UNIT-1

INTRODUCTION TO DBMS AND ER MODEL

Syllabus:

- Introduction to database systems,
- File System vs. Database Systems,
- Database system structure,
- Views of data in a database system,
- Data models and Database languages.
- Introduction to Entity-Relationship data model,
- Elements of an ER model,
- Constructing ER diagrams,
- Modelling of constraints,
- Reduction of ER diagrams to relational tables.

INTRODUCTION TO DBMS AND ER MODEL

DATA:
It is any fact that can be recorded (or) it is a raw material that can be processed on any computer machine.

EXAMPLE: Text(names),Numbers(Phone_numbers),Images,Pictures,Videos, Speech etc.....

DATA BASE (DB): It is collection ofrelated data and describe the activities of one (or) more organization .

EXAMPLE:

- ➔ University database contains entities such as student,courses, faculty and class rooms.
- ➔ Relationship between entities such as student enrollment in courses,faculty teaching courses and use class rooms for courses.
- ➔ They are different kinds of data base:
  1.Traditional Data Base
  2.Multimedia Data Base
  3.GIS Data Base
  4.Relational Data Base
  5.Data ware Housing

1.Traditional Data Base:
It contains names and numbers.

2.Multimedia Data Base:

It contains videos like speech ,songs,movies etc.....

EX: YOU TUBE

3.GIS (Geographic Information System)Data Base:

It contains images of satilite systems (or) other things.

EX:NASA having GIS

4.Relational Data Base:

It contians tables with rows and columns in 2D form.

5.Data Ware Housing:

It us a kind of Data Base and store huge amount of data.

EX: Any organization with past 100 years data.

DATA BASE MANAGEMENT SYSTEM(DBMS):

It is a software (or) Set of programmes that allow user to define,create and maintain a Data Base and provided control access of data.

EX: mysql , DB2 , oraclle long , 11g etc....

DATA BASE SYSTEM (DBS):-It is collection of data base and data base managment system.

DBS = DB + DBMS

EX: DBS= notebook+pen (without computer)

*Where DBMS being used? (or) **Applications of DBMS:**

1. **Airlines:** Reservations,Schedules etc.
2. **Telecom:** Callsnade,Customer Details,Network usage.
3. **Universities:** Registration,Grades,Result etc.
4. **Banking:** All trasactions.
5. **Sales:** products,purchases,customer etc.
6. **Finance:** It is like storing sales,Holding information,Finance statements etc.
7. **Online Shopping:** It has become a very trend of modern days.It contains purchase information,Invites,Payments.
8. **Social media sites:** It contains accounts like Facebook,Twitter,Google+ etc.
9. **Library Management System:** It contains book issue dates,Name of the book,author,availability of books etc.
10. **Railway Management System:** It contains records of ticket booking, Train departure,Railway status etc.
11. **Human Resource Management:** It contains Employee salary,tax etc.
12. **Manufacturing department:** It contains product details like quantity,bill purchase,expiry dates etc.
13. **Military System:** It contains millions of soldiers.they keep files in secure and safe mode.

File System Vs Database System:

- **File:** It is a sequence of records stored in binary format.

- **File System:** It is a method of Storing and Organising the computer files and whatever data they contain easy to find and access them.

Characteristics of File System:

➔ It is a group of files storing the data of any Oraganisation.
➔ Each file is called as “Flat” file.(There is no particular structure and format).
➔ Files are designing by using programming language such as C,C++.

Limitations/Disadvantages of File System:

(1)Seperated and isolated data.
(2)Duplication of data.
➔ It takes additional storage space.
➔ There is a loss of data integrity.
➔ It is most cost and time.

(3)Data Dependency:

One file is depended on another file and these are contains paticular physical format that are developed by programmers.

(4)Data Redundancy:

It means the same amount of data stored in a different places that is data redundancy.

(5)Data Security:

The security of data is low in the file system that can be easy to accesible.

(6)Transaction Problem:

File systems are not support the transaction and ACID properties.

A - Atomicity
C - Consistency
I - Isolation
D – Durability

(7)Concurrent Problems:

- ➔ When multiple users acces the same amount of data at same interval of time is called as concurrent operating system.
- ➔ When two or more users read the data simultaneously then there is no problem but when they like to update the file,there is a problem.

ADVANTAGES OF DATABASE SYSTEM

To overcome the properties of file system,we use database system.

(1)Data Independency:

The DBMS provides an abstract view of data and hide such details.

(2)Reduce DataRedundancy:

- ➔ In file system the data can be stored in a different places and there is a chances of duplication of data and also there is a waste of space.
- ➔ But in DBS ,the files are stored in single database.The hold data is stored only once so there is no chance of duplication of data.

(3)Sharing of Data:

- ➔ In DBS,the data can be shared between authorised users of DB.
- ➔ All the users have own rights to access the data.
- ➔ The DBA(Database Administrator)has complete access of DB and he can assign user to access the data.

(4)Data Consistency:

It means control the data redundancy which is mean that sysytem with high data consistency.

(5)Data Integrity:

It means that data in database is always accurate or correct that can be decrease the duplication of data also data inconsistency.

(6)Security:

- ➔ It means protecting your data from unauthorised users.
- ➔ Data in a DB we kept secure and safe.

➔ Only authorised users should be grant to access the DB.

(7)Providing Backup and Recovery:

The Backup and recovery of DBS from software/hardware failures with facilities like the recover the data and backup the data.

DISADVANTAGES OF DATABASE SYSTEM:

(1) Increased cost:

Database System required sophisticated modern hardware/software with more cost.

(2)Complexity:

Developers,Designers,DBA and endusers of DB must have complete skills if they want to use it properly otherwise there is a chances of loss of data (or) database failure.

(3)Technical Staff Requirement:

Any organisation have many employees,it is not easy to for work on DBMS and they well known in DBS.

(4)DataBase Failure:

All the files are stored in a single DB.So, there is a chances of failure become more.Any accidental failure of component may cause loss of valuable data.

(5)Size:

A DBMS become big software,Lots of space and the memory to run its applications and gains bigger size.

(6)Currency Maintainence:

DBMS should be update according to the current scenario and new threats comes daily,so DBMS update itself daily.

(7)Performance:

The total data can be stored in a same DB and there is a chances of low in the perforamance of Computer System.

COMPONENTS OF DATABASE SYSTEM:

There are 4 types of components of database.

(1)Data
(2)Hardware
(3)Software
(4)Users

(1)Data:

Data stored in database include numerical and non-numerical data,audio,video etc.

(2)Hardware:

It includes Various storage devices,input and output devices.

(3)Software:

It includes operating systems,Network Software and application programs.

(4)Users:

There are different types os database users in DBS like DBA(Database Administrator),Database designer,Endusers and Application users.

DATABASE USERS:

There are two types of database users.

1. Actors on the scene
2. Workers behind the scene

Actors on the scene:

Those who are use and control the database content and those who design,develop and maintain database application are called as Actors on the scene.

Database Administrator(DBA):

A person who performs all activities related to maintaining a database responsibilies includes designing,implementating and maintaining a database.

Responsibilies of DBA(or)functions of DBA:

1.Installing and upgrading the sql server:

DBA responsible for installing sql server(or)upgrading sql server with service tax.

2.Monitoring sql server:

The DBA also responsible for sql server is running with particular performance or not and monitor each and every time.

3.Using storage properly:

It means proper use of storage with space requirement and adding new storage space (disk drives)when required.

4.Performing Backup and Recovery duties:

The DBA responsible for Backup and Recovery of software or hardware failures.

5.Managing database users and security:

The DBA is also responsible for assigning database users and proper security level for each user.

6.Working Developers:

The DBA responsible for a person well-known in sql server.

7.Transferring of data:

The DBA responsible for importing and exporting the data from the sql server.

8.Providing 24 hours access:

The database server must stay up,and always protected and provide information to database users.

9.Data ware housing:

It provides huge amount of storage like past hundred years data.

2.DATABASE DESIGNERS:

They are responsibel for identifying the how much of data can stored and way to organising.

3.END USERS:

- They are responsible for access the database for quering,updating and report generation.

They are different types of End users.....

(a) Casual End users:

They use database occationally and need different information each time using sql.

Ex: Middle(or)high-level managers.

(b)Naive/Parametric Endusers:

They are biggest group of users and frequently update the database using “canned trasactions”.

Ex:

1.Bank tellers check account balance,withdraw,deposites.
2.Reservations gor airlines and hotels etc checks the availability of seats or rooms and make reservations.

(c)Sophisticated Endusers:

Engineers,Scientists,Business Analyst are came under “Sophisticated Endusers”.

(d)Standalone Endusers:

They are maintain personal database with particular software package.

Ex: Financial

(4)System Analyst,Application Programmers and Software Engineers:

(a) System Analyst: They are determine the needs of particular endusers particularly “Naive endusers”.

(b) Application Programmers:

They always implement,test,document and maitain a program.

2.Workers Behind the Scene:

Those eho design and develop the DBMS software and related tools and computer system operators.

(a)Tool Developers:

Design and develop the software tools related to DBS design,performance ,monitoring etc.

(b)Operators:

They are responsible for day to day operations of Computer System.

DATABASE MODELS:

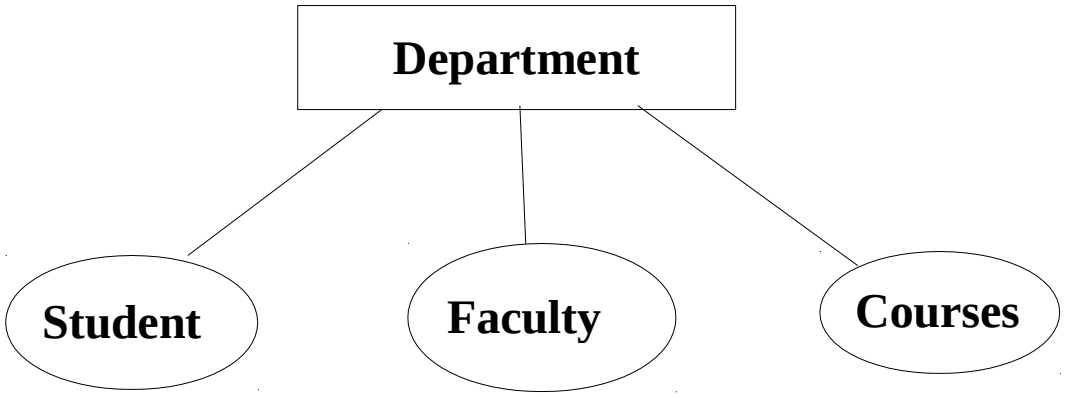
It is design and structure of database and defining how data will be stored,accessed and updated in a DBMS.
There are different database models in DBMS:

(1)Hierarchiral Model
(2)Network Model
(3)Entity-Relationship Model
(4)Relational Model

(1)Hierarchiral Model:

- ➔ In this model data in the form of tree like structure with a single root to which all other data is linked.
- ➔ In this model childs have single parent node.
- ➔ In this model data organised in One-to-Many Relationship.

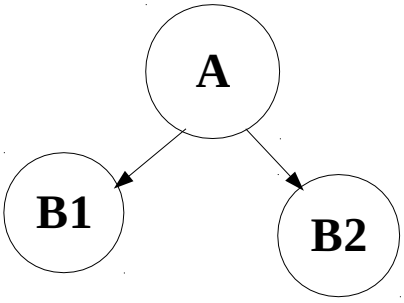
EX:

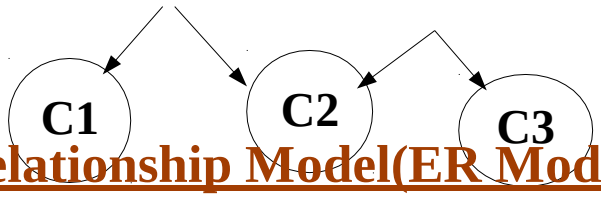


(2)Network Model:

- ➔ It is extension of Hierarchical Model.
- ➔ It is a graph like structure.
- ➔ The parent may have multiple childs also the child may have multiple parents.
- ➔ In this model data organised in Many-to-Many Relationship.

Ex:

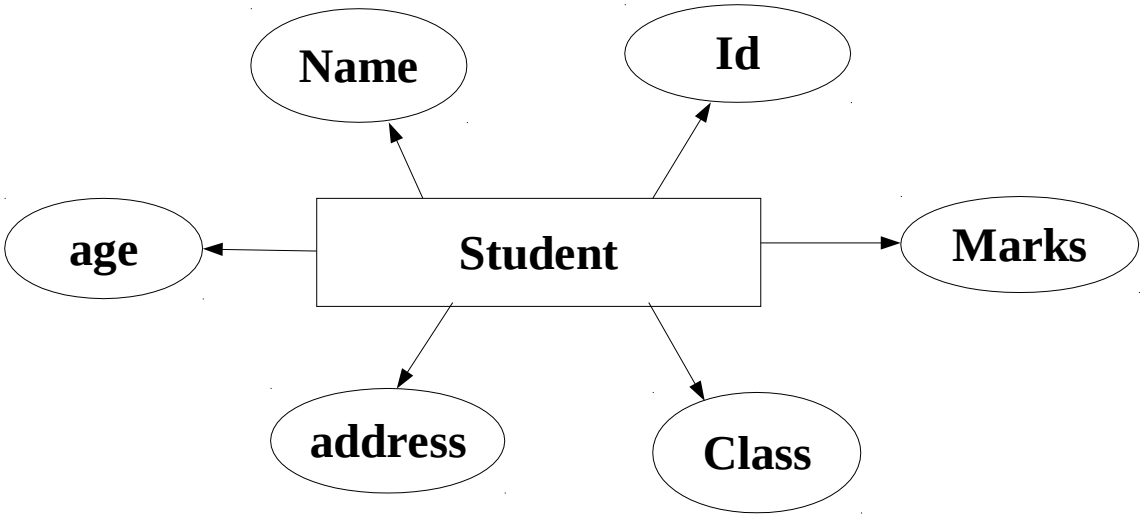




(3)Entity-Relationship Model(ER Model):

- ➔ The pictorial form of data is known as “E-R Model”.
- ➔ In this model data in a database designed in good way.
- ➔ For example, College database contains student as entity and attributes are name,id,age,marks,address etc.

Ex:



(4)Relational Model:

- ➔ In this model data organised in two dimentional table(2D form).
- ➔ It is a collection of table with data and relationship.
- ➔ This model introduced by “EF codd” in 1970.
- ➔ It is most widely used database model.

➔ The basic structure of data in Relational Models in Tables.

➔ Tables also known as “Relation”.

Ex:

Name	Id	Age	Marks	Address
Ram	103	23	89	kadapa
Raj	107	23	92	chittoor
Rani	100	27	90	Nelloor

DATABASE LANGUAGE:

➔ Database Language also called as “SQL”.

➔ SQL stands for Structured Query Language.

➔ It is not a Database,it is a programming language.

➔ It is standard language for Relational Database.

What can SQL can do?

➔ SQL can create new database.

➔ SQL can create new tables in a database.

➔ SQL can insert records in a database.

➔ SQL can delete records in a database.

➔ SQL can update records in a database.

➔ SQL can retrive data from a database.

➔ SQL can execute Queries against database.

There are 4 types of database languages:

(1)DDL- Data Definition Language.

(2)DML- DataManipulation Language.
(3)TCL- Trasaction Control Language.
(4)DCL- Data Control Language.

DDL(Data Definition Language): This Language used create amd modify the structure of database Objects.

(1)Create:

Create command used to used to create a table (or) database.

Syntax: create table table_name(column1 datatype1,column2 datatype2,.....);

Ex: create table student (name varchar(20),sid int,age int);

In the above example student is the table_name and it contains three columns namely are name,id,age.

(2)Alter:

Alter command is used for alternation of data in a database.

- To add column to the exist table.
- To add multiple columns to existing table.
- Drop a column.
- Modify a column.
- Rename a column.

Add column to the existing table:

Syntax: alter table table_name add(column name datatype);

Ex: alter table student add(address varchar(20),marks int);

Drop a column:

Syntax: alter table table_name drop(column_name datatype);

Ex: alter table student drop(age int);

Modify a column:

Syntax: alter table table_name modify(column_name datatype);

Ex: alter table student modify(marks int);

Rename a column:

Syntax: alter table table_name rename(old_column name to new_column name);

Ex: alter table student rename id to sid;

(3)Truncate Command:

- ➔ This command is used for remove all the records from the table but it can't destroy the table structure.
- ➔ When we apply truncate command on a table it's primary key is initialised.

Syntax: Truncate table table_name;

Ex: truncate table student;

Sid	Name	Age
501	--	--
502	--	--
503	--	--

(4)Drop command:

- ➔ This command is used for remove table from the database.
- ➔ It destroy the table structure.

Syntax: drop table table_name;

Ex: drop table student;

(5)Rename command:

This command is used for rename a tablename.

Syntax: rename old_tablename to new_tablename.

Ex: rename student to student_record;

DML(Data Manipulation Language):

- ➔ This Language used for managing (or) changing the data in a database.
- ➔ DML commands are “not auto-committed”.
- ➔ Insert,Select,Delete,Update are the DML commands.

(1)INSERT command:

This command is used for to insert data into a table.

Syntax: insert into table_name
values(data1,data2,. ,datan);

Ex: insert into student values(501, “abc”,20);
insert into student values(502, “xyz”,21);
insert into student values(503, “lmn”,22);

(2)SELECT command:

This command is used to retrive data from database.

Syntax: select *from table_name;

Ex: select *from student;

Sid	Name	Age
501	abc	20
502	xyz	21

503	lmn	22

(3)DELETE command:

This command is used to delete data from table.It also delete particular row.

Syntax: delete from table_name where condition;

Ex: delete from student where sid=502;

Sid	Name	Age
501	abc	20
503	lmn	23

(4)UPDATE command:

This command is used to update a row of a table.

Syntax: update table_name set column_name=value where condition;

Ex: update student set age=27 where sid=501;

select *from student;

Sid	Name	Age
501	abc	27
503	lmn	23

TCL(Trasaction Control Language):

- ➔ There are used to manage the changes made by DML statements.
- ➔ commit,rollback and savepoint are TCL commands.

➔ These are used to control the transaction in a database.

(i) COMMIT: It commits the current transaction means making changes permanently.
Syntax: commit;

(ii) ROLLBACK: It rollbacks the current transaction means cancelling it changes and it restore the database.

Syntax: rollback to savepoint;
Ex: rollback to A;

(iii) SAVEPOINT: It is used to store data permanently.
Syntax: savepoint save_point;
Ex: savepoint A;

Example for TCL:

sid	name
501	abc
502	xyz

insert into student values(503, “lmn”);
commit;
update student set name= “ram” where sid=502;
insert into student values(504, “raj”);
savepoint A;
insert into student values(505, “rock”);
savepoint B;

insert into student values(506, “rajesh”);
savepoint C;
select *from student;

sid	name
501	abc
502	ram
503	lmn
504	raj
505	rock
506	rajesh

rollback to A;
select *from student;

sid	name
501	abc
502	ram
503	lmn
504	raju

DCL(Data Control Language):

- ➔ It is used to provide control the data in a database.
- ➔ grant,revoke are DCL commands.

(i)GRANT:

It is used to provide access (or) privilages (or) permission on the databases objects to the user.

Syntax: grant privilage_name on object_name to {user_name/public/role_name}[with grant option];
where,

Privilage name: Privilage granted to DDL (or) DML commands with user like insert, alter,drop,update etc.

Object Name: It names names of database like name of the table.

User name: It means name of the user who access right is being granted.

Public: It means grant access rights to all users.
With grant option: It allow a user to grant access rights to other user.
Ex: grant select on student to user 1(ram);

(ii)REVOKE:

It is used to removes access rights (or) privilages to the databases objects.
Syntax: revoke privilage_name on object_name from {username / public / role_name }

Ex: revoke select on student from ram;

Views in Database System:

- ➔ It is a kind of virtual table.
- ➔ It also contains rows and columns in real table in the database.

- ➔ It contains equal coulumns or less columns compare with real table.
- ➔ We can creat a view by selecting fields (columns) of one or more table present in a database.
- ➔ A view can either have all the rows of the table or specific rows based on certain condition.

Views are two types:

i. Simple – It takes data from single table.

ii.Complex –It takes data from two tables.

Syntax for creat view:

creat view view_name as select column 1 , column 2,..... from table name where condition;

Here “view_name” is ‘name of the view’,Table name – Name of the table(real table) and condition means to select rows.

Student table (real table)

Sid	Name	Age
501	abc	20
502	def	21
503	ghi	22
504	pqr	23
505	xyz	24

Ex:- Create view student_view as select name , age from student where age>21;
select * from student_view;

Name	Age
Imn	22
pqr	23
xyz	24

Insert rows in a view:

Syntax: Insert into view_name(column 1,column 2) values value1(name),value2(age);
Ex: Insert into student_view(name,age) values (‘jesus’,37);

Output:

Name	Age
Imn	22
pqr	23
xyz	24
jesus	37

Delete from view:

Syntax: Delete from view_name where condition;
select *from student_view;
Ex: Delete from student_view where age=24;
select *from student_view;

Output:

Name	Age

Imn	22
pqr	23
xyz	24

Update the view:

Syntax: update view_name set coumnname= “value” where condition;
select *from student_view;

Ex: update student_view set name= “priya”where age=23;
select *from student_view;

Output:

Name	Age
Imn	22
priya	23
xyz	24

Truncate:

Syntax: truncate view view_name;
select *from view_name;

Ex:
truncate view student_view;
select *from view_name;

Output:

Sid	Name	Age
501	--	--
502	--	--

503	--	--
504	--	--
505	--	--

Drop:

Syntax:drop view view_name;
select *from view_name;
Ex: drop view student_view;
select *from student_view;
Output:
table doesn't exist

Schema and Instance:

- **Schema:**It is a skeleton structure of a data base or logical view of database.
- It defines how the data is organized and how relation among the entities.

Instance: The data stored in database at particular moment of time is called as "Instance"

- Data base schema defines variable declaration in table but database instance gives values to the variables of table.

Data independence:

- The capacity of change the schema at one level without changing schema at the highest level.
- These are two types

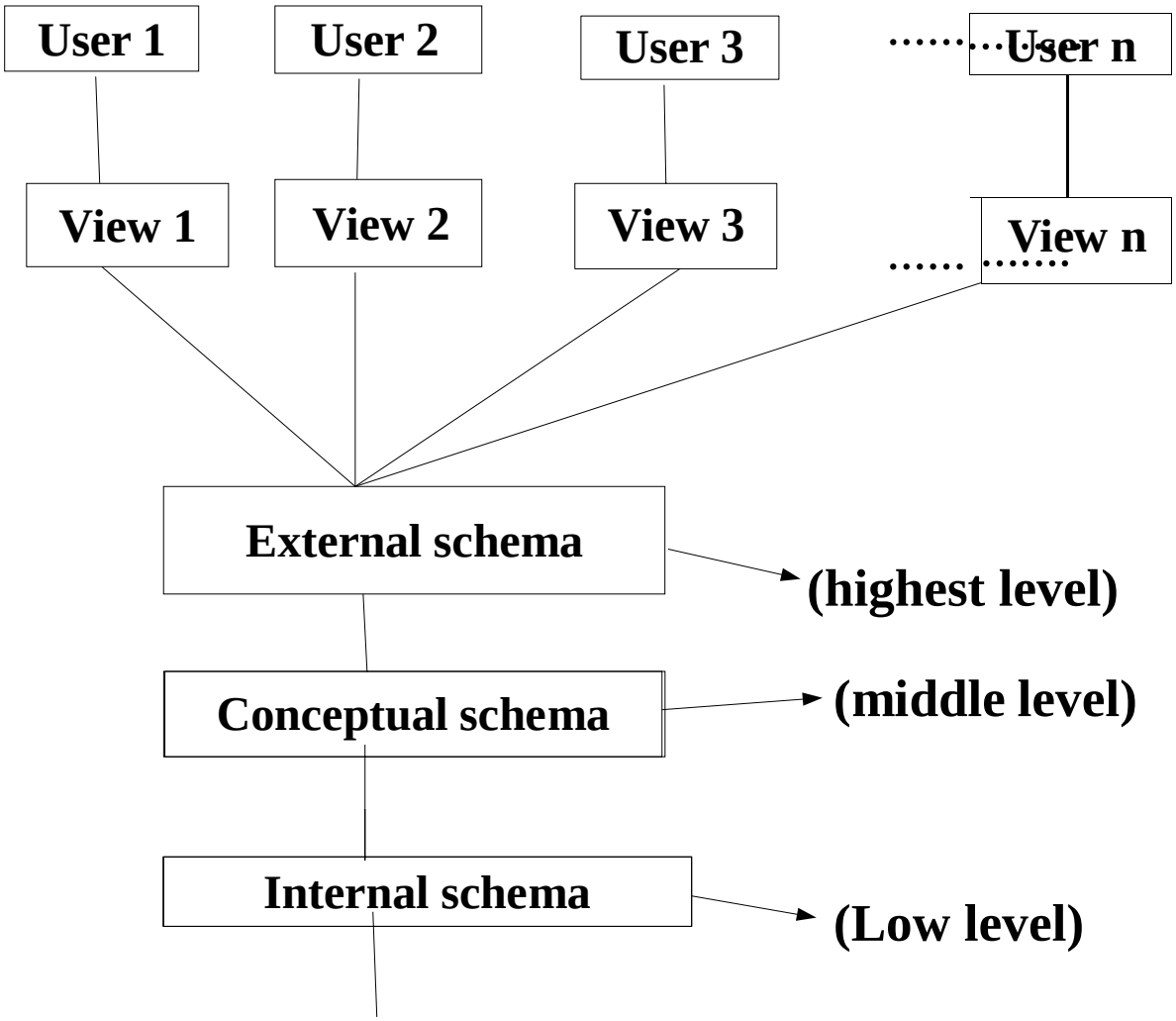
1.Physical data inpedence:

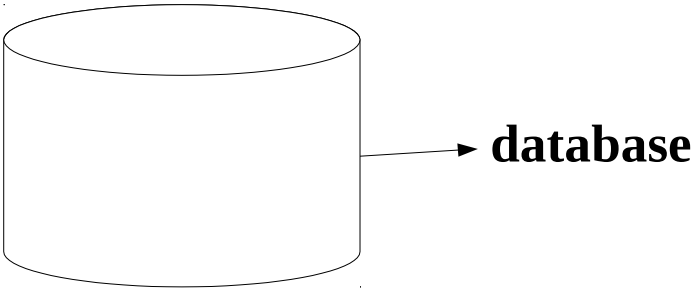
It is the capacity of change the schema at internal level without changes min the schema at conceptual level.

2.Logical data inpedence:

The capacity of change the schema at conceptual level without changes in the schema at external level.

Three tier schema architecture of DBS (or) levels of DBS:





There are three different levels in Database System

1.Internal level
2.Conceptual level
3.External level

(1)INTERNAL LEVEL:

- ➔ It is also called as “physical level” or “low level”.
- ➔ It decribes physical representation of data and also describe additional storage details.
- ➔ How the data can be stored in database can be understand by this ,Record placement.

2.CONCEPTUAL LEVEL:

- ➔ It is also called as “logical level” or “middle level”.
- ➔ It describes logical representation of data that can be stored in database.
- ➔ In relational DBMS the conceptual level describe all the relation stored in database.
- ➔ what data is stored in database and relationship among the data can give this.
- ➔ This level represent entities,attribute,relations etc.

➔ For example university database contains entities such as students,course,faculty,classrooms.

3.EXTERNAL LEVEL:

➔ It is also called as “highest level” .
➔ This level consists of number of different external views od database.
➔ It describe prticular group of users and provide powerful and flexible security to the users.
➔ It permits user to access the data in database in a particular way and same data can be seen by authorized users at same amount of time.

Introduction to E-R model:

E-R definition:

➔ The pictorial or graphical form of data is known as “Entity-Relationship(E-R) model”.
➔ It is conceptual view of database.

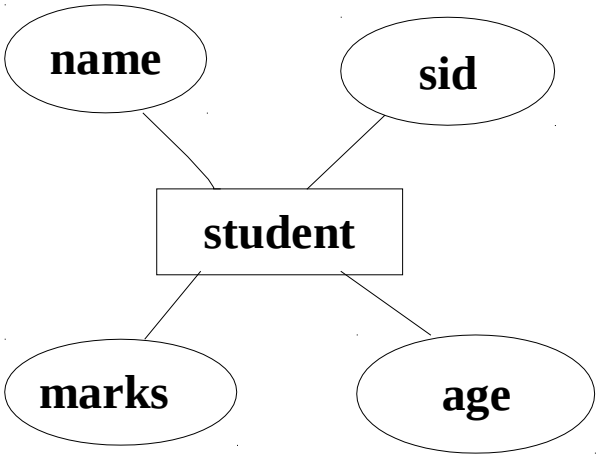
Elements/Concepts of E-R Model:

1.Entity
2.Attributes
3.Relationship

1.Entity:

➔ It is an object in a real world that can be different from other objects.
➔ It is represented as Rectangle ( )

Ex: University database contains entities like students,faculty,courses,class rooms etc.



In above diagram student is an entity.

Entity set:

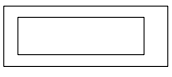
- ➔ It is a collection of entities with similar types.
- ➔ It is extension of entity type.

Entity type:

- ➔ It is a collection of entities that have same attributes.
- ➔ It is intension of entity set.

Ex:E1 is an entity having entity type student and set of all students is called “entity set”.

Weak Entity:

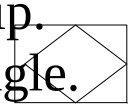
- ➔ The entity doesnot have primary key.
- ➔ It is represented as double rectangle. 

Strong entity:

- ➔ The entity have a primary key is called as “Strong entity”.

Associative entity:

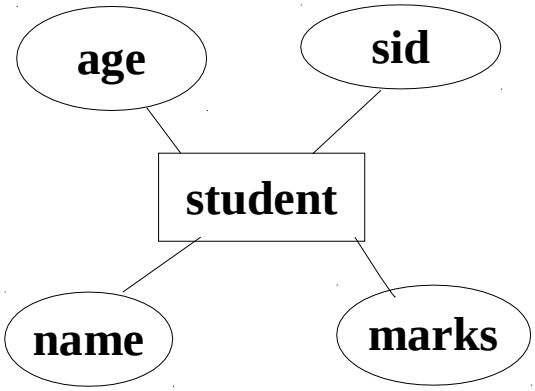
- ➔ The entity which look like a relationship.
- ➔ It is represented as diamond in a rectangle.



2. Attributes:

- ➔ The property or characteristics of an entity is called as "Attributes".
- ➔ It is represented as eclipse or oval in E-R model.

- ➔ **Ex:**student is an entity which contains name,sid,marks,age.



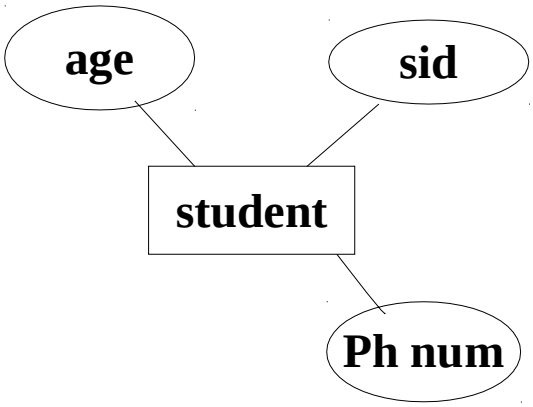
Types of Attributes:

a.Simple vs Composite Attributes
b.Single valued vs Multivalued Attributes
c.Stored vs Derived Attributes
d.Complex Attributes

a.Simple Attributes:

- ➔ Attribute which contains single atomic value which can't be divide further.

Ex: Student id,Student ph.no,ph num etc.

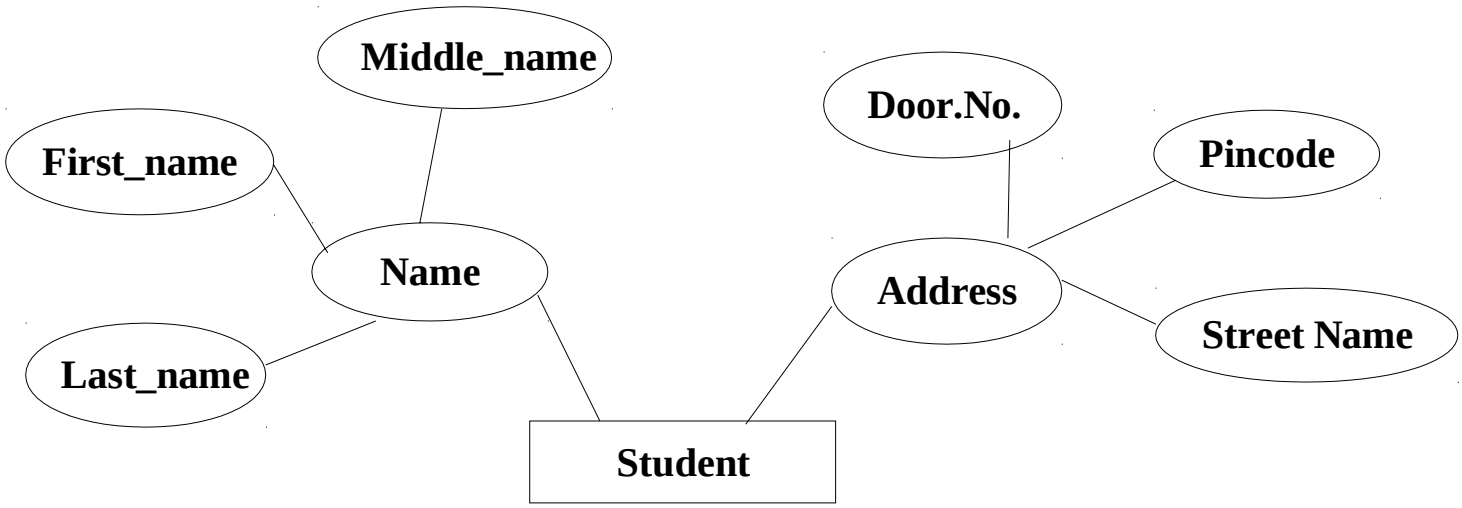


Composite Attributes:

➔ The Attributes which contains no more than one simple attributes .

➔ **Ex:**

- 1)Student complete name contains firstname,last name,middle name.
- 2)Address also contains doorno. ,street name,pincode etc.



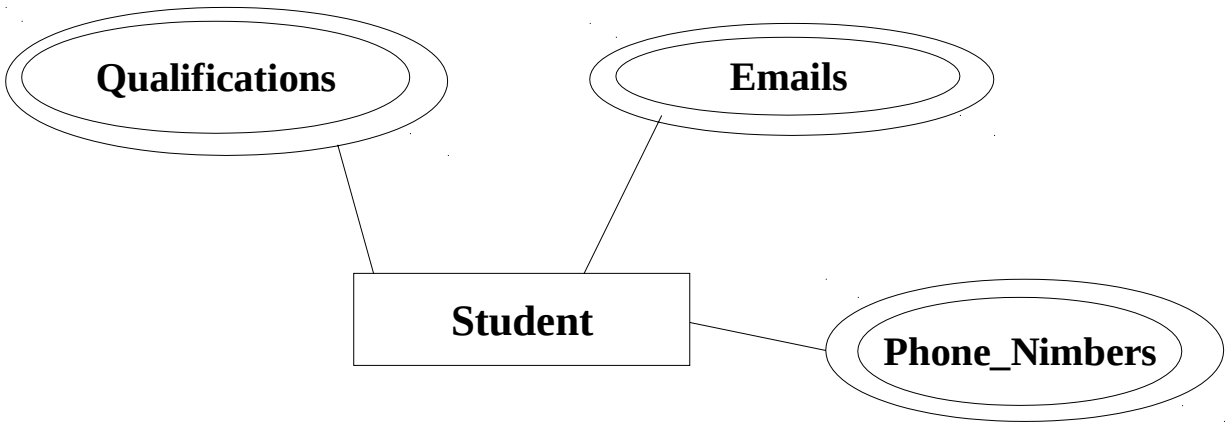
b)Single valued vs Multivalued Attributes:

Single valued attributes: The attribute which contain only one value.
Ex:blood group,pan card number etc....

Multivalued attributes: The attribute which contain more than one value.

➔ It is represend as double eclipse.

Ex:A person may have more than one phone number,email ids,qualifications etc.



C.Stored vs Derived Attributes:

Stored Attribute: The attribute which store the value and specify value to the derived Attribute.

Ex:DOB

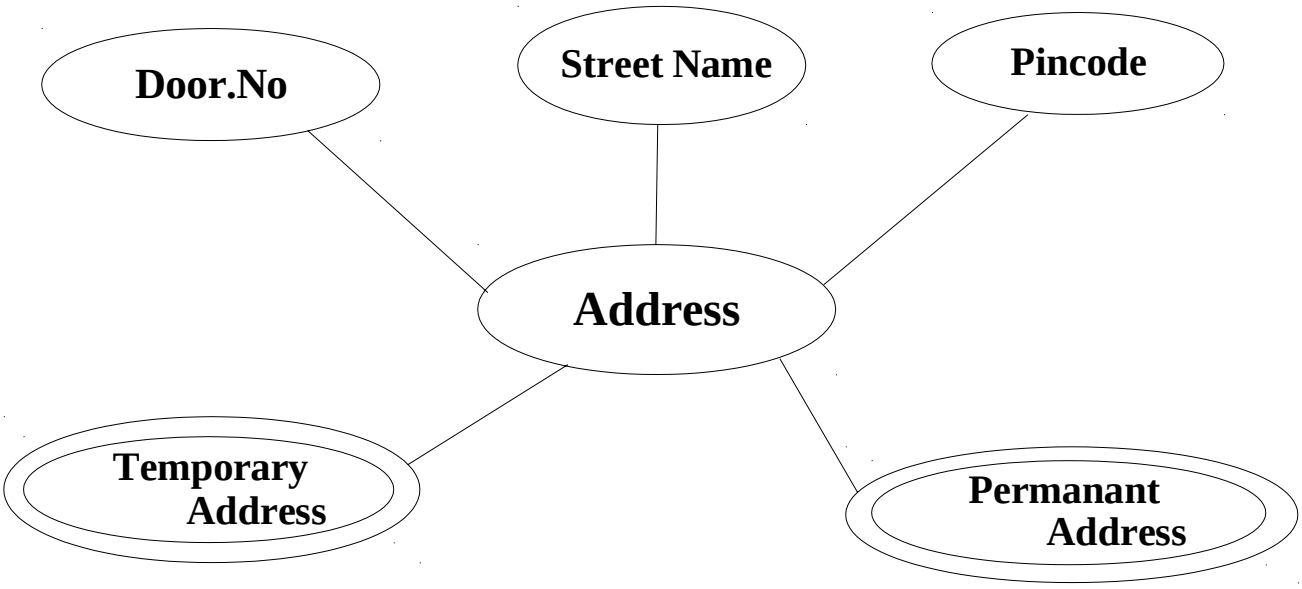
Derived Attributes: The attributes which are derived from stored Attribute.

➔ It is represented as dotted eclipse

Ex: Student age

D.Complex Attribute:

➔ It is the collection of both composite and multivalued Attributes.

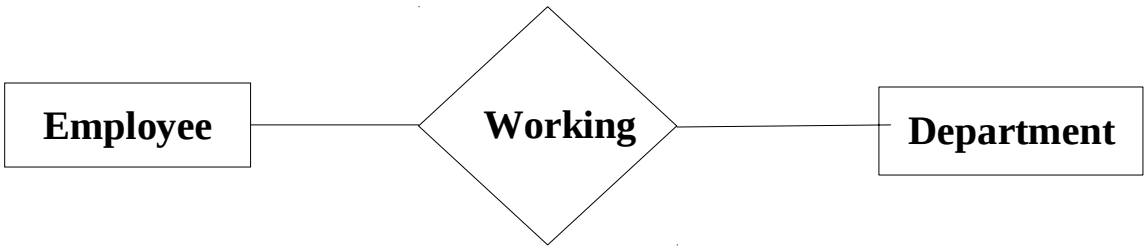


3.Relationship:

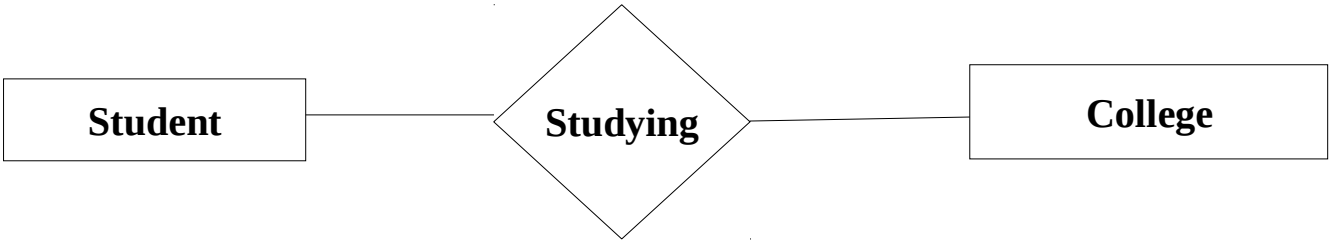
➔ Relation between entity or association among entities is called as “Relationship”.

➔ It is represented as diamond in ER model.

Ex1: Employee is working in a department



Ex-2: student are studying in college.



In above examples working,studying is called as "R**elationship**".

Relationship set:The set of relationship of similar type is called "Relationship set".

Degree of relationship: The no.of entities participate in a relationship.

Types of Relationship:

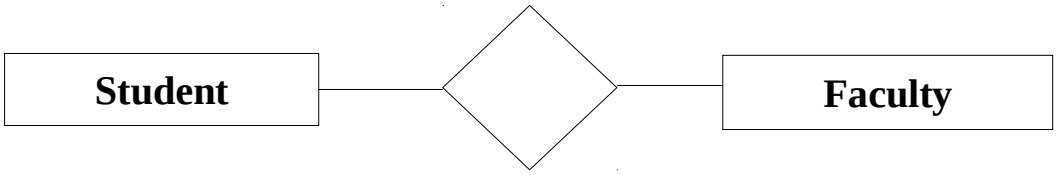
There are 3 types of Relationships in ER model:

a)Binary Relationship

b)Ternary Relationship
c)n-ary Relationship

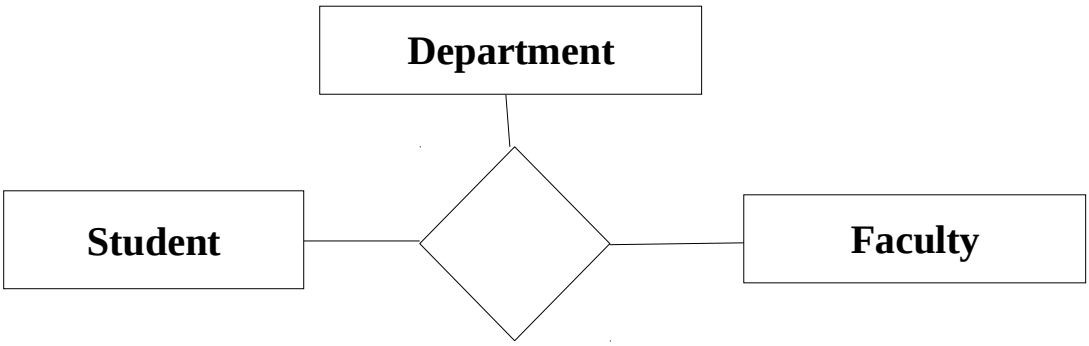
a)Binary Relationship:

➔ It means relation between two entities its degree is two.



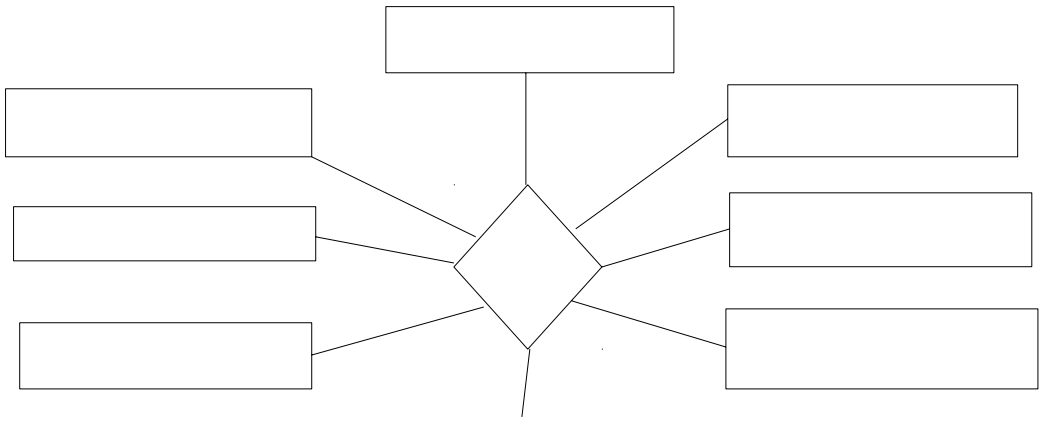
b)Ternary Relationship:

➔ It means relation between three entities and its degree is three.



c)n-ary Relationship:

It means relation between n entities and its degree is n



Binary Relationship is furher divided into

1.One to One mapping(1-1)
2.One to Many mapping(1-N)
3.Many to One mapping(N-1)
4.Many to Many mapping(N-N)

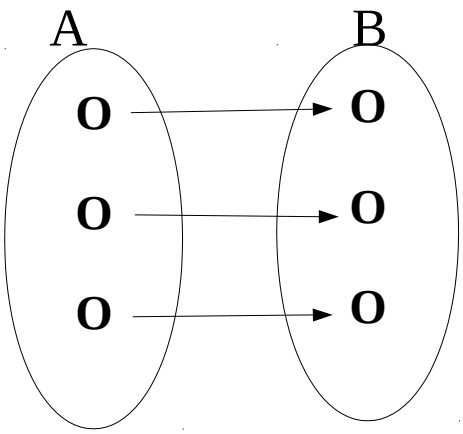
Cardinality:

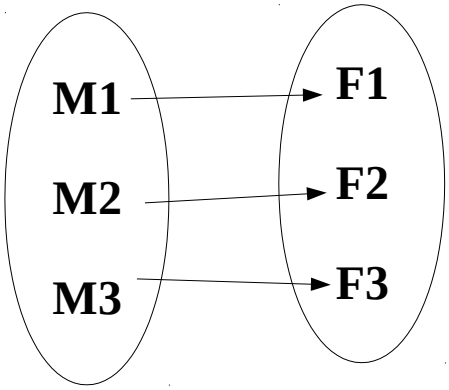
The no.of times an entity of entity set participated in a relationship is known as “Cardinality”

1.One to One mapping(1-1):

➔ One entity from entity set A that can be relation with atmost 1 entity in entity set B and viceversa.

➔ Let us consider example of a male married to female and female married to male.

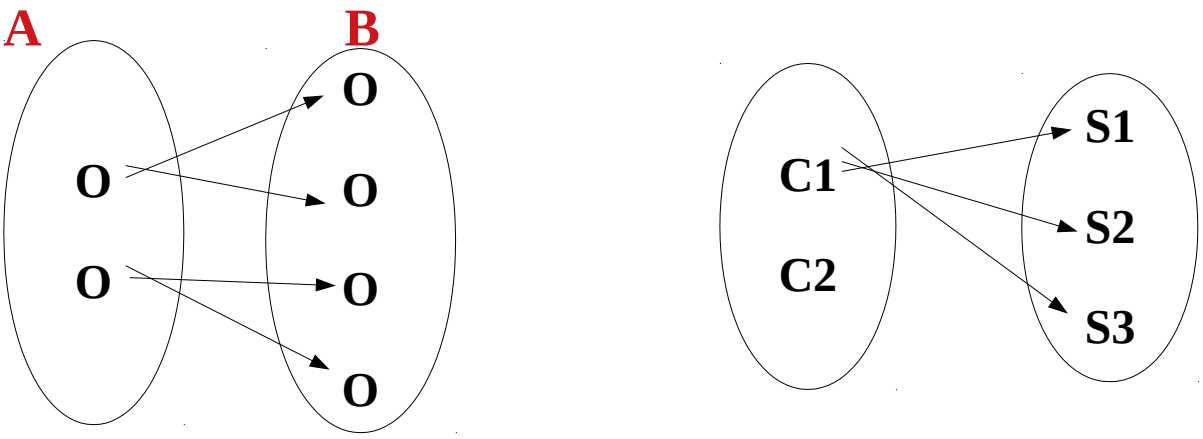




2.One to Many mapping(1-N):

➔ One entity from entity set A that can be relation with more than one entity from entity set B can be relation with atmost one entity from entity set A.

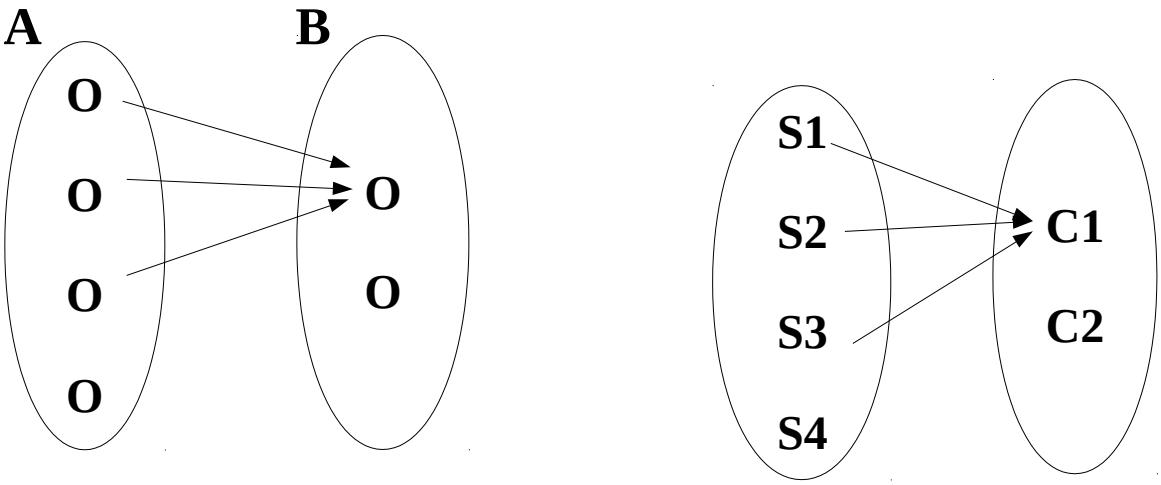
➔ For example one course taken by many students however many students taken one course.



3.Many to One Mapping(N-1):

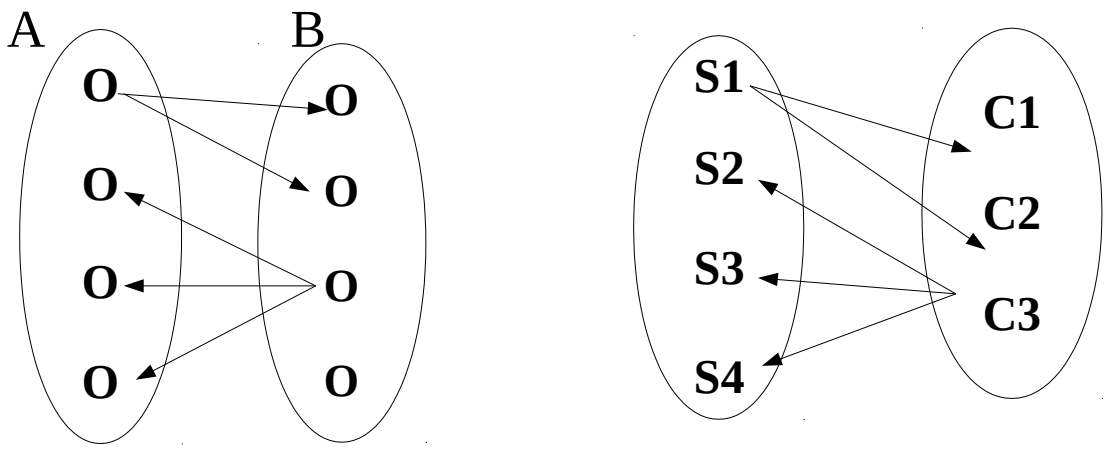
➔ More than one entity from entity set A that can be relation with atmost one entity in entity set B and vice versa.

➔ For example many students taken one course.However one course taken many students.



4.Many to Many Mapping(N-N):

- ➔ One entity from entity set A that can be relation with more than one entity from entity set B and one entity from entity set B that can be relation with more than one entity in entity set A.
- ➔ Ex:One couse taken by many students and one student taken many courses.



Constraints in ER model:

Integrity Constraints:

- ➔ These are contains set of rules and used to maintain quality of information.
- ➔ These are ensure that you can insert,update and other process have to performed in such a way that data integrity is not effected.

➔ It is used to guard against accidental failure of database.

Types of Integrity Constraints:

1.Structural Constraints
2.Key Constraints
3.Domain Constraints
4.Entity integrity Constraints
5.Referential integrity Constraints

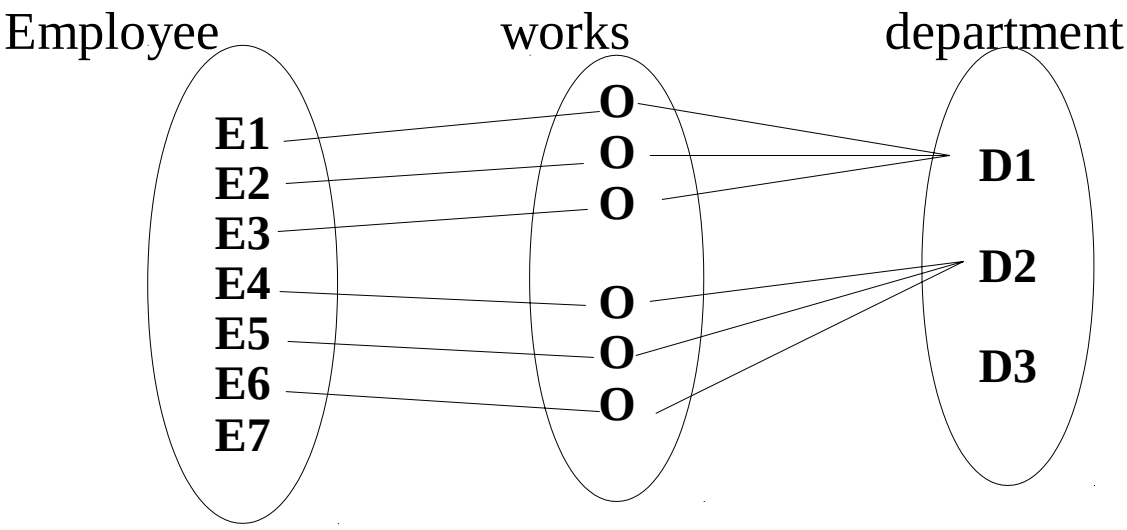
1.Structural Constraints:

These are furher divided into two types.

a)Cardinality ratio
b)Participation

a)Cardinality ratio: The maximum no.of times an entity participated in a relationship.

Statement:Every employee works for exactly one department and a department can have many employee,newly formed department do not have any employee.



In above example cardinality of employee e1 is 1,the cardinality of d1 is 2

b)Participation:

- ➔ The minimum no.of times an entity participated in a relationship.
- ➔ Some times it is also caleed as minimum cardinality.
- ➔ It is further divided into two types
  a)Total participation
  b)Partial participation

a)Total Participation: If all entities are participated in a relationship.

- ➔ It is represented as “Double lines” in ER diagram.

b)Partial participation:

- ➔ If some of the entities are participated in a relationship.
- ➔ It is represented as single line in ER diagram.

Note point: Minimum cardinality is 1 then total participation.

- ➔ Maximum cardinality is 1 , if an entity participated only one relationship.
- ➔ Maximum cardinality is one,if an entity participated in a relationship n times.

2.Key Constraints:

- ➔ These are uniquely identified in a table.

➔ Key attributes or set of attributes that are uniquely identifies an entity within entity set.

They are two types

1.Primary key

2.Foreign key

a.Primary key:

➔ primary key contains unique values and never contains new values.

➔ It is unique column in a table.

➔ A table can have only one primary key which consists of one or more columns.

Ex:

```
create table student(sid int,name varchar(20),age
int,marks int,primary key(sid));
```

student table

sid	name	age	marks
501	Raj	40	99
502	Raju	39	98
503	Ram	41	97
504	Ramu	18	92

b.Foreign key:

➔ It means it likes two different tables together and column in one table that can be pointing to the primary key in another table.

➔ They act as cross reference between tables.

Course table

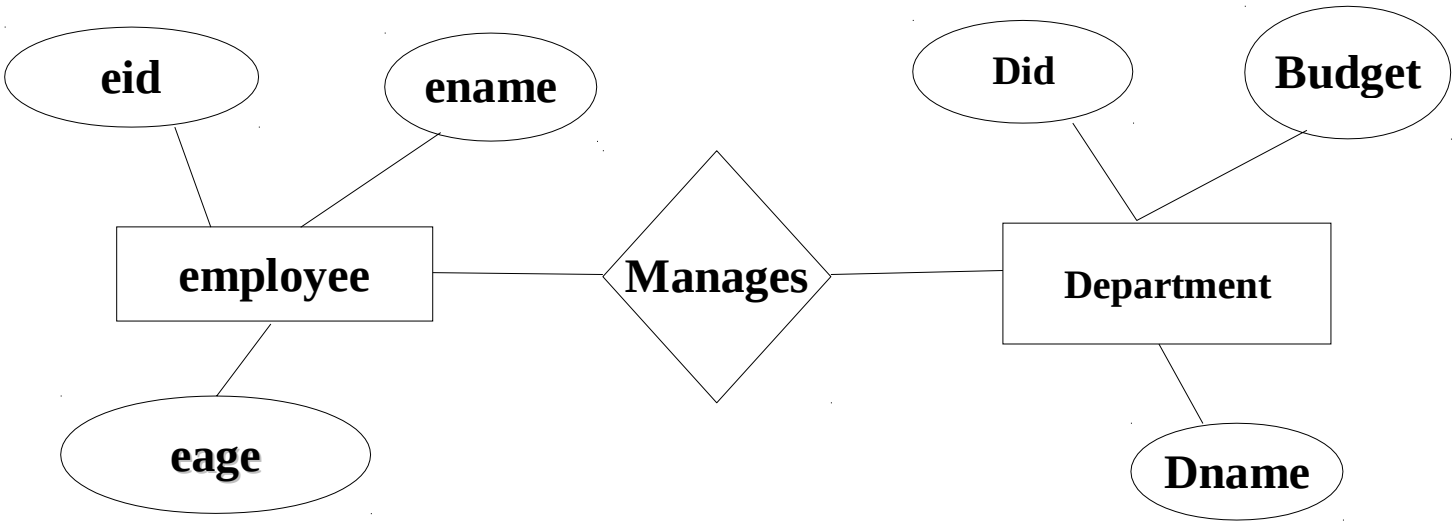
cid	sid	cname

1	501	x
2	502	y
3	503	z

- ➔ In above table courses and student table.
- ➔ In student table sid column is primary key in course table sid column is foreign key.
- ➔ A foreign key is formed with two or more tables.

Ex: create table course(cid int,sid int,cname varchar(20),primary key(sid,cid),foreign key(sid) references student(sid));
desc course;

Diagram for Key constraiants:



3.Domain Constraints:

- ➔ It means that valid set of values of an attribute.
- ➔ You need to define
  - ➢ data type.
  - ➢ length or size
  - ➢ null value is allowed
  - ➢ default value.
  - ➢ It is the value used is unique.
  - ➢ Range value
- ➔ Domain Constraints=data type+constraints(not null/unique/primary key/foreign key/default).
- ➔ Each table has set of columns and each column has allows same type of data based on its data type[int,char].
- ➔ The column doesnot accept values of any other data type.
- ➔ Every attribute is bound to specific range of values for example,age can not less than zero and a telephone num cannot contain a digit outside 0 to 9.
- ➔ The data type related with domains include character,integer,date,time,currency etc.

Ex:

sid	name	age
1	a	20
2	b	21
3	c	a

‘a’ is not allowed because it is an integer attribute.

4.Entity Integrity Constraints:

- ➔ It defines primary key value can not be NULL.
- ➔ This is because primary key value is used to identify individual rows in a table and if primary key has a null value then we can't identify those rows.
- ➔ A table can contain a NULL value other than primary key field column.

Ex: sid is the primary key

sid	name	age
1	a	20
2	b	21
-	c	a

- ➔ It is not allowed because primary key can not contain NULL value.

5.Referential Integrity Constraints:

- ➔ It can be specified between two tables.
- ➔ It maintain consistency and accuracy between tables.

Rules:

- ➢ **(i)** You can not delete record from primary table if matching record found in secondary table.
- ➢ **(ii)** You can not change a primary key in primary table if that related to the foreign key of the secondary table.
- ➢ **(iii)** Secondary table must be NULL on available in primary table.

student table
(primary table)

sid	name	age
501	a	20
502	b	21
503	c	20
504	d	22

Course table
(Secondary table)

sid	cid	cname
501	1	abc
502	2	xyz
503	3	----
504	4	Pqr

Reduction/Conversion of ER diagrams to tables/Relation:

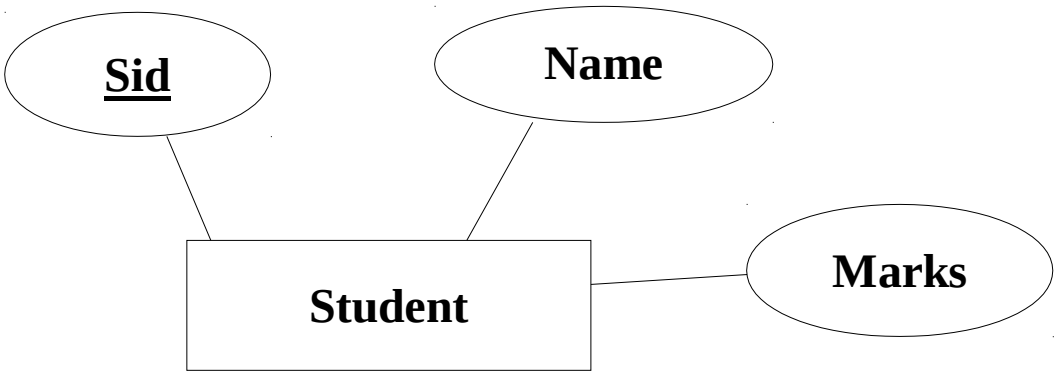
- ER diagram is converted into tables in relational model because relational models can be easily implemented by RDBMS like mysql,oracle etc.
- Following are used for converting ER diagrams into tables.

RULE 1: For strong entity set wuth only simple attributes.

- A strong entity with only simple attributes will require only one table in relational model.
- Attributes of the table will be attributes of an entity set.

➢ The primary key of the table will be the key Attribute of the entity.

Ex:



Schema: Student(sid,name,marks)

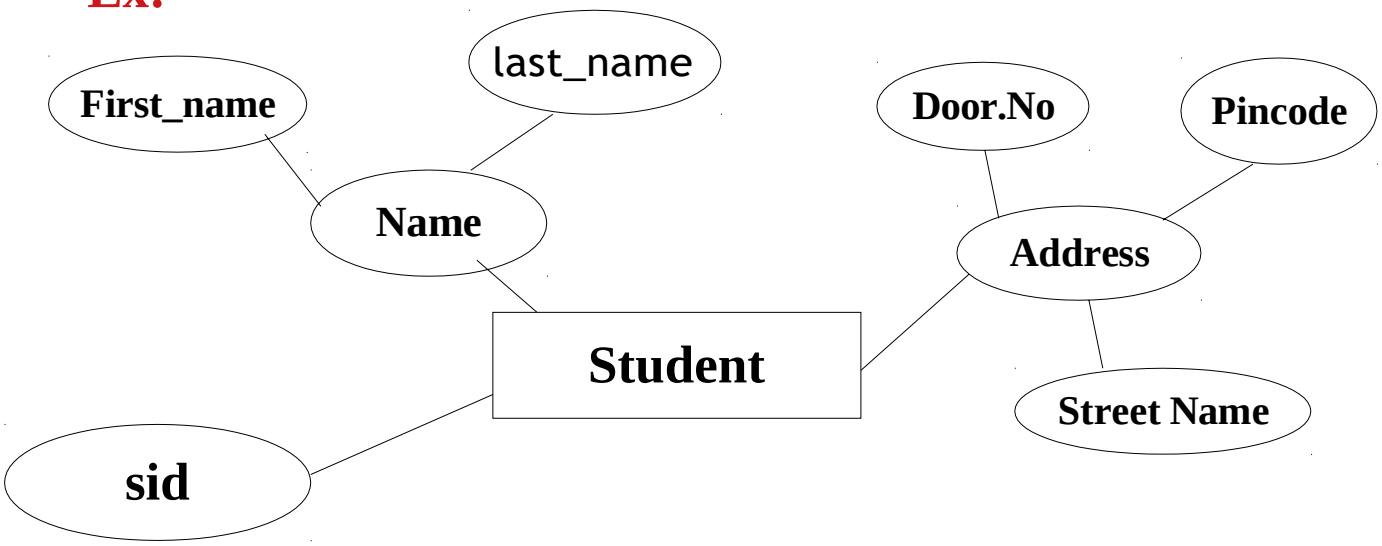
Roll No	Name	Marks

RULE 2: For strong entity set with composite attributes

➔ A strong entity set with any no.of composite attributes will require only one table in relational model.

➔ While conversion,simple attributes of the composite attributes are taken into account and not the composite attribute itself.

Ex:



Sid	First_name	Last_name	Door.No	Streetname	pincode

Schema:

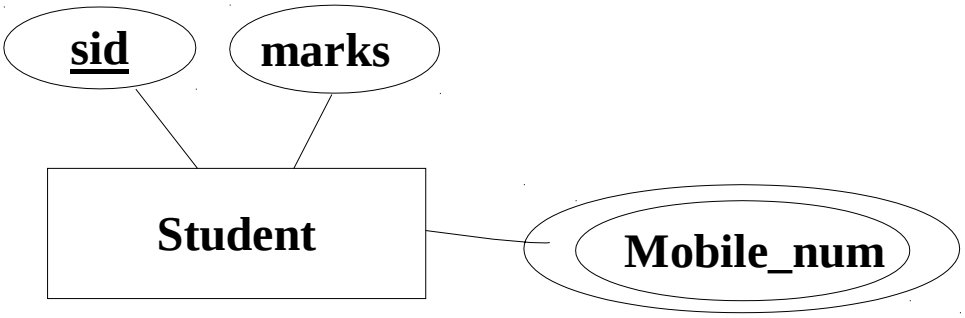
student(sid,firstname,lastname,dno,streetname,pincode)

RULE 3:For strong entity set with multi valued attributes .

- ➔ A strong entity set with any no.of multivalued attributes require two tables in relational model.
- ➔ a)one table will contain all the simple attribute with the primary key.

➔ b)other table will contain the primary key and all multivalued attributes.

Ex:



First table:

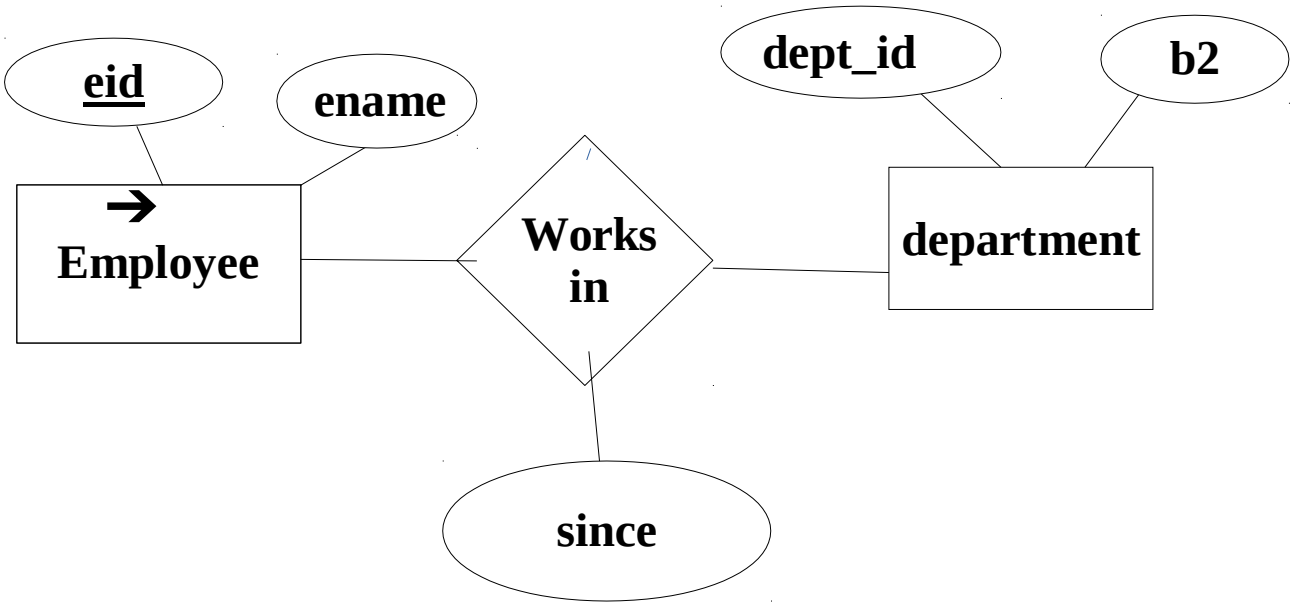
Sid	marks

Second Table:

sid	marks	Mobile num

RULE 4: Translating relationship into tables.

- A relationship set will require one table in relational model
- Attribute of the table are
  a)primary key attributes of the participating entity sets.
  b)its own descriptive attributes if any.



Schema: worksin(eid,deptid,since)

eid	deptid	since

Note: if we consider the overall ER diagram,three table will be required in relational model.

- One table for entity set “employee”.

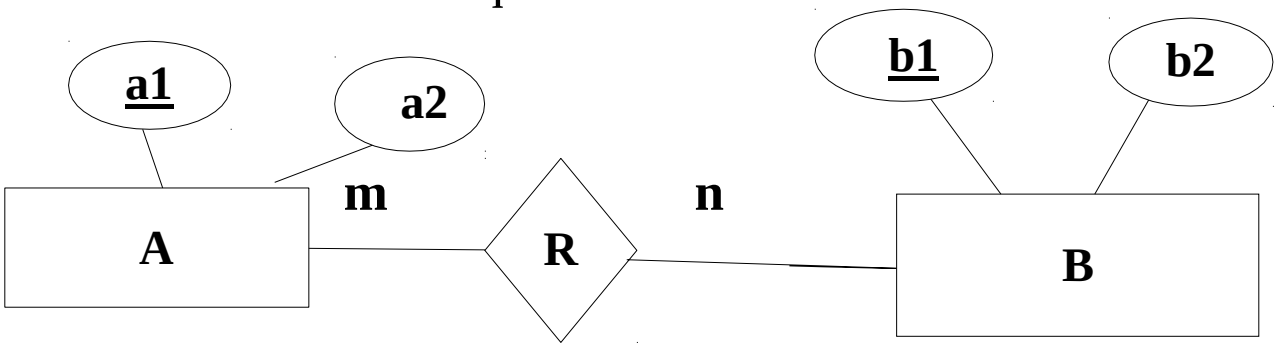
➢ one table for entity set “department”.
➢ one table for relationship set “works in”.

RULE 5:For binary relationship with cardinality ratios

➔ 4- case 1:binary relationship with cardinality ratio 1:1
➔ 2-case 2:binary relationship with cardinality ratio 1:m
➔ 3-case 3:binary relationship with cardinality ratio m:1
➔ 1-case 4:binary relationship with cardinality ratio m:n

Case 01:Binary relationship with cardinality ratio m:n

Here 3 tables are required...



i)A(**a1**,a2)

a1	**a2**

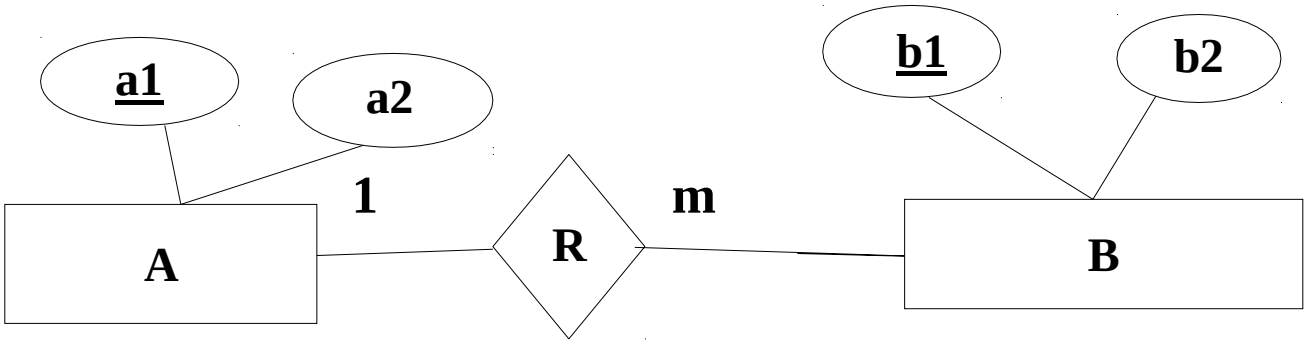
ii)R(**a1**,**b1**)

a1	**b1**

iii)B(**b1**,b2)

b1	**b2**

CASE 02:Binary relationship with cardinality ratio 1:m



here two tables required

1.A(**a1**,a2) FK

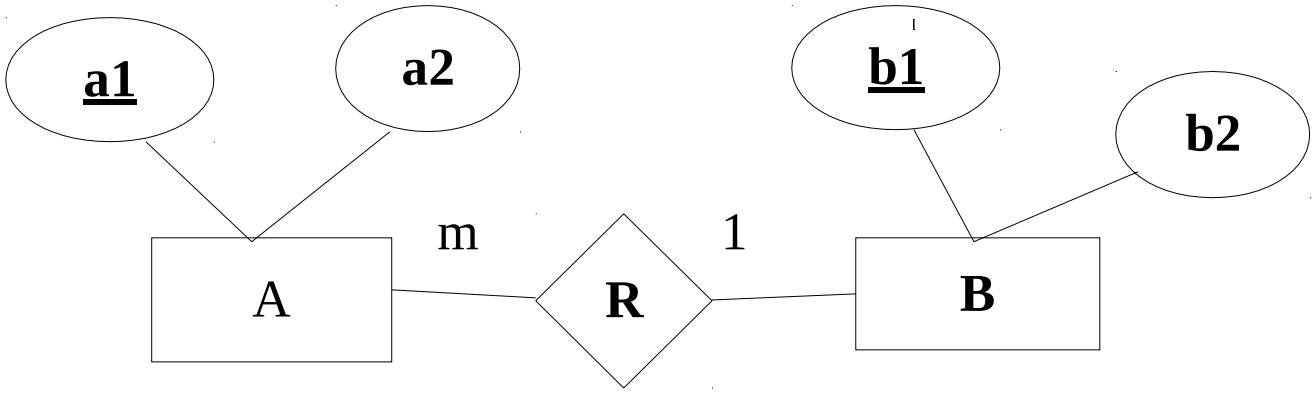
2.BR(**a1**,**b1**,b2)

Here “b1” is foreign key.

Note: here combine table will be drawn for entity set B and relationset R.

Ex: R(a1,b1) B(b1,b2) BR(a1,b1,b2).

CASE 03:Binary relationship with cardinality ratio m:1



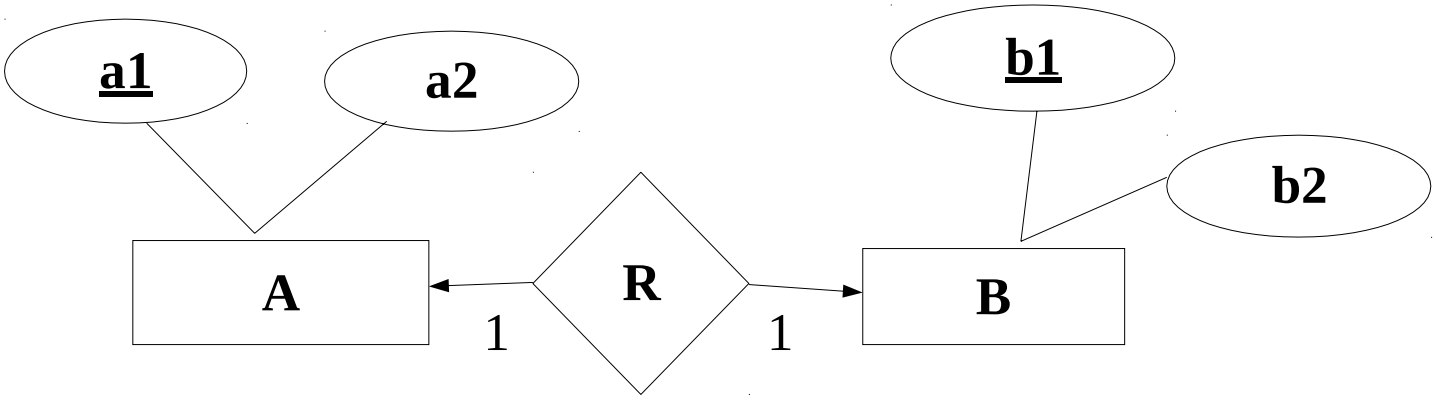
here two tables will be required

i)AR(a1,a2,b1) here b1 is foreign key

ii)B(b1,b2)

Note:here combined table will be drawn from entity set A and relationship set R.

CASE 04:Binary relationship with cardinality ratio 1:1



here two tables will be required either combine R with A or B.

way 01: 1.AR(a1,a2,b1)
2.B(b1,b2)
here b1 is the foreign key

a1	a2	b1

b1	b2

Way 02: 1.A(a1,a2)
2.BR(a1,b1,b2)
here "a1" is the foreign key.

a1	a2

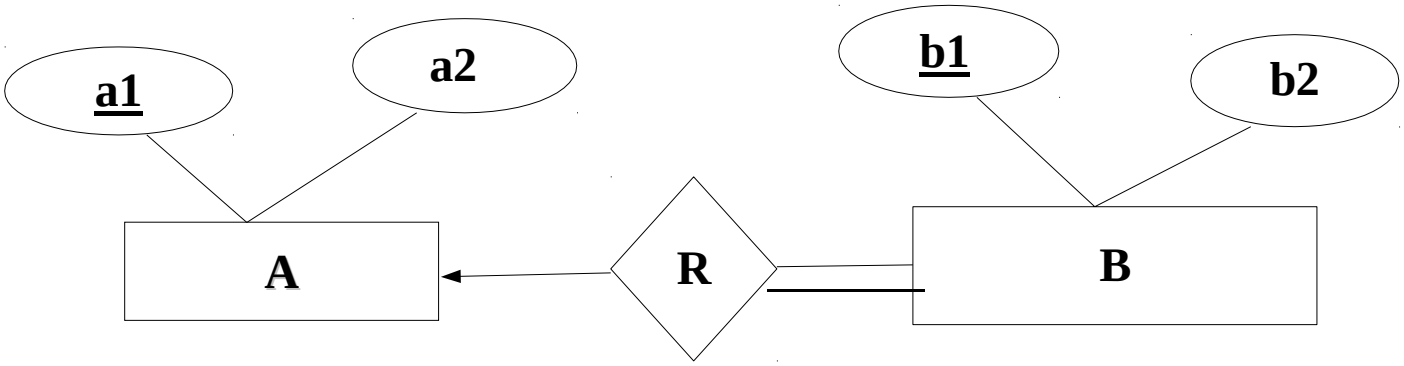
a1	b1	b2

Points to remember:

- While determining the minimum no.of tables required for binary relationship with given cardinality ratio,following rules to remember.
- a)for binary relationship with cardinality ratio m:m,separate and individual table will be drawn for each set and relationship.
- b) for binary relationship with cardinality ratio either 1:m or m:1 always remember many side will consume the relationship i.e, combined table will be set for many side entity set and relation set.
- c) for binary relationship with cardinality ratio 1:1,two tables will be required .you can combine the relationship set with any one of the entity sets.

RULE 06: for binary relationship with cardinality and participation constraints.

Case 01: for binary relationship with cardinality constraints and total participation from one side.



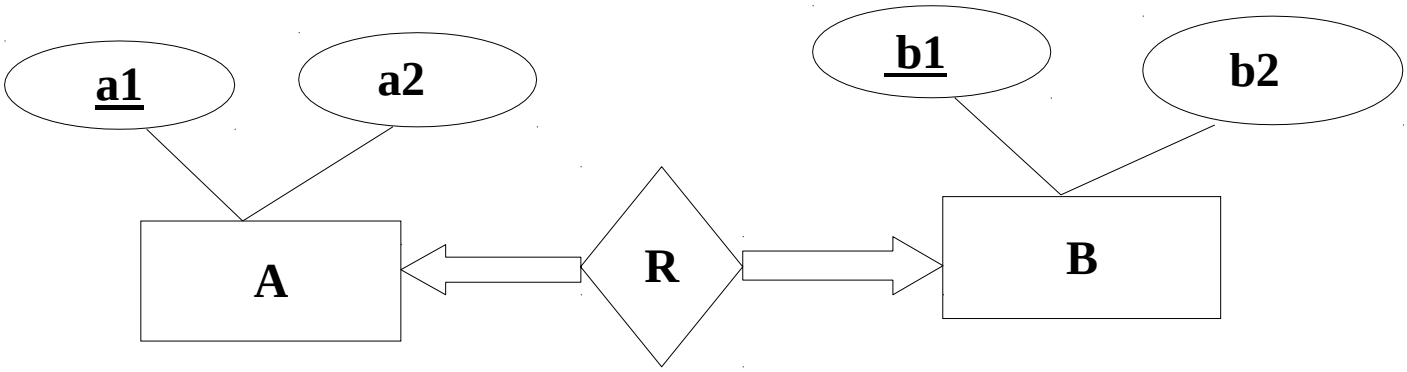
Because cardinality ratio 1:n,so will combine the entityset_B and relation set R.then,two tables will be required.

1.A(a1,a2)
2.BR(a1,b1,b2)

Here a1 is foreign key.

Case 02: for binary relationship with cardinality constraints and total participation from both sides.

- If there is a key constraints from both the sides of an entity set with total participation,then relationship is represented using only one table.

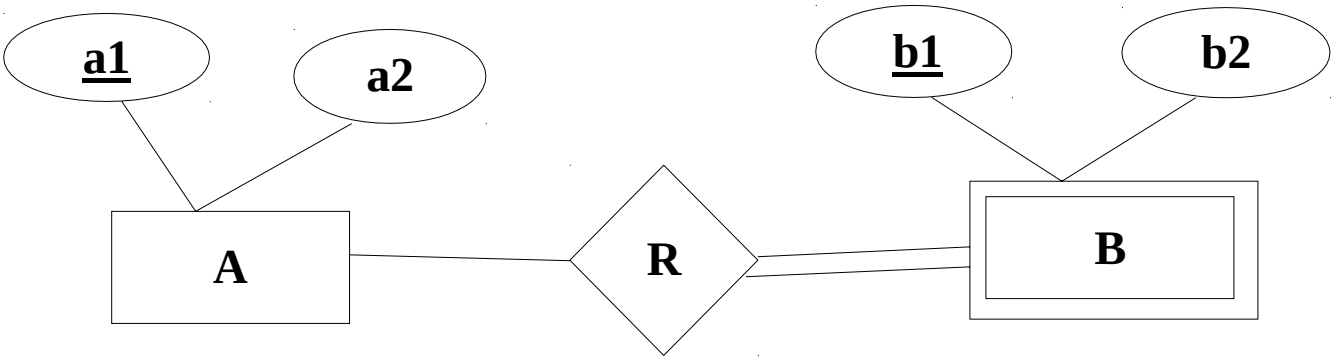


here only one table required.

ARB(a1,a2,b1,b2)

RULE 07: For binary relationship with weak entity.

weak entity set always appears in association with identifying relationship with total participation constrains.



here two tables will be required

1.A(a1,a2)

2.BR(a1,b1,b2)