

OPERATING SYSTEMS AND ADBMS LABORATORY

[As per Choice Based Credit System (CBCS) scheme]

(Effective from the academic year 2016 -2017)

SEMESTER – I

Subject Code	16SCS16	IA Marks	20
Number of Lecture Hours/Week	01+03	Exam Marks	80
Total Number of Lecture Hours	50	Exam Hours	03

CREDITS – 02

Course objectives: This course will enable students to

- To provide students with contemporary knowledge in Data Compression and Coding.
- To equip students with skills to analyze and evaluate different Data Compression and Coding methods
- To be instrumental to handle multi dimension data compression
- To acquire practical knowledge on advanced databases and its applications.
- To analyze and work on areas like Storage, Retrieval, Multi valued attributes, Triggers and other complex objects, Algorithms etc related to ADBMS.
- To design and implement recent applications database for better interoperability

PART – A OS LABORATORY WORK:

1. Design and Develop a UNIX/LINUX shell program that should support at least 10 commands(Assume suitable application). OR Design a front-end application upon click of a button corresponding shell command should be executed.
- 2.
3. 2.Design and develop a program to implement lazy buddy system algorithm.
- 4.
5. 3.Write a multi-class multithreaded program that simulates multiple sleeping barbers, all in one barbershop that has a finite number of chairs in the waiting room. Each customer is instantiated from a single customer class; each barber is instantiated from a single Barber

class.

- 6.
7. 4. Create two processes and demonstrate the usage of Shared segment by the above processes (use shmget, signal, fork etc. to simulate the working environment of the program).
- 8.
9. 5. Design and develop a program to realize the virus classification, such as boot sector infector, file infector and macro

PART – B ADBMS LABORATORY WORK

Note: The following experiments may be implemented on MySQL/ORACLE or any other suitable RDBMS with support for Object features

1. Develop a database application to demonstrate storing and retrieving of BLOB and CLOB objects.

- a. Write a binary large object (BLOB) to a database as either binary or character (CLOB) data, depending on the type of the field in your data source. To write a BLOB value to the database, issue the appropriate INSERT or UPDATE statement and pass the BLOB value as an input parameter. If your BLOB is stored as text, such as a SQL Server text field, pass the BLOB as a string parameter. If the BLOB is stored in binary format, such as a SQL Server image field, pass an array of type byte as a binary parameter.
- b. Once storing of BLOB and CLOB objects is done, retrieve them and display the results accordingly.

2. Develop a database application to demonstrate the representation of multi valued attributes, and the use of nested tables to represent complex objects. Write suitable queries to demonstrate their use .

Consider Purchase Order Example: This example is based on a typical business activity: managing customer orders. Need to demonstrate how the application might evolve from relational to object-relational, and how you could write it from scratch using a pure object-oriented approach.

- a. Show how to implement the schema -- Implementing the Application under the Relational Model -- using only Oracle's built-in data types. Build an object-oriented application on top of this relational schema using object views

3. Design and develop a suitable Student Database application by considering appropriate attributes. Couple of attributes to be maintained is the Attendance of a student in each subject for which he/she has enrolled and Internal Assessment Using TRIGGERS, write active rules to do the following:

- a. Whenever the attendance is updated, check if the attendance is less than 85%; if so, notify the Head of the Department concerned.
- b. Whenever, the marks in an Internal Assessment Test are entered, check if the marks are less than 40%; if so, notify the Head of the Department concerned.

Use the following guidelines when designing triggers:

- Use triggers to guarantee that when a specific operation is performed, related actions are performed.
- Use database triggers only for centralized, global operations that should be fired for the triggering statement, regardless of which user or database application issues the statement.
- Do not define triggers that duplicate the functionality already built into Oracle. For example, do not define triggers to enforce data integrity rules that can be easily enforced using declarative integrity constraints.
- Limit the size of triggers (60 lines or fewer is a good guideline). If the logic for your trigger requires much more than 60 lines of PL/SQL code, it is better to include most of the code in

a stored procedure, and call the procedure from the trigger.

- Be careful not to create recursive triggers. For example, creating an AFTER UPDATE statement trigger on the EMP table that itself issues an UPDATE statement on EMP causes the trigger to fire recursively until it has run out of memory.
1. **Design, develop, and execute a program to implement specific Apriori algorithm for mining association rules. Run the program against any large database available in the public domain and discuss the results.**

Association rules are if/then statements that help uncover relationships between seemingly unrelated data in a relational database or other information repository. An example of an association rule would be "If a customer buys a dozen eggs, he is 80% likely to also purchase milk."

Course Outcomes

The students should be able to:

- Work on the concepts of Software Testing and ADBMS at the practical level
- Compare and pick out the right type of software testing process for any given real world problem
- Carry out the software testing process in efficient way
- Establish a quality environment as specified in standards for developing quality software
- Model and represent the real world data using object oriented database
- Embed the rules set in the database to implement various features of ADBMS
- Choose, design and implement recent applications database for better interoperability

Conduction of Practical Examination:

- 1 . All laboratory experiments (nos) are to be included for practical examination.
- 2 . Students are allowed to pick one experiment from **each part and execute both**
- 3 . Strictly follow the instructions as printed on the cover page of answer script for breakup of marks
- 4 . **PART –A:** Procedure + Conduction + Viva: **10 + 20 +10 (40)**
- 5 . **PART –B:** Procedure + Conduction + Viva: **10 + 20 +10 (40)**
- 6 . **Change of experiment is allowed only once and marks allotted to the procedure part to be made zero.**