Normalization

Date 12/10/2019

Expt. No. _____

(1) It is used to reduce data redundancy (having the same data but at different places.

(2) There are basically two problems arises due to redundancy 1st one is increases the size of the database, as the same data is repeated in many places.
2nd is inconsistency problem arises during insert, delete and update operations.

1NF - If a database contains composite or multivalued attributes, it violates 1NF. A relation is in 1NF if every attribute in that relation is single valued attri.

2NF - A relation must be in 1NF and relation must not contains any partial dependency.
A relation is in 2NF iff it has no partial dependency. i.e. no nonprime attribute (attribute which are not part of any candidate key) is dependent on any proper subset of any candidate key of the table.

3NF - A relation is in 3NF there is no transitive dependency, for nonprime attribute is it in 2NF.
A relation is in 3NF iff at least one of the follow- condition holds in every nontrivial functional depen.

$X \rightarrow Y$

(i) X is superkey
(ii) Y is prime attribute (i.e. each element of Y is part of some candidate key).

Teacher's Signature _____

Example: Consider the relation R(A, B, C, D, E)

$A \rightarrow BC$, $CD \rightarrow E$, $B \rightarrow D$, $E \rightarrow A$

All possible candidate keys in the above relation are {A, E, CD, BC} all attribute are in right side of functional dependency are prime.

Transitive dependency :→ Suppose we have a relation Table: Student (stud_no, stud_name, stude_phone, stud_state, stud_country, stud_age).

FD are - ① {stud_no → stud_name}

② {stud_no → stud_state}

③ {stud_state → stud_country}

④ {stud_no → stud_age}

Candidate Key: {stud_no}

for the above relation -

{stud_no → stud_state} and {stud_state → Stud_Country} are true. So {stud_country} is transitively dependent on {stud_no}. It violates the 3NF.

To convert it in 3NF, we will decompose the relation

Table1 Student {stud_no, stud_name, stud_phone, stud_age}

Table2 State_Country {state, country}.

Date

Expt. No.

Page No.

Boyce-Codd Normal Form (BCNF)

A relation is in BCNF if R is in 3NF and for every functional dependency, LHS is superkey.
A relation is in BCNF iff in every non trivial FD, $X \rightarrow Y$; X is a superkey.

Example :- Find the highest normal form of a R(A,B,C,D,E) with FD set as { $BC \rightarrow D$, $AC \rightarrow BE$, $B \rightarrow E$ }

Step1 - As we can see $(AC)+ = \{A, C, B, E, D\}$ but none of its subset can determine all the attribute of relation. So AC will be the candidate key. A and C can't be derived from any of the attribute of the relation. so there will be only one candidate key {AC}.

Step2 - Prime attribute are those attribute which are part of candidate key {A, C} in this example and other are non prime attribute {B, D, E}.

Step3 - The relation R is in 1NF as it doesnot allow multivalued or composite attribute.
The relation R is 2NF because $BC \rightarrow D$ is in 2NF (BC is not a proper subset of candidate key AC)
$AC \rightarrow BE$ is in 2NF (AC is candidate key).
$B \rightarrow E$ is in 2NF (B is not a proper subset of AC)

Teacher's Signature

The relation is not in 3NF because in $BC \rightarrow D$ (neither BC is superkey nor D is a prime attribute) in $B \rightarrow E$ (neither B is a superkey nor E is prime attr.) But to satisfy 3NF, either LHS of a FD should be superkey or RHS should be prime attribute).

So, the highest N.F. of above relation will be 2NF.

1. BCNF is free from redundancy.
2. If a relation is in BCNF, then 3NF also satisfied.
3. If all attribute of a relation are prime attribute then the relation is always in 3NF.
4. A relation in a RDBMS is always atleast in 1NF.
5. ~~A relation~~ Every Binary relation (A relation with only two attributes) is always in BCNF.
6. If a relation has only singleton candidate keys (i.e. every candidate key consists of only one attribute) then the relation is always in 2NF (because no partial dependency possible.
7. Sometimes going for BCNF form may not presesve FD, in that case go for BCNF only if the lost FD may not required, else normalize till 3NF only.

Date

Expt. No.

Page No.

Q. Find the highest NF in relation under R(A,B,C,D,E) following FDs - ABC → D, CD → AE.

Sol: ① Always checking from BCNF then 3NF and so on.
② If any FD satisfies a NF then there is no need to check for lower NF. ABC → D is in BCNF (Because ABC is superkey) so no need to check this dependency for lower normal form.

* Functional Dependency

FD is a constraints between two sets of attributes in a relation from a DB. A FD is denoted by (→) if an attribute A functionally determines B, then it is written as A → B.
for example - empid → name means empid functionally determines the name of employee.

A → B means for all instances of a particular value of A there is the same value of B.

* Trivial functional dependency

X → Y is trivial only when Y is subset of X.
ABC → AB ; ABC → A ; ABC → ABC.

Non Trivial FD - X → Y is a nontrivial FD when Y is not subset of X.

Teacher's Signature

* Some Important points on FD.

① For FD we find attribute closure using FD set F.
② If B is subset of A+, then A→B is true.
③ Candidate key is minimal superkey.
④ Attributes which are part of any candidate key of relation are called prime attribute.

Q.1 Consider a relation schema R=(A, B, C, D, E, H) on which the following FD holds -

{ A→B, BC→D, E→C, D→A, }

What are the candidate key of R?

Sol? $(AE)^+ = \{A, B, C, D, E\}$ which is not a set of all attribute. SO AE is not a candidate key.

$(AEH)^+ = \{A, B, C, D, E, H\}$ $(BEH)^+ = \{A, B, C, D, E, H\}$

$(BCH)^+ = \{B, C, H, D, A\}$ which is not set of all attribute.

SO, Ans is {AEH, BEH, DEH}.

Dependency Preserving in DBMS Decompotition

The dependency preserving decompotition is another property of decomposed relational DB. schema D in which each functional dependency X→Y specified in F either appeared directly in one of the relational schemas Ri in the decomposed D or could

be inferred from the dependencies that appear in some R_i.

A decomposition $D = \{R1, R2, R3, \ldots\ldots Rn\}$ of R is dependency preserving w.r.t. a set F of FD $(F1 \cup F2 \cup F3 \cup \ldots\ldots fm)^+ = F+$.

Consider a Relation R, $R \dashrightarrow F$ {with some FD} R is decomposed of R1 with FD {f1} and R2 with FD {f2}; there can be three cases:

(i) $f1 \cup f2 = f$; decomposition is dependency preserving.
(ii) $f1 \cup f2$ is a subset of f; Not dependency preserving.
(iii) $f1 \cup f2$ is a superset of f; this case is not possible.

Q. Let a relation R(A,B,C) and a functional D. {AB → C, C → D, D → A}. Suppose relation R is decomposed into R1(A,B,C) & R2(C,D). Check whether a decomposition is dependency preserving or not?

Sol) Let us find the closure of f1 and f2.

closure of f1

closure of (A) = {A} Trivial
closure of (B) = {B} Trivial
closure of (C) = {C, A}
But D can't be closure as D is not present in R1.
= {C, A}
C ---> A Removing C from right side as it is trivial attribute

Teacher's Signature

Closure of (AB) = {A, B, C, D}

AB ---> C = {A, B, C} (Incomplete)

Q.1 Let R(A,B,C,D) be a relational schema with the following FDs. A → B, B → C, C → D & D → B. The decomposition of R into (A,B), (B,C), (B,D). Gives lossless join & is dependency preserving

Ans

Q.2 Let a relation R(A,B,C,D) and FDs { AB → C, C → D, D → A } Let R is decomposed into R1(A,B,C) & R2(C,D). Check whether the decomposition is dependency presesving or not?

Sol^n. since f1 ∪ f2 ⊂ f so, it is not.

Canonical Cover

Whenever a user updates the DB, the system must check whether any of the FD are getting violated in this process. If there is a violation of dependencies in the new DB state the system must roll_back.

Working with a huge set of FD can cause unnecessary added computational time. This is where the canonical cover comes into play.

A canonical cover of a set of FD f is a simplified set of FD that has a same closure as the original set F.

Extraneous attribute :→ An attribute of a FD is said to be extraneous, if we can remove it without changing the closure of the set of FDs.

Date

Expt. No.

Page No.

A canonical cover F_c of a set of FDs F such that all the following properties are satisfied -

(i) F logically implies all dependencies in F_c.
(ii) F_c logically implies all dependencies in F.
(iii) No FD in F_c contains an extraneous attribute.
(iv) Each left side of FD in F_c is unique i.e. there are no two depencies.

$\alpha_1 \to \beta_1$ & $\alpha_2 \to \beta_2$ in such that $\alpha_1 \to \alpha_2$.

Example - Consider the following set of FD F -

$F = \{ A \to BC,\ AB \to C,\ B \to C,\ A \to B \}$.

steps to find the canonical cover.

Sol?- 1. There are two FD with the same set of attribute on the left hand side i.e. $A \to BC$, $A \to B$ these two can be combined to get $A \to BC$.
Now, the combined set becomes

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

2. There is an extraneous attribute in $AB \to C$ because even after removing $AB \to C$ from the set F we get the same closure. This is because $B \to C$ is already a part of F.
Now F becomes

$F = \{ A \to BC,\ B \to C \}$

Teacher's Signature

3. C is an extraneous attribute in $A \to BC$ also $A \to B$ is logically implied by $A \to B$ & $B \to C$ (by transitivity) so, $F = \{ A \to B, B \to C \}$

4. After this step, F doesnot change anymore hence the required Canonical Cover is

$F_C = \{ A \to B, B \to C \}$ Ans:

Equivalence of FD

For understanding equivalence of FD sets basic idea about attribute closure is -

Given a relation with different FD sets for that relation, we have to find out whether one FD set is subset of other or both are equal

→ **How to find relationship between two FD sets :-**

Let FD1 & FD2 are two FD sets in a relation R,

(i) If all FDs of FD1 can be derived from FD present in FD_2, we can say that $FD1 \supseteq FD_2$

(ii) If all the FDs of FD_2 can be derived from FD present in FD1, we can say that $FD2 \supseteq FD1$

(iii) If both are true, then $FD1 = FD2$

Concurrency Control

Date 15/10/2019

Expt. No. _____

Serializability

Page No. _____

If any transaction (non serial) produce an outcome which is equal to the outcome of that schedule's transaction executed serially, then we can call that transaction schedule, a serializable schedule.

(i) serial schedule have less resource utilization and low throughput. To improve this we use concurrency.
(ii) Concurrency of transaction may lead to inconsistency in database.
(iii) To avoid this, we need to check whether these concurrent schedules are serializable or not.

These are two types { view serializable, conflict serializable.

Conflict Serializable.

A schedule is called conflict, if it can be transformed into a serial schedule by swapping non conflicting operations.

Two operations are said to be conflicting if all conditions satisfy :-

(i) They belongs to different transactions.
(ii) They operate on same data.
(iii) Atleast one of them is a write operation.

Ex: conflicting operations pair $[R_1(A), W_1(A)]$ because they belongs to two different transaction on same data A and one of them is write.

Teacher's Signature _____

Some Important points

→ To check for conflict serializable, we need to make precedence Graph. If the graph contains the cycle, then its not conflict serializable. Hence their is no cycle the schedule is conflict serializable.

→ Now to check for recoverable, we need to check for a dirty read operations (Write by T_i transactions followed by T_j read operation before T_i Commit), between any pair of operations. If no dirty read operation the recoverable schedule.

→ If there is no interleaving of operation the conflict

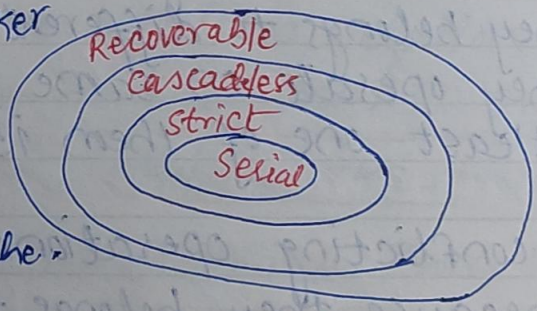
T_1 T_2
W

Q. Which of the following scenarios may lead to an irrecoverable error in a database system.

Ans A transaction read a data item after it is written by an uncommitted transaction.

(i) Cascadele schedule is stricker than recoverable schedule.

(ii) Strict schedule is striker or subset of cascadeless sche.



(iii) Serial schedule satisfy constrainsts of all schedule recoverable, cascadeless and strict schedule & hence it is a subset of strict schedule.

Expt. No. _____ Date _____

Page No. _____

→ There are basically two reasons for allowing concurrency
(i) Improved throughput and resource utilization.
(ii) Reduced waiting time also reduce avg. response time.

The motivation for concurrent execution in DB is same as using multiprogramming in o/s.

Example 1: consider the following schedule

T1	T2	T3
R(A)		
	W(A)	
	commit.	
W(A)		
		W(A)
commit.		commit.

Ans. (i) first of all it is view serializable as view equal to serial schedule T1 → T2 → T3.
which satisfies the initial read & updated read and final write on variable A.

(ii) Now there is a write-write pair done by transaction T1 & T3, which is violating the above mention conditions of strict schedule.

So, The given schedule is serializable but not strict recoverable.

Two-phase Locking Protocol (2PL)

A simple lock based protocol does not guarantee serializability (Also known as binary lock).

To guarantee serializability, 2-phase locking is used that concern the positioning of locking and unlocking operations in every transactions.

"2-phase locking ensures serializability"

A transaction is said to follow 2PL protocol if locking & unlocking can be done in 2-Phase -

(i) Growing phase – New locks on data items may be acquired, but none can be released.

(ii) Shrinking phase – Existing locks may be released but no new locks can be acquired.

[S(a)] → shared lock ; X(a) → Exclusive locks.

* If lock conversion is allowed then upgrading of lock from S(a) to X(a) is allowed in growing phase.

And

* Downgrading of lock from X(a) to S(a) must be done in shrinking phase.

* Strict 2PL – It ensures that the schedule is recoverable and cascadeless but deadlock is possible.

* Rigorous 2PL – Lock being 2Phase, all exclusive(X) and shared(S) locks. It is more restricted than strict 2PL.

Example

→ strict 2PL.
→ Rigourous 2PL

Expt. No. _____ Page No. _____

	T1	T2
1	Lock-S(A)	—
2	—	Lock-S(A)
3.	Lock-X(B)	
4	—	—
5	unlock(A)	—
6	—	Lock-X(C)
7	unlock(B)	—
8	—	unlock(A)
9	—	unlock(C)
10		

Transaction T1 :→ Growing phase from 1-3.
Shrinking phase from 5-7.
Lockpoint at 3.

Transaction T2 :→ Growing phase from 2-6.
Shrinking phase from 8-9.
Lockpoint at 6.

Lockpoint → The point at which growing phase ends.

Drawbacks of 2-Phase Locking protocol

(i) Cascading rollback is possible under 2PL.
(ii) Deadlocks and starvation is possible.

Teacher's Signature _____

Timestamp ordering Protocol

(1) It is used to order the transaction based on their timestamps. The order of transaction is nothing but the ascending order of the transaction creation.

(2) The priority of the older transaction is higher that's why it executes first. To determine the timestamp of the transaction, this protocol uses system time or logical counter.

(3) The lock based protocol is used to manage the order between conflicting pairs among transactions at the execution time.

(4) Timestamp based protocol starts working as soon as transaction is created.

(5) It ensures conflict serializable & free from deadlock.

(6) One Drawback of basic TO is that it cascading rollback is possible. Also may not be recoverable.

Granularity :→ It is the size of the data items allowed to lock. Now multiple granularity means hierarchical breaking up the database into ~~database~~ blocks which can be locked and track.

multiple choice Questions

Date 16/10/19

Expt. No. _____ Page No. _____

(1) Rigorous 2PL protocol permits releasing all locks at the end of transaction.

(2) A scheme that creates a new version of a data item for each transaction is defined by multiversion concurrency control scheme.

(3) A 2phase locking protocol variant that requires all locks to be held until transaction commit is called Rigorous 2-phase locking protocol.

(4) Deadlock prevention scheme that requires each transactions to locks all data items before it begins execution.

(5) Deadlock prevention scheme named wound wait is a preemptive technique.

(6) A transaction that is inserting a new tuple into DB is given an. Exclusive lock.

(7) multiple granularity locking protocol utes multiple lock modes to ensure serializability.

(8) cascading rollback can be avoided by strict 2PL Protocol.

(9) A process known to receive msges from transaction and in response send msg is - Lock manager.

(10) In a preemptive technique, transactions are assigned a unique timestamp to decide whether it should wait or rollback.

(11) Snapshot isolation is a scheme of multiversion concurrency control.

(12) Two modes of locking data items, are termed as shared & Exclusive.

Teacher's Signature _____

(13) A set of rules applied over a transaction that may lock & unlock each of data items in DB, is known to be Locking protocol.

(14) A protocol that permits a transaction to lock a new data items only if it has not yet unlocked any data items is called 2PL protocol.

(15) Exclusive locks are released at the end of transaction to ensure - Recoverability & cascadelessness.

(16) For dealing with not prevented deadlocks system construct wait-for graph.

(17) Snapshot isolation scheme doesnot support Serializability.

(18) No. of actions need to be taken in order to recover from deadlock is/are - 3.

(19) A way of guarding data items against deadlock prevention is to - order data items.

(20) To ensure deadlock prevention, we can use preemptive approach and Rollbacks.

(21) A write operation in multiversion timestamp ordering may result into the - Rollback of transaction.

(22) A protocol that requires each transactions to be executed in 2 or 3 diff phases in its lifetime is called validation protocol.

(23) ~~multiversion 2PL protocol requires transactions to be~~ declared as Readonly & updates.

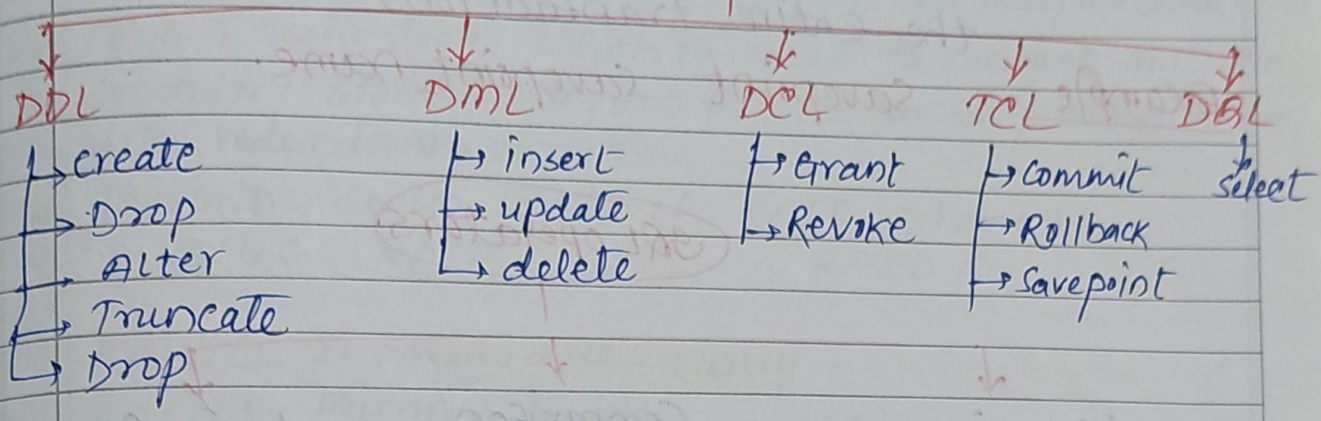
(24) Deadlock prevention scheme named wait-Die is a non preemptive scheme.

Date

Expt. No.

SQL commands

Page No.



(1) DDL (Data Definition Language)

→ It changes the structure of the table like creating a table, deleting a table, altering a table etc.

→ It is auto-committed i.e. permanently save all the changes in the database.

Truncate - It is used to delete all the rows from the table & free the space containing the table.

Ex: Truncate table employee;

(2) DML (Data manipulation Language).

→ It is used to modify the DB.

→ It is not auto-committed that means it can't permanently save all the changes in DB. They can be rollback.

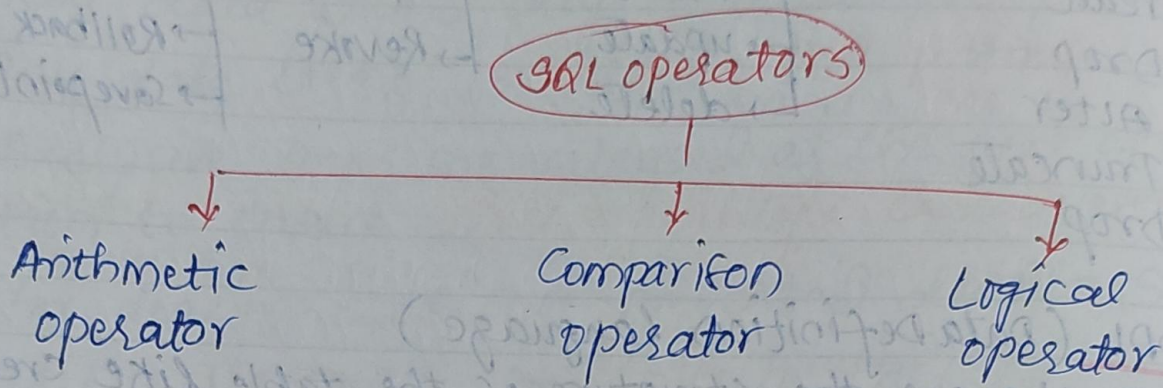
(3) TCL (Transaction Control Language)

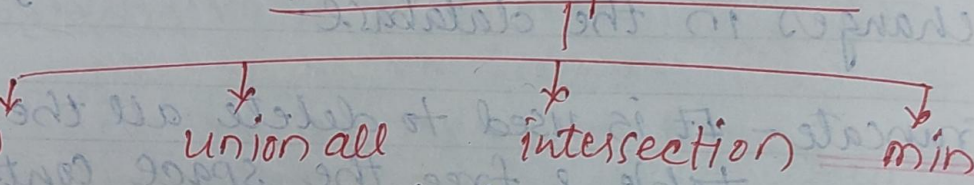
Teacher's Signature

It is only used with DML like insert, delete etc.

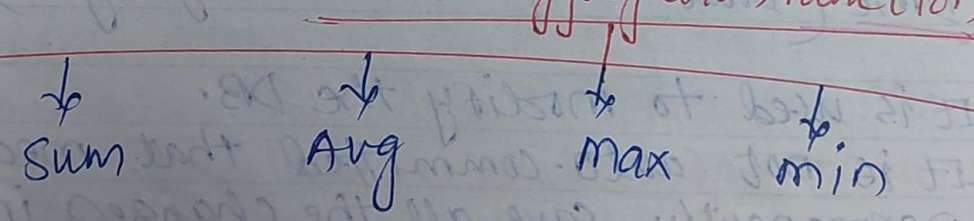
Savepoint – It is used to roll the transactions back to a certain points without rolling back the entire transactions.

Example Savepoint savepoint_name.





→ minus operators are used to display the rows which are present in the 1st query but absent in the second query.



→ count function is used to count no. of rows in a DB table. It works on both numeric & non numeric datatypes.

→ Count(*) returns all the rows also considers duplicates (Null).

Kendriya Vidhyalaya 2019

* Data stripping → data can place across multiple disks. If one disk fails, then all data in the array lost

Expt. No. _____ **RAID** Page No. _____

→ It refers to Redundancy array of the independent Disk.
→ It is a technology which is used to connect multiple secondary storage devices for increased performance, data redundancy or both.
→ There are seven levels of RAID schemes —
RAID 0, RAID 1, ... , RAID 6.

RAID 0	It provides data stripping.
RAID 1	Mirroring of data
RAID 2	bit level stripping using hamming code parity
RAID 3	Byte level stripping with dedicated parity.
RAID 4	Block level stripping with parity disk.
RAID 5	slight modification of RAID 4, distributed parity.
RAID 6	

Indexing

- ordered indices
- primary index
  - Dense index
  - sparse index.
- clustering index
- Secondary index

→ Indexing is used to optimize the performance of the DB by minimizing the no. of disk access of the data in the DB table quickly.

Teacher's Signature

Dense index – It is primary index based on primary key. The dense index contains an index record for every search key value in the data file. It makes searching faster.

The no. of records in the index table is same as the no. of records in the main table. Therefore it needs more space to store index record itself.

The index record have searchkey and a data reference/pointer to the actual record in the disk.

Index record

search key	data reference/pointer

Sparse index – In the data file index record appears only for a few items. each item points to a block. The index points to the records in the main table in gap.

Expt. No. _____

B+ Tree

Page No. _____

(1) The B+ tree is a balanced binary search tree. It follows a multilevel index format.

(2) In this, leaf node denote actual data pointers. B+ tree ensures that all leaf node remain at the same height.

(3) Leaf nodes are linked using Link list. Therefore B+ tree can support random access as well as sequencial access.

(4) An internal node of B+ tree can contain atleast n/2 record pointers except the root node. At most, an internal node of the tree contains n pointers

(5) The leaf node of the B+ tree can contains atleast n/2 record pointers and n/2 key values.

Date 08/01/2020

View Sesializability

view equivalence

reads get the same view

Let t_1 & t_2 have two transactions.

1. x $r_1(x)$ $r_2(x)$: initial read
2. x $w_1(x)$ $w_2(x)$ final write.

3. x $T_1 \leftarrow T_2$ S

   $T_1 \leftarrow T_2$ S'

two schedules S, S'

view equivalence to same sesial schedule.

Ex: ① S: $r_1(a)$ $w_1(a)$ $r_2(a)$ $w_2(a)$ $r_1(b)$ $w_1(b)$ $r_2(b)$ $w_2(b)$.

T_1 T_2 a: $r_1(a)$ $w_1(a)$ $r_1(b)$ $w_1(b)$ $r_2(a)$ $w_2(a)$ $r_2(b)$ $w_2(b)$.

($r_1(a)$: initial; $r_1(b)$: initial; $w_2(a)$: final; $w_2(b)$: final)

that's why it is view sesializable.

Ex: ② S: $r_1(a)$ $w_2(a)$ $w_1(a)$ $w_3(a)$

T_1 T_2 T_3 : $r_1(a)$ $w_1(a)$ $w_2(a)$ $w_3(a)$

view sesializable.

× $T_2 T_1$: $r_2(a)$ $w_2(a)$ $r_2(b)$ $w_2(b)$ $r_1(a)$ $w_1(a)$ $r_1(b)$ $w_1(b)$

Date

Expt. No. Page No.

Ex: ③ S: $r_1(a)$ $w_2(a)$ $w_1(a)$

$T_1 T_2$: $r_1(a)$ $w_1(a)$ $w_2(a)$ ×

$T_2 T_1$: $w_2(a)$ $r_1(a)$ $w_1(a)$. ×

not view serializable

conflict Vs view serializable

Note: ① Every conflict serial. is also view serializable.
i.e. conflict serial. is ∉ striker than view serial.

② But Every view S. is not conflict serial.

③ Constrained write assumption × C.S. ≡ V.S.

$w_i(\phi(r(x)))$

unconstrained writes / Blind writes

Conflict Serializability.

I_i of T_i conflict I_j of T_j (logical temperal order.)

1. Access Same data item
2. at least one of them is a write operation.

conflict sesializability

S: conflict equivalence S'

Exa: S: $r_1(a)\ w_1(a)\ r_2(a)\ w_2(a)\ r_1(b)\ w_1(b)\ r_2(b)\ w_2(b)$

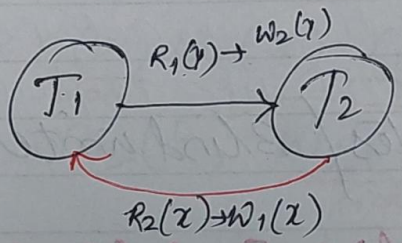
T1: T2; $r_1(a)\ w_1(a)\ r_1(b)\ w_1(b)\ r_2(a)\ w_2(a)\ r_2(b)\ w_2(b)$

→ Any view sesializable schedule that is not conflict sesializable, must contains a Blind write. "Blind write is simply when a transaction writes without reading".

→ Blind write happens in view sesializability.

Ex: ① $R_1(x)\ R_1(y)\ R_2(x)\ R_2(y)\ W_2(y)\ W_1(x)$

precedence Graph

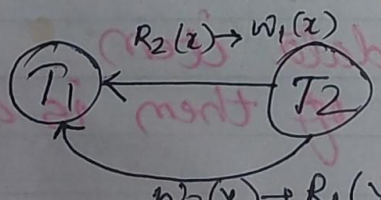


cyclic

Hence Non conflict Sesializable.

Ex: ② - $R_1(x)\ R_2(x)\ W_2(y)\ R_1(y)\ W_1(x)$

precedence Graph



Acyclic conflict Sesializable order T2 : T1

Kameshwari d2011994852 Kameshwari@12

Date 09/1/2020

Expt. No. Page No.

(4) Which normal formal is considered as adequate for usual DB design - 3NF

(1) A table on the many side of a one to many or many to many relationship must:

Ans: Have a composite key. The relation in 2NF is also in 1NF and no partial dependencies on any column in primary key.

(2) Table in 2NF - Eliminate all hidden dependencies.

(3) Which one of the following statements about normal forms is false :→ (a) BCNF is strictei than 3NF

(b) Lossless, dependency preserving decomposition into 3NF is always pissible.

(c) Lossless, dependency preseiving decomposition into BCNF is always possible.

(d) Any relation with two attributes is BCNF.

(4) which is bottom-up approach to db design that design by examining the relationship between attributes -
(a) FD (b) Database modeling (c) Normalization (d) Decomposition

(5) A table is in 2NF if it is in 1NF and if:
no column that is not a part of the primary key is dependent on only a portion of the primary key.

(6) A normal form on which every determinant is a key BCNF. A relation is in BCNF if ever determinant of the relation is a candidate key.
A relation is in BCNF if it is not in 4NF.

Teacher's Signature

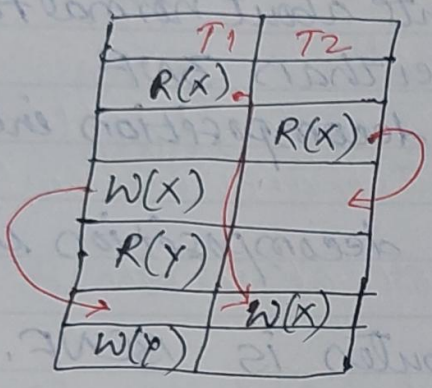
Precedence Graph

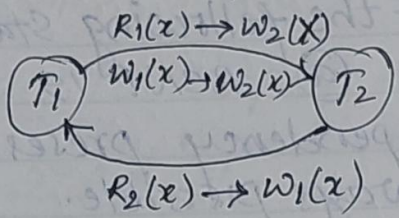
$R_1(A) \longrightarrow W_2(A)$ Arrow from $T_1 \rightarrow T_2$

$W_1(A) \longrightarrow R_2(A)$ Arrow from $T_1 \rightarrow T_2$

$W_1(A) \longrightarrow W_2(A)$ Arrow from $T_1 \rightarrow T_2$

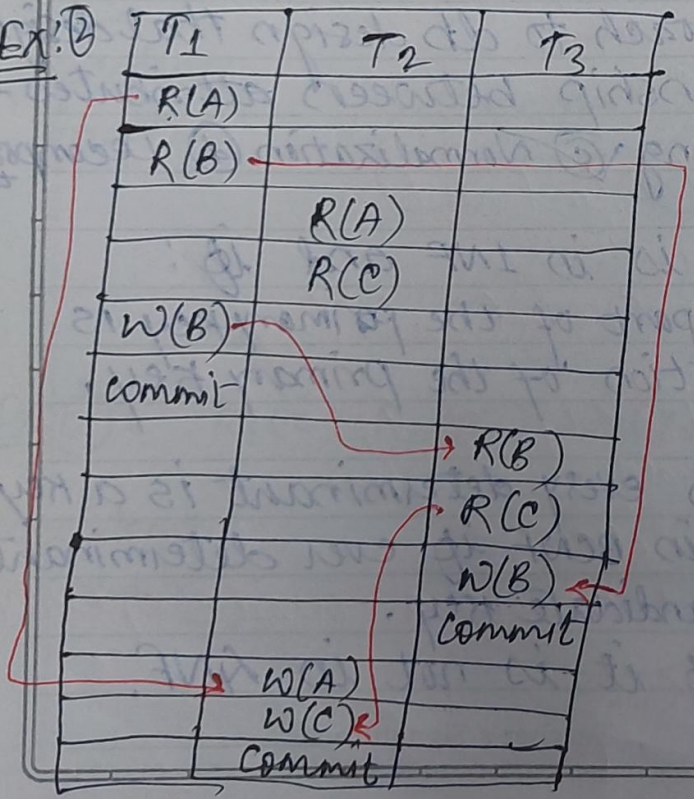
Ex: (A) $D = R_1(x)\ R_2(x)\ W_1(x)\ R_1(y)\ W_2(x)\ W_1(y)$

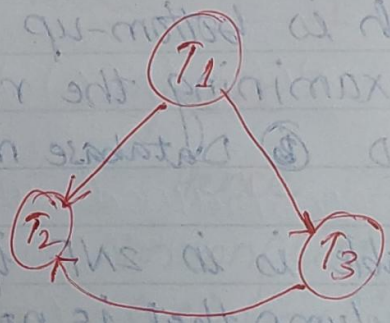




cyclic Hence Not Conflict Serializable.

Ex: (B)





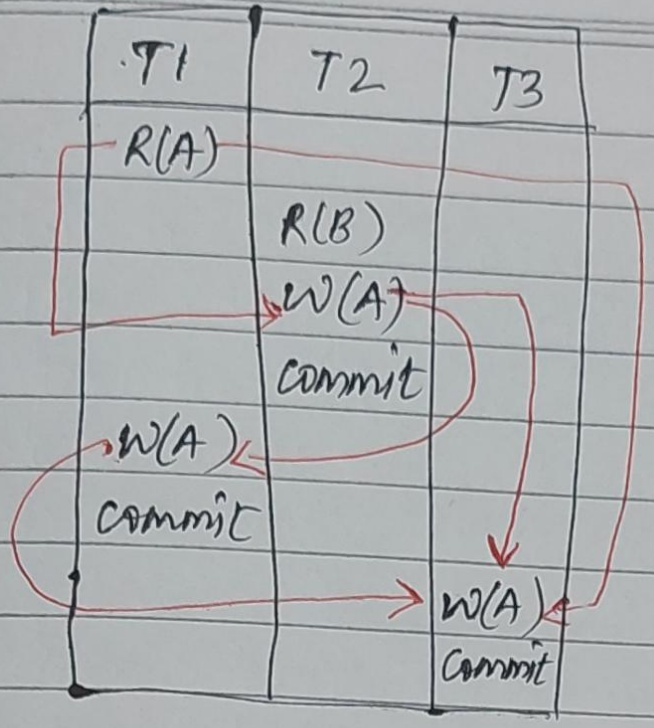
Acyclic then conflict Serializable.

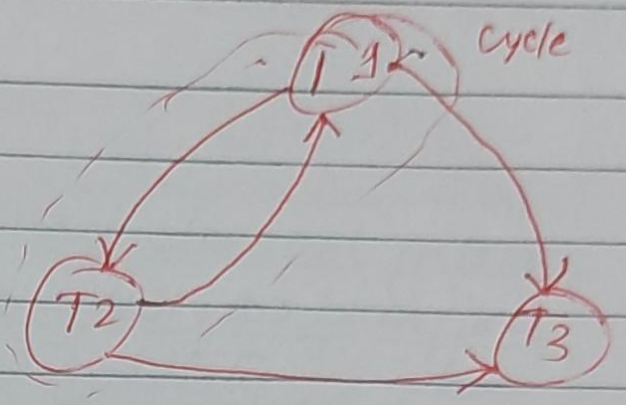
Date

Expt. No.

Page No.

Ex: (3)





cycle: Hence not conflict serializable.

Q. Consider the following two statements about transaction schedules:

I Strict 2-phase locking protocol generates conflict serializable schedules that are also recoverable.

II Timestamp ordering concurrency control protocol with Thomas "write rule can generate view serializable schedules that are not conflict serializable.

Ans. Both statements are true.