ANDHRA KESARI UNIVERSITY ::ONGOLE

Model Syllabus for 4-Year UG Honours in B.Sc. (Data Science) as Major in consonance with Curriculum framework w.e.f. AY 2025-26

Prepared by Adikavi Nannaya University, Rajahmundry

COURSE STRUCTURE (for Semester I to VI)

Year	Semester	Course	Title of the Course	No. of Hrs /Week	No. of Credits
I	I	1	Computer Fundamentals and Office Automation	3	3
			Computer Fundamentals and Office Automation Lab	2	1
		2	Problem Solving Using C	3	3
			Problem Solving Using C Lab	2	1
	II	3	Python Programming and Data Structures	3	3
			Python programming and Data Structures lab	2	1
		4	Statistical Foundations for Data Science	3	3
			Statistical Foundations for Data Science lab	2	1
II	III	5	Database Management Systems	3	3
			Database Management Systems Lab	2	1
		6	Data Science with R	3	3
			Data Science With R lab	2	1
		7	Web Technologies	3	3
			Web Technologies Lab	2	1
	IV	8	Data Mining	3	3
			Data Mining Lab	2	1
		9	Python for Data Analysis and Visualization	3	3
			Python for Data Analysis and Visualization lab	2	1
		10	Document Oriented Database	3	3
			Document oriented Database lab	2	1
III	V	11	Business Intelligence Tools	3	3
			Business Intelligence Tools Lab	2	1

SEMESTER-III

COURSE 5: DATABASE MANAGEMENT SYSTEMS

Theory	Credits: 3	3 hrs/week

Course Objectives:

1. To understand the fundamentals of data, information, and the evolution from file-based systems to modern database management systems.
2. To develop the ability to design conceptual data models using Entity-Relationship (ER) and Enhanced ER diagrams.
3. To explore relational model principles, such as keys, integrity constraints and normalization.
4. To perform data definition and manipulation using SQL commands including queries, joins, subqueries, views, and set operations.
5. To apply procedural logic using PL/SQL, incorporating control structures, functions, procedures, and database triggers.

Course Outcomes:

At the end of the course, students will be able to:

1. **Describe** the fundamentals of data, database systems, and the differences between file-based and database approaches. **Compare and classify** various DBMS architectures, data models, and their components, including the three-schema architecture.
2. **Design** conceptual data models using Entity-Relationship and Enhanced ER diagrams, applying generalization, specialization, and constraints.
3. **Apply** relational model concepts, including CODD rules and normalization techniques.
4. **Construct and execute** SQL queries for data definition, manipulation, aggregation, joining, and subqueries, including views and set operations.
5. **Develop** PL/SQL programs incorporating control structures, procedures, and functions to manage database behavior effectively.

Unit 1. Overview of Database Management System:

Introduction to data, information, database, database management systems, file-based system, Drawbacks of file-Based System, database approach, Classification of Database Management Systems, advantages of database approach, Various Data Models, Components of Database Management System, three schema architecture of data base, costs and risks of database approach.

Unit 2. Entity-Relationship Model:

Introduction, the building blocks of an entity relationship diagram, classification of entity sets, attribute classification, relationship degree, relationship classification, reducing ER diagram to tables, enhanced entity-relationship model (EER model), generalization and specialization, IS A relationship and attribute inheritance, multiple inheritance, constraints on specialization and generalization, advantages of ER modeling.

Unit 3. Relational Model:

Introduction, CODD Rules, relational data model, concept of key, relational integrity, relational algebra, relational algebra operations, advantages of relational algebra, limitations of relational algebra, Functional dependencies and normal forms.

Unit 4. Structured Query Language:

Introduction, Commands in SQL, Data Types in SQL, Data Definition Language, Selection Operation, Projection Operation, Aggregate functions, Data Manipulation Language, Table Modification Commands, Join Operation, Set Operations, View, Sub Query.

Unit 5. PL/SQL:

Introduction, Shortcomings of SQL, Structure of PL/SQL, PL/SQL Language Elements,Data Types, Operators Precedence, Control Structures, Steps to Create a PL/SQL, Program, Iterative Control, Procedures, Functions.

Textbooks:

1. Database System Concepts, Avi Silberschatz, Henry F. Korth,S. Sudarshan, Seventh Edition, McGraw-Hill
2. Database Management Systems by Raghu Ramakrishnan, McGrawhill

Reference Books:

1. Fundamentals of Database Systems, Elmasri Navathe Pearson Education
2. An Introduction to Database systems, C.J. Date, A.Kannan, S.Swami Nadhan, Pearson

Activities:

Outcome: Describe the fundamentals of data, database systems, and the differences between file-based and database approaches. Compare and classify various DBMS architectures, data models, and their components, including the three-schema architecture.

Activity: Create a comparative presentation or infographic illustrating:

- File-based vs. DBMS approaches
- Types of DBMS architectures (1-tier, 2-tier, 3-tier)
- Data models and the three-schema architecture

Evaluation Method: Rubric-based assessment of the presentation covering clarity, accuracy, and depth of comparison. Include a short quiz to test conceptual understanding.

Outcome: Design conceptual data models using Entity-Relationship and Enhanced ER diagrams, applying generalization, specialization, and constraints.

Activity: Model a university or hospital database using ER and Enhanced ER diagrams that shows:

- Entity sets, relationships
- Generalization/specialization
- Participation and cardinality constraints

Evaluation Method: Diagram submission with peer review and instructor feedback. Use a checklist to assess completeness, correctness, and notation usage.

Outcome: Apply relational model concepts, including CODD rules, and normalization techniques.

Activity: Normalize a given Database upto 3NF.

Evaluation Method: Written assignment graded on:

- Correctness of normalization steps
- Short-answer questions on CODD rules

Outcome: Construct and execute SQL queries for data definition, manipulation, aggregation, joining, and subqueries, including views and set operations.

Activity: Implement a mini-project (e.g., Library or Inventory DB) using SQL. Include:

- Table creation (DDL)
- Data manipulation (DML)
- Aggregation, joins, subqueries, views, and set operations

Evaluation Method: Lab-based practical test with query execution and output validation. Include a viva to explain logic and optimization.

Outcome: Develop PL/SQL programs incorporating control structures, procedures and functions to manage database behaviour effectively.

Activity: Build a PL/SQL-based payroll or student grading system using:

- Procedures and functions
- Control structures (IF, LOOP)
- Triggers for automated updates

Evaluation Method: Code review and demonstration. Evaluate based on:

- Syntax correctness
- Logical flow

SEMESTER-III

COURSE 5: DATABASE MANAGEMENT SYSTEMS

Practical **Credits: 1** **2 hrs/week**

Experiment 1 : Database: Inventory Management

Table 1: Products

Structure:

Column Name	Data Type	Constraints
product_id	INT	PRIMARY KEY
product_name	VARCHAR(50)	NOT NULL
price	DECIMAL(10,2)	CHECK(price > 0)
stock_qty	INT	CHECK(stock_qty >= 0)

Sample Data:

product_id	product_name	price	stock_qty
1	Pen	10.00	100
2	Notebook	50.00	200
3	Stapler	120.00	50
4	Marker	25.00	80
5	File Folder	60.00	150

Table 2: Suppliers

Structure:

Column Name	Data Type	Constraints
supplier_id	INT	PRIMARY KEY
supplier_name	VARCHAR(50)	NOT NULL
contact_no	VARCHAR(20)	UNIQUE
product_id	INT	FOREIGN KEY REFERENCES Products(product_id)

Sample Data:

supplier_id	supplier_name	contact_no	product_id
101	StationeryMart	9876543210	1
102	PaperWorld	9876500000	2
103	OfficeSupplies	9876512345	3
104	MarkerHub	9876522222	4
105	FileDepot	9876533333	5

Section A: DDL (Data Definition Language)

1. Create a database called InventoryDB.
2. Create a table Products and table Suppliers with the specified columns and constraints:

Section B: DML (Data Manipulation Language)

4. Insert at least 5 rows into the Products table.
5. Insert at least 5 rows into the Suppliers table.
6. Update the stock quantity of product 'Pen' to 120.
7. Delete a supplier with a specific supplier_id.
8. Write a query to rename 'Notebook' to 'NoteBook A4'

Section C: DQL (SELECT Queries)

9. Display all records from the Products table.
10. Display only product_name and price of all products.
11. List all products that have a stock quantity less than 100.
12. Show all products between 20 and 100 price range.
13. Find all suppliers whose contact number starts with '98765'.
14. Find the average price of products.
15. Display the total number of products in the inventory.
16. Show the maximum and minimum stock quantities.
17. Count how many suppliers supply each product.
18. Show all products where price > 50 AND stock_qty > 100.
19. Show all products where price < 20 OR stock_qty < 80.
20. Display suppliers whose supplier_name contains the word 'Mart'
21. List all suppliers along with the product they supply (use INNER JOIN).
22. Display suppliers whose name starts with 'S'.
23. Find products whose name has exactly 5 characters
24. Find suppliers who supply products costing more than 100.

Experiment 2 : ONLINE BOOKSTORE DB

An online book store wants to implement a BOOKSTORE DB for managing their online transactions by using the following tables.

Authors Table

Column Name	Data Type	Constraints
author_id	INTEGER	PRIMARY KEY
first_name	VARCHAR	NOT NULL
last_name	VARCHAR	NOT NULL
nationality	VARCHAR	NULL allowed

Books Table

Column Name	Data Type	Constraints
book_id	INTEGER	PRIMARY KEY

Title	VARCHAR	NOT NULL
author_id	INTEGER	FOREIGN KEY REFERENCES Authors
publication_year	INTEGER	
Price	DECIMAL	

Customers Table

Column Name	Data Type	Constraints
customer_id	INTEGER	PRIMARY KEY
first_name	VARCHAR	NOT NULL
last_name	VARCHAR	NOT NULL
Email	VARCHAR	UNIQUE, NOT NULL
Address	VARCHAR	NOT NULL

Orders Table

Column Name	Data Type	Constraints
order_id	INTEGER	PRIMARY KEY
customer_id	INTEGER	FOREIGN KEY REFERENCES Customers
book_id	INTEGER	FOREIGN KEY REFERENCES Books
order_date	DATE	NOT NULL
quantity	INTEGER	NOT NULL

SAMPLE DATA SET for BOOKSTORE DB

Authors Table

author_id	first_name	last_name	nationality
1	Jane	Austen	British
2	George	Orwell	British
3	Gabriel	Garcia Marquez	Colombian
4	Toni	Morrison	American
5	Mark	Twain	American
6	Harper	Lee	American
7	Fyodor	Dostoevsky	Russian

Books Table

book_id	Title	author_id	publication_year	price

101	Pride and Prejudice	1	1813	12.99
102	1984	2	1949	9.50
103	One Hundred Years of Solitude	3	1967	15.00
104	Beloved	4	1987	11.25
105	Animal Farm	2	1945	8.75
106	Adventures of Huckleberry Finn	5	1884	10.50
107	To Kill a Mockingbird	6	1960	14.00

Customers Table

customer_id	first_name	last_name	Email	address
201	Alice	Smith	alice.s@example.com	12 Oak St, London
202	Bob	Johnson	bob.j@example.com	45 Pine Ave, Oxford
203	Charlie	Brown	charlie.b@example.com	78 Maple Rd, Bristol
204	Diana	Prince	diana.p@example.com	34 Queen St, York
205	Edward	Norton	edward.n@example.com	22 River Ln, Leeds
206	Fiona	Hall	fiona.h@example.com	56 Lake Dr, Bath
207	Greg	Miller	greg.m@example.com	89 Park Ave, Glasgow

Orders Table

order_id	customer_id	book_id	order_date	Quantity
301	201	101	2025-07-20	1
302	202	102	2025-07-21	2
303	201	105	2025-07-22	1
304	203	103	2025-07-23	1
305	204	106	2025-07-24	1
306	205	107	2025-07-25	3
307	206	104	2025-07-26	2

Section A: DDL (Schema Design & Constraints)

1. Write SQL statements to create all 4 tables (Authors, Books, Customers, Orders) with:
    - Primary Keys
    - Foreign Keys
    - Appropriate data types
    - NOT NULL constraints where necessary.
2. Alter the Books table to add a constraint that price must be greater than 0.

3. Add a new column phone_number to the Customers table (VARCHAR(15)) and ensure it is unique.
4. Drop the phone_number column from the Customers table.

Section B: DML (Data Manipulation)

5. Insert at least 7 records for each table (use sample dataset above).
6. Update the price of the book titled *Animal Farm* by increasing it by 10%.
7. Delete all orders made before 2025-07-21.
8. Change the nationality of Gabriel Garcia Marquez to "Latino-American".

Section C: SELECT Queries (Data Querying)

9. List all books published between 1900 and 2000.
10. Find all customers whose email contains "example.com".
11. Retrieve books whose price is between 10 and 15 and published before 1950.
12. Show authors who are either 'British' or 'American'.
13. Find books that have a price less than 10 or are published after 1980.
14. Display all orders placed after 2025-07-22.
15. List all books written by author with author_id = 2.
16. Find customers whose last name starts with B.
17. Show all books with a price NOT between 9 and 13.
18. Display books whose publication_year is in (1813, 1945, 1987).
19. Find authors whose nationality is NOT 'British'.
20. List customers whose address contains the word Park.
21. Show all books sorted by price in descending order.
22. List authors in alphabetical order by last_name.
23. Display orders sorted by order_date (latest first).

Use of Date Functions

24. Show all orders placed in July 2025.
25. Show all orders with an estimated delivery date (5 days after order date).
26. Show customers who placed an order on a weekend.
27. Calculate how many days have passed since the last order was placed.

Aggregate Functions (COUNT, SUM, AVG, MIN, MAX)

28. Count the total number of books in the database.
29. Find the average price of all books.
30. Show the highest-priced book.
31. Count how many orders each customer has placed.
32. Calculate the total sales (price × quantity) for each customer.

GROUP BY and HAVING

33. Count how many books are written by each author.
34. Group orders by customer_id and display total quantity ordered.
35. Show customers who have ordered more than 2 books in total (use HAVING).
36. Find the total number of books sold per author (GROUP BY author).

Experiment 3: EMPLOYEE DB

An enterprise wants to automate its employee management process by implementing an Employee Database.The goal is to replace manual record-keeping with a centralized system that stores employee, department, and project details. Use the following table structures and data set to implement Employee DB.

EmployeeDB - Table Structures

1. Departments Table

Column	Type	Constraints
dept_id	INT	PRIMARY KEY
dept_name	VARCHAR	UNIQUE, NOT NULL
location	VARCHAR	NOT NULL

2. Employees Table

Column	Type	Constraints
emp_id	INT	PRIMARY KEY
first_name	VARCHAR	NOT NULL
last_name	VARCHAR	NOT NULL
email	VARCHAR	UNIQUE, NOT NULL
phone	VARCHAR	CHECK (phone LIKE '--____')
hire_date	DATE	NOT NULL
job_title	VARCHAR	NOT NULL
salary	DECIMAL	CHECK (salary > 0)
dept_id	INT	FOREIGN KEY REFERENCES Departments(dept_id)
manager_id	INT	FOREIGN KEY REFERENCES Employees(emp_id) (self-referential)

3. Projects Table

Column	Type	Constraints
project_id	INT	PRIMARY KEY
project_name	VARCHAR	NOT NULL
start_date	DATE	NOT NULL
end_date	DATE	NULL
dept_id	INT	FOREIGN KEY REFERENCES Departments(dept_id)

4. Employee_Project Table (Many-to-Many)

Column	Type	Constraints
emp_id	INT	FOREIGN KEY REFERENCES Employees(emp_id), PRIMARY KEY(emp_id, project_id)
project_id	INT	FOREIGN KEY REFERENCES Projects(project_id)
hours_allocated	INT	CHECK (hours_allocated > 0)

Sample Data Set

Departments Table

dept_id	dept_name	Location
1	HR	New York
2	IT	San Francisco
3	Finance	Chicago
4	Marketing	Boston
5	Operations	Seattle
6	Legal	Washington D.C.
7	Sales	Dallas
8	R&D	Austin
9	Procurement	Denver
10	Customer Care	Miami

2. Employees Table

emp_id	first_name	last_name	Email	phone	hire_date	job_title	salary	dept_id	manager_id
101	Alice	Johnson	alice.j@corp.com	123-456-7890	2020-03-15	HR Manager	75000	1	NULL
102	Bob	Smith	bob.s@corp.com	234-567-8901	2019-05-20	IT Analyst	65000	2	104
103	Charlie	Brown	charlie.b@corp.com	345-678-9012	2021-01-10	Finance Executive	58000	3	106

104	Diana	Prince	diana.p@corp.com	456-789-0123	2018-07-12	IT Manager	90000	2	NULL
105	Ethan	Hunt	ethan.h@corp.com	567-890-1234	2022-02-25	Marketing Lead	62000	4	NULL
106	Fiona	Hall	fiona.h@corp.com	678-901-2345	2017-11-01	Finance Manager	85000	3	NULL
107	Greg	Miles	greg.m@corp.com	789-012-3456	2023-04-15	IT Support	45000	2	104
108	Hannah	White	hannah.w@corp.com	890-123-4567	2021-09-05	HR Executive	50000	1	101
109	Ian	Scott	ian.s@corp.com	901-234-5678	2020-11-20	Operations Analyst	56000	5	NULL
110	Julia	Adams	julia.a@corp.com	012-345-6789	2019-12-18	Legal Advisor	70000	6	NULL

3. Projects Table

project_id	project_name	start_date	end_date	dept_id
201	Payroll System	2023-01-01	NULL	3
202	Website Upgrade	2023-02-10	NULL	2
203	Recruitment Drive	2023-03-05	NULL	1
204	Ad Campaign	2023-05-20	NULL	4
205	New CRM Tool	2023-04-15	NULL	7
206	Compliance Portal	2023-06-10	NULL	6
207	Inventory System	2023-07-01	NULL	5
208	AI Research	2023-08-05	NULL	8
209	Customer Feedback	2023-09-10	NULL	10
210	Procurement System	2023-10-01	NULL	9

4. Employee_Project Table

emp_id	project_id	hours_allocated
102	202	120

104	202	80
103	201	100
106	201	150
101	203	50
105	204	70
107	202	60
109	207	90
110	206	110
108	203	40

Section A: DDL (Schema Creation & Modification)

1. Write SQL statements to create the above tables with the specified constraints
2. Alter the Employees table to add a column bonus DECIMAL(8,2) with default value0.
3. Drop the column bonus from Employees.

Section B: DML (Insert, Update, Delete)

4. Insert at least 10 rows into Departments, Employees, Projects, and Employee_Project.(use the above data set)
5. Try inserting an employee with a negative salary (should fail due to CHECK constraint).
6. Update the salary of the employee with emp_id = 103 by 15%.
7. Delete an employee record who has resigned (choose any emp_id).
8. Increase all employees' salaries in the IT department by 5%.
9. Change the department of an employee to "Research".(should fail due to FK constraint)

Section C: DQL (Select Queries)

10. List all employees and their details.
11. Show all employees in the "HR" department.
12. Find employees with salaries between 50,000 and 80,000.
13. Retrieve employees hired after 2020.
14. Show employees who are in either the IT or Finance department.
15. Find employees whose email ends with "@corp.com".
16. List all employees with salary > 60,000 AND located in "New York".
17. Display employees in descending order of salary.
18. Count the number of employees in each department.
19. Show the average salary of employees department-wise.
20. Display departments where the average salary is greater than 70,000.
21. Find the number of employees in each project.
22. Display departments with more than 3 employees.
23. Show the sum of all salaries department-wise.
24. List all distinct department IDs from the Employees table.

25. Show employee names with the year they were hired.
26. Show employees grouped by the year of hire.
27. List employees hired in the last 90 days.
28. List the no of years of experience of all the employees

Section D: Joins

29. List all employees with their department names (INNER JOIN).
30. Display all departments along with employees, including those departments without employees (LEFT JOIN).
31. Show employees and the projects they are working on (JOIN 3 tables: Employees, Employee_Project, Projects).
32. List projects along with total hours allocated by employees.
33. Write a query to find employees who are working on more than one project.
34. Show all projects handled by the 'Finance' department.

Section E: PL/SQL Programming

1. Write a procedure GetEmpInfo that takes emp_id as input and displays name, salary, and department.
2. Write a PL/SQL block that checks if an employee's salary is above 50,000.
   If yes, print "High Salary" ;Otherwise print "Standard Salary".
3. Write a PL/SQL program to display the top 10 rows in the Emp table based on their job and salary
4. Write a stored procedure GiveBonus that takes department ID and a designation as input, along with a bonus amount, and updates the salary of all employees in that department who have the specified designation by adding the bonus amount to their current salary.
5. Create a trigger to prevent inserting employees with a salary less than 30,000.
6. Create a trigger to avoid any transactions(insert, update, delete) on the EMP table on Saturday & Sunday.

SEMESTER-III

COURSE 6: DATA SCIENCE WITH R

Theory	Credits: 3	3 hrs/week

Course Objectives

1. Introduce the data science process, lifecycle, and applications in real-world domains.
2. Build proficiency in R programming for data manipulation, exploration, and visualization.
3. Train students in handling structured, unstructured, and time-based data effectively.
4. Familiarize with basic machine learning and statistical modeling using R.
5. Develop awareness of ethical, interpretability, and responsible use of data science.

Course Outcomes

At the end of the course, students will be able to:

1. Explain the Data Science process and perform EDA (Exploratory Data Analysis).
2. Write R programs using variables, functions, loops, and packages for basic analytics.
3. Perform data wrangling, cleaning, and visualization with R libraries (dplyr, tidyr, ggplot2).
4. Build and evaluate basic machine learning models such as regression and clustering.
5. Apply data science techniques to practical case studies.

Unit 1. Introduction to Data Science Process:

Introduction- Definition - Data Science in various fields - Examples - Impact of Data Science - Data Analytics Life Cycle - Data Science Toolkit - Data Scientist - Data Science Team, Exploratory Data Analysis (EDA), Feature Engineering & Data Transformation

Unit 2. Basics of R Programming:

Introduction to R and RStudio, Data Types, Variables, Operators, Control Structures (if, loops, apply), Functions and Packages, Data Input/Output (CSV, Excel, XML, JSON).

Unit 3. Data Handling & Visualization in R:

Data Frames, Lists, Matrices, Data Wrangling with dplyr and tidyr, Handling Missing Data, Working with Date/Time in R. Visualization with ggplot2: grammar of graphics, aesthetics, geometries, scales.

Faceting and layering techniques,Visualizing categorical and numerical data, Customizing and exporting plots

Unit 4. Applications & Case Studies in Data Science:
Simple Linear Regression, Multiple Regression
Model Evaluation Method: Accuracy, Confusion Matrix, ROC.
K-Means Clustering, Text Mining & Word Clouds, Recommender Systems Basics, Ethical Issues in Data Science

Unit 5. Advanced Topics in Data Science with R :
Introduction to Time Series Analysis in R (ARIMA basics)- Concept of time series (trend, seasonality, noise), Time series objects in R (ts, zoo, xts), Plotting and decomposing time series, Stationarity and differencing, Autocorrelation & Partial Autocorrelation (ACF/PACF), AR, MA, ARIMA model basics, Forecasting using forecast package
Creating interactive visualizations with plotly packages-Converting ggplot2 plots to interactive plots
Animations and sliders in plotly

R Shiny: Building interactive web applications-Introduction to Shiny framework, UI and server functions, Reactive expressions and reactivity in Shiny, Input and output widgets (sliders, dropdowns, text), Layouts and dashboard design

Textbooks

1. An Introduction to Statistical Learning with Applications in R, Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, Springer, 2nd Edition, 2021
2. R for Data Science, Hadley Wickham and Garrett Grolemund, O'Reilly Media, 2017.

Reference Books

1. The Art of R Programming, Norman Matloff,, No Starch Press, 2011.
2. Modern Applied Statistics with S, W.N. Venables & B.D. Ripley, Springer, 2002.
3. Introduction to Data Science: Data Analysis and Prediction Algorithms with R, Rafael A. Irizarry, CRC Press, 2020.
4. Data Science from Scratch: First Principles with Python (for conceptual clarity only), Joel Grus,

Activities:

Outcome: Explain the Data Science process and perform EDA (Exploratory Data Analysis).

Activity: Use a real-world dataset (e.g., Titanic or COVID data) to:

- Outline the steps of the Data Science workflow
- Perform EDA using summary statistics and visualizations (histograms, boxplots, scatterplots)

Evaluation Method: Presentation and checklist (10-point scale):

- Clear explanation of workflow stages
- Quality of EDA insights
- Use of appropriate plots and summaries

Outcome: Write R programs using variables, functions, loops, and packages for basic analytics.

Activity: Write an R script that:

- Reads a CSV file
- Uses if, for, and while loops
- Defines and calls custom functions with arguments and return values

Evaluation Method: Code review and execution test to verify (10-point scale):

- Correctness of the syntax and logic
- Functionality of control structures
- Output accuracy and modularity

Outcome: Perform data wrangling, cleaning, and visualization with R libraries (dplyr, tidyr, ggplot2).

Activity: Clean a messy dataset using:

- dplyr for filtering, selecting, and mutating
- tidyr for reshaping and handling missing values
- Time-based operations (e.g., filling gaps, formatting dates)

Evaluation Method: Before-and-after comparison (10 point score):

- Completeness of cleaning steps
- Use of appropriate functions
- Handling of missing/time data

Outcome: Implement basic machine learning models and evaluate performance using appropriate metrics and visual tools.

Activity: Build a simple classification model (e.g., logistic regression or decision tree) using R:

- Train/test split
- Predict outcomes
- Evaluate using confusion matrix, accuracy, precision, recall

Evaluation Method: Model report and demo (10 point scale):

- Correct implementation of model
- Use of evaluation metrics

SEMESTER-III

COURSE 6: DATA SCIENCE WITH R

Practical **Credits: 1** **2 hrs/week**

List of Practicals:

1. Compute Mean, Median, Mode, Variance, and Standard Deviation
2. Visualize Binomial, Normal, and Poisson Distributions
3. Perform t-test and Chi-Square Test in R
4. Calculate Correlation and Build a Simple Linear Regression Model
5. Conduct Exploratory Data Analysis (EDA) on a Real-World Dataset
6. Apply Feature Engineering: Scaling, Normalization, and Encoding
7. Practice R Programming: Variables, Control Structures, and Functions
8. Read and Write Data from CSV, Excel, JSON, and XML Files
9. Use dplyr and tidyr for Data Wrangling Tasks
10. Handle Missing Data and Detect Outliers
11. Work with Dates and Times in R
12. Visualize Data Using ggplot2 (Bar, Scatter, Histogram, Boxplot)
13. Perform K-Means Clustering and Visualize Clusters
14. Evaluate Models Using Confusion Matrix, Accuracy, and ROC Curve
15. Perform Text Mining and Create a Word Cloud
16. Time Series Forecasting with ARIMA on a real dateset (e.g., monthly airline passengers, stock prices, or temperature data).
17. Create interactive bar, line, and scatter plots using plotly. On a real dataset (e.g., COVID-19 cases, sales data, or student marks).
18. Develop a Shiny app that lets users upload a CSV file.

SEMESTER-III

COURSE 7: WEB TECHNOLOGIES

Theory	Credits: 3	3 hrs/week

Course Objectives

1. Understand the principles of web design and distinguish between web and desktop application architectures.
2. Develop static web pages using HTML elements, attributes, and multimedia integration techniques.
3. Style web pages effectively using CSS, including layout control, responsive design, and UI enhancements.
4. Implement dynamic behaviors and form validations using JavaScript and the Document Object Model (DOM).
5. Explore JSON and jQuery for handling structured data and simplifying client-side scripting in web development.

Course Outcomes

At the end of the course, students will be able to:

1. Design and structure HTML-based webpages incorporating text, images, tables, forms, and multimedia content.
2. Apply CSS styling rules to manage layout aesthetics, interactivity, and responsiveness across devices.
3. Use JavaScript for string manipulation, event handling, arrays, object operations, and basic validation.
4. Employ client-side scripting to enhance form functionality, create dialog interactions, and add animations via events.
5. Parse JSON data and use jQuery to simplify DOM manipulation, AJAX calls, and build dynamic, data-driven web applications.

Unit 1.HTML:

Introduction to web designing, difference between web applications and desktop applications, introduction to HTML, HTML structure, elements, attributes, headings, paragraphs, images, tables, lists, blocks, symbols, embedding multi-media components in HTML, HTML forms

Unit 2.CSS:

CSS home, introduction, syntax, CSS combinators, colors, background, borders, margins, padding, height/width, text, fonts, tables, lists, position, overflow, float, pseudo class, pseudo elements, opacity, tool tips, image gallery, CSS forms, CSS counters.

Unit 3.Java Script:

What is DHTML, JavaScript, basics, variables, operators, statements, string manipulations, mathematical functions, arrays, functions. objects, regular expressions, exception handling.

Unit 4. Client-Side Scripting:

Accessing HTML form elements using Java Script object model, basic data validations, data format validations, generating responsive messages, opening windows using java script, different kinds of dialog boxes, accessing status bar using java script, embedding basic animative features using different keyboard and mouse events.

Unit 5. JSON and jQuery

Introduction to JSON: Need for data exchange formats, JSON syntax, JSON vs XML, parsing JSON, creating JSON objects and arrays, accessing nested JSON data, reading/writing JSON in JavaScript.

Working with jQuery: Introduction to jQuery, selectors, filters, DOM manipulation, event handling, animations, effects, and chaining.

Text Book(s)

1. Web Programming: Building Internet Applications, Chris Bates, Wiley, Second Edition.
2. An Introduction to Web Design plus Programming, Paul S. Wang, Sanda S. Katila, Thomson.
3. Learning jQuery – Jonathan Chaffer, Karl Swedberg, Packt Publishing.
4. JSON at Work – Tom Marrs, O'Reilly Media.

Reference Books

1. Head First HTML and CSS, Elisabeth Robson, Eric Freeman, O'Reilly Media Inc.
2. An Introduction to HTML and JavaScript: for Scientists and Engineers, David R. Brooks, Springer.

3. Schaum's Easy Outline: HTML, David Mercer, McGraw Hill Professional.
4. jQuery in Action, Bear Bibeault, Yehuda Katz, Manning Publications.
5. Beginning JSON, Ben Smith, Apress.

Activities:

Outcome: Design and structure HTML webpages with text, images, tables, forms, and multimedia.
Activity: Create a personal profile webpage with all these elements.
Evaluation Method: Checklist (10 pts) - correct tags, structure, working forms/media.

Outcome: Apply CSS rules for layout, aesthetics, interactivity, and responsiveness.
Activity: Style the profile page using colors, fonts, Flexbox/Grid, hover effects, and media queries.
Evaluation Method: Rubric (10 pts) - visual appeal, responsiveness, selector/layout usage, clean code.

Outcome: Use JavaScript for string manipulation, events, arrays/objects, and validation.
Activity: Add form validation, greeting message, array/object display, and button click handling.
Evaluation Method: Demo & testing (10 pts) - correct syntax, validation, event handling, output behavior.

Outcome: Employ client-side scripting for dialogs, animations, and event-based interactions.
Activity: Add show/hide sections, confirmation dialog on submit, and animations via mouse/keyboard events.
Evaluation Method: Live demo (10 pts) - smooth interaction, correct event listeners, proper animations, good UX.

Outcome: Parse JSON and use jQuery for DOM manipulation, AJAX, and dynamic data display.
Activity: Fetch JSON via jQuery AJAX, render as table/list, add filter/sort, and compute summaries.
Evaluation Method: Demo & checklist (10 pts) - JSON parsing, jQuery usage, filter/sort functionality, error handling.

SEMESTER-III

COURSE 7: WEB TECHNOLOGIES

Practical **Credits: 1** **2 hrs/week**

List of Experiments:

1. Create an HTML document with the following formatting options:

   (a) Bold, (b) Italics, (c) Underline, (d) Headings (Using H1 to H6 heading styles), (e) Font (Type, Size and Color), (f) Background (Colored background/Image in background), (g) Paragraph, (h) Line Break, (i) Horizontal Rule, (j) Pre tag
2. Create an HTML document which consists of:

   (a) Ordered List (b) Unordered List (c) Nested List (d) Image
3. Collect any ten images of your choice. Using table tag, align the images as follows:

4. Create a form using HTML which has the following types of controls:

   (a) Text Box (b) Option/radio buttons (c) Check boxes (d) Reset and Submit buttons
5. Embed a calendar object in your web page.
6. Create a form that accepts the information from the subscriber of a mailing system.
7. Apply CSS to design a student registration form (use different selectors, colors, borders, spacing).
8. Create a responsive webpage using CSS Flexbox/Grid.
9. Add hover effects and transitions on images and buttons using CSS.
10. Write a JavaScript program to perform string operations (reverse, substring, count vowels).
11. Create a JavaScript form validation program (check email format, password length, required fields).

12. Develop a webpage that displays greetings based on the current time (morning, afternoon, evening).
13. Use JavaScript to manipulate arrays and objects (add, delete, sort, search).
14. Fetch and display student information stored in a JSON object on a webpage.
15. Fetch real-time weather data from an open API (e.g., OpenWeatherMap) in JSON format and display temperature, humidity, and conditions dynamically on a webpage.
16. Use jQuery to simplify DOM manipulation (hide, show, fade, slide, toggle).

SEMESTER-IV

COURSE 8: DATA MINING

Theory	Credits: 3	3 hrs/week

Course Objectives:

- Provide an understanding of data warehousing concepts, architecture, and OLAP operations for effective storage, modeling, and analysis.
- Develop knowledge of data mining fundamentals, tasks, and preprocessing techniques to prepare data for mining.
- Introduce students to association rule mining algorithms for discovering hidden patterns and relationships in large datasets.
- Enable learners to apply classification techniques (decision trees, Bayesian, nearest neighbor, rule-based) for predictive modeling.
- Equip students with knowledge of clustering paradigms and algorithms (partitioning, hierarchical, density-based, categorical) for data grouping and pattern discovery.

Course Outcomes:

Upon successful completion of the course, the student will be able to:

1. Explain the architecture, schemas, and OLAP operations of data warehousing and distinguish it from traditional database systems.
2. Apply preprocessing techniques (data cleaning, dimensionality reduction, feature selection, transformation, and similarity measures) to prepare raw data for analysis.
3. Implement various association rule mining algorithms (Apriori, Partition, FP-Growth, etc.) to uncover meaningful relationships within large datasets.
4. Build and evaluate classification models using decision tree algorithms (ID3, C4.5, CART), rule-based classifiers, Bayesian classifiers, and nearest-neighbor methods.
5. Analyze and implement clustering techniques such as K-Means, K-Medoid, DBSCAN, BIRCH, and categorical clustering methods (STIRR, ROCK, CACTUS) for grouping and pattern discovery in different types of datasets.

Unit-1: Data Warehousing:

Introduction to Data Ware House, Differences between Database systems and Data Ware House, Data Ware House characteristics, Data Ware House Architecture and its components,

Data Modeling, Schema Design, star and snow-Flake Schema, Fact Constellation, Fact Table, OLAP cube, OLAP Operations.

Unit-2: Data Mining:

What is Data Mining? Data Mining: Definitions, KDD vs Data Mining, Data Mining Tasks, Data Preprocessing- Data Cleaning, Missing Data, Dimensionality Reduction, Feature Subset Selection, Discretization and Binarization, Data Transformation; Measures of similarity and Dissimilarity-Basics.

Issues and Challenges in DM, DM Applications- Case Studies

Unit-3: Association Analysis:

Association Rules: What is an Association Rule?, Methods to Discover Association Rules, A Priori Algorithm, Partition Algorithm, Pincer-Search Algorithm, Dynamic Itemset Counting Algorithms, FP-Tree Growth Algorithm, Generalized Association Rule, Association Rules with Item Constraints

Unit-4: Classification:

Definition, What is Decision Tree?, Tree Construction Principle, Best Split, Splitting Indices, Splitting Criteria, Decision Tree Construction Algorithms: CART, ID3, C4.5, Method for Comparing Classifiers, Rule Based Classifiers, Nearest Neighbor Classifiers, Bayesian Classifiers.

Unit-5: Clustering Techniques:

Clustering Paradigms, Partitioning Algorithms (K-Means), k-Medoid Algorithms, Hierarchical Clustering: DBSCAN, BIRCH, Categorical Clustering Algorithms: STIRR, ROCK, CACTUS

Textbooks:

1. Data Mining Techniques, Arun K Pujari, 3rd Edition, Universities Press
2. Data Mining: Concepts and Techniques, Jiawei Han, Micheline Kamber, Jian Pei, 3rd Edition, Morgan Kaufmann Publishers

Reference Books:

1. K.P. Soman , Shyam Diwakar, V.Ajay ,2006, Insight into Data Mining Theory and Practice, Prentice Hall of India Pvt. Ltd - New Delhi.

2. Introduction to Data Mining, Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, Vipin Kumar, 2nd edition,

Activities:

Outcome: Explain the architecture, schemas, and OLAP operations of data warehousing and distinguish it from traditional database systems.

Activity: Students will design a **conceptual schema** for a data warehouse using **star or snowflake schema** for a given business case (e.g., sales, hospital, or university records). They will also **demonstrate OLAP operations** (roll-up, drill-down, slice, dice) on a sample dataset using a spreadsheet or OLAP tool.

Evaluation Method: Assessment will be based on correctness and completeness of schema design, appropriateness of fact/dimension tables, and accuracy of OLAP operation outputs. A short viva/quiz will be used to test conceptual understanding.

Outcome 2: Apply preprocessing techniques (data cleaning, dimensionality reduction, feature selection, transformation, and similarity measures) to prepare raw data for analysis.

Activity: Students will be given a **real-world noisy dataset** (with missing values, outliers, redundant features). They will perform:

- Data cleaning (handling missing values, outliers)
- Dimensionality reduction (e.g., PCA)
- Feature selection (e.g., filter/wrapper methods)
- Similarity/dissimilarity measure calculations.

Evaluation Method: Evaluation will consider correctness of preprocessing steps, justification of chosen methods, and clarity of intermediate outputs. Students will submit a short report with before/after comparisons.

Outcome 3: Implement various association rule mining algorithms (Apriori, Partition, FP-Growth, etc.) to uncover meaningful relationships within large datasets.

Activity: Using a **transaction dataset** (e.g., market basket data), students will implement Apriori/FP-Growth in **Python (mlxtend library) or Weka**. They will generate frequent itemsets, derive association rules, and interpret them under given **support, confidence, and lift thresholds.**

Evaluation Method: Submissions will be graded on correctness of generated rules, clarity of code/parameters, quality of interpretation, and ability to link discovered rules to practical business insights.

Outcome 4: Build and evaluate classification models using decision tree algorithms (ID3, C4.5, CART), rule-based classifiers, Bayesian classifiers, and nearest-neighbor methods.

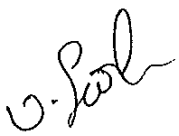
Activity: Students will implement at least **two classification algorithms** (e.g., Decision Tree + Naïve Bayes or KNN) on a dataset like **Iris, Titanic, or Student Performance**. They will evaluate models using **confusion matrix, accuracy, precision, recall, and F1-score** and compare results.

Evaluation Method: Assessment will consider correctness of implementation, clarity in performance comparison, and visualization of decision trees/rules. Students must explain why certain models performed better.

Outcome: Analyze and implement clustering techniques such as K-Means, K-Medoid, DBSCAN, BIRCH, and categorical clustering methods (STIRR, ROCK, CACTUS).

Activity: Students will implement at least two clustering algorithms (e.g., K-Means and DBSCAN or BIRCH) on real datasets (e.g., customer segmentation, text data, student groups). They will compare clusters using Silhouette Score, Davies-Bouldin Index, and visualize results using scatter plots/dendrograms.

Evaluation Method: Evaluation will be based on accuracy of clustering implementation, choice of parameters (e.g., k in K-Means, eps in DBSCAN), visualization quality, and clarity in interpretation of clusters.



SEMESTER-IV

COURSE 8: DATA MINING

Practical	Credits: 1	2 hrs/week

List of Experiments:

Recommended datasets: **weather.arff, iris.arff, supermarket.arff, vote.arff, contact-lenses.arff**, or custom CSV datasets.

1. Load datasets in WEKA and explore data formats (ARFF/CSV)
2. Perform data cleaning and handle missing values using filters
3. Apply normalization and discretization on numeric attributes
4. Reduce data using attribute selection and PCA
5. Summarize and visualize data using statistical tools and class-wise comparison in WEKA.
6. Generate association rules using the Apriori algorithm
7. Apply multilevel association rule mining using hierarchical attributes
8. Apply K-means clustering and interpret the cluster outputs.
9. Perform hierarchical clustering and visualize results using dendrograms.
10. Apply Expectation-Maximization (EM) clustering and analyze cluster summaries.
11. Build a decision tree classifier using J48 and evaluate its performance.
12. Perform Naive Bayes classification and compare with decision tree results.
13. Apply rule-based classification using PART or JRip algorithms.
14. Compare classifiers using confusion matrix, accuracy, and ROC curves.
15. Perform basic text preprocessing and clustering using TF-IDF and K-means.

SEMESTER-IV

COURSE 9: PYTHON FOR DATA ANALYSIS AND VISUALIZATION

Theory	Credits: 3	3 hrs/week

Course Objectives:

1. Introduce foundational concepts of NumPy arrays and array operations for efficient numerical computing.
2. Teach key data structures and manipulation techniques using Pandas.
3. Enable students to perform data input/output operations and implement basic data cleaning workflows.
4. Explore string processing methods and feature engineering strategies in Pandas.
5. Guide learners in advanced data wrangling tasks including merging, reshaping, hierarchical indexing and visualization.

Course Outcomes:

1. Demonstrate proficiency in creating and manipulating NumPy arrays for mathematical operations and simulations.
2. Apply Pandas Series and DataFrame operations for structured data handling and analysis.
3. Read, write, and clean diverse data formats using Python tools, addressing missing values and outliers.
4. Implement vectorized string operations and create derived features for enhanced model readiness.
5. Perform complex data wrangling tasks such as merging datasets, reshaping data structures, generating group-level statistics, Visualize the data.

Unit 1. NumPy Essentials:

NumPy ndarray: A Multidimensional Array Object, Creating ndarrays, Data Types for ndarrays, Arithmetic with Arrays, Basic Indexing and Slicing, Boolean Indexing, Fancy Indexing, Transposing Arrays, Swapping Axes, Universal Functions: Element-wise Operations, Basic Mathematical and Statistical Functions, Random Number Generation (basic use)

Unit 2. Pandas Basics and Data Structures:

Series, DataFrame, Index objects, Indexing and Selection, Filtering and Boolean Indexing, Arithmetic and Data Alignment, Sorting and Ranking, Dropping Entries, Handling Duplicate Indexes

Unit 3. Data Input, Output, and Cleaning:

Reading and Writing Data in Text Format (CSV, TXT), Working with JSON, Reading Microsoft Excel Files, Handling Missing Data, Dropping and Filling Missing Values, Replacing Values, Renaming Axis Indexes, Removing Duplicates, Filtering Outliers, Transforming Data Using Mapping or Functions

Unit 4. String Operations and Feature Engineering:

String Methods in pandas, Basic Regular Expressions, Vectorized String Functions, Creating Dummy/Indicator Variables, Permutation and Random Sampling.

Unit 5. Data Wrangling, Reshaping & Visualization:

Merging and Joining Datasets, Concatenating Along an Axis, Combining Data with Overlap, Reshaping with Pivot, Stack, and Unstack, Basic Hierarchical Indexing, Summary Statistics by Group or Level

Introduction to matplotlib: plots, customization, styling, Seaborn for statistical data , visualization, Plotly for interactive charts and dashboards.

Textbooks

1. Python for Data Analysis: Data Wrangling with pandas, NumPy, and Jupyter, Wes McKinney, 3rd Edition, O'Reilly Media, 2022.

2. Python Programming-An Object Oriented Approach, Anita Goel, Universities Press

3. Python for Data Science For Dummies, Yuli Vasiliev, 2nd Edition, Wiley, 2022.

Reference Books

1. Python Data Science Handbook: Essential Tools for Working with Data, Jake VanderPlas, 2nd Edition, O'Reilly, 2022.

2. Introduction to Machine Learning with Python, Andreas Müller & Sarah Guido, O'Reilly Media, Reprint Edition, 2023.

3. Foundations for Analytics with Python: From Non-programmer to Hacker, Clinton Brownley, 2nd Edition, Pearson, 2020.

Activities:

Outcome: Demonstrate proficiency in creating and manipulating NumPy arrays for mathematical operations and simulations.

Activity: Create a NumPy-based simulation:

- Generate a 2D array representing temperature data over time
- Apply mathematical operations (mean, std, element-wise addition)
- Simulate random noise and visualize the effect

Evaluation Method: Code-based assessment (10-point scale):

- Correct use of np.array, np.random, and math functions
- Accuracy of simulation logic
- Output clarity and reproducibility

Outcome: Apply Pandas Series and DataFrame operations for structured data handling and analysis.

Activity: Analyze a CSV dataset (e.g., sales or COVID data):

- Load into a DataFrame
- Perform Series operations (filter, map, value_counts)
- Apply DataFrame methods (groupby, sort, describe)

Evaluation Method: 10-point scale checklist and peer review to verify:

- Proper use of Series vs DataFrame
- Logical data manipulation
- Insightful summary statistics

Outcome: Read, write, and clean diverse data formats using Python tools, addressing missing values and outliers.

Activity: Work with multiple formats:

- Read CSV, Excel, and JSON files
- Identify and handle missing values (dropna, fillna)
- Detect and treat outliers using IQR or Z-score

Evaluation Method: Rubric-based evaluation to check (10-point scale):

- File handling accuracy

- Cleaning completeness
- Outlier detection logic

Outcome: Implement vectorized string operations and create derived features for enhanced model readiness.

Activity: Prepare text data for modelling to:

- Use str methods to clean and standardize strings
- Extract features (e.g., domain from email, length of name)
- Encode categorical variables (e.g., get_dummies, LabelEncoder)

Evaluation Method: Feature report to check (10-point scale):

- Efficiency of vectorized operations
- Relevance of derived features
- Readiness for ML input

Outcome: Perform complex data wrangling tasks such as merging datasets, reshaping data structures, generating group-level statistics and Visualization.

Activity: Integrate and reshape datasets:

- Merge two datasets on a common key
- Reshape using pivot, melt, stack, unstack
- Generate group-level stats (e.g., mean sales per region)
- Visualize the Datasets

Evaluation Method: Before-and-after comparison to validate:

- Accuracy of merge and reshape
- Correct use of aggregation
- Final structure suitability for analysis

SEMESTER-IV

COURSE 9: PYTHON FOR DATA ANALYSIS AND VISUALIZATION

Practical | **Credits: 1** | **2 hrs/week**

C9P: Python for Data Analysis and Visualization Lab

List of Practicals:

1. Create and Manipulate NumPy ndarrays; Explore Data Types
2. Perform Arithmetic Operations and Element-wise Calculations on Arrays
3. Practice Indexing, Slicing, Boolean, and Fancy Indexing on ndarrays
4. Use Universal Functions and Compute Basic Mathematical/Statistical Functions with NumPy
5. Create and Manipulate Pandas Series and DataFrames
6. Perform Indexing, Selection, Filtering, and Boolean Indexing in Pandas
7. Conduct Arithmetic Operations and Data Alignment in DataFrames
8. Sort, Rank, Drop Entries and Handle Duplicate Indexes in Pandas
9. Read and Write Data in CSV, TXT, JSON, and Excel Formats
10. Handle Missing Data: Detect, Drop, Fill, and Replace Missing Values
11. Rename Axis Indexes, Remove Duplicates, and Filter Outliers
12. Transform Data Using Mapping Functions and Apply String Operations
13. Perform String Operations and Use Regular Expressions on DataFrames
14. Create Dummy Variables and Perform Permutations and Random Sampling
15. Merge, Join, and Concatenate Datasets Using Pandas
16. Reshape Data Using Pivot, Stack, Unstack, and Perform Hierarchical Indexing
17. Compute Summary Statistics Grouped by Levels or Categories
18. Basic Visualizations using Matplotlib

SEMESTER-IV

COURSE 10: DOCUMENT ORIENTED DATABASE

Theory	Credits: 3	3 hrs/week

Course Objectives

1. To introduce students to the concepts of NoSQL databases and their significance compared to traditional relational databases.
2. To provide hands-on experience with MongoDB for performing CRUD operations, querying, and advanced data handling.
3. To develop skills in schema design, data modeling, and working with embedded and referenced documents.
4. To explore MongoDB's advanced features such as indexing, aggregation framework, replication, and transactions.
5. To prepare students for real-world applications of MongoDB in scalable, high-performance data-driven applications.

Course Outcomes

On successful completion of this course, students will be able to:

1. Differentiate between SQL and NoSQL databases, and explain the architecture and features of MongoDB.
2. Perform CRUD operations and construct queries using MongoDB Query Language (MQL).
3. Apply schema design strategies and use appropriate data modeling techniques for different application scenarios.
4. Utilize advanced features like indexing, aggregation, GridFS, and transactions to optimize data handling.
5. Implement replication concepts and ensure high availability, fault tolerance, and scalability in MongoDB-based applications.

Unit 1. Introduction to NoSQL & Fundamentals of MongoDB:

What is NoSQL DB? History & evolution of NoSQL, Features of NoSQL databases, CAP theorem & BASE properties, Types of NoSQL (Key-Value, Document, Column, Graph), Difference between RDBMS & NoSQL, Why and when to use NoSQL?, NoSQL Database misconceptions, Benefits & real-world use cases of NoSQL, Comparison of popular NoSQL systems (Redis, Cassandra, CouchDB, Neo4j), Introduction to JSON & BSON

Installation & Setup: Installing MongoDB locally and using Atlas (MongoDB's cloud service), connecting via Mongo shell or GUI.

Unit 2. MongoDB Architecture, Data Modeling and Basics:

MongoDB Architecture: Database, Collection, Document concepts, BSON format, Advantages of MongoDB over RDBMS,

MongoDB Datatypes (String, Number, Date, Boolean, Array, ObjectId, Embedded Documents, Null)

Data Modeling in MongoDB: Schema design strategies, Embedded vs Referenced documents

Database & Collection Management: Create & Drop Database, Create & Drop Collection

Unit 3. CRUD Operations and Querying in MongoDB:

CRUD Operations: Insert Documents (insertOne, insertMany), Query Documents (find, operators, conditions), Update Documents (updateOne, updateMany, replaceOne), Delete Documents (deleteOne, deleteMany)

Query operators ($gt, $lt, $in, $nin, $and, $or, $not), Regular expression queries, Bulk operations

Working with Arrays.

Unit 4. Data Modelling and Aggregation:

Data Modelling and Aggregation: Data Models: Introduction to embedded vs normalized models, advantages and trade-offs

Embedded Data Models: Use cases, benefits, and limitations

Normalized Data Models: References between documents, when to normalize data Relationships Between Documents, Data Model Using an Embedded Document, Data Model Using Document References

Aggregation Basics: Introduction to MongoDB Aggregation Framework, simple pipelines and operators

Unit 5. Advanced Query Processing and Optimization in MongoDB:

Query Optimization & Operations: Projection, Limiting & Skipping Records, Sorting Records

Indexing in MongoDB (single field, compound, multikey, text index), Aggregation Framework (pipelines, stages, operators), Replication Concepts: Replica sets, failover, consistency

Textbooks:

1. MongoDB: The Definitive Guide, Shannon Bradshaw, Eoin Brazil, Kristina Chodorow, 3rd Edition, O'Reilly Media, 2019.
2. MongoDB Recipes: With Data Modeling and Query Building Strategies, Subhashini Chellappan, Dharanitharan Ganesan , Apress

Reference Books:

1. MongoDB in Action, Kyle Banker, 2nd Edition, Manning Publications, 2016.
2. MongoDB Applied Design Patterns, Steve Francia, O'Reilly Media, 2013.
3. MongoDB for Developers, Rick Copeland, O'Reilly Media, 2013.

Web Resources:

1. Official MongoDB Documentation: https://www.mongodb.com/docs/
2. MongoDB University Free Courses: https://learn.mongodb.com/
3. W3Schools, Tutorialspoint, geeksforgeeks

Activities:

Outcome: Differentiate between SQL and NoSQL databases, and explain the architecture and features of MongoDB

Activity: Students create a comparative chart (SQL vs NoSQL) and draw MongoDB architecture diagram.

Evaluation Method: Assess based on accuracy, clarity of explanation, and presentation.

Outcome: Perform CRUD operations and construct queries using MongoDB Query Language (MQL)

Activity: Implement CRUD operations and write 5 different queries on a sample dataset (e.g., Library or E-commerce).

Evaluation Method: Marks for correct execution, query correctness, and output validation.

Outcome: Apply schema design strategies and use appropriate data modeling techniques for different application scenarios

Activity: Design schema for a university management system (students, courses, faculty) using MongoDB data modeling techniques.

Evaluation Method: Evaluate on schema correctness, use of embedding/referencing, and justification of design.

Outcome: Utilize advanced features like indexing, aggregation, GridFS, and transactions to optimize data handling

Activity: Create indexes and implement an aggregation pipeline to generate sales report from a dataset.

Evaluation Method: Marks for index implementation, aggregation correctness, and efficiency of results.

Outcome: Implement replication concepts and ensure high availability, fault tolerance, and scalability in MongoDB-based applications

Activity: Configure a replica set with primary and secondary nodes, and demonstrate failover.

Evaluation Method: Assess on setup correctness, successful failover demonstration, and report/documentation.

SEMESTER-IV

COURSE 10: DOCUMENT ORIENTED DATABASE

Practical | **Credits: 1** | **2 hrs/week**

1. Installation and setup of MongoDB, connecting to Mongo Shell and Compass.
2. Creating and using databases, creating collections, inserting documents.
3. Basic queries using find(), filtering with comparison operators.
4. Using logical operators ($and, $or, $not, $nor) for complex queries.
5. Updating documents with $set, $unset, $inc, $rename.
6. Deleting documents using deleteOne() and deleteMany().
7. Using projection to display selective fields.
8. Sorting documents, limiting output, skipping records.
9. Designing an Embedded Data Model for a student-course enrollment system.
10. Designing a Normalized Data Model using document references.
11. Modeling relationships: One-to-One, One-to-Many, Many-to-Many in MongoDB.
12. Implementing schema validation using JSON Schema in MongoDB.
13. Creating and testing single-field and compound indexes.
14. Using text search and multikey indexes.
15. Building aggregation pipelines with $match, $group, $project, $sort.
16. Advanced aggregation operators: $lookup, $unwind, $bucket.
17. Configuring and testing replication with a replica set (minimum 3 nodes).
18. Storing and retrieving large files using GridFS.
19. Using MongoDB Transactions for multi-document consistency.
20. Case Study: Developing a mini-application (e.g., Library Management / E-commerce Cart) using MongoDB CRUD, Aggregation, Indexing, and Replication.