185545

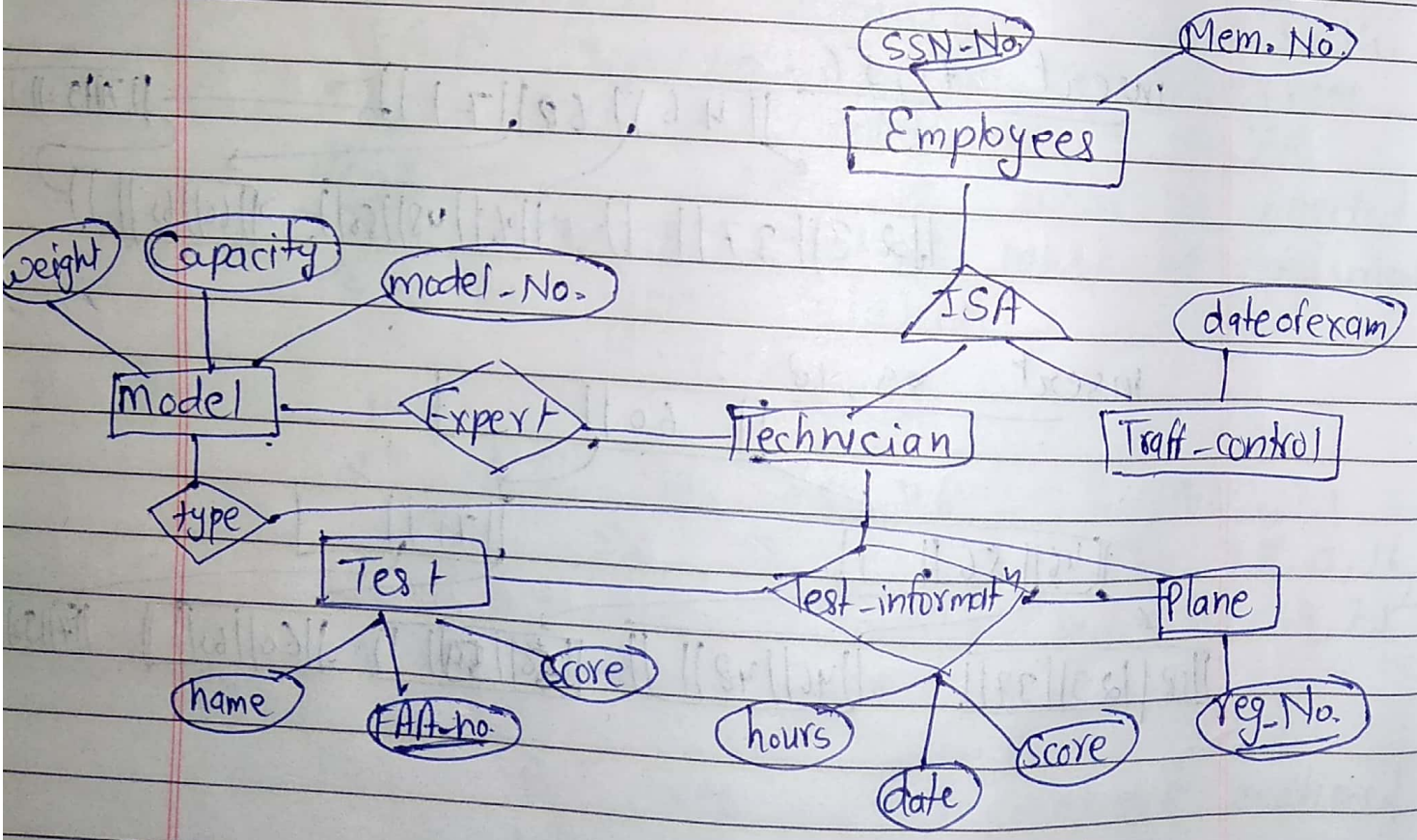
(1)

Date / /
Page No.

Name - Neetu Kumawat
Branch - CSE - DD
Roll No. - 185545
Subject - DBMS
Subject Code - CSD - 313
Semester - 5th

Ans (1) E-R diagram for airport Database :-

Registration no. = Reg-no.
Specific model = model
model-number = model-No.



Scanned with CamScanner

In this ER diagram every employee has there SSN_No. & Union_member no. SSN_No. is unique for every employee. so SSN_No. is primary key in Employee table.

↳ Every plane has their reg_no. as primary key.

↳ Every model has their weight & capacity & model No. here model_no. will be unique for every model. so model_no. will be my primary key.

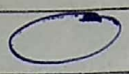
↳ In test_information table; it will have attibutes like exam hours, date & Score.

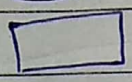
↳ here ISA (is a relationship) The upper entity type connected to the apex of the ISA Triangle is the more general entity type (Supertype) from which the lower entities inherit its attibutes.

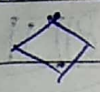
Scanned with CamScanner

185545
Date / /
Page No.

In the ER - Diagram :-

 → it shows attributes of entities

 → it shows the entities

 → it shows relationship between 2 or more entities

↳ underlined attributes is the primary Key.

The FAA passes a regulation that tests on a plane must be conducted by a technician who is expert on that model. We can not note the expert technician constraint the FAA requires in an ER diagram. There is no notation in an ER diagram & this is what is needed:- The expert relation must be equivalent to the type relation.

Ans (3) Items :-

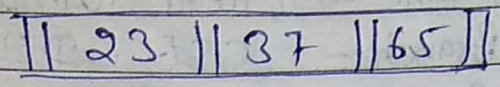
23, 65, 37, 60, 46, 92, 48, 71, 56, 59, 18, 21, 10, 74, 78, 15, 16, 20, 24, 28, 39, 43, 47, 40, 69, 75, 8, 49, 33, 30

(a) B^+ tree for above mentioned data :-

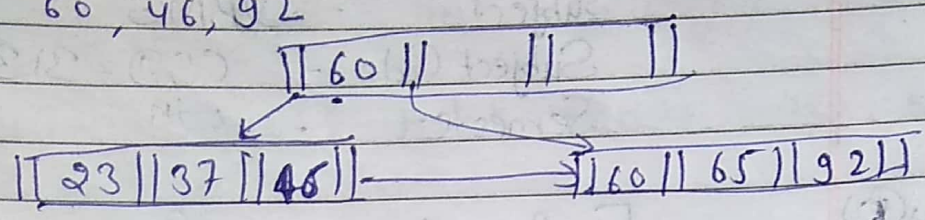
185545

Date / /
Page No.

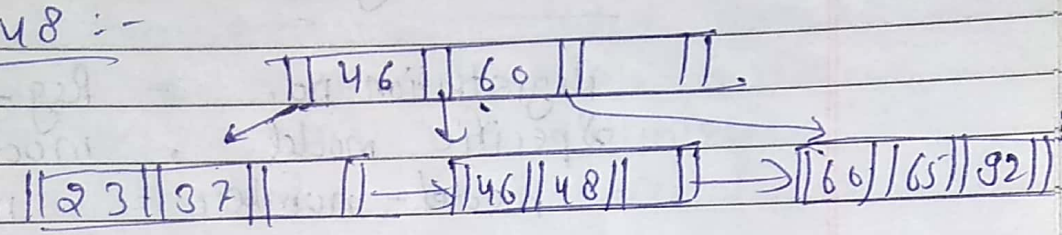
insert 23, 65, 37



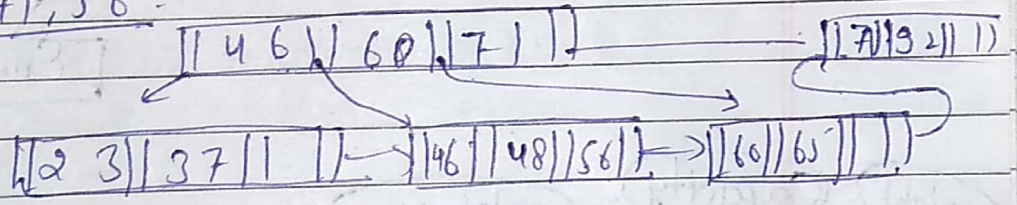
insert 60, 46, 92



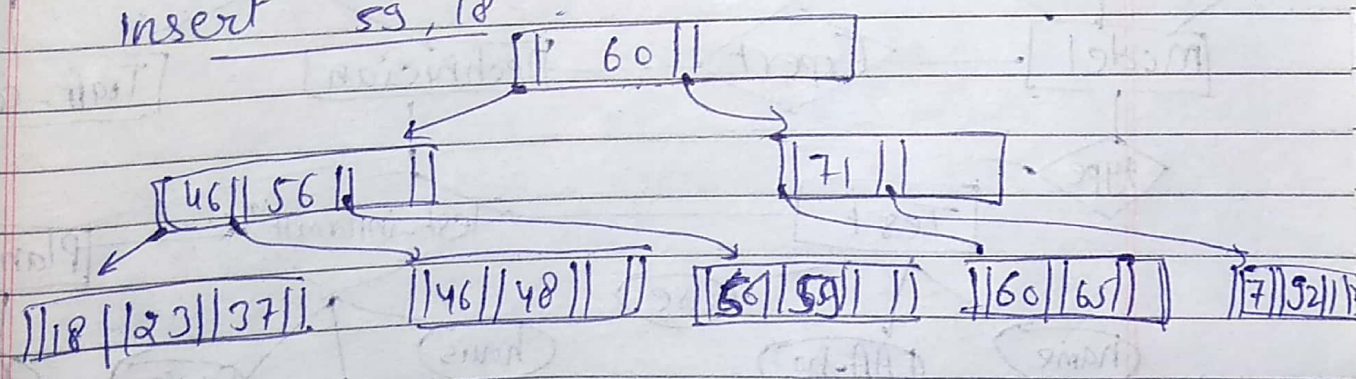
insert 48 :-



insert 71, 56 :-



insert 59, 18 :

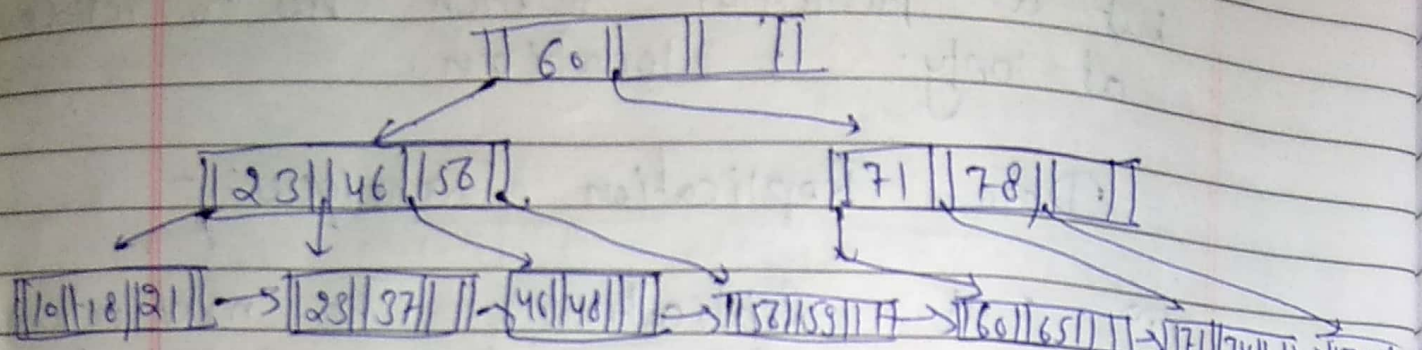


185545

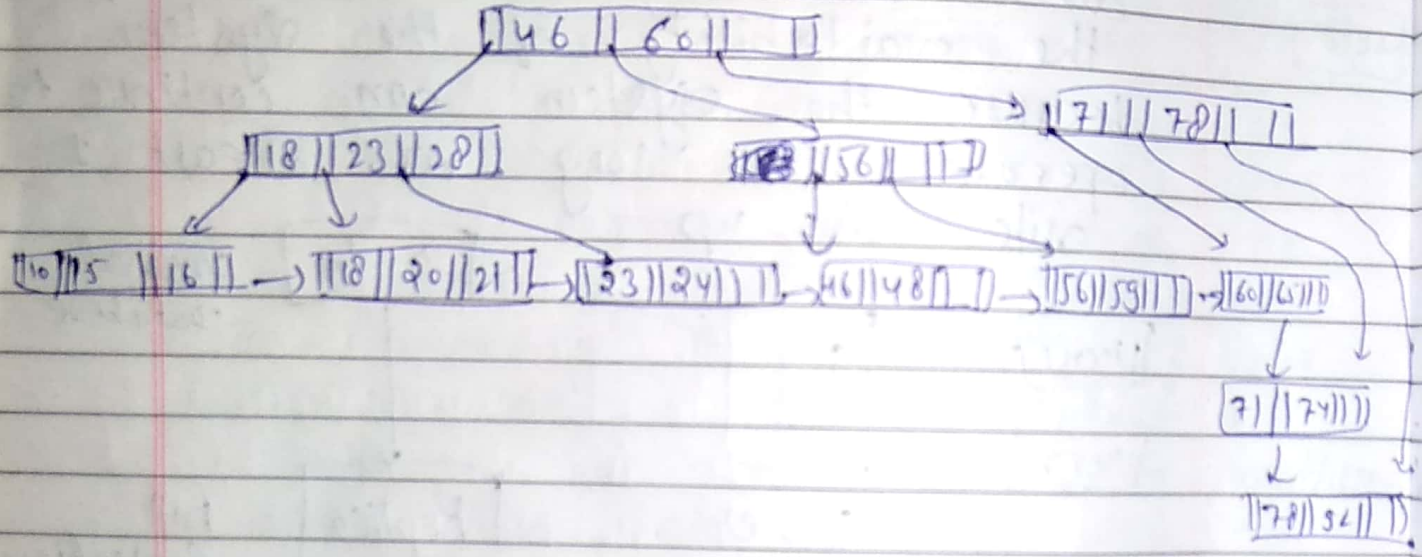
Date / /

Page No.

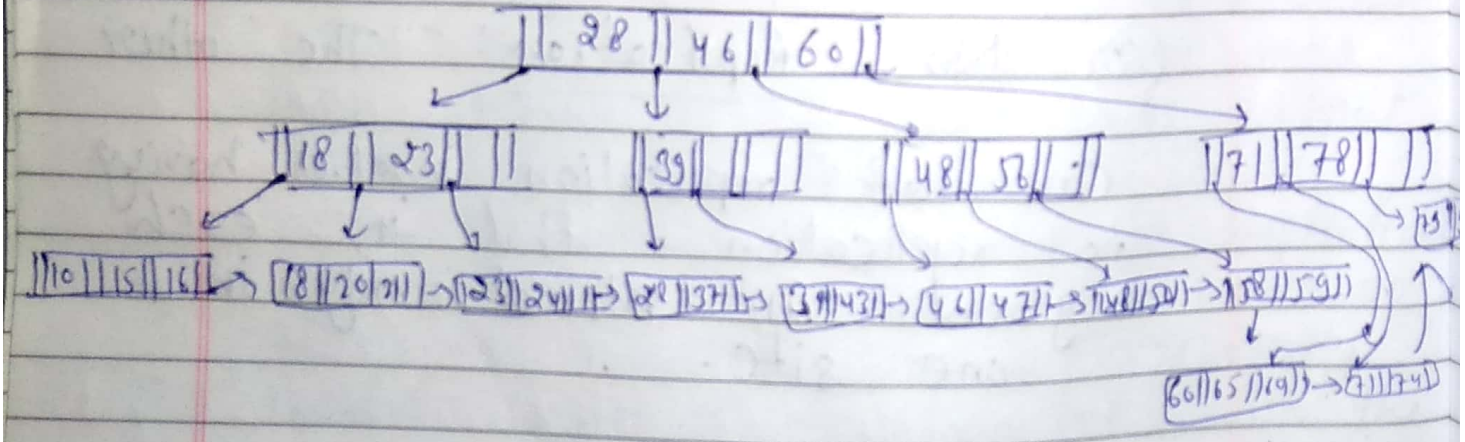
insert 21, 10, 74, 78 :-



insert 15, 16, 20, 24, 28, 39 :-

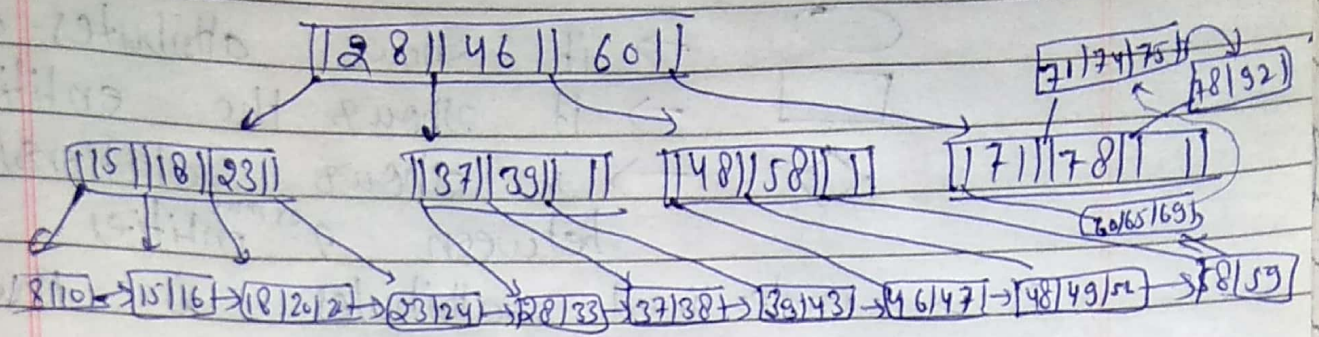


insert 43, 47, 50, 69, 75 :-



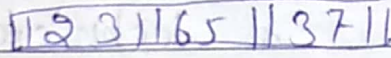
Date 18/5/25
Page No.

after insert 8, 49, 33, 38

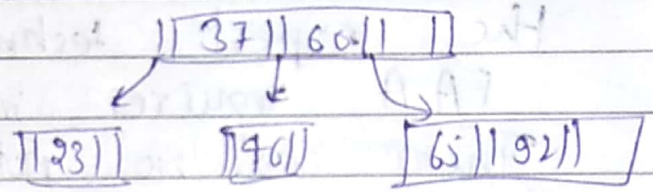


(b) B tree with order 4.

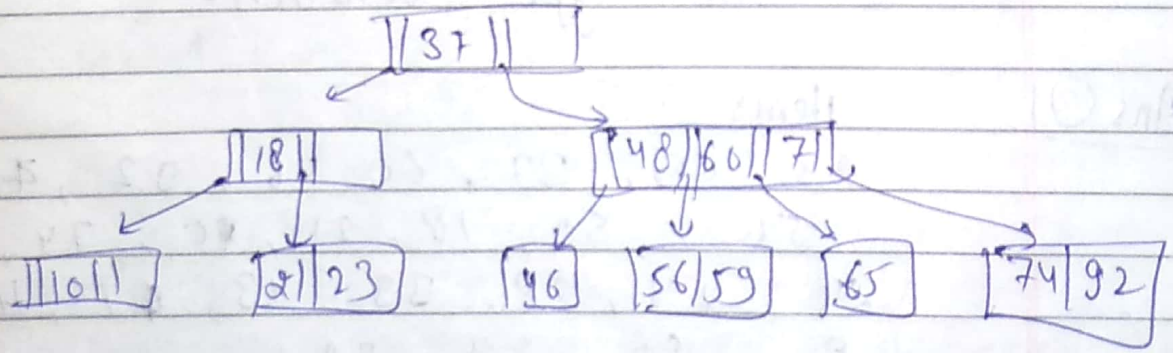
insert 23, 65, 37



insert 60, 46, 92



insert 18, 21, 10, 74 :-



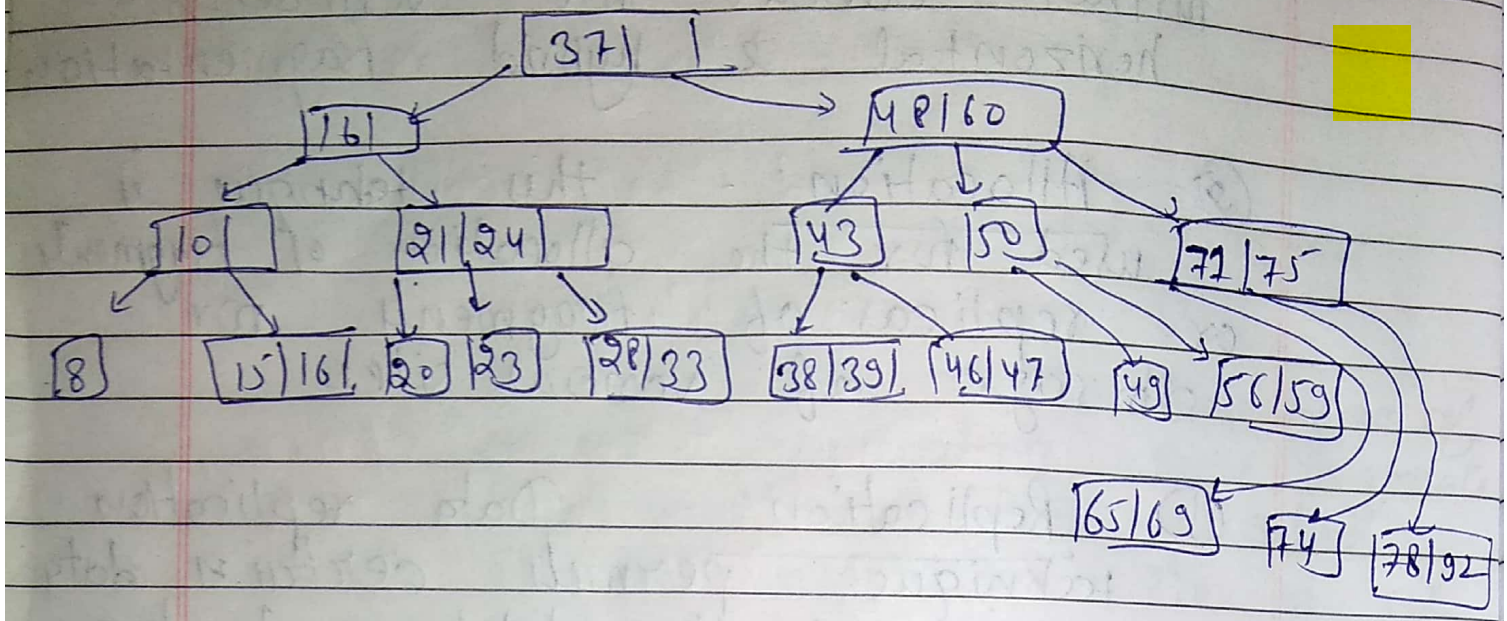
185545

Date / /

Page No.

(7)

insert 47, 50, 69, 75, 8, 49, 33, 78 :-



Ans (6) Data Replication :- it is the process of storing data in more than one site or node. it is useful in improving the availability of data. it is simply copying data from a database from one server to another server so that all the users can share the same data without any inconsistancy. The result is a distributed DataBase in which users can access data relavant to their tasks without interfering with the work of others.

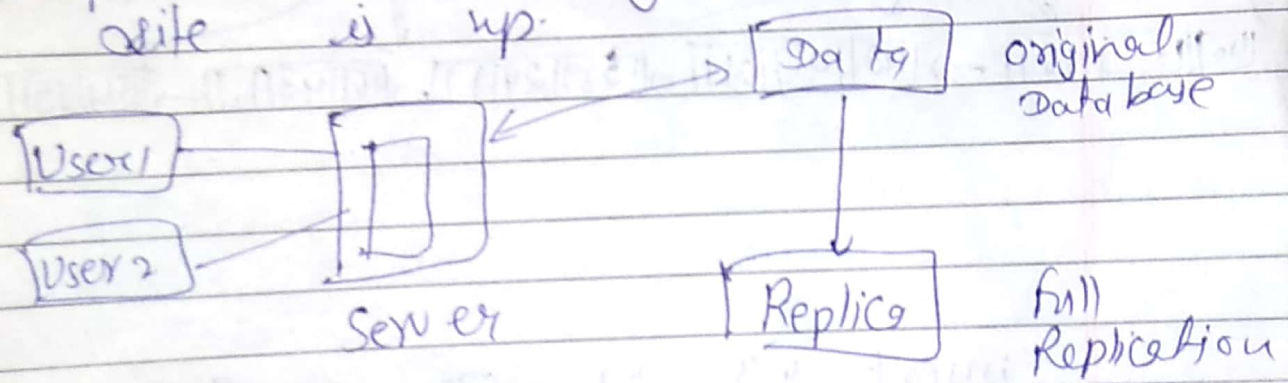
Data replication encompasses duplication of transaction on an ongoing basis, so that the replicate in a consistently updated state & synchronized with the source. However in data replication

Date / /
Page No.

data is available at different locations, but a particular relation has to reside at only 1 location.

Data Replication Techniques:-

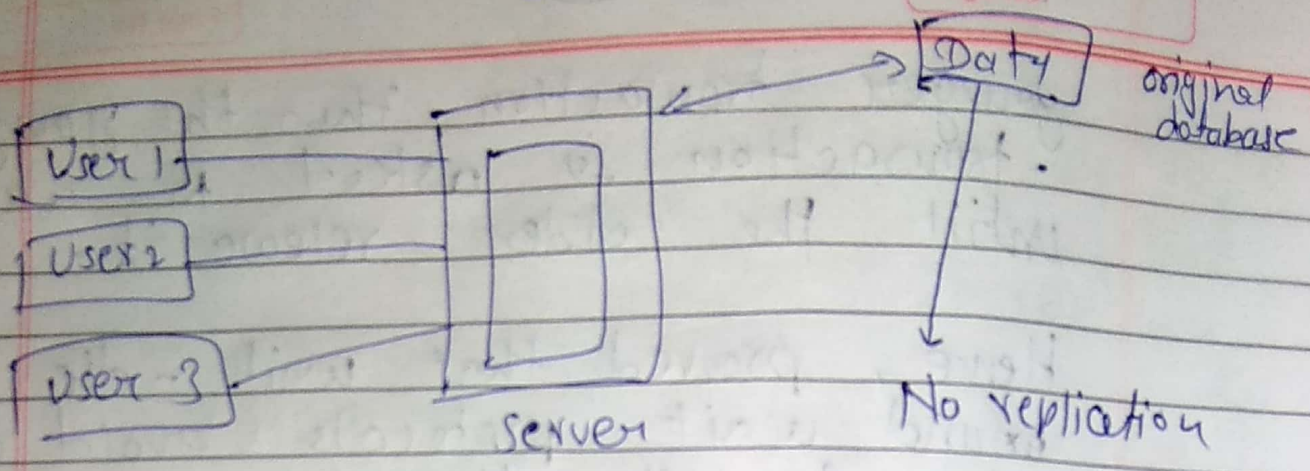
① **Full Replication:-** The most extreme case is replication of the whole database at every site in the distributed system. This will improve the availability of the system because the system can continue to operate as long as atleast 1 site is up.



② **No Replication:-** The other case of replication involves having no replication - that is, each fragment is stored only in one site.

Scanned with CamScanner

Date / /
Page No.



(2) Partial Replication :- In this type of ~~partial~~ replication some fragments of the database may be replicated whereas other may not. The no. of copies of the fragment may range from one to the total no. of sites in the distributed DBMS.

Data allocation in Distributed Database:-

Regardless of offering so many benefits over centralized database system the allocation of data is one of the key design issue of a distributed DB system

Techniques :-

(1) Fragmentation :- Technique that is used to break up the database into logical units is known as fragmention. fragmentation can be assigned for storage at different sites

The fragmentation techniques can be further divided into vertical, horizontal & hybrid fragmentation.

(2) Allocation:- this technique is used for the allocation of fragments or replicas of fragments for storage of various sites.

(3) Replication:- Data replication technique permits certain data or copies of the database to be stored in more than one site.

Wait - Die Scheme:- in this scheme, if a transaction requests for a resource which is already ~~defined~~ held with a conflicting lock by another transaction then the DBMS simply checks the timestamp of both transactions. It allows the older transaction to wait untill the resource is available for execution.

let's assume we have two Transactions T_i & T_j. TS is time stamp

(1) Check if $TS(T_i) < TS(T_j)$
→ if T_i is older transaction

& T_j has held some resource, then T_j is allowed to wait until the data available for execution. That means if the older transaction is waiting for a resource which is locked by the younger transaction, then the older transaction is allowed to wait for resource until it is available.

2) check if $TS(T_i) < TS(T_j) \rightarrow$ if T_i is older transaction & has hold some resource & if T_j is waiting for it, then T_j is killed & restarted later with the random delay but with the same timestamp.

Wound - wait protocol :- in wound wait protocol scheme, if the older transaction requests for a resource which is held by the younger transaction, then older transaction forces younger one to kill the transaction & release the resource. After the minute delay, the younger transaction is restarted but with the same timestamp.

if the older transaction has held a resource which is requested by the

younger transaction, then the younger transaction is asked to wait untill the older release it.

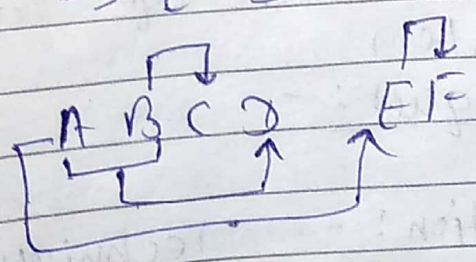
Hence, prooved that wait-die & wound wait protocols avoid dead-lock & starvation.

Ans(2) (Trans-no., Item no., price, sold-quantity, Seller, District-seller) is replaced with A, B, C, D, E, F respectively.

So relation scheme => (A, B, C, D, E, F)

Function Dependency => $(A, B) \rightarrow D$
$B \rightarrow C$
$E \rightarrow F$
$A \rightarrow E$

$B \rightarrow C$
$A \rightarrow E$ } Partial dependency



Candidate Key => (AB)
because AB) closure
$(AB)^+ = ABCDEF$

we can see that $B \to C$, $A \to E$ are the partial dependency because (AB) is a candidate key. So this table is not in 2nd NF Form.

↳ So the given relation is in ~~form~~ 1st NF.

Now, we will convert this into 2NF.

R (ABCDEF)

↳ R_1 (ABD)
↳ R_2 (BC)
↳ R_3 (AEF)

So transformed relation: -

R_1 (Trans-No, Item-No, sold-quantity)
R_2 (Item-No, price)
R_3 (Trans-no, District-seller)

Ans (5) (1) R_1(P), R_2(T), R1(T), R3(P), R3(Q); W1(P), C1, W3(Q), C3; R2(Q), W2(T), W2(Q), C2

Scanned with CamScanner

185545

Date / /
Page No.

T_1	T_2	T_3
$R_1(P)$		
	$R_2(T)$	
$R_1(T)$		
		$R_3(P)$
		$R_3(Q)$
$W_1(P)$		
C_1		
		$W_3(Q)$
		C_3
	$R_2(Q)$	
	$W_2(T)$	
	$W_2(Q)$	
	C_2	

it is recoverable, cascadless & strict ~~so~~ recoverable because first of all T_1 commits & the T then T_3 commits & Then T_2 commits.

The commits order is same as they are performing write operation on various datatype. firstly T_3 writes the data Q values in the database (∵ commits T3 before T2).

So the T_2 transaction will read the actual value of Q (from database).

(2)

T1	T2	T3
R1(P)		
	R2(T)	
R1(T)		
		R3(P)
		R3(Q)
W1(P)		
		W3(Q)
	R2(Q)	
	W2(T)	
	W2(Q)	
C1		
	C2	
		C3

In this dirty read is exists beweem W3(Q) & R2(Q) because transaction T3 makes some changes on data item (Q) but it is not commited on that time & our transaction T2 read that item & perform write operation on Q data item so the changes that are made by T3 would be lost.

So my given schedule is not recoverable because T2 commits before T3.

Scanned with CamScanner

185545

Date / /
Page No.

(3)

T_1	T_2	T_3
R1 (P)		
	R2(T)	
		R3 (P)
R1(T)		
	R2 (Q)	
		R3 (Q)
W1 (P)		
C1	W2 (T)	
		W3(Q)
W2(Q)		C3
	C2	

There is no dirty read or no phantom read problem. The given schedule is recoverable & cascadless schedule because They are cascadelling the data items but it is not strict schedule because b/w W3 (Q) & W2 (Q) after committing the T1 is performing some operation on Q data item & that data item is reading the value from the buffer not from actual database.

Ans (4) (1)

T1	T2	T3
R1 (y)		
		R3(y)
W1(Y)		
	R2(Y)	
		W3(Y)

The given schedule

Ans (4) let's there be 3 transactions T1, T2, T3. They are executed concurently & produce a schedule S.

S is serializable if it can be reproduced as at least one serial schedule (T1 → T2 → T3 or T1 → T3 → T2 or T2 → T1 → T3 or T2 → T3 → T1 or T3 → T1 → T2 or T3 → T2 → T1)

(a) This schedule is not serializable because T1 reads Y (r1 (X)) before T3 but T3 read Y. (r3(y)) before T1 writes Y (w1(Y)), wher Y is common data item. The operation r2 (X) of T2 does not affect the schedule at all so its position in the schedule is irrelavent. In a serial schedule T1, T2, T3 the operation w1(Y) comes after r3 (X). which does not happen in the gus

(b) This schedule is not serializable because T1 reads Z (r1(Z)) before T3 but T3 writes Z (w3(Z)) before T1 writes Z (w1(Z)). The operation r2(Z) of T2 does not affect the schedule at all so its position in the schedule is irrelevant. In a serial schedule T1, T3, & T2 r3(Z) & w3(Z) must come after w1(Z); which does not happen in the ques.

(c) This schedule is serializable because all conflicting operations of T3 happens before all conflicting operations of T3 happens before all conflicting operation of T1. T2 has only one operation, which is a read on Y (r2(Y)), which does not ~~affect~~ conflict with any other operation. Thus serializable schedule is:
r2(Y); r3(Y); w3(Y); r1(Y); w1(Y) (eg. T2 → T3 → T1) is a serial schedule.

d) This is not serializable schedule because T3 reads z (r3(z)) before T1 reads z (r1(z)) but r1(z) happens before T3 writes z (w3(z)). In a serial schedule T3, T2, & T1, r1(z) will happen after w3(z), which does not happen in the ques.

Scanned with CamScanner