stored in database

Credit card transactions

For purchase of credit cards and all the other transactions are made possible only by DBMS. A credit card holder knows the importance of their information that all are secured through DBMS.

Social Media Sites

We all are on social media websites to share our views and connect with our friends. Daily millions of users signed up for these social media accounts like facebook, twitter, and Google plus. But how all the information of users are stored and how we become able to connect to other people, yes this all because DBMS.

Telecommunications

Any telecommunication company cannot even think about their business without DBMS. DBMS is must for these companies to store the call details and monthly post paid bills.

Finance

Those days have gone far when information related to money was stored in registers and files. Today the time has totally changed because there are lots f thing to do with finance like storing sales, holding information and finance statement management etc.

Military

Military keeps records of millions of soldiers and it has millions of files that should be keep secured and safe. As DBMS provides a big security assurance to the military information so it is widely used in militaries. One can easily search for all the information about anyone within seconds with the help of DBMS.

Online Shopping

Online shopping has become a big trend of these days. No one wants to go to shops and waste his time. Everyone wants to shop from home. So all these products are added and sold only with the help of DBMS. Purchase information, invoice bills and payment, all of these are done with the help of DBMS.

Human Resource Management

Big firms have many workers working under them. Human resource management department keeps records of each employee's salary, tax and work through DBMS.

Manufacturing

Manufacturing companies make products and sales them on the daily basis. To keep records of all the details about the products like quantity, bills, purchase, supply chain management, DBMS is used.

Airline Reservation system

Same as railway reservation system, airline also needs DBMS to keep records of flights arrival, departure and delay status.

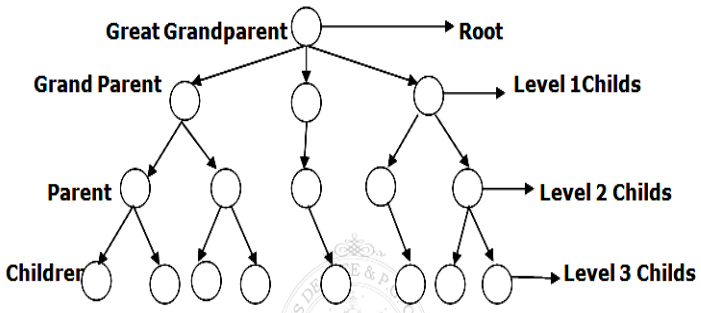
Data Models

Data Model :- Documenting rules and policies of an organization is called Data Modeling. Data Modeling is the first step in designing a database. The following are the Data models.

1) Hierarchical Model
2) Network Model
3) Relational Model
4) ER-Model
5) Object-oriented Model

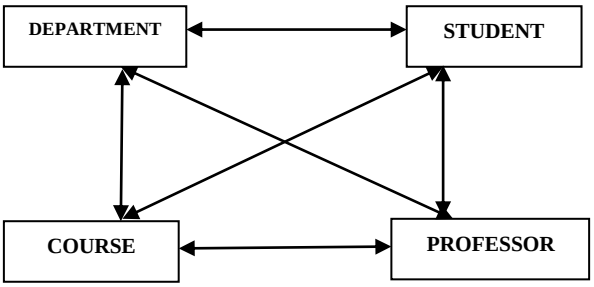
1) Hierarchical Data Model :-

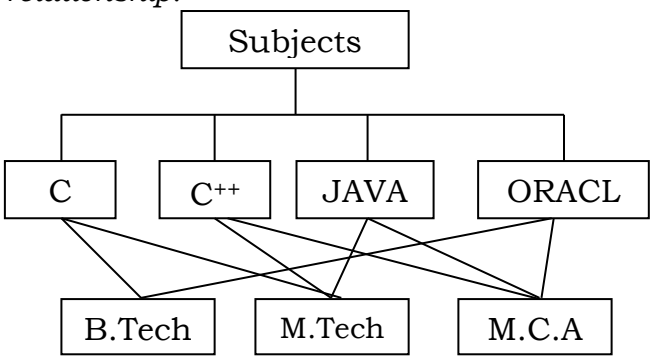
- The Hierarchical Model was developed in 1960 to manage large amount of data.
- The Hierarchical Model organizes the data in a "**Tree Structure**".
- In this model, the data is arranged in Top-down approach.
- The data element in the Top-level will be called as "***Root***".
- The data elements in the remaining levels will be called as "***Childs***".
- The parent data element can have many child data elements, but a child has only one parent. This is known as *One-to-Many relationship*.



2) Network Model :-

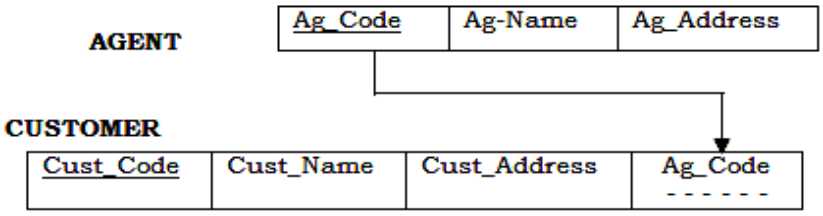
- The Network Model was developed to represent complex data.
- This is powerful model and extension of Hierarchical model.
- In this model, the data is arranged in the form of Graph.
- Hence, a parent data element can have many child data elements.
- A child data element can have many parent data elements.
- The Network Model supports *Many-to-Many relationship*.





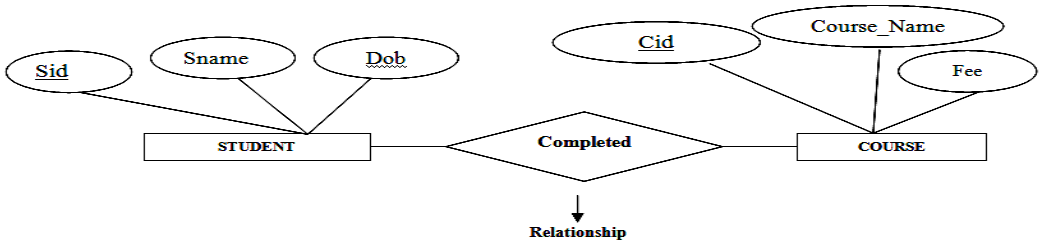
3. Relational Model:-

- The Relational Model was developed by Dr.E.F.Codd in 1970.
- It is the most common model to represent the data in the database applications.
- It supports different types of relationships
- It uses referential integrity constraints to establish relationship between tables.
- In this, tables are called ***relations***, columns are called ***fields***, and rows are called ***tuples***
- In this model the most widely used language is ***SQL***

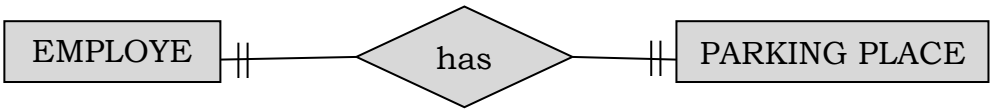


4. Entity Relation ship Model (ER):-

- The Entity Relationship Model is a detailed graphical representation of data for an org.
- An ER-Model (ERM) is normally represented with ER-Diagrams (ERD).
- The ER-Model contains the components like Entities, Attributes, and Relationships._By using these components, we can draw ER-Diagrams.

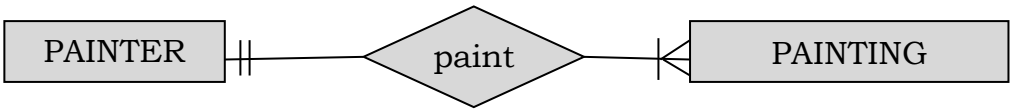


Ex (1): One – to – One Relation Ship (1 : 1)



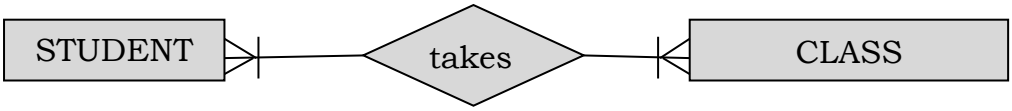
Each *Employee* has one *Parking place*; Each *Parking place* is allotted by one *Employee*.

Ex (2): One-to-Many Relation Ship (1 : M)



A *Painter* paints many *Paintings*; Each *Painting* is painted by one *Painter*.

Ex (3): Many – to – Many Relation Ship (M : M)

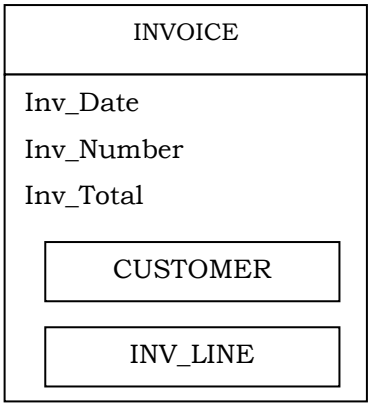


Each *Student* takes many *Classes*; Each *Class* contains many *Students*.

5. Object Oriented Model:-

- In this model, both data and relationships are represented in a single structure called an ***Object***.
- The main advantage of this model is to represent complex data such as video, audio, and graphics etc.
- In this model, an object is represented in a box, all the object attributes and relationships are included in that box

Ex:-

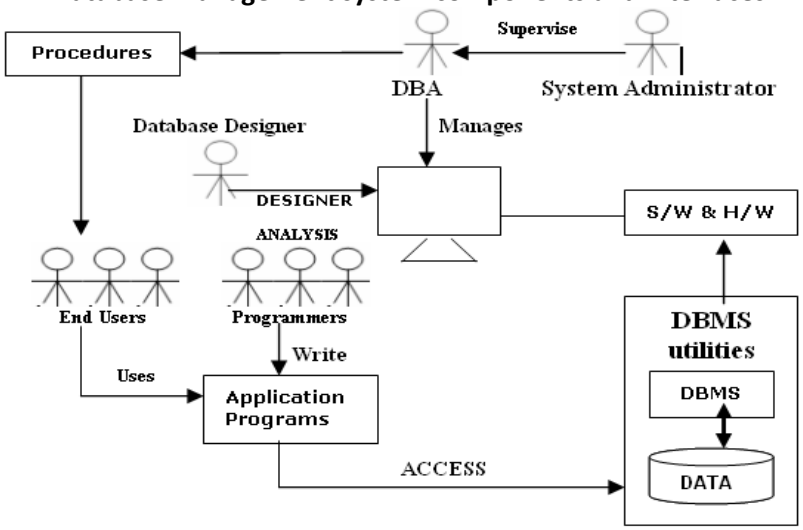


Database Environment & Data Base Users

The database system refers to an organization some components. Those components are used for collection, storage, management and use of data with in a database environment. The database system is components are 5 types. They are

1) Hardware
2) Software
3) **People / Users**
4) Procedures
5) Data

Database management system components and interfaces



1) Hardware:- Hardware refers to all of the system's physical devices. For example Computers (Micro, Mainframe, Workstations) Storage devices, Printers, Network devices etc.

2) Software:- Software is a set of programs Or Collection of programs it is used for managing Hardware. There are three types of Software's are needed in data base environment they are

1) Operating System
2) DBMS Software
3) Application Programs and Utilities Software

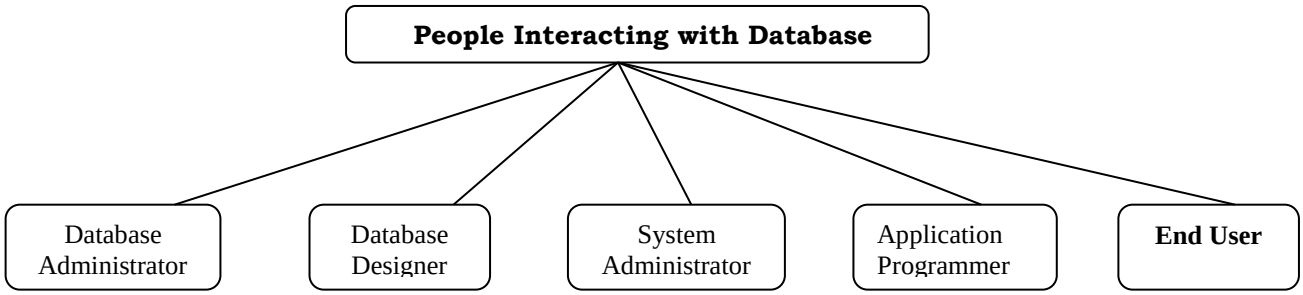
3) Procedures :-

- Procedures are the rules and instructions.
- The procedure may contain information about the DBMS, start and stop the DBMS.
- Procedure on how to identify the failed component, how to recover the database

4) Data:-

- Data is collection of raw facts stored in database
- The data in the database is related
- The data are accessible in database is without difficulty

5) People :- This component includes all users of the database system. Five types of users can be identified in a database system.



i) Database Administrator:-

- These users manages the DBMS and ensure the functionality of the database
- These users to control the database environment and associated software
- DBA also monitors the recovery and backup and provide technical support.
- DBA repairs damage caused due to hardware and/or software failures.

ii) Database Designer:-

- Database designer can be either logical database designer or physical database designer.
- Logical database designer is concentrate on identifying the data, relationships and constraints.
- The physical database designer takes the logical data and decides where it can be physically implemented (Stored).

iii) System Administrator:-

- These users officially supervises the database system's general operations.

iv) Application programmers:-

- These users design and develop the application programs
- Application programmers write application programs and interacts with the data base through host Language like Pascal, C and Cobol

iv) End Users:-

- Who use the application programs to run the organizations daily operation are called End Users.

v) System Analyst :

- System Analyst is a user who analyzes the requirements of parametric end users. They check whether all the requirements of end users are satisfied.

vi) Sophisticated Users :

- Sophisticated users can be engineers, scientists, business analyst, who are familiar with the database.
- These users can develop their own database applications according to their requirement.
- They don't write the program code but they interact the database by writing SQL queries directly through the query processor.

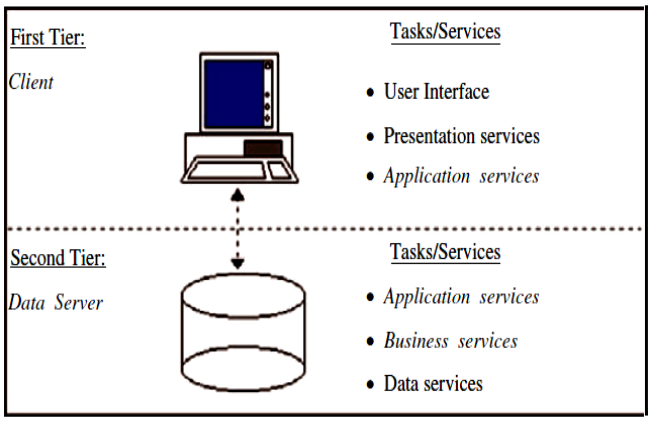
Database System Architecture in DBMS

The database architecture can be classified into 3 types. They are

1. Two- Tire Architecture
2. Three-Tire Architecture
3. Multitier Architecture

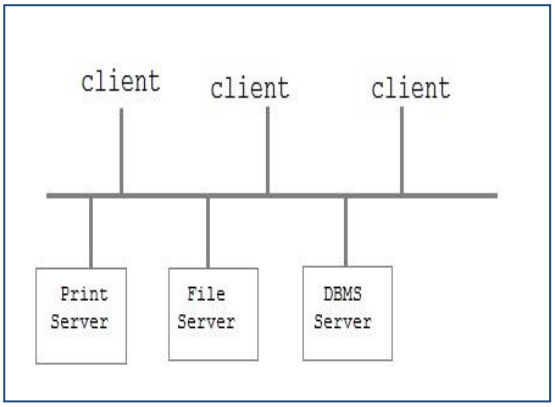
1. Two- Tire Architecture:-

1. The two-tier architecture is a ***Client–Server*** architecture,
2. This architecture the ***Client*** contains the presentation code & SQL queries for data access.
3. The **Database server** processes the SQL statements and sends results back to the client.
4. The Two-tier client/server provides a basic separation of tasks.
5. The "Client" Or First tier, is responsible for the ***presentation*** of data to the user
6. The "Server" Or Second tier, is responsible for supplying ***data services*** to the client



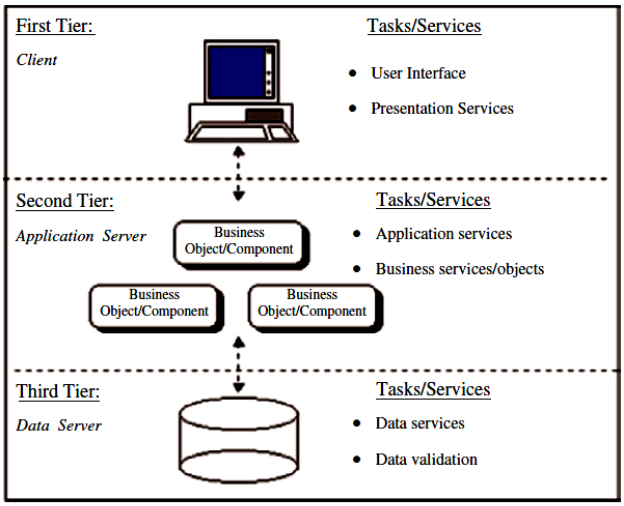
Two-tier client–server architecture

Or



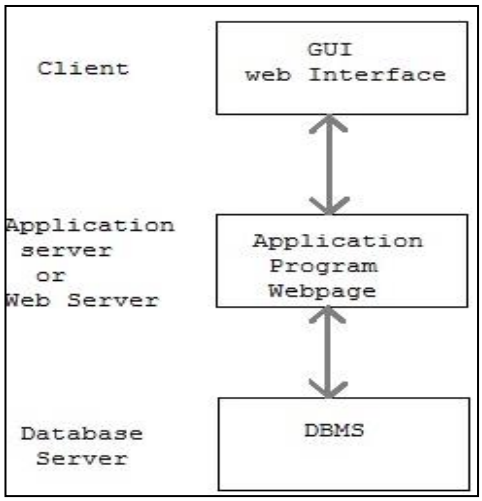
2. Three-Tier architecture

1. The Three-tier" or "N-tier," architecture provides greater application scalability, lower maintenance, and increased reuse of components.
2. Three-tier architecture is commonly used for web applications
3. The Application server stores the web connectivity software and Business services. which are used to access the right amount of data from the database server. This layer acts like medium for sending data between the database server and the client.



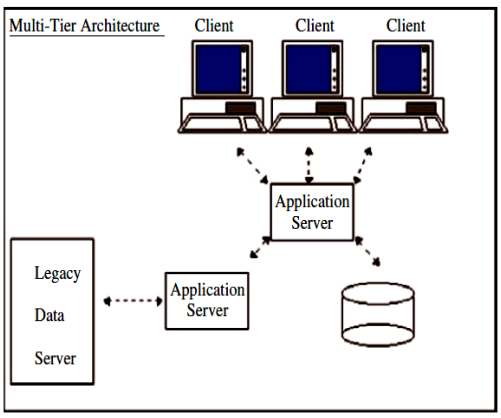
Three-tier client–server architecture

Or



3. Multi-Tier architecture

1. The Multi-Tier architecture is the most general client–server architecture.
2. The implementation of this architecture most difficult.
3. Multi-Tier architecture can provide the most benefits in terms of scalability, flexibility, and interoperability.



Multiple-tier architecture

Ex:- The following diagram the client sends a request to *Application Server #1* for accessing main frame application. But the *Application Server #1* doesn't directly access the mainframe application, So, on that time the *Application Server #2* provides the services to access the data from the mainframe application, which satisfies the client request.

Database View / Abstraction

The main purpose of data abstraction is to hide irrelevant data and provide an abstract view of the data. With the help of data abstraction, developers hide irrelevant data from the user and provide them the relevant data. By doing this, users can access the data without any hassle, and the system will also work efficiently.

The main objective of DBMS is to store and retrieve information efficiently. But it is not necessary for the users to know physical database storage details. The developers hide the complexity from users through 3 levels of abstraction. They are

1. Physical level schema Or Internal level
2. Logical level schema Or Conceptual level
3. External Schema Or View level

1) Physical Level: *How (or) Where the Data is actually Stored in DB*

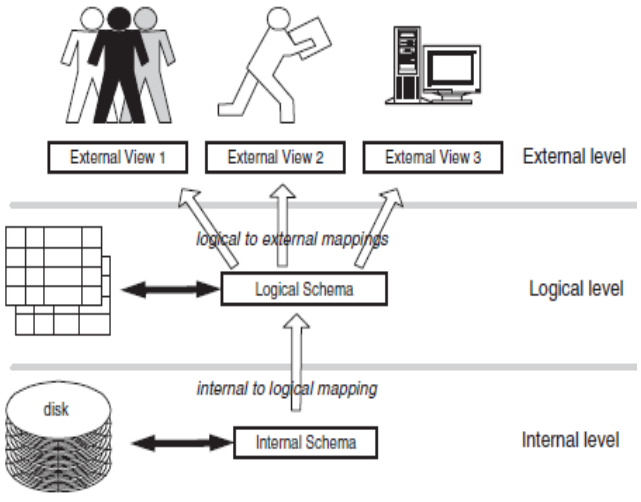
It is lowest level of abstraction, The physical level ***contains storage of the information***. It provides the internal view of the actual physical storage of data. It displays low-level in the architecture. It defines complex data structures in detail, so it is very complex to understand, which is why it is kept hidden from the end user.

2) Logical Level :- *What Data is stored and what are the relationships them*

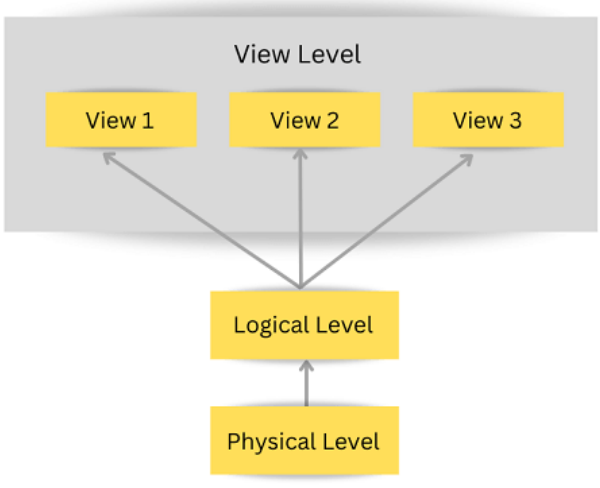
It is next level of abstraction, Logical (Conceptual) level represents ***what type of data are stored in the database***, and what are the relationships exist among those data. The implementation of simple structure in the logical level may involve complex physical level structures. It displays Middle-level in the architecture

3) View / External Level / User Level :- *Who Stores and access Data*

It is a top level of abstraction, It is also known as **User view.** This level is for the end-user interaction; at this level the user to access the data in the database, The user views are like reports, transactions etc.



ANSI/SPARK data model



Levels of Data Abstraction in DBMS

DATABASE LANGUAGES

Database Languages are used to create and maintain database on computer. There are large numbers of databases like Oracle, MySQL, MS Access, dBase, FoxPro etc. SQL statements commonly used in Oracle and MS Access can be categorized as

1. Data definition language (DDL)
2. Data control language (DCL)
3. Data manipulation language (DML)
4. Transaction Control Language (TCL)

DDL Commands

The term DDL means "Data Definition Language". In SQL the DDl commands are used to define, modify and remove the database objects etc. In SQL the DDL commands are classified into different types. They are. **i) Create ii) Alter iii) Drop iv) Truncate v) Describe vi) Rename**

i) Create :- In SQL the database primary object is "table". The table contains rows and columns in which the data is stored. In a table each column is identified by a column name and a given data type with specified width. In SQL the tables are created by using "Create table" command. This command creates only the structure of the table.

Syntax:- Create table <table name>(Columnname1 data type(width), Columnname2 data type(width),….);
Ex:- Create table student(htno number(8), sname varchar2(15), course varchar(5));

STUDENT

Htno	Sname	Course

ii) Alter:- The alter command is used to modify the structure of the existing table.
Syntax:- Alter table <table name> add/modify(column name data type(width),……);

Ex 1:- Alter table student add(dob date);

The above command adds new column "dob" to the Student table.

Ex 2:- Alter table student modify(htno varchar(10));

The above command modify the htno "number" data type to "varchar."

3) Drop:- The drop command is used to remove the database tables permanently

Syntax:- Drop table <table name>;

Ex:- Drop table student;

The above example removes the table "Student"

4) Truncate:- This command is used to remove the all the data permanently from the table, but not structure. **Syntax:-** Truncate table <table name>;

Ex:- Truncate table student;

The above command removes all the data permanently from the table "student" but it does not remove the structure of "Student " table.

5) Describe:- This command is used to display the structure of the table.

Syntax:- Describe <table name>; or Desc <table name>;

Ex:- desc student;

6) Rename:- This command is used to change the database object name from old name to new

Syntax:- Rename <old file name> to <new file name>;

Ex:- Rename student to stud;

DML Commands

The DML means "Data Manipulation Language". The DML commands are used to manipulate data in a database. i.e., The DML commands are used to insert data, modify data, delete data and retrieve data in the table. In SQL the DML commands are classified into mainly 4 types. They are

i) Insert 2) update 3) Delete 4) Select

i) Insert:- This command is used to add records into a table. This command is also used for inserting data into particular columns of table.

Syntax:- insert into <table name> values (val1,val2……;

Ex:- insert into student values(101,'venu','CSE');

The above insert command is used for inserting values into all the columns. For inserting values for a particular columns we have to use the following systax.

Syn:- insert into <table name> (field1,field2,……..) values (val1,val2……);

Ex:- insert into student (sno, sname) values (101,'kumar');

The above insert command is inserts only "sno, sname" values to "student" table.

Syntax for inserting Multiple rows

Syntax:- insert into <table name> values ('&field1','&field2',……..);

Ex:- insert into student values('&sno','&sname','&course');

The above insert command takes the values for "Sno, Sname, Phno" at the time of query execution. This insertion is also called "bulk insertion".

ii) Update:- The update command is used to modify the existing data with new data in a table. This command used "Set" clause.

Syntax:- update<table name> set <field1>=val1, <field2>=val2,……;

Ex:- Update student set course= 'AIML';

The above update command modifies the all records(row) in a specified table.

For modifying only the specified records(rows), we have to use "where" clause.

Syntax :- update <table name> set <field>=val1,<field2>=val2,….. where <condition>;

Ex:- update student set course= 'CSE' where sno=101;

The above command modifies only particular records that will satisfy the condition sno=101 from the table "student".

iii) Drop:- The delete command is used to delete the specific records (or) all records from the table. For deleting a particular record in the table, we have to use "Where" clause.

Systax:- delete from <table name>[where <condition>];

Ex 1:- delete from student;

The above command deletes all rows from the table

Ex 2:-delete from student where sno=101;

The above command deletes specific rows that will satisfy the condition sno=101 fro the table "student"

iv) Select : This command is also called "DQL" command. It is used to retrieve (or) display the data from the table. By using this command and we can select all columns (or) specific columns all rows (or) specific rows from the table.

Syntax:- select *from <table name> [Where <condition>];

The above syntax '*' represents all columns in the table. Here 'where' clause is optional.

Ex 1:- Select *from student;

The above command displays all columns and all rows from the "student" table.

Ex 2:- select *from student where sno=101;

The above command displays all columns and specific rows that will satisfy the condition sno=101;

Ex 3:- Select sno, sname, course from student where sno=101;

The above command displays particular column values i.e, sno, sname, course and specific rows that will satisfy the condition sno= 101 in the table "student".

DCL Commands

The term DCL means "Data control language". The DCL commands are used to control user, access to the database objects. These commands are used to giving and canceling privileges to other users. There are two commands under this category.

i) Grant ii) Revoke

i) Grant:- The Grant command is used to giving all (or) particular privileges to other users. The privileges are insert, delete, update and select commands.

Syntax:- grant <privileges> on <table name> to <username>;

Ex:- Grant select on student to haritha;

The above command provides "select" privilege on student table to the user "Haritha"

After getting privileges "select "from 'scott' user to "Haritha" user, we write the following query for display data from the "student" table

Ex:- select *from scott.student;

ii) Revoke:- The revoke command is used to canceling the all or particular privileges that are given by "grant" command.

Syntax:- Revoke <privileges> on <table name> from <user name>;
Ex:- revoke select on student from haritha;

The above command cancels the "select " privilege on student table from the user "Haritha".

TCL Commands

The term TCL means "Transaction Control Language". A "transaction" is logical unit of work done on the database tables. The transaction may be "simple" or "complex". The simple transaction constraints one DML operation. But the complex transaction contains more than one DML operation by default all the changes can be made temporarily on the database. To make the changes permanently on a data base and to cancel the previously performed transaction, we use TCL commands. In SQL a transaction begins with an SQL executable statement (DML operations) and ends with commit (or) rollback. In SQL, the TCL commands are classified into different types they are

i) Commit ii) Roll back iii) Save point

i) Commit:- It is used to end a transaction. It makes the changes permanently to the database.

Syntax:- Commit; or Commit work;

Ex:-
```
Sql> delete from student where htno=101;
Sql> insert into student values(102, 'jyothi', 'AIML');
Sql> commit;
```

ii) Rollback :- it is used to cancel the work done in the current transaction. The roll back command is used to cancel the entire transaction or particular part of the transaction.

To cancel entire transaction:- **Syntax:-** Rollback or Rollback work;
To cancel part of the transaction:- **Syntax:-** Rollback to save point id;

iii) Save point:- The save points are like the markers, which are used to divide a complex transaction into simple transaction and give a name to each transaction. The save points are always used with Rollback command to cancel particular part of a transaction. In SQL the save points are created by using save point command.

Syntax:- savepoint Savepoint Id;

Ex:-
```
1) SQL> delete from student where course='CSE';
   SQL> roll back;
2) SQL> insert into student values(10,'kumar', 'CSE');
   SQL> savepoint s1;
   SQL> update student set course = 'AIML' where htno=101;
   SQL> roll back to s1;
```

In the above first example, the rollback command cancels the entire transaction. In the second example the rollback command cancels only the update operation.

Application and Uses of Database Management System (DBMS)

Now a days many companies are wants to store and retrieve data in the database for future reference. So some of the applications and uses of database management system (DBMS) as shown below**.**

Railway Reservation System

Database is required to keep record of ticket booking, train's departure and arrival status. Also if trains get late then people get to know it through database update.

Library Management System

There are thousands of books in the library so it is very difficult to keep record of all the books in a copy or register. So DBMS used to maintain all the information relate to book issue dates, name of the book, author and availability of the book.

Banking

We make thousands of transactions through banks daily and we can do this without going to the bank. So how banking has become so easy that by sitting at home we can send or get money through banks. That is all possible just because of DBMS that manages all the bank transactions.

Universities and colleges

Examinations are done online today and universities and colleges maintain all these records through DBMS. Student's registrations details, results, courses and grades all the information are stored in database

Credit card transactions

For purchase of credit cards and all the other transactions are made possible only by DBMS. A credit card holder knows the importance of their information that all are secured through DBMS.

Social Media Sites

We all are on social media websites to share our views and connect with our friends. Daily millions of users signed up for these social media accounts like facebook, twitter, and Google plus. But how all the information of users are stored and how we become able to connect to other people, yes this all because DBMS.

Telecommunications

Any telecommunication company cannot even think about their business without DBMS. DBMS is must for these companies to store the call details and monthly post paid bills.

Finance

Those days have gone far when information related to money was stored in registers and files. Today the time has totally changed because there are lots f thing to do with finance like storing sales, holding information and finance statement management etc.

Military

Military keeps records of millions of soldiers and it has millions of files that should be keep secured and safe. As DBMS provides a big security assurance to the military information so it is widely used in militaries. One can easily search for all the information about anyone within seconds with the help of DBMS.

Online Shopping

Online shopping has become a big trend of these days. No one wants to go to shops and waste his time. Everyone wants to shop from home. So all these products are added and sold only with the help of DBMS. Purchase information, invoice bills and payment, all of these are done with the help of DBMS.

Human Resource Management

Big firms have many workers working under them. Human resource management department keeps records of each employee's salary, tax and work through DBMS.

Manufacturing

Manufacturing companies make products and sales them on the daily basis. To keep records of all the details about the products like quantity, bills, purchase, supply chain management, DBMS is used.

Airline Reservation system

Same as railway reservation system, airline also needs DBMS to keep records of flights arrival, departure and delay status.

Prepared by

Venukumar DVH

M.Sc.,M.Ed.,M.Tech(CSE)

Asst.Professor., Dept.of CSE., Geethanjali Institute of Science and Technology : Gangavaram : Nellore

Entity – Relationship Model

Introduction :-

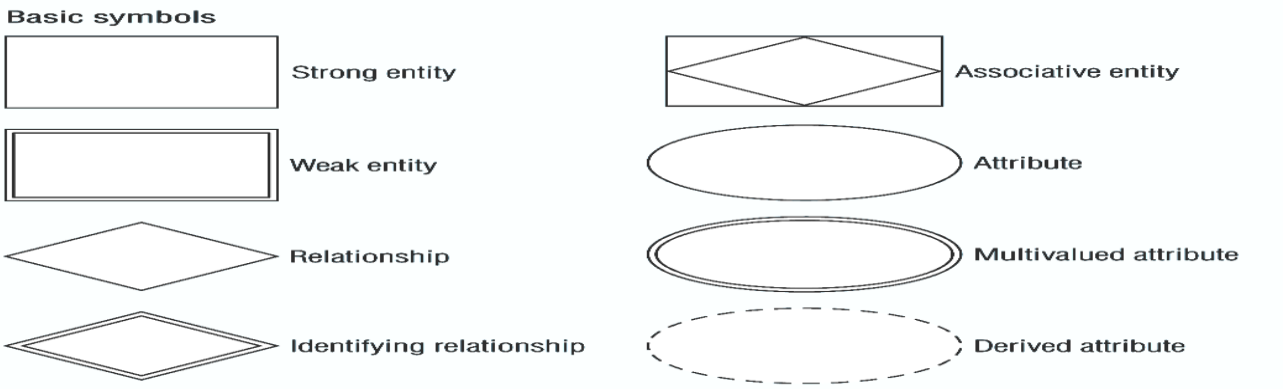
The E-R Model Stands for Entity Relationship Model. It is a detailed graphical representation of the data for an organization. An E-R Model is normally expressed in terms of E-R Diagrams i.e Entities, Relationships and Attributes of both the entities and their relationships.

Basic Building Blocks of E-R Model (OR) Basic Components of E-R Model / Conceptual View of ER Diagrams

E-R Model:- The E-R Model Stands for Entity Relationship Model. It is a detailed graphical representation of the data for an organization. An E-R Model is normally expressed in terms of E-R Diagrams i.e Entities, Relationships and Attributes of both the entities and their relationships.

The Basic Components of E-R Model is

1. Entities
2. Attributes
3. Relationships



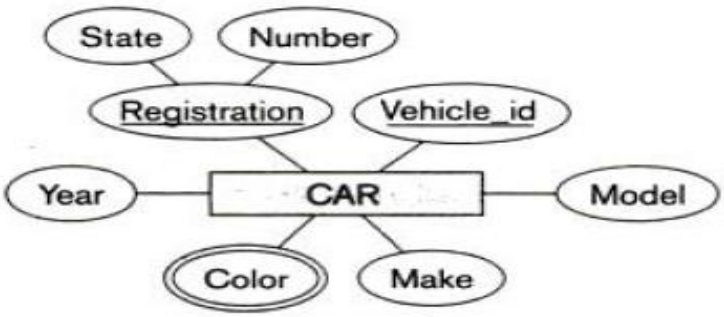
1. Entity:-

Entity is a real world object Or An Entity is a person, place, object, event or concept in the user environment about which the organization wants to maintain the data. Or An Entity may be an object with a physical existence for Ex. A person, car, house, Or it may be an object with a conceptual existence i.e a company, a job, or a university course..

Person : Employee, Student, Customer, Patient etc.
Place : City, State etc
Object : Building, Vehicle etc
Event : Order, Sale
Concept : Accounts, Course etc..

Entity Type:-

The collection of entities that share common properties will be called as an "Entity type". Each and every entity type must contain a name. The name must be specified in capital letters in an E-R Diagram. Entity type must be placed inside the rectangular box.



Entity Set / Entity Instance:-The single occurrence (Record) of an entity type will be called as an "Entity Instance **or** Entity Set' . Generally an entity type is described once, many entity instances of that entity types may be represented.

Ex:- Entity Type : CUSTOMER

C_Id	C_Name	C_Add
100	XYZ	Nellore
101	PQR	Kavali
102	STU	Tirupati
103	VWX	Bapatla

Attributes

Entity Instances

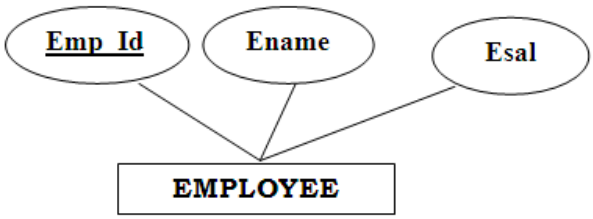
The entity types are divided into Three types they are

a) Strong Entity Type
b) Weak Entity Type
c) Associative Entity Type

a) Strong Entity Type:-

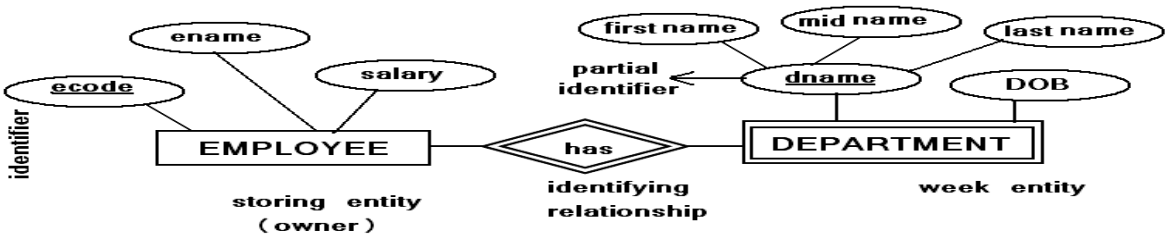
A Strong Entity is an entity. Which don't depends on any other entity. It is independent entity. A strong entity type have one or more identifier attributes. In E-R Diagrams strong entities are indicated by **Rectangle box.**

Ex:-



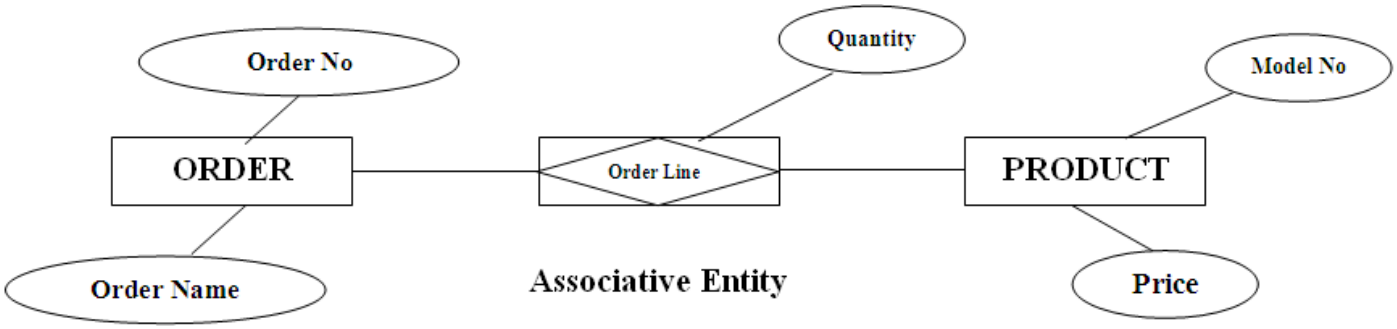
b) Weak Entity Type :-

A Weak entity is an entity, which always depends on some other entity . In ER diagram weak entity can be indicated by double line Rectangle box.



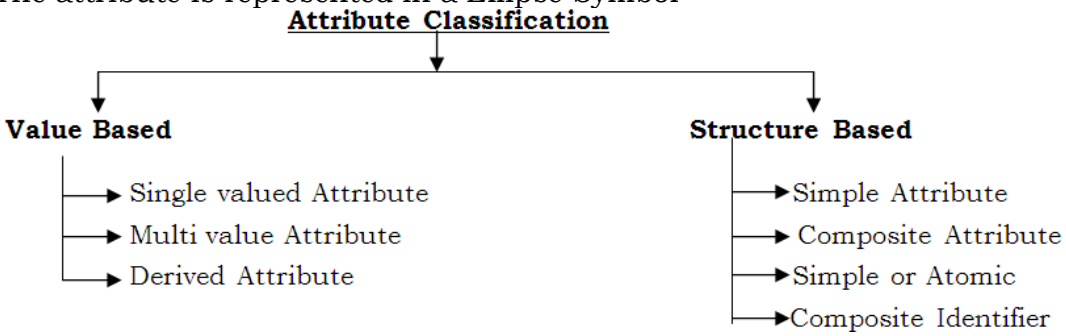
c) Associative Entity:-

An associative entity is an entity it shows the relationship based on an identifier attribute/not. In ER Diagrams Associative Entity represented as Rectangluer Box enclosed with Diamond symbol.



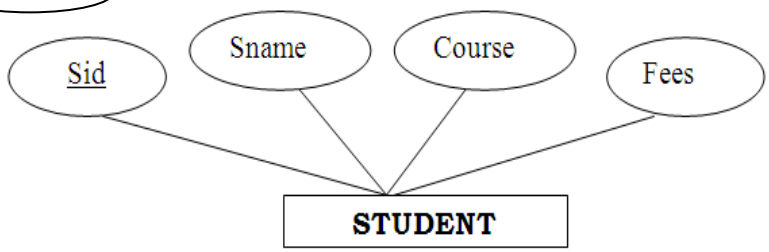
2. Attribute (Attribute Classification):

An attribute is a property or Characteristic of an entity type. Each entity type has a set of attributes. The attribute is represented in a Ellipse Symbol



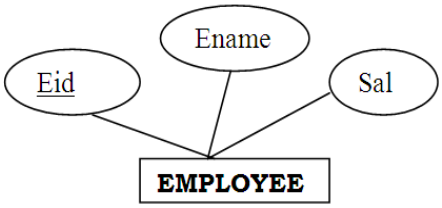
a) Simple Attribute:-

Simple attribute is attribute it is not divisible into smaller components, it is denoted by "**Ellipse Symbol**" 



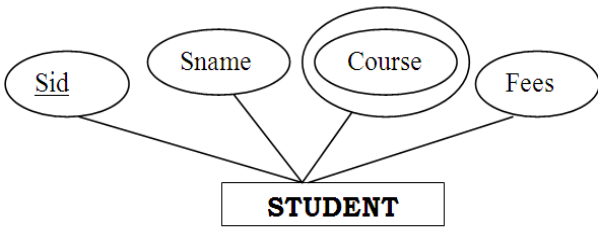
b) Single – Valued Attribute:

A Single valued attribute is an attribute that can take only one value in a given entity instance. For example, the entity type employee has an attribute namely 'sal' takes only one value for a given entity instance. So, salary is a single-valued attribute. In ER diagrams, single valued attribute is represented by using 'general ellipse'.



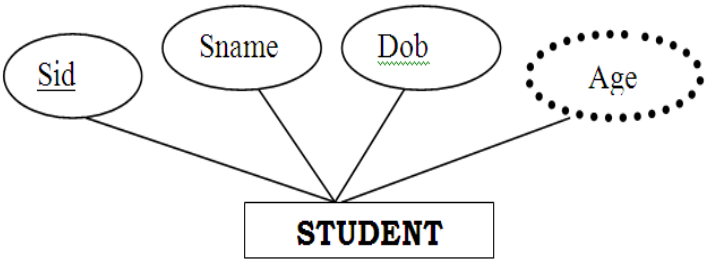
c) Multi value Attribute:

An attribute that may take more than one value for a given entity instance is called multi value attribute. In ER diagrams, multi valued attributes are represented by using '**Double-Lined Ellipse'.** 



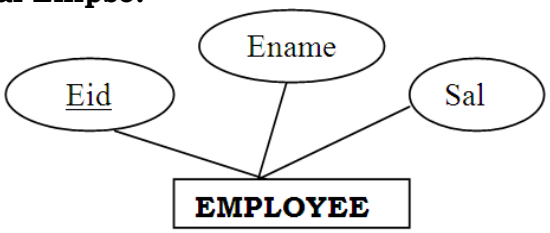
d) Derived or Stored Attribute:

Derived attribute is an attribute whose values can be calculated from the related attribute values. In ER diagrams derived attributes are indicated by using **Dotted Ellipse.** 



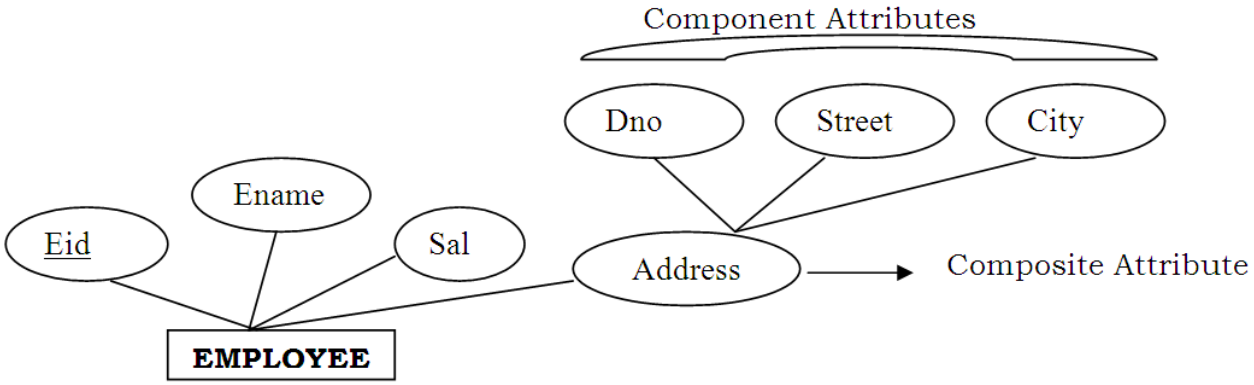
f) Simple or Atomic Attribute:

A Simple attribute is an attribute that cannot be broken down into smaller components. For example, an entity type employee has an attribute salary. Here, the attribute salary cannot be broken the smaller components. So, salary is a simple attribute in ER diagrams. Simple attribute are indicated by using **General Ellipse.**



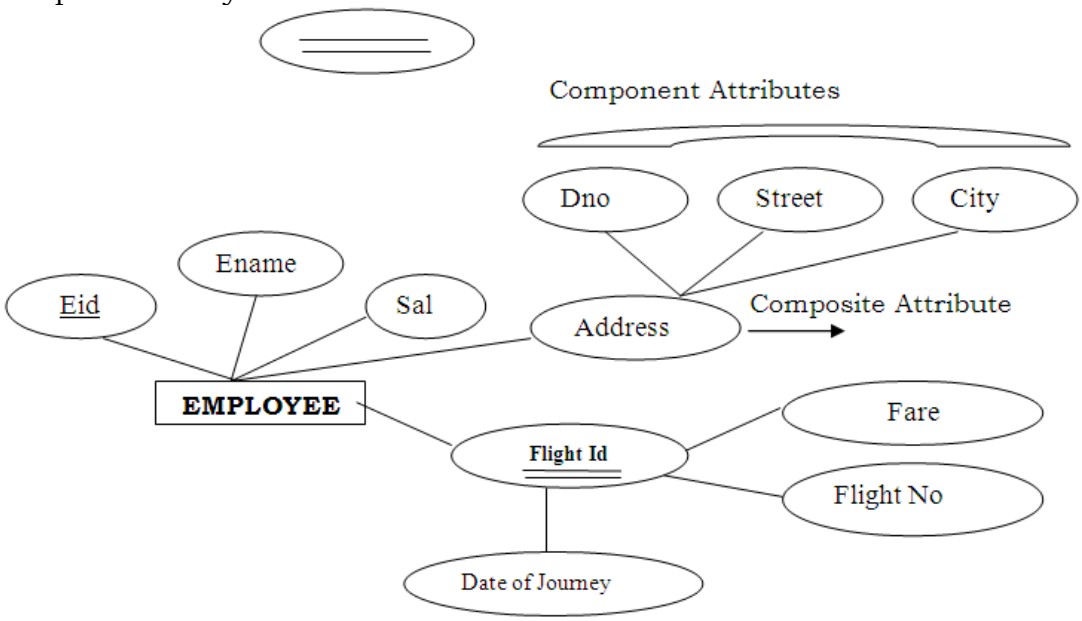
g) Composite Attribute:

A Composite attribute is an attribute that can be broken down into meaningful components. The most common example, composite attribute is address, which can usually broken down into components like Dno, Street, City, Pincode etc. These smaller attributes are called component attributes.



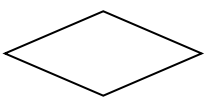
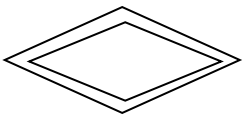
h) Composite Identifier Attribute:

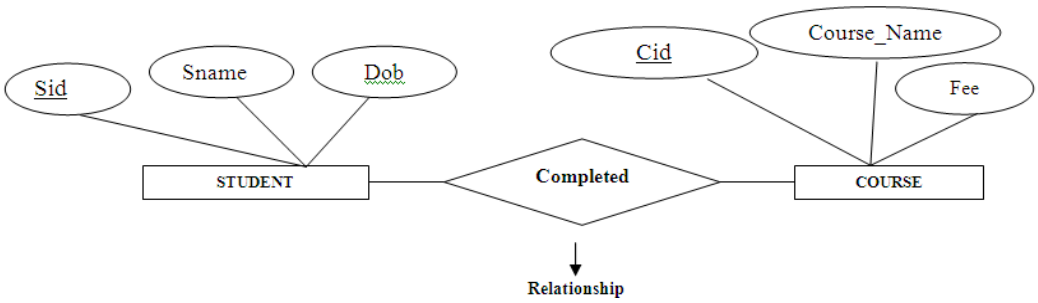
An identifier that consists of a composite attribute is called the composite identifier attribute. Is represented by **Double Lined Under Line**

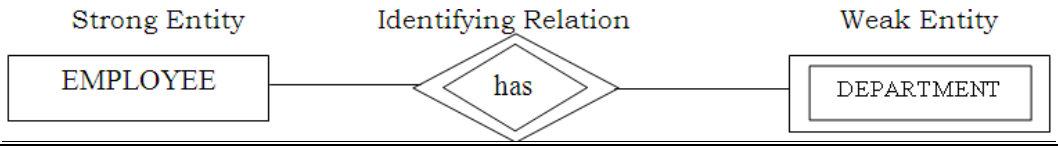


3. Relation Ship:

A relationship is an association between the instances of one or more entity types .In an E-R Diagram , the relationship is represented by a **Diamond Symbol** containing the name of the relationship . These are 2 types 1) Relationship 2) Has / Identifying Relationship

1)  2) 

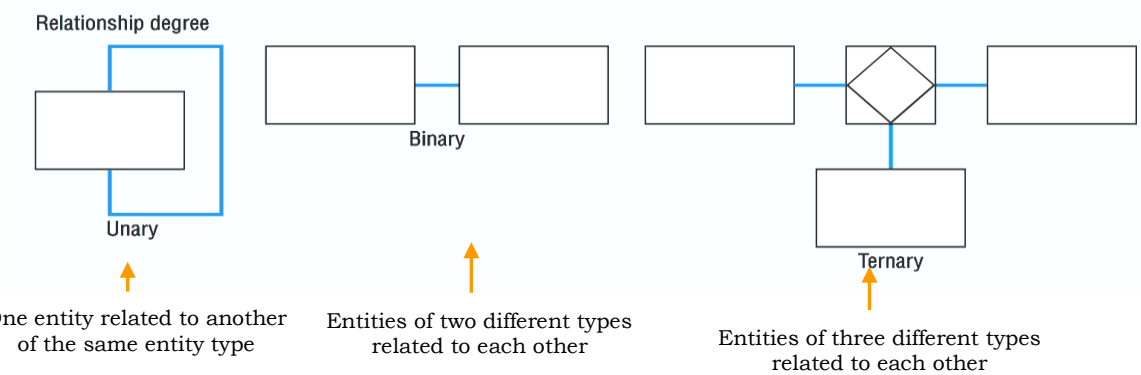




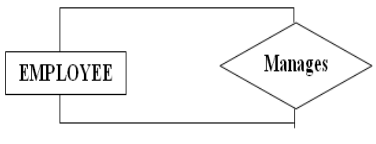
Degree of Relationship / Degree of Relationship set

The degree of the relationship is the number of entity types that participates in that relationship. In ER-diagrams, there are 3 most common relationship degrees are available. They are:

a) Unary Relationship
b) Binary Relationship
c) Ternary Relationship

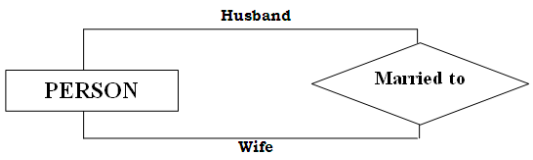


a) Unary Relationship: A Unary Relationship shows relationship between the instances of single entity type.

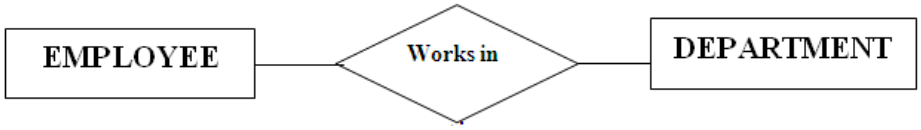


Ex 1:-

Ex 2:-



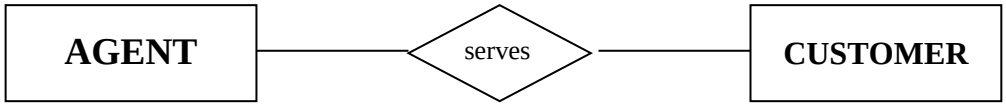
b) Binary Relationship: A Binary Relationship shows relationship between the instances of two entity types.



c) Ternary Relationships: A Ternary Relationship shows relationship between the instances of three entity types.

Relationship Classification

A relation ship is an association between the Entities. The relation must be specified with the help of diamond symbol. For example A relation ship exists between CUSTOMER and AGENT An agent can serves many customers and each customer may be served by one agent



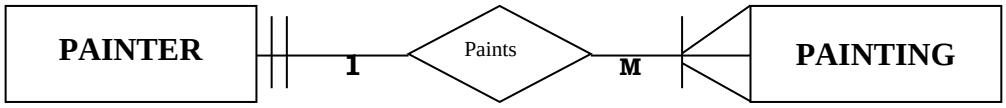
The Relationship Classified into 3 types

1) One-to-Many Relation Ship (1 : M)
2) Many-to-Many Relation Ship (M :M)
3) One-to-One Relation Ship (1 : 1)

1) One-to-One Relationship (1:1):- In 1:1 relationship one entity can be related to only one other entity. Consider the EMPLOYEE has PARKING PLACE ex:-

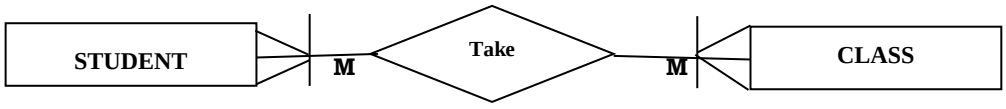
- The above example each Employee has one and only parking place and each parking place is allotted by one Employee.
- There is one row in EMPLOYEE table is associated with one row in PARKING PLACE table.

2) One-to-Many Relation Ship (1 : M) :- 1:M relationship is the standard type in the relational model. Consider the PAINTER paints PAINTING example.



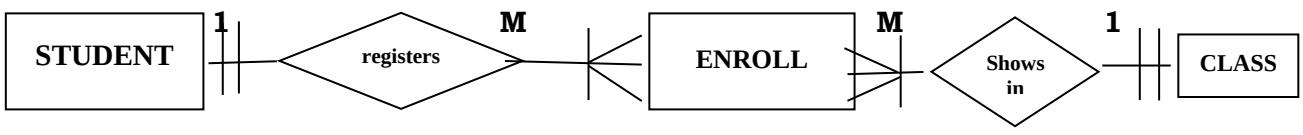
- The above example. Each painting is painted by one and only one painter, but each painter could have painted many paintings.
- There is one row in PAINTER table is associated with many row in PAINTING table

3) Many-to-Many Relation ship(M:N):- The M:N relation ship is not supported directly in the relational environment. Therefore M:N relationships can be implemented by creating a new entity in 1:M relationships.



- Each STUDENT takes many CLASSES, Each CLASS contains many STUDENTS.
- There is many row in STUDENT table is associated with many row in CLASS table.

From the above two tables, the relational operation become very complex. It leads to data redundancy. So avoid this problem, we can create a new table ENROLL, and link that table in between STUDENT and CLASS tables.



The above example shows changing the M:N relation ship into 1:M relationships.

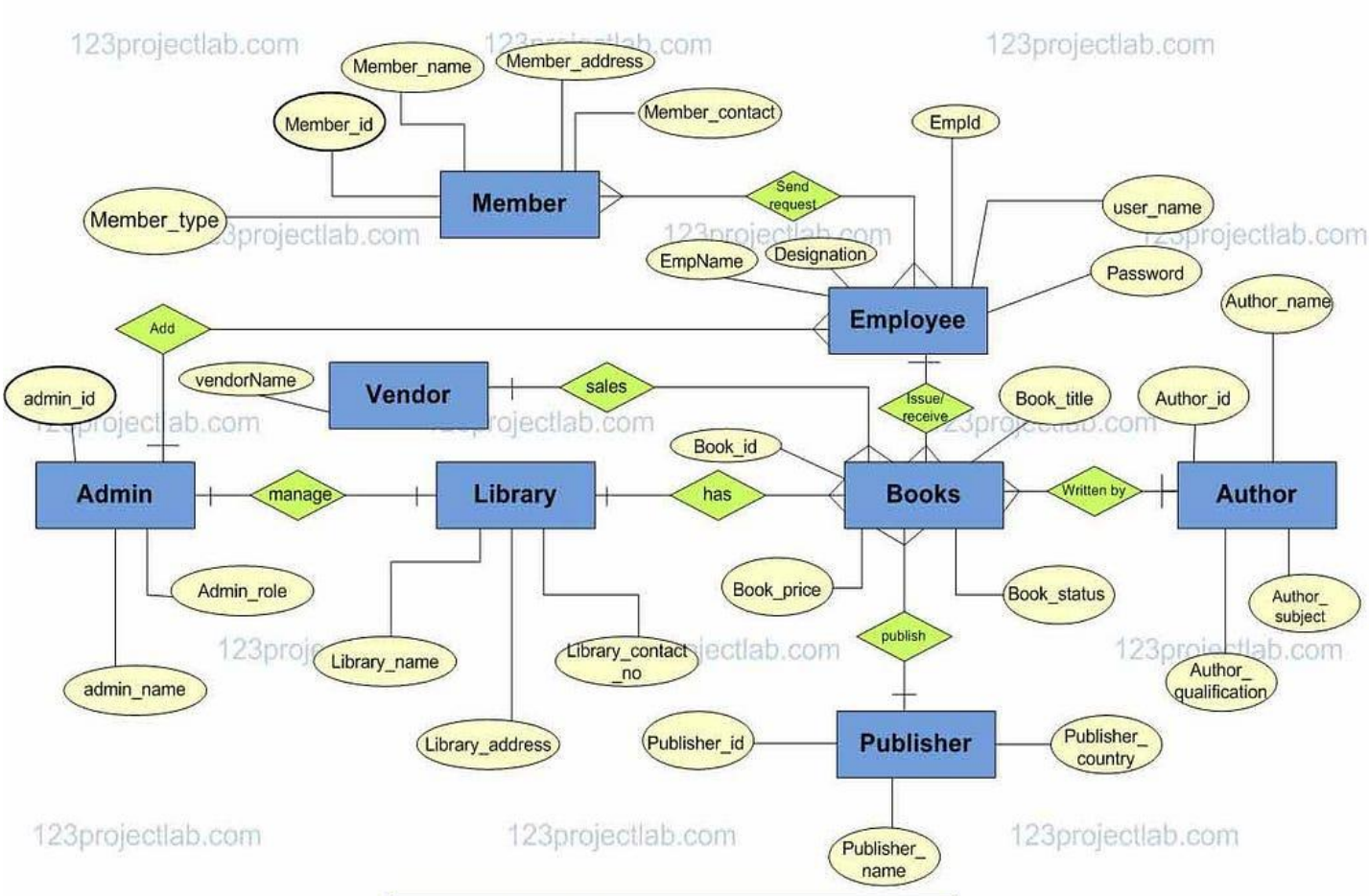
ER-Diagram for Library Management System

The ER (Entity Relationship) Diagram of Library management system is an ER model of online library management system entities. EDR for Library management system is a graphical representation of database tables and relations between library, member, book, admin etc.

The main purpose of ER diagram for online library management system is to represent data objects and their relationships. The main entity of this ER diagram is library, library admin, employee, member, book, vendor, author and publisher.

The Library Management System database working is based on the following assumptions-

→The system keeps track of all users with a single point authentication via user id and password.
→The special feature access is granted based on users' role (such as admin, employ or member)
→The library admin has full control over the entire system. He can add new employee and member. Admin can assign access right to different users.
→Employee of the library maintain book details of the library. They can issue or receive books from library member. Each employee has unique login id and password.
→Each book has unique book id, title, price and book status (available/issued).
→Each vendor has unique id. Similarly, each publisher also has unique publisher id.
→For better database handling we have used ids in the database for any transaction. Such as at the time of book issue/receive we store book id in the database instead of book title and similarly instead of member name we store member id.
→Employee can generate various library reports such as member report, book report, vendor report etc. Such reports are very helpful in decision making.
→The entire system has several checks to avoid mismanagement. Any employee can not again issue a book which is already issued to someone. There is a check that prevents it from doing so. Similarly, entry of any master record such as book, member, vendor, author or publisher can not be deleted if its related record exists in any transaction table.



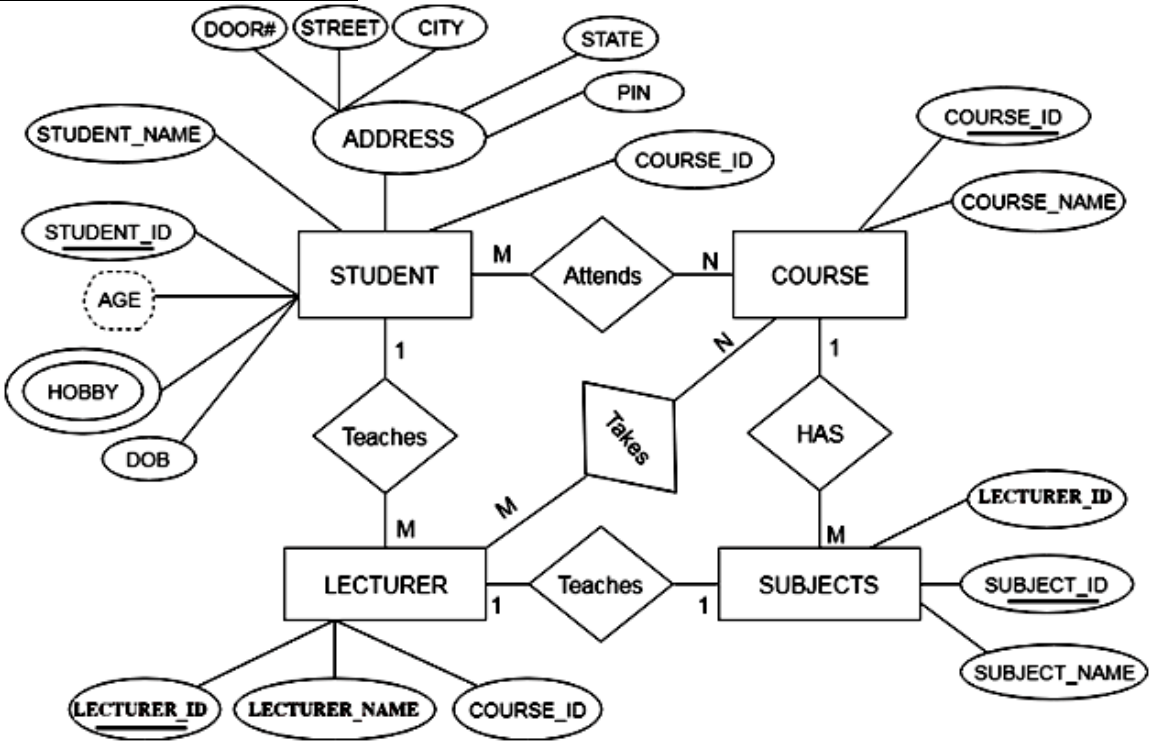
ER-Diagram for Library Management System

Reducing of ER diagram to Table

The database can be represented using the notations, and these notations can be reduced to a collection of tables.

In the database, every entity set or relationship set can be represented in tabular form.

The ER diagram is given below:



Converting the ER diagram to the table:

- **Entity type becomes a table.**
  In the given ER diagram, LECTURE, STUDENT, SUBJECT and COURSE is individual tables.
- **All single-valued attribute becomes a column for the table.**
  In the STUDENT entity, STUDENT_NAME and STUDENT_ID form the column of STUDENT table. Similarly, COURSE_NAME and COURSE_ID form the column of COURSE table and so on.
- **A key attribute of the entity type represented by the primary key.**
  In the given ER diagram, COURSE_ID, STUDENT_ID, SUBJECT_ID, and LECTURE_ID are the key attribute of the entity.
- **The multi valued attribute is represented by a separate table.**
  In the student table, a hobby is a multi valued attribute. So it is not possible to represent multiple values in a single column of STUDENT table. Hence we create a table STUD_HOBBY with column name STUDENT_ID and HOBBY. Using both the column, we create a composite key.
- **Composite attribute represented by components.**
  In the given ER diagram, student address is a composite attribute. It contains CITY, PIN, DOOR#, STREET, and STATE. In the STUDENT table, these attributes can merge as an individual column.
- **Derived attributes are not considered in the table.**
  In the STUDENT table, Age is the derived attribute. It can be calculated at any point of time by calculating the difference between current date and Date of Birth.

Prepared by

DVH. Venu Kumar

M.Sc.,M.Ed.,M.Tech(CSE)

Asst.Professor., Dept.of CSE., Geethanjali Institute of Science and Technology : Gangavaram : Nellore

A Solder fighting without a weapon is the same as a teacher giving a lesson without reading a book