CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

UNIVERSITY INSTITUTE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE &ENGINNERING

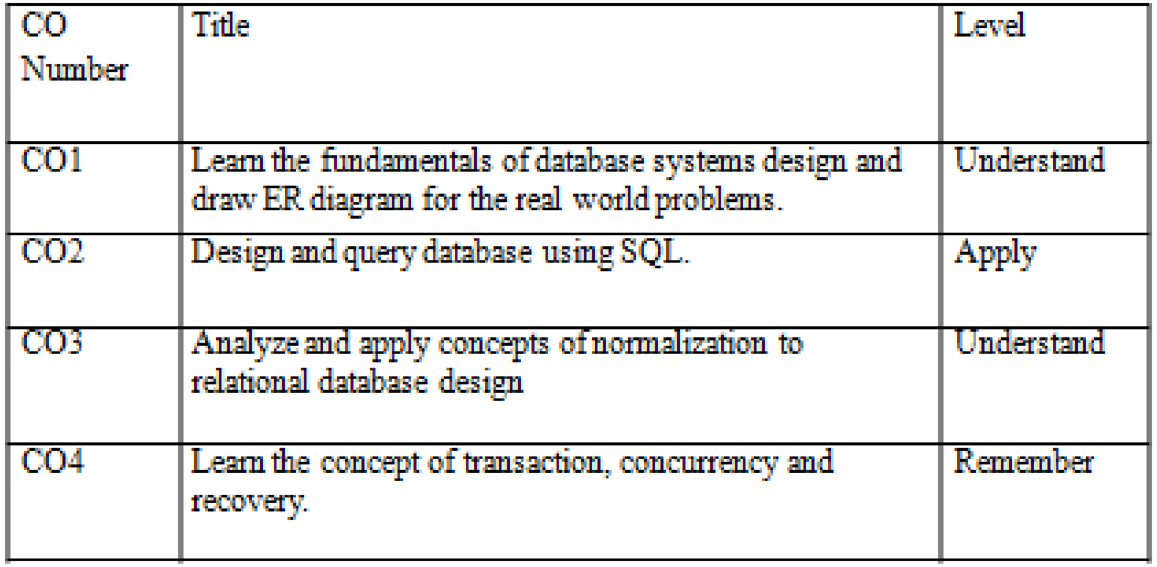
Bachelor of Engineering (Computer Science & Engineering)

Database Management System and CST-227

Prepared By: Ms. Heena Arora

Database Management System

Course Outcome



CO Number	Title	Level
CO1	Learn the fundamentals of database systems design and draw ER diagram for the real world problems.	Understand
CO2	Design and query database using SQL.	Apply
CO3	Analyze and apply concepts of normalization to relational database design	Understand
CO4	Learn the concept of transaction, concurrency and recovery.	Remember

DATABASE RECOVERY

Introduction

Types of Error

Recovery Techniques

Log Based Recovery

Shadow Paging

INTRODUCTION

Recovery : is the technique in which we retrieve the data after failure of system or after system crash.

Through recovery we maintain the consistency of the data.

Maintain the integrity of data.

TYPES OF ERROR

There are different types of errors in the system.

If transaction gets failed due to deadlock cause error in system.

If system gets failed due to failure of RAM cause error in system.

TECHNIQUES OF RECOVERY OF DATA

Transaction Log:

For recovery of data we maintains log of transactions.

For every transaction in the system one log file is maintained.

UPDATION OF DATA

Data Update:

- **Immediate Update:** there maintains log of transaction in cache first than final modification are performed at disks.
- **Deferred Update**: All the changes should performed in disks as series of transactions are performed.

CACHING

Data Caching:

- Every changes are initially stored at the cache memory , than it writes to final disks.
- It is the responsible of cache manager to manage the cache memory.

UNDO AND REDO OPERTAIONS

- **Undo**: Rollback the transaction at its initial state, if there exists system failure.
- **Redo**: Restore data from the log after successfully completion of data.

STORAGE MODEL

- Cache memory is used to maintain the logs after system crash
- All the logs are maintained by system in cache..

STABLE STORAGE

- Write(P) overwrites all of P on the disk
- If Write is unsuccessful, the error might be detected on the next read ...
  - e.g. page checksum error => page is corrupted
- … or maybe not
  - Write correctly wrote to the wrong location
- Write is the only operation that's atomic with respect to failures and whose successful execution can be determined by recovery procedures.

CACHE

- Cache memory is divided into pages.
- In this there we used dirty bit.
- Its dirty bit describe the status the page is updated or not.

CONT…

- In cache different operations are to be performed.
- Like fetch page lush page
- Fetch(P) – used to read the page in cache
- Flush(P) – when page is dirty used flush operation.

LOG FILE

- Log file is maintained in system.
- It describes every events occurred in the transaction.

RECOVERY MANAGER

- Processes Commit, Abort and Restart
- Recovery Manger manages the record of file.
- It maintains the record like which process has been completed, restart.

LOG BASED RECOVERY

- In the log there should maintain one log file
- It maintain the record when any transaction gets started..
- It maintains the record of read and write operations performed in the transactions.
- Two approaches using logs based recovery described as follows.
    - Deferred database modification
    - Immediate database modification

DEFERRED DATABASEMODIFICATION

Deferred Database Modification:

- During recovery after a crash, a transaction needs to be redone if and only if both <T_i **start**> and<T_i **commit**> are there in the log.
- Redoing a transaction T_i (**redo**T_i) sets the value of all data items updated by the transaction to the new values.
- Crashes can occur while
  - the transaction is executing the original updates, or
  - while recovery action is being taken
- example transactions T_0 and T_1 (T_0 executes before T_1):

T_0: **read** (A)

A: - A - 50

Write (A)

read (B)

B:- B + 50

write (B)

T_1 : **read** (C)

C:- C- 100

write (C)

SHADOW PAGING

Shadow Paging:

- **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

CONT…

- When a transition has to be committed:

1. Copy the current data into disks.
2. In this there is no need to use any recovery ,we can directly refer shadow page.

Characteristics of Shadow Paging

- Advantages of shadow-paging over log-based schemes
  - no overhead of writing log records
  - recovery is trivial
- Disadvantages :
  - Copying the entire page table is very expensive
    - Can be reduced by using a page table structured like a B^+-tree
      - No need to copy entire tree, only need to copy paths in the tree that lead to updated leaf nodes
  - Commit overhead is high even with above extension
    - Need to flush every updated page, and page table
  - Data gets fragmented (related pages get separated on disk)
  - After every transaction completion, the database pages containing old versions of modified data need to be garbage collected
  - Hard to extend algorithm to allow transactions to run concurrently
    - Easier to extend log based schemes

BACKUP MODES

Backup Modes:

- Hot backup
  - allows backup of the database while the database is running and available to users.
  - performance degrades during the backup period
  - takes longer than a cold backup
- Cold backup
  - requires database shutdown before backup begins
  - physical files are backed up while shutdown
  - database is unavailable to users during backup period
  - faster than a hot backup

ORACLE BACKUP

Oracle Backup Options:

- Off-line physical backup
  - is a cold backup technique
  - copies data files, log files, init files, and control files after shutdown
- On-line physical backup
  - is a hot backup technique
  - temporarily sets tablespaces into backup mode
- Logical Backup
  - is a hot backup technique
  - exports all or part of the database by creating SQL scripts necessary to recreate the objects

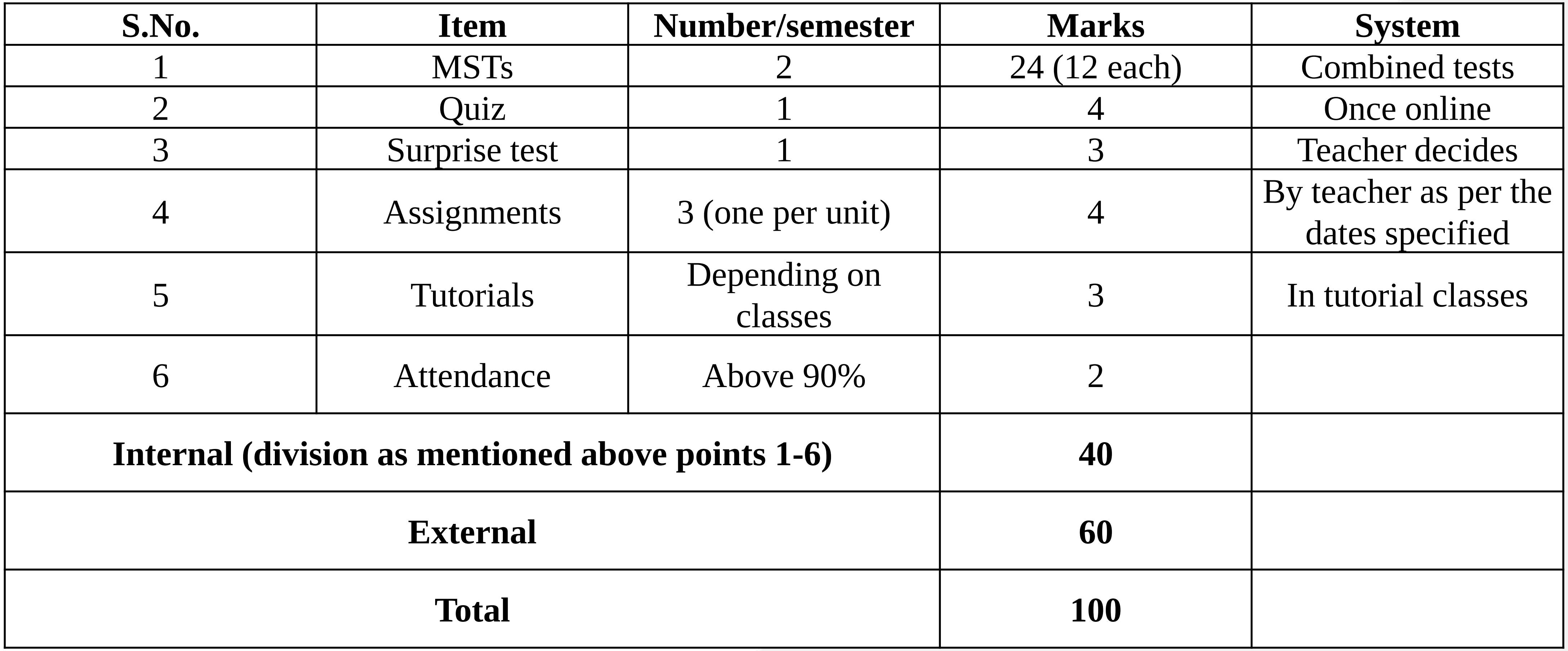
BACKUP TYPE

- Complete (Full)
  - copy all database and related files
  - delete the archive log files
- Cumulative (Differential)
  - copy blocks that have changed since last full backup

    or
  - copy all archive log files generated since last full backup
- Incremental
  - copy blocks that have change since the last partial backup

    or
  - copy all log files generated since last partial backup
- Complete (Copy)
  - copy all target data
  - Don't include the set in backup set logic

Assessment Pattern



S.No.	Item	Number/semester	Marks	System
1	MSTs	2	24 (12 each)	Combined tests
2	Quiz	1	4	Once online
3	Surprise test	1	3	Teacher decides
4	Assignments	3 (one per unit)	4	By teacher as per the dates specified
5	Tutorials	Depending on classes	3	In tutorial classes
6	Attendance	Above 90%	2	
Internal (division as mentioned above points 1-6)			**40**	
External			**60**	
Total			**100**	

REFERENCES

- “Data Structures and Algorithms Made Easy: Data Structures and Algorithmic Puzzles” by Narasimha Karumanchi
- “Data Structures Using C” by Aaron M. Tenenbaum
- “Data Structures and Algorithms” by Alfred V. Aho
- “Fundamentals of Data Structures in C by Ellis Horowitz, Sartaj Sahni, and Susan Anderson-Freed

THANK YOU

For queries

Email: Heenae7725@cumail.in