

Computing Lab

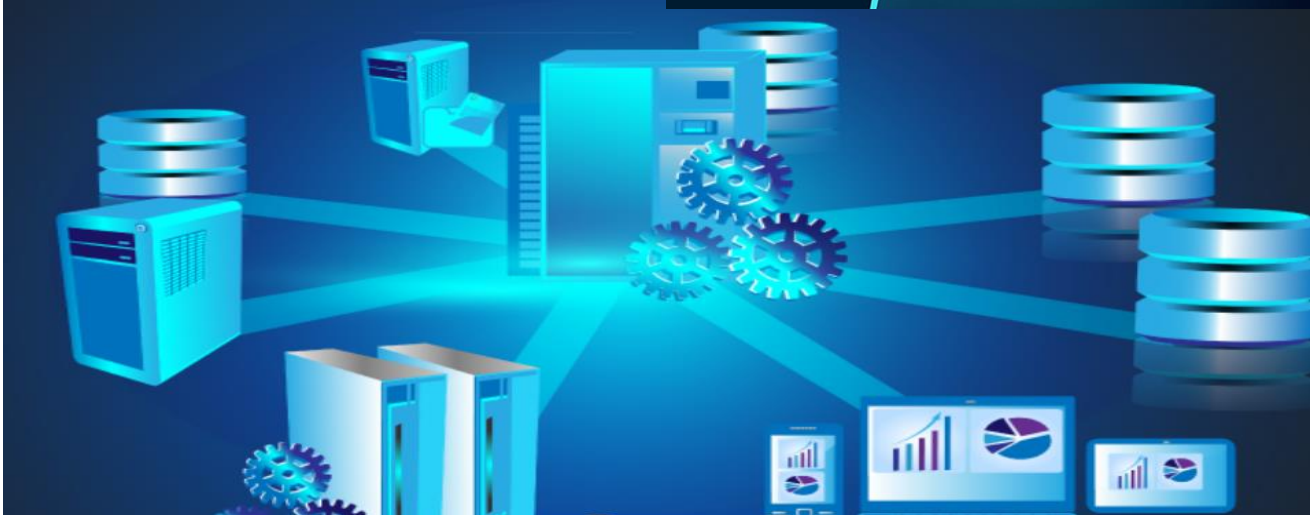
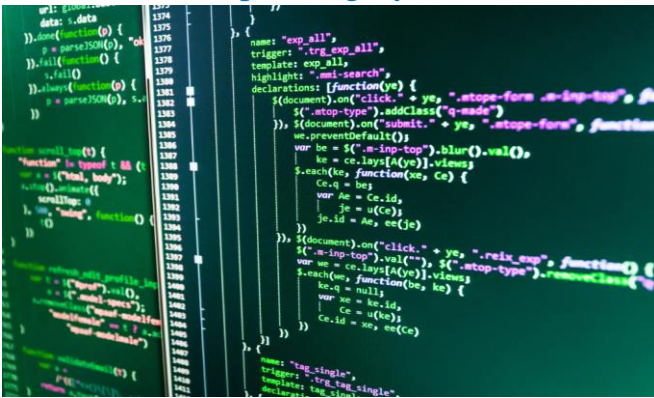
Associated Laboratories

- Database management systems
- Python programming
- Data science using Python
- Novel Engaging Course LABs

Major Equipment List:

Desktops with following configurations:

- Processor: Intel core i7 (6700/4790) @ 3.40 Ghz
- RAM: 4 Gb
- Operating System: Windows 10 / pro 8 / 8



In charge:

Dr. Nidhi Saxena

Associated Faculties:

Physical In Charge:

Mr. Imteyaz Husain

Department of Information Technology



Computing Lab

Code of Conducts

GENERAL PRACTICES TO BE FOLLOWED IN LABORATORY:

- You are expected to leave your computer workstation in the same condition as you found it.
- You are responsible for reading and abiding by all signs posted in the computer labs.
- Software downloaded from the Internet is not to be installed on any lab computer for any purpose.
- Documents should be saved to your network drive, to a flash drive or to a cloud storage account.
- The computer labs are an academic resource. As such, please respect the needs of others by not monopolizing the computers for non-academic use.

Do's:

1. Remove your shoes or wear foot socks before you enter the lab.
2. Always keep quiet. Be considerate to other lab users.
3. Report any problems with the computer to the person in charge.
4. Shut down the computer properly.



Don'ts:

1. Do not bring any food or drinks in the computer room.
2. Do not touch any part of the computer with wet hands.
3. Do not hit the keys on the computer too hard.
4. Don't damage, remove, or disconnect any labels, parts, cables or equipment. Do not install or download any software or modify or delete any system files on any lab computers.



Computing Lab

Database Management System Lab

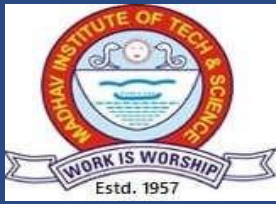
List of Experiments

While creating tables, databases the name should have a prefix of your roll number. Ex. If your roll number is 55 then every table name must start with 55 TABLE_NAME. 1. Write program name 2. Write description of command used for executing the query. 3. Write commands in bold letters. 4. Take the screenshot of the output.

1. Implementation of DDL commands of SQL with suitable examples. a. Create table; b. Alter table; c. Drop Table
2. Implementation of DML commands of SQL with examples. a. Insert; b. Update; and c. Delete
3. Implementation of different type of function with suitable example a. Number function; b. Aggregate function; c. Character function; d. Conversion function; e. Data function
4. Implementation of different type of operators in SQL. a. Arithmetic operators; b. Logical operators; c. Set operator; f. Comparison Operator; g. Special operator.
5. Implementation of type of joins. a. Inner Join; b. Outer Join; and Natural Join etc.
6. Study and implementation of a. Group by & having clause; b. order By clause; c. Indexing.
7. Study of Implementation of a. Sub queries; b. Views; 8. Study & implementation of different type of constraints.
8. Study & implementation of database backup & recovery command.
9. Study & implementation of Rollback, commit, save point.
10. Study & implementation of Rollback, commit, save point.

Course outcomes

- CO1. construct database schema for a given problem domain.
- CO2. apply integrity constraints on a database schema using a state-of-the-art RDBMS.
- CO3. apply SQL queries using DDL and DML to design and access database systems.
- CO4. make use of operators and functions used in query.
- CO5. distinguish Tables and Views for database systems.
- CO6. develop a small project for a real-world scenario.



Computing Lab

Data science using Python

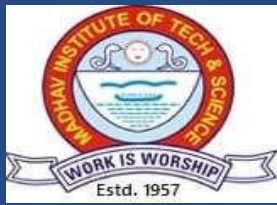
List of Experiments

1. Write a python program to compute
 - a. Central Tendency Measures: Mean, Median, Mode
 - b. Measure of Dispersion: Variance, Standard Deviation
2. Study of Python Basic Libraries such as Statistics, Math, Numpy and Scipy
3. Study of Python Libraries for data science such as Pandas and Matplotlib
4. For a given set of training data examples stored in a .CSV file, implement and demonstrate the Candidate-Elimination algorithm to output a description of the set of all hypotheses consistent with the training examples.
5. Write a Python program to implement Simple Linear Regression
6. Implementation of Multiple Linear Regression for House Price Prediction using sklearn.
7. Implementation of Decision tree using sklearn and its parameter tuning
8. Write a program to demonstrate the working of the decision tree based ID3 algorithm. Use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample.
9. Write a program to implement the naïve Bayesian classifier for a sample training dataset stored as a .CSV file. Compute the accuracy of the classifier, considering few test data sets.
10. Implementation of KNN using sklearn

Course outcomes :

Students will be able to

- CO1. develop relevant **programming** abilities.
- CO2. demonstrate proficiency with statistical **analysis of data**.
- CO3. develop the ability to build and assess data-based **models**.
- CO4. execute statistical analyses with professional statistical **software**.
- CO5. demonstrate skill in **data management**.



Computing Lab

Python Programming Lab

List of Experiments

1. Write a program to demonstrate different number data types in python.
 2. Write a program to perform different arithmetic operations on numbers in python.
 3. Write a program to create, concatenate and print a string and accessing substring from given string.
 4. Write a python program to create, append and remove lists in python.
 5. Write a program to demonstrate working with tuples in python.
 6. Write a program to demonstrate working with dictionaries in python.
 7. Write a python program to find the factorial of a number using recursion.
 8. WAP to swap two integers without using a third variable. The swapping must be done in a different method in a different class.
 9. WAP to read content of a file and write into another file.
 10. Write a python program to define a module and import a specific function in that module to another program.
-

Course outcomes

- CO1. write, test, and debug simple Python programs.
- CO2. solve computational problem using python language.
- CO3. familiar with basics syntax and features of python programming language.
- CO4. use Python lists, tuples, dictionaries for representing compound data.
- CO5. design a program utilizing the features of object-oriented concept.
- CO6. utilize some of the libraries available for solving problems



Computing Lab

Design and Analysis of Algorithm

List of Experiments

1. WAP to implement the following using array as data structure and analyze its time complexity.
 - a. Insertion sort
 - b. Selection sort
 - c. Bubble sort
 - d. Quick sort
 - e. Merge sort
 - f. Bucket sort
 - g. Heap sort
2. WAP to implement Linear and Binary Search and analyze its time complexity.
3. WAP to implement Strassen's Matrix Multiplication.
4. WAP to implement Matrix Chain Multiplication and analyze its time complexity.
5. WAP to implement Longest Common Subsequence Problem and analyze its time complexity.
6. WAP to implement Optimal Binary Search Tree Problem and analyze its time complexity.
7. WAP to implement 0/1 knapsack using dynamic programming.
8. WAP to implement Dijkstra's Algorithm and analyze its time complexity.
9. WAP to implement Bellman Ford Algorithm and analyze its time complexity.
10. WAP to implement DFS and BFS and analyze their time complexities.

Course outcomes

- CO1. relate the principles of algorithm design in solving problems.
- CO2. demonstrate basic algorithms and different problem solving strategies.
- CO3. build creativeness and confidence to solve non-conventional problems.
- CO4. analyze running times of algorithms using asymptotic analysis.
- CO5. compare various algorithm design approaches for solving real world problems.
- CO6. design and implement optimization algorithms in specific applications.

