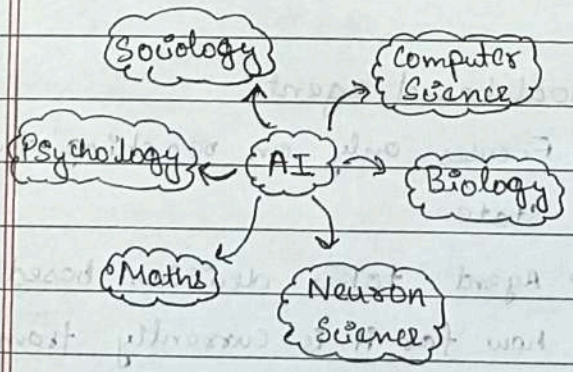
Artificial Intelligence

- Goals of AI :-
  1. Building a machine that can do human intelligence task
  2. Intelligent connection b/w perception and action

AI → Artificial [Man-made]
AI → Intelligence [Power of thinking]

- Reasons of Boost in AI:
  1. Software or device can be made to solve Real-time Problems
  2. Creation of Virtual assistant [SIRI, CORTANA]
  3. Robots development
  4. New Job opportunities

- Applications of AI
  1. AI in Gaming
  2. AI in NLP
  3. Expert System
  4. Computer vision
  5. Speech Recognition



AI Comprised of :-

Advantages of AI	Disadvantages of AI
1. Accuracy ↑ Error ↓	Cost ↑
2. Fast Decision Making	Can't perform beyond the limit
3. Reliability is more	No feelings/Emotions
4. usefulness is Risky Area	dependency ↑

Classification of AI :-

1. weak AI → Cant perform beyond its field or limitations.
   - Examples → Flying machine
   - → using logic
   - → Apple Siri
   - → Playing chess
2. Strong AI → Its the Study & design of machines that Simulate human mind to perform intelligent tasks.
   - It includes borrowing ideas from Psychology and neuroScience.
   - Forgetting things, Genetics, languages they keep all this in mind before performing an action.

③ Evolutionary AI :- Its the study & design of machines that simulate (follow) simple creatures and evolve (learn) for eg → ants, bees

④ Super AI → ~~Its an laptop~~

→ Its a hypothetical concept for now.
→ designing a machine that can perform better than humans.

★ AI agents :-

→ AI System is composed of
- Agent
- Environment

An agent is anything like :-
→ Perceiving its environment through (sensors)
→ Acting upon the environment through (actuators)

Types of an agent

① Simple Reflex agent
→ works only on Current Situation/ Perception and ignores the history of previous State.
→ It will only action if the condition is true.

Limitations of Simple Reflex Agent
1) Very limited intelligence
2) No knowledge about non-perceptual parts of state
3) Can go into infinite loop.

② Model - based agent
It makes decisions based on its current perception of the environment, taking into account its internal model or representation of the world

Unlike simple reflex agents, that only consider the current percept, model-based reflex agent consider the history of past percepts and actions.

For example → vacuum cleaning task.
The agent's internal model would include information about :-
1) The current location of the agent
2) The State of the surrounding area
3) Its goal to keep the environment clean

③ Goal based agent
→ Focuses only on reaching the goal state.
→ Agent takes decision based on how far it is currently from the Goal State
→ Every action is taken to minimize distance to Goal State.
→ more flexible agent.

④ Utility - Based Agents

→ useful when there are multiple possible alternatives and agent has to choose in order to perform best action.

→ Act based not only on goals but also the best way to achieve goal.

⑤ Learning Agents

→ can learn from its past experiences

→ Starts to ACT with basic knowledge and then able to act by adapting learning.

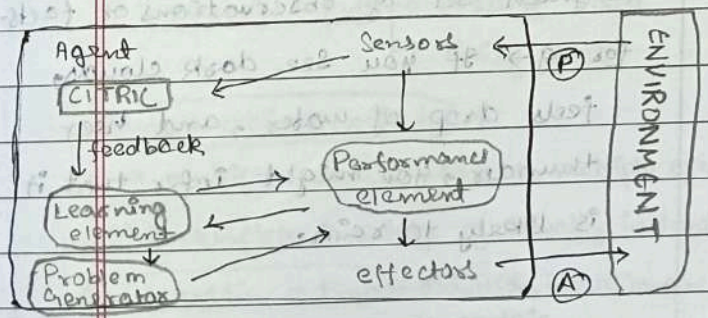
→ components :-

a) Learning Element → Makes improvement in System by learning from environment

b) Critic → Gives feedback about agent's Performance based on standards.

c) Performance element → Selects the action to perform

d) Problem Generators → Suggests the action



★ PEAS : Grouping of AI Agents used to group Similar type of agents together.

PEAS

→ Performance measure → Its the output we get from an agent

→ Environment → All Surrounding things and conditions

→ Actuators → devices through which an agent performs

→ Sensor → Devices from which agent perceives observation from environment

Example → Self driving car

(P) → Comfort, Safety, Time, Legal driving

(E) → Condition of roads, Crossing, Traffic Signals

(A) → Steering, Brakes, Horn, Accelerators

(S) → Camera, GPS, Speedometer

★ Classification of Environment

(1) Accessible and Inaccessible

↓

Agent can obtain complete & accurate info about states environment

→ for eg → Room with temp as State (accessible)

→ Event on Earth (Inaccessible)

(2) Deterministic, or Non Deterministic, → [stochastic]

↓

Agent's Current State and Selected action can completely determine environment's next State. Doesn't have to worry about — uncertain

③ Static or Dynamic

↓

Enviroment doesn't change its state with passage of time.

for Eg → Crossword Puzzles (Static)

→ car driving (Dynamic)

④ Discrete or Continous :-

↓

finite no. of percepts and actions

for eg → Chess game (Discrete)

→ ~~Sudoku~~ Self driving (continous)

⑤ observable or Partially observable

↓ ↓

Agent Sensor can access complete State of enviroment at each point of time

In this some part of enviroment is not in the reach of agent

★ Different tasks in AI

① Mundane (Easiest to learn)

→ Machine / Human Learns Mundane (ordinary) tasks since their birth.

→ Learn by → Perception
→ Speaking
→ using language
→ locomotives

→ ⓐ Perception
→ computer vision
→ Speech, Noise

ⓑ NLP
→ understanding
→ Language generate
→ Lang translation

ⓒ Reasoning

ⓓ Robotics (Locomotive)

② Formal task

Its little complex compare to mundane but not as complex as Expert systems.

- Mathematical Equations
- Geometry
- Logic
- Game theory

③ Expert task → Its more complex than mundane and Formal task.

- Engineering
- Manufacturing
- Monitoring
- Scientific, finance, Medical.

NOTES

- Abduction → Its a form of reasoning where one generates the best possible explanation / hypothesis to account for a given set of observations or facts.

for eg → If you see dark clouds, feels drop of water, and hear thunder, you might infer that it is likely to rain.

- **Induction** → Its a mode of reasoning in which general conclusions or patterns are derived from specific observation or examples.
for eg → If you observe multiple instances of apples falling from trees and each time you see it, they fall toward the ground, you might induce that all apples fall toward the ground.

- **deduction** → Its a mode of reasoning where conclusions logically follow from given premise or statements
for eg → All man are mortal (general premise). Socrates is a man (specific premise). Therefore, Socrates is mortal (deduction conclusion)

★ **Uncertainity**

Its defined as the lack of exact info or knowledge that helps us to find correct conclusion.

Sources of Uncertainity

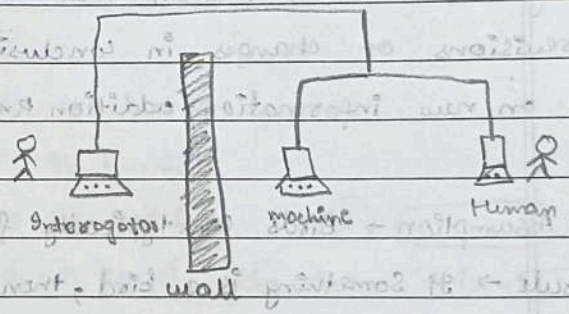
① Uncertain Inputs → Missing Data
→ Noisy Data

② Uncertain knowledge →
→ multiple causes leads to multiple effects
→ Incomplete knowledge of casuality in domain

③ Uncertain outputs → Abduction, induction are uncertain
→ Default reasoning, Incomplete deduction inference

Uncertainity may be caused by problems with data such as :-
→ Missing / unavailble data
→ Unreliable / ambiguous data
→ Imprecise / Inconsistent representation of data

★ **Turing Test**
→ Introduce by Alan Turing in 1950
→ Its used to determine whether or not machines can think intelligently like human
→ There will be a (human Interrogator) on one side of wall and other side a (machine) and (human)
→ It States that if the human Interrogator cannot distinguish the response given by machine and human then the machine has passed the test & is intelligent.
→ i.e if the answer is same = machine is intelligent.



★ **AI technique**

It refers to a specific method, algorithm, or approach used in the field of AI to enable machine to perform intelligent tasks, learn from data, make decisions or solve problems.

Single-State problem	multiple-State Problem
→ Exact prediction is Possible	→ Semi-exact prediction is Possible
→ State is exactly known after any action Sequence	→ not exactly known, but limited to a Set of possible States.
→ info can be achieve through Sensors	→ we can use Reasoning to gather information.
→ Consequences of action is known to the agent.	→ Consequences are not always known to the agent
→ Its Simple but restricted.	→ Less Restricted, more Complex
→ for eg → Vaccum world	→ for eg → Vaccum world but no Sensors.

Notes:-

- Non-monotonic → It refers to a type of reasoning System or formalism that allow for revisions or changes in conclusion based on new information/addition knowledge

for eg → Assumption → Birds can typically fly.
Default rule → If Something is a bird, then it can fly

Lets consider the following Statements:-
Statement 1: Tweety is a bird
Statement 2: Tweety is a penguin

Based on our default rule, we would initially conclude that tweety can fly because its a bird. However, we encounter Statement 2 that Specifies Tweety as a penguin, which is a flightless bird, we have new information that contradicts our initial assumption.
→ one disadvantage → cant use for theorm prove.

- Default reasoning
Its very common form of non-monotonic reasoning. Conclusions are drawn based "on what is most likely to be true"

→ Default logic is a way of reasoning that allows us to make assumptions or "best guess" conclusions when we dont have complete information. These assumptions are called default rules & represent what we typically expect to be true in certain Situations.

→ for example, lets Say we have a default rule that Says, "If its raining outside, people usually carry umbrellas." In default logic, we would assume that if we see Someone walking outside on a rainy day, they are likely carrying an umbrella, even if we didnt actually see the umbrella.

- Monotonic Reasoning
Once the conclusion is taken, then it will remain Same even if we add Some other info to existing info in our knowledge.

→ Few disadvantages of monotonic :-

- Can't represent real world scenarios
- New knowledge from the real world can't be added.

★ Production System

Helps in structuring AI programs in a way that facilitates describing and performing the search process.

Production System consist of :-

1) Set of Rules → what action to take if the condition is true.
2) Knowledge Base → collection of rules, facts and information that the system uses to make decisions or perform tasks.
3) Control Strategy → It determines how the production system selects and applies rules from the rule base to achieve its goals or objectives.
4) Rule applier → It applies rules on the control strategy.

Steps to Solve the problem :-

→ first reduce Problem so that it can be shown in a precise statement.
→ Problem can be solved by searching a Path through space. [Start → Goal]
→ Solving process can be modelled.

Characteristics :-

1) monotonic PS → we can't apply a new rule after applying a rule later on.
2) Partially commutative PS → It's a type of Production System where the order of a rule application can be rearranged without affecting the final result.
3) Commutative PS → Both monotonic and Partially commutative.

★ Propositional Calculus

word	Symbol	example
not	¬	¬X (not X)
and	∧	X∧Y
or	∨	X∨Y
implies	→	(X→Y) if X then Y
if and only if	↔	(X↔Y)

X : It's hot
Y : It's humid
Z : It's raining

① If it is (humid) then it is (hot) = (Y→X)
② If it is (hot) and (humid), then it's not (raining) = (X∧Y)→¬Z

(P, Q, R) are variables

Set of equivalence relations or laws :-

Commutative	$P \wedge Q \cong Q \wedge P$, $P \vee Q \cong Q \vee P$
Associative	$(P \wedge Q) \wedge R \cong P \wedge (Q \wedge R)$ $(P \vee Q) \vee R \cong P \vee (Q \vee R)$
Double Negation	$\sim(\sim P) \cong P$
De-Morgan	$\neg(P \vee Q) = \neg P \wedge \neg Q$ $\neg(P \wedge Q) \cong \neg P \vee \neg Q$
Absorption	$P \wedge (P \vee Q) \cong P$ $P \vee (P \wedge Q) \cong P$
Law of contradiction	$P \wedge \neg P \cong$ False ($P=1, \neg P=0$ ⓪) ($P=0, \neg P=1$ ⓪)
Law of excluded middle	$P \vee \neg P =$ True ($P=1, \neg P=0$ ①) ($P=0, \neg P=1$ ①)
Law of impotency	$P \wedge P \cong P$ $[P=1 \to 1, P=0 \to 0]$

Rules of Inference :-

① Modus Ponens → If 'p' and 'p → Q' is given to be true, then we can infer that 'Q' is true.

P: Its a holiday

Q: The School is closed

P → Q: If its a holiday, then School is closed

② Modus Tollens → If (P → Q) and ~P → ~Q (not of P implies not of Q) are given to be true, then we can infer that ¬P is true

¬Q = School is not closed

if its not a holiday, then School is not closed

¬P its not a holiday (True)

★ Truth table

It shows how the truth or falsity of a compound statement depends on the truth or falsity of simple statements

Some of truth table example :-

① Negation

P	~P
T	F
F	T

② AND : (P ∧ Q) → True when P and Q both are true

Simple Statement → P, Q

compound statement → P ∧ Q

P	Q	P ∧ Q
T	T	T
T	F	F
F	T	F
F	F	F

★ Tautology → Its a formula which is always true.

→ opposite is Contradiction (always false)

Eg → Show that (P → Q) ∨ (Q → P) is tautology

P	Q	P → Q	Q → P	(P → Q) ∨ (Q → P)
T	T	T	T	T
T	F	F	T	T
F	T	T	F	T
F	F	T	T	T

all are true

∴ tautology

$\ulcorner S \urcorner$ → negetation of S

Date
Page 9

★ First order Predicate logic

This helps us describe and analyze the relationship b/w objects and properties using symbols, variables, and logical connectives. It allows us to make statements and draw conclusion based on logical rules.

Key components of FOPL :-

1) objects → It allows us to talk about objects, such as people, animals, or any other entities in given domain. we can represent objects using variables like x, y, or constants like a, b

2) Predicates → It describe properties or relationships between objects. They are statements that can be either true/false depending on the objects involved.

3) Quantifiers → It helps us express general statements or specify specific instances. for eg → "for all" (∀) and "there exists" (∃) to make statements about all objects or atleast one object.

universal

Existensial

4) Connectives → and (∧), or (∨) and not (¬)

Properties of Sentance

Every atomic Sentence is a Sentence

i) If (S) is Sentence the $\ulcorner S \urcorner$ is Sentence

ii) If S_1, S_2 are Sentances :-

(a) then $S_1 \wedge S_2$ is a Sentence (conjuction)

(b) $S_1 \vee S_2$ is a Sentance (disjunction)

(c) $S_1 \rightarrow S_2$ is a Sentance (implication)

(d) $S_1 \cong S_2$ is a Sentance (equivalent)

iii) If x is variable and S is Sentance then

$\forall x S$ is a Sentence → for all x

iv) If (x) is variable and (S) is Sentance then $\exists x S$ is Sentance

There exist some x for which x is a Sentance

Example of Quantifiers

(1) All Boys like football

$\forall x$: Boys(x) → Like (x, football)

(2) Some Boys like football

$\exists x$: Boys(x) ∧ Like (x, football)

Representing class membership / class ∧ inclusion

(1) Isa → Represents relationship of class inclusion

(2) Instance → Represent class membership relationship

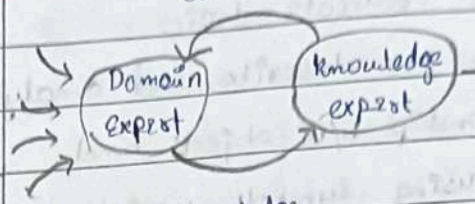
eg → Two facts → Penguin is a Bird; Pingu is Penguin

Bird → Penguin : all the attributes of Bird are inherited by Penguin class

So, class (Penguin, Bird) → ISA

Instance (Pingu, Penguin) → Instance

Predicate calculus	Propositional calculus
Predicates are used that involves constants, variables, relation, function	uses propositions in which complete sentence is denoted by Symbol (Its rainy day) → X
Can represent Tall (Aman)	cant represent individual entities (Aman is tall)
✓ No_of_Sides ( )	cant express generalization specialization or pattern (Square has 4 sides)



Tasks in knowledge Base:-

(i) Collect → Acquiring knowledge from expert

(ii) Interpret → Review & identify key parts

(iii) Analyze → Forming theories, Strategies

(iv) Design → forming better understanding of Problem

Knowledge acquisition techniques:-

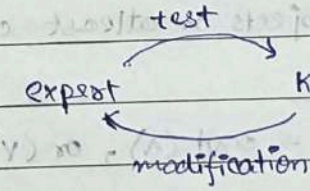
a) Introspection → Expert acts as both expert and knowledge engineer

b) Observation → expert closely observed work

c) Induction → Converting Set of examples into rules.

d) Protocol Analysis → expert is asked to perform task and Verbalized through proces

e) Prototyping → expert + K.E = System



f) Interviewing → experts verify the Knowled

★ Knowledge acquisition

Its the process of extracting, structuring and organizing knowledge from one or more sources. Knowledge is a collection of specialized facts, procedures and judgment rules.

→ In knowledge acquisition, we acquire knowledge from multiple different sources

→ In knowledge elicitation, we acquire knowledge from human experts only

Knowledge Representation
- → Propositional logic
- → First order logic
- → Rule - based System
- → Semantic networks
- → Frames

Date 18
Page

Architecture of Knowledge Based System :-
KBS is a System that draws upon knowledge of Human Expts captured in knowledge Base to Solve problems that normally require human expertise

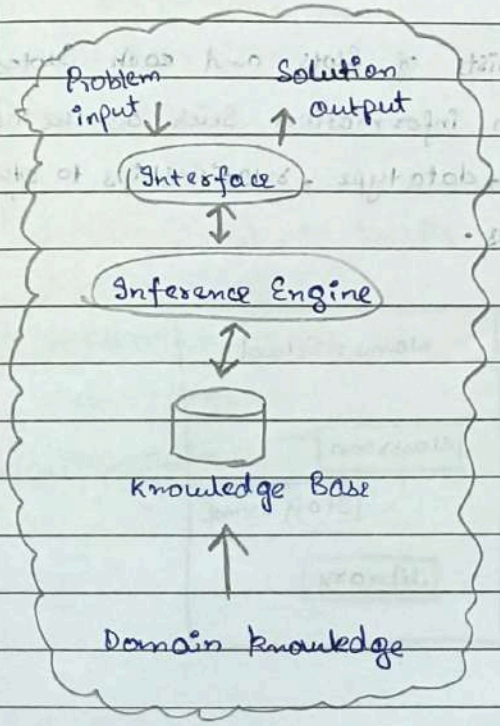
ES is a Specialized type of KBS, not all KBS are designed as Expert Systems.

Components :-

Knowledge Base → Contains organized collection of Facts about a System domain

Inference Engine → Tries to derives answers from knowledge Base. Its the Brain of any expert System.

User Interface → Enables the user to communi-cates with KBS.



• Knowledge Engineering → Its the process of digitizing the knowledge and utilizing it to resolve complicated problems. Its an Essential part of AI.

Role of knowledge Engineer is to develop Software or Systems by gathering all relevant information and putting it into Systematic format.

Types of knowledge :-

① Procedural knowledge → Describes (How) to Solve Problem
→ Also known as Imperative knowledge
→ Provides direction on (how) to do Something
→ Includes Rules, Strategies and procedures

② Declarative knowledge → Describes (what) about problem
→ Tells us facts (what) things are
→ Includes Concepts, Facts, objects.

③ Meta knowledge
→ Describing knowledge about another knowledge
→ used to Pick other knowledge that is best Suited for Solving a problem.

④ Heuristic knowledge → Its often employed in complex & uncertain domains where finding an optimal Soln. may be difficult. They dont rely on facts, logical reasoning. It allows to use Shortcuts to arrive at a good solution.

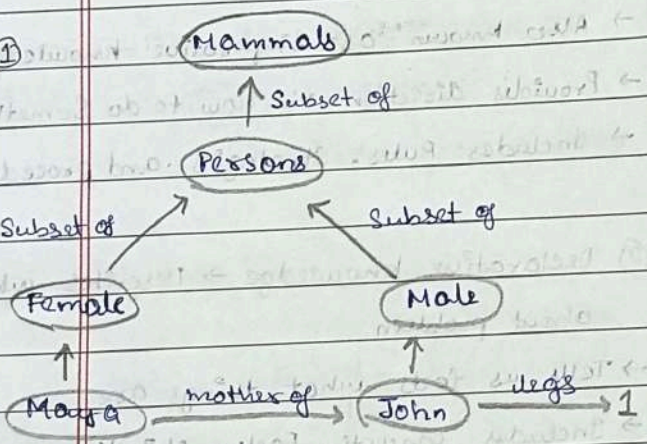
⑤ Structural knowledge →

→ Its the basic knowledge to problem Solving
→ describes relationships b/w various concepts
→ describes an expert overall mental model of a problem

★ Semantic Network

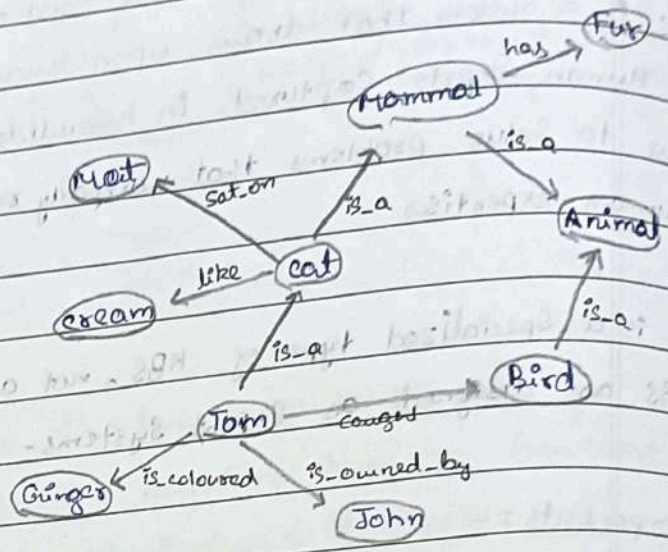
→ Graphical notation for representing knowledge in interconnected nodes/pattern.
→ Popular in AI and NLP: It represents knowledge or Support reasoning.

Eg → ①



Eg → ②

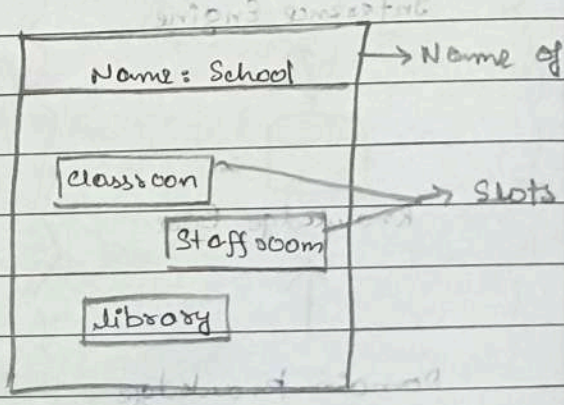
Tom is a cat
Tom caught a bird
Tom is owned by John
Tom is ginger in colour
Cats like cream
The cat Sat on the mat
A cat is a mammal
A bird is an animal
All mammals are animals
Mammals have fur



★ Frames

Its a knowledge representation technique used to organize and structure information about objects, concepts, or entities in a hierarchical manner. Frames provide a way to represent and store knowledge about a paticular domain or subject area.

It consists of Slots and each Slotes can contain information such as the name, value, datatype, relationships to other frames.



operator	Precedence
( )	3
* /	2
+ -	1

∴ Its easy to evaluate postfix for the machine hence we convert infix to postfix / prefix

classmate
Date
Page 13

★ Infix, Prefix and Postfix Notations to write an Expression

Eg→ Infix	Prefix	Postfix
a x b	* a b	a b *
a - b	- a b	a b -
A * (B + C) x D		A B C + * D *
	* + a b - c d	a b + c d - *
x - y * z	- x * y z	x y z * -
P - q - r / a	- - P q / r a	
(m - n) * (P + q)		m n - P q + *

① x - y * z to prefix and postfix

Prefix → Postfix

→ (x - (y * z)) → (x - (y * z))

→ (x - (* y z)) → (x - (y z *))

→ - x * y z → x y z * -

② P - q - r / a to prefix

→ ((P - q) - (r / a))

→ ((- P q) - (/ r a))

→ - - P q / r a

③ (m - n) * (P + q) to postfix

((m - n) * (P + q))

((m n -) * (P q +))

(m n -) (P q +) *

m n - P q + *

★ Different Sorting techniques :-

① Radix Sort

Here, we dont do any comparison between the numbers. It Sorts elements based on their digits or characters.

Eg→				
	904	001	001	001
	046	062	904	005
	005	904	005	046
	074	074	046	062
	062	005	062	074
	001	046	074	904

∴ Just placing each unit's place in ascending order

② Stable VS Unstable Sort

3	5	2	1	5'	10	unsorted array
1	2	3	5	5'	10	Stable Sort
1	2	3	5'	5	10	Unstable Sort

In case of Stable Sort, the order of the element in which they occur in a list or array is maintained even after the Sort.

③ Internal Sorting

In this style of Sorting, the complete Data will be there in memory at a time. Because of this the Sorting can be completed in memory itself.

Different types of internal Sort:-

- bubble Sort
- Selection Sort
- Insertion Sort
- Shell Sort

} $O(n^2)$

- Quick Sort → (worst → $O(n^2)$)
- Merge Sort
- Heap Sort

} $O(n \log n)$

- Radix Sort

} $O(n)$

④ External Sorting

If the Data Collection is too big to fit into the memory then Sorting can not be completed in memory alone. Such techniques are called External Sorting.

⑤ Counting Sort

→ It provide the size of the input and the range of the input (that will be given in the Question)

→ Suppose, inputs → 2, 1, 2, 3, 1, 2, 4

Range → (1 ↔ 5)

1	II
2	III
3	I
4	I
5	O

↓ No. of occurence

It works in linear time but the range is predefined, we can't go outside the range. We might have to take a lot of space unnecessarily just be of one no.

∴ Counting Sort → 1, 1, 2, 2, 2, 3, 4

∴ TC → $O(n+k)$ SC → $O(k)$

⑥ Bucket Sort

→ work on floating point number between range 0.0 to 1.0

→ Inputs should be uniformly and independently distributed across (0,1) to get a running time of $O(n)$

Eg → 0.79, 0.13, 0.64, 0.39, 0.20, 0.89, 0.53, 0.42, 0.06, 0.94

0	0.06
1	0.13
2	0.20
3	0.39
4	0.42
5	0.53
6	0.64
7	0.79
8	0.89
9	0.94

∵ This is the best case where the elements are uniformly distributed.

→ $O(1) \times n$

→ $O(n)$

Eg of worst case → 0.79, 0.78, 0.795, 0.74

0	
1	
2	
3	
4	
5	
6	
7	0.74 — 0.78 — 0.79 — 0.795
8	
9	
10	

(worst case → $O(n) \times (n)$

→ $O(n^2)$

★ Scripts → used for knowledge represent-ation.

→ Its a Structure that prescribes a set of circumstances which could be expected to follow on from one another.

→ ~~conditi~~ considered to consists of a no. of slots or frames but with more specialized roles.

Components of a Script:-

① Roles → Person involved in Event. (Student)

② Props → objects involved in Event (Pen, answer sheet)

③ Entry Conditions → Conditions that needs to be satisfied before event occur in Script (ID card should be there)

④ Results → Condition that will be true after event in Script

⑤ Track → Variations on the Script. (Exam centre)

⑥ Scenes → Sequence of events that occurs

exam hall entry
↓
Getting allocated seat
↓
Getting Question Paper
↓
writing answers
↓
Submition Sheet

Advantages

→ Event Prediction is Possible

Disadvantages

→ Less general than frames

→ May not be Suitable to represent all kind of knowledge.

★ Baye's Theorem

Describes the probability of an Event, based on prior knowledge of conditions.

$$P(B/A) = \frac{P(A/B) \cdot P(B)}{P(A)}$$

P(B/A) → likelihood (Prob. of evidence)

P(A/B) → Posterior (Prob of A when B is true)

P(B) → marginal Prob. (Prob of evidence)

P(A) → Prior Prob. (Prob of hypothesis)

Q) whats the Probability that Person has disease dengue with neck pain?

Given → 80% of time dengue causes neck pain

$P(a/b) = 0.8$

→ P(dengue) = 1/30000 → P(b)

→ P(neck pain) = 0.02 → P(a)

Sol) Supp. a = Proposition that Person has neck Pain

b = Person has dengue

$$P(b/a) = \frac{P(a/b) \cdot P(b)}{P(a)}$$

$$\Rightarrow \frac{0.8 \cdot 1/30000}{0.02} = \boxed{0.00133}$$

Applications of Baye's theorem in AI:

1. Robot/Automatic machine } next step is calculated based on Previous step.
2. Forecasting → weather
3. helps in Solving Monty Hall problem

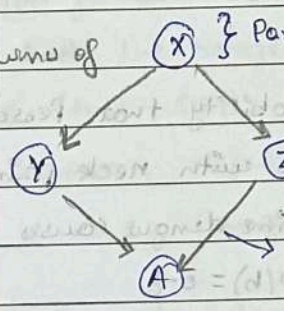
★ Bayesian Belief N/w in AI

~~Bayesian probabilistic independencies~~

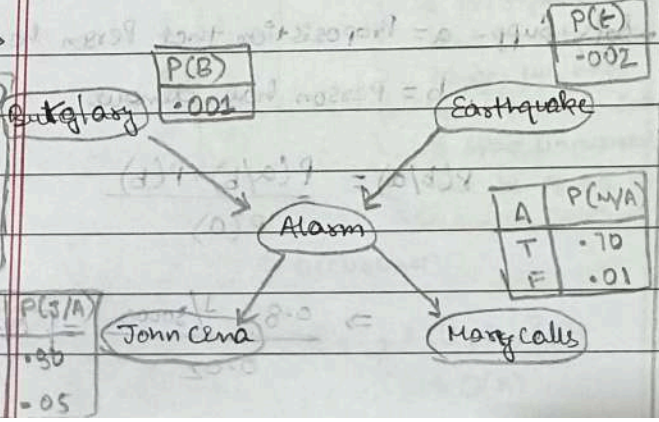
Its a probabilistic graphical model which represents a set of variables and their conditional dependencies using a directed acyclic graph

- It consist of Directed Acyclic graph and
- Table of conditional probability.
- It also consist of Nodes and arc

Direct influence of (X) on (Y)



Problem →



B	E	P(A|B,E)
T	T	.95
T	F	.94
F	T	.29
F	F	.001

A	P(J/A)
T	.90
F	.05

A	P(M/A)
T	.70
F	.01

Q) what is the probability that the alarm has Sounded but neither a burglary nor an earthquake has occured, & both John and Meery call?

$P(j \wedge m \wedge a \wedge \neg b \wedge \neg e) \Rightarrow P(j/a)\ P(m/a) P(a/\neg b, \neg e) P(\neg b) P(\neg e)$

j is called
m is called
a is there
no (b), no (e)

$\Rightarrow 0.90 \times 0.70 \times 0.001 \times 0.999 \times 0.998$

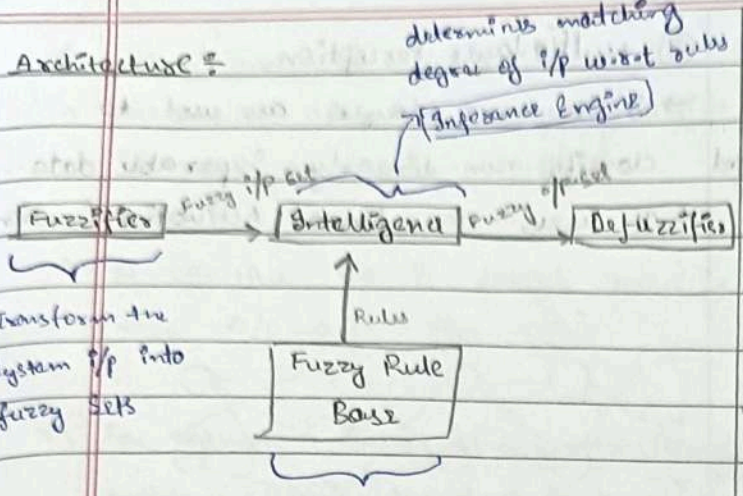
$\Rightarrow \boxed{0.00062}$

~~★ Probability theory from Data~~

★ Fuzzy logic

- Its a type of reasoning that deals with uncertainty and imprecision. It allow for the representation of vague concepts by assigning degrees of truth to statements or variables, rather than strict true or false values.
- In simple terms, fuzz logic mimics human decisions-making by considering shades of grey instead of relying solely on black and white.
- Its useful in situations where information or data is in complete, ambiguous, or subjective.

Architecture :-

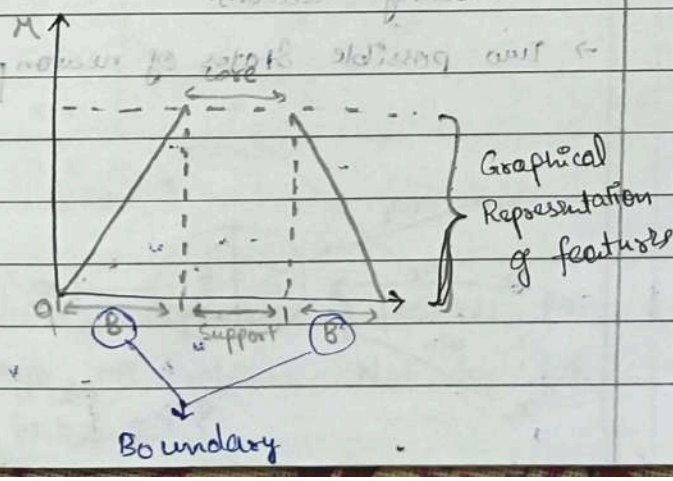


★ Membership function in Fuzzy Systems

→ Membership function is used to represent degree of truth in fuzzy logic.
→ Represented by Graphical Forms.
→ output always lies b/w 0 to 1

Features of membership function :-

(i) CORE → If MF = 1
(ii) Support → MF > 0
(iii) Crossover → MF = 0.5
(iv) Boundary → 1 > MF > 0



★ Learning

Learning refers to ability of a machine or computer system to Acquire knowledge or skills on its own, without explicit programming for every specific task.

Components of learning Systems :-

- Learning Components → (main) acquire knowledge, make changes / improvements.
- Performance Components → Task is performed by choosing an action that needs to be taken.
- Problem generator → Suggests Problems / actions that would lead to generation of new examples to improve learning.
- Critic → Gives feedback

Learning can be achieved by :-

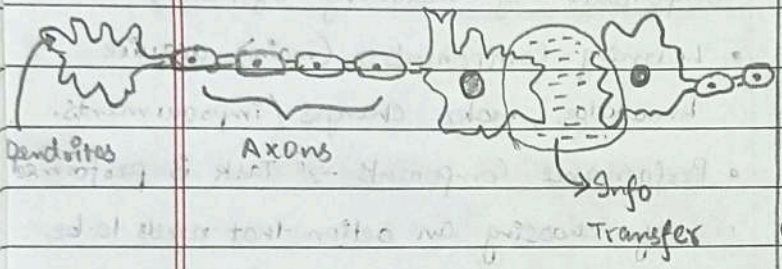
→ Taking Advice
→ From observations
→ From Examples
→ By direct Instruction

★ Bio-logical model of Neuron (Spiking Neural model)

→ Mathematical description of the properties of certain cells in nervous system that generates sharp electrical potentials across their cell membrane.
→ Human Brain is composed of Neuron cells called Neurons and they are connected to other thousand cell by Axons.

★ Bio-logical model of Neuron

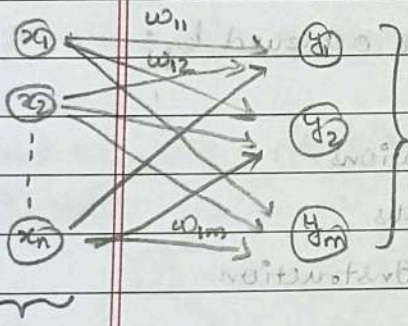
→ Stimulus from external environment or input from sensory organ are accepted by Dendrites.



★ Different types of Artificial Neural Network

① Single Layer Feed forward network

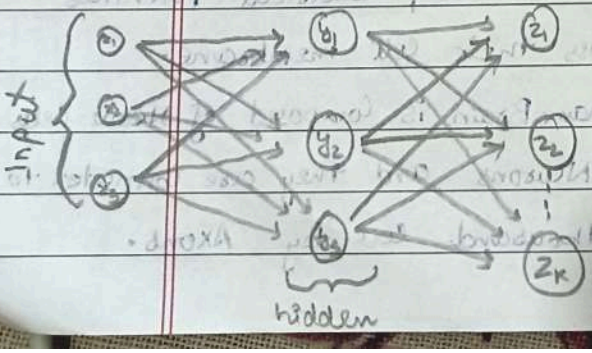
only two layers → input and output



② Multiple layer feed forward network

→ has a hidden layer

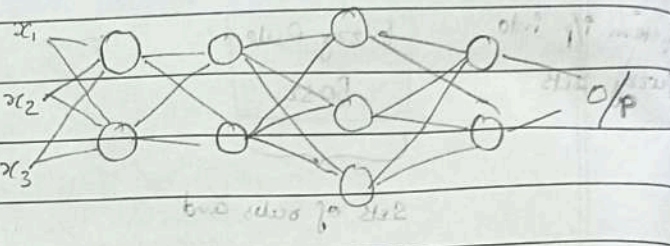
→ computationally more Stronger



③ Multiple layer Perception

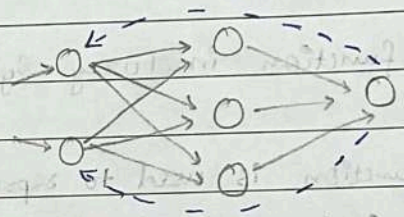
→ 3 or more layers are used to classify non-linearly seperable data.

→ It uses non-linear Activation function



④ Feedback ANN

feedback is provided to adjust parameters



from here, few input goes back to the first layer to minimize the error.

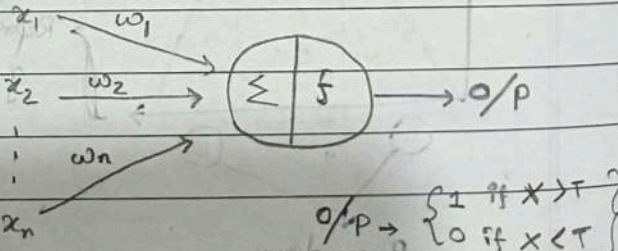
★ MCP Neuron Model. (McCulloh Pitts)

→ first mathematical model of biological neuron.

→ Basic building block of neural network

→ Directed weighted Graph is used for connecting neurons.

→ Two possible States of neuron → Active(1), Silent(0)



★ Bias/ Threshold

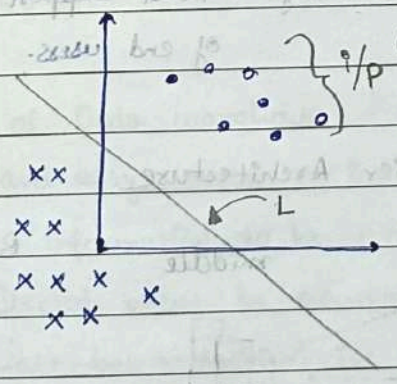
→ Minimum value of weighted active i/p for a neuron to fire

→ If effective i/p is larger than T, then o/p → (1). Else (0)

For example → In a binary classification problem where the goal is to predict whether an email is Spam or not, a model may assign a probability score to each mail. By setting a threshold such as 0.5, if the predicted ~~problem~~ probability of an email being Spam exceeds the threshold, its classified as Spam; otherwise its classified as not Spam.

★ Linearly Seperable Patterns

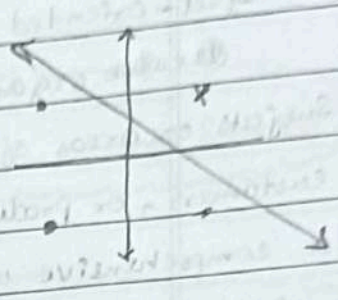
When two classes of patterns that can be separated by decision boundary represented by linear, then they are said to be linearly separable.



Linearly Seperable is possible for AND, OR Problems. Not for XOR Problems.

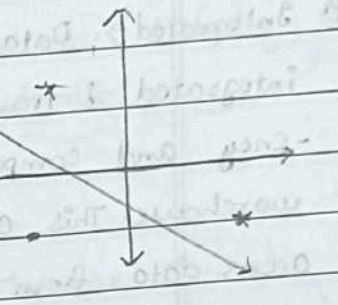
AND Problems:

x	y	o/p
-1	-1	-1
-1	1	-1
1	-1	-1
1	1	1



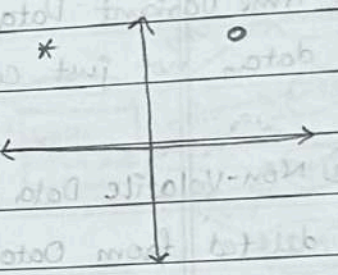
OR Problem

x	y	o/p
-1	-1	-1
-1	1	1
1	-1	1
1	1	1



XOR Problem

x	y	o/p
-1	-1	-1
-1	1	1
1	-1	1
1	1	-1



LS → not possible here

★ Data ware house

In Data warehouse, data is extracted, transformed, and loaded from multiple ~~opera~~ operational systems, such as database, spreadsheets, or transactional system. The data is then organized, cleaned and standarized to ensure consistency and quality. This process is commonly known as data integration.

Key characteristics :-

① Subject-Oriented → A data warehouse is organized around specific subjects or areas of interest, such as sales, customers, or products. It focuses on providing a comprehensive view of data related to these subjects.

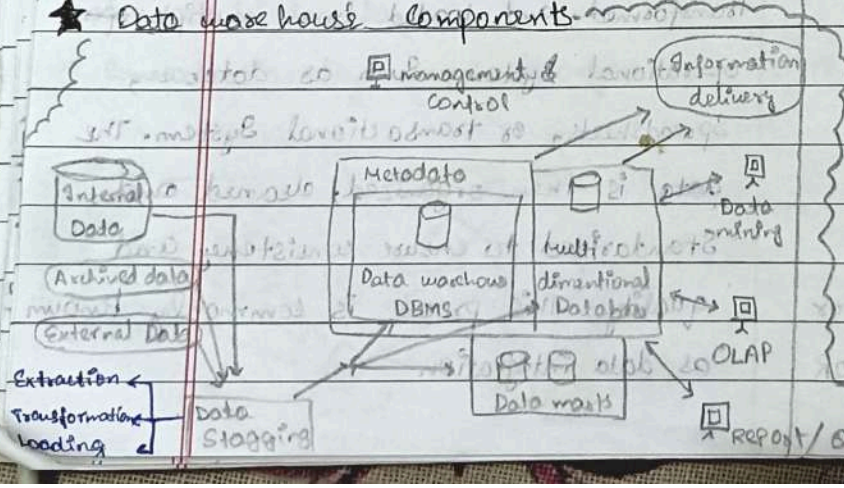
② Integrated → Data from different sources is integrated & transformed to ensure consistency and compatibility within the data warehouse. This allows users to analyze and access data from multiple systems in a unified and standardized format.

③ Time Variant Data → It contains historical data, not just current data.

④ Non-Volatile Data → Data is not updated/deleted from Data warehouse in Real-time

⑤ Data Granularity → In Data warehouse, it is efficient to keep data summarized at different levels.

★ Data warehouse components

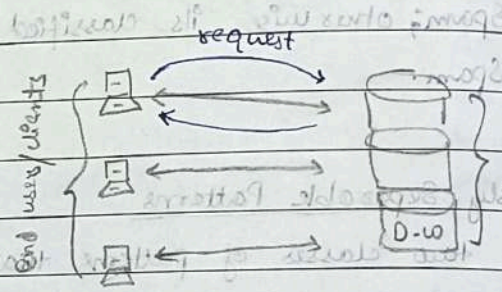


• Management and Control → Coordinates services & activities within Data warehouse

• OLAP → Its a technology and approach used for analyzing and querying multidimensional data from a data warehouse or other data sources. It allows users to perform complex and interactive analysis of data to gain insights and make informed decisions.

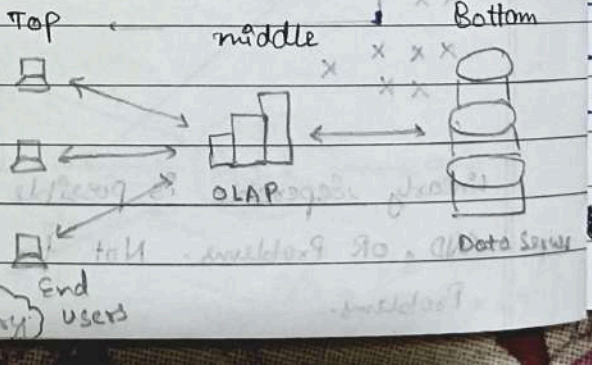
★ Different type of Data warehouse architecture :-

① 2-Tier Architecture



→ Easy to maintain
→ Fast communication
→ Disadvantage → Doesn't support large no. of end users.

② 3-Tier Architecture



→ Its most widely used
→ Bottom tier → After cleaning, transformation data is loaded
→ Middle tier → OLAP Server
→ Top tier → Front end client layers

③ 4-TIER Architecture

→ end user
→ OLap
→ Presentation (New layer) also known as the user interface or client tier, is responsible for interacting with users and presenting information to them. It includes components such as web browsers, mobile apps, or desktop application that users directly interact with.

★ Need for Data warehouse
→ For taking quick & effective decisions
→ Helps users in providing necessary inform.
→ Creating and managing efficient data repository.

★ Goals of Data warehouse
→ Secured and easy access of information to user
→ Consistent information to be provided
→ Data collected must be accurate, verified.
→ Data must be adaptive in nature.

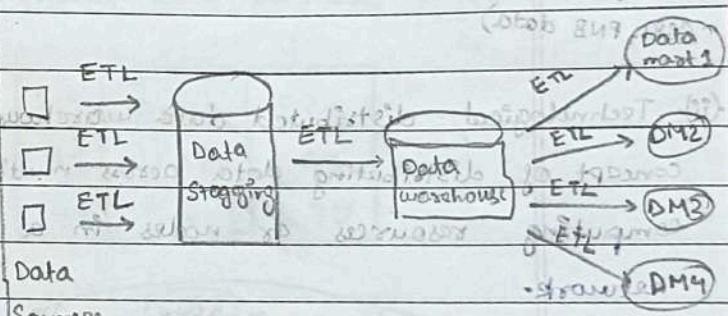
★ Disadvantage → Construction of D.W for large organization is complex task & can take years to complete.

★ ETL Process → (Extraction, Transform & Load)

Extracting
(Data brought up from external Sources)

Transforming
(Converting the extracted data into Standard format)
(Conversion, cleaning duplicates, filtering, Standardizing)

Loading
(Import transformed database into a large database or data warehouse.)

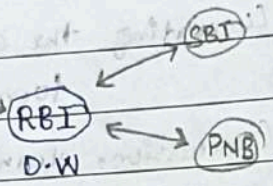


Data warehouse	OLTP
→ It Support pre-define operations. (Retraiving of Data, Insert, delete & update)	→ It only Support Predefined operations.
→ Query accesses thousand or million of records	→ few records at a time
→ Store historical data	→ No historical data.

★ Distributed data warehouse

In this the data are shared across multiple data repositories, for the purpose of OLAP (Online Analytics) and where each data warehouse may belong to one or more organisation.

for eg →



Categories of distributed data warehouse :-

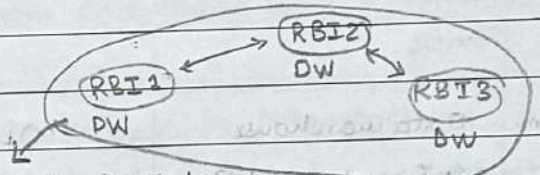
(i) Local and Global data warehouse

↓ data of a particular site (SBI, PNB data)

↓ Integrate data (RBI data)

(ii) Technological distributed data warehouse :-

Concept of distributing data across multiple computing resources or nodes in a network.



If we summarize a report, then all the 3 gonna be treated as 1.

(iii) Independently evolving distributed data warehouse :-

here multiple data warehouse or data marts evolve independently over time. and they have its own schema, data model, data sources and business rules.

★ Data mart

Datamart is a subset of the DW that is usually oriented to a specific purpose.

Reasons for Creating Data mart

→ Easy access of frequent data
→ Improved end-user response time
→ Less cost

Different types of Data mart :-

- Dependent Data mart → In this the Data mart is built by drawing Data from Central Data warehouse that already exists.
- Independent Data mart → Datamart is created without help of Data warehouse

Disadvantage :-

(i) unorganized development
(ii) Increase in Data mart size leads to problems. Such as performance degradation, data Inconsistency.

Difference b/w data warehouse & datamining

→ Data warehouse Store detailed data, allowing Complex analysis across multiple dimensions. while data marts often Store Summarized and aggregated data optimized for specific business areas.

→ Building and maintaing a data warehouse is complex and time-consuming, involving data. while data marts are quicker and easier to implement due to their narrow scope and simple requirements.

★ 5 Steps in data mast :-

(i) Designing →
- understanding the department's requirement
- Integrating and Summarizing data
- ensure data quality
- providing user-friendly access
-

(ii) constructing →
- Creating Schema objects (table, index)
- Determing best table access Strategy.

(iii) Population → (Loading the data)
- getting data from the Source (ETL)
- cleasing and transforming the data.

(iv) Accessing →
- Involves putting the data to use.

(v) Managing
- Involves managing the data over its lifetime
- Secure the data
- Managing the growth of data.

★ Dimensional modelling

It uses a multidimensional Schema having measure (1,2,3) and dimension (a,b,c) attribute

- Facts are the numerical measures or quantities by which one can analyze relationship b/w dimensions. The relations containing such multidimensional data are called Fact tables.

Bid	tid	number
B1	1	25
B2	2	26

- Dimensions → A Dimension table is a table associated with logically related attributes

Bid	Author	Price
B1	ABC	40
B2	XYZ	70

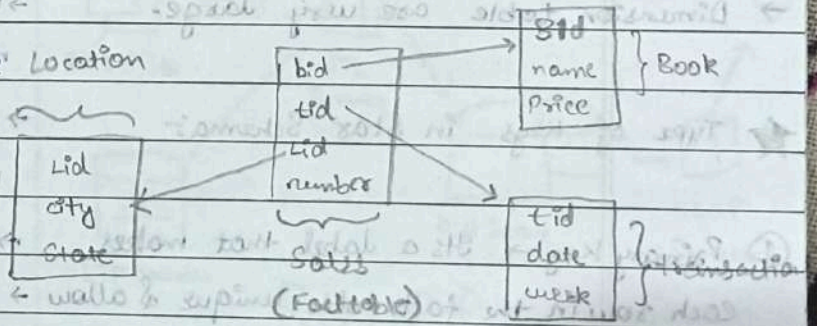
★ Data ware house Schemas

→ (i) STAR SCHEMA

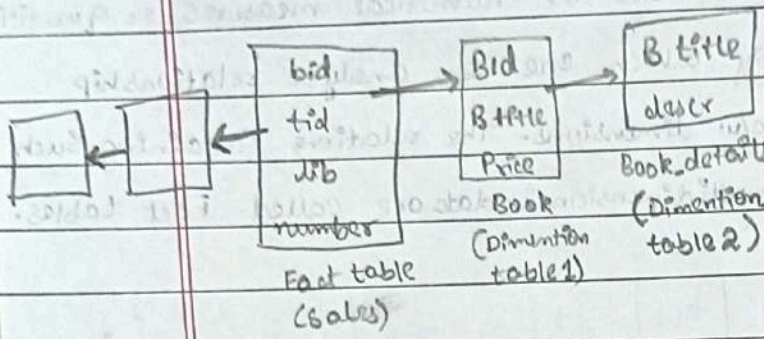
⇒ It consist of Fact table with a single table for each dimension

→ Simplest & easy

→ most Suitable for Query processing.



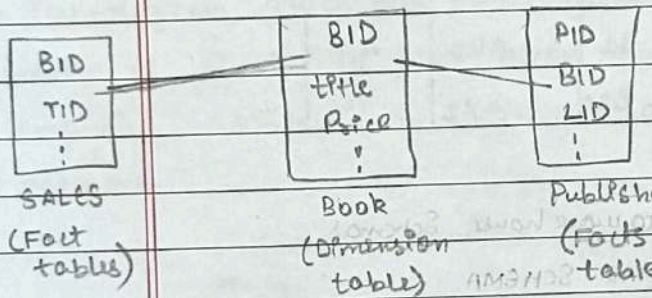
(ii) Snowflake Schema → Its an extension of the Star Schema, which organizes data into a central fact table connected to multiple dimension tables.



→ Dimention tables are easier to update.

→ Disadvantage → Complex Schema.

(iii) Fact constellation Schema → Also known as Galaxy Schema. In this, multiple Fact tables shares the Dimension tables.



Disadvantage

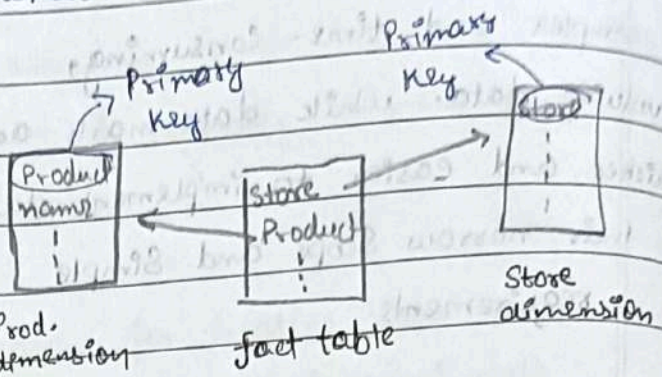
→ Complex due to multiple fact tables.

→ difficult to manage

→ Dimension table are very large.

★ Type of Keys in Star Schema:-

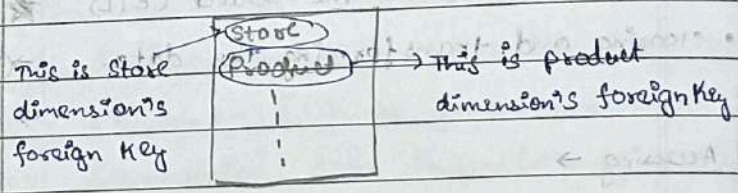
① Primary Key → Its a label that makes each row in the fact table unique & allow us to link it to relevant information in the dimension table.



(ii) Surrogate Keys → They are created by the System and have no inherent meaning or relationship to the data they represent. Surrogate Keys help ensure that each record in a database table has a distinct identifier, making it easier to handle and organize the data

for Eg → MUMBAI123
DELHI123

(iv) Foreign Key → Primary key of each dimension table must be a foreign key in the fact table.



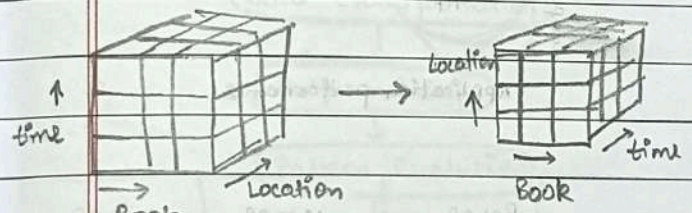
Multidimensional Data modelling	Relational Data modelling
→ Data is stored in multidimentional database	→ RDBMS (Relational database)
→ Storage of data in multidimensional Cubes	→ 2D Tables
→ Data's in denormalized form	→ normalized form
→ Mainly used for OLAP (Analytical Purposes)	→ OLTP (Transactional Pur
→ MDX language is used	→ SQL lang is used

- OLAP (Online Analytical Processing)
It's a category of software technology that enables analysts managers to analyze the complex data derived from the Data warehouse.

- OLTP (Online transaction Processing)
It is refers to a class of systems and applications designed to manage and process real-time transactions. (Insert, update, delete)

★ Different OLAP operations

- Pivoting → It's the technique of changing from one-dimensional orientation to another. Also called rotation.



- Slice and Dice → They help uncover patterns, trends, and relationship within the data and enable users to make informed decisions based on the insights gained.

→ Slicing is like taking a slice out of the larger dataset to examine and analyze a particular portion of it.

→ Dicing is like cutting the data into different slices and examing each slice separately.

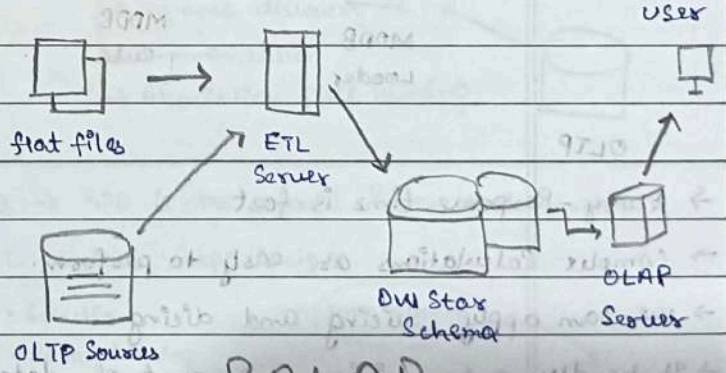
- Rollup and Drill Down
If we come to confined data from large data then its called Rollup.
If we go from confine data to large data then its called Drill Down.

★ OLAP Rules By (CODD) → Person

→ Multidimensional conceptual view
→ Transparency
→ Accesibility
→ Consistency
→ client/ Server
→ Genesic dimensionality
→ Dynamic Sparse matrix
→ Multi user
→ unrestricted
→ Flexible
→ Unlimited

★ ROLAP (Relational OLAP)

→ It work with relational database whereas OLAP used to work with multidimensional database.
→ (Fact) and (Dimension) Table are Stored as Relations.
→



MDDB → multidimentional database.

Advantages of ROLAP

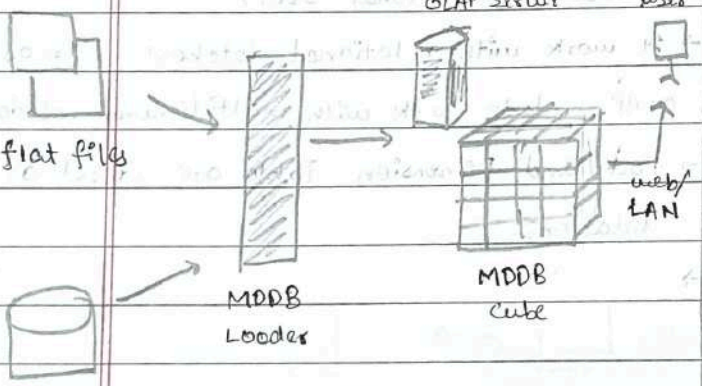
→ Handle large amount of data
→ 2D relational tables can be viewed in multiple multidimensional forms.
→ Any SQL reporting tool can access data.
→ time needed to load data is less.

Disadvantages of ROLAP

→ Difficult to perform complex Calculation.
→ Long Query time for large data size

★ MOLAP (Multidimensional OLAP)

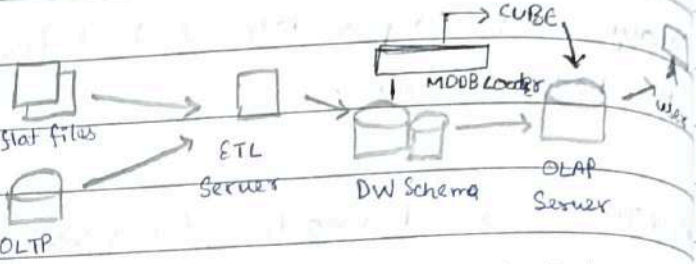
→ MOLAP has a 3-tier architecture (OLTP → OLAP → users)
→ Its a multidimensional database and Storing optimization that helps in better performance
→ Array are used. (we can find cell's location using array
→ Compression is also used (so we can Store more data in less Space)
→ Eg → Oracle's Express Server



→ Query - Response time is fast
→ Complex Calculations are easy to perform.
→ we can apply Slicing and dicing
→ It handles only minimum amount of data.
→ Loading the dataset is slow & high complexity.

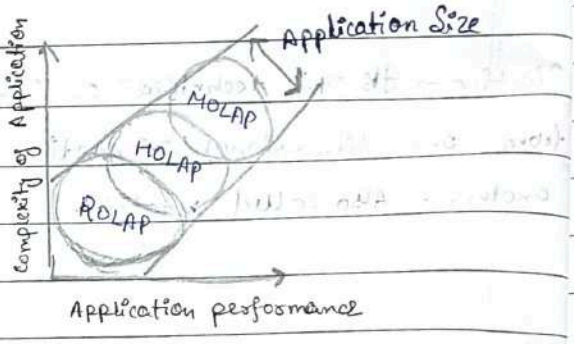
★ HOLAP (Hybrid OLAP)

This System includes the best of ROLAP and MOLAP.



→ The complex queries problem that we had in ROLAP can be Solved using MDDB Loader

★ Comparison of OLAP Servers :-



	ROLAP	MOLAP	HOLAP
(i) Detail data Storage Location	Relational database	MDDB	Relational Database
(ii) Aggregate data storage location	Relational database	MDDB	MDDB
(iii) Space req.	Large	Medium	Small
(iv) Query Response ~~Performed~~ time	Slow	fast	medium
(v) Processing time	Slow	fast	fast.

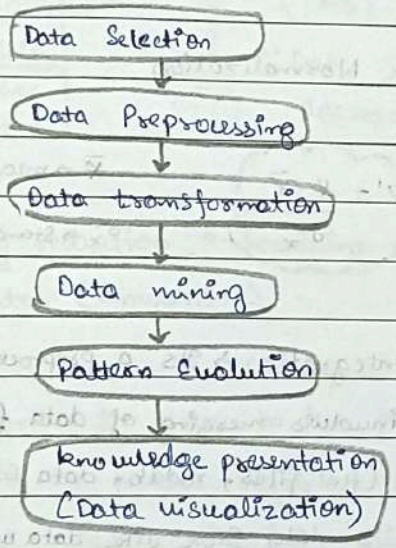
★ Data mining → Process of discovering or mining knowledge from a large amount of data.

→ Another term for Data mining → KDD (Knowledge Discovering from data)

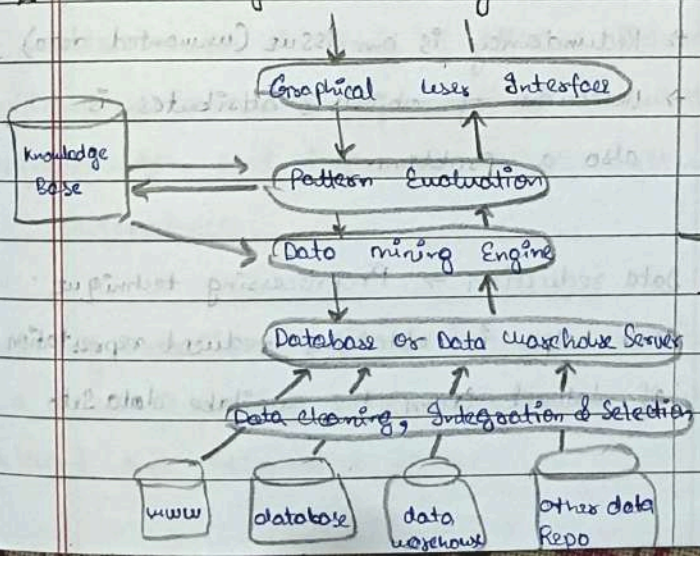
→ Various Data mining technologies are the following.

① Statistics
② Artificial Intelligence
③ Machine Learning

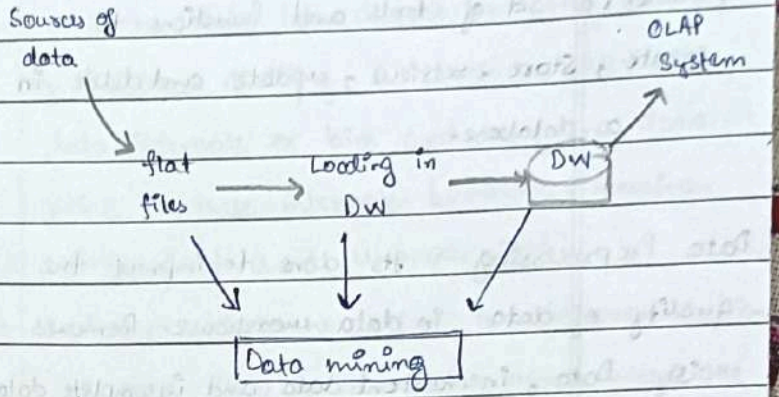
★ Steps in Data mining Process / Phases of KDD in Database :-



★ Architecture of Data mining



★ Data mining in the Datawarehouse Environment



★ Data mining applications :-

① Customer Segmentation → (to understand customers)
② Fraud Detection
③ Demand Prediction

Benefits of Data mining Tasks :-

① PREDICTIVE → Predict values of Data by making use of known results from a different set of Sample data.
- Classification
- Regression
- Prediction
- Time Series Analysis

② DESCRIPTIVE → Enables you to determine patterns & relationship in a Sample data.
- clustering
- Sequence discovery
- Summarization
- Association rule mining.

KDD → KDD is the overall Process that involves various Steps such as
- Selection
- data cleaning
- Preprocessing
- data transformation
- data mining → one of the Step in KDD

DBMS → Its a Software that helps manage and organize data in a Structured way. It provides a Set of tools and functions to Create, Store, retrieve, update and delete in a database.

Data Preprocessing → Its done to improve the quality of data in data warehouse. Removes noisy Data, inconsistent data and incomplete data. (Data with missing values)

Data Cleaning → It cleans the data by filling in the missing values, Smoothing noisy data, resolving the inconsistency and removing the outliers.

ways to Handle missing data during cleaning
→ Manual Entry of missing data
→ using attribute mean
→ using most probable value. (Predicting the value)
→ using Global constant (NA)
→ Ignore the tuple.

Data transformation → Its a data preprocessing technique that tranforms or coordinate the data into alternate forms appropriate for mining. It involves:-

(i) Smoothing → removing noisy data
(ii) Aggregation → Constructing a data Cube (OLAP)
(iii) Generalization → low-level concepts are replaced with high level

(iv) Normalization → Attributes values are normalized by Scaling their value So that they fall in Specified range. for Eg →

2
40
36
47

← we have this table & we have converted all the numbers into range (0-1) thats normalization.

• Min-Max Normalization

$$v' = \frac{v - min_x}{max_x - min_x}$$

v' → new value
v → original value
min_x → minimum value of the attribute
max_x → maximum value of the attribute

• Z-Score Normalization

$$v' = \frac{v - \bar{x}}{\sigma_x}$$

$\bar{x}$ → mean of attribute(x)
σ_x → Standard deviation

Data Integration → Its a preprocessing method that involves merging of data from different Sources (flat files, mddb, data Cubes) in order to form a data Store like data warehouse.

→ Redundancy is an issue (unwanted data)
→ Detection of objects & attributes is also a problem.

Data reduction → Preprocessing technique that helps in obtaining reduced representation of dataset from the available data Set.

→ Splitting → dividing the attributes in topdown approach
→ Merging → dividing the attributes in bottom up approach
→ Supervised →
→ Unsupervised

classmate
Date
Page 29

→ Integrity of the original data should even after reduction in data volume
→ It should produce same analytics result as an original data.

Data Cube Aggregation
Its a process in which information is gathered and expressed in a summary form. (Statistical)

for eg →

Year 2017	
HY	Sales
H1	500
H2	300

Year 2018	
HY	Sales
H1	600
H2	100

Year	Sales
2017	800
2018	700

∴ Data Cube aggregation form

Dimensionality reduction → Removing redundant attributes (unwanted)

Data Compression → It refers to the process of reducing the size or storage requirements of data without losing significant information.

It provides benefits in terms of efficient storage utilization, faster transfer, reduced bandwidth requirements, and improved overall system performance.

Numerosity reduction → No. of features or attributes of a dataset is reduced while preserving the most relevant and informative ones.

{Discretisation} and Concept hierarchy
Discretization is the process of transformin continous data into discrete or categorical for It involves dividing a continous attribute into intervals or bins and assigning data points to these intervals. [Splitting, Merging, Supervised and Unsupervised]
→ Concept hierarchy refers to the organizat of data in a hierarchical manner, where th data is structured into levels or categories of increasing detail.

→ Think of a concept hierarchy like a tree a tree structure. At the top level, you have a broad category or concept, and as you mo down the hierarchy, you encounter more specific subcategories or concepts that provides more detailed information. [Binning, Histogram analysis, cluster analysis
→ for eg → consider a concept hierarchy for the attribute 'Time'. At the top level, you ma have the concept of 'year', followed by 'Quarter' 'month' 'week' and finaly 'Day' a the lowest level. Each level represents a different level of granularity or detail.

Dimensionality reduction → It repeats the original data in compressed or reduced form by applying data encoding or transformation
→ Lossless → If original data can be reconstructed from compressed data without loosing any information.

→ Lossy → If original data can be reconstructed from compressed data without loosing of information.

Numerosity reduction → It reduces the data volume by choosing alternative smaller forms of data representation.

↳ parametric → dimensionality reduction that involves the use of Statistical models or determine the most relevant feature in a dataset. (Regression, log-linear models)

→ Non-parametric → data is stored in the form of histogram, clustering, Sampling.

Supervised → Input data is provided to the model along with the output target variables is given, the goal is to train the model so that it can predict the output when its given new data.

1) classification → It predict / classify the discrete values such as, Male / Female True / False

2) Regression → It predict continous values such as Price, Salary, etc.

Unsupervised learning → Here only the input data is provided to the model and it does not ~~really~~ rely on predefine labels or target outputs for training.

1) clustering → Clustering algorithm group similar data points together based on their inherent characteristics or proximity in the feature space.

② Dimensionality reduction → (done)

Semi-Supervised learning → A blend of Supervised and unsupervised learning. A small portion of the data is labelled, similar to supervised learning, where each labelled example consists of input data and corresponding output labels. However, the majority of the data remains unlabeled. The goal is to utilize the labeled data to guide the learning process & leverage the unlabeled data to extract additional information and improve generalization to eg → medical prediction

Reinforcement learning → Focus on making decisions based on previous experience.

Noise in Data → Its a random error or variance in a measured variable. There are three techniques to remove noise:-

① Smoothing by BIN means:-
Value of the bin is replaced by mean value (average)

(ii) Bin medians → value of bin is replaced by medians value

(iii) Bin boundaries → value of bin is replaced by minimum and maximum value

Steps of Data cleaning Process

(i) Detection of discrepancy
- ↳ manual error
- ↳ Inconsistent data.

(ii) Transformation of data . for eg →

Degree

B-Tech	B-Tech
B.E	B-Tech
B-Tech	B-Tech
B.E	B-Tech

Q) Divide the given Sample Data in two (2) clusters using K-Means Algorithm

	Height (H)	weight(w)	centroid
1	185	72	(185, 72)
2	170	56	(170, 56) → C_2 $\left(\frac{170+168}{2}, \frac{60+56}{2}\right)$
3	168	60	C_2 [169, 58] ← This will be our new C_2
4	179	68	
5	182	72	
6	188	77	
7	180	71	
8	180	70	
9	183	84	
10	180	88	
11	180	67	
12	177	76	

E.D for row 3 :-
- → $\sqrt{(168-185)^2+(60-72)^2} = 20.80$
- → $\sqrt{(168-170)^2+(60-56)^2} = 4.48$

Since this is smaller, 3rd row will be in 2nd cluster as cluster 1 and cluster 2

→ we will consider first two row and then Solve for the third row to check whether its 1st or 2nd row.

★ Memory - Based Reasoning → It uses known instances of model to predict unknown instances. When a new record arrives for evaluation, the algo finds neighbours similar to the new record, then uses characteristics of the neighbours for prediction and classification

Link Analysis → useful for finding patterns from relationships. applications of Link analysis are :-

(i) Associations Discovery → Algo find combination where presence of one item suggests the presence of another.

foreg → There are chances that a customer might buy Butter when buying bread. So there's a link b/w Butter & Bread.

(ii) Sequential Pattern Discovery → Discovers patterns where one set of items follows another specific set.

(iii) Similar Time Sequence Discovery → It refers to the process of identifying patterns, trends, or relationships in a sequence of events or data points that occur over time.

Association rule mining → By this, analysts can discover interesting relationships b/w items or attributes within transactional data. It has applications in market basket analysis, recommendation Systems.

Parameters →

i) Finding all items that appears frequently in transaction → min. Support count.

ii) Finding Strong associations among frequent items → confidence.

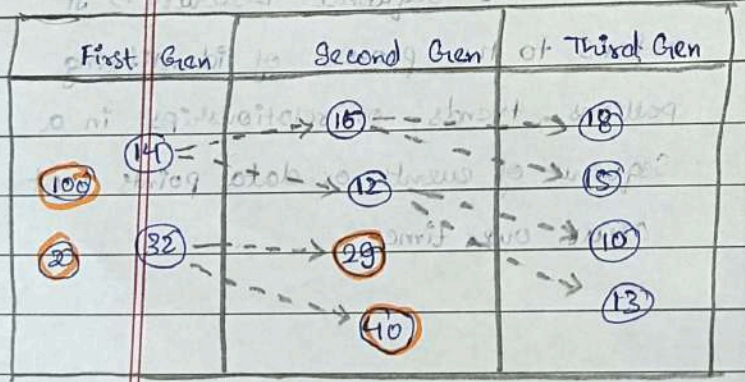
But finding Strong associations among frequent items.

Genetic Algorithm

It Selects the best attributes in every iteration, then it will crossover those attributes and will make few changes with the help of mutation in the order to improve next Successive generation.

for eg →

Suppose we have (4) different attributes & we are finding the most optimised Solution



0 → not Selected

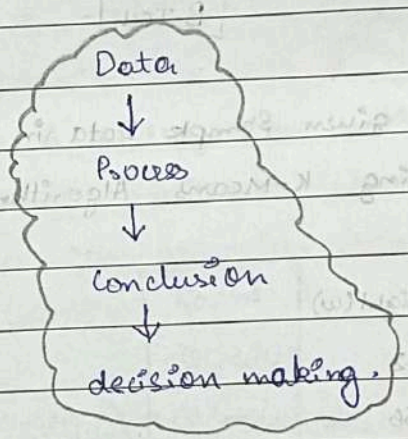
DBMS

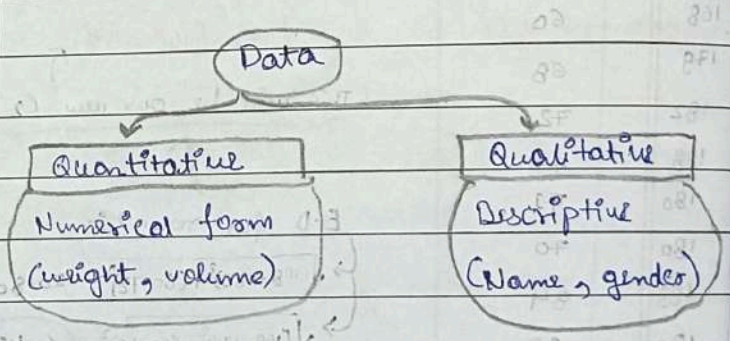
what is data?

→ Data is a collection of raw, unorganized facts and details like text, observations, figures, Symbols and descriptions of things etc.

→ Data is measured in terms of bits and bytes. which are basic unit of information in the context of computer Storage and processing.

→ Data can be recorded and doesn't have any meaning unless processed.

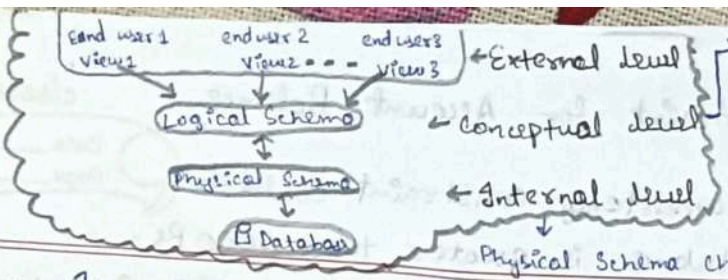




what is information?

→ Its processed, organized and structured data

→ It provide context of the data and enable decision making.



→ Logical Schema is the most important in terms of effect on application programs as programmers construct apps by using logical Schema.

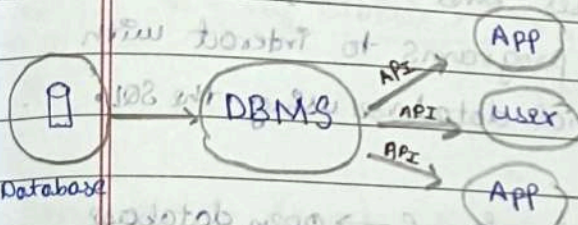
Physical Schema change should not affect logical Schema/app. progra

Database → Its a system where data is stored in a way that it can be easily accessed, managed and updated.

DBMS → Its a collection of interrelated data and a set of programs to access those data.

→ DBMS is the database itself, along with all the software and functionality. Its used to perform different operations, like addition, access, updating and deletion of the data.

→ Its an interface between users and database



DBMS vs file Systems

DBMS and file Systems are both used for manging data, but they have significant differences in terms of its structure, functionality and usage.

Here are disadvantages of file Systems:-

→ Data redundancy and inconsistency

→ Difficulty in accessing data

→ Data isolation

→ Integrity problems

• (These are advantages of DBMS)

Schema and different levels in Schema

A Schema refers to the structure or blueprint of a database. It defines how the data is organized, the relationships between different data elements, and the rules or constraints that govern the data.

Different levels in Schema :- (abstraction)

(i) **Physical / Internal Schema** →

It describes how the data is stored.

It deals with low-level details, including data structures, file organization.

(ii) **Conceptual / logical Schema** →

It serves as a high-level abstraction that focuses on the essential aspect of the database design, providing a global perspective of the entire database system.

It represents the overall logical structure and organization of the entire database.

(iii) **External / view Schema**

→ The external Schema represents the view of the database from the perspective of individual users or groups of users

→ These Schemas provide a simplified and customized view of the database by selecting specific tables, attributes, or view that are relevant to a particular user's needs.

Example → Consistency Constraint for Account Balance

In a banking System, one consistency constraint could be ensuring that the account balance is greater than 1000 Rs.

Abstraction in DBMS

It refers to the process of hiding complex details and presenting users with a simplified and intuitive view of the database. It allows users to interact with the database without needing to understand the underlying complexities of how the data is stored and manged.

Difference b/w instances and Schemas

→ Collection of information stored in the DB at a paticular moment is called an instance of DB.

→ Schema doest change frequently. Data may change frequently.

Data models

It describe the (Design) of a database at logical level. It describes data, data relationships data Semantics and consistency constraints

for eg → ER model
→ Relationships model
→ object-oriented model
→ object-relational data model etc.

Database language

→ DDL (Data definition language)
include CREATE, ALTER, and DROP

→ DML (Data manipulation Language)
Example of DML Statements include SELECT (to retrieve data), INSERT (to insert new records), UPDATE (to modify existing records), and

Query language → SQL (Structured Query language) is the most widely used query language, Providing a Standardized way to interact with relational database. SQL combines elements of DDL and DML to define database Structure and manipulate data.

Interface b/w Java, C and SQL

The interface between Java and SQL is typically facilitated through the use of a Java Database Connectivity. (JDBC). It provides a Standard set of classes and interface that allow Java programs to interact with relational database using the SQL.

Similarly for C → open database connectivity (ODBC)

☆ DBA → Database administrator.
Its a professional responsible for the design, implementation, maintenance, and management of a database System.

→ DBA works at logical level whereas users work at view level. DBA works at Internal level as well but mostly at logical level.

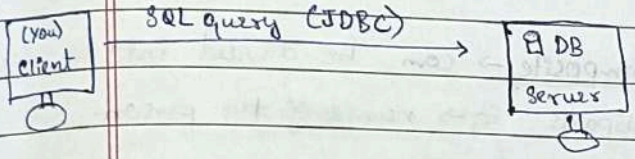
→ DBAs have responsibilities that

include designing and setting up databases, ensuring data security, monitoring performance, resolving issues, maintaing backups, upgrading software, and providing support to users. They play a vital role in managing and maintaing the database environment, ensuring its reliability and efficient operation.

Data application Architecture

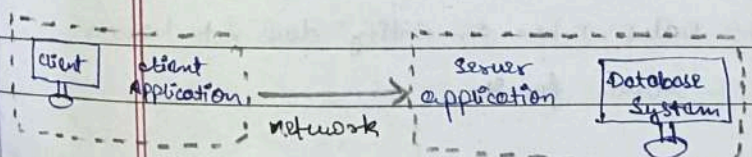
① T1 Architecture → The client, server and DB all present on the same machine.

② T2 Architecture



→ Here, App is partitioned into 2-components.
→ API standards like ODBC and JDBC are used to interact between client & server.

Here, client application will communicate to server application, then server application will call database System and retrieve the data. then database System will return the data to server application. the server application modifies the data and sent it back to client application.

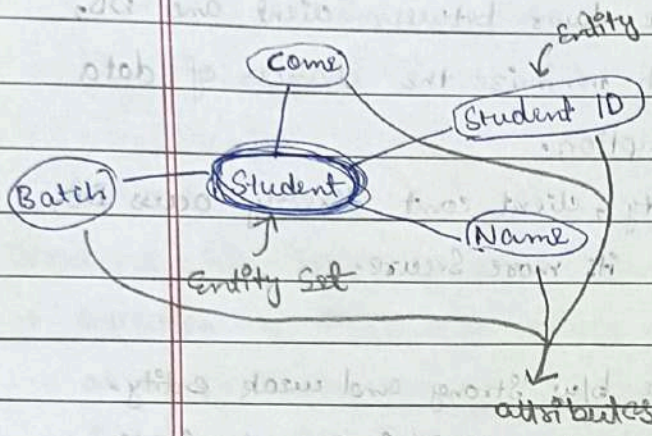


→ T3 architecture are best for Web/W application
→ Scalability due to distributed application servers.
→ Data integrity, App • Server acts as a middle layer between client and DB, which minimize the chances of data corruption.
→ Security, client cant directly access DB, hence its more secure.

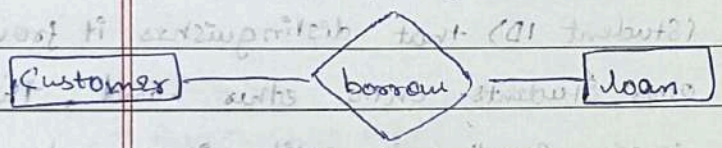
Difference b/w Strong and weak entity.
→ we have this topic in Er model (Entity-relationship model)
→ An Entity is a "thing" or "object" in the real world that is distinguishable from all objects.
→ The Student entity is a strong entity because it can exist independently and has its own attributes. It has unique identifier (Student ID) that distinguishes it from other Students. Onthe other hand, the course enrollment entity is a weak entity because it relies on the existence of both the Student and Course entities. It cannot exist on its own and requires foreign keys to link it to the associated Student and course.
→ To represent the relationship between the Strong and weak entities, the database schema would typically include foreign keys in the weak entity table to establish the connection with the Strong entities. This ensures data integrity and

maintaining the relationship between the entities in the database.

Entity Set



ER model → using er model, developers can create a clear and structured representation of the data and its relationships, making it easier to design and implement a database system that accurately represents the real-world entities and their connections. for eg →



Relationship b/w two Entity.

Attribute consistency constraints → It ensures that values stored in an attribute are there to maintain specific rules or conditions. These constraints help maintain data integrity and ensure that the attribute values meet certain criteria. for eg:-

→ Data type constraint → If an attribute is defined as an integer type, the constraint ensures that only integer values can be stored in the attribute.

→ Range Constraint = It defines the minimum and maximum values that an attribute can take.

→ Not Null Constraint → It ensures that an attribute cannot have a null or empty value.

Types of Attribute

① Simple → Attributes which cant be divided further. Eg → Customer's bank number

② Composite → can be divided into subparts Eg → name of the person.

③ Single-valued → only one value attribute for eg → Student ID

④ Multi-valued → Attributes having more than one value. for eg → Phone no.

⑤ Derived → value of this type of attribute can be derived from the value of other related attributes. for eg → age of a person can be derived from his D.O.B

⑥ Null value → An attribute takes a null value when an entity does not have a value for it.

Degree of Relationship

1) Number of entities participating in a relationship

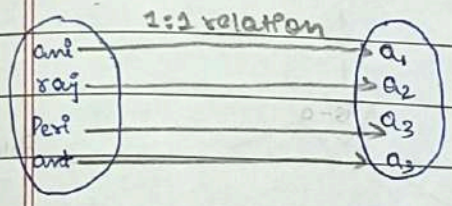
2) Unary → only one entity participates eg → Employee manages employee.

3) Binary → two entities participates eg → Student takes Course. (Binary are common)

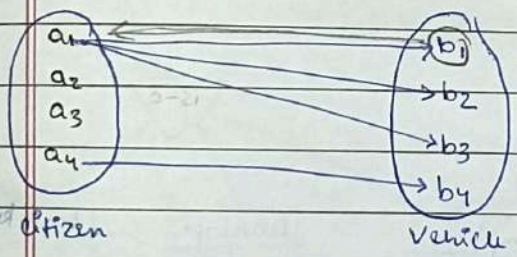
4) Ternary relationship, three entities participates eg → Employee work-on branch, employee works on job.

Mapping Cardinality → Number of entities to which another entity can be associated via a relationship.

- One to one



- One to many

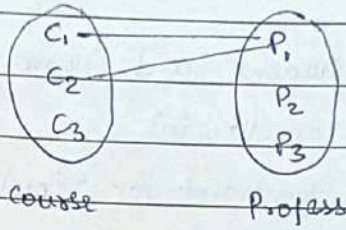


→ A person can have many vehicle, but a vehicle can only be owned by single person

→ defination :-

Entity in A associated with N entity in B. while entity in B is associated with at most one entity in A.

- Many to one → Entity in A associated with at most one entity in B. while entity in B can be associated with N entity in A. for eg → Course taken by professor.



- Many to many → for Eg → Customers buys Product

- Participation Constraints

① Total participation → Total participation means that every instance of the entity on one side of the relationship must participate in the relationship. It indicates that the participation is mandatory, and without it, the relationship cannot exist.

② Partial participation → It means that not every instance of the entity on one side of the relationship needs to participate. It allows for optional participation, indicating that some instances may choose to participate while others may not.

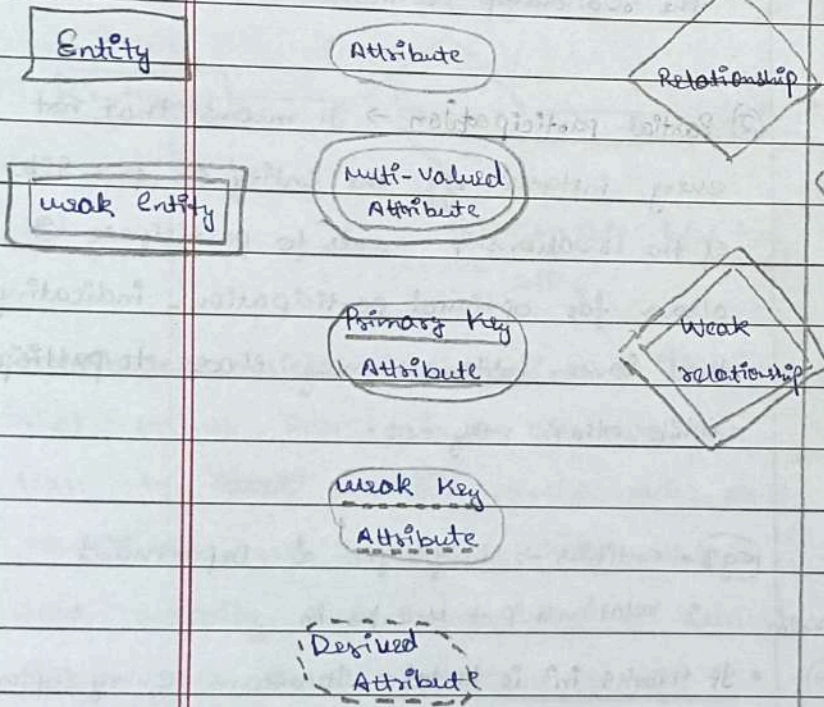
Eg 1 • entities → Employee & Department
• relationship → works in

• If 'works in' is total = It means every employee must work in a department. An employee cannot exist without being associated with a department

- If participation constraint is partial → It allows for some employees to not be assigned to any department, giving them the option of not participating in the relationship.

Eg 2 → • entities → Student and Course
- relationship → 'Enrolls in'
- If participation constraint for 'Enroll in' relationship is total → It means every student must be enrolled in atleast one course. Without enrolling in a course, a student cannot exist in the system.
- If participation constraint is partial → It allows for some students to not be enrolled in any course, giving them the option of not participating in the relationship

Symbols used in ER Diagram

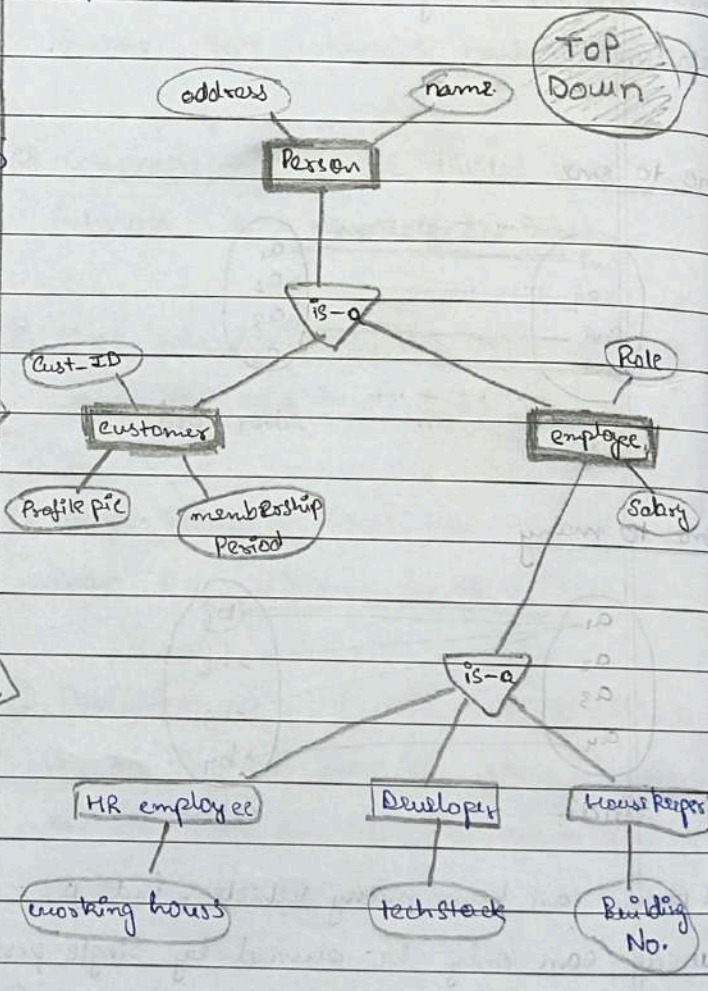


☆ Specialisation → Its splitting up the entity set into further sub entity sets on the basis of their functionalities, Specialities and features.

→ Its a Top-down approach

→ we have 'is-a' relationship b/w Superclass and Subclass. Depicted by triangle component.

→ DB designer can show the distinctive features of the sub entities through Specialization.



DBMS is continued after Pg.