1. Introduction to DBMS & E-R Model

* **Data** :- Any form of Raw-material which can be stored & is known as data.

(or)

Any fact that can be recorded.

Ex: text, numbers, Audio, video

* **Database** :- Collection of related data

* **Data** :- It is any type that can be recorded (or) that It is any type of raw material that can be processed on any computer.

Ex: Text (names), numbers (phno:), Images (pictures), videos, Speech Etc...

* **Database** : It is a collection of related data & describe the activities of one or more organisations.

Ex: University Database contains Entities Such as Student, courses, faculty and classmates / classrooms.

* The relation between Entities Such as student Enroll-ment in courses, faculty teaching courses and used classrooms for courses.

* There are different kinds of databases :-

i) **Traditional database** :- It Contains names & numbers

ii) **multimedia database** :- It contains videos like speech, Songs, Movies Etc...

Ex:- Youtube

iii) Geographic Information System (GIS) :- It contains images of satellite system (or) other things

Ex: Nasa NASA having GIS

iv) Relational database :- It contains tables with rows & columns in 2D form.

Data warehousing :- It is a kind of database and store huge amount of data.

Ex: Any organisation with past 100 yrs data.

DBMS :-

It is a software (or) sets of programs that allow users to define, create and maintain a database and provide control access of data.

Ex: mySQL, DB2, oracle log, lig etc.

DBS :- (Data base system)

It is a collection of database and database management system ie. DBS = DB + DBMS (Notebook + pen) without computer

Where DBMS being used (or) Applications of DBMS

1. Airlines :- Reservations, schedules Etc.
2. Telecom :- calls made, customer details, network usage
3. Universities :- Registrations, grades, results etc.
4. Banking :- All transactions
5. Sales :- products, purchases, customers etc
6. Finance department :- Its like storing sales, getting information and finance statements.
7. Online shopping :- It has become a Trend of modern days. It contains purchase information, payments, Invoice.

8. Social media Sites :- It contains accounts like Fb, whatsapp, Twitter etc.

9. Library Management System :- It contains book issue dates, name of the book, author & availability of book.

10. Railway Reservation System :- It contains records of ticket booking, Train departure & Railway status.

11. Human Resource management :- It contains contains Employees salary, Tax.

12. Manufacturing :-

It contains product details like quantity, bill purchased, Expiry dates etc.

13. Military system :-

It contains records of millions of soldiers that keep the files in secure and safe mode.

* File system vs Database system

File : It is a sequence of records stored in Binary format

Filesystem : 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 :-

i) It is a group of files storing the data of any organisation

ii) Each file is called as flat file (There is no particular structure & format).

iii) Files are designed by using programming languages such as C, C++

Limitations (or) disadvantages of file system

i) Seperated and isolated data

ii) Duplication of data

1) It takes additional storage space
2) There is a loss of data integrity
3) It is more cost and time

iii) Data Dependency :- One file is dependent to other files and these are contains particular physical format that are developed by the programmers.

iv) Data redundancy :- It means the same amount of data can be stored in a different places that is data Re

v) Data Security :- The security of data is low in the file system that can be easily accessible.

vi) Transaction problems :- File systems are not support the transaction / ACID properties

A - Atomicity
C - Consistancy
I - Isolation
D → Durability

vii) Concurrent problems :- When multiple users access the same amount of data at same interval of time is called as concurrency of a 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.

r ✓
up ✗

Advantages of DataBase System :- To work on the

To Overcome the problems of file system we use a DBS :-

i) Data independency :- The DBMS provides an abstract view of data and hide such details.

ii) Reduce data redundancy :-

1) In file system the data can be stored in a differen places and there is a chances of duplication of data

2) And also there is a waste of space

* But in database system all the files are stored in a single database. The whole data is stored in a only once so there is no chance of duplication of data.

iii) Sharing of data :- In database system data can be saved / shared between authorised users of database

2) All the users have own right to access the data.

3) The database Administrator has complete access of database and he can assign users to access the data.

iv) Data consistancy :- It means control / reduce the data redundancy which is mean that system with high data consistancy.

v) Data integrity :- It means that data in a database is always accurate (or) correct that can be decreased the duplication of data also data inconsistancy.

vi) Data security :- It means protecting your data from unauthorised users.

2) Data in a database be kept secure and safe

3) Only authorised users should be grant to access the database

4) providing backup and Recovery :- The backup and recovery of database system from software (or) hardware failures with facilities like recover the data and packup the data.

Disadvantages of Database System :-

i) increased cost :- Database system require sophisticated hardware and software (modern) with more cost

ii) Complexity :- Developers, Designers, Database Administrators and End users of Database must have complete skills if they want to use it properly otherwise there is a chances of loss of data (or) database failure.

iii) Technical staff Requirement :- Any Organisation have many Employees, it is not Easy for work on DBMS and they Wellknown the Databasesystem.

iv) Database failure :- All the files are stored in a single database so there is a chances of failure become more

* Any accidental failure of component may cause loss of valueble data.

v) Size :- DBMS become big software, lots of space & memory to run the big applications Efficiently and it gains bigger size.

vi) Currency maintanance :- DBMS should be updates according to the currency scenario and new threats comes daily so DBMS require update itself daily.

vii) Performance :- The total data can be stored in a same database and there is a chances of loss in performance of a system.

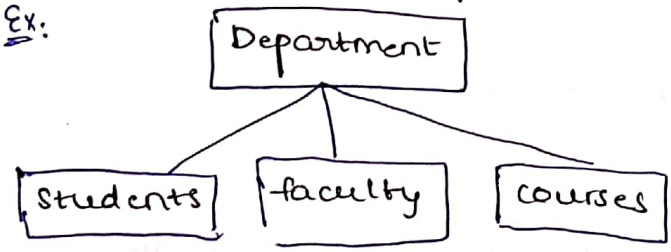
Database models : It is logical design and structure of Database and define how database is stored, accessed and updated in a DBMS.

There are Different Database models in a DBMS :-

i) Hierachial model
ii) Network model
iii) Entity - Relationship model
iv) Relational model

i) Hierachial model : In this model data in 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:

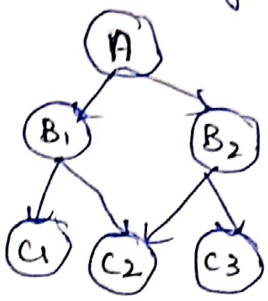


ii) Network model :

* It is the Extension of Hierachial model
* It is drop 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:

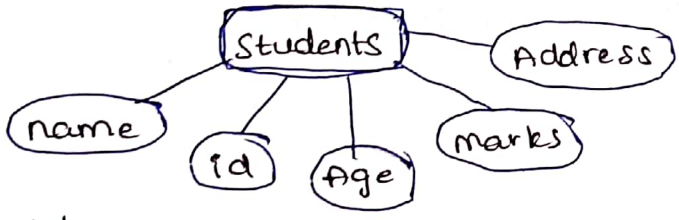


[M-N]

(iii) E-R Model:- The pictorial form of data is known as E-R Model

* In this model data in a database designed in a good way
* For Example college database contains students as Entity and attributes are name, idno, Age, marks, Address etc-

* Ex:



(iv) Relational Model: **

* In this model the data organised in two dimensional table (2D table)
* It is a collection of table with a data and Relationship
* This Model Introduced By EF codd in 1970
* It is most ~~quite~~ widely used database model
* The basic structure of Relational Model is tables
* The tables are also known as Relational
* Ex: Student

S name	idno:	Age	marks	Section
Bhuvana	0162084	18	95	CSE-1
Srija	0161587	18	92	CSE-1

10/7/19

Components of database system :-

There are 4 type components of DBS

1. Data :-

* Data stored in a DB include numerical and non numerical data, audio, video etc.

2. Hardware :-

* It includes various storage devices, input and output devices.

3. Software:-

* It includes OS, network software and the application program.

4. Users :-

* There are different types of database users in DBS like Database administrator, Database designers, Endusers, applicatio programmers etc.

Database users:

There are 2 types of database users

1. Actors on the Screen
2. Workers behind the Scene.

i. Actors on the Screen: Those who are use and control the database content and those who design, develop and maintain DB applications are called as actors on the Scene.

ii) Database Administrator :- A person who performs all the activities related to maintaining a database.

* Responsibilities include designing, implementing and maintai -ning a database.

Responsibilities and functions of DBA :-

i) Installing and updating SQL Server

→ The DBA responsible for installing SQL Server or upgrading SQL Server with Service packs.

ii) Monitoring SQL Server

→ The DBA also responsible for SQL Server is running with particular performance or not and monitor Each and Every time.

iii) Using storage properly

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

iv) performing backup and recovery duties

→ The DBA responsible for backup and recovery of software or hardware failures

(v) Managing database users and security

→ The DBA is also responsible for assigning DB users and determining the proper Security level for Each user.

→ It is also responsible for work coith network administra -tors

vi) Working with developers

→ The DBA responsible for a person well known in SQL Server.

vii) Transferring of data

→ The DBA also responsible for importing and Exporting the data from the SQL Server.

viii) providing 24 hours access

→ The DB Server must stay up and always protected and provide information to database users.

(ix) Datawore housing

→ It provides huge amount of storage like past 100 years data.

(b) Database designers :- They are responsible for identifying the how much of data can be stored and way to Organise it.

(c) End users: They are responsible for access the database for querying, updating, and report generation.

→ There are different types of End users.

* Casual End users :- They use DB ocassionally or regularly and need different information Each time by using SQL.

Ex: middle (or) high level managers.

* Naive or parametric End users :- They are biggest group of users and they frequently update the DB using canned transactions.

Ex: (i) Bank tellers check account balance, withdraw, deposit

(ii) Reservation clerks for airlines or hotels etc check availability of seats or rooms and make reservation.

* Sophisticated End users:

Engineers, Scientists, business analysts all are come under Sophisticated End users.

* Standalone end users: They are maintaining personal DB with particular Software packages.

Ex: financial

(d) System analysts, application programmers and software Engineers :-

System analysts : They are determine the need of End users particularly naive End users.

Application programmers :- They always implement, test, document and maintain a program.

2. Workers behind the Scene : Those who design and develop the DBMS Software and related tools and computer system operators.

(i) Tool developers : Design and develop the software tools related to DBS design, performance monitoring etc

(ii) Operators : They are responsible for day to day operations of computer system.

11/7/19

Database languages

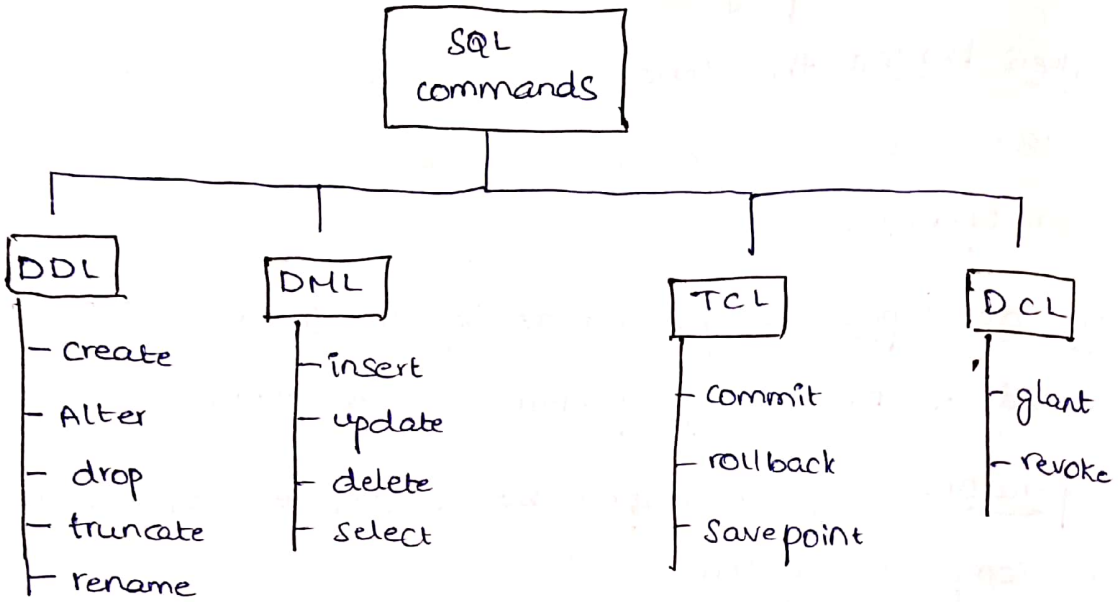
* These are also called as SQL
* SQL stands for structured query language
* It is not a database, it is a programming language
* It is standard language for relational database

What can SQL can do ?

* SQL can create new database
* SQL can create new table in a database
* SQL can insert records in a database
* SQL can delete records in a dB
* SQL can update records in a dB
* SQl can retrieve data from a database
* SQl can Executes Queries against the database
*

There are 4 types of database languages:-

1. DDL - Data definition language
2. DML - Data manipulation language
3. TCL - Transcation control language
4. DCL - Data control language



1) DDL (Data Definition language):- This Language is used for create and Modify the dat structure of Database objects.

1) create command: It is used to create a table (or) DB.

Syntax:

```
Create table tablename (column_name1  datatype1,
                        column-name2  datatype2,
                        column_name3  datatype3
                                :
                                :  );
```

Ex: create table student (name varchar(20),
Sid int,
Age int);

* In above Example student is the table name & it contain 3 columns names are name, Sid, Age.

2) Alter command :- It is used for Alternation of data in a DB.

- To add column to the Existant table
- To add multiple columns to the Existing table
- Drop a column
- modify a column
- Rename a column

* Add column to the Existing table :

Syntax : and Ex

```
Alter table tablename add (address varchar(20),
                           marks   int);
```

(column3 Datatype)

* Drop a column :

Syntax :

Alter table tablename drop (columnname datatype);

* Modify a column :

Syntax :

Alter table tablename modify (columnname datatype);

Ex:

Alter table tablename modify (address varchar(20));

* Rename a column :

Syntax :

[Alter table tablename Rename (data columnname datatype)]

Ex:

Alter table tablename Rename address to location

Syntax for Rename :

Alter table tablename rename old-column.name to new-column.name;

3) Truncate : This command is used for remove all the records from the table but it cannot destroy the table structure.

* When we apply truncate command on a table its primary key is initialising / initialised.

Syntax :

Truncate table tablename;

Ex :

Truncate table student;

Sid	name	Age
501	Bhuvana	18
502	Srija	18

→

sid	name	Age
501	-	-
502	-	-

4) Drop command : This command is used for remove table from the database.

* It destroys the table structure

Syntax :

Drop table tablename;

Ex :

Drop table student;

5) Rename command : This command is used for rename a table name.

Syntax :

Rename old_table_name to new_table_name;

Ex :

Rename student to student_record;

ii) DML :- Data manipulation language

→ These language used for managing or changing the data in a DB.

→ DML commands are not auto-commited

→ Insert, Select, delete, update are DML commands.

i) Insert - command

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

Syntax: insert into tablename value (data1, data2, ---);

Ex: insert into student value (501, 'viswitha', '18');

insert into student value (2083, 'Nichala', '20');

ii) Select - command

→ This command is used to retrieve data from database.

Syntax:

Select * from tablename;

Eg: Select * from student;

Sid	name	age
501	Suppo	18
502	swathi	19

iii) Delete - command:

→ This command is used to delete data from a table.

→ It also deletes particular row.

Syntax: delete from tablename where condition;

Eg: delete from student where sid = 501;

Select * from student;

(iv) update - command: This command is used to update a row of a table.

Syntax: update tablename set columnname = value

where condition;

Eg: update student set age = 19 where sid = 2083;

3. TCL (Transaction control language): These are used to control the transaction in a Database.
(or)
manage

* These are used to manage the changes made by DML statements.
* Commit, Rollback, Savepoint are Tcl commands.

i) Commit: It commits the current transaction means making changes permanently. The syntax

Syntax

```
commit ;
```

ii) Rollback: It Rollbacks the current transaction means cancelling its changes and it restore the database.

Syntax:

```
rollback to savepointname ;
```

(ii) Ex:- rollback to A;

(iii) Savepoint: It is used to store data temporarily.

Syntax:

```
Savepoint savepointname;
```

Ex:

```
savepoint A;
```

Example

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, 'Bhuvana');

> savepoint A;

> insert into student values (505, 'srija');

> savepoint B;

> insert into student values (506, 'supriya');

> savepoint c; > Select * from student;

output

sid	name
501	abc
502	ram
503	lmn
504	Bhuvana
505	srija
506	supriya

> rollback to A;

> Select * from student;

Output

sid	name
501	abc
502	ram
503	lmn
504	Bhuvana

4. DCl: 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) permissions on the database objects to the user.

Syntax:

grant privilage_name on object_name to {username (or) public (or) role_name}

[with grant option];

i) privilage_name: privilage granted to DDL (or) DML commands with user like select / Alter / update etc..

(ii) object_name: It means name of the database / table,

iii) user_name: It means name of the user who access rights is being accessed / granted.

(iv) public: It means grant access right to all other users

(v) with grant option:- It means allow a user to grant access rights to other users.

Ex: grant select on student to ram;

2. revoke command: It is used to remove access rights (or) privilages to the database objects.

Syntax:

revoke privilage_name on objectname from {username / public / rolename}

[with grant option];

Ex:

revoke select on student from ram;

Views: It is a kind of virtual table.

* It also contains rows & columns as they are in real table in the database.
* It contains Equal columns or less columns compare with real table / main table.
* We can create a view by selecting fields of one or more tables present in a database.
* A view can either have all the rows of tables or specific rows based on certain condition.

Syntax for creating a view:

Create view View_name as select column1, column2, ... from tablename where condition;

→ Here View name is name of the view, table_name is name of the table (real / main table) and condition, it means to select rows.

Student table (real table)

Sid	name	age
501	abc	20
502	def	21
503	lmn	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
lmn	22
pqr	23
xyz	24

Insert rows in a view

Syntax: insert into view_name (column 1, column 2)
values (value 1, value 2);

Ex: insert into student_view (name, age) values ('ram', 25)

Delete rows in a view:

Syntax: delete from view_name where condition;

Ex: delete from student_view where name = 'Nani'

Update row in a view:

Syntax: update view_name set columnname = value where condition;

Drop a view:

Syntax: drop view view_name;

Simple view: Taking content from one table

Complex view: Taking content from two tables

Schema and Instance

Schema: It is a skeleton structure of database (or) logical view of database. It defines how data is Organised & how relation among the Entities.

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

* Database schema defines variable declaration of a table but database instance gives values to the variables of a table.

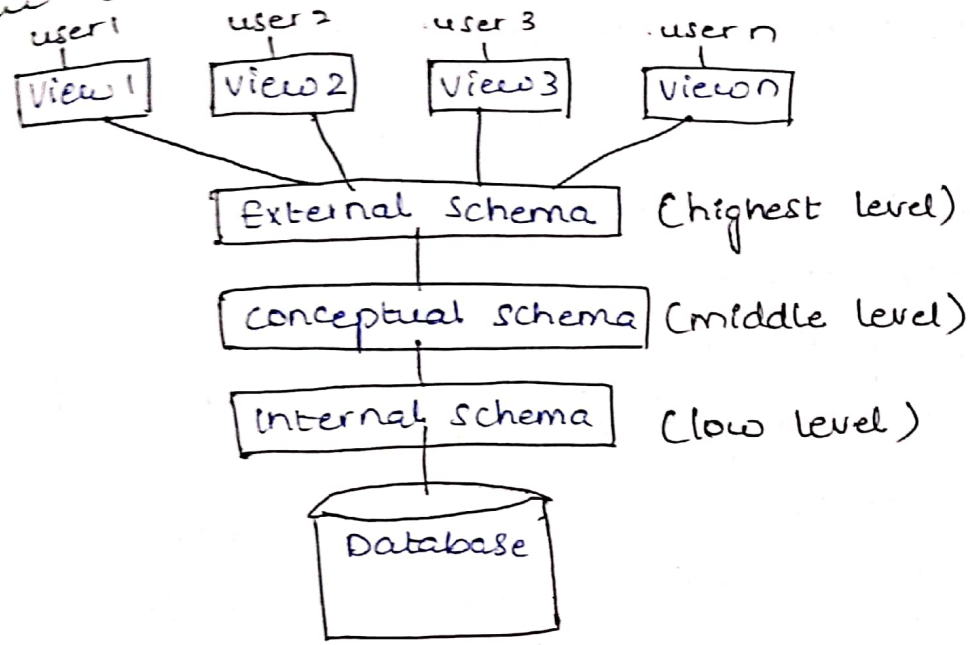
Data independence: The capacity of change the schema and one level without change the schema at the highest level.

* These are two types:

1. physical data independency: It is a capacity of change the schema at internal level without changes in the schema at conceptual level.

2. Logical data independence: It is a capacity of change the schema at conceptual / logical 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

Internal level: It is also called as physical level or low level.

* It describes physical representation of data and also describes additional storage details.
* How the data stored in a database.
* Record placement

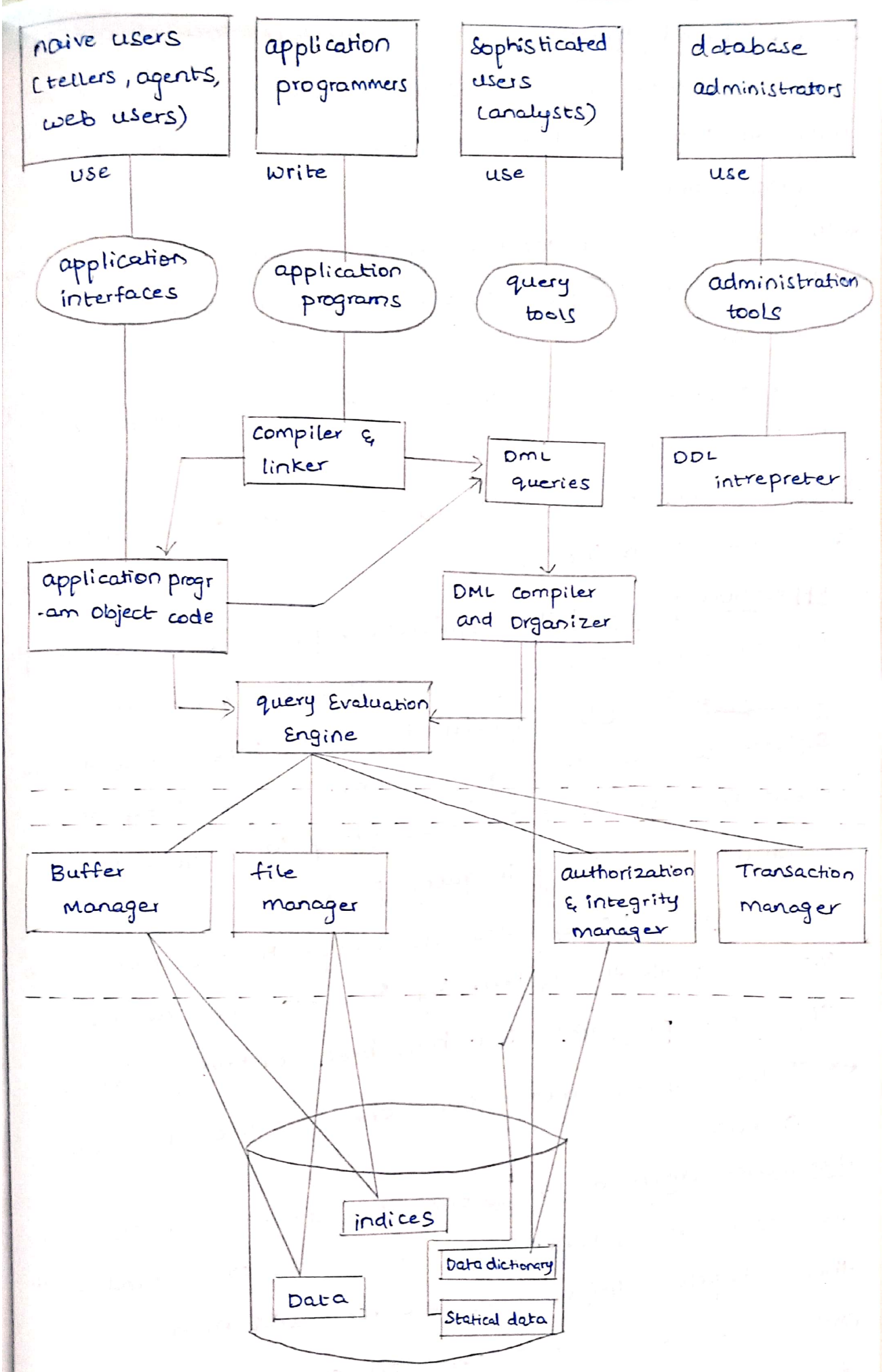
Conceptual level: It is also called as logical level or middle level.

* It describes logical representation of data that can be stored in a database.
* In relational database management system (RDBMS) the conceptual level describe the all the relation stored in a database.

* what data is stored in a database and relationship among the data.
* This level represent entities, attributes, relations etc.
* For Example university database contains Entities such as student, course, Faculty and classrooms.

3) External level: It is also called as highest level.

* This level consists of number of different External views of the database.
* It describe particular group of users and provide powerful and flexible security to the users.
* It permits user to access the data in a database in a particular way and same data can be seen by different authorized users. at the same amount of time.



Structure of Dbms

Database Architecture: Database architecture focuses on design, development, implementation and maintainance of computer programs that store and organize information for organisation.

* The architecture of database system is greatly influenced by the underlying computer system on which database is running.

i) Centralized ii) client-server iii) parallel(multi-processor) iv) Distributed.

Database users: users are differentiated by the way they expect to interact with the system.

→ Application programmers: Application programmers are computer professionals who write application programs.

⇒ Sophisticated users: Sophisticated users interact with the system without writing programs. Instead, they form their requests in a database query language. They submit each query to query processor.

→ Naive users: Naive users are unsophisticated users who interact with the system by invoking one of the application programs that have been written previously.

Ex: Teller transfers money from one account to another account.

Database administrator: Co-ordinates all the activities of the database system. DBA has a good understanding of the enterprises information resources and needs.

Duties: Schema definition: The DBA creates the original database schema by executing a set of data definition statements in DDL.

* storage structure and access method definition.
* schema and physical organization modification.
* Granting user authority to access the database
* Monitoring performance and responding to changes in requirements.

Query processor: The query processor will accept query from user and solves it by accessing the database.

DDL interpreter: This will interprets DDL statements and fetch the definitions in the data dictionary.

DML compiler: This will translates DML statements in a query language into low level instructions that the query Evaluation Engine understands.

Query Evaluation Engine: This Engine will Execute low-level instructions generated by DML compiler on DBMS.

Storage Manager: A storage manager is a program module which acts like interface between the data stored in a database and the application programs and queries submitted to the system.

* Storage manager is responsible for storing, retrieving and updating data in the database.

Introduction to E-R model:

E-R model: The pictorial or graphical form of data is known as E-R model. (or) E-R diagram.

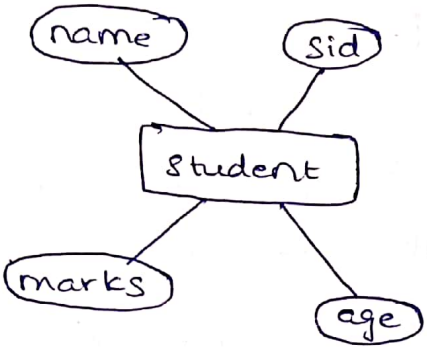
* It is conceptual view of database.

Elements / components / concepts of ER model: There are 3 Elements in an 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 a rectangle ( )
* For ex, University database contains Entities like Student, faculty, courses, classrooms 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 collection of Entities that have same attributes.

* It is intension of Entity set.

Ex: E_1 is an Entity having Entity type student and set of all students is called as Entity set.

Weak Entity: The Entity doesn't have primary key is called as weak Entity.

* It is represented as Double rectangle ( )
* Weak entity always depends on strong Entity.

Strong Entity: The Entity having a primary key is known as Strong Entity.

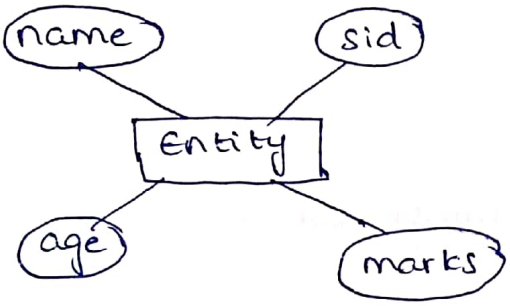
Associative Entity: The Entity which look like a relationship

* It is represented as ' '

2. Attributes : The properties or characteristics of an Entity is called as an attributes.

* It is represented by Eclipse symbol (◯) in E-R diagram.

* For Example, the student is an Entity which contains name, sid, age, marks as attributes.



Types of Attributes:

(a) Simple Vs composite Attributes

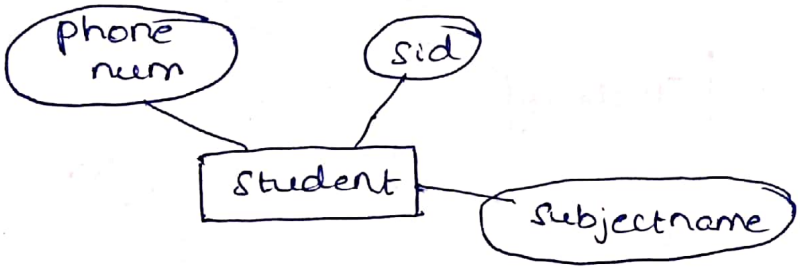
(b) Single valued Vs multivalued attributes

(c) Stored Vs derived attributes

(d) Complex attributes.

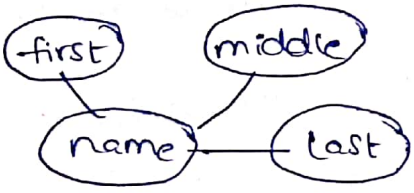
Simple: The attribute which contains single atomic value and cannot divide further.

Ex: Student id, Student phone number, Subject Name Etc



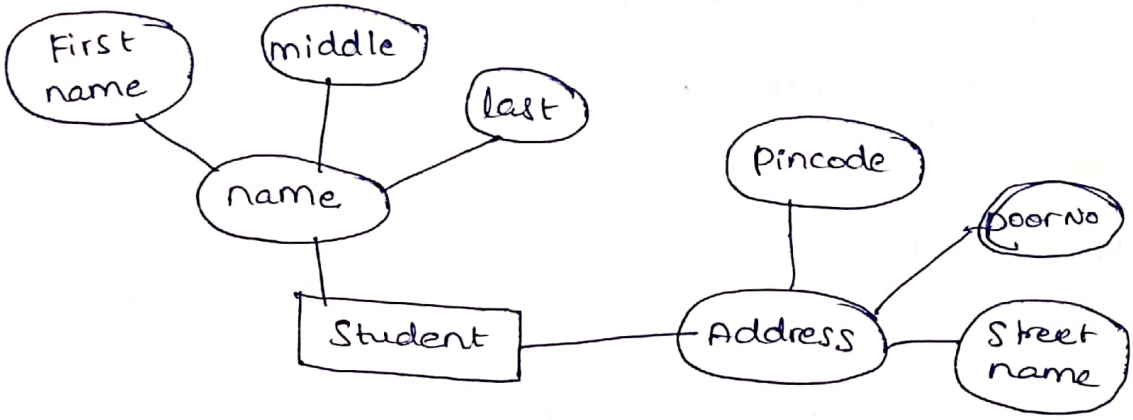
Composite Attribute: The attribute which contains more than one Simple attributes.

* It is represented as



* For Ex 1. student complete name contains first name,

2. Address also contains door num, street name, pincode etc



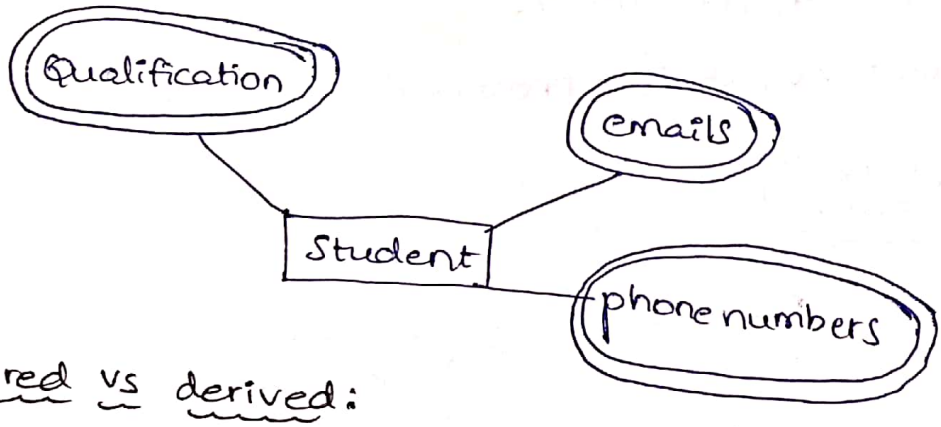
(b) Single vs Multivalued valued

1. Single valued: The attribute which contains only one value.

Ex: Blood group, pan card Number.

2. Multivalued: The attribute which contains more than one value.

* It is represented as double Eclipse (◎)
* Ex A person may have more than one phone number, Email Id's, Qualifications etc.



(c) Stored vs derived:

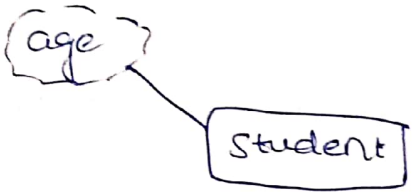
1. Stored Attributes: The attribute which stores the value and supply the value to the derived attributes.

Ex: Date of Birth (DOB)

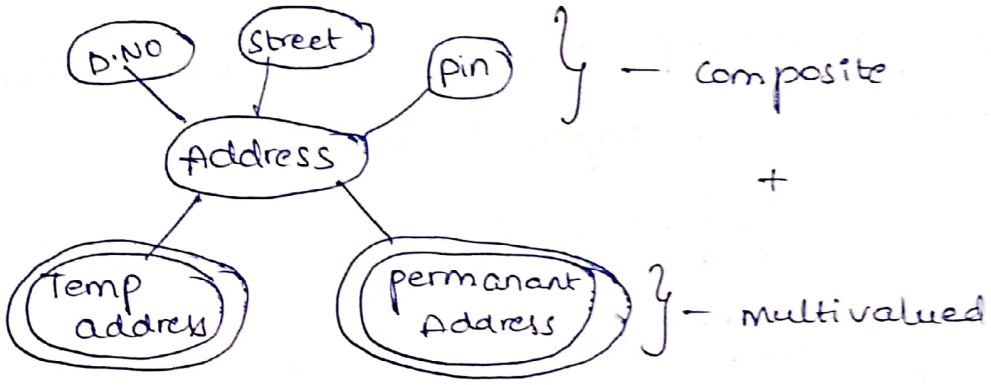
2. Derived: The attributes which are derived from the stored attributes.

* It is represented as dotted Eclipse (<...>)

Ex:



(d) Complex attributes: It is a collection of composite & multivalued attributes.

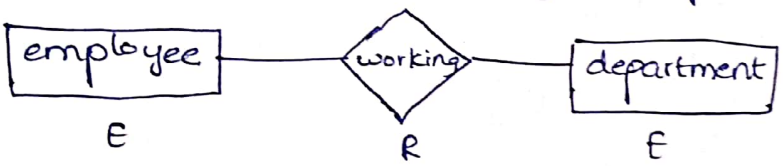


3. Relationship:-

→ Relation between Entities (or) association among Entities

→ It is represented as [◇] Diamond in E-R model.

→ Foreg:- Employee is working in department.



Binary Entity

→ Ex:-2
- Students are studying in college
- Entities are in noun form, Relations verbform
- In above Examples working, studying are called as relationship.

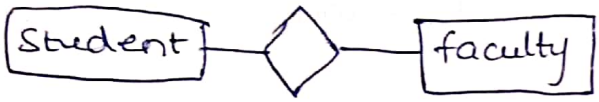
→ Relationship Set: The collection of / set of relationship of similar type is called as Relationships.

→ Degree of relationship: No. of Entities participated in a relationship.

* Types of relationships :- There are 3 types of relationship in E-R model.

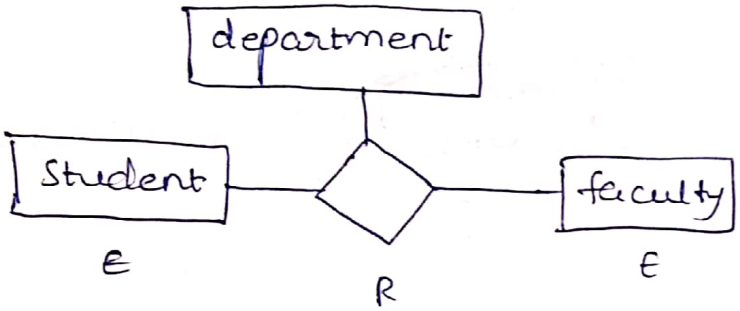
a) Binary relationship: It means it contains relationship b/w 2 Entities. It's degree is 2.

Ex:

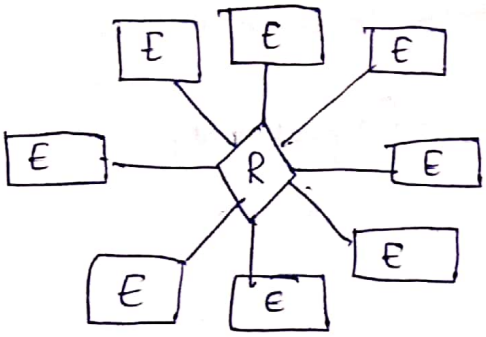


b) Ternary relationship :- It means it contains relation b/w 3 Entities & it's degree is 3.

Ex:



c) n-ary relationship :- It means it contains relation b/w more than 4 Entities degree = n.



→ Binary RS is further divided into

i) one - to - one mapping (1-1)

ii) one - to - many mapping (1-M)

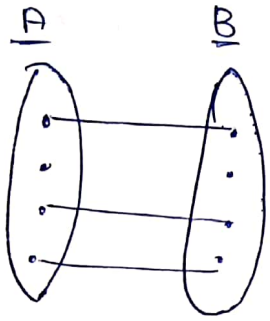
iii) many - to - one " (M-1)

iv) many - to - many " (m-M)

Cardinality: The no. of times an Entity of Entity set participated in a relationship is known as cardinality

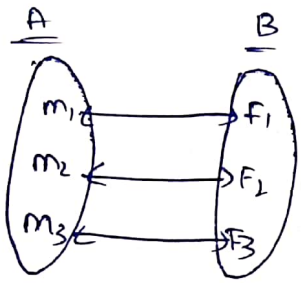
→ One-to-one mapping :- One Entity from Entity set A that can be in a relation with almost one Entity in Entity Set B and vice versa.

form:-



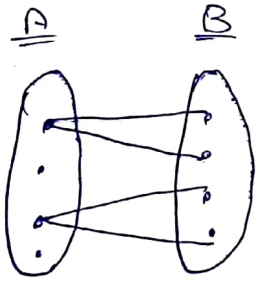
* Let us consider Example of a male married to female & female married to male.

Ex:

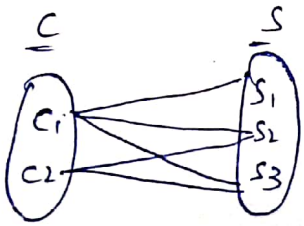


2. One-to-many mapping : One Entity from Entity set A participated in a relationship with more than one Entity from Entity Set B. However more than one Entity from Entity Set B relation with atmost Entity in Entity Set A.

form :-

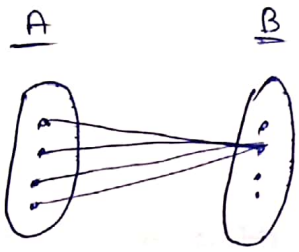


Ex: One course taken By many Students. However, many Students take one course.

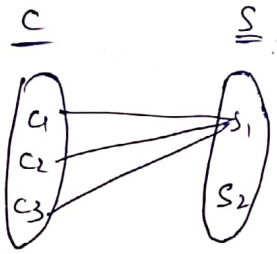


3. many-to-one mapping: more than one Entity from Entity set A participated in relationship with one Entity of Entity B. However one Entity of Entity set B is in relation with more than 1 entity from Entity A.

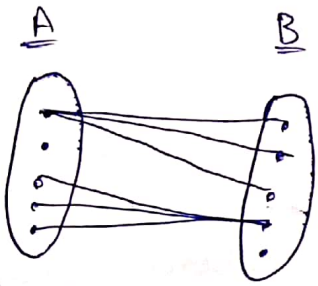
Form:



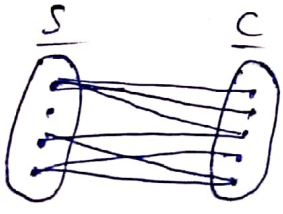
Ex: many Courses taken by one student (or) One student taking many courses.



4. many-to-many mapping: One Entity from Entity set A that can be related with more than one Entity in the Entity set B. and one Entity from entity set B that can be related with more than one Entity in Entity set A.



Ex: One Course taken by many students & one student taken many Courses.



23/7/19 Constraints in E-R model

There are 2 types of constraints

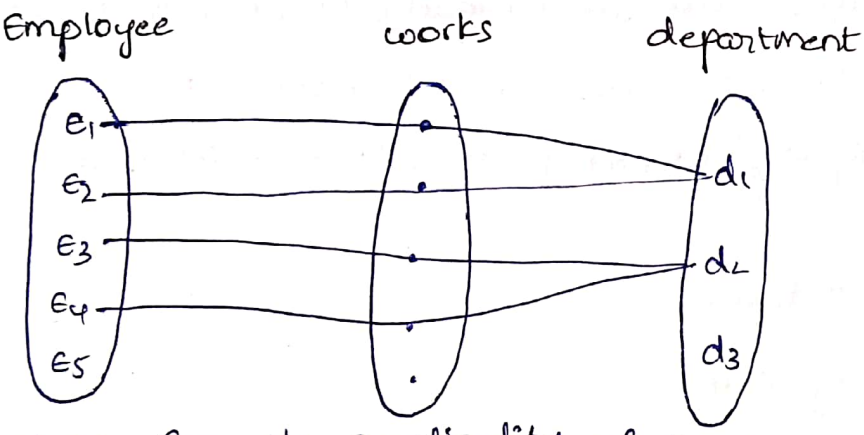
1. Structural constraints
2. Key constraints

1. Structural constraints : These are further divided into two types.

a. Cardinality ratio
b. participation.

a. Cardinality ratio : The maximum no. of times an Entity parti-cipated in a relationship.

Statement: Every Employee works for Exactly one department can have many Employees. newly formed department not have any Employee.



In above Example cardinality of Employee e_1 is 1, cardinality of dept d_1 is 2.

participation : The min no. of times an Entity participated in a relationship. Sometimes, it is also called as minimum cardinality.

It is further divided into 2 types

a. Total participation
b. partial participation

Total participation : If all Entities participated in a relation-ship.

→ It is represented as "=" in E-R diagram.

Partial participation: If some of Entities are not participated in a relationship.

→ It is represented as "—"

Note:

(i) minimum cardinality is 'zero' then it is partial participation.

(ii) min cardinality is "1" then it is total participation.

(iii) Max cardinality is "1" if an Entity participated only once in a relationship.

(iv) Max cardinality is "n" if an Entity participated 'n' times in a relationship

2. Key Constraints: These are uniquely Identified in a table. These are Attributes (or) set of Attributes that are uniquely Identifies an Entity within a Entity set.

These are two types.

1. primary key
2. Foreign key

Primary key: The primary key contains unique values & never contains null values.

→ It is unique column in a table.

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

Ex: Student table

Sid	name	age	marks
501	A	30	90
502	B	40	95
503	C	50	97
504	D	60	100

Primary key

Syntax:

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

Foreign key: It means it links 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 b/w tables.

cid	sid	cname
1	501	k
2	502	H
3	503	N
4	504	Z

In above course table and student table.

In student table sid column is primary key, in course table sid column is foreign key.

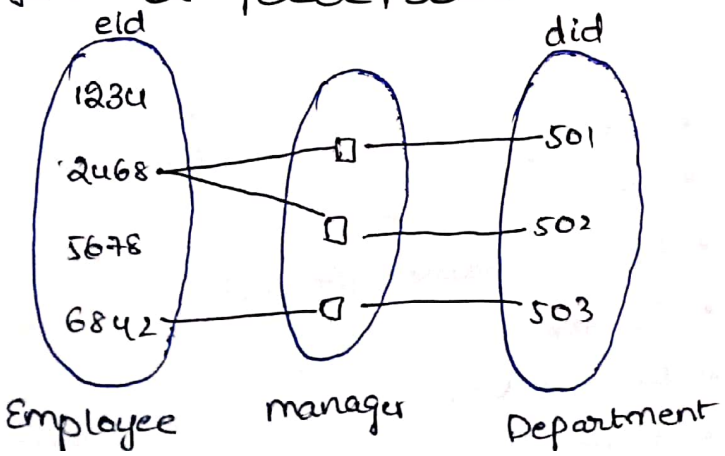
Note: If foreignkey is formed with two or more tables

Ex:

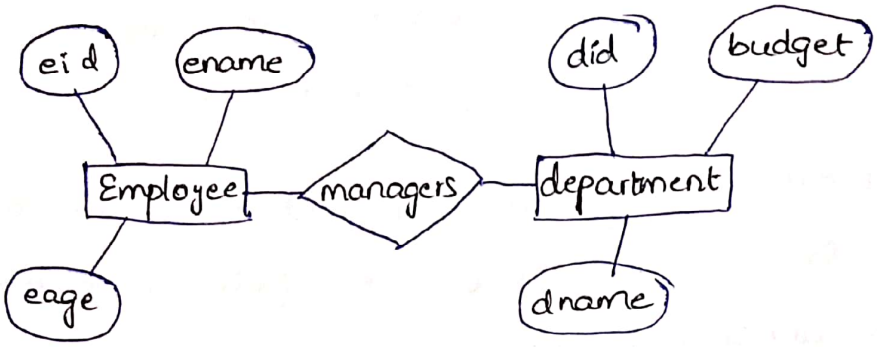
```
Create table course(cid int, sid int, c name varchar(20),
                    primary key(sid, cid), foreign key(sid),
                    references. Student (sid));
```

```
desc course;
```

2) Diagram for participation



i) Diagram for Key constraints



Integrity constraints

a) Key constraints
b) Structural constraints
c) domain constraints
d) Entity integrity constraints
e) Referential integrity constraints

I.C 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 be performed in such a way that data integrity is not affected.

* This is used to guard against accidental failures of database.

* The above five are the Types (or) modelling of constraints.

c) Domain constraints: It means that valid set of values of an attribute.

* We need to define
  - Datatype
  - length (or) size
  - is null value allowed
  - is the value unique
  - Defaut value
  - Range of values

∴ Domain constraints = Datatype + constraints (foreignkey / default / NOT NULL / unique / primary key)

* Each table has set of columns and each column allows same type of data based on its datatype.
* The column doesn't accept values of any other datatype
* Every attribute is bound to specific range of values for example age cannot less than '0' and telephone no. cannot contain a digit outside 0-9.
* The datatype related with domains include character, integer, date, time, currency etc..

Ex:

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

Not allowed because it is an integer attribute.

d) Entity Integrity constraints: It defines the primary key value cannot be Null.

* This is because primary key value is used to identity individual rows in a table and if primary key has a null value then we cannot identify those rows.
* A table can contain a Null value otherthan primary key field / column.

Eg

sid	name	age
1	a	20
2	b	21
-	c	22

Not allowed because primary key cannot contain null value

e) Referential Constraints: It can be specified b/w 2 tables

* It maintain consistancy & accuracy b/w tables.

Rules:

1) We cannot delete a record (row) from primary table if matching record found in Secondary table.

2) you cannot change a primary key in primary table if that related to the foreign key of the Secondary table.

3) Secondary table must be null (or) available in primary table.

Ex:

Student table
(primary table)

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

↓
PK

course table
(secondary table)

Sid	cid	cname
501	1	abc
502	2	lmn
503	3	
504	4	xyz
505	5	pqr

↓
fk

Not allowed because Sid = 505 is not defined in primary key of student table.

Reduction/conversion to E-R diagrams to table/relation

→ ER diagrams is converted into tables in Relational model because Relational model can be Easily implemented by RDBMS like mySQL, Oracle etc...

→ Following rules are used for converting E-R diagram into the tables

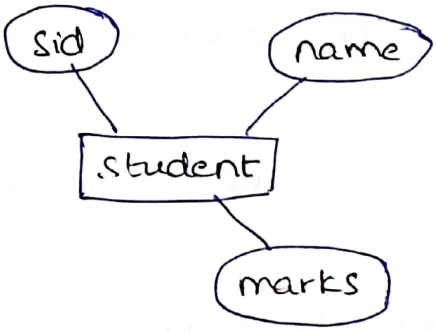
Rule-01: For strong Entity Set with only simple attributes

→ A strong Entity set with only simple attributes will require only one table in relational model

→ Attributes of the table will be attributes of Entity Set

→ The primary key of the table will be the key attribute of the Entity k Set.

Ex:



Schema : Student (sid, name, marks)

Ex

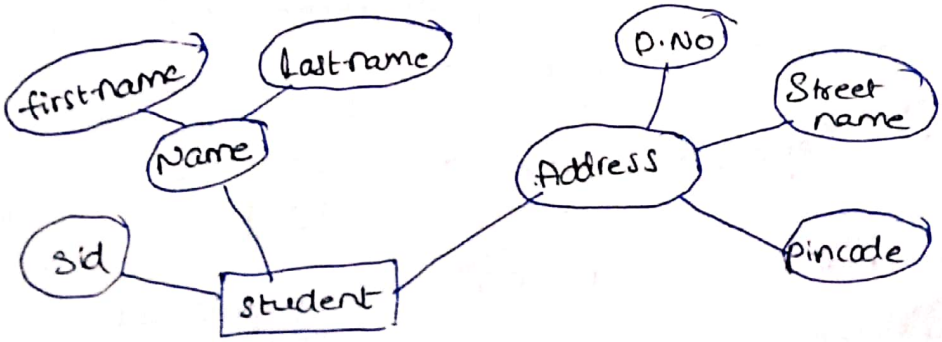
Rollno	name	marks

→ Schema : Student ( )

Rule 02: For strong Entity Set with Composite Attributes

* A strong Entity set with any one 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	DNO	Streetname	pincode

Schema : Student (sid, first-name, last-name, DNO, street name, pincode)

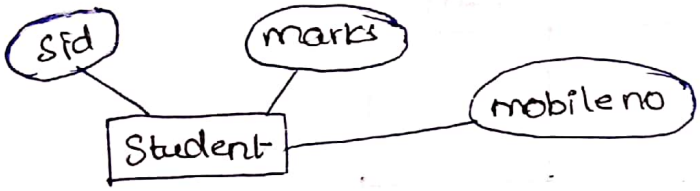
Rule-03 : For strong Entity set with multivalued Attributes.

→ A strong entity set with any no. of multivalue of Attributes will require two tables in relational model.

→a) one table will contain all the simple attributes with the primary key

b) Other table will contain the primary key and all the multivalued Attributes.

Ex



1.

Sid	marks

Sid	mobileno

Rule-04 : Translating relationship set into table

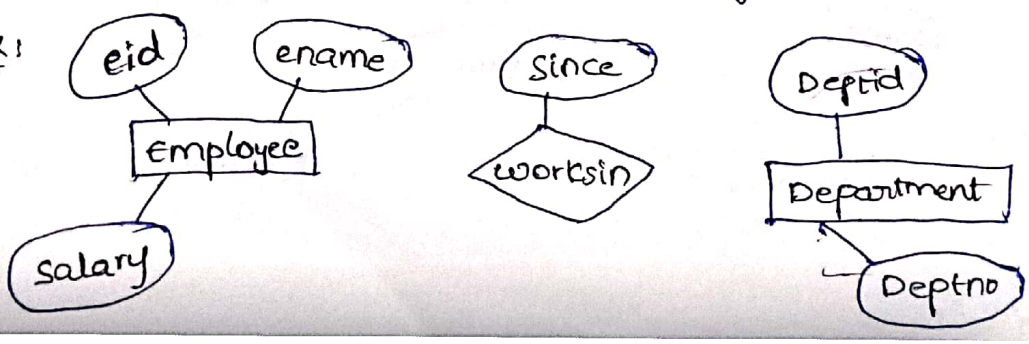
A relationship set will require one table in relational model

Attributes of the table are:

a) primary key attributes of the participation Entity sets.

b) Its an descriptive attributes if any

Ex:



eid	Did	Since

Schema: works in (Eid, dept-id, since)

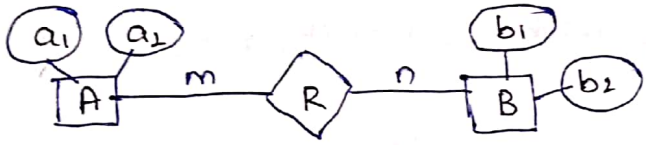
Note: If we consider the Overall ER diagram three table will be required in relational model.

→ One table for Entity set "Employee"
→ One table for Entity set "Department"
→ One table for relationship "works-in".

Rule 05: For Binary relationships with cardinality ratio

case 01: Binary relationship with cardinality ratio 1:1
case 02: Binary relationship with cardinality ratio 1:n
case 03: Binary relationship with cardinality ratio n:1
case 04: Binary relationship with cardinality ratio m:n

case 1: Binary relationship with cardinality ratio m:n



3 tables

A

a_1	a_2

R

a_1	b_1

B

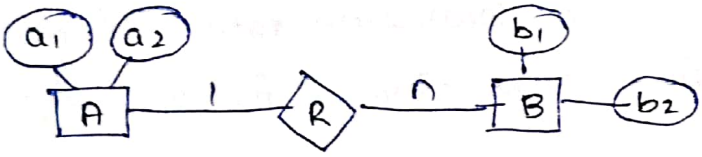
b_1	b_2

(i) A (a_1, a_2) → FK

(ii) R (a_1, b_1) → FK

(iii) B (b_1, b_2) → FK

case 2: Binary relationship with cardinality ratio 1:n



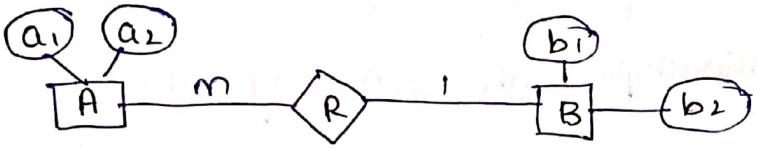
Here two tables are required

1. A (a_1, a_2)
2. BR $(a_1, b_1 b_2)$ → Fk

Note: The combined table will be drawn for Entity set B and relationship set R

Ex: R (a_1, b_1)
B (b_1, b_2)
BR (a_1, b_1, b_2)

Case 3: For Binary relationship with Coordinality ratio n:1



there, two tables will be required

1. AR (a_1, a_2, b_1)
2. B (b_1, b_2)

Note: Here combines table will be drawn from Entity set A and relationship set R.

Case 4: Binary relationship with Cardinality ratio 1:1

there, two tables will be required. Either combine 'R' with 'A' or B

way-01: AR (a_1, a_2, b_1)
B (b_1, b_2)

way-02 A (a_1, a_2)
BR (a_1, b_1, b_2)

Rule points to remember: while determining the minimum no. of tables required for being relationship with given co-ordination ratio, following rules to remember

a) For binary relationship with Cordinality ratio m:n, Seperate and individual table will be thrown for Each Entity 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 drawn for many side (entity set and relationship set

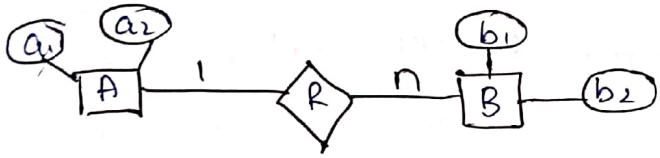
c) For binary relationship with cardinality ratio 1:1, two tables will be required you can combine the relationship set with any one of the Entity sets.

Rule -06

For Binary relationship with cardinality constraints & participation constraints.

case -01 :

Binary relationship with cardinality constraints and Total participation from one side



Because, cardinality ratio 1:n, so it will combine the Entity set B and relationship R.

Then, two tables will be required 1. A (a_1, a_2)

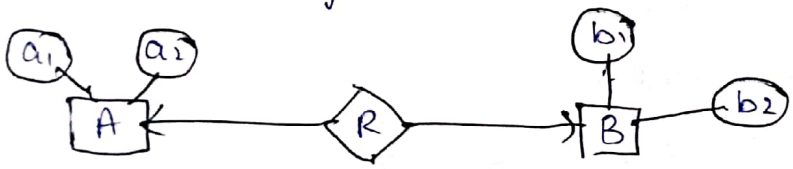
2. BR (a_1, b_1, b_2)

fk

Case 02:

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 participate then relationship is represented in only one table



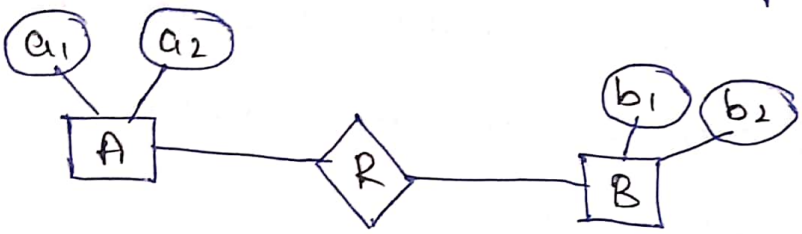
there, only one table required

ARB (a_1, a_2, b_1, b_2)

Rule -07:

Binary relationship with weak entity set

Weak Entity set always appears in association with identifies relationship with total participation constraint



there two tables will be required

1. A (a1, a2) → Fk
2. BR (a1, b1, b2)
   - a1 ↓ Fk
   - b1 ↓ FK