INTRODUCTION TO DBMS & ER MODEL

DEF: DBMS is collection of interrelated data and programs used to handle that data. Primary goal is to store and retrieve information from Database in convenient and efficient manner.

- For managing the Database two important task are conducted.
  - Define structure for storage of information.
  - Provide mechanism to manipulate information.

In addition database must also ensure safety of info stored.

CHARACTERISTICS:

- Rrepresentation of some aspects of real world application.
- Systematic management of information
- Representing the data by multiple view
- Effecient & Easy implementation of operations
- Maintain data for specific purpose
- Represents logical relationship between records and data.

PURPOSE:

- Earlier they were created to manage commercial data
- This data typically stored in files. To allow user to manipulate these files various programs are written.
- Thus with new needs new program apps were written.
- This typical file processing system is supported by OS

ADVANTAGES

- removes data redundancy
- retrieve required data.
- Data isolation for efficient use
- Maintains Data integrity
- Atomicity can be maintained
- Allows concurrent access
- Security policies can be applied.

DISADVANTAGES

- complex design
- Large amount of investment
- Damaged part affects all
- High conversion cost
- Proper trained individuals required.

Atomicity: changing one entry is reflected by whole database

Conversion: converting data base structure

OPERATIONS:

Addition of new data
Updating data
Deleting data
Sorting data.

FILE PROCESSING SYSTEMS

The system that needs different program to store or add the permanent records stored as file

DISADVANTAGES:

Data redundancy	dublication of data may occur as different programmer has different file.
Data inconsistency	When various copies of data may not match example: Employees address
Difficulty in accessing data.	File system doesnt allow convenient access of desired data.
Data isolation	As data is scattered in different files and all files have different format. it becomes difficult to retrieve data.
Integrity problems	Data values entered must fall within specific range or format. With several files enforcing such constraint. data integrity difficult to assess.
Atomicity problem	Particular operation must be carried out entirely or not at all. This is difficult to ensure
Concurrent access anomalies.	Synchronized data access not possible thus accessing data concurrenty is not possible
Security problem	ad hoc manner of application program makes it difficult to implement security.

	Database system	File System.
redundancy	less	more
Security	high	low
inconsistency	less	more
physical address	User doesnt know	User needs to locate
retrieval	in desired format	not in desired format
concurrency	is present	is not present
Definition of data in system	structured manner with co-relation among data	un structured data usually in isolated form
Use case.	high constraint systems	less [constraint / demanding] systems

DATABASE SYSTEM APPLICATION

1. Accounting maintaining employee salaries, payroll taxes etc.
2. Manufacturing supply chain management & product tracking
3. Banking customer info, account, loans etc.
4. Universities student info, course registration etc.
5. Reservation system: for ailines, railways, transport systems.
6. Telecommunication: records of call, monthly bills, maintaining balances, storing info, etc.

VIEW OF DATA

Database is a collection of interrelated data and set of programs that allow user to access or modify the data.

Abstract view of system is a view in which system hides certain detail of how the data are stored and maintained.

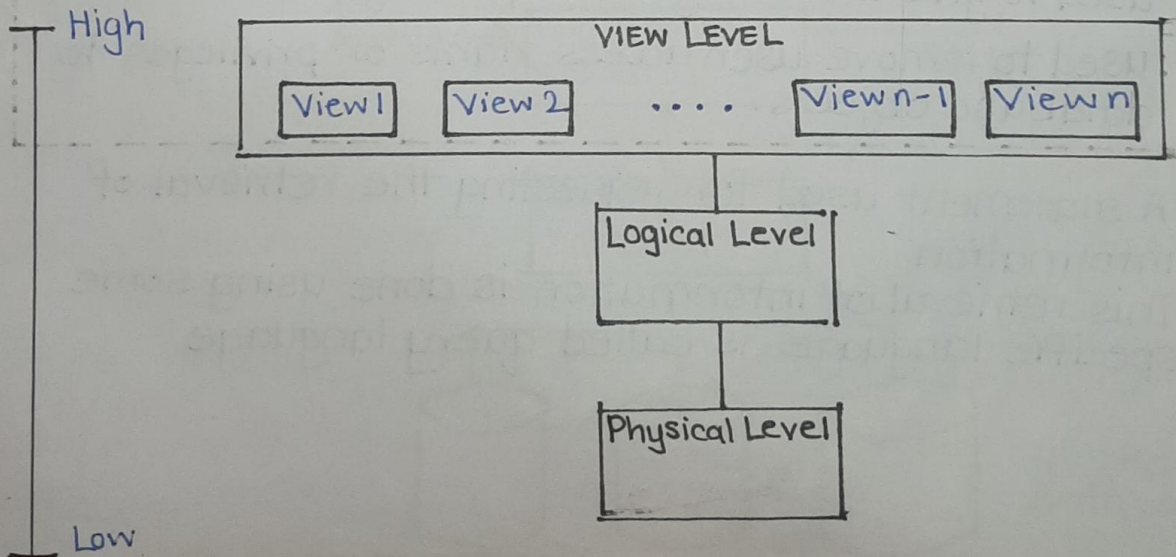
The main purpose of database is to provide abstract view

This helps users retrieve data efficiently

For simplifying the user interaction there are multiple levels

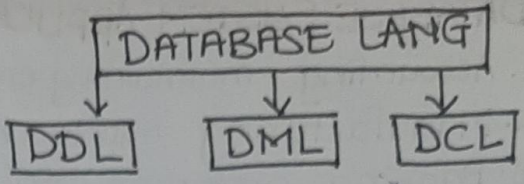
Physical Level	Logical Level	View Level
• This is lowest level	• Next higher level	• Highest level
• Describes how data is stored.	• Describes relation between data	• can provide access to part of database
• stores data at physical level	• Describes entire database in terms of small number of small structures	• Helps in simplyfing interaction with system
• Describes complex low-level data structure	• Decides what info to hide	• Can provide multiple view of same system

DIAGRAM



DATABACE LANGUAGES

There are three types of Languages



DDL

- Data Definition Language
- Specifies database schema by set of definitions
- Language used for creating and modifying the structures of tables, views, indexes etc.
- Also used to specify additional properties of data.
- Common Commands : Create, Alter, Drop

Create: build new table
Alter: add additional column or drop existing.
Drop: Delete table / view

DML

- Data manipulation Language
- Enable users to access or manipulate data as organized by appropriate data model
- Type of access : Retrieval, Insertion, Deletion, Modification
- Types of DML :

Procedural: Requires user to specify what data are needed and how to get that data.
Declarative: Requires user to specify what data are needed but not how to get that data.

DCL

- Data control Language
- Used to control access to data stored also called as authorization
- Typical command used are:

Grant: used to give access rights or privilege to database
Revoke: used to remove user access rights or privileges to database objects.

QUERY: A statement used for requesting the retrieval of information
This retrieval of information is done using some specific language is called query language

DATABASE SYSTEM STRUCTURE

Three schema Architecture.

Goal: to seperate user application from physical address.

It is divided into three levels:

INTERNAL

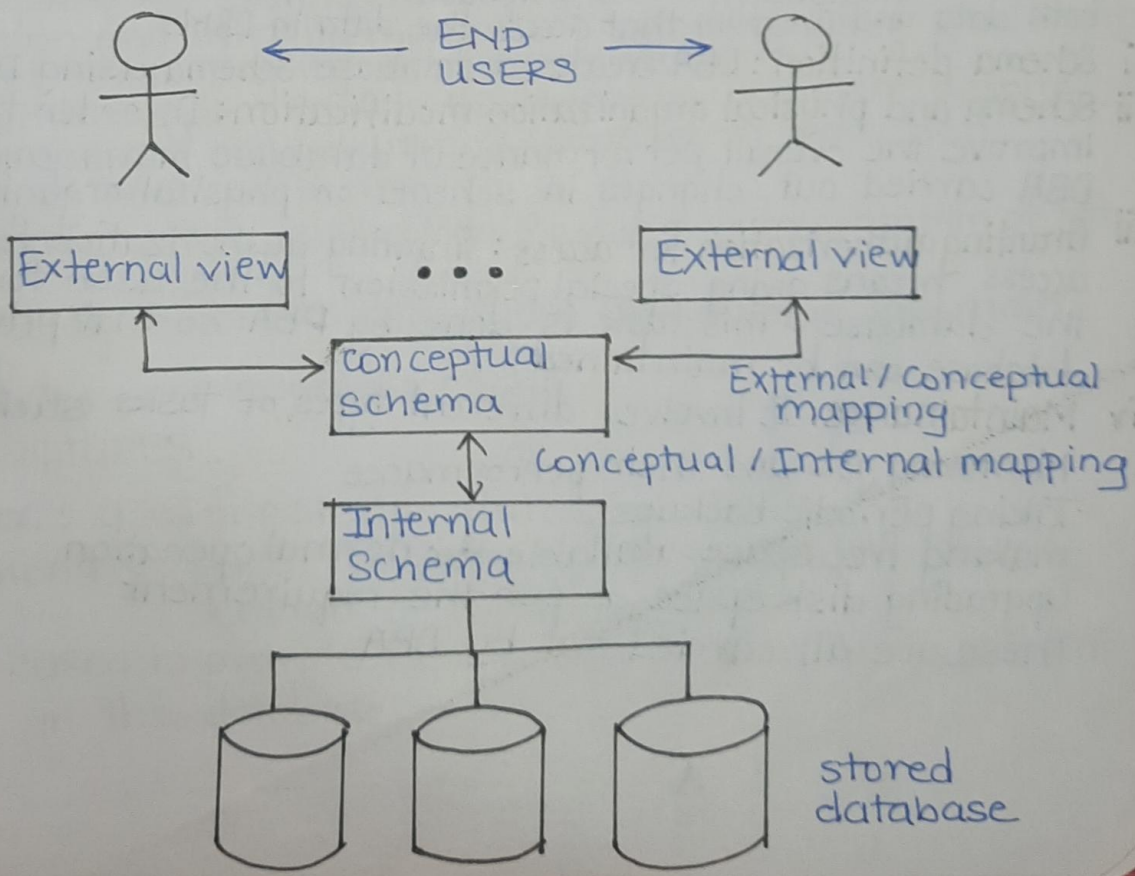
- contains internal schema
- represents physical storage structure maintained by
- Software and user not allowed to modify it.
- closest to physical storage
- Typically describes record layout of files, types of file access path etc.

CONCEPTUAL

- contains conceptual schema
- hides the internal level's details
- Also called logical level as it contains construct used for designing database.
- Contains info like table name, column index, constraints etc
- Representational data model is used to describe conceptual schema when database system is implemented.

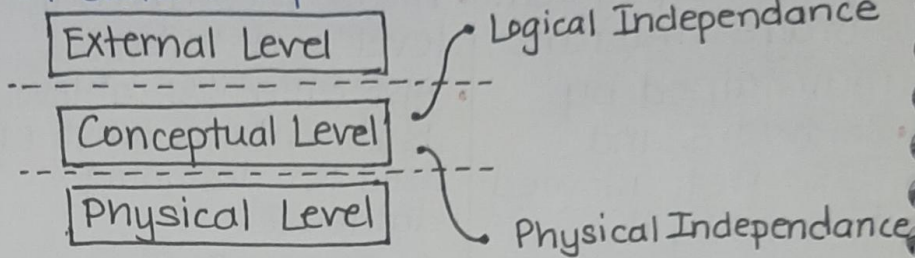
EXTERNAL

- Contains external schema / user view
- At this level user can only see data stored in database.
- They can see whole data values or specific records
- They will not have any information about how they are stored in the database.



DATA INDEPENDANCE

- ability to change data at one level without change at another level.
- Data independance is one of the important characteristics of database management system.
- By this property, the structure of the database or the values stored in the database can be easily modified by without changing the application program
- There are two types of data independance.



i. Physical Independance :
- Allows modification of physical schema without affecting conceptual schema

ii Logical Independance :
- This is the kind which allows manipulation of conceptual schema without affecting external schema.

RESPONSIBILITIES OF DBA

Database administrator (DBA) is a person who has a central control over both data and program that access the data in DBMS

i Schema definition : DBA creates a database schema using DDL statement

ii Schema and physical organization modification : In order to ~~keep~~ improve the overall performance of database management system DBA carried out changes in schema or physical organization

iii Granting authorization for access : Granting authorization for data access means giving special permission to the user for accessing the database. This task is done by DBA so that privacy of database can be maintained

iv Maintainance : It involves different types of tasks such as
Montoring job and their performance
Taking periodic backups
making free space available for normal operation
upgrading disk space as per the requirement
These are all carried out by DBA

DATA BASE USERS

There are four different types of user differentiated by the way they interact with the system

i Naive users : interact with the help of previously created program known as application

ii Application programmer : professionals who write applications usually Rapid action development (RAD) tools are used

iii Sophisticated user : users who interact with system without writing program like query or prompts

iv Specialized user : They are like sophisticated users who write specialized database application which dont fit into traditional data processing framework.

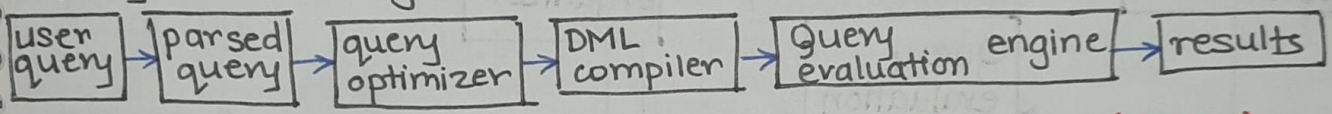
QUERY PROCESSORS :

- Helps database systems to simplify and facilitate access to data.
- with following components various functions are performed.

i DDL interpreter : translator for DDL statements in Data dictionaries

ii DML compiler : translates DML query language into evaluation plan.

iii Query evaluation engine : executes low-level instruction by DML compiler



STORAGE MANAGER :

(database) (query)

- component which provides interface low-level data & application program
- It is responsible for storing, retrieving and updating data

i Authorization and integrity manager : validates user who want access and tests integrity of constraints.

ii Transaction manager : Ensures DB remains consistent despite of system failure and concurrent transaction execution proceeds without conflicting

iii File manager : Manages allocation of space & representation of info on disk

iv Buffer manager : fetches data from disk storage into main memory.
Also decides what data to cache

DATA STRUCTURES

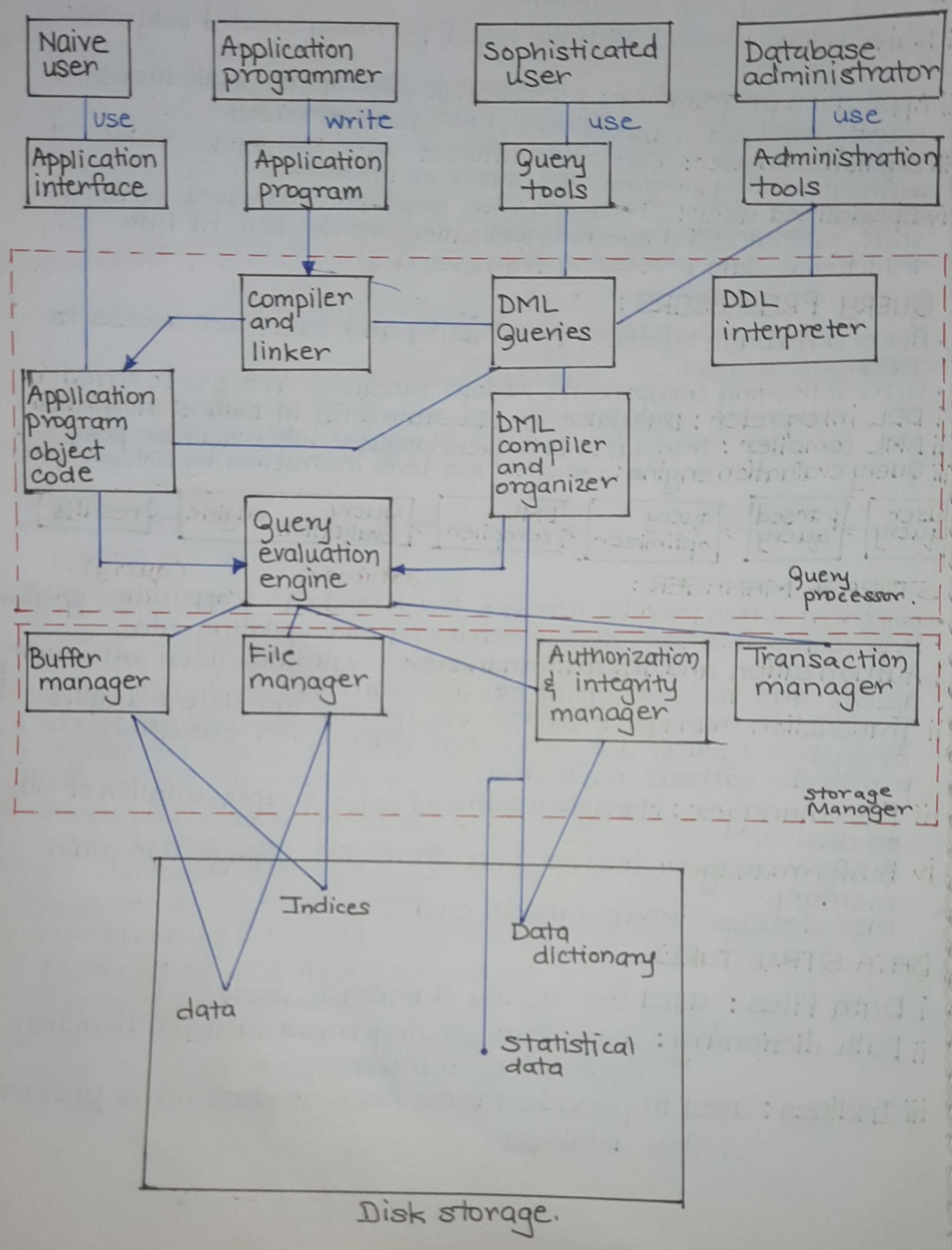
i Data files : used for storing database itself

ii Data dictionary : used for storing meta data particularly schema of database.

iii Indices : used to provide fast access to data items present in the database.

E-R model relevant here

ARCHITECTURE OF DBMS



DATA MODELS

- It is a collection of conceptual tools for describing data, relationships among data, semantics of data and constraints
- Data model is a structure below database
- Data model provides a way to describe the design of database at physical, logical and view level
- There are various data models used in database system and these are as follows

• RELATIONAL MODEL

- The relational model consists of collection of tables which store data and also represent the relationship among data.
- The table is also known as relation
- The table contains one or more columns and each column have unique names
- Each table contains record particular type, and each record type defines a fixed number of fields or attribute
- Example:

Seat no.	Name	Marks	City
101	Ram	98	Pune
102	Shyam	76	Mumbai

Advantages

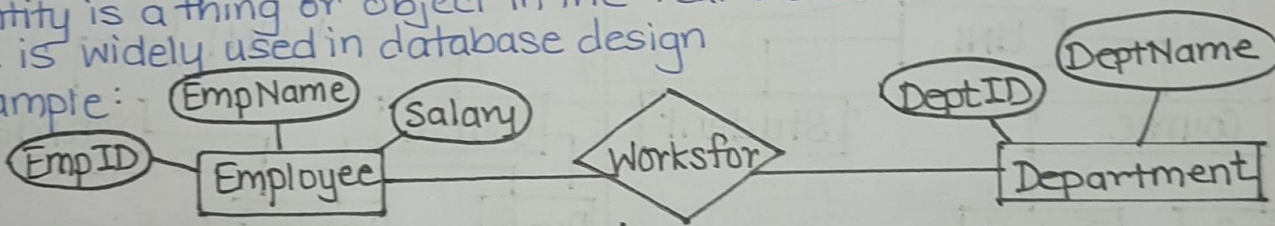
- Structural Independance: making change without affecting others.
- Conceptual Simplicity: Easy to understand
- Query capability:
- Easy design maintainance & usage

Disadvantages

- Requires poweful hardware and large data storage device
- May lead to slower processing time
- Poor design leads to poor implementation.

• ENTITY RELATIONSHIP MODEL

- As the name suggests the entity relationship model uses collection of basic objects called entity & relationships
- Entity is a thing or object in the real world
- It is widely used in database design
- Example:



Advantages

- Simple
- Easy to understand
- Effective
- Integrated
- Easy conversign

Disadvantages

- Loss of information
- Limited relationship
- No representation for data manipulation
- No industry standard.

E-R model relevant here

- OBJECT BASED DATA MODEL

object oriented languages lead to object based data model

It combines object oriented features with ~~reb~~ relational model

Advantages

- Enriching modelling: capable of modeling real world objects
- Reusability
- Support for schema evolution: tight coupling between data and application
- Improved performance: use of objects increase performance

Disadvantages

- Lack of universal data model
- Lack of experience: use is limited and model is dependant on skilled programmer
- Complex: More functionalities thus design is complex.

- SEMI STRUCTURED DATA MODEL
- It permits the specification of data where individual data items of same type may have different set of attributes
- Extensible Markup Language (XML) is widely used to represent semi structured language.

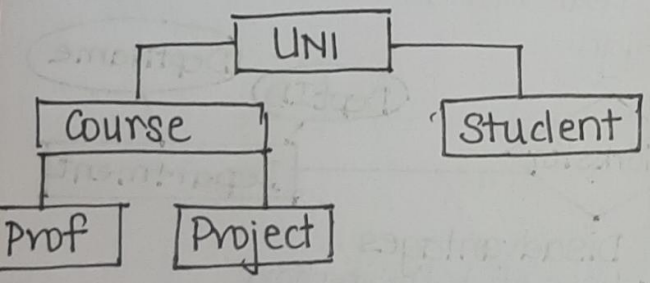
Advantage

- Data not constrained by fixed schema.
- Flexible
- Portable

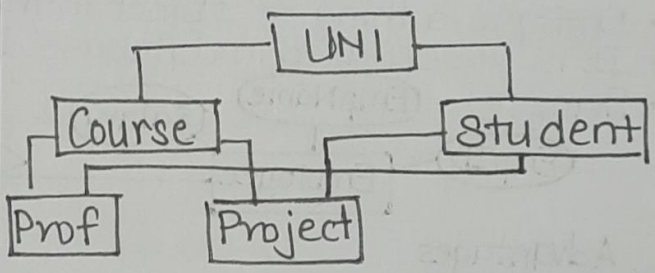
Disadvantage.

- Queries are less efficient

- HIERARCHICAL MODEL
- In this each entity only has one parent but can have multiple children
- At top only root node present.



- NETWORK MODEL
- Enhanced Hierarchical model
- No single parent concept
- Can have many relationship.



DATABASE DESIGN AND ER MODEL

- Data Modelling in database management system is based on Entity Relationship Modelling
- ER model is based on identifying entities and represent how they are related.
- ER model represents overall logical structure of database
- It is useful in mapping the meanings and interactions of real world entities onto a conceptual schema or database
- The er model consists of three basic concepts

ENTITIES & ENTITY SET

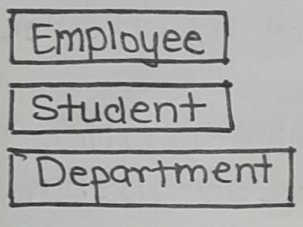
- Entity : An object that exists and is distinguishable from other objects
- Entity set : The entity set is a set of entities of same types

RELATIONSHIPS

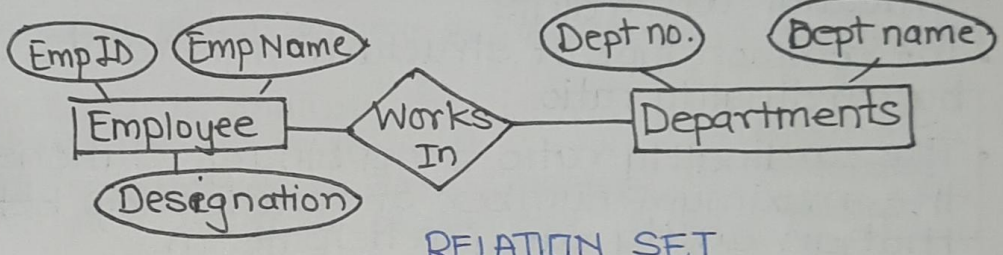
- Relationship : It is an association among two or more entities
- Relationship set : A collection of similar relationships

ATTRIBUTES

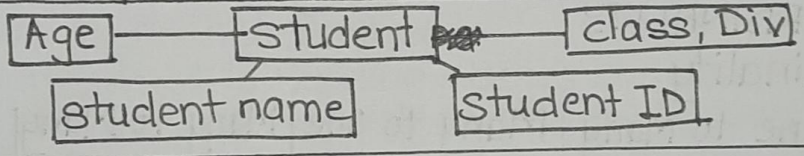
It defines properties of data object of entity



ENTITIES



RELATION SET

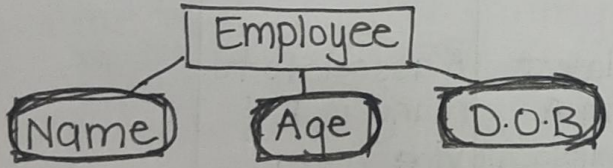


ATTRIBUTES

TYPES OF ATTRIBUTES

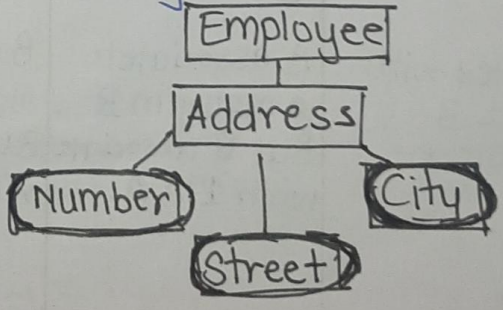
1. Simple Attribute

Attribute drawn from atomic value

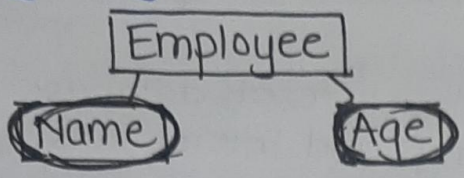


Composite Attribute

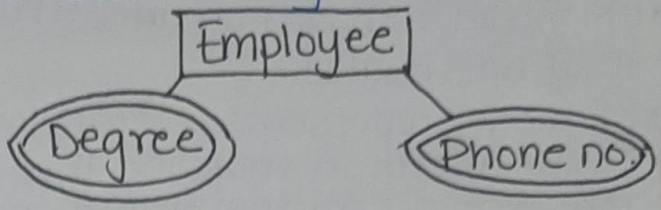
Attribute that contain hierarchy of attribute



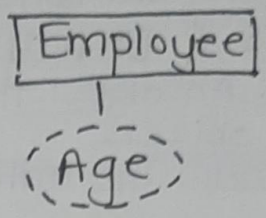
2 Single valued
Attributes represented using single value



Multivalued
Attributes with a set of values for each entity



3. Derived attribute
An attribute that contains value which can be calculated from other attribute.
like Age from D.O.B



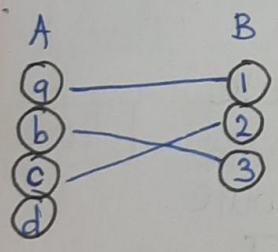
CONSTRAINTS

- Relationship types have certain rules that limit the possible combination of entities that can take part in a relationship. These rules or restrictions are called structural constraints
- The common type of structural constraints are represented by cardinality ratio
- The cardinality ratio for a binary relationship specifies the maximum number of relationship ~~between~~ instance that an entity can participate in.

TYPES OF CARDINALITY

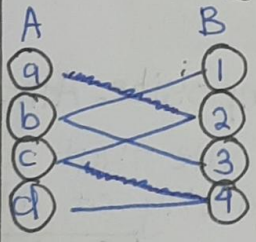
- Mapping Cardinality

1. One to one



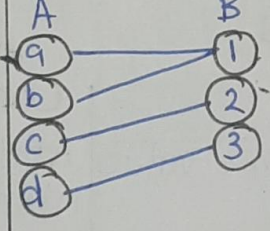
A associated with atleast 1 B & vice versa

One to Many



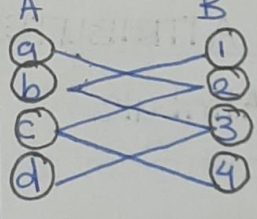
A associate to many in B But B associate with 1 in A.

Many to One



B associate to many in A But A associate with 1 in B

Many to Many



A associate to many in B & vice versa

Keys

used to identify entities uniquely from the given entity set.

A key can be a attribute or a set of attribute that help identify

Also can be used to identify relationship uniquely thus distinguish relationship from each other

The primary key of an entity set allowed us to distinguish among various entities of set

DESIGN PROCESS

i understand what data to be stored in database, what application must be built what are all those operations that are frequently used. by system.
- informal process, requires proper communication automated tools can be used for this purpose

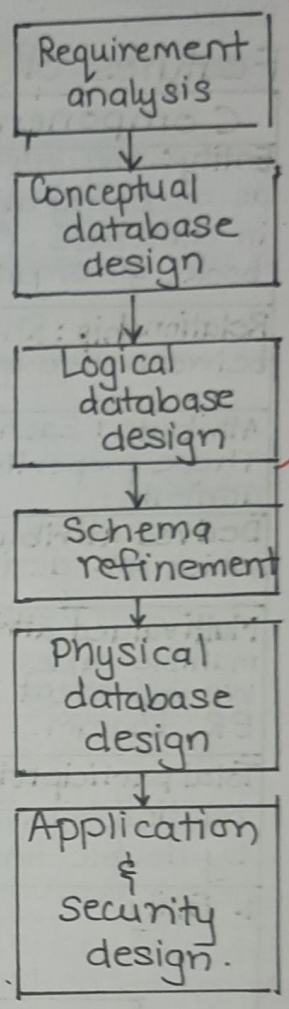
ii E-R model is built here (high level data model)
- goal is to create a simple description of the data that matches with requirement of user.

iii In this ER model is turned into relational database schema sometimes called as ~~relational schema~~ logical schema in relational data model

iv Relational database is analyzed to identify the potential problem and to refine it
The scheme refinement can be done with the help of normalizing and restructuring the relations.

v Database is refined further
- Task performed is building indexes on tables and clustering tables, redesigning some parts of schema obtained from earlier design steps.

vi Using design methodologies like UML (Unified Machine Language) the design of the database can be accomplished
- The role of each entity in every process must be reflected in the application task.
- For each role, there must be the provision for accessing some part of database and prohibited access to other parts of database
- Thus some access rules must be enforced on the app to protect the secure features

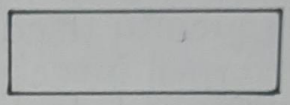
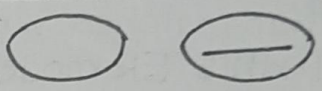
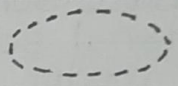
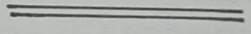


Database Design Process

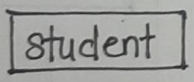
ER DIAGRAM

ER model is a model for identifying entities to be represented in the database and representation of how those entities are related.

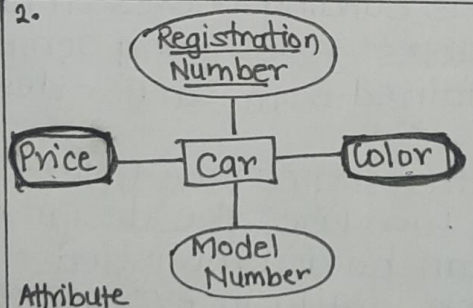
- ER model specifies enterprise schema that represents the overall logical structure of database graphically
- ER diagrams are used to model the real world objects like a person, a car, a company and relation between these objects.
- Features of ER model

Component	Symbol
Entity: Any real world object can be represented as an entity about which data can be stored in database. All the real world objects like a book, an organization, a product, a car, a person	
Relationship: Rhombus is used to setup relationship between two or more entities.	
Attribute: Each entity has a set of properties. These properties of each entity are termed as attribute.	
Derived attribute: Derived attributes are thpse which are derived based on other attribute	
Multivalued attribute: An attribute that can hold multiple values is known as multivalued attribute we represent it with double Ellipse in an ER diagram.	
Total participtation: Each entity is involved in the relationship. Total participation is represented by double line	

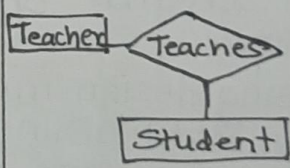
1. Entity



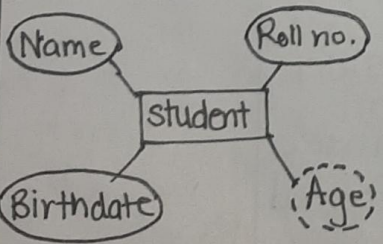
2.



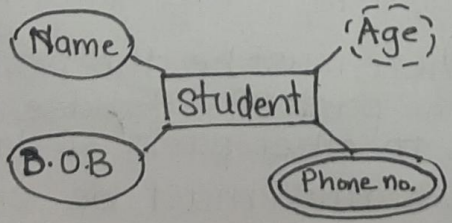
3. Relationship



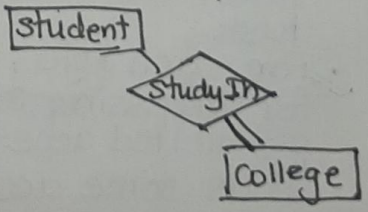
4. Derived attribute



5. Multivalued.

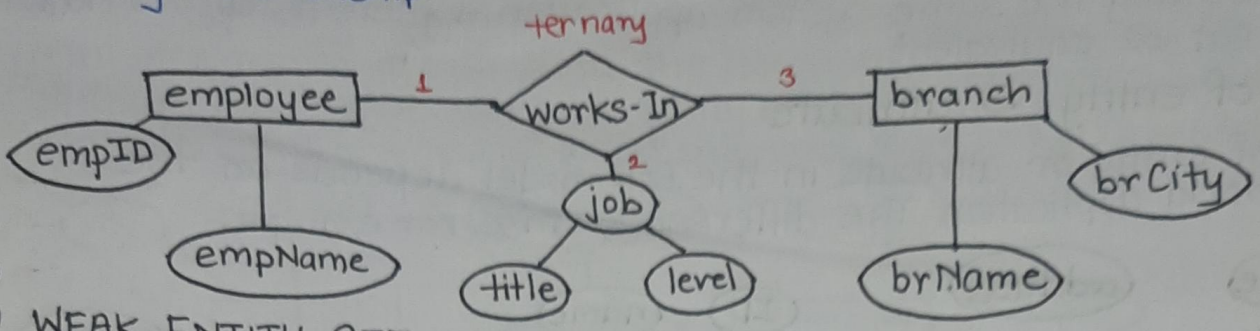


6. Total participation



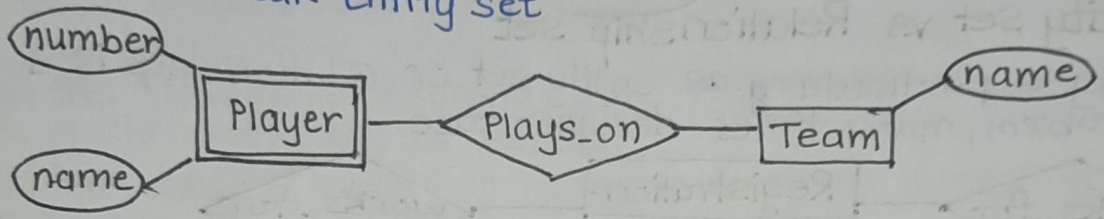
TERNARY RELATIONSHIP

A relationship where three entities are involved is called ternary relationship



WEAK ENTITY SET

A weak entity is a entity that cannot be uniquely identified by its attributes alone. The entity set which does not have sufficient attributes to form a primary key is called as weak entity set



STRONG ENTITY SET

The entity set that has primary key is called a strong entity set.

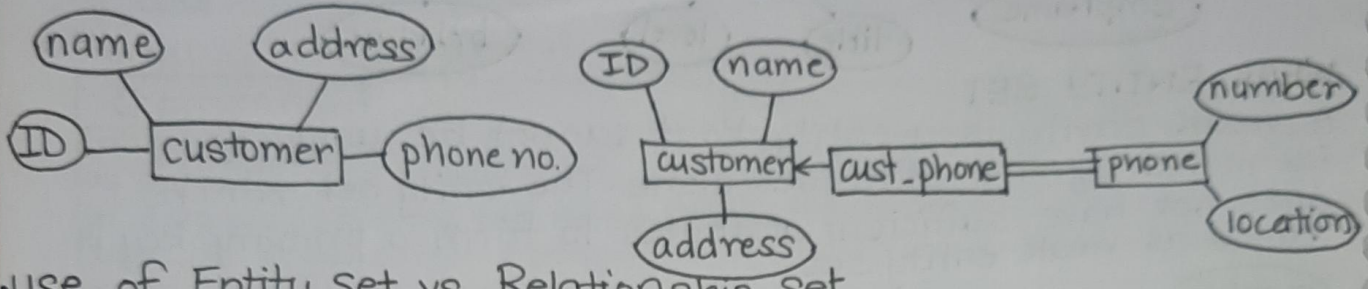
STRONG ENTITY SET	WEAK ENTITY SET
It has it own primary key	Doesnt have sufficient attributes to form a primary key on its own
Represented by a rectangle	Represented by double rectangle
Represents primary key which is underlined	Represents partial key/discriminator which is dashed underline
member is called dominant entity set	member is called subordinate entity set
relationship between two strong is ~~called~~ represented by diamond	relationship between strong and weak is represented by double diamond.
primary key uniquely identifies members	primary key is made of partial key of weak + primary key of strong.

DESIGN ISSUES

ER diagram can be designed in several different ways for representing the same system application. Different representation may not be equivalent

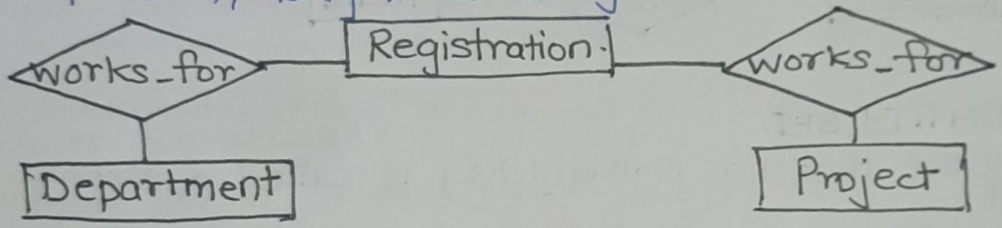
- use of entity. vs attribute

use of entity or attribute in the E-R model depends on the real world application The differences arise for example.



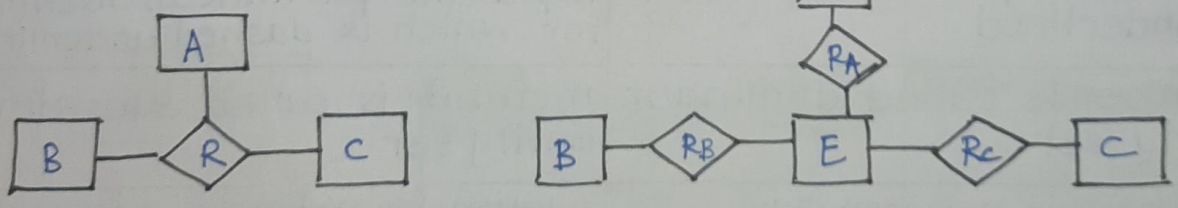
- use of Entity set vs Relationship set

Representing a particulary as entity set or relationship is a common problem, many times designer face.



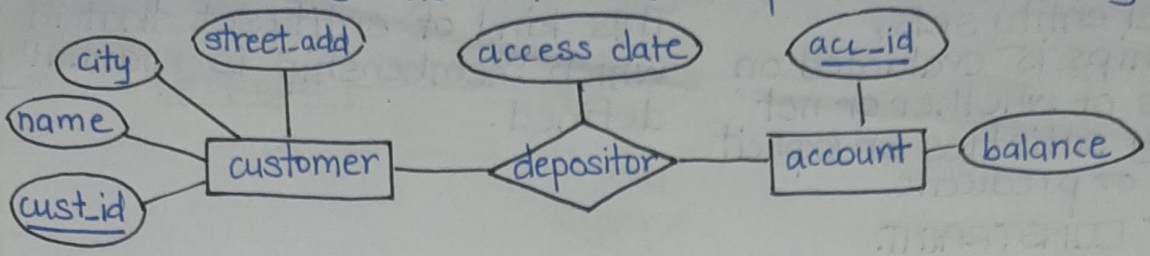
- Binary vs n-ary Relationship set.

Genrally relationship described in the database are binary relationships. Howeher, non-binary relationship can be represented by making several binary relationships we can specify some mapping cardinality on relationship with degree > 2.



PLACING RELATIONSHIP ATTRIBUTE

The relationship set can have descriptive attribute. For example in the following figure the relationship depositor has descriptive attributes named access_date the decision of placing the specific attribute as a relationship or entity attribute should passes the characteristics of the real world enterprise that is being modelled.



Extended ER Functions

SPECIALIZATION AND GENERALIZATION

Some entities have relationship that form hierarchy
In this relationship heirarchy, some entity acts as superclass and some other entity acts as subclass.

SUPER CLASS : A entity that represents a general concept at a high level
SUB CLASS : A entity that represents a specialized concept at a low level

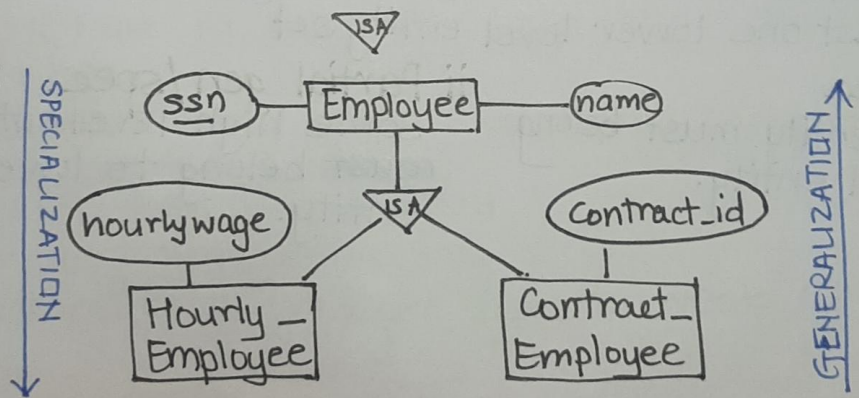
The subclass is said to inherit from superclass. When a subclass is said to inherit from multiple super class it inherits all their attributes. In addition it can have its own special attributes.

SPECIALIZATION : The process of making subclass from a general concept is called specialization.
It is a top down approach. The subgroups are identified within an entity set which has attributes not shared by other entities.

GENERALIZATION : The process of making super class from subclasses is called generalization.
It is a bottom up approach. In this multiple sets are synthesized into high level entities.

The symbols used for Generalization / Specialization is:

EXAMPLE



CONSTRAINTS ON SPECIALIZATION & GENERALIZATION

1. MEMBERSHIP CONSTRAINT

These are type of constraint ~~on~~ that involve determining which entities can be member of a given lower level entity.
There are two types of membership constraints

i Condition defined

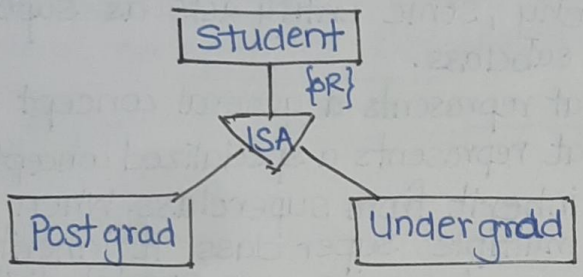
lower level entity sets, memberships is evaluated on the basis of whether or not an entity satisfies an explicit condition or predicate.

ii User defined.

This kind of entity set that in which membership is manually defined.

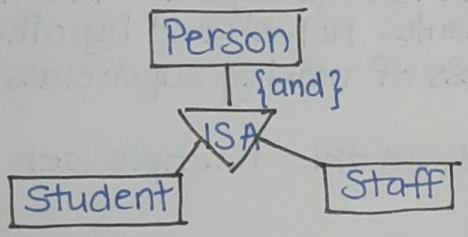
2. DISJOINT CONSTRAINT.

The disjoint constraint only applies when a superclass has more than one subclass. If the subclass are disjoint, then an entity occurance can be a member of only one of the subclasses.



3. OVERLAPPING

When some entity can be a member of more than one subclass



4 COMPLETENESS

It specifies whether or not an entity in the higher level set must belong to atleast one lower level entity set

i Total gen/spec

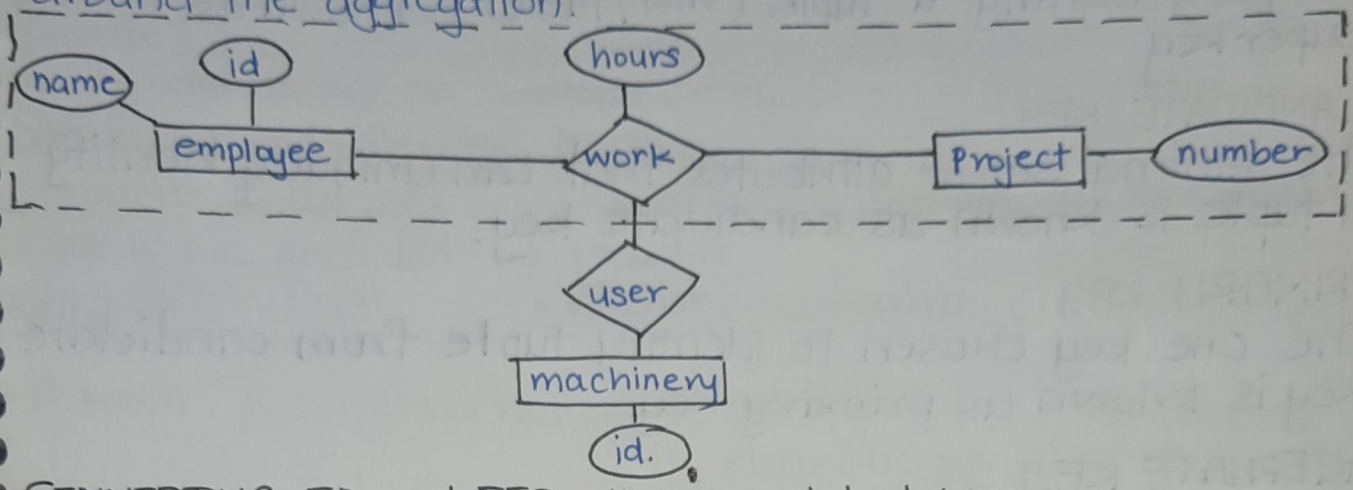
Each higher entity must belong to lower level entity.

ii Partial gen/spec.

Some high level entity may ~~must~~ belong to lower level entity.

AGGREGATION

This feature of the entity set/model allows a relationship set to participate in another relationship set. This is indicated on an ER diagram by drawing a dashed box around the aggregation.



CONVERTING ER and EER diagrams into tables

Mapping Entityset to Relationship.

An entity set is mapped to a relation in a straight forward way
Each attribute of entity set becomes an attribute of the table
The primary key attribute of entity set becomes an entity of the table

Mapping Relationship Sets (without constraint) to tables.

Create a table for the relationship set.
Add all primary keys of the participating entities as fields
Add a field for each attribute of relationship
Declare a primary key
Declare foreign key.

Mapping Relationship sets (with constraints) to tables.

relationship set involves n entity set and some m of them are linked via arrows in the ER diagram, The key for anyone of these m entity sets constitute a key for the relation
Hence we have m candidate keys and one of them is primary key

Mapping weak entity set to Relational Mapping.

Create table for weak set
Make each attribute a field.
Add primary and foreign key
The attributes of relation goes to weak entity

TYPES OF KEYS.

SUPER KEY

A set of attributes or combination of attributes that can uniquely identify a tuple (row/table) is known as super key

CANDIDATE KEY

The minimal set of attributes that can uniquely identify a tuple is known as candidate key.

PRIMARY KEY.

The one key chosen to identify tuple from candidate key is known as primary key.

ALTERNATE KEY

The candidate keys other than primary keys are called alternate key.

FOREIGN KEY

A reference to another table with the attribute of that table is called foreign key.

COMPOSITE KEY

Sometime one attribute is not strong/ unique enough to form a key thus multiple attributes combine to form a key this is called composite key.

UNIT 2

SQL & PL/SQL

Characteristics & Advantages.
Standard language for creating and manipulating Database
Very simple & easy to learn
allows user to create update delete or retrive data from database
used to create view stored procedures and functions
allows user to set permissions on table.

Data types and Literals

1) Numeric data types :
- Int, BigInt, Small Int
- Real, Double, Float
- Decimal, Numeric (n,m)

2) Character string datatype.
- Char(n)
- Varchar, long varchar

3) Large object
- Clob (for char)
- Blob (for bits)

4) Boolean : True False or Null

5) Date datatype
- date (yyyy-MM-DD)
- timestamp
- Interval

Classification of SQL comments.

DDL	DML	DCL	DQL	TCL
create	Insert	Grant	Select	Commit
alter	update	Revoke	(without any clauses, From, Into)	Rollback.
Drop	delete			

Create: used for creating databases, tables, views.
Create [Database | Table | View] [D.name | T.name | V.name];

INSERT: used for insertion operation into tables, views.
Insert Into tablename (col1 ...) values (...);

UPDATE: used for modifying the table [data]
Update tablename SET col1=val1... where condition;

ALTER: used for modifying the table [structure]
ALTER table table_name [ADD | DROP] column column_name;

DELETE: used for deleting one or more records.
Delete From table_name where condition;

DROP: used for deleting table, view, database with structure.
Drop [Database, Table, View] [D.name, T.name, V.name];

WHERE used for setting up condition on select statement

CLAUSES.

1. Order By.
   - used for arranging data in particular order. ASC or DESC.
   - Select col1...
     From table_name
     Order By col1... ASC|DESC.

2. Group By.
   - used for grouping rows that have same values
   - optionally used with aggregate fxn
   - Select column....
     From table_name
     where condition
     Group By column_name(s)

3. Having
   - Filters records summarized by GroupBy result.
   - Having works on groups while where works on individual rows
   - Select col From tablename where cond GroupBy col Having cond;

LOGICAL OPERATORS.

AND.
only when all cond are True.
where cond1 AND cond2;

OR
only when atleast 1 cond is true.
where cond1 OR cond2;

NOT
only when reverse is true
where NOT cond;

STRING OPERATIONS (=, <, >, <=, >=, <>)

LIKE

(%) matches zero, one or multiple

(_) matches single char.

Example:

EmpName
Sunil
Supria
Sonia
Suraj

∴ Select EmpName from Emp where EmpName Like 'S%a'

Output: Supria, Sonia.
(Begins with S ends with a)

∴ Select EmpName from Emp where EmpName Like 'S___'

Output: Sunil, Sonia, Suraj
(Begins with S followed by 4 letters)

BETWEEN used for specifing a range for the condition.
Select col From Tablename where col Between val & val;
BUILT IN FXN: ABS: returns absolute non negative value.
Select ABS(-9) Result
output

Result
9

ABS(m) Tan(n)
MOD(m,n) SQRT(n)
POWER(m,n) EXP(n)
ROUND(m,n) LOG(n2,n1) n2=base, n1=val.
TRUNC(m,n) CEIL(n) smallest int ≥ = n.
SIN (n) FLOOR(n) gretest int < or = n.
COS (n) SIGN(n) -1 if n<0, 0 if n=0, 1 if n>0.

EXIST to check if reqord exist returns bool.
NOT EXIST to check if record doesnt exist returns bool
UNIQUE to check if record is unique returns bool.

PRIMARY key: Create table (col1, col2 ... primarykey(col));
FOREIGN key: Create table (col1, col2 ... Foreign key(col) references tablename(col));
Composite key: Create table tablename (col1, col2 ...
constraint name Primarykey (col1, col2));

CONSTRAINTS.
UNIQUE To check if data inserted isunique.
NOT NULL To check if null values are inserted (doesnt allow Null)
CHECK TO impose value constraints example (cost < 100) etc.
IN allows to specify multiple values in where clause.

SET OPERATIONS.

UNION Performs union of 2 queries
INTERSECT Performs intersection of 2 queries
EXCEPT Performs difference of 2 queries.

(Select query) [union | Intersect | EXCEPT] (select query)

JOIN OPERATION.

Inner Join Select t1.col ... From table1 as t1 Innerjoin table2 as t2
on t1.colcomm = t2.colcom ;
common columns.

RIGHT/LEFT Join: Select t1.col... From table1 as t1 [RIGHT | LEFT]
Join table2 as t2 On t1.colcom = t2.colcom ;

FULL JOIN. Select t1.col ... from table_name Full join table2
ON t1.colcom = t2.colcom;

VIEWS

Rules for updating.
should be based on only 1 table
Select should not have distinct.
Shouldnt have Not null values.
Shouldnt be created from complex query
Select should not include clauses.
Should not have Aggregate fxn fields.

Aggregate Functions:

Sum( )
Count( )
Min( )
Max( )

STORED PROCEDURE

A group of statements which can be called by its name.

```
[create|Replace] Procedure procedure_name
(Parameter_name [IN, OUT, INOUT] Type ...)
[IS|AS]
BEGIN.
<Body>
END;
```

Called by.

```
EXEC procedure_name(parameters);
```

FUNCTION.

It take one or more values and returns one value.

```
[create|Replace] Function Function_name
 [Parameters]
Return Datatype
[IS|AS]
[Declaration section]
BEGIN
<body>
END function_name;
```

Called by.

```
var = Function_name (Parameters);
```

CURSOR

- Any SQL query is processed in a context area
- A cursor is a pointer to this context area.
- All info about query ~~porces~~ processing is controlled by cursor

☐ Implicit cursor:

Default cursor created when Select update delete operation executed.

- %ROWCOUNT returns no. of rows affected.
- %NOTFOUND T/F if rows not affected → T
- %FOUND T/F if rows affected → T
- %ISOPEN T/F if cursor open → T

☐ Explicit Cursor:

```
Cursor cursor_name Is (select query);
Open cursor_name;
FETCH cursor_name into var;
close cursor_name;
```

TRIGGERS.
invoked with some event occurs.
Events involved are create, Alter, Drop (DDL)
Update, Insert, Delete (DML)

Create | Replace trigger trigger_name
[Before | After] [event] of col on Table_name
[Referencing old as O New as N]
For EACH ROW
When cond
DECLARE
<declaration>
BEGIN
<execution>
END.

Types. : 7 and more.

1) Before Insert
2) Before Update
3) Before Delete.
4) After Insert
5) After update
6) After delete.
7) Instead of, (for views)

ASSERTION.
These are set predicates database must always follow.
CREATE ASSERTION <assertion_name> CHECK <predicate>

ROLES & PRIVILEGES
set of priviledges that can be granted to users.
Create role <name>
Identified By <pswd>
Grant create table .... To <name>
Revoke create table From <name>

System priviledge
Create, Alter, Drop

Object priviledge.
Select, Insert, Delete