UNIT-1

Database :- A DBMS is a collection of interrelated data and a set of programs to access those data.

→ The collection of data usually referred to as the data base, contains information relevant to an enterprise.

Goals

→ The primary goal of a database MS is to provide way to store and retrieve database information that is both convinient & efficient.

→ Database systems are designed to managed large bodies of information.

Database system Application.

1) **Banking :-** for customer info, accounts loans and banking transactions.

2) **Airlines :-** for reservation and schedule info

3) **universities :-** for student info, registration results etc.

Credit card Transmission :- You purchase credit card & generating of monthly statements.

Telecommunication :- Record of calls, monthly bills.

Sales :- You customers product & purchase info.

DBMS Vs File system

File processing system is supported by a conventional operating system. The system stores permanent records in various files & it needs diff application program to enter record & add record to the appropriate files.

→ Keeping organizational information in a file processing system has a no. of major disadvantages.

→ Data redundancy (duplicacy) & inconsistency.

→ Difficulty in accessing data.

→ Integrity problems.

→ Atomicity (uniqueness) problem

→ Security problem

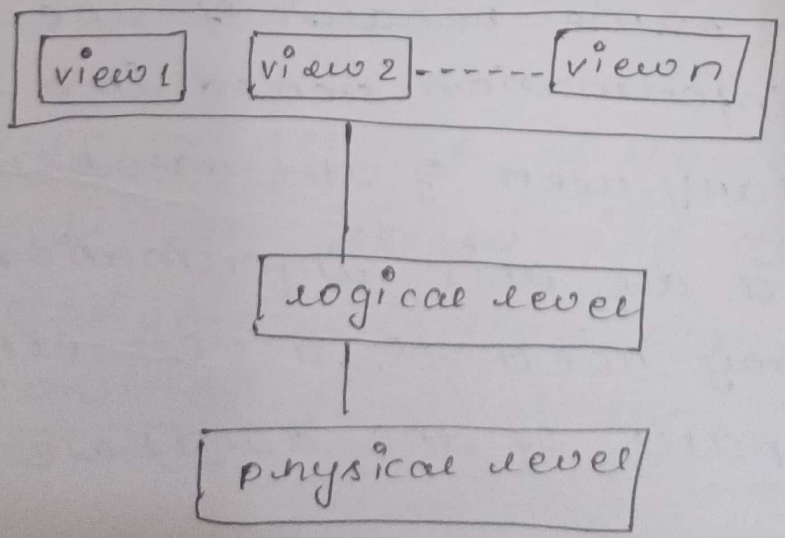
→ A major purpose of database is to provide users with an abstract view of the data that is system hides certain details of how the data are stored and maintained

→ Data abstraction

★ To retrieve data efficiently.

★ The need for efficiency have led designers to use complex data structure to represent data in the data base.

★ Many database users are not computer learners, developers hide the complexity from users through several leo levels of abstraction, to simply users, interaction with the system



Three levels of data abstraction

Physical level :- The lowest level of abstraction describes how the data

are actually stored. The physical level describes complex low level data structure in detail.

Logical level :- The next higher level of abstraction describes what we store in data base, and what relationship exists among those data the logical level thus describe the entire data base in terms of a small no. of relatively simple structure

View level :- The highest level of abstraction describes only part of the entire database. Even though the logical level uses simple structure, complexity remains same because of the variety of information stored in a large database. Many user of the databases do not need all this information, instead they need to access only the part of the database.

→ Instances & schemas

↓

means value at particular time

→ Database changes overtime as information is inserted & deleted. The collection of information stored in the database at the particular moment is called instance of the database

→ The overall design of the database is called the database schemas (schemas are change infrequently)

e.g, student

Name	Roll No.	class	Major

course.

course name	course No.	Credit	department

Data model : "A collection conceptual tools for describing data, data relationships, data semantics and consistency constraints

A data model provides a way to describe e.g. the design of the data base, at a physical, logical & view level".

Data models are classified into four categories.

Relational model uses a collection of tables to represent both data and relationship among those data. each table has multiple rows, columns and each column have unique name.

Entity Relationship model (ER model)

ER model is based on the perception of a real world that consists of a collection of basic object called entities and relationship among those objects. An entity is a "thing" or "object" in the real world that is distinguished among those objects.

Object base model :- Is extending the ER model which notions of encapsulation, methods and object identities.

the object relational datamodel combines features of object oriented data model (OM) and relational OM

Semi structure data model :- Permits the specification of data where

individual data items of the same type may have different set of attributes.

Extensible markup language (XML) is widely used to represent this data model

Network and hierarchical data model preceded the RDM (relational data model)

Data base languages

1. Data manipulation language (DML)

DML is a language that enables users to access or manipulate as organised by the appropriate data model. The type of access are

- Retrival of information stored in the database
- Insertion of new information into the data base
- Deletion of information from the data base
- Modification of information stored in the data base

There are basically two types

⇒ Procedural DML's require a user to specify what data are needed and how to get those data.

⇒ Declarative DML's (also refer to as non procedural DML's) requires a user to specify what data are need without specifying how to get those data.

2. Data definition language (DDL)

⇒ To specify a database schema
⇒ These specify the storage structure and access methods used by the data bases by the a set of statements in the special type of DDL called data storage and definition language.

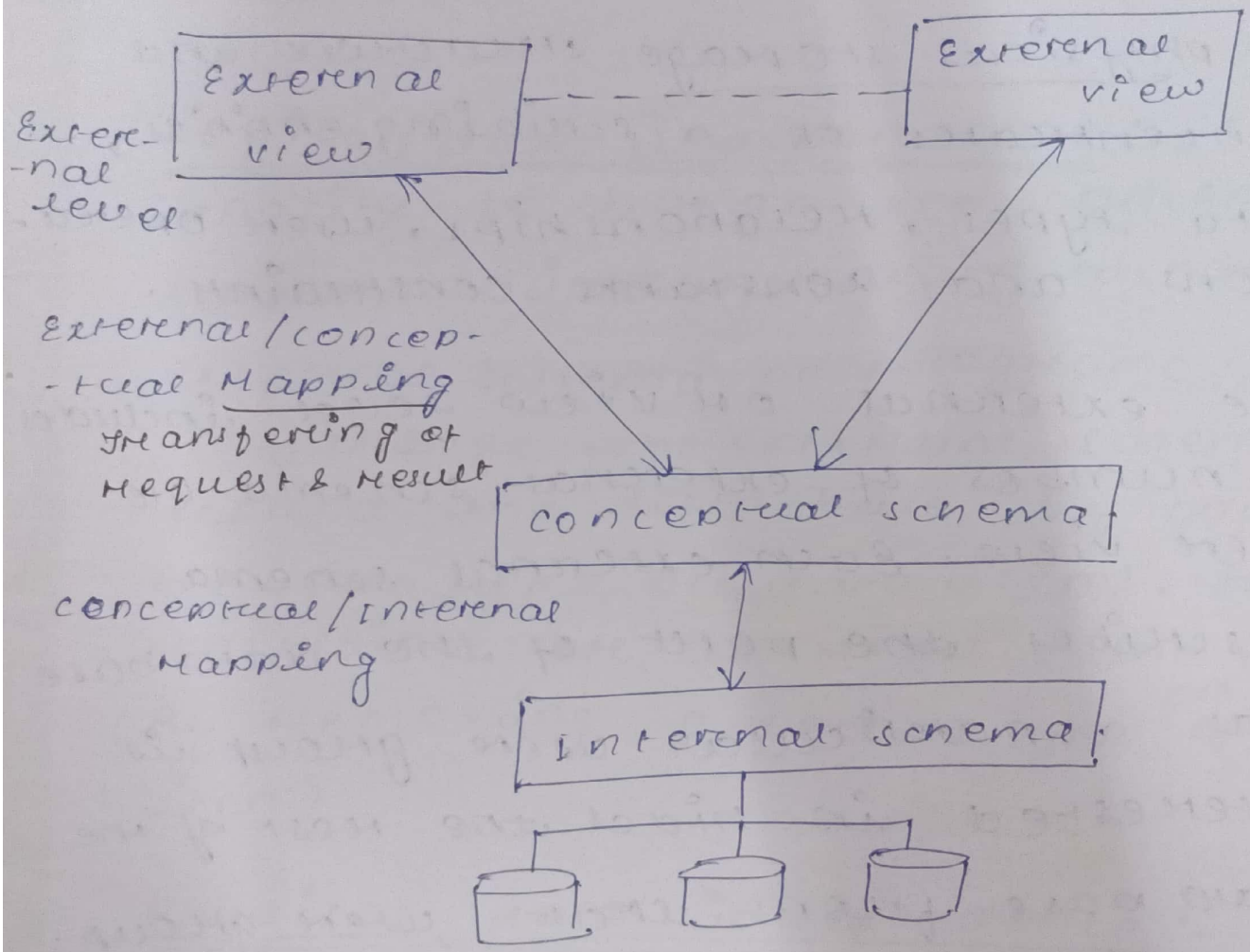
Data base approach

There are four important characteristic of data base approach.

⇒ self describing nature of a data base system
⇒ Insulation of program and data.
⇒ Support of multiple views
⇒ use of a catalog to store data base

Three schema architecture and data independency

Three schema architecture and data independency.



The internal level has an internal schema which describe the physical storage structure of the database. The internal schema used a physical data model and describe the complete details of the data storage and access path for the database.

⇒ The conceptual level has a conceptual schema which describe the structure of the whole data base for a community community of users. The conceptual schema hides the details of physical storage structure and concentrates on describing entities, data types, relationships, user operations and ~~constraints~~ constraints.

⇒ The external or view level includes a number of external schema or user view. Each external schema describes the part of the data base that a particular user group is interested in hides the rest of the data base from that user group.

⇒ The process of transferring request and result is called mapping.

Data Inaependence

The three schema architecture can be used to further explain the concept of data independence, which can be defined as the

capacity to change the schema at one level of the data base without having to change the scheme the next higher level. We can define the two types of the data indepen -dence.

Logical data independence :- It the capacity to change the conceptual schema without having to change external schema or programe. We may change the external schema to expand the data base (by adding a record type or data item), to change constraints or to reduce the database (By removing a record type or data item)

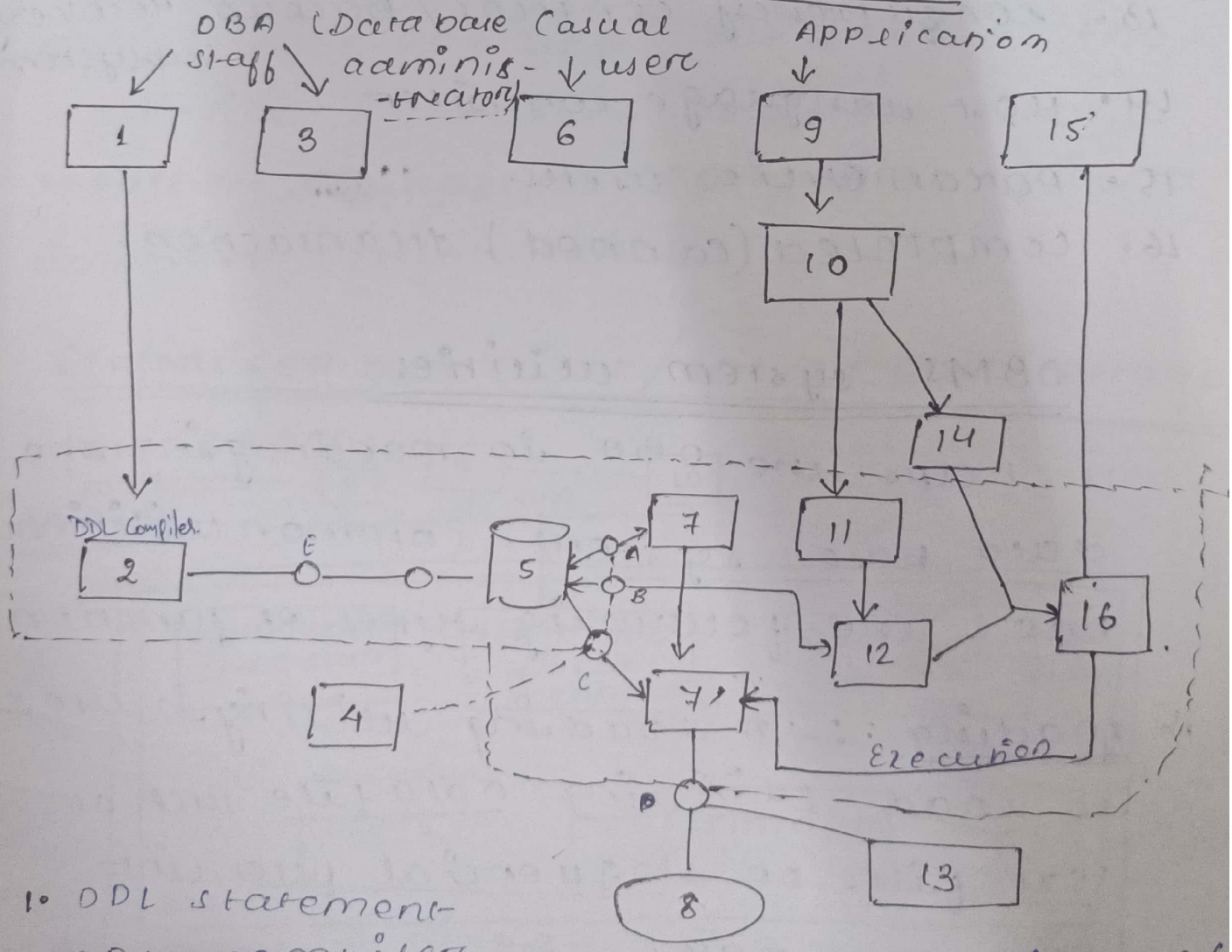
Physical data independence :- Is the capacity to change the internal schema without having to change the conceptual schema. Hence the external schema need not to be change because some physical files has to be recognised. For e.g, By creating additional excess structure to improve the performance of

retrieval and update. If the data as before remains in the data, we should not have to change the conceptual schema.

DBMS interface The user friendly interfaces provided by DBMS may include the following

- Menu based interfaces for web client or browsing. (for requesting something)
- Forms based interfaces → To display a form to each other.
  Fill the form so that DBMS will retrieve matching data
- Graphical user interfaces (GUI)
  Typically displays a schema to the user in diagrammatic form.
  The user can then specify a query by manipulating a diagram. GUI's utilize both menu and forms

- Natural language interface :- It has its own "schema" which is similar to the database conceptual schema. as well as dictionary of important words.
- Interface for parametric user :- such as bank tellers, often have a small set of operations that they must perform repeatedly
- Interfaces for a DBA → Privileged Commands (like creating accounts -)
  Database system environment-



1. DDL statement
2. DDL compiler
3. Preixiledge comands
4. Stored data manger
5. System catalog/ data dironary

6. Interactive query
7. Query compiler
7'- Runtime database processor
8 - stored database.
9. Application programmer.
10 the compiler
11. DML statements.
12. DML compiler
13. concurrency control / backup recovery subsystem.
14. Host language compiler.
15. Parametric users.
16. compiled (canned) transaction.

DBMS system utilities

It helps the DBA in managing the data base system. common utilities have the following types of function.

1) Loading :- A loading utility is used to load existing data file such as text files or sequencial files into the data base.

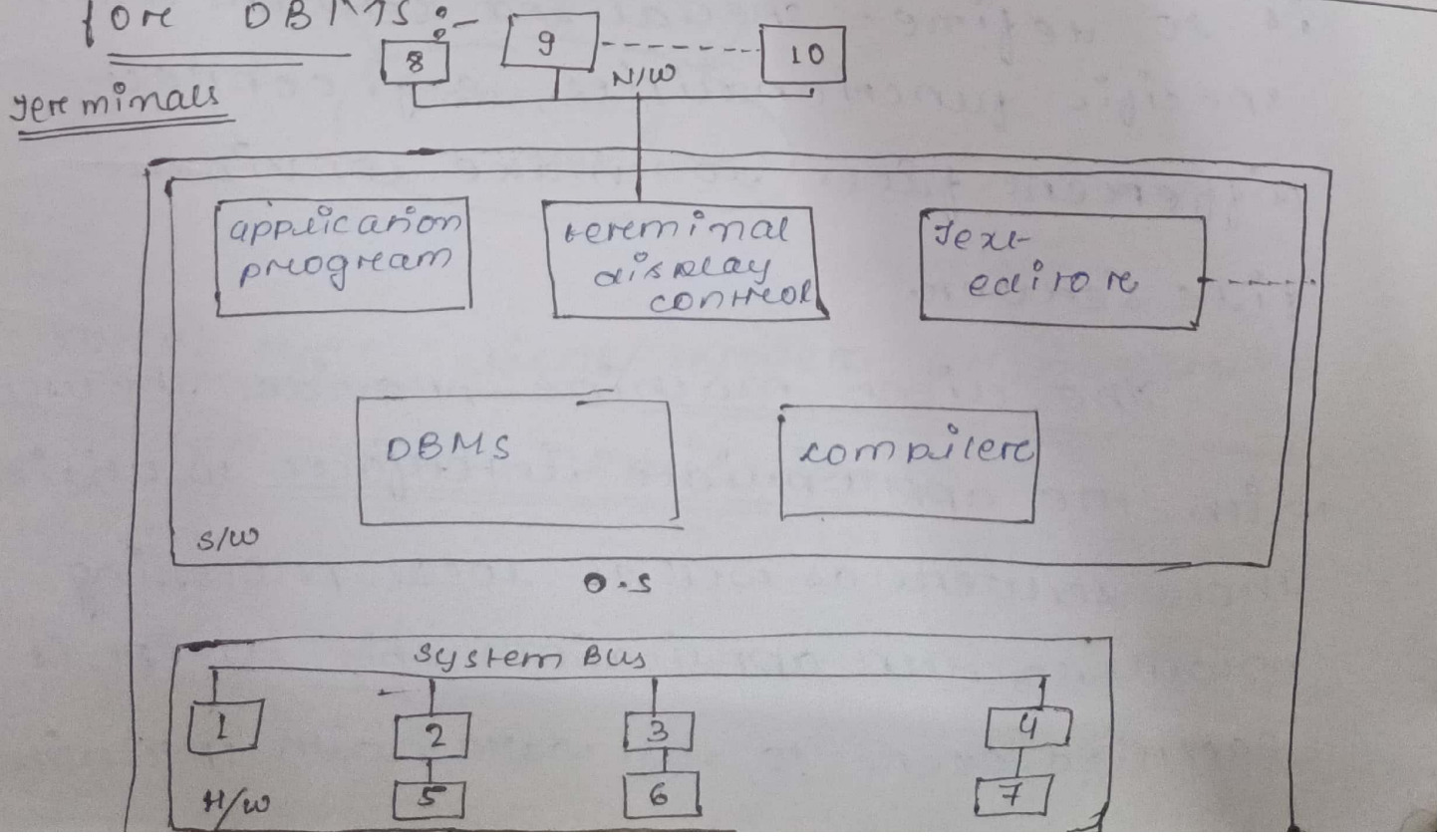
2) Back up :- A backup utility creates a backup copy of the database

usually by dumping the entire database onto tape.

3) File reorganisation :- This utility can be used to reorganise the database file into a different file organisation to improve performance

4) performance monitoring :- such a utility monitor database usage and provides statistic to the DBA. The DBA uses the statistics in making decisions such as weather or not to reorganise files to improve performance.

Centralised and client/server architecture for DBMS :-



1. CPU

2, 3, 4 - controller

5 - Memory

6 - Disk

7 - I/P O/P devices (printers, tape drive

8 - }
9 - } - Display monitor.
10 - }

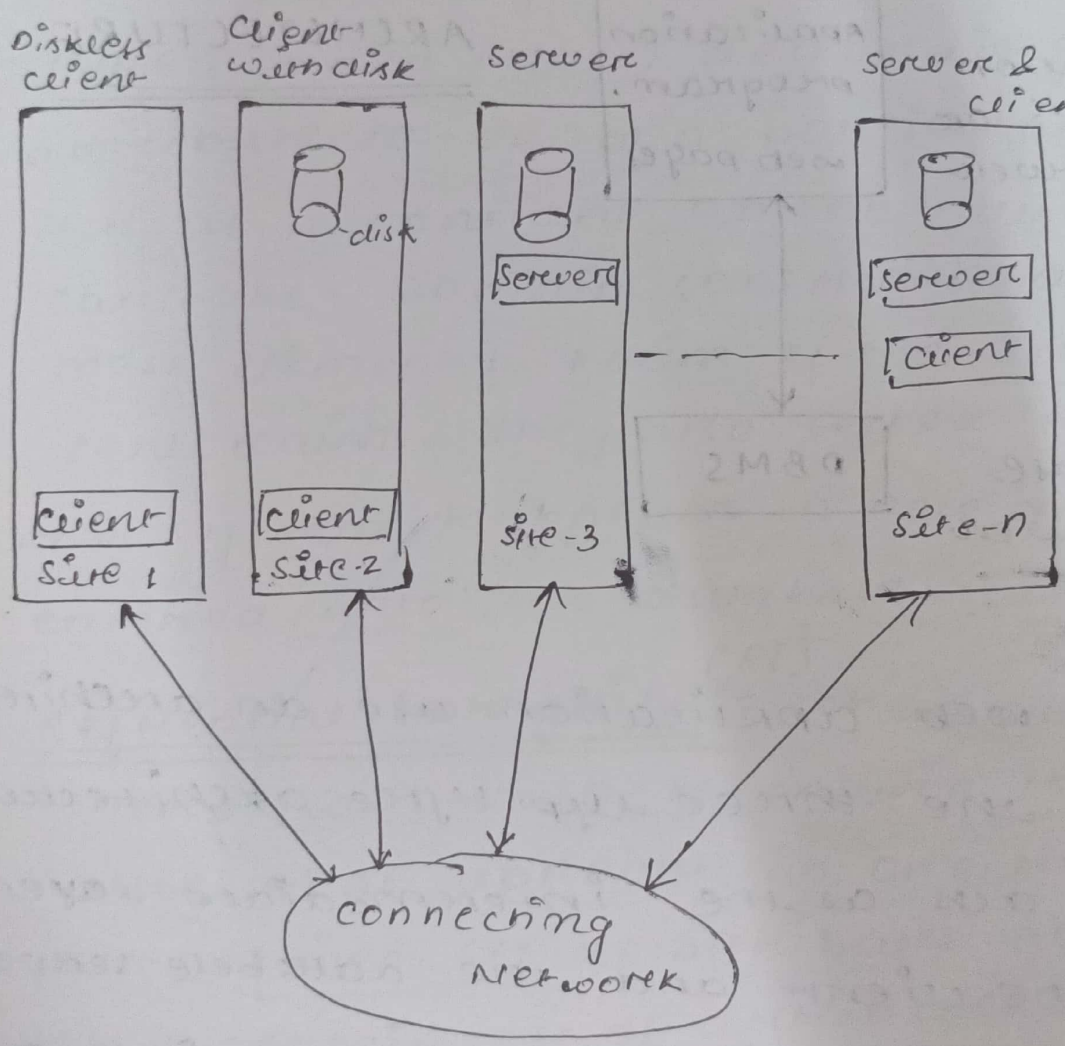
Basics client/server architecture :-

client/server architecture was developed to deal with computing environment in which a large no. of PC's, workstations, file servers, printers, data base servers, web servers and other equipments are connected via a network. The idea is to define specialized servers with specific functionalities. e.g, connect different files to make convinent file servers.

The client machine provides the user with the appropriate interfaces to utilize these servers as well as local processing power to run applications this concept is carried over to software, with specialize

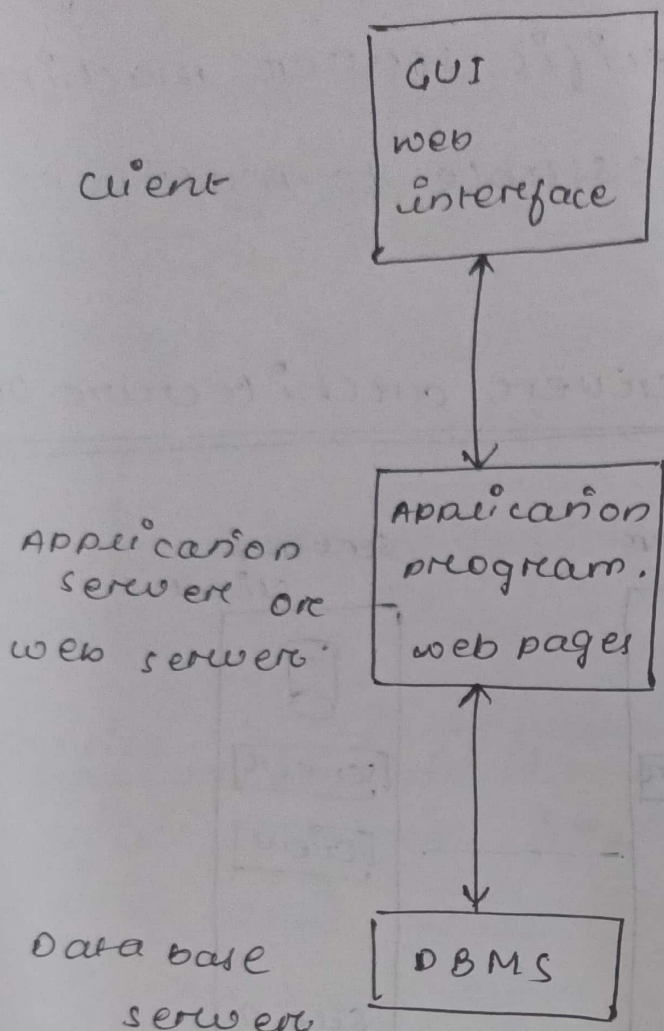
software such as DBMS, CAD package being stored on specific server machines and being made accessable to multiple clients

Two type client/server architecture DBMS's



Three type client/server architecture for web application :-

P.T.O.



LOGICAL THREE TYRE CLIENT SERVER ARCHETECTURE

Many web application use an architecture called the three tyre architecture which acts as the intermediate layer b/w the client and the database server the intermediate layer or middle tyre is sometimes called the application server or web server.

Data modelling using the entity relationship model

① Data base language The data base language are DML and DDL

② Domain constraints :- constraints :- A domain of possible values must be associated with every attribute (for e.g., integar type, character type, data types). declaraing an attribute to be of a particular domain acts as a constant on the values that it can take. Domain constraints are the most elementy form of the integrity constraint. They are tested easily by the system whenever a new data item is entered into the database

③ Refrential Integrity (RI) :- These are cases where we wish to ensure that a value that appears in one relation for a given set of attributes also appears for a certain set of attributes in another relations (refrential integrity). DataBase modification can cause voxib violation of refrential integrity. when a RI constraints are violated, the normal procedure is to reject the action that cause cost the violation

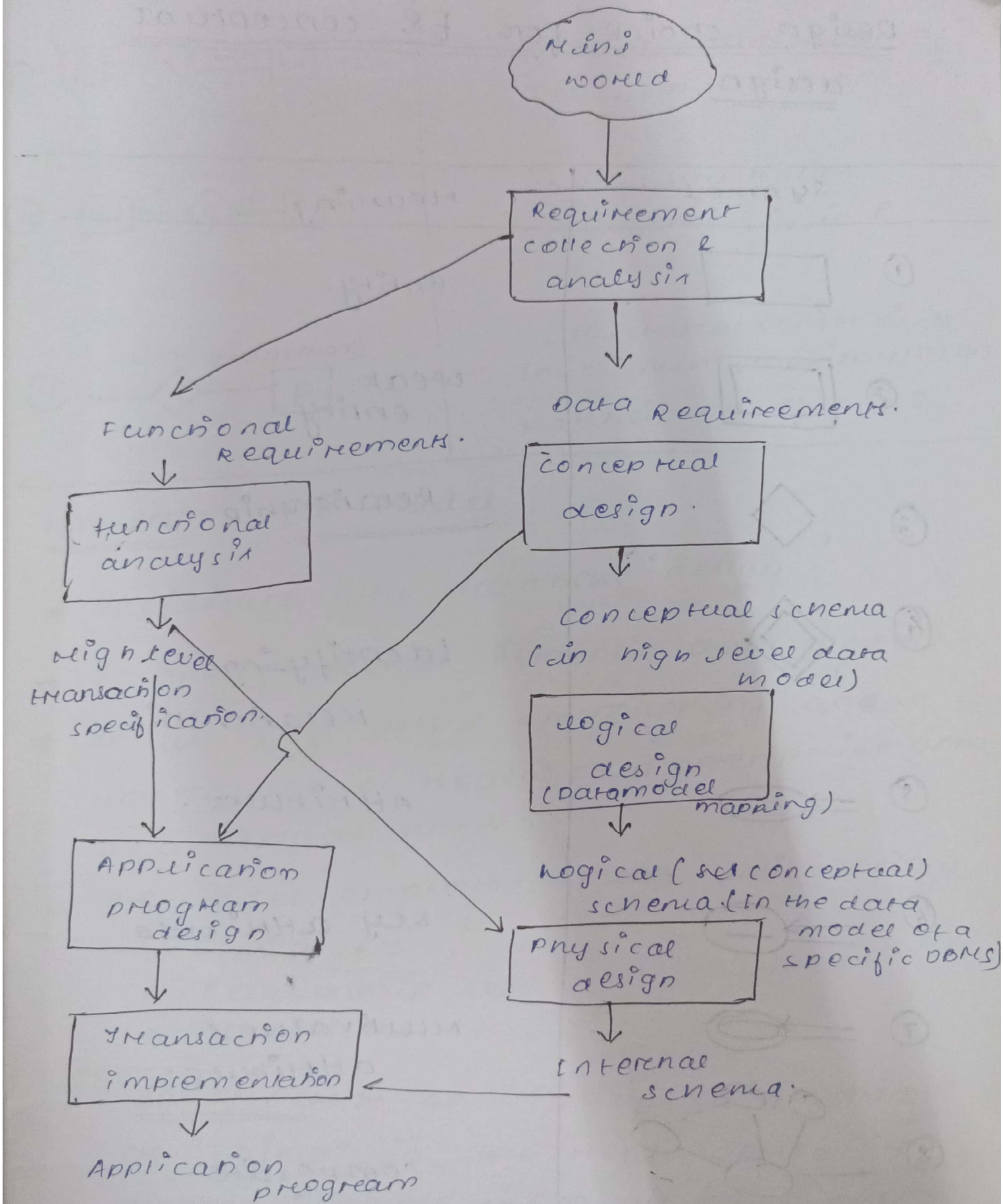
④ Assertion :- An assertion is any condition that the data base must always satisfy. Domain constraints and RI are special forms of assertion. Example :- Minimum balance of ₹1000 in bank accounts.

When an assertion is created, the system test it for validity. If the assertion is valid then any future modification to the database is allowed only if it does not cause that assertion to be violated.

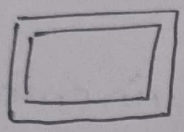
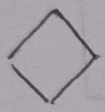
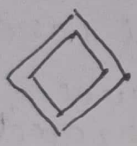
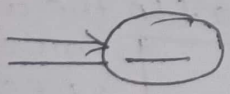
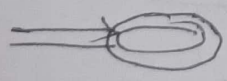
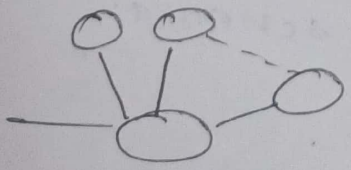
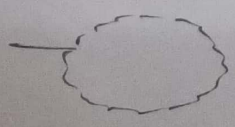
⑤ Authorisation :- Read Authorisation
⇒ Insert Authorisation
⇒ Update Authorisation
⇒ Delete Authorisation

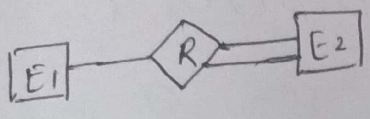
⑥ Phases of Database design

P.T.O.

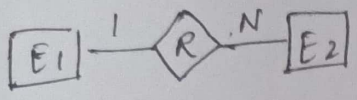

Mini world
Requirement collection & analysis
Functional Requirements.
Data Requirements.
functional analysis
Conceptual design.
High level transaction specification.
Conceptual schema (in high level data model)
logical design (Data model mapping)
Application program design
logical (set conceptual) schema (in the data model of a specific DBMS)
Physical design
Transaction implementation
Internal schema.
Application program

Design choice for ER conceptual design

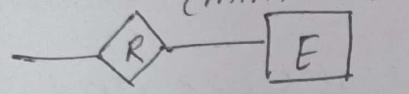
symbol	Meaning
① 	Entity
② 	weak entity
③ 	Relationship
④ 	identifying relationship
⑤ 	Attributes.
⑥ 	key attribute
⑦ 	Multivalued attribute.
⑧ 	composite attribute.
⑨ 	Derived attribute

(10) 

Total participation of E_1 : E_2 in a relation R.

(11) 

Cardinality ratio.
1 : N for E_1 : E_2 in R.

(12) 

Structural constraints (minimum, maximum) on participants of E in R.

13.2.2014 **ER Model**

⇒ Facilitate the database design

⇒ Is one of several semantic data models.

⇒ It is very useful in mapping and interaction of real world enterprise onto a conceptual schema.

⇒ ER model employees three basic notions
- Entity sets
- Relationship sets.
- Attributes

Entity sets :- An "Entity" is a "thing" or "object" in the real world i.e, distinguishable from all other objects For e.g, Each person is a enterprise in an entity.

⇒ An entity has a set of properties, and the value for some sets of properties may uniquely identify an entity

Example ⇒ Person_id

⇒ Entity set is a set of all entities of the same types that shows the same properties or attributes.

Example → Bank ~~and~~ customer

⇒ Each entity has a value for each of its attributes

⇒ An entity set do not need to be disjoint. Example → It is possible to define the set of all employees of a bank and customer. A person entity may be an employee entity, a customer entity, both or neither.

Relationship set :-

⇒ A relationship is an association among several entities

E.g

id	Name	Address
123	XYZ	—
323	ABC	—

Loan No.	Amount
729	1000000
323	20000

Attributes

⇒ For each attribute there is a set of permitted values, called the domain or value set of that attribute
For example → customer name might be a set of all text strings of certain length

⇒ An entity may have several attributes, each entity can be describe by a set of (attribute, data values) pairs, one pair for each attribute of the entity set
For example → customer can describe a set {(customer_id, 12345), (customer_name, xyz), (customer_address, abc), (customer_city, Delhi)} meaning that, ~~[illegible]~~ 'xyz' whose customer id is 12345 and who resides at abc in city Delhi.

Attributes type :-

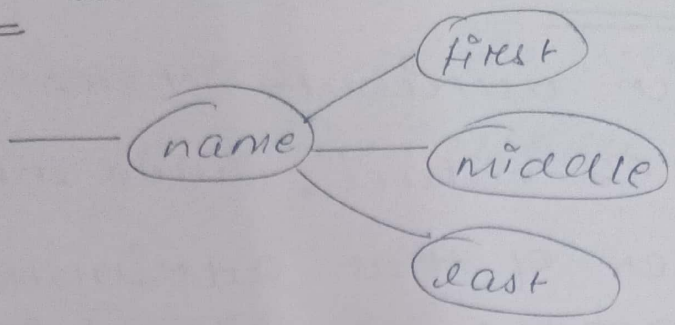
- **Simple and composite attribute :-**
  Simple :- The attribute have been simple i.e, they have not been divided into sub parts. For eg,

  (id)

  On the other hand attributes can be divided into sub parts called

composite attribute.

FOR. eg,



- single and multi valued attributes

An entity have a single value

FOR e.g, loan_no. have specific loan entity refers to only one loan number.

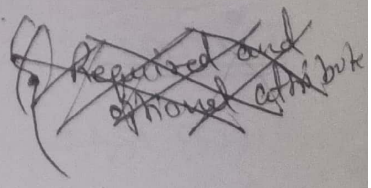
Multivalued entity FOR. eg, A customer had many oto many phone_no.

- Derived attributes and stored

Attribute can be derived from the values of other related attributes or entities. E.g, loan

- Key and Non key attribute
- Complex attribute → composite + multivalued

representation



CONSTRAINTS

1) Mapping constraints
2) Key constraints
3) Participation constraints.

Mapping constraints

Mapping cardialities express the no. of entries to which another entity can be associated by a relationship set.

⇒ Mapping cardialities are most useful in describing relationship set i.e, one to one, one to many, many to one, many to many.

KEYS

⇒ The integrity of the information stored in a database is controlled by keys

⇒ A key or key feild is a colourn value in a table that is use to either uniquely identify a row of data in a table or establish a relationship with another table

Customer

acc no.	name	add

Loan

loan no.	amount

A key is normally correlated with one column in table, although it might be associated with multiple columns.

⇒ Key is used to store the data. It also

Referred as some key, index or keyword.

Key can be of different types:-

1) **Super key**:- A super key is a set of one or more attribute that taking collectively, allow us to identify uniquely an entity in the entity set.

e.g, customer_id or customer_name and customer_id.

Customer_name alone can't be a super key but customer_id and customer_name, customer_id can be a super key.

2) **Candidate key**:- A combination of one or more fields whose values uniquely identifies a record in a table i.e, not no two records in a table has a unique code.

Suppose the combination of customer-name and customer_street is sufficient to distinguish among members of the customer entity set. Then both

{customer_id} & {customer_name, customer_street} are candidate key.

Although the attribute customer_id and customer_name together can distinguish customer entries. But their combination does not form a candidate key, since the attribute customer_id alone is a identifying key.

Primary key :- It denotes a key i.e, choosen by a data base designer as the principal means of identifying unique record ~~in~~ in a table.

The primary key should be choosen in such a way that its value must not (or rarely) change

For example → Roll_no., account_no. are primary key.

Composite key :- composite key is a key compose of more than one column. Sometime it also known as concatenated or structured key.

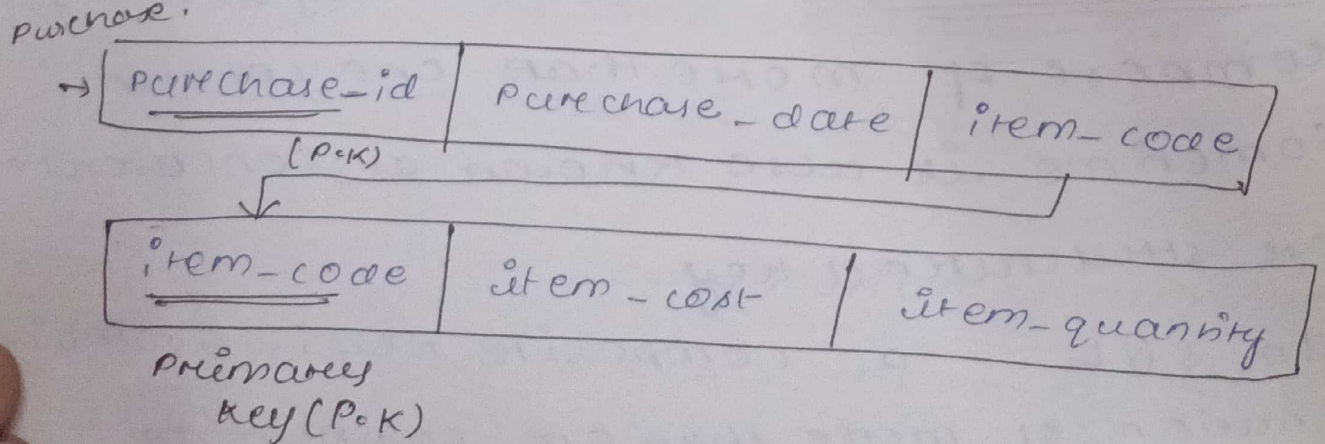
In SQL a composite key is a combination of more than one column to identify a unique row in a table.

Example :-

Invoice	product id	Quantity	Discount
153	21	2000	5%
153	16	3000	2%
102	21	2000	30%
106	32	1000	20%
124	11	5000	10%
132	16	6000	11%

Invoice table each row is uniquely identified. Here invoice no. and product id forms a composite key.

Foreign key :- A Foreign key is a combination of one or more column in a table (parent table) and reference at a primary key in another table (child table)



unique key:- It may be one or more than one

⇒ It cannot be a candidate key.

⇒ It can be null while primary key never be null.

Alternate key:- A candidate key(s) other than one choosen as the primary key is known as alternate key.

secondary key:- is a key use to speed up the searches and retrival. contrary to primary key, a secondary key does not necessary contains unique values.

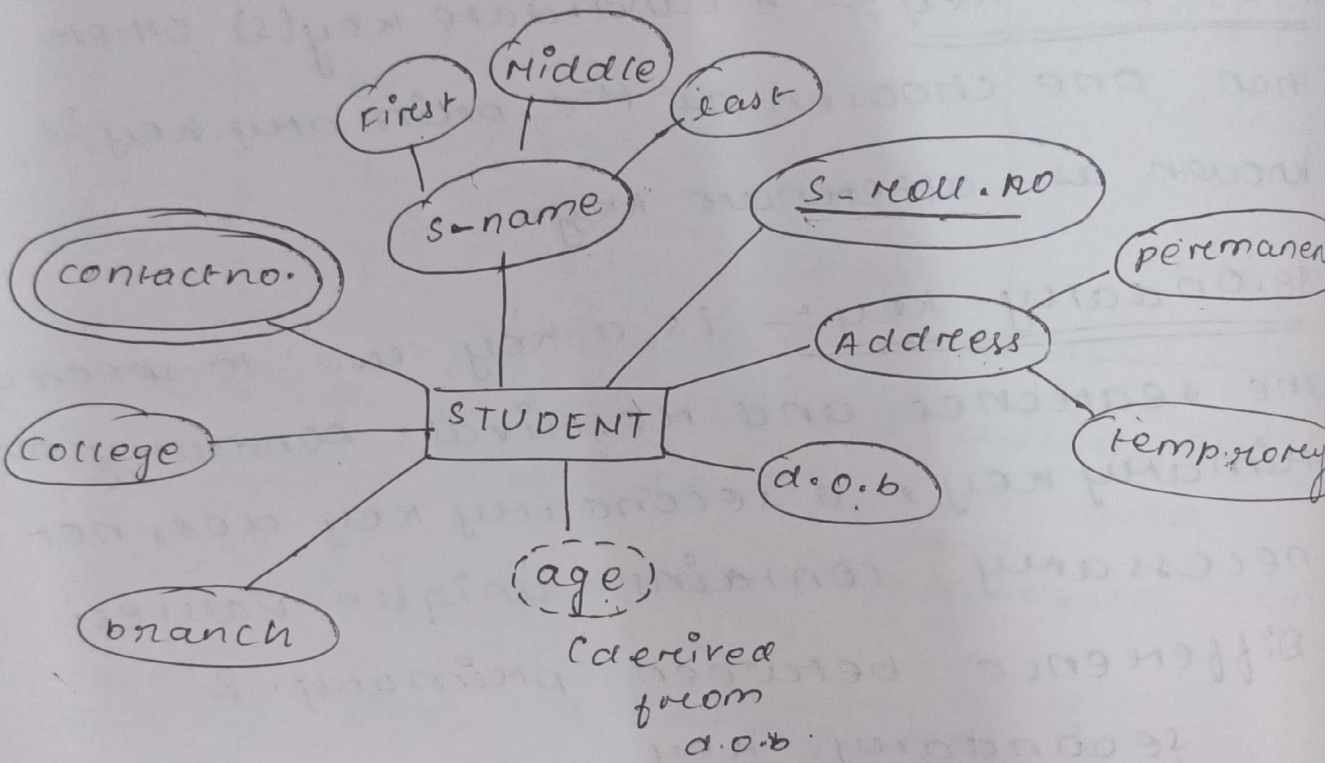
Difference between primary & secondary key

Primary	secondary
1) It is used for unique identification of rows	1) It is used for identification of rows but not usually unique
2) We have only one primary key per table	2) We have multiple secondary keys per table
3) A primary key is one of the candidate key	3) Attributes used for secondary key are not the one's used for super key i.e,

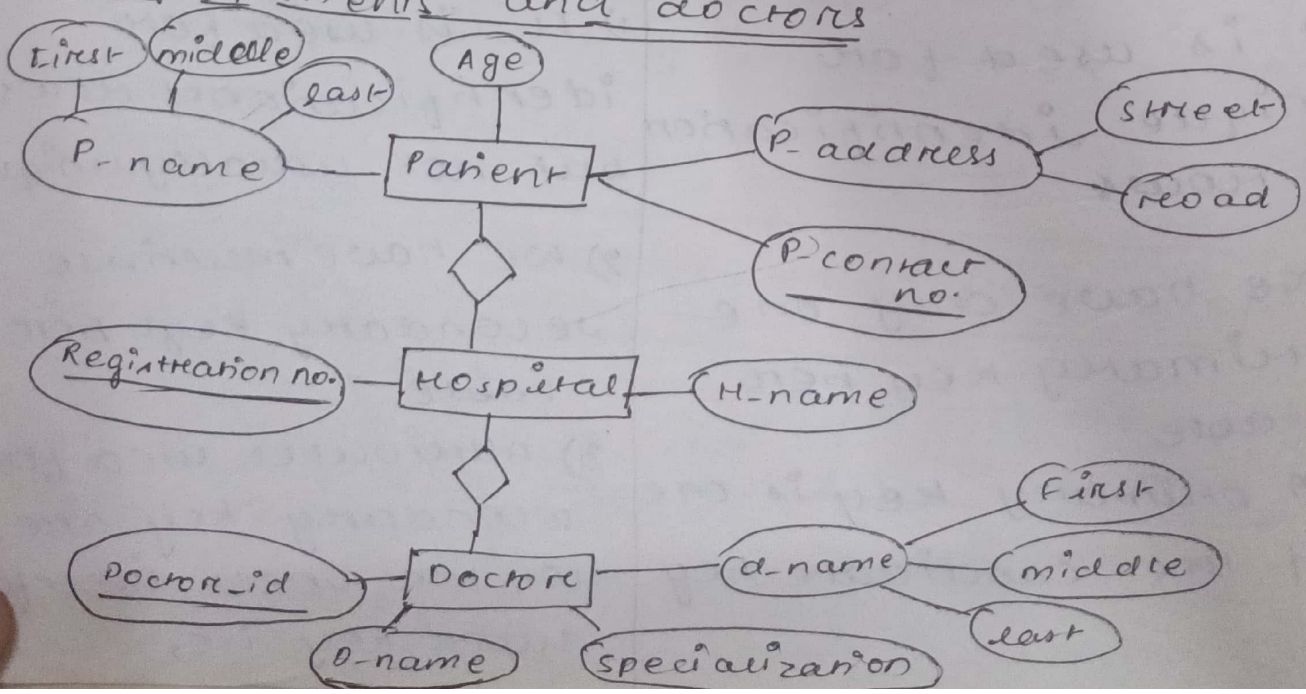
secondary key is not even be one of the super key.

Example of ER diagram.

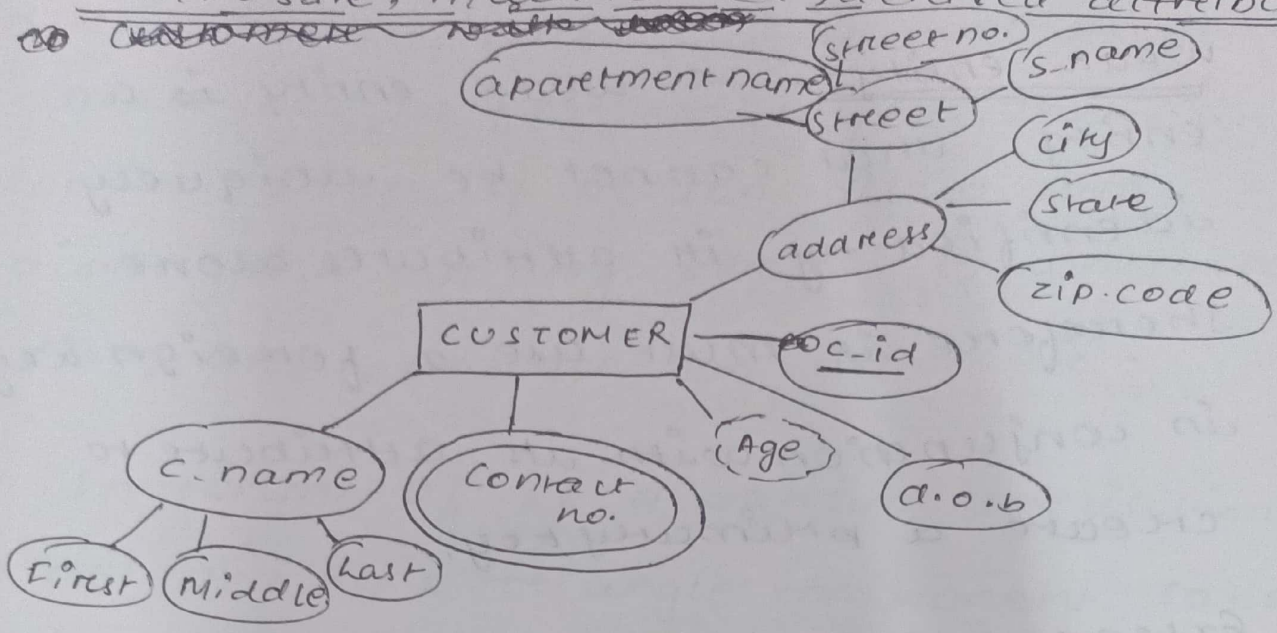
ER diagram for student.



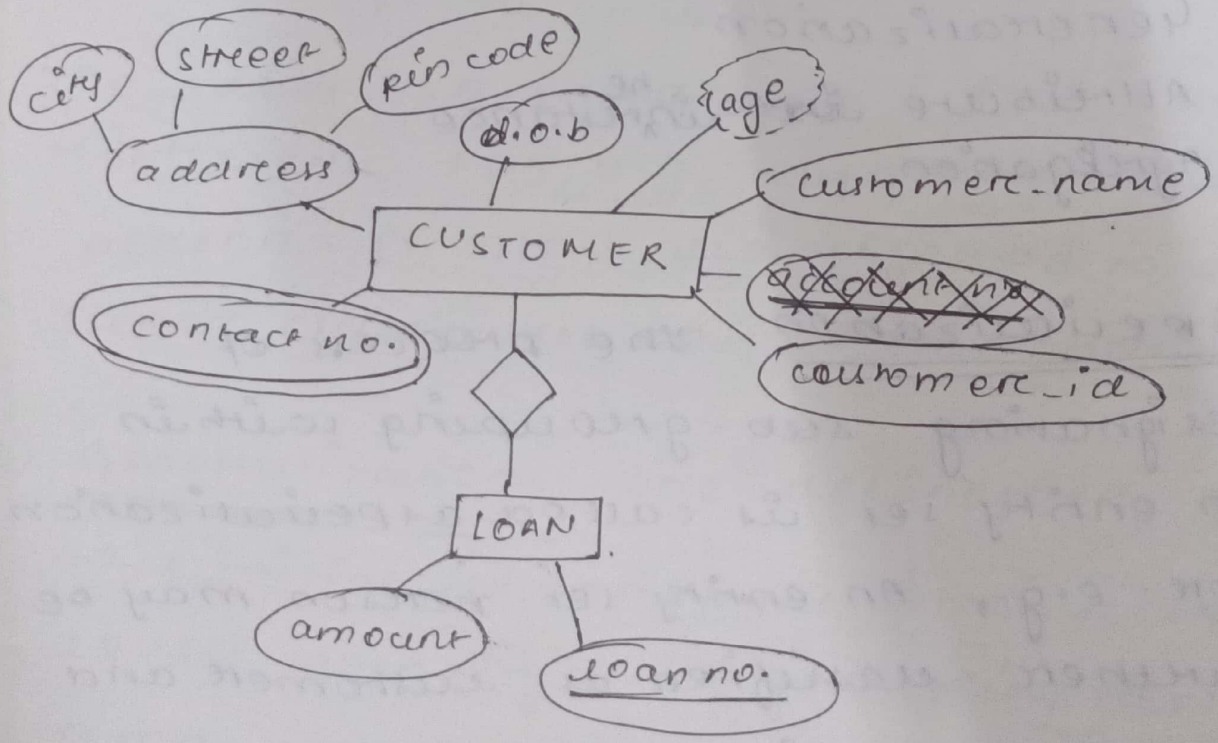
ER diagram for hospital with a set of patients and doctors



composite, multivalued, derived attributes



Customer with loan



Weak entity:- A weak entity is a entity that cannot be uniquely identified by its attributes alone. Therefore it must use a foreign key in conjunction with its attribute to create a primary key.

Extended E-R model (EER model)

There are basically four types of EER model

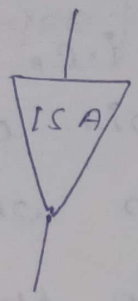
1) Specialization
2) Generalization
3) Attribute inheritance.
4) Aggregation

1) **Specialization** The process of designating sub-grouping within an entity set is called a specialization For e.g, An entity set person may be further classified as customer and employee having some attributes.

- Each of these person type is described by a set of attributes that include all the attributes, For e.g - customer entities may be described by the

attribute customer_id as well as employee employee_id for employee. The specialization of person allow us to distinguish among person to whether they are employee or customer.

In terms of ER diagram specialization is denoted by a triangle component labeled ISA



The label ISA stands for "Is A" and represent for e.g. customer "~~refer~~ ISA" person. It may also referred to as a super class - sub class relationship.

2) Generalization The refinement from an initial entity set into successive levels of entity subgrouping represents a top-down design process in which distinction are made explicitly. The design process may also process in bottom up manner. In which multiple entity set are synthesised into a

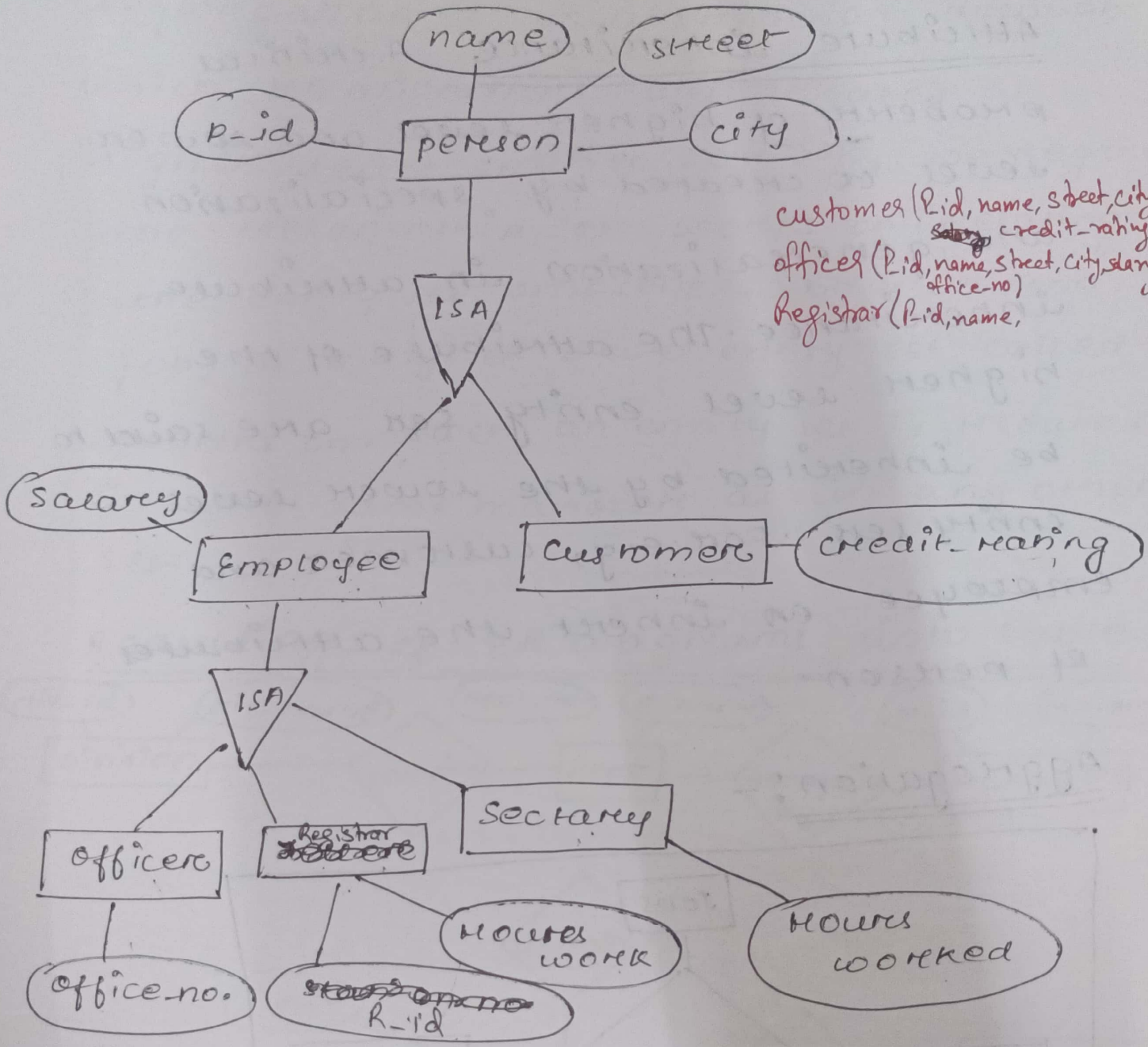
higher level entity set on the basis of common feature. The data base designers may have identify a customer entity set with attribute name, street, city and customer_id, An employee entity set with the attribute name street, city, employee_id and salary. i.e, several entities are common. This commonly express by generalization which is a containment relationship that exist between a higher level entity set and one or more lower level ses

" Generalization is a simple inversion of specialization"

* Diagram for EER diagram

for specialization and generalization.

P.T.O



customer (P_id, name, street, city, ~~salary~~ credit_rating)
officer (P_id, name, street, city, salary, office_no)
Registrar (P_id, name,

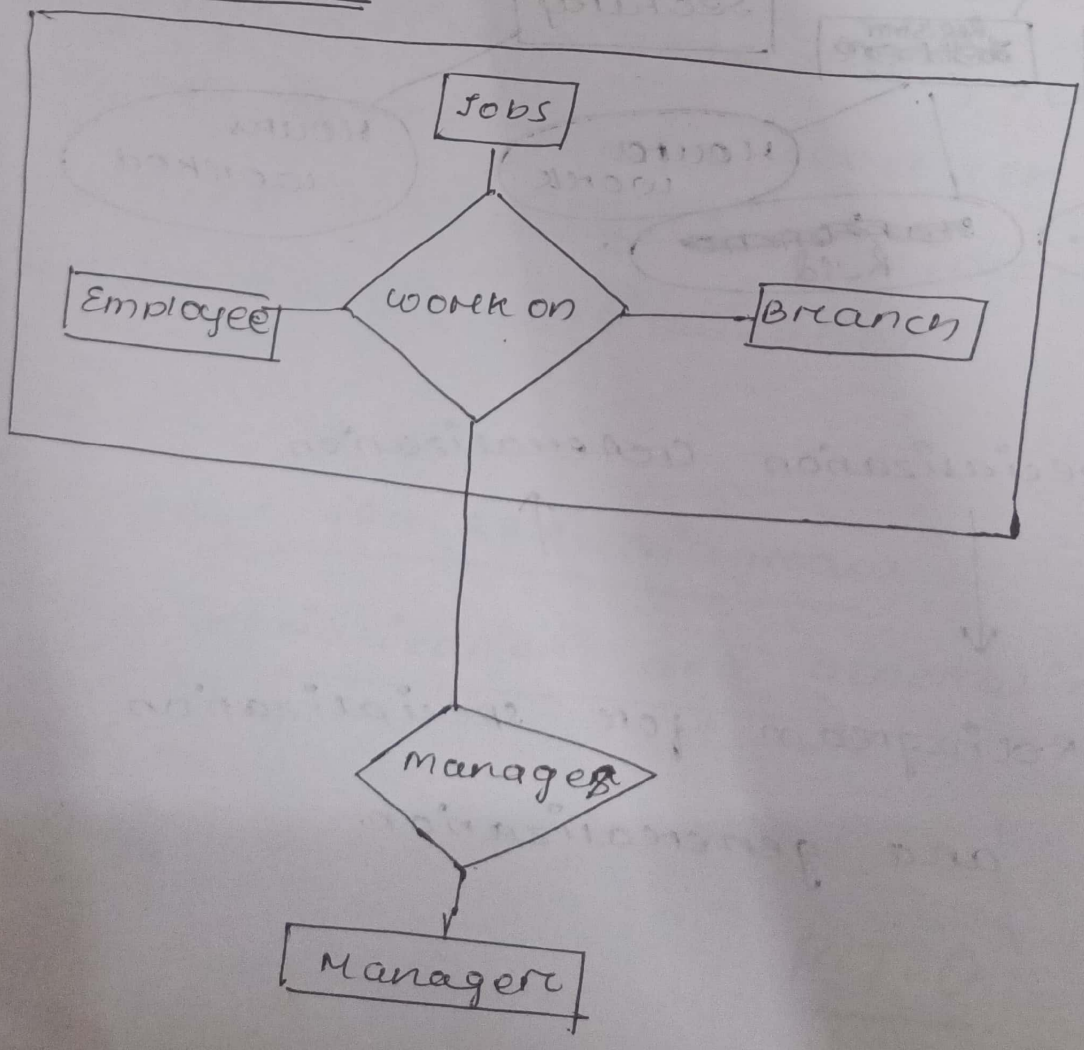
Specialization ↓ Generalization ↑

ER-Diagram for specialization and generalization.

Attribute inheritance

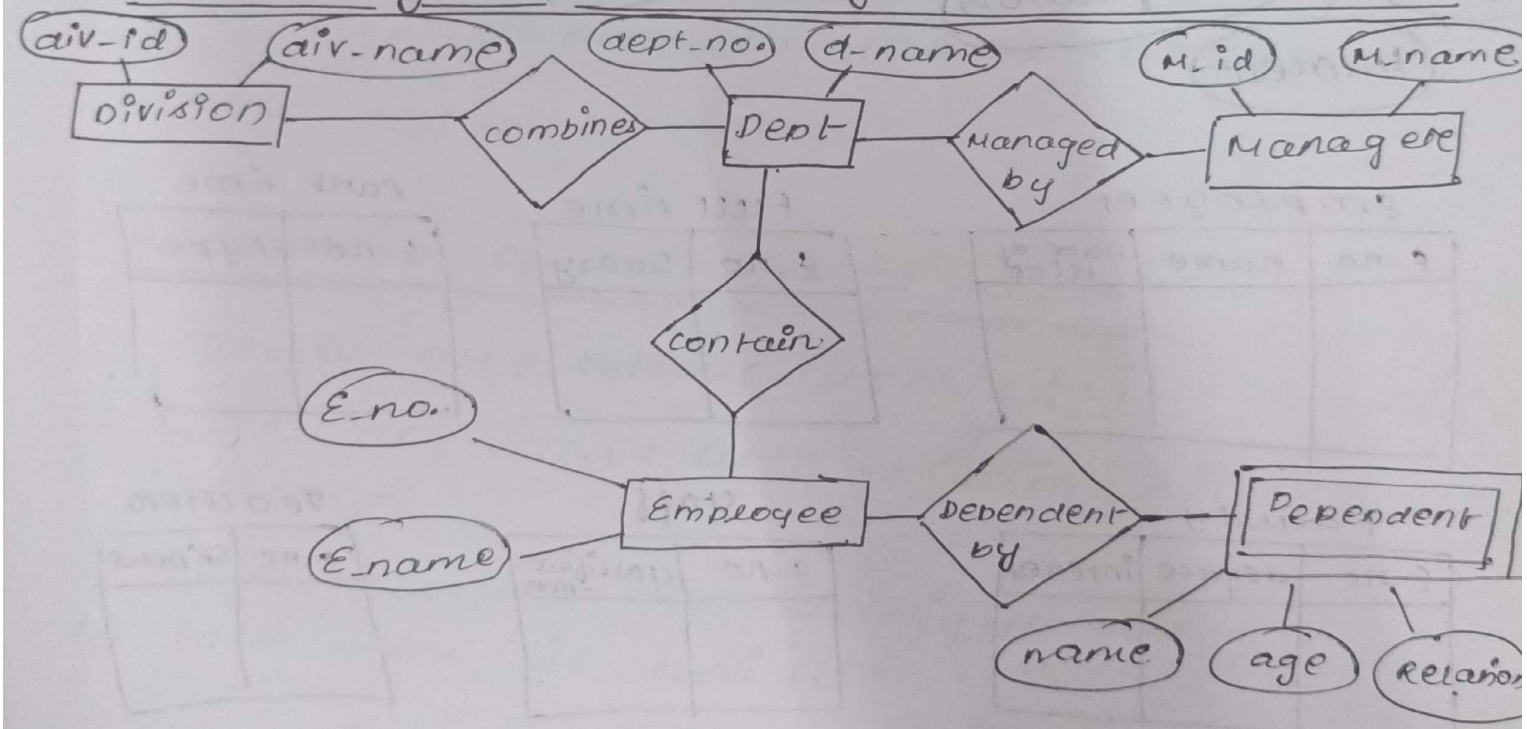
A critical property of higher level and lower level is created by specialization and generalization in attribute inheritance. The attribute of the higher level entity sets are said to be inherited by the lower level entity sets. For e.g., customer and employee can inherit the attributes of person.

Aggregation :-



Aggregation is an abstraction through which relationships are treated as higher level entities. For e.g., we regard the relationship set works on (relating the entity set employee, branch and job) as a higher level entity set called works on. Such an entity set is treated in the same manner as in any other set.

Reducing of ER diagrams into table



div_no.	div_name

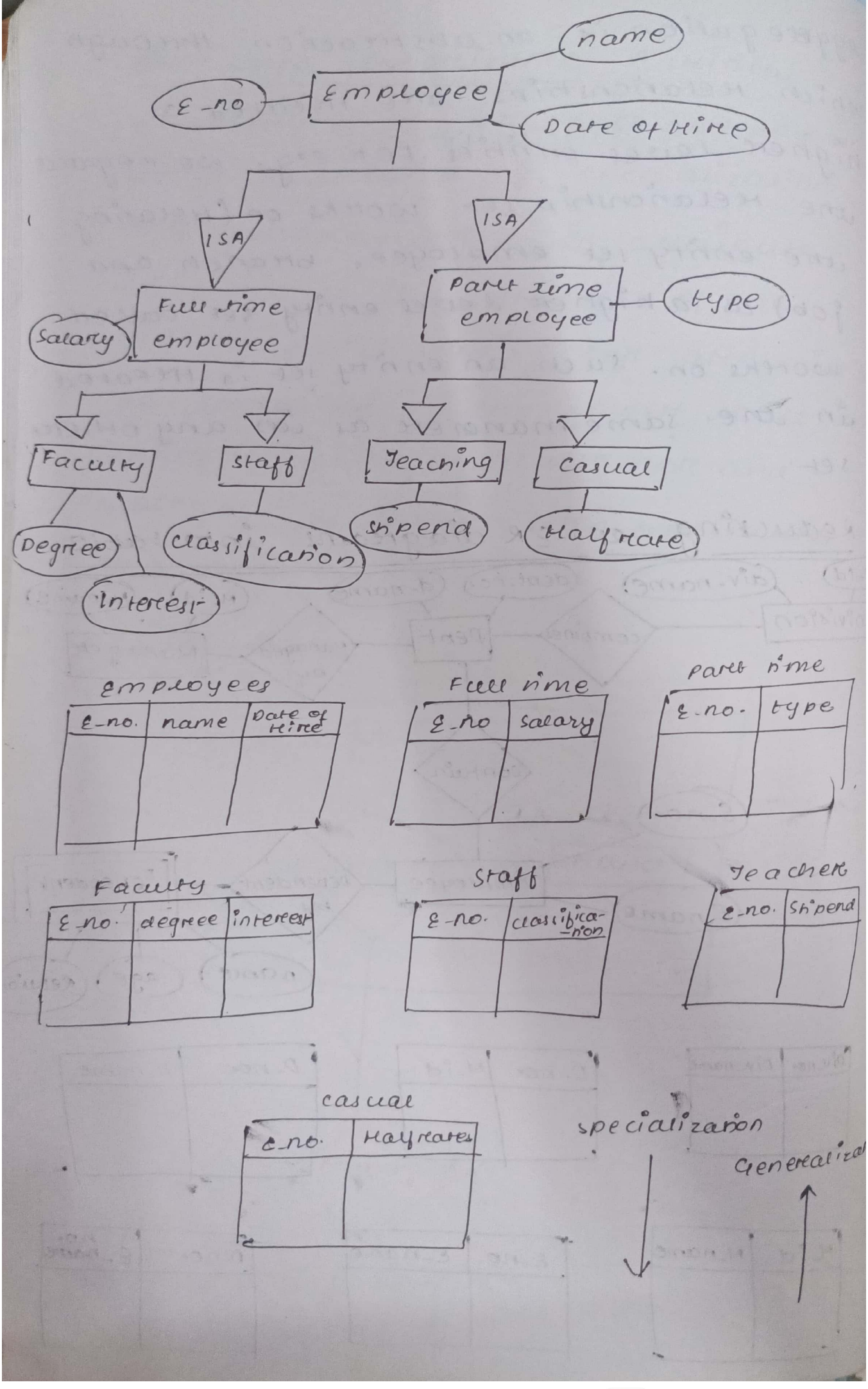
D_no.	M_id.

D_no.	D_name

M_id	M_name

E_no	E_name

D_no.	E_no. name



employees

E_no.	name	Date of Hire

Full time

E_no	salary

part time

E_no.	type

Faculty

E_no.	degree	interest

Staff

E_no.	classifica-tion

Teacher

E_no.	Stipend

casual

E_no.	Hafrates

specialization

Generalization