Deadlock prevention strateqies :-

① Deadlock prevention protocols ensure that the system will never enter into a deadlock state.

② Some prevention strategies are:-

i) locks all data before its execution.

ii) either all are locked in one step or none are locked.

iii) Two main disadv :-

a) It is hard to predict, before transaction begins, what data items need to be locked.

b) data item utilization may be very slow.

ii) Impose an ordering of all data item.

a) ——

b)

iii) Use preemption and transation roll-back.

i) when T2 is request lock othat hold by T1, T1 get rollback & granting the lock to T2. They use timestamp alg.

Wait-Die scheme

Wound-Wait scheme.

X—o—X

Deadlock prevention strategies:-

Deadlock prevention protocols ensure that the system does not go in deadlock state:-

① lock all data before its execution.

i)

ii) It is hard to predict, before transaction, which data need to be locked

② Use preemption & transation roll back → Wait-die
Wound-wait

③ Impose an ordering of all data items.

Q.3] explain Deadlock Detection & Recovery mechanism.

a] Deadlock detection:-

① aborting a transaction is not always a practical approach.

② deadlock avoidance mechanism can be used to detect any deadlock situation in advance.

③ method like wait-for graph are suitable only those system where transactions are lightweight having fewer instances.

④ In bulky system, deadlock prevention tech. may work well.

b] Deadlock Recovery:-

① The most common soln. is to roll back one or more transaction to break the deadlock.

② Three actions need to be taken:

1. Selection of victim:

i) Given a set of deadlocked transactions we must determine which transaction to roll back to break the deadlock.

We should roll back those transaction that will incur minimum cost.

2. Roll back:

i) once we determine that particular transaction must be rolled back, we must determine how far transaction should be rolled back.

ii) The simplest soln is total roll back:

a) abort the transaction & then restart it.

iii) partial rollback requires the system to maintain additional info. about state of all running transaction.

3] ~~Start~~ starvation:-

① In a system where the selection of victims is based on minimum cost factors, it may happen that same transaction is always picked is victim. as a result, the transaction never completes its designated task, thus there is starvation.

x—o—x

detection - ① abort is not possibly
② deadlock prevention tech.
③ wait for graph → lighweighed
④ bulky system → deadlock prevention tech.

Recover → ① Roll back
② 1) selection of victim
2) Roll back
3) starvation.

failure:- 1] Transfer failure
① ~~syst~~ logical error:-
② system error
2] system crash:-
bugs →
3] Disk failure:-

x—o—x

log based :- stable storage
log record, updating activities
Ti start
<Ti start> log record
write (x) <Ti, Xj, V1, V2>
<Ti commit>

Q.6] What are diff. types of failure?

1] **Transaction failure**:-

There are two types of ~~failure~~ error that may cause a transaction fail:

① **logical** error:-

transaction can no longer continue with its normal executioon because some internal condition.

like → bad input, data not found, overflow

② **System** error:-

The system has entered in undesirable state. ex. deadlock.

as a result transaction cannot continue with its normal execution.

2] **System crash**:-

① There is hardware malfunction / bug in database software or operating system. that causes the loss of content of volatile storage & brings transaction processing to halt.

② The content of non-volatile is not currepted.

③ The assumption that hardware errors & bugs in software bring system to a halt.

④ but do not currept nonvolatile storage content is known as fail-stop assumption.

3] **Disk failure**:-

① A disk block loses its content as a result of either a head crash / failure during data transfer operation.

② Copies of data on other disk, archival backup on tertiary media such as tapes, are used to recover from failure.

1) volatile storage

1) system crash
2) main memory, cache
3) access fastly
4) we can directly access data

Non volatile

— survive in system cras
— disk, magnetic tapes
— online storage, archival storage
— slower than volatile

Q.9] log-based Recovery mechanism:-

① A log is kept on stable storage.

② log is sequence of log records & maintain record of update activities.

③ When transaction T_i starts, it register itself by writing $<T_i$ start$>$ log record

④ Befor T_i execute write(x), a log record $<T_i, X_j, V_1, V_2>$ is written.

V_1 → value of x before write

V_2 → Value to be written to x

X_j → value V_1 before the write & will have value V_2 after the write.

⑤ When T_i finishes its last statement, log record $<T_i$ commit$>$ is written.

⑥ Two approaches using logs

— Deffered ~~data~~ database modification

— Immediate —||—