| **Core Practical 3**<br>For the students admitted from A.Y. 2023-2024 & onwards | | |
|---|---|---|
| Offering Department: **Computer Application** | Offered to: **Master of Computer Application** | |
| **Semester - I** | | |
| Course Code | Course Title | Course Credit and Hours |
| **23MCACC107** | **Core Practical 2 :** Databases for Enterprises Applications | **2 Credits - 4 hrs/wk** |

| **Course Description:**<br>This course offers an overview of database management systems, covering both relational databases using Oracle and NoSQL databases using MongoDB. Students will learn the basics of table creation, data manipulation, and SQL query language. The course will also introduce document-oriented data modelling, querying, and data management using MongoDB. Students will gain practical experience through hands-on exercises and projects, allowing them to design and manage databases using both technologies and perform data analysis on large data sets. Upon completion, students will have a solid understanding of both relational and NoSQL databases. |
|---|
| **Course Purpose:**<br>The purpose of this course is to provide students with a comprehensive understanding of database management systems, with a focus on both Oracle and MongoDB. Students will learn to design, create, and manage both relational and NoSQL databases, using SQL and MongoDB query language. By the end of the course, students will be proficient in designing database schema, manipulating data, and querying large datasets using both technologies. The course aims to equip students with practical skills and knowledge necessary to pursue a career in database administration, data analysis, or related fields. |

**Course Outcomes:** Upon completion of this course, the learners will be able to

| **CO No.** | **CO Statement** | **Bloom's Taxonomy Level ($K_1$ to $K_6$)** |
|---|---|---|
| $CO_1$ | Understand the fundamental principles of database management systems, including the differences between relational and non-relational databases. | K1,K2 |
| $CO_2$ | Demonstrate proficiency in creating and managing Oracle databases, including configuring database parameters, managing tablespaces, and monitoring database performance. | K2,K3 |
| $CO_3$ | Learn how to use MongoDB to store and retrieve data, including understanding the differences between document-oriented databases and relational databases. | K2 |
| $CO_4$ | Demonstrate proficiency in creating and managing MongoDB databases, including configuring data models, indexing data for efficient retrieval, and managing data replication. | K2,K3 |
| $CO_5$ | Analyze and troubleshoot common database issues, including query optimization, data backup and recovery, and performance tuning. | K4 |

| **Course Content** | **Hours** |
|---|---|
| **Exercise-I: SQL Statements DDL, DML** | **12 hrs** |
| ***Create a database schema for a fictional online bookstore that includes tables for customers, orders, and books.***<br>1. Add columns to an existing table using ALTER TABLE statement.<br>2. Rename a table using RENAME statement.<br>3. Drop a table from the database using DROP statement.<br>4. Create a backup copy of a table using the CREATE TABLE AS statement. | |

| **Course Content** | **Hours** |
|---|---|
| ***Categorize the main database objects:***<br>1. Identify the main database objects in a sample database schema.<br>2. Categorize the objects by their functionality and purpose.<br>3. Explain the relationships between the different objects in the schema. | |
| ***Review the table structure:***<br>1. Examine the structure of a table in a sample database schema.<br>2. Identify the columns, data types, and constraints used in the table.<br>3. Evaluate the effectiveness of the table structure for storing and retrieving data. | |
| ***List the data types available for columns:***<br>1. Create a list of data types available for columns in Oracle and MongoDB databases.<br>2. Compare and contrast the different data types in terms of their functionality and usage. | |
| ***Create a simple table:***<br>1. Create a table to store employee information, including columns for name, age, job title, and salary.<br>2. Specify appropriate data types and constraints for each column.<br>3. Insert some sample data into the table. | |
| ***Decipher how constraints can be created at table creation:***<br>1. Create a table with various constraints such as NOT NULL, PRIMARY KEY, UNIQUE, CHECK, and FOREIGN KEY constraints.<br>2. Demonstrate how each constraint can be added to the table at creation time. | |
| ***Describe how schema objects work:***<br>1. Create a schema for a sample database and explain how it is used to organize database objects.<br>2. Identify and describe the different types of schema objects, including tables, views, sequences, and indexes.<br>3. Demonstrate how schema objects can be manipulated using SQL commands. | |
| ***Data Manipulation Statements:***<br>1. Describe the different types of DML statements, including INSERT, UPDATE, DELETE, and MERGE statements.<br>2. Create a table with sample data and use DML statements to insert, update, and delete rows from the table.<br>3. Demonstrate how changes made using DML statements can be saved or discarded using COMMIT and ROLLBACK statements. | |
| **Exercise-II: DCL, TCL** | **12 hrs** |
| ***Retrieve Data using the SQL SELECT Statement:***<br>1. Write a SELECT statement to retrieve all data from a specific table in a database.<br>2. Use the WHERE clause to filter the data retrieved by specifying a condition.<br>3. Use the ORDER BY clause to sort the retrieved data in ascending or descending order. | |
| ***List the capabilities of SQL SELECT statements:***<br>1. Create a list of capabilities of SQL SELECT statements, including filtering, sorting, joining, and aggregating data.<br>2. Identify examples of each capability and demonstrate how it can be used in a SELECT statement. | |
| ***Generate a report of data from the output of a basic SELECT statement:***<br>1. Write a SELECT statement to retrieve data from a specific table.<br>2. Use the formatting options in SQL to generate a report of the data retrieved.<br>3. Include column headings, formatting options for data types, and other formatting elements to make the report more readable. | |
| ***Select All Columns:***<br>1. Write a SELECT statement that retrieves all columns from a specific table in a database.<br>2. Compare the results of the SELECT statement to a SELECT statement that retrieves only a subset of columns.<br>3. Discuss the advantages and disadvantages of selecting all columns versus selecting only | |

| Course Content | Hours |
|---|---|
| specific columns. | |
| ***Select Specific Columns:***<br>1. Write a SELECT statement that retrieves only specific columns from a specific table in a database.<br>2. Use column aliases to rename the columns in the output.<br>3. Discuss the advantages of selecting specific columns versus selecting all columns. | |
| ***Use Column Heading Defaults:***<br>1. Write a SELECT statement that uses the default column headings in the output.<br>2. Compare the results of the SELECT statement to a SELECT statement that uses custom column headings.<br>3. Discuss the advantages and disadvantages of using default column headings versus custom column headings. | |
| ***Use Arithmetic Operators:***<br>1. Write a SELECT statement that uses arithmetic operators to perform calculations on columns in the output.<br>2. Include examples of arithmetic operators such as addition, subtraction, multiplication, and division.<br>3. Discuss the advantages of using arithmetic operators in a SELECT statement. | |
| ***Understand Operator Precedence:***<br>1. Write a SELECT statement that uses multiple operators in a single expression.<br>2. Explain the rules of operator precedence in SQL and how they affect the outcome of the expression.<br>3. Use parentheses to control the order of operations in an expression.<br>4. Learn the DESCRIBE command to display the table structure:<br>5. Use the DESCRIBE command to display the structure of a specific table in a database.<br>6. Identify the different columns in the output and their corresponding data types and constraints.<br>7. Discuss the advantages of using the DESCRIBE command to examine the structure of a table. | |
| ***Restricting and Sorting Data:***<br>1. Write a SELECT statement that includes a WHERE clause to restrict the output retrieved from a specific table in a database.<br>2. Use comparison and logical operators in the WHERE clause to specify a condition.<br>3. Use the ORDER BY clause to sort the output in ascending or descending order. | |
| **Exercise-III: Joining, Grouping and Sub queries - 1** | **12 hrs** |
| ***Aggregate Data Using the Group Functions:***<br>1. Write a SELECT statement that uses an aggregate function such as SUM, AVG, MIN, or MAX to produce a report on a specific table in a database.<br>2. Use the GROUP BY clause to group the data by one or more columns.<br>3. Use the HAVING clause to exclude groups of data that do not meet a specific condition. | |
| ***Divide the retrieved data in groups by using the GROUP BY clause:***<br>1. Write a SELECT statement that uses the GROUP BY clause to group the data retrieved from a specific table in a database.<br>2. Group the data by one or more columns to create meaningful reports.<br>3. Use aggregate functions to summarize the data within each group. | |
| ***Exclude groups of data by using the HAVING clause:***<br>1. Write a SELECT statement that uses the HAVING clause to exclude groups of data that do not meet a specific condition.<br>2. Use aggregate functions to create the condition for excluding data.<br>3. Compare the results of the SELECT statement with and without the HAVING clause to understand its impact. | |
| ***Display Data from Multiple Tables Using Joins:***<br>1. Write SELECT statements to access data from two or more tables in a database. | |

| **Course Content** | **Hours** |
|---|---|
| 2. Use the JOIN keyword to join the tables based on a common column.<br>3. Use inner joins, left joins, right joins, or full outer joins to view data that may or may not meet the join condition. | |
| ***View data that generally does not meet a join condition by using outer joins:***<br>1. Write a SELECT statement that uses outer joins to retrieve data from two or more tables in a database.<br>2. Use the LEFT JOIN or RIGHT JOIN keywords to include data from one table even if it does not meet the join condition.<br>3. Use the FULL OUTER JOIN keyword to include data from both tables even if it does not meet the join condition. | |
| ***Join a table to itself by using a self-join:***<br>1. Write a SELECT statement that joins a table to itself based on a common column.<br>2. Use the alias keyword to differentiate between the two instances of the table.<br>3. Use inner joins or outer joins to view data from the self-joined table. | |
| **Exercise-IV: Joining, Grouping and Sub queries - 2** | **12 hrs** |
| ***Use Sub-queries to Solve Queries:***<br>1. Write a sub-query to solve a problem that cannot be solved with a simple SELECT statement.<br>2. Define a sub-query that retrieves data from a single table or multiple tables.<br>3. List the types of sub-queries, including single-row sub-queries, multiple-row sub-queries, and correlated sub-queries. | |
| ***Write single-row and multiple-row sub-queries:***<br>1. Write a single-row sub-query that retrieves data from a specific table in a database.<br>2. Write a multiple-row sub-query that retrieves data from two or more tables in a database.<br>3. Use the IN operator with a sub-query to retrieve data that matches one of several possible values. | |
| ***Multiple-Column Sub queries:***<br>1. Write a multiple-column sub-query that retrieves data from multiple columns in a table.<br>2. Use a multiple-column sub-query to solve a problem that cannot be solved with a single-row sub-query.<br>3. Combine multiple sub-queries using the AND or OR operator to retrieve data that meets specific criteria. | |
| ***Pair wise and Non-pair wise Comparison:***<br>1. Write a sub-query that uses a pair-wise comparison to retrieve data from a specific table in a database.<br>2. Write a sub-query that uses a non-pair wise comparison to retrieve data from two or more tables in a database.<br>3. Compare the results of the two types of sub-queries to understand their differences. | |
| ***Scalar Sub query Expressions:***<br>1. Write a sub-query that returns a single value, such as a count or a sum.<br>2. Use a scalar sub-query to solve a problem that cannot be solved with a simple SELECT statement.<br>3. Combine multiple scalar sub-queries to create a complex expression that retrieves data from a database. | |
| ***Solve problems with Correlated Sub queries:***<br>1. Write a correlated sub-query that retrieves data from multiple tables in a database.<br>2. Use a correlated sub-query to solve a problem that cannot be solved with a non-correlated sub-query.<br>3. Compare the performance of correlated and non-correlated sub-queries to understand their impact on query execution time. | |
| ***Update and Delete Rows Using Correlated Sub queries:***<br>4. Write a correlated sub-query that updates data in a specific table in a database.<br>5. Write a correlated sub-query that deletes data from a specific table in a database. | |

| Course Content | Hours |
|---|---|
| 6. Use a correlated sub-query to solve a problem that cannot be solved with a simple UPDATE or DELETE statement. | |
| ***The EXISTS and NOT EXISTS operators:*** <br> 1. Write a sub-query that uses the EXISTS operator to retrieve data from a specific table in a database. <br> 2. Write a sub-query that uses the NOT EXISTS operator to retrieve data that does not match a specific condition. <br> 3. Compare the results of the two operators to understand their differences. | |
| ***Invoke the WITH clause:*** <br> 1. Write a sub-query that uses the WITH clause to create a temporary table. <br> 2. Use the temporary table to solve a problem that cannot be solved with a simple SELECT statement. <br> **3.** Compare the performance of the WITH clause to other methods for creating temporary tables. | |
| **Exercise V- Introduction to NoSQL & MongoDB** | **12 hrs** |
| ***Exercise for Creating Documents in MongoDB:*** <br> 1. Write a program to insert a single document using the insertOne() method. <br> 2. Write a program to insert multiple documents using the insertMany() method. <br> 3. Create a collection of products with the following fields: name, category, price, quantity. Insert at least 5 documents into the collection using insertOne() and insertMany() methods. <br> 4. Write a program to handle duplicate documents in the collection. <br> ***Exercise for Querying Documents in MongoDB:*** <br> 1. Write a program to retrieve all documents from the products collection using the find() method. <br> 2. Write a program to retrieve documents with price greater than 100 using query operators. <br> 3. Write a program to retrieve documents sorted by price in descending order and limited to 5 results. <br> 4. Write a program to retrieve documents with only name and price fields. <br> ***Exercise for Updating Documents in MongoDB:*** <br> 1. Write a program to update the quantity field of a specific document using updateOne() method. <br> 2. Write a program to update the price field of all documents with quantity greater than 10 using updateMany() method. <br> 3. Write a program to add a new field 'description' to a specific document using updateOne() method. <br> 4. Write a program to upsert a document with a specific _id. <br> ***Exercise for Deleting Documents in MongoDB:*** <br> 1. Write a program to delete a specific document using deleteOne() method. <br> 2. Write a program to delete all documents with price less than 50 using deleteMany() method. <br> 3. Write a program to delete all documents from the collection. <br> 4. Write a program to handle concurrency in MongoDB CRUD operations. <br> ***Exercise for Querying MongoDB:*** <br> 1. Write a program to retrieve documents with a specific filter, sorted by a specific field and limited to a specific number of results. <br> 2. Write a program to group documents by a specific field and calculate the sum of a specific field in each group using the aggregation pipeline. <br> 3. Write a program to find documents near a specific location using geospatial queries and indexes. | |

**Text books:**

- Json Price, Oracle Database 12c SQL, Master SQL, Oracle Press “Oracle Database SQL Language Reference 12c” Release 1

**Reference books:**

- Database Systems Concepts, design and Applications 2/e Singh, S. K., PearsonEducation, New Delhi, 2011
- An introduction to Database Systems, C J Date, Addition-Wesley.
- Silberschatz, Korth, “Data base System Concepts”., McGraw hill, 2008.
- Raghu Ramakrishnan and Johannes Gehrke, Database Management Systems (3/e), McGraw Hill, 2003.
- Sommerville, “Software Engineering”, 8th Edition, Pearson Education
- Peter Rob and Carlos Coronel, Database System- Design, Implementation and Management (7/e), Cengage Learning, 2007.

**Pedagogic tools:**

- Chalk and Board
- Power point presentation
- Seminar
- Videos

**Methods of Assessment & Tools:**

Components of CIA: 40 marks

| **Sr. No.** | **Component** | **Content** | **Duration (if any)** | **Marks** | **Sub Total** |
|---|---|---|---|---|---|
| **A** | Practical Skill Assessment | All Experiments | 3 hours | 30 (Set for 30) | 30 |
| **B** | Observation and Book of Records | All Experiments | - | 10 (10 marks) | 10 |
| | | | | **Grand Total** | **40** |

| | |
|---|---|
| **Lab Methods:** | • Hands-on exercises<br>• Programming assignments<br>• Demonstrations<br>• Code reviews |
| **Assessment Tools:** | • Self-assessments<br>• Peer assessments<br>• Code documentation |