DBMS

10/4/24

Week-11

Lec. 11

Backup: copy of data containing all necessary contents

Two Type of Backup:

1) Physical Backup: copy of physical dataset file such as data, control file, log file, archived redo logs.

2) Logical Backup: copy of logical data that is extracted from a database consisting Table, procedure, views, funct"

⇒ Recovery is the process of restoring database

Why backup necessary?

1) Disaster Recovery:

2) Client side change: client want to modify existing applicat" and then also want previous version.

3) Auditing:

4) Down-time: time period for which system dont works properly.

Type of Backup Data:

1) Business Data: include personal info of client, employee, contractors etc.

2) System Data: include log file, software dependency data, disk image

3) Media: photo, video, generally larger size

Backup strategies:

1) Full Backup: a full backup must be done before any type of backup is done.
→ done on daily basis

advantage:
i) complete copy of all data
ii) easy to setup
iii) loss of single day backup doesnot affect ability to recover other backup.

disadv.:
i) take large time
ii) large amount of space
iii) result in longest system downtime during backup process

2) INCREMENTAL Backup:

↳ backup only data that has changed since last backup.

Sunday	monday	Tuesday	Thursday
100 MB	5 MB	5 MB	5 MB
full	increment.	increment.	increment.

advantage :

i) less storage needed
ii) downtime minimize
iii) less cost than full backup
iv) faster backup

dis adv:

i) restoring data become complex & time consuming
ii) incremental backup depend on previous backup.
iii) chance of data loss

3) DIFFERENTIAL Backup:

↳ backup all the changes that have occured since most recent full backup

How its work:

the ↳ i) Full backup
ii) differential Backup: backup those data that has changed since last full backup.

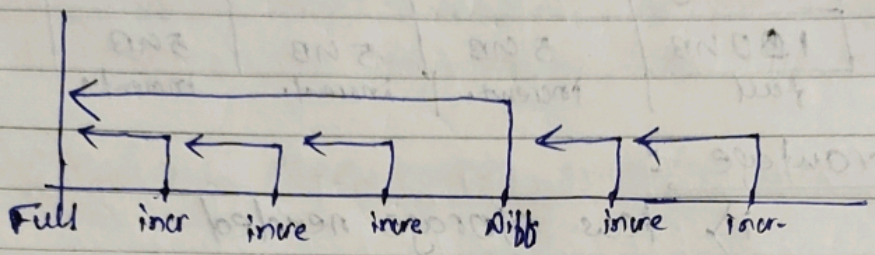
advantage :-

i) recoveries require fewer backup set
ii) provide better recovery options
iii) faster, less complex

disadv:

i) if done after long time, differential backup can reach size of full backup

Friday	sat	sun	mon	tue	wed
Full	incre.	incre	incre.	Diffrent.	incre



HOT BACKUP: Know as like backup.

↳ method of backing while system is still running

↳ Hot backup is preferable wherever possible

Advantage:

i) database always available to end user

ii) recovery is easier to achieve

iv) most efficient while dealing with dynamic & modularized data.

dis adv:

i) Not fesible when dataset is huge or monolothic

ii) fault tolerance is less.

iii) costly.

Transact" logging as HOT Backup:

↳ Hot bakup mainly used for Transact" Log Backup.

↳ cold backup like differential, incremental are preffered for data backup.

FAILURE CLASSIFICATION:

Transaction failure:
1) Logical error
2) System error (ex: deadlock)
3) System crash (power failure)
4) Disc failure

STORAGE STRUCTURE

1) VOLATILE STORAGE:
   - dont survive system crash
   - ex: main memory, cache memory

2) Non-volatile:
   - survive system crashes
   - disc, tape, flash memory
   - but still fail, losing data

3) STOable storage:
   - survives all failure
   - by maintaining multiple copies on distinct non-volatile media.

Database System Recovery:

A ~~- acid~~
A - atomicity
C - consistency
I - isolation
D - durability

→ concurrency control guarantees I, contribute to C

→ Applicat° program guarantee C

→ Recovery guarantee A & D, contribute to C

DATA Access:

→ physical blocks (non-volatile): blocks reside on disc

→ system buffer block (volatile): Block reside temprory in main memory.

→ Input (B) → transfer physical block B to main memory
Output (B) → transfer buffer block B to disc

Recovery & Atomicity.

- Log-based Recovery Mechanism
- Shadow paging

→ A log is kept on stable storage.
→ log is sequence of log records.

→ Immediate -modificat° allows update of transaction before commit.

→ deferred -modification allows update of transaction after commit.

→ restore all the data to their older value going backward.

Undo : Ctrl + Z

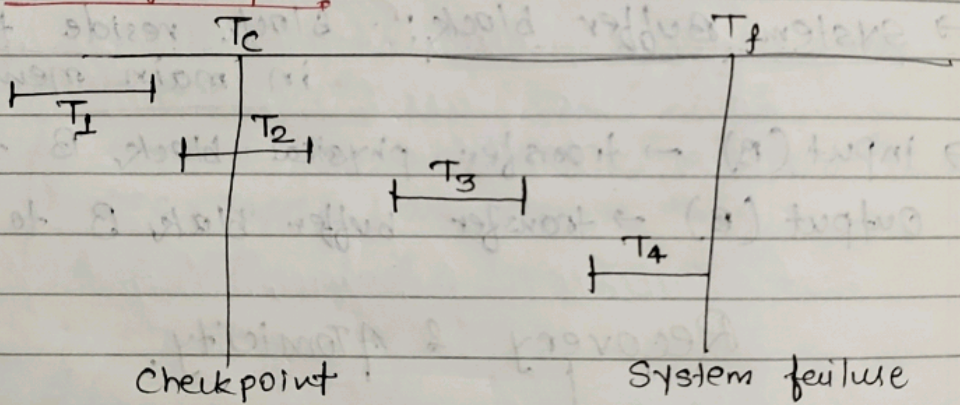
Redo : Ctrl + Y

↳ set the value of all data to new value going forward

→ Without commit if failure happens then UNDO

→ after commit if failure happen then REDO

Checkpoints:



→ Any transaction that commited before last checkpoint should be ignored.

→ T_1 can be ignored. (no recovery require)

→ Any transaction committed ~~after~~ since last checkpoint need to be **Redone**

→ T_2, T_3 redone

→ Any transaction that were running at the time of failure needs to be **undone**.

→ T_4 undone.

Tue 1103

HOT backup	cold backup
→ used for Transact" Log backup	→ ex: differential, Incremental.
→ dynamic	→ statics.
→ available 24 x 7	→ not available 24x7
→ live backup	→ takes when database is offline

TRANSACTION LOG:

- → are much smaller in size than actual data.
- → backup of inconsistent state of database can be restored to consistant state using transact" log.

Recovery: process of bringing a database back to a consistant & usable state.

→ a freshly recovered database is not always consistant.

Restore: process of retrieving data from backup copies.

→ freshly ~~recovered~~ restored database is always consistant.

lec 40.4

Recovery with Early Lock Release

→ index used in processing a transaction, such as B^+-tree, can be treated as normal data.

→ to increase concurrency, B^+-tree concurrency control often allows locks to release early.

Logical undo Logging

- Operation like B^+-tree insertion & deletion release locks early.
  - → they can't be undone by restoring old values (Physical undo)
  - → insert", delet" are undone by executing a delet" operat" (logical undo)

physical Redo

- → don't conflict with early lock release.
- → Redo information is logged physically.

- if crash / roll back occurs before operat" complete:
  → operation - end log record is not found
  → physical undo information is used to undo operation.

- if crash/rollback occurs after the operat complete:
  → operat"-log record is found, and
  → Logical undo is performed.
  → physical undo is ignored.