

Laboratory Manual

for

Database Management Systems Lab.

(MCA-165)

MCA - I Semester

Compiled by:

Dr. Rakhee

(Assistant Professor, BVICAM, New Delhi)



**Bharati Vidyapeeth's
Institute of Computer Applications
and Management (BVICAM)**

A-4, Paschim Vihar, Rohtak Road, New Delhi-63

Visit us at: www.bvicam.in

Index

List of Abbreviations

Declaration

1. Vision of the Department	5
2. Mission of the Department	5
3. Programme Educational Objectives (PEOs)	5
4. Programme Outcomes (POs)	6
5. Institutional Policy for Students' Conduct	8
6. Learning Outcomes of Laboratory Work	9
7. Course/Lab Outcomes (COs)	10
8. Mapping of COs with POs	10
9. Course/Lab Description	10
10. Grading Policy	11
11. Lesson Plan	11
12. Lab Exercises/Problems	13
13. Advanced Lab exercises/Problems	17
<i>Appendix – A (Index of Lab File)</i>	19

List of Abbreviations

BTL	Bloom's Taxonomy Level
CE	Communication Efficacy
CICP	Conduct Investigations of Complex Computing Problems
CK	Computational Knowledge
CO	Course Outcome
DAC	Departmental Advisory Committee
DDS	Design and Development of Solutions
I&E	Innovation and Entrepreneurship
I&T	Individual & Team Work
IQAC	Internal Quality Assurance Cell
LLL	Life-Long Learning
MTU	Modern Tool Usage
PA	Problem Analysis
PE	Professional Ethics
PEO	Programme Educational Objective
PMF	Project Management and Finance
PO	Programme Outcome
SEC	Societal and Environmental Concern
SED	Stream Editor

Declaration

Department : Department of Computer Science and Applications

Course, Year and the Semester to which Lab is offered : MCA - I Year, I Semester

Name of the Lab Course : Database Management Systems Lab.

Course Code : MCA-165

Version No. :

Name of Course/Lab Teacher : Dr. Rakhee

Laboratory Manual Committee : 1. Dr. Ritika Wason
2. Prof. P. S. Grover
3. Mr. Amit Sharma, Alumni & Industry Expert
4. Dr. Sunil Pratap Singh

Approved by : DAC

Approved by : IQAC

Signature
(Course Teacher)

Signature
(Head of Department)

Signature
(IQAC Coordinator)

1. Vision of the Department

To become a centre of excellence in the field of Computer Science and Applications to produce quality professionals in software development.

2. Mission of the Department

- M1** To produce quality software professionals as per global industry standards.
- M2** To foster innovation, entrepreneurial skills, research capabilities and bring all-round development amongst budding professionals.
- M3** To promote analytical and collaborative life-long learning skills, among students and faculty members.
- M4** To inculcate strong ethical values and professional behaviour while giving equal emphasis to social commitment and nation building.

3. Programme Educational Objectives (PEOs)

The PEO's for the MCA programme are as follows:

- PEO1** Exhibit professional competencies and knowledge for being a successful technocrat.
- PEO2** Adopt creative and innovative practices to solve real-life complex problems.
- PEO3** Be a lifelong learner and contribute effectively to the betterment of the society.
- PEO4** Be effective and inspiring leader for fellow professionals and face the challenges of the rapidly changing multi-dimensional, contemporary world.

4. Programme Objectives (POs)

PO1: Computational Knowledge (CK)

Demonstrate competencies in fundamentals of computing, computing specialisation, mathematics, and domain knowledge suitable for the computing specialisation to the abstraction and conceptualisation of computing models from defined problems and requirements.

PO2: Problem Analysis (PA)

Identify, formulate, and analyze complex real-life problems in order to arrive at computationally viable conclusions using fundamentals of mathematics, computer sciences, management and relevant domain disciplines.

PO3: Design and Development of Solutions (DDS)

Design efficient solutions for complex, real-world problems to design systems, components or processes that meet the specifications with suitable consideration to public health, and safety, cultural, societal, and environmental considerations.

PO4: Conduct Investigations of Complex Computing Problems (CICP)

Ability to research, analyze and investigate complex computing problems through design of experiments, analysis and interpretation of data, and synthesis of the information to arrive at valid conclusions.

PO5: Modern Tool Usage (MTU)

Create, select, adapt and apply appropriate technologies and tools to a wide range of computational activities while understanding their limitations.

PO6: Professional Ethics (PE)

Ability to perform professional practices in an ethical way, keeping in mind cyber regulations & laws, responsibilities, and norms of professional computing practices.

PO7: Life-Long Learning (LLL)

Ability to engage in independent learning for continuous self-development as a computing professional.

PO8: Project Management and Finance (PMF)

Ability to apply knowledge and understanding of the computing and management principles and apply these to one's own work, as a member and leader in a team, to manage projects in multidisciplinary environments.

PO9: Communication Efficacy (CE)

Ability to effectively communicate with the technical community, and with society at large, about complex computing activities by being able to understand and write effective reports, design documentation, make effective presentations, with the capability of giving and taking clear instructions.

PO10: Societal and Environmental Concern (SEC)

Ability to recognize and assess societal, environmental, health, safety, legal, and cultural issues within local and global contexts, and the consequential responsibilities applicable to professional computing practices.

PO11: Individual & Team Work (I&T)

Ability to work in multi-disciplinary team collaboration both as a member and leader as per need.

PO12: Innovation and Entrepreneurship (I&E)

Ability to apply innovation to track a suitable opportunity to create value and wealth for the betterment of the individual and society at large.

5. Institutional Policy for Students' Conduct

The following guidelines shall be followed:-

- 5.1 All the students in their introductory Lab. shall be assigned a system, which shall be their workplace for the complete semester. Students can store records of all their Lab. assignments on their individual workstations.
- 5.2 Introductory Lab. shall include an introduction to the appropriate software/tool, followed by a basic Introductory Assignment having Practice Questions. All the students are expected to complete this assignment within a week time, as the same shall be assessed through a lab. test.
- 5.3 Each week, the instructor, in parallel to respective topics covered in the theory lecture, shall assign a set of practical problems to the students. The problems in these assignments shall be divided into two parts. The first set of problems shall be compulsory for all the students and its record need to be maintained in the Practical File, having prescribed format, as given in Appendix - A. All the students should get the Practical File checked and signed, weekly, by the respective teacher. The second set of problems is Advanced Problems and shall be optional. Student may solve these advanced problems for their further practice.
- 5.4 Cellular phones, pagers, CD players, radios and similar devices are prohibited in the classrooms, laboratories and examination halls.
- 5.5 Laptops, Tablets may be used in lectures/labs for the purpose of taking notes or working on team-projects.
- 5.6 The internal practical exam shall be conducted towards the end of the semester and shall include the complete set of lab exercises conducted as per syllabus. However, students shall be assessed on continuous basis through overall performances in regular lab. tests, both announced and surprise and viva-voce.
- 5.7 The respective faculty shall prepare and submit sufficient number of practical sets of computing problems to the Dean (Examinations), atleast two

weeks prior to the actual exam. It is the responsibility of the faculty to ensure that a set should not be repeated for more than 5 students in a given batch.

5.8 The internal practical exam shall be of 3 hours duration where the student shall be expected to implement solutions to his/her assigned set of problems on appropriate software tools in the lab.

5.9 Once implemented, student shall also appropriately document code implemented in the assigned answer sheets, which shall be submitted at the end of the examination. All the students shall also appear for viva-voce examination during the exam.

5.10 Co-operate, collaborate and explore for the best individual learning outcomes but copying or entering into the act of plagiarism is strictly prohibited.

6. Learning Outcomes of Laboratory Work

The student shall demonstrate the ability to:

- Verify and Implement the concepts and theory learnt in class.
- Code and use Software Tools to solve problems and present their optimal solutions.
- Apply numerical/statistical formulas for solving problems/questions.
- Develop and apply critical thinking skills.
- Design and present Lab as well as project reports.
- Apply appropriate methods for the analysis of raw data.
- Perform logical troubleshooting as and when required.
- Work effectively as a member of a team in varying roles as need be.
- Communicate effectively, both oral and written.
- Cultivate ethics, social empathy, creativity and entrepreneurial mindset.

7. Course/Lab Outcomes (COs)

CO1	Translate an information model into a relational database schema and to implement the schema using RDBMS. (BTL2)
CO2	Apply advanced SQL features like views, indexes, synonyms, etc. for database management. (BTL3)
CO3	Analyze PL/SQL structures like functions, procedures, cursors and triggers for database applications. (BTL4)
CO4	Examine database administration concepts like GRANT, REVOKE etc. through SQL commands. (BTL4)
CO5	Work in teams to design solutions for real-world problems/case studies by creating efficient database schema. (BTL6)
CO6	Translate an information model into a relational database schema and to implement the schema using RDBMS. (BTL2)

8. Mapping of CO's with PO's

Table 1: Mapping of CO's with PO's

PO/CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓		✓					✓		
CO2	✓	✓	✓			✓				✓		
CO3	✓	✓	✓			✓				✓		
CO4	✓	✓	✓				✓			✓		
CO5	✓	✓	✓	✓		✓				✓		
CO6	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

9. Course/Lab Description

Course (Lab) Title : Database Management Systems Lab.

Course (Lab) Code : MCA-165

Credits	: 01
Pre-requisites	: Basic Understanding of Databases, ER diagrams and various architectures
Academic Session	: July to December
Contact Hours/Week	: 02 (01 Labs of 02 Hours/Week)
Internal Assessment	: 40 Marks
External Assessment	: 60 Marks

10. Grading Policy

Item	Points	Marks	Remarks
Weekly Lab Exercises including Practical Files	10	10	Closed Book/Open Book
Internal End-Term Practical Examination	20	10	Closed Book
Viva-Voce	20	20	Closed Book
External End-Term Examinations	60	60	Closed Book (conducted and evaluated by the University)
Total		100	

11. Lesson Plan

Week No.	Lab No.	Topics/Concepts to be Covered
1.	1.	To familiarize with DDL and DML commands.
2.	2.	To access desired records from tables, using where clauses with different operators and using nested queries
3.	3.	To work with joins for joining multiple tables
4.	4.	To apply and use primary key and foreign key concepts.
5.	5.	To familiarize with built-in functions including numeric, character and date functions.
6.	6.	To familiarize with the structure of PL/SQL programs and learn basic concepts.

Week No.	Lab No.	Topics/Concepts to be Covered
7.	7.	To develop programs based on PL/SQL control structures.
8.	8.	To use implicit and explicit cursors for accessing multiple records from tables.
9.	9.	To work with exception handling in PL/SQL programs.
10.	10.	To use the features of procedures and functions.
11.	11.	To develop triggers for some events such as insert, delete or update.
12.	12.	To learn, use and differentiate before and after triggers.

12. Lab Exercises/Problems

P1	<p>Consider the following table:</p> <p style="text-align: center;">Table : Book</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Field Name</th> <th>Data type</th> <th>Size</th> <th>Constraint</th> </tr> </thead> <tbody> <tr> <td>Book_Code</td> <td>Varchar2</td> <td>5</td> <td>Primary key</td> </tr> <tr> <td>ISBN No.</td> <td>Varchar2</td> <td>8</td> <td>Not Null/</td> </tr> <tr> <td>Book_Name</td> <td>Varchar2</td> <td>8</td> <td>Unique</td> </tr> <tr> <td>Publisher</td> <td>Varchar2</td> <td>6</td> <td></td> </tr> <tr> <td>Price</td> <td>Number</td> <td>5,2</td> <td>>=100</td> </tr> <tr> <td>Author_Name</td> <td>Varchar2</td> <td>8</td> <td></td> </tr> <tr> <td>Date_of_Launch</td> <td>Date</td> <td></td> <td></td> </tr> </tbody> </table> <p>Write queries to perform the following:</p> <ol style="list-style-type: none"> 1. Insert few records in it. 2. Increase the size of the Book_Name field to 10 characters. 3. Display the total price of all the books of "TMH" publishing house. 4. Display the details of all the books of author "Yashwant Kanetkar" and "Robert Lafore". 5. Delete all the books belonging to "PHI" publisher. 6. Update the price of books by 5% which belong to 'BPB' publishers. 7. Create a new table "Author" with author details such as author id (primary key), Name of the author, subject, contact details, research area etc. 8. Insert records in it and display the records of those authors who have 'Web mining' as their research area. 9. Add a new column "Author_Id" in "Book" table and make it a foreign key on "author id" of "Author" table. 10. Update all the previous records of Book table to add information of author id (newly added column) 11. Display the records of the books where book name starts with 'C' or 'G' or 'K'. 12. Create a view for the user to access the Book Code, Name and Price of books which are published by 'Pearson Education'. 	Field Name	Data type	Size	Constraint	Book_Code	Varchar2	5	Primary key	ISBN No.	Varchar2	8	Not Null/	Book_Name	Varchar2	8	Unique	Publisher	Varchar2	6		Price	Number	5,2	>=100	Author_Name	Varchar2	8		Date_of_Launch	Date						
Field Name	Data type	Size	Constraint																																		
Book_Code	Varchar2	5	Primary key																																		
ISBN No.	Varchar2	8	Not Null/																																		
Book_Name	Varchar2	8	Unique																																		
Publisher	Varchar2	6																																			
Price	Number	5,2	>=100																																		
Author_Name	Varchar2	8																																			
Date_of_Launch	Date																																				
P2	<p>Consider the following table:</p> <p style="text-align: center;">Table : Student</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Field Name</th> <th>Data type</th> <th>Size</th> <th>Constraint</th> </tr> </thead> <tbody> <tr> <td>Roll_no</td> <td>Number</td> <td>2</td> <td>Primary Key</td> </tr> <tr> <td>Name</td> <td>Varchar2</td> <td>20</td> <td>Not Null</td> </tr> <tr> <td>Address</td> <td>Varchar2</td> <td>50</td> <td></td> </tr> <tr> <td>City</td> <td>Varchar2</td> <td>30</td> <td></td> </tr> <tr> <td>Marks_Sub1</td> <td>Number</td> <td>5,2</td> <td>>=0 and <=100</td> </tr> <tr> <td>Marks_sub2</td> <td>Number</td> <td>5,2</td> <td>>=0 and <=100</td> </tr> <tr> <td>Marks_sub3</td> <td>Number</td> <td>5,2</td> <td>>=0 and <=100</td> </tr> <tr> <td>Total</td> <td>Number</td> <td>5,2</td> <td>>=0 and <=300</td> </tr> </tbody> </table> <ol style="list-style-type: none"> 1. Insert few records except total marks. 2. Create a view on the table to access roll no, name and total marks of those students who belong to 'Delhi'. 3. Calculate total marks of each student and save it. 4. Display the name of the student who has got the highest marks. 5. Display the records of those students who have got marks more than the marks of 'Kiran'. 6. Display the percentage of all the students with their names. 	Field Name	Data type	Size	Constraint	Roll_no	Number	2	Primary Key	Name	Varchar2	20	Not Null	Address	Varchar2	50		City	Varchar2	30		Marks_Sub1	Number	5,2	>=0 and <=100	Marks_sub2	Number	5,2	>=0 and <=100	Marks_sub3	Number	5,2	>=0 and <=100	Total	Number	5,2	>=0 and <=300
Field Name	Data type	Size	Constraint																																		
Roll_no	Number	2	Primary Key																																		
Name	Varchar2	20	Not Null																																		
Address	Varchar2	50																																			
City	Varchar2	30																																			
Marks_Sub1	Number	5,2	>=0 and <=100																																		
Marks_sub2	Number	5,2	>=0 and <=100																																		
Marks_sub3	Number	5,2	>=0 and <=100																																		
Total	Number	5,2	>=0 and <=300																																		

	<ol style="list-style-type: none"> 7. Display the names of those students who have got maximum marks in sub1 and more than 50 in sub2 and sub3. 8. Update the marks of all those students who are fail, with grace marks 5. (fail means less than 40) 9. Delete the record of student who got minimum marks. 10. Display record of student who got second highest marks. 11. Display the average total marks of all the student's city wise excluding the students of 'Gurgaon'. 12. Get the names of students with Roll_no less than 30 and whose total marks is more than the total marks of at least one student with Roll_no greater than or equal to 30.
P3	<p>Consider the following three relations:</p> <p>Primary Keys are indicated by # Foreign Keys are underlined.</p> <p>Employee (Eid#, EName, Address, DateOfJoining, Department) Project (Pid#, PName, StartDate, TerminationDate) AssignedTo (Eid, Pid)</p> <p>Now write queries for the following in relational algebra as well as in SQL:</p> <ol style="list-style-type: none"> 1. Find the employees working on 'Banking' project. 2. Find the projects assigned to the employees of D01 or D02. 3. Find the employees who belong to Delhi and work on either 'University' project or 'ShareMarket' project 4. Find the employees who do not work on 'University' project. 5. Find the employees who work on all projects. 6. List the employees who have not been assigned any project. 7. Find the employees who joined the department after the commencement of 'Bank' project. 8. Display the start and termination date of projects which are allotted to 'Jai Prakash'.
P4	<p>Consider the following relations:</p> <p>Primary Keys are indicated by # Foreign Keys are underlined.</p> <p>Student (Roll#, SName, City, Age, Gender) Course (Cid#, CName, Semester, Credits, Fee) Enrollment (Roll, Cid, DateOfReg)</p> <p>Now write SQL queries for the following:</p> <ol style="list-style-type: none"> 1. Find the details of students registered in MCA course. 2. Display the total number of students enrolled in MBA. 3. Display the names of students who are enrolled in the course in which "Anita" is registered. 4. Display the names of all the students with their date of enrollment in the courses. 5. Display the name of the course in which "Kunal" is registered. 6. Display the names of boys-students enrolled in MCA. 7. Display the number of students registered in MCA course who live outside Delhi.
P5	<p>Consider any table of your choice and execute following functions on numeric, character and date type fields :</p> <p>• sqrt() • power() • ceil() • max() • min() • avg() • count() • exp() • mod() () • sum () • floor() • abs() • greatest () • round() • to_date()</p>

	<ul style="list-style-type: none"> • months_between() • last_day() • add_months() • next_day() • sysdate • round() • to_char() length() • substr() • lower() • initcap() • instr() • concat() • lpad() • rpad() • ltrim() • rtrim() • replace()
P6	Write a PL/SQL code that will reverse a given number like 12345 to 54321.
P7	Write a PL/SQL code to accept emp_id from user and display his / her record. Table: Employee (EmpId, EName, DOJ, DOB, Salary, Address)
P8	Write a PL/SQL code that calculates factorial of first five even numbers and stores the same in the given table. Table : Factorial (No, Fact)
P9	The bank manager of Delhi branch decides to mark the status of all those accounts as 'Inactive' on which there are no transactions performed in last 6 months. Whenever any such update takes place the corresponding record is inserted in another table 'InactiveAccounts' with the name and account no of the account holder and his balance. Table: Account (AccNo, Name, Address, PANNo, Mobile, Status)
P10	Write a Cursor(PL/SQL code) to display the Employee_name, DateofBirth, and Designation whose basic salary is greater than 15000, if not found then show the proper error message. (Use Exception handing) Table : Employee(EmpId, EName, DOJ, DOB, Salary, Address)
P11	Create a function that accepts emp id from the calling procedure and returns his salary. Table: Employee (EmpId, EName, DOJ, DOB, Salary, Address) Create a function that accepts emp id from the calling procedure and returns his salary. Table: Employee (EmpId, EName, DOJ, DOB, Salary, Address)
P12	Write a PL/SQL function ODDEVEN to return value TRUE if the number passed to it is EVEN else will return FALSE

13. Advanced Lab Exercises/Problems

<p>AQ1</p> <p>A</p>	<p>Consider the following relations: Primary Keys are indicated by # Foreign Keys are underlined. Student (Roll#, SName, City, Age, Gender) Course (Cid#, CName, Semester, Credits, Fee) Enrollment (Roll, Cid, DateOfReg)</p> <p>Now write SQL queries for the following:</p> <ol style="list-style-type: none"> 1. Find the details of students registered in MCA course. 2. Display the total number of students enrolled in MBA. 3. Display the names of students who are enrolled in the course in which "Anita" is registered. 4. Display the names of all the students with their date of enrollment in the courses. 5. Display the name of the course in which "Kunal" is registered. 6. Display the names of boys-students enrolled in MCA. 7. Display the number of students registered in MCA course who live outside Delhi.
<p>AQ2</p>	<p>Write a PL/SQL code that accepts the train no, source and destination from the user, finds the records of the train in the given table. Display the record if record found otherwise display an appropriate message. Table : Train (TrainNo, Name, Source, Destination, NoOfStops, NoOfCoaches)</p>
<p>AQ3</p>	<p>Create a function that accepts book id from the calling procedure, finds if it is a 'TMH' publication. If 'yes' then return its price.</p>
<p>AQ4</p>	<p>Given Student Report table, in which student marks assessment is recorded. In such schema, create a trigger so that the total and average of specified marks is automatically inserted whenever a record is insert. Table : Student (RollNo, SName, DOB, Sub1, Sub2, Sub3, Total, Average, Grade)</p>
<p>AQ5</p>	<p>To create a row level trigger for the CUSTOMERS table that would fire for INSERT or UPDATE or DELETE operations performed on the CUSTOMERS table. The trigger should display the difference in amount paid between the old values and new values. Table: Customer (CustId, Cname, Address, OrderNo, AmountPaid, DateofPayment)</p>

Appendix – A: Index of Lab File

Week No.	Lab. Ex. No.	Detailed Description of the Lab Exercise	Outcome Mapping		Page No./Link of Online Document	Signature of Teacher with Date
			CO	BTL		
1.						
2.						
3.						
4.						
5.						
6.						
7.						
8.						
9.						
10.						
11.						
12.						

Note: The students should use header and footer, mentioning their roll number & name in header and page number in footer.