ST. XAVIER’S COLLEGE

**Maitighar,Kathmandu**

**(Affiliated to Tribhuvan University)**



**Database Management System**

**Lab Assignment #9**

**Submitted By**

Ajita Khatiwada

B.Sc. CSIT

Year II/IV Semester

013BSCIT004

**Submitted To**

Er. Sanjay Kumar Yadav

Lecturer,

Department of Computer Science

St. Xavier’s College

Maitighar, Kathmandu

**Submitted On**

October 1, 2015

1. **Databse recovery**

Retrieving deleted/inaccessible data from electronic storage media (hard drives, removable media, optical devices, etc...)

Typical causes of loss include:

* 1. Electro-mechanical Failure
  2. Natural Disaster
  3. Computer Virus
  4. Data Corruption
  5. Computer Crime

1. **purpose of data recovery**

A database may become inconsistent because of a

* 1. transaction failure (abort)
  2. database system failure (possibly caused by OS crash)
  3. media crash (disk-resident data is corrupted)

The recovery system ensures the database contains exactly those updates produced by committed transactions

* 1. I.e. atomicity and durability, despite failures

1. **Types of failure**

**1.transaction failures**

* 1. **Logical errors**: transaction cannot complete due to some internal error condition
  2. **System errors**: the database system must terminate an active transaction due to an error condition (e.g., deadlock)

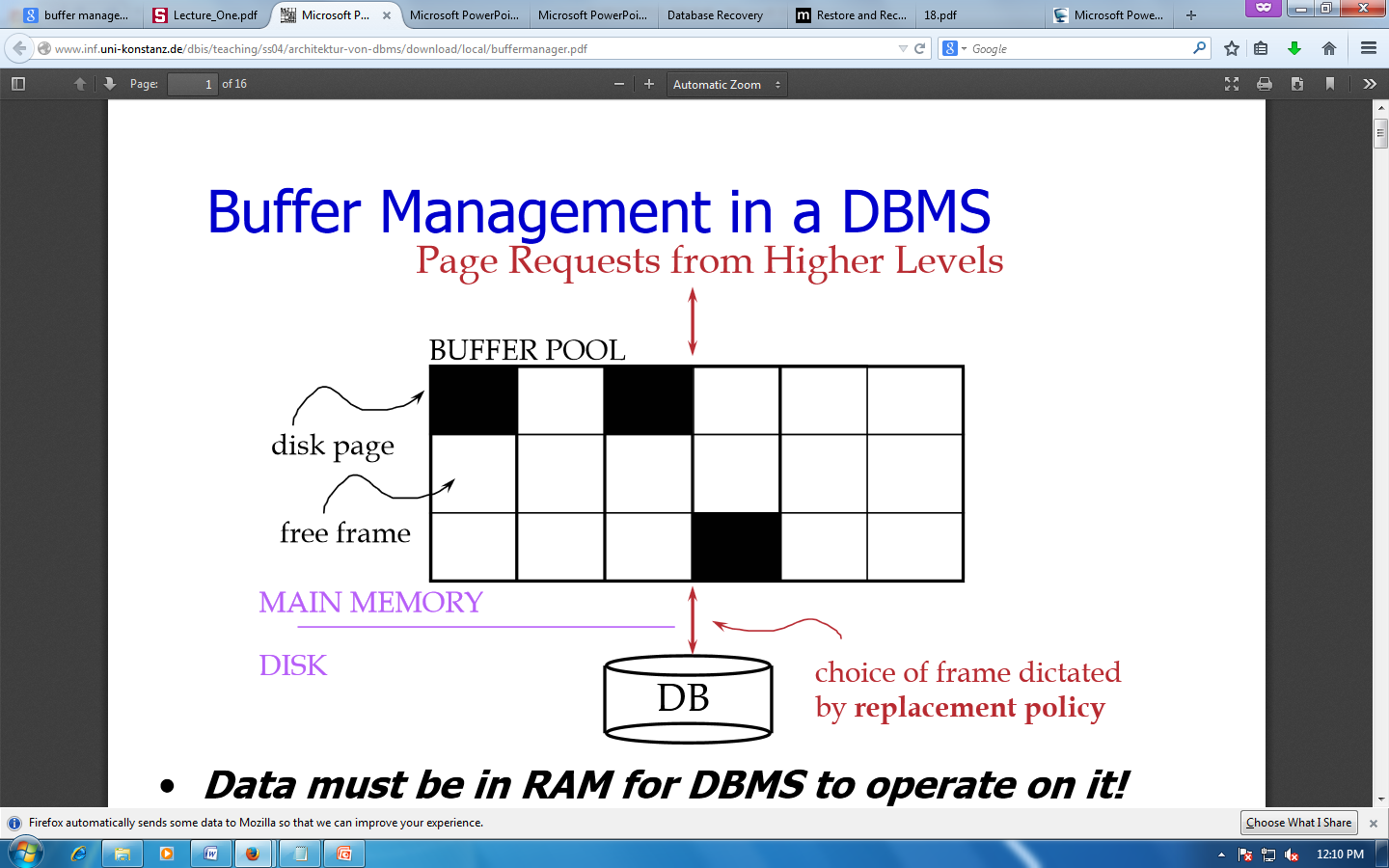
**2.System crash:** a power failure or other hardware or software failure causes the system to crash.Fail-stop assumption: non-volatile storage contents are assumed to not be corrupted by system crash.Database systems have numerous integrity checks to prevent corruption of disk data

**3** .**Disk failure**: a head crash or similar disk failure destroys all or part of disk storage

**4. Media failures**

* problems with disk head, unreadable media surface
* (parts of ) information on secondary storage may be lost
* lose a disk or a disk controller fails
* a head crash
* file corruption
* the overwriting or deletion of a datafile
* sny of the multiplexed control files are deleted or lost
* any datafile belonging to the system or the undo tablespace is deleted or lost.
* sn entire redo log group is lost.[1]

1. the storage hierarchy
2. buffer management



Data must be in RAM for DBMS to operate on it. Buffer Mgr hides the fact that not all data is in RAM

* When a Page is Requested
  + Buffer pool information table contains:

<frame#, pageid, pin\_count, dirty>

* If requested page is not in pool:
  + Choose a frame for replacement (only un-pinned pages are candidates)
  + If frame is “dirty”, write it to disk
  + Read requested page into chosen frame
  + Pin the page and return its address.
* Once we choose a page to remove
  + A page is dirty, if its contents have been changed after writing
  + Buffer Manager keeps a dirty bit
  + Say we choose to evict, if P is dirty, we write it to disk [\*]

1. **transaction log**

A **transaction** is an action, or a series of actions, carried out by a single user or an application program, which reads or updates the contents of a database.

* The transaction log records the details of all transactions
  + Any changes the transaction makes to the database
  + How to undo these changes
  + When transactions complete and how
  + The log is stored on disk, not in memory
  + If the system crashes it is preserved
* Write ahead log rule:

The entry in the log must be made before COMMIT processing can complete

1. data updates
2. data caching
3. **transaction roll back(undo)&roll forward**

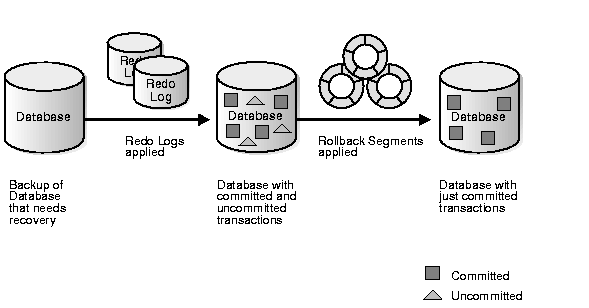
**transaction roll back**

ROLLBACK signals the unsuccessful end of a transaction

* + Any changes made by the transaction should be undone
  + It is now as if the transaction never existed

**transaction roll for ward**

The first step of recovery from an instance or disk failure is to roll forward, or reapply all of the changes recorded in the redo log to the datafiles. Because rollback data is also recorded in the redo log, rolling forward also regenerates the corresponding rollback segments. This is called cache recovery.Rolling forward proceeds through as many redo log files as necessary to bring the database forward in time. Rolling forward usually includes online redo log files and may include archived redo log files.

****

#### Fig : Basic Recovery Steps: Rolling Forward and Rolling Back

1. check pointing,shadow paging

A checkpoint is a procedure to limit the amount of work for Restart.

**Shadow paging** is an alternative to log-based recovery; this scheme is useful if transactions execute serially

Idea: maintain *two* page tables during the lifetime of a transaction –the **current page table**, and the **shadow page table**

Store the shadow page table in nonvolatile storage, such that state of the database prior to transaction execution may be recovered.

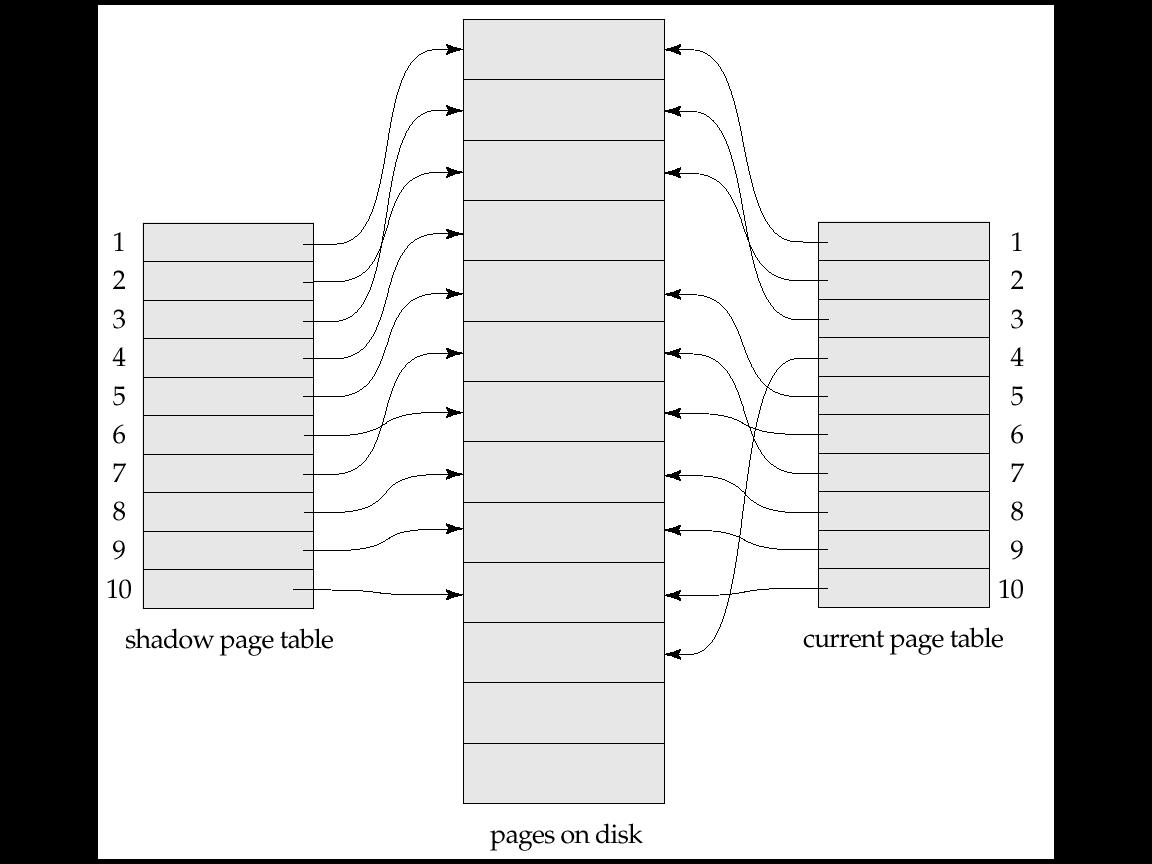
* + Shadow page table is never modified during execution

To start with, both the page tables are identical. Only current page table is used for data item accesses during execution of the transaction.

Whenever any page is about to be written for the first time

* + A copy of this page is made onto an unused page.
  + The current page table is then made to point to the copy

The update is performed on the copy



1. recovery schemes
2. **Failure with loss of non volatile storage (general concepts** )

Technique similar to checkpointing used to deal with loss of non-volatile storage

* 1. Periodically **dump** the entire content of the database to stable storage
  2. No transaction may be active during the dump procedure; a procedure similar to checkpointing must take place
     1. Output all log records currently residing in main memory onto stable storage.
     2. Output all buffer blocks onto the disk.
     3. Copy the contents of the database to stable storage.
     4. Output a record <**dump**> to log on stable storage.

1. recovery in multidatabase systems

reference :

[1]” <http://webdocs.cs.ualberta.ca/~zaiane/courses/cmput391/slides/L9-391-04.pdf>”, October 1, 2015.

\*]” <http://www.inf.uni-konstanz.de/dbis/teaching/ss04/architektur-von-dbms/download/local/buffermanager.pdf>” oct-1 2015

\* http://codeidol.com/community/sql/the-storage-hierarchy/3437/