DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

LIST OF COURSES OFFERED FOR MINOR PROGRAM (R23)

Course code	Course Title	Contact hours/week				Credits
		L	T	P	Total	
23CSM1	Fundamentals of data structures	3	0	3	3	3
23CSM2	Principles of Object-oriented programming.	3	0	3	3	3
23CSM3	Operating Systems processes & concepts	3	0	0	3	3
23CSM4	Introduction to Database systems	3	0	3	3	3
23CSM5	Software Engineering Fundamentals	3	0	0	3	3
23CSM6	Introduction to Design and Analysis of Algorithms.	3	0	0	3	3
23CSM7	Fundamentals of data structures Lab	0	0	3	3	1.5
23CSM8	Principles of Object-oriented programming lab	0	0	3	3	1.5
23CSM9	Introduction to Database systems lab	0	0	3	3	1.5

B.Tech-CSE	23CSM1 – FUNDAMENTALS OF DATA STRUCTURES	L	T	P	Cr.
		3	0	0	3

Pre-requisite : Programming Language

Course Educational Objectives:

The objective of the course is to make students familiar with writing algorithms to implement different data structures like stacks, queues, trees and graphs, and various sorting techniques.

Course Outcomes (COs): *At the end of this course, the student will be able to*

CO 1: Write the algorithms for various operations on list using arrays and linked list and analyze the time complexity of its operations.**(Understand - L2)**

CO 2: Apply linear data structures like stack and queue in problem solving.(**Apply - L3**)

CO 3: Demonstrate various searching and sorting techniques and compare their computational complexities in terms of space and time.**(Understand - L2)**

CO 4: Write the algorithms for various operations on binary trees,binary search trees and AVL trees.(**Understand - L2**)

CO 5: Demonstrate graph traversal techniques and hashing techniques.(**Understand - L2**)

UNIT - I

Introduction Abstract Data Type (ADT)

List : List ADT, List using arrays and linked list- Singly Linked List, Doubly Linked List, Circular LinkedList.

UNIT – II

Stacks: Stack ADT, Implementation using arrays and linked list.

Applications of stacks: Infix to postfix expression conversion, Evaluation of Postfix expressions and balancing the symbols.

Queues:

Queue: Queue ADT, Implementation of Queue using arrays and linked list, circular queue

UNIT - III

Sorting: Bubble sort**,** Insertion Sort, Selection sort, Merge Sort, Quick Sort & Heap Sort

Searching: linear & Binary Search

UNIT - IV

Trees: Introduction, Binary Tree Representation using array and Linked list, Tree traversals- In order, pre order and Post order , Binary Search Trees and it's operations – Insert, delete, and search.

UNIT - V

Graphs: Fundamentals, Representation of graphs, Graph Traversals: BFS, DFS.

Hashing: Hash Table, Hash Function, Collision resolution Techniques- separate Chaining, Open addressing, rehashing.

TEXTBOOKS:

1. Mark Allen Weiss, "Data Structures and Algorithm Analysis in C", Pearson Education, 2nd edition,2011
2. Reema Thareja, Data Structures using c, Oxford Publications, 1st Edition,2014.

REFERENCE BOOKS:

1. Langson, Augenstein &Tenenbaum, 'Data Structures using C and C++', PHI, 2nd Edition , 2015.
2. RobertL.Kruse, Leung and Tando, 'Data Structures and Program Design in C', PHI, 2nd edition, 2011.

B.Tech-CSE 23CSM2 -PRINCIPLES OF OBJECT-ORIENTED PROGRAMMING

L	T	P	Cr.
3	0	0	3

Pre-requisite : Programming for Problem Solving using C.

Course Educational Objective: The objective of the course is to learn the constructs of the Java programming language along with built-in facilities to create different applications such as console & graphical user interfaces. In the process of learning the language, they will be applying knowledge of object-oriented programming; they will get the fundamental knowledge reason collection framework.

Course Outcomes: At the end of this course, the student will be able to

CO 1 Demonstrate the fundamentals of object-oriented programming and basic building blocks of Java. **(Understand- L2)**

CO 2 Apply object-oriented programming principles for the development of reusable applications. **(Apply - L3)**

CO 3 Understand the importance of abstraction, user defined package creation and handling different exceptions. **(Understand- L2)**

CO 4 Develop multitasking applications using JAVA multithreaded programming and perform different operations upon various data structures by using collection framework. **(Apply – L3)**

CO 5 Develop GUI applications using AWT (Abstract Window Toolkit). **(Apply- L3)**

UNIT-I

Introduction to OOP: Programming paradigms, procedural programming language versus object-oriented language, principles of OOP.

Introduction to JAVA: Data types, variables, keywords, operators, and control statements.

UNIT-2

Introduction to Classes and Object: Class definition, variables, and methods. Declaring Objects, Constructors, and this keyword.

Classes and objects: overloading methods and constructors, parameter passing, returning objects, recursion. Access control, nested and inner classes, final and static keyword, variable and command-line arguments.

UNIT-3

String handling classes: String, StringBuffer, StringTokenizer.

Inheritance and polymorphism: Inheritance, types of inheritance, super keyword, polymorphism (overloading & overriding), dynamic method dispatch, abstract class, using final with inheritance.

UNIT-4

Interfaces and packages: Interface methods, inheritance in interfaces. API : The built-in JAVA packages and creating and managing user defined packages, importance of CLASSPATH.

Exception Handling: Exception hierarchy, importance of try, catch, throw, throws and finally. Block creation of user-defined exceptions, Assertions.

UNIT-5

Multithreading: Introduction, thread life cycle, creation of threads, naming a thread, joining a thread, thread priorities, daemon thread, thread pool, thread group, thread synchronization, Inter- thread communication.

TEXTBOOKS:

1. Herbert Schildt, "Java: The complete reference", TMH Publications, 7th edition, 2006.
2. Cay S. Horstmann, "Core Java Volume I – Fundamentals", Pearson, Eleventh edition, 2018.

REFERENCE BOOKS:

1. Dr.R.NageswaraRao, "Core JAVA: An Integrated Approach", Dreamtech Press, 1st Edition,2008.
2. E. Balaguruswamy, "Programming with JAVA", TMH Publications, 2ndEdition, 2000.
3. Patrick Niemeyer & Jonathan Knudsen, "Learning Java", O'REILLY Publications, 3rd Edition, 2005.
4. Benjamin J Evans & David Flanagan, "Java–in a Nutshell – A desktop quick reference", O'REILLY Publications, 6th Edition, 2014.

B.Tech-CSE	23CSM3 - OPERATING SYSTEMS PROCESS & CONCEPTS	L	T	P	Cr.
		3	0	0	3

Pre-requisite : Knowledge of Computer fundamentals & Data structures& Algorithms

Course Educational Objective: The objective of the course is to provide basic knowledge of computer operating system structure and functioning, understand how Operating Systems evolved with advent of computer architecture, and comprehend the different CPU scheduling algorithms, page replacement algorithms, disk scheduling and identify best one.

Course Outcomes: At the end of this course, the student will be able to

CO1: Demonstrate the underlying principles and techniques of operating system **(Understand-L2)**

CO2: Interpret scheduling and communication methods of processes handled by operating systems **(Understand-L2)**.

CO3: Distinguish the process synchronization methods and deadlock handling approaches employed in operating systems **(Understand-L2)**.

CO4: Classify memory management techniques and virtual memory mechanisms **(Understand-L2)**.

CO5: Interpret the strategies of disk scheduling algorithms and file system architecture **(Understand-L2)**.

Unit-1:

Operating System Structures: Operating-System Services, User Operating-System Interface, System Calls, Types of System Calls, System Programs, Operating-System Design and Implementation, Operating-System Structure, Virtual Machines, Operating-System Generation, System Boot.

Unit-2:

Processes: Process concept, Inter-process Communication,
Threads: Overview, Multithreading Models
Process Scheduling: Scheduling Criteria, Scheduling Algorithms (FCFS, SJF, PRIORITY, ROUNDROBIN)

Unit-3:

Process Synchronization: The Critical-Section Problem, Peterson's Solution, Synchronization Hardware, Semaphores, Classic Problems of Synchronization, Monitors.

Unit-4:

Deadlocks: System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention. Deadlock Avoidance, Deadlock Detection, Recovery from deadlock.

Memory Management Strategies: Swapping, Contiguous Memory Allocation, Paging, Structure of the Page Table, Segmentation.

.

Unit-5:

Virtual Memory Management: Demand Paging, Page Replacement, Allocation of Frames, Thrashing

Implementing File System: File-System Structure, Allocation Methods, Free-Space Management

TEXTBOOKS:

1. Silberschatz& Galvin, "Operating System Concepts", Wiley, 7th edition, 2007.

REFERENCE BOOKS:

1. William Stallings, "Operating Systems", PHI, 5th Edition, 2004.
2. B.A. Forouzan& R.F. Giberg, "Unix and shell Programming", Thomson, New Delhi, 1st Edition, 2003.
3. http://codex.cs.yale.edu/avi/os-book/OS9/slide-dir/index.html
4. https://swayam.gov.in/nd1_noc19_cs50/preview

B. Tech-CSE	23CSM4 - INTRODUCTION TO DATABASE SYSTEMS	L	T	P	Cr.
		3	0	0	3

Pre-requisite **:** Data Structures

Course Educational Objective: The Objective of this course is to know about basic concepts of DBMS, Database Languages, Database Design, Normalization Process, Transaction Processing, Indexing.

Course Outcomes: At the end of this course, the student will be able to

CO1: State the Basic Components of Database Management System and data modelling using Entity-Relationship Diagrams. **(Understand -L2)**
CO2: Examine the relational model using Structured Query Language (SQL). **(Apply- L3)**
CO3: Employ principles of normalization for effective database design. **(Apply- L3)**
CO4: Demonstrate the necessity of transaction processing, Concurrency control mechanisms and recovery strategies in DBMS. **(Understand- L2)**
CO5: Describe file organization, indexing techniques and the competency in selecting NoSQL Database. **(Understand- L2)**

UNIT – I

Introduction: An overview of Database Management System, Database System Vs File System, Database System Concepts and Three Schema Architecture, Data Models, Database Schema and Instances, Data Independence, Database Languages, Database Structure.

Data Modelling using the Entity Relationship Model: ER model concepts, Notation for ER Diagram, Mapping Constraints, Keys, Concepts of Super Key, Candidate Key, Primary Key, Generalization, Aggregation, Reduction of an ER Diagrams to Tables.

UNIT – II

Relational Data Model and Language: Relational Data Model Concepts, Integrity Constraints: Entity Integrity, Referential Integrity, Key Constraints, Domain Constraints.

Introduction to SQL: Characteristics of SQL, Advantage of SQL. SQL Data types and Literals, Insert, Update and Delete Operations, Tables, Views and Indexes, Nested Queries, Aggregate Functions, Joins, Unions, Intersection, Minus, Cursors in SQL, Triggers in SQL.

UNIT – III

Normalization: Functional Dependencies, Normal Forms - First, Second, Third Normal Forms, BCNF, Inclusion Dependences, Loss Less Join Decompositions, Multi Valued Dependencies, Fourth Normal Form, Join Dependencies and Fifth Normal Form.

UNIT – IV

Transaction Processing Concepts: Transaction System, Testing of Serializability, Serializability of Schedules, Conflict & View Serializability, Recoverability, Deadlock Handling.

Concurrency Control Techniques: Concurrency Control, Locking Techniques for Concurrency Control, Time Stamping Protocols for Concurrency Control, Validation Based Protocol, Multiple Granularity, Recovery with Concurrent Transactions.

UNIT – V

Crash Recovery: Log Based Recovery, Checkpoints.

Physical Database Design: Storage and file structure, indexed files, hashed files, B+ trees, files with dense index; files with variable length records.

.

TEXTBOOKS:

1. Henry F. Korth, Abraham Silberschatz, S.Sudarshan, "Database System Concepts", McGrawHill, 6th edition, 2009.
2. Shashank Tiwari, " ProfessionalNoSql", John Wiely& Sons, 2011.

REFERENCE BOOKS:

1. Raghu Ramakrishnan, JohanneseGehrke, ―Database Management System‖, McGrawHill, 3rd edition, 2000.
2. Date C J, ―An Introduction to Database System, Pearson Education, 8th edition, 2003.
3. RamezElmasri, ShamkanthB.Navathe, "Fundamentals of Database Systems", Addison Wesley, 6th edition, 2010.

B.Tech-CSE **23CSM5 – SOFTWARE ENGINEERING FUNDAMENTALS**

L	T	P	Cr.
3	0	0	3

Pre-requisite : Object Oriented Programming

Course Educational Objective: The objective of the course is to provide understanding of different s/w process models and how to choose one among them by gathering the requirements from a client and specifying them. Using those requirements in the design of the software architecture based on the choices with the help of modules and interfaces. To enable s/w development, by using different testing techniques like unit, integration and functional testing, the quality assurance can be achieved.

Course Outcomes: At the end of this course, the student will be able to

CO 1 Understand the fundamentals of software engineering concepts and software process models. **(Understand-L2)**

CO 2 Apply the requirement elicitation techniques for preparing SRS and design engineering. **(Apply-L3)**

CO 3 Understanding the basic building blocks of UML, Class and object diagrams. **(Understand-L2)**

CO 4 Apply the behavioral models for real world applications. **(Apply-L3)**

CO 5 Demonstrate different software testing approaches for testing the real time applications. **(Understand-L2)**

UNIT – I:

Software and software Engineering: The evolving role of Software, Characteristics of Software, Importance of software Engineering, Changing nature of software, Legacy Software, Software Myths.

UNIT – II:

Software Process and Process Models: Layered technology, Process frame work, The process and Product, software process models, the water fall model, incremental model, the spiral and V Model, Component based s/w development, Unified process model,

UNIT – III:

Requirements Analysis and Software design: Requirements gathering and analysis, software requirements specifications (SRS).
Design Engineering: overview of design process, Design Concepts, Architectural Concepts

UNIT – IV:

Design Using UML: Building Blocks of UML, Defining things, relationships and diagrams, Common Mechanism in UML, Class and Object Diagrams

Behavioral Modeling: Interactions, Interaction diagrams, use cases, Use case Diagrams, Activity Diagrams, Events and signals, state machines, processes and Threads, time and space, state chart diagrams

UNIT – V:

Architectural Modeling: Component, Deployment, Component diagrams and Deployment diagrams.

Testing Techniques: Software testing fundamentals, Unit testing, Integration testing, Black box testing, white box testing, Debugging, System testing.

TEXTBOOKS:

1. Roger S. Pressman, "Software engineering- A practitioner 's Approach", TMH International Edition, 6th edition, 2005.

2. Grady Booch, James Rumbaugh, Ivar Jacobson, "The Unified Modelling Language User Guide", PEARSON ,4th Impression, 2012.

REFERENCE BOOKS:

1. Ugrasen Suman, "Software Engineering - Concepts and practices", Cengage learning,2nd Edition,2011
2. Mahesh P. Matha, "Object-oriented analysis and design using UML", PHI,3nd Edition,2015
3. Rajib Mall , "Fundamentals of Software Engineering", PHI,3rd Edition, 2010
4. https://onlinecourses.nptel.ac.in/noc20_cs68

B.Tech-CSE	23CSM6 – INTRODUCTION TO DESIGN AND ANALYSIS OF ALGORITHMS	L	T	P	Cr.
		3	0	0	3

Pre-requisite : Programming language and Data structures.

Course Educational Objective: The Objective of the course is to learn various algorithm design techniques and analyze the computing resources of the algorithms ,and motivate the students to design new algorithms for various problems

Course Outcomes: At the end of this course, the student will be able to

CO1: Identify the characteristics of an algorithm and analyze its time and space complexity. **(Understand- L2)**
CO2: Apply the divide-and-conquer method for solving problems like searching and sorting. **(Apply- L3)**
CO3: Design Greedy algorithms for the optimization problems like knapsack problem, minimum cost spanning tree, single source shortest path problem. **(Apply - L3)**
CO4: Apply dynamic programming paradigm to solve optimization problems like travelling salesperson problem,0/1 knapsack problem, Optimal binary search tree. **(Apply - L3)**
CO5: Apply the backtracking method on optimization problems like N-queens, sum of subsets, Hamiltonian circuit and graph coloring. **(Apply - L3)**

UNIT – I

Introduction: Algorithm definition, Specifications, Performance Analysis- Time Complexity, Space Complexity. Asymptotic Notations-Big-Oh, Omega, Theta.

UNIT – II

Divide and Conquer: General Method, Binary Search, Finding Maximum and Minimum, Merge Sort, Quick sort, Strassen's matrix multiplication, Closest Pair of Points using Divide and Conquer algorithm

UNIT – III

The Greedy Method – General Method, Knapsack Problem, Job sequencing with deadlines, Minimum-cost spanning trees, Optimal storage on tapes, Single source shortest paths, Huffman coding.

UNIT - IV

Dynamic Programming - General method, Multistage graph, All pairs shortest path, Single Source Shortest path, Optimal Binary search trees, 0/1 Knapsack, Reliability design, the travelling salesman problem.

UNIT-V:

Back tracking - The General Method, The 8-Queens Problem, Sum of subsets, Graph Coloring, Hamiltonian cycles.

TEXTBOOK(S):

1. Ellis Horowitz, SartajSahni,S Rajasekaran , "Fundamentals of Computer Algorithms", University press, 2nd edition, 2012.

REFERENCE BOOKS:

1. Mark Allen Weiss, "Data Structures and Algorithm Analysis in C++", Pearson, 3rd edition , 2007.
2. Aho, Hopcroft & Ullman, "The Design and Analysis of Computer Algorithms", Addison Wesley publications, 2008.
3. Thomas H.Corman et al, "Introduction to Algorithms", PHI, 3rd edition, 2008
4. Anany Levitin, "Introduction to the Design and Analysis of Algorithms", PEA,
5. P. H. Dave, H. B. Dave, "Design and Analysis of Algorithms‖, Pearson Education", 2008.
https://onlinecourses.nptel.ac.in/noc20_cs68

B.Tech-CSE 23CSM7 – FUNDAMENTALS OF DATA STRUCTURES LAB	L	T	P	Cr.
	0	0	3	1.5

Pre-requisite : Programming language and Data structures.

Course Educational Objective: The Objective of the course is to learn various algorithm design techniques and analyze the computing resources of the algorithms, and motivate the students to design new algorithms for various problems

Course Outcomes: At the end of this course, the student will be able to

CO1: Implement Linear Data Structures using array and Linked list. (Apply - L3)

CO2: Implement Various Sorting Techniques. (Apply - L3)

CO3: Implement Non-Linear Data Structure such as Trees. (Apply - L3)

CO4: Improve individual / teamwork skills, communication & report writing skills with ethical values.

List of Experiments:

1. Write a C program to implement list using arrays.
2. Write a C Program to implement various operations on single linked list.
3. Write a C Program to implement various operations on double linked list.
4. Write a C Program to implement stack using arrays.
5. Write a C Program to implement stack using linked list
6. Write a C Program to evaluate postfix expression.
7. Write a C Program to convert infix expression into postfix expression
8. Write a C Program to implement queue using arrays.
9. Write a C Program to implement queue using linked list.
10. Write a C Program to implement various sorting techniques: Bubble sort, selection sort, insertion sort, merge sort, quick sort.
11. Write a C Program to implement searching techniques: linear search and binary search.
12. Write a C Program to implement various operations on Binary tree.

TEXTBOOK(S):

Ellis Horowitz, SartajSahni,S Rajasekaran , “Fundamentals of Computer Algorithms”, University press, 2nd edition, 2012.

B.Tech-CSE 23CSM8 – PRINCIPLES OF OBJECT- ORIENTED PROGRAMMING LAB

L	T	P	Cr.
0	0	3	1.5

Prerequisites: Computer Programming Lab

Course Objectives: The aim of this course is to

- Practice object-oriented programming in the Java programming language
- Implement Classes, Objects, Methods, Inheritance, Exception, Runtime Polymorphism, User defined Exception handling mechanism
- Illustrate inheritance, Exception handling mechanism
- Construct Threads, Event Handling, implement packages

Course Outcomes: After successful completion of the course the students are able to

CO1: Implement basic concepts of the java programming language. (Apply-L3)
CO2: Implement object-oriented programming concepts and exception handling (Apply- L3).
CO3: Design multithreaded applications. (Apply-L3)
CO4: Improve individual / teamwork skills, communication & report writing skills with ethical values.

Experiments covering the Topics:

- Object Oriented Programming fundamentals- data types, control structures
- Classes, methods, objects, Inheritance, polymorphism,
- Exception handling, Threads, Packages, Interfaces

Exercise – 1

a. Write a JAVA program to display default value of all primitive data type of JAVA
b. Write a java program that display the roots of a quadratic equation ax2+bx=0. Calculate the discriminate D and basing on value of D, describe the nature of root.

Exercise - 2

a. Write a JAVA program to search for an element in a given list of elements using binary search mechanism.
b. Write a JAVA program to sort for an element in a given list of elements using bubble sort
c. Write a JAVA program using String Buffer to delete, remove character.

Exercise - 3

a. Write a JAVA program to implement class mechanism. Create a class, methods and invoke them inside main method.
b. Write a JAVA program implements method overloading.
c. Write a JAVA program to implement constructor.
d. Write a JAVA program to implement constructor overloading.

Exercise – 4

a. Write a JAVA program to implement Single Inheritance
b. Write a JAVA program to implement multilevel Inheritance
c. Write a JAVA program for abstract class to find areas of different shape

Exercise - 5

a. Write a JAVA program give example for "super" keyword.
b. Write a JAVA program to implement Interface. What kind of Inheritance can be achieved?
c. Write a JAVA program that implements Runtime polymorphism

Exercise - 6

a. Write a JAVA program that describes exception handling mechanism
b. Write a JAVA program Illustrating Multiple catch clauses
c. Write a JAVA program for creation of Java Built-in Exceptions
d. Write a JAVA program for creation of User Defined Exception

Exercise - 7

a. Write a JAVA program that creates threads by extending Thread class.First thread display "Good Morning "every 1 sec, the second thread displays "Hello "every 2 seconds and the third display "Welcome" every 3 seconds,(Repeat the same by implementing Runnable)
b. Write a program illustrating is Alive and join ()
c. Write a Program illustrating Daemon Threads.
d. Write a JAVA program Producer Consumer Problem

Exercise – 8

Write a JAVA program that import and use the user defined packages.

B. Tech-CSE	23CSM9- INTRODUCTION TO DATABASE SYSTEMS LAB	L	T	P	Cr.
		0	0	3	1.5

Course Objectives: This Course will enable students to

1. Populate and query a database using SQL DDL/DML Commands
2. Declare and enforce integrity constraints on a database
3. Writing Queries using advanced concepts of SQL
4. Programming PL/SQL including procedures, functions, cursors and triggers

Course Outcomes: After successful completion of the course the students are able to

CO1: Implement SQL queries using DDL/DML commands**.(Apply-L3)**
CO2: Apply different Integrity constraints & Normalization techniques for effective database design. **.(Apply-L3)**
CO3: Implement PL/SQL including procedures, functions, cursors and triggers. **.(Apply-L3)**
CO4: Improve individual / teamwork skills, communication & report writing skills with ethical values.

Experiments covering the topics:

- DDL, DML, DCL commands
- Queries, nested queries, built-in functions,
- PL/SQL programming- control structures
- Procedures, Functions, Cursors, Triggers,
- Database connectivity- ODBC/JDBC

Sample Experiments:

1. Creation, altering and droping of tables and inserting rows into a table (use constraints while creating tables) examples using SELECT command.
2. Queries (along with sub Queries) using ANY, ALL, IN, EXISTS, NOTEXISTS, UNION, INTERSET, Constraints. Example:- Select the roll number and name of the student who secured fourth rank in the class.
3. Queries using Aggregate functions (COUNT, SUM, AVG, MAX and MIN), GROUP BY, HAVING and Creation and dropping of Views.
4. Queries using Conversion functions (to_char, to_number and to_date), string functions (Concatenation, lpad, rpad, ltrim, rtrim, lower, upper, initcap, length, substr and instr), date functions (Sysdate, next_day, add_months, last_day, months_between, least, greatest, trunc, round, to_char, to_date)
5.
i. Create a simple PL/SQL program which includes declaration section, executable section and exception –Handling section (Ex. Student marks can be selected from the table and printed for those who secured first class and an exception can be raised if no records were found)
ii. Insert data into student table and use COMMIT, ROLLBACK and SAVEPOINT in PL/SQL block.
6. Develop a program that includes the features NESTED IF, CASE and CASE expression. The program can be extended using the NULLIF and COALESCE functions.

7. Program development using WHILE LOOPS, numeric FOR LOOPS, nested loops using ERROR Handling, BUILT –IN Exceptions, USE defined Exceptions, RAISE- APPLICATION ERROR.
8. Programs development using creation of procedures, passing parameters IN and OUT of PROCEDURES.
9. Program development using creation of stored functions, invoke functions in SQL Statements and write complex functions.
10. Develop programs using features parameters in a CURSOR, FOR UPDATE CURSOR, WHERE CURRENT of clause and CURSOR variables.
11. Develop Programs using BEFORE and AFTER Triggers, Row and Statement Triggers and INSTEAD OF Triggers
12. Create a table and perform the search operation on table using indexing and non- indexing techniques.

Design Database for any one of the following Case Studies

Case Study1: Hospital Management System

Aim: XYZ hospital is a multi-specialty hospital that includes a number of departments, rooms, doctors, nurses, compounders, and other staff working in the hospital. Patients having different kinds of ailments come to the hospital and get checkup done from the concerned doctors. If required they are admitted in the hospital and discharged after treatment. The aim of this cases study is to design and develop a database for the hospital to maintain the records of various departments, rooms, and doctors in the hospital. It also maintains records of the regular patients, patients admitted in the hospital, the checkup of patients done by the doctors, the patients that have been operated, and patients discharged from the hospital.

Description: In hospital, there are many departments like Orthopedic, Pathology, Emergency, Dental, Gynecology, Anesthetics, I.C.U., Blood Bank, Operation Theater, Laboratory, M.R.I., Neurology, Cardiology, Cancer Department, Corpse, etc. There is an OPD where patients come and get a card (that is, entry card of the patient) for check up from the concerned doctor. After making entry in the card, they go to the concerned doctor's room and the doctor checks up the irailments. According to the ailments, the doctor either prescribes medicine or admits the patient in the concerned department. The patient may choose either private or general room according tohis/her need. But before getting admission in the hospital, the patient has to fulfill certain formalities of the hospital like room charges, etc. After the treatment is completed, the doctor is charges the patient. Before discharging from the hospital, the patient again has to complete certain formalities of the hospital like balance charges, test charges, operation charges (if any), blood charges, doctors' charges, etc. Next we talk about the doctors of the hospital. There are two types of the doctors in the hospital, namely, regular doctors and call on doctors. Regular doctors are those doctors who come to the hospital daily. Calls on doctors are those doctors who are called by the hospital if the concerned doctor is not available.

Table Description:

Following are the tables along with constraints used in Hospital Management Database

Constraint: Identity number is unique for each doctor and the corresponding department should exist in DEPARTMENT table.

1. **DEPARTMENT:** This table consists of details about the various departments in the hospital. The information stored in this table includes department name, department location, and facilities available in that department.
Constraint: Department name will be unique for each department.

2. **ALL_DOCTORS:** This table stores information about all the doctors working for the hospital and the departments they are associated with. Each doctor is given an identity number starting with DR or DC prefixes only.

3. **DOC_REG:** This table stores details of regular doctors working in the hospital. Doctors are referred to by their doctor number. This table also stores personal details of doctors like name, qualification, address, phone number, salary, date of joining, etc. Constraint: Doctor's number entered should contain DR only as a prefix and must exist in ALL_DOCTORS table.

4. **DOC_ON_CALL:** This table stores details of doctors called by hospital when additional doctors are required. Doctors are referred to by their doctor number. Other personal details like name, qualification, fees per call, payment due, address, phone number, etc., are also stored.

Constraint: Doctor's number entered should contain DC only as a prefix and must exist in ALL_DOCTORS table.

5. **PAT_ENTRY:** The record in this table is created when any patient arrives in the hospital for a checkup. When patient arrives, a patient number is generated which acts as a primary key. Other details like name, age, sex, address, city, phone number, entry date, name of the doctor referred to, diagnosis, and department name are also stored. After storing the necessary details patient is sent to the doctor for checkup.

Constraint: Patient number should begin with prefix PT. Sex should be M or F only. Doctor's name and department referred must exist.

6. **PAT_CHKUP:** This table stores the details about the patients who get treatment from the doctor referred to. Details like patient number from patient entry table, doctor number, date of checkup, diagnosis, and treatment are stored. One more field status is used to indicate whether patient is admitted, referred for operation or is a regular patient to the hospital. If patient is admitted, further details are stored in PAT_ADMIT table. If patient is referred for operation, the further details are stored in PAT_OPR table and if patient is a regular patient to the hospital, the further details are stored in PAT_REG table.
Constraint: Patient number should exist in PAT_ENTRY table and it should be unique.

7. **PAT_ADMIT:** When patient is admitted, his/her related details are stored in this table. Information stored includes patient number, advance payment, mode of payment, room number, department, date of admission, initial condition, diagnosis, treatment, number of the doctor under whom treatment is done, attend an tname, etc.

Constraint: Patient number should exist in PAT_ENTRY table. Department, doctor number, room number must be valid.

8. PAT_DIS: An entry is made in this table whenever a patient gets discharged from the hospital. Each entry includes details like patient number, treatment given, treatment advice, payment made, mode of payment, date of discharge, etc.
Constraint: Patient number should exist in PAT_ENTRY table.

9. PAT_REG: Details of regular patients are stored in this table. Information stored includes date of visit, diagnosis, treatment, medicine recommended, status of treatment, etc.

Constraint: Patient number should exist in patient entry table. There can be multipleentriesofonepatientaspatientmightbevisitinghospitalrepeatedlyforcheckupandthere will been try for patient's each visit.

10. PAT_OPR: If patient is operated in the hospital, his/her details are stored in this table. Information stored includes patient number, date of admission, date of operation, number of the doctor who conducted the operation, number of the operation theater in which operation was carried out, type of operation, patient's condition before and after operation, treatment advice, etc.

Constraint: Patient number should exist in PAT_ENTRY table. Department, doctor number should exist or should be valid.

11. ROOM_DETAILS: It contains details of all rooms in the hospital. The details stored in this table include room number, room type (general or private), status (whether occupied or not), if occupied, then patient number, patient name, charges per day, etc.

Constraint: Room number should be unique. Room type can only be G or P and status can only be Y or N.

CaseStudy2: Railway Reservation

Aim: The railway reservationsystemfacilitatesthepassengerstoenquireaboutthetrainsavailable on the basis of source and destination, booking and cancellation of tickets, enquire about the status of the booked ticket, etc. The aim of case study is to design and develop a database maintaining the records of different trains, train status, and passengers. The record of train includes its number, name, source, destination, and days on which it is available, where as record of train status includes dates for which tickets can be booked, total number of seats available, and number of seats already booked. The database has been developed and tested on the Oracle.

Description:
Passengers can book their tickets for the train in which seats are available. For this, passenger must provide the desired train number and the date for which ticket is to be booked. Before booking a ticket for a passenger, the validity of train number and booking date is checked. Once the train number and booking date are validated, itis checked whether the seat is available. If yes, the ticket is booked with confirm status and corresponding ticket IDis generated which is stored along with other details of the passenger. After all the available tickets are booked, certain numbers of tickets are booked with waiting status. If waiting lot is also finished, then tickets are not booked and a message of non-availability of seats is displayed. The ticket once booked can be cancelled at any time. For this, the passenger must provide the ticket ID (the unique key). The

ticket ID is searched, and the corresponding record is deleted. With this, the first ticket with waiting status also gets confirmed.

List of Assumption

Since the reservation system is very large in reality, it is not feasible to develop the case study to that extent and prepare documentation at that level. Therefore, a small sample case study has b0een created to demonstrate the working of the reservation system. To implement this sample case study, some assumptions have been made, which are as follows:

1. The number of trains has been restricted to 5.
2. The booking is open only for next seven days from the current date.
3. Only two categories of tickets can be booked, namely, AC and General.
4. The total number of tickets that can be booked in each category AC and General is 10.
5. The total number of tickets that can be given the status of waiting is 2.
6. The in-between stop page stations and their bookings are not considered.

Description of Tables and Procedures:

Tables and procedures that will be created are as follows:

1. **TrainList:** This table consists of details about all the available trains. The information stored in this table includes train number, train name, source, destination, fair for AC ticket, fair for general ticket, and weekdays on which train is available.
Constraint: The train number is unique.
2. **Train_Status:** This table consists of details about the dates on which ticket can be booked for a train and the status of the availability of tickets. The information stored in this table includes train number, train date, total number of AC seats, total number of general seats, number of AC seats booked, and number of general seats booked.
Constraint: Train number should exist in Train List table.

3. **Passenger:** This table consists of details about the booked tickets. The information stored in this table includes ticket ID, train number, date for which ticket is booked, name, age, sex and address of the passenger, status of reservation (either confirmed or waiting), and category for which ticket is booked.
Constraint: Ticket ID is unique and the train number should exist in Train List table.

4. **Booking:** In this procedure, the train number, train date, and category is read from the passenger. On the basis of the values provided by the passenger, corresponding record is retrieved from the Train_ Status table. If the desired category is AC, then total number of AC seats and number of booked AC seats are compared in order to find whether ticket can be booked or not. Similarly, it can be checked for the general category. If ticket can be booked, then passenger details are read and stored in the Passenger table.

5. **Cancel:** In this procedure, ticket ID is read from the passenger and corresponding record is searched in the Passenger table. If the record exists, it is deleted from the table. After deleting the record (if it is confirmed), first record with waiting status for the same train and same category searched from the Passenger table and its status is changed to confirm.

CaseStudy3: Painting Hire Business

System Description

A local business woman has decided to start her own Internet business, called Masterpieces Ltd, Hiring paintings to private individuals and commercial companies. Because of your reputation database designer, she has called upon your services to design and implement a database to support her new business. At the initial planning meeting, to discuss the design, the following user requirements were requested. The system must be able to manage the details of customers, paintings and those paintings currently on hire to customers. Customers are categorized as B(bronze), S (silver), G (gold) orP (platinum). These categories entitle a customer to a discount of0%,5%,10%or15%respectively.

Customers often request paintings by a particular artist or theme (e.g. animal, landscape, seascape, naval, still-life, etc). Over time a customer may hire the same painting more than once.

Each painting is allocated a customer monthly rental price defined by the owner. The owner of the painting is then paid 10% of that customer rental price. Any paintings that are not hired with in six months are returned to the owner. However, after three months, an owner mayre submit a returned painting. Each painting can only have one artist associated with it .Several reports are required from the system. Three main ones are :For each customer, are ports howing an overview of all the paintings they have hired or are currently hiring.
For each artist, are port of all paintings submitted for hiref or each artist, areturnsreportforthosepaintingsnothirdoverthepastsixmonthsremembertoidentifykeyattributes and any foreign key attributes.

Text Books/Suggested Reading:

1. Oracle: The Complete Reference by Oracle Press
2. Nilesh Shah, "Database Systems Using Oracle", PHI, 2007
3. Rick F Vander Lans, "Introduction to SQL", Fourth Edition, Pearson Education, 2007