

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

MCA - II Year - I Sem.

DATA BASE MANAGEMENT SYSTEMS

Objectives:

The objective of the course is to learn the foundation of Database Technologies. In this course, student will learn to design a database, normalize it by eliminating the redundancies, Querying, Improve the database performance by Indexing, Hashing and finally optimize the queries and Transaction Management.

Outcomes: Demonstrate the basic elements of a relational database management system.

- Ability to identify the data models for relevant problems.
- Ability to design entity relationship and convert entity relationship diagrams into RDBMS and formulate SQL queries on the respective data.
- Apply normalization and professional attitudes for the development of application software's.

UNIT I

Database System Applications, Purpose of Database Systems, View of Data – Data Abstraction, Instances and Schemas, Data Models – the ER Model, Relational Model, Other Models – Database Languages – DDL, DML, Database Access from Applications Programs, Transaction Management, Data Storage and Querying, Database Architecture, Database Users and Administrators, History of Data base Systems. Introduction to Data base design, ER diagrams, Beyond ER Design, Entities, Attributes and Entity sets, Relationships and Relationship sets, Additional features of ER Model, Conceptual Design with the ER Model, Conceptual Design for Large enterprises. Relational Model: Introduction to the Relational Model – Integrity Constraints over Relations, Enforcing Integrity constraints, Querying relational data, Logical data base Design, Introduction to Views – Destroying /altering Tables and Views.

UNIT II

Relational Algebra and Calculus: Relational Algebra – Selection and Projection, Set operations, Renaming, Joins, Division, Examples of Algebra Queries, Relational calculus – Tuple relational Calculus – Domain relational calculus – Expressive Power of Algebra and calculus.

Form of Basic SQL Query – Examples of Basic SQL Queries, Introduction to Nested Queries, Correlated Nested Queries, Set – Comparison Operators, Aggregate Operators, NULL values – Comparison using Null values – Logical connectives – AND, OR and NOT – Impact on SQL Constructs, Outer Joins, Disallowing NULL values, Complex Integrity Constraints in SQL Triggers and Active Data bases.

UNIT III

Introduction to Schema Refinement – Problems Caused by redundancy, Decompositions – Problem related to decomposition, Functional Dependencies - Reasoning about FDS, Normal Forms – FIRST, SECOND, THIRD Normal forms – BCNF –Properties of Decompositions- Loss less- join Decomposition, Dependency preserving Decomposition, Schema Refinement in Data base Design – Multi valued Dependencies – FOURTH Normal Form, Join Dependencies, FIFTH Normal form, Inclusion Dependencies.

UNIT IV

Overview of Transaction Management: The ACID Properties, Transactions and Schedules, Concurrent Execution of Transactions – Lock Based Concurrency Control, Deadlocks – Performance of Locking – Transaction Support in SQL.

Concurrency Control: Serializability, and recoverability – Introduction to Lock Management – Lock Conversions, Dealing with Deadlocks, Specialized Locking Techniques – Concurrency Control without Locking.

Crash recovery: Introduction to Crash recovery, Introduction to ARIES, the Log, Other Recovery related Structures, the Write-Ahead Log Protocol, Check pointing, recovering from a System Crash, Media recovery.

UNIT V

Overview of Storage and Indexing: Data on External Storage, File Organization and Indexing – Clustered Indexes, Primary and Secondary Indexes, Index data Structures – Hash Based Indexing, Tree based Indexing, Comparison of File Organizations.

Storing data: Disks and Files -The Memory Hierarchy – Redundant Arrays of Independent Disks.

Tree Structured Indexing: Intuitions for tree Indexes, Indexed Sequential Access Methods (ISAM). B+ Trees: A Dynamic Index Structure, Search, Insert, and Delete.

Hash Based Indexing: Static Hashing, Extendable hashing, Linear Hashing, Extendible Vs. Linear Hashing.

TEXT BOOKS:

1. Data base Management Systems, Raghu Ramakrishnan, Johannes Gehrke, TMH, 3rd Edition, 2003.
2. Data base System Concepts, A.Silberschatz, H.F. Korth, S.Sudarshan, McGraw hill, VI edition, 2006.
3. Fundamentals of Database Systems 5th edition, Ramez Elmasri, Shamkant B.Navathe, Pearson Education, 2008.

REFERENCE BOOKS:

1. Database Management System Oracle SQL and PL/SQL, P.K.Das Gupta, PHI.
2. Database System Concepts, Peter Rob & Carlos Coronel, Cengage Learning, 2008.
3. Database Systems, A Practical approach to Design Implementation and Management Fourth edition, Thomas Connolly, Carolyn Begg, Pearson education.
4. Database-Principles, Programming, and Performance, P.O'Neil, E. O'Neil 2nd ed., ELSEVIER.
5. Fundamentals of Relational Database Management Systems, S.Sumathi, S.Esakkirajan, Springer.
6. Introduction to Database Management, M.L.Gillenson and others, Wiley Student Edition.
7. Database Development and Management, Lee Chao, Auerbach publications, Taylor & Francis Group.
8. Introduction to Database Systems, C.J.Date, Pearson Education.
9. Database Management Systems, G.K.Gupta, TMH.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

MCA - II Year - I Sem.

COMPUTER NETWORKS

Objectives:

- To introduce the fundamentals of various types of computer networks.
- To demonstrate the TCP/IP and OSI models with merits and demerits.
- To explore the various layers of OSI Model.
- To introduce UDP and TCP Models.

Outcomes:

- After this course student could understand and explore the basics of Computer Networks and Various Protocols. She/he will be in a position to understand the World Wide Web concepts.
- After this course he/she will be in a position to administrate a network and flow of information.

UNIT I

Overview of the Internet: Protocol, Layering Scenario, TCP/IP Protocol Suite: The OSI Model, Internet history standards and administration; Comparison of the OSI and TCP/IP reference model
Data Link Layer – design issues, CRC Codes, Elementary Data link Layer protocols, sliding window protocol

UNIT II

Multiple Access Protocols –ALOHA, CSMA, Collision free protocols, Ethernet- Physical Layer, Ethernet Mac Sub layer, data link layer switching & use of bridges, learning bridges, spanning tree bridges, repeaters, hubs, bridges, switches, routers and gateways.

UNIT III

Network Layer: Network Layer Design issues, store and forward packet switching connection less and connection oriented networks-routing algorithms-optimality principle, shortest path, flooding, Distance Vector Routing, Count to Infinity Problem, Hierarchical Routing, Congestion control algorithms, admission control.

UNIT IV

Internetworking: Tunneling, Internetwork Routing, Packet fragmentation, IPv4, Ipv6 Protocol, IP addresses, CIDR, ICMP, ARP, RARP, DHCP.
Transport Layer: Services provided to the upper layers elements of transport protocol-addressing connection establishment, connection release, Crash Recovery.

UNIT V

The Internet Transport Protocols UDP-RPC, Real Time Transport Protocols, The Internet Transport Protocols- Introduction to TCP, The TCP Service Model, The TCP Segment Header, The Connection Establishment, The TCP Connection Release, The TCP Connection Management Modeling, The TCP Sliding Window, The TCP Congestion Control, The future of TCP.
Application Layer-Introduction, providing services, Applications layer paradigms, Client server model, Standard client-server application-HTTP, FTP, electronic mail, TELNET, DNS, SSH.

TEXT BOOKS:

- 1) Computer Networks, Andrew S. Tanenbaum, David J Wetherall, Pearson Education, 5th Edition.
- 2) Computer Networks A Top-Down Approach, Behrouz A Forouzan, Firouz Mosharrarf, TMH.

REFERENCE BOOKS:

1. An Engineering Approach to Computer Networks-S.Keshav, 2nd Edition, Pearson Education.
2. Understanding communications and Networks, 3rd Edition, W.A.Shay, Cengage Learning.
3. Computer Networks, L.L.Peterson and B.S.Davie, 4th edition, ELSEVIER.
4. Computer Networking: A Top-Down Approach Featuring the Internet, James F.Kurose, K.W.Ross, 3rd Edition, Pearson Education.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

MCA - II Year - I Sem.

OPERATING SYSTEMS

Objectives:

- Understand main components of OS and their working.
- To study the operations performed by OS as a resource manager.
- Understand the scheduling policies of OS.
- Understand the different memory management techniques.
- Understand process concurrency and synchronization.
- Understand the concepts of input/output, storage and file management.
- To study different OS and compare their features.

Outcomes:

- Apply optimization techniques for the improvement of system performance.
- Ability to understand the synchronous and asynchronous communication mechanisms in their respective OS.
- Learn about minimization of turnaround time, waiting time and response time and also maximization of throughput with keeping CPU as busy as possible.
- Ability to compare the different OS.

UNIT I

Operating System Introduction: Operating Systems objectives and functions, Computer System Architecture, OS Structure, OS Operations, Evolution of Operating Systems - Simple Batch, Multi programmed, time-shared, Personal Computer, Parallel, Distributed Systems, Real-Time Systems, Special -Purpose Systems, Operating System services, User OS Interface, System Calls, Types of System Calls, System Programs, Operating System Design and Implementation, OS Structure, Virtual Machines.

UNIT II

Process and CPU Scheduling - Process concepts-The Process, Process State, Process Control Block, Threads, Process Scheduling-Scheduling Queues, Schedulers, Context Switch, Preemptive Scheduling, Dispatcher, Scheduling Criteria, Scheduling algorithms, Multiple-Processor Scheduling, Real-Time Scheduling, Thread scheduling, Case studies: Linux, Windows.

Process Coordination – Process Synchronization, The Critical Section Problem, Peterson's solution, Synchronization Hardware, Semaphores, and Classic Problems of Synchronization, Monitors, Case Studies: Linux, Windows.

UNIT III

Memory Management and Virtual Memory - Logical & Physical Address Space, Swapping, Contiguous Allocation, Paging, Structure of Page Table, Segmentation, Segmentation with Paging, Virtual Memory, Demand Paging, Performance of Demanding Paging, Page Replacement Page Replacement Algorithms, Allocation of Frames, Thrashing, Case Studies: Linux, Windows.

UNIT IV

File System Interface - The Concept of a File, Access methods, Directory Structure, File System Mounting, File Sharing, Protection, File System Implementation - File System Structure, File System Implementation, Allocation methods, Free-space Management, Directory Implementation, Efficiency and Performance, Case Studies: Linux, Windows.

Mass Storage Structure – Overview of Mass Storage Structure, Disk Structure, Disk Attachment, Disk Scheduling, Disk Management, Swap space Management

UNIT V

Deadlocks - System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection and Recovery from Deadlock.

Protection – System Protection, Goals of Protection, Principles of Protection, Domain of Protection, Access Matrix, Implementation of Access Matrix, Access Control, Revocation of Access Rights, Capability-Based Systems, Language-Based Protection, Case Studies: Linux, Windows.

TEXT BOOKS:

1. Operating System Principles, Abraham Silberschatz, Peter B. Galvin, Greg Gagne, 8th Edition, Wiley Student Edition.
2. Operating Systems – Internals and Design Principles, W. Stallings, 6th Edition, Pearson.

REFERENCE BOOKS:

1. Modern Operating Systems, Andrew S Tanenbaum, 3rd Edition, PHI.
2. Operating Systems A concept-based Approach, 2nd Edition, D.M.Dhamdhare, TMH.
3. Principles of Operating Systems, B.L.Stuart, Cengage learning, India Edition.
4. Operating Systems, A.S.Godbole, 2nd Edition, TMH.
5. An Introduction to Operating Systems, P.C.P. Bhatt, PHI.
6. Operating Systems, S.Haldar and A.A.Aravind, Pearson Education.
7. Operating Systems, R.Elmasri, A.G.Carrick and D.Levine, Mc Graw Hill.
8. Operating Systems in depth, T.W. Doepfner, Wiley.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

MCA - II Year - I Sem.

OBJECT ORIENTED ANALYSIS AND DESIGN USING UML

Objectives

- Concisely define the following key terms: class, object, state, behavior, object class, class diagram, object diagram, operation, encapsulation, constructor operation, query operation, update operation, scope operation, association, association role, multiplicity, association class, abstract class, concrete class, class-scope attribute, abstract operation, method, polymorphism, overriding, multiple classification, aggregation, and composition.
- Describe the activities in the different phases of the object-oriented development life cycle.
- State the advantages of object-oriented modeling vis-à-vis structured approaches.
- Compare and contrast the object-oriented model with the E-R and EER models.
- Model a real-world application by using a UML class diagram.
- Provide a snapshot of the detailed state of a system at a point in time using a UML (Unified Modeling Language) object diagram.
- Recognize when to use generalization, aggregation, and composition relationships.
- Specify different types of business rules in a class diagram.

Outcomes:

Master student can able to take up the case studies and model it in different views with respect user requirement such as use case, logical, component and deployment and etc, and preparation of document of the project for the unified Library application.

UNIT I

Introduction to UML: Importance of modeling, principles of modeling, object oriented modeling, conceptual model of the UML, Architecture, Software Development Life Cycle.

UNIT II

Basic Structural Modeling: Classes, Relationships, common Mechanisms, diagrams.

Advanced Structural Modeling: Advanced classes, advanced relationships, Interfaces, Types and Roles, Packages.

Class & Object Diagrams: Terms, concepts, modeling techniques for Class & Object Diagrams.

UNIT III

Basic Behavioral Modeling-I: Interactions, Interaction diagrams.

Basic Behavioral Modeling-II: Use cases, Use case Diagrams, Activity Diagrams.

UNIT IV

Advanced Behavioral Modeling: Events and signals, state machines, processes and Threads, time and space, state chart diagrams.

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

UNIT V

Patterns and Frameworks, Artifact Diagrams. Case Study: The Unified Library application

TEXT BOOKS:

1. Grady Booch, James Rumbaugh, Ivar Jacobson : The Unified Modeling Language User Guide, Pearson Education 2nd Edition.
2. Hans-Erik Eriksson, Magnus Penker, Brian Lyons, David Fado: UML 2 Toolkit, WILEY-Dreamtech India Pvt. Ltd.

REFERENCE BOOKS:

1. Meilir Page-Jones: Fundamentals of Object Oriented Design in UML, Pearson Education.

2. Pascal Roques: Modeling Software Systems Using UML2, WILEY-Dreamtech India Pvt. Ltd.
3. Atul Kahate: Object Oriented Analysis & Design, The McGraw-Hill Companies.
4. Mark Priestley: Practical Object-Oriented Design with UML, TMH.
5. Applying UML and Patterns: An introduction to Object – Oriented Analysis and Design and Unified Process, Craig Larman, Pearson Education.
6. Object-Oriented Analysis and Design with the Unified Process By John W. Satzinger, Robert B Jackson and Stephen D Burd, Cengage Learning.
7. UML and C++, R.C.Lee, and W.M.Tepfenhart, PHI.
8. Object Oriented Analysis, Design and Implementation, B.Dathan, S.Ramnath, Universities Press.
9. OODesign with UML and Java, K.Barclay, J.Savage, Elsevier.
10. Learning UML 2.0, Russ Miles and Kim Hamilton, O'Reilly, SPD.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

MCA - II Year - I Sem.

JAVA PROGRAMMING

Objectives

- Understand the concept of OOP as well as the purpose and usage s of inheritance, polymorphism, and encapsulation principles.
- Identify classes, objects, members of a class and the relationships among them needed for a specific problem.
- Develop Java application programs using sound OOP practices(ex. Interfaces and APIs)
- Develop programs using the Java collection APIs as well as Java standard class library.

Outcomes:

- Understanding of OOP concepts and basics of java programming (Console and GUI based).
- The skills to apply OOP and Java programming in problem solving.
- Should have the ability to extend his knowledge of Java programming further on his/her own.

UNIT I

Java Basics - History of Java, Java buzzwords, comments, data types, variables, constants, scope and life time of variables, operators, operator hierarchy, expressions, type conversion and casting, enumerated types, control flow-block scope, conditional statements, loops, break and continue statements, simple java program, arrays, input and output, formatting output, Review of OOP concepts, encapsulation, inheritance, polymorphism, classes, objects, constructors, methods, parameter passing, static fields and methods, access control, this reference, overloading methods and constructors, recursion, garbage collection, building strings, exploring string class, Enumerations, autoboxing and unboxing, Generics.

UNIT II

Inheritance – Inheritance concept, benefits of inheritance, Super classes and Sub classes, Member access rules, Inheritance hierarchies, super uses, preventing inheritance: final classes and methods, casting, polymorphism- dynamic binding, method overriding, abstract classes and methods, the Object class and its methods. **Interfaces** – Interfaces vs. Abstract classes, defining an interface, implementing interfaces, accessing implementations through interface references, extending interface. **Inner classes**– Uses of inner classes, local inner classes, anonymous inner classes, static inner classes, examples. **Packages**-Defining, Creating and Accessing a Package, Understanding CLASSPATH, importing packages.

UNIT III

Data structures creation and manipulation in java – Introduction to Java Collections, Overview of Java Collection frame work, Commonly used Collection classes– ArrayList, LinkedList, HashSet, HashMap, TreeMap, Collection Interfaces – Collection, Set, List, Map, Legacy Collection classes – Vector, Hashtable, Stack, Dictionary(abstract), Enumeration interface, Iteration over Collections – Iterator interface, ListIterator interface. Other Utility classes – String Tokenizer, Formatter, Random, Scanner, Observable, java.util. **Files** – streams- byte streams, character streams, text Input/output, binary input/output, random access file operations, File management using File class, java.io. **Networking** – Introduction, Manipulating URLs, Ex. Client/Server Interaction with Stream Socket Connections, Connectionless Client/Server Interaction with Datagrams, java.net.

UNIT IV

Exception handling – Dealing with errors, benefits of exception handling, the classification of exceptions- exception hierarchy, checked exceptions and unchecked exceptions, usage of try, catch, throw, throws and finally, rethrowing exceptions, exception specification, built in exceptions, creating own exception sub classes.

Guide lines for proper use of exceptions. **Multithreading** - Differences between multiple processes and multiple threads, thread states, creating threads, interrupting threads, thread priorities, synchronizing threads, interthread communication, thread groups, daemon threads.

UNIT V

GUI Programming with Java - The AWT class hierarchy, Introduction to Swing, Swing vs. AWT, MVC architecture, Hierarchy for Swing components, Containers – Top-level containers – JFrame, JApplet, JWindow, JDialog, Light weight containers – JPanel, A simple swing application, Overview of several swing components- JButton, JToggleButton, JCheckBox, JRadioButton, JLabel, JTextField, JTextArea, JList, JComboBox, JMenu, Java's Graphics capabilities – Introduction, Graphics contexts and Graphics objects, color control, Font control, Drawing lines, rectangles and ovals, Drawing arcs, Layout management - Layout manager types – border, grid, flow, box. **Event Handling** - Events, Event sources, Event classes, Event Listeners, Relationship between Event sources and Listeners, Delegation event model, Semantic and Low-level events, Examples: handling a button click, handling mouse and keyboard events, Adapter classes.

Applets – Inheritance hierarchy for applets, differences between applets and applications, life cycle of an applet - Four methods of an applet, Developing applets and testing, passing parameters to applets, applet security issues..

TEXT BOOKS:

1. Java: the complete reference, 8th edition, Herbert Schildt, TMH.
2. Java for Programmers, P.J.Deitel and H.M.Deitel, Pearson education / Java: How to Program P.J.Deitel and H.M.Deitel, 8th edition, PHI.

REFERENCE BOOKS:

1. Java Programming, D.S.Malik, Cengage Learning.
2. Core Java, Volume 1-Fundamentals, eighth edition, Cay S.Horstmann and Gary Cornell, Pearson Education.
3. An introduction to Java programming and object oriented application development, R.A. Johnson-Cengage Learning.
4. Advanced Programming in Java2, K.Somasundaram, Jaico Publishing House.
5. Programming in Java, S.Malhotra and S.Choudhary, Oxford Univ. Press.
6. Object Oriented Programming with Java, R.Buyya, S.T.Selvi, X.Chu, TMH.
7. Object Oriented Programming through Java, P. Radha Krishna, Universities Press.
8. An introduction to programming and OO design using Java, J.Nino, F.A.Hosch, John Wiley & Sons.
9. Java and Object Orientation, an introduction, John Hunt, second edition, Springer.
10. Maurach's Beginning Java2, D.Lowe, J.Murach, A.Stelman, SPD.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

MCA - II Year - I Sem.

DATABASE MANAGEMENT SYSTEMS LAB

Objectives:

This lab enables the students to practice the concepts learnt in the subject Database management systems.

Outcomes:

- Ability to design and implement a database schema for given problem.
- Be capable to Design and build a GUI application.
- Apply the normalization techniques for development of application software to realistic problems.
- Ability to formulate queries using SQL DML/DDDL/DCL commands.

List of Sample Problems:

- 1) Creation, altering and dropping 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) Creation of 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) Program development using creation of package specification, package bodies, private objects, package variables and cursors and calling stored packages.
- 11) Develop programs using features parameters in a CURSOR, FOR UPDATE CURSOR, WHERE CURRENT of clause and CURSOR variables.
- 12) Develop Programs using BEFORE and AFTER Triggers, Row and Statement Triggers and INSTEAD OF Triggers.

Example Problems:

1. Creating tables for various relations (in SQL):

```
CLIENT- MASTER :( CLENTNO, NAME, ADDRESS1, ADDRESS2, CITY, PINCODE, STATE, BALDUE).
PRODUCT_MASTER : (PRODUCTNO, DESCRIPTION, PROFITPERCENT, UNITMEASURE,
QTY_ON_HAND, REORDERLVL,
SELLPRICE, COSTPRICE)
SALESMAN_MASTER: (SALESMANNO, SALESMANNAME, ADDRESS1, ADDRESS2, CITY,
```

PINCODE,
STATE, SLAMT, TGTTOGET, YTDSALES, RESALES)
SALES_ORDER : (OREDRENO, CLIENTNO, ORDERDATE, DELYADDR,
SALESMANNO, DELYTYPE, BILLYN, DELYDATE, ORDERSTATUS)
SALES_ORDER_DETAILS :(ORDERNO, PRODUCTNO, QTYORDERED, QTYDISP, PRODUCTRATE)

- i) Insert data into their respective table.
- ii) Exercise on retrieving records from table.
 - a. Find out the names of all clients.
 - b. Retrieve the entire contents of the client_master table.
 - c. Retrieve the list of names, city and the state of all clients.
 - d. List the various products available from the Product_master table.
 - e. List all the clients who are located in Mumbai.
 - f. Find the names of salesmen who have a salary equal to Rs.3000.
- iii. Exercise on updating records in a table
 - a. Change the city of client No 'C00005' to 'Bangalore'.
 - b. Change the BalDue of Client No 'C00001' to Rs.1000.
 - c. Change the cost price of 'Trousers' to Rs.950.00.
 - d. Change the city of the salesman to Pune.
- iv. Exercise on deleting records in a table
 - a. Delete all sales men from the salesman_master whose salaries are equal to Rs.35000.
 - b. Delete all products from product_master where the quantity on hand is equal to 100.
 - c. Delete from Client_Master where the column state holds the value 'Tamilnadu'.
- v. Exercise on altering the table
 - a. Add a column called 'Telephone' of data type 'number' and size='10' to the Client Master table.
 - b. Change the size of Sell Price column in Product_Master to 10, 2.
- vi. Exercise on deleting the table structure along with the data
 - a. Destroy the table Client_Master along with the data.
- vii. Exercise on renaming the table
 - a. Change the name of the Salesman_Master table to Sman_mast.

2. Using the tables created previously generate the SQL statements for the operations mentioned below.

The tables in user are as follows:

Client_Master
Product_Master
Salesman_Master
Sales_Order
Sales_Order_Details

- i) Perform the following computations on table data:
 - a. List the names of all clients having 'a' as the second letter in their names.
 - b. List the clients who stay in a city whose first letter is 'M'.
 - c. List all clients who stay in 'Bangalore' or 'Mangalore'.
 - d. List all clients whose BalDue is greater than value 10000.
 - e. List all information from the Sales_Order table for orders placed in the month of June.
 - f. List the order information for Client No'C00001' and 'C00002'.
 - g. List products whose selling price is greater than 500 and less than or equal to 750.
 - h. List products whose selling price is more than 500. Calculate a new selling price as, original selling price * .15. Rename the new column in the output of the above query as new_price.
 - i. List the names, city and state of clients who are not in the state of 'Maharashtra'.
 - j. Count the total no of orders.
 - k. Calculate the average price of all the products.
 - l. Determine the maximum and minimum products prices. Rename the output as max_price and min_price respectively.
 - m. Count the no of products having price less than or equal to 500.
 - n. List all the products whose Qty On Hand is less than reorder level.

ii) Exercise on Date Manipulation

- a. List the order number and day on which clients on placed their order.
- b. List the months (in alphabets) and date when the orders must be delivered.
- c. List the Order Date in the format 'DD-Month-YY'. E.g.12-February-02.
- d. List the date, 15 days after today's date.

iii). Exercises on using Having and Group by Clauses:

- a. Print the description and total qty sold for each product.
- b. Find the value of each product sold.
- c. Calculate the average qty sold for each client that has a maximum order value of 15000.00.
- d. Find out the total of all the billed orders for the month of June.

iv). Exercises on Joins and Correlation:

- a) Find out the products, which have been sold to 'Ivan Bay Ross'.
- b) Find out the products and their quantities that will have to be delivered in the current month.
- c) List the product no and description of constantly sold products (i.e. rapidly moving products).
- d) Find the names of clients who have purchased 'Trousers'.
- e) List the products and orders from customers who have ordered less than 5 units of 'Pull Overs'.
- f) Find the products and their quantities for the orders placed by 'Ivan Bay Ross' and 'Mamta Muzumdar'.
- g) Find the products and their quantities for the orders placed by Client No 'C00001' and 'C00002'.

v). Exercise on Sub-queries:

- a. Find the Product No and Description of non_moving products i.e. Products not being sold.
- b. List the customer Name, Address1, Address2, City and Pin Code for the client who has placed order no 'O19001'.
- c. List the client names that have placed orders before the month of May'02.
- d. List if the product 'Lycra Top' has been ordered by any client and print the Client_no, Name to whom it was sold.
- e. List the names of clients who have placed orders worth Rs.10, 000 or more.

3) Creating Views

4) Writing Assertions

5) Writing Triggers

6) Implementing Operations on relations (tables) using PL/SQL

Ex: Write a PL/SQL code block to calculate the area of a circle for a value of radius varying from 5 to 9. Store the radius and the corresponding values of calculated area in an empty table named Areas(radius, area).

7) Creating FORMS.

8) Generating REPORTS.

TEXT BOOKS:

- 1. Introduction to SQL, Mastering the relational DB languages, IV th edition, Rick F. Vander Lans, Pearson ed., 2007.
- 2. Oracle PL/SQL, 3rd edition, Benjamin Rosenzweig, Elena Silvestrova, Pearson ed., 2004.
- 3. Oracle, Forms Developer's Hand Book. Albert Lulushi, Pearson ed.,
- 4. Oracle Database 11g PL/SQL Programming, M. McLaughlin, TMH.
- 5. Oracle Pl/Sql. Programming, IV Edition, Steven Feuerstein. SPD,rp-2007.
- 6. SQL & PL/SQL for Oracle 10g, Black Book, Dr.P.S. Deshpande.
- 7. Database Systems using Oracle: A Simplified Guide to SQL and PL/SQL, Shah, PHI.

Additional Problems:

i) Databases :

Objective: This lab enables the students to practice the concepts learnt in the subject Databases by developing a database for an example company named "Roadway Travels" whose description is as follows. The student is expected to practice the designing, developing and querying a database in the context of example database "Roadway travels". Students are expected to use "Mysql" database.

Roadway Travels

"Roadway Travels" is in business since 1997 with several buses connecting different places in India. Its main office is located in Hyderabad.

The company wants to **computerize its operations** in the following areas:

- Reservations and Ticketing
- Cancellations

Reservations & Cancellation:

Reservations are directly handled by booking office. Reservations can be made 30 days in advance and tickets issued to passenger. One Passenger/person can book many tickets (to his/her family).

Cancellations are also directly handed at the booking office.

In the process of **computerization of Roadway Travels** you have to design and develop a Database which consists the data of Buses, Passengers, Tickets, and Reservation and cancellation details. You should also develop query's using SQL to retrieve the data from the database.

The above process involves many steps like 1. Analyzing the problem and identifying the Entities and Relationships, 2. E-R Model 3. Relational Model 4. Normalization 5. Creating the database 6. Querying. **Students are supposed to work on these steps week wise and finally create a complete "Database System" to Roadway Travels.** Examples are given at every experiment for guidance to students.

Experiment 1: E-R Model

Analyze the problem carefully and come up with the entities in it. Identify what data has to be persisted in the database. This contains the entities, attributes etc.

Identify the primary keys for all the entities. Identify the other keys like candidate keys, partial keys, if any.

Example: **Entities:**

1. BUS
2. Ticket
3. Passenger

Relationships:

1. Reservation
2. Cancellation

PRIMARY KEY ATTRIBUTES:

1. Ticket ID (Ticket Entity)
2. Passport ID (Passenger Entity)
3. Bus_NO (Bus Entity)

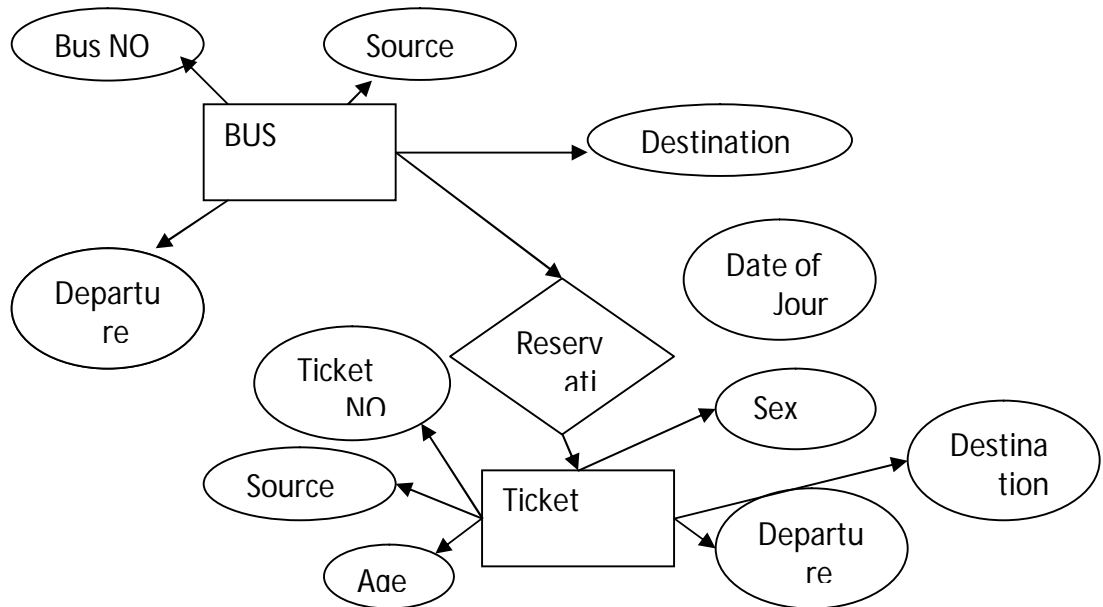
Apart from the above mentioned entities you can identify more. The above mentioned are few.

Note: The student is required to submit a document by writing the Entities and Keys to the lab teacher.

Experiment 2: Concept design with E-R Model

Relate the entities appropriately. Apply cardinalities for each relationship. Identify strong entities and weak entities (if any). Indicate the type of relationships (total / partial). Try to incorporate generalization, aggregation, specialization etc wherever required.

Example: E-R diagram for bus



Note: The student is required to submit a document by drawing the E-R Diagram to the lab teacher.

Experiment 3: Relational Model

Represent all the entities (Strong, Weak) in tabular fashion. Represent relationships in a tabular fashion. There are different ways of representing relationships as tables based on the cardinality. Represent attributes as columns in tables or as tables based on the requirement. Different types of attributes (Composite, Multi-valued, and Derived) have different way of representation.

Example: The passenger tables look as below. This is an example. You can add more attributes based on your E-R model. This is not a normalized table.

Passenger					
Name	Age	Sex	Address	Ticket_id	<u>Passport ID</u>

Note: The student is required to submit a document by Represent relationships in a tabular fashion to the lab teacher.

Experiment 4: Normalization

Database normalization is a technique for designing relational database tables to minimize duplication of information and, in so doing, to safeguard the database against certain types of logical or structural problems, namely data anomalies. For example, when multiple instances of a given piece of information occur in a table, the possibility exists that these instances will not be kept consistent when the data within the table is updated, leading to a loss of data integrity. A table that is sufficiently normalized is less vulnerable to problems of this kind, because its structure reflects the basic assumptions for when multiple instances of the same information should be represented by a single instance only.

For the above table in the First normalization we can remove the multi valued attribute Ticket_id and place it in another table along with the primary key of passenger.

First Normal Form: The above table can be divided into two tables as shown below.

Passenger				
Name	Age	Sex	Address	<u>Passport ID</u>

<u>Passport ID</u>	Ticket_id

You can do the second and third normal forms if required. Anyhow Normalized tables are given at the end.

Experiment 5: Installation of Mysql and practicing DDL commands

Installation of MySQL. In this week you will learn Creating databases, How to create tables, altering the database, dropping tables and databases if not required. You will also try truncate, rename commands etc.

Example for creation of a normalized "Passenger" table.

```
CREATE TABLE Passenger (
  Passport_id INTEGER PRIMARY KEY,
  Name VARCHAR (50) Not NULL,
  Age Integer Not NULL,
  Sex Char,
  Address VARCHAR (50) Not NULL);
```

Similarly create all other tables.

Note: Detailed creation of tables is given at the end.

Experiment 6: Practicing DML commands

DML commands are used to for managing data within schema objects. Some examples:

- SELECT - retrieve data from the a database
- INSERT - insert data into a table
- UPDATE - updates existing data within a table
- DELETE - deletes all records from a table, the space for the records remain

Inserting values into "Bus" table:

Insert into Bus values (1234,'hyderabad', 'tirupathi');
Insert into Bus values (2345,'hyderabd','Banglore');
Insert into Bus values (23,'hyderabd','Kolkata');
Insert into Bus values (45,'Tirupathi','Banglore');
Insert into Bus values (34,'hyderabd','Chennai');

Inserting values into "Passenger" table:

Insert into Passenger values (1, 45,'ramesh', 45,'M','abc123');
Insert into Passenger values (2, 78,'geetha', 36,'F','abc124');
Insert into Passenger values (45, 90,'ram', 30,'M','abc12');
Insert into Passenger values (67, 89,'ravi', 50,'M','abc14');
Insert into Passenger values (56, 22,'seetha', 32,'F','abc55');

Few more Examples of DML commands:

Select * from Bus; (selects all the attributes and display)
UPDATE BUS SET Bus No = 1 WHERE BUS NO=2;

Experiment 7: Querying

In this week you are going to practice queries (along with sub queries) using ANY, ALL, IN, Exists, NOT EXISTS, UNION, INTERSECT, Constraints etc.

Practice the following Queries:

1. Display unique PNR_no of all passengers.
2. Display all the names of male passengers.
3. Display the ticket numbers and names of all the passengers.
4. Find the ticket numbers of the passengers whose name start with 'r' and ends with 'h'.
5. Find the names of passengers whose age is between 30 and 45.
6. Display all the passengers names beginning with 'A'
7. Display the sorted list of passengers names

Experiment 8 and Experiment 9: Querying (continued...)

You are going to practice queries using Aggregate functions (COUNT, SUM, AVG, and MAX and MIN), GROUP BY, HAVING and Creation and dropping of Views.

1. Write a Query to display the Information present in the Passenger and cancellation tables. **Hint:** Use UNION Operator.
2. Display the number of days in a week on which the 9W01 bus is available.
3. Find number of tickets booked for each PNR_no using GROUP BY CLAUSE. **Hint:** Use GROUP BY on PNR_No.
4. Find the distinct PNR numbers that are present.
5. Find the number of tickets booked by a passenger where the number of seats is greater than 1. **Hint:** Use GROUP BY, WHERE and HAVING CLAUSES.
6. Find the total number of cancelled seats.

Experiment 10: Triggers

In this week you are going to work on Triggers. Creation of insert trigger, delete trigger, update trigger. Practice triggers using the above database.

Eg: **CREATE TRIGGER updcheck BEFORE UPDATE ON passenger
FOR EACH ROW
BEGIN
IF NEW.TickentNO > 60 THEN
SET New.Tickent no = Ticket no;
ELSE
SET New.Ticketno = 0;
END IF;**

END;

Experiment 11: Procedures

In this session you are going to learn Creation of stored procedure, Execution of procedure and modification of procedure. Practice procedures using the above database.

```
Eg: CREATE PROCEDURE myProc()  
BEGIN  
SELECT COUNT(Tickets) FROM Ticket WHERE age>=40;  
End;
```

Experiment 12: Cursors

In this week you need to do the following: Declare a cursor that defines a result set. Open the cursor to establish the result set. Fetch the data into local variables as needed from the cursor, one row at a time. Close the cursor when done

```
CREATE PROCEDURE myProc(in_customer_id INT)  
BEGIN  
DECLARE v_id INT;  
DECLARE v_name VARCHAR (30);  
DECLARE c1 CURSOR FOR SELECT stdId,stdFirstname FROM students WHERE  
stdId=in_customer_id;  
OPEN c1;  
FETCH c1 into v_id, v_name;  
Close c1;  
END;
```

Tables

BUS

Bus No: Varchar: PK (public key)
Source : Varchar
Destination : Varchar

Passenger

PPNO: Varchar(15)) : PK
Name: Varchar(15)
Age: int (4)
Sex:Char(10) : Male / Female
Address: VarChar(20)

Passenger_Tickets

PPNO: Varchar(15)) : PK
Ticket_No: Numeric (9)

Reservation

PNR_No: Numeric (9): FK
Journey_date : datetime (8)
No_of_seats : int (8)
Address : Varchar (50)
Contact_No: Numeric (9) --> Should not be less than 9 and Should not accept any other character other than Integer
Status: Char (2): Yes / No

Cancellation

PNR_No: Numeric (9) : FK

Journey_date : datetime(8)

No_of_seats : int (8)

Address : Varchar (50)

Contact_No: Numeric (9) --> Should not be less than 9 and Should not accept any other character other than Integer

Status: Char (2): Yes / No

Ticket

Ticket_No: Numeric (9): PK

Journey_date : datetime(8)

Age : int (4)

Sex:Char(10) : Male / Female

Source: Varchar

Destination: Varchar

Dep_time : Varchar

TEXT BOOKS:

1. Introduction to SQL, Rick F.Vander Lans, Pearson education.
2. Oracle PL/SQL, B.Rosenzweig and E.Silvestrova, Pearson education.
- 3.Oracle PL/SQL Programming,Steven Feuerstein,SPD.
4. SQL & PL/SQL for Oracle 10g, Black Book, Dr.P.S.Deshpande, Dream Tech.
5. Oracle Database 11g PL/SQL Programming, M. Mc Laughlin, TMH.
6. SQL Fundamentals, J.J.Patrick, Pearson Education.

- c) Write a Java program that displays the number of characters, lines and words in a text file.
- d) Write a Java program to change a specific character in a file.

Note: Filename , number of the byte in the file to be changed and the new character are specified on the command line.

16. Write a Java program that illustrates the following
 - a) Creation of simple package.
 - b) Accessing a package.
 - c) Implementing interfaces.
17. a) Write a Java program that creates three threads. First thread displays "Good Morning" every one second, the second thread displays "Hello" every two seconds and the third thread displays "Welcome" every three seconds.
 - b) Write a Java program that correctly implements producer consumer problem using the concept of inter thread communication.
18. Write Java programs that illustrates the following
 - a) Handling predefined exceptions
 - b) Handling user defined exceptions
19. Write a Java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -, *, % operations. Add a text field to display the result.
20. a) Develop an applet in Java that displays a simple message.
 - b) Develop an applet in Java that receives an integer in one text field, and computes its factorial Value and returns it in another text field, when the button named "Compute" is clicked.
21. Write a Java program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a Number Format Exception. If Num2 were Zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.
22. Write a Java program that implements a simple client/server application. The client sends data to a server. The server receives the data, uses it to produce a result, and then sends the result back to the client. The client displays the result on the console. For ex: The data sent from the client is the radius of a circle, and the result produced by the server is the area of the circle. (Use java.net)
23. a) Write a Java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green. When a radio button is selected, the light is turned on, and only one light can be on at a time.No light is on when the program starts.
 - b) Write a Java program that allows the user to draw lines, rectangles and ovals.
24. a) Write a Java program to create an abstract class named Shape that contains an empty method named numberOfSides (). Provide three classes named Trapezoid, Triangle and Hexagon such that each one of the classes extends the class Shape. Each one of the classes contains only the method numberOfSides () that shows the number of sides in the given geometrical figures.
 - b) Suppose that a table named Table.txt is stored in a text file. The firstline in the file is the header, and the remaining lines correspond to rows in the table. The elements are separated by commas. Write a java program to display the table using Jtable component.
25. Write a Java program for handling Key events.
26. Write a Java program for handling mouse events. (Use Adapter classes).
27. Write Java programs that use both recursive and non-recursive functions for implementing the following searching methods:
 - a) Linear search
 - b) Binary search
28. Write Java programs to implement the List ADT using arrays and linked lists.
29. Write Java programs to implement the Stack ADT and Queue ADT using arrays.
30. Write Java programs to implement the following using a singly linked list.
 - a) Stack ADT
 - b) Queue ADT .
31. Write Java programs for implementing the following sorting methods:
 - a) Bubble sort
 - b) Selection sort
 - c) Quick sort
32. Write a Java Program to perform the following:
 - a) Create a binary search tree of elements.
 - b) Search the above binary search tree for a key value.
 - c) Traverse the above binary search tree in inorder.