SCHEME OF INSTRUCTION
MASTER OF COMPUTER APPLICATIONS (MCA)
SEMESTER – II

SNo	Course Code	Course Title	Hours/ Week			Scheme of Examination				No of Credits
						Max Marks			Duration (hrs)	
THEORY			L	T	P	CIE	SEE	Total Marks	SEE	Cr
1	PCC 201	Operating Systems	4	-	-	30	70	100	3	4
2	PCC 202	Database Management System	4	-	-	30	70	100	3	4
3	PCC 203	Design and Analysis of Algorithms	3	1	-	30	70	100	3	4
4 *	PCC 204	Data Engineering with Python	4	-	-	30	70	100	3	4
5	PCC 205	Machine Learning	3	-	-	30	70	100	3	3
6	MGC 206	Operations Research	3	-	-	30	70	100	3	3
PRACTICALS										
7	LCC 251	Operating Systems Lab	-	-	3	25	50	75	3	1.5
8 *	LCC 252	Data Engineering with Pytl	-	-	3	25	50	75	3	1.5
9	LCC 253	Database Management Systems Lab	-	-	3	25	50	75	3	1.5
10	SIP 321	Summer Internship/ Mini Project*	-	-	-	-	-	-	-	-
			20	3	9	255	570	825	-	26.5

***Summer Internship**: After second semester, the students are expected to do summer internship/ Mini Project and Its grade will be credited in the third semester memo after evaluation.

Abbreviation	Full Form	Abbreviation	Full Form
PCC	Professional Core Course	CIE	Continuous Internal Evaluation
PEC	Professional Elective Course	SEE	Semester End Evaluation
HSC	Humanities and Social Science Course	L	Lecture
LCC	Laboratory Core Course	P	Practical

Note : Each lab should be made with 30 students for batch

PCC201 Operating Systems

Credits : 4

Instruction	4 L hrs per week	Duration of SEE	3 hours
CIE	30 marks	SEE	70 marks

Course Objectives

1. To gain the understanding of operating system and unix operating system in specific
2. To comprehend the details of process.
3. To learn the types and architecture of computer memory
4. To study file system and its implementation
5. To realize the operating system concepts into case studies.

Course Outcomes – Learners on completion of the course, be able to

1. Explain operating systems and Unix OS, illustrate the workings of various OS components.
2. Analyze the process, its states and process scheduling algorithms.
3. Demonstrate paging, demand paging, page replacement and segmentation with illustrations.
4. Elaborate the file access and allocation methods and mass storage structures.
5. Describe concrete implementations of Linux system and Windows 7.

UNIT-I

Unix: Introduction, commands, file system, security and file permission, regular expression and grep, shell programming, awk.
Introduction to Operating Systems: OS structure and strategies, Process concepts, Multithreaded Programming, Process scheduling, Process synchronization, Deadlocks.

UNIT-II

Memory management strategies with example architectures: Swapping, Contiguous allocation, Paging, Segmentation, Segmentation with paging, Virtual memory management : Demand paging, Page replacement, Thrashing.

UNIT-III

File System Interface: File concepts, Access methods and protection. File system implementation: File system structure, Allocation methods, Directory implementation of file systems, Mass storage structures, I/O systems

UNIT-IV

System Protection : Principles and Domain, Access Matrix and implementation, Access control and access rights, Capability based systems, Language based Protection,
System Security: Problem, Program threats, cryptography, user authentication, implementing security defenses, Firewalling, Computer security Classification

UNIT-V

Case Studies: The Linux System–Design principles, Kernel modules, Process management, Scheduling, Memory management, File systems, Input and Output, Inter process

communication. Windows 7 –Design principles, System components, Terminal services and fast user switching File systems, Networking, Programmer interface.

Suggested Readings:

1. Abraham Silberschatz, Peter B Galvin, Operating System Concepts, 9th edition, Wiley, 2016
2. William Stallings, Operating Systems-Internals and Design Principles, 8th edition, Pearson, 2014
3. Andrew S Tanenbaum, Modern Operating Systems, 4th edition, Pearson, 2016.

PCC202 Database Management System

Credits : 4

Instruction	4L hrs per week	Duration of SEE	3 hours
CIE	30 marks	SEE	70 marks

Course Objectives

1. Introduce database concepts along with ER modelling
2. Learn about relational databases and SQL query language
3. Define advanced SQL
4. Study DB transactions and explore concurrency concepts
5. Introduce NoSQL

Course Outcomes

1. Explain the DB concepts and model requirements as ER-model
2. Suggest relational algebra queries from text specification
3. Write SQL queries for the given questions
4. Elaborate indexing and hashing and describe concurrency control concepts
5. Comprehend NoSQL technology

UNIT – I

Introduction: Database System Applications, Purpose of Database Systems, View of Values, Nested Sub-queries, Complex Queries, Views, Modification of the Database, Joined Relations Data, Database Languages, Relational Databases, Database Design, Object–based and Semi-structured Databases, Data Storage and Querying, Transaction Management, Data Mining and Analysis, Database Architecture, Database Users and Administrators. Database Design and the **E-R Model**: Overview of the Design Process, The Entity- Relationship Model, Constraints, Entity-Relationship Diagrams, Entity – Relationship Design Issues, Weak Entity Sets, Extended E-R Features, Database Design for Banking Enterprise, Reduction to Relational Schemas, Other Aspects of Database Design

UNIT – II

Relational Model: Structure of Relational Databases, Fundamental Relational-Algebra Operations, Additional Relational – Algebra Operations, Extended Relational - Algebra Operations, Null Values, Modification of the Databases. Structured Query Language: Data Definition, Basic Structure of SQL Queries, Set Operations, Aggregate Functions, Null

UNIT – III

Advanced SQL: SQL Data Types and Schemas, Integrity Constraints, Authorization, Embedded SQL, Dynamic SQL, Functions and Procedural Constructs, Recursive Queries, Advanced SQL Features. Relational Database Design: Features of Good Relational Design, Atomic Domains and First Normal Form, Functional-Dependency Theory, Decomposition using Functional Dependencies.

UNIT – IV

Indexing and Hashing: Basic Concepts, Ordered Indices, B+-tree Index Files, B-tree Index Files, Multiple-Key Access, Static Hashing, Dynamic Hashing, Comparison of Ordered Indexing and Hashing, Bitmap Indices. Index Definition in SQL Transactions: Transaction Concepts, Transaction State, Implementation of Atomicity and Durability, Concurrent Executions, Serializability, Recoverability, Implementation of Isolation, Testing for Serializability

UNIT – V

Concurrency Control: Lock-based Protocols, Timestamp-based Protocols, Validation-based Protocols, Multiple Granularity, Multi-version Schemes, Deadlock Handling, Insert and Delete Operations, Weak Levels of Consistency, Concurrency of Index Structures. Recovery System: Failure Classification, Storage Structure, Recovery and Atomicity, Log-Based Recovery, Recovery with Concurrent Transactions, Buffer Management, Failure with Loss of Nonvolatile Storage, Advanced Recovery Techniques, Remote Backup Systems

NoSQL: Need for NoSQL, aggregate data models, more details on data models, distribution models, consistency, version stamps, map-reduce, key-value databases, document databases, column-family stores, graph databases, Schema Migrations

Suggested Readings

1. Abraham Silberschatz, Henry F Korth, S Sudarshan, Database System Concepts, McGraw-Hill International Edition, 6th Edition, 2010.
2. Ramakrishnan, Gehrke, Database Management Systems, McGraw-Hill International Edition, 3rd Edition, 2003.
3. Elmasri, Navathe, Somayajulu, Fundamentals of Database Systems, Pearson Education,4thEdition,2004.
4. Shashank Tiwari, “Professional NoSQL”, 1st Edition , Wiley publishers, 2011.

PCC203 Design and Analysis of Algorithms

Credits : 4

Instruction	4(3L+1T) hrs per week	Duration of SEE	3 hours
CIE	30 marks	SEE	70 marks

Course Objectives

1. Learn algorithms time complexity
2. Learn divide and conquer approach
3. Learn greedy method
4. Learn dynamic programming
5. Learn backtracking

Course Outcomes

1. Carry out algorithms time complexity
2. Explain divide and conquer approach
3. Illustrate greedy method
4. Elaborate dynamic programming
5. Explore backtracking

Unit I

Introduction to Algorithms: Algorithm Specification, Performance Analysis, Randomized Algorithms. **Elementary Data Structures**: Stacks and Queues, Trees, Dictionaries, Priority Queues, Sets and Disjoint Set Union, Graphs.

Unit II

Divide and Conquer: Binary Search, Finding the Maximum and Minimum, Merge Sort; Quick Sort, Selection sort, Strassen's Matrix Multiplication, Convex Hull.

The Greedy Method: Knapsack Problem, Tree Vertex Splitting, Job Sequencing with Deadlines, Minimum-Cost Spanning Trees, Single Source Shortest Paths.

Unit III

Dynamic Programming: General Method, Multistage Graphs, All-Pairs Shortest Paths, Single-Source Shortest Paths, Optimal Binary Search Trees, 0/1 Knapsack, The Traveling Salesperson Problem.

Basic Traversal and Search Techniques: Techniques for Binary Trees, Techniques for Graphs, Connected Components and Spanning Trees, Biconnected Components and DFS.

Unit IV

Back Tracking: General Method, 8-Queens Problem, Sum of Subsets, Graph Coloring, Hamiltonian Cycles, Knapsack Problem. **Branch-Bound**: The Method, 0/1 Knapsack Problem, Traveling Sales Person.

Unit V

NP-Hard and NP-Complete Problems: Basic Concepts, Cook's Theorem, NP-Hard. Graph Problems, NP-Hard Scheduling Problems, NP-Hard Code Generation, Some Simplified NP-Hard Problems.

Suggested Readings

1. E Horowitz, S Sahni, S Rajasekaran, "Fundamentals of Computer Algorithms", Second Edition, Universities Press, 2007.
2. R. Pannerselvam, "Design and Analysis of Algorithms", PHI, 2007.
3. Hari Mohan Pandey, "Design, Analysis and Algorithm", University Science Press, 2009.
2. TH Cormen, CE Leiserson, RL Rivert, C Stein, "Introduction to Algorithms", Third Edition, PHI, 2010.

PCC 204 # Data Engineering with Python

Instruction	3 Periods per week
Duration of University Examination	4 Hours
University Examination	70 Marks
Sessional	30 Marks
Credits	4

Course Objectives: The main objectives of this course are to teach

- how to extract raw data,
- clean the data,
- perform transformations on data,
- load data and visualize the data

Outcomes:

At the end of the course the student will be able to:

- Understand the basics of Python Programming Language
- Handle different types of files and work with text data
- Use regular expression operations
- Use relational databases via SQL
- Use tabular numeric data
- Use the data structures: data series and frames
- Use PyPlot for visualization
- Use Python for basic Machine Learning

Unit – I

Introduction, Parts of Python Programming Language, Control Flow Statements, Functions, Strings [Reference 2 – Chapter 1 to Chapter 5]

Unit- II

Lists, Dictionaries, Tuples and sets, Files, Regular expressions [Reference 2- Chapter 6 to Chapter 10]

Unit-III

Introduction to Data Science [Reference 2- Chapter 12] **, Data Science**: Data Analysis Sequence, Data Acquisition Pipeline, Report Structure [Reference 1(Chapter 1-Unit1 to Unit 3)]]
Files and Working with Text Data: Types of Files, Creating and Reading Text Data, File Methods to Read and Write Data, Reading and Writing Binary Files, The Pickle Module, Reading and Writing CSV Files, Python os and os.pathModules. [Reference 2, Chapter 9)]
Working with Text Data: JSON and XML in Python[Reference 2, Section12.2]

Working with Text Data: Processing HTML Files, Processing Texts in Natural Languages [Reference 1(Chapter3 –Unit 13, and Unit16)

Regular Expression Operations: Using Special Characters, Regular Expression Methods, Named Groups in Python Regular Expressions, Regular Expression with *glob* Module [Reference 2-Chapter 10]

Unit – IV

Working with Databases: Setting Up a MySQL Database, Using a MySQL Database: Command Line, Using a MySQL Database, Taming Document Stores: MongoDB [Reference 1 (Chapter4-Unit17toUnit20)]

Working with Data Series and Frames: Pandas Data Structures, Reshaping Data, Handling Missing Data, Combining Data, Ordering and Describing Data, Transforming Data, Taming Pandas File I/O [Reference 1 (Chapter 6-Unit 31 to Unit 37)]

Plotting: Basic Plotting with PyPlot, Getting to Know Other Plot Types, Mastering Embellishments, Plotting with Pandas [Reference 1 (Chapter8-Unit 41 to Unit 44)]

Unit – V

Probability and Statistics: Reviewing Probability Distributions, Recollecting Statistical measures, Doing Stats the Python way [Reference 1 (Chapter9-Unit 45 to Unit 47)]

Machine Learning: Designing a Predictive Experiment, Fitting a linear regression, Grouping Data with K- means Clustering. Surviving in Random Decision Forests. [Reference 1(Chapter 10 - Unit 48 to Unit-51)]

Suggested Reading

1. Data Science Essentials in Python: Collect, Organize, Explore, Predict, Value. Dmitry Zinoriev, The Pragmatic Programmers LLC, 2016
2. Introduction to Python Programming**.** Gowrishankar S., Veena A. CRC Press, Taylor & Francis Group, 2019
3. Python for Everybody: Exploring Data Using Python 3. Charles R Severance, 2016
4. Python Data Analytics – Data Analysis and Science using Pandas, matplotlib and the Python Programming Language. Fabio Nelli, Apress, 2015
5. Website Scraping with Python. Using BeautifulSoup and Scrapy. GáborLászlóHajba, Apress, 2018
6. Machine Learning with Python Cookbook:.Practical Solutions from Preprocessing to Deep Learning. Chris Albon, O'Reilly 2018

PCC205 Machine Learning

Credits : 3

Instruction	3L hrs per week	Duration of SEE	3 hours
CIE	30 marks	SEE	70 marks

Course Objectives

1. Learn regression techniques
2. Learn dimensionality reduction methods
3. Learn classification schemes
4. Learn clustering mechanisms
5. Learn evaluation metrics

Course Outcomes

1. Solve regression problems
2. Apply dimensionality reduction methods
3. Analyze classification schemes
4. Explore clustering mechanisms
5. Explain evaluation metrics

Unit I

Basic Maths: Probability, Linear Algebra, Convex Optimization **Background**: Statistical Decision Theory, Bayesian Learning (ML, MAP, Bayes estimates, Conjugate priors)

Unit II

Regression: Linear Regression, Ridge Regression, Lasso **Dimensionality Reduction**: Principal Component Analysis, Partial Least Squares

Unit III

Classification: Linear Classification, Logistic Regression, Linear Discriminant Analysis, Quadratic Discriminant Analysis, Perceptron, Support Vector Machines + Kernels, Artificial Neural Networks + Back Propagation, Decision Trees, Bayes Optimal Classifier, Naive Bayes.

Unit IV

Evaluation measures: Hypothesis testing, Ensemble Methods, Bagging, Adaboost Gradient Boosting, Clustering, K-means, K-medoids, Density-based Hierarchical, Spectral

Unit V

Miscellaneous topics: Expectation Maximization, GMMs, Learning theory

Introduction to Reinforcement Learning **Graphical Models**: Bayesian Networks.
Use Cases of various ML Algorithms in Manufacturing, Retail, Transport, Healthcare, weather, insurance sectors.

Suggested Readings

1. Ethem Alpaydin. Introduction to Machine Learning 3e(Adaptive Computation and Machine Learning Series). The MIT Press, 2004.
2. Tom M. Mitchell, Machine Learning McGraw Hill Education, 2013

MGC206 Operations Research

Credits : 3

Instruction	3L hrs per week...	Duration of SEE	3 hours
CIE	30 marks	SEE	70 marks

Course Objectives

1. Learn linear programming
2. Learn transportation problem
3. Learn assignment problem
4. Learn dynamic programming
5. Learn gaming theory

Course Outcomes

1. Solve linear problems
2. Apply transportation problems
3. Analyze assignment problems
4. Explore dynamic programming
5. Explain gaming theory

UNIT I

Linear Programming: Introduction, Concept of Linear Programming Model, Development of LP models, Graphical Method, Linear Programming Methods, Special cases of Linear Programming, Duality, Sensitivity Analysis.

UNIT II

Transportation Problem: Introduction, Mathematical Model for 'Transportation Problem, Types of Transportation Problem, Methods to solve Transportation Problem, Transshipment Model.

UNIT III

Assignment Problem: Introduction, Zero-One Programming Model, Types of Assignment Problem, Hungarian Method, Branch-and-Bound Technique for Assignment Problem.

Integer Programming: Introduction, Integer Programming Formulations, The Cutting-Plane Algorithm, Branch-and-Bound Technique, Zero-One Implicit Enumeration Algorithm.

UNIT IV

Dynamic Programming: Introduction, Applications of Dynamic Programming, Solution of Linear Programming Problem through Dynamic Programming.
Basics of Queuing theory.

UNIT V

Game Theory: Introduction, Game with Pure Strategies, Game with Mixed Strategies, Dominance Property, Graphical Method for 2 x n or m x 2 Games, Linear Programming Approach for Game Theory.

Suggested Reading:

1. Pannarselvam, "*Operations Research*", 3rd Edition, PHI, 2009.
2. Prem Kumar Gupta, DS Hira, "*Problems in Operations Research*", S. Chand, 2010.
3. Rathindra P Sen, "*Operations Research - Algorithm and Application*", PHI, 2010.
4. J K Sharma, "*Operations Research*", Fourth Edition, MacMillan, 2009.

LCC251 Operating Systems Lab

Credits : 1.5

Instruction	3P hrs per week	Duration of SEE	3 hours
CIE	25 marks	SEE	50 marks

Course Objectives

1. Learn shell commands and scripting
2. Learn CPU scheduling algorithms
3. Learn memory management algorithms
4. Learn synchronization problems
5. Explore file allocation strategies and disk scheduling algorithms

Course Outcomes

1. Be able to execute shell commands and write shell scripts
2. Be able to write programs on CPU scheduling
3. Be able to create memory management algorithms
4. Be able to execute programs to demonstrate synchronization problems
5. Be able to implement file allocation methods and be able to create disk scheduling algorithms

Programs

1. Unix Shell Commands
    a) File handling commands
    b) Directory handling commands
    c) General purpose commands
2. Unix Shell Scripts
    a) Print Multiplication table of a given no. using all loops
    b) Perform all arithmetic operations
    c) Print the type of a file
    d) Rename all files whose names end with .c as .old
    e) Display the no. of lines in each of text file in a given dir
3. Simulate the following CPU scheduling algorithms.
    a. FCFS
    b. SJF
    c. Round Robin
    d. Priority.
4. Write a C program to simulate producer-consumer problem using Semaphores
5. Write a C program to simulate the concept of Dining-philosophers problem.
6. Simulate MVT and MFT.
7. Write a C program to simulate the following contiguous memory allocation techniques

   a. Worst fit
   b. Best fit
   c. First fit.
8. Simulate following page replacement algorithms
   a. FIFO
   b. LRU
   c. OPTIMAL
9. Simulate following File Organization Techniques
   a. Single level directory
   b. Two level directory
10. Simulate following file allocation strategies
   a. Sequential
   b. Indexed
   c. Linked.
11. Simulate Bankers Algorithm for Dead Lock Avoidance.
12. Simulate Bankers Algorithm for Dead Lock Prevention.
13. Write a C program to simulate disk scheduling algorithms.
   a. FCFS
   b. SCAN
   c. C-SCAN

LCC252 Data Engg. With Python Lab

Credits : 1.5

Instruction	3P hrs per week	Duration of SEE	3 hours
CIE	25 marks	SEE	50 marks

Course objectives:

- Understand the process of Importing and Exporting the data.
- Learn how to collect, store and manage data from multiple data sources.
- Know the insights of data using statistical methods
- Identify different techniques for data analysis and data visualization.
- Put into practice the ETL (extract, transform, load) pipeline which will extract raw data, clean the data, perform transformations on data, load data and visualize the data.

Course Outcomes: students would be able to:

- Demonstrate various data types in python and develop programs using files, exception handling, functions, classes in Python.
- Examine the process for importing and exporting the data.
- Apply appropriate data collection and pre-processing methods.
- Identify different data analysis Techniques suitable for a given applications
- Demonstrate data visualization techniques for Data Analysis.

Libraries
In this course students are expected to extract, transform and load input data that can be text files, CSV files, XML files, JSON, HTML files, SQL databases, NoSQL databases etc.,. For doing this, they should learn the following Python libraries/modules:
pandas, numpy, BeautifulSoup, pymysql, pymongo, nltk, matplotlib

Datasets
For this laboratory, appropriate publicly available datasets, can be studied and used. Example:
MNIST (http://yann.lecun.com/exdb/mnist/),
UCI Machine Learning Repository(https://archive.ics.uci.edu/ml/datasets.html),
Kaggle(https://www.kaggle.com/datasets)
Twitter Data

Exercises

1. Write programs to parse text files, CSV, HTML, XML and JSON documents and extract relevant data. After retrieving data check any anomalies in the data, missing values etc.
2. Write programs for reading and writing binary files
3. Write programs for searching, splitting, and replacing strings based on pattern matching using regular expressions
4. Design a relational database for a small application and populate the database. Using SQL do the CRUD (create, read, update and delete) operations.
5. Create a Python MongoDB client using the Python module pymongo. Using a collection object practice functions for inserting, searching, removing, updating, replacing, and aggregating documents, as well as for creating indexes

6. Write programs to create numpy arrays of different shapes and from different sources, reshape and slice arrays, add array indexes, and apply arithmetic, logic, and aggregation functions to some or all array elements
7. Write programs to use the pandas datastructures: Frames and series as storage containers and for a variety of data-wrangling operations, such as:
    - Single-level and hierarchical indexing
    - Handling missing data
    - Arithmetic and Boolean operations on entire columns and tables
    - Database-type operations (such as merging and aggregation)
    - Plotting individual columns and whole tables
    - Reading data from files and writing data to files

Additional Exercises (for learning and practice) :

1. Introduction to Python Programming:
    A. Running instructions in Interactive interpreter and a Python Script.
    B. Write a program to purposefully raise Indentation Error and Correct it
    C. Write a program to compute distance between two points taking input from the user
    D. Write a program add python that takes 2numbers as command line arguments and prints its sum.
    E. Program to display the following information: Your name, Full Address, Mobile Number, College Name, Course Subjects
    F. Write a Program for checking whether the given number is a even number or not.

2 Control Structures, Lists
    A. Program to find the largest three integers using if-else
    B. Program that receives a series of positive numbers and display the numbers in order and their sum
    C. Program to find the product of two matrices and
    D. Program to display two random numbers that are to be added, the program should allow the student to enter the answer.
    E. If the answer is correct, a message of congratulations should be displayed.
    F. If the answer is incorrect, the correct answer should be displayed.
    G. Using a for loop, write a program that prints out the decimal equivalents of 1/2,1/3,1/4, .1/10.
    H. Write a program using a while loop that asks the user for a number, and prints a countdown from that number to zero.

3 Functions andRecursion
    A. Write recursive and non-recursive functions for the following
    B. To find GCD of two integers
    C. To find the factorial of positive integer
    D. To print Fibonacci Sequence up to given number n
    E.To display prime number from2 to n.
    F. Function that accepts two arguments: a list and a number n. It displays all of the numbers in the list that are greater than n
    G. Functions that accept a string as an argument and return the number of vowels and consonants that the string contains

4 Files, Exceptions, Lists, Sets, Random Numbers
    A. Program to write a series of random numbers in a file from 1 to nanddisplay.
    B. Program to write the content in a file and display it with a line number followed by a colon
    C. Program to display a list of all unique words in a textfile
    D.Program to analyse the two text files using set operations
    E. Write a program to print each line of a file in reverse order.
    F. Write a program to count frequency of characters in a given file. Can you use character frequency total whether the given file is a Python program file, C program file or a text file?
    G. Write a program combine lists that combines these lists in to a dictionary.

5 Object Oriented Programming
A .Program to implement the inheritance
B. Program to implement the polymorphism

6. Demonstrate data analysis using NumPy
   a. Create an array of 10 zeros
   b. Create an array of even integers upto 50
   c. Create a 3x3 matrix
   d. Generate an array of 25 random numbers sampled from a standard normal distribution.
   e. Create an array of 20 linearly spaced points between 0 and 1
   f. Demonstrate slicing and indexing operations
   g. Get the sum of all columns in matrix

7. Write a Program in Python to create and combine student and subject data frames in Pandas.

8. Create a data frame 'Book' that contains three vectors [Name, Price, Author]. Convert this data frame into a matrix and list the object using the operator 'as'.

9. Performing Exploratory data analysis on web scraped data of 2021-22 NBA player stats (http://www.basketball-reference.com/)
   Perform data cleaning
   Handle missing values by replacing with 0
   Write to CSV file
   Which player scored the most points per game?
   Which player had the highest 3-point field goals per game?
   Demonstrate Group By() function

10. Data visualization through Sea born for the above program 9.
    Box plot of points scored grouped by position
    Compute the correlation matrix

11. To determine the mean of a set of numbers. To plot the numbers in a bar plot and have a straight line run through the plot at the mean.

12. To determine the median of a set of numbers. To plot the numbers in a barplot and have a straight line run through the plot at the median.

13. To determine the standard deviation. To plot the numbers in a bar plot and have a straight line run through the plot at the mean and another straight line run through the plot at mean + standard deviation.

More dataset to perform data analysis

Source of the Data: https://www.kaggle.com/chirin/africa-economic-banking-and-systemic-crisis-data/downloads/africa-economic-banking-and-systemic-crisis-data.zip/1

Data set: https://www.kaggle.com/khalidative/crimeanalysis

LCC253 Database Management Systems Lab

Credits : 1.5

Instruction	3P hrs per week	Duration of SEE	3 hours
CIE	25 marks	SEE	50 marks

Course Objectives

1. Learn SQL queries
2. Learn PL/SQL stored procedures
3. Learn Triggers
4. Learn report generation methods
5. Learn database application creation

Course Outcomes

1. Write SQL queries
2. Write stored procedures
3. Write triggers
4. Use file locking and table locking facilities
5. Create small full-fledged database application

Creation of database (exercising the commands for creation)

1. Simple to Complex condition query creation using SQL Plus.
2. Usage of Triggers and Stored Procedures.
3. Creation of Forms for Student information, Library information, Pay roll etc.
4. Writing PL/SQL procedures for data validation.
5. Report generation using SQL reports.
6. Creating password and security features for applications.
7. Usage of File locking, Table locking facilities in applications.
8. Creation of small full- fledged database application spreading over 3 sessions.

Note: The creation of sample database for the purpose of the experiments is expected to be pre-decided by the instructor.

SIP321 Summer Internship/ Mini Project *

Program Description

The Internship Program/ Mini Project allows MCA students to gain practical experience in the workplace before receiving their graduate degrees.

The internship is a required academic course. The student identifies companies willing to hire him/her on a full time basis for 6-week period (minimum required), usually in the summer. The Internship Program supervises the students and awards academic credits (2) upon successful completion of all the required assignments.

Those students who wish to do a Mini Project can use Problem statements and Data Sources from good quality sources and implement a solution. The Student will be evaluated based on the working system that is presented in Semester III of this course.

Intended Learning Outcomes

Upon successful completion of the internship, you should be able to

1. Communicate a practical understanding of how a technology actually operates
2. Demonstrate the ability to integrate and apply theoretical knowledge and skills developed in various courses to real-world situations in a business organization
3. Exhibit the ability to effectively work in a professional environment and demonstrate work ethic and commitment in a work-based environment
4. Demonstrate the ability to successfully complete internship assignments.
5. Reflect on personal and professional development needs and set strategic goals for advancing along an intended career path
6. Communicate effectively in a professional environment in both English and regional language, orally and in writing.