K.K.Wagh Institute of Engineering Education and Research, Nashik
(Autonomous from Academic Year 2022-23)

T. Y. B. Tech. Computer Engineering
Pattern 2022 Semester: V
COM223003: Database Management System

Teaching Scheme:	**Credit Scheme:**	**Examination Scheme:**
Theory: 03 hrs/week	**03**	**Continuous Comprehensive Evaluation: 20 Marks** **InSem Exam: 20 Marks** **EndSem Exam: 60 Marks**

Prerequisite Courses: - COM222001: Fundamentals of Data Structure
COM222012:Advanced Data Structures

Companion Course:- COM222004: Database Management System Lab

Course Objectives:
- To understand the fundamentals of database management System and database query languages
- To know the principles of database design and transaction management
- To study database system architecture and NOSQL databases

Course Outcomes: On completion of the course, students will be able to

	Course Outcomes	**Bloom's Level**
CO1	Illustrate applications of databases, and features of RDBMS	2-Understand
CO2	Build database queries using SQL, PL/ SQL and NoSQL queries using MongoDB.	3-Apply
CO3	Construct ER diagram to represent logical design of a database	3-Apply
CO4	Apply different normalization techniques to minimize redundancy and anomalies	3-Apply
CO5	Explain various protocols of transaction management and concurrency control in databases	2-Understand

COURSE CONTENTS

Unit I	**Relational Model and SQL**	**(08 hrs)**	**CO1, CO2**

Introduction: Basic concepts, Advantage of DBMS over file processing system, Data Abstraction, Database Language, Structure of DBMS, Data Modeling, database applications.
RDBMS: Basic concepts, Attributes and Domain, Integrity Constraints.
SQL: Introduction to Relational Algebra and Tuple Relational Calculus, Introduction to SQL, SQL Data types and Literals, DDL, DML, DCL, TCL, SQL Select Query and Clauses.
Topic for Self-Study : Codd's Rules

Unit II	**Advanced SQL and PLSQL**	**(06 hrs)**	**CO2**

SQL Advanced Features: Set Operation, Aggregate Function, Null Values, Nested Sub Query, View, Joins, Sequence, Index, Introduction to Embedded and Dynamic SQL.
Introduction to PL/SQL: Data types, Procedures, Functions, Cursor, Trigger, Package, Assertions, Roles and Privileges.
Topic for Self-Study : Oracle Database Architecture

Unit III	**Database Design: Entity- Relationship Model and Relational Database Design**	**(08 hrs)**	**CO3**

Database Design and ER Model: ER Model, Extended E-R Features, converting ER model and EER model to tables, schema diagrams.
Relational Database Design: Functional Dependency, Normalization 1NF, 2NF and 3NF
Topic for Self-Study : BCNF.

Unit IV	**NO SQL Database**	**(08 hrs)**	**CO4**

Database-system Architecture: Centralized and Client-Server Architecture, Server System Architecture, Introduction to Parallel and Distributed databases.
NoSQL Databases: Structured, Unstructured Data and Semi-Structured Data, Comparison of RDBMS and NoSQL, CAP theorem and BASE property.
Types of NoSQL Databases: Key-value store, document store, graph, wide column stores.
Mongo DB: Data types, CRUD operations, Aggregation, Indexing, Sharding.

Unit V	**Transaction Management**	**(06 hrs)**	**CO5**

Transaction: Transaction concept, Transaction state, Transaction Property, Concurrent Executions
Serializability: Conflict serializability, View Serializability, Testing for Serializability, Deadlock prevention, Deadlock Detection and Recovery from deadlock.
Concurrency Control Protocols: Two phase Locking, Timestamp-based protocol.
Recovery: Failure classification, Shadow-Paging and Log-Based Recovery

Text Books

1. Abraham Silberschatz, Henry F. Korth and S. Sudharshan, "Database System Concepts", 6 th Edition Tata McGraw Hill Publishers, ISBN 0-07-120413-X.
2. Kristina Chodorow, "MongoDB: The Definitive Guide", 3rd Edition, Oreilly Publications, ISBN 1491954469

Reference Books

3. C J Date, "An Introduction to Database Systems" ,Addison-Wesly, ISBN:0201144719
4. Pramod J. Sadalage, Martin Fowler, "NoSQL Distilled", Addisen Wesley publication, ISBN:0201144719

Strength of CO-PO PSO Mapping

	PO												PSO	
	1	2	3	4	5	6	7	8	9	10	11	12	1	2
CO1	3	2	2	-	-	-	-	-	-	-	-	3	3	2
CO2	3	2	2	-	2	-	-	-	-	-	-	**2**	2	2
CO3	3	2	3	-	2	-	-	-	-	-	-	2	2	-
CO4	2	3	-	-	-	-	-	-	-	-	-	-	-	-
CO5	2	2	2	-	-	-	-	-	-	-	-	-	-	-
Average	3	2	2.	-	2	-	-	-	-	-	-	2	2.	2.

Guidelines for Continuous Comprehensive Evaluation of Theory Course		
Sr. No.	**Components for Continuous Comprehensive Evaluation**	**Marks Allotted**
1	Quiz on Unit 1, Unit 2, Unit 3, Unit 4 (Quiz 15 marks each and will be converted to 15 Marks)	15
2	Theory assignment on Unit-5 (One Assignment on Unit 5 of 10 marks will be converted to 5 Marks)	5
	Total	**20**

K.K.Wagh Institute of Engineering Education and Research, Nashik
(Autonomous from Academic Year 2022-23)

T. Y. B. Tech. Computer Engineering **Pattern 2022 Semester: V** **COM223004: Database Management System Lab**		
Teaching Scheme:	**Credit Scheme:**	**Examination Scheme:**
Practical: 02 hrs/week	**01**	**Termwork: 25 Marks** **Practical Exam: 25 Marks**
Prerequisite Courses: - COM222007: Data Structures Lab, COM222017: Advanced Data structures Lab		
Companion Course: - COM222003: Database Management System		

Course Objectives:

- To understand the fundamentals of database management System and database query languages
- To know the principles of database design and transaction management
- To study database system architecture and NOSQL databases

Course Outcomes: On completion of the course, students will be able to–

	Course Outcomes	**Bloom's Level**
CO1	Make use of normalized relational database schemas to represent real-world scenarios	3-Apply
CO2	Build simple and complex SQL queries and PL/ SQL code to retrieve, manipulate relational database	3-Apply
CO3	Construct ER diagram to represent logical design of a database	3-Apply
CO4	Build database queries using MongoDB to retrieve, manipulate NoSQL databases	3-Apply
CO5	Develop database-driven applications using programming languages and frameworks that interact with relational database systems or NoSQL databases	3-Apply

List of Laboratory Experiments / Assignments

Sr. No.	**Laboratory Experiments / Assignments**	**CO Mapped**
1	**SQL Queries** Consider the given Database Schema: employee (employee-name, street, city) works (employee-name, company-name, salary) company (company-name, city) manages (employee-name, manager-name) Write SQL queries for the following 1. Find the names of all employees who work for First Bank Corporation. 2. Find the names and cities of residence of all employees who work for First Bank Corporation 3. Find the names, street addresses, and cities of residence of all employees who work for First Bank Corporation and earn more than Rs.10,000. 4. Find all employees in the database who live in the same cities as the companies for which they work.	CO1, CO2

	5. Find all employees in the database who live in the same cities and on the same streets as do their managers. 6. Find all employees in the database who do not work for First Bank Corporation. 7. Find all employees in the database who earn more than each employee of Small Bank Corporation. 8. Assume that the companies may be located in several cities. Find all companies located in every city in which Small Bank Corporation is located. 9. Find all employees who earn more than the average salary of all employees of their company. 10. Find the company that has the most employees. 11. Find the company that has the smallest payroll. 12. Find those companies whose employees earn a higher salary, on average, than the average salary at First Bank Corporation.	
2	**Index, Sequence and View** Consider the given relational table: employee(empno , empname, designation, city, salary, zipcode, county) Write SQL queries for the following 1. Create a sequence used to generate employee numbers for the empno column of the emp table. 2. Create an Index on the county. 3. Find the country whose zipcode = 071 and check whether the query uses the Index and write your observation. 4. Create a view for employees having salary < 50000 and stays in 'Mumbai' 5. Display a Count of employees who stays in 'Mumbai' 6. Find average salary of employees of a created view 7. Display employee names who stays on same street of a view	CO1, CO2
3	**SQL Joins** Consider the given database schema: Student (studentid , studentname,instructorid,studentcity) Instructor(instructorid,Instructorname,instructorcity,specialization) Use all types of Joins 1. Find the instructor of each student. 2. Find the student who is not having any instructor. 3. Find the student who is not having any instructor as well as instructor who is not having student. 4. Find the students whose instructor's specialization is computer. 5. Create a view containing the total number of students whose instructor belongs to "Pune".	CO1, CO2
4	**ER Modelling and Normalization:** Conceptual Design using ER features using tools like ERD plus, ER Win etc. (Identifying entities, relationships between entities, attributes, keys, cardinalities, generalization, specialization etc.) Convert the ER diagram into relational tables and normalize the Relational data model.	CO3
5	**PL/SQL block** Create a database with following schemas Borrower(Rollin, Name, DateofIssue, NameofBook, Status) & Fine(Roll_no,Date,Amt) 1. Write a PL/SQL block to accept input for Borrower table.	CO1, CO2

	2. Write a PL/SQL block using control structures to calculate fine by using the following rules: a. check the number of days (from date of issue), if days are between 15 to 30 then fine amount will be Rs 5 per day b. If no. of days>30, per day fine will be Rs 50 per day c. for days less than 30, Rs. 5 per day. After submitting the book, status will change from I to R. If condition of fine is true, then details will be stored into fine table.	
6	**Cursors** Write a block in PL/SQL to print a report which shows that, the employee id, name, hire date, and the incentive amount they achieved according to their working experiences, who joined in the month of current date. Use explicit cursor	CO1, CO2
7	**Database Trigger** Create a Library database with the schema Books(AccNo, Title, Author, Publisher, Count). a. Create a table Library_Audit with same fields as of Books and Date and status column b. Create a before trigger to insert records into Librry_Audit table if there is deletion in Books table, insert date of deletion and status as deleted Create a after trigger to insert records into Librry_Audit table if there is updation in Books table , insert date of updation and status as updated	CO1, CO2
8	**Database Connectivity**: Write a program to implement Menu driven MySQL/Oracle database connectivity with any front end language for Python/Java/PHP to implement Database navigation operations (add, delete, edit etc.)	CO5
9	**MongoDB Queries** Implement the following MongoDb Query 1. Create a collection named books. 2. Insert 5 records with field TITLE,DESCRIPTION,BY,URL,TAGS AND LIKES 3. Insert 1 more document in collection with additional field of user name and comments. 4. Display all the documents whose title is 'mongodb'. 5. Display all the documents written by 'Ajay' or whose title is 'mongodb'. 6. Display all the documents whose title is 'mongodb' and written by 'Ajay'. 7. Display all the documents whose like is greater than 10. 8. Display all the documents whose like is greater than 100 and whose title is either 'mongodb' or written by 'Ajay'. 9. Update the title of 'mongodb' document to 'mongodb overview' 10. Delete the document titled 'nosql overview'. 11. Display exactly two documents written by 'Ajay'. 12. Display the second document published by 'Ajay'. 13. Display all the books in the sorted fashion. Insert a document using save method.	CO4

10	**MongoDB Aggregation and Indexing** Create the collection Books having the following fields TITLE, DESCRIPTION, BY, URL, TAGS AND LIKES. Implement the following Aggregation and Indexing Queries 1. Find the number of books published by "Ajay" 2. Find books which have minimum likes and maximum likes published by "Ajay". 3. Find the average number of likes of the books published by Ajay. 4. Find the first and last book published by "Ajay".. 5. Create an index on the author name. Display the books published by "Ajay" and check if it uses the index which we have created	CO4
11	**Mini Project:** Form a group of 3 or 4 students and Using the database concepts covered, develop an application with following details: 1. Define a problem statement 2. Follow the Software Development Life cycle and other concepts learnt in Software Engineering Course throughout the implementation. 3. Develop application considering: Front End: Java/Perl/PHP/Python/Ruby/.net/any other language Backend : MongoDB/ MySQL/Oracle 4. Test and validate applications using Manual/Automation testing.	CO1 to 5
Additional Lab Assignments		
1	**ER Modeling** Conceptual Design using ER features using tools like ERD plus, ER Win etc. (Identifying entities, relationships between entities, attributes, keys, cardinalities, generalization, specialization etc.) Convert the ER diagram into relational tables and normalize the Relational data model. ER model of a Hospital management using the following description . Each of these entities have their respective attributes which are – Patients - ID(primary key), name, age,visit_date Tests- Name(primary key), date, result Doctor- ID(primary key), name, specialization	CO3
2	**SQL Queries** Consider the following schema account(acc-no,branch-name,balance) depositor(cust-name,acc-no) borrower (cust-name, loan-no) loan (loan - no, branch - name, amount) Write following queries using SQL 1. Create tables using proper primary keys 2. Update information of particular customer 3. Find the customers having loan less than 1 lac	CO1, CO2

	4. Display account number and customer name starting with 'P' 5. Display name of the depositor with balance 6. Find names of all customers who have a loan at the 'Redwood branch'. 7. Find all customers who have an account and loan or both. 8. Find all customers who do not have loan 9. Find average account balance at each branch. 10. Find the name of borrower having maximum loan amount	
3	**PLSQL Block** Write a Stored Procedure namely proc_Grade for the categorization of students. If marks scored by students in examination is <=1500 and marks>=990 then students will be placed in distinction category if marks scored are between 989 and 900 category is first class, if marks 899 n 825 category is Higher Second Class and Less than 825 and > 600 have 'Pass Class'. Insert the result in Result table for all Write a Stored Procedure for calculating Number of students getting each class e.g Distinction - 10 students, First class -5 students. Insert count in the Analysis table Write a PL/SQLblock to use procedures created with the above requirement. Stud_Marks(roll, name, total_marks) Result(Roll,Name, Class) Analysis(class , count)	CO1, CO2
4	Cassandra Queries: Design and Develop Queries using CRUD operations	CO4

Guidelines for Laboratory Conduction

Use of coding standards and Hungarian notation, proper indentation and comments.
Use of open source software is to be encouraged.
Operating System recommended: - Linux or its derivative
Programming tools recommended: - Open Source line gcc/g++

Guidelines for Student's Lab Journal

The laboratory assignments are to be submitted by students in the form of a journal. Journal consists of Certificate, table of contents, and handwritten write-up of each assignment (Title, problem statement, theory concepts in brief, algorithm, flowchart, test cases and conclusions). Program codes with sample outputs shall be submitted in soft form

Guidelines for Termwork Assessment

Continuous assessment of laboratory work shall be based on overall performance of a student. Assessment of each laboratory assignment shall be based on rubrics that include R1- timely completion (10), R2- understanding of assignment (10) and R3- presentation/clarity of journal writing (10) (Coding standard, Indentation, Hungarian notation, input validation etc)

Strength of CO-PO PSO Mapping

	PO												PSO	
	1	2	3	4	5	6	7	8	9	10	11	12	1	2
CO1	3	2	2	-	-	-	-	-	-	-	-	3	3	2
CO2	2	2	2	-	2	-	-	-	-	-	-	**2**	2	2
CO3	3	2	3	-	2	-	-	-	-	-	-	2	2	-
CO4	2	3	-	-	3	-	-	-	-	-	-	-	-	-
CO5	2	2	2	-	3	-	-	-	2	-	-	-	-	-
Average	2.40	2.20	2.25	-	2.5	-	-	-	2	-	-	2.33	2.33	2.00