11/11/18

DBMS (8-10 Marks)

Faculty → Ravi Kumar Sir

Syllabus

1) Integrity Constraints && ER Model
2) Normalization
3) Queries (RA, SQL, RC)
4) File Organisation & Indexing
5) Transaction && Concurrency Control

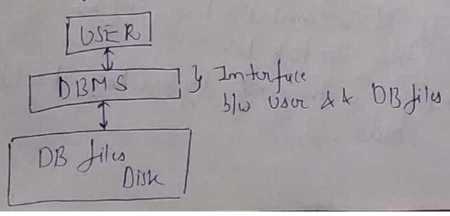
Data Base :-

Collection of related data.

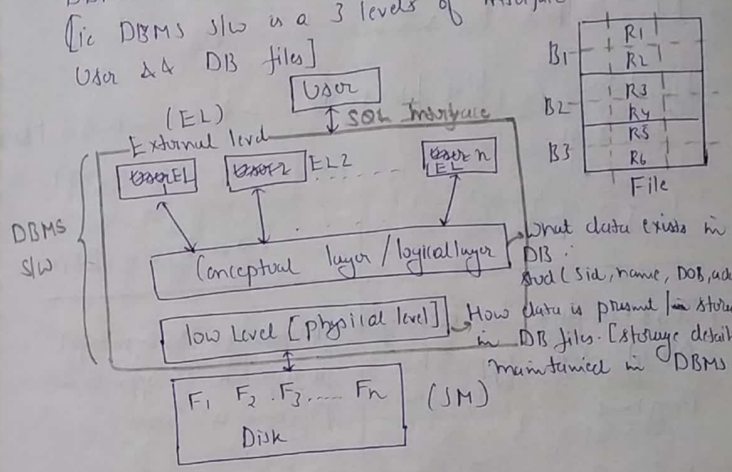
Ex: Set of students info.

DBMS System

It is a s/w used to manage database files in more efficient ways.



DBMS architecture is a 3-level architecture
[i.e DBMS s/w is a 3 levels of interface b/w User && DB files]

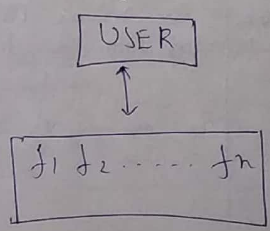


→ Data base managed without DBMS s/w is called Flat file System

Flat File System

(OS File System)

DB files managed by user using OS



→ How user maintains storage details of files.

→ FFS failed to manage large DB

Limitations of flat file systems & Advantages of DBMS
file system

Limitation FFS	Advantage of DBMS
→ Too complex to develop & manage application programs.	→ Because of SQL interface development of application programs is easier.
→ Require more I/O cost to access required data from DB files	→ because of indexing, less ~~access cost~~ I/O cost to access required data from DB files.
→ less degree of concurrency. ~~using DBMS~~	→ More degree of concurrency
→ Too complex to eliminate (or) reduce redundant data	→ because of normalization can manage DB with less redundancy / no redundancy

Integrity Constraints

study under assumptions of RDBMS

[RDBMS: Relational DBMS]

→ RDBMS proposed by R.J. Codd

→ RDBMS also called Codd's data ~~base~~ Model

→ Codd proposed 12 rules to design RDBMS S/W [RDBMS guidelines]

Rule 1 :- Data in DB file must be in Tabular format [set of rows & set of columns]

Ex:- Stud

Relational Instance

Sid	Sname	age
S1	A	20
S2	B	20
S3	A	18
S4	D	19

age ← attribute / field

20 ← tuple / record.

Arity :- no. of attributes of the table is called arity

Cardinality :- no of records of the table.

Relational Schema :- definition (or) structure of the DB table

$\boxed{\text{Stud (Sid, Sname, age)}}$

abstract of DB table

~~bs~~ Relational Instance :- Record set of the DB table
(Snapshot)

Rule 2 [RDBMS , Guideline]

No 2 records of the DB table must be same.
(ie one or more attribute value must differ for any 2 records)

Candidate key :- minimal set of attributes used to differentiate all records of the DB table uniquely

Ex: Stud (Sid, Sname, age)   Candidate key :- Sid
(Sid, Sname) → Not a Ck bcz not minimal

Ex: Enroll (Sid, Cid, fee) i) Stud can enroll in many courses (and) Course can be enrolled by many students.

S1	C1	5000
S1	C2	4000
S3	C2	3000
S2	C2	4000

Cand. key = (Sid, Cid)

ii) Stud can enroll in at most one course, a course can be ~~enroll~~ enrolled by multiple students

S1	C1
S2	C1
S3	C2
S4	C2

Cand. key = (Sid)

Ex: R

A	B	C
4	7	6
4	7	5
4	5	6
4	5	5
6	7	6

Cand. key → (ABC)

ABC cand key of R only if ABC should differentiate records uniquely && no proper subset of ABC can differentiate records uniquely

Ex:

emp

cid	Cname	DOB	panID	PP no.	ifsc code	A/c no
e1	A			X10	SB101	101
e2	Null			X 2	SB 101	102
e3	A			NULL	IC101	101
e4	B			X5	IC101	102
e5	Null			NULL	SB102	103

Cand. keys { eid, $\underbrace{\text{panID, ppno, \{ifsc code A/c no\}}}_{\text{alternative Keys.}}$ }

eid ↓ primary key

Null : denote unknown value / Un existed

Primary key :- Any one candidate key of RDBMS table whose field values must be "not NULL"

For any relation at most 1 primary key is allowed and Primary key fields not allowed NULL values.

Alternative keys : All candidate keys of the relations except Primary key are alternative keys.

Alternative key fields are allowed to be "Null".

```
CREATE TABLE Emp
( eid, varchar (10) PRIMARY KEY,
  ename varchar (30),
  DOB date,
  PanID varchar (10) UNIQUE NOT NULL,
  PPno varchar (15) UNIQUE,
  ifsc code varchar (6),
  A/c no integer (10), UNIQUE (ifsccode, A/c no)
);
```

Pass port no → PPno

Prime attribute [Key Attribute]

Attribute which belongs to some ~~can~~ (at least one) candidate key of relational schema.

Ex: {eid, PanId, PPno, ifscode, A/c no}

Prime attribute set of emp

Attributes not related to any candidate key is called non-prime attribute

Non-prime attribute : (ename, DOB)
set of emp

RDBMS Constraint :

At least one Cand. key of RDBMS table whose field values must "not be Null". [(or) "NOT NULL"]

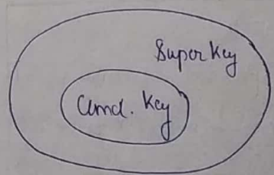
Super key (Super set of Candidate key)

Set of attributes which can differentiate records of relation uniquely. ~~is called~~

Stud (Sid, Sname, age) minimal Super key [candidate key]

Sid : Cand. key

Set of Super keys : { Sid, Sid Sname, Sid age, SidSnameage }



Q R(A, B, C, D)

How many Super keys possible if {A} is Cand key ?

Sol: $\underset{1}{A}\ \underset{2}{\_}\ \underset{2}{\_}\ \underset{2}{\_} = 8$ choice

Super key :-

A { Any subset of B, C, D }

A	AB	ABC	ABCD
	AC	ACD	
	AD	ABD	

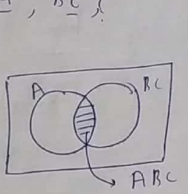
Q R(ABCDE)

How many Super keys possible if A, BC ?

Sol: Super key $= 2^4 + 2^3 - (2^2)$

$= 16 + 8 - 4$

$= 20$



Super key $= S(A) + S(BC) - S(A \cap BC)$

$= 16 + 8 - 4$

$= 20$

Q R(ABCDEF)

How many Super keys possible in R if A, BC, CD are cand. keys ?

S.k's by
DEF
↓
ABC [DEF]
↑

Sol: Super key : $S(A) + S(BC) + S(CD) - S(A \cap BC) - S(BC \cap CD) - S(A \cap CD) + S(A \cap BC \cap CD)$

$= 32 + 16 + 16 - 8 - 8 - 8 + 4$

$= 64 - 24 + 4$

$= 44$

Q R($A_1, A_2, A_3, \ldots A_n$) How many super keys possible in relation R if {A_1, A_2A_3, $A_3A_4A_5$ ..} ?

Sol: Super key $= S(A_1) + S(A_2A_3) + S(A_3A_4A_5) - S(A_1 \cap A_2A_3) - S(A_2A_3 \cap A_3A_4A_5) - S(A_1 \cap A_3A_4A_5) + S(A_1 \cap A_2A_3 \cap A_3A_4A_5)$

$= 2^{n-1} + 2^{n-2} + 2^{n-3} - 2^{n-3} - 2^{n-4} - 2^{n-4} + 2^{n-5}$

$= 2^n \left(\frac{1}{2} + \frac{1}{4} + \cancel{\frac{1}{8}} - \cancel{\frac{1}{8}} - \frac{1}{16} - \frac{1}{16} + \frac{1}{32} \right)$

$= 2^n \left(\frac{1}{2} + \frac{1}{4} - \frac{1}{8} + \frac{1}{32} \right)$

$2^n \left(\frac{16 + 8 - 4 + 1}{32} \right)$

$= 2^n \left(\frac{21}{32} \right)$

$= 2^{n-5} (21)$.

Q R(A1, A2, A3, ... An)

How many Super Keys possible in R ?

Sol: $2^n - 1$ Super key possible in R if each attribute of R is cand. key.

Foreign key [Referential key]

→ Used to relate data b/w tables
defined over 2 tables
→ Referenced relation
→ Referencing relation

Stud (Sid, Sname, login)   Enroll (StudID, CourseID, fee)

Foreign key

Sid	login
S1	a1
S2	a2
S3	a3
S4	a4
S5	a5

referenced relation

Primary key (P.K)   Alternative key

referencing relation

StudID	CourseID
S1	C1
S1	C2
S2	C2
S3	C2
S3	C4
S5	C1
S7	C1

invalid
entries should belong to Sids of Stud

```
CREATE TABLE Enroll
(
  StuID     varchar(10),
  CourseID  varchar(10),
  fee       integer(5),
  primary key (StudID, CourseID),
  FOREIGN KEY (StudID) REFERENCES Stud (Sid)
  ON DELETE CASCADE ON UPDATE CASCADE
);
```

(Imp) If nothing is specified by default it will refer PK of Stud.

Def :- Foreign key is set of attributes references to primary key or alternative key of same relation or some other relation.

X ← Y

primary key/alternative key   foreign key

Ex. Emp (eid, ename, SupID)

PK   also must be emp.

eid	ename	SupID
e1	A	Null
e2	A	e1
e3	B	e1
e4	A	e2
e5	C	Null
e6	A	e8

← invalid entry

PK   AK   FK

R Referenced relation

A	B	C
2	15	2
3	Null	2
4	16	4
5	Null	20
6	17	20
7	18	Null

S Referencing relation

D	E
15	e1
15	e2
16	e3
Null	e4
Null	e5
18	e6

→ F.k allows Null values / FK

→ FK attributes values can be NULL

→ NULL values of FK field records not related to referenced table. referencing table → which contains F.K

NOTE: Each record of referencing relation table related to at most one record of referenced table.

Each record of referenced relation related to many (0 or more) records of referencing relation.

[1 : M] Mapping b/w reference relations

Referential (f.k) Integrity Constraints (V.V Imp)

Referenced Relation

a) Insertion :- No violation

b) Deletion :- may cause violation

User query
delete from stud where Sid = "S2";

To preserve integrity constraint

a) ON DELETE NO ACTION ← by default if no constraint specified by user.

~~or~~ deletion of referenced record is restricted

ii) ON DELETE CASCADE
(~~deletion of~~ deletes both referencing record && referenced record)

iii) ON DELETE SET NULL.
if (f.k) fields values allowed to set null then allowed to delete referenced record && sets NULL in related f.k values.

c) Updation of Referenced Key Values :-

may cause violation.

i) ON UPDATE NO ACTION
updation of referenced key not allowed

ii) ON UPDATE CASCADE
updates both referenced record && referencing record

iii) ON UPDATE SET NULL.
update referenced record && set referencing record to NULL.

2) Referencing Relation

a) Insertion
may cause violation

b) deletion → No voilation

c) Updation → may cause voilation

If voilation occurs bcz of insertion (or) updation of referencing table, then restrict updation / insertion

Normalization (Schema Refinement)

→ Used to reduce / eliminate redundancy in Relations.

→ Redundancy can occur in relation if 2 or more independent relations stored in single relation :

1. Sid → Sname, age
2. Cid → Cname, Instructor
3. Sid Cid → fee

Sid	Sname	age	Cid	Cname	Inst	fee
S1	A	20	C1	DBMS	Korth	5000
S2	A	20	C1	DBMS	Korth	4000
S3	B	22	C1	DBMS	Korth	5000
S3	B	22	C2	OS	Galvin	6000
S3	B	22	C3	OS	Hayes	4000

redundancy   redundancy   Sid Cid → primary key

Problems bcz of redundancy

1) Redundancy causes DB anamalies

causes inconsistency

i) Insertion Anamaly — Student record can be inserted only if he has enrolled some course.
To insert some data, forced to insert other independent data

ii) Deletion Anamaly — delete from Stud where Cname=OS; OS taught by Hayes also deleted.
To delete some data also loose other independent data

iii) Updation Anamaly
Some redundant copies updated, while some failed to update which cause inconsistency.

Normalization of DB

decompose relations into two or more sub relations to eliminate / reduce redundancy && anamalies.

R1 (Sid, Sname, age)

Sid	Sname	age
S1	A	20
S2	A	20
S3	B	22

R2 (Sid, Cid, fee)

Sid	Cid	fee
S1	C1	–
S2	C1	–
S3	C1	–
S3	C2	–
S3	C3	–

R3 (Cid, Cname, Inst)

Cid	Cname	Inst
C1	DB	
C2	OS	
C3	CO	

Normalized DB design
[no redundancy && No anamalies]

Functional Dependency

X, Y some attribute set over R and
t_1, t_2 any tuples of R.

X determines Y

R	X	Y	. . . . . .
	2	8	
	2	8	
	3	6	
	4	6	
	5	7	
	2	8	
	4	6	

X → Y
For each "X" value there must be only one "Y" value.

X → Y implied in Relation R

ie if for any two tuples X value is same then Y value must also be same

X → Y FD exists in Relation R
→ if $t_1.x = t_2.x$ then $t_1.y = t_2.y$

A	B	C
4	7	5
4	7	7
6	5	5
6	5	7
6	4	5

A→B ✗ B→A ✓
A→C ✗ C→A ✗
B→C ✗
AB→C ✗
BC→A ✓
AC→B ✗

FD set for given instance
{ B→A , BC→A

A → B ✗
Not implied in R

1) Trivial FD

L.H.S attributes are superset of R.H.S attributes

X→Y FD trivial FD iff $X \supseteq Y$

Ex: Sid → Sid
Sid Sname → Sname
Cid → Cid
Sid Sname → Sid Sname

	Sid	Sname	Cid
t_1	S1	A	C1
t_2	S1	A	C2
	S2	B	C2
	S3	B	C3

Every trivial FD always in Relational Schema
Trivial FD's are default FD's.

2) Non-Trivial FD

X → Y FD is non-trivial FD only if no common attribute in X, Y attribute sets.

Sid → Sname
Sid Cid → Sname } Non-Trivial FD's.

3) Semi Non Trivial FD

Combination of both trivial && non-trivial FD's.

i) Sid → Sid Sname   X → YZ ⇒ X → Y, X → Z
[Sid → Sid , Sid → Sname]

ii) Sid Cid → Cid Sname
[Sid Cid → Cid , Sid Cid → Sname]

Scanned by CamScanner

Armstrong Rules over FD's

X, Y, Z some set of attributes over relation R

1. Reflexivity

   $X \to X$ [always true]

2. Transitivity

   if $X \to Y$ & $Y \to Z$ then $X \to Z$

3. Augmentation

   if $X \to Y$ ~~&~~ Then $XZ \to YZ$

   Ex:- Sid → Sname ⇒ Sid Cid → ~~Cid~~ Sname Cid

4. Split Rule

   if $X \to YZ$ Then $X \to Y$ & $X \to Z$

5. Merge Rule (Union rule)

   if 2 FD's ~~with Same~~ with same left side

   $X \to Y$ & $X \to Z$ then $X \to YZ$

Attribute Closure (X^+)

$X^+ = \{$ Set of all attributes determined by X $\}$

Given Set of FD's

$\{A \to B,\ D \to E,\ B \to C,\ AC \to D,\ EF \to G\}$

$(A)^+ = \{A, B, C, D, E\}$ means $A \to ABCDE$

$(B)^+ = \{B, C\}$

$(C)^+ = \{C\}$

$(D)^+ = \{D, E\}$

$(AF)^+ = \{A, F, B, C, D, E, G\}$ means $AF \to AFBCDEG$ P.K

$(BD)^+ = \{B, D, C, E\}$ means $BD \to BCDE$

$(BF)^+ = \{B, C, F\}$

Super Key:-

X is super key of relational schema R iff

X^+ should determine all attributes of relation R.

R(ABCDE) $\{AB \to C,\ C \to D,\ B \to E\}$

$(AB)^+ = \{A, B, C, D, E\}$

ie AB is SK hence any superset of AB is will be Sk

$(BC)^+ = \{B, C, D, E\}$

not SK

*** Candidate key (minimal Super key) (V.V.V Important)

X is candidate key of relational Schema R

iff ① X must be Super key of Relational Schema R

(and)

② No proper Subset of X is superkey of relational Schema R. ie X is minimal super key

$\forall Y \subset X$ such that

$(Y)^+ = \{$ not all attributes of $R\}$

Q Find Candidate keys of given relations ?

① $R(ABC)$ $\{A \to B\}$

$A^+ = AB$ $C^+ = C$ $AC^+ = ABC$ ← C.k

$B^+ = B$ $AB = AB$

② $R(ABCDEF)$ $\{A \to E, F \to B, C \to A\}$

$CDF^+ = \{CDFAEB\}$

Cand key : CDF^+

③ $R(A B' C' D' E' F')$

$\{AB \to C, C \to D, CD \to EF, F \to B\}$ ← prime

$A^+ = \{A\}$

$AB^+ = \{A, B, C, D, E, F\}$

Cand key : AB^+

for any non-trivial FD, determiner is prime attribute.

*** if $X \to$ Prime then more than one ck in R

$AC^+ = \{A, C, D, E, F, B\}$

$AD^+ = \{A, D\}$

$AE^+ = \{A, E\}$

$AF^+ = \{A, F, B, C, D, E, F\}$

prime attributes $\{$ ~~A, B, F~~ $\}$ AB, AF

NOTE: to get next cand key replace B in AB by L.H.S of $F \to B$ bcz $F \to B$ is of the form $X \to$ prime

we have $CD \to EF \Rightarrow CD \to E$, $CD \to F$

replace F by CD in AF

$(ACD)^+ = \{A, C, D, E, F, B\}$ $AC^+ = \{ACDEFB\}$

~~prime attributes~~ { but here AC^+ determines all attributes hence ACD is not minimal super key ie ACD is not candidate key.

Prime attributes $\{AB, AF, AC\}$ are candidate keys

④ R(ABCD) = { AB → CD, C → B, D → A }

prime attribute AB, BD, AC, CD

$A^+ = \{A\}$ $C^+ = \{C, B\}$

$B^+ = \{B\}$ $D^+ = \{D, A\}$

$AB^+ = ABCD$ ✓ $AD^+ = AD$ ✗

$AC^+ = ACBD$ ✓ $BC^+ = BC$

~~ABD = ABCD~~ $BD^+ = BDAC$ ✓

~~ACD = A~~ $CD^+ = CDAB$ ✓

Pk → AB, AC, BD, CD

⑤ R(ABCDEF)

{AB → C, C → D, CD → AE, DE → F, EF → B}

ABCDEF

$C^+ = CDAEFB$ ✓

$AB^+ = ABCDEF$ ✓

prime attributes {C, AB, } EF

$\boxed{\begin{array}{l} AD^+ = AD \\ AE = AE \\ AF = \end{array}}$ $AEF^+ \to AEFBCD$ ✓ $ADE \to ADEFBCD$ ✓

~~EF⁺ = EFB~~

⑥ R(ABCDEF) C → D & C → E

{AB → C, C → DE, E → F, F → A}

prime attribute B, AB, BF, BE, B-----

$BA^+ \to ABCDEF$ ✓

$BF \to BFACDE$ ✓

$BE^+ \to BEFACD$ ✓

~~BCD⁺ = CDEFAB~~

$BC^+ = BCDEFA$ ✓

Finding ~~Calculating~~ Candidate key using Closure method is NP-complete problem

R(A_1 A_2 ..... A_n)

$A_1^+ =$	$A_1A_2^+$	$A_1A_2A_3^+$ ...	......	$A_1A_2 \ldots A_n^+$
$A_2^+ =$	$A_1A_3^+$			
$A_3^+ =$	$A_1A_4^+$			
⋮ =	⋮		......	
A_n^+				
$^nC_1$	$+ \, ^nC_2$	$+ \, ^nC_3$	$+ \ldots$	$+ \, ^nC_n$

∴ In worst case

$^nC_1 + {^nC_2} + {^nC_3} + {^nC_4} + \ldots + {^nC_n} = 2^n$

Hence 2^n closure required to compute to find cand key of R.

→ Finding Ck of R is NPC problem.

Q R(ABCDEF) Ans - 7

{AB → C, C → A, BC → D, ACD → B, BE → C,

EC → FA, CF → BD, D → E}

prime attribute EB, ECF, EC

Sol: ~~EA⁺ = EA~~ $EB^+ = EBCADF$ ✓

~~ECF⁺ = ECFBDAF~~ ~~EAB⁺ → EABCDF~~

~~EACD⁺ = EACDBF~~ ~~ECD⁺ =~~

$ED^+ = EDF$

$EC^+ = ECFABD$ ✓

~~F~~ $AB \to C$, $C \to A$, $BC \to D$, ~~$AC \to D$~~, $CF \to B$
$ACD \to B$, $BE \to C$, $EC \to FA$, $CF \to BD$, $D \to E$

EB, EC,

$EB^+ = EBCADF$ ✓
$EC^+ = ECAFBD$ ✓
~~$ED^+ = ED$~~
~~EAB~~ $AB^+ = ABCDEF$ ✓
$EF^+ = EF$ ✗ $BD^+ = BDECFA$ ✓
$BC^+ = BCDEFA$ ✓ $CD^+ = CDAEFB$ ✓
~~$AD^+ = ADE$~~
~~CF~~ $CF^+ =$

{EB, EC. AB, BC
DB, DC,
E, B, C

Ans: AB, BC, BE,
BD, CF, EC,
CD ✓
∴ E is prime

~~Q~~ R(ABCDEF) Ans:- Cand key {AB, BC, BE, BD, CF, EC, CD}

No one is determining E, so E must be present in cand. key

$E^+ = E$ $EA^+ = EA$ $EB^+ = EBCADF$ ✓
$EC^+ = ECAFBD$ ✓ $ED^+ = ED$ $EF^+ = EF$

prime attribute {EB, EC}, now see where E is present at R.H.S, and replace its corresponding LHS to get next key {D→E}

$DB^+ = DBECAF$ ✓ $DC^+ = DCAEFB$ ✓

prime attributes {EB, EC, DB, DC}, similarly check for every prime attribute where it occurs at R.H.S and replace it by corresponding LHS to get cand keys

Check for B {ACD → B, CF → B} check for all possible cand keys

EACD, ECF, DACD, DCF from EACD, ECF, DACD, DCF to get remaining keys

EA^+ ✗, EC ✓, ED ✗, $AC^+ = AC$ ✗, $AD^+ = ADE$ ✗, CD ✓
ECF ⇒ EC ✓, EF ✗, $CF^+ = CAFBDE$ ✓ ← new cand. key.

Membership Test :-

X → Y FD member of FD set (F)
X → Y must determine "Y"
iff X^+ in FD set (F)

F = { . . . . }
if X^+ = { . . . . Y }
X → Y is member of FD set (F)

otherwise
X → Y is not member of FD set (F)

Q {AB → C, BC → D, D → E, CD → F}.

Test FD's members of FD set or not?

i) AB → F ii) CD → E iii) BC → A

Sol: i) $AB^+ = ABCDEF$ ✓ AB → F member of FD set
ii) $CD^+ = CDEF$ ✓ CD → E member of ~~FD~~ set
iii) $BC^+ = BCDEF$ ✗ BC → A not member of FD set

Equality of FD Sets :-

F & G two FD sets equal iff

(1) F cover G means $F \supseteq G$

i.e every FD of G set member of F set.

(and)

(2) G covers F means $G \supseteq F$

i.e every FD of F set must be member of G set.

F covers G	(and) G covers F	
True	True	$F \equiv G$
True	False	$F \supset G$
False	True	$F \subset G$
False	False	F & G not comparable

Q(1) $F = \{A \to B,\ B \to C,\ C \to A\}$

$G = \{A \to BC,\ B \to AC,\ AB \to C,\ BC \to A\}$

Which is true?

a) $F \subset G$ b) $F \supset G$ c) $F \equiv G$ d) None

Sol: F covers G

$A \to BC$ $A^+ = ABC$ ✓

$B \to AC$ $B^+ = BCA$ ✓

$AB \to C$ $AB^+ = ABC$ ✓

$BC \to A$ $BC^+ = BCA$ ✓

G covers F

$A \to B$ $A^+ = ABC$ ✓

$B \to C$ $B^+ = BAC$ ✓

$C \to A$ $C^+ = C$ ✗

(Q2) $F = \{A \to BCDEF,\ BC \to ADEF,\ D \to E,\ B \to F\}$

$G = \{A \to BC,\ BC \to AD,\ D \to E,\ B \to F\}$

a) $F \subset G$ b) $F \supset G$ c) $F = G$ d) none

Sol: F covers G

$A \to BC$ ✓ A^+ in F $A^+ = ABCDEF$

$BC \to AD$ ✓ $BC^+ = BCADEF$

$D \to E$ ✓

$B \to F$ ✓

G covers F

$A \to BCDEF$ ✓ A^+ in G $A^+ = ABCDEF$

$BC \to ADEF$ ✓ $BC^+ \to BCADEF$

$D \to E$ ✓

$B \to F$ ✓

Canonical Cover [Minimal Cover]

Canonical cover of given FD set (F) is minimal set of FD's which are logically equal to F

Ex: $F = \{A \to B,\ B \to C,\ A \to C,\ AB \to C\}$

F_m: $A \to C$ ✗ bcz $A \to B$ $B \to C$

$AB \to C$ ✗ bcz $B \to C \Rightarrow AB \to AC$ → $AB \to A$, $AB \to C$

$F_m = \{A \to B,\ B \to C\}$

$\{F \equiv F_m\}$

Scanned by CamScanner

12/11/18

Procedure

Given FD set F

I Remove extraneous attributes from determinants of FD set F.

$\underbrace{X}_{\text{determinant}} \rightarrow \underbrace{Y}_{\text{determined}}$

Extraneous attributes :-

$\{WXY \rightarrow Z,\ W \rightarrow X\}$

$\{WY \rightarrow Z,\ W \rightarrow X\}$

Find extraneous attribute

1. $\{AB \rightarrow C,\ A \rightarrow B\} = \{A \rightarrow C,\ A \rightarrow B\}$
   extraneous

2. $\{ABCD \rightarrow E,\ \underbrace{A \rightarrow F,\ F \rightarrow CD}_{A \rightarrow CD}\}$
   extraneous

   $\{AB \rightarrow E,\ A \rightarrow F,\ F \rightarrow CD\}$

3. $\{AB \rightarrow C,\ A \rightarrow B,\ B \rightarrow A\}$

   $\{B \rightarrow C,\ A \rightarrow B,\ B \rightarrow A\}$ $\{A \rightarrow C,\ A \rightarrow B,\ B \rightarrow A\}$

   either A determine B or B determine A

4. $\{ABCD \rightarrow E,\ A \rightarrow D,\ B \rightarrow F,\ F \rightarrow C\}$

   $\{AB \rightarrow E,\ A \rightarrow D,\ B \rightarrow F,\ F \rightarrow C\}$

   bcz $\underbrace{B \rightarrow F,\ F \rightarrow C}_{B \rightarrow C}$ & $A \rightarrow D$

5. $\{ABCD \rightarrow E,\ A \rightarrow D,\ BD \rightarrow F,\ F \rightarrow C\}$

   $\{AB \rightarrow E,\ A \rightarrow D,\ B \rightarrow F,\ F \rightarrow C\}$

6. $\{ABCD \rightarrow E,\ AB \rightarrow C,\ BC \rightarrow A\}$ either A or C extraneous attribute

   $AB^+ = ABC$ $CD^+ = CD$
   $AC^+ = AC$ $ABC^+ = ABC$
   $AD^+ = AD$ $ABD^+ = ABCD$ $ABD \rightarrow C$
   $BC^+ = BCA$ $ACD^+ = ACD$
   $BD^+ = BD$ $BCD^+ = BCDA$ $BCD \rightarrow A$

II) Remove redundant FD's from result of step I

$X \rightarrow Y$ FD from FD set F is redundant iff $X \rightarrow Y$ must be member of $\{F - \{X \rightarrow Y\}\}$ FD set.

Find redundant FD's ?

1. $\{A \rightarrow B,\ B \rightarrow C,\ A \rightarrow C\}$ redundant

   bcz without $A \rightarrow C$ FD A^+ will determine C

   $A^+ = ABC$

2. $\{AB \rightarrow C,\ AB \rightarrow D,\ C \rightarrow D\}$
   redundant

   $AB^+ = ABCD$

   ABD
   ABCD
   ABC

Scanned by CamScanner

3. $\{ \cancel{AB} \rightarrow A \rightarrow B,\ AC \rightarrow B \}$

$A \rightarrow B$ ✓ $\quad A^+ = A$

$AC \rightarrow B \quad AC^+ = ACB$

this is redundant.

4. $\{ A \rightarrow BC,\ B \rightarrow C,\ C \rightarrow B \}$

$\{ A \rightarrow B,\ A \rightarrow C,\ B \rightarrow C,\ C \rightarrow B \}$

either of them redundant.

a) $\{ A \rightarrow B,\ B \rightarrow C,\ C \rightarrow B \}$

b) $\{ A \rightarrow C,\ B \rightarrow C,\ C \rightarrow B \}$

Q Find minimal cover of given FD set.

$\{ A \rightarrow BCDEF,\ BC \rightarrow ADEF,\ B \rightarrow F,\ D \rightarrow E,\ AD \rightarrow E,\ AE \rightarrow F \}$

Sol: Remove extraneous attributes

BF $\quad B^+ = BF \quad C^+ = C$

AD $\quad A^+ = ABCDEF \quad D^+ = DE$

$A\cancel{D} \rightarrow E$ extraneous

AE $\quad A^+ = ABCDEF \quad E^+ = E$

extraneous

$XYZ \rightarrow P$

For Y to be extraneous

$X^+ \rightarrow Y$ (or) $Z^+ \rightarrow Y$

(or) $XZ^+ \rightarrow Y$

L.H.S

NOTE: Extraneous attribute possible only when ~~RHS~~ has more than one attributes.

Removal of Extraneous attribute over determinants

extraneous attributes possible

$A \rightarrow BCDEF$

$BC \rightarrow ADEF$

$B \rightarrow F$

$D \rightarrow E$

$A\cancel{D} \rightarrow E \qquad A \rightarrow E$

$A\cancel{E} \rightarrow F \qquad A \rightarrow F$

Removal of redundant FD's. $\left.\begin{array}{l} A \rightarrow BCDEF \\ BC \rightarrow ADEF \end{array}\right\}$ split

$A \rightarrow B$	$BC \rightarrow A$	$B \rightarrow F$
$A \rightarrow C$	$BC \rightarrow D$	$D \rightarrow E$
$A \rightarrow D$ X	$BC \rightarrow E$ X	
$A \rightarrow E$ X	$BC \rightarrow F$ X	
$A \rightarrow F$ X		

Temporarily hide $A \rightarrow B$ and check whether A^+ determines B from remaining FD's. If it does then $A \rightarrow B$ is redundant.

$F_m = \{ A \rightarrow BC,\ \cancel{A \rightarrow} BC \rightarrow AD,\ B \rightarrow F,\ D \rightarrow E \}$

NOTE: Minimal cover of FD set may not be unique but all minimal covers are logically equal.

$\{ F_{m_1} = F_{m_2} = F \}$

[also equal to F]

NOTE: To find minimal cover of F, check apply equality test b/w F and F_m

Properties of Decomposition

1. Loss Less Join decomposition
2. Dependency Preserving Decomposition

Loss Less Join Decomposition

Relational Schema R with instance "r" (using reused set) decomposed into R1, R2, R3, ... Rn sub relations.

⋈ : Natural Join

In General
$\{R_1 \bowtie R_2 \bowtie R_3 \dots \bowtie R_n\} \supseteq R$

If $\{R_1 \bowtie R_2 \bowtie R_3 \dots \bowtie R_n\} \equiv R$
Then decomposition is lossless join decomposition

If $\{R_1 \bowtie R_2 \bowtie R_3 \dots \bowtie R_n\} \supset R$
Then lossy Join decomposition

R → R_1, R_2

1) $R_1 \bowtie R_2 \equiv R$ → LLJ
2) $R_1 \bowtie R_2 \supset R$ → LJ
3) $R_1 \bowtie R_2 \subset R$ Never possible

{ Sid → Sname }
Sid Cid ⇒ Cand Key

R

Sid	Sname	Cid
S1	A	C1
S1	A	C2
S2	B	C2
S3	B	C3

r is decomposed into

Sid → Sname
Sid : Cand Key

R1

Sid	Sname
S1	A
S2	B
S3	B

R2

Sid	Cid
S1	C1
S1	C2
S2	C2
S3	C3

$R_1 \bowtie R_2$

Sid	Sname	Cid
S1	A	C1
S1	A	C2
S2	B	C2
S3	B	C3

loss Less Join Decomposition

ii) decomposed into

R_1

Sid → Sname

Sid cand key

Sid	Sname
S1	A
S2	B
S3	B

R_2

Sname Cid

cand key

Sname	~~Course~~ Cid
A	C1
A	C2
B	C2
B	C3

$R_1 \bowtie R_2$

Sid	Sname	Cid
S1	A	C1
S1	A	C2
S2	B	C2
S2	B	C3
S3	B	C2
S3	B	C3

extra records
Spurious tuples
[causes inconsistency]

lossy Join $R_1 \bowtie R_2 \supset R$

Drawback

→ lossy join causes spurious tuples, which causes inconsistency.

→ If while decomposition of R into sub relations common attribute is not unique, then it causes lossy join.

NOTE: Testing of Loss less join using FD set

Relational schema R with $\boxed{\text{FD set } F}$, decomposed into R_1, R_2.

decomposition is loss less join iff

① $R_1 \cup R_2 = R$

(and)

② $R_1 \cap R_2 \rightarrow R_1$ (or) $R_1 \cap R_2 \rightarrow R_2$

ie ~~(B)~~ common attribute of both the relation must be key for either of the relations.

Q

R (ABCDE)

{AB → C, C → D, B → E}

decomposed into

i) {ABC, CD}   ~~ii) {ABC, DE}~~   ~~iii) {AB~~

$AB^+ = ABCDE$
C.K

ABCDE

ABC
AB → C

CD   C → D

Loss less

i) { ABC, CD }

ii) { ABC, DE }

iii) { ABC, CDE }

iv) { ABCD, BE }

Sol: i) ABCDE → ABC, CD

E is missing hence lossy

ii) ABCDE → ABC, DE

no common attribute lossy

iii) ABCDE → ABC, CDE

$AB^+ = ABCDE$

AB → C

C not key

C → D

not determine E

lossy

B → E not preserved

iv) ABCDE → ABCD, BE

$AB^+ = ABCDE$

B → E

AD → C

C → D

Loss less join

Q R (ABCDEFGHIJ)

{ AB → C, A → DE, B → F, F → GH, D → IJ }

i) { DIJ, ADE, FGH, BF, ABC }

ABCDEFGHIJ

~~$AB^+ = ABCDE$~~

if $R_i \cap R_j \longrightarrow R_i / R_j$ — if common attribute is key for either of sub relation

then $R_i \bowtie R_j$ [LLJ decomposition]

~~AB~~ i) { DIJ, ADE, FGH, BF, ABC }

$AB^+ = ABCD$ ~~AB~~ $EFGHIJ$

$A^+ = ADEIJ$

$B^+ =$

ABCDEFGHIJ → DIJ, ADE, FGH, BF, ABC

$D^+ = DIJ$

$F^+ = GHF$

DIJ ⋈ ADE = ADEIJ

FGH ⋈ BF = BFGH

ADEIJ ⋈ ABC = ~~ADL~~ ABCDEIJ

ABCDEIJ ⋈ BFGH = ABCDEFGHIJ

lossless join

ii) { DIJ, ACE, FGH, BF, ACD }

D → IJ

F → GH

B → F

all FD's are not preserved

$D^+ = DIJ$

$AC^+ = ACDEIJ$

DIJ ACE FGH BF ACD

FGH ⋈ BF = BFGH

DIJ ⋈ ACD = ACDIJ

ACDIJ ⋈ ACE = ACDEIJ

ACDEIJ ⋈ BFGH → not possible bcz no common attribute

Dependency Preserving Decomposition

Relational Schema "R" with FD set "F" ~~decomposition~~ decomposed into $R_1 R_2 \ldots R_n$ sub relations.

Assume $F_1, F_2, F_3, \ldots F_n$ FD sets for each sub relations

In general

$\{F_1 \cup F_2 \cup \ldots \cup F_n\} \subseteq F$

if $\{F_1 \cup F_2 \cup F_3 \ldots \cup F_n\} = F$

Then DP decomposition

if $\{F_1 \cup F_2 \cup F_3 \ldots \cup F_n\} \subset F$

Then not DP decomposition

R with FD set $F = \{\ldots\ldots\}$

R_1 R_2

1. $\{F_1 \cup F_2\} = F$ DP decomp
2. $\{F_1 \cup F_2\} \subset F$ not DP decomp
3. $\{F_1 \cup F_2\} \supset F$

Never possible

Q $R(ABCD)$

$\{AB \to CD,\ D \to A\}$

decomposed into $\{BCD, AD, ABC\}$

Sol $\{BCD, AD\cancel{A}, ABC\}$

$D \to A$ $AB \to C$

$R_1(BCD)$	$R_2(AD)$	$R_3(ABC)$
(F_1)	(F_2)	(F_3)

To compute FD set of $R_1(BCD)$

$F = \{BD \to C\}$ B^+, C^+, D^+, BC^+, CD^+, BD^+

not BCD bcz $BCD \to BCD$ is trivial.

We only test for

D^+

DB^+

DC^+

bcz $D \to A$ is in FD set hence D with other attribute may determine all the attributes of R_1

$D^+ = DA$

$DB^+ = DABC$ ✓ ~~AB~~ $DB \to C$

$DC^+ = DAC$

Hence FD set of $R_1(BCD)$ will have $BD \to C$ only

R_2 (AD)
$A^+ = A$
$D^+ = AD$

$F = \{ D \to A \}$

R_3 (ABC)
$A^+ = A$ $\quad C^+ = C$
$B^+ = B$ $\quad AB^+ = ABCD$
$AC^+ = AC$ $\quad BC^+ = BC$

$F = \{ AB \to C \}$

FD set for $F_1 \cup F_2 \cup F_3$

$\{ BD \to C, D \to A, AB \to C \}$

$AD \to CD$ +
$D \to A$ ✓
~~$F_1 \cup F_2 \cup F_3 \not\subseteq F$~~
Hence FD not preserved

NOTE:

$F_1 \cup F_2 \cup F_3 \subseteq F$

Check $\{F_1 \cup F_2 \cup F_3\}$ covers F or not.

if every FD of F present in $\{F_1 \cup F_2 \cup F_3\}$ then FD preserving decomposition

If every FD of F not present in $\{F_1 \cup F_2 \cup F_3\}$ then ~~FD~~ not FD preserving decomposition

Q R(ABCDE)
$\{ A \to B, B \to C, C \to D, D \to BE \}$
decomposition to
(AB, BC, CD, DE)

Sol: $A^+ = AB$ $\quad B^+ = BCDE$ $\quad C^+ = CDBE$ $\quad C^+ = CDBE$ $\quad D^+ = DBEC$

AB	BC	CD	DE
$A \to B$	$B \to C$	$C \to D$	$D \to E$
	$C \to D$	$D \to C$	

$F_1 \cup F_2 \cup F_3 \cup F_4 = \{ A \to B, B \to C, C \to D, D \to E \}$

$\{F_1 \cup F_2 \cup F_3 \cup F_4\} \subseteq F$

$A \to B$ ✓ $\quad C \to D$ ✓
$B \to C$ ✓ $\quad D \to BE$ ✓

$\therefore \{F_1 \cup F_2 \cup F_3 \cup F_4\}$ covers F

Hence FD preserving decomposition.

Q $R(ABCDE) = \{ AB \to C, C \to D, D \to E, E \to A \}$
decomposed to $\{ABC, CD, DE\}$

Sol:

ABC	CD	DE
$C^+ = CDEA$	$C^+ = CDEA$	$D^+ = DEA$
$AB^+ = ABC$	$D^+ = DEA$	$E^+ = EA$
$AC^+ = ACDE$	$C \to D$	$D \to E$
$BC^+ = BCDEA$		
$C \to A$ $\quad AB \to C$		
$BC \to A$		

$F_1 \cup F_2 \cup F_3 = \{ AB \to C, \overset{C \to A}{\wedge} BC \to A, C \to D, D \to E \}$

$AB \to C$ ✓ $\quad D \to E$ ✓
$C \to D$ ✓ $\quad E \to A$ ✗

not FD preserving decomposition

Scanned by CamScanner

13/11/18

Normal Forms (Important)

→ Used to identify degree of redundancy in DB relations.

Normal Forms

i) 1NF   ii) 2NF   iii) 3NF   iv) BCNF   v) 4NF

i)–iv): defined over FD ($X \to Y$)
FD: Single valued dependency

v): over MVD ($X \to\to Y$)
MVD: Multi Valued Dependency.

redundancy in Relation

Over Non-Trivial FD
$X \to Y$
X: not superkey

~~design DB into~~
~~BCNF~~; to eliminate redundancy over FD decompose relation into BCNF

In BCNF 0% redundancy over FD's
Redundancy may exists over MVD's.

Over Non-Trivial MVD
$X \to\to Y$
X: not superkey

To eliminate redundancy over MVD's decompose relation into 4NF.

In 4NF 0% redundancy over FD's && MVD's.

First Normal Form

Relational schema R in 1NF
iff no multi valued attributes in relation R.
[every attribute of R must be single valued or atomic]

[default normal form of an RDBMS table is 1NF]

R (Sid Sname Cid) → multi valued attribute

Sid	Sname	Cid
S1	A	C1/C2
S2	B	C2/C3
S3	B	C3

⇒ Not 1NF relation.

⇓ 1NF design: design Cand key using FD's

R(Sid Sname Cid) { Sid → Sname } — Sid: not SK

Sid	Sname	Cid
S1	A	C1
S1	A	C2
S2	B	C2
S2	B	C3
S3	B	C3

1NF relation

cand key : Sid Cid

Combine cand. key with multi valued attribute to design 1NF relation

(2) R(Sid Sname dob Cid pho email)

Sid	Sname	dob	Cid	pho	email
S1	A	1995	C1/C2	P1/P2/P3/P4	e1/e2/e3/e4
S2	B	1999	C2/C3	P6/P8/P9	e5/e6

2 * 4 * 4 = 32

2 * 3 * 2 = 12

FD: { Sid → Sname dob } — Sid: not SK

no of records for A

2 * 4 * 4 = 32

⇓ 1NF design

R(Sid Sname dob Cid pho email)

Sid	Sname	dob	Cid	pho	email
S1	A	1995	C1	P1	e1
S1	A	1995	C2	P1	e2
⋮	⋮	⋮	⋮	⋮	⋮
S2	B	1999	C2	P6	e5
⋮	⋮	⋮	⋮	⋮	⋮

32 tuples for A

12 tuples for B

degree of redundancy is very high

Cand key Sid Cid Pho Email

R(ABCD) FD: { A → BC }

then we take cand keys as AD bcz D is not functionally dependent on A, hence D must be multi-valued.

Second Normal Form (2NF)

⇒ X → Y FD forms redundancy in relational schema R

iff 1) ~~Also~~ X → Y is non trivial FD

2) X is not super key of R.

⇒ X → Y FD not forms redundancy in relational schema R

iff 1) X → Y FD is trivial FD ie $X \supseteq Y$

(OR)

2) non trivial X → Y FD ; X is super key of R.

Ex:- R(Sid Sname age) { Sid → Sname age }

Sid	Sname	age
S1	A	28
S2	A	20
S3	B	25
S4	B	25
S5	B	22

age 20 is of S1 and not of A

Here ~~Sid is~~ it is non-trivial FD & Sid is super key.

R(Sid, Pin, City) { Sid → Pin, Pin → City }

Sid: S.K — Pin: not S.K — forms redundancy

Sid	Pin	City
S1	P1	C5
S2	P1	C5
S3	P2	C2
S4	P2	C2
S5	P2	C2
S6	P1	C5

3) R(ABCDEF)

$\{ AB \to C$
$B \to D$
$D \to E$
$AE \to F$
$C \to A \}$

Sol: R(ABCDEF)

$AB \to C,\ B \to D,\ D \to E,\ AE \to F,\ C \to A$

~~$AB^+ = ABC$~~ $B^+ = BDE$

$AB^+ = ABCDEF$ ✓

$BC^+ = BCDAEF$ ✓

$BD^+ = BDE$

$BE^+ = BE$

$BF^+ = BF$

{A, B, C}
prime attributi

SK {AB, BC}

Here $B \to D$, $D \to E$, $AE \to F$, $C \to A$ forms redundancy

$$\left.\begin{array}{l} B \to D \\ D \to E \\ AE \to F \\ C \to A \end{array}\right\}$$

$$\boxed{\begin{array}{c} \text{Non Trivial FD} \\ X \to Y \Rightarrow \text{Causes redundancy} \\ \uparrow \text{not S.K} \end{array}}$$

X: not allowed
✓: allowed.

(Important)

FD forms redundancy Non trivial FD $X \to Y$ with X not super key	1NF	2NF	3NF	BCNF
1. proper subset of C.K → Non prime	✓	X	X	X
2. Non-prime → Non-prime	✓	✓	X	X
3. proper subset of C.K as non-prime → non prime	✓	✓	X	X
4. proper subset of cand key → proper subset of other cand key	✓	✓	✓	X

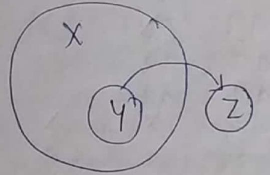
$\underbrace{\text{X}}$ not allowed any FD with which forms redundancy

NOTE:
⇒ if R in 3NF but not in BCNF
Then $\left\{ \text{proper subset of cand key} \to \text{proper subset of another cand key} \right\}$ in R
Which forms redundancy.

Second Normal form (2NF) :-

Relation Schema R in 2NF iff
No partial dependency in R.

partial dependency



X : any cand key of R
$Y \subset X$
Z : non prime attribute of R.

$\boxed{Y \to Z}$ partial dependency

~~Ex~~ NOTE: if cand. key is single attribute then partial key is not possible.

~~Ex. ***~~

Third Normal Form (3NF)

Relation Schema R in 3NF iff every non trivial FD $X \to Y$ in R with

i) X must be Super Key
(or)
ii) Y must be prime attribute,

$X \to Y$ (Sup.key; Prime / non-prime) (OR) $X \to Y$ (X not sup.key; Y must be prime.)

Boyce Codd Normal Form [BCNF]

Relational Schema R in BCNF iff every non-trivial FD, $X \to Y$ in relation R with X must be super key.

{ R in BCNF } <=> { 0% redundancy in R over FD set }

***Q.1 R(ABCDE) with FD set
{ABD → C, BC → D, CD → E}
What is highest NF satisfied by R?

a) 1NF b) 2 NF c) 3NF d) BCNF

Sol:- ABD → C, BC → D, CD → E   R(ABCDE)

$AB^+ = AB$
$ABC^+ = ABCDE$ ✓   {ABC, ABD} ← CK
$ABD^+ = ABDCE$ ✓   P.A { A, B, C, D }
$ABE^+ = ABE$

	ABD → C	BC → D	CD → E	BC → D & CD → E
				BC → E
BCNF	✓	×	×	
3NF	✓	✓	×	
2NF	✓	✓	Not PD ✓	
~~1NF~~				

2NF test: If Closure of proper subset of CK determine only prime attribute then R is in 2NF else not in 2NF

{proper subset of CK}$^+$ = {only Prime}

1. ABD^+ = {A, B, D, AB, AD, BD} ← proper subset of ABD.
   $A^+ = A$
   $B^+ = B$
   $D^+ = D$
   $AB^+ = AB$
   $AD^+ = AD$
   $BD^+ = BD$

2. ABC^+ = {A, B, C, AB, AC, BC}
   $AC^+ = AC$
   $BC^+ = BCDE$ → non prime

   So $BC \to E$ is partial dependency
   So relation R is not 2NF

Hence R must be 1NF.

Q2 R(ABCD)
{$AB \to C$, $C \to A$, $A \to D$} → partial dependency

Sol: Cand key {AB, BC}
$AB^+ = ABCD$ ✓
$BC^+ = BCAD$ ✓
$BD^+ = BD$

Not 2NF
Highest NF is 1NF

Q3 R(ABCD) {$AB \to C$, $BC \to D$}.

Sol: $AB^+ = ABCD$ ✓ Cand key → {AB}

$AB \to C$ not PD
$BC \to D$ not PD

Hence in 2NF

AB = {A, B}
$A^+ = A$
$B^+ = B$

*** Q4 R(ABCD) {$AB \to C$, $C \to A$, $AC \to D$}

Sol: $AB^+ = ABCD$ ✓ Cand key {AB, BC}
$BC^+ = BCAD$ ✓
$BD^+ = BD$

$AB \to C$ not PD
$C \to A$ not P.D
$AC \to D$ not P.D

AB = {A, B}
$A^+ = A$
$B^+ = B$

BC = {B, C}
$B^+ = B$
$C^+ = CAD$ ✓ non-prime
$C \to D$ is partial dependency

Hence 1NF.

*** Q5 R(ABCDEF)
{$AB \to C$, $C \to D$, $CD \to AE$, $DE \to F$, $EF \to B$}

Sol: $A^+ = A$
$B^+ = B$
$C^+ = CDAEFB$ ✓
$D^+ = D$
$E^+ = E$
$F^+ = F$
$AB^+ = ABCDEFL$
$AEF^+ \to AEFBCD$ ✓
$ADE^+ \to ADEFBC$ ✓

Scanned by CamScanner

Candidate key { C, AB, AEF, ADE }

	$AB \to C$	$C \to D$	$CD \to AE$	$DE \to F$	$EF \to B$
BCNF	✓	~~✓~~	✓	X	
3NF	✓	✓	✓	✓	✓

Highest NF is 3NF

Q6 $R(\acute{A}B\acute{C}D\acute{E}\acute{F})$

$\{ AB \to C, C \to DE, E \to F, F \to A \}$

Sol:
$AB^+ = ABCDEF$ ✓
$BC^+ = BCDEFA$ ✓
$BD^+ = BD$
$BE^+ = BEFACD$ ✓
$BF^+ = BFACDE$ ✓

Candidate key { AB, BC, BE, BF }

	$AB \to C$	$C \to DE$	$E \to F$	$F \to A$
BCNF	✓	X		
3NF	✓	X		
2NF	not PD	PD		

Highest NF is 1NF

Q7 $R(\acute{A}\acute{B}C\acute{D})$ $\{ A \to B, B \to AC, C \to D \}$

Sol: $A^+ = ABCD$ ✓
$B^+ = BACD$ ✓
$C^+ = CD$ *
$D^+ = D$

$CD \xrightarrow{+} CD$

Candidate key { A, B }

Candidate key is single attribute, hence partial dependency not possible

	$A \to B$	$B \to AC$	$C \to D$
BCNF	✓	✓	X
3NF	✓	✓	X
2NF	not PD	not PD	not PD

Highest NF is 2NF

Q8 $R(\acute{A}\acute{B}\acute{C}\acute{D}E)$

$\{ A \to B, B \to C, C \to D, D \to A \}$

Sol: $AE^+ = AEBCD$ ✓
$BE^+ = BECDA$ ✓
~~CD~~ $CE^+ = CEDAB$ ✓
$DE^+ = DEABC$ ✓

Candidate key { AE, BE, CE, DE }

	$A \to B$	$B \to C$	$C \to D$	$D \to A$
BCNF	X			
3NF	✓	✓	✓	✓

Highest NF is 3NF

Q9) refer (7)

S1 :> If all candidate keys of relational schema R is simple cand key mean single attribute key

Then R always guaranteed in 2NF.

by may/may not 3NF

refer (5)

Q10) Statement

if every attribute of relational schema R is prime attribute

Then R always 3NF but may/may not BCNF

Q11) Statement

Relational schema R with no non-trivial FD's

R always in BCNF but may/may not 4NF

Ex: R(ABC) { no non trivial FD's }

mean <u>ABC</u> is cand. key

{ No non trivial FD's in R } => { No redundancy over FD's in R }

then R guranteed in BCNF

(Imp)

Q12) Relation schema R with only two attribut

R always in BCNF also in 4NF, 5NF, 6NF

R (AB) possible FD's

BCNF	$\{A \rightarrow B\}$	$\{\underline{A}\}$
BCNF	$\{B \rightarrow A\}$	$\{\underline{B}\}$
BCNF	$\{A \rightarrow B, B \rightarrow A\}$	$\{\underline{A}, \underline{B}\}$
BCNF	{no nontrivial FD}	$\underline{AB}$

Decomposition of Relation into Higher NF :

Q R(ABCDE)

$\{AB \rightarrow C, B \rightarrow DE\}$

Sol: Cand key : AB

$\because AB^+ = ABCDE$

$B \rightarrow DE$ is partial dependency

R is in 1NF but not 2NF

2NF decomposition

R1 $\boxed{ABC}$ R2 $\boxed{\underline{B}DE}$ } LLJ, FD preserved

$AB \rightarrow C$ $\quad B \rightarrow DE$

2NF satisfied ✓

3NF ✓

BCNF ✓

this B can be Foreign key

14/11/18

Q R(ABCDE)

{AB → C, B → D, D → E}

i) decompose into 2NF

Sol. $AB^+ = ABCDE$

Cand key: AB.

B → D partial dependency & B → E also Partial dependency

$B^+ = BDE$

R1 $\boxed{ABC}$ AB → C, R2 $\boxed{BDE}$ B → D, $\underbrace{D \to E}$ not in 3NF } LLJ, DP, 2NF ✓

ii) decompose into 3NF

$\boxed{ABC}$ AB → C, ~~BD~~, $\boxed{BD}$ B → D, $\boxed{DE}$ D → E } 3NF ✓ BCNF ✓

F.K F.K

Q R(ABCDEF)

{AB → C, B → D, D → E, A → F}

ii) decompose in 2NF

Sol. $AB^+ = ABCDEF$

Cand key: AB

Partial dep: B → D & A → F
B → E

ABCDEF

$B^+ = BDE$

$A^+ = AF$

{ $\boxed{BDE}$ B → D, $\underbrace{D \to E}$; $\boxed{AF}$ A → F ; $\boxed{ABC}$ AB → C } LLJ, DP, not PD, 2NF ✓

ii) Decompose into 3NF

$\boxed{DE}$ D → E, $\boxed{BD}$ B → D, $\boxed{AF}$ A → F, $\boxed{ABC}$ AB → C

Q*** R(ABCD)

{A → B, C → D}

i) decompose into 2NF?

Sol. $AC^+ = ACBD$

Cand key: AC

Partial dep: A → B
C → D

$A^+ = AB$

$C^+ = CD$

LLJ, DP, 2NF ✓ { $\boxed{AB}$ A → B, $\boxed{CD}$ C → D, ~~ABCD~~ AC, $\boxed{AC}$ no non-trivial FDs }

Scanned by CamScanner

Q R(ABCDEF)

$\{A \to D, B \to E, C \to F\}$

decompose into 2NF

Sol:- $(ABC)^+ = ABCDEF$

Cand Key : ABC

PD : $\{A \to D, B \to E, C \to F\}$

$A^+ = AD$ $B^+ = BE$ $C^+ = CF$

F.K's

$\boxed{AD}$ $\boxed{BE}$ $\boxed{CF}$ $\boxed{ABC}$

$A \to D$ $B \to E$ $C \to F$ no non-trivial FD's.

LLJ
DP
2NF

Q R(ABCDE)

$\{A \to B, B \to D, D \to E, E \to BE\}$

decompose into 3NF

Sol: $AC^+ = ACBDE$ ✓ prime attributes $\{A, C\}$

Cand Key : AC

PD : $\{A \to B\}$

$A^+ = ABDE$

~~PD~~ $A \to B$ ✗ B

$B \to D$ ✗

$D \to E$ ✗

$E \to BE$

$\boxed{ABDE}$ $\boxed{AC}$

$A \to B$, $B \to D$, $D \to E$, $E \to BE$; no non-trivial FD's.

LLJ
DP
2NF

$B^+ = BDE$

$D^+ = DEB$

$E^+ = BED$

$\boxed{AB}$ $\boxed{BDE}$ $\boxed{AC}$

$A \to B$; $B \to D$, $D \to E$, $E \to BE$

~~$D^+ = BEB$~~

~~$E^+ = BED$~~

$\boxed{AB}$ $\boxed{BDE}$ $\boxed{AC}$

$A \to B$; $B \to D$, $D \to E$, $E \to BE$

3NF

Q R(ABCDE)

$\{A \to B, B \to C, C \to D, D \to BE\}$

decompose into 3NF

Sol: $A^+ = ABCDE$ ✓

Cand Key : A

no PD. hence R is in 2NF

$B^+ = BCDE$ $D^+ = DBEC$

$C^+ = CDBE$

$\boxed{AB}$ $\boxed{BCDE}$

$A \to B$; $B \to C$, $C \to D$, $D \to BE$

LLJ
DP
3NF ✓

Q R(ABCDE)

$\{A \to B, B \to C, C \to D, D \to E\}$

decompose into 3NF

Sol: Cand Key : A hence R in 2NF bcz no PD

$B^+ = BCDE$

$C^+ = CDE$

$D^+ = DE$

$\boxed{AB}$ $\boxed{BCDE}$: ~~$\boxed{BCD}$ $\boxed{CDE}$ $\boxed{DE}$~~

$B \to C$

$C \to D$, $D \to E$; $C^+ = CDE$, $D^+ = DE$

BCDE → BC, CDE

CDE → CD ✓, DE ✓

$\left\{ \boxed{AB} \; \boxed{BC} \; \boxed{CD} \; \boxed{DE} \right\}$ LLJ, DP

$A \to B \quad B \to C \quad C \to D \quad D \to E$

Q R(ABCDE)

{AB → C, BC → A, AC → B}

Cand key: {ABDE, BCDE, ACDE}

decompose into BCNF?

Sol: Cand key: ABDE, BCDE, ACDE

given R is in 3NF but not in BCNF

$ABDE^+ = ABDEC$

$BCDE^+ = BCDE\,A$

$ACDE^+ = ACDE\,B$

$\boxed{\cancel{ABCDE}}$ $\boxed{\cancel{ABDE}}$

$AB^+ = ABC$ $\quad BC^+ = BCA$ $\quad AC^+ = ACB$

$\boxed{ABC}$: AB → C, BC → A, AC → B

$\boxed{ABDE}$: no non-trivial FD's

} LLJ, DP, BCNF

Cand key {AB, BC, AC}

Q R(ABC)

{AB → C, C → A}

Cand key {AB, BC}

R in 3NF but not in BCNF

decompose into BCNF

Sol: Cand key {AB, BC}

ABCD

AB → C

C → A

$C^+ = CA$

$\boxed{\cancel{CB}}$ $\boxed{\cancel{AC}}$ $\boxed{\cancel{CA}}$ $\boxed{\cancel{BCD}}$

~~C → A~~ C → A

$\boxed{AC}$: C → A

$\boxed{BC}$: no non-trivial FD's.

} LLJ, BCNF, Not DP

here Dependencies are lost.

if $\boxed{ABC}$: AB → C, C → A } LLJ, DP, not BCNF

NOTE: DP && BCNF decomposition not possible for above relation.

(V.V. Important)

DB design Goals as per Normalization	1NF	2NF	3NF	BCNF	4NF
*** 1) Loss less join decomposition	Yes	Yes	Yes	Yes	may not
** 2) Dependency preserving decomposition	Yes	Yes	Yes	may not	may not
* 3) 0% redundancy	No	No	No	Yes over FD's, no over MVD's	Yes over FD, no over MVD's.

lossy Join decomp
⇓
Spurious tuples
⇓
Inconsistency

Not DP decomposition
⇓
loss of cand. key

Not 0% redundancy
⇓
DB Anomalies Occurs

NOTE: 3NF is most accurate NF

* 5NF [Join Dependency NF] ~~JD~~ JDNF :-

if 4NF decomposition satisfied LLJ then DB in 5NF

* 6NF [Domain Key NF] DKNF :-

if 4NF decomposition satisfied LLJ && ~~DP~~ dependency preserving decomposition. Then DB is in 6NF

ER Model :-

Entity Relationship diagrams used for high level DB design (or) Diagrammatic representation of DB design.

Main Components to draw ER-diagram :-

1) Attributes
2) Entity Sets
3) Relationship Sets

- Attributes :
- Key attribute
- Multi valued attributes

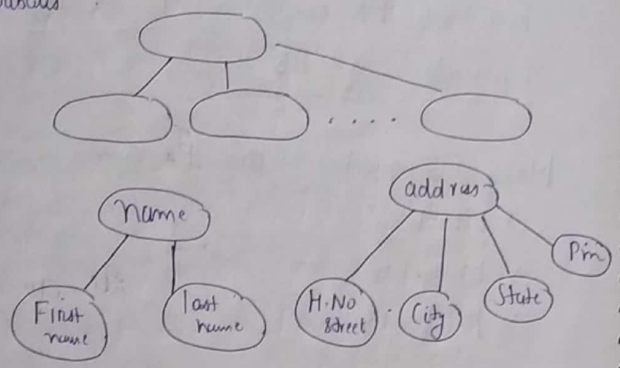
Derived Attribute :- value of attribute derived from other stored attribute.

Ex: DOB — stored attribute

age — derived attribute

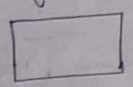
Composit Attribute

attribute which can represent as 2 or more attributes :



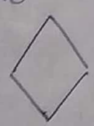
Entity Set :-

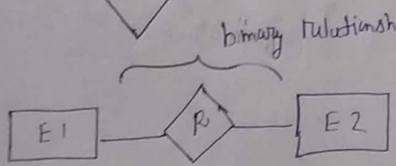
Set of similar entities (or) tuples

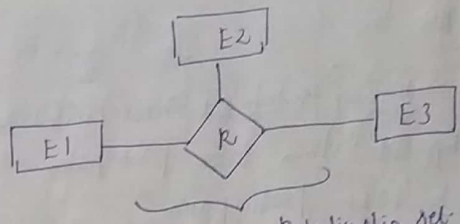


Relationship Set :-

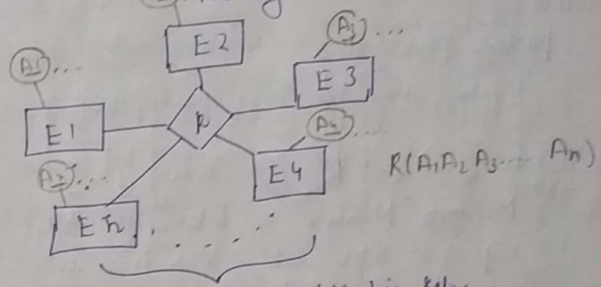
Used to relate two or more entity set



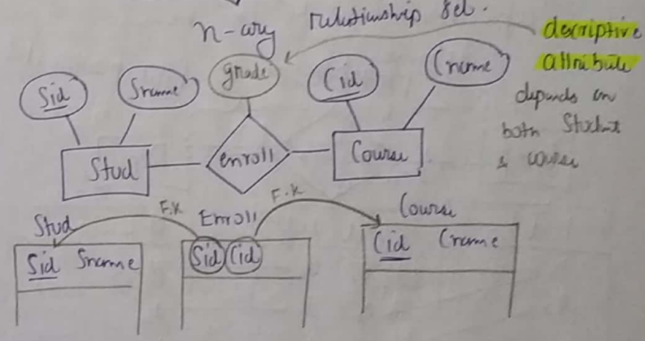




Ternary Relationship set



Ex:



main integrity Constraint of entity set is PK && AK's

Foreign Key constraint is must for relationship set

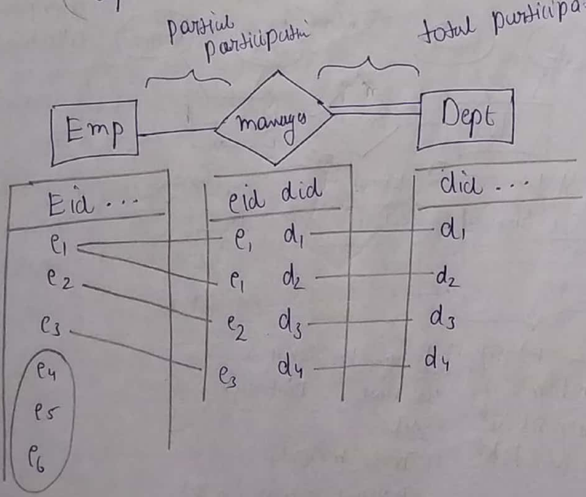
ie main integrity constraint of relationship set is F.K.

Participation :-

If every entity of entity set must related to relationship set then total participation [must be 100% participation]

Otherwise
partial participation
[may / may not 100% participation]

EX (specification) Emp && Dept are Entity sets.
~~Manages~~ Manages is Relation set b/w emp && dept set. Such that every dept there must be one manager



DB design Steps

High level design
1. Requirements :- What type of data / operation
2. design ER diagrams
3. Convert ER diagram to RDBMS table.
   only guarantee 1NF.
   Apply Normalization :
4. Design Schemas of DB Table.
5. Design Indexing
6. design Application program && User access control.
   [Front end Screen] *

low level design

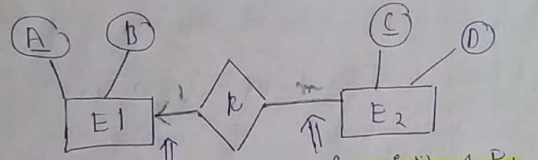
Cardinality [Mapping] of Relationship Set :-

for binary relationship set
1. One to One mapping
2. One to many mapping
3. many to one mapping
4. many to many mapping

based on required mapping design candidate keys of relationship set.

1) One : Many

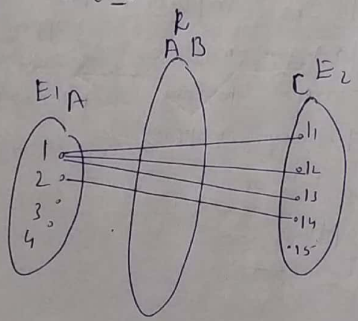
Scanned by CamScanner



each entity of E_2 related at most one entity of E_1

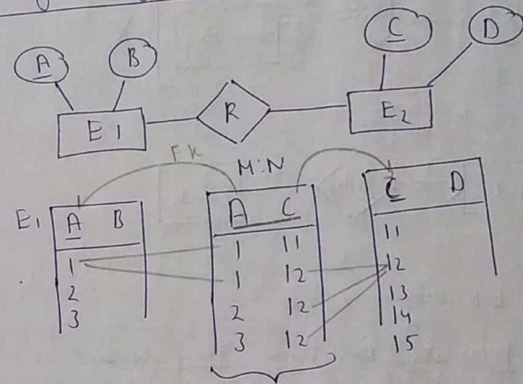
each entity of E_1 can relate to many (0 or more) entities of E_2

to maintain 1:M mapping relationship set cand key of R(A C) is {C}



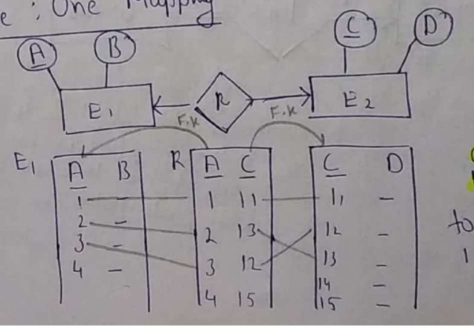
⇒ For [1:M] one : many mapping cand key comes from many side participating entity

Many : Many Mapping



AC: cand key for relationship set R to maintain M:N Mapping

One : One Mapping

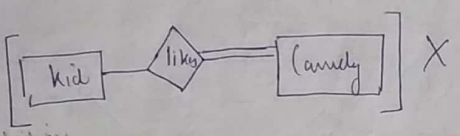


{A, C} are candidate keys of relset R to maintain 1:1 mapping

Q Draw ER diag for given specification

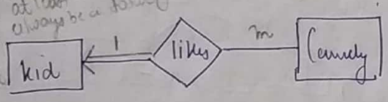
"each candy likes by <u>atmost one kid</u> and each kid likes <u>some</u> candy"
at least one

Sol.



X

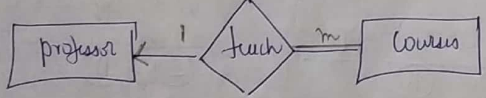
If there is a kid in kid entity set, then he must like at least one candy hence there will always be a total participation from kid entity set



Q draw ER diag

"each professor can teach <u>many</u> courses" and each course ~~teach~~ taught by <u>only one</u> professor".

Sol.

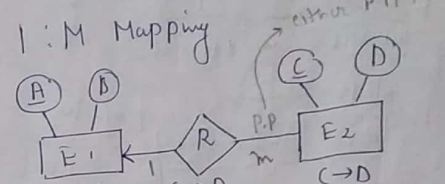


If there is a course in course entity set, then there must be some professor to teach it. hence there will be total participation in course set

15/11/18

RDBMS table Design for given ERD

1 : M Mapping



Min 2 RDBMS tables for given ERD && 1 F.K.

A → B | C → A | C → D

E1 (A, B) | R (C A) | E2 (C D)

E1 A	B	R C	A	E2 C	D
a1	—	a1	c1	c1	—
a2	—	a4	c2	c2	—
a3	—	a2	c3	c3	—
a4	—			c4	—
				c5	—

R: key ⇑ (C) — E2: key ⇑ (C)

F.K.

E1 (A B)

A	B
a1	—
a2	—
a3	—
a4	—

E2R (C D A)

C	D	A
c1	—	a1
c2	—	a1
c3	—	a2
c4	—	NULL
c5	—	NULL

C → DA

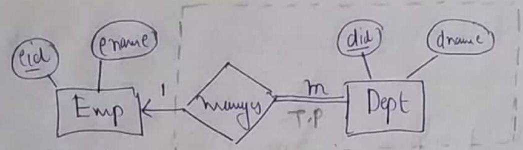
{C}

primary key

A is FK references to E1

because of partial participation at E2 and foreign key "A" values can be "NULL" values

Ex: Emp && Dept entity sets. Manages relationship set with each employee can manage <u>many</u> depts && each dept should managed by one employee



eid → ename   did → eid   did → dname

Emp

<u>eid</u>	ename
e_1	
e_2	
e_3	
e_4	

eid → ename

Dept - Manages   foreign key

<u>did</u>	dname	eid
d_1	x	e_1
d_2	y	e_1
d_3	z	e_2
d_4	z	e_2

did → dname eid

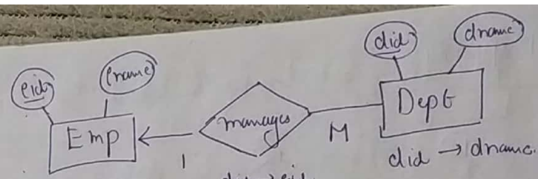
did : primary key

eid : foreign key references to emp
FK values are not NULL because
total participation at Dept end

<u>NOTE:</u> In 1:M mapping relationship set combines with entity having "many" participation.

Disadvantage of 1:M mapping

1 : M relationship set
if Merges with Left side entity set :-



eid → ename   did → eid   did → dname.

Emp - Manages

eid	ename	did
e_1	A	d_1
e_1	A	d_2
e_2	B	d_3
e_2	B	d_4

$\left\{\begin{array}{l}\text{did} \to \text{eid} \\ \text{eid} \to \text{ename}\end{array}\right\}$

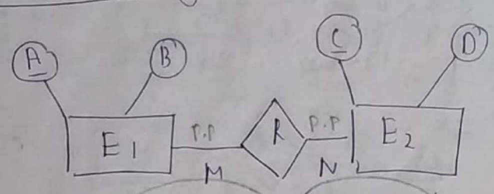
did : primary key

** ⇒ partial participation at Left side entity set is failed
[Not possible to store employee who are not managers] bcz did can not be NULL]

⇒ forms data redundancy [allowed in 1NF design]
(eid → ename non-trivial FD with eid is not super key forms redundancy)

Scanned by CamScanner

M:N Mapping :-



E1(<u>A</u> B)	R(<u>A C</u>)	E2(<u>C</u> D)
a_1 —	a_1 c1	c1 —
a_2 —	a_1 c2	c2 —
a_3 —	a_2 c2	c3 —
a_4 —	a_3 c2	c4 —
A → B	no non-trivial FD	C → D

{ Min 3 RDBMS tables required && 2 FK's }

E1R

A	B	C
a_1	b_1	c_1
a_1	b_1	c_2
a_2	b_1	c_2
a_3	b_2	c_2
a_4	b_2	NULL

not allowed bcz AC is primary key

This is Wrong Design

{A → B}

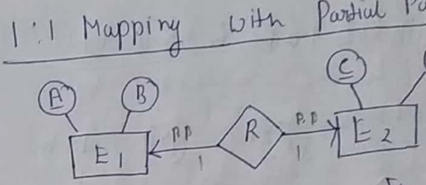
AC : primary key

Hence not possible to store E_1 entities not related to relationship

* data redundancy

if R combines with E_2 then also same problem occurs

1:1 Mapping with Partial Participation both Ends :-



E1

<u>A</u>	B
a1	—
a2	—
a3	—
a4	—

A → B

R

<u>A</u>	C
a1	c1
a2	c2
a3	c3

A → C

C → A

E2

<u>C</u>	D
c1	—
c2	—
c3	—
c4	—

C → D

{ Min 2 relational tables && 1 Foreign key }

E1R

<u>A</u>	B	C
a_1	—	c_1
a_2	—	c_2
a_3	—	c_3
a_4	—	NULL

E2

<u>C</u>	D
c_1	—
c_2	—
c_3	—
c_4	—

{C → D}

A → BC

C → A

{A, C} : Cand keys

primary key : A

"C" F.K references to E_2

E1RE2

A	B	<u>C</u>	D
a1		c1	—
a2		c2	—
a3		c3	—
a4		NULL	NULL
NULL	NULL	c4	—

P.P at E2 side

P.P at E1 side

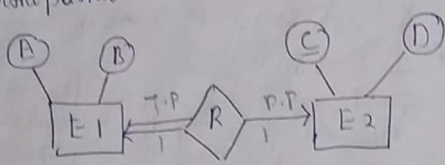
A → BC

C → AD

{<u>A</u>, <u>C</u>} Cand key

No cand key whose field value "NOT NULL"

1:1 Mapping with at least one end total participation

i) 

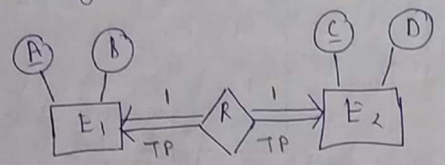
M:m
1
Relational table (No FK's)

$E_1 R E_2$ (A B C D)

A	B	C	D
		c1	—
a1	—		
		c2	—
a2	—		
a3	—	c3	—
Null	Null	c4	—

b/z of P.P at E_2 side

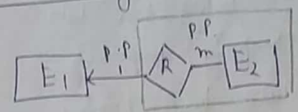
$A \rightarrow BC$
$C \rightarrow DA$
{ A, C }
⇑ ⇑
alternate key, primary key

ii) 

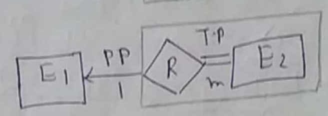
$E_1 R E_2$ (A B C D)
A ⇑ primary key
C ⇑ Unique Not Null

$A \rightarrow BC$
$C \rightarrow DA$
{ A, C }

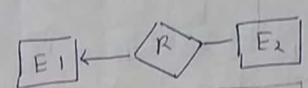
Summary

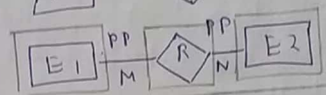


Mm 2 tables
1 F.K

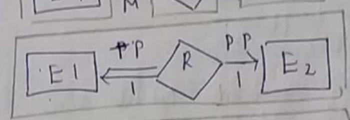


Mm 2 tables
1 F.K

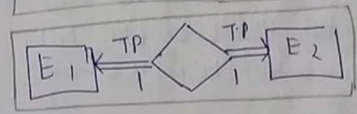




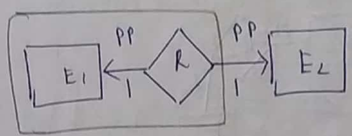
Mm 3 tables
2 FK



mm 1 table
&& no FK



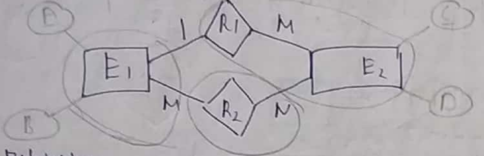
mm 1 table
&& no F.K



mm 2 tables
&& 1 FK

Q1) E_1, E_2 entity sets & R_1, R_2 rel set related b/w E_1, E_2 entity sets with 1:M & M:N mapping respectively. How many min related tables required

a) 2 b) 3 c) 4 d) 5

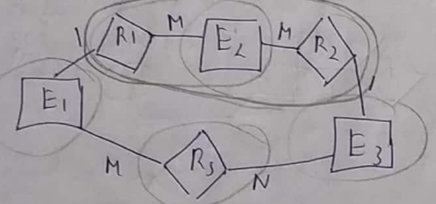


E_1 (A B)
E_2 (C D A)
R_2 (A C)
FK → 3

related tables
E_1, R_2, E_2R_1 = 3

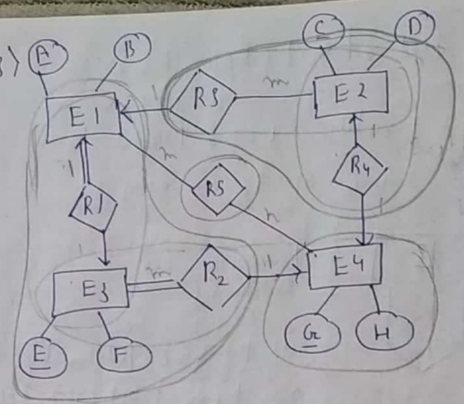
Q E_1, E_2, E_3 entity sets. R_1 rel set b/w E_1, E_2 with 1:M. R_2 rel set b/w $E_2 E_3$ with M:1. R_3 Rel set b/w $E_1 E_3$ with M:N. How many min relational tables?

Sol



relational set tables
E_3, E_1, (E_2R_1, E_2R_2), R_3 = 4
$E_2R_1R_2$

Q3)



rel table
E2 R3 R4, ~~E3R2E4~~,
~~E1R1, R5~~
E1R1E3, E4, R5

E2R3R4 (C D A G)

E1R1E3 (A B E F G)
AK PK

xxx i) Min rel tables?
~~5~~ 4

ii) # FKs required for ~~minimal~~ minimized DB design?
5 FK's

ii) # Attributes for minimized DB design?
5+4+2+2 = 13

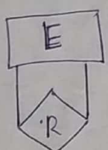
$E_1R_1E_3R_2$	(A B E F G)	{A, E} cand key
$E_2R_3R_4$	(C D A G)	{C, G} cand key
E_4	(G H)	{G} cand key
R_5	(A G)	{AG} cand key

Scanned by CamScanner

Self Referential Relationship Set :-

Entities of Entity Set (E) related to same other entities of same entity set

Recursive Relationship Set

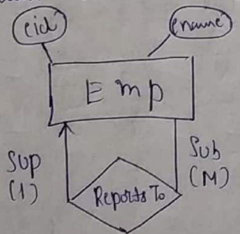


} Recursive entity set

} Self referential relationship set also binary relationship set can be 1:M, M:1, M:N Mapping.

Ex: Emp(eid, ename) entity set & "Reportsto" is relationship set related b/w Supervisor, subordinates.

i) each supervisor can supervise many subordinates and each subordinate reports to one supervisor.



} min 1 table and 1 F.K

Reports To

Sup	Sub
e_1	e_2
e_1	e_3
e_2	e_4

Emp

eid	ename
e_1	A
e_2	A
e_3	B
e_4	C

Emp Reports To

F.K

eid	ename	Sup
e_1	A	Null
e_2	A	e_1
e_3	B	e_1
e_4	C	e_2

eid of emp

eid of sup

Assignment

ii) repeat it for M:1 } 1 table & 1 F.K

iii) 1:1 } 1 table } 1 table & 1 FK

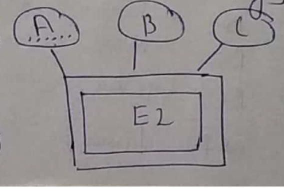
iv) M:N } 2 tables & 2 FK

Weak Entity Set :-

Entity Set with No key

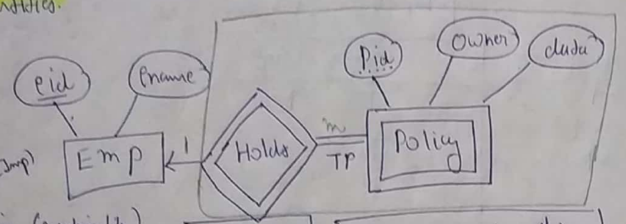
[Attributes of weak entity set not sufficient to differentiate entities uniquely]

Partial key



A, B, C not sufficient to differentiate entities uniquely for each weak entity set there must be identifier (OWNER) which is strong entity set

Weak entity set entities are depending entities. { Min 2 tables and 1 F.K }



Grade (Imp)

* Mapping (cardinality) b/w Identifier && weak entity set mostly 1:M [always 1:M]

eid	Pid
e1	P1
e2	P1
e2	P2
e3	P2

Weak Relation set

Pid	amount	data
P1	50000	2005
P1	50000	2005
P2	80000	2006
P2	80000	2005

F.K

eid	ename
e1	—
e2	—
e3	—
e4	—

Policy holds

eid	Pid	amount	date
e1	P1	50000	2005
e2	P1	50,000	2005
e2	P2	80,000	2006
e3	P2	80,000	2005

eid Pid : Primary Key

** → Participation at weak entity set end must be Total participation.

** → Weak entity set and its relationship set must be one RDBMS table

** → Multi valued attribute and its weak entity set allowed to represent in ERD, but not allowed in RDBMS tables.

16/11/18

QUERIES

Procedural Query Language

formulation of
- What data retrieved from DB (or)
- How to retrieve data from DB

*** Relational Algebra

Non Procedural Query lang

formulation of
- What data retrieved from DB tables

Relational calculus [uses First Order logic && Predicate calculus formulas]

* Tuple Relational Calculus [TRC]	Domain Relational calculus [DRC]
*** SQL Queries close to TRC formulas	QBE [Query by Example] Queries close to DRC formula

Relational Algebra Queries

Basic Operators :-

- π : Projection } Unary operator
- σ : Selection } Unary operator
- $\times$: Cross product
- ρ : Rename } Unary operator
- $\cup$: Union
- $-$: Set difference (minus)

Derived Operators :

- $\cap$: Intersection { Using "$-$" }
- $\bowtie$: Join { Using $\times$, σ, π }
- *** / or $\div$: Division { Using π, $\times$, $-$ }

⇒ DB table is set of records.

⇒ bag representation

R

A	B
2	4
2	4
3	5
4	5

} set of records
can be duplicate records : bag

⇒ Result of any RA expression is always distinct tuples.

can be bag
⇓
RA Query
⇓
always set

$A = \{1, 2, 3, 4\}$ → set

$B = \{1, 2, 3, 3, 4\}$ → bag

Projection (π) :-

Used to project required attributes from relation R.

$\pi_{\text{attribute list}}(R)$

Selection (σ) :-

Used to select records of relation R those are satisfied condition (P)

$\sigma_P(R)$

R

A	B	C
4	6	8
4	6	8
7	6	8
9	3	5
9	3	5

$\pi_{BC}(R)$

B	C
6	8
3	5

$\sigma_{A>5}(R)$

A	B	C
7	6	8
9	3	5

Q Which is true

i) σ is commutative

ii) π is commutative

~~a)~~ only i) True

b) only ii) True

c) both true

d) ~~both i, ii +~~ none

Sol $\sigma_{C_2}(\sigma_{C_1}(R)) = \sigma_{C_1}(\sigma_{C_2}(R))$

C_2: age > 25, C_1: sal > 50,000; C_1: sal > 50,000, C_2: age > 25

$\pi_{list2}(\pi_{list1}(R)) \neq \pi_{list1}(\pi_{list2}(R))$

Q Which LHS Query can not replace by RHS Query.

~~a) $\sigma_{C_2}(\sigma_{C_1}(R))$~~

a) $\sigma_{C_2}(\sigma_{C_1}(R)) \rightarrow \sigma_{C_1}(\sigma_{C_2}(R))$

b) $\sigma_{C_1}(\pi_{list1}(R)) \rightarrow \pi_{list1}(\sigma_{C_1}(R))$

✓ c) $\pi_{list1}(\sigma_{C_2}(R)) \rightarrow \sigma_{C_2}(\pi_{list1}(R))$

d) None

eid	sal	dept
e1	10	d1
e2	20	d2
e3	40	d2
e4	20	d3

$\pi_{eid, sal}(\sigma_{dept=2}(Emp))$

e2	20
e3	40

$\sigma_{dept=2}(\pi_{eid, sal}(Emp))$

no result.

R(ABCD)

b) $\sigma_{B>10}(\pi_B(R)) \Rightarrow \pi_B(\sigma_{B>10}(R))$

c) $\pi_B(\sigma_{A>5}(R)) \not\Rightarrow \sigma_{A>5}(\pi_B(R))$

Cross Product (X) :-

R X S : results all attributes of R followed by all attributes of S and each record of R pairs with every record of S.

R (arity : x; n distinct tuples)

A	B	C
4	5	6
7	3	4
8	5	6

S (arity : y; m distinct tuples)

C	D
6	7
5	5
9	2

R X S (arity : x + y)

A	B	C	C	D
4	5	6	6	7
4	5	6	5	5
4	5	6	9	2
7	3	4	6	7
7	3	4	5	5
7	3	4	9	2
8	5	6	6	7
8	5	6	5	5
8	5	6	9	2

cardinality : n * m

Gate
Q R with n distinct tuples. S with 0 tuples. How many records in R X S results: 0 tuples

JOIN Operators :-

1) Natural Join ($\bowtie$)

$R \bowtie S \equiv \underset{\text{distinct attributes}}{\pi} \left(\underset{\text{Equality condition b/w all common attributes of R \& S}}{\sigma} (R \times S) \right)$

R(A B C) S(C D)

$R \bowtie S = \underset{ABCD}{\pi} \left(\underset{R.C = S.C}{\sigma} (R \times S) \right)$

R(A B C) $\bowtie$ S(C D)

A	B	C
4	5	6
7	3	4
8	5	6

$\bowtie$

C	D
6	7
5	5
9	2

=

A	B	C	D
4	5	6	7
8	5	6	7

EX: T_1(A B C) T_2(B C D)

$T_1 \bowtie T_2 = \underset{ABCD}{\pi} \left(\underset{(T_1.C = T_2.C) \wedge (T_1.B = T_2.B)}{\sigma} (T_1 \times T_2) \right)$

EX3 T_1(A B) T_2(C D)

$\boxed{T_1 \bowtie T_2 = T_1 \times T_2}$

NOTE: if no common element then Natural Join equal to cross product

2) Conditional Join ($\bowtie_c$)

$R \underset{c}{\bowtie} S = \sigma_c (R \times S)$

$R \underset{R.C > S.C}{\bowtie} S = \underset{R.C > S.C}{\sigma} (R \times S)$

R

A	B	C
4	5	6
7	3	4
8	5	6

$\underset{R.C > S.C}{\bowtie}$ S

C	D
6	7
5	5
9	2

=

A	B	C	C	D
4	5	6	5	5
8	5	6	5	5

NOTE: $\boxed{\text{Arity of } \bowtie \le \text{Arity of } \times \quad \text{Arity of } \bowtie_c = \text{Arity of } \times}$

EX $R \underset{A<5}{\bowtie} S = \underset{A<5}{\sigma} (R \times S)$

A	B	C	C	D
4	5	6	6	7
4	5	6	5	5
4	5	6	9	2

Scanned by CamScanner

Ex: $\sigma_{A<5}(R \bowtie S)$

A	B	C	D
4	5	6	7
8	5	6	7

$\Downarrow$ A>5

A	B	C	D
4	5	6	7

$(R \bowtie S)_{A<5}$

A	B	C	C	D
4	5	6	6	7
4	5	6	5	5
4	5	6	9	2

3) Outer Joins

i) Left Outer Join (⟕)

R ⟕ S : results ~~[R ⋈ S]~~

[R ⋈ S] U [records of R those failed Join condition with NULL values in other attributes]

R

A	B	C
4	5	6
7	3	4
8	5	6

⟕ S

C	D
6	7
5	5
9	2

=

A	B	C	D
4	5	6	7
8	5	6	7
7	3	4	Null

ii) Right Outer Join (⟖)

R

A	B	C
4	5	6
7	3	4
8	5	6

⟖ S

C	D
6	7
5	5
9	2

=

A	B	C	D
4	5	6	7
8	5	6	7
Null	Null	5	5
Null	Null	9	2

iii) Full Outer Join (⟗)

R ⟗ S = (R ⟕ S) U (R ⟖ S)

R

A	B	C
4	5	6
7	3	4
8	5	6

⟗ S

C	D
6	7
5	5
9	2

=

A	B	C	D
4	5	6	7
8	5	6	7
7	3	4	Null
Null	Null	5	5
Null	Null	9	2

iv) $R ⟗_{R.C>S.C} S$

A	B	C
4	5	6
7	3	4
8	5	6

$⟗_{R.C>S.C}$ S

C	D
6	7
5	5
9	2

=

R	B	C	C	D
4	5	6	5	5
8	5	6	5	5
7	3	4	Null	Null
Null	Null	Null	6	7
Null	Null	Null	9	2

Usage of Joins

⋈ | $\bowtie_c$ used to (Satisfy) retrieve some | any | atleast one record from cross product.

R(A....) S(B...)

Retrieve records of R those "A" values more than "some" 'atleast one | any "B" values of S

R

A...
10
20
30
40
45

S

B....
15
25
35

Same

20 > {15}

30 > {15, 25}

40 > {15, 25, 35}

45 > {15, 25, 35}

R

A..	B..
20	15
30	
40	
45	

R X S

	A...	B...
X	10	15
X	10	25
X	10	35
✓	30	15
✓	30	25
	30	35
	⋮	⋮

$\pi_A \left(R \bowtie_{R.A > S.B} S \right)$

R

A
20
30
40
45

EX: Stud (Sid, age)   faculty (fid, age)

Retrieve Sid's whose age more than some faculty age

$= \pi_{Sid} \left(Stud \bowtie_{Stud.age > faculty.age} faculty \right)$

(or)

$\pi_{Sid} \left(\sigma_{Stud.age > faculty.age} (Stud \times faculty) \right)$

Stud (Sid age)   X   faculty (fid, age)

Sid	age
S1	18
S2	22
S3	25

fid	age
f1	20
f2	24

Sid.age > faculty.age

Sid	age	faculty	age
S1	18	f1	20
S1	18	f2	24
S2	22	f1	20
S2	22	f2	24
S3	25	f1	20
S3	25	f2	24

⇓

Sid	age	faculty	age
S2	22	f1	20
S3	25	f1	20
S3	25	f2	24

$\bowtie_c$: Some / any / at least one

***

R (A...)   S (B...)

Retrieve records of 'R' those "A" values more than every / all 'B' values of S.

Concept: [Use complementation to access required data]

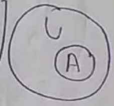
$\sim (> B) = \leq (\text{Some } B)$

$\neg(> \text{all } B) \equiv \leq (\text{some } B)$

$\left[\begin{array}{c}\text{all "A" val} \\ \text{of } R\end{array}\right] - \left[\begin{array}{c}\text{"A" value of } R \\ \text{those } \leq \text{some "B" of } S\end{array}\right]$

Mc

$= \neg \left[\text{"A" value of } R \text{ those } \leq \text{some "B" of } S \right]$



$U - A = A^c$

$= \neg (\leq \text{some } B)$

$= > (\text{all } B)$

$$\boxed{\Pi_A(R) - \Pi_A\left(R \underset{R.A \leq S.B}{\bowtie} S\right)}$$

Q Stud (Sid, age)

faculty (fid, age)

"Retrieve Sid's whose age more than every every faculty age"

Sol: $= \Pi_{sid}(\text{Stud}) - \Pi_{sid}\left(\text{Stud} \underset{\text{Stud.age} \leq \text{faculty.age}}{\bowtie} \text{faculty}\right)$

$\left[\begin{array}{c}\text{Sid of all} \\ \text{Stud}\end{array}\right] - \left[\begin{array}{c}\text{Sid of those stud whose} \\ \text{age} \leq \text{some } \cancel{\text{age of}} \text{ faculty age}\end{array}\right]$

Mc

(a) Retrieve Sid's whos age less than all faculty's age.

$= \Pi_{sid}(\text{Stud}) - \Pi_{sid}\left(\text{Stud} \underset{\text{Stud.age} \geq \text{faculty.age}}{\bowtie} \text{faculty}\right)$

ρ : Rename Operator

Used to Rename table / attributes to process Query

Stud (Sid Sname age)

ρ (Temp, Stud) $\Rightarrow$ Temp (Sid, Sname, age)

$\rho_{I,N,A}$ (Stud) $\Rightarrow$ Stud (I, N, A)

$\rho_{Sid \to I,\ Age \to A}$ (Stud) $\Rightarrow$ Stud (I, Sname, A)

$\Rightarrow$ Every possible two records of "R" should compare : R X R [Self cross product]

"Retrieve Sid's whose age more than some stud age"

Stud (Sid age)

Sid	age
S1	18
S2	22
S3	25

Stud (Sid , age)

Sid	age
S1	18
S2	22
S3	25

$\times$ $\sigma_{Stud.age > Stud.age}$ (Stud $\times$ Stud)

sid	age	sid	age

ambiguity

✓ $\pi_{Stud.sid} \left(Stud \bowtie_{Stud.age > Temp.age} \rho(Temp, Stud) \right)$

rename table

(or)

$\pi_{Stud.sid} \left(Stud \bowtie_{Stud.age > Stud.A} \rho_{I,A}(Stud) \right)$

rename attribute

(or)

age > A

Q Emp(<u>eid</u> dno sal)

eid	dno	sal
e_1	4	20
e_2	5	40
e_3	40	60
e_4	5	80
e_5	4	90

i) Retrieve eids whose salary less than some emp of dept 5

$Emp \bowtie_{Emp.sal < S} \rho_{I,D,S}(Emp)$

$\pi_{eid} \left(Emp \bowtie_{(Sal < S) \wedge (D = 5)} \rho_{I,D,S}(Emp) \right)$ (or) $\pi_{eid} \left(Emp \bowtie_{Sal < S} \rho_{I,D,S} \left(\sigma_{dno=5}(Emp) \right) \right)$

2) Retrieve eids whose salary less than every emp salary of dept 5.

$\sim [< (\text{all dps})]$

$\geq$ (some dep5)

{all eid's} − { eid's whose sal $\geq$ some emp of dep 5 }

$\pi_{eid}(Emp) - \pi_{eid} \left(Emp \bowtie_{Sal \geq S \wedge D = 5} \rho_{I,D,S}(Emp) \right)$

Q Emp(eid, gen, sal)

eid	gen	sal
e_1	fem	80
e_2	male	85
e_3	fem	90
e_4	male	95
e_5	fem	80

i) Retrieve female eid's whose salary more than sal of any (at least one) male emp.

$\neg(> (\text{any male})) \Rightarrow \leq (\text{all male})$

$\pi_{eid} \left[\left(\sigma_{gen=fem}(Emp) \right) \bowtie_{Sal > S} \rho_{I,G,S} \left(\sigma_{gen=male}(Emp) \right) \right]$

(or)

$\pi_{eid} \left[(Emp) \bowtie_{Sal > S \wedge gen = fem \wedge G = male} \rho_{I,G,S}(Emp) \right]$

Scanned by CamScanner

ii) Retrieve female eid's whose sal more than all male employees.

Sol. ¬ (Sal > all male)

(Sal ≤ some male)

$= \pi_{eid}(\sigma_{gen=fem}(Emp)) - \pi_{eid}\left(\sigma_{gen=fem}\left(Emp \bowtie_{(Sal \le S) \wedge (gen=fem) \wedge (G=male)} \rho_{I,G,S}(Emp)\right)\right)$

[all female emp id's] - [female emp id's whose salary ≤ some male salary]

Q Emp(eid, Sal)

Retrieve eids who gets highest salary.

[eids whose salary maximum] = [all emp id's] - [emp Id's whose salary not maximum]

[eid whose sal ≥ every sal of emp] = [all emp id's] - [eid's whose sal < some emp]

Concept: A ≥ All B

~(≥ All B) = < some B

A value ≥ all B = [All A val's] - (A < some B)

eid	Sal
e1	70
e2	90
e3	60
e4	50
e5	90

eid	Sal
e1	70
e2	90
e3	60
e4	50
e5	90

$= \pi_{eid}(Emp) - \pi_{eid}\left(Emp \bowtie_{Sal < S} \rho_{I,S}(Emp)\right)$

ii) Retrieve Emp id whose salary is minimum.

(ie ≤ Sal of every emp)

[all emp ids] - [emp ids > some emp id]

$\pi_{eid}(Emp) - \pi_{eid}\left(Emp \bowtie_{Sal > S} \rho_{I,G,S}(Emp)\right)$

Q Emp(eid, dno, Sal)

Retrieve eids whose Sal maximum for each dept.

Emp (

eid	dno	Sal
e1	2	70
e2	2	80
e3	2	60
e4	3	70
e5	3	90

[eid's whose sal max for each dept] = [all eids of emps] - [eids whose not max for each dept]

[eids whose sal ≥ every sal of his/her dept (same dept)] = [eid of all emp] - [eid of emp whose sal < some emp's sal of his/her dept]

$$\Pi_{eid}(Emp) - \Pi_{eid}\left(Emp \bowtie_{Sal < S \wedge D_{no} = D} \rho_{I,D,S}(Emp)\right)$$

Q family (parent child child DOB)

parent	child	child DOB
A	B	2000
A	C	2005
A	D	2002
B	E	2005
B	E	2008

Retrieve Oldest child of each parent

[Child whose DOB ≤ all child DOB of same parent] = [all Child] - [Children whose (DOB) > some children's DOB of same parent]

$$\Pi_{child}(family) - \Pi_{child}\left((family) \bowtie_{(childDOB > DOB) \wedge (parent = P)} \rho_{P,C,DOB}(family)\right)$$

DIVISION [/ or ÷] (V.V.V. Important)

Used to retrieve attribute values which paired with every value of other relation

R

A	B
a_1	b_1
a_1	b_2
a_1	b_3
a_2	b_1
a_2	b_2
a_3	b_1

pair with all S

S

B
b_1
b_2
b_3

$$\Pi_{AB}(R) / \Pi_{B}(S) = \begin{array}{|c|}\hline A \\ \hline a_1 \\ \hline\end{array}$$

Retrieves "A" values of R those are paired with every "B" of S

EX: Enroll (Sid Cid) Course (Cid ...)

Retrieve Sid's enrolled in every course

Enroll

Sid	Cid
S1	C1
S1	C2
S1	C3
S2	C1
S2	C2
S3	C1

Course

Cid ...
C1
C2
C3

$$\Pi_{Sid\,Cid}(Enroll) / \Pi_{Cid}(Course) = \begin{array}{|c|}\hline Sid \\ \hline S1 \\ \hline\end{array}$$

Expansion of "/" :-

$\Pi_{Sid\,Cid}(Eroll) / \Pi_{Cid}(Course)$: Sid's enrolled every course

1) Sid's not enrolled every course
[Sid's enrolled proper subset of courses]

$$\left(\Pi_{Sid}(E) \times \Pi_{Cid}(C) - \Pi_{Sid\,Cid}(E)\right)$$

Sid	Cid
~~S1~~	~~C1~~
~~S1~~	~~C2~~
S1	C3
~~S2~~	~~C1~~
~~S2~~	~~C2~~
S2	C3
~~S3~~	~~C1~~
S3	C2
S3	C3

every stud enrolled every course

−

Sid	Cid
S1	C1
S1	C2
S1	C3
S2	C1
S2	C2
S3	C1

$\underbrace{\qquad\qquad}_{\text{Sid's enrolled courses}}$

Sid	Cid
S2	C3
S3	C2
S3	C3

→ these records tells in which course corresponding student has not enrolled into.

Sid
S2
S3

⇐ disqualified Sid's for "/"

2) Step 2) {

{Sid's enrolled every course} = {Sid's enrolled in some course} − {Sid's enrolled proper subset of courses}

Sid
S1

=

S1
S2
S3

−

S2
S3

$$\boxed{\Pi_{Sid\,Cid}(E) / \Pi_{Cid}(C) = \Pi_{Sid}(E) - \Pi_{Sid}\left(\Pi_{Sid}(E) \times \Pi_{Cid}(C) - \Pi_{Sid\,Cid}(E)\right)}$$

$$\boxed{\Pi_{AB}(R) / \Pi_{B}(S) = \Pi_{A}(R) - \Pi_{A}\left(\Pi_{A}(R) \times \Pi_{B}(S) - \Pi_{AB}(R)\right)}$$

"A" values of R those are paired with every B of S

Q Given DB table
Stud (Sid Sname age)
Course (Cid, Cname, Instructor)
Enroll (Sid Cid fee)

i) Retrieve Sid's enrolled <u>some course</u> taught by KORTH.

Enroll (Sid Cid) X Course (Cid Inst)

Sid	Cid
S1	C1
S1	C2
S2	C1
S2	C3
S2	C2
S3	C4
S4	C4

Cid	Inst
C1	Korth
C2	Korth
C3	Navathe
C4	Ullmann
C5	Navathe

Query 1)

$$\pi_{sid}\left(\sigma_{Enroll.Cid = Course.Cid \wedge Inst = Korth}(Enroll \times Course)\right)$$

30 tuples in "X"
30×2 = 60 comparison
[Cost of query]

(or)

Query 2)

$$\pi_{sid}\left(\sigma_{Enroll.Cid = Course.Cid}\left(Enroll \times \sigma_{Inst = Korth}(Course)\right)\right)$$

· 5 comparison
· 12 tuples in "X"
· 5+12 = 17 comp
[Cost of Query]

(or)

Query 3)

$$\pi_{sid}\left(Enroll \bowtie_{Eroll.Cid = Course.Cid \wedge Inst = Korth} Course\right)$$

Query 4)

$$\pi_{sid}\left(Enroll \bowtie \sigma_{Inst = Korth}(Course)\right)$$

Query 2) & Query 4) are efficient

ii) Retrieve Sid's enrolled in all the courses (or) every taught by Korth..

E(Sid Cid) pair with every C(Cid Inst) $\left\{\begin{matrix} C1 & Korth \\ C2 & Korth \end{matrix}\right\}$

$$\pi_{Sid,Cid}(Enroll) \,/\, \pi_{Cid}\left(\sigma_{Inst = Korth}(Course)\right) = \begin{array}{|c|} \hline Sid \\ \hline S1 \\ \hline \end{array}$$

$$\begin{bmatrix} S1 & C1 \\ S1 & C2 \\ S2 & C1 \\ S2 & C3 \\ S3 & C2 \\ S4 & C4 \end{bmatrix} \quad \begin{bmatrix} C1 \\ C2 \end{bmatrix}$$

$$\pi_{sid}(Enroll) - \pi_{sid}\left[\underbrace{\left(\pi_{Sid}(Enroll) \times \pi_{Cid}\left(\sigma_{Inst = Korth}(Course)\right)\right) - \pi_{Sid,Cid}(Enroll)}_{\text{every stud enrolled every Korth course}}\right]$$

↑ enrolled some course $\begin{pmatrix} S1 \\ S2 \\ S3 \\ S4 \end{pmatrix}$

↑ $\begin{pmatrix} S2 \\ S3 \\ S4 \end{pmatrix}$ Enrolled proper subset of Korth course

⇒ Works (eid, Pid)
Project (Pid, Pname)

Retrieve eid's who works for some "DB" (Pname) projects

$\Pi_{eid} \left(\text{Works} \bowtie \left(\sigma_{Pname = DB} (\text{Project}) \right) \right)$

2) Retrieve eid's who works for every "DB" project

Sol :- $\Pi_{eid\,Pid} \,/\, \Pi_{Pid} \left(\sigma_{Pname = DB} (\text{Projects}) \right) =$

Set Operator :-

$\cup$: Union

$-$: Set difference

$\cap$: Intersection

* To Apply Set ~~operation~~ operators

Union Computable

R & S Union Computable iff

i) Arity of R = Arity of S

&& ii) domain of each attribute of R must same domain for S attributes respectively

domain (Attr A) : possible values accepted by Attr A.

$\Pi_{Sid\,Sname} (\ldots\ldots) \cup \Pi_{Sid} (\ldots\ldots)$ ✗

$\Pi_{Sid\,Sname} (\ldots\ldots) \cap \Pi_{Sid\,age} (\ldots\ldots)$ ✗

$\Pi_{Sid\,Sname} (\ldots\ldots) - \Pi_{StudID\,Stud\,name} (\ldots\ldots)$ ✓

↑ Set of Stud ID's

R ∪ S, R − S, R ∩ S set operations

i) Resulted record set always distinct tuples.

ii) Result Schema is same as Schema of R

R

A	B
2	4
2	4
3	5
4	5
4	5
3	5

S

C	D
2	4
2	4
4	5
4	5
3	6

R ∪ S

A	B
2	4
3	5
4	5
3	6

R ∩ S

A	B
2	4
4	5

R − S

A	B
3	5

Intersection ($\cap$)

(Derived Operator)

$R \cap S = R - (R - S)$ (or) ~~$(R-S) \cup (S-R)$~~

$= (R \cup S) - ((R-S) \cup (S-R))$

$R \cup S = \{ X / X \in R \vee X \in S \}$ OR

$R \cap S = \{ X / x \in R \wedge x \in S \}$ AND

$R - S = \{ X / x \in R \wedge x \notin S \}$ But not

NOTE:

=> T_1

A	B
2	4
2	5
2	6

T_2

A	B
2	4
3	5
2	8

① $T_1 \cap T_2 = T_1 \bowtie T_2 =$

A	B
2	4

② $T_1 \cup T_2 = T_1 ⟗ T_2 =$

A	B
2	4
2	5
2	6
3	5
2	8

if T_1 & T_2 with same name attributes then

NOTE: Set theory distribution law, Demorgan's laws are not applicable in Relational Algebra.

Q Stud (Sid name age)

Course (Cid Cname Inst)

Enroll (Sid Cid fee)

=> Retrieve Sid's whose age more than 20 (or) enrolled in same course taught by Korth.

Sol: $\pi_{Sid}(\sigma_{age>20}(Stud)) \cup \pi_{Sid}(Enroll \bowtie (\sigma_{Inst=Korth}(Course)))$

↑ Sid's whose age more than 20

↑ Sid's enrolled same Korth course.

=> Retrieve Sid's enrolled same course taught by Korth (or) same course taught by Navathe

$\pi_{Sid}(Enroll \bowtie (\sigma_{Inst=Korth}(Course))) \cup \pi_{Sid}(Enroll \bowtie (\sigma_{Inst=Navathe}(Course)))$

(or)

$\pi_{Sid}(Enroll \bowtie (\sigma_{Inst=Korth \vee Inst=Navathe}(Course)))$

Q Which query is to right :-

Query 1) $\pi_{sid}\left(\text{Enroll} \bowtie \sigma_{\substack{Inst = Korth \\ \wedge\ Inst = Navathe}}(\text{Course})\right)$ ✓ Empty

Wrong

Query 2) $\left[\pi_{sid}\left(\text{Enroll} \bowtie \sigma_{Inst = KORTH}(\text{Course})\right)\right]$ S1

$\cap \left[\pi_{sid}\left(\text{Enroll} \bowtie \sigma_{Inst = Navathe}(\text{Course})\right)\right]$

S1
S2

set difference

=> Retrieve Sid's enrolled only Navathe courses

~~$\left(\text{Enroll} \bowtie \left(\sigma_{Inst = Navathe}(\text{Course})\right)\right)$~~

Concept:

$\left.\begin{array}{l}\text{Only} \\ \text{But not}\end{array}\right\}$ Using " $-$ "

$\left\{\begin{array}{l}\text{Sid's enrolled} \\ \text{Some courses}\end{array}\right\} - \left\{\begin{array}{l}\text{Sid's enrolled some} \\ \text{non Navathe Course}\end{array}\right\}$

$S2 = \begin{bmatrix} S1 \\ S2 \\ S3 \end{bmatrix} - \begin{bmatrix} S1 \\ S3 \end{bmatrix}$

$\Uparrow$

~~Only in~~ Enrolled only in Navathe course

$\pi_{sid}(\text{Enroll}) - \pi_{sid}\left(\text{Enroll} \bowtie \left(\sigma_{Inst \neq Navathe}(\text{Course})\right)\right)$

Q ~~Enroll~~ Retrieve Sid's enrolled at least two courses.

Enroll (T_1) $\times$ Enroll (T_2)

Here we take self cross product

Sid	Cid
S1	C1
S1	C2
S1	C3
S2	C1
S2	C2
S3	C1

$\times$

Sid	Sid
S1	C1
S1	C2
S1	C3
S2	C1
S2	C2
S3	C1

Compare every possible two records $(t_1 ; t_2)$ of Enroll rel. Such that if
$(t_1.sid = t_2.sid$ and $t_1.cid \neq t_2.cid)$
Then t_1 sid enrolled at least two courses.

$$\pi_{sid}\left(\sigma_{\substack{sid = S \\ cid \neq C}}\left(\text{Enroll} \times \rho_{S,C,F}(\text{Enroll})\right)\right)$$

(or)

$$\pi_{sid}\left((\text{Enroll}) \bowtie_{\substack{(sid = S) \\ \wedge (cid \neq C)}} \rho_{S,C,F}(\text{Enroll})\right)$$

=> Retrieve Sid's enrolled at least three courses

$$\underbrace{\overbrace{\text{sid} \quad \text{cid} \quad \text{sid} \quad \text{cid} \quad \text{sid} \quad \text{cid}}^{\text{cid} \neq \text{cid} \neq \text{cid}}}_{\text{sid} = \text{sid} = \text{sid}}$$

$\rho(T_1, Enroll)$

$\rho(T_2, Enroll)$

$\rho(T_3, Enroll)$

$$\pi_{T_1.Sid}\left(\sigma_{\substack{T_1.Sid = T_2.Sid \ \wedge \\ T_2.Sid = T_3.Sid \ \wedge \\ T_1.Cid \neq T_2.Cid \ \wedge \\ T_2.Cid \neq T_3.Cid \ \wedge \\ T_3.Cid \neq T_1.Cid}}(T_1 \times T_2 \times T_3)\right)$$

=> Retrieve Sid's with at least 2 students of same name.

Sid	Sname
S1	A
S2	A
S3	B

Sid	Sname
S1	A
S2	A
S3	B

~~(Enroll) ⋈ (Enro~~

$$\pi_{Sid}\left((Stud) \bowtie_{\substack{Sid \neq S \ \wedge \\ Sname = N}} \rho_{S,N}(Stud)\right)$$

=> Retrieve Sid's enrolled ~~at most one course~~ only one course

[Sid's enrolled atleast one course but not " — " at atleast two course]

$$\pi_{Sid}(Enroll) - \pi_{Sid}\left(Enroll \bowtie_{\substack{Sid = S \\ \wedge\ Cid \neq C}} \rho_{S,C,F}(Enroll)\right)$$

Sid
S3

S1, S2, S3	−	S1, S2

- Atleast two $\sigma_p(T_1 \times T_2)$
- Atleast three $\sigma_p(T_1 \times T_2 \times T_3)$
- only two } set difference with at least....
- at most two

=> Sid's enrolled at most one course

[0 1] 2 3 4

[all students who enrolled 0 or more course] − [Sid's enrolled at least two courses]

$$\pi_{Sid}(Stud) - \pi_{Sid}\left((Enroll) \bowtie_{\substack{Sid = S \\ \wedge\ Cid \neq C}} \rho_{S,C,F}(Enroll)\right)$$

=) Sids enrolled at most two courses

[All students who enrolled 0 or more course] − [All students who enrolled in at least 3 courses]

SQL [Structured Query Language]

Sub languages of SQL :-

1) **Data Definition lang [DDL]**

Used to define / modify structure [definition] of the DB Table.

Used to modify Integrity Constraints PK/AK/FK

CREATE TABLE
DROP TABLE
ALTER TABLE
i) Add/remove attributes
ii) Modify IC.

2) **Data Manipulation lang [DML]**

Used to modify data records & & Used to access required data from DB tables

INSERT INTO tname ....
DELETE FROM tname
UPDATE tname SET....

SELECT *
FROM Tables
WHERE P;
} data access command

3) **Data Control lang [DCL]**

i) Data Control For Transaction Management. [To avoid inconsistency because of concurrent execution]
lock (data)
Unlock (data)
commit
Rollback etc.

ii) Data Control for Security [Access Control]
- Grant Access // User
- Revoke access // User

SQL Query Equal to RA Query :-

SELECT DISTINCT A1, A2, ... An FROM R1, R2, ... Rm WHERE P;

SELECT DISTINCT → equivalent to "π" of RA
FROM → equivalent to "X" of R.A
WHERE → equivalent to "σ" of R.A

$\sigma_P (R1 \times R2 \times R3 \dots \times R_m)$

$$\pi_{A1, A2 \dots An} \left(\sigma_P (R1 \times R2 \times R3 \times \dots R_m) \right)$$

Works (eid Pid)
Project (Pid pname)

Retrive eids who ~~went~~ works for some db project.

RA → $\pi_{eid} \left(Works \bowtie_{works.pid = Project.pid \wedge pname = db} Project \right)$

(or)

$$\pi_{eid} \left(\sigma_{pname = db \wedge works.pid = Project.pid} (Works \times Project) \right)$$

~~SELECT~~ Rename Table

$\boxed{\text{Table AS } T_1 \text{ (or) Table } T_1}$

3. SELECT DISTINCT eid
1. FROM WORK AS T, PROJECT AS T_2
2. WHERE T_1.pid = T_2.pid AND Pname = 'DB';

Works

eid	pid
e1	P1
e1	P2
e2	P2
e3	P3

Project

Pid	Pname
P1	DB
P2	DB
P3	NW

got this after where clause →

eid	pid	Pid	Pname
e1	P1	P1	DB
e1	P2	P2	DB
e2	P2	P2	DB

Π_{eid} → e1, e2

SELECT eid → e1, e1, e2

$\boxed{\text{SELECT DISTINCT} \equiv \Pi \text{ (Projection)}}$

Basic SQL Clauses :-

(6)

(5) SELECT [DISTINCT] A1, A2 .... An
(1) FROM R1, R2 .... Rm
(2) [WHERE P] ← tested for each record
(3) [GROUP BY (Attributes)
(4) [HAVING (Condition)]] ← tested for each group
(7) [ORDER BY (attributes) [DESC]];

Execution flow :-

FROM (Cross Product X)
⇓
WHERE (Condition Operator σ)
⇓
GROUP BY
⇓
HAVING
⇓
SELECT
⇓
DISTINCT } (Projection Operator Π)
⇓
ORDER BY // Sort resulted records in ASC|DESC based specific attr's value.

Aggregate functions :-

Count ( )
Sum ( )
AVG ( )
MIN ( )
MAX ( )

} Computes aggregation of non NULL values [NULL values discards by Aggregate fun's]

Count(*) include Row ID also.

default attribute associated with every table and User is not allowed to access it.

CREATE TABLE R
(
A int,
B int
);

It is UNIQUE AND NOT NULL

R:

Row ID	A	B
1	20	5
2	Null	20
3	40	Null
4	Null	Null
5	20	25
6	60	40

Scanned by CamScanner

# of records

# of non null values of A

⇒ Select Count(*) AS A1, Count(A) AS A2,
Count (DISTINCT A) AS A3,
Count (A, B) AS A4

FROM R;

* A DISTINCT A (A,B)

o/p

A1	A2	A3	A4
6	4	3	5

Select Sum(A) A1, SUM(DISTINCT A) A2,
Avg (A) A3, AVG (DISTINCT A) A4,
MIN (A) A5, MAX(A) A6

⇒ FROM R;

o/p

A1	A2	A3	A4	A5	A6
140	120	35	40	20	60

$$AVG(A) = \frac{SUM(A)}{Count(A)} \qquad AVG(DISTINCT\ A) = \frac{SUM(DISTINCT\ A)}{Count(DISTINCT\ A)}$$

~~Select AVG (A,B)~~

~~Select AVG(A, B)~~
~~FROM R;~~

Incorrect usage of aggregation

⇒ Select AVG(A), B
FROM R; } ERROR

AVG(A)	B
35	5
	20
	NULL
	⋮

X

if Aggregate function Used in SELECT clause not allowed to select any attribute in SELECT clause if GROUP BY clause NOT Used.

Select *
FROM R
WHERE $\boxed{A = max(A)}$; } 4 records in result

each record

20 = max(A) ✓ True

NULL = max( ) Unknown

NOTE: Comparison with NULL treated as Unknown [neither True nor False]
Unknown result discards by Where clause.

⇒ Aggregate funⁿ in Where clause computes for each record.

Select *
FROM R
Where A > 30;

X 20 > 30 false
X Null > 30 Unknown
✓ 40 > 30 True
X NULL > 30 Unknown
X 20 > 30 false
✓ 60 > 30 True

o/p

A	B
40	Null
60	40

Select *
FROM R
Where not (A > 30);

20 > 30 True
20 > 30 True

o/p

A	B
20	5
20	25

Here also Unknown values are discarded.

GROUP BY :-

Used group data records based on Group by attributes values

if Group by clause used in Query

Then ① In SELECT clause must select every attribute of Group by.

② Allowed to select aggregate function in SELECT clause
[computes aggregation of each group]

③ Not allowed to select any other Attribute in select clause.

SELECT X, Avg(Y)
FROM R
GROUP By $\underbrace{X}_{\text{Set of attribute}}$

~~R A B C~~
~~a1 b5~~
~~Null b2~~
~~a3~~
~~a1~~
~~a2~~
~~Null~~
~~a1~~
~~a2~~
~~a3~~

R

A	B	C
a_1	b_5	70
Null	b_2	40
a_3	b_2	NULL
a_1	b_5	80
a_2	b_3	80
Null	b_2	60
a_1	b_1	50
a_2	b_3	90
a_3	b_2	90

Group by(A) →

A	B	C
a_1	b_5	70
a_1	b_5	80
a_1	b_1	50
Null	b_2	40
Null	b_2	60
a_3	b_2	Null
a_3	b_2	90
a_2	b_3	80
a_2	b_3	90

1) Select A
FROM R
GROUP BY (A);

A
a_1
Null
a_3
a_2

2) Select A, AVG(C) ← each group
FROM R
GROUP BY A;

O/P

A	C
a_1	66.66
Null	50
a_3	90
a_2	85

3) Select B
FROM R
GROUP BY (A); } ERROR

4) Select A, B
FROM R
Group BY A; } ERROR

A	B
a_1	b_5, b_1
a_3	b_2, b_2
Null	b_2
a_2	b_3, b_3

← Single A value, multiple B value. Hence WRONG

5) Select A , AVG(C)
FROM R
Group by (A,B);

⇒ SQL Standard will give ERROR
Most of langs allowed

Groups

A	B
a_1	b_2
a_1	b_3
a_1	b_1
Null	b_2
a_3	b_2
a_2	b_3

O/P

A	Avg(C)
a_1	70
a_1	80
a_1	50
Null	50
a_3	90
a_2	85

these "a_1" are from different groups.

6) Select A, B
FROM R
Group by (A,B);

O/P

HAVING Clause :-

⇒ Used to select groups based on having clause condition ie only those groups are selected which satisfy having clause condition

⇒ Having clause condition tested for each groups.

⇒ Having clause condition must be over aggregate function (or) special defined function some(), all() every() etc but not over direct attribute.

1) ④ Select A
① FROM R
② Group by A
③ having AVG(C) > 60

O/P

A
a_1
a_3
a_2

2) Select A
FROM R
Group by A
having $\underbrace{\text{every (C) > 75}}_{\text{every 'C' of Group > 75}}$;

O/P

A
a_2

3) Select A
FROM R
GROUP BY A
Having Some(C) > 75;

O/P

A
a_1
a_3
a_2

4) Select A
FROM R
Group By A
having C > 75;

ERROR

C > 75 → not suitable to test group level

5) Select A
FROM R
GROUP BY A
having Some (C) > 75
order by A;

A
a_1
a_3
a_2

⇒

A
a_1
a_2
a_3

Emp(eid, dno, Sal, gender)

Retrieve Eid's who gets highest salary

RA

① $\pi_{eid}(Emp) - \pi_{eid}\left(Emp \bowtie_{\cancel{Emp.}Sal < S} \rho_{I,D,S,G}(Emp)\right)$

②
```
SELECT eid
FROM Emp
```
$-$
```
EXCEPT
Select T1.eid
FROM Emp T1, Emp T2
WHERE T1.Sal < T2.Sal ;
```

③ Using max()
```
Select eid
FROM Emp
Where Sal = ( Select max(Sal)
              FROM Emp) ;
```

④
```
Select eid
FROM Emp, ( Select Max(Sal) AS Sal FROM Emp) temp
WHERE emp.Sal = Temp.Sal.
```

Retrieve eid's who gets highest salary for each department

~~SELECT eid, did~~.

RA $\pi_{eid}(Emp) - \pi_{eid}\left((Emp) \bowtie_{Sal<S,\ Dno=D} \rho_{I,D,S,G}(Emp)\right)$

②
```
SELECT eid
FROM Emp
```
} All emp ID's

$-$
```
EXCEPT
SELECT T1.eid
FROM Emp T1, Emp T2
WHERE T1.Sal < T2.Sal AND
      T1.dno = T2.dno ;
```
} eid's Sal < Same emp Sal of Same dept.

③
```
Select eid
FROM Emp, ( Select dno, MAX(Sal) FROM Sal
            FROM Emp
            Group By (dno)) temp
WHERE Emp.dno = temp.dno AND
      Emp.Sal = temp.Sal ;
```

Emp

eid	dno	sal
e1	2	70
e2	2	90
e3	4	80
e4	4	60

Temp

dno	max(Sal)
2	90
4	80

Enroll (sid cid fee)

⇒ Retrieve sid who paid max fee's for each course

NOTE: different students can enroll same course

Enroll

Sid	Cid	fee
S1	C1	5000
S2	C1	4000
S3	C1	9000
S2	C2	6000
S3	C2	8000
S4	C3	7000

Temp

Cid	fee
C1	9000
C2	8000
C3	7000

O/P

S3	C1
S3	C2
S4	C3

temp.cid → here grouping done for cid and we want sid, so nested query is required

Select Emp.sid, ~~cid~~.

FROM ~~(Select cid~~ Emp, (Select cid , MAX(fee) fee

FROM Enroll

Group by ((cid)) temp

Where Enroll. cid = temp. cid

AND Enroll. fee = temp. fee ;

Q Retrieve eid's who gets 2nd highest Salary.

RA $\rho\left(Temp,\ \pi_{eid,Sal}\left(Emp \underset{Sal<S}{\bowtie} \rho_{I,D,S,G}(Emp)\right)\right)$

$\pi_{eid}(Temp) - \pi_{eid}\left(Temp \underset{Sal<S}{\bowtie} \rho_{I,D,S,G}(Temp)\right)$

SQL : With Clause

Used to reuse sub query result many times for Query processing

[Avoid's re-computation]

WITH Temp(eid, Sal) AS

(Select DISTINCT T_1.eid, T_1.sal

FROM Emp T_1, Emp T_2

WHERE T_1.Sal < T_2.Sal)

Select eid

FROM Temp ~~T_1, Temp T_2~~

~~WHE~~ EXCEPT

Select T_1.eid

FROM Temp T_1, Temp T_2

WHERE T_1.sal < T_2.sal ;

(2) Using aggregate function

III

Select eid

FROM Emp

II

WHERE Sal = (Select MAX(Sal)

FROM Emp

I

WHERE Sal < (Select MAX(Sal)

FROM Emp)) ;

eid	Sal
e1	20
e2	40
e3	60
e4	30

eid	Sal
e1	20
e2	40
~~e3~~	~~60~~
e4	30

eid	Sal
e1	20
e2	40
e3	60
e4	30

Retrieve dno's with only two employees.

{ dno's with at least 2 employees } − { dnos with at least 3 employees }

①

```
Select T1.dno
FROM Emp T1, Emp T2
WHERE  T1.dno = T2.dno AND
       T1.eid <> T2.eid
EXCEPT
Select
FROM Emp T1, Emp T2, Emp T3
WHERE  T1.dno = T2.dno AND T2.dno = T3.dno
   AND T1.eid <> T2.eid AND
       T2.eid <> T3.eid AND
       T3.eid <> T1.eid ;
```

② Using Group by and Having

```
Select dno
FROM Emp
GROUP BY dno
HAVING count(*) = 2;
```

Gate NOTE: HAVING clause condition is replaceable by WHERE clause

```
Select dno
FROM( Select dno, count(*) AS C
      FROM Emp
      Group by dno)
WHERE C = 2;
```

Every SQL Query using having clause can replace using Where clause of Nested Query.

i) Retrieve dno whose Avg salary more than Avg salary of all male employees.

①

```
Select dno, AVG(Sal) as Sal
FROM Emp
WHERE T Sal >(Select AVG(Sal)
               from Emp
               WHERE gender = male)
GROUP BY (dno)
```

②

```
Select dno
FROM Emp
GROUP BY dno
HAVING AVG(Sal) > ( Select AVG(Sal)
                    FROM Emp
                    Where gender = male)
```

③

```
Select dno
FROM ( Select dno, AVG(Sal) as X
       FROM Emp
       GROUP BY dno)
WHERE X > ( Select Avg (Sal)
            FROM Emp
            Where gender = male );
```

Q Retrieve dno such Avg Sal of female Employee of each dept is more than Avg salary of all male employees.

Sol:-

```
Select dno , AVG( Sal) as S
FROM Emp
WHERE gen = Female And S > ( Select AVG(Sal)
GROUP BY (dno)                FROM Emp
                              Where gen = male)
```

Q Select Em'eid ERROR

```
From Emp
Where Sal > ( Select Sal
              From Emp
              Where dno = 5);
```

→ inner query result is not scalar value

Sal
90
70

this results in error

```
Select dno
FROM ( Select dno, AVG (Sal) as Sal
       FROM Emp
       Where gen = Female) group by dno.
Where Sal > ( Select AVG(Sal)
              FROM Emp
              Where gen = male)
```

Functions Used for Nested Queries

- IN / NOT IN
- ANY
- ALL
- EXISTS / NOT EXISTS

IN :- Used for membership test

X IN

Y
⋮

} Set of Val of inner query result

TRUE if Attribute X value is belongs to Y set

$\boxed{\text{"IN" fun can Used for equi join queries}}$

① R(A B) = S(C D) — B value equals to at least one value of C

R

A	B
?	2
–	4
–	6
–	7

S

C	D
3	–
4	–
5	–
6	–

$\pi_{AB}\left(R \underset{R.B = S.C}{\bowtie} S\right)$ =

```
Select *
FROM R
WHERE B IN ( Select C
             From S)
```

Equijoin : (= at least one)

② R(A, B) S(C D)

A	B
2	4
2	5
3	4

C	D
2	4
2	6
4	5

$\pi_{AB}\left(R \underset{\substack{R.A = S.C \\ \wedge R.B = S.D}}{\bowtie} S\right)$ =

```
Select *
From R
Where (A, B) IN ( Select C, D
                  From S );
```

Q R(A B) S(C D) DB tables

What is result of SQL query

correct

```
Select A,B
From R                         R-(R-S)        R-S
Where (A,B) NOT IN ( Select A,B
                     From R
                     Where (A,B) NOT IN ( Select C,D
                                          From S) ) ;
```

a) R-S

b) R∩S

c) RUS

d) $R \bowtie S$

$$R-(R-S) = R \cap S$$

ANY && ALL functions :-

Preceded by comparison operator.

<, <=, >, >=, =, <>

X < ANY

Y
⋮

True if X less than atleast one value of Y set.

X < ALL

Y
⋮

TRUE if X less than every val of Y set.

Stud (sid age)   faculty (sid age)

Retrieve sid's whose age less some faculty age.

$$\pi_{sid}\left(Stud \bowtie_{Stud.age < faculty.age} faculty\right)$$ RA

```
Select sid
FROM Stud
WHERE age < ANY ( Select age
                  FROM faculty ) ;
```

Retrieve sid's whose age less than every faculty age

[all sid's] - [sids whose age ~~more~~ >= some faculty]

$$\pi_{sid}(Stud) - \pi_{sid}\left(Stud \bowtie_{Stud.age \geq faculty.age} faculty\right)$$

```
Select sid
FROM Stud
WHERE age < ALL ( Select age
                  FROM faculty);
```

Grate

NOTE: • IN fun equal to (= ANY) fun

• (NOT IN) fun equal (<> All) fun

Q

R

A	B
4	6
3	7
5	5

S

C	D
7	6
4	8
3	5

How many tuples in result of SQL query ?

i) Select *
FROM R
Where B > ANY (Select C
FROM S
Where D >10);

Concept :-

$X \text{ op}^n \underbrace{\text{ANY}\ \boxed{Y}}_{\text{false}}$ } empty

ii) Select *
FROM R
Where B > ALL (Select C
From S
Where D >10);

$X \text{ op } \underbrace{\text{ALL}\ \boxed{Y}}_{\text{true}}$ } empty

it look for at least one value to be True

Sol i) 0 Where $\overbrace{B > ANY}^{\text{false}}$ $\boxed{C}$ empty } $\pi_{AB}\left(R \bowtie_{R.B > S.C} (\sigma_{D>10}(S))\right)$

it look for at least one value to be False

3 ii) Where $\overbrace{B > ALL}^{\text{True}}$ $\boxed{C}$ empty

Concept : $\exists x\ (x > 10)$

True : for some x s.t
x >10

False : for every x s.t
not x >10

Concept: $\forall x\ (x>10)$

True : for every x s.t
x >10 at least one

false : for some x s.t
not x >10

$\pi_{AB} - \pi_{AB}\left(R \bowtie_{R.B > S.C} (\sigma_{D>10}(S))\right)$

Nested Queries

Nested Query without Co-relation :-

```
SELECT   Attributes
FROM     Relations
WHERE    Condition
GROUP BY Attributes
HAVING   Condition
ORDER BY Attributes
```

② Top

[Inner Query]

① Bottom

⇒ Inner Query Independent of Outer Query.

Inner Query Computes only once.

Execution flow : Bottom - Top.

Co-related Nested Query :-

⇒ Inner query Where clause Uses attributes defined in Outer Query.

⇒ Co-related Nested Queries allowed in WHERE clause && HAVING Clause of Outer query.

Nested queries allowed in

- FROM Clause
- WHERE Clause
- HAVING Clause

Co-relation in Where clause

```
③ Select R.A
① FROM R                3 2 X 1
   WHERE ( Select  Count (x)
       ② FROM S
          Where  R.A < S.B) < 2
```

R	A
	10
	20
	30
	40

S	B
	5
	15
	25
	45

o/p → A

30

40

→ If Co-relation in Where clause inner query recomputes for each record of Outer query.

→ If Co-relation in HAVING clause, Inner query recomputes for each group of Outer query.

Q How many records in the result of query

R	A
	5
	10
	15
	25
	35
	45

S	B
	5
	12
	14
	24
	50

```
Select A
From R                    0 1 3 4 4 4
Where ( Select Count (x)
        FROM S
        Where R.A > S.B) ≤ 3
```

5 > 5 (count=0)

10 > 5 (count=1)

15 > 5, 15 > 12, 15 > 14 (count=3)

25 > 5, 25 > 12, 25 > 14, 25 > 24 (count=4)

eid's Whose salary 3rd highest ?
(k^{th})

eid	Sal
e1	70
e2	70
e3	30
e4	40
e5	40
e6	35

eid	Sal
e1	70
e2	70
e3	30
e4	40
e5	40
e6	35

Select eid
FROM Emp T_1
Where (Select Count (Distinct T_2.sal)
FROM Emp T_2
Where T_2. sal > T_1.sal) = 2 ;
[k-1]

Q
i) eid Whose salary 3rd lowest ?
ii) Retrieve eids who gets top 3 salaries ?
iii) Retrieve eid Who get three least salaries ?

Sol i) Select eid
FROM Emp T_1
Where (Select Count (DISTINCT T_2. sal)
FROM Emp T_2
Where T_2.sal < T_1. sal) = 2 ;

Sol ii) Select eid
From Emp
Where (Select count (DISTINCT T_2.sal)
From Emp
Where T_1. Sal < T_2.Sal) $\leq$ 2 ;

Sol - iii) Select eid
From Emp
Where (Select count (DISTINCT T_2.sal)
From Emp
Where T_1.sal > T_2.sal) $\leq$ 2

SET Operations of ~~RA~~ SQL is exactly similar to SET operations of RA

UNION
INTERSECTON
EXCEPT

Select A
FROM R
UNION
SELECT B
FROM S

A
2
2
3

B
2
2
3
4

=>

A
2
3
4

distinct

Scanned by CamScanner

18/11/18

IN, ANY, ALL } Used for Nested Queries Without corelation

EXISTS, NOTEXISTS } Used for Co-related Nested Queries

EXISTS :-

Used to test result of inner query empty / not empty

EXISTS(Inner Query)

TRUE if result of inner query not empty
[At least one record in result]

EX: $\pi_A (R \bowtie_{R.A < S.B} S)$: "A" values of R those are less than some "B" of S.

```
Select A
FROM R
WHERE EXISTS (Select *
              from S
              Where R.A < S.B)
```

R	A
	5
	10
	25

S B	5	10	25
8	✓	×	×
12	✓	✓	×
20	✓	✓	×

o/p	A
	5
	10

Emp (eid dno sal)

Retrieve Eid's Whose salary more than any employee sal of dept 5

RA $\pi_{eid} \left(Emp \bowtie_{Sal > S} \rho_{I,D,S} (\sigma_{dno=5} (Emp)) \right)$

```
⑥ Select T1.eid
① FROM Emp T1
⑤ Where EXISTS (④ Select *
               ② FROM Emp T2
               ③ Where T1.Sal > T2.Sal And T2.dno = 5)
```

SQL ~~Nested~~ Corelated Query
How for every outer tuple inner query is recomputed

SQL JOIN

```
Select distinct T1.eid
FROM Emp T1 , Emp T2
Where T1.Sal > T2.sal AND T2.dno = 5 ;
```

SQL Nested Query

```
Select eid
FROM Emp
Where sal > ANY ( Select Sal
                  FROM Emp
                  Where dno = 5)
```

	Nested	Corelated
Simple Query	efficient	Not efficient
Complex query	Not efficient	efficient

Emp(eid dno sal)

Retrieve Eids whose salary more than every employee of dept 5.

$$\pi_{eid}(Emp) - \pi_{eid}\left(Emp \underset{sal \le S}{\bowtie} \left(\underset{\substack{dno=5 \\ I,D,S}}{\sigma}(Emp)\right)\right)$$

Concept :-

EXISTS = TRUE [at least one record in the result]

NOT EXISTS = TRUE [no record in result]

$\neg(\le \text{Some}) \equiv > \text{All}$

```
Select T1.eid
FROM Emp T1
Where NOT EXISTS ( Select *
      (every)     FROM Emp T2
                  Where T1.Sal <= T2.Sal AND
                        T2.dno = 5) ;
```

***

Q R(A....) S(B....)

Retrieve "A" val of R those are more than every "B" of S

Sol: CR

```
Select A
FROM R
Where NOT EXISTS ( Select *
                   FROM S
                   Where R.A <= S.B)
```

R.A ≤ Some B

R.A > All B

RA

$$\underset{A}{\pi}(R) - \pi_A\left(R \underset{R.A \le S.B}{\bowtie} S\right)$$

***

Division Query (Important)

SQL query which is equal to "/" of RA :-

Q Enroll(Sid Cid) Course (Cid Inst)

Retrieve Sids enrolled every course taught by KORTH.

RA :- $\cancel{\pi_{Sid\,Cid}(Enroll) / \pi_{Cid}(Course)}$

$$\pi_{Sid\,Cid}(Enroll) \Big/ \pi_{Cid}\left(\sigma_{Inst = KORTH}(Course)\right)$$

SQL 3:-

T_1 Enroll

Sid	Cid
S1	C1
S1	C2
S1	C3
S2	C1
S2	C4
S3	C4

Course

Cid	Inst
C1	Korth
C2	Korth
C3	Korth
C4	Navathe

−

T_2 Enroll

Sid	Cid
S1	C1
S1	C2
S1	C3
S2	C1
S2	C4
S3	C4

= ϕ Empty

Retrieve T_1.Sid from Enroll(T_1)

only if { [all Cid's of Korth] − [Cids of Enroll (T_2) which are enrolled by T_2.sid] = ϕ empty }

④ Select distinct T_1.Sid
① FROM Enroll T_1
③ Where NOT EXISTS (Select Cid
FROM ~~Enroll T_2~~ Course
Where Inst = Korth
② EXCEPT
Select Cid
FROM Enroll T_2
Where T_2.Sid = T_1.Sid) ;

Q
Catalog (Sid Pid)
Parts (Pid Color)
Retrieve Sid's who supplied every red part

Select distinct C_1.Sid
FROM Catalog C_1
Where NOT EXISTS (Select Pid
FROM Parts
Where color = red
EXCEPT
Select Pid
FROM Catalog C_2
Where C_1.Sid = C_2.Sid)

Catalog

Sid	Pid
S1	P1
S1	P2
S1	P3
S2	P3
S2	P4
S3	P5

Parts

Pid	Color
P1	R
P2	R
P3	R
P4	G
P5	B

−

Catalog

Sid	Pid
S1	P1
S1	P2
S1	P3
S2	P3
S2	P4
S3	P5

= ϕ Empty

Important Model

Stud (Sid Sname) with 200 tuples
Enroll (Sid Cid) with 100 tuples

Q1 How many (max, min) tuples in result of Stud ⋈ Enroll ?

a) (200, 0) b) (100, 100)
c) (200, 100) d) (20000, 0)

Stud (Sid Sname)

200 tuples

Sid	Sname
2	A
3	B
4	B
5	A
6	C
7	D
8	B

Enroll (sid cid)

100 tuples

sid	cid
2	1
2	2
3	1
3	2
4	1

Sid	Sname	Sid	Cid
1	S1	1	C1
1	S1	1	C2
1	S1	1	C3

Max tuples $S \bowtie E$ = 100 tuples

at least 100 tuples Min tuples $S \bowtie E \geq 100$ tuples

NOTE: Min tuples is 100 if we consider F.K constraint.
Min tuples is 0 if we do not consider F.K constraint.

Q2 R(A B C) No Null's with n tuples
S(B D E) with m tuples
max tuples in result of $R \bowtie S$: ~~min(n,m)~~ n
min tuples in result of $R \bowtie S$: ___

min tuples { ~~min(n,m)~~ n with FK ; 0 without FK

A	B	C
1	2	6
2	2	7
3	2	6
4	2	7
5	3	6

B	D	E
2	5	6
3	7	5

for some questions max tuples in $R \bowtie S$ is determined by # of tuples in relation which contain foreign key

Q3 R(A B C) with n tuples
S(A D E) with m tuples
maximum tuples in $R \bowtie S$: min(m,n)
Min tuples in $R \bowtie S$: 0 without FK
min(m,n) with FK

Q4 R(A B C) with n tuples
S(D B E) with m tuples
max tuples in $R \bowtie S$ result: $n * m$
min tuples in $R \bowtie S$ result: 0

Sol: 3) R(A B C) —n— one to one —m— S(A D E)

R.A	S.A
1	1
2	2
3	3
4	

Sol: 4) R(A B C) S(D B E)

$\bowtie = 0$ then all different

A	B		D	B
1	a_1		5	b_1
2	a_2		6	b_2
3	a_3		7	b_3

(or)

A	B		D	B
1	a_1		5	a_1
2	a_1		6	a_1
3	a_1		7	a_1

Q R(ABC) S(BDF) With 1000 && 2000 tuples respectively.

{A → B, B → C, B → DF} FD's

max tuples in R ⋈ S ?

Sol: R(A B C) S(B D F)

$A^+ = BC$ $B^+ = DF$

max = 1000 ~~With F.K~~

~~Without F.K~~

R(A B C) S(B D F)

A	B		B
1	a1		a1
2	a1		a2
3	a2		
4			
5			

m : n

Imp

Q Rel R with X set of attributes && n distinct tuples

Rel S with Y set of attributes && m distinct tuples

i) to apply R/S required condition : $X \supset Y$

ii) Resulted attribute set of R/S : $X - Y$

iii) Cardinality (min, max tuples) in result of R/S : {0 tuples to $\lfloor \frac{n}{m} \rfloor$ tuples}

Sol. i) $\pi_{AB}(E) / \pi_{B}(C)$ ie $B \subset AB$

$\pi_{sid\ cid}(E) / \pi_{cid}(C)$ ⇒ sid

sid	cid
s1	c1
s1	c2
s1	c3

cid
c1
c2
c3

ie no of tuples in S must be proper subset of no of tuples in R.

ii) $\pi_{ABC}(R) / \pi_{C}(S)$ =

A	B	C
2	4	2
2	4	4
2	4	6

C
2
4
6

=

A	B
2	4

iii) $\pi_{ABC}(R) / \pi_{C}(S) = \lfloor \frac{n}{m} \rfloor$

n {

A	B	C
2	4	2
2	4	4
2	4	6
3	5	2
3	5	4
3	5	6

m {

C
2
4
6

A	B
2	4
3	5

$\lfloor 6/3 \rfloor = 2$

Imp

Q R

A	B
2	3
2	4
3	5
4	5

S

B

} 0 tuples

How many tuples in R/S ?

a) 0 b) 2 c) 3 ✓ d) 5

$\pi_{AB}(R) / \pi_{B}(S) \equiv \pi_{A}(R) - \pi_{A}(R)\left(\pi_{A}(R) \times \pi_{B}(S) - \pi_{AB}(R)\right)$

$= \begin{vmatrix} 2 \\ 3 \\ 4 \end{vmatrix}$ − empty

$\underbrace{\begin{matrix} 2 \\ 3 \\ 4 \end{matrix} \times \phi}_{\phi}$ − $\begin{vmatrix} 2 & 3 \\ 2 & 4 \\ 2 & 5 \\ 4 & 5 \end{vmatrix}$

$= \begin{vmatrix} 2 \\ 3 \\ 4 \end{vmatrix}$

NOTE :- R : with n distinct tuples
S : with 0 tuples.
R/S result : Atmost n tuples.

R & S' Relations with n & m distinct tuples respectively

(Imp)

RA Query	Cardinality
R X S	$n * m$
xxx R ⋈ S, R $⋈_c$ S	{ 0 to $n*m$ }
R ⟕ S	{ n to $n*m$ tuples }
R ⟗ S	{ max(n,m) to $n*m$ tuples }
R ∪ S	{ max(n,m) to n + m tuples }
R ∩ S	{ 0 to min (n, m) tuples }
R − S	{ 0 to n tuples }
xxx R/S	{ 0 to $\lfloor \frac{n}{m} \rfloor$ tuples }

Tuple Relational Calculus

⇒ Non procedural query lang.

⇒ Uses predicate calculus & First Order logic

⇒ p, q predicates
$p \vee q$
$p \wedge q$
$\neg p$
$p \rightarrow q$
$\exists x \in Rel (P(x))$
$\forall x \in Rel (P(x))$
$x \in Rel$

formulas uses in Relational Calculus Queries.

format of TRC Query

Result { T | P(T) }

T : tuple Variable
P(T) : formula over tuple variable T.

Retrieves set of tuples (T) those satisfied P(T)

{ T | T ∈ Stud ∧ T.age > 20 }
⇓
T retrieves tuples which belongs to Stud table and satisfy condition T.age > 20
ie students whose age more than 20

RA : $\sigma_{age > 20}$ (Stud)

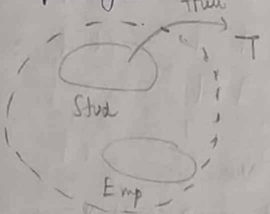
Scanned by CamScanner

Tuple Variable

→ Free Tuple Variable	→ Bounded Tuple Variables
→ Universal variables	→ bounded by $\exists, \forall$ quantifiers

Ex: $\{T \mid T \in Stud \wedge T.age > 30\}$

→ T is universal variable not bounded by any quantifier.



→ Tuple variable used for result of TRC Query must be free variable

$\exists T_1 \in Stud\ (P(T_1))$

True if some T_1 belongs to Stud rel which should satisfy $P(T_1)$

$\forall T_2 \in Stud\ (P(T_2))$

TRUE if every T_2 belongs to rel which should satisfy $P(T_2)$

$T_1\ T_2$ are bounded variables

Q

Stud (Sid Sname age)

Course (Cid Cname Inst)

Enroll (Sid cid fee)

i) Retrieve Sids enrolled some course taught by korth

Concept $\exists_x \exists_y (x+y > 10)$

for some x there must be some y such that $x+y > 10$.

$\{T / \exists T_1 \in Enroll\ \exists T_2 \in Course\ (T_1.cid = T_2.cid \wedge T_2.Inst = korth \wedge T = T_1.sid)\}$

$\boxed{\sigma_p(T_1 \times T_2) \equiv \exists T_1 \exists T_2 (P(T_1, T_2))}$

Enroll

Sid	Cid
S1	C1
S2	C2

Course

Cid	Inst
C1	Korth
C2	Ullmann

ii) Retrieve Sids whose age more than 20 or enrolled course taught by korth.

$\left[\pi_{sid}\left(\sigma_{age>20}(Stud)\right)\right] \cup \left[\pi_{sid}\left(Enroll \bowtie (\sigma_{Inst=korth}(Course))\right)\right]$

$\boxed{R \cup S = \{X / X \in R \vee X \in S\}}$

Scanned by CamScanner

$\{ T / \exists T_1 \in Stud (T_1.age > 20 \wedge T = T_1.sid) \vee$

$\exists T_2 \in Enroll \; \exists T_3 \in Course \; (T_2.cid = T_3.cid \wedge$
$T_3.Inst = korth \wedge$
$T = T_2.sid) \}$

Stud

Sid	age
S1	17
S2	25
S3	18

Enroll

Sid	Cid
S1	C1
S2	C2
S3	C2

Course

Cid	Inst
C1	korth
C2	Ullman

8: $R \cup S = R \vee S$

$R \cap S = R \wedge S$

$\underbrace{R - S}_{RA} = \underbrace{R \wedge \neg S}_{TC}$

Sid's enrolled at least one course

Q
i) $\{ T / (\exists T_1 \in Enroll (T = T_1.Sid)) \wedge$

$\neg \exists T_1 \in Enroll \; \exists T_2 \in Enroll \; (T_1.Sid = T_2.sid$
$\wedge T_1.cid \neq T_2.cid$
$\wedge T = T_1.sid)$

Sid's enrolled at least two courses

Sid's enrolled only one course.

ii) Sid's enroll at most one course.

$\{ T / \exists T_1 \in Stud (T = T_1.sid) \wedge$
$\neg \exists T_1 \in Enroll \; \exists T_2 \in Enroll \; (T_1.sid = T_2.sid \wedge$
$T_1.cid \neq T_2.cid \wedge$
$T = T_1.sid) \}$

all stud Sid's

$\{ T / (\exists T_1 \in Stud (T = T_1.sid)) \wedge$
$\neg (\exists T_1 \in Stud \; \exists T_2 \in Stud \; (T_1.age < T_2.age \wedge T = T_2.sid)) \}$

Stud sids whose age more than some student's age (ie Students whose age is more than at least one)

$$\pi_{sid}(Stud) - \pi_1 \left(Stud \underset{age < A}{\bowtie} \rho_{I,M,A}(Stud) \right)$$

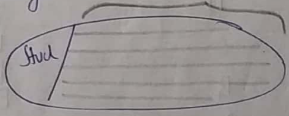
Unsafe TRC Query (Imp)

TRC query with infinite records in result

$\{ T / T \notin Stud \}$
$\{ T / \neg T \in Stud \}$
} Unsafe TRC Query

Retrieves set of tuples (T) those are not belongs to student relation.

(Infinite Tuples)



T: Universal

(Imp) Expressive Power Comparison (Gate)

[Expressive Power of Safe "TRC" Queries] $\Longleftrightarrow$ [Expressive power of "basic RA" Queries]

Basic RA Queries

RA queries which can formulate (express) using $\{\pi, \sigma, \times, \rho, \cup, -, \cap, \bowtie, \bowtie_c, ⟕, ⟗, /\}$ operators

Queries failed to express using basic RA :-

⇒ Count of records, Count of attribute values, Sum of attribute values, Avg of attribute values

NOTE: min, max can be expressed in RA

⇒ Ordering of records in Result of Query etc

P-54 T-4

family (parent, child, child DOB)

parent	child
G	D
D	C
C	B
B	A
A	Amy

C
↑
B
↑
A
↑
Amy

$$\pi_{T_2.parent}\left(\sigma_{T_1.child = Amy \wedge T_1.parent = T_2.child}\left(\rho(T_1, family) \times \rho(T_2, family)\right)\right) \cup$$

$$\pi_{T_3.parent}\left(\sigma_{T_1.child = Amy \wedge T_1.parent = T_2.child \wedge T_2.parent = T_3.child}(T_1 \times T_2 \times T_3)\right)$$

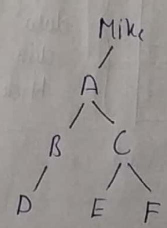
$$\vdots$$

$$\pi_{T_n.parent}\left(\sigma_c(T_1 \times T_2 \times T_3 \cdots \times T_n)\right)$$

Queries failed to express using RA
(or)
Not constant length RA query

Here we need to know the length of generation to determine how many "X" we need to do. and for that we need count() , which is not possible to ~~expressive~~ express in RA.

parent	child
mike	A
A	B
A	C
B	D
C	E
C	F
G	H



$$\pi_{child}\left(\sigma_{parent = Mike}(family)\right) \cup$$

$$\pi_{T_2.child}\left(\sigma_{T_1.child = T_2.parent}(T_1 \times T_2)\right) \cup$$

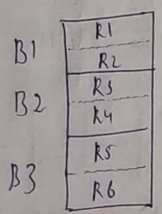
$$\vdots$$

$$\pi_{T_n.child}\left(\sigma_c(T_1 \times T_2 \times T_3 \ldots T_n)\right)$$

Not constant length RA Query

File Organization & Indexing

DB is collection of files (Tables)
Files is collection of blocks (pages)
Block is collection of records



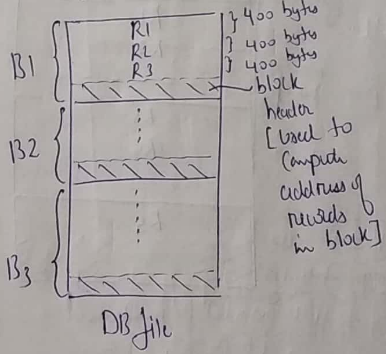
Stored in Disk
data access from disk to main memory block by block

Records in DB file can be

1) Fixed length Records :-

All records in DB file same size

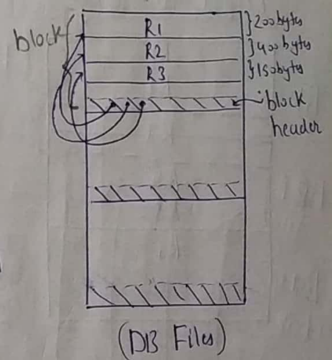
```
Create table R
(
  A char (100),
  B char (100),
  C char (200).
);
```



2) Variable length Records :-

All records of DB are variable length

```
Create table S
(
  D char (100)
  F text
);
```



ERD N
IC
S/w developer

Index TAC
* DBMS Component
RAD

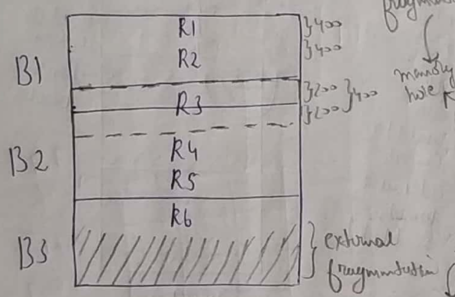
Records Organization in DB file

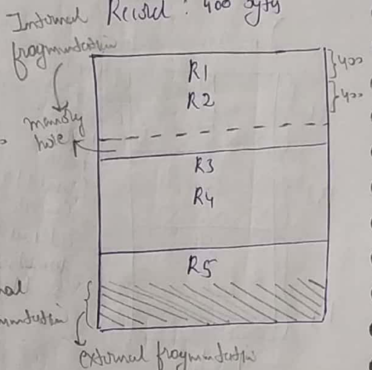
Record Organization can be of 2 types

i) Spanned ~~Organization~~ Organization

ii) Unspanned Organization

i) Spanned Organization	ii) Unspanned Organization
=> Record allowed to span in more than one block	Complete record must be in a block
Block: 1000 bytes Record: 400 bytes	Block: 1000 bytes Record: 400 bytes





=> ~~Ao~~ More no. of access i.e. (more access cost) to access span records	=> Require less access cost. Easy to organize.
=> Possible to allocate DB file without internal fragmentation.	May not possible to avoid internal fragmentation in DB file
Spanned Organization preferred to store DB file with variable length records.	Unspanned Organization preferred to store DB file with fixed length records.

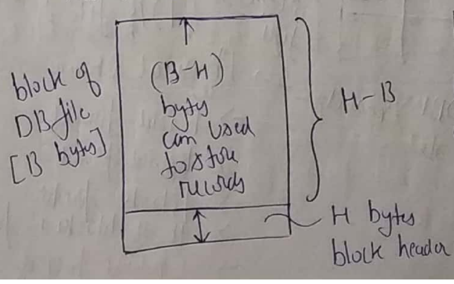
NOTE:- Spanned Organization for fixed length records is not preferred bcz there will be lot of internal fragmentation. And if unspanned Organization is not preferred for variable length records

X

Block factor

max no. of records ~~per block~~ that can be stored in DB block

Record Size: R bytes [Fixed length]



Block factor of DB file $= \left\lfloor \frac{B-H}{R} \right\rfloor$

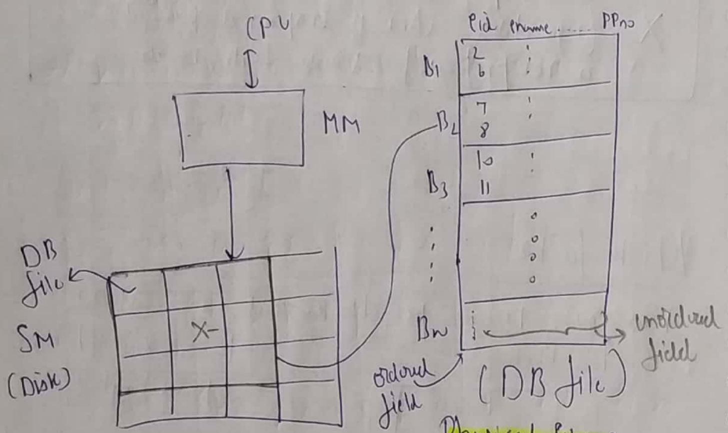
$$\boxed{\text{block factor of DB file} = \left\lfloor \frac{B-H}{R} \right\rfloor \text{ records / block}}$$

[bcz of unspanned organization]

I/O Cost [Access Cost]

of SM [Disk] blocks required to transfer from Disk to Main Memory in order to access required record.

(SM = Secondary Memory)



DBMS S/W maintain → Physical Schema
- ⇒ Field format
- ⇒ record format
- ⇒ Block factor
- ⇒ # of blocks in file
- ⇒ Ordered field of file

I/O Cost to access record from DB file with n block without index

a) based on Ordered field

```
Select *
FROM Emp
Where eid = x;
```
(eid: ordered field)

} $\lceil \log_2 n \rceil$ blocks access cost

binary search can be applied.

~~n = # of records~~

b) based on Unordered field

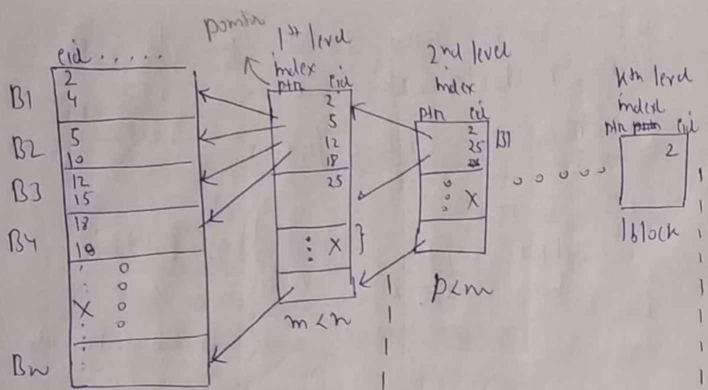
```
Select *
From Emp
Where PPno = 4;
```
(PPno: unordered field)

} n blocks access cost

NOTE: For flat file system cost to access record is always n blocks.

Indexing

Index :- used to reduce I/O cost to access data records from DB file



⇒ $\lceil \log_2 m \rceil + 1$ block access cost using 1st level index

⇒ k+1 blocks access cost to access record using multilevel index

Index file :-

Each entry of index file has two fields

$\boxed{\text{(Search key, pointer)}}$

Set of (key, pointer) pairs in index block

⇒ **Block factor of index file**

max possible (key, pointer) pairs can store in one block

block size : B bytes

Search key : K bytes

pointer size : P bytes

block header : H bytes

block factor of index file $= \left\lfloor \frac{(B-H)}{K+P} \right\rfloor$ index entries / block

Multilevel index :-

index to index until last level must be only one block index.

(k+1) blocks access cost to access record using multilevel index when indexing done upto kth level.

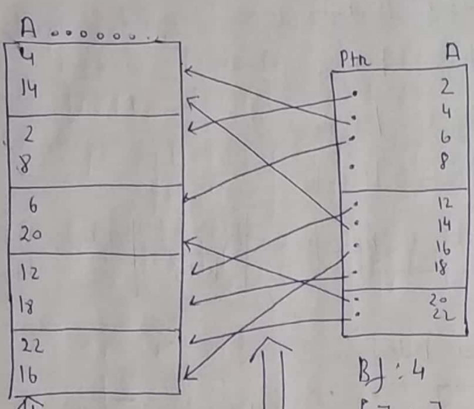
Categories of Index

1) ~~Dense~~ Dense Index

For each DB file record there exists entry in index file.

$\boxed{\text{\# of index file entries} = \text{\# of records of DB file}}$

19/11/18



Bf : 4
[Index]

Unordered file of DB file

Dense index

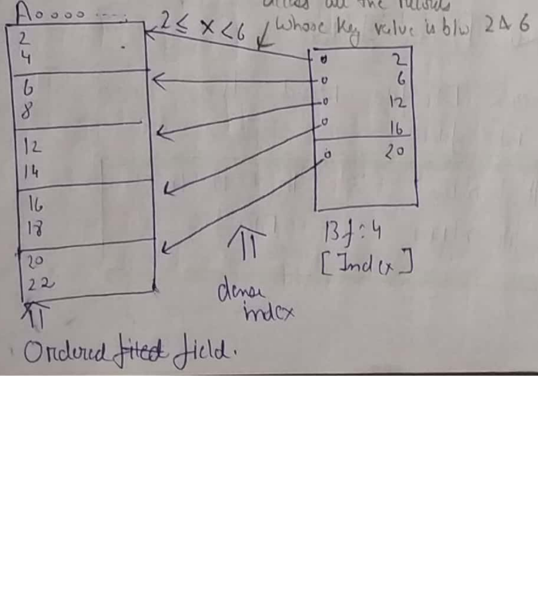
Dense index can be over ordered field (or) unordered field of DB file.

2) Sparse Index

for set of DB records there exists entry in index file.

Using this pointer we access all the records whose key value is b/w 2 & 6

$2 \le x < 6$



Bf : 4
[Index]

dense index

Ordered ~~filed~~ field.

→ Sparse index possible only over ordered field of DB file.

Sparse index entries < # of DB records.

if sparse index over key field

$$\boxed{\text{\# of index file entries} = \text{\# of blocks of DB file}}$$

Q) Consider DB file with 100,000 records block size 1024 bytes. Record size 100 bytes
Search key ~~pointer~~ 12 bytes. pointer : 15 bytes.

i) I/O cost to access record without index
 a) over ordered field
 b) over unordered field

ii) if sparse index used
 *** a) # of 1st level index blocks ?
 b) I/O cost using 1st level index ?

iii) if dense index used
 *** a) # of 1st level index blocks ?
 b) I/O cost using 1st level index ?.

Sol: $n = 100000$

block size = 1024 bytes

record size = 100 bytes

$Bf = \left\lfloor \frac{\text{block size}}{\text{record size}} \right\rfloor = \frac{1024}{100} = \lfloor 10.24 \rfloor = 10$ records/block

# of blocks / DB file = $\left\lceil \frac{100,000}{10} \right\rceil = 10,000$

~~a) Over ordered field~~

~~log~~

ii) if sparse index used

# of index file entries = # of blocks = 10,000

a) 1st level index block = $\left\lceil \frac{10,000}{37} \right\rceil = \lceil 270.27 \rceil$

$= 271$ blocks

index field size = 12 + 15

= 27 bytes

$Bf = \left\lfloor \frac{1024}{27} \right\rfloor = 37$ ~~records~~ index field / blocks

b) I/O cost $= \lceil \log_2 271 \rceil + 1$

$\lceil 8.08 \rceil + 1$

$= 9 + 1$

= 10 block access

iii) if cluster index used

# of index file entries = # of records of DB file

= 100,000

$B.f = \left\lfloor \frac{1024}{27} \right\rfloor = 37$ index file entries / block

# of 1st level index blocks = $\left\lceil \frac{100,000}{37} \right\rceil = \lceil 2702.2 \rceil$

= 2703 blocks

b) I/O cost = ~~$\lfloor \log_2 2703 \rfloor + 1$~~

$= \lceil \log_2 2703 \rceil + 1$

$= \lceil 11.4 \rceil + 1$

$= 12 + 1$

= 13 block access

i) Without index

a) Over ordered field

~~$\lceil \log_2 \lfloor 100,000 \rfloor \rceil$~~ $\lceil \log_2 10,000 \rceil$

$= \lceil 13.28 \rceil$

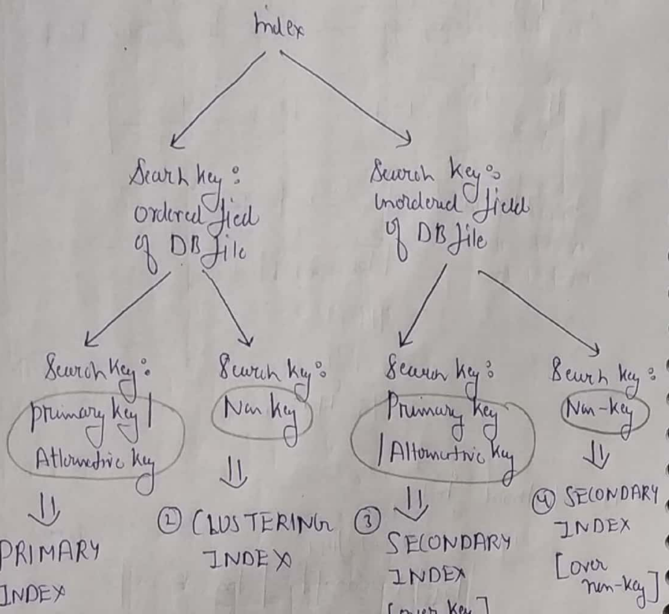
= 14 blocks access

b) Over unordered field

10,000 blocks access

Types of index

Search key : field of DB file used for indexing

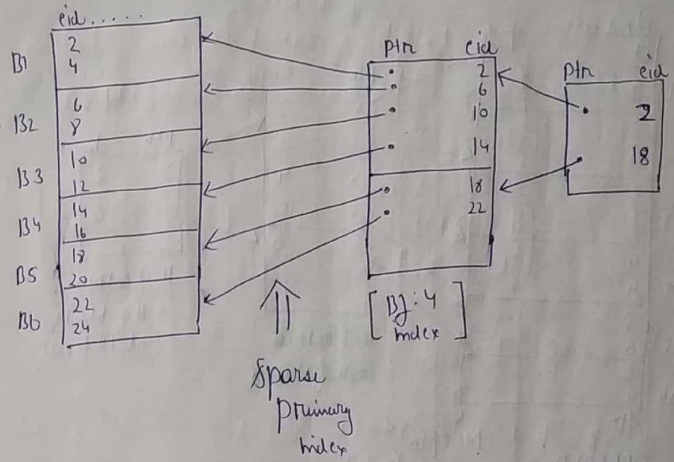


NOTE :-

different types of indexing is only possible at level 1, from next level we have sparse indexing. (ie for every block at level 1 index, entry is maintained at level 2 index.

Primary Index :-

Search key : (ORDERED FIELD) and (KEY)



→ I/O cost to access record using PI with MLI : (k+1) blocks.

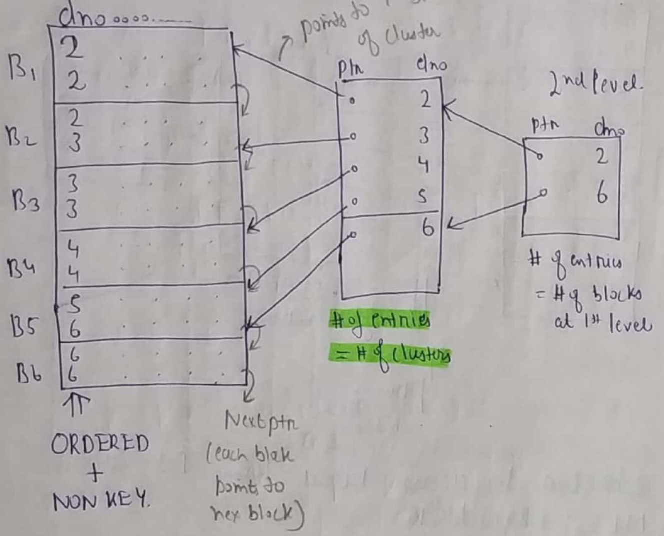
→ Primary index can be either dense (or) sparsed indexed. ∴ [sparse PI is preferred bcz file size is small]

→ For any DB relation at most one primary index is possible.

Scanned by CamScanner

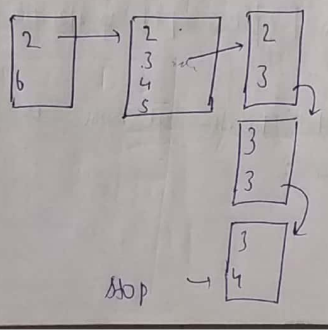
Clustering Index

Search key : (ORDERED field) and (non key)



I/O Cost

Select *
FROM Emp
Where dno=3

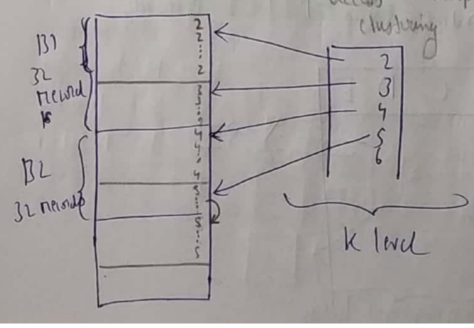


I/O cost to access cluster of records using clustering index with multilevel index is
{ k + one or more blocks access from DB file until first block of next cluster }

Q Cluster size : 16 records
block size factor of DB : 32 records
I/O cost to access cluster using CI with k level MLI?

a) k+1 blocks ✓ b) k+2 blocks
c) k blocks d) k+3 blocks

Sol. each contain = 32 records.
# of cluster / block = $\frac{32}{16}$ = 2 cluster
Using CI with k level MLI
I/O cost = k+2 | In worst case we might have to access multiple blocks bcz of clustering



→ Clustering index mostly sparse index
[Clustering index can be dense if each cluster with one record]

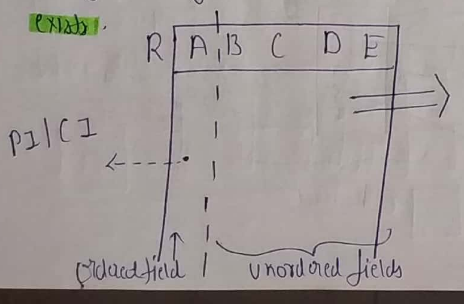
→ For any DB relation at most one clustering index is possible, because search key is ordered field.

→ For any DB relation either PI or CI is possible but not both.

Secondary Index :-

Search key : (Unordered field)
(and)
Key | Non key

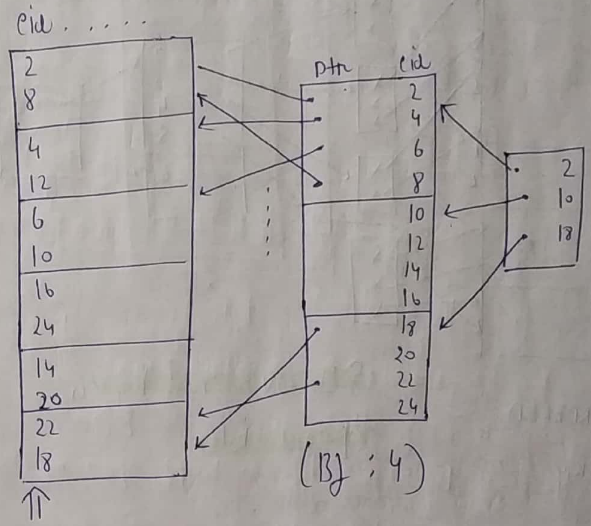
SI is alternative index to access data from DB file using index, even PI/CI already exists.



⇒ Many secondary indexes possible for any DB relation.

i) SI Over key :-

1st level index must be dense index



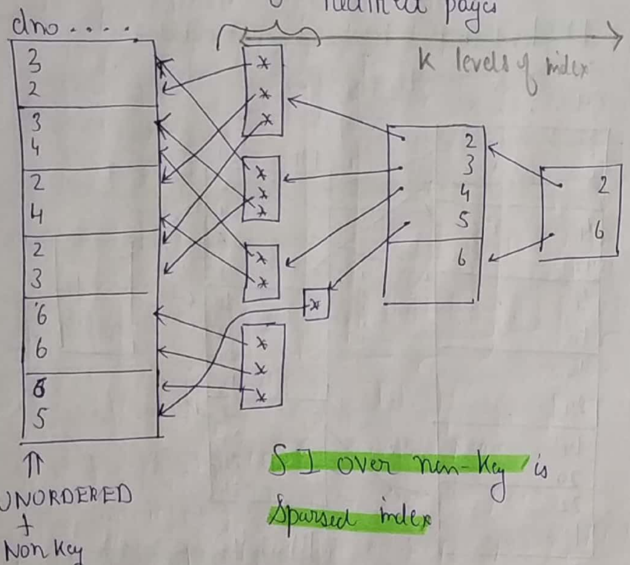
(Bf : 4)

Unordered field + key

I/O Cost to access record using SI over key with MI is (k+1) blocks.

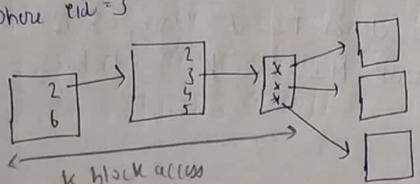
Secondary index over key is always dense index

iii) SI over non key :-



SI over non-key is Sparsed index

Select *
FROM Emp
Where eid = 3



# of block access is # of pointers in the redirect page.

I/O cost to access record using SI over non-key with Multi level index is

k + # of ~~blk~~ pointers in redirect pages

~~k+~~ [# of blocks of DB file access]

Imp
P-65 (Work book)

Q11

	*** SI over key (a)	*** PI (b)	CI (c)	SI over non key (d)
# of levels of index	1191	37	74	74
# of index blocks	3	2	2	2
I/O cost	k+1 blocks 4	k+1 blocks 3	3	19

Sol.. (X^2, Y) — key, non key

1
1
2
2
3
.
.

⇑ unordered — ordered.

records = 16536
size of record = 32 bytes
size of search key = 6B
blocksize = 1024 bytes
pointer size = 12 bytes

block factor for DB file
$\Rightarrow \left\lfloor \frac{1024}{32} \right\rfloor = 2^5 = 32$

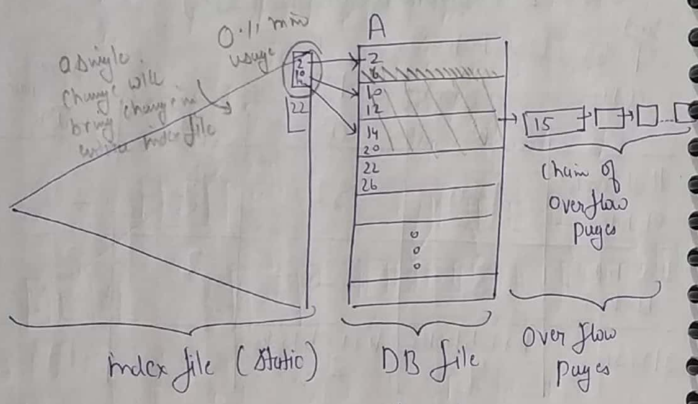
block factor for ~~disk~~ indexfile

$\left\lfloor \frac{1024}{18} \right\rfloor = 56$

Scanned by CamScanner

Static Multilevel Index

Index file not allowed to modify even data base records inserted / deleted



Insertion of records : In overflow pages
I/O cost increases to access overflow pages
Deletion of records : left as Empty space in DB file

Disadvantages of Static MLI

Because of chain of overflow pages in worst case access time can be O(n)
[Almost full DB access]

→ Because of records deletion from DB file Minimum usage of index block can be 0 %.

Dynamic MLI :-

→ No overflow pages used
→ Index file should update base database file records insertion / deletion / updation
→ Standard data structures used for Dynamic MLI
i) B Tree
ii) B+ Tree

B Tree and B+ Tree are balanced Search Tree.

Max height of search tree for n distinct should not exceed $O(\log n)$
[Height restricted search tree]

Advantage of balanced search tree is in worst case TC to search a key is $O(\log n)$

Scanned by CamScanner

balanced Search Tree

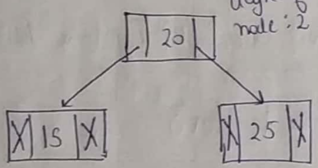
- AVL Tree
- Red-black Tree
- B Tree
- B^+ tree

Used for DB index bcz no restriction on degree of node

AVL Tree, Red-black Tree

→ balanced binary Search Tree

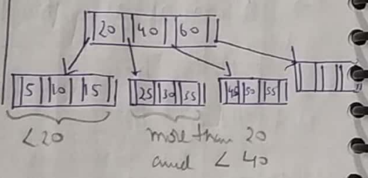
degree of node : 2



B Tree, B^+ tree

→ balanced Search Trees

degree of node : p



Imp Why B/B+ index used for DB file rather than balanced binary search trees (Gate)

DB file in Disk [SM] 44

Index file to DB file also must be in disk

Data access from disk block by block

If AVL Tree used for DB index

} One block should allocate for one AVL Tree node

More levels of index

More I/O cost to access records

More wastage disk space allocated for index

If B/B+ tree index used

Then

Disk block allocated for one B/B+ Tree node

→ less I/O cost bcz less levels index

→ based on block size order P can compute for B/B+ Tree node such that less wastage of Disk space which is allocated for index.

Main Memory Search Application

n keys should store same DS in MM

∆∆ DS should be use for mainly search operation

⇒ less # of keys } AVL tree DS preferred

n : 100 keys

Huge # of keys
n : lacks of keys } B Tree DS is preferred

If 10 lacks keys and AVL Tree used
then # of pointers required = 20 lacks.
~~hence~~ hence pointers become overhead.
therefore we go for B Tree DS.

Secondary Memory Search Application

[Search over Disk block of data]
(DB Indexing)

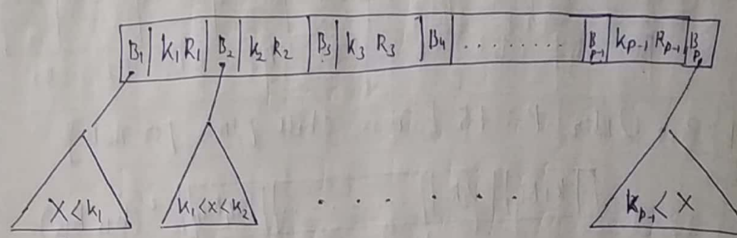
less # of keys
n : 100 keys
Huge # of keys
n : lack of keys } B+ tree DS preferred

20/11/18

B - Tree definition :-

Order P : max possible child pointers (block pointers) can store in B Tree Node.

1) Node Structure :- P Block ptr (P-1) keys & (P-1) Record ptr



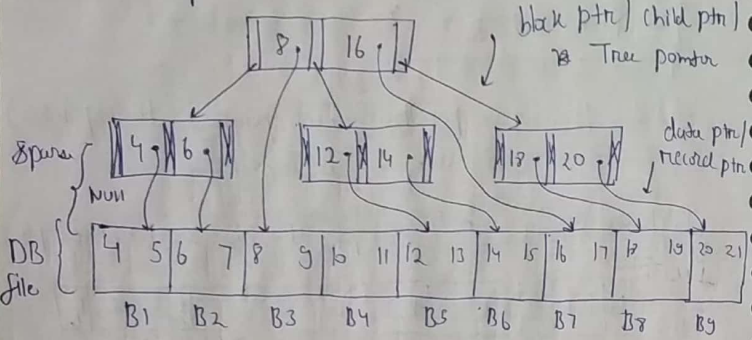
Balancing Conditions

2) Every internal node except root must be at least $\lceil P/2 \rceil$ block ptrs & $(\lceil P/2 \rceil - 1)$ keys
at most p block ptrs & (p-1) keys

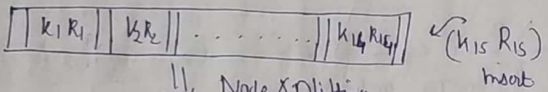
3) Root can be at least 2 block ptrs & 1 key
at most p block ptrs & (p-1) keys

4) All leaf nodes must be at same level.

p=3

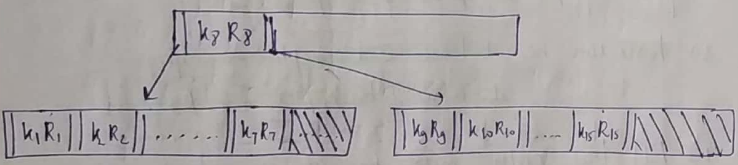


EX2 Order P : 15 [max child ptr's per node]



⇓ Node Splitting

$$\underbrace{k_1 < k_2 \ldots k_7}_{\text{left child}} < \underbrace{k_8}_{\text{middle key become root}} < \underbrace{k_9 < k_{10} \ldots k_{15}}_{\text{right}}$$



Min

$\lceil \frac{P}{2} \rceil = 8$ block ptr

& 7 keys

Every node min 50% should be filled

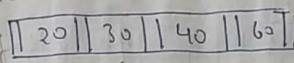
Construct B Tree with Order P : 5

(max bp per node) and sequence of

keys :- 40, 60, 20, 30, 80, 90, 25, 10, 15, 50, 70, 28, 35, 38, 5, 12, 18

So/n			
max	5bp	4k	Internal nodes
min	3bp	2k	

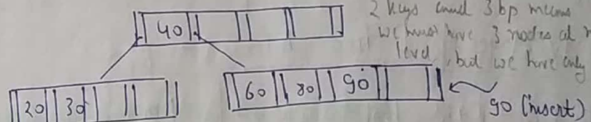
Keys are always present in ascending order in a node.



⇓ Splitting

20 30 (40) 60 80

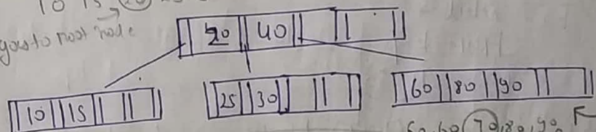
insertion of new node is always done at leaf node b/z if insertion done at internal node, then we have 2 keys and 3 bp means we must have 3 nodes at next level, but we have only 2 nodes



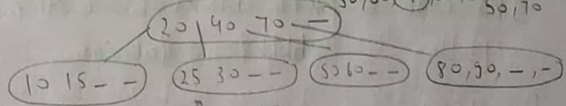
Insert (25, 10, 15) splitting

10 15 (20) 25 30

20 goes to root node

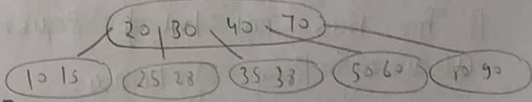


50, 60 (70) 80, 90 ← 50, 70



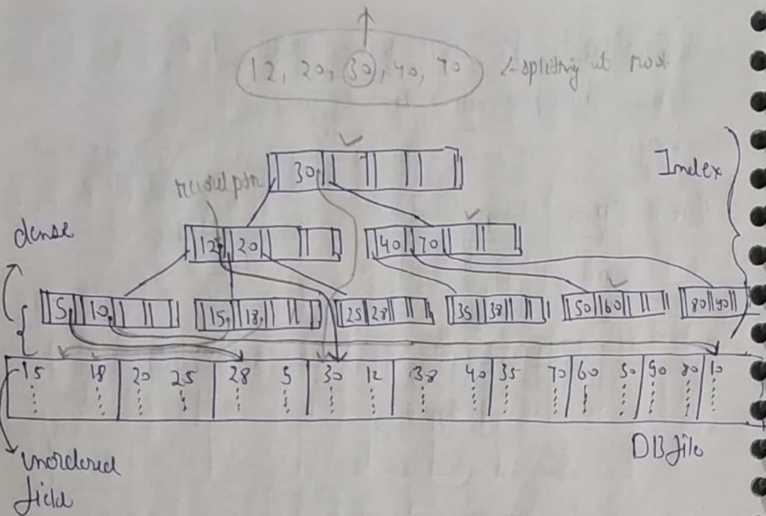
28, 35, 38

25, 28 (30) 35, 38



5, 12, 18

5 10 (12) 15 18



Advantage of B Tree index :-

B Tree index best suitable for random access of any one record.

Ex: Select *
FROM Emp
Where A = 50; (one record access)

I/O cost = (k+1) blocks in Worst case
2 blocks in Best Case

Disadvantage of B Tree index

B Tree index not best for Sequential access of Range of records.

Ex: Select *
FROM R
Where A ≥ 25 and A ≤ 60; (range of records access)
X records

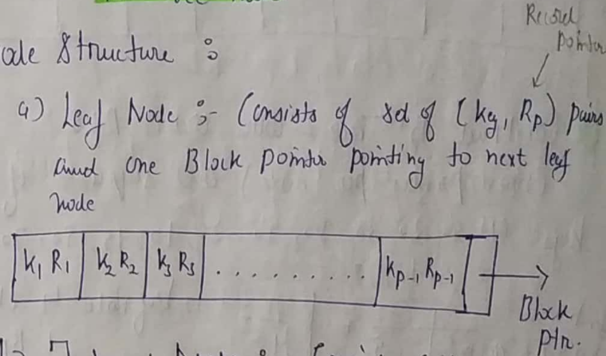
I/O Cost = X [k+1] blocks in Worst Case
$\Theta(k \cdot X)$ k = Level of indexing

B+ Tree definition :-

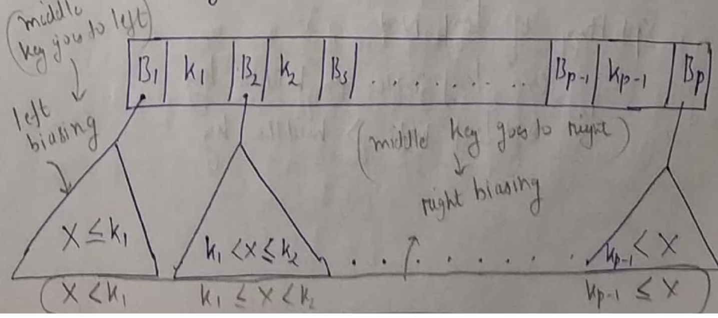
Order P: max possible pointers can store in B+ Tree node.

1) Node Structure :

a) Leaf Node :- Consists of set of (key, R_p) pairs and one Block pointer pointing to next leaf node

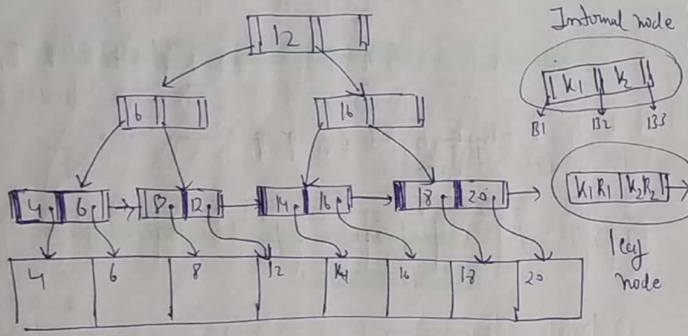


b) Internal Node :- Consists only keys && block pointers



Scanned by CamScanner

Balancing Conditions of B+ Tree same as B Tree



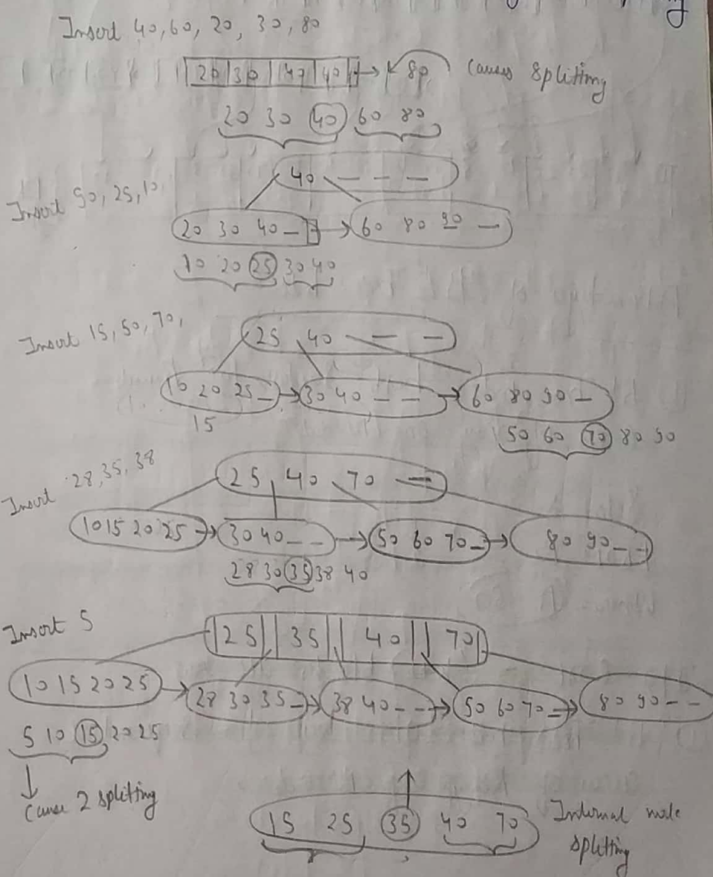
Q Construct B+ Tree with order P: 5 (max pointers per node) and sequence of keys: 40, 60, 20, 30, 80, 90, 25, 10, 15, 50, 70, 28, 35, 38, 5, 12, 18

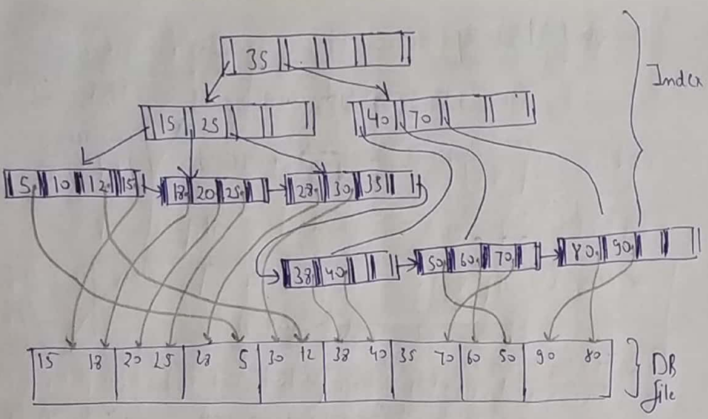
	Internal node	leaf node
max	5	4
min	3	2

leaf node 4 (K, Rp) pair
1 Bp

Splitting of leaf Node :-
one leaf node ≤ middle key
2nd leaf node > middle key

Internal Node Splitting :-
one Internal node < middle key
2nd Internal node > middle key
} Same as B Tree Node splitting





Advantages of B+ Tree Index

① B+ Tree index best suitable for Random access of any one record

Select *
FROM R
Where A = 50;

one record access

I/O cost = (k+1) blocks all cases

② B+ Tree index also best for sequential access of Range of records.

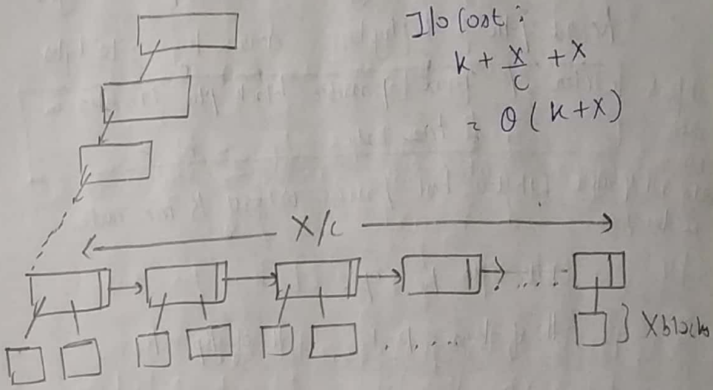
Select *
FROM R
Where A ≥ 25 and A ≤ 60

X records

I/O cost:

$k + \frac{X}{c} + X$

$= O(k+X)$



If left biasing used, keys stored in internal nodes are maximum key of each leaf node except last ~~leaf~~ leaf node

$\boxed{\text{\# of keys stored in internal nodes = \# of leaf nodes} - 1}$

If right biasing used, keys stored in internal nodes are minimum key of each leaf node except First leaf node

***

Q Find best possible order of B/B+ tree node for given block size

Model-I

Q1 block size : 1024 bytes   block ptr : 9 bytes

Record ptr : 12 bytes   Search key : 10 bytes

default order when order not specified in the question

$\boxed{\text{Order } p: \text{max possible block ptr can store in B tree node}}$

What is best possible order of B Tree node.

Sol:

Let order be p.

# of keys = p−1   # of Bp = p

$9*p + 10(p-1) \le 1024$

$9p + 10p - 10 \le 1024$

$19p \le 1034$

$p \le \frac{1034}{19}$

$p \le \lfloor 54.42 \rfloor$

$p = 54$

$9*p + (10+12)(p-1) \le 1024$

$9p + 22p - 22 \le 1024$

$31p \le \cancel{1034}\ 1046$

$p \le \frac{\cancel{1034}\ 1046}{31}$

$p \le \lfloor 33.35 \rfloor$ $p = 33$

Concept:- One disk block is allocated to one node

$\boxed{p * B_p + (p-1)(k + R_p)}$ } block [B Tree node]

Order p : max Bp per node

Q2 block : 512 bytes

Search key : 5 bytes   block ptr = Record ptr = 10 bytes

i) $\boxed{\text{Order } p: \text{max possible keys can store in B Tree node}}$

What is best possible order of B Tree node?

Sol:

p → (k, Rp) pairs

(p+1) → Bp

$10*(p+1) + (5+10)(p) \le 512$

$10p + 10 + 15p \le 512$

$25p \le 502$

$p \le \frac{502}{25}\ 20.8$

$p \le \lfloor 20.8 \rfloor$

$p = 20$

ii) Order p: b/w 1 to 2p keys for Root

b/w p to 2p keys for other nodes of B Tree

What is best possible order of B Tree node?

Sol. ~~First Root node~~ Order is taken based on max order

$B_p(2p+1) + (2p)(k + R_p) \le \text{Block}$

$10(2p+1) + (2p)(5+10) \le 512$

$20p + 10 + \cancel{15p}\ 30p \le 512$

$50p \le 502$

$p \le \frac{502}{50}$ 10.04

$p \le 10.04$

$p = \lfloor 10.04 \rfloor$

$p = 10$

Concept :- node size is ~~size is~~ decided based on maximum order possible.

Order : p : max 2p keys per node

Q3

Q) block : 1024 bytes   block ptr : 9 bytes

Record ptr : 12 bytes   Search key : 10 bytes

Order p : max possible pointers can store in B^+ Tree node.

What is best possible order of B+ tree

i) Internal node   ii) leaf node

Sol: Internal node

$\boxed{p * Bp + (p-1)k \le Block}$

$p*9 + (p-1)10 \le 1024$

$19p - 10 \le 1024$

$p \le \frac{1034}{19}$

$p \le 54.42$

$p = \lfloor 54.42 \rfloor = 54$

ii) leaf node

$\boxed{(p-1)(k+Rp) + Bp \le block}$

$(p-1)(10+12) + 9 \le 1024$

$22p - 22 + 9 \le 1024$

$22p \le 1024 + 13$

$p \le \frac{1037}{22}$

$p = \lfloor 47.136 \rfloor$

$p = 47$

~~if size of (Bp) = size of (Rp Then (B+ Tree Internal~~

Q4) Block : 512 bytes.

Search Key : 5 bytes $\boxed{\text{Block ptr} = Rp = 10 \text{ bytes}}$

i) Order p : max possible keys can be stored in B+ Tree node

What is best possible order of B+ Tree node ?

Sol: Concept : if size of (Bp) = size of (Rp)

Then (B+ tree Internal node order) = (B+ tree leaf node order)

using definition of internal node.

$\boxed{(p+1)Bp + p(k) \le block}$

$(p+1)10 + p(5) \le 512$

$15p \le 502$

$p \le \frac{502}{15}$ 33.7

$p = \lfloor 33.7 \rfloor = 33$ [max keys per node]

iii) Order p: b/w 1 to 2p keys for Root

b/w p to 2p keys for Other nodes of B+ Tree.

What is best possible order of B+ tree?

Sol: Using definition of internal node

$(2p)\,k + (2p+1)\,B_p \le \text{block}$

$(2p)\,5 + (2p+1)\,10 \le 512$

$30p \le 502$

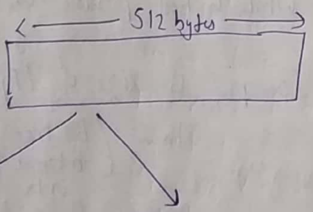
$p \le \frac{502}{30}$

$p \le 16.733$

$p = \lfloor 16.733 \rfloor$

$p = 16$

$R_p = B_p = 10$



B Tree Node
Order p: 20
[max 20 keys per node]

B+ tree node
Order p: 33
[max 33 keys per node]

Imp

If Disk block size allocated for B Tree && B+ tree Nodes are same size

Then (1)

(1) Order P of B Tree Node < Order P of B+ Tree Node

(2) (# of B Tree Nodes for n distinct keys) > (# of B+ Tree Node for n distinct keys)

# of index blocks — # of index blocks

(3) # of levels of B Tree index for n distinct keys (I/O cost) > # of levels of B+ Tree index for n distinct keys (I/O cost)

For DB Index B+ Tree preferred than B Tree because B+ Tree index access cost less than B Tree index access cost for both Random access queries && Range Query

Malte-2

21/11/18

Q Find max/min keys in B/B+ Tree for given l levels [Hight h]

Q Find max keys/Nodes in B/B+ Tree with Order P : 5 [max Cp per node] and 4 levels?

child pointer

Sol Order P : 5

max	5 Bp	4 k
min	3 Bp	2 k

	level	max node	Max Bp	Max keys
Root	1	1	5	4
	2	5	5*5	4*5
	3	25	25*5	25*4
leaf	4	125	~~125*5~~	125*4

here all Bp points to Null

Max possible node in B/B+ tree of 4 levels = 156

Max keys possible in B Tree of 4 level = 624.

Max keys in B+ Tree of 4 level = 500 (bcz all keys are present at leaf node)

Q2) Find min key/nodes in B Tree/B+ Tree with order P : 5 and 4 levels. Assume order P : b/w 2 to p bp's for Root

b/w $\lceil \frac{P}{2} \rceil$ to p bps for other internal nodes

b/w $(\lceil P/2 \rceil - 1)$ to $(p-1)$ keys for leaf nodes.

Order P : 5 ⇒ Min 2 Bp && 1 key for root

Min 3 Bp && 2 keys for Other internal node

	level	min Node	min Bp	min keys
Root	1	1	2	1
	2	2	2*3	2*2
	3	6	6*3	6*2
leaf	4	18	=	18*2

min Node possible in B/B+ tree = 27

min keys possible = 53 for B tree

min keys for B+ Tree = 36 of 4 levels

Q3) Order P : b/w 1 to 2p keys for Root

b/w p to 2p keys for other nodes except root.

What for order p : 5 and level 4 B/B+ Tree

i) min keys/nodes ?

ii) max keys/nodes

Sol: Root : 2 bp 1 k

Other node : 3 bp 2 K

level	min node	min Bp	min key
1	1	2	1
2	2	2*3	2*2
3	6	6*3	6*2
4	18	=	18*2

min node = 27
min key Btree = 52
min key B+tree = 36

ii) max nodes/keys

root: 2p bps 2p-1 keys
10 bps 9 keys
Other nodes: 10 bps 9 keys

level	max node	min Bp	min keys
1	1	10	9
2	10	10x10	10x9
3	100	100x10	100x9
4	1000		1000x9

max node = 1111
max key in B Tree = 9999
max key in B+ Tree = 9000

i) Order : p = 5

root : 2 Bp 1K
Other node : 6 Bp 5K

level	min node	min Bp	min key
1	1	2	1
2	2	2*6	2*5
3	12	12*6	12*5
4	72	=	72*5

min node = 87
min keys B Tree = 430
min keys B+ Tree = 360

ii) max nodes / keys

Order p : 9 Bps 10 keys

level	max node	max Bp	max keys
1	1	11	10
2	11	11*11	11*10
3	121	[illegible]	99*10
4	901	=	901*10

max node = 1012

level	max nodes	max Bp	max keys
1	1	11	10
2	11	11*11	11*10
3	221	221*11	121*10
4	1331	=	1331*10

max nodes 1464

Max keys in B+Tree =

ii) order p:5

root : 11 Bp 10k

internal node 11Bp 10k

level	max node	max Bp	max keys
1	1	11	10
2	11	11*11	11*10
3	121	121*11	121*10
4	1331	=	1331*10

max nodes = 1464

max keys in B Tree = 14640

max keys in B+ tree = 13310

Q Find max keys in B+ Tree with order P : 5 [max keys per node] and 4 levels?

Sol: Order p: 5   6Bp   5k

	level	max node	max Bp	max keys
root	1	1	6	5
	2	6	6*6	6*5
	3	36	36*6	36*5
leaf	4	216	=	216*5

(B Tree) max keys = 1295

max nodes = 259

max keys in B+ tree = 1080

Bulk loading B+ Tree :- (Marks 3)

Construction of B+ Tree in bottom up approach (ie leaf to Root Construction)

1. Sort the keys which are used for indexing in ASC order.

~~2. Design le~~

2. Design leaf nodes :- [based on keys]

distribute keys into leaf nodes based on design criteria.

[Min keys per node or max keys per node].

result : # of leaf nodes

3. Design Internal Nodes :- [based on Block ptrs]

If "m" nodes at level "l"

Then m child ptrs at level (l-1)

Distribute m child ptrs into Nodes based on design criteria

[min Bp per Node (or) max Bp per Node]

result : # of Nodes at level "l-1"

Repeat Until Root

Ex. Order P: 5 [max pointers per node]

keys : 40, 60, 20, 30, 80, 90, 25, 10, 15, 50, 70, ~~80~~ 28, 35, 38, 5, 12, 18.

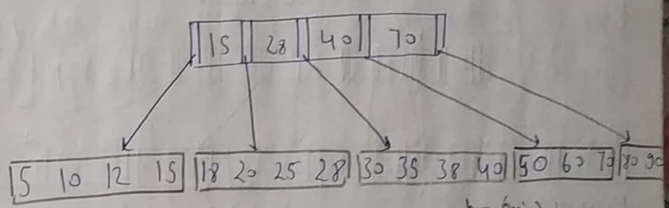
Construct bulk loading B+ Tree with

i) Min levels ? // Max keys per node for given keys (bcz min level is asked we go for max keys per node)

Min : 2 keys

Min : 6 B+ Tree nodes

Here we have 5 nodes at previous level Here we need only 5 Bptr



$\left\lceil \frac{17}{4} \right\rceil = 5$ Nodes

↑ max keys per node

bcz (min) → 3 Bp 2 K.
we should not violate min condition. Hence we take ceil

ii) Max levels ?

bcz max level is asked so we go for min keys per node

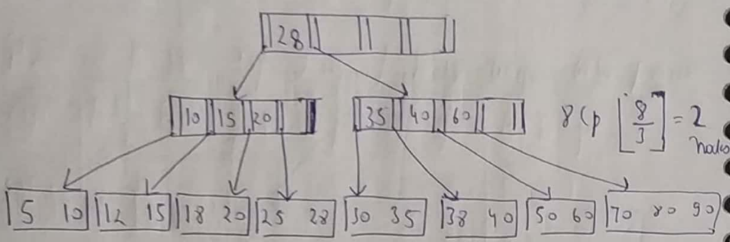
Order P : 5 [Max Pointers]

min $\left\lceil \frac{P}{2} \right\rceil = 3$ ptr 44 2 keys

For Given keys

→ Max 3 levels of index

→ Max 11 B+ Tree Nodes



$\left\lfloor \frac{17}{2} \right\rfloor$ = 8 nodes

min 2 keys per node

We don't take ceil fun bcz we need to satisfy min key per node criteria

| 70 80 | | 90 | ← not permissible

Hence here we take floor fun!

Q> Given 2500 keys with order p : 7 [max ptrs B in B+ Tree nodes]

i) How many min levels && min nodes of B+ Tree?

ii) How many max levels && max nodes of B+ tree?

Sol. min levels // we go for max keys per node

order p : 7 7 Bptr 6 keys

$\left\lceil \frac{2500}{6} \right\rceil = 417$ — (leaf) l # of nodes = 489

$\left\lceil \frac{417}{7} \right\rceil = 60$ l-1 # of levels = 5

$\left\lceil \frac{60}{7} \right\rceil = 9$ l-2

$\left\lceil \frac{9}{7} \right\rceil = 2$ l-3

$\left\lceil \frac{2}{7} \right\rceil = 1$ l-4

ii) max levels // we go for ~~max~~ min keys per node

Order p : 7 4 Bptr 3 keys

$\left\lfloor \frac{2500}{3} \right\rfloor = \lfloor 833.33 \rfloor = 833$ — leaf

$\left\lfloor \frac{833}{4} \right\rfloor = \lfloor 208.25 \rfloor = 208$ — l-1

$\left\lfloor \frac{208}{4} \right\rfloor = 52$ → l-2 # of levels = 6

$\left\lfloor \frac{52}{4} \right\rfloor = 13$ → l-3 # of nodes = 1110

$\left\lfloor \frac{13}{4} \right\rfloor = 3$ → l-4

$\left\lfloor \frac{3}{4} \right\rfloor = 1$ → l-5

Scanned by CamScanner

if # of keys given To find Min/Max levels of B+ Tree, ~~can be~~ ~~used~~ use bulk loading B+ Tree design

if # of levels given, To find min/max keys design Root to ~~leaf~~ leaf

Hence, Bottom-Up design / bulk design only possible for B+ Tree [Not possible for B Tree].

JOIN Algorithm

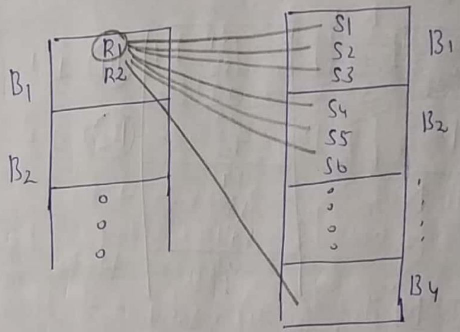
i) Nested loop Join
ii) block Nested loop Join

① Nested loop Join

Assume Relation R with n tuples occupied X blocks
Relation S with m tuples occupied Y blocks

Nested loop Join Algorithm uses record numbers of R & S as loop Variable.

```
R ⋈ S => for (i=1; i≤n; i++)
{
    for (j=1; j≤m; j++)
    {
        Join (ith Record of R,
              jth Record of S)
    }
}
```



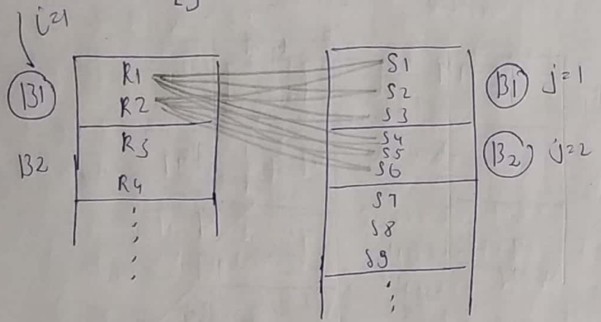
Disadvantage: More I/O cost to join R & S if Main Memory space allocated for join is limited.

② Block Nested loop Join Algorithm

R with n tuples in X blocks
S with m tuples in Y blocks
block Nested loop join Algorithm uses block numbers of R & S as loop variable

$R \bowtie S \Rightarrow$ for (i=1 ; i ≤ X ; i++)

```
{
    for (j=1 ; j ≤ Y ; j++)
    {
        Join (ith block records of R,
              jth block records of S)
    }
}
```

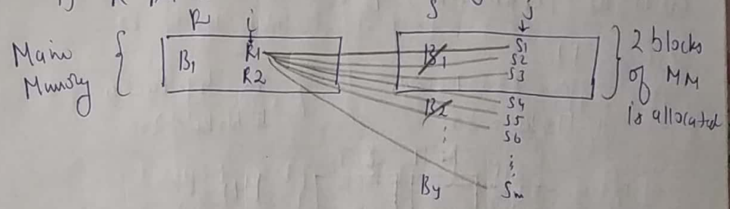


Advantage :

~~Yess~~ less I/O cost compare to Nested loop Join algorithm if MM space allocated for Join is very limited.

Assume only two blocks of MM allocated to store records of R & S. to perform Join at any time

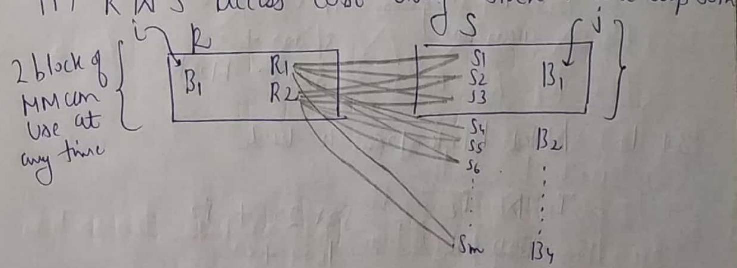
i) $R \bowtie S$ Access Cost using Nested loop Join



"Y" blocks of S Rel access cost for one record Join of R

$$\boxed{\left.\begin{matrix} R \bowtie S \\ \text{access cost} \end{matrix}\right\} = (X + nY) \text{ blocks}}$$

ii) $R \bowtie S$ access cost using Block Nested loop Join



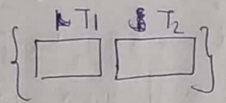
"Y" blocks of S Rel access cost for block Join of R.

$$\boxed{\left.\begin{matrix} R \bowtie S \\ \text{access cost}\end{matrix}\right\} = (X + XY) \text{ blocks}}$$

Q 18-19 (workbook) P-66

Sol: T1 — $n = 2000$ records, # blocks = 80

T2 — $m = 400$ records, # blocks = 20



Sol:-18) Nested loop join is used

$$\left.\begin{matrix} T_1 \bowtie T_2 \\ \text{access cost}\end{matrix}\right\} = 2000 * 20 + 80 = 40{,}000 + 80 = 40{,}080 \text{ blocks}$$

$$\left.\begin{matrix} T_2 \bowtie T_1 \\ \text{access cost}\end{matrix}\right\} = 400 * 80 + 20 = 32000 + 20 = 32{,}020 \text{ blocks}$$

Sol: 19) Nested block join is used

$$\left.\begin{matrix} T_1 \bowtie T_2 \\ \text{access cost}\end{matrix}\right\} = 80 * 20 + 80 = 1600 + 80 = 1680 \text{ blocks}$$

$$\left.\begin{matrix} T_2 \bowtie T_1 \\ \text{access cost}\end{matrix}\right\} = 20 * 80 + 20 = 1620 \text{ blocks}$$

reduction in # of block access = 32,020 − 1620 = 30,400

Q 20 P-66

Sol: # records of R < # of records of S

if Nested ~~join~~ loop join

$R \bowtie S$ preferred

$S \bowtie R$ X

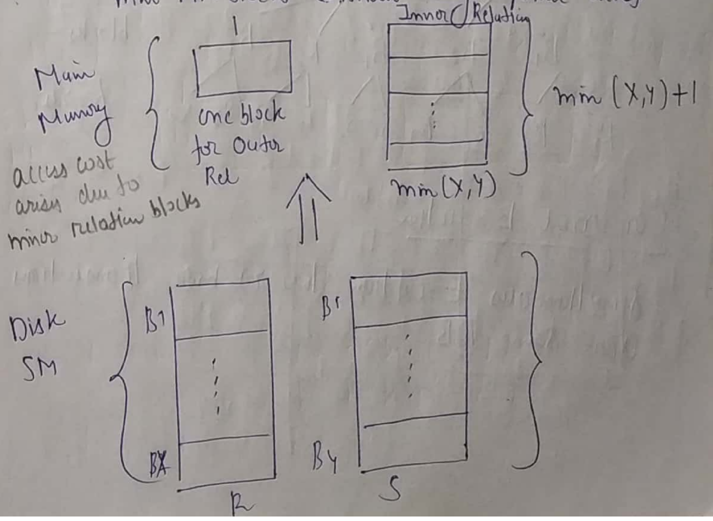
(Imp)

Q 36 P-68

Sol: T1 — # $n = 2000$ records, # disk blocks = 80

T2 — # $m = 400$ records, # disk blocks = 20

min MM blocks without any repeated access

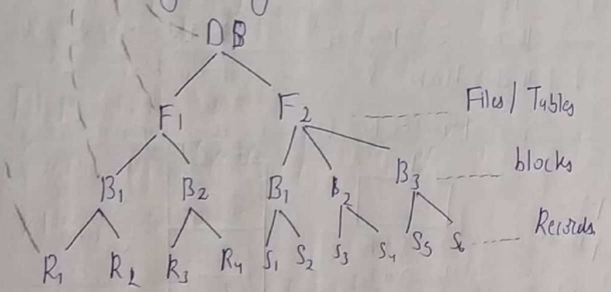


Transaction && Concurrency Control :-

Transaction :- logically related R/W operations to perform unit of works.

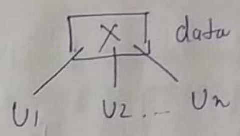
Data Item : [Shared Resource]

database element which may required to access by many transactions.



Concurrent Execution

Simultaneous Execution two (or) more transactions over same DB.



Concurrent Execution may lead inconsistency.

Concurrency Control :-

. Avoid inconsistency because of concurrent execution of many transaction over same DB.

Degree of Concurrency :-

# of users can use DB simultaneously.

Goal :

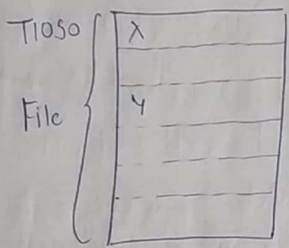
Design concurrency controller such that it should provide more degree of concurrency && should avoid inconsistency because of ~~Concurrency~~ concurrent execution

Flat File System

[OS File] no concurrency

because concurrency is control by O.S, it locks the data over file level.

[File] : Resource



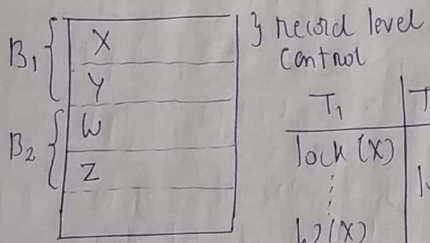
T_1	T_2
lock (file)	
W(X)	
	lock (file)
	{W(Y)} denied

Degree of Concurrency very less using Flat File System because Concurrency control at File level.

DBMS file System

Concurrency control possible over record level.

[record] : resource.



T_1	T_2
lock (X)	
	lock (Y)
W(X)	
	W(Y)

More degree of Concurrency using DBMS file System because Concurrency Control over record level.

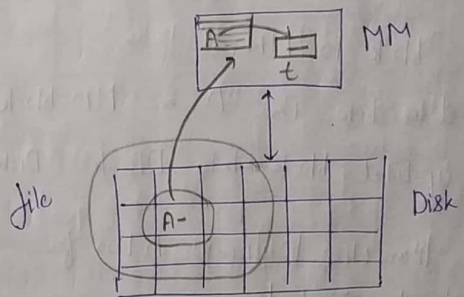
* degree of Concurrency

$\boxed{\text{over record level} > \text{Over block level} > \text{over file level} > \text{over DB}}$

Main Operations :-

Read (A) : [A : data item]

accessing of data item "A" value from (disk) DB file to main memory to use value of A in transaction logic.



Read (A)

atomic
1. Find block which contain data item "A" in DB file
2. Transfer block to MM.
3. File address of data item "A" in MM & copy value of A into programmed variable

Write (A) :-

Updation of data item "A" in DB file [disk]

Write (A) :-

atomic
1. find block of DB file (DISK) which contain A.
2. Transfer block to MM.
3. Update ~~data~~ data item "A" in MM block
4. Replace updated block into DB file (Disk)

Ex: Consider a transaction to transfer Rs 1000 from account 101 to account 102

```
begin Trans T1
    Update Account bal = bal-1000;
    Where Aid = 101;
    Update Acount bal = bal +1000;
    Where Aid = 102;
End Trans;
```

|| SQL Parser / SQL Executer

```
begin trans T1
Trans (T1)
    Read ( bal of Aid = 101);
    bal = bal - 1000;
    Write ( bal of Aid = 101);
    Read ( bal of Aid = 102);
    bal = bal + 1000;
    Write ( bal of Aid = 102);
    Commit ; // Transaction executed successfully
    end transaction ;
```

```
begin Trans T1
    Update Account bal = 5000
    Where Aid = 101;
~~end Transact T1~~
End Trans;
```

blind write

Trans (T_2)

begin trans
    Write (bal : 5000 where aid = 101)
    Commit
end trans

Trans (T_3)
R(A)
R(B)
W(A)
W(C)
W(D)
Commit ;

Set of R/W opⁿ

W(C), W(D) } blind write

NOTE :

To preserve integrity [Correctness]

Transactions must satisfy ACID rules

ACID Properties

A : Atomicity
D : Durability } Maintained by "Recovery Management Component" of DBMS s/w.

{ C : Consistency } Maintained by user [DB administrator / DB developer]

user

I : Isolation : } Maintained by "Concurrency Control Component" of DBMS s/w.

$\boxed{\text{A, D I } \} \rightarrow \text{maintained by DBMS s/w.}}$

A : Atomicity :-

Execute all operations of transactions including commit

(OR)

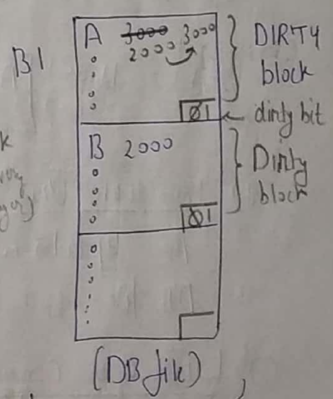
Execute none of the operations of transaction to terminate transaction (Roll back)

Trans (T_1) : Transfer 1000 from A to B

begin Trans (T_1
    R(A)
    A = A - 1000 ;
    W(A)
    R(B)
    B = B + 1000
    W(B)
    Commit ;
end Trans

Trans failed

Roll back (Recovery Manager)



Data in DB file is incorrect bcz atomicity failure

Responsibility of Recovery Management Component :-

→ RMC rolls back the transaction which fails any where before commit.

Roll back (Abort) :-

~~Undo~~ Undo modification of DB file which are done by failure transaction

Transaction log :-

File is maintained by RMC in disk to record all activities of transactions

W(A) block { A: — }

There are two ways to maintain transaction log

① Immediate Write (Protocol)
"A" updates in log file && also updates in DB file.

② Write after Commit (Protocol)
"A" update only in log file, after commit during redo updates in DB file

Trans log

T_1: begin
T_1: R, A, 3000
T_1: W, A, 3000, 2000
T_1: R, B, 2000
T_1: W, B, 2000, 3000
T_1: commit

log entry format

T_1 : Write, data item, old val, new val

Dirty block :- DB block which is updated by uncommitted transaction.

Redo Operation :- DB changes which are required because of transaction commit.
[all changes of DB file bcz of trans. commit]

1. If Trans. uses Write immediate, then change dirty block to non dirty block
2. If Trans. uses Write after Commit ~~use~~ then DB file should update based on new values of log entry.
3. Clean log entries of transaction

Undo Operation :- All changes of DB file because of trans. roll back.

① If Write Immediate Used

Then 1. Write Old value of log into DB file && change Dirty block to Non Dirty block

Otherwise

No changes in DB file

② Clean log entries of transactions

Drawback

T_1 : Commit $\Rightarrow T_1$ redo

T_2 : Commit $\Rightarrow T_2$ redo

T_3 : Commit $\Rightarrow T_3$ redo

⋮

Overhead of redo becomes more if redo performs for each commit.

Thus redo cost is more

To overcome this drawback we go for check point.

Consistency Check Point :-

→ redo not performed after immediate commit if DBMS issues check point

Then
- All transactions which are in committed state until previous check point will perform redo.
- All trans of roll back state until previous check point will perform Undo

9:00 AM Check point
- T_1 : Commit
- T_2 : Roll back
- T_3 : begin
- T_4 : Commit

9:05 AM Check point → T_1, T_4 : redo T_2 : Undo
- T_5 : begin
- T_6 : Commit
- T_7 : rollback

9:10 AM Check point → T_6 : redo T_7 : Undo
- T_8 : begin
- T_5 : Commit
- T_9 : Commit

System Crash T_5, T_9 : redo
T_3, T_8 : Undo

Crash Recovery

=> All Committed transaction ~~undtt~~ until previous check point perform redo.

=> All Uncommitted transaction in entire system change state to roll back & perform undo.

Q: Consider log of entries

1. T_1 begins
2. T_1; W, A, 0, 20
3. T_2 begin
4. T_2; W, B, 5, 10
5. T_2 commit
6. Check point
7. T_3 begins
8. T_3; W, B, 10, 20
9. T_4 begins
10. T_4: W, C, 50, 60
11. T_4 commit
12. System crash

What happens because of system crash?

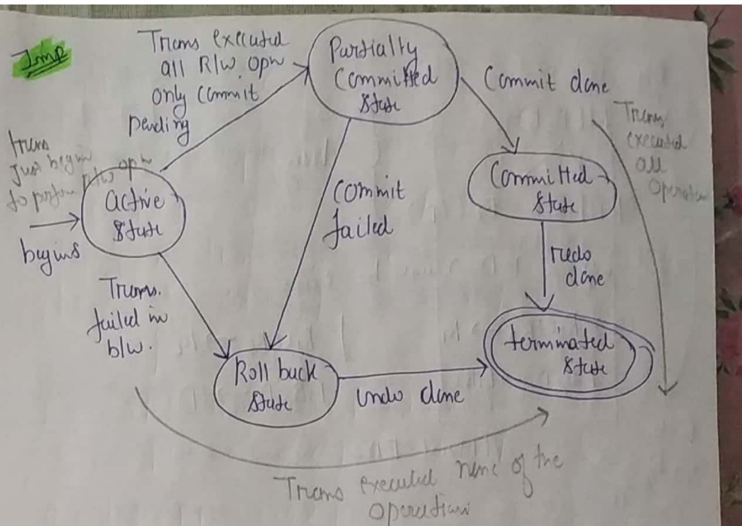
a) T_2 T_4 redo T_1 T_3 undo

b) T_4 redo T_1 T_3 undo ✓

c) T_1 T_2 redo T_3 T_4 undo

d) T_2 redo T_1 T_3 T_4 undo

Imp



Normalization (Work book)

Q6

$S = \{A_1 A_2 \dots A_n\}$ (with $A_1 A_2$ marked T)

$T \subseteq S$

T: Cand key

1. $T \rightarrow S - T$ $\{A_1 A_2 \rightarrow A_3 A_4 \dots A_n\}$

2. $\exists P \subset T$ $P \rightarrow S - P$
   ⇑ proper subset of CK $\{A_1 \rightarrow A_2 A_3 \dots A_n\}$ ✗

3. $\forall Q \supseteq T$ $Q \rightarrow S - Q$
   ↑ Super key $\{A_1 A_2 A_3 \rightarrow A_4 \dots A_n\}$ ✓

Q14

Sol: R(ABCD) → R_1(ABC), R_2(CD) } : lossy

Which FD must be exists

a) $R_1 \cup R_2 \rightarrow R_1$ | $R_1 \cup R_2 \rightarrow R_2$

ABCD → ABC | ABCD → CD

Trivial FD's

Q37) R & S

R	S	
✓	✓	LLJ
✓	✓	3NF
✓	may not	DP
✗	✓	BCNF

22/11/18

D: Durability

Transaction should able to recover under any case of failure.

During Redo && Undo if failure occurs ; ~~the~~ Transaction should recover by performing redo && undo again

Reasons for Transaction failure

1. Power failure
2. S/W Crash
   → DBMS restarted
   → OS restarted
3. DBMS Concurrency controller may kill transaction

log file && Transaction states stored in DISK

4. H/W Crash
   Disk Crash

Redundant Array of Independent disk ← RAID design of DISK should be used

Used to recover from disk crash.

every failure should able to Rollback : Durability

C: Consistency [Maintained by USER]

DB operations [SQL Queries] requested by user to perform transaction must be logically correct.

Transaction T_1

A+B by exec^n of T_1

equal

A+B after exec^n of T_1

Trans (T_1)
R(A)
A = A - 50
W(A)
R(B)
B = B + 50
W(B)
Commit

Transfer 50 from A to B.

Isolation :-

[Maintained by Concurrency Control Component]

Concurrent execution of two or more transactions results must be equal to result of any serial execution of transaction. (Serializable Schedule)

Schedule :- Time Order execution ~~execution~~ sequence of two or more transactions.

Time Order exec^n of T_1, T_2, T_3

S :	T_1	T_2	T_3
	$r_1(A)$		
	$w_1(A)$		
		$r_2(A)$	
			$w_3(A)$
	$r_1(B)$		
	$w_1(B)$		
		$r_2(B)$	
			$w_3(B)$

Serial Schedule

Transactions should execute one after other.

T_1 : Transfer 500 from A to B

T_2 : Display Total balance of A, B.

T_1	T_2
R(A)	
A = A - 500	
W(A)	
R(B)	
B = B + 500	
W(B)	
Commit	
	R(A)
	R(B)
	Commit

[T_1 : T_2 Serial]

followed by

T_1	T_2
	R(A)
	R(B)
	Commit
R(A)	
W(A)	
R(B)	
W(B)	
Commit	

[T_2 : T_1 Serial]

Scanned by CamScanner

Adv of Serial ~~schro~~ schedule

Result of any serial schedule always preserves integrity (correctness)

Disadv :- Less degree of Concurrency

less Throughput.

Resource Utilization not good.

Response time is more

Concurrent Schedule

→ ~~Transactions~~ (Concurrent / Simultaneously / Interleaved execution of two or more transactions.

→ Serial / Non-serial Schedule (ie all possible schedules)

A: 2000
B: 3000

T_1	T_2	
R(A)		
W(A)		
	R(A)	: 1500
R(B)		
W(B)		
C1		
	R(B)	: 3500
	C2	5000

Equal to $T_1 : T_2$ Serial
[Isolation Satisfied]

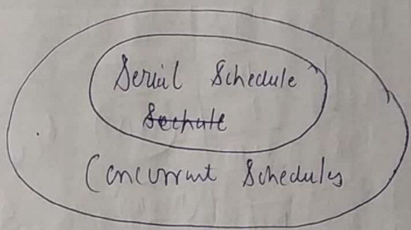
T_1	T_2	
R(A)		
W(A)		
	R(A)	; 1500
	R(B)	; 3000
	C2	4500
R(B)		
W(B)		
C1		

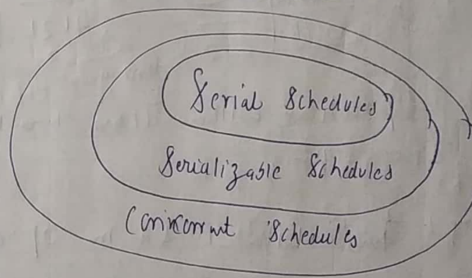
← Not equal to $T_2 : T_1$ serial

~~Incorr~~ Incorrectness bcz of Concurrent Execn.

Not Equal to $T_1 : T_2$ serial.

Correct Schedule equal to $T_1 : T_2$ Serial	Incorrectness because of Concurrent execution
Isolation Satisfied	Not equal to $T_2 : T_1$ Serial
(Serializable Schedule)	Not equal to $T_1 : T_2$ Serial
	(Isolation failed)
	(Non-Serializable Schedule)





NOTE:-

n! possible ~~Sch~~ Serial Schedule for given n transactions

Modle-I

Q T_1: $r_1(A)$ $w_1(A)$ $r_1(B)$ $w_1(B)$

T_2: $r_2(A)$ ~~w2~~ $r_2(B)$

How many possible ~~concurrent~~ concurrent schedule ~~sets~~ b/w T_1 & T_2.?

Sol:

S	T_1	T_2
1	$r_1(A)$	
2	2	$r_2(A)$
3	$w_1(A)$	
4		$r_2(B)$
5	$r_1(B)$	
6	$w_1(B)$	

$^6C_4 * ^2C_2$

$= \frac{6!}{4!\,2!}$ [Concurrent Schedule]

arrangement eliminate T_1 & T_2

T_1 T_2 Operations within Trans ~~must~~ must in same sequence

Q T_1 T_2 Trans with n & m operations

~~#~~ # of concurrent schedule $= \frac{(n+m)!}{n!\,m!}$

$^{n+m}C_n \cdot ^mC_m$

Q T_1 T_2 T_3 trans with n, m, p operations respectively.

# of concurrent schedule $= \frac{(n+m+p)!}{n!\,m!\,p!}$

$^{n+m+p}C_n * ^{m+p}C_m * ^pC_p = \frac{(n+m+p)!}{n!\,m!\,p!}$

Q Process P, Q with

P : P_1, P_2

Q : Q_1, Q_2

Instruction Set

How many interleaving Execution possible b/w P, Q ?

Sol: $\frac{(2+2)!}{2!\,2!} = 6$ (or) $^4C_2 * ^2C_2$

Q ~~A: a_1, a_2, a_3 sorted order~~

~~A~~ A, B arrays with 3 & 5 sorted elements each. How many possible results if A, B merges into single sorted array?

a_1	a_2	a_3

$a_1 < a_2 < a_3$

b_1	b_2	b_3	b_4	b_5

$b_1 < b_2 < b_3 < b_4 < b_5$

C

a_1	_	a_2	_	a_3	_	_	_

$^8C_3 \cdot ^5C_5 = 56$

(Imp)

Classification of Schedules based on Serializability :- [Serializability testing]

1. Conflict Serializable Schedule
2. View Serializable Schedule

1) Conflict Serializable Schedule :-

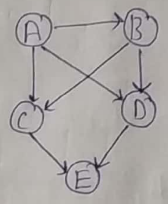
i) topological order
ii) Conflict pair
iii) Conflict Equal Schedules
iv) Precedence Graph

Topological Order
[Graph Traversal Algorithm only for (DAG) Directed Acylic Graphs]

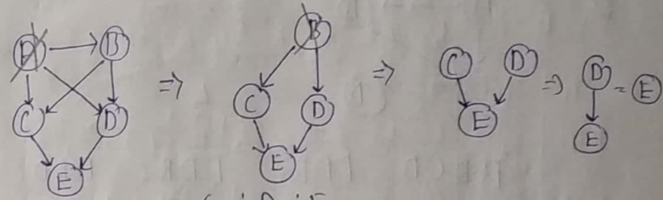
1. Visit Vertex(V) whose Indegree '0' && delete "V" from Graph
2. Repeat ① for all Vertices of Graph

Visited Order : Topological Sorting (order)

Ex:

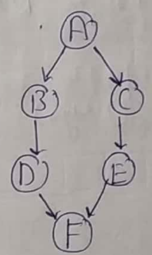


Topological Sort :- AB CDE, ABD CE } topological orders of the graph



A : B < C : D : E ; D : C : E

Ex:



Find # of topological orders?

A < BCDE~~AB~~F ; CBDE F

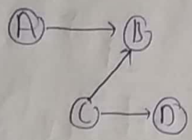
A < BDCEF ; CEBDF

A < BDCEDF ; CBDE F

[A _ _ _ _ E]
BD CE
$^4C_2 * ^2C_2$
$= 6$

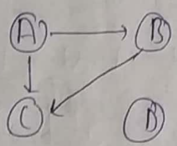
D will always be followed by B

Q3



ACDB CDAB
ACBD CADB CABD

Q4

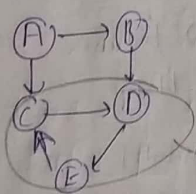


ABCD DABC ADBC ABDC
~~ABC~~

Q5) How many topological orders in NULL Graph of n vertices
(Null graph : n vertices 0 edges)

Sol $n!$

Q5)



~~ABCD~~
~~ACBD~~
~~AB..~~

→ cycle exists not DAG
topological sort not possible

Conflict pair

Two operations from schedule (S) is conflict pair iff

1) at least one write operation
&&
2) Over same data item
&&
3) from different transactions

S: ... $r_i(A)$ .... $w_j(A)$ ....
S: .... $w_i(A)$ ..... $r_j(A)$ ....
S: .... $w_i(A)$ ...... $w_j(A)$ ....
} Conflict pairs

S: ...... $r_i(A)$ ..... $r_j(A)$ ......
S: .... $r_i(A)/w_i(A)$ ..... $r_j(B)/w_j(B)$ ....
} Non-Conflict pairs

S: $r_1(A)\ r_1(B)\ w_2(A)\ w_1(B)\ w_2(B)$

Non conflict pairs
$r_1(A)\ r_1(B)$
$r_1(A)\ w_1(B)$
$r_1(B)\ w_1(B)$

T_1	T_2
$r_1(A)$	
$r_1(B)$	
	$w_2(A)$
$w_1(B)$	
	$w_2(B)$

Conflict pairs

(Operations from same tran. in schedule (S) neither conflict pair nor non conflict pair)

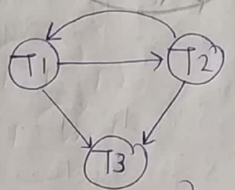
Precedence Graph :-

Vertices [V] : Transactions of Schedule

Edges [E] : Conflict pair precedence of schedule.

S : $r_1(A)\ w_2(A)\ w_1(B)\ w_1(A)\ w_3(B)\ w_3(A)$

T_1	T_2	T_3
$r_1(A)$		
	$w_2(A)$	
$w_1(B)$		
$w_1(A)$		
		$w_3(B)$
		$w_3(A)$



Precedence Graph of S

Cycle exists hence non-serializable

Conflict Equal Schedules :-

S_1 S_2 schedules conflict equal iff S_2 derived by interchanging of consecutive non-conflict pair of S_1.

S1 : $r_1(A)\ w_2(A)\ w_1(B)\ w_1(A)\ w_3(B)\ w_3(A)$

S2 : $r_1(A)\ w_1(B)\ w_2(A)\ w_3(B)\ w_1(A)\ w_3(A)$

Conflict equal

S1 S2 conflict equal iff

ie S1 & S2 must have same transaction set.

1) each trans of S1 must be exactly same in S2.

S1 : .... T_i .... $r(A)$ $r(B)$ $r(B)$ $w(B)$

S_2 : .... T_i .... $r(A)$ $r(B)$ $w(B)$

(AND)

2) Every conflict pair precedence of S1 must be same precedence in S2.

Q S_1 :- $r_1(A)\ w_1(B)\ r_2(A)\ w_2(B)\ r_3(A)\ w_3(B)$

S_2 $r_2(A)\ r_3(A)\ r_1(A)\ w_1(B)\ w_2(B)\ w_3(B)$

S_1

T_1	T_2	T_3
$r_1(A)$		
$w_1(B)$		
	$r_2(A)$	
	$w_2(B)$	
		$r_3(A)$
		$w_3(B)$

S_2

T_1	T_2	T_3
	$r_2(A)$	
		$r_3(A)$
$r_1(A)$		
$w_1(B)$		
	$w_2(B)$	
		$w_3(B)$

Conflict Equal

T_1, T_2, T_3 are same for S_1 S_2

Conflict pair precedence is same for S_1 & S_2

NOTE:-

⇒ If S_1 & S_2 are conflict equal schedules Then precedence graph of S_1 & S_2 same.

⇒ If S_1 & S_2 schedules with not same precedence graph S_1 and S_2 not conflict Equal.

⇒ If S_1 & S_2 with same precedence graph then S_1 & S_2 may or may not conflict Equal.

S_1 : $r_1(A)\ w_2(A)\ w_1(A)\ r_2(A)$ precedence not same

S_2 : $r_1(A)\ w_2(A)\ r_2(A)\ w_1(A)$

although same precedence graphs but not conflict equal.

NOTE:

If S_1 and S_2 schedules

i) Same set of transactions

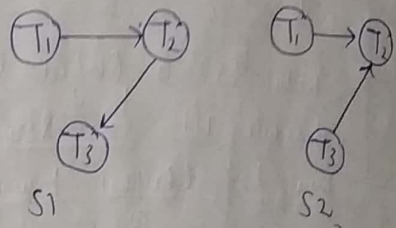
(and) ii) Same precedence graph.

(and) iii) Acyclic Precedence graph.

Than S_1 and S_2 are Conflict Equal Schedules

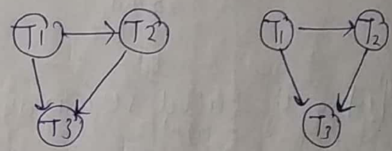
To test S_1 S_2 conflict Equal

Case 1:



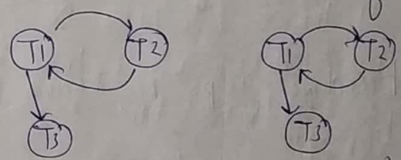
S1 & S2 are not conflict equal.

Case 2:



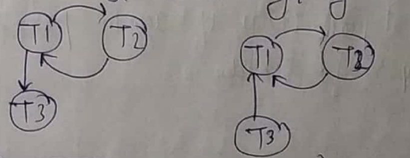
S1 & S2 are conflict equal

Case 3:



S1 & S2 may/may not conflict equal

Case 4:



S1 & S2 not conflict Equal.

1) Conflict Serializable Schedule

Schedule (S) is Conflict Serializable Schedule iff some serial schedule (S') must be Conflict Equal to S.

~~S' (Given Schedule)~~ $\quad S \xLeftrightarrow[\text{equal}]{\text{Conflict}} \exists S'$: Serial Schedule

S (given schedule)

Testing of Conflict Serializable Schedule :-

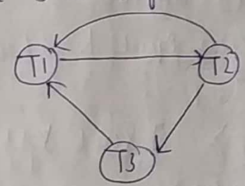
Schedule (S) is Conflict Serializable Schedule iff

- Precedence graph of schedule (S) must be acyclic
- Conflict equal serial schedules are topological orders of acyclic precedence graph of schedule S.

Q $S: r_1(A)\ w_1(A)\ w_2(A)\ w_2(B)\ r_3(B)\ w_1(B)$

Test S is conflict serializable schedule or not

Sol:



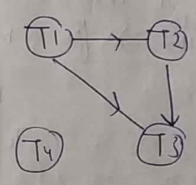
not conflict serializable

Q $S: r_1(A)\ r_2(A)\ r_3(A)\ r_4(A)\ w_1(B)\ w_2(B)\ w_3(B)$

i) Test S is conflict serializable or not?

ii) How many serial schedule conflict equal to S

Sol:- i)



Sol:- ii)

Topological sort: _T_1_T_2_T_3 T_4

Serial schedule conflict equal to given schedule (S):

$T_4\ T_1\ T_2\ T_3$ $T_1\ T_2\ T_4\ T_3$

$T_1\ T_4\ T_2\ T_3$ $T_1\ T_2\ T_3\ T_4$

23/11/18

View Serializable Schedule

Schedule (S) is View Serializable iff some Serial Schedule (S') View equal to Schedule (S)

Schedule [S] $\xleftrightarrow[\text{equal}]{\text{View}}$ ∃ Serial Schedule (S')

View Serializable Schedule

View Equal Schedule :-

S1 & S2 Schedules View equal iff

1) Initial Read :-

If transaction Ti reads X from ~~initial~~ initial database in S_1 then in Schedule S_2 also T_j must read X from initial database

initial DB — initial read

S1 : ....Ti..... S_2 : ......Ti....

R(X)   R(X)

(AND) 2) Update Read :-

S_1 : .... Ti .... Tj .... S_2 : .... Ti .... Tj ....

R(X) ← W(X)   R(X) ← W(X)

If Ti reads X which is updated by Tj in S1 then in S2 also Ti should read X updated by Tj.

S1: T_1	T_2	T_3	S2: T_1	T_2	T_3
R(A)					
		R(A)			
		W(A)			
	R(A)				
R(B)					

(AND) 3) Final Write :-

S_1 : ..... Ti ..... S_2 : ..... Ti .....

W(X)   W(X)

final write of X by Ti   final write of X by Ti

Scanned by CamScanner

If in Schedule S_1, transaction T_i does final write over data X, then in S_2 also T_i must do final write over data X.

Ex: S_1:

T_1	T_2	T_3
R(A)		
		R(A)
		W(A) ← final write
	R(A)	
R(B)		
	W(B)	
		R(B)
		R(C)
W(C)		
	W(C) ← final write	
W(D)		
	W(D)	
		W(D)

S_2:

T_1	T_2	T_3
		R(A)
R(A)		
R(B)		
		final write W(A)
	R(A)	
	W(B)	
		R(B)
		R(C)
W(C)		
	W(C) ← final write	
	W(D)	
W(D)		
		W(D)

$S_1 \xleftrightarrow[\text{Equal}]{\text{View}} S_2$

Q Test Given S_1 & S_2 View Equal or not

1) S_1: $r_1(A)$ $r_2(A)$ $w_1(A)$ $w_2(A)$

S_2: $r_1(A)$ $w_1(A)$ $r_2(A)$ $w_2(A)$

S_1

T_1	T_2
$r_1(A)$	
	$r_2(A)$
$w_1(A)$	
	$w_2(A)$

S_2

T_1	T_2
$r_1(A)$	
	~~$w_2(A)$~~
$w_1(A)$	
~~$r_2(A)$~~	$r_2(A)$
	$w_2(A)$

Not View Equal

2) S1: $r_2(A)$ $r_2(B)$ $w_2(A)$ $r_1(C)$ $w_1(A)$ $r_3(C)$ $r_3(A)$ $w_1(B)$ $w_2(B)$ $w_3(B)$

S2: $r_3(C)$ $r_2(A)$ $r_1(C)$ $r_2(B)$ $w_2(A)$ $w_1(A)$ $r_3(A)$ $w_2(B)$ $w_1(B)$ $w_3(B)$

S_1

T_1	T_2	T_3
	$r_2(A)$	
	$r_2(B)$	
	$w_2(A)$	
$r_1(C)$		
$w_1(A)$ (final write)		
		$r_3(C)$
		$r_3(A)$
$w_1(B)$		
	$w_2(B)$	
		$w_3(B)$

S_2

T_1	T_2	T_3
		$r_3(C)$
	$r_2(A)$	
$r_1(C)$		
	$r_2(B)$	
	$w_2(A)$	
final write $w_1(A)$		
		$r_3(A)$
~~$w_1(B)$~~	$w_2(B)$	
$w_1(B)$		
		$w_3(B)$

	S1	S2
A:	T_1	T1
B:	T_3	T_3

View Equal

S_1 S_2 Conflict Equal
Every conflict pair
(R–W conflict pair
W–R "
W–W ")
of S_1 S_2 must be same precedence

$\Rightarrow$ implied

S_1 S_2 View Equal
Every
(Initial Read
Updated read
final write)
of S_1 S_2 must be same

NOTE:
if S_1 S_2 conflict Equal
Then $S1$ $S2$ also View Equal
if $S1$ $S2$ not conflict Equal
Then $S1$ $S2$ may/may not View equal.

View Serializable Schedule Testing:

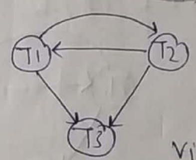
[Schedule (S)] $\longleftrightarrow$ $\exists$ (View Equal Serial Sched (S'))

S: View Serializable

Q S:

T_1	T_2	T_3
	$r_2(A)$	
$W_1(A)$		
	$W_2(A)$	
		$W_3(A)$

SG:



not conflict serializable.

Given (S) — View Equal — S': Serial

FW: A: T_1 T_2 T_3
final write

$\Rightarrow$ $(T_1, T_2) \rightarrow T_3$
they can execute in any order

IR:

data item	Initial Read	Write
A	T_2	T_1 T_2 T_3

$\Rightarrow$ $T_2 \rightarrow (T_1, T_3)$
they can execute in any order

T_2 : T_1 : T_3 Serial Schedule View equal to Sched (S).

S is View Serializable Sched.

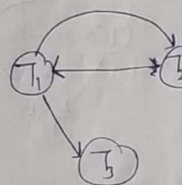
Q

S: $r_2(A)\ r_2(B)\ w_2(A)\ r_1(C)\ w_1(A)\ r_3(C)\ r_3(A)\ w_1(B)\ w_2(B)\ w_3(B)$

T_1	T_2	T_3
	$r_2(A)$	
	$r_2(B)$	
	$w_2(A)$	
$r_1(C)$		
$w_1(A)$		
		$r_3(C)$
		$r_3(A)$
$w_1(B)$		
	$w_2(B)$	
		$w_3(B)$

FW

FW



not Conflict ~~Equal~~ Serializable

	Given Sched (S)		S' : Serial
FW :	A : T_2 T_1	$\Rightarrow$	$T_2 \rightarrow T_1$
	B : T_1 T_2 T_3	$\Rightarrow$	$(T_1, T_2) \rightarrow T_3$
assume	$\boxed{C : T_1\ T_3}$	$\Rightarrow$	$T_1 \rightarrow T_2$

$T_2\ T_1\ T_3$

then not view serializable

Write $T_2 : T_1 : T_3$ in Serial Order

remaining 5 ~~Serial~~ ~~of~~ serial are eliminated

(or)

Continue Analysis

S'

T_1	T_2	T_3
	$r_2(A)$	
	$r_2(B)$	
	$w_2(A)$	
	$w_2(B)$	
$r_1(C)$		
$w_1(A)$		
$w_1(B)$		
		$r_3(C)$
		$r_3(A)$
		$w_3(B)$

S

T_1	T_2	T_3
	$r_2(A)$	
	$r_2(B)$	
	$w_2(A)$	
$r_1(C)$		
$w_1(A)$		
		$r_3(C)$
		$r_3(A)$
$w_1(B)$		
	$w_2(B)$	
		$w_3(B)$

S (given) $\xleftrightarrow{\text{view equal}}$ S' : serial

IR : data item	IR	W	S' : serial
A	T_2	$T_2 T_1$	$T_2 \rightarrow T_1$
B	T_2	$T_1 T_2 T_3$	$T_2 \rightarrow (T_1, T_3)$
C	$T_1 T_3$	—	—

no need to check b/c no write on C

UR : $W_1(A) \longrightarrow R_3(A)$ && Other Trans. Writes data item A : T_2 | $T_1 \dashrightarrow 3$ $[T_2]$ ✗

$T_2 : T_1 : T_3$ satisfy all IR, FW, & UR simultaneously.

Hence $T_2\ T_1\ T_3$ View Equal to given Schedule S. Hence S is View Equal Serializable

Q S': $r_1(A)\ r_2(A)\ w_1(C)\ r_3(A)\ r_4(A)\ r_3(C)$
$w_1(B)\ w_2(B)\ w_3(B)\ w_4(B)\ w_4(C)$

How many Serial Schedule View Equal to Schedule (S)?

Sol:

	given (S)	S': Serial
FW:	A:	
	B: $T_1\ T_2\ T_3\ T_4$	$(T_1\ T_2\ T_3) \longrightarrow T_4$
	C: $T_1\ T_4$	$(T_1) \longrightarrow T_4$ _ _ _ _ T_4

IR:

data item	IR	W
A	$T_1 T_2 T_3 T_4$	—
B	~~T_4~~ —	$T_1 T_2 T_3 T_4$
C	—	T_1, T_4

$T_1 \longrightarrow (T_2\ T_3\ T_4)$

~~$T_1 \longrightarrow T_4$~~

UR: $W_1(C) \longrightarrow r_3(C)$

Other Trans writing "C" T_4

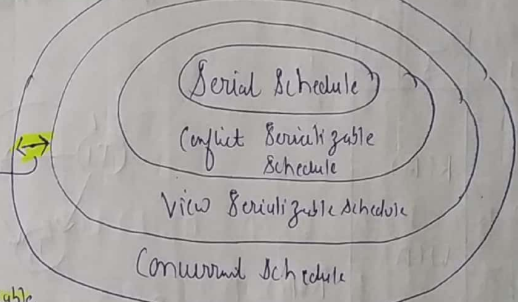
$T_1 \xrightarrow{(T_4) * \times} T_3$

T_4 not allowed in b/w T_1 & T_3

_ _ _ T_4

$(T_1 T_3)$ $^3C_2 * {}^1C_1 = 3$

$T_1\ T_3\ T_2\ T_4$
$T_2\ T_1\ T_3\ T_4$
$T_1\ T_2\ T_3\ T_4$
} Serial Sched View equal S



Non Serializable Schedule ≡ Not VSS

NOTE:-
• Conflict Serializable Schedule testing condition is "Only sufficient but not necessary" (implied) Condition for Serializability testing

$p \rightarrow q$
if p then q
sufficient but not necessary

• View Serializable Schedule testing condition is "both sufficient and necessary" Condition for Serializability testing.

if and only if

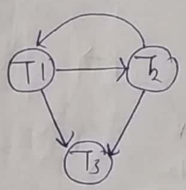
$p \longleftrightarrow q$
p if and only if

Sufficient && necessary

⇒ If Schedule (S) is not conflict serializable Schedule (and) No blind Writes in Schedule (S)

Then S is not View Serializable Schedule.

T_1	T_2	T_3
R(A)		
	R(A)	
W(A)		
	W(A)	
		R(A)
		W(B)



S is not CSS.
&& no blind Write

Hence S is not View Serializable Schedule.

If not CSS && some blind Writes exists in S
Then S may / may not VSS.

Q 23 (WB) P-61

Sol: T_1: $r_1(A)\ r_1(B)\ w_1(B)$

T_2: $r_2(A)\ r_2(B)\ w_2(B)$

Concurrent Sched. $= \frac{6!}{3!\,3!} = 20$

Find HOW many are Serializable

Serializable

equal to $T_1 T_2$ Serial

S'

T_1	T_2
$r_1(A)$	
$r_1(B)$	
$w_1(B)$	
	$r_2(A)$
	$r_2(B)$
	$w_2(B)$

equal

S:

$r_1(A)\ r_1(B)\ w_1(B)\ r_2(B)\ w_2(B)$

$r_2(A)$ not conflict with any pair hence it can be placed at 4 possible places

4

equal to $T_2 T_1$ Serial

T_2	T_1
$r_2(A)$	
$r_2(B)$	
$w_2(B)$	
	$r_1(A)$
	$r_1(B)$
	$w_1(B)$

$r_2(A)\ r_2(B)\ w_2$: $r_1(B)\ w_1(B)$

$r_1(A)$

4

4 serializable Sched equal to $T_2 : T_1$ Sched.

4 Serializable Sched equal to $T_1 : T_2$ Sched.

Hence $20 - 8 = 12$ are non-serializable

Scanned by CamScanner

20

- 8 Serializable
  - 2 Serial Sched
  - 6 Non Serial Sched which are Serializable Schedule
- 12 Non-Serializable

Q.22

Sol: $T_1 : r_1(A)\ w_1(A)\ r_1(B)\ w_1(B)$

$T_2 : w_2(B)\ w_2(A)$

No Non-Serializable Sched b/w T_1 & T_2 can be Serializable

Sched equal to $T_1 : T_2$ Serial

S: ..... $w_1(B) : w_2(B)$ ....

Sched equal to $T_2 : T_1$ Serial

S: .... $w_2(A)\ r_1(A)$ ...

${}^6C_4 = 15$: Concurrent

- 2 Serializable Schedule
  - 2 Serial Sched
  - 0 Non-Serial Sched.
- 13 Non Serializable Schedule

Classification of Schedule based on Recovery ~~Rec~~ Recoverability :-

Concurrent execution of transactions may leads

1. Irrecoverable Problem
2. Cascading Rollback problem
3. lost update problem.

Can Occurs even Schedule is Serializable.

Irrecoverable Schedule :-

Schedule S irrecoverable iff some trans T_j reads data item X which is updated by T_i and commit of T_j is before C/R of T_i

C → commit, R → rollback

T_i	T_j
⋮	⋮
W(X)	
	R(X)
	⋮
	Commit
[Commit] rollback	

Irrecoverable Schedule.

T_1 : Withdraw 3000 from "A"

T_2 : check balance of "A"

A : 10,000

T_1	T_2
R(A)	
A=A-3000	
W(A)	
	R(A)
	Commit

System Crash — Roll back

Rollback of T_2 not possible because T_2 already committed [irrecoverable]

Dirty Read [Uncommitted Read]

T_i	T_j
W(X)	
	R(X)
C|R	

T_j reads "X" which is updated by uncommitted trans T_i

Recoverable Schedule :-

Sched (S) is recoverable iff

1) No dirty reads in schedule (S)

(OR)

2) If T_j reads X which is updated by T_i Then commit of T_j must delay until commit (or) rollback of T_i.

T_i	T_j
W(X)	
	R(X) ← dirty read
C|R	
	Commit

Recoverable Sched

Cascading Rollback Problem

because of failure of some transaction forced to rollback some set of other transactions.

T_1	T_2	T_3	T_4	T_5	T_6
W(A)					
	R(A)				
	W(A)				
	W(B)				
		W(A)			
			R(B)		
				R(B)	
				W(B)	
					R(B)

System crash

[T_2 T_4 T_5 T_6] forced to Rollback because of T_1 rollback

Disadvantage :-

Wastage of CPU Execution Time

I/O access cost && resource utilization

Cascadeless Rollback Recoverable Schedule :-

[Dirty reads not allowed]

T_i	T_j
W(X)	
C/R	
	R(X)

If T_i writes X W(X) then R(X) of T_j must delay until C/R of T_i

commit

rollback

Lost Update Problem

T_i	T_j
W(X)	
	W(X)
C/R	

if T_i writes X && T_j also writes b4 T_i C/R then lost update problem can occur.

EX: T_1 : Set A : 1000
T_2 : Set B : 2000

T_1	T_2
W(A)	
	W(A)
	C2

failed.

[Incorrect result]

A = ∅
1000 <- by T_1
2000 <- by T_2

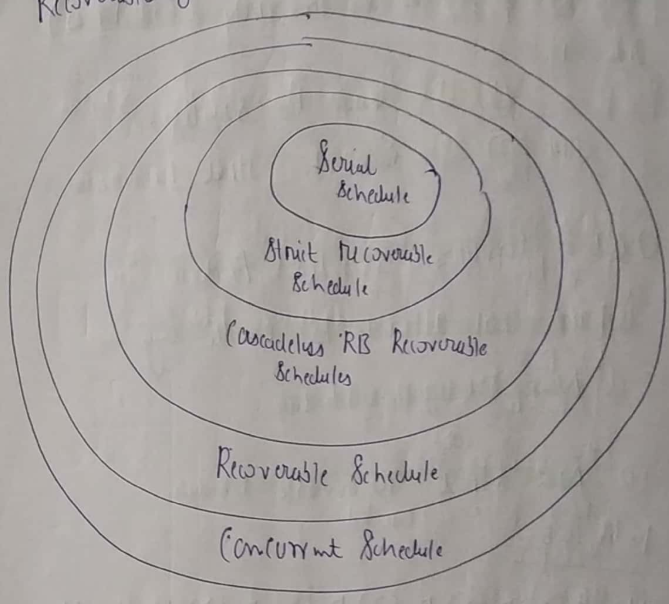
Strict Recoverable Schedule :-

[Cascadeless RB recoverable] and [Avoid Lost Update Problem]

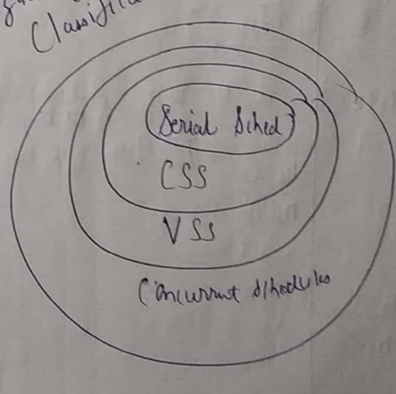
T_i	T_j
W(X)	
⋮	
C|R	
	R(X)/W(X)

If Ti writes X
Then R(X)/W(X) of
Tj must delay untill Ti
commit | rollback

Recoverability Classification



Serializability Classification



Schedule (S) Preserves integrity (correctness) only if

① S must be serializable

(AND) ② S must be strict recoverable

Goal of Concurrency Control Protocol :-

Should not allowed to execute any

① Non-serializable schedule

(OR)

② Non-strict recoverable schedule.

T2 P-61 (WB)

Q S1: $r_1(X)\ r_2(Z)\ r_1(Z)\ r_3(X)\ r_3(Y)\ w_1(X)\ C_1$
$w_3(y)\ C_3\ r_2(y)\ w_2(z)\ w_2(y)\ C_2$

T_1	T_2	T_3
$r_1(x)$		
	$r_2(z)$	
$r_1(z)$		
		$r_3(x)$
		$r_3(y)$
$w_1(x)$		
C_1		
		$w_3(y)$
		C_3
	$r_2(y)$	
	$w_2(z)$	
	$w_2(y)$	
	C_2	

Strict Schedule

S_2

T_1	T_2	T_3
$r_1(x)$		
	$r_2(z)$	
$r_1(z)$		
		$r_3(x)$
		$r_3(y)$
$w_1(x)$		
		$w_3(y)$
	$r_2(y)$	
	$w_2(z)$	
	$w_2(y)$	
C_1		
	C_2	
		C_3

Strict ×
Cascadeless ×
recoverable ×
Non-recoverable

S_3:

T_1	T_2	T_3
$r_1(x)$		
	$r_2(z)$	
		$r_3(x)$
$r_1(z)$		
	$r_2(y)$	$r_3(y)$
$w_1(x)$		
C_1		
	$w_2(z)$	
		$w_3(y)$
	$w_2(y)$	
		C_3
	C_2	

Strict ×
Cascadeless ✓

Locking Protocol

Lock : Variable used to identify status of data item.

Trans (T)

lock (A) <~ grants by concurrency control protocol.
R (A)
W (A)
lock (B) <— denied
{ waiting state
lock (B) <— granted
W (B)
unlock (A)
unlock (B)

Single Mode of lock :-

=> Not possible to differentiate ready only data & Read/Write operation

T_1
{ lock(A)
  R(A)   <— grants
{ lock(B)
  W(B)   <— ~~grant~~ grants

=> less degree of concurrency

T_1	T_2
lock(A)	
R(A)	
	lock(A) <- denied
	R(A)

Consistent Schedule but not allowed to execute.

Multi Mode locks :-

1) Shared lock [S] :-
Read only lock

2) Exclusive lock [X] :-
Read/write lock
S (A) <— grant
R (A)
X (B) <— grants
{ R(B)
  W(B)

T_1	T_2
S(A)	
	S(A) <- granted
S(B)	
	X(B) <- denied
	[T_2 must wait untill T_1 unlocks "B"].

Hold by T_i

data item A	S	X
requesting by T_j S	Yes	No
requesting by T_j X	No	No

Scanned by CamScanner

Two Phase locking Protocol [2PL]

~~def~~: [Guaranteed Serializability]

def: lock requests of Trans (T) in any data item in any mode allowed untill first unlock of Trans (T).

Trans (T)	
S(A)	Growing Phase [locking phase]
X(B)	
S(C)	
X(D)	* — lock point of Trans (T)
U(B)	* Shrinking phase [unlocking phase]
U(A)	
U(D)	
U(C)	

first unlock of T → U(B)

T_1: Trans 500 from A to B	T_2: disp A+B bal	T_3: Set A, B, C values as 1000
T_1	T_2	X(A)
X(A)	S(A)	W(A)
R(A)	R(A)	X(B)
W(A)	S(B)	W(B)
U(A)	~~R(B)~~	X(C)
X(B)	U(A)	U(A)
R(B)	R(B)	U(B)
W(B)	U(B)	W(C)
U(B)	Commit	U(C)
Commit		Commit
2PL	2PL	2PL

not 2PL

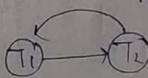
Concurrent execution of T_1 T_2 T_3

guaranteed serializability

<u>EX</u>: S_3

T_1	T_2
$r_1(A)$	
$w_1(A)$	
	$r_2(A)$
	$r_2(B)$
$r_1(B)$	
$w_1(B)$	

Non-serializable



Not CSS + no blind write

By 2PL

S:

T1	T2
X(A)	
r_1(A)	
W(A)	
X(B)	
U(A)	
	S(A)
	r(A)
	S(B)
	r(B) ← denied.
	must delay until T_1 unlocks B
r_1(B)	
W_1(B)	
U(B)	

if Schedule (S) not Conflict Serializable Schedule
Then Schedule (S) not allowed to execute by 2PL.

if Schedule (S) allowed to execute by 2PL
Then Schedule (S) is Conflict Serializable Schedule.

Every Schedule allowed by 2PL is also Conflict Serializable but,
Not every Conflict Serializable Schedule is allowed by 2PL.

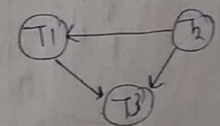
2PL Schedules ⊂ CSS

Q $S: W_2(A)\ W_1(A)\ W_3(A)\ W_2(B)\ W_1(B)\ W_3(B)$
Which is true?

a) S is CSS && in 2PL
b) S is CSS but not in 2PL ✓
c) S not CSS && not in 2PL
d) S not CSS && allowed in 2PL

Ex:

T_1	T_2	T_3
	W(A)	
W(A)		
		W(A)
	W(B)	
W(B)		
		W(B)



it is CSS.

T_1	T_2	T_3
	X(A)	
	W(A)	
	X(B)	
	U(A)	
X(A)		
W(A)		
X(B)		
U(A)	U(B)	
		X(A) ← denied
		W(A)
	W(B)	
W(B)		
		W(B)

not possible

b/c of $W_2(B)$ after $W_3(A)$

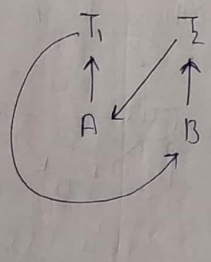
Hence S is not in 2PL.

Limitations of 2PL

→ 2PL restriction may lead to deadlock.

T_1 : $r_1(A)$ $r_1(B)$ [S(A) S(B) U(A) U(B)]   denied

T_2 : $w_2(B)$ $w_2(A)$ [X(B) X(A) U(B) U(A)]   denied



dead lock

T_1	T_2
S(A)	
$r_1(A)$	
	X(B)
	$W_2(B)$
denied → S(B)	
$r_1(B)$	
	X(A) ← denied
	$W_2(A)$

T_1 waiting for T_2

Q $O_1, O_2 \ldots O_n$ are data items

Assume each trans should lock data item in sequence of $O_1, O_2 \ldots O_n$ irrespective of read/write data items order

Protocol :

1. lock all data items
2. perform R/W's of T
3. unlock all data items of T.

True ?

a) free from deadlock ✓
b) suffer from deadlock
c) May execute non-serializable schedule
d) Free from starvation

Sol :-

T_1	T_2
$X(O_1)$	
$X(O_2)$	
$W(O_1)$	
$U(O_1)$	
	$X(O_1)$
	$X(O_2)$ — denied
	$W(O_2)$
$W(O_2)$	
$U(O_2)$	
	$W(O_1)$

2) 2PL restriction may lead to starvation

	T_1	T_2	T_3	T_4 ...
		S(A)		
denied wait for T_2 →	X(A)		S(A)	
		U(A)		
denied wait for T_3 →	X(A)			S(A)
			U(A)	
denied wait for T_4 →	X(A)			
	⋮			

T_1 : Starvation

3) Basic 2PL restrictions not sufficient to avoid IRRECOVERABLE
Cascading RB
& Lost update Problem.

T_1	T_2
X(A)	
W(A)	
X(B)	
U(A)	
	X(A)
	R(A)
W(B)	
U(B)	
	X(B)
	W(B)
	U(B)
	C2
C1	

Strict 2PL Protocol :-

① Basic 2PL :-
Lock requests of T not allowed in unlock phase of T.

(and)

② Strict Recoverable :- All exclusive locks of Trans (T) must hold until C/R of Trans (T)

T_1	T_2
X(A)	
W(A)	
⋮	
C/R	
U(A)	R(A)/W(A)

Trans (T)

S(A)
R(A)
X(B)
W(B)
S(C)
R(C)
X(D)
W(D) } locking phase

U(A)
U(C)
Commit
U(B)
U(D) } unlocking phase

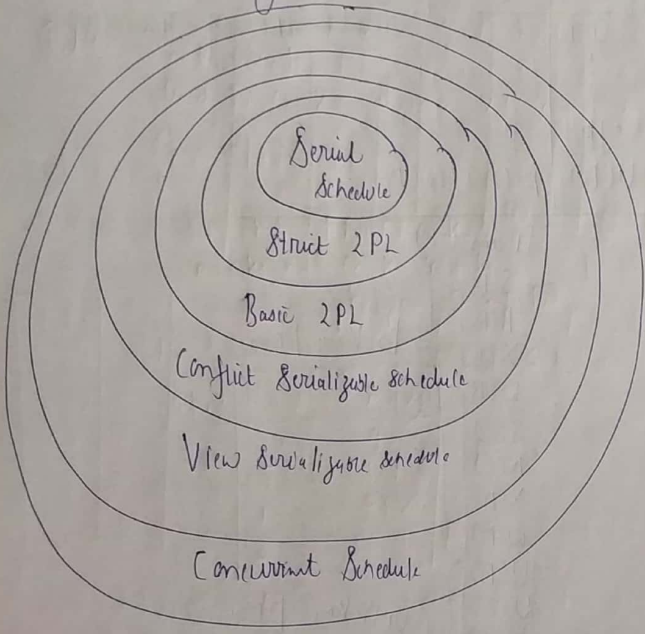
Strict 2PL

↳ guaranteed serializability

↳ guaranteed strict recoverability

↳ not free from deadlock && starvation

NOTE:- Strict 2PL correct protocol for concurrency control implementation



Topic	Imp [Notes + Extra Prob]	Less Imp [Notes]
Normalization	• Find CK's • LLJ/DP testing • 1NF to BCNF dej / comparison • Highest NF of given R	- Canonical Cover • Decomposition into higher NF (2NF, 3NF, BCNF)
Queries	RA / SQL	TRC
Index	B / B+ Tree [Dynamic MLI]	Static & MLI (PI / CI / SI) with MLI
Transaction & Concurrency Control	ACID properties *** CSS / VSS testing Recoverability Rec / CLR / Strict Rec 2PL / strict 2PL	
ERD && IC	CK / PK / AK / SK FK *** ER : RDBMS design for given ERD	

Q12 Chp-3 (SQL | RA | TRC)

Enroll (Sid cid) X paid (Sid amount)

plan 1 : $\pi_{cid} \left(\text{Enroll} \bowtie \sigma_{amount > X} (\text{paid}) \right)$

plan 2 : $\pi_{cid} \left(\sigma_{amount > X} (\text{Enroll} \bowtie \text{paid}) \right)$

Imp (Q3.17 P-55 ○ (GB))
(Q19)

Sol :-

Cid	Rank
C1	1
C2	1
C3	3
C4	3
C5	5
C6	6

A

Cid	bal
C1	80
C2	80
C3	70
C4	70
C5	60
C6	40

B

Cid	bal
C1	80
C2	80
C3	70
C4	70
C5	60
C6	40

C1	2
C2	2
C3	4
C4	4
C5	5
C6	6

SQL Query 1) Wrong

FROM ACC, A , ACC, B
Where A.bal ≤ B.bal
⇓
Group by A.Cid
⇓
Select A.cid , Count (B.cid)

Ans
Hence both Queries are wrong , And both the queries will produce same o/p if every customer has distinct balance.

Cid

2 { C3 70 C1 80
C3 70 C2 80

2 { C4 70 C1

4 { C5 60 C1
C2
C3
C4

5 { C6 40 C1
C2
C3
C4
C5

SQL Query 2) Wrong

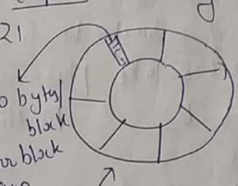
FROM A, B
Where $\boxed{\text{A.bal < B.bal}}$
⇓ no row selected for C1 & C2 hence
Group by A.Cid rank 1 will not be printed.
⇓
Select A.cid , Count (B.cid) +1

C3	3
C4	3
C5	5
C6	6

for C1 & C2 it is
NULL + 1 } → NULL hence
NULL + 1
No rank is assigned

Chp-5 Indexing

Q 21



200 bytes record size

Bf = 5

Sector [block] = $\frac{4000}{32}$ = 1250 bytes

1000 bytes can use to store records

Q26 (Work Book) P-61 T & CC

T_1 : $r_1(A)$ $w_1(A)$ $r_1(B)$ $w_1(B)$

T_2 : $r_2(A)$ $w_2(A)$ $r_2(B)$ $w_2(B)$

T_1	T_2
$r_1(A)$	
$w_1(A)$	
$r_1(B)$	
$w_1(B)$	
	$r_2(A)$
	$w_2(A)$
	$r_2(B)$
	$w_2(B)$

T_2	T_1
$r_2(A)$	
$w_2(A)$	
$r_2(B)$	
$w_2(B)$	
	$r_1(A)$
	$w_1(A)$
	$r_1(B)$
	$w_1(B)$

Ans 6

similar to $T_1 T_2$

I case

possible places $r_2(A)$ $w_2(A)$ can take preserving precedence

$r_1(A)$ $w_1(A)$ | $r_1(B)$ | $w_1(B)$ | : $r_2(B)$ $w_2(B)$

$r_2(A) w_2(A)$ } $w_2(A)$ $w_2(A)$ 3

$r_2(A) w_2(A)$ $w_2(A)$ 2

$r_2(A) w_2(A)$ 1

} 6

$^3C_1 + {}^3C_2$

$= 3 + 3 = 6$ either both can take same space (or) different spaces

6 Schedule Serializable as $T_1 T_2$

Q 19 P-60 (WB) Gate-2017

Sol. T_1 : $r_1(X)$ $w_1(X)$ $r_1(Y)$ $w_1(Y)$

T_2 : $r_2(Y)$ $w_2(Y)$ $r_2(Z)$ $w_2(Z)$

total (conflict) serializable schedule to $T_1 \rightarrow T_2$

T_1	T_2
$r_1(X)$	
$w_1(X)$	
$r_1(Y)$	
$w_1(Y)$	
	$r_2(Y)$
	$w_2(Y)$
	$r_2(Z)$
	$w_2(Z)$

$r_1(X)$ $w_1(X)$ $r_1(Y)$ $w_1(Y)$: $r_2(Y)$ $w_2(Y)$ $r_2(Z)$ $w_2(Z)$

$T_1 T_2$ is only one way, no other schedule possible b/z last opn of T_1 and 1st opn of T_2 conflict pairs form

Now, how many schedule (conflict) serializable to $T_2 \rightarrow T_1$

Here we fix T_1 & find valid positions of $w_2(Y)$

T_2	T_1
$r_2(Y)$	
$w_2(Y)$	
$r_2(Z)$	
$w_2(Z)$	
	$r_1(X)$
	$w_1(X)$
	$r_1(Y)$
	$w_1(Y)$

Here cases are possible while preserving the precedence of conflicting pairs.

I) $w_2(Y)$

$r_1(X)$
$w_1(X)$
$r_1(Y)$
$w_1(Y)$

II) $r_1(X)$
$w_2(Y)$
$w_1(X)$
$r_1(Y)$
$w_1(Y)$

III) $r_1(X)$
$w_1(X)$
$w_2(Y)$
$r_1(Y)$
$w_1(Y)$

Concept:- Here we fix T_1 & ~~first~~ find valid place for $r_2(Z)$, $w_2(Z)$ for corresponding valid place of $w_2(Y)$

I case

$\xrightarrow{\quad}$ $r_2(Y)$
$W_2(Y)$
$\xrightarrow{\quad}$
$r_1(X)$
$\xrightarrow{\quad}$
$W_1(X)$
$\xrightarrow{\quad}$
$r_1(Y)$
$\xrightarrow{\quad}$
$W_1(Y)$
$\xrightarrow{\quad}$

Possible places $r_2(Z)$ ~~r_2~~ $W_2(Z)$ can take

$\underbrace{{}^5C_1 + {}^5C_2}$ $= 5 + 10 = 15$ — either both can take same space (or) different space

for each of these 15 positions, $r_2(Y)$ can take take only place bcz it has to come b4 $W_2(Y)$

$\therefore$ total $15 \times 1 = 15$ Schedule possible

II case

$\xrightarrow{\quad}$ $\}$ $r_2(Y)$
$r_1(X)$
$W_2(Y)$
$\xrightarrow{\quad}$
$W_1(X)$
$\xrightarrow{\quad}$
$r_1(Y)$
$\xrightarrow{\quad}$
$W_1(Y)$
$\xrightarrow{\quad}$

possible places $r_2(Z)$ $W_2(Z)$ can take

$\Rightarrow {}^4C_1 + {}^4C_2 = 4 + 6 \Rightarrow 10$

for each of these 10 positions, ~~r_2 can take 2~~ $r_2(Y)$ can take 2 positions

$\therefore$ total schedule $= 2 \times 10 = 20$ Schedule possible

III >

$\xrightarrow{\quad}$
$r_1(X)$ $\}$ r_2
$W_1(X)$
$\xrightarrow{\quad}$
~~r_2~~ $W_2(Y)$
$\xrightarrow{\quad}$
$r_2(Y)$
$\xrightarrow{\quad}$
$W_2(Y)$
$\xrightarrow{\quad}$

possible positions $r_2(Z)$ $W_2(Z)$ can take

${}^3C_1 + {}^3C_2 = 3 + 3 = 6$

for each of these 6 positions, r_2 can take 3 positions

$\therefore$ total schedule $= 3 \times 6 = 18$

Total Schedule (conflict) serializable to $T_2 \to T_1$

$15 + 20 + 18 = 53$.

$\therefore$ ~~An~~ Total (conflict) serializable schedule.

$(T_2 \to T_1) + (T_1 \to T_2)$

$= 53 + 1$

Ans $= 54$

Q18 P-60 (WB) (Gate 2017)

Sol. Concept : ~~Wait Wound~~

i) Wait Die

ii) Wound - Wait

Wound wait

If $TS(P_1) < TS(P_2)$

ie (P_1 Older process than P_2)

~~When P_1 requests for resource ~~R~~, currently held by P_2 (which is younger process) then P_1 is allowed to wait.~~
~~Otherwise if P_1 is younger process and P_2 is older process than P_1 is rolled back~~

When P_1 requests for resource, currently held by P_2 (which is younger process) then P_1 (older) Wounds/ kills P_2 and restart it later with same time stamp

Otherwise if P_1 is younger process and P_2 is older than P_1 is allowed to wait.

> If $TS(P_1) < TS(P_2)$
>
> P_2 is killed
>
> else
>
> P_1 waits

Wound wait is preemptive.

It is deadlock && starvation free.

Wait -die

If $TS(P_1) < TS(P_2)$

(ie P_1 is Older process than P_2)

When P_1 requests for resource currently held by process P_2, the P_1 allowed to wait.

If P_1 is younger and P_2 is older than P_1 & aborts (die) and restart later with same time stamp.

Wait die non- preemptive

Wait die is not deadlock free and starvation free

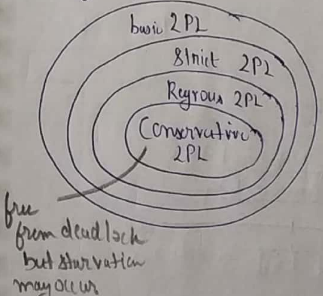
DBMS Revesion after Aptitude

TSO(Time Stamp (Ordering) Protocol

→ guarantee conflict serializability → disadv
It leads to starvation

→ free from deadlock

Graph based protocol
-free from deadlock



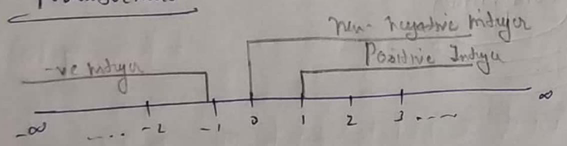
29/11/18

Reasoning & Aptitude

Number System

- Number line
- factorials
- Irrational | Rational
- Divisibility Rule
- Cyclicity
- Factors
- Remainder
- Base Method
- LCM | HCF
- Miscellaneous

Number line



Any no. that can be represented on an imaginary line $-\infty$ to $+\infty$ is known as real no.

Natural No $(1,\infty)$

Set of all +ve integers.

2, 7,

2.1 ✗

Whole no $(0,\infty)$

Set of all non-negative integers $(0,\infty)$

0, 1, ...

2.1 ✗

Even No

Any natural no divisible by 2

2, 4, 6, ...

Odd no

Any natural no not divisible by 2

1, 3, 5, ...

→ 0 is neither even nor odd

Prime No

Any natural no having exactly 2 distinct factors/divisors ie 1 and number n itself.

→ 2 is the only even prime no.

→ $(3,\infty)$ all prime no are odd

Q $p \to$ prime no & $p > 10{,}000$

$\frac{p^{3458}}{6} \to$ Remainder ?

Rule :- Except ② and ③ all other prime no's belongs to the family of $6k+1$ or $6k-1$ (Where $k \in 1,2,3,\ldots$)

NOTE: Note this is necessary condition but not sufficient Ex:- $25 = 6\times4+1 \to$ not prime
$23 = 6\times4-1 \to$ prime

5	7	11	13	17	19
$6\times1-1$	$6\times1+1$	$6\times2-1$	$6\times2-1$	$6\times3-1$	$6\times3+1$

23	29
$6\times4-1$	$6\times4+1$

$\frac{p^{3458}}{6}$ $\to \frac{(6k+1)^{3458}}{6} \to ①$ Ans - ①

$\to \frac{(6k-1)^{3458}}{6} \to ①$

Composite no

Any natural no. having at least 3 factors

$\to 1$ is neither prime nor composite

4 6 8 9 . . . . .

4	6	8
1	1	1
2	2	2
4	3	4
	6	8

Factorials

$n!$ or $\underline{|n}$

$n! = n(n-1)(n-2)\ldots\ldots 3.2.1$

$0! = 1$

No of Zero's in the End

Q How many 0's will come in the end of $5\times10\times\ldots$

$5\times10\times15\times20\times25\times30\times35\times40\times45$

Sol: 0 is generated by pairs of (2,5) bcz $2\times5=10$

$(5)^{10}(2)^{7} = (5,2)^{7} = 7$ 7 pairs

$(5)^{10}(3)^{4}(7)^{1}(2)^{7} = (5,2)^{7} - 7$

Q $12! = \underbrace{12\times11\times10\times9\times8\times7\times6\times5\times4\times3\times2\times1}_{6+3+1=10}$

power of 2 $\frac{12}{2}=6$

$\frac{6}{2}=3$

$\frac{3}{2}=1$

Scanned by CamScanner

Q $31!$ find $\cancel{2^a \times 3^b \times 7^c \times}$

$31! = 2^a \times 3^b \times 5^c \times 7^d \times \ldots 31^1$

$2^a = \frac{31}{2} = \frac{15}{2} = \frac{7}{2} = \frac{3}{2} = 1 = 26$

$3^b = \frac{31}{3} = \frac{10}{3} = \frac{3}{3} = 1 \Rightarrow 14$

$5^c = \frac{31}{5} = \frac{6}{5} = 1 \Rightarrow 7$

$7^d = \frac{31}{7} = \frac{4}{7} \Rightarrow 4$

Q $100! = 2^a \times 3^b \times 5^c \times 7^d \times \ldots 97^1$

$\frac{100}{2} = \frac{50}{2} = \frac{25}{2} = \frac{12}{2} = \frac{6}{2} = \frac{3}{2} = 1 \Rightarrow 97$

$\frac{100}{3} = \frac{33}{3} = \frac{11}{3} = \frac{3}{3} = 1 \Rightarrow 48$

$\frac{100}{5} = \frac{20}{5} = \frac{4}{5} \Rightarrow 24$

$\frac{100}{7} = \frac{14}{7} = \frac{2}{7} \Rightarrow 16$

Q $1024! \rightarrow$ How many zeros

$\frac{1024}{5} = \frac{204}{5} = \frac{40}{5} = \frac{8}{5} = 1$ (253)

Q $80! = 3^b$

$\frac{80}{3} = \frac{26}{3} = \frac{8}{3} = 2 \Rightarrow 36$

Q How many 0's will come in the end of

$(40!)^{50!}$

a) $8^{50!}$ b) $8 \times 50!$ c) $9^{50!}$ d) $9 \times 50!$

Sol: $\frac{40}{5} = \frac{8}{5} = 1 \Rightarrow 9$ $(5^9)^{50!} = (5)^{9 \times 50!}$

# of 0's $= 9 \times 50!$

Q How many 0's will come in the end of

$(6!)^{6!} \times (8!)^{8!} \times (10!)^{10!}$

a) $6! + 8! + 10!$ b) $6! \times 8! \times 10!$

c) $6! + 8! + 2(10!)$ d) $6! \times 8! \times 2(10!)$

Sol: $(6!)^{6!} = (5^1)^{6!}$

$(8!)^{8!} = \left(\frac{8}{5}\right) = \frac{1}{5} = (5^1)^{8!}$

$(10!)^{10!} = \frac{10}{5} = \frac{2}{5} = (5^2)^{10!}$ $(5)^{2 \times 10!}$

$5^{6!} + 5^{8!} + 5^{2 \times 10!}$

$= 5^{6! + 8! + 2 \times 10!}$

Q How many 0's will come in the end

$18! + 19!$

Sol :- $\frac{18}{5} = \frac{3}{5} = 3$ ~~(5^3) + (5^3)~~ =

$\frac{19!}{5} = \frac{3}{5} = 3$

$18! + 19! \Rightarrow 18! + 19 \times 18! \Rightarrow 18!(1+19)$

$\Rightarrow 18! \times 20$

$(5)^3 \times 5^1 = (5)^4$

$\therefore$ 4 zeros.

Exponents (Highest Power)

Q What is the highest power of 2 that can divide 100!

Sol :- $\frac{100!}{(2)^x} = \frac{\underbrace{2^{97} \times 3^{48} \times 5^{24} \times \ldots\ldots 97^1}_{Q}}{(2^{97})}$

$N = DQ + R$ $D = 2^{97}$

if $R = 0$ $N = D \times Q$

Ans - 97

$\frac{100!}{3^x} = \frac{3^{48} \times Q}{3^{48}}$ Ans :- 48

Q $\frac{100!}{8} = \frac{2^{97}}{(2^3)^x} = (2^3)^{32} \cdot 2 = 32$ Ans

$\frac{97}{3} = 32$

Q $\frac{100!}{9} = \frac{3^{48}}{9} = (3^2)^{24} = 24$

H.W

Q $\frac{100!}{(72)^x} = \frac{2^{97}}{2^3} \times \frac{3^{48}}{3^2}$

$= (2^3)^{32} \cdot 8 \; (3^2)^{24}$

$= 24$

Rational / Irrational :

$\frac{p}{q}$ Simplest form / Reduced form

p, q are integers

$q \neq 0$

Rational No

Irrational No

$\frac{1}{\sqrt{3}}$, $\frac{7\sqrt{2}}{5\sqrt{3}}$, ...

Surds, log value (except $\log 10^n$)

exponential e^x

Rational no
- terminating: $\frac{1}{8} = 0.125$, $\frac{1}{4} = 0.25$
- Non terminating Recurring: $\frac{1}{3} = 0.333.... 33... \infty$

$\pi = \frac{22}{7} = 3.142857\,142857.....\infty$

$\frac{1}{6} = 0.1666.....6..\infty$

Irrational (Non-terminating + Non-~~to~~ recurring)

$\pi = 3.14281561781 9....$

$\sqrt{2} = 1.4143561.....$

Q[Gate] Simplest form $\frac{1}{1357924861(7)}$ 4 2^7 $5^{?}$ 2^x5^y

a) Terminating

~~b)~~ Non-terminating & recurring

c) Non-terminating & non-recurring

d) none of these

Sol:- Condition for terminating (in rational no)

(i) $\frac{1}{2^n} = \frac{1}{4}, \frac{1}{8}$ ...

(ii) $\frac{1}{5^x} = \frac{1}{5}, \frac{1}{625}$, ...

iii) $\frac{1}{2^x.5^y} = \frac{1}{10}$ ...

Denominator must contain either 2^x or 5^x or 2^x5^y...

Conversion of Non-terminating Recurring no into $\frac{p}{q}$ form

EX $2.7777....$

$x = 2.\overline{7}$

$10x = 27.\overline{7}$

$9x = 25$

$x = \frac{25}{9}$

don't use this method

1Q $12.\overline{785}$ — no of digits under bar

$\frac{12785 - 12}{999}$

2Q $2.\overline{1738}$

$\frac{21738 - 2}{9999}$

3Q $12.\overline{78}596$ — no of digits after decimal & bar

Sol $\frac{1278596 - 1278}{99900}$

Q4 ~~$12.\overline{785}$~~

$12.34\overline{567}$

$\frac{1234567 - 12345}{99000}$

Scanned by CamScanner

Gate 2014

$x = 0.abcabcabc\ldots$

$x = 0.\overline{abc}$

$x \times \underset{\text{natural}}{y} = \underset{\text{+ve integer}}{z}$

Which among the following is possible value of y?

a) 9999 b) 99999 ~~c)~~ 1998 d) none of these

$\frac{abc}{999} \times \begin{pmatrix} \text{y must be 999} \\ \text{multiple of } \\ 999 \end{pmatrix} = z$

abc
0.123
$\frac{123}{999}$ - 9999

Divisibility Rule

$\boxed{\text{Rule of 2}}$ $\frac{N}{2} \longrightarrow$ last digit → Even, 0

$\boxed{\text{Rule of 3}}$ $\frac{N}{3} \longrightarrow \frac{\text{Sum of digits}}{3}$

Ex: $\frac{1654}{3} \Rightarrow \frac{16}{3} = 1$ (Sum of digits)

Remainder = ?

Ex: $\frac{1111\ldots}{3}$ 100 times Remainder = ?

$\Rightarrow \frac{100}{3} = 1$

$\boxed{\text{Rule of 4}}$ $\frac{N}{4} \longrightarrow$ last 2 digits of no are divisible by 4.

- 00
- $\frac{ab}{4}$ divisible by 4

$\frac{7312}{4}$?

$7312 = 7300 + 12$

7300 divisible by 4; 12 must be divisible by 4 for sum to be divisible by 4.

$\boxed{\text{Rule of 5}}$ $\frac{N}{5} \longrightarrow$ last digits → 5, 0

$\boxed{\text{Rule of 6}}$ $\frac{N}{6} \longrightarrow \frac{N}{2} \cdot \frac{N}{3}$ must be divisible by 2 & 3 both

$\boxed{\text{Rule of 8}}$ $\frac{N}{8} \longrightarrow$ last 3 digits

- 000
- $\frac{abc}{8}$

$7512 = 7000 + 512$

$= 7 \times 1000 + 8 \times 8 \times 8$

$= 7 \times 125 \times 8 + 8 \times 8 \times 8$

$\boxed{\text{Rule of 9}}$ $\frac{N}{9} \longrightarrow \frac{\text{Sum of the digits of N}}{9}$

$7254 = 7000 + 200 + 50 + 4$

$= 7 \times 1000 + 2 \times 100 + 5 \times 10 + 4$

$= 7 \times (999 + 1) + 2(99 + 1) + 5 \times (9 + 1) + 4$

$= \underbrace{7 \times 999 + 2 \times 99 + 5 \times 9}_{\text{divisible by 9}} + \underbrace{7 + 2 + 5 + 4}$

Rule of 11   ab cd

$(b+d) - (a+c) = 0$ (or) $= 11k$

$7524 = 7\times1000 + 5\times100 + 2\times10 + 4$

$= 7\times(1001-1) + 5(99+1) + 2\times(11-1) + 4$

$= \underbrace{7\times1001 + 5\times99 + 2\times11}\ \underbrace{+4 - 7 + 5 - 2}_{0 \text{ or } 11k}$

$\boxed{1001 = 7\times11\times13 \text{ — always divisible by } 11}$

Q $N = 123456789$

is divisible by — Remainder ?

a)	3 ✓	0
b)	4	1
c)	8	5
d)	9 ✓	0
e)	11	5

Q $N = 1568X354$ — is divisible by 88

What is X & Y ?

Sol. $11 \to (16+4) - (X+12) = 0$ or $11k$

$8 \to \frac{354}{8} = 4\overline{4}2$ ... $Y=2$ must be 2

~~354 = 8 × 2 + 4~~

$18 - X - 12 = 11k$ or 0

$6 - X = 11k$ or 0

$X = 6$

Cyclicity (To get the last digit)

$(N)^P = (\_\_\_\_\_ x)^P$

always divide power $P/4$

Remainder	raise remainder over x	Solve
1	x^1	↓
2	x^2	last
3	x^3	digit
0	x^4	

Ex. $(7732)^{5127}$ $\qquad 27/4 = \text{rem} = 3$

$x = 2 \qquad 2^3 = 8$

1) $(5163)^{6690}$ $\qquad \frac{90}{4} = 2 \leftarrow \text{rem}$

$(3)^2 = 9$

2) $(2167)^{1144}$ $\qquad x = (7) \qquad (7)^4 = 343 \times 7 = 2401$

$\frac{44}{4}$ rem $= 0$ $\qquad \frac{44}{4} = 0$

3) $(5162)^{4488}$ $\qquad (2)^4 = 16$

4) $(213)^{111111} = (3)^3 = 512 = 2$

Q 118 (WB) P-85
Q 126 (WB) P-86

Sol: 118 $\quad 211^{870} + 146^{127} \times 3^{424} = ?$

$(1)^2 = 1 + (6)^3 = 6 \times (3)^4 (1)$

$1 + 6 * 1 = 7$

Q 126 $\quad (2171)^7 + (2172)^9 + (2173)^{11} + (2174)^{13}$

Sol $\quad 1 + 2 + 7 + 4 = 14$

Remainder Theory

$D \overline{)N(} Q$

$\overline{\quad R \quad}$

$N = DQ + R$

General rule

$$\frac{4739 \times \cancel{10^{40}}}{5 \times \cancel{10^{40}}} = 4 \times 10^{40}$$

generally cancellation is banned

When cancellation is done multiply the cancelled no with the result

$\frac{\cancel{10}}{\cancel{6}} \frac{5}{3} = 2 \times 2 = 4$

Here we cancelled with 2 so multiply the result with 2.

$\frac{N^P}{x}$

$\frac{(x+1)^P}{x} \qquad \frac{(x-1)^P}{x} \qquad \frac{(x+y)^P}{x}$

$(x+1)^n = xQ + 1$

reduce

$(x+1)^1 = x(A) + 1$

$(x+1)^2 = x^2 + 2x + 1$
$= x(x+2) + 1$
$= x(B) + 1$

Ex. $\frac{5^3}{4} = \frac{(4+1)^3}{4}$

Rem $= \frac{4^3 + 3 \cdot 4^2 + 3 \cdot 4 + 1^3}{4}$

$= \frac{4(4^2 + 3 \cdot 4 + 3) + 1^3}{4}$

$= \frac{4(31) + 1}{4}$

$$\boxed{(x+1)^P \xrightarrow[\text{reduce}]{} x(Q)+1}$$ Concept

1) $\frac{7^{80}}{6} = \frac{(6+1)^{80}}{6} = \frac{(6Q+1)}{6} = 1$

2) $\frac{7^{80}}{48} = \cancel{(48-1)}\ \cancel{\frac{(48+1)}{48}} = \frac{(7^2)^{40}}{48} = \frac{(48+1)^{40}}{48} = 1$

3) $\frac{7^{85}}{48} = \frac{(48+1)^{42} \cdot 7^1}{48} = \frac{(48Q+1) \cdot 7^1}{48} = \frac{7}{48} = 7$

4) $\frac{7^{100}}{342} = \frac{(7^3)^{33} \cdot 7^1}{342} = \frac{(342+1)^{33} \cdot 7^1}{342} = 7$

$$\boxed{7^1 = 7 \quad 7^2 = 49 \quad 7^3 = 343 \quad 7^4 = 2401}$$

5) $\frac{13 \times 7^{81}}{48} = \frac{13 \times (7^2)^{40} \cdot 7^1}{48}$

$= \frac{13 \times (48+1)^{40} \cdot 7}{48} = \frac{91}{48} = 43$

II case

$(x-1)^P$ — $(x-1)^{odd} = xQ-1$

$(x-1)^{even} = xQ+1$

Remainder	Ans
-1	$x-1$
$-y$	$x-y$

$(x-1)^{odd} = xQ-1$

$= x[(Q-1)+1]-1$

$= x(Q-1)+x-1$

$\frac{4^3}{5} = (5-1)^3$

$= 5^3 - 3 \cdot 5^2 \cdot 1 + 3 \cdot 5 \cdot 1 - 1^3$

$= 5(5^2 - 3 \cdot 5 + 3) - 1$

$= \frac{5(13) - 1}{5}$

$= \frac{5(12+1) - 1}{5}$

$= \frac{5(12) + 5 - 1}{5}$

Q $\frac{7^{50}}{50} \Rightarrow \frac{(7^2)^{25}}{50} = \cancel{\frac{(49+1)^{25}}{50}} = \frac{(50-1)^{25}}{50} = \frac{(50Q-1)^{25}}{50}$

Ans: $50-1 = 49$

Q $\frac{7^{100}}{344} = \frac{(344-1)^{33} \cdot 7}{344} = \frac{(344Q-1) \cdot 7}{344} = \cancel{\frac{344 \times 7}{344}} = \cancel{344Q}$: $\cancel{\frac{(344Q - 7)}{344}}$

Ans: $(344-7) = 337$

$\frac{\cancel{7 \times 344Q} - 7}{\cancel{344}} = \frac{-7}{344} = -7$

Concept

$(x+y)^p \xrightarrow[\text{reduce}]{} \frac{x0+y^p}{x} \Rightarrow \frac{y^p}{x}$

Q $\frac{9^{60}}{7} = \frac{(7+2)^{60}}{7} = \frac{2^{60}}{7} = \frac{(2^3)^{20}}{7} = \frac{(7+1)^{20}}{7} = \frac{70+1}{7} = 1$

Q $\frac{10^{100}}{7} = \frac{(7+3)^{100}}{7} = \frac{3^{100}}{7} = \frac{(7+2)^{50}}{7} = \frac{2^{50}}{7}$

$\cancel{8}2^2 \frac{(7+1)^{16}}{7} = 2^2 = 4$

(or) $\boxed{1001 = 7 \times 11 \times 13}$

Q 6 (WB) P-49

Sol:- $\frac{7^{84}}{342} = \frac{(343)^{28}}{342} = \frac{(342+1)^{28}}{342} = \frac{342\,0+1}{342} = 1$

Reasoning

Factors

Type-1

Total no of factors ?

$36^2 = 2^2 \times 3^2$ ← factorized form

How many factors 72 has?

$72 = 2^3 \times 3^2 \longrightarrow 3^0, 3^1, 3^2$

$= (3+1)(2+1) = 4 \times 3 = 12$

$2^0, 2^1, 2^2, 2^3$

$(2^0+2^1+2^2+2^3)(3^0+3^1+3^2)$

$(1+2+4+8)(1+3+9)$

$(1, 3, 9, 2, 6, 18, 4, 12, 36, 8, 24, 72)$

Factorized form

$N = a^p \times b^q$

$\boxed{\text{total no of factors} = (p+1)(q+1)}$

ii) $360 = 2^3 \times 3^2 \times 5^1$

no of factors $= (3+1)(2+1)(1+1) = 4 \times 3 \times 2 = 24.$

Q 1 (WB) P-49

Sol factors of 1800

$1800 = 2^3 \times 3^2 \times 5^2$

$= (3+1)(2+1)(2+1)$

$= 36$

Type-2

Sum of the factors

$72 = 2^3 \times 3^2 = (2^0+2^1+2^2+2^3)(3^0+3^1+3^2)$

$= (15)(13)$

$= 195$

Scanned by CamScanner

$360 = 2^3 \times 3^2 \times 5^1$

$(2^0+2^1+2^2+2^3)(3^0+3^1+3^2)(5^0+5^1)$

$= (15)(13)(6)$

$= 195 \times 6$

$= 1170$

Q $N = a^p \times b^q$

$(a^0+a^1+\ldots a^p)(b^0+b^1+\ldots+b^q)$

$= \left(\frac{1(a^{p+1}-1)}{a-1}\right)\left(\frac{1(b^{q+1}-1)}{b-1}\right)$

$$\boxed{N = \left(\frac{a^{p+1}-1}{a-1}\right)\left(\frac{b^{q+1}-1}{b-1}\right)}$$

Type 3, 4 → Composite factors

↓ prime factors

Q How many factors of 72 are — prime no's / Composite no's

$72 = 2^3 \times 3^2$

= 2, 3 are prime factors

Count prime no's in factorized form = # of prime factors

$360 = 2^3 \times 3^2 \times 5^1$

2, 3, 5 are prime factors. ie 3 prime factors

Q 2 P-49 (WB)

Sol. $30^7 \times 22^5 \times 34^{11}$ ↙ factorized form

$$\boxed{2^{23} \times 3^7 \times 5^7 \times 11^5 \times 17^{11}}$$

5 prime factors

$$\boxed{\text{\# Composite factors} = \text{total factors} - (\text{Prime factors} + 1)}$$

$72 \Rightarrow 12 - (2+1) = 9$

$360 \Rightarrow 24 - (3+1) = 18$

Type 5, 6 → odd factors

↓ Even factors

How many factors of 72 are — odd / even

$72 = 2^3 \times 3^2$

$(2^0+2^1+2^2+2^3)(3^0+3^1+3^2)$

$(1+2+4+8)(1+3+9)$

odd factors → 1 3 9   8 24 72

2 6 18

4 12 36

To get odd factors = delete 2^n from the factorized form & count the remaining factors of result

$360 = \cancel{2^3} \times 3^2 \times 5^1$

$(2+1)(1+1) = 6$ odd

even = total − odd

$= 24 - 6 = 18$

Q8 P-49 (WB)

Sol:- N = 10800

# of even factors.

$10800 = 2^2 \times 3^3 \times 2^2 \times 5^2$

$= 2^4 \times 3^3 \times 5^2$

total = (60)

odd = 12

even = 48

Base System

base	digits
10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
2	0, 1
3	0, 1, 2
9	0, 1, 2, 3, 4, 5, 6, 7, 8
16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

I Conversion of ~~Any~~ No. from any base to decimal

$(123)_{10} \longrightarrow (123)_{10}$

$1\times10^2 + 2\times10^1 + 3\times10^0$

$(123)_8 \longrightarrow (83)_{10}$

Q $(1101101)_2 \longrightarrow (109)_{10}$

Q $(453)_7 \longrightarrow (234)_{10}$

Q $(2AB)_{15} \longrightarrow (611)_{10}$

II $(\text{Number})_{10} \longrightarrow (\quad)_{\text{any base}}$

Q $(83)_{10} \longrightarrow (123)_8$

83 10 1

Q $(143)_{10} \longrightarrow (10001111)_2$

143 71 35 17 8 4 2 1

Q $(746)_{10} \longrightarrow (2114)_7$

106 15 2

Q $(1311)_{10} \longrightarrow (51F)_{16}$

i) Sum >= base then $\frac{\text{Sum}}{\text{base}} <^{Q}_{R}$

III) Addition

$$\begin{array}{r} 345 \\ +\ 131 \\ \hline (476)_8 \end{array} \qquad \begin{array}{r} {\scriptstyle 1\ 1} \\ 347 \\ +\ 236 \\ \hline (605)_8 \end{array} \qquad \begin{array}{r} {\scriptstyle 1\ \ 1} \\ 1347 \\ +\ 2136 \\ \hline (3505)_8 \end{array}$$

$\frac{13}{8}$ Q = 1, R = 5

IV) Subtraction

$$\begin{array}{r} 564 \\ -\ 137 \\ \hline (425)_8 \end{array} \qquad \begin{array}{r} 3264 \\ -\ 1537 \\ \hline (1525)_8 \end{array}$$

Q 62 (WB) P-79

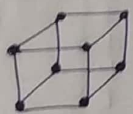
Sol: $(7526)_8 - (Y)_8 = (4364)_8$

$(7526)_8 - (4364)_8 = (Y)_8$

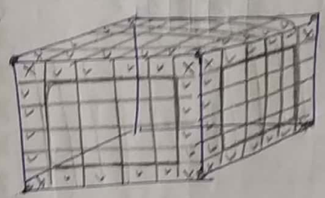
$$\begin{array}{r} 7526 \\ -\ 4364 \\ \hline (3142) \end{array} = Y$$

Reasoning

Cube, Dice, & Direction



8 corners
6 faces
12 edges



Cube $(6 \times 6 \times 6) = n\,(1 \times 1 \times 1)$

$n = 6 \times 6 \times 6$
$= 216$ small cubes $(1 \times 1 \times 1)$

All 216 small cubes $(1 \times 1 \times 1)$ are P

How many cubes $(1 \times 1 \times 1)$ will have exactly

a) 1 face painted: 6 faces × 16 = 96

b) 2 face painted: 12 edges × 4 = 48

c) 3 face painted: 8 corners × 1 = 8

a) No face painted

$(n-2)(n-2)(n-2)$

$(6-2)(6-2)(6-2)$

$= 4 \times 4 \times 4$

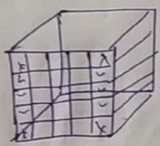
$= 64$

(or) $216 - (~~18~~\ 96 + 48 + 8)$

$216 - (152)$

$= 64$

Q1-4 P-59



Sol 1) $a^3 = n(\ )$

$a^3 = 125\ (1 \times 1 \times 1)$

$a = 5$

~~125~~ 2) $8 \times 1 = 8$

3) $12 \times 3 = 36$

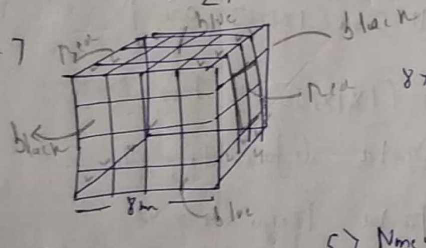
4) $6 \text{ faces} \times 9 = 54$

1) $125 - (54 + 36 + 8)$

$= 125 - (98)$

$= 27$

Q 5-7



$8 \times 8 \times 8 = n(2 \times 2 \times 2)$

$64 = n$

6) $4 \times 4 = 16$

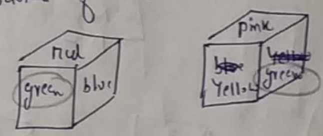
7) $2 \times 4 = 8$

5) None of these

Dice

6 faces of a dice are painted with 6 different color i.e. blue, brown, red, green, pink, yellow

2 orientation of dice are shown below



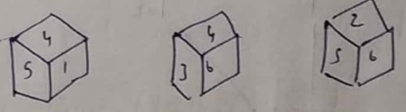
What is the color on the face opposite to "brown" color face? Ans - green

Sol:- green — red ✗, blue ✗, pink ✗, Yellow ✗, brown ✓

take out the common value

Q 8-9, 11 P-59

Sol.



8) 6 — 1 ✓, 2 ✗, 3 ✗, 4 ✗, 5 ✗

9) 5 — 1 ✗, 2 ✗, 3 ✓, 4 ✗, 6 ✗

(Q10)

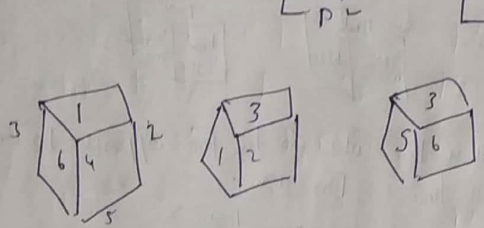
Opposite to A

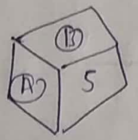
h - A ✗, B ✗, h ✗, M ✗, P ✓

H - A ✓, B ✗, K ✗, M ✗, P ✗

(Q11)

Sol)





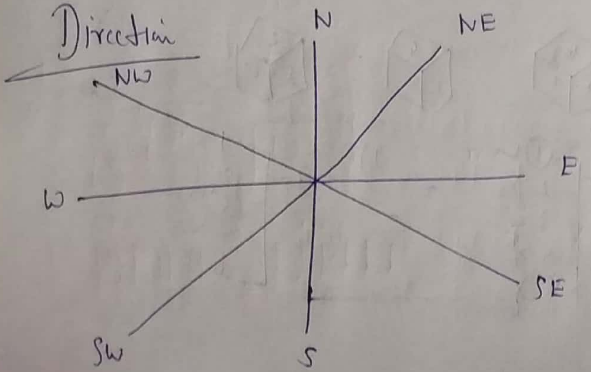
find A & B

3 - 1 ✗, 2 ✗, 4 ✓, 5 ✗, 6 ✗

6 - 1 ✗, 2 ✓, 3 ✗, 4 ✗, 5 ✗

Ans - (a) 2 and 3

Direction



29/11/18

Q Aman moves 40 m North

then takes Right turn and moves 50 m

" " Left " " " 60 m

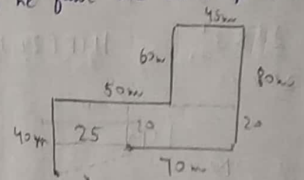
" " Right " " " 45 m

" " Right " " " 80 m

" " Right " " " 70 m

How far is he from the start point

Sol:-

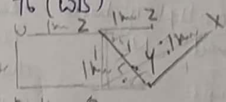


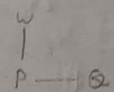
$x = \sqrt{625 + 400} = \sqrt{1025} = 32.01$

(Imp)

Q37 P - 76 (WB)

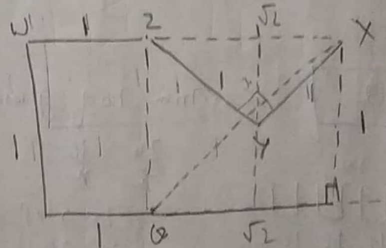
Sol:-





$1 - 0.25$

$1 + 2\sqrt{0.75}$



QX = √3

LCM | HCF

LCM → Least Common Multiple

HCF / GCD → Highest Common factor / Greatest Common Divisor

LCM → take the highest power of each prime

HCF → take the lowest power of each prime

Ex: $24 = 2^3 \times 3^1$ $\quad$ $HCF(24, 36) = 2^2 \times 3^1 = 12$

$36 = 2^2 \times 3^2$

LCM

N	N
24	36
24	36
48	72
72	108
96	144
144	⋮
⋮	

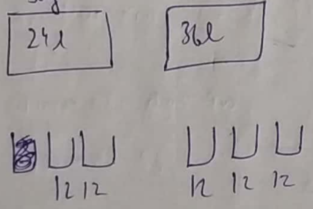
→ Smallest no multiple of both and > N

divisible

N	N
24	36
12	12

↓ largest ~~Smallest~~ no which divides both and < N

Q. Jug



=> n = 5

min no of glasses?

$\frac{24}{x}$ $\quad$ $\frac{36}{x}$

$24 = 2^3 \times 3^1$

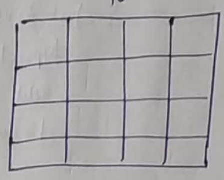
$36 = 2^2 \times 3^2$

$2^2 \times 3^1 = 12$

Gate Q

16

12



All square tiles are of the same size → integer

min no of tiles?

x  x

$\frac{16}{x}$ $\quad$ $\frac{12}{x}$

$16 = 2^4 \times 1$

$12 = 2^2 \times 3$

$2^2 \times \cancel{3^1} = 4$

HCF (4)

Type-3

$\frac{N}{12}$ $\quad$ $\frac{N}{15}$ $\quad$ find N such that

a) Smallest no, $LCM(12, 15) = 60$

b) 3rd Smallest 60, 120, 180 ✓

c) largest (3) digit no.

960

d) Smallest (4) digit no.

Sol c) 60) 999 (16
60
399
360
39

$999 - 39 = 960$

1 digit less

Sol d) Least 4 digit no = greater (3) digit + LCM

$= 960 + 60$

$= 1020$

Type-2

Q $\frac{N}{12}$ $\frac{N}{15}$

Rem 7 7

Just solve like type-1 and then add the rem to get answer

a) Smallest no 67

b) 3rd smallest no = 187

c) greatest ③ digit no = 967

60) 999 (16
60
399
360
39

999 − 39 = 960 ⇒ 960 + 7 = 967

d) Smallest ④ digit no = 1020 + 7 = 1027

Type-3

$\frac{N}{12}$ $\frac{N}{15}$

12−4 = 8, 15−7 = 8

rem 4 7

a) Smallest no 60 − 8 = 52

b) greatest ③ digit no

960 − 8 = 952

c) Smallest ④ digit no.

1020 − 8 = 1012

d) 3rd smallest

180 − 8 = 172

Type-4

$\frac{N}{12}$ $\frac{N}{15}$

Rem 7 13

HW

a) Least no? → 43

b) greatest ③ digit no.

Sol. 12 + 7 = 19 19/5 = 4

24 + 7 = 31 31/5 = 1

36 + 7 = 43 43/5 = 13

48 + 7 = 55

LCM (12, 15) = 60

43 + 60

43 + 120

43 + 180

⋮

43 + 900 = 943

Percentage

→ Overall % change
→ Single % change
→ Successive % change
→ Application

Reciprocal

$\frac{1}{1} = 100\%$	$\frac{1}{11} = 9.09\%$
$\frac{1}{2} = 50\%$	$\frac{1}{12} = 8.33\%$
$\frac{1}{3} = 33.33\%$	$\frac{1}{13} = 7.7\%$
$\frac{1}{4} = 25\%$	$\frac{1}{14} = 7.14\%$
$\frac{1}{5} = 20\%$	$\frac{1}{15} = 6.66\%$
$\frac{1}{6} = 16.66\%$	$\frac{1}{16} = 6.25\%$
$\frac{1}{7} = 14.28\%$	$\frac{1}{17} = 5.88\%$
$\frac{1}{8} = 12.5\%$	$\frac{1}{18} = 5.55\%$
$\frac{1}{9} = 11.11\%$	$\frac{1}{19} = 5.26\%$
$\frac{1}{10} = 10\%$	$\frac{1}{20} = 5\%$

$\frac{1}{4} = 25\%$	$\frac{1}{3} = 33.33\%$	$\frac{1}{7} = 14.28\%$
$\frac{1}{8} = 12.5\%$	$\frac{1}{6} = 16.66\%$	$\frac{1}{14} = 7.14\%$
$\frac{1}{16} = 6.25\%$	$\frac{1}{12} = 8.33\%$	

all 1's $\frac{1}{9} = 11.11\ldots1\%$	$\frac{1}{19} = 5.26\%$
all 5's $\frac{1}{18} = 5.5\ldots5\%$	$\frac{1}{20} = 5\%$
all 90's $\frac{1}{11} = 9.09090\ldots\%$	$\frac{1}{21} = \cancel{5.76}\ 4.76$

all 3's $\frac{1}{30} = 3.33\ldots\%$	$\frac{1}{13} = 7.7\%$
all 6's $\frac{1}{15} = 6.666\ldots\%$	$\frac{1}{17} = 5.88\%$

Overall % change

initial $a \longrightarrow b$ final

$$\text{overall \% change} = \frac{b-a}{a} \times 100\%$$

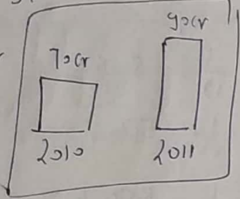
$120 \longrightarrow 130$ $\quad \frac{10}{120} = \frac{1}{12} = 8.33\%$

$170 \longrightarrow 180 = \frac{10}{170} = 5.88\%$

$\frac{19}{34} = \frac{17+2}{34} = 50\% + 5.88 = 55.88\ \%$

$\frac{19}{34} = 0.5588$

Q.



$70 \longrightarrow 90$

$\frac{20}{70} = 2 \times \frac{1}{7} = 2 \times (14.28)$
$= 28.56\ \%$

Single % Change

$x \xrightarrow{r\%\uparrow} x\left(\frac{100+r}{100}\right)$

$x \xrightarrow{r\%\downarrow} x\left(\frac{100-r}{100}\right)$

$x \xrightarrow{13\%\uparrow} x \times 1.13$

$x \xrightarrow{\downarrow 7\%} x * (1.07)$

$x \xrightarrow{17\%\uparrow} x * (1.17)$

$x \xrightarrow{8\%\downarrow} x(0.92)$

$x \xrightarrow{9\%\downarrow} x(0.91)$

Q Salary of Ram has gone up 30%. The new salary is Rs 182. Find initial salary

So. $\cancel{\frac{30}{182} \times 100 = x}$ 3%

$x \longrightarrow 182$

$x \times (1.3) = 182$

$x = \cancel{\frac{1820}{13}}\ 140$

Q A is 20% more than B

1) by what % is B of A

2) by what % B less than that of A?

Sol:- 1) $B = 100$
$A = 120$

$\frac{100}{120}\ \frac{5}{6} = 5 \times (16.66)$
$\sim 83.30\%$

~~2) $\frac{120}{120} \times 100 = \frac{100}{120}\ 3 = 83.33\%$~~

$B \longrightarrow A = (1.2)B$

$\frac{B}{A} = \frac{10}{12} = 83.3\%$

2) $\frac{20}{120} \times 100 = \frac{2 \times 100}{12} = 16.66\%$

Successive % change (More than one change)

$$x \xrightarrow{\uparrow a\%} x\left(\frac{100+a}{100}\right) \xrightarrow{b\%} x\left(\frac{100+a}{100}\right)\left(\frac{100+b}{100}\right)$$

$$x \xrightarrow{\uparrow 20\% \quad \uparrow 30\%} x\times(1.2)\times(1.3)$$

$$x\times(1.56)$$

overall

56% change

$$x \xrightarrow{a\%\uparrow \quad b\%\uparrow} \left(a+b+\frac{ab}{100}\right)x$$

overall % change

$$x\left(\frac{100+a}{100}\right)\left(\frac{100+b}{100}\right)$$

$$\frac{x}{100^2}\left(100^2+100a+100b+ab\right)$$

$$1+\frac{x}{100^2}\left[100a+100b+ab\right]$$

$$1+x\left[\frac{100a+100b+ab}{100}\right]\frac{x}{100}$$

$$= 1+\left[a+b+\frac{ab}{100}\right]\% \text{ of } x$$

$$x \xrightarrow{20\%\uparrow \quad 30\%\uparrow} \left(20+30+\frac{20\times30}{100}\right) \Rightarrow 56\%$$

overall % change

Case I

$$x \xrightarrow{10\%\uparrow \quad 20\%\uparrow} \quad 10+20+\frac{10\times20}{100} = 32\%$$

Case II

$$x \xrightarrow{\downarrow 10\% \quad \downarrow 20\%} \quad \cancel{20}\ -10-20\ \frac{-10\times-20}{100} \Rightarrow -28\%\downarrow$$

Case III

$$x \xrightarrow{\uparrow 20\% \quad \downarrow 10\%} \quad 20-10+\frac{20\times-10}{100} = 10-2 = 8\%\uparrow$$

$$x \xrightarrow{\downarrow 50\%\ \text{discount} \quad \downarrow 50\%\ \text{discount}} -50-50+\frac{(-50)\times(-50)}{100} = -75\%\downarrow$$

3 changes

$$x \xrightarrow{10\%\uparrow \quad 20\%\uparrow \quad 30\%\uparrow} x(1.1)(1.2)(1.3)$$

$x(1.32)(1.3)$

$x(1.716)$

ie 71.6% overall

↓

32%

⇓

71.6%

132 × 13 = 1716

$$x \xrightarrow{10\%\uparrow \quad 20\%\uparrow \quad 10\%\uparrow \quad 20\%\uparrow}$$

Scanned by CamScanner

Application of %

1) Expenditure = Price × Qnty

P ↑ 20%.
Q ↑ 10%.

E'%, $E' = (1.2)P \times (1.1)Q$
$= (1.2)(1.1)PQ$
$E' = (1.32)PQ$
$E' = (1.32)E$
i.e. 32% ↑

2) Sales = Price × Qty

= ↑ 30%   ~~P~~ ↓ 20%.

$= (1.3)P \times (0.8)$ Qty
$= (1.04)$ P × Qty
$= +4\%$ ↑

3) Rectangle

$A = l \times b$ l ↑ 20%. b ↑ 20%.   ~~new~~ A% ↑

$A' = (1.2)l \times (1.2)b$
$= (1.44)lb$

44% ↑

P-50

Q3 ~~40l~~   Mixture
Sol.

25% water   10l   30l milk

~~75%~~ $\frac{3}{4}$ → $\frac{3}{5}$

$\frac{3}{5} - \frac{3}{4}$

$\frac{12-15}{20}$

$-\frac{3}{20} \times \frac{1}{4}$

Q6
Sol.

$x \xrightarrow{5\%} (0.95)x \xrightarrow{75\% \uparrow} (0.95)(0.25)x = 950$

$x = \frac{950}{(0.95)(0.25)x}$

Q8
Sol.

100
75% V   25% IV
39   36 → ~~2184~~ 3024

8400 = 100
.84 = 1
36 × 84

51-17
(34)

$\frac{46}{12}$

Imp
Q10 P-73
Sol:

5   3 $\frac{34}{3}$ + 4 = ~~$\frac{34}{3}$~~ $\frac{40}{3}$ 5

$$\left[\left[x \times \frac{2}{3} + 4\right] \times \frac{3}{4} + 3\right] \times \frac{1}{2} + 2 = 17$$

$x \times \frac{2}{3} = 32$ $x = \overset{16}{32} \times \frac{3}{2}$

$x = 48$

Q7

Sol:- ~~1.25~~ × SP $\xrightarrow{25\%\uparrow\ 25\%\downarrow}$ ~~(1.5625)~~ SP

$225 = (1.5625)\,SP$

$\frac{225}{1.5625} = SP$

SP $\xrightarrow{25\%\uparrow\ 25\%\downarrow}$ $x \times 1.25 \times \cancel{1.25}^{0.75} = 225$

$x \times \frac{5}{4} \times \frac{3}{4} = 225$

$x = \cancel{225}^{15} \times \frac{16}{\cancel{15}}$

$x = 240$

Calendar

Any year ⟶ leap year

Non-Century year — divisible by 4

Normal year: 100, 200, 300, 500, 600, 700, 900, ...

Century year — divisible by 400: 400, 800, 1200, 1600

Probability for year to be leap year $= \frac{97}{400}$

1 — 100 : 24

101 — 200 : 24

201 — 300 : 24

301 → 400 : 25

→ 97

in 1st 100 yr, 100th yr is not leap yr, it is Normal year.

Odd days

Concept :- $\frac{\text{total days}}{7}$ = remainder = odd days

Normal yr $\frac{365}{7} = 1$ ⟹ odd day

$7 \overline{)365}(52$

35

15

14

1

Q In Normal yr, find prob of 53rd Monday?

Sol:- 52 weeks ⟶ 1 odd days

31st dec

M T W T F S Sun

$\frac{1}{7}$

leap year $\frac{366}{7}$ = 2 odd days

100 Years

1 — 100

76 Normal — 76×1 = 76

24 leap yr — 2×24 = 48

$\frac{124}{7} = 5$ odd day

1 — 100 5 $\frac{10}{7} = 3$
101 — 200 5
201 — 300 5 $\frac{15}{7} = 1$
301 — 400 6 $\frac{21}{7} = 0$

yr	odd days
100	5
200	3
300	1
400	0

Total odd day

Remainder	day
0	Sunday
1	Monday
2	Tuesday
3	Wednesday
4	Thursday
5	Friday
6	Saturday

301 — 400
Normal = 75 75×1 = 75
leap yr = 25
2×25 = 50
75 + 50 = 125
$\frac{125}{7} = 6$

Normal	Odd day
~~Normal~~ yr	1
leap yr	2

Jan $\frac{31}{7} = 3$

~~Feb $\frac{28}{7} = 0$~~

Feb — leap $\frac{29}{7} = 1$
Feb — Normal $\frac{28}{7} = 0$

March $\frac{31}{7} = 3$

April $\frac{30}{7} = 2$

Today date → 30 November 2018

Years gone 2017 | Current yr 2018

2017
2000 + 17
400 → 0
2000 → 0
2

Quotient will give leap year
4)17(4
4 ~~odd day~~ leap year

17 — 13 Normal = 13
17 — 4 leap year = 8
21 odd day

Current yr	
Jan	3
Feb	0
Mar	3
April	2
May	3
June	2
July	3
Aug	3
Sept	2
Oct	3

$\}$ 24

Nov → 30

yr 21
Month 24
days 30
75

$\frac{75}{7} = 5$ → Friday

Day → 15 August 1947

1946
1900 + 46 | Current yr

1900 — 1600 → 0
1900 — 300 ~~km~~ → 1

4)46(11
44
2

11 leap year

46 — 35 → 35
46 — 11 → 22
$\}$ 58

Jan	3
Feb	0
Mar	3
April	2
May	3
Jun	2
July	3

$\}$ 16

Aug → 15

Month = 16
day = 15
yr = 58
$\frac{89}{7} = 5$ Friday

Q 26 Jan 1950

1949
1900 + 49 < 37, 12
1600 → 0, 300 → 1

Day = 26
Month = 0
Yr = 62

$\frac{88}{7} = 4$ ⟶ Thursday

1 + 24 + 37 = 62

Shifting Technique

Q 30 Nov 2018 ⟶ Friday

Then 30 Nov 2020 ⟶ ?

30 Nov 2018 ⟶ Saturday
Normal +1
30 Nov 2019 ⟶ Saturday
leap +2
30 Nov 2020 ⟶ Monday

Q 26 Jan 1950 ⟶ ~~Thursday~~ Thursday
+1
26 Jan 1951 ⟶ Friday
+1
26 Jan 1952 ⟶ Saturday
29th Feb 1952 +2
26 Jan 1953 ⟶ Monday

P - 49

Q2

Sol:-

11th Aug 2010 — Wednesday
2009 -1
2008 -2
2007 -1
2006 -1
2005 -1
2004 -2
2003 -1
2002 -1
2001 -1 — Saturday

9 years: 2 leap year, Normal

$\frac{11}{7} = 4$

2×2=4+7=11 odd days

Q3 August 1988 Friday

88 + 1900 < 1600, 300 — 1
88 → 22

88 < 66 - 66, 22 + 44 = 110

Q4 Sol:

16th July 2000 — Sunday

~~16th July~~ 20th December 2000

16th Aug 2000 → 3
16th Sept 2000 → 3
16th Oct 2000 → 2
16th Nov 2000 → 3
16th Dec 2000 — Tuesday

$\frac{16}{7} \Rightarrow 2$

Q5 Sol: 375 days

Q4 16th July 2000 → Sunday 15

July − 15/7 = 1
Aug − 3
Sept − 2
Oct − 3
Nov − 2
20th Dec − 20/7 = 6 → Wednesday

$\frac{17}{7} = 3$

Q5
Sol: 385 days 8 days in week

$\frac{385}{8} = 1$

Q6
Sol: In leap year Jan 26th → Friday

Jan → 5/7 = 5
Feb → 1
March → 3
Apr → 2
May → 3
Jun → 2
July → 3
15th Aug → 15/7 = 1 → Thursday

$\frac{20}{7} = 6$

Profit loss And Discount

CP —Profit/loss→ SP —discount→ MP

CP 100, SP 140, discount 30%, 60, MP 200

(30% of 200) = 60

SP > CP Profit = 140 − 100 = 40

SP < CP Loss.

Rule discount is always calculated on MP

$$\boxed{CP + Profit = SP = MP - discount}$$

If no discount is given

CP + Profit = SP

CP − loss = SP

Q Ram sells a book at 40% profit. If the SP is 182. Find CP

$\frac{40}{100}C + C = SP$

$\frac{140}{100} C = 182$

$C = \frac{182 \times 100}{140}$

C = 130

Q Ram sells a book at 6% profit. had he sold it at 17% profit, then he would have earned 88 Rs more. Find CP

Sol:

$$x + \frac{6}{100}x = SP$$

$$x + \frac{17}{100}x = 88 + SP$$

$$\left(x + \frac{17}{100}x\right) - \left(x + \frac{6}{100}x\right) = 88$$

$$\frac{117x}{100} - \frac{106x}{100} = 88$$

$$x = \frac{88 \times 100}{11}$$

$$x = 800$$

Q Ram sells a book at 8% loss. Had he ~~received~~ sold it at 7% profit, then he would have received 105 more. Find CP

S:

$$\left(\frac{107}{100}x\right) - \left(\frac{92}{100}x\right) = 105$$

$$x = \frac{105 \times 100}{15}$$

$$x = 700$$

Q MP of a chair is increased by 50% over the cost price. If the discount is offered by 20%. Then what will be overall profit %.

Sol. CP = 100

MP = 150

Discount = 30

SP = 120

P% = 20%

Q Even after giving a discount of 11.11% shopkeeper still can earns profit of 14.28%. by what % the shopkeeper has marked up CP.

Sol:

CP —— SP —— MP

$\frac{8x}{9}$ x

$$CP + \frac{1}{7}CP = \frac{8x}{9}$$

$$CP\left(\frac{8}{7}\right) = \frac{8x}{9}$$

$$CP = \frac{7}{9}x$$

$$\cancel{x - \frac{7}{9}x = \frac{2}{9}x = \frac{2}{9}x \quad 22.22\%}$$

$$\frac{CP}{MP} = \frac{7}{9} \qquad CP \longrightarrow MP$$

$$7 \underbrace{\qquad}_{2} 9$$

$$\frac{2}{7} \times 100 = 2 \times 14.28 = 28.56$$

Scanned by CamScanner

Q CP of 15 books = SP of 12 books

overall Profit %.

Sol. SP of 15 books − Profit = SP of 12 books

SP of 3 books = Profit

Profit % = $\frac{\text{SP of 3 books}}{\text{SP of 12 books}} \times 100 = 25\%$.

LCM (15, 12) = 60

15 books → 60

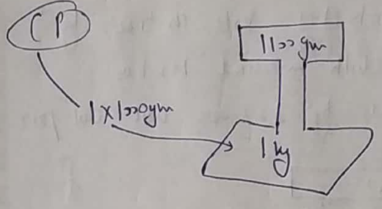
CP 1 book → 4 SP of 1 book → 5

P% = $\frac{1}{4} \times 100 = 25\%$.

Faulty Series

Trader has marked up his good by 20%. faulty balance shows 1100 ~~gram~~ grams for a kg find overall profit / loss ? 20% ↑

Sol. Assume CP = 1 Rs/gm MP = 1.2 Rs/gm



1100 × 1.2 = 1320

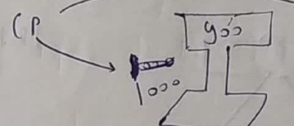
Profit = 320

P% = $\frac{320}{1000} \times 100 = 32\%$.

Q Trader has marked up his goods by 10%. and faulty balance 900 gm / kg. What is overall profit / loss ?

10% ↑

Sol:-



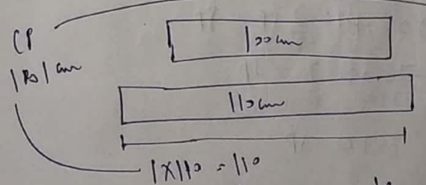
MP 1.1 × 900 = 990

Loss = $\frac{10}{1000} \times 100 = 1\%$.

Faulty Scale

Q During summers meter scale expand by 10%. if there is no marker on the CP. Find overall profit / loss %. No mark up

Sol:

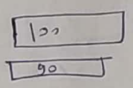


1 × 110 = 110 ↑ CP

100 × 1 = 100 ↑ SP

$\frac{10}{110} \times 100 = 9.09\%$ loss

Q During winter meters scale contract by 10%. If the cloth merchant has marked up his goods by 20% over ~~cost~~ cost price find profit %.

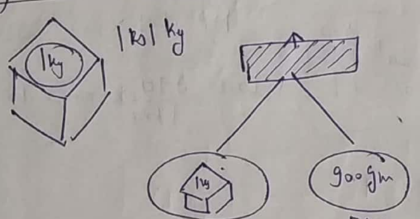


Soln: CP = 90 $\quad$ SP = 1.2 × 100 = 120

$P\% = \frac{30}{90} \times 100 = \frac{1}{3} = 33.33\%$

faulty Weight

1 Rs / kg $\quad$ 1 Rs / kg



CP = 900 $\quad$ SP = 1000

$P\% = \frac{100}{900} \times 100 = 11.11\%$

Q 14 (WB) P - 51

Soln: $7290 \div \frac{2}{7} SP = SP$

$\cancel{7290} \times \frac{7}{9} = SP$ (810)

5670 = SP

Q 8
Soln:

CP ——— SP ——— MP

SP = 75 $\quad$ −25 $\quad$ MP = 100

$CP + \frac{20}{100} \times CP = 75$

$CP = \cancel{75} \times \frac{100}{120} = \frac{75}{2}$

$\frac{CP}{MP} = \frac{75}{200}$

CP $\quad$ MP
75 $\quad$ 200

$\frac{125}{75} \times 100 = 5 \times 33.33 = 5$

$CP = \frac{75}{2}$ $\quad$ MP = 100

$MP - CP = \frac{\frac{125}{2}}{\frac{75}{2}} \times 100 = \frac{125}{75} \times 100 = \frac{500}{3} = 1$

CP ——— SP ——— MP

SP = $\frac{3}{4}x$ $\quad$ MP = x

$CP + \frac{20}{100} \times CP = \frac{3}{4}x \Rightarrow CP + \frac{1}{5} CP = \frac{3}{4}x$

$CP\left(\frac{120}{100}\right)$ $\quad$ $CP\left(\frac{6}{5}\right) = \frac{3}{4}x$

$\frac{CP}{MP} = \frac{5}{8}$

CP $\quad$ MP
5 $\quad$ 8

$\frac{3}{5} \times 100 = 60\%$

Scanned by CamScanner

Q9 $216 = x(0.9)(0.8)$

$\frac{216 \times 100}{0.72} = x$

$x = 300$

Q10

Sol: CP ——— SP ——— MP

100 5% 150

$100 + 5 = SP$

$SP = 105$

$\frac{45}{15} \times 100 = 30\%$

Q 12

Sol: CP of 12 = SP of 15

SP of 12 − Profit = SP of 15

Profit = − SP of 3

loss = SP of 3

$\text{loss}\% = \frac{SP \text{ of } 3}{SP \text{ of } 15} \times 100 = 20\%$

Q 1

Sol. SP = 24k — P = 20%

SP = 24k — L = 20%

$CP + \frac{20}{100} \times CP = 24k$

$CP - \frac{20}{100} CP = 24k$

$CP = \frac{24000}{2000} \times \frac{100}{1}$

$CP = 20{,}000$

$\boxed{P\% = \frac{4000}{20{,}000} \times 100 = 20\%}$

$P\% = \frac{4000}{20{,}000} \times 100 = 20\%$

$CP_2 = \frac{2400}{300} \times \frac{100}{1}$

$CP_2 = 30{,}000$

$L = \frac{6000}{30{,}000} \times 100$

$= 20\%$

Total CP = 50,000

SP = 48,000

$L\% = \frac{2000}{50{,}000} \times 100 = 4\%$

~~Overall = 20 − 20 + 20×20/100 = 18 − 1/5 = 18~~

~~Overall = 20 − 10 − 20×10/100 = 4% loss~~

Q 2

Sol:- Overall loss $= 10 - 10 + \frac{10 \times -10}{100} = 1\%$ loss

~~$CP_1 = 3500 \times \frac{100}{110}$~~

$3500 + \frac{10}{100} \times 3500 = SP$

$3500 \times \frac{110}{100} = SP_1$

$3850 = SP_1$

$P\% = \frac{350}{3500} \times 100 = 10\%$

$3500\left(\frac{90}{100}\right) = SP$

$3150 = SP_2$

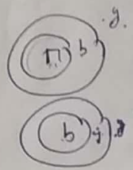
$L\% = \frac{350}{3500} \times 100$

$= 10\%$

Total SP = 3850 + 3150
= 7000
CP = 7000

~~Sitting Arrangement~~

Syllogism (logical Reasoning)



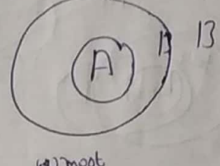
Stmnt : ① All red are blue
② All blue are green

Conclusion : ① All green are red ×
② Some red are green ✓

a) only 1
Ⓑ only 2 ✓
Ⓒ (i) ✓ & (ii) ✓
Ⓓ (i) × & (ii) ×

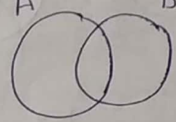
Cases ① ② ③
All All Some
All Some Some

1) All A's are B's



(at most)
2) Some A's are B's

A B



Stmnt : All ~~studen~~ Students are Indians

Conclusion : Some Students are Indians

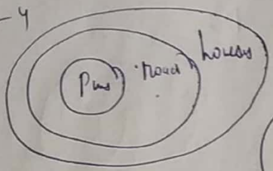
3 Rules

1) Draw all possibilities from the given stmnts
2) For a conclusion to be true, it has to be true in all possibilities.

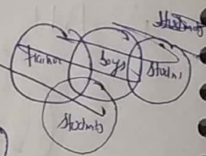
③ If a conclusion is false, even ~~one~~ at one possibility, it will be considered false for ever.

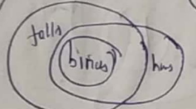
Q 1-4

i)

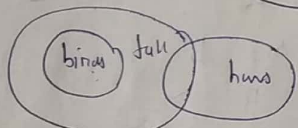


iii)

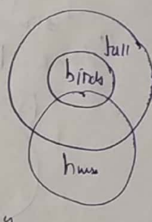




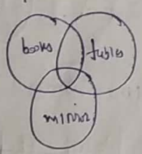
ii)



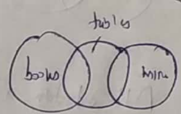
(or)



iv)



(or)

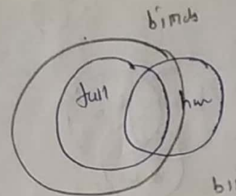


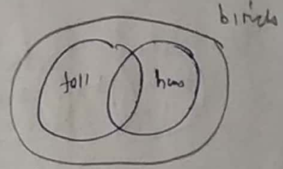
Scenario 2

i) All tall are birds

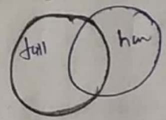
ii) Some tall are hens

First process "Some"





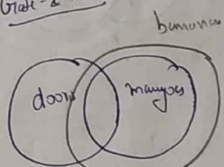
bird overall tall

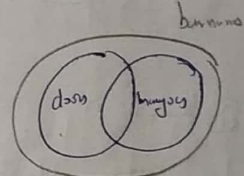


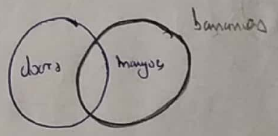
P-62

Q9 (WB) Grade-2016

Sol:







Scenario ③

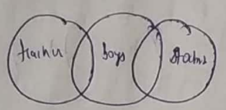
Stmnt: ① Some teacher are boys

② Some boys are students

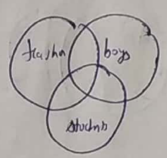
Conclusion i) Some students are boys

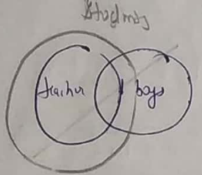
ii) Some teachers are students

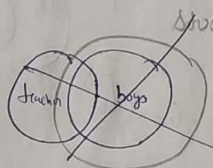
1 — 3



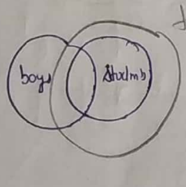
1 3







2 — 3



Q 7

Sol. Stmnt

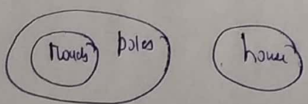
All roads are poles

No pole is a house

Conclusion

Some roads are houses F

Some houses are poles F



Ratio Proportion, Alligation (Mixture & Solution), Replacement Method

Q Divide 784 into 4 parts such that 4a : 3b : 2c : ...

Sol:- $4a = 3b = 2c = 12D$

$\frac{a}{b} = \frac{3}{4}$ $\quad \frac{b}{c} = \frac{2}{3}$ $\quad \frac{c}{d} = \frac{12}{2}$

$a : b : c : d$

$6 : 8 : 12 : 2$

$a : b : c = 3 : 4 : 6$

$a : b : c : d = \cancel{3 : 4 :} 6 : 8 : 12 : 2$

$a = 6 \times \frac{\cancel{784}}{\cancel{28}}$ 28

$a = 168$ $\quad b = 8 \times 28 = 224$ $\quad c = 12 \times 28$

$\boxed{D = 2 \times 28 = 56}$

$4a = 3b = 2c = 12D = k$

$\frac{k}{4} : \frac{k}{3} : \frac{k}{2} : \frac{k}{12}$ $\quad k = 1$

$a : b : c : d = 3 : 4 : 6 : 1$

$d = \frac{1}{14} \times 784 = 56$

Q 1 Re coin $\quad$ 50 P coin $\quad$ 25 Paise coin

Total value of ~~coin~~ = coin = 210 Rs

no of coins 5 : 6 : 8

How many 25 Paise coins in it?

Sol:- ~~Value of 25P coin = $\frac{8 \times 210}{15}$ =~~

Ratio of Value of coin = 5 : 3 : 2

Value of 25P coin = $\frac{2}{10} \times 210 = 42$ Rs

\# of 25P coin = $\frac{42 \times 4}{1} = 168$ coins

Combining the ratios

Ram : Shyam : Mohan

a : b

c : d

ac : bc

bc : bd

ac : bc : bd

Ram : Shyam : Mohan : Gautam

a : b

c : d

e : f

ace : bce

cbe : dbe

edb : fdb

ace : bce : bde : bdf

Q1 P-53

Sol A : B = 3 : 2 A : B : C = 15 : 10 : 8

B : C = 5 : 4

C : D = 3 : 7 A : B : C : D = 45 : 30 : 24 : 56

Q2

Sol:- no of coins = 3 : 7 : 9

Value of coin = $\frac{3}{4} : \frac{7}{5} : \frac{9}{10}$

= 15 : 28 : 18

61 = 61

1 ≡ 1

Value of 20p = 28 Rs

\# of 20p coins = $\frac{28}{1} \times 5 = 140$

I | II | III | IV Proportion

1st proportion

a, b, c ⟶ 1st proportion

x ; a, b, c

x : a :: b : c

$\frac{x}{a} = \frac{b}{c}$

$x = \frac{ab}{c}$

4th proportion

a, b, c, x

a : b :: c : x

$\frac{a}{b} = \frac{c}{x}$

$x = \frac{cb}{a}$

Scanned by CamScanner

Mean proportion (2nd proportion)

a, x, b

$a : x :: x : b$

$\frac{a}{x} = \frac{x}{b} \Rightarrow \boxed{x = \sqrt{ab}}$

3rd Proportion

a, b, x

$a : b :: b : x$

$\frac{a}{b} = \frac{b}{x}$

$\boxed{x = b^2 a}$

Q 3

Sol:- $M : W = 6 : 5$

$\cancel{\frac{6+16}{5+16}} = \frac{10}{9}$

$\frac{6x+16}{5x+16} = \frac{10}{9}$

$54x + 144 = 50x + 160$

$4x = 16$

$x = 4$

$M = 24$

$W = 20$

Proportion

Direct   Indirect

Direct proportion

$x \propto y$

$x = ky$

$\frac{x}{y} = k$ (Const?

$\frac{x_1}{y_1} = \frac{x_2}{y_2}$

Volume Constant

$P \propto T$

$P = kT$

$\frac{P}{T} = k$ (Volume Const)

$\boxed{\frac{P_1}{T_1} = \frac{P_2}{T_2}}$

Time (Constant

$S \propto D$

$S = k \cdot D$

$\frac{S}{D} = k$ (Time (const.)

$\boxed{\frac{S_1}{S_2} = \frac{D_1}{D_2}}$

Time Constant

Q In 100m race, A beats B by 10m.
In 200m race, B beats C by 20m.
In new 400m race, By how what distance A will beat C?

Sol. $\frac{A}{B} = \frac{100}{90} = \frac{10}{9}$ $\frac{B}{C} = \frac{200}{180} = \frac{20}{18}$

$\frac{A}{C} = \frac{A}{B} = \frac{10 \times 2}{9 \times 2}$ $\frac{B}{C}$ $A : B : C = \frac{200}{180}$ $\frac{180}{182}$

180 m — 182

$\frac{A}{B} = \frac{100}{90} = \frac{10}{9}$

$\frac{B}{C} = \frac{200}{180} = \frac{10}{9}$

$\frac{A}{B} = \frac{100}{90}$ $\frac{B}{C} = \frac{90}{81}$

$\frac{A}{C} = \frac{100}{81}$

100 — 19

400 — $\frac{400}{100} \times 19 = 76$

Q Price of a diamond is directly proportional to the square of its weight. The diamond breaks into 2 pieces in the ratio 7 : 3 (by weight). What is overall % change in value?

Sol

$P \propto W^2$

$P = kW^2$

$\frac{P}{W^2} = k$

Value = $P \Rightarrow kW^2$

$P = k(100)$

$7+3 = W$

$\left(\frac{7}{10}W\right)$ $\left(\frac{3}{10}W\right)$

$\frac{7}{10} \times \sqrt{P}$ $\frac{3}{10}\sqrt{P}$

new

$W = 7+3$

$= 10$

$\left\{ \begin{array}{ll} P \text{ of } 7y & k(7)^2 \\ P \text{ of } 3y & k(3)^2 \end{array} \right.$

42 58k

$100k \xrightarrow{42} 58k$

$\frac{42}{100} \times 100 = 42\%$ decrease in value

Q Height of Ram is directly proportional to the square root of his age. What is the height of Ram after 7 years if he was 4 feet tall at 9 years ago

Sol

$H \propto \sqrt{A}$

$4 = k\sqrt{9}$

$\frac{4}{3} = k$

$H = \frac{4}{3}\sqrt{16}$

$\frac{4}{3} \times 4 = \frac{16}{3}$

$= \frac{40}{3}$ approx

$= 1\frac{1}{3}$ feet

Indirect Proportion

$x \propto \frac{1}{y}$

$x = \frac{k}{y}$

$xy = k$ (const)

if Temp const.

$P \propto \frac{1}{V}$

$P = k\frac{1}{V}$ Temp

$\boxed{PV = k}$ (const)

Distance const

$S \propto \frac{1}{T}$

$S = \frac{k}{T}$

$S \cdot T = k$ (distance) const

Expenditure = price × Qty

↑ const

price $\propto \frac{1}{Qty}$

P-50

Q 9

Sol.

$E = P \times Q$

↑ 10% ↓ x%

E = const

$\left(\frac{11}{10}\right)P \times \left(\frac{10}{11}\right)Q = k$

$Q - \frac{10}{11}Q = \frac{1}{11}Q = 9.09\%$

Q Due to reduction in price of apples by 20%, Ram was able to purchase 25 apples more. If the total expenditure is Rs 600.

i) find the original price of apple

ii) find original quantity purchased

Sol: $n \times q = 600$

$\downarrow 20\%$

$\left(\frac{80}{100}\right)$

$$\boxed{q\left(\frac{4}{5}\right)\left(\frac{5}{4}\right)n = 600 \qquad 600 = n \times q \qquad n - \frac{5}{4}n = -\frac{1}{4}}$$

$600 = \left(\frac{5}{4}\right)n\left(\frac{4}{5}\right)q$

$\frac{5}{4}n - n = 25$

$n = 100$

Price $= \frac{600}{100} = 6$

Q Ram

Home S —— office $T = x$ min

Today speed of Ram $\downarrow 20\%$ due to which he was able to reach office 11 min late. What is his usual time

$\frac{5S_1}{4S_1} = \frac{T_2}{T_1}$

$\frac{5}{4} = \frac{T_2}{T_1}$ $\qquad T_1 = T_2 \times 4/5$

$5T_1 = (T_1 + 11)4$

$5T_1 = 4T_1 + 44$

$T_1 = 44$ min

$S \longrightarrow T = x$

$4/5\,S \longrightarrow \frac{5}{4}T = x + 11$

$\frac{1}{4}T = 11$

$T = 44$ min

Partnership

Partners A & B

Capital $I_A : I_B$

time $T_A : T_B$

Profit $I_A \times T_A : I_B \times T_B$

$$\frac{P_A}{P_B} = \frac{I_A \times T_A}{I_B \times T_B}$$

Q 9 P-54

sol: C 5 : 8

8 months $\longrightarrow$ $\frac{1}{2} = \frac{5 \times 8}{8 \times t_B}$

$t_B = 10$

Mixture & Solution

Alligation

	A	B
Rice	9/kg	11/kg
Qty	7kg	3kg

$$\text{Avg} = \frac{9 \times 7 + 11 \times 3}{7+3} = \frac{96}{10} = 9.6/\text{kg}$$

c d

V_c V_d

$$m = \frac{c \cdot V_c + d \cdot V_d}{V_c + V_d} \Rightarrow m(V_c + V_d) = cV_c + dV_d$$

$$\boxed{\frac{V_c}{V_d} = \frac{d-m}{m-c}}$$

Nature of c, m, d must be same

Q11 P-54

So: c 62/kg d 72/kg

c	m	d
62	64.5	72

V_c V_d

$$\frac{V_c}{V_d} = \frac{72 - 64.5}{64.5 - 62} = \frac{7.5}{2.5}$$

$$\frac{V_c}{V_d} = \frac{3}{1}$$

Scanned by CamScanner

Q
Sol
M_1

m w
4 8
V_m V_w
1 1

$\frac{1}{1} =$

Replacement Concept

I. 120 l milk — $-12l$ 10%, 80l taken out, replace 12l water

II. m : w = 92l : 10l ... -9 $-1l$ ... replace by 10l water ... 81, 19

III. 81l : 19l ... $-12l$ 8.1 1.9 ... replace by 10l water ... 72.9 17.1

IV. m w 72.9 27.1

$$\text{final milk} = \text{initial milk}\left(1 - \frac{\text{taken out Qty}}{\text{Total volume}}\right)^n$$

TS P - 54
Sol
m w : 3 5 ; 15l 25l ; 40l ... 8l : 3 5
m w : 12l 24l
$\frac{m}{w} = \frac{1}{2}$

Q 10
Sol
m w : 4 3, +5, 8 ; 4/8
m w
$4+8 = 12 = 12$
M 20, W = 20
$\frac{4x}{3x+5} = \frac{1}{1}$
$4x = 3x + 5$
$x = 5$

m w : 120l 80l → 40l : 24l 16l ; 96l 64l ; 200

m w : 96l 84l → 40l : 21.3 18.6 ; 74.7 65.4 ; 200

final milk = $(120)(0.8)(0.8) = 120 \times 0.64 = 76.8l$

Water = $200 - 76.8 = 123.2l$

Q 4
Sol: $P/d = 2/4 = 1/2$ $1+2 = 400$, $3 = 400$, $1 = \frac{400}{3}$

$P = 2 \times \frac{400}{3}$

$P + d = 150$ $P/d = 1/2$ $2P = d$
$3P = 150$
$P = 50$ $d = 100$

Q 7
Sol: 5 chairs = 12 stools
7 stools = 2 table
3 table = 2 sofa = 2×175 = 350
5 sofa = 875
1 sofa = 175

8 stools = $2 \times \frac{350}{3} \times \frac{1}{7} \times \frac{12}{5}$ = 700
table = $\frac{350}{3}$ = 80

21.3, 74.7, 18.6, 96 × 4/3, 8 × 4/3

Seating Arrangement

Row
Circular

P-60
Q 1
Sol

D B (A) E C

DBA
A_E C

Q 7-Q 8
Sol

D (Ex) B F (Ex)

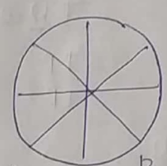
A (Ex) E C (Ex)

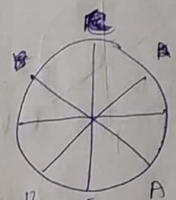
D _F
~~E~~
C E C

Circular Arrangement

1. Always divide circle into equal parts.
2. Always assume that all occupants are facing center, accordingly decided left (or) right
3. C is sitting between A and B

Explore 2 cases

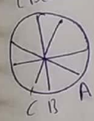




4. Left means immediate left (IL)
   Right means immediate right (IR)
5. A and B are not sitting together
   means not adjacent to each other
6. A, B, C are sitting together

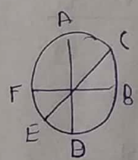
clockwise

A B C → anticlockwise



C B A

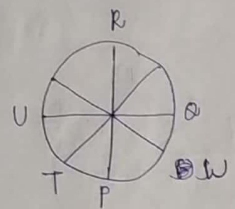
Q2-3
Sol:
E
AFED
ACB



Q4-6

S:

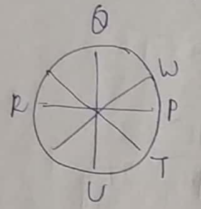
PQRSTUVW

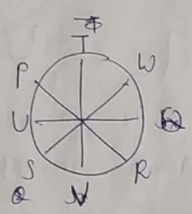


TWQ

TP

TP



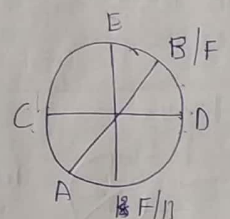


Q

TP

Q10-9

Q



ABCDEF

AFDB

E_A_D

Time & Work

→ Ratio

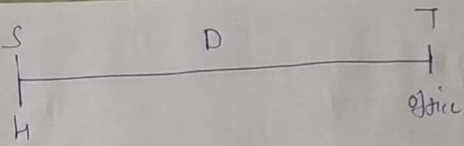
→ Avg speed

→ Relative speed

→ Circular speed

→ Miscellaneous

Q



S

Speed ↑ by 20% and he was able to reach office 7 min early. Find usual Time

$T_2 = \frac{5}{6} T_1$ $\quad (T_1 - 7) = \frac{5}{6} T_1$

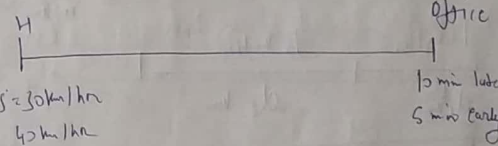
$6T_1 - 42 = 5T_1$

$T_1 = 42$ m.

$S_1 = \frac{S_1}{S_2} = \frac{5}{6} = \frac{T_2}{T_1}$

$S_1 = \frac{5}{6}$

Q



Find d = ? time ? usual speed ?

$\frac{3}{4} = \frac{T_2}{T_1} = \left(\frac{t-5}{t+10}\right) = \frac{3}{4}$

$4t - 20 = 3t + 30$

$t = 50$ min

~~d = 30 × 50/2 = 25 km~~ $\quad d = 30 \times \frac{60}{60} = 30$ km

~~S = 25 × 60/50 = 30 km/hr~~ $\quad S = \frac{30}{50} \times 60 = 36$ km/hr

Average Speed

$$S_{avg} = \frac{\text{Total distance travelled}}{\text{Total time}}$$

A km/hr | B km/hr

t_1 | t_2

$$\boxed{S_{avg} = \frac{At_1 + Bt_2}{t_1 + t_2}}$$

if time is const in each phase of Journey

if $t_1 = t_2 = t$

$$S_{avg} = \frac{(A+B)t}{2t} = \frac{A+B}{2}$$

arithmetic mean

A km/hr | B km/hr

d_1 km | d_2 km

$$\boxed{S_{avg} = \frac{d_1 + d_2}{\frac{d_1}{A} + \frac{d_2}{B}}}$$

if distance is constant in each ~~ca~~ phase of journey

$$S_{avg} = \frac{d+d}{\left(\frac{d}{A} + \frac{d}{B}\right)} = \frac{2}{\left(\frac{1}{A} + \frac{1}{B}\right)} = \frac{2AB}{A+B}$$

Harmonic Mean

Raman · Home | Office

Morning $S = 30$ km/hr

← Evening $S = 40$ km/hr

$$\text{avg Speed} = \frac{2}{\frac{1}{30} + \frac{1}{40}} = 2 \times \frac{30 \times 40}{70} = \frac{240}{7} = 34.28 \text{ km/hr}$$

Relative Speed

Relative speed is applicable only if both things are simultaneously start.

Q. Delhi — Thief → 40 km/hr, 7 AM — Gurugram 120 km — Jaipur

10 AM Police → 50 km/hr

hounds dog → 60 km/hr

by the time police was able to catch the thief. What is the overall distance travelled by Dog?

$$T = \frac{120}{20} = 6 \text{ hr}$$

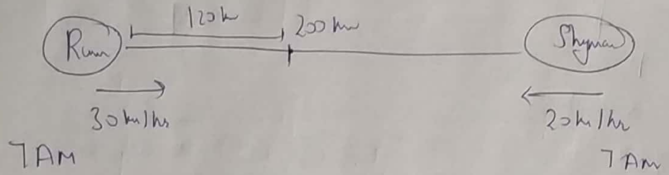
~~D = 6×60 = 360 km~~

~~Thief = $\frac{360}{4}$~~

$$t = \frac{120}{10} = 12 \text{ hr}$$

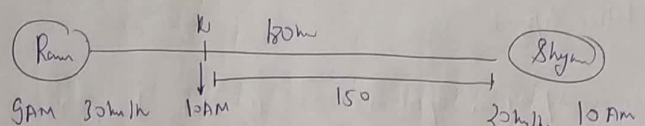
$$D = 60 \times 12 = 720 \text{ km}$$

↑ Overall distance travelled by dog



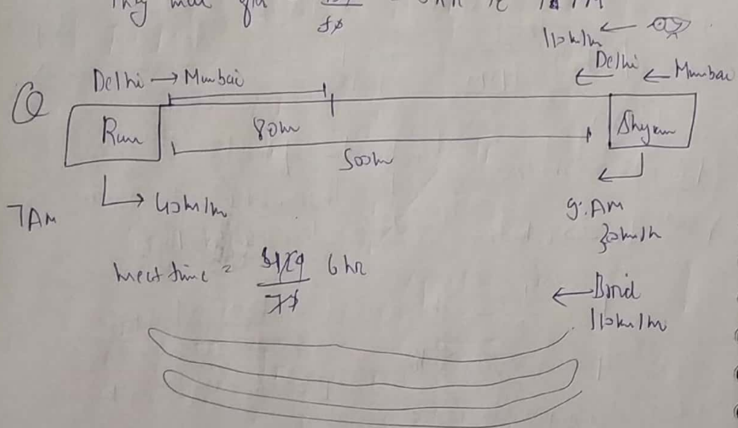
They will meet after = $\frac{200}{50}$ = 4hr i.e at 11AM

~~distance~~ distance = 4x30 = 120km from Ram



They meet after = $\frac{15\cancel{0}}{5\cancel{0}}$ = 3hr ie 1PM

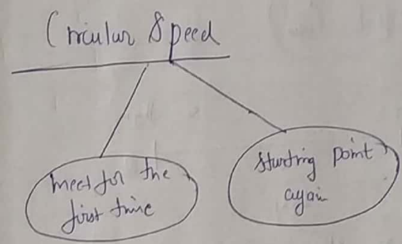
Q

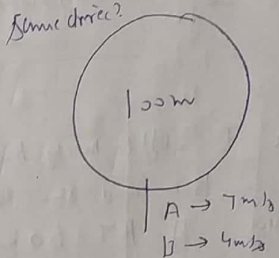


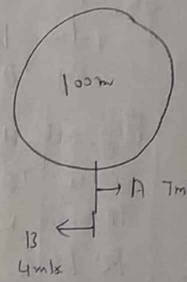
Meet time = $\frac{540}{90}$ 6 hr

~~overall distance~~ By time Ram & Shyam meet what is overall distance travelled by the bird

6x110 = 660 km

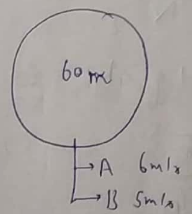






i) Meet first time = $\frac{100}{7-4} = \frac{100}{3}$

ii) Meet first time = $\frac{100}{7+4} = \frac{100}{11}$ sec



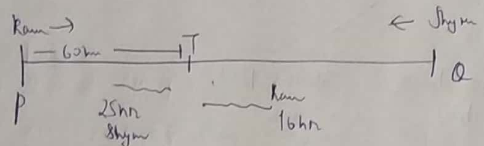
$t_A = 10$

$t_B = 12$

meeting at starting point = LCM(10,12) = 60

find each time to complete the track. Take LCM of time

Miscellaneous (Special Cases)



both Ram & ~~Shyam~~ ~~are~~ Shyam are heading towards each other

Ram P → Q

Shyam Q → P

They met at Point T . After meeting Ram covers TQ distance in 16hr and Shyam covers TP distance in 25 hr . If PT = 60 km What is ~~PQ~~ PQ = ?

Sol) $\frac{S_R}{S_{Sh}} = \frac{T_{Sh}}{T_R}$

$\frac{S_R}{S_K} = \frac{T_{Sh}}{T_R}$

$S_{Sh} = \frac{60}{25} = \frac{12}{5}$ km/hr

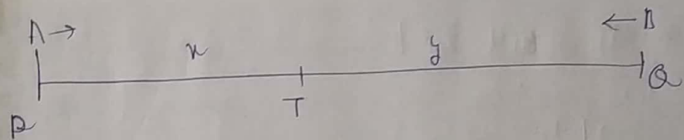
$\frac{S_R}{12/5} = \frac{25hr}{T_R}$

Speed of Shyam $= \frac{60}{25} = \frac{12}{5}$

$\frac{S_R}{S_{Sh}} = \sqrt{\frac{T_{Sh}}{T_R}}$

$S_R = \sqrt{\frac{25}{16}} \times S_{Sh}$

$S_R = \frac{5}{4} \times \frac{12}{5} = 3$ km/hr



When they meet at T

$$\boxed{\frac{S_A}{S_B} = \frac{x}{y}} \quad -① $$

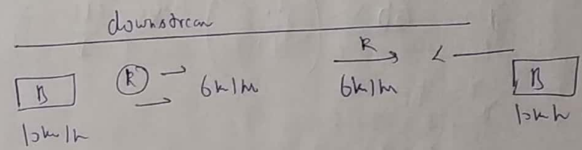
$y = S_A \times T_A$

$x = S_B \times T_B$

$\frac{S_A}{S_B} = \frac{S_B \times T_B}{S_A \times T_A} \Rightarrow \frac{S_A^2}{S_B^2} = \frac{T_B}{T_A}$

$$\boxed{\frac{S_A}{S_B} = \sqrt{\frac{T_B}{T_A}}}$$

Boat & River Stream



$D = B + R$

$= 10 + 6$

$D = 16$ km/hr

$U = B - R$

$= 10 - 6$

$U = 4$ km/hr

punit.singh@gmail.com
www.facebook.com/UPSCACADS
09999 178844

Time and Work

- → Ratio
- → General concepts
- → Alternate work
- → Pipe & Cistern
- → Miscellaneous

Men	hr	Tree
40	8	60
32	12	x

less more

$$x = 60 \times \left(\frac{32}{40}\right) \times \left(\frac{12}{8}\right)$$

$x = 72$

General Concept

$$\text{no of days} = \frac{\text{total work}}{\text{Per day work}}$$

Q total work = 30 unit

Ram —— 10 days 3 unit / day R+S → 5 unit / day

Shyam ——→ 15 day 2 unit / day 6 days

Q total work = 60

A → 10	6	
B → 12	5	
C → 15	4	$\frac{60}{20} = 3$ days
D → 20	3	
E → 30	2	
	per day	

Q total work = 30 unit

A+B → 10 days

A alone → 15 day

A+B = 3

B = 1

B alone = $\frac{30}{1}$ = 30 days

Q total work = 60 unit

A+B → 10 days 6

B+C → 12 days 5

C+A → 15 days 4

$2(A+B+C) = 15$

i) $A+B+C = \frac{15}{2}$ unit / days $\frac{60 \times 2}{15}$ = 8 days

ii) A alone = $\frac{60 \times 2}{5}$ = 24 days

iii) B alone = $\frac{60 \times 2}{7} = \frac{120}{7}$ days

iv) C alone = $\frac{60 \times 2}{3}$ = 40 days

Q

48

$A+B \longrightarrow 12$ days 4

$B+C \longrightarrow 16$ days 3

C alone = ?

48

end of 5th day — A left

end of 7th day — B left

end of 13th day — work finished

A+B+C — A left — B left — work finish

2×3 — 6x — = 13

6 unit

A+2B+C = 7
A+C+2B = 7
A − C = 1
A = C
C+2B+C = 7
B+C = 7/2
~~A+C~~
(B+C) = 7/2
A+B+C = 7 − 7/2
A = 7 − 7/2
= 7/2
x +

~~11x + 6 = 48~~
~~11x = 42~~
~~x =~~

~~13(11x + 6x) = 48~~
~~13(11x + 6x) = 48~~
~~11x + 3x + 78 = 48~~

A + 2B + C = 7
x + 2B + x = 7
2B = 7 − 2x
(B + x) = 7/2
B = 7/2 − x

~~7/2 − x + A = 4~~
~~7/2 − 4 = A − x = 1/2~~
~~A = 1/2 + x~~

~~20 + 6~~
~~20 + C^{5 days} + 6 + C^{6 days} = 48~~

~~26 day + C^{11} days =~~

$20 + C^{5\text{days}} + 6 + C^{6\text{days}} = 48$

$26 + C^{11\text{days}} = 48$

$C^{11\text{days}} = 22$ $\qquad C^{1\text{day}} = 2$ unit

$C^{\text{alone}} = \frac{48}{2} = 24$ days

Alternate Work

Total work = 24 unit

Sita ⟶ 8 hr

Geeta ⟶ 12 hr

Sita & gita are working in alternate hrs work start at 9 AM. If Sita starts the work

i) What time work is finished? 6:30 PM

ii) Who will finish the work? Geeta

T.W = 24 unit

Sol: S ⟶ 8 hr 3 U/hr

g ⟶ 12 hr 2 U/hr

S — SG SG ...

$\left(\frac{24}{5}\right)$ 4 cycle → 8 ~~days~~ hr + $\frac{4}{3}$ S ⟶ 3U

G ⟶ $\frac{1}{2}$

9 ~~days~~ hr + 30 min

If Geeta starts works ?

$\frac{[G\,3]}{5}\,5$

$4 \times 2 = 8$ hr

5 hr $+ \frac{2}{3}$ hr

5 hr $+ 40$ min

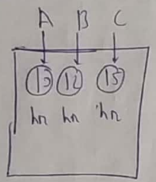
Work finish at 9:40 min

Sita finish the work

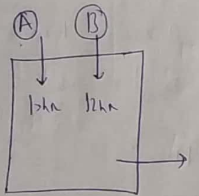
$4 \to 2 \to G$

$\frac{2}{3}$

Pipe and Cistern

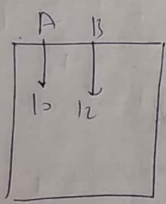


$A+B+C = \frac{60}{6+5+4} = \frac{60}{15} = 4$ hr



$A+B+C \to \frac{60}{6+5-3} = 7.5$ hr

Q



Tank empty

A + B are opened together

after x minutes pipe B is closed

it was found that the tank is full

in total 8 min.

find x ?

$T = 60$

$A \to 6$

$B \to 5$

$A+B \to 11$

$11x + (8-x)5 = 60$

$6x = 20$

$x = \frac{20}{6} \times 60 = 200$ min

$x = 200$ min

$11x + (8-x)6 = 60$

$5x = 12$

$x = 2.4 = 144$ min

P-56

Q 12

Sol

$A \to 15$ min ~~60/4 hr~~

~~$B \to -60$ hr -1 U/hr~~

~~$A+B \to 60$ min~~

60 w

$A \to -60$ m -1 U/m

$A+B \to 15$ min $+ 4$ U/m

$-1 + B = 4$

$B = 5$

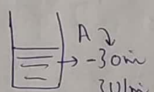
$\frac{1 - 1}{5 \text{ min}}$

$B \to 60/5 = 12$ min

P-77

Q 45

Sol



$B \to x$ U/m

$x - 3 =$

$\frac{1}{x} - \frac{1}{30} = \frac{1}{5}$

$\frac{1}{x} = \frac{1}{5} + \frac{1}{30}$

$\frac{1}{x} = \frac{6+1}{30}$ $x = \frac{30}{7}$

$= 4$

Miscellaneous

Type ①

3 Man or 4 Woman ⟶ 43 days

7 Man + 5 Woman ⟶ ? days

Sol

3 M ⟶ 43
4 W ⟶ 43
} ⟶ 3 M = 4 W

$1W = \frac{3}{4} M$

	Men	days
	3	43
7M + 5W 7M + $\frac{15}{4}$ M $\frac{43}{4}$ M	$\frac{43}{4}$ M	x

less

$x = 43 \times \left(\frac{3}{43/4}\right) = 12$ days

Type ②

5M ⟶ 6 days

10W ⟶ 5 days

3M + 7W ⟶ ? days

$3\cancel{0}M = 5\cancel{0}W$

$\frac{3M}{5} = W$

$3M + \frac{21}{5}$

$\frac{36}{5}$ 5M 6

$\frac{36M}{5}$ x

$x = 6\left(\frac{5\times5}{36}\right)$

$x = \frac{25}{6}$ 4.16

Work is Constant

Men efficiency $\propto \frac{1}{\text{day}}$

efficiency ⟹ $\boxed{\text{Men} \propto \frac{1}{\text{day}}}$

M × D = K (work const.)

5 × 6 = 30

1 × 30 = 30 unit

5 × 6 = 30

1 × (5 × 6) = 30

5 Men ⟶ 6 days

1 Men ⟶ 30 days

(or) 30 Men ⟶ 1 day

10W ⟶ 5 days

1W ⟶ 50 days

5 × 6 M = 10 × 5 W

$3\cancel{0}$ M = $5\cancel{0}$ W

$M = \frac{5}{3} W$

$\frac{3}{5} M = W$

$3M + \frac{3}{5} \times 7M$

$3M + \frac{3 \times 7}{5}M$

$\left(\frac{15 + 21}{5}\right) M$

$\left(\frac{36}{5}\right) M$

	Men	day
	5M	6
~~3M+5W~~ 3M + 7W	$\frac{36}{5}$ M	x

less

$x = 6 \times \left(\frac{5}{36/5}\right)$

Ans $x = \frac{25}{6}$

Alternate Methods

total work = 150 U

$1M \longrightarrow 30$ days $\longrightarrow 5\ U/D$

$1W \longrightarrow 50$ days $\longrightarrow 3\ U/D$

₹ $3M + 7W \longrightarrow \frac{150}{36}$

$3\times5 + 7\times3$

Type-3

$12M + 16B \longrightarrow 5$ days $\qquad 5(12M + 16B) \longrightarrow 1$ days

$13M + 24B \longrightarrow 4$ days $\qquad 4(13M + 24B) \longrightarrow 1$ days

$60M + 80B = 52M + 96B$

$8M = 16B$

$M = 2B$

$7M + 10B \longrightarrow ?$ days

	Boys	days
12M + 16B	40B	5
₹		
7M + 10B	24B	x

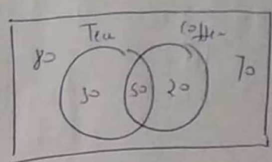
$x = 5\left(\overset{\text{more}}{\frac{40}{24}}\right) = \frac{50}{6} = 8\frac{2}{6}$ days

Set Theory

100 Students ⟶ each like at least ① drink

80 — Tea

70 — Coffee



How many like

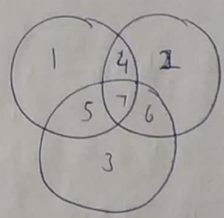
i) both drink 50

ii) Exactly ① drink 50

iii) Tea but not Coffee 30

– set difference

3 Sets

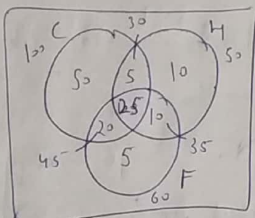


	Region
each like atleast ①	1, 2, 3, 4, 5, 6, 7
Exactly ①	1, 2, 3
Exactly ②	4, 5, 6
All ③	7
at least ②	4, 5, 6, 7
A and B but not C	4
A but not C	1, 4

Scanned by CamScanner

Q 125 Students - each like at least one game

100 - Cricket
50 - Hockey
60 - Football
30 - C∩H
35 - H∩F
45 - F∩C



125

Exactly (1) game = 65

Exactly (2) game = 35

All (3) game = 25

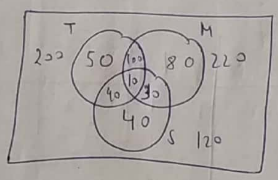
At least (2) games = 60

Cricket and Hockey but not football = 5

Cricket but not football = 55

Q 8-11

Sol

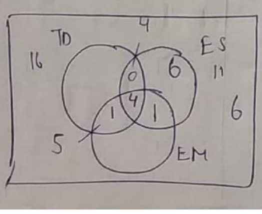


500

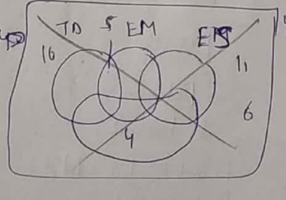
P-79

Q 68

Sol:



(ES∩EM) - TD



140

~~Clocks~~ P-50

Q 72

$n(E \cup S) = n(E) + n(S) - n(E \cap S)$

80% + 70% - ?

$\underset{\text{both fail}}{n(E \cup S)} = \underset{\text{fail}}{n(E)} + \underset{\text{fail}}{n(S)} - \underset{\text{both fail}}{n(E \cap S)}$

= 20% + 30% - 15%

= 35%. So 65% passed both the subjects

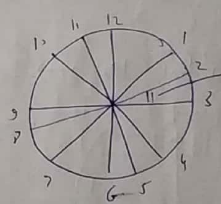
Let total stud be x

(65%) of x = 195

x = 300

Clocks

→ frequency
→ angle
→ meeting time



$\frac{360}{12} = 30°$

HH	MH
60 min → 30°	60 min → 360°
1 min → $\frac{1}{2}$°	1 min → 6°

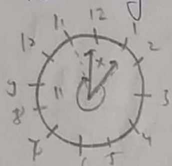
In every min

HH		MH		SH
$\frac{1}{2}$°	:	6°	:	360°

HH	:	MH	:	SH
1	:	12	:	720

Frequency

How many times in a day MH/HH

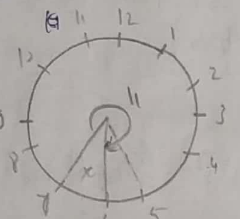


12hr → 11

24hr → 22

They do no coincide b/w 12 – 1
b/w 11-12 they coincide at 12

① Coincide → 22 times.

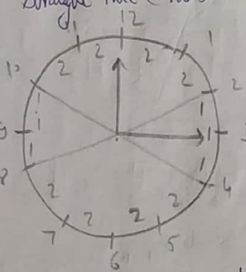


12 hr → 11

24 hr → 22 form straight line

They do not ~~coincide~~ b/w 6-7
b/w 5-6 they ~~coincide~~ at 6 form straight line

② Straight line (180°) → 22 times.



⊥ at 3 is common for 2-3 & 3-4

~~the~~

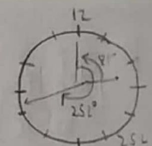
Similary ⊥ at 9 is common for 8-9 & 9-10

12 hr → 22

24 hr → 44 times

③ Perpendicular (⊥)(90°)
44 times.

Angle

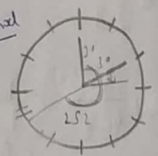


Current Time ⇒ 2 : 42

$$\begin{array}{r} 252 \\ -81 \\ \hline 171 \end{array}$$

What is the angle b/w HH/MH?

1st Method



HH

1 min → 1/2°

42 min → 21°

30 + 30 + 21 = 81°

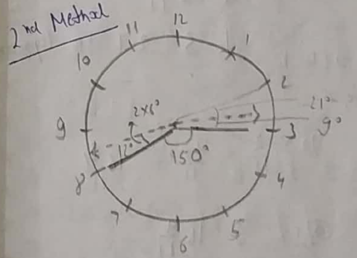
252 - 81 = 171°

MH →

1 min → 6°

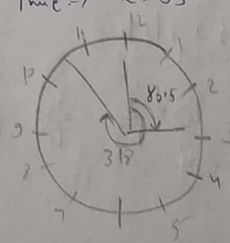
42 min → 42×6 = 252°

2nd Method



12° + 150° + 9 = 171°

Q Time ⇒ 2 : 53



(53×6) – (53/2 + 30 + 30)

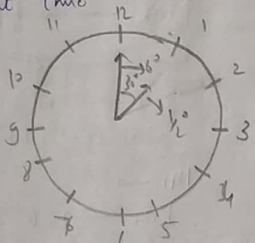
318 – (53/2 + 30 + 30)

318 – (86.5)

231.5

Meeting Time

B/w 1'O clock — 2'O clock
at what time HH – MH coincide.



$$\frac{30^\circ}{6-\frac{1}{2}} = 30 \times \frac{2}{11} = \frac{60}{11} = 5\frac{5}{11}$$

$1 : 5\frac{5}{11}$ min

$1 : 5 : 27$ sec (approx)

P-55

Q1. b/w 3'0 clock – 4'0 clock

$$\text{Coincide} = \frac{90^\circ}{6-\frac{1}{2}} = 90 \times \frac{2}{11} = \frac{180}{11} = 16\frac{4}{11}$$

3 hr : 16 : 21 sec (or) $3 : 16\frac{4}{11}$

Q b/w 4'0 clock & 5'0 clock

$$\text{Coincide} = \frac{120}{6-\frac{1}{2}} = 120 \times \frac{2}{11} = \frac{240}{11} = 21\frac{9}{11}$$

$4 : 21\frac{9}{11}$ min

b/w 1'O clock — 2'O clock
HH | MH become straight?

I-case

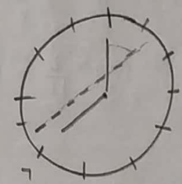
initial gap ⟶ 0 ⟶ 180°
30°

$30^\circ < 180^\circ$

$$\frac{30^\circ + 180^\circ}{6-\frac{1}{2}} = \frac{210}{11/2} = \frac{210 \times 2}{11} = \frac{420}{11} = 38\frac{2}{11}$$

$1 : 38\frac{2}{11}$ min

II-case



Initial gap

210° — 200 — 0 — 180°
↓
180°

$210^\circ > 180^\circ$

$$\frac{210 - 180^\circ}{6-\frac{1}{2}} = \frac{30}{11/2} = \frac{60}{11} = 5\frac{5}{11}$$

$1 : 5\frac{5}{11}$ min

Prob. B/w 1'o clock – 2'o clock at what times they are perpendicular

I case   initial gap

$30° \longrightarrow 0° \longrightarrow 90°$

$\frac{30+90}{6-\frac{1}{2}} = \frac{240}{11} = 21\frac{9}{11}$ min $= 1:21\frac{9}{11}$ min

II case

$30° \longrightarrow 0° \longrightarrow 270°$

$\frac{30+270°}{6-\frac{1}{2}} = 300 \times \frac{2}{11} = \frac{600}{11} = 54\frac{6}{11}$

$\boxed{1:54\frac{6}{11}}$

B/w 4'O clock – 5'o clock at what time they are perpendicular

$90°$

$120° — 0 —$

$\frac{30}{6-\frac{1}{2}} = \frac{60}{11} = 5\frac{5}{11}$

$4:5\frac{5}{11}$ min

$120° — 0 — 90°$

$\frac{120+90}{6-\frac{1}{2}} = \frac{210}{\frac{11}{2}} = \frac{420}{11} = 38\frac{2}{11}$

$4:38\frac{2}{11}$

Permutation Combination

1. number based questions
2. general concepts
3. Circular arrangements

Group Distributions

Using digits 1, 2, 3, 4, 5, 6, 7 form 4 digit no

i) without repetition of digits

$\underline{7}\ \underline{6}\ \underline{5}\ \underline{4} = 840$

ii) with repetition of digits

$\underline{7}\ \underline{7}\ \underline{7}\ \underline{7} = 7^4$

Using digits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 form 3 digit no

i) without repetition of digits

$\underline{9}\ \underline{9}\ \underline{8} = 648$

ii) with repetition of digits

$\underline{9}\ \underline{10}\ \underline{10} = 900$

Using digits 1, 2, 3, 4, 5, 6, 7 form 4 digit even no

i) without repetition of digits

$\overline{6}\ \overline{5}\ \overline{4}\ \overline{(2,4,6)} = 120 \times 3 = 360$

ii) with repetition

$\overline{7}\ \overline{7}\ \overline{7}\ \overline{(2,4,6)} = 7^3 \times 3$

Gate - 2014/16

Q Using digits 1, 2, 3, 4, 5 form 5 digit no without repetition of digits. What is the sum of all such no.s?

Sol:- $\overline{5}\ \overline{4}\ \overline{3}\ \overline{2}\ \overline{1} = 120$

Concept E: $\cancel{\frac{120}{5} \times (15) = 360}$

3 digit 1, 2, 3

10^2	10^1	10^0
12	12	12

1	2	3
1	3	2
2	1	3
2	3	1
3	1	2
3	2	1
13	3	2

$\frac{6 \text{ nos}}{3}$ = ie every digit we will get 2 times

2 (1, 2, 3)

2 (1+2+3)

= 12

$= 12 \times 10^2 + 12 \times 10^1 + 12 \times 10^0$

$= 12(10^2 + 10^1 + 10^0)$

$= 12(100 + 10 + 1)$

$= 12(111)$

$= 12 \times 111$

$= 1332$

1	2	3	4	5
1	2	3	5	4
⋮				
5	4	3	2	1
360	360	360	360	360
399	399	399	396	
399	9	9	6	0

← required sum

$\frac{120}{5} = 24$ times

24 (1+2+3+4+5)

24 (15)

360

$$\text{Sum of the no's} = (\text{Sum of given digits}) \times \frac{\text{Total no's}}{\text{no of digits given}} (111\ldots1)$$

$= 15 \times \frac{120}{5} (11111)$

$= 3999960$

General Concept

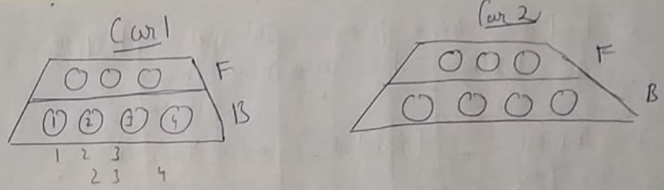
arrangement (P)	Selection (C)
1) All things are different	
$^{n}C_{r} * r!$	$^{n}C_{r}$
2) Same things are same	
$n = p + q$ $\frac{n!}{p!\,q!}$	$n = p + q$ $(p+1)(q+1)$ at least one $= (p+1)(q+1) - 1$
3) All things are same (ie all are identical)	
1	1

identical things

AAAA

{ select any

AAA 3



Girls: They will sit together
+
will sit in the back row

How many seating arrangements?

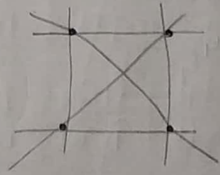
Sol: $\left(\underset{\text{car}}{^{2}C_{1}} \times \underset{\text{seat}}{^{2}C_{1}} \times \underset{\text{girl arrange}}{^{3}C_{3} \times 3!} \right) \times \left(^{11}C_{5} \times 5! \right)$

Lines

n points on a plane

No 3 points are collinear. How many lines?

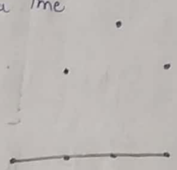
Sol: $^{n}C_{2}$



7 Points on a plane. Out of it, 4 points are collinear. How ~~many~~ many lines?

Total cases − limitation + extra line

$^7C_2 - {}^4C_2 + 1$



Triangles

n points on a plane.

No 3 points are collinear. How many △'s?

$= {}^nC_3$

7 points on a plane, out of it, 4 points are collinear How many △'s?

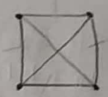
Total cases − limitation

$= {}^7C_3 - {}^4C_3$

Diagonals

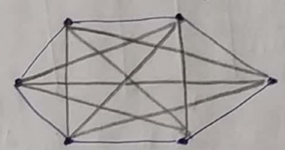
n points on a plane No 3 points are collinear. How many diagonals?

δ:- $^nC_2 - n$



Q 6 Points no of diagonals

$^6C_2 - 6 = 15 - 6 = 9$



Handshake

n people # handshake = $^nC_2$

Total Arrangements

MISSISSIPPI

$\overset{1}{M}\ \overset{4}{S}\ \overset{4}{I}\ \overset{2}{P} = \frac{11!}{1!\,4!\,4!\,2!}$

MISSISSIPPI AAO = 14

|M|S|S|S|S|P|P| $= \frac{7!}{4!2!} \times \frac{^8C_4 \times 7!}{4!2!}$

Q $\boxed{AAO}$ BCDFG

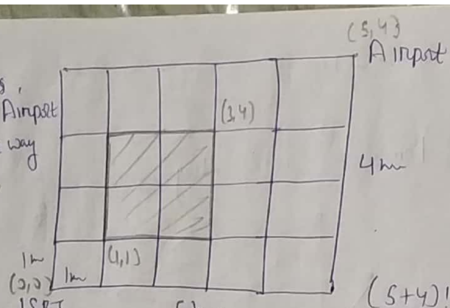
Vowels are always together ~~3!~~ $6! \times \frac{3!}{2!}$

$6! \times \frac{3!}{2!}$

Scanned by CamScanner

Q

From ISBT to Airport
all roads are one way
at any intersection
either move ↑ →
How many different
ways can you reach
airport ! ~~SE~~



$\frac{(5+4)!}{5!\,4!} = 126$

(or)

$^{5+4}C_5$

$\frac{9!}{5!\,4!} = \frac{9\times 8\times 7\times 6\times}{4\times 6} = 126$

HW

$$\boxed{\frac{2!}{1!\,1!}\left(\frac{9!}{5!\,4!}\right) - \left(\frac{2!}{1!\,1!} + \frac{3!}{2!\,1!}\right)}$$

(1+1)! (2+1)!

$= 126 - 5$

$= 121$

5 = 3 identical + 2 identical
AAA BB

Total selections $(3+1) \times (2+1) = 12$

at least ① $[(3+1)(2+1)] - 1 = 11$

× A / AA / AAA   B / BB   AB / ABB   AAB / AABB   AAAB / AAABB

If we have at least n different things

Total selections

$2^n = {}^nC_0 + {}^nC_1 + {}^nC_2 + {}^nC_3 + \dots\dots {}^nC_n$

at least one

$2^n - {}^nC_0 = {}^nC_1 + {}^nC_2 + \dots + {}^nC_n$

	AAAA 4 identical things	ABCD 4 different things
Total Selection	$(4+1)$	2^4
at least 1	$(4+1) - 1$	$2^4 - 1$
Select 2	1	$^4C_2$

	3 identical Pizza	4 identical Croissants	5 different cakes
Total selection	$(3+1)$ ×	$(4+1)$ ×	2^5
at least ①	$((3+1)$ ×	$(4+1)$ ×	$2^5) - 1$
Select ② of each	1 ×	1 ×	$^5C_2 = 10$

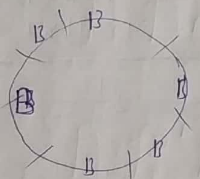
Scanned by CamScanner

Circular Arrangements

n things can be arranged circularly in $(n-1)!$ ways.

Q 6 boys, 3 girls

Circular arrange $\rightarrow$ no 2 girls are together



boys girls

$5! \times {}^{6}C_{3} \times 3!$

Garland (flower)

Necklace (Diamond)

$$\boxed{\frac{(n-1)!}{2}}$$

3 diff flowers A B C

How many different Garlands?

$\frac{(n-1)!}{2}$ $\quad \frac{(3-1)!}{2} = \frac{2!}{2} = 1$

Group Distribution Theory

identical Things distribution

n identical things if distributed among r persons or groups, where each can get $0, 1, 2, \ldots n$ things

Then total ways of distribution are ${}^{n+r-1}C_{r-1}$

$r \geq 0$
$S \geq 0$
$M \geq 0$

$\underbrace{\text{Ram} + \text{Shyam} + \text{Mohan}}_{r=3} = \underbrace{10 \text{ identical balls}}_{n=10}$

${}^{10+3-1}C_{3-1} \Rightarrow {}^{12}C_{2} = 66$ ways

$S \geq 0$
$R \geq 0$
$M \geq 1$
$S \geq 3$
$G \geq 2$

Ram + Shyam + Mohan + Sita + Geeta = 20 identical Chocolate

1 3 2

$\underbrace{\text{Ram} + \text{Shyam} + \text{Mohan} + \text{Sita} + \text{Geeta}}_{r=5} = 14$

${}^{14+5-1}C_{5-1} \Rightarrow {}^{18}C_{4}$

II case

$\underbrace{R + S + M + S + G}_{r=5} = 3$ identical Chocolate

$n = 3$

$r > n$

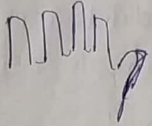
${}^{3+5-1}C_{5-1}$ $\quad$ ~~5+3=16~~

$\boxed{= {}^{7}C_{4}}$

How many ways 3 identical rings can be worn into 5 fingers.

$n = 5 \quad r = 3$

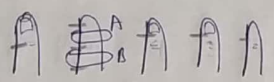
$^{3+5-1}C_{5-1} = {}^{7}C_{4}$

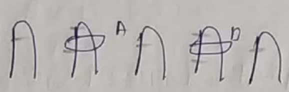


3 different rings in 3 different fingers

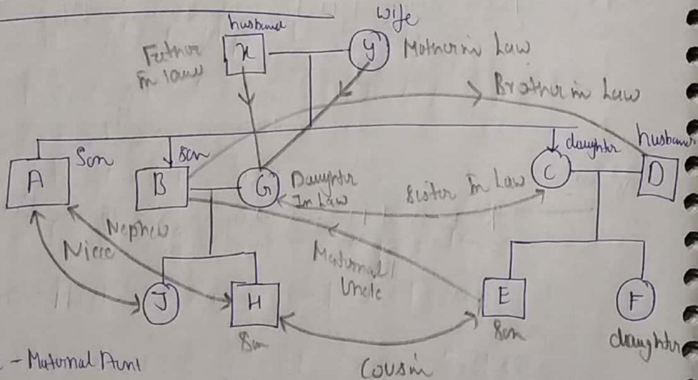
A B C

$5 \times 6 \times 7$





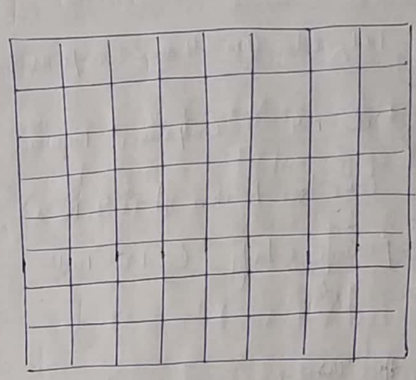
Blood Relation



Mossie - Maternal Aunt

Bua - Paternal Aunt

8 X 8 Chess board



How many total squares in a chess board?

$= 8\times8 + 7\times7 + 6\times6 + 5\times5 + 4\times4 + 3\times3 + 2\times2 + 1\times1$

$= 64 + 49 + 36 + 25 + 16 + 9 + 4 + 1 = 204$

$= \sum 8^2 = \frac{n(n+1)(2n+1)}{6}$

# of rectangles = $^9C_2 \times {}^9C_2$

How many different types of rectangle in a chess board?

1x1							
1x2	2x2						
1x3	2x3	3x3					
1x4	2x4	3x4	4x4				
1x5	2x5	3x5	4x5	5x5			
1x6	2x6	3x6	4x6	5x6	6x6		
1x7	2x7	~~3x7~~	4x7	5x7	6x7	7x7	
1x8	2x8	~~3x8~~	4x8	5x8	6x8	7x8	8x8
		3x7					
		3x8					
8	7	6	5	4	3	2	1

$8+7+6+5+4+3+2+1$

$= 36$

$\sum 8 = 36$

P-550

Q3.16

Sol:- To find red color cars 4 comparison required.

To find green car 4 comparison required

(because Cars Relation has 4 rows)

Above comparison satisfy C.color = "red" & C.color = "green"

I- For red color car

C.color = "red" —— 4 comp. cid (102, 104)

To satisfy R.cid = C.cid

Reservation relation has 10 tuples and to find red cars each tuple need to be compared with cid's 102 & 104.

Hence total comp = $10 \times 2 = 20$.

Here we get {22, 22, 31, 31, 64} 5 # did's

II- For green color car

C.color = "green" —— 4 comp (103) cid

R.cid = C.cid —— $10 \times 1 = 10$ comp.

Here we get {22, 31, 74} 3 did's

Now we need to perform Intersect opn

we will take distinct dids in each set

{22, 31, 64} ∩ {22, 31, 74}

To comparison = 1+2+3 = 6

Hence total comparison required while executing SQL query

~~4+4+4+20~~ (4 + 20) + (4 + 10) + 6 =

finally we get did → 22, 31

Now we need to find 22, & 31 in Drivers relation

for 22 ⟶ 1 comparison

for 31 ⟶ 3 comparison

Total comparison

(4 + 20) + (4 + 10) + (1+2+3) + (1+3) = 48

find did who reserves red cars; find did who reserves green car; perform Intersect; find did in Drivers

Ans: Range of n is 44 — 48

~~Q5.17~~ Q 4.19 P - 565

· $\{ e.name \mid employee(e) \land$
$(\forall x) [\neg employee(x) \lor x.supervisorName \neq e.name$
$\lor x.sex = \text{"male"}]$

(OR is commutative)

$\{ e.name \mid employee(e) \mid$
$(\forall x) [\neg (employee(x) \land x.sex \neq \text{"male"}) \lor$
$SupervisorName \neq e.name]$

$\{ e.name \mid employee(e) \mid$
$(\forall x) [(employee(x) \land x.sex = \text{"Female"}) \Rightarrow x.SupervisorName \neq e.name]$

Means retrieve those e.name, who is not supervisor of any female employees

ie retrieve those e.name, with no immediate female subordinate.

it can be read as, retrieve e.name from employee relation such that $\forall$ x' $x \in$ Employee

ie for every tuple that belongs to employee, and sex is female., employee "e" is not supervisor of her

Q 4.21 P - 565

Sol: $R = \{a, b, c\}$ $S = \{c\}$

I. $\Pi_{R-S}(r) - \Pi_{R-S}(\Pi_{R-S}(r) \times S - \Pi_{R-S,S}(r))$

II. $\{t \mid t \in \Pi_{R-S}(r) \land \forall u \in S (\exists v \in r (u = v[S]$
$\land t = v[R-S]))\}$

means select tuple $t \in r$ ~~from~~ such that for all tuples u in S, there exists a tuple v in r such that $u = v[S] \land t = v[R-S]$

I, II both correspond to R/S.

P - ~~6~~565
Q 4.23

Suppliers (sid, sname, city, street)

$\{ \underline{sid} \rightarrow sname \quad \underline{sid} \rightarrow street \}$
$\underline{sid} \rightarrow city$
~~street~~
$(\underline{sname, city}) \rightarrow sid$
$(\underline{sname, city}) \rightarrow street$

non-trivial FD's

Hence given relation is in BCNF

given each supplier and each street within a city has unique name means, (Sname, city) — C.K

(Street, City) — C.K

Sid — P.K (given in question).

Hence for any FD, LHS is a super key.

Hence given Schema is in BCNF

Q given Relation $R(A_1 A_2 A_3 \ldots A_{15})$

with $(A_1, A_2, A_3, A_4, A_5, A_6)$ are simple Candidate keys

# of super keys?

Sol. # of sup keys with n ~~single~~ att simple Candidate keys

$2^n - 1 \Rightarrow 2^6 - 1 = 63$

non prime attributes = $15 - 6 = 9$

~~Each~~ # of superkeys = 63×2^9

MT1 (2017) <u>Q55</u>

given two Transactions $T_1 T_2$. if no ~~8~~ non-serial schedule is conflict serializable to the ~~given~~ $T_1 T_2$ serial Schedule and $T_2 T_1$ serial. Then this schedule also fails 2PL locking.

Q find # of Conflict Equivalent Schedule to given schedule

S_1: $W_1(A) \mid \underbrace{W_1(B) \mid r_2(A)}_{r_2(A)\ 2\text{ ways}} \mid W_2(B) \mid r_3(A)\ W_3(B)$

$r_3(A)$ 4 ways for each of these 2 ways — total $2 \times 4 = 8$

these write opn must execute in this order only, their relative position must not change.

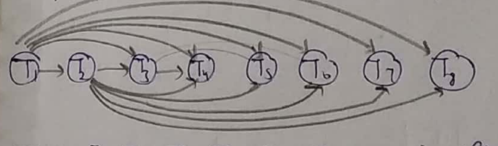
MT4 (2018)

Q

S: $W_1(x)\, r_2(x)\, W_3(x)\, r_4(x)\, W_5(x)\, r_6(x)\, W_7(x)\, r_8(x)$

# of serial schedule view equal to schedule (given) but not conflict equal to given schedule.

T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8
W(x)							
	r(x)						
		W(x)					
			r(x)				
				W(x)			
					r(x)		
						W(x)	
							r(x)



$T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_8$ ~~are~~ is Conflict equal to given

For View Equal

$\underbrace{W_1(x)\ r_2(x)}, \underbrace{W_3(x)\ r_4(x)}, \underbrace{W_5(x)\ r_6(x)}, \underbrace{W_7(x)\ r_8(x)}$

Here final write is by $W_7(x)$ & update read final is by $\underline{W_7(x)\ r_8(x)}$, Hence they must execute last

Other update read pairs can arrange in 3! ways

~~$W_1(x)\ r_2$~~

total $3! = 6$ View equal schedule

# of view equal but not conflict equal $= 6 - 1 = 5$