



JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR
 (Established by Govt. of A.P., ACT No.30 of 2008)
 ANANTAPUR – 515 002 (A.P) INDIA

Master of Computer Applications

Ist Year Semester - 1 (Theory - 6, Lab - 3)						
S.No	Course No	Course Name	L	T	P	Credits
1.	20F00101	Mathematical Foundations of Computer Science	3	0	0	3
2.	20F00102	Foundations of Data Science	3	0	0	3
3.	20F00103	Computer Organization & Architecture	3	0	0	3
4.	20F00104	Python Programming	2	0	0	2
5.	20F00105	Data Structures	4	0	0	4
6.	20F00106	Database Management Systems	3	0	0	3
7.	20F00107	Data Science and Python Laboratory	0	0	4	2
8.	20F00108	Data Structures using C Laboratory	0	0	4	2
9.	20F00109	Database Management Systems Laboratory	0	0	4	2
Total						24

Ist Year Semester - II (Theory - 6, Lab – 3, M-1)						
S.No	Course No	Course Name	L	T	P	Credits
1.	20F00201	Operating Systems	3	0	0	3
2.	20F00202	Software Engineering	3	0	0	3
3.	20F00203	Artificial Intelligence	3	0	0	3
4.	20F00204	Computer Networks	3	0	0	3
5.	20F00205a 20F00205b 20F00205c 20F00205d 20F00205e	<i>Elective – I (MOOCs)</i> 1. E-Commerce 2. Operations Research 3. Digital Marketing 4. Data Mining and Business Intelligence 5. Internet of Things	3	0	0	3
6	20F00206a 20F00206b 20F00206c 20F00206d 20F00206e	<i>Elective-II</i> 1. Cloud Computing 2. Formal Languages and Automata Theory 3. Computer Graphics 4. Design and Analysis of Algorithms	3	0	0	3
7.	20F00207	Operating Systems & Computer Networks Laboratory	0	0	4	2
8.	20F00208	Software Engineering Laboratory	0	0	4	2
9.	20F00209	Artificial Intelligence using R Laboratory	0	0	4	2
10..	20F00210	*Mandatory Audit Course-1				NA
Total						24

II year Semester - 1 (Theory - 6, , Lab – 3, M-1)						
S.No	Course No	Course Name	L	T	P	Credits
1.	20F00301	Web & OOA Design	3	0	0	3
2.	20F00302	Big Data Analytics	3	0	0	3
3.	20F00303	Dev. Ops & Agile Programming	3	0	0	3
4.	20F00304a	Elective – III 1. Software Architecture & Design Patterns	3	0	0	3
	20F00304b	2. Cyber Security				
	20F00304c	3. Multimedia Systems & Tools				
	20F00304d	4. Machine Learning				
	20F00304e	5. Compiler Design				
5.	20F00305a	Elective-IV 1. Software Project Management	3	0	0	3
	20F00305b	2. Wireless Sensor Networks				
	20F00305c	3. Service Oriented Architecture				
	20F00305d	4. Deep Learning				
	20F00305e	5. Android Programming				
6.	20F00306a	Elective-V 1. Software Quality Assurance & Testing	3	0	0	3
	20F00306b	2. Block Chain Technologies				
	20F00306c	3. Augmented Reality/Virtual Reality				
	20F00306d	4. Natural Language Processing				
	20F00306e	5. Mobile Application Development				
7.	20F00307	Web & OOA Design using Java Laboratory	0	0	4	2
8.	20F00308	Big Data Analytics Laboratory	0	0	4	2
9.	20F00309	Dev. Ops & Agile Programming Laboratory	0	0	4	2
10	20F00310	*Mandatory Audit Course-2				
Total						24

II year Semester - II (Project & Comprehensive Viva))			
S.No	Course No	Course Name	C
1.	20F00401	Project work	16
2.	20F00402	Comprehensive Viva-Voce	2
			18

(20F00101) MATHEMATICAL FOUNDATIONS OF COMPUTER SCIENCE

Course Objectives

- To explain about the Boolean Algebra, Graph theory and Recurrence relations.
- To demonstrate the application of basic methods of discrete mathematics in Computer Science problem solving.
- To elucidate solving mathematical problems from algorithmic perspective.
- To introduce the mathematical concepts which will be useful to study advanced courses Design and Analysis of Algorithms, Theory of Computation, Cryptography and Software Engineering etc.

UNIT- I

Statements and Notation, Connectives- Negation, Conjunction, Disjunction, Conditional and Bi-conditional, Statement formulas and Truth Tables. Well-formed formulas, Tautologies, Equivalence of Formulas, Duality Law, Tautological Implications.

Normal Forms: Disjunctive Normal Forms, Conjunctive Normal Forms, Principal Disjunctive Normal Forms (PDNF), Principal Conjunctive Normal Forms (PCNF), Ordering and Uniqueness of Normal Forms.

The Theory of Inference for the Statement Calculus: Rules of Inference, Consistency of Premises and Indirect Method of Proof.

The predicate Calculus, Inference theory of the Predicate Calculus.

Learning Outcomes:

At the end of this unit, the student will be able to

- Describe logical sentences in terms of predicates, quantifiers, and logical connectives (L1)
- Evaluate basic logic statements using truth tables and the properties of logic (L5).
- Apply rules of inference to test the consistency of premises and validity of arguments (L3).
- Verify the equivalence of two formulas and their duals (L4).
- Find the Principal Conjunctive and Principal Disjunctive Normal Forms of a statement formula (L1).

UNIT-II

Set Theory: Basic concepts of Set Theory, Representation of Discrete structures, Relations and Ordering, Functions, Recursion.

Lattices and Boolean Algebra: Lattices as Partially Ordered Sets, Partial order relation, poset - Lattices, Hasse diagram, Boolean Functions, Representation and Minimization of Boolean Functions.

Learning Outcomes:

At the end of this unit, the student will be able to

- Describe equivalence, partial order and compatible relations (L1).
- Compute Maximal Compatibility Blocks (L3).
- Identify the properties of Lattices (L2).
- Evaluate Boolean functions and simplify expression using the properties of Boolean algebra (L5).

UNIT-III

Algebraic Structures: Semi group - Monoid – Groups, Cyclic group - Permutation group(S_n and D_n)
- Substructures - Homomorphism of semi group, monoid and groups - Cosets and Lagrange Theorem
– Normal Subgroups - Rings and Fields (Definition and examples only)

Learning Outcomes:

At the end of this unit, the student will be able to

- Infer Homomorphism and Isomorphism (L4).
- Describe the properties of Semi groups, Monoids and Groups (L1).
- Describe the properties Normal subgroups.
- Describe the properties Rings and Fields.

UNIT-IV

Elementary Combinatorics: Basics of Counting, Combinations and Permutations, Enumeration of Combinations and Permutations, Enumerating Combinations and Permutations with repetitions, Enumerating Permutations and Combinations with constrained Representations, Binomial Coefficients, The Binomial and Multinomial Theorems, The Principle of Inclusion and Exclusion.

Learning Outcomes:

At the end of this unit, the student will be able to

- Explain fundamental principle of counting (L2).
- Examine the relation between permutation and combination (L4).
- Solve counting problems by applying elementary counting techniques using the product and sum rules (L3).
- Apply permutations, combinations, the pigeon-hole principle, and binomial expansion to solve counting problems (L3).

UNIT-V:

Recurrence Relations: Generating Functions of Sequences, Calculating Coefficients of Generating Functions, Recurrence Relations, Solving Recurrence Relations by Substitution and Generating Functions, The method of Characteristic Roots, Solution of Inhomogeneous Recurrence Relations.

Learning Outcomes:

At the end of this unit, the student will be able to

- Find the generating functions for a sequence (L1).
- Design recurrence relations using the divide-and-conquer algorithm (L6).
- Solve linear recurrence relations using method of Characteristic Roots (L3).
- Outline the general solution of homogeneous or Inhomogeneous Recurrence Relations using substitution and method of generating functions (L2).
- Solve problems using recurrence relations and recursion to analyze complexity of algorithms (L3).

Text Books:

1. Joe L. Mott, Abraham Kandel and Theodore P. Baker, "Discrete Mathematics for Computer Scientists & Mathematicians", 2nd Edition, Pearson, 2008. (for Units III to V).
2. J P Trembly and R Manohar, "Discrete Mathematical Structures with Applications to Computer Science", 1st Edition, McGraw Hill, 2017(For Unit I&II).

Reference Books:

1. Ralph P. Grimaldi and B.V. Ramana, "Discrete and Combinatorial Mathematics, an Applied Introduction", 5th Edition, Pearson, 2016.
2. Narsingh Deo, "Graph Theory with Applications to Engineering", Prentice Hall, 1979.
3. D.S. Malik and M.K. Sen, "Discrete Mathematics theory and Applications", 1st Edition, Cenegage Learning, 2012.
4. C L Liu and D P Mohapatra, "Elements of Discrete Mathematics, A computer Oriented approach", 4th edition, MCGRAW-HILL, 2018.

After completion of this course the student would be able to

- Evaluate elementary mathematical arguments and identify fallacious reasoning (L5).
- Understand the properties of Compatibility, Equivalence and Partial Ordering relations, Lattices and Has see Diagrams (L1).
- Understand the general properties of Algebraic Systems, Semi Groups, Monoids, Groups (L1), Normal subgroups, Rings and Fields
- Design solutions for problems using breadth first and depth first search techniques (L6)
- Solve the homogeneous and non-homogeneous recurrence relations (L3).

(20F00102) FOUNDATIONS OF DATA SCIENCE

Course Objectives

Ideally for a student to understand Data Science, he/she should have exposure to the following. This will give a basic feel about Data Science. In the following, the topics highlighted in light blue is minimum needed and those highlighted in yellow will help to get a feel about the subject.

Overall it covers the following:

- Basics of probability
- Basics of statistics
- Pattern Recognition
- Machine Learning
- Introduction on Deep Neural Networks.

UNIT-I

Introduction to Probability and Statistics:

Descriptive Statistics: Measures of central tendency—mean, median, mode, harmonic mean and geometric mean; Measures of dispersion – mean deviation from mean, standard deviation and variance. Central moments. Linear and rank correlation. Covariance and correlation; Statistics and sampling distributions; Hypothesis testing of means, proportions, variances and correlations Definition of random variable and probability, (problems depending on counting –taught in MFCS), discrete probability distributions: Bernoulli, Binomial, Poisson; Continuous probability distributions: Gaussian, Exponential, Chi-square. Definition of Bayesian probability.

Probability theory and axioms; Random variables; Probability distributions and density functions (univariate and multivariate); Expectations and moments;; Confidence (statistical) intervals; Correlation functions; White-noise process. Probability, Hypothesis and Inference, Gradient Descent.

Learning Outcomes:

At the end of this unit, the student will be able to

- Understand how to summarise data through measures of central tendency, dispersion, moments and correlation.
- Understand random variable and associated probability distributions
- It lays foundations for further UNITS.

UNIT-II

Exploratory Data Analysis (EDA), Data Science life cycle, Descriptive Statistics, Basic tools (plots, graphs and summary statistics) of EDA, Philosophy of EDA.

Data Visualization: Scatter plot, bar chart, histogram, boxplot, heat maps etc.

Learning Outcomes:

At the end of this unit, the student will be able to

- Learn to analyse the data through plotting: from raw-scatter plot to summarized visualization.
- Understand the functionality of various phases of data Science.
- Analyze the characteristics of data

UNIT-III

Introduction to Pattern Recognition and Machine Learning

Patterns, features, patten representation, curse of dimensionality, dimensionality reduction. Supervised and Unsupervised learning. Classification—linear and non-linear. Bayesian, Perceptron, Nearest neighbor classifier, Support vector machine, use of kernels, Logistic regression, Naïve-bayes, decision trees and random forests; boosting and bagging. Clustering---partitional and hierarchical; k-means clustering.

Regression. Least squares. Evaluation metrics: RMSE, MAE and Coefficient of Determination (R-square)

Cost functions, training and testing a classifier. Cross-validation. Class-imbalance – ways of handling, Exploratory data analysis (EDA), evaluation metrics— Precision, Recall, RoC, AUC; Confusion matrix, Classification accuracy

Learning Outcomes:

At the end of this unit, the student will be able to

- Understand the meaning of pattern, feature, importance of pattern representation.
- Classification methods, their working and evaluation. Minimising cost function
- Regression and evaluation.
- Exploratory data analysis (outcome of Unit-I; can be tried out in the associated lab)

UNIT-IV

Introduction to Deep Learning

Multilayer perceptron. Back propagation. Loss functions. Epochs and Batch sizes. Hyper parameter tuning. Applications to classification, regression and unsupervised learning. Overview (introduction to the terms) of RNN, CNN and LSTM.

Learning Outcomes:

At the end of this unit, the student will be able to

- Understanding of deep neural networks as extension of multi-layer perceptron
- Unsupervised and supervised learning in deep neural networks
- Understand what is deep learning
- Minimizing loss functions

- How a deep learning models are trained
- Familiarity of popular deep learning models

UNIT-V

Overview of Data Science Models.

Applications to text, images, videos: recommender systems, image classification, Social network graphs

Learning Outcomes:

At the end of this unit, the student will be able to

- Applications of data science/machine learning
- Familiarity of applications

Textbooks:

1. Cathy O’Neil, Rachel Schutt, Doing Data Science, Straight Talk from the Frontline. O’Reilly, 2013.
2. Christopher Bishop, Pattern Recognition and Machine Learning, Springer, 2007.
3. Ian Goodfellow, YoshuaBengio, Aaron Courville, Deep Learning, MIT Press, 2016
4. Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements of Statistical Learning, Springer 2009.
5. Erwin kreyszig, Advanced Engineering Mathematics, 10th Edition, John Wiley & Sons,2011.

References:

1. Michael Nielsen, Neural Networks and Deep Learning, Determination Press, 2015.
2. Francois Chollet, Deep Learning with Python, 1/e, Manning Publications Company, 2017
3. [UNIT-1] Robert V Hogg, Allen Craig and J.W. McKean. Introduction to Mathematical Statistics. Pearson. (Indian Edition)
4. [UNIT-II] EMC2: 4) Data Science and Big Data Analytics, EMC Education Services, EMC 2 , Wiley Publication, 2015.
5. [UNIT-III] V. Susheela Devi and M. NarasimhaMurty. Pattern Recognition – An Introduction. Universities Press (Indian Edition; there is an expensive Springer version of the same)
6. [UNIT-II] Christopher M. Bishop. Pattern Recognition and Machine Learning. Springer (Indian Edition)
7. [UNIT-III]Ian Goodfellow and YoshuaBengio and Aaron Courville. Deep Learning. MIT Press. Book available online at <https://www.deeplearningbook.org/>.
8. [UNIT-IV] J. Leskovec, A. Rajaraman, J.D. Ullman. Mining of Massive Datasets. Cambridge University Press. (Indian Edition; Online pdf is available for download)

(20F00103) COMPUTER ORGANIZATION & ARCHITECTURE

Course Objectives:

- Learn the fundamentals of computer organization and its relevance to classical and modern problems of computer design
- Understand the structure and behavior of various functional modules of a computer.
- Discuss the techniques that computers use to communicate with I/O devices
- Study the concepts of pipelining and the way it can speed up processing.
- Describe the basic characteristics of multiprocessors

UNIT - I

Basic Structure of Computer: Computer Types, Functional Units, Basic operational Concepts, Bus Structure, Software, Performance, Multiprocessors and Multicomputer.

Machine Instructions and Programs: Numbers, Arithmetic Operations and Programs, Instructions and Instruction Sequencing, Addressing Modes, Basic Input/output Operations

INTEL-8086: CPU architecture, Addressing modes - generation of physical address- code segment registers, Zero, one, two, and three address instructions. INTEL 8086 ASSEMBLY LANGUAGE INSTRUCTIONS-Data transfer instructions, input- output instructions, arithmetic, logical, shift, and rotate instructions, Conditional and unconditional transfer.

Learning Outcomes:

At the end of this unit, the student will be able to

- Understand the basic functional units and the ways they are interconnected to form a computer system
- Illustrate various addressing modes for accessing register and memory operands
- Describe the instruction sequencing and various types of instructions

UNIT - II

Arithmetic: Addition and Subtraction of Signed Numbers, Design of Fast Adders, Multiplication of Positive Numbers, Signed-operand Multiplication, Fast Multiplication, Integer Division, Floating-Point Numbers and Operations.

Basic Processing Unit: Fundamental Concepts, Execution of a Complete Instruction, Multiple-Bus Organization, Hardwired Control, Multiprogrammed Control.

Learning Outcomes:

At the end of this unit, the student will be able to

- Understand the arithmetic operations on signed numbers
- Represent the floating-point numbers and describe the operations performed on them

- Distinguish between hardwired and microprogrammed control units

UNIT - III

The Memory System: Basic Concepts, Semiconductor RAM Memories, Read-Only Memories, Speed, Size and Cost, Cache Memories, Performance Considerations, Virtual Memories, Memory Management Requirements, Secondary Storage.

Learning Outcomes:

At the end of this unit, the student will be able to

- Understand various types of memories
- Analyze the performance of cache memory
- Apply effective memory management strategies

UNIT - IV

Input/output Organization: Accessing I/O Devices, Interrupts, Processor Examples, Direct Memory Access, Buses, Interface Circuits, Standard I/O Interfaces.

Learning Outcomes:

At the end of this unit, the student will be able to

- Understand the basics of I/O data transfer synchronization
- Analyze the interrupt handling mechanisms of various processors
- Describe various techniques for I/O data transfer methods.

UNIT - V

Pipelining: Basic Concepts, Data Hazards, Instruction Hazards, Influence on Instruction Sets.

Large Computer Systems: Forms of Parallel Processing, Array Processors, The Structure of General-Purpose multiprocessors, Interconnection Networks.

Learning Outcomes:

At the end of this unit, the student will be able to

- Understand the use of pipelining and multiple functional units in the design of high-performance processors
- Design and analyze a high performance processor
- Describe the interconnection networks for multiprocessors.

TEXT BOOKS:

1. Computer Organization, Carl Hamacher, Zvonko Vranesic, Safwat Zaky, McGraw Hill Education, 5th Edition, 2013.
2. Microprocessors and Interfacing, Douglas Hall, Tata McGraw-Hill.

REFERENCE BOOKS:

1. Computer System Architecture, M.Morris Mano, Pearson Education, 3rd Edition.
2. Computer Organization and Architecture, Themes and Variations, Alan Clements, CENGAGE Learning.
3. Computer Organization and Architecture, Smruti Ranjan Sarangi, McGraw Hill Education.
4. Computer Architecture and Organization, John P.Hayes, McGraw Hill Education

Course Outcomes:

At the end of this unit, the student will be able to

- Demonstrate computer architecture concepts related to design of modern processors, memories and I/Os
- Able to explore the hardware requirements for cache memory and virtual memory
- Ability to design algorithms to exploit pipelining and multiprocessors
- Ability to use memory and I/O devices effectively
- Detect pipeline hazards and identify possible solutions to those hazards

(20F00104) PYTHON PROGRAMMING

Course Objectives:

- To learn the fundamentals of Python
- To elucidate problem-solving using a Python programming language
- To introduce a function-oriented programming paradigm through python
- To get training in the development of solutions using modular concepts
- To introduce the programming constructs of python

Unit – I

Introduction: What is a program, Running python, Arithmetic operators, Value and Types.

Variables, Assignments and Statements: Assignment statements, Script mode, Order of operations, string operations, comments.

Functions: Function calls, Math functions, Composition, Adding new Functions, Definitions and Uses, Flow of Execution, Parameters and Arguments, Variables and Parameters are local, Stack diagrams, Fruitful Functions and Void Functions, Why Functions.

Learning Outcomes:

At the end of this unit, the student will be able to

- List the basic constructs of Python.
- Solve the problems by applying modularity principle.

Unit – II

Case study: The turtle module, Simple Repetition, Encapsulation, Generalization, Interface design, Refactoring, docstring.

Conditionals and Recursion: floor division and modulus, Boolean expressions, Logical operators, Conditional execution, Alternative execution, Chained conditionals, Nested conditionals, Recursion, Infinite Recursion, Keyboard input.

Fruitful Functions: Return values, Incremental development, Composition, Boolean functions, More recursion, Leap of Faith, Checking types,

Learning Outcomes:

At the end of this unit, the student will be able to

- Apply the conditional execution of the program.
- Apply the principle of recursion to solve the problems.

Unit – III

Iteration: Reassignment, Updating variables, The while statement, Break, Square roots, Algorithms.

Strings: A string is a sequence, len, Traversal with a for loop, String slices, Strings are immutable, Searching, Looping and Counting, String methods, The in operator, String comparison.

Case Study: Reading word lists, Search, Looping with indices.

Lists: List is a sequence, Lists are mutable, Traversing a list, List operations, List slices, List methods, Map filter and reduce, Deleting elements, Lists and Strings, Objects and values, Aliasing, List arguments.

Learning Outcomes:

At the end of this unit, the student will be able to

- Use the data structure list.
- Design programs for manipulating strings.

Unit – IV

Dictionaries: A dictionary is a mapping, Dictionary as a collection of counters, Looping and dictionaries, Reverse Lookup, Dictionaries and lists, Memos, Global Variables.

Tuples: Tuples are immutable, Tuple Assignment, Tuple as Return values, Variable-length argument tuples, Lists and tuples, Dictionaries and tuples, Sequences of sequences.

Files: Persistence, Reading and writing, Format operator, Filename and paths, Catching exceptions, Databases, Pickling, Pipes, Writing modules.

Classes and Objects: Programmer-defined types, Attributes, Instances as Return values, Objects are mutable, Copying.

Classes and Functions:

Learning Outcomes:

At the end of this unit, the student will be able to

- Apply object orientation concepts.
- Use data structure dictionaries.
- Organize data in the form of files.

Unit – V

Classes and Functions: Time, Pure functions, Modifiers, Prototyping versus Planning

Classes and Methods: Object oriented features, Printing objects, The init method, The __str__ method, Operator overloading, Type-based Dispatch, Polymorphism, Interface and Implementation

Inheritance: Card objects, Class attributes, Comparing cards, decks, Printing the Deck, Add Remove shuffle and sort, Inheritance, Class diagrams, Data encapsulation.

The Goodies: Conditional expressions, List comprehensions, Generator expressions, any and all, Sets, Counters, defaultdict, Named tuples, Gathering keyword Args,

Learning Outcomes:

At the end of this unit, the student will be able to

- Plan programs using object orientation approach.
- Illustrate the principle of inheritance.

Text books:

1. Allen B. Downey, “Think Python”, 2nd edition, SPD/O’Reilly, 2016.

Reference Books:

1. Martin C.Brown, “The Complete Reference: Python”, McGraw-Hill, 2018.
2. Kenneth A. Lambert, B.L. Juneja, “Fundamentals of Python”, CENGAGE, 2015.
3. R. Nageswara Rao, “Core Python Programming”, 2nd edition, Dreamtech Press, 2019

Course Outcomes:

At the end of this unit, the student will be able to

- Apply the features of Python language in various real applications.
- Select appropriate data structure of Python for solving a problem.
- Design object oriented programs using Python for solving real-world problems.
- Apply modularity to programs.

(20F00105) DATA STRUCTURES

Course Objectives:

- To illustrate the basic concepts of C programming language.
- To discuss the concepts of Functions, Arrays, Pointers and Structures.
- To familiarize with Stack, Queue and Linked lists data structures.
- To explain the concepts of non-linear data structures like graphs and trees.
- To learn the different types of searching and sorting techniques.

Unit-1

Introduction to C Language - C Language Elements, Variable Declarations and Data Types, Operators and Expressions, Decision Statements - If and Switch Statements, Loop Control Statements - while, for, do-while Statements.

Introduction to Functions, Storage classes, Arrays, Structures, Unions, Pointers, Strings and Command line arguments.

Learning Outcomes:

At the end of this unit, the student will be able to

- Use C basic concepts to write simple C programs. (L3)
- Design and develop C programs using functions and arrays. (L6)
- Apply String handling functions and pointers. (L3)

Unit – 2

Data Structures, Stacks and Queues- Overview of Data Structure, Representation of a Stack, Stack Related Terms, Operations on a Stack, Implementation of a Stack, Evaluation of Arithmetic Expressions, Infix, Prefix, and Postfix Notations, Evaluation of Postfix Expression, Conversion of Expression from Infix to Postfix, Recursion, Queues - Various Positions of Queue, Representation of Queue, Insertion, Deletion, Searching Operations.

Learning Outcomes:

At the end of this unit, the student will be able to

- Describe the operations of Stack. (L2)
- Explain the different notations of arithmetic expression. (L5)
- Develop various operations on Queues. (L6)

Unit – 3

Linked Lists–Pointers, Singly Linked List, Dynamically Linked Stacks and Queues, Polynomials Using Singly Linked Lists, Using Circularly Linked Lists, Insertion, Deletion and Searching Operations, Doubly linked lists and its operations, Circular linked lists and its operations.

Learning Outcomes:

At the end of this unit, the student will be able to

- Analyze various operations on singly linked list. (L4)
- Interpret the operations of doubly linked lists. (L2)
- Apply various operations on Circular linked lists. (L6)

Unit-4

Trees- Tree terminology, representation, Binary tree, representation, Binary tree traversals. Binary Tree Operations, **Graphs**- Graph terminology, Graph representation, Elementary Graph Operations, Breadth first search (BFS) and Depth first search (DFS), Connected Components, Spanning Trees.

Learning Outcomes:

At the end of this unit, the student will be able to

- Develop the representation of Tree. (L3)
- Identify the various Binary tree traversals. (L3)
- Illustrate different Graph traversals like BFS and DFS. (L2)

Unit-5

Searching and Sorting–Sequential, Binary, Exchange (Bubble) Sort, Selection Sort, Insertion Sort, Quick Sort, Merge Sort, Heap Sort. Searching- Linear and Binary Search Methods.

Learning Outcomes:

At the end of this unit, the student will be able to

- Design the different sorting techniques (L6)
- Apply linear search and Binary search methods. (L3)

Text Books:

1. The C Programming Language, Brian W Kernighan and Dennis M Ritchie, Second Edition, Prentice Hall Publication.
2. Fundamentals of Data Structures in C, Ellis Horowitz, Sartaj Sahni, Susan Anderson-Freed, Computer Science Press.
3. Programming in C and Data Structures, J.R.Hanly, Ashok N. Kamthane and A. Ananda Rao, Pearson Education.
4. B.A.Forouzon and R.F. Gilberg, “COMPUTER SCIENCE: A Structured Programming Approach Using C”, Third edition, CENGAGE Learning, 2016.
5. Richard F. Gilberg & Behrouz A. Forouzan, “Data Structures: A Pseudocode Approach with C”, Second Edition, CENGAGE Learning, 2011.

Reference Books:

1. PradiDey and ManasGhosh, Programming in C, Oxford University Press, 2nd Edition 2011.
2. E.Balaguruswamy, "C and Data Structures", 4th Edition, Tata Mc Graw Hill.
3. A.K.Sharma, Computer Fundamentals and Programming in C, 2nd Edition, University Press.
4. M.T.Somashekara, "Problem Solving Using C", PHI, 2nd Edition 2009.

Course Outcomes:

At the end of this unit, the student will be able to

- Analyse the basic concepts of C Programming language. (L4)
- Design applications in C, using functions, arrays, pointers and structures. (L6)
- Apply various operations of Stacks and Queues in solving the problems. (L3)
- Explain operations on Linked lists. (L5)
- Demonstrate various tree traversals and graph traversal techniques. (L2)
- Design searching and sorting methods (L3)

(20F00106) DATABASE MANAGEMENT SYSTEMS

Course objectives:

At the end of this unit, the student will be able to

- Train in the fundamental concepts of database management systems, database modeling and design, SQL, PL/SQL and system implementation techniques.
- Enable students to model ER diagram for any customized application
- Inducting appropriate strategies for optimization of queries.
- Provide knowledge on concurrency techniques
- Demonstrate the organization of Databases

UNIT-I:

Introduction: Database systems applications, Purpose of Database Systems, view of Data, Database Languages, Database Design, Database Engine, Database and Application Architecture, Database Users and Administrators.

Introduction to Relational Model: Structure of Relational Databases, Database Schema, Keys, Schema Diagrams, Relational Query Languages, Relational Algebra

Learning Outcomes:

At the end of this unit, the student will be able to

- Distinguish between Database and File System
- Categorize different kinds of data models
- Define functional components of DBMS

UNIT-II:

Introduction to SQL: Overview of the SQL Query Language, SQL Data Definition, Basic Structure of SQL Queries, Additional Basic Operations, Set Operations, Null Values, Aggregate Functions, Nested Sub-queries, Modification of the Database. **Intermediate SQL:** Joint Expressions, Views, Transactions, Integrity Constraints, SQL Data Types and Schemas, Index Definition in SQL, Authorization.

Advanced SQL: Accessing SQL from a Programming Language, Functions and Procedures, Triggers, Recursive Queries, Advanced Aggregation Features.

Learning Outcomes:

At the end of this unit, the student will be able to

- Outline the elements of the relational model such as domain, attribute , tuple, relation and entity
- Distinguish between various kinds of constraints like domain, key and integrity
- Define relational schema
- Develop queries using Relational Algebra and SQL
- Perform DML operations on databases

UNIT-III:

Database Design and the E-R Model: Overview of the Design Process, The Entity-Relationship Model, Complex Attributes, Mapping Cardinalities, Primary Key, Removing Redundant Attributes in Entity Sets, Reducing E-R Diagrams to Relational Schemas, Extended E-R Features, Entity-Relationship Design Issues, Alternative Notations for Modeling Data, Other Aspects of Database Design.

Relational Database Design:

Features of Good Relational Designs, Decomposition Using Functional Dependencies, Normal Forms, Functional-Dependency Theory, Algorithms for Decomposition using Functional Dependencies, Decomposition Using Multivalued Dependencies, More Normal Forms, Atomic Domains and First Normal Form, Database–Design Process, Modeling Temporal Data, Indexing.

Learning Outcomes:

At the end of this unit, the student will be able to

- Develop E-R model for the given problem
- Derive tables from E-R diagrams
- Differentiate between various normal forms based on functional dependency
- Apply normalization techniques to eliminate redundancy

UNIT-IV:

Query Processing: Overview, Measures of Query cost, Selection Operation, Sorting, Join Operation, Other Operations, Evaluation of Expressions, Query Processing in Memory.

Query optimization: Overview, Transformation of Relational Expressions, Estimating Statistics of Expression Results, Choice of Evaluation Plans, Materialized views, Advanced Topics in Query Optimization.

Learning Outcomes:

At the end of this unit, the student will be able to

- Identify variety of methods for effective processing of given queries.
- Obtain knowledge related to optimization techniques.

UNIT V:

Transaction Management:

Transactions: Transaction Concept, A Simple Transactional Model, Storage Structure, Transaction Atomicity and Durability, Transaction Isolation, Serializability, Transaction Isolation and Atomicity, Transaction Isolation Levels, Implementation of Isolation Levels, Transactions as SQL Statements.

Concurrency Control: Lock-Based Protocols, Deadlock Handling, Multiple Granularity, Insert Operations. Delete Operations and Predicate Reads, Timestamp-Based Protocols, Validation- Based Protocols, Multiversion Schemes, Snapshot Isolation, Weak Levels of Consistency in Practice, Advanced Topics in Concurrency.

Recovery System: Failure Classification, Storage, Recovery and Atomicity, Recovery Algorithm, Buffer Management, Failure with Loss of Non-Volatile Storage, High Availability Using Remote Backup Systems, Early Lock Release and Logical Undo Operations, ARIES, Recovery in Main-Memory Databases.

Learning Outcomes:

At the end of this unit, the student will be able to

- Understand various properties of transaction.
- Design atomic transactions for an application.
- Gain the knowledge about log mechanism and check pointing techniques for system recovery.

TEXT BOOKS:

1. A.Silberschatz, H.F.Korth, S.Sudarshan, “Database System Concepts”, 7/e, TMH 2020

REFERENCE BOOKS:

1. Shamkant B. Navathe, “Database Management System” 6/e RamezElmasri PEA
2. “Database Principles Fundamentals of Design Implementation and Management”, Carlos Coronel, Steven Morris, Peter Robb, Cengage Learning.
3. Raghurama Krishnan, Johannes Gehrke, “Database Management Systems”, 3/e, TMH

Course Outcomes

At the end of this unit, the student will be able to

- Design a database for a real world information system
- Define transactions which preserve the integrity of the database
- Generate tables for a database
- Organize the data to prevent redundancy
- Pose queries to retrieve the information from database

(20F00107) DATA SCIENCE AND PYTHON LABORATORY

Course Objectives:

- To train the students in solving computational problems
- To elucidate solving mathematical problems using Python programming language
- To understand the fundamentals of Python programming concepts and its applications.
- Practical understanding of building different types of models and their evaluation

List of Topics

1. Write a program to demonstrate a) Different numeric data types and b) To perform different Arithmetic Operations on numbers in Python.
2. Write a program to create, append, and remove lists in Python.
3. Write a program to demonstrate working with tuples in Python.
4. Write a program to demonstrate working with dictionaries in Python.
5. Write a program to demonstrate a) arrays b) array indexing such as slicing, integer array indexing and Boolean array indexing along with their basic operations in NumPy.
6. Write a program to compute summary statistics such as mean, median, mode, standard deviation and variance of the given different types of data.
7. Write a script named copyfile.py. This script should prompt the user for the names of two text files. The contents of the first file should be the input that to be written to the second file.
8. Write a program to demonstrate Regression analysis with residual plots on a given data set.
9. 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.
10. Write a program to implement the naïve Bayesian classifier for a sample training data set stored as a .CSV file. Compute the accuracy of the classifier, considering few test data sets.
11. Write a program to implement k-Nearest Neighbour algorithm to classify the iris data set. Print both correct and wrong predictions using Java/Python ML library classes.
12. Write a program to implement k-Means clustering algorithm to cluster the set of data stored in .CSV file. Compare the results of various “k” values for the quality of clustering.
13. Write a program to build Artificial Neural Network and test the same using appropriate data sets.

Textbooks:

1. Francois Chollet, Deep Learning with Python, 1/e, Manning Publications Company, 2017
2. Peter Wentworth, Jeffrey Elkner, Allen B. Downey and Chris Meyers, “How to Think Like a Computer Scientist: Learning with Python 3”, 3rd edition, Available at <http://www.ict.ru.ac.za/Resources/cspw/thinkcspy3/thinkcspy3.pdf>
3. Paul Barry, “Head First Python a Brain Friendly Guide” 2nd Edition, O’Reilly, 2016
4. Dainely.Chen “Pandas for Everyone Python Data Analysis” Pearson Education, 2019

(20F00108) DATA STRUCTURES USING C LABORATORY

Course Objectives:

- To get familiar with the basic concepts of C programming.
- To design programs using arrays, strings, pointers and structures.
- To illustrate the use of Stacks and Queues
- To apply different operations on linked lists.
- To demonstrate the Binary tree traversal techniques.
- To design searching and sorting techniques.

Week 1

Write C programs that use both recursive and non-recursive functions

- i) To find the factorial of a given integer.
- ii) To find the GCD (greatest common divisor) of two given integers.
- iii) To solve Towers of Hanoi problem.

Week 2

- a) Write a C program to find both the largest and smallest number in a list of integers.
- b) Write a C program that uses functions to perform the following:
 - i) Addition of Two Matrices
 - ii) Multiplication of Two Matrices

Week 3

- a) Write a C program that uses functions to perform the following operations:
 - i) To insert a sub-string in to a given main string from a given position.
 - ii) To delete n Characters from a given position in a given string.

Week 4

- a) Write a C program that displays the position or index in the string S where the string T begins, or – 1 if S doesn't contain T.
- b) Write a C program to count the lines, words and characters in a given text.

Week 5

- a) Write a C Program to perform various arithmetic operations on pointer variables.
- b) Write a C Program to demonstrate the following parameter passing mechanisms:
 - i) call-by-value
 - ii) call-by-reference

Week 6

Write a C program that uses functions to perform the following operations:

- i) Reading a complex number
- ii) Writing a complex number
- iii) Addition of two complex numbers
- iv) Multiplication of two complex numbers

(Note: represent complex number using a structure.)

Week 7

Write C programs that implement stack (its operations) using

- i) Arrays
- ii) Pointers

Week 8

Write C programs that implement Queue (its operations) using

- i) Arrays
- ii) Pointers

Week 9

Write a C program that uses Stack operations to perform the following:

- i) Converting infix expression into postfix expression
- ii) Evaluating the postfix expression

Week 10

Write a C program that uses functions to perform the following operations on singly linked list.

- i) Creation
- ii) Insertion
- iii) Deletion
- iv) Traversal

Week 11

Write a C program that uses functions to perform the following operations on Doubly linkedlist.

- i) Creation
- ii) Insertion
- iii) Deletion
- iv) Traversal

Week 12

Write a C program that uses functions to perform the following operations on Circular linkedlist.

- i) Creation
- ii) Insertion
- iii) Deletion
- iv) Traversal

Week 13

Write a C program that uses functions to perform the following:

- i) Creating a Binary Tree of integers

ii) Traversing the above binary tree in preorder, inorder and postorder.

Week 14

Write C programs that use both recursive and non-recursive functions to perform the following searching operations for a Key value in a given list of integers:

- i) Linear search
- ii) Binary search

Week 15

Write a C program that implements the following sorting methods to sort a given list of integers in ascending order

- i) Bubble sort
- ii) Selection sort

Week 16

Write a C program that implements the following sorting methods to sort a given list of integers in ascending order

- i) Insertion sort
- ii) Merge sort
- iii) Quick sort

Text Books:

1. Programming in C and Data Structures, J.R.Hanly, Ashok N. Kamthane and A. Ananda Rao, Pearson Education.
2. B.A.Forouzon and R.F. Gilberg, "COMPUTER SCIENCE: A Structured Programming Approach Using C", Third edition, CENGAGE Learning, 2016.
3. Richard F. Gilberg & Behrouz A. Forouzan, "Data Structures: A Pseudocode Approach with C", Second Edition, CENGAGE Learning, 2011.

Reference Books:

1. Pradip Dey and ManasGhosh, Programming in C, Oxford University Press, 2nd Edition 2011.
2. E.Balaguruswamy, "C and Data Structures", 4th Edition, Tata Mc Graw Hill.
3. A.K.Sharma, Computer Fundamentals and Programming in C, 2nd Edition, University Press.
4. M.T.Somashekara, "Problem Solving Using C", PHI, 2nd Edition 2009.

Course Outcomes

At the end of this unit, the student will be able to

- Demonstrate basic concepts of C programming language. (L2)
- Develop C programs using functions, arrays, structures and pointers. (L6)
- Illustrate the concepts Stacks and Queues. (L2)
- Design operations on Linked lists. (L6)
- Apply various Binary tree traversal techniques. (L3)
- Develop searching and sorting methods. (L6)

(20F00109) DATABASE MANAGEMENT SYSTEMS LABORATORY

Course Objectives:

- To implement the basic knowledge of SQL queries and relational algebra.
- To construct database models for different database applications.
- To apply normalization techniques for refining of databases.
- To practice various triggers, procedures, and cursors using PL/SQL.
- To design and implementation of a database for an organization

Week-1: CREATION OF TABLES

1. Create a table called Employee with the following structure.

Name	Type
Empno	Number
Ename	Varchar2(20)
Job	Varchar2(20)
Mgr	Number
Sal	Number

- a. Add a column commission with domain to the Employee table.
- b. Insert any five records into the table.
- c. Update the column details of job
- d. Rename the column of Employee table using alter command.
- e. Delete the employee whose empno is 19.

2. Create department table with the following structure.

Name	Type
Deptno	Number
Deptname	Varchar2(20)
location	Varchar2(20)

- a. Add column designation to the department table.
- b. Insert values into the table.

- c. List the records of emp table grouped by dept no.
- d. Update the record where dept no is 9.
- e. Delete any column data from the table

3. Create a table called Customer table

Name	Type
Cust name	Varchar2(20)
Cust street	Varchar2(20)
Cust city	Varchar2(20)

- a. Insert records into the table.
- b. Add salary column to the table.
- c. Alter the table column domain.
- d. Drop salary column of the customer table.
- e. Delete the rows of customer table whose ust_city is 'hyd'.
- f. Create a table called branch table.

Name	Type
Branch name	Varchar2(20)
Branch city	Varchar2(20)
asserts	Number

- 4. Increase the size of data type for asserts to the branch.
 - a. Add and drop a column to the branch table.
 - b. Insert values to the table.
 - c. Update the branch name column
 - d. Delete any two columns from the table

5. Create a table called sailor table

Name	Type
Sid	Number
Sname	Varchar2(20)
rating	Varchar2(20)

- a. Add column age to the sailortable.
- b. Insert values into the sailortable.

- c. Delete the row with rating>8.
- d. Update the column details of sailor.
- e. Insert null values into the table.

6. Create a table called reserves table

Name	Type
Boat id	Integer
sid	Integer
day	Integer

- a. Insert values into the reserves table.
- b. Add column time to the reserves table.
- c. Alter the column day data type to date.
- d. Drop the column time in the table.
- e. Delete the row of the table with some condition.

Week-2: QUERIES USING DDL AND DML

1.
 - a. Create a user and grant all permissions to the user.
 - b. Insert the any three records in the employee table and use rollback. Check the result.
 - c. Add primary key constraint and not null constraint to the employee table.
 - d. Insert null values to the employee table and verify the result.
2.
 - a. Create a user and grant all permissions to the user.
 - b. Insert values in the department table and use commit.
 - c. Add constraints like unique and not null to the department table.
 - d. Insert repeated values and null values into the table.
3.
 - a. Create a user and grant all permissions to the user.
 - b. Insert values into the table and use commit.
 - c. Delete any three records in the department table and use rollback.
 - d. Add constraint primary key and foreign key to the table.
4.
 - a. Create a user and grant all permissions to the user.
 - b. Insert records in the sailor table and use commit.
 - c. Add save point after insertion of records and verify save point.
 - d. Add constraints not null and primary key to the sailor table.
5.
 - a. Create a user and grant all permissions to the user.
 - b. Use revoke command to remove user permissions.
 - c. Change password of the user created.
 - d. Add constraint foreign key and no tnull.
6.
 - a. Create a user and grant all permissions to the user.
 - b. Update the table reserves and use save point and rollback.
 - c. Add constraint primary key , foreign key and not null to the reserves table
 - d. Delete constraint not null to the table column

Week-3:QUERIES USING AGGREGATE FUNCTIONS

1.
 - a. By using the group by clause, display the names who belongs to dept no 10 along with average salary.
 - b. Display lowest paid employee details under each department.
 - c. Display number of employees working in each department and their department number.
 - d. Using built in functions, display number of employees working in each department and their department name from dept table. Insert dept name to dept table and insert dept name for each row, do the required thing specified above.
 - e. List all employees which start with either B or C.
 - f. Display only these ename of employees where the maximum salary is greater than or equal to 5000.
2.
 - a. Calculate the average salary for each different job.
 - b. Show the average salary of each job excluding manager.
 - c. Show the average salary for all departments employing more than three people.
 - d. Display employees who earn more than the lowest salary in department 30
 - e. Show that value returned by sign (n)function.
 - f. How many days between day of birth to current date
3.
 - a. Show that two substring as single string.
 - b. List all employee names, salary and 15% rise in salary.
 - c. Display lowest paid emp details under each manager
 - d. Display the average monthly salary bill for each deptno.
 - e. Show the average salary for all departments employing more than two people.
 - f. By using the group by clause, display the eid who belongs to dept no 05 along with average salary.
4.
 - a. Count the number of employees in department20
 - b. Find the minimum salary earned by clerk.
 - c. Find minimum, maximum, average salary of all employees.
 - d. List the minimum and maximum salaries for each job type.
 - e. List the employee names in descending order.
 - f. List the employee id, names in ascending order by empid.
5.
 - a. Find the sids ,names of sailors who have reserved all boats called "INTERLAKE
Find the age of youngest sailor who is eligible to vote for each rating level with at least two such sailors.
 - b. Find the sname , bid and reservation date for each reservation.
 - c. Find the ages of sailors whose name begin and end with B and has at least 3characters.
 - d. List in alphabetic order all sailors who have reserved red boat.
 - e. Find the age of youngest sailor for each rating level.
6.
 - a. List the Vendors who have delivered products within 6 months from or derdate.
 - b. Display the Vendor details who have supplied both Assembled and Subparts.
 - c. Display the Sub parts by grouping the Vendor type (Local or Non Local).

- d. Display the Vendor details in ascending order.
- e. Display the Sub part which costs more than any of the Assembled parts.
- f. Display the second maximum cost Assembled part

Week-4: PROGRAMS ON PL/SQL

1. a. Write a PL/SQL program to swaptwonumbers.
b. Write a PL/SQL program to find the largest of three numbers.
2. a. Write a PL/SQL program to find the total and average of 6 subjects and display the grade.
b. Write a PL/SQL program to find the sum of digits in a given number.
3. a. Write a PL/SQL program to display the number in reverse order.
b. Write a PL/SQL program to check whether the given number is prime or not.
4. a. Write a PL/SQL program to find the factorial of a given number.
b. Write a PL/SQL code block to calculate the area of a circle for a value of radius varying from 3 to 7. Store the radius and the corresponding values of calculated area in an empty table named areas, consisting of two columns radius and area.
5. a. Write a PL/SQL program to accept a string and remove the vowels from the string. (When 'hello' passed to the program it should display 'Hll' removing e and o from the word Hello).
b. Write a PL/SQL program to accept a number and a divisor. Make sure the divisor is less than or equal to 10. Else display an error message. Otherwise Display the remainder in words.

Week-5: PROCEDURES AND FUNCTIONS

1. Write a function to accept employee number as parameter and return Basic +HRA together as single column.
2. Accept year as parameter and write a Function to return the total net salary spent for a given year.
3. Create a function to find the factorial of a given number and hence find NCR.
4. Write a PL/SQL block to print prime Fibonacci series using local functions.
5. Create a procedure to find the lucky number of a given birth date.
6. Create function to the reverse of given number

Week-6: TRIGGERS

1. Create a row level trigger for the customers table that would fire for INSERT or UPDATE or DELETE operations performed on the CUSTOMERS table. This trigger will display the salary difference between the old values and new values:

CUSTOMERS table:

ID	NAME	AGE	ADDRESS	SALARY
1	Alive	24	Khammam	2000
2	Bob	27	Kadappa	3000
3	Catri	25	Guntur	4000

4	Dena	28	Hyderabad	5000
5	Eeshwar	27	Kurnool	6000
6	Farooq	28	Nellur	7000

- Creation of insert trigger, delete trigger, update trigger practice triggers using the passenger database.
Passenger(Passport_ id INTEGER PRIMARY KEY, Name VARCHAR (50) NotNULL, Age Integer Not NULL, Sex Char, Address VARCHAR (50) NotNULL);
 - Write a Insert Trigger to check the Passport_id is exactly six digits ornot.
 - Write a trigger on passenger to display messages '1 Record is inserted', '1 record is deleted', '1 record is updated' when insertion, deletion and updation are done on passengerrespectively.
- Insert row in employee table using Triggers. Every trigger is created with name any trigger have same name must be replaced by new name. These triggers can raised before insert, update or delete rows on data base. The main difference between a trigger and a stored procedure is that the former is attached to a table and is only fired when an INSERT, UPDATE or DELETEoccurs.
- Convert employee name into uppcase whenever an employee record is inserted or updated. Trigger to fire before the insert orupdate.
- Trigger before deleting a record from emp table. Trigger will insert the row to be deleted into table called delete _emp and also record user who has deleted the record and date and time ofdelete.
- Create a transparent audit system for a table CUST_MSTR. The system must keep track of the records that are being deleted or updated

Week-7: PROCEDURES

- Create the procedure for palindrome of given number.
- Create the procedure for GCD: Program should load two registers with two Numbers and then apply the logic for GCD of two numbers. GCD of two numbers is performed by dividing the greater number by the smaller number till the remainder is zero. If it is zero, the divisor is the GCD if not the remainder and the divisors of the previous division are the new set of two numbers. The process is repeated by dividing greater of the two numbers by the smaller number till the remainder is zero and GCD isfound.
- Write the PL/SQL programs to create the procedure for factorial of givennumber.
- Write the PL/SQL programs to create the procedure to find sum of N naturalnumber.
- Write the PL/SQL programs to create the procedure to find Fibonacciseries.
- Write the PL/SQL programs to create the procedure to check the given number is perfect ornot

Week-8: CURSORS

1. Write a PL/SQL block that will display the name, dept no, salary of fist highest paid employees.
2. Update the balance stock in the item master table each time a transaction takes place in the item transaction table. The change in item master table depends on the item id is already present in the item master then update operation is performed to decrease the balance stock by the quantity specified in the item transaction in case the item id is not present in the item master table then the record is inserted in the item master table.
3. Write a PL/SQL block that will display the employee details along with salary using cursors.
4. To write a Cursor to display the list of employees who are working as a Managers or Analyst.
5. To write a Cursor to find employee with given job and dept no.
6. Write a PL/SQL block using implicit cursor that will display message, the salaries of all the employees in the 'employee' table are updated. If none of the employee's salary are updated we get a message 'None of the salaries were updated'. Else we get a message like for example, 'Salaries for 1000 employees are updated' if there are 1000 rows in 'employee' table

Week-9: CASE STUDY: BOOK PUBLISHING COMPANY

A publishing company produces scientific books on various subjects. The books are written by authors who specialize in one particular subject. The company employs editors who, not necessarily being specialists in a particular area, each take sole responsibility for editing one or more publications.

A publication covers essentially one of the specialist subjects and is normally written by a single author. When writing a particular book, each author works with on editor, but may submit another work for publication to be supervised by other editors. To improve their competitiveness, the company tries to employ a variety of authors, more than one author being a specialist in a particular subject for the above case study, do the following:

1. Analyze the data required.
2. Normalize the attributes.

Create the logical data model using E-R diagrams

Week-10: CASE STUDY GENERAL HOSPITAL

A General Hospital consists of a number of specialized wards (such as Maternity, Pediatric, Oncology, etc). Each ward hosts a number of patients, who were admitted on the recommendation of their own GP and confirmed by a consultant employed by the Hospital. On admission, the personal details of every patient are recorded. A separate register is to be held to store the information of the tests undertaken and the results of a prescribed treatment. A number of tests may be conducted for each patient. Each patient is assigned to one leading consultant but may be examined by another doctor, if required. Doctors are specialists in some branch of medicine and may be leading consultants for a number of patients, not necessarily from the same ward. For the above case study, do the following.

1. Analyze the data required.
2. Normalize the attributes.

Create the logical data model using E-R diagrams

Week-11: CASE STUDY: CAR RENTAL COMPANY

A database is to be designed for a car rental company. The information required includes a description of cars, subcontractors (i.e. garages), company expenditures, company revenues and customers. Cars are to be described by such data as: make, model, year of production, engine size, fuel type, number of passengers, registration number, purchase price, purchase date, rent price and insurance details. It is the company policy not to keep any car for a period exceeding one year. All major repairs and maintenance are done by subcontractors (i.e. franchised garages), with whom CRC has long-term agreements. Therefore the data about garages to be kept in the database includes garage names, addresses, range of services and the like. Some garages require payments immediately after a repair has been made; with others CRC has made arrangements for credit facilities. Company expenditures are to be registered for all outgoings connected with purchases, repairs, maintenance, insurance etc. Similarly the cash inflow coming from all sources: Car hire, car sales, insurance claims must be kept of file. CRC maintains a reasonably stable client base. For this privileged category of customers special credit card facilities are provided. These customers may also book in advance a particular car. These reservations can be made for any period of time up to one month. Casual customers must pay a deposit for an estimated time of rental, unless they wish to pay by credit card. All major credit cards are accepted. Personal details such as name, address, telephone number, driving license, number about each customer are kept in the database. For the above case study, do the following:

1. Analyze the data required.
2. Normalize the attributes.

Create the logical data model using E-R diagrams

Week-12: CASE STUDY: STUDENT PROGRESS MONITORING SYSTEM

A database is to be designed for a college to monitor students' progress throughout their course of study. The students are reading for a degree (such as BA, BA (Hons) M.Sc., etc) within the framework of the modular system. The college provides a number of modules, each being characterized by its code, title, credit value, module leader, teaching staff and the department they come from. A module is coordinated by a module leader who shares teaching duties with one or more lecturers. A lecturer may teach (and be a module leader for) more than one module. Students are free to choose any module they wish but the following rules must be observed: Some modules require pre-requisites modules and some degree programmes have compulsory modules. The database is also to contain some information about students including their numbers, names, addresses, degree they read for, and their past performance i.e. modules taken and examination results. For the above case study, do the following:

1. Analyze the data required.
2. Normalize the attributes.
3. Create the logical data model i.e., ER diagrams.
4. Comprehend the data given in the case study by creating respective tables with primary keys and foreign keys wherever required.
5. Insert values into the tables created (Be vigilant about Master- Slave tables).
6. Display the Students who have taken M.Sc course

7. Display the Module code and Number of Modules taught by each Lecturer.
8. Retrieve the Lecturer names who are not Module Leaders.
9. Display the Department name which offers 'English' module.
10. Retrieve the Prerequisite Courses offered by every Department (with Department names).
11. Present the Lecturer ID and Name who teaches 'Mathematics'.
12. Discover the number of years a Module is taught.
13. List out all the Faculties who work for 'Statistics' Department.
14. List out the number of Modules taught by each Module Leader.
15. List out the number of Modules taught by a particular Lecturer.
16. Create a view which contains the fields of both Department and Module tables. (Hint- The fields like Module code, title, credit, Department code and its name).
17. Update the credits of all the prerequisite courses to 5. Delete the Module 'History' from the Module table.

Reference Books:

1. Ramez Elmasri, Shamkant, B. Navathe, "Database Systems", Pearson Education, 7th Edition, 2020.
2. Peter Rob, Carles Coronel, "Database System Concepts", Cengage Learning, 7th Edition, 2008.

Web References:

<http://www.scoopworld.in>

SOFTWARE AND HARDWARE REQUIREMENTS FOR A BATCH OF 24 STUDENTS:

HARDWARE: Desktop Computer Systems: 24 nos

SOFTWARE: Oracle 11g.

Course Outcomes

At the end of this unit, the student will be able to

- Design database for any real world problem
- Implement PL/SQL programs
- Define SQL queries
- Decide the constraints
- Investigate for data inconsistency