The Key Responsibilities of a Database Administrator

Software installation and Maintenance.

Data Extraction, Transformation, and Loading.

Specialised Data Handling.

Database Backup and Recovery.

Security.

Authentication.

Capacity Planning.

Performance Monitoring.

Data Dictionary

A data dictionary contains metadata i.e data about the database. The data dictionary is very important as it contains information such as what is in the database, who is allowed to access it, where is the database physically stored etc. The users of the database normally don't interact with the data dictionary, it is only handled by the database administrators.

The data dictionary in general contains information about the following −

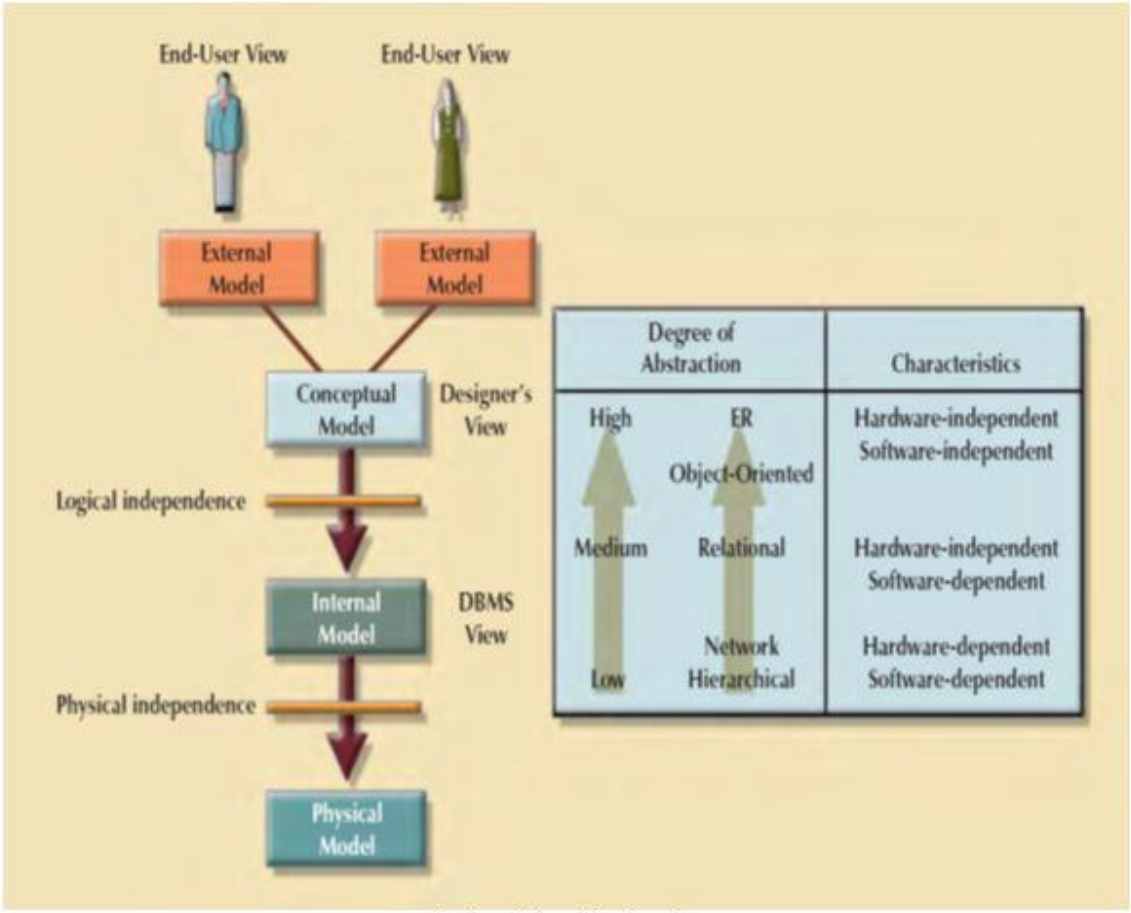
- Names of all the database tables and their schemas.
- Details about all the tables in the database, such as their owners, their security constraints, when they were created etc.
- Physical information about the tables such as where they are stored and how.
- Table constraints such as primary key attributes, foreign key information etc.
- Information about the database views that are visible.

Two types: Active Data Dictionary and Passive Data Dictionary.

Parameter	PRIMARY KEY	UNIQUE KEY
Basic	Used to serve as a unique identifier for each row in a table.	Uniquely determines a row which isn't primary key.
NULL value acceptance	Cannot accept NULL values.	Can accepts NULL values.
Number of keys that can be defined in the table	Only one primary key	More than one unique key
Index	Creates clustered index	Creates non-clustered index
Auto Increment	A Primary key supports auto increment value.	A unique key does not supports auto increment value.
Modification	We cannot change or delete values stored in primary keys.	We can change unique key values.

Candidate key is a collection of one or more unique attributes.

Super key should have at least one candidate key with other normal attributes.



Data abstraction levels

Data Abstraction Levels:

1. External Level
2. Conceptual Level.
3. Internal Level.

Data Model Level:

1. Conceptual Model.
2. Logical Model.
3. Physical Model.

Data Independence Types:

1. Logical Data Independence.
2. Physical Data Independence.

A relation is said to be in 1NF (first normal form), if it doesn't contain any multi-valued attribute. In other words you can say that a relation is in 1NF if each attribute contains only atomic(single) value only.

BCNF Example table: (Boyce Codd Normal Form)
Fid, faculty_name, c_id,course_name
=> fid, faculty_name ; => c_id, course_name => fid,c_id

4NF Example table:
Student_id, name, course;
=> student_id, name ; => student_id course.

DDL - Data Definition Language (CREATE, DROP, ALTER, TRUNCATE)

DML- Data Manipulation Language (INSERT, SELECT, DELETE, UPDATE)

DQL- Data Query Language (SELECT)

DCL- Data Control Language (GRANT, REVOKE)

DCL (Data Control Language):

DCL includes commands such as GRANT and REVOKE which mainly deal with the rights, permissions, and other controls of the database system.

List of DCL commands:

- GRANT: This command gives users access privileges to the database.
- REVOKE: This command withdraws the user's access privileges given by using the GRANT command.

Though many resources claim there to be another category of SQL clauses TCL – Transaction Control Language. So we will see in detail about TCL as well. TCL commands deal with the transaction within the database.

List of TCL commands:

- COMMIT: Commits a Transaction.
- ROLLBACK: Rollbacks a transaction in case of any error occurs.
- SAVEPOINT:Sets a savepoint within a transaction.
- SET TRANSACTION: Specify characteristics for the transaction.

Aggregate functions in SQL:

- COUNT()
- SUM()
- AVG()
- MIN()
- MAX()

Dense and Sparse Index =>

Dense Indexing And Sparse Indexing Explained in Hindi l DBMS Course

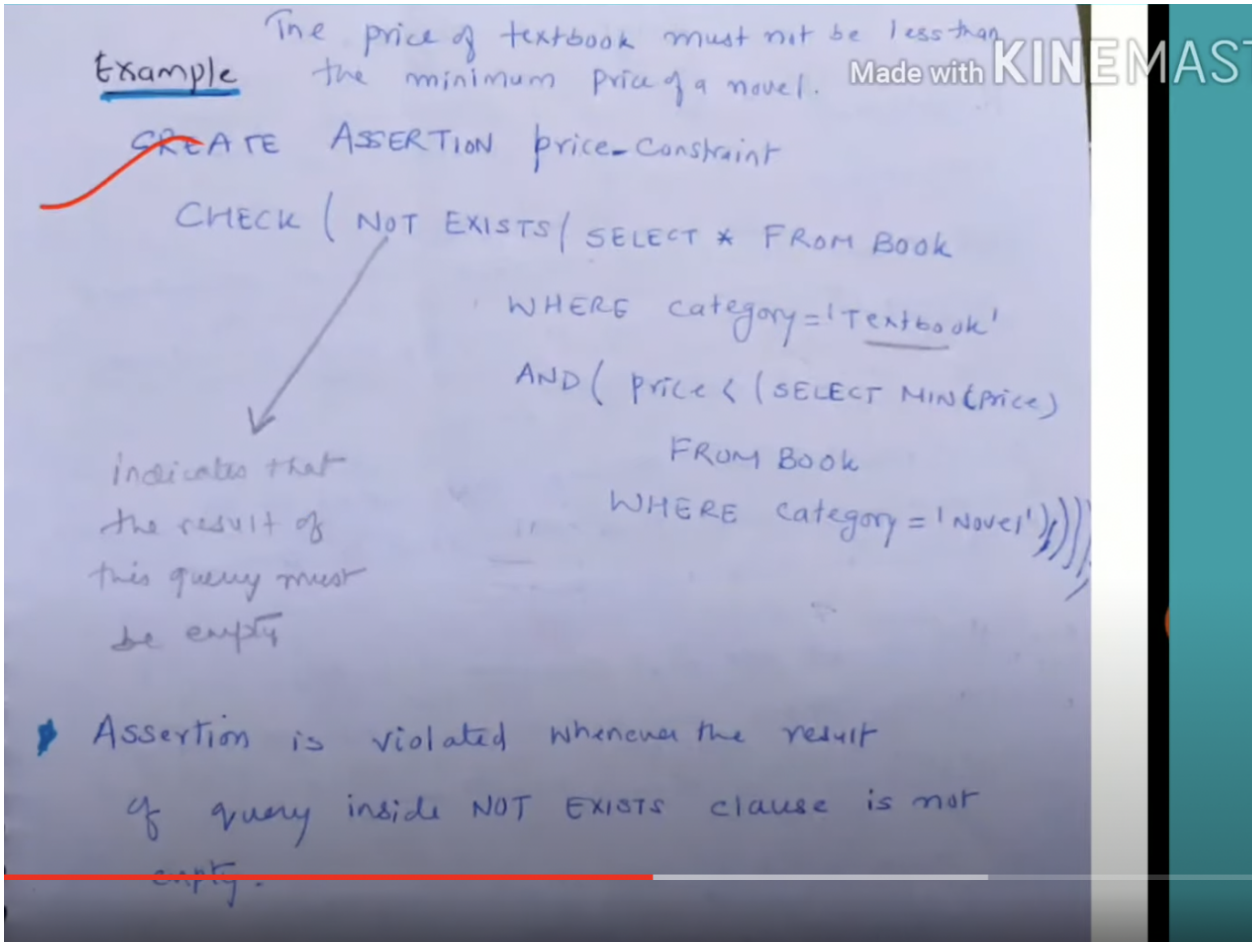
Assertions :

- Used to specify additional types of constraints that are outside the scope of the built-in relational model constraints (integrity constraints, entity constraint, referential constraints).
- Assertion is an expression or constraint that should always be true.
- DBMS always checks assertion whenever a modification corresponding with the table is created.

Syntax:

CREATE ASSERTION <assertion-name>
CHECK <condition>

<> or **!=** is Not equal to operator.



Triggers:

It is a procedure that starts automatically or gets triggered if specified changes/modifications occur in the database.

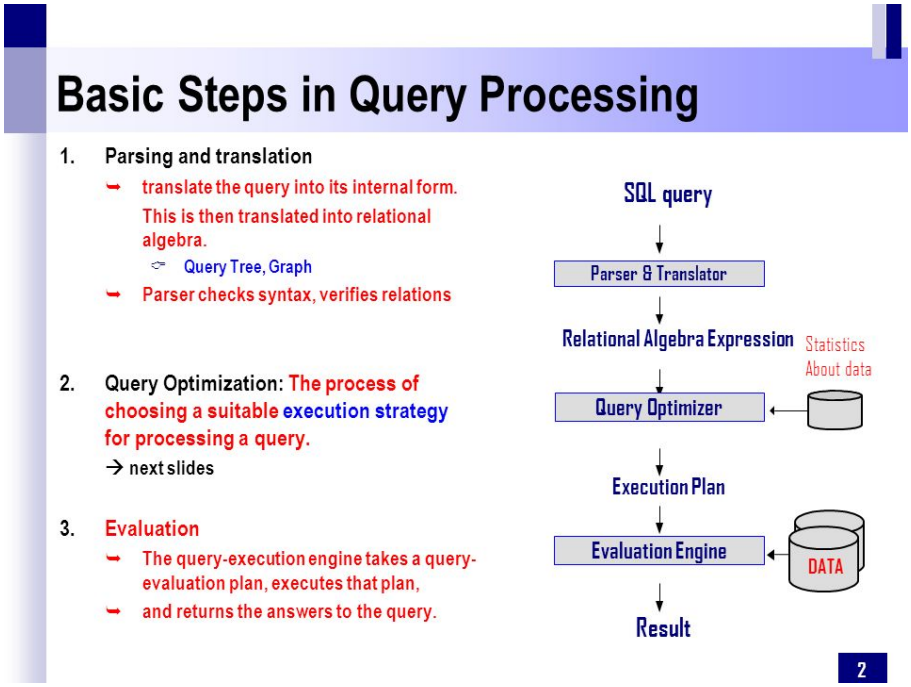
3 parts of db trigger:
Trigger event, trigger condition (optional), trigger action.
Events can be insert, update or delete.

Syntax:

CREATE TIGGER <trigger_name>
Before|After insert|update|delete ON <table_name>
For each row | For each Statement
When

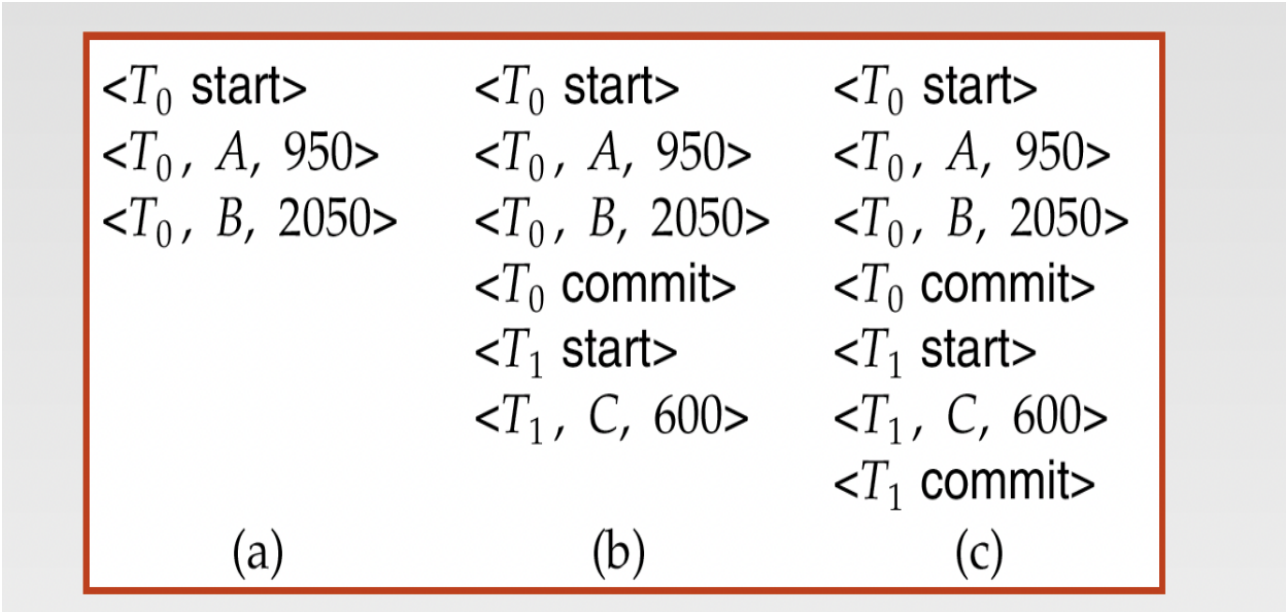
<condition>
Command.

Example:
When the data is added on the employee table, increase the salary by 100.
CREATE TRIGGER emp_sal_increase
After insert ON Employee
For each row
When
(employeeSalary>500)
UPDATE employee SET salary = salary+100;



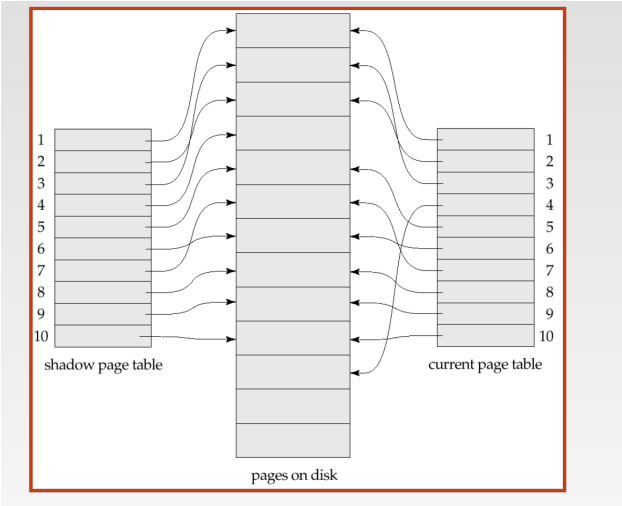
Crash Recovery;

1. Log based recovery
    a. Deferred database modification. (only redo)
    b. Immediate database modification. (has both undo and redo)



Deferred Method : a) Nothing b) To Redo and Nothing c) To redo and T1 redo.
Immediate Method: a) Undo b) To Redo and Undo T1 c) To redo and T1 redo.

2. Shadow Paging (https://www.youtube.com/watch?v=YA0sXVDoHig)
   - Alternative to log based recovery
   - Modified version of shadow copy recovery. In shadow copy recovery, entire database copy is made; whereas in shadow paging, only required parts copy are made
   - Has tables, shadow page table and current page table.
   -



Deadlock Avoidance:

Based on TimeStamps :

Wait-Die Scheme
In this scheme, if a transaction requests to lock a resource (data item), which is already held with a conflicting lock by another transaction, then one of the two possibilities may occur −

If TS(Ti) < TS(Tj) − that is Ti, which is requesting a conflicting lock, is older than Tj − then Ti is allowed to wait until the data-item is available.

If TS(Ti) > TS(tj) − that is Ti is younger than Tj − then Ti dies. Ti is restarted later with a random delay but with the same timestamp.

This scheme allows the older transaction to wait but kills the younger one.

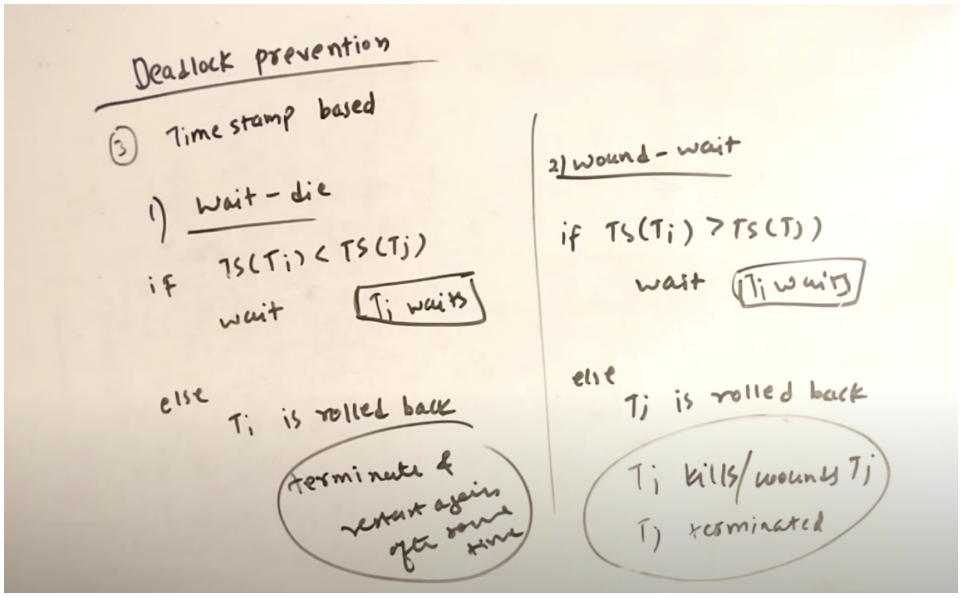
Wound-Wait Scheme
In this scheme, if a transaction requests to lock a resource (data item), which is already held with conflicting lock by some another transaction, one of the two possibilities may occur −

If TS(Ti) < TS(Tj), then Ti forces Tj to be rolled back − that is Ti wounds Tj. Tj is restarted later with a random delay but with the same timestamp.

If TS(Ti) > TS(Tj), then Ti is forced to wait until the resource is available.

This scheme allows the younger transaction to wait; but when an older transaction requests an item held by a younger one, the older transaction forces the younger one to abort and release the item.

In both the cases, the transaction that enters the system at a later stage is aborted.



Difference between serial schedule and serializable schedule

Serial schedule:

- The transaction is executed completely before the start of another transaction; this condition is termed as serial schedule.
- It is involved in a cycle in which the next transaction is started only upon completion of the former that cycles in a loop.

Serializable schedule :

- Serializable is used to find a non-serial type in existence, this process allows any transaction to execute concurrently without inferences and identifies if they are correct.
- Transaction execution has interleaving, meanwhile serial is always a serializable one.

Two phase commit Protocol

- There are three different types of commit protocols in the distributed system; They are :
  I. one phase commit protocol
  2. Two phase commit protocol
3. Three phase commit protocol.

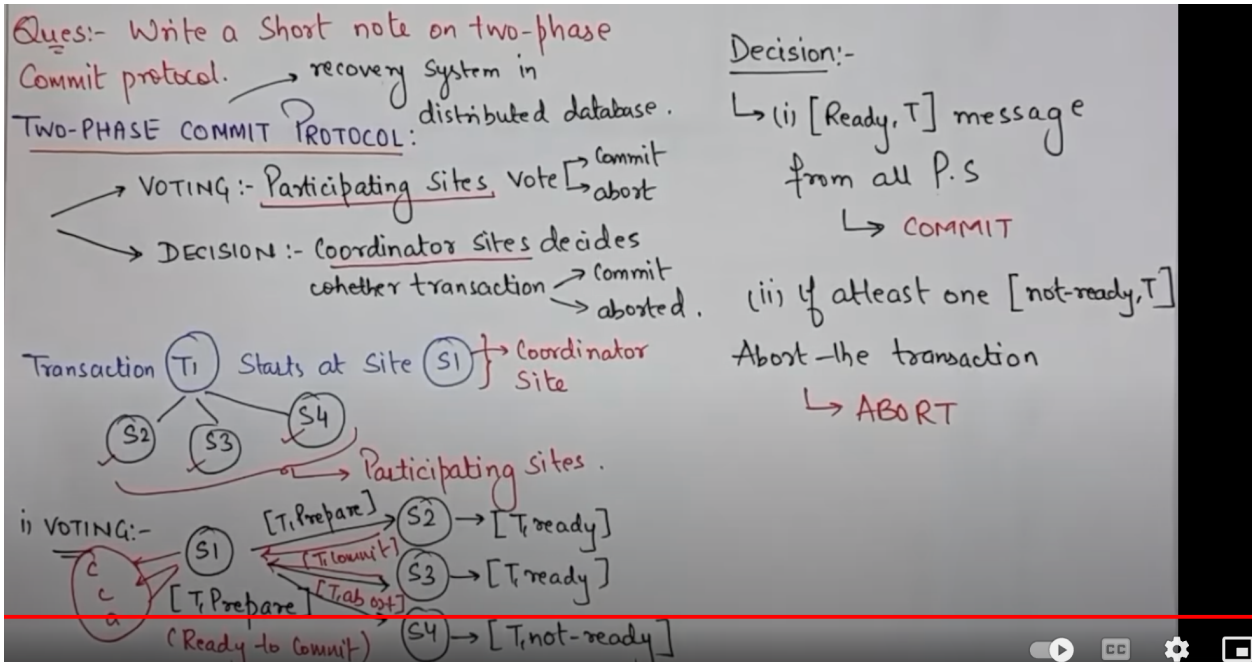
Two phase commit protocol has two phases;

1. Voting phase.
2. Decision Phase.

In this system, we make one coordinator; and other server in the system will be participant. In the system, where the transaction has been initialized is coordinator. (The coordinator is responsible for initializing the protocol).

Voting Phase :
In voting phase, the sends <T, prepare> or “is ready to commit ?” request to the participants.
The participants, based on the their state returns <T, commit> or <T, abort>.
If the participants is ready, <T,ready> is maintained in their log file; else <T, noready> is maintained.

Decision Phase:
In decision phase, the coordinator counts the number of votes from participants and gives decision.
If <T,ready> is from all participants, then a commit message is given to all participants server; and the Participants server, commits the transaction.
Else if any one the <t,abort> is received, then the abort signal is given to participants to transaction.


Ques:- Write a Short note on two-phase Commit protocol.
recovery system in distributed database.
TWO-PHASE COMMIT PROTOCOL:
VOTING :- Participating sites vote commit / abort
DECISION :- Coordinator sites decides cohether transaction commit / aborted.
Transaction T1 Starts at Site S1 Coordinator Site
S2
S3
S4
Participating sites.
i) VOTING:-
S1
[T1,Prepare]
[T1,commit]
[T1,abort]
[T1,Prepare]
(Ready to Commit)
S2 [T1,ready]
S3 [T1,ready]
S4 [T1,not-ready]
Decision:-
(i) [Ready, T] message from all P.S
COMMIT
(ii) If atleast one [not-ready,T]
Abort the transaction
ABORT