UNIT – I

OVERVIEW OF DATABASE MANAGEMENT SYSTEM

INTRODUCTION:

Generally The Computers are used to Store the data, Manipulate the Data and Present the Data to the users. From IIIGeneration of the computer the data storage is played a essential role for the programmers. They were some personal computers are used to store the data.

A data base management system is a collection of logically interrelated and set of programs to access this data. Here the collection of logically related data is called data base. Mainly the data base systems were developed to overcome the limitations of file processing system. The primary objective of DBMS is to provide a comfortable environment to store and retrieve database information.

Database management system was first introduced in 1960. The file processing system was still dominated during this period in this period the data base management systems were used primarily for large and complex applications. During the 1970 the use of DBMS becomes a commercial reality. This model is generally regarded as 1st generation of DBMS. The DBMS permits only one user to access at a single time.

To overcome this limitation in 1980 Dr.E.F.Codd and others developed a new model called “The Relational Data Base Management system”. This model is considered as second generation of DBMS. In this model the fourth generation language called SQL is used for data manipulation.

At the period of 1990 introduced the new model object oriented relational database management system (ORDBMS). This model is consider 3rd generation of DBMS.

Basic concepts & definitions

Data: - Data is a collection of “Raw facts" from which conclusions can be drawn. For example in a sales person’s database the database the data would be included the facts such as customer no, customer name, address, contact no etc.
Information: - Data that have been processed to increase the knowledge of a person, who uses the data, is called information.
Data base: - It is a collection of “Related and Meaningful information “stored centrally at one location.

- ☞ The main objective of database is record the history for the future use.
- ☞ The data base always stores the data in the hard disk in form of records

Meta data: - Data that describes the properties of another data is called Meta data.

For example the employee details to the following.

Employee details

Eno	Ename	Job	sal	Deptno
101	Rajini	Analyst	8000	10
102	Madhu	Research	9000	20
103	Harini	HR	20000	30

In the above example the Meta data eno, ename, job, sal, deptno describes the properties of employee data. It provides complete description of tables. That’s why it is important.

Data base system: - The data base and DBMS software together is called data base system.

Database Management System (DBMS) :It is a collection of programs working together to control Data Manipulations (add , change ,remove), Retrievals (Read) and sharing on Database. (Or)

It is program or software it manages the data in the database based on the request from user.

Data Inconsistency: - Data inconsistency exists when different copies of the same data appear in different places.

Data Redundancy: - as application programs are often developed independently, there is more chance to duplicated data. This unnecessarily repeated data is called “Data redundancy”.

Data dictionary:

- ☞ Data dictionary is a collection of system defined tables.
- ☞ Data dictionary is the main source of information for any RDBMS.
- ☞ All data dictionary tables belong to SYS user account.
- ☞ The system defined tables are created automatically whenever the data base is created.

File:- A file is a collection of records that are stored regarding a specific entity. Each file is defined with a definite structure with well – defined fields.

Record: - A record is a row that represents several properties of an entity. Each row in a table represents information about exactly one entity instance.

Field: - A field is an attribute or property of an entity type. A field is used to define and store data.

1. Discuss about data and information

Data: Data is a collection of “Raw facts" from which conclusions can be drawn. For example in a sales person’s database, the data would be included the facts such as customer no, customer name, address, contact no etc.

- Data may be defined as collection of any facts, Figures, Observations, assumptions or occurrences in the real world.
- Data may be represented using numbers and letters or combination of both.
- Data will always be in raw shape.
- Data is unorganized facts and figures.
- It can be generated or prepared from various departments and other sources.
- Data individually carries less meaning.
- Data individually may not be useful for decision making.
- It sometimes as information.

Example: Collection of the applicant, address, age, etc. are facts related to people, these facts are represented by using Alphabets, Numbers or combination both.

Information: Data that have been processed in such a way is to increase the knowledge of a person, who uses the data, is called information.

- Information may be defined as processed data.
- Information is more refined in its shape.
- Information is more meaningful when compared with data.
- Information can be used for decision making.
- Information may also be defined as organized or classified data.
- Information can be distributed to various departments and people in the organization.
- It sometimes appears as data.

Example:
If we collect Marks obtained by 75 students in Computers subject --- This is **data**.
Where as if we prepare a list of highest or lowest marks of the same ---This is**Information**

Data vs. information: The term data and information are closely related, and in fact are often called interchangeably. For example consider following the list facts.

Madhu	24001
Prasad	24002
Murali	24003

The above facts satisfy our definition ‘Data’. But the above data is useless in the present form. By adding a few additional data terms (column names & headings) and providing some structure to the above data, then anybody can recognize the above data.

Student details

Student Name	Student ID
Madhu	24001
Prasad	24002
Murali	24003

The above process data satisfies our definition ‘information’

<u>Meta data</u>: - Data that describes the properties of another data is called Meta data.

For example the employee details to the following.

Employee details

EmpId	Ename	Job	Sal	Deptno
24001	Madhu	Analyst	25000	10
24002	Mahesh	Lecturer	30000	20
24003	Murali	Research	20000	30

In the above example the Meta data empId, Ename, Job, Sal, Deptno describes the properties of employee data. It provides complete description of tables. That’s why it is important.

2. Discuss about database and database management system.

<u>Data base</u>: - It is a collection of “Related and Meaningful information “stored centrally at one location.

- ☞ The main objective of database is record the history for the future use.
- ☞ The data base always stores the data in the hard disk in form of records.

A database is a computer based record keeping records and maintains information. The database is a single large repository of data, which can be used simultaneously by many departments and users.

To maintain huge database, we need certain operations which help in maintaining the data in database efficiently. The most commonly used operations performed on the database are: Insertion, selection, Updating and sorting.

Database examples in our daily life:

Files on your computer hard drive

A telephone book

Employee details in a company

TV guide

Airline reservation system

Motor vehicle registration system

Database Management System (DBMS) :

It is a collection of programs working together to control Data Manipulations (add,change, remove), Retrievals (Read) and sharing on Database. (OR)

It is program or software it manages the data in the database based on the request from user. (OR)

A Database Management System (DBMS) is a collection of programs that manages the databasestructure and controls access to the data stored in the database.

The DBMS serves as the intermediary between the user and database. The database structure itself isstored as a collection of files and the only way to access the data in those files is through the DBMS. Itpresents the end user or application program with single, integrated view of data in the database.

The DBMSreceives all application requests and translates them into the complex operations to fulfill those requests. TheDBMS hides much of the database's internal complexity from the application programs and users. Theapplication program might be written by a programmer using a programming language such as VisualBasic.NET, Java or C++, or it might be created through a DBMS utility program.

Examples:

Oracle RDBMS, IBM DB2, Microsoft SQL Server

Discuss in detail DBMs objectives.

All these organizations are making a huge profit just because of database management system. This is why because DBMS provides a lot benefits to all these companies and there are lots of objectives of using a database management system.The major objectives of database management system are:

1. Data Availability
2. Data integrity
3. Data security
4. Data independence
5. Mass storage
6. Multiple users access
7. Backup and recovery

Data availability: Data availability refers to the fact that the data are made available to large variety of users in a meaningful format at reasonable cost so that the users can easily access the data.

Data Integrity: Data integrity refers to the completeness, correctness and consistency of data. It ensures that data entered into the database must be complete, accurate, valid, and consistent. Integrity is related to the quality of data, which is maintained with the help of integrity constraints. These constraints are the rules that are designed to keep data consistent and correct.

Data Security: Data security refers to the fact that only authorized users can access the data. Data security can be enforced by passwords. If two separate users are accessing a particular data at the same time, the DBMS must not allow them to make conflicting changes.

Data Independence: DBMS supports the concept of data independence since it represents a system for managing data separately from programs that use the data. DBMS allows the user to store, update and retrieve data in an efficient manner. In order to store the information efficiently, complex data structures are used to represent the data. The system hides certain details of how the data are stored and maintained.

Mass Storage

DBMS can store a lot of data in it. So for all the big firms, DBMS is really ideal technology to use. It can store thousands of records in it and one can fetch all that data whenever it is needed.

Multiple Users Access

No one handles the whole database alone. There are lots of users who are able to access database. So this situation may happen that two or more users are accessing database. They can change whatever they want, at that time DBMS makes it sure that they can work concurrently.

Data Backup and recovery

Sometimes database failure occurs so there is no option like one can say that all the data has been lost. There should be a backup of database so that on database failure it can be recovered. DBMS has the ability to backup and recover all the data in database.

Write about evolution of database management system

At the time of computers there was no data processing on those days. Computers are only used for engineering and scientific calculations. Gradually computers were introduced into the business world. For business applications computers must able to store and manipulate large amount of data.

For this purpose introduce a new concept called file processing system. But file processing system has number of limitations is there. To overcome the limitations one more new system introduced called **Database management System**.

A database is collection of logically interrelated data. Database is designed to meet the information needs of an organization.

DBMS were first introduced during the 1960s and have continued to evolve during subsequent decades. The database management system supports different types of database technologies or architectures or models. The Evolution of Database systems are

- File processing systems
- Hierarchical database System
- Network Database System
- Entity Relationship Model
- Relational Database System
- Object based Relational database Management System

File processing systems

1960s: Traditional file systems and first database management systems were introduced.

The earliest business computer systems were used to process business records and produce information. They were generally faster and more accurate than equivalent manual systems. These systems stored groups of records in separate files, and so they were called **file processing systems.**

Hierarchical database System:

1970s: Hierarchical and Network data base models also known as first generation DBMS. Hierarchical database model is one of the traditional database models developed in 1960's. In this model, the records are represented in the form upside – down tree (i.e. hierarchical structure). This tree structure model consists of many levels (segments), which are similar to record type in file system. Here the top files is called root and the bottom files are called leaves.

Mainly hierarchal data model is used in transaction processing systems (TPS). Management information systems (MIS) applications.

Network Database System:

Network Database Models were developed in 1970's in order to overcome the problem encountered in using hierarchical models.

Entity Relationship Model:

1976: Peter Chen presented Entity Relationship model, which is widely used in database design.

Relational Database System:

1980s: Relational model also known as second generation DBMS. In Relational model, all data are represented in the form of tables. A relatively simple fourth generation language called SQL (for Structured Query Language) is used for data retrieval.

Object based Relational Database Management System:

1990s: Object-Oriented and Object-Relational data model was introduced. Any database design based on OOPS concepts known as ORDBMS.

Classification of Database Management System

The DBMS can be classified into different categories on the basis of several criteria such as the data model they are using, number of users they support, number of sites over which the database is distributed, and the purpose they serve.

Based on Data Models:

Depending on the data model they are use, the DBMSs can be classified as hierarchical, network, and relational.

Hierarchical DBMS organizes the data records in a tree structure i.e Hierarchy of parent – child relationship. In a hierarchical data – base, a parent record may have more than one child, but a child always has only one parent. This is called 'one – to- many relationship'.

Network DBMS organizes the data records liked to one another through pointers, which is an association between two records. A network database is similar to a hierarchical database except that each child can have more than one parent record. This is called 'many- to – many relationship'.

A relational DBMS organizes the data records in the form of table and relationship among the tables are set using fields. It is simple in nature because data is simply represented in tabular format.

Based on Number of users:

Depending on the number of users the DBMS support, it is divided into two categories, namely, single user system and multi user system.

In single user system, database resides on one computer and is only accessed by one user at a time. The user may design, maintain, and write programs for accessing and manipulating the database according to the requirements.

In multi – user system, multiple users can access the database simultaneously. In multi – user DBMS, the data is both integrated and shared. For example, the online book database is a multi user database system in which the data of books, authors, and publishers are stored centrally and can be accessed by many users.

Based on Number of sites:

Depending on the number of sites over which the database is distributed, it is divided into two types, namely, centralized and distributed database systems.

Centralized database system runs on a single computer system. Both the database and DBMS software reside at a single computer site. The user interacts with the centralized system through a dummy terminal connected to it for information retrieval.

In distributed database systems, the database and DBMS software are distributed over several computers located at different sites. The computers communicate with each other through various communication media such as high – speed networks or telephone lines.

Distributed database can be further classified as homogeneous and heterogeneous. In homogeneous distributed database system, all sites have identical DBMS software, whereas in heterogeneous distributed database system, sites use different DBMS software.

Based on the Purpose:

Depending on the purpose the DBMS serves, it can be classified as general purpose or specific purpose.

DBMS is a general purpose software system. It can however, be designed for specific purposes such as airline or railway reservation. Such systems cannot be used for other applications. These database systems fall under the category of online transaction processing (OLTP) systems. Online transaction processing systems is specially used for data entry and retrieval. It supports large number of concurrent transactions without excessive delays. An automatic Teller Machine for a bank is an example of online commercial transaction processing application. The OLTP technology is used in various industries, such as banking, airlines, supermarkets, manufacturing etc.

UNIT –I I

OVERVIEW OF DATABASE MANAGEMENT SYSTEM

1. What is file based system? Write its characteristics.

The file system method of organizing and managing data was certain improvement compare to manual system. But with the file system approach is that even the simplest data retrieval task requires extensive programming in 3GL examples of 3GL are **COBO**L (Common Business-Oriented Language), **BASIC** (Beginner's All-Purpose Symbolic Instruction Code), **FORTRAN** (Formula Translation), C^{++}.

Ex:- We can retrieve the Co_Name, Co_Pno, Co_Zip form CUSTOMER table by using 3GL and 4GL programming Languages.

Characteristics of File based system:

- Some important characteristics of file – based system are:
- It is a group of files storing data of an organization
- Each file is independent from one another.
- Each file is called a flat file.
- Each file contained and processed information for one specific task, such as accounting or inventory.
- Files are designed by using programs written in programming languages such as C, C++.
- Each file must have its own file management system.

2. What is file processing system and explains problems with file system data management?

Traditional file processing system

At the beginning of computers there was no data processing on those days computers are only used for engineering and scientific calculations. Gradually computers were introduced into the business world. For business applications computers must able to store and manipulate large amount of data. For this purpose introduce a new concept called file processing system. But the file processing systems had a number of limitations. As a result these file processing system have been placed by database management systems. Some of the **disadvantages** associated with processing system are.

Disadvantages (Problems) of file system

1. Structural and data dependence
2. Field definition and naming conventions.
3. Data Redundancy
4. Data inconsistency
5. Limited data sharing
6. Lengthy development times
7. Excessive program maintenance
8. Limited data security

<u>1</u>**<u>. Structural and data dependence</u>: -** The application programs that access to data file is dependent on its structure. Whenever any change is made to a data file, then all the application programs that access that data file must also be refined. The changes in the characteristics of data such as changing data type of a field, changing the memory capacity of a field cause changes in all the programs that access the file.

<u>2. Field definition and naming conventions</u>:-while designing a data file, it is important to design every field name of a record. These field names describe the characteristics of actual data values.

For example the Field name "CUST_ ID" describes the ID value of a customer. To design the field names, we follow some naming conventions.

Example

CUS_NAME	Customer name
CUS_CITY	Customer city
EMP_ID	Employee ID

<u>3. Data Redundancy:</u> - The file system data management forces the storage of same basic data in different locations. As application programs are often developed independently, there is more chance of duplicated data. This unnecessarily repeated data is called "Data Redundancy". The data stored in different locations need to be updated consistently. Otherwise it leads to the problem of data inconsistency.

Example

The agent names and phone numbers occurred in both the CUSTOMER and AGENT files. You need only one correct copy of agent names and phone numbers.

Having them occur in more than one place produces data redundancy. Data redundancy exists when the same data are stored unnecessarily at different places.

4. Data Inconsistency: - Data Inconsistency exists when different copies of same data appear in different places.

Example:

Suppose you change the agent phone number or address in the AGENT file. If you forget to make corresponding changes in the CUSTOMER file, the file contains different data for the same agent. This leads data inconsistency.

5. Limited data sharing:-In file processing system each application has its own private files and uses have little opportunity to share data outside their own application.

6. Lengthy development times: - In file processing system, each new application requires its own file and also it requires new application program to access different types of data. It is time consuming process.

7. Excessive program maintenance: - The file processing system requires heavy program maintenance. In fact 80% of the total information systems development budget many are devoted to program maintenance.

8. Limited data security: - In file processing system we cannot have any feature to provide security feature to provide security on files so in file processing security is very minimum.

3. Discuss about the reasons brings you to choose the data base than file system. (OR)
What is DBMS? Explain the advantages of data base approach.

A database management system is a collection of interrelated data and a set of programs to access the data. The primary objective of DBMS is to provide a convenient environment to store and retrieve database information. The database approach provides a number of advantages compared to file processing system. They are

1. Program independence
2. Minimal data redundancy(duplication of data)
3. Improved data sharing
4. Improved productivity application development

5. Improved data quality
6. Enforcement of standards.
7. Improved data accessibility
8. Reduced program maintenance
9. Improved data security

1. Program data independence: - With database approach data description (Meta data) are stored in a central location called **data dictionary** or **repository.** The separation of data description from the data is called data independence.

2. Minimal redundancy: - In file processing system each application has its own private files. In data base approach tables are integrated into a single logical unit called database. Each primary fact (data (or) records) is recorded into the database.

3. Improved data sharing: - A data base is designed as shared resource. Authorized users are granted permission to use the data base.

4. Improved productivity application development: - A major approach is that, it reduces the cost and time for developing a new business application. The data base approach contain specific built in programs for different tasks.

5. Improved data quality: - The database approach provides numbers of tools to improve data quality database designers can specify integrity constraints to improve data quality a constraints is a rule that cannot be violated by database

6. Enforcement of standards: - With central control of database the database administrator (DBA) establisher and enforce data standards

7**. Improved data accessibility: -** Improved data accessibility:- with a relational database and user can retrieve and display data without programming experience. For example, the end-users can display data from tables by using simple select commands.

Ex: - sql>select * from emp;

8. Reduced program maintenance: - In file processing environment the file structure and data are stored into a single logical unit. As a result changes the data formats need to modify application programs. In a database environment data are more independent. So the application, if the user change the data format. As a result, program maintenance can be reduced in data base environment.

9. Improved data security: - In database approach the data base developers contains so many security features to provide security tables so in database approach security is very high.

What are the functions and services of DBMS

A DBMS performs several important functions that guarantee the integrity and consistency of the data in the data base. Most of those functions are transparent to end users, and most can be achieved only through the use of DBMS. They include data dictionary management, data storage management, data transformation and presentation, security management, multi user access control, backup and recovery management, data integrity management.

1. Data dictionary management: - The DBMS stores definitions of the elements and their relationships in a data dictionary. The programs that access that access the data in the data base work through the DBMS. The DBMS uses the data dictionary for required data component structure and the relationships. By maintaining data dictionary DBMS achieves program dependence.

2. Data storage management: - The DBMS creates and manages the complex structures required for data storage physically. A modern database provides storage not only for the data, but also for related data entry forms or screen definitions, report definitions, data validation rules, procedural code, structure to handle video and picture formats and so on. Data storage management is also important for data base performance tuning.

3. Data transformation and presentation: - The DBMS transforms entered data to conform to required data structures. For example, imagine an enterprise data base used by a multinational company. An end user in England would expect to enter data such as July 11, 2011 as “11/07/2011” In contrast the same date would be entered in the United States as “07/11/2011”. Regardless of data presentation format, the DBMS must manage the date in the proper format for each country.

4. Security management:-The DBMS creates a security system that enforces user security and data privacy. Security rules determine which users can access the data base, which data items can access and which operations the users perform. All data base users may be authenticated to the DBMS through a user name and password or though biometric authentication such as a finger print scan.

5. Multi – user access control: - To provide data integrity and data consistency, the DBMS uses sophisticated algorithms to ensure that multiple users can access the data base concurrently without compromising the data integrity of the data base.

6. Backup and recovery management: - The DBMS provides backup and recovery to ensure data safety and integrity. Current DBMS systems provide special utilities that allow the DBA to perform routine and special backup and restore procedures. Recovery management deals with the recovery of the data base after failure, such as bad sector in the disk or power failure.

7. Data integrity management: - The DBMS promotes and enforces integrity rules, thus minimizing data redundancy and maximizing data consistency. The data relationships stored in the data dictionary are used to enforce data integrity.

8. Data Definition Services: The DBMS accepts the data definitions such as external schema, the conceptual schema, the internal schema, and all the associated mappings in source form.

9. Database Communication Interfaces: The end-user's requests for database access are transmitted to DBMS in the form of communication messages.

10. Data Manipulation Management: A DBMS furnishes users with the ability to retrieve, update and delete existing data in the database.

Write about the Components of database environment.

The major components of database environment are

1. Computer aided software engineering tools(CASE tools)
2. Repository
3. Database
4. Database management system(DBMS)
5. User interface(Forms)
6. Application programs
7. Database administrator(DBA)
8. System developers
9. End users

CASE Tools: - Computer Aided Software Engineering Tools. Some predefined tools (built in-tools) are used to design database and application programs. These tools are called Case tools. Some Case tools are:

- Form generator
- Report generators
- Code generators
- Diagramming tools

Repository (data dictionary):- A repository is a centralized memory unit in the database generally; it is used to store table structures (Meta data), constraints information, database object information, database users' information etc.

<u>Data base</u>: - It is a collection of "Related and Meaningful information "stored centrally at one location.

- The main objective of database is record the history for the future use.
- The data base always stores the data in the hard disk in form of records

<u>Database Management System (DBMS)</u> :It is a collection of programs working together to control Data Manipulations (add , change ,remove) , Retrievals (Read) and sharing on Database. (Or)

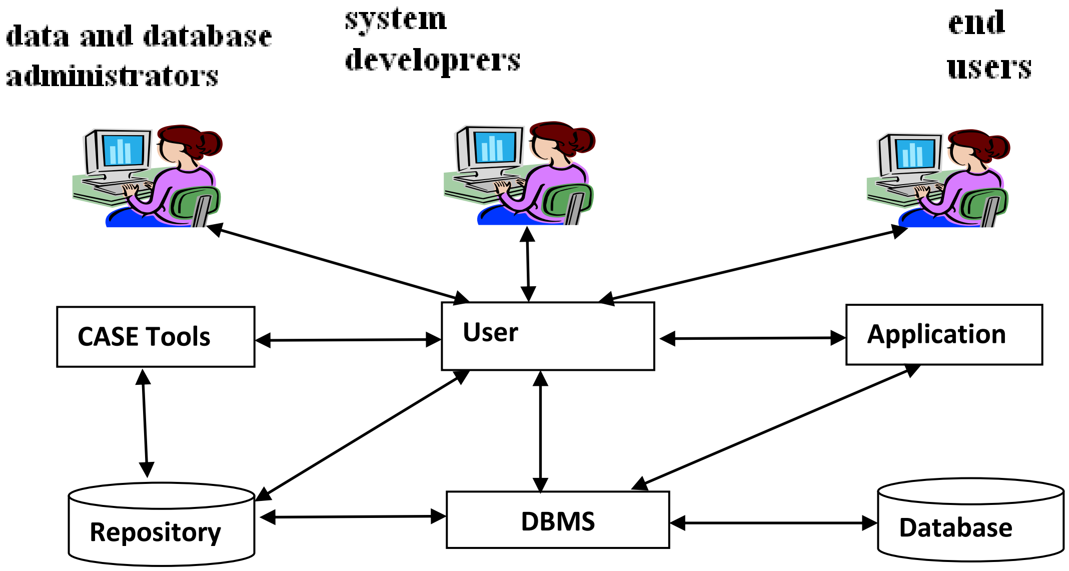
<u>Application programs:</u>-Application programs are the programs that are developed by the programs generally application programs are used to manipulate the data in the data base. These programs provide information to users.

<u>User interfaces: </u>- User interfaces are the programs that are used to manipulate data from the database easily generally user interface are created through the case tool form generator

<u>Database administrator</u>: - Database administrators are the persons who are responsible for the over all database in an organization database administrator uses case tools to improve database planning and design

<u>System developers</u>:-Persons such as analyst, designer's programmers and testers who developer's new application system developers uses case tools for system requirements analysis and design

<u>End users:</u>-Persons throughout the organizations who manipulated data in the database and who requests and receive information from the database.



<u>Components of the database environment</u>

Explain Codd's relational data base rules.

Edgar Frank Codd (19 August 1923 – 18 April 2003) defined 12 important relational database rules that make a database system as relational database system. He defined these rules to standardize the definition of relational database system. Any database system that follows minimum 6 Codd's rules is known as RDBMS.

The information rule: The data stored in a database, may it be user data or metadata, must be a value of some table cell. Everything in a database must be stored in a table format.

The rule of guaranteed access: Every item of data must be logically addressable with the help of a table name, primary key value and column name.

The systematic treatment of null values: The RDBMS must be able to support null values (these values are different from zeroes and spaces) to represent missing or accessible to users with appropriate authority and are stored in the data dictionary.

Active online catalog: - The structure description of the entire database must be stored in an online catalog, known as data dictionary, which can be accessed by authorized users.. These are accessible to users with appropriate authority and are stored in the data dictionary.

Comprehensive data sub language: A relational database may support multiple languages. But it should also support an additional well – defined declarative language that provides support for data definition, integrity constraints, data manipulation, transaction management and view definition.

The view updating rule: All views that are theoretically updatable must also be updatable by the RDBMS.

The High level insert and update rule: A database must support high-level insertion, updating, and deletion. This must not be limited to a single row, that is, it must also support union, intersection and minus operations to yield sets of data records.

The physical independence rule: The data stored in a database must be independent of the applications that access the database. Any change in the physical structure of a database must not have any impact on how the data is being accessed by external applications.

The logical data independence rule: The logical data in a database must be independent of its user's view (application). Any change in logical data must not affect the applications using it. For example, if two tables are merged or one is split into two different tables, there should be no impact or change on the user application. This is one of the most difficult rule to apply

The integrity independence rule: A database must be independent of the application that uses it. All its integrity constraints can be independently modified without the need of any change in the application. This rule makes a database independent of the front-end application and its interface.

The distribution independence rule: The end-user must not be able to see that the data is distributed over various locations. Users should always get the impression that the data is located at one site only. This rule has been regarded as the foundation of distributed database systems.

The non-subversion rule: If the system supports low – level access to the data, there must not be a way to bypass the integrity rules of the database. This is necessary for data integrity.

THE RELATIONAL DATABASE MODEL:

The Relational model represents both data (entities) and relationships among data in the form of tables. Each table has multiple columns and each column has a unique name. Consider the following relational model.

Customer Relation			
Customer_name	**Street**	**City**	**Account No**
Raju	Janathpeta	Kavali	31183960958
Madhu	B.V Nagar	Nellore	31183960959
Aasritha	Ramnagar	Ongole	31196586652

Account Relation	
Account No	**Balance**
31183960958	50000
31183960959	30000
31196586652	25000

The description of data in terms of tables called as relations, from the above customer and account relations, we can make a condition that customer details are maintained in customer relational database and their deposit details are maintained in account relation database.

What is data integrity? Explain rules regarding data integrity.
Write about logical view of data integrity rules.

Data integrity constraints:

It is a correctness and completeness of the data. The data available in the database must be complete and correct data. To maintain correctness and completeness of data oracle has given some data integrity constraints. Relational data base integrity rules are very important to good data base design. Many RDBMSs enforce integrity rules automatically.

Constraint: -It is rule or restriction.

- A set of predefine rules applied on table columns while creating tables or after creation.
- They are automatically activated when ever DML operations are performed on tables.
- They are used to impose to restrictions on table columns.
- They are also activated when tables are manipulated by other users or by other application software's.
- They provide high security on tables.

Constraints are classified into 3 types.

1. Domain integrity constraints
   a. Not null
   b. Check
2. Entity integrity constraints
   a. Unique
   b. Primary key
3. Referential integrity constraints
   a. References

1. Domain integrity constraints: - It is used to restrict duplicate values into table columns. It checks the validity of entries for a given column. We can enforce domain integrity by restricting the type, the format, or the range of possible values.

a. Not null: - It is used to restrict null values, any number of duplicate values allowed.

Example of not null constraint

Sql>create table emp(eno number(3) not null,ename varchar2(12) not null, job varchar2(12)not null, sal number(5) not null,deptno number(2));

b. Check: - It is used to provide conditional restrictions on table columns.

Sql>create table emp3(eno number(3) not null, ename varchar2(12) not null,

job varchar2(12) check(job in('manager','analyst','salesman','clerk')),

sal number(8,2) check(sal between 5000 and 15000),deptno number(2));

2. Entity integrity constraints: - It is used to provide conditional; restrictions on table columns. Entity integrity rules ensure that it should be easy to identify each entity in the database.

a. Unique: - Used to restrict duplicate values but any no.of null values are allowed.

Sql>create table emp3(eno number(3) unique not null,ename varchar2(12) not null, job varchar2(12) check(job in('manager','analyst','salesman','clerk')),sal number(8,2)check(sal between 5000 and 15000),deptno number(2));

b. Primary key: - Not Null + Unique + Index

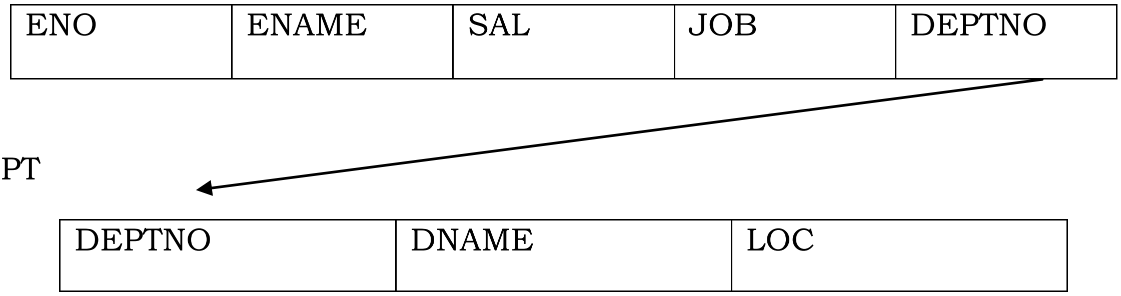
- It is used to define the Key column of a table.
- It can be used only once in Table definition.
- It will not allow Null values and Duplicate values into Key column.
- It is supported with an Index automatically.

Sql>create table emp3(eno number(3) primary key, ename varchar2(12) not null, job varchar2(12)check(job in('manager','analyst','salesman','clerk')),sal number(8,2)check(sal between 5000 and 15000),deptno number(2));

3. Referential integrity constraint: - It is used to establish relation between two tables. To define referential integrity constraint it is necessary to define the foreign key first.

Foreign key: Used to define relationship between 2 Tables. It allows Null and duplicates values. It can be related to either Primary key or unique constraint column of other Table.

PK / UNQ <-----> FK



EMPLOYEE

ENO	ENAME	SAL	JOB	DEPTNO

DEPT

DEPTNO	DNAME	LOC

Sql>create table emp3(eno number(3) primary key, ename varchar2(12) not null, job varchar2(12)check(job in('manager','analyst','salesman','clerk')),sal number(8,2)check(sal between 5000 and 15000),deptno number(2) constraint emp_deptno_fk references dept(deptno));

Explain about Relational Set Operators.

Relational algebra defines theoretical way of manipulating capabilities of the relationalmodel using the eight
relational operators:
1. UNION
2. INTERSECT
3. DIFFERENCE
4. PRODUCT
5. SELECT
6. PROJECT
7. DIVIDE
8. JOIN

UNION: It combines all rows from two tables without duplicate rows. The tables must have the sameattribute characteristics to be used in the UNION. When two ormore tables share the same number of columns, when the columns have the same names and when they sharethe same (or compatible) domain, they are said to be union-compatible.
INTERSECT: It gives only the rows that appear in both tables. Thetables have same structure. we cannot use INTERSECT if one of theattributes is numeric and one is character-based.
DIFFERENCE: This operator returns all the rows in the first table minus rows in the second table. In other words, it returns the rows present in the first table but not present in the second table.
PRODUCT: It returns possible pairs of rows from two tables also known as the Cartesian product. Thereforeif one table has six rows and the other has three rows, the PRODUCT gives a list composed of 6 X 3 = 18rows.
SELECT: It is also known as RESTRICT. It returns values for all rows found in a table that satisfy a condition.SELECT can be used to list all the rows values or it can yield only those row values that match a specifiedcriterion. In other words, SELECT returns a horizontal subset of the table.
PROJECT: It yields all values for a selected attributes. In other words, PROJECT yields a vertical subset ofa table.
DIVIDE: The DIVIDE operation uses one single-column table (i.e. column a) as the divisor and one 2-column table (i.e. columns a and b) as the dividend. The tables must have a common column (i.e. column a).The output of the DIVIDE operation is a single column with the values of column ‘a’ from the dividend tablerows where the value of the common column (i.e. column a) in both tables match.
JOIN: It allows information to be combined from two or more tables. JOIN is the real power behind therelational database, allowing the use of independent tables linked by common attributes.

Explain relationships within the relational data base.
Explain various relationship cardinalities.

A relationship is a meaningful association among several entities, relationship cardinalities or cardinality ratios express the number of entities to which another entity can be associated through a relationship.
Relationship cardinality is very important in relational data modeling to build a logical and well structured data base design.

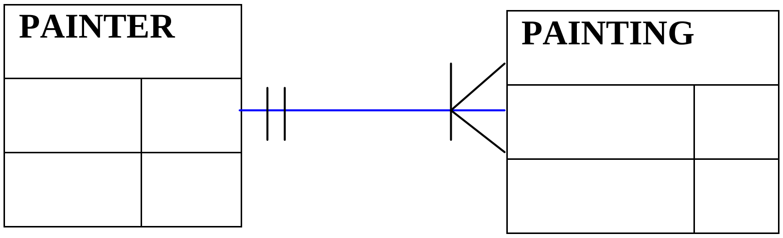
The number of entities that participate in a relationship set is called the degree of relationship. Based on this degree we can categorize relationship into unary, binary and ternary relationships. Usually the relationships in relational data model are binary relationships.

Based on the cardinality of relationships they are classified into the following categories.

1. One – to – many relationship (1:M)
2. Many- to –many relationship (M:M)
3. One – to – one relationship (1:1)

Example:

A painter painting many different paintings but each one of them is painted by only one painter.

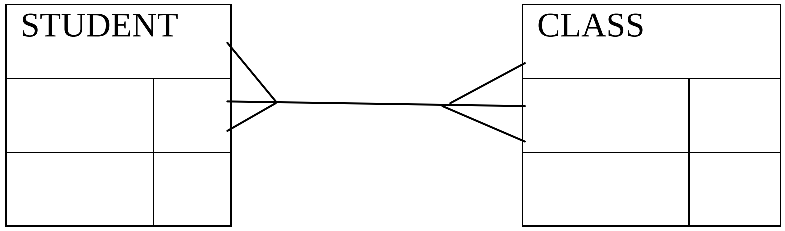


ONE TO MANY RELATIONSHIP BETWEEN COURSE AND CLASS

2. MANY- TO- MANY RELATIONSHIP: If an entity in A is associated with multiple entities in B and an entity in B is also associated with multiple entities in A, then the relationship is called M: N relationship.

Example:

A student can take many classes and each class can be taken by many students.

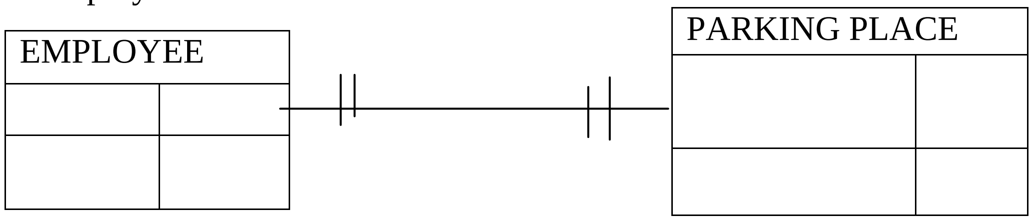


MANY TO MANY RELATIONSHIP BETWEEN STUDENT AND CLASS

ONE –TO- ONE RELATIONSHIP: If an entity in A is associated with atmost one entity in B and an entity in B is associated with at most one entity in A, then the relationship is called 1:1 relationship.

Example:

Each employee is assigned exact only one parking place and each parking place must be assigned one employee.



ONE TO ONE RELATIONSHIP BETWEEN EMPLOYUEE AND PARKING PLACE

UNIT –III

ENTITY – RELATIONSHIP MODEL

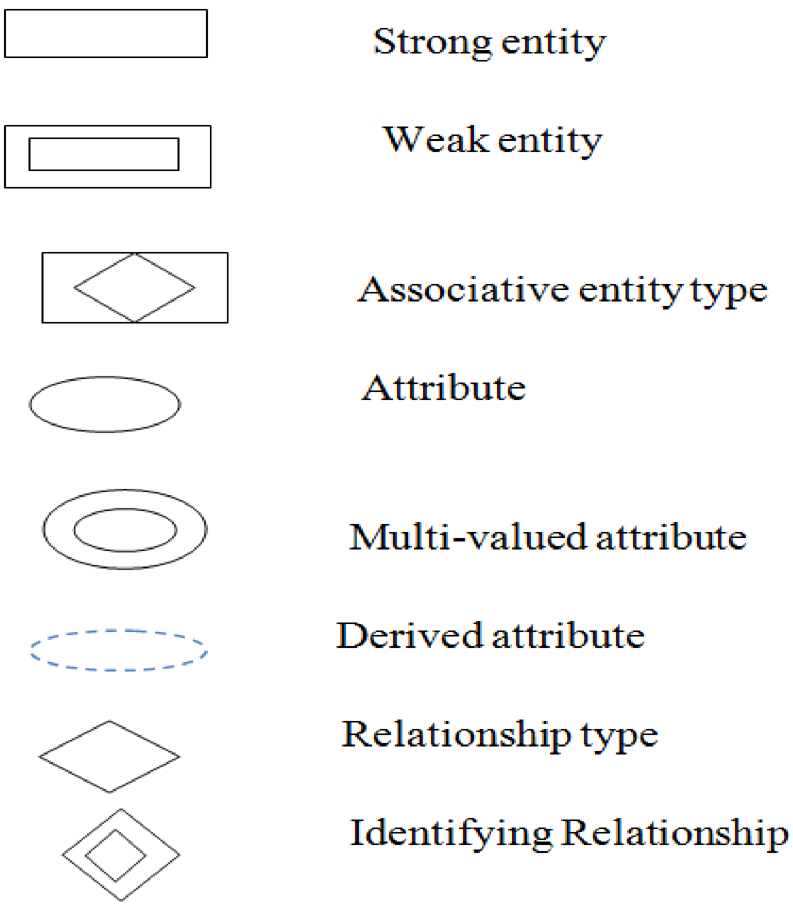
ER Modeling:

The Entity relationship model is a detail graphical representation of data from an organization. An ER Model is normally expressed in terms of ER diagrams, which is graphical representation of an ER Model. The ER model is expressed in terms of entities, relationship among these entities and their relationships.

Benefits of ER Modeling

- Documents information for the organization in clear, precise format.
- Provides an easily understood pictorial map for the database design.

The base notations for an ER model are as follows.



Write about entity relationship model. (OR)

Explain the constructs of ER model. (OR)

Write about Components of ER Modeling.

The Entity relationship model is a detail graphical representation of data from an organization. An ER Model is normally expressed in terms of ER diagrams, which is graphical representation of an ER Model. The ERD represents the conceptual database as viewed by the end user.

ER Model components are three types they are:

1. Entities
2. Attributes
3. Relationships

1. Entities:

It can be anything in the real world, which has got certain properties, which is identified.

Entity can be a person or location or organization or thing, for which the database system store the information. Some examples of each of these types of entities are as follows.

PERSON: Employee, Student, Patient, Customer etc.

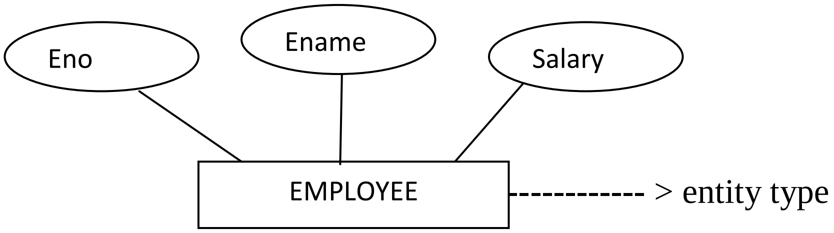
EVENT: Sales, Purchase, Registrations etc.

OBJECT: Machine, building, Automobile etc.

PLACE: Store, State etc.

CONCEPT: Account, Course etc

Entity type: - An entity type is a collection entities that can be share common properties. Each entity type in an ER Diagram is given a name. Entity type name must be in capital letters. In an entity Relationship diagram entity type names must be placed inside the rectangular box.



Entity instance: - An entity instance is a single occurrence (record) of an entity type.

EMPLOYEE---------> Entity type

Eno	Ename	Salary ---------->attribute	
1	Madhu	30000	
2	Uma	15000	**Entity instance**
3	Harini	70000	

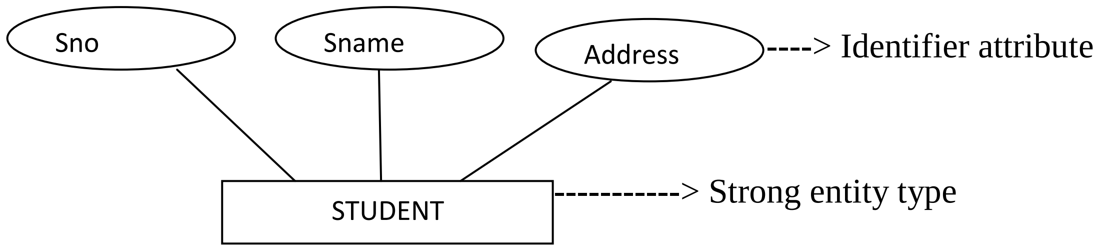
There are two types of Entities. They are as follows,

1. Strong entity
2. Weak entity

Strong entity: - Strong entity is the one that does not depend on other entities. A strong entity type has one or more identifier attributes. Most of the entity types in an organization are strong entity type. In ER diagrams strong entity types are indicated by rectangle.

For example, a chairman of a company does not depend on anyone for final decisions. Hence, chairman is strong entity.

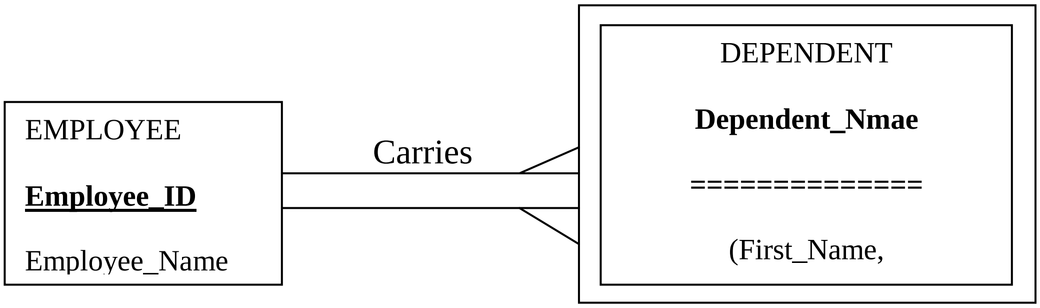
Example



Weak entity: - Weak entity is the one that depends on other entities for existence. For example, if an employee is retired then we do not need to store the details of his dependents. Hence the dependents (children) entity is weak entity.

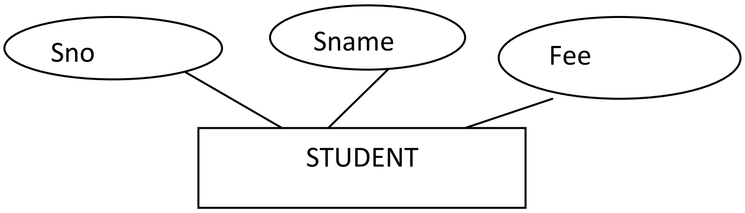
Example: - CUSTOMER, STUDENT, COLLEGE, EMPLOYEE

Example:

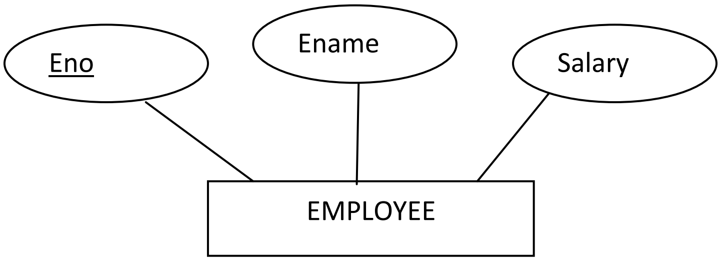


2. Attributes: - The characteristic of an entity is known as property or attribute. Table is an entity set. In Table each record represents one real time entity.

For example the student entity includes, among many others, the attributes STU_LNAME, STU_FNAME, and STU_INITIAL. In the original Chen notation, attributes are represented by ovals and are connected to the entity rectangle with a line.



Identifier attribute: - An attribute is uniquely identifies individual entity instances of an entity type. In ER Diagrams identifier attributes can be underline.

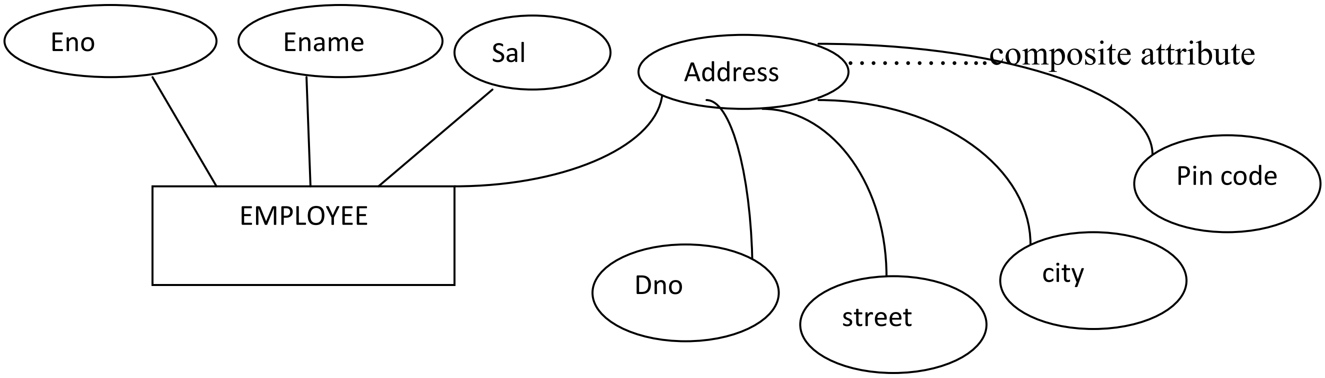


Required attributes: - An attribute that must have a value for every entity (or relationship) instance.

Optional attribute: - An attribute that may not have a value for every entity (or relationship) instance.

Simple attributes: - These attributes are also called as atomic attributes. These cannot be subdivided further. For example, the attributes like, roll number and class of ‘student’ entity cannot be further subdivided.

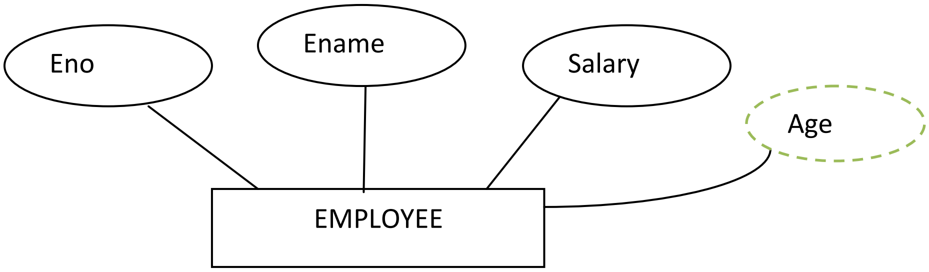
Composite attributes: - An attribute which can be further divided into smaller components are called composite attributes. For example, the attribute name can be further divided into first name, middle name and last name. Similarly, the attribute address can be further divided into house number, city, street, country.



Single – valued attribute: - Certain attribute take only a single value in all instances. For example, the age of the person is a single – valued attribute as man cannot have two ages.

Multi - valued attributes: - Attributes that can have more than one value at a time for an instance are called multi-valued attributes. For example, the color of the product. A product might be multicolored in which case it takes more than one value at a time.

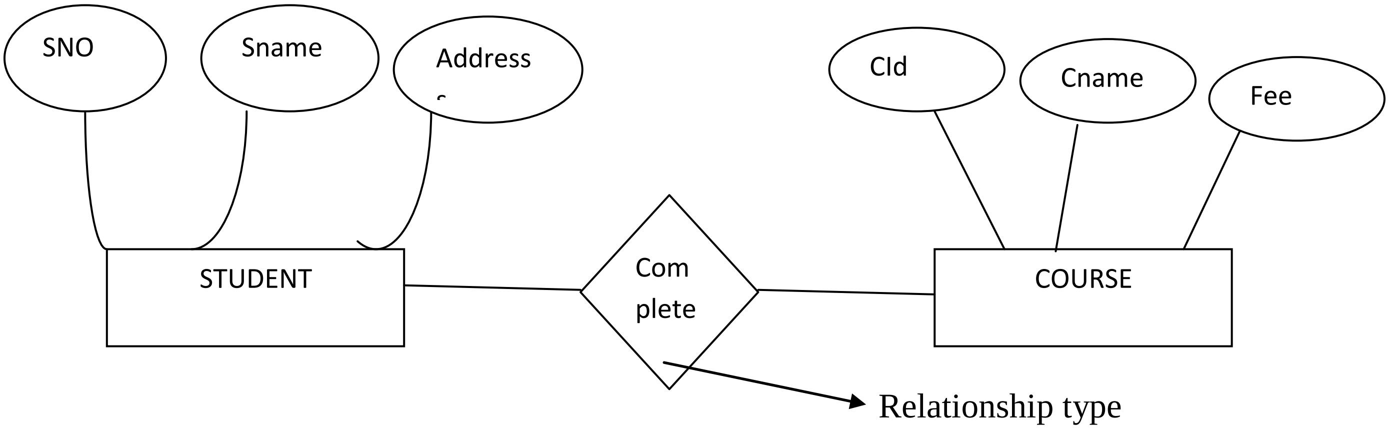
Stored and derived attribute: - Some attributes need not be stored but can be derived from available other attributes. For example, the total number of students in a class can be calculated by calculating the number of student records. Similarly the age of the student can be calculated, by subtracting the date of birth field from present date. The age field is called **derived attribute**. And date of birth is called **stored attribute**.



3. Relationships: - A relationship is an association between entities. The entities that participate in a relationship are also known as participants. For example a STUDENT takes CLASS, a PROFESSOR teaches a CLASS, a DEPARTMENT

employs a FOFESSOR. Relationship between entities always operates in both directions.
A CUSTOMER may generate many INVOICEs.
Each INVOICE is generated by one CUSTOMER.

Relation type: - A relation type is a meaningful association between the entity types. In ER Diagrams relationship types are indicated by diamond symbol.



Explain relationship degree in detail.

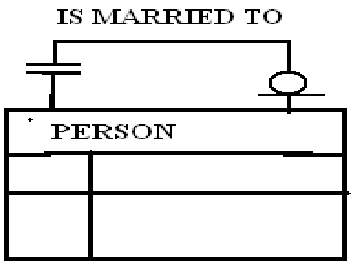
Relationship degree:

The degree of a Relationship is the no. Of entity types that participate in that relationship. In ER model, there are 3 most common relationship degrees. They are.

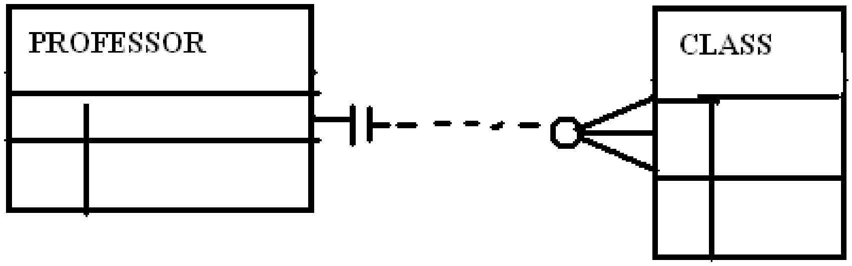
1. Unary Relationship (1^0 degree relationship)
2. Binary relationship (2^0 degree relationship)
3. Ternary relationship (3^0 degree relationship)

1. Unary relationship: - Unary relationship is a relationship between the maintenance of one entity type. Unary relationship is also called as recursive relationship.

In the below case "is married to" relationship means that PERSON requires another PERSON to be married. That is PERSON has a relationship with itself. Such a relationship is known as recursive relationship.



2. Binary relationship: - Binary relationship is a relationship between the instances of two entity types. It is the most common type of relationship in ER model. A binary relationship associated when two entities are associated in a relationship.binary relationships are most common. In the below case “a PROFESSOR teaches one or more CLASSES” represents a binary relationship.

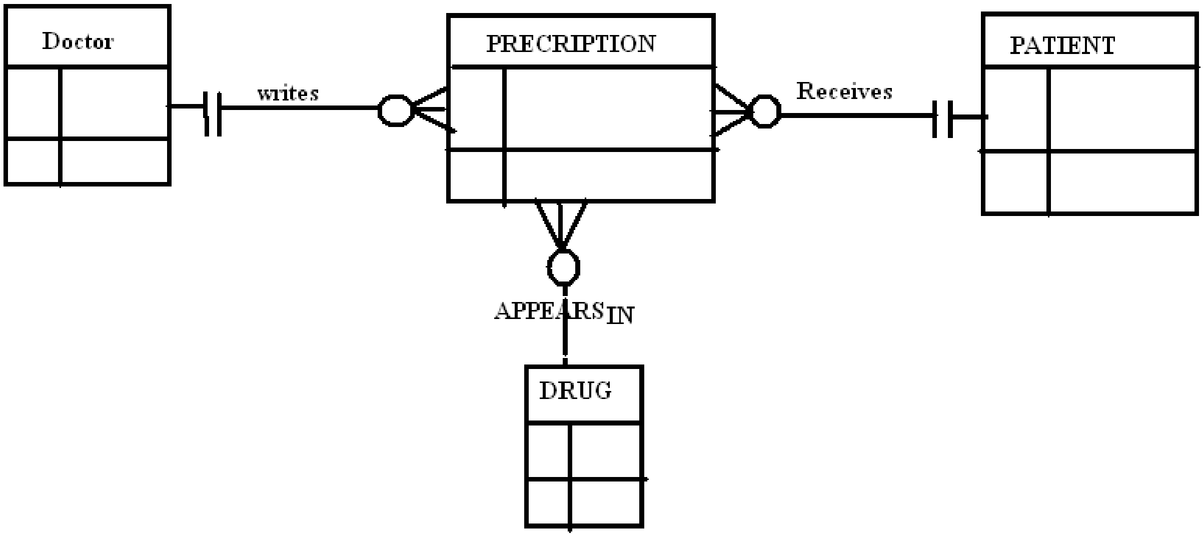


3. Ternary Relationship: - A Ternary Relationship is a relationship between the instances of thee entity type. In the below case

A DOCTOR writes one or more PRESCRIPTIONS

A PATIENT may receive one or more PRESCRIPTIONS

A DRUG may appear in one or more PRESCRIPTIONS



What is normalization? Explain the need for normalization in database design.

In the logical database design, we transform the ER diagrams into relations. Before proceeding with the physical database design we need a method to validate the logical design. Normalization is a primary tool to validate and improve the logical design. So Normalization is a systematic approach of decomposing tables to eliminate data redundancy and undesirable characteristics like Insertion, Update and Deletion Anomalies. It is a multi-step process that puts data into tabular form by removing duplicated data from the relation tables.

Normalization is used for mainly two purposes,

- Eliminating redundant (useless) data.
- The data must be stored in logical order.

Need for normalization: **-** If a table have redundant data, the users have to face the errors (Or) inconsistency at the time of modifying the data. These types of errors are called ‘anomalies’ generally anomalies are three types these are

1. Insertion anomaly
2. Deletion anomaly
3. Update anomaly

Problem without Normalization

Without Normalization, it becomes difficult to handle and update the database, without facing data loss. Insertion, Updation and Deletion Anamolies are very frequent if Database is not Normalized. To understand these anomalies let us take an example of **Student** table.

S_id	S_Name	S_Address	Subject_opted
401	Aravind	Hyderabad	Comp
402	Ashok	Nellore	Maths
403	Santhi	Amaravathi	Maths
404	Aravind	Hyderabad	Physics

Updation Anamoly : To update address of a student who occurs twice or more than twice in a table, we will have to update **S_Address** column in all the rows, else data will become inconsistent.

Insertion Anamoly : Suppose for a new admission, we have a Student id(S_id), name and address of a student but if student has not opted for any subjects yet then we have to insert **NULL** there, leading to Insertion Anamoly.

Deletion Anamoly : If (S_id) 401 has only one subject and temporarily he drops it, when we delete that row, entire student record will be deleted along with it.

Explain about Basic normal forms:

1. First normal form (1 NF)
2. Second normal form (2 NF)
3. Third normal form (3NF)

First normal form (1NF):- A relation schema is said to be in First normal form if the values in the relation are atomic. In simple words, there should be no repeating groups in particular column. A value can be defined as an atomic value if it doesn't contain any multi valued attribute and no composite attribute.

Ex:

For example consider a table which is not in First normal form.

Student Table:

Student	**Age**	**Subject**
Aravind	15	Comp, Maths
Ashok	14	Maths
Santhi	17	Maths

In First Normal Form, any row must not have a column in which more than one value is saved, like separated with commas. Rather than that, we must separate such data into multiple rows.

Student Table following 1NF will be :

Student	**Age**	**Subject**
Aravind	15	Comp
Aravind	15	Maths
Ashok	14	Maths
Santhi	17	Maths

Using the First Normal Form, data redundancy increases, as there will be many columns with same data in multiple rows but each row as a whole will be unique.

Rule: Any multi- valued attributes (repeating groups) have been removed, so there is a single value at the intersection of each row and column of the table

Functional dependency:

Normalization is based on the analysis of functional dependency. A functional dependency is a constraint between two or more attributes.

For example in a relation ‘R’ attributes ‘B’is functionally dependent on the attribute ‘A’ means the value of ‘A’ uniquely determines the vale of ‘B’. The functional dependency is represented by a right arrow (→).

A→B

An attribute may be functionally dependence on two or more attributes. For example the relation STUDENT contains attribute like Sno, Sname, course_id and date completed.

In the above example, the attribute date _complete is functionally dependent on course_Id.

Sno, Course_Id →date completed.

Determinant:- The attributes on the left hand side of the arrow in a functional dependency is called determinant. In the above example determinants are Sno, Course_Id.

Second normal form (2 NF):-A relation is second normal form, if it is in first normal form and contains no partial functional dependency. That means every non-key attribute is fully functionally dependent on the full set of primary key attribute. If a relation is in second normal form if any one of the following conditions applies

i. A relation contains only one primary key.
ii. No non-key attribute in the relation.

For example, a relation student contains attribute like Sno, Sname, Course_Id and date completed. In this example, the primary key is a composite key of Sno, course_Id. Here the relation STUDENT is not in second normal form because there is a partial functional dependency. Here the non-key attributes Sname, Address is functionally dependent on the part of the primary key (Sno). So we decompose the above relation into new relations.

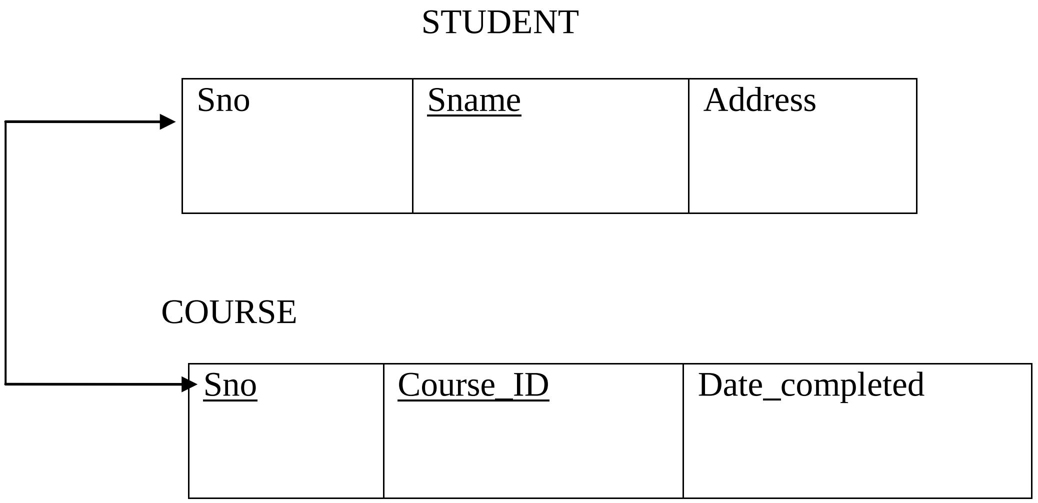
STUDENT

Sno	Sname	Address	Course_ID	Date_completed

Sno ------------> Sname (partial functional dependency)

Sno ------------> Address (Partial functional dependency)

Sno, Course_Id ----------------->Date _completed (fully functional dependency)



STUDENT

Sno	Sname	Address

COURSE

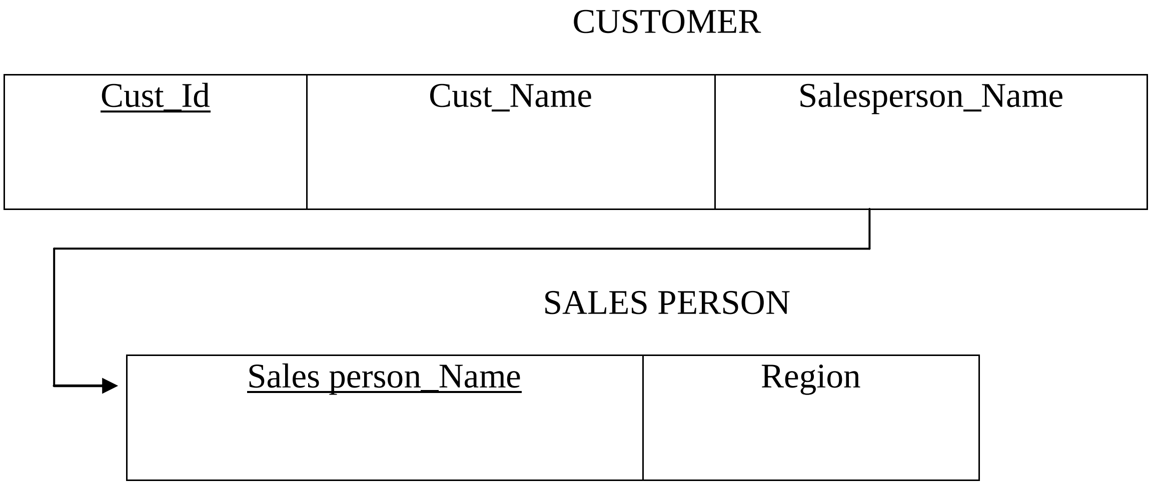
Sno	Course_ID	Date_completed

Third Normal form:- if a relation is in 3rd normal form, if it is in second normal form and contains no transitive dependency. Here transitive dependency means a functional dependency between two non – key attributes. For example consider a relation CUSTOMER with attributes like Cust_id, cust_name, sales person name and region. Here the primary attribute is ‘Cust_id’.

CUSTOMER

Cust_Id	Cust_name	Sales person_name	Region

The above relation contains a transitive dependency because the non-key attribute region is functionally dependent on another non-key attribute sales person_Name. So the relation customer is not in third normal form. So we decompose the above relation into two meaningful relations.



CUSTOMER

Cust_Id	Cust_Name	Salesperson_Name

SALES PERSON

Sales person_Name	Region

Write about Boyce-codd normal form (BCNF):-

A relation is in Boyce-codd normal form, if it is in third normal form and all determinants are cardinalities keys (primary keys).

For example consider a relation STUDENT with attributes like Sno, Subject and Advisor. Here primary key is a composite key of Sno, Subject. The above relation contains two functional dependencies.

STUDENT

Sno	Advisor	Subject

Sno, Advisor ------------------->Subject

Subject -------------------------> Advisor

In the above example, in the second functional dependency determinant is subject. It is not a candidate key. So the relation STUDENT is not in Boyce Codd normal form. So we decompose the above relation into two meaningful relations.

STUDENT

Sno	Advisor

ADVISOR

Advisor	Subject