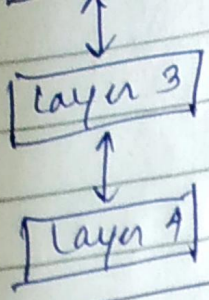
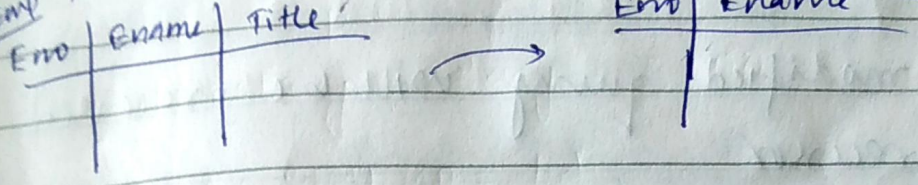


Database view is a subset of a database and it is derived from base relations as a result of a query



View provides security from unauthorised access.

CREATE VIEW SYSAN (ENO, Ename) AS SELECT ENO, Ename from Emp;

View in distributed DBMS

It may be derived from fragmented rel^{n} stored at diff. sites.

When a view is defined its name and its retrieval query are stored in a catalogue.

View definitions can be centralised ~~and~~ at one side, partially replicated or fully replicated.

However evaluating views in distributed systems can be costly. In centralised system views can be modified as a base rel^{n} unlike in distributed rel^{n}

Query Modification

Mapping the query expressed on views into a query expressed on base relⁿ. is done using query modification

Select Ename, Eno, RESP from
Emp, Asg
where Emp.Eno = Asg.Eno

↳ join query

Eno	Ename
10	
11	
12	

PNO	Phone no.
10	-
11	
12	-
13	-

Eno	Ename	Phone no.
10		
10		
12		
11		
12		
13		

Replicate ×

and the modified query will be processed by the query processor

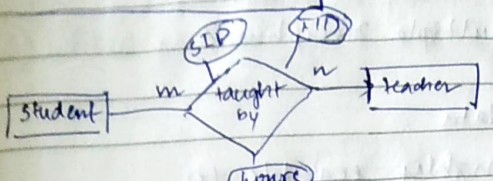
For distributed DBMS also we use query modification for mapping. The qualification defining a view is found in the distributed database catalogue and then merged w/ the query to provide query on base relⁿ. Such a query is called distributed query.

To avoid view derivation actual versions of the view are maintained called materialized views.

It stores the tuples of the views like other database tuples through indices which makes the access to the materialized views faster.

```
p = x * y
s = x + y
if (p <= 1000){
    return p;
else
    return s;
```

Actors in use case → Ext
↓ classdiag
↓ DFD
↓ ERD



$y = f(x)$ one to one

many to many → many to one + one to many

Join multiple tables

- Basic join - cartesian pdt / classic join (matrix multi)
  - ↳ lot of redundant / repititive data.
- cross join
- Inner " / Natural join / Equi join
- Outer " → left, right, full

$O(n^x)$
↳ n in primary memory & accessible

- Self join
- Semi Join

Common attr
data type & name same

Cross join select * from one, two;

Inner join select *
from one, two
where one.x = two.x

x	A	x	B
2	B	2	X

inner

x	A	B
2	B	X

natural

Foreign key . primary key relationship.

Select * from one inner join two;
one.x = two.x ; } natural

Theta join → comparison operator other than =

Transmission impairment

- attenuation
- delay distortion
- noise

Why confluence of Multiple Disciplines?

- Tremendous amt. of data
  - Algos must be highly scalable to handle such as tetra-bytes of data
- High-dimensionality of data
  - Micro array may have tens of thousands of dimensions.
- High complexity of data
  - Data str

Applications of Data Mining

- Web page analysis:
- collaborative analysis & recommendation systems
- Basket data analysis to targeted marketing
- Biological and medical data analysis

Composite primary key

primary key (x, y));

Network layer : (3rd layer)

1) Host to host (src to src) delivery.
2) Logical / IP address
   - Network
   - Host
3) Routing
   - RIP
   - OSPF

ATTITUDE

Attitudes are evaluative statements or judgements concerning object people and event.

Components of attitude :- ABC components of Attitude.

- Cognitive
- effective → emotional / feeling part of attitude
- behavioural → intention to behave in a certain way.

→ The opinion / belief segment of attitude : Cognitive

1. Discrimination is wrong → Cognitive
2. I don't like John because he discriminates → effective
3. I might choose to avoid John because of my feeling towards him. → behavioural

- Types :-

1. Job satisfaction - refers to an individuals general attitude towards his or her job

   The 5 specific dimensions of a job :-
   ① pay ② work itself ③ promotional opportunities ④ supervision ⑤ coworkers.

   2 measures of job satisfaction :-
   ① job descriptive index (JDI)
   ② minnesota satisfaction Questionnaire (MSQ)

2. Job involvement : degree to which a person identifies with his or her job

3. Organizational commitment.
The degree to which an employee identifies with a particular organization and its goals and wishes to maintain membership in the organization

Forcedly → after a predefined no of updates it is refreshed.

Lazily → it is refreshed just before a query is evaluated on the review.

Efficiency issues → for large changes recomputing from scratch is the most efficient way. But for small changes compute the view incrementally by computing only the changes

Materialised view maintaining algo :-

It is the optimal algorithm which computes exactly the view tuples that are inserted or deleted.
But it requires access to the base rels

	a1	a2	a3	a4	a5	a6	a7	a8	a9
01	1	5	9	6	3	8	5	4	2
02	2	8	7	7	3	6	7	5	5
03	10	15	10	12	12	12	10	10	10
04	23	25	27	24	31	33	35	37	39
05	6	8	5	9	10	12	15	18	19
06	5	50	101	75	23	65	72	76	77

01 $\frac{1+5+9+6+3+8+5+4+2}{9} = \frac{43}{9}$

02

Distributed Access Control

The addit additional problem in DBMS is due to the fact that objects and subjects are distributed and the messages with sensitive data can be read by unauthorized user.

the problems are:-

1. Remote user authentication
2. Management of discritionary access rules
3. Handling of views and user groups
4. Enforcing multi level access control

Solutions for authentication:-

1. authentication info is maintained at a central site for global users which can be authenticated only once and then accessed from multiple sites but the problem is the central site can be a point of failure. The info for authenticating user is replicated at all sites in the catalogue. It is more costly in terms of directory management.
3. All sites identify and authenticate themselves intersite communication is protected by the use of site password ~~is~~. Once the initiating site has been authenticated there is no need for authenticating their remote users. This solution is necessary when user info

is not replicated

Authorization - Distributed authorization rules must be stored in the catalogue. They can be either fully replicated at each site or stored at the site of referenced object. views are considered as objects by the authorization mechanism ∴ granting access to the view which is composed object translates into granting access to the underlying objects.

If the view definition and authorization rules are fully replicated then the translation is simple and can be done locally. The translation becomes harder when the view def^n and objects are stored seperately. In this situation, the translation is totally distributed operation

The authorisation granted on views depend on the access right of the view creater on the underlying object. A solution is to record the association information at the site of each underlying object

Handling user groups for the purpose of authorisation simplifies distributed system administration as a particular user group is referred to as public in case of distributed systems. The management of groups in distributed systems posses some problems since the subjects of a group can be located at various sites. and the access to the object may be granted to several groups.

Sol^{n}:-

(1) Enforces access rights by performing a remote query to the node holding the group defn.

(2) Replicates a group defn at each node containing an object that maybe accessed by subjects of the group

Semantic Integrity control:

- Primary key should not be null

  COLG-ID NOT NULL IN COLG

- Foreign key.

Conditions:-

check on COLG (AGE >= 18 AND AGE <= 30)

check on COLG WHEN DELETE (AGE >= 30)

check on clg1 : STCET , clg2 : STCET
(clg1.COLG-ID = clg2.COLG-ID)

check on clg1 : COLG , stcet : STCET
(SUM (clg1.YEAR WHERE clg1.COLG-ID = stcet.COLG-ID))

	Gender	Student	Married	Pizza	V
O1	M	Y	Y	Y	N
O2	F	Y	P	Y	Y
O3	M	N	N	P	P
O4	F	N	Y	N	N

	O1	O2	O3	O4
O1	0			
O2	d(2,1)	0		
O3	d(3,1)	d(3,2)	0	
O4	d(4,1)	d(4,2)	d(4,3)	0
On	d(n,1)			

Method 1

dissimilarity = $\frac{p-o}{p}$

o → how many places same value occured

p → no of properties

Method 2

Yes, P → 1 contingency table for binary data

No → 0

Obj i \ Obj j	1	0	sum
1	q	r	q+r
0	s	t	s+t
sum	q+s	r+t	p

$$d(i,j) = \frac{r+s}{q+r+s+t}$$ → dissimilar

for asymmetric $d(i,j) = \frac{r+s}{q+r+s}$

$$sim_{jaccard}(i,j) = \frac{q}{q+r+s}$$ → measure for asymmetric binary variables

$$\text{coherence}(i,j) = \frac{sup(i,j)}{sup(i) + sup(j) - sup(i,j)} = \frac{q}{(q+r)+(q+s)-q}$$

Name	Gender	Fever	Cough	Test-1	Test-2	Test-3	Test-4
Jack	M	Y	N	N O	P	N O	N O	N O
Mary	F	Y	Y	N O	P 1	N O	P 1	N O
Jim	M	Y	Y	P 1	N O	N O	N O	N O

~~Test~~ ~~01~~ ~~02~~ ~~03~~ ~~04~~

01
02 5/7
03
04

$$\frac{7-2}{7} = \frac{5}{7}$$

2

Mary

Jack	1	0	sum
1	2	0	2
0	1	3	4
sum	3	3	

$$d(1,2) \quad \frac{1}{3} = 0.33$$

Jim

Jack	1	0
1	1	1
0	1	2

$$d(jack, jim) = \frac{1+1}{2+1} = \frac{2}{3}$$

$$= 0.67$$

Mary

Jim	1	0
1	1	1
0	2	1

$$d(jim, mary) = \frac{1+2}{3+1} = \frac{3}{4}$$

$$= 0.75$$

Importance of perception

1. Individuals at work tend to perceive the same stimulus differently. Eg: Strike situation in a factory is perceived differently by managers & trade unions in case of most of the industrial disputes it is actually a perceptual difference between the parties.

2. In a factory when there is an accident the worker would blame the variation in machine layout whereas the supervisor would blame the workers careless attitude

3. Perception is what helps individuals use the knowledge they have in an entirely diff situation from that in which they learnt it

[5, 5, 3, 9, 2]

R = [1, 4]
B = [5, 3]
[1, 5], [4, 3]

5 + 4 = 9 (true)
1 + 3 = 4 (false)

Tandem Bicycle :

1. take two array i/p.
2. make the pairs

→ int blue[] = new int[n]
for (i = 0 to < n)
blue[i] = sc.nextInt()

[[1, 5], [4, 3]]

int [][] arr = new int[n][n]

1 4
3 5
9 1
3 5
4 5

1 2 3 4 5
3 4 5 6 7

3 4 5 6 7

Arrays.sort()

for (int num : arr)
→ i = 0 to n
traverse arr.

2 3 5 5 9
1 2 3 6 7

9 5 5 3 2
1 2 3 6 7

1 + 2 + 3 + 3 + 2 ⇒ 11

9 + 5 + 5 + 6 + 7 = 32

(A) Quality
(B) Punctuality
(C) Accuracy
(D) Conduct

Factors effecting perception.

(1) Internal characteristics of the percieved
(a) Need and desires - An individuals perception varies (b) with his needs and desires that is present in himself.

(b) learning and experience - Succesful experience will help a percieved understand accurately.
Eg - interview or competition

(c) Mood - How the percieved feels at the time of perception determines whether the target will be +vely or -vely percieved.

(d) External factors / Factor or characteristics of the target -
(i) Intensity - bright light, loud voice, catchy colors.
(ii) Size - newspaper advertisement.
(iii) Contrast - underline words, make them bold.
(iv) Repitition -
(v) motion.
(vi) ~~Intro~~ Nobelty.

(e)

Attribution theory
Our perception of people develop from the fact we make inferences about the actions of people. The theory suggests that when we observe we attempt to determine whether it was internally or externally caused depending on 3 factors -
(a) distinctiveness
(b) consensus
(c) consistency

(a) whether the individual displays different behaviour in different situations or similar manner in all situations

Low (I) High (E)

it is attributed to an internal behaviour or it can be " to an external one.

(b) Conserves - when everyone who is faced with a similar situation responds in the same way we can say that the behaviour shows conformity.

High (E) Low (I)

Select sname, cname from salespeople natural join customers on salespeople.snum = customers.snum

group by, having → cannot use where

Select customers.cname natural join salesperson.sname on customers.city = salesperson.city group by comm. Having rating.

Theta join ≠ inequality

Select sname, cname from customers, salesperson where sname < cname and rating > 200

~~Join more than 2 tables~~

Select onum, cnum, orders.cnum, orders.snum from salesperson, customers, orders where customers.city <> salesperson.city and orders.snum = salesperson.snum and orders.snum = customers.snum;

Select onum, comm * amt from salesperson, orders, customers where rating > 10 and orders.snum = salesperson.snum and orders.cnum = customers.cnum;

~~set works~~
syntax written to fetch data → fetch means reading → i.e. query
∴ it is query and not commands.

OSI Ref. model.

Network layer:
1. Logical addressing
2. Routing.

→ source to dest. delivery.
→ choose most optimal route

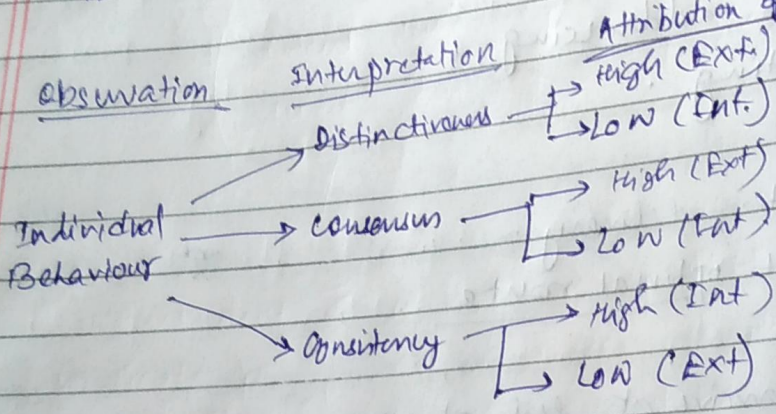
~~expect~~
msg should arrive at dest at uncorrupted manner
→ reliability utmost imp
or
→ throughput "

- Routing - the decision that must be taken by a device called router
  → process of choosing a route from source to dest when a no. of alternate routes are available in order for choosing a particular route the router must take a collective decision. so that the traffic along that route at that point of time is optimal and that route chosen must also be the optimal one based on several criteria like data transfer speed, reliability or throughput.

consistency means that does the person respond the same way over time

Eg: coming to office late for work is not perceived in the same way for the employee for whom it is an unusual behaviour

The more consistent the behaviour is the more the observer is inclined to an internal cause.



$Z_1 = \frac{5 - \mu_1}{\sigma_1}$

	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9		μ
	-0.77	-0.30	-0.16	-0.15	0.17	0.46	2.31	-1.23	-0.61		
01	5	8	7	9	15	13	25	2	6	μ_1, σ_1	90/9 = 10
02	9	18	27	23	30	33	35	25	21	μ_2, σ_2	221/9 = 24.5
03	5	10	12	15	18	19	23	2	9	μ_3, σ_3	113/9 = 12.56
04	7	5	4	5	6	2	1	0	-1	μ_4, σ_4	27/9 = 3
05	9	3	2	1	0	10	15	5	7	μ_5, σ_5	52/9 = 5.78

$$\sigma = \sqrt{\frac{1}{n}\sum (x_i - \mu)^2}$$

$$\sigma_{01} = \sqrt{\frac{1}{9}(-5 - 2 - 5 - 1 + 5 + 3 + 15 - 8 - 4)^2}$$

$$= \sqrt{\frac{1}{9}(0)^2} = 0$$

$$\sigma_{02} = \sqrt{\frac{1}{9}(-18.5 - 6.5 + 2.5 - 1.5 + 5.5 + 8.5 + 10.5 + 0.5 - 2.5)^2}$$

$$= \sqrt{\frac{2.25}{9}} = \sqrt{0.25} = 0.5$$

$$\sigma_{03} = \sqrt{\frac{1}{9}(-7.56 - 2.56 - 0.56 + 2.44}$$

$$= \sqrt{\frac{(0.04)^2}{9}} = \frac{0.04}{3} = 0.013$$

$$\sigma_{01} = \sqrt{\frac{1}{9}(25 + 4 + 9 + 1 + 25 + 9 + 225 + 64 + 16)}$$

$$= \sqrt{\frac{378}{9}} = \sqrt{42} = 6.48$$

$$\sigma_{02} = \sqrt{\frac{1}{9}(20.25 + 42.25 + 6.25 + 2.25 + 30.25 + 72.25 + 110.25 + 0.25 + 6.25)}$$

$$= \sqrt{\frac{510.25}{9}} = \sqrt{56.69} = 7.52$$

Individual
The constraint definition is sent to all the sites and the constraint must be compatible with the rel^{n} data at each sites

Compatibility can be checked at two levels -
predicate and data

if student then clg = "true"
if student then clg = "false"

A constraint c is not compatible with fragment predicate p if "c is true" implies that "p is false" and is compatible with p otherwise then it is globally rejected.

Manhattan distance $h = 1$

$$d(i,j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \dots + |x_{ip} - x_{jp}|$$

Euclidean distance $h = 2$

$$d(i,j) = \sqrt{(|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j2}|^2 + \dots + |x_{ip} - x_{jp}|^2)}$$

Supremum distance $h = \infty$

$$d(i,j) = \lim_{h \to \infty} \left(\sum_{f=1}^{p} |x_{if} - x_{jf}|^h \right)^{1/h}$$

$$= \max_{f}^{p} |x_{if} - x_{jf}|$$

x_1	1	2
x_2	3	5
x_3	2	0
x_4	4	5

$d_m(x_1, x_2) = |1-3| + |2-5| = 5$

$d_e(x_1, x_2) = \sqrt{(|1-3|^2 + |2-5|^2)} = \sqrt{4+9} = \sqrt{13}$

$d_s(x_1, x_2) = 3$

$d(x_3, x_3) = \sqrt{(|1-2|^2 + |2-0|^2)} = \sqrt{1+4} = \sqrt{5}$

Manhattan	Euclidean	Supremum
$d(x_1, x_2) = 5$	$d(x_1, x_2) = \sqrt{13} = 3.61$	3
$d(x_1, x_3) = 3$	$d(x_1, x_3) = \sqrt{5}$	2
$d(x_1, x_4) = 6$	$d(x_1, x_4) = \sqrt{18} = 4.24$	3
$d(x_2, x_3) = 6$	$d(x_2, x_3) = \sqrt{26} = 5.1$	5
$d(x_2, x_4) = 1$	$d(x_2, x_4) = 1$	1
$d(x_3, x_4) = 7$	$d(x_3, x_4) = \sqrt{29} = 5.39$	5

Dissimilarity matrices

Manhattan

	x1	x2	x3	x4
x1	0			
x2	5	0		
x3	3	6	0	
x4	6	1	7	0

Euclidean

	x1	x2	x3	x4
x1	0			
x2	3.61	0		
x3	2.24	5.1	0	
x4	4.24	1	5.39	0

Supremum

	x1	x2	x3	x4
x1	0			
x2	3	0		
x3	2	5	0	
x4	3	1	5	0

$$\cos(d_1, d_2) = (d_1 \cdot d_2) / \|d_1\| \, \|d_2\|$$

$\cdot$ → dot pdt

$\|d\|$: → length of vector d.

Transparent layer

1) Process to process delivery
2) Error control
3) Flow control
4) connection control
5) cogestion control
6) Segmentation & Reassembly

- database audit -

SQL serve .mdf (microsoft data files) ldf → log data files

Access → mdb a

mysql → .sql

oracle → .ora .osc

Expressions in Subquery

Q. Find all customers whose cnum is 1000 above the snum of 'xyz'

select * from customers where cnum = (select snum + 1000 from salespeople where sname like 'xyz');

Salespeople

SubQueries in Having:

These can use their own aggregate functions as long as they donot produce multiple values or use group by or having themselves.

7) Count the customers with ratings above SanJose's average

select rating, count (DISTINCT cnum) from customers group by rating having 'rating' > (select avg (rating) from customers where city like 'SanJose');

* multiple subquery under ANSI is prohibit

group up by

6. Find the total amt in orders for each salesman for whom their total is greater than the amount of the largest order in the table

select snum, sum(amount) from orders group by snum
having sum(amount) > (select max(amount) from orders);

Subqueries and Join

9) write a query that produces the names and ratings of all customers who have above avg orders.

Select DISTINCT cname, rating from customers, orders where
Amount > (select avg(amount) from orders) and orders.cnum = customers.cnum;
select DISTINCT cname, rating from customers natural join orders on customers.cnum = orders.cnum where
Amount > (select avg(amount) from orders);

SHUT UP.
Ghusha Binish Aditya Shabnam.

Correlated SubQuery
inner query returns multiple values.
can refer outer in the inner query to the table in the from clause of the outer query the subquery is executed repeatedly, one for each row of the main query's table

* alias
x.snum = y.snum and x = variable

Q. Find all customers with orders on Oct 3rd
select * from customers outer where '10-Oct-03' IN (select odate from orders inner where outer.cnum = inner.cnum);

YOU ARE THE CAUSE OF MY EUPHORIA

of outer query

The value in the column field varies, the inner query must be executed seperately for each row of the outer query. The row of the outer query for which the inner query is being executed at any time is called candidate row. The procedure to evaluate a correlated subquery is this:

1. Select a row from the table named

Type of research study in which participants are relatively homogeneous subsets experimental units

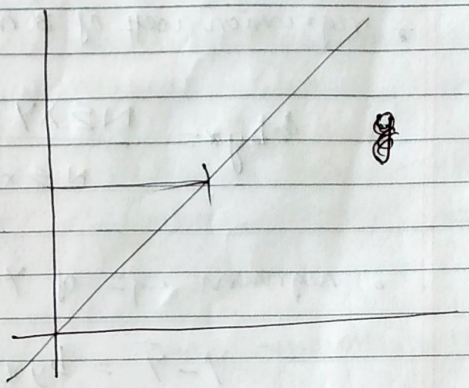
Generalized randomized block design

- study interaction b/w blocks & treatments. (cont=)

- each treatment replicated twice in each block (at least=)

this replication allows the estimation & testing of an interaction item in the linear model

y	x
2	5
8	16
3	9
6	12



Main Ladka PO POPO TO LADKI POPOPO

$$y = a + bx$$

$$r = \frac{(SP_{xy})^2}{(SS_x)(SS_y)}$$

$$b = \frac{SP_{xy}}{SS_x}$$

$$s = \sqrt{\frac{RSS}{n-2}}$$

$$SS_x = \Sigma x^2 - \frac{(\Sigma x)^2}{n}$$

$$RSS = SS_y - \text{regression } SS$$

$$SP_{xy} = \Sigma xy - \frac{\Sigma x \Sigma y}{n}$$

$$\text{regression } SS = \frac{(SP_{xy})^2}{SS_x}$$

$$a = \bar{y} - b\bar{x}$$

I LOB UUU UUU

y	x	x^2	y^2	xy
1	10	100	1	10
2	11	121	4	22
3	15	225	9	45
4	12	144	16	48
5	17	289	25	85
6	18	324	36	108
7	20	400	49	140
$\Sigma y = 28$	$\Sigma x = 103$	$\Sigma x^2 = 1603$	$\Sigma y^2 = 140$	$\Sigma xy = 458$

$$\bar{x} = \frac{\Sigma x}{N} = \frac{103}{7} = 14.71$$

$$\bar{y} = \frac{\Sigma y}{N} = \frac{28}{7} = 4$$

- regression coeff of Y on X

$$b_{yx} = \frac{N\Sigma XY - (\Sigma x)(\Sigma Y)}{N\Sigma x^2 - (\Sigma x)^2}$$

- regression eq of Y on X

$$Y - \bar{Y} = b_{yx}(X - \bar{x})$$

$$b_{yx} = \frac{7 \times 458 - (28 \times 103)}{7(1603) - (103)^2} = \frac{3206 - 2884}{11221 - 10609}$$

$$= \frac{322}{612}$$

$$= 0.52$$

each iteration of inner loop

each situation of outer loop.

candidate row compared w/ outer query predicate → outer reference

Salespeople (snum, sname, city, comm)
Customer (cnum, cname, city, Rating, snum)
Orders (onum, amount, odate, cnum, snum)

Q) Find all customers with orders on October 3rd.

Select * from customers outer where '10-Oct-03' IN
(select odate From Orders inner where outer.cnum = inner.cnum)

or

Slect * from customers first, Orders second
where first.cnum = second.cnum AND second.odate = 10/03/1990

→ this will produce duplicate records.

Q) Find the names and numbers of all salespeople who have more than one customer.

Select sname, cnum from Salespeople main WHERE 1 <
(SELECT COUNT(*) from customers WHERE snum =
main.snum);

Q) Find all orders with above average amounts for their customers.

Select * from orders WHERE ~~inner.cnum = outer.cnum~~
amt > (select AVG(amt) FROM Orders inner WHERE
inner.cnum = outer.cnum);

↓
multiple values

for each → group by
<> → not equal

Exists operator

Exists is an operator that produces a true or false value, i.e. A Boolean expression

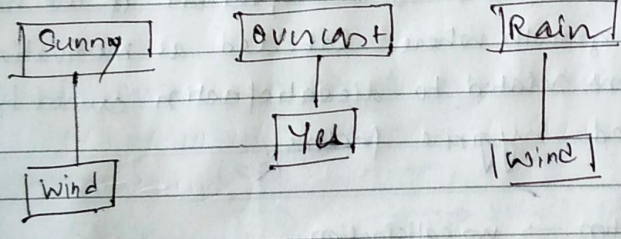
Q. Extract some data from customers table if and only if one or more of the customers are located in London

```
select cnum, cname, city from customers
where EXISTS (select * from customers where
city like 'London');
```

Q. Find salespeople who have multiple customers

```
Select DISTINCT snum from customers outer
where EXISTS (select * from customers
inner where inner.snum = outer.snum
and inner.cnum <> outer.cnum);
```

Day	Outlook	Temp	Humidity	Wind	PlayCricket
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	H	Strong	No
3	Overcast	Hot	H	Weak	Y
4	Rain	Mild	H	W	Y
5	Rain	Cool	Normal	W	Y
6	Rain	Cool	N	S	No
7	Overcast	Cool	N	S	Y
8	Sunny	Mild	High	W	N
9	Sunny	Cool	Nrmal	W	Y
10	Rain	Mild	N	W	Y
11	Sunny	Mild	N	S	Y
12	Overcast	Mild	H	S	Y
13	Overcast	Hot	N	W	Y
14	Rain	Mild	H	S	N



(entropy)

expected info

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

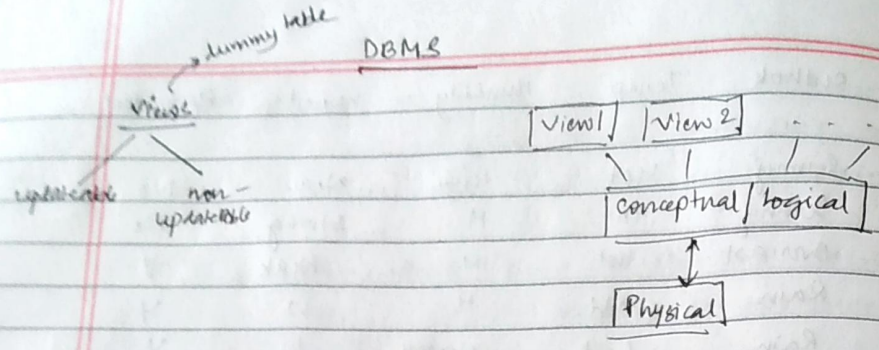
info needed

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$$

info gained

$$Gain(A) = Info(D) - Info_A(D)$$

DBMS



create view <view name> as
select . . . .
. . . . ;

Create view student_accss as
select studid, sname, dept, sem, dob from student;

Select * from student_accss.
↳ displays prev. selected columns in last view ~~stated~~ statement.

TWO FACTOR THEORY

Work conditions related to satisfaction of the need for psychological growth where labelled as motivational factors. Work conditions related to dissatisfaction caused by discomfort were labelled hygiene factors.

Satisfaction → no satisfaction
no satisfaction → less "

The motivation factors where identified as responsibility, recognition, advancement & work itself. When these factors are present they improve performance & effort on the part of job incumbents.

Hygiene factors were company policies & administration technical supervision interpersonal relation, working cond^{n} salary & status.

DBMS

Relational Algebra

Execution perspective

SQL

Procedural → non procedural

Procedural SQL is how you execute a query its details are stored physically somewhere

Parser has no stored docs ⇒ non procedural.

Relational algebra → works on procedural.

" calculus → generally non procedural

- tuple relational calculus → has some stored info
- domain relational calculus

6 fundamental operators of relational algebra and some derived operators:-

1. SELECT (σ) → where clause in SQL
2. PROJECT (π) → select clause in SQL
3. RENAME (ρ) → alias clause in SQL
4. CARTESIAN PRODUCT ($\times$) → cross join in SQL
5. UNION ($\cup$) → union clause in SQL
6. SET DIFFERENCE ($-$) → minus clause in SQL

derived -

1. intersection ($\cap$) (intersect in SQL) — natural ⋈, left outer ⟕, right outer ⟖, full outer ⟗
2. division ($\div$)
3. join
4. aggregates A_{max} A_{min} A_{avg} A_{count} A_{sum}
5. group by G

subnet mask address?

An organization is given 16 class C addresses with 2.4.80.0. What is the supernet mask?

DBMS

Select
$\sigma_{roll=5 \text{ and/}\wedge \text{ dept}='CSE'}$ (student)

Select * from student
where roll = 5 ;

$\sigma_p(r) = \{t \mid t \in r \text{ and } p[t]\}$ → where clause

p → selection predicate.
R → table name / relational R
r → set of attributes

branch (branch_name, branch_city, assets)
customer (customer_name, customer_street, customer_city)
account (account_number, branch_name, balance)
loan (loan_number, branch_name, amount)
depositor (customer_name, account_number)
borrower (customer_name, loan_number)

$\Pi_{A1, A2, \ldots Ak}(r)$ → select clause

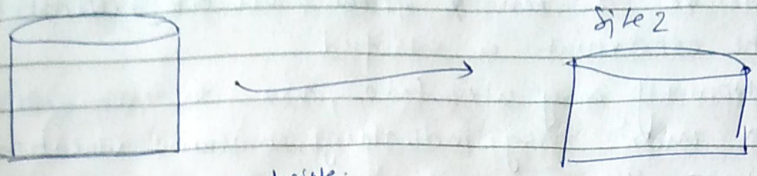
$\Pi_{1,2}$ (student)

$\Pi_{name, roll, dept, \ldots}$ (student)

- if multiple values present 1st occurence printed.
  ↳ inbuilt distinct operator.

$\Pi_{name}(\sigma_{roll=5 \wedge dept='CSE'}(student))$

Enforcement of Distributed Integrity Constraint communicate site: 1.



Individual

It depends on the constraints

Individual → update/modify
→ delete
→ inserting statement

set oriented aggregated

If update is an insert statement all the new tuples are provided to the user. Each side checks the constraint on their own. Now if the update is delete/modify then the query processor will execute the operation for each fragment and the resulting tuples will be combined into a temp. rel^n in case of delete and in case of modify it will be combined in to two rel^n R+ & R-

R+ → has modification
R- → lacks modification

Set oriented constraints - Eg. foreign key - ENO IN EMP

ENO IN EMP DETERMINES (EMP, INSERT, C)

C → (∀ E ∈ EMP)(∀ NEW1 ∈ EMP)

NEW1.ENO (∀ NEW2 ∈ EMP)

(NEW1.ENO = E.ENO ⇒ NEW1.ENAME = E.ENAME) ∧
(NEW1.ENO = NEW2.ENO ⇒ NEW1.ENAME = NEW2.ENAME)

Update qualification will be executed by query processor and it will return one or more temporary relation & it will be sent to all the site that are storing the original database.

The site where the update is submitted must receive a message from each site indicating that the constraint is satisfied

Now if any site fails, then the site will send a ~~und~~ msg indicating constraint violation then the integrity manager will decide if the entire transaction should be ~~deleted~~ or not

Eg:

PNO IN ASG REFERENCES PNO ON PROJ.

Now the update will be computed at the site where it was sent and then the enforcement will be done by the query master.

3) 2

DBMS

Select name from student where dept like 'CSE';

$\Pi_{name}(\sigma_{dept='CSE'}(student))$

DUAL → generally oracle specific

System defined dummy table which allow users to refer to system library.

Select 2+5 from DUAL; all arithmetic

Select tan(a) from dual;
" log(x) " "
" ln(a) " "

Select dob from student minus select sysdate from dual;

ER left to right & top to bottom } reading.

Entities → always nouns
rel^ships → verbs.

ext. entity & actors have to be same.

select to_date (dob) from dual;
select toupper ("abcd") from dual;

select month ('15-feb-23') from dual;

Data stores → as tables in DFD
↳ same entity must also be present.

DFD
entity : noun
process : verb
dataflow → in context level, inflow outflow must be same in next level

Design phase in s/w.

mdf
ldf

Group Behaviour

Group → Formal
Group → Informal

Formal groups are those defined by organisation structure, with designated work assignment, establishing tasks. In formal groups the behaviour that one should engage in are stipulated and directed towards organisational goal

Command group → a group composed of individuals who report directly to a given manager.

Task group → those working together to complete a job task

Informal group → are alliances that are neither formally structured nor organizationally determined. These groups are natural formations in their work environment

that appear in response to the need for social contact.

Task-set group → Those working together to attain a specific objective with which each is concerned.

Those brought together because they share one/more common characteristic

Stages of Group formation -

i) Forming
ii) Storming
iii) Norming
iv) Performing
v) Adjourning

Forming → 1st stage in group development characterised by much uncertainty

Storming → 2nd stage characterised by intra group conflict

Norming → 3rd stage characterised by close relationship and group cohesiveness

Performing → Where the group is fully functional & highest productivity

Adjourning → the final stage in group development for temporary groups characterised by wrapping up activities rather than task performance

Characteristics of group:-

1. Size 2. Purpose 3. Efficient 4. Interdependence
5. Interaction 6.

branch (branch_name, branch_city, assets)
customer (customer_name, customer_street, customer_city)
account (account_number, branch_name, balance)
loan (loan_number, branch_name, amount)
depositor (customer_name, account_number)
borrower (customer_name, loan_number)

Q Write queries in RA & SQL

① Find all loans above Rs 12000
② Find loan number for each loan above Rs 12000
③ Find the names of all customers who have a loan & account or both from the bank.
④ Find names of all customers who have a loan at Bangalore branch.
⑤ Find names of all customers who have a loan at the Bangalore branch but do not have an account at any branch of the bank.
⑥ Find the largest account balance w/o using aggregate func.

① $\sigma_{amount > 12000}(loan)$

② $\Pi_{loannumber}(\sigma_{amount > 12000}(loan))$

③ $\Pi_{cust\_name}(borrower) \cup \Pi_{custn}(depositor)$

Select customer_na.

④ $\Pi_{customer\_name}(\sigma_{branch\_name = 'Bangalore'}((borrower \bowtie loan)))$

borrower.loanno = loan.loanno

If we use natural join common attributes have to be mentioned.

⑤ $\Pi_{customer}(depositor)$ ← n

Select customer name from borrower, loan where loan.loannumber = borrower.loannumber and branch_name = "Bangalore" minus select customer_name from depositor.

① Find those balances that are not the largest

– Rename the account relation as D so that we can compare each account balance w/ all others

② use set diff operator to find those account balances that were not found in the earlier steps.

$\Pi_{balance}(account) - \Pi_{account.balance}(account.balance < d.balance (account \times \rho_d(account)))$

$\Pi_{balance}(account) - (\Pi_{account.balance}(account) - \Pi_{account.balance}(account.balance < d.balance (account \times \rho_d(account))))$

$\Pi_{balance}(account) - (\Pi_{account.balance}(account) - \Pi_{account.balance}(account.balance < d.balance (account \times \rho_d(account))))$

$\max(balance)(account)$

Client name
Part number
Salesman name
Sales order
Item order details

SQRT()
ABS()
~~TIME STAMP~~
CURRENT_TIMESTAMP
~~ROUND()~~ ROUND (15.19, 1) o/p 15.2

SELECT ADD_MONTHS (SYSDATE, 5) FROM DUAL;

SELECT MONTHS_BETWEEN ('02-JUL-01', '02-JAN-01') FROM DUAL;

SELECT TO_CHAR (SYSDATE, 'DD/MM/YYYY') FROM DUAL;

COMM number(4) DEFAULT 0 check (COMM <1500),

Query Processor steps:-

- Query decomposition
  1. Query normalisation
  2. Analysis - Semantic analysis of normalized query
  3. Elimination - the semantically corrected query is simplified by reducing the redundant queries.
  4. rewriting -
- Data localization
- Global Query optimization
- Distributed Query execution.

Baye's Theorem

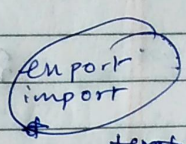
Let $A_1, \ldots, A_n$ be a collection of mutually exclusive and exhaustive events with $P(A_i) > 0$ for $i = 1 \ldots n$. Then for any other event B for which $P(B) > 0$.

$$P(A_K \mid B) = \frac{P(B \mid A_K)\, P(A_K)}{P(B)} = \frac{P(B \mid A_K)\, P(A_K)}{\sum_{i=1}^{n} P(B \mid A_i)\, P(A_i)}$$

$K = 1, \ldots, n$

Pitfalls of database

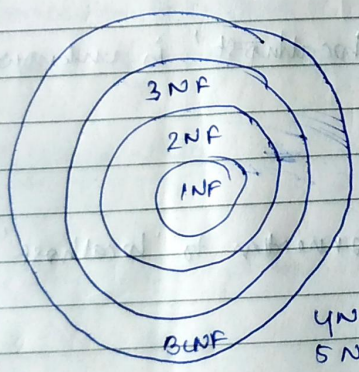
1. redundancy of data
2. Unable to store irrelevant
3. Insert
4. update
5. delete



textual format

Database Normalization. => SQL

Vararray
allows multiple values to be stored



Armstrong axioms.

Problems of normalization.

Functional dependencies

attributes are mathematically related.

roll → name

determined

$A \rightarrow B$

$t[\alpha_1] \neq t[\alpha_2]$

Socket Programming

Q. Write client server programs using Java such that the client will be able to access the most recent date and time at the server.

Ans

```
import java.net.*;
import java.util.*;
import java.io.*;
import java.util.Date;
public class TimeClient {
    p s v m (String args[]) throws IOException
    {
        Socket timeSocket = null;
        ObjectInputStream timeIn = null;
    try {
        timeSocket = new Socket("localhost", 1313);
        timeIn = new ObjectInputStream(timeSocket.getInputStream());
    }
catch (UnknownHostException e)
    {
        sout("Server 'localhost' is unknown");
        System.exit(1);
    }
catch (IOException e) {
sop ln ("Error during connection to 'localhost'");
    System.exit(1);
}
try {
Date ServerDate = (Date) timeIn.readObject();
System.out.println("Current server time: " + serverDate);
}
catch (classNotFoundException e)
{
    sout("Not a Date object returned");
    System.exit(1);
}
timeIn.close();
timeSocket.close();
}
```

OOPs
OS
N/W
DB

```
import java.net.*;
import java.io.*;
import java.util.Date;
```

Photo

Name -
Address -
E-mail -
Mob -
Aim -

Academic Credentials

Degree	Institute	University/Board	CGPA/Percent	Year of passing
BTech			7.2 (upto 6th sem)	2024 (pursuing)
~~10th~~ XII			• as per board • all subs avg.	
X				

Co Curricular Activities

- The name of Project/Paper, _____ (DOI number), Duration -

Projects/Internship.

→ "

Extra-Curricular Activities *after class 10.

Personal Details

Date of Birth -
Languages known -
Hobbies -

Signature
Date

5 marks fill in

1. FD candidate key set
2. defn + identify from schema which is primary, foreig
3. sql
4.

$AB \rightarrow C$

$B \rightarrow C$

A becomes redundant

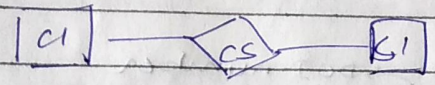
course (cno, cname, sno, sname)

candidate key → cno, sno,

partial dependency

cno → cname

sno → sname



1NF

C1 (cno, cname)

S1 (sno, sname)

CS (cno, sno)

2NF is in 1NF & has no partial dependency