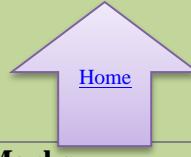


**Savitribai Phule Pune University**  
**Third Year of Computer Engineering (2019 Course)**  
**310246:Database Management Systems Laboratory**


  
Home

<b>Teaching Scheme</b> <b>Practical: 04 Hours/Week</b>	<b>Credit Scheme: 02</b>	<b>Examination Scheme and Marks</b> <b>Term work: 25 Marks</b> <b>Practical: 25 Marks</b>
---	--------------------------	---

**Companion Course:** Database Management Systems (310241)

**Course Objectives:**

- To develop Database programming skills
- To develop basic Database administration skills
- To develop skills to handle NoSQL database
- To learn, understand and execute process of software application development

**Course Outcomes:**

On completion of the course, learners will be able to

- CO1:** Design E-R Model for given requirements and convert the same into database tables  
**CO2:** Design schema in appropriate normal form considering actual requirements  
**CO3:** Implement SQL queries for given requirements , using different SQL concepts  
**CO4:** Implement PL/SQL Code block for given requirements  
**CO5:** Implement NoSQL queries using MongoDB  
**CO6:** Design and develop application considering actual requirements and using database concepts

**Guidelines for Instructor's Manual**

The instructor's manual is to be developed as a reference and hands-on resource. It should include prologue (about University/program/ institute/ department/foreword/ preface), curriculum of the course, conduction and Assessment guidelines, topics under consideration, concept, objectives, outcomes, set of typical applications/assignments/ guidelines, and references.

**Guidelines for Student's Laboratory Journal**

The laboratory assignments are to be submitted by student in the form of journal. Journal consists of Certificate, table of contents, and handwritten write-up of each assignment (Title, Date of Completion, Objectives, Problem Statement, Software and Hardware requirements, Assessment grade/marks and assessor's sign, Theory- Concept in brief, algorithm, flowchart, test cases, Test Data Set(if applicable), mathematical model (if applicable), conclusion/analysis. Program codes with sample output of all performed assignments are to be submitted as softcopy. As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers as part of write-ups and program listing to journal must be avoided. Use of DVD containing students programs maintained by Laboratory In-charge is highly encouraged. For reference one or two journals may be maintained with program prints in the Laboratory.

**Guidelines for Laboratory /Term Work Assessment**

Continuous assessment of laboratory work should be based on overall performance of Laboratory assignments by a student. Each Laboratory assignment assessment will assign grade/marks based on parameters, such as timely completion, performance, innovation, efficient codes, and punctuality.

**Guidelines for Practical Examination**

Problem statements must be decided jointly by the internal examiner and external examiner. During practical assessment, maximum weightage should be given to satisfactory implementation of the problem statement. Relevant questions may be asked at the time of evaluation to test the student's understanding of the fundamentals, effective and efficient implementation. This will encourage, transparent evaluation and fair approach, and hence will not create any uncertainty or doubt in the minds of the students. So, adhering to these principles will consummate our team efforts to the promising start of student's academics.

## **Guidelines for Laboratory Conduction**

The instructor is expected to frame the assignments by understanding the prerequisites, technological aspects, utility and recent trends related to the topic. The assignment framing policy need to address the average students and inclusive of an element to attract and promote the intelligent students. Use of open source software is encouraged. Based on the concepts learned. Instructor may also set one assignment or mini-project that is suitable to respective branch beyond the scope of syllabus.

Operating System recommended :- 64-bit Open source Linux or its derivative

Programming tools recommended: - MYSQL/Oracle, MongoDB, ERD plus, ER Win

### **Virtual Laboratory:**

- <http://vlabs.iitb.ac.in/vlabs-dev/labs/dblab/labs/index.php>

### **Suggested List of Laboratory Experiments/Assignments**

**Assignments from all Groups (A, B, C) are compulsory**

Sr. No.	Group A: SQL and PL/SQL
<b>1. ER Modeling and Normalization:</b> Decide a case study related to real time application in group of 2-3 students and formulate a problem statement for application to be developed. Propose a 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 Relational data model.  Note: Student groups are required to continue same problem statement throughout all the assignments in order to design and develop an application as a part Mini Project. Further assignments will be useful for students to develop a backend for system. To design front end interface students should use the different concepts learnt in the other subjects also.	
<b>2. SQL Queries:</b> <ol style="list-style-type: none"> <li>a. Design and Develop SQL DDL statements which demonstrate the use of SQL objects such as Table, View, Index, Sequence, Synonym, different constraints etc.</li> <li>b. Write at least 10 SQL queries on the suitable database application using SQL DML statements.</li> </ol> Note: Instructor will design the queries which demonstrate the use of concepts like Insert, Select, Update, Delete with operators, functions, and set operator etc.	
<b>3. SQL Queries - all types of Join, Sub-Query and View:</b> Write at least 10 SQL queries for suitable database application using SQL DML statements.  Note: Instructor will design the queries which demonstrate the use of concepts like all types of Join, Sub-Query and View	

<p>4.</p>	<p><b>Unnamed PL/SQL code block: Use of Control structure and Exception handling is mandatory.</b></p> <p>Suggested Problem statement:</p> <p>Consider Tables:</p> <ol style="list-style-type: none"> <li>1. Borrower(Roll_no, Name, DateofIssue, NameofBook, Status)</li> <li>2. Fine(Roll_no, Date, Amt)</li> </ol> <ul style="list-style-type: none"> <li>• Accept Roll_no and NameofBook from user.</li> <li>• Check the number of days (from date of issue).</li> <li>• If days are between 15 to 30 then fine amount will be Rs 5 per day.</li> <li>• If no. of days &gt; 30, per day fine will be Rs 50 per day and for days less than 30, Rs. 5 per day.</li> <li>• After submitting the book, status will change from I to R.</li> <li>• If condition of fine is true, then details will be stored into fine table.</li> <li>• Also handles the exception by named exception handler or user define exception handler.</li> </ul> <p style="text-align: center;"><b>OR</b></p> <p>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, consisting of two columns, radius and area.</p> <p>Note: Instructor will frame the problem statement for writing PL/SQL block in line with above statement.</p>
<p>5.</p>	<p><b>Named PL/SQL Block: PL/SQL Stored Procedure and Stored Function.</b></p> <p>Write a Stored Procedure namely proc_Grade for the categorization of student. If marks scored by students in examination is <math>\leq 1500</math> and <math>\text{marks} \geq 990</math> then student will be placed in distinction category if marks scored are between 989 and 900 category is first class, if marks 899 and 825 category is Higher Second Class.</p> <p>Write a PL/SQL block to use procedure created with above requirement.</p> <p>Stud_Marks(name, total_marks)      Result(Roll,Name, Class)</p> <p>Note: Instructor will frame the problem statement for writing stored procedure and Function in line with above statement.</p>
<p>6.</p>	<p><b>Cursors: (All types: Implicit, Explicit, Cursor FOR Loop, Parameterized Cursor)</b></p> <p>Write a PL/SQL block of code using parameterized Cursor that will merge the data available in the newly created table N_RollCall with the data available in the table O_RollCall. If the data in the first table already exist in the second table then that data should be skipped.</p> <p>Note: Instructor will frame the problem statement for writing PL/SQL block using all types of Cursors in line with above statement.</p>

7.	<p><b>Database Trigger (All Types: Row level and Statement level triggers, Before and After Triggers).</b></p> <p>Write a database trigger on Library table. The System should keep track of the records that are being updated or deleted. The old value of updated or deleted records should be added in Library_Audit table.</p> <p>Note: Instructor will Frame the problem statement for writing PL/SQL block for all types of Triggers in line with above statement.</p>
8.	<p><b>Database Connectivity:</b></p> <p>Write a program to implement MySQL/Oracle database connectivity with any front end language to implement Database navigation operations (add, delete, edit etc.)</p>
<b>Group B: NoSQL Databases</b>	
1.	<p><b>MongoDB Queries:</b></p> <p>Design and Develop MongoDB Queries using CRUD operations. (Use CRUD operations, SAVE method, logical operators etc.).</p>
2.	<p><b>MongoDB - Aggregation and Indexing:</b></p> <p>Design and Develop MongoDB Queries using aggregation and indexing with suitable example using MongoDB.</p>
3.	<p><b>MongoDB - Map reduces operations:</b></p> <p>Implement Map reduces operation with suitable example using MongoDB.</p>
4.	<p><b>Database Connectivity:</b></p> <p>Write a program to implement MongoDB database connectivity with any front end language to implement Database navigation operations (add, delete, edit etc.)</p>
<b>Group C: Mini Project</b>	
1.	<p>Using the <b>database concepts covered in Group A and Group B</b>, develop an application with following details:</p> <ol style="list-style-type: none"> <li>Follow the same problem statement decided in Assignment -1 of Group A.</li> <li>Follow the Software Development Life cycle and other concepts learnt in <b>Software Engineering Course</b> throughout the implementation.</li> <li>Develop application considering: <ul style="list-style-type: none"> <li>Front End : Java/Perl/PHP/Python/Ruby/.net/any other language</li> <li>Backend : MongoDB/MySQL/Oracle</li> </ul> </li> <li>Test and validate application using Manual/Automation testing.</li> <li>Student should develop application in group of 2-3 students and submit the Project Report which will consist of documentation related to different phases of Software Development Life Cycle: <ul style="list-style-type: none"> <li>Title of the Project, Abstract, Introduction</li> <li>Software Requirement Specification</li> <li>Conceptual Design using ER features, Relational Model in appropriate Normalize form</li> <li>Graphical User Interface, Source Code</li> <li>Testing document</li> <li>Conclusion.</li> </ul> </li> </ol> <p><b>Note:</b></p> <ul style="list-style-type: none"> <li>Instructor should maintain progress report of mini project throughout the semester from project group</li> <li>Practical examination will be on assignments given above in Group A and Group B only</li> <li>Mini Project in this course should facilitate the Project Based Learning among students</li> </ul>

**@The CO-PO Mapping Matrix**

PO/CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
<b>CO1</b>	-	1	3	-	3	1	1	1	3	1	-	1
<b>CO2</b>	2	2	3	-	2	-	1	-	3	-	1	-
<b>CO3</b>	-	1	2	-	2	1	-	1	3	-	-	2
<b>CO4</b>	-	1	2	-	2	-	-	-	3	2	1	-
<b>CO5</b>	-	1	2	-	2	-	2	-	3	1	-	1
<b>CO6</b>	2	2	3	-	3	1	-	-	3	-	2	1